UNIVERSITY OF TOURS

DOCTORAL SCHOOL MIPTIS

BDTLN and RFAI Teams
LIFAT (EA 6300)


# THESIS presented by :


Praveen SONI


Presenting on : **13th December 2024**

to obtain the degree of: **Doctor of the University of Tours**

Discipline: **Computer Science**

# Automatic generation of dashboard

**THESIS directed by :**

| | |
|---|---|
| **Pr. Fatma Bouali** | Co-Director, University of Lille |
| **Dr. Cyril De Runz** | Co-Director, University of Tours |
| **Pr. Gilles Venturini** | Director, University of Tours |

**REVIEWERS:**

| | |
|---|---|
| **Pr. Hanène Azzag** | Sorbonne Paris Nord University |
| **Pr. Arnaud Sallaberry** | University of Montpellier |

---

**JURY :**

| | |
|---|---|
| **Pr. Fatma Bouali** | Co-Director, University of Lille |
| **Dr. Cyril De Runz** | Co-Director, University of Tours |
| **Pr. Gilles Venturini** | Director, University of Tours |
| **Pr. Hanène Azzag** | Reviewer, Sorbonne Paris Nord University |
| **Pr. Arnaud Sallaberry** | Reviewer, University of Montpellier |
| **Pr. Guy Mélançon** | Examiner, President of jury, University of Bordeaux |

# Preface

The motivation for this thesis stems from the increasing importance of data in decision-making across diverse fields, such as data visualization, healthcare, education, and public administration. As data generation continues to grow at an unprecedented rate, the ability to transform this raw information into actionable insights has become crucial. Dashboards, as visual tools for data interpretation, play a vital role in this process, enabling users to monitor trends, explore patterns, and make informed decisions. However, the creation of effective dashboards presents a significant challenge, particularly for novice users who may lack expertise in data visualization.

My journey toward this research began with a key question: How can we simplify the process of dashboard creation so that it becomes accessible to everyone, regardless of their technical skills? This question arose from the observation that, despite the availability of numerous dashboard creation tools, many require substantial manual input and a deep understanding of visualization principles. These requirements can be daunting for individuals without specialized training, creating barriers to effective data use. Recognizing this gap, I sought to explore how automation could make dashboard creation more intuitive and user-friendly.

Throughout the research process, I explored a wide range of data visualization tools and methodologies, both from academic literature and industry practices. This exploration highlighted the complexities involved in creating dashboards that are not only visually appealing but also functionally effective. Existing tools often focus on the incremental selection of individual visualizations, which can hinder the creation of a cohesive dashboard that tells a clear, data-driven story. This realization reinforced the need for a system that could automate the process while ensuring that the final dashboard is informative and well-structured.

This thesis presents the SimpleVis, an Automatic Dashboard Recommendation System (ADRS) designed to address these challenges. SimpleVis automates the key aspects of dashboard creation, including the selection of visualizations and the layout of the dashboard, reducing the burden on users. By considering the overall structure and coherence of the dashboard, SimpleVis ensures that the final product is both effective and easy to understand, even for those with limited experience in data visualization.

I would like to extend my deepest gratitude to my academic advisors, whose guidance and expertise were instrumental in shaping this research. Their support and insights were invaluable at every stage of the thesis. I also thank my colleagues and peers for their constructive feedback and discussions, which helped refine my ideas. I am also grateful to the users who participated in the evaluations of the SimpleVis. Their feedback was crucial in testing and improving the system, ensuring it meets the needs of real-world users. Lastly, I express my heartfelt appreciation

to my family and friends for their unwavering support throughout this journey.

It is my hope that this thesis contributes to the field of data visualization by making dashboard creation more accessible. By empowering users of all backgrounds to harness the power of data, I aim to facilitate better decision-making and outcomes across various domains.

# Acknowledgement

Completing this thesis has been a journey filled with challenges, learning, and growth, and I am deeply grateful to all those who have supported and guided me along the way. First and foremost, I would like to express my sincere gratitude to my academic advisors, Pr. Gilles Venturini, Pr. Fatma Bouali, and Dr. Cyril De Runz, whose wisdom, encouragement, and guidance have been instrumental in the successful completion of this research. Your insightful feedback and unwavering support have pushed me to refine my ideas and develop this work to its fullest potential. I am deeply appreciative of the time and effort you invested not only in my academic journey but also journey of life. My learning from you extends beyond my professional life to help me navigate real-world challenges as well.

I would also like to extend my thanks to the members of my thesis committee, Pr. Fatma Bouali, Dr. Cyril De Runz, Pr. Gilles Venturini, Pr. Hanène Azzag, Pr. Arnaud Sallaberry, and Pr. Guy Mélançon, for their valuable suggestions and constructive criticism. Your expertise and input were crucial in shaping the direction of this thesis and ensuring its academic rigor.

A special thanks to my colleagues and peers in the LIFAT lab at the University of Tours. Your camaraderie, support, and insightful discussions have not only enriched this research but also made this journey more enjoyable. I would also like to thank two students, Mr. Ayush Baliyan and Miss Daniela Pantoja Meneses, who contributed to student projects as part of my thesis. I am particularly grateful to specific colleagues Mr. Bhola Kumar and Mr. Ashish Sharma for the many hours of brainstorming and collaboration that significantly contributed to the technical part of the thesis.

I am also indebted to the participants of the user evaluations, whose honest feedback and engagement were critical in testing and refining the SimpleVis during the comparative study with PowerBI. Your contributions were invaluable in ensuring that the system meets the needs of real-world users. Great thanks to Pr. Sandeep Singh Kang and Dr. Suresh Kaswan for support in finding the potential participants for system evaluations.

I would like to acknowledge the financial support provided by ANR as part of the BI4PEOPLE project, without which this research would not have been possible. The resources and opportunities you provided were crucial in bringing this project to fruition.

On a personal note, I am profoundly grateful to my family and friends. To my parents, Mr. Mahesh and Mrs. Suman Soni, and my sisters, Mrs. Priyanka, Mrs. Rakhi, and Mrs. Neha, thank you for your unwavering love, encouragement, and belief in my abilities. Your support has been the foundation upon which not only my PhD thesis but the entire journey of my life has been built. To my friends, thank you for your patience, and understanding, and for being a constant source

of positivity and strength. I would also like to thank my friends Poonam, Yatin, Nilayam, and Akash for their unconditional love and support.

Lastly, I would like to acknowledge the countless researchers and professionals whose work laid the groundwork for this thesis. The literature and tools developed in the field of data visualization provided the essential foundation upon which this research was built.

This thesis is a culmination of collective efforts, and I am profoundly thankful to everyone who has been a part of this journey. Your contributions, both large and small, have made this achievement possible. Thank you.

# Publications

| No. | Cited by | Year | Publications |
|---|---|---|---|
| 1 | 9 | 2024 | Soni, Praveen, Cyril de Runz, Fatma Bouali, and Gilles Venturini. "A survey on automatic dashboard recommendation systems." Visual Informatics 8, no. 1 (2024): 67-79. |
| 2 | 0 | 2024 | Soni, Praveen, Cyril de Runz, Fatma Bouali, and Gilles Venturini. "Un outil de génération automatique de tableaux de bord." Atelier «La place des usagères et usagers dans les outils de fouille et d'exploration de données»(PAUL@ EGC 2024) (2023). |
| 3 | 1 | 2023 | Soni, Praveen, Cyril De Runz, Fatma Bouali, and Gilles Venturini. "A genetic algorithm for automatic dashboard generation: first results." In 2023 27th International Conference Information Visualisation (IV), pp. 77-82. IEEE, 2023. |
| 4 | 1 | 2022 | Soni, Praveen, Cyril de Runz, Fatma Bouali, and Gilles Venturini. "Challenges for automatic dashboard generation systems in the context of novice users." In 18e journées Business Intelligence & Big Data (EDA 2022), pp. 111-112. Editions RNTI, 2022. |

# Abstract

Dashboards have become crucial tools for visualizing and interpreting complex datasets, particularly in fields like business intelligence, education, and public administration. They help users monitor processes, explore data, and derive insights through structured visual representations. However, the process of creating effective dashboards presents significant challenges, especially for novice users who may lack expertise in data visualization. These challenges include selecting the right visualizations, accurately mapping data attributes, and organizing these visual elements into a coherent and insightful dashboard.

Existing tools for dashboard creation, though often equipped with user-friendly interfaces, require substantial manual input and specialized knowledge. These tools tend to focus on the incremental selection and placement of individual visualizations without adequately addressing the need for a cohesive overall structure. As a result, users—particularly those who are new or with less experience—may find it difficult to create dashboards that effectively communicate complex, multidimensional data.

This thesis proposes SimpleVis, an Automatic Dashboard Recommendation System (ADRS) designed to assist users, particularly novices, in generating effective dashboards with minimal manual intervention. This includes the selection of visualizations, mapping of data attributes, and layout of the dashboard. Unlike traditional tools, SimpleVis provides a complete set of interconnected visualizations that work together to form a cohesive and informative dashboard.

In this thesis, a comprehensive review of existing multi-view generation systems and commercial dashboard tools was conducted, highlighting their limitations in supporting novice users. SimpleVis developed in this thesis addresses these limitations by offering fully automated recommendations. We consider the global structure of the dashboard, ensuring that the visualizations are not only individually relevant but also collectively coherent. This holistic approach simplifies the dashboard creation process, making it more accessible to users with varying levels of expertise.

Genetic algorithm (GA) plays the vital role in automatic generation of dashboards because GA is well-suited for optimization problems where the solution space is vast and complex. In this context, the goal is to create an optimal dashboard (DBo) with an objective function that effectively combines visualizations and data attributes to meet user needs. GA is an appropriate choice because of optimization in complex spaces, adaptability, customization, and efficiency.

SimpleVis user interface allows users to generate the dashboard as per their needs independent of user type. The system supports completely automatic recommendations of a set of dashboards without any user interventions. Considering the expert users SimpleVis also allows manual editing (metadata, visualization, and dashboard) and parameter setting (number of visualizations, type of visualization, GA operators, data attribute importance). The system allows users to create person-

alized DBo using a set of five best-recommended dashboards. The system provides tutorials, help, and suggestions if any user is lost while using the tool.

To validate the effectiveness of SimpleVis in contrast to other existing tools we perform a comparative study. This thesis includes user evaluations involving both novice and experienced users. The results demonstrate that the system significantly reduces the complexity and efforts required to create dashboards while improving the overall quality and relevance of the visualizations. The SimpleVis is particularly beneficial for novice users, enabling them to produce professional-quality dashboards without the need for extensive manual adjustments or specialized knowledge.

In conclusion, this thesis makes a significant contribution to the field of data visualization by developing SimpleVis which democratizes the dashboard creation process. By automating the selection and layout of visualizations, SimpleVis empowers users of all skill levels to effectively harness data, facilitating better decision-making and deeper insights. This thesis lays the foundation for future research in automatic dashboard generation and offers a practical solution to the challenges faced by novice users in an increasingly data-driven world.

# Contents


**1 Introduction** **1**
- 1.1 Context of research . . . 1
- 1.2 Statement of the research problem . . . 2
- 1.3 Objectives of the thesis . . . 3
- 1.4 Research methodology and thesis overview . . . 4

**2 State of the art** **7**
- 2.1 Definition for visualizations and dashboards . . . 8
- 2.2 Survey scope and methodology . . . 9
  - 2.2.1 Research methodology . . . 9
  - 2.2.2 Related previous surveys . . . 11
  - 2.2.3 Specific inclusion or exclusion of systems . . . 11
- 2.3 Dimensions used to describe ADRSs . . . 13
  - 2.3.1 Year . . . 14
  - 2.3.2 Considered visualizations . . . 14
  - 2.3.3 Type of visualization suggestion method . . . 14
  - 2.3.4 Utility functions . . . 16
  - 2.3.5 Layout . . . 16
  - 2.3.6 User feedback . . . 17
  - 2.3.7 User interface and user evaluation . . . 17
  - 2.3.8 License . . . 17
- 2.4 A classification of ADRSs . . . 18
  - 2.4.1 Overview . . . 18
  - 2.4.2 Year . . . 19
  - 2.4.3 Included visualizations . . . 19
  - 2.4.4 Suggestion mechanisms . . . 23
  - 2.4.5 Utility functions . . . 26
  - 2.4.6 User interfaces and evaluation . . . 27
  - 2.4.7 Market accessibility . . . 27
  - 2.4.8 Transverse categorization of systems based on methodologies and features . . . 29
- 2.5 Discussion . . . 31
  - 2.5.1 About this survey . . . 31
  - 2.5.2 Analysis of system launched year . . . 32
  - 2.5.3 No dominant methodology and lack of experimental comparison 32
  - 2.5.4 Novice users . . . 32
  - 2.5.5 Dashboard interface . . . 33
  - 2.5.6 User feedback . . . 33
  - 2.5.7 Lack of open source and freeware ADRS . . . 34
  - 2.5.8 Thesis goals and objective . . . 34

**3 Genetic algorithms** **37**
3.1 Model and Schema . . . 38
3.1.1 Overview of our system . . . 38
3.1.2 Reasons for the selection of a genetic algorithm . . . 39
3.1.3 Data model . . . 40
3.1.4 User model . . . 40
3.1.5 Visualization model . . . 41
3.1.6 Data and visual attribute type mapping . . . 41
3.1.7 Data to visualization mapping . . . 43
3.2 Genetic Algorithm . . . 44
3.2.1 Overview . . . 44
3.2.2 Dashboard genetic representation . . . 46
3.2.3 Random generation . . . 47
3.2.4 Genetic operators . . . 51
3.2.5 Fitness function . . . 58
3.3 Results . . . 62
3.3.1 Experimental set-up . . . 62
3.3.2 Parameter study . . . 63
3.3.3 Examples of generated dashboards . . . 63

**4 User interface for dashboard recommendation** **69**
4.1 System design . . . 70
4.1.1 System requirements and goals . . . 70
4.1.2 System Architecture . . . 71
4.1.3 Tech Stack Selection . . . 72
4.2 Data upload and in-build dataset . . . 72
4.3 System manual editing and parameter setting . . . 74
4.3.1 Metadata . . . 74
4.3.2 User data . . . 75
4.3.3 User preferences setting . . . 76
4.4 Data visualization and dashboard recommendation . . . 78
4.4.1 Main Dashboard and its control features . . . 78
4.4.2 Parameter Setting . . . 83
4.4.3 Five best recommended dashboards . . . 85
4.4.4 Personalized dashboard . . . 86
4.4.5 Dashboard Rating System and Additional Help . . . 87
4.4.6 Miscellaneous features . . . 90
4.5 Conclusion . . . 95

**5 User Evaluation: Comparative study** **97**
5.1 Participants and Dataset . . . 99
5.2 Evaluation protocol . . . 100
5.3 Results and Statistical Analysis . . . 102
5.3.1 Task 1 with Power BI and SimpleVis . . . 103

5.3.2 Task 2 with Power BI and SimpleVis . . . 107
5.4 Questionnaire Analysis . . . 111
5.5 User type based analysis of results . . . 115
5.5.1 Novices vs Experienced users: Task 1 with Power BI and SimpleVis . . . 115
5.5.2 Novices vs Experienced users: Task 2 with Power BI and SimpleVis . . . 121
5.6 System evaluation analysis summary . . . 127

**6 Conclusion and Perspective 131**
6.1 Conclusion . . . 131
6.2 Perspective . . . 132
6.2.1 Short Term . . . 132
6.2.2 Long Term . . . 133
6.2.3 Broader and Generalize Perspective . . . 134

**Appendices 137**

**A System use cases 139**
A.1 Novice user use cases . . . 140
A.1.1 Novice use case 1 . . . 141
A.1.2 Novice use case 2 . . . 143
A.1.3 Novice use case 3 . . . 146
A.2 Expert user use cases . . . 150
A.2.1 Expert use cases 1 . . . 151
A.2.2 Expert use case 2 . . . 164
A.3 Use cases summary . . . 171

**B System feedback 173**
B.1 System feedback . . . 173

**Bibliography 177**

# List of Figures


1.1 Proposed an order for reading the manuscript. The black arrows represent a sequence between chapters. . . . . 4

2.1 On the left side, a single visualization with one graph (bar chart) is presented while on the right side, multiple visualizations with four graphs (scatterplot, histogram, pie chart, line chart) are presented. . 8

2.2 PRISMA flow adapted from [Moher *et al.* 2009]. . . . . 10

2.3 The timeline representing the systems studied in our survey. . . . . 19

3.1 Our system takes input for (A) Preprocessed data file and four models: (B) Data Model, (C) Visulization Model, (D) Task Model, (E) User Model. With a GA, it produces an (F) optimized DBo, determines layout, and visualizes output with Vega-lite. A user can interact with the (G) user interface to recommend DBos and also regenerate DBos with modifications in the user and task model. . . . 38

3.2 Schema for selection of Genetic Algorithm(A) Algorithms (B) metaheuristics, (C) Several types of metaheuristics, (D) Genetic Algorithm, (E) Optimisation algorithms important features. . . . . 39

3.3 Data Model for the SAPS dataset: each data attribute has a name, a type, a number of distinct values, and user importance. . . . . 40

3.4 Visualizations model: a visualization has a name ("visualization" column), an identifier ("vis_ID"), and one or several visual attribute(s). If several visual attributes are necessary to define a visualization, we use several rows (one for each attribute). For each visual attribute, we represent its name ("VA"), type ("VAT"), the corresponding data attribute type it can be matched to ("CorresdDT"), the efficiency of the mapping with the data attribute ("VA_imp"), and the maximum number of distinct values supported by this visual attribute ("Max#Val"). . . . . 42

3.5 Mapping efficiency scores ("imp_value" column) derived from Mackinlay's matrix for evaluating the quality of a mapping between a data attribute and a visual attribute. . . . . 43

3.6 Complexity score assigned to each visualization. The complexity score of the visualization changes when the user type also changes as shown in columns NV_Complex, BE_Complex, and EX_Complex for novices, beginners, and experts respectively. . . . . 44

3.7 Both the figures collectively show the mapping between visual attributes and data attributes. On the left three visual attributes X, Y, and C are presented on the cartesian plane on the X-axis, Y-axis, and legend. On the right, a scatter plot is presented. These three visual attributes X, Y, and C correspond to data attributes 'Discussion', 'raisedhands', and 'class' on X-axis, Y-axis, and legend axis respectively. . . . . 45

3.8 Both the figures collectively show the mapping between visual attributes and data attributes. On the left visual attribute Angle ($\Theta$) is presented on the Cartesian plane as an angle. On the right, a pie chart is presented. While other visual attribute ColorHue(C) corresponds to legends in data attribute 'parentsschoolSatishfaction'. . . . 46

3.9 The table on top of the figure presents the visualization model with "visual attributes" and their type in columns $VA$ and $VAT$ respectively. Both columns are used to calculate the "visual attributes importance" given in column $VA_imp$. At the bottom of the figure, three visualizations (pie chart, bar chart, and scatterplot) are generated from this table. The feasibility of these visualizations is found based on a mapping between "visual attributes types" and corresponding "data attributes types" given in columns $VAT$ and $CorresDT$ respectively. Finally "visual attributes" are mapped with "data attributes" of the data set as indicated with a yellow, green, and red box in the table and numbering on visualizations. . . . . 47

3.10 The figure showed how the dashboard is generated from 4 individual visualizations (Scatterplot, Bar chart, Pie chart, and Histogram). Each visualization row shown in figure 3.11 is converted to a graphical chart with the help of Vega-lite encoding. . . . . 48

3.11 The DBo representation used in our GA: the four rows, are visualizations present in the DBo. Columns represent the data attributes. Values of the matrix are the visual attribute names and represent the mapping between the data attributes and the visual attributes. . . . 48

3.12 The figure presents parent 1 as a dashboard. It contains five visualizations: Scatterplot, Bar chart, Histogram, and two Pie charts . . . 51

3.13 The figure presents parent 2 as a dashboard. It contains four visualizations: Bar chart, Scatterplot, Histogram and Pie chart . . . . . 52

3.14 The figure presents both the parent's dashboard (parent-1 and parent-2) in the form of genetic representation. . . . . 53

3.15 The figure presents the cutting points (with red broken line) of the parent 1 dashboards for the crossover operator. This cutting point divides the parent 1 into two parts. The visualizations below this cutting point are exchanged. . . . . 54

3.16 The figure presents the cutting points (with red broken line) of the parent 2 dashboards for the crossover operator. This cutting point divides the parent 2 into two parts. The visualizations below this cutting point are exchanged. . . . . 55

3.17 The genetic representation of both dashboards for crossover operator. The red broken line presents the cutting point. The visualizations below this cutting point are exchanged. . . . . 56

3.18 Offspring 1, generated after crossover operator. This dashboard contains three visualizations in the first row from parent 1 and two visualizations in the second row from parent 2. . . . . 57

3.19 Offspring 2, generated after crossover operator. This dashboard contains two visualizations in the first row from parent 2 and two visualizations in the second row from parent 1. . . . . 58

3.20 The genetic representation of both offspring dashboards after crossover operator. . . . . 59

3.21 Add visualization mutation: Mutation is implemented on offspring 1. A new visualization "Bar Chart" is added to the second row of dashboard (present in red box). After mutation, there are six visualizations in the dashboard. . . . . 60

3.22 Remove visualization mutation: Mutation is implemented on offspring 1. An existing visualization "scatterplot" is removed from the first row of the dashboard. After mutation, there are four visualizations in the dashboard. . . . . 61

3.23 Mapping between "visual attributes" and "data attributes" for mutation: Mapping mutation is implemented on offspring-1. An existing visualization remains in the dashboard but the mapping between "visual attributes" and "data attributes" is changed for a few visualizations. The mapping is changed for "Scatterplot", "Histogram", and "Pie Chart" as shown in the red box. . . . . 62

3.24 The genetic representation of all three mutated dashboards. On the top, "add visualization mutation", where "BarChartNominal" row is added at the end of DBo. In the middle "remove visualization mutation", where "ScatterPlotOrdinal" row is removed between data attribute names, and "BarCharOrdinal". At the bottom Mapping between "visual attributes" and "data attributes" mutation. The mapping is changed only for the first and last two visualizations shown in the red box. The boxes (green, red, and orange) represent the old position of visual attributes while the arrows (green, red, and orange) represent the new position. . . . . 63

3.25 GA Operators Probability. (A) The selection of two pairs of DBo was done randomly from the population. (B) Selection of two parents (P1 and P2) for crossover based on binary tournament. (C) The crossover will be performed with a probability of 0.35 if a number of data attributes in the dataset is more than 5 else with a probability of 0.01. (D) The mutation will be performed with a probability of 0.65 if a number of data attributes in the dataset is more than 5 else with a probability of 0.99. (E) Add Visualization mutation will be performed with a probability of 0.8. (F) Remove Visualization mutation will be performed with a probability of 0.1. (G) Mapping mutation will be performed with a probability of 0.1. (H) Replacement can be possible in three ways: through crossover, mutation, or directly after binary tournament . . . . . 65
3.26 A typical run of the GA (population size of 50 and over 5000 generations): each curve represents respectively, the best (in blue), the average (yellow) and the worst (red) fitness values of the population. This run takes around 1.4s. . . . . 66
3.27 Two examples of DBo generated with our tool and for behavior analysis scenario. In (a), a DBo generated during the initial population with a fitness value of 0.099, and in (b) a final DBo generated after 20000 generations with a fitness value 0.729. This run takes around 6.4 seconds. . . . . 67

4.1 SimpleVis system welcome screen . . . . . 72
4.2 Data upload: There are 5 in-build datasets for testing SimpleVis tool 73
4.3 Data upload: uploading new dataset from local machine (the format of uploaded file must be .csv) . . . . . 74
4.4 On the left top (A) Metadata and on the left bottom (B) User data preview, with (C) user preferences setting on the right. . . . . 75
4.5 The first two rows of metadata are fixed: Data attribute name and Unique values. User can edit the last two rows: Data attribute types and user importance. . . . . 75
4.6 User dataset uploaded by user. It represents rows and columns in a dataset. . . . . 76
4.7 Four sets of user preferences: (A) Dashboard Generation Type, (B) User Type, (C) Number of Visualizations, and (D) Selected Visualizations . . . . . 77
4.8 Screen of step 3 contains 11 components: (A) Main Dashboard as generated by GA (B) Data attributes and their importance, (C) Selected Visualizations to be appear in the generated DBos (D) Number of Visualizations, (E) GA Parameters, (F) User type configuration (G) the five best Dashboards, (H) Personalized Dashboard, (I) Main DBo features control, and (J) Rating system for feedback and (K) Additional Help. . . . . 79

4.9 When a user clicks on (A) Info Icon it shows pop up with information: Detailed information about the dashboard fitness evaluation and data attributes information. . . . . 80
4.10 Each visualization of the main dashboard has 5 interactive icons with different functions. (A) Add visualization to personalized DBo, (B) Download this visualization, (C) View in full screen and edit, (D) Delete from this DBo, (E) Information about this visualization for detail study. There are two additional functionalities when the user hovers and single clicks on each visualization. . . . . 82
4.11 Each visualization of the main dashboard has 5 interactive icons with different functions. There are two additional functionalities when the user hovers and single clicks on each visualization. . . . . 83
4.12 There are 5 best dashboards that are automatically recommended by the system. It contains interactive features like saving, viewing information, downloading, and deleting. . . . . 84
4.13 There are 5 best dashboards that are automatically recommended by the system. It contains interactive features like saving, viewing information, downloading, and deleting. . . . . 85
4.14 Personalized dashboard can be generated from 5 best-recommended dashboards. It contains interactive features like locking/unlocking the dashboard, viewing information, downloading, and deleting. . . . 86
4.15 Rating system for dashboard shows that user selected problems 1, 2, 4, 5, and 8 out of a total 8 problems for dashboard recommendation improvements . . . . . 87
4.16 Rating system for dashboard shows the suggested solutions for problems 1, 2, 4, 5, and 8 for dashboard recommendation improvements. . 88
4.17 The system tutorial acts as the complete guide for the new users to make them familiar with system uses and features. . . . . 89
4.18 System evaluation form with section 1 question and section 2 question from Q1 to Q3. . . . . 90
4.19 System evaluation for with question from Q4 to Q11 of section 2 . . 91
4.20 Contact Us form for user help and support . . . . . 92
4.21 Scatter plot with alternative data attribute features keeping visualization constant. When user clicks on edit icon (shown in red box), an window will appear at top of visualization with data attributes corresponding to visual attribute (shown in green box). When user clicks on drop down of respective data attribute all alternative data attribute of same type will appear. . . . . 92
4.22 In this Scatter plot, the visual attribute 'color' have total 4 data attributes options for mapping. . . . . 93
4.23 Generation of alternative visualization of keeping data attributes constant. This feature is only in the case of simple bar charts and pie charts. . . . . 94

4.24 SimpleBar Chart is replaced with alternative 'Pie Chart' keeping data attributes constant . . . . . 94

5.1 Level of Mastery: a couple of questions to participant related to their level of mastery in using the application for data visualization or dashboard generation. The participant needs to answer first if he/she is already familiar with such applications (Power BI, Tableau, Qlik Sense, or similar) or not. If the participant chooses no for the first question, we consider them novice users. In case of yes they need to choose their mastery level from 1 to 4 as follows: 1 for beginner, 2 for learners, 3 for advanced, and 4 for expert. . . . . 98
5.2 Results analysis schema based on tasks and user types: on the left workflow of task based results analysis and on the right workflow of user type based results analysis. . . . . 102
5.3 The box represent summary statistics for task 1 done with Power BI while on the other side red box represent task 1 done using SimpleVis. The lower value for T1-SV with respect to T1-PBI for different parameters indicates that participant perform the T1 with SimpleVis is more faster than PowerBI. . . . . 105
5.4 The box represents the summary statistics of the response time for task 2 completed with Power BI, while the red box represents task 2 performed using SimpleVis. The lower values for T2-SV compared to T2-PBI across various parameters indicate that participants completed task 2 faster with SimpleVis than with Power BI (and the difference is significant from a statistical point of view with a p-value of 0.00013 during the statistical test). . . . . 109
5.5 The first 4 (Q1 to Q4) are related to individual tools. . . . . 112
5.6 The next 7 comparative questions (Q5 to Q11) for PowerBI and SimpleVis. . . . . 113
5.7 The box represent summary statistics for task 1 done with PowerBI by experienced and novices users in blue and red respectively. While on the other side yellow and green box represent task 1 done using SimpleVis by experienced and novice users respectively. . . . . 117
5.8 The boxes represent the summary statistics for Task 2, with SimpleVis results shown in blue for experienced users and red for novice users. On the other hand, the yellow and green boxes represent Task 2 completed using SimpleVis by experienced and novice users, respectively. . . . . 125

A.1 We have presented (A) five use case, (B) three for novices and (C) two for experts. Each of it objective is mentioned below each of them. 140

# List of Tables


2.1 The different survey or state-of-the-art papers considered in our survey, with three categories: on top, general surveys, and information about dashboards. In the middle, surveys focused on specific features in dashboards. On the bottom, are surveys related to visualization recommendations. P denotes the number of studied papers, when relevant. . . . . 12
2.2 The 8 studied dimensions for the 19 systems included in our survey. . 13
2.3 Different acronyms and their meaning. . . . . 21
2.4 Overview of the studied ADRSs. . . . . 22
2.5 A summary of the user evaluations performed in ADRSs. U denotes the number of users involved in the evaluation. . . . . 28

3.1 Coefficients values as per user types . . . . . 60
3.2 Sample of results obtained during the grid search: values represent the best fitness values obtained at each run and averaged over ten runs. Values in parenthesis represent the standard deviations and the best result is highlighted. For each run, the running time is less than 10 seconds. . . . . 64
3.3 Fitness value analysis: DBo (Initial Population) vs DBo (after GA run) 66
3.4 Data attribute analysis: DBo (Initial Population) vs DBo (after GA run) . . . . . 67

5.1 Statistical Analysis of time taken in seconds by users for each sub-task of T1 (with a p_value of 0.00273 during statistical t-test). . . . . 103
5.2 Quality analysis of task 1 performed by participants. NV is for novices and EXP is for experienced users. . . . . 106
5.3 Statistical Analysis of response time taken in seconds by users for each sub-task of T2 with a significant difference between the two means (p-value of 0.00013 during the statistical t-test). . . . . 107
5.4 Quality analysis of task 2 performed by participants. NV is for novices and EXP is for experienced users. . . . . 110
5.5 Post-task questionnaire mean and standard deviation for questions Q1 to Q11. . . . . 114
5.6 Statistical Analysis of time taken in seconds by novices user and experienced users to complete the Task 1 with Power BI. . . . . 116
5.7 Quality analysis of task 1 performed by novices and experienced participants with PBI. . . . . 118
5.8 Statistical Analysis of time taken in seconds by novices user and experienced users to complete the Task 1 with SimpleVis. . . . . 119

5.9 Quality analysis of task 1 performed by novices and experienced participants with SimpleVis. . . . . 120
5.10 Statistical Analysis of time taken in seconds by novices user and experienced users to complete the Task 2 with PowerBI. . . . . 121
5.11 Quality analysis of task 2 performed by novices and experienced participants with PBI. . . . . 123
5.12 Statistical Analysis of time taken in seconds by novices user and experienced users to complete the Task 2 with SimpleVis. . . . . 124
5.13 Quality analysis of task 2 performed by novices and experienced participants with SV. . . . . 126
5.14 Task Based Analysis Summary . . . . . 127
5.15 User Type Based Analysis Summary . . . . . 129

# List of Acronyms

**DBo** Dashboard

**ADRSs** Automatic dashboard recommendation systems

**MpV** Multiple View

**"CCP"** Composition and Configuration Patterns

**KPIs** Key performance indicators

**MT** Mark Type

**MV** Many Visualizations

**SI** Single and Independent

**SiD** Single and Dependent

**GO** Global Optimization

**DM** Direct Matching

**T** Predefined Dashboard Templates

**UIAs** Design from User's Initial Actions

**VQL** Visualization Query Language

**SD** Statistical Properties of the Dataset

**BB** Behavior-Based

**ML** Machine Learning

**IN** Insights

**NF** Not Found

**MS** Manual Size

**MLa** Manual Layout

**IG** Imposed on a Grid

**OL** Optimized Layout

**OS** Optimized Size

**PT** Predefined Templates

**DF** Direct Feedback

**BF** Behavioral Feedback

**DD** Drag and Drop

**SM** Selection Mechanisms

**C** Commercial

**OS** Open-Source

**F** Freeware

**GA** Genetic Algorithm

**SAPS** students' Academic Performance Dataset

**CRUD** Create, Remove, Update, and Delete

**SV** SimpleVis

**PBI** PowerBI

**DGR** Dashboard Genetic Representation

Chapter 1

# Introduction

Contents


1.1 Context of research . . . . . 1
1.2 Statement of the research problem . . . . . 2
1.3 Objectives of the thesis . . . . . 3
1.4 Research methodology and thesis overview . . . . . 4


## 1.1 Context of research

In the realm of data, visualization is crucial for succinctly conveying complex and abundant information through essential visual representations. A commonly utilized tool in this context is the dashboard [Pauwels *et al.* 2009, Sarikaya *et al.* 2018, Tundo *et al.* 2020]. As we will detail further in this thesis, there is not a single definition for this visual representation. We can attempt to simply define a dashboard as a structured set of visualizations, with interactions, that allow users to monitor a process, explore data, and get insights about it. Dashboards are very common in many domains like business intelligence, learning, data analysis, decision-making, data storytelling, and presentation.

As part of the national funding (ANR, French national research agency), we worked on the BI4PEOPLE project. In BI4PEOPLE project there are several work packages manage by other partners in France including other labs and companies. My work package in more focus about building system user interface that automatically recommend dashboards.

Creating an efficito the userent dashboard is not as easy as it seems. Indeed, in the industry [Mackinlay *et al.* 2007a, Gowthami & Kumar 2017, QlikTech International 2013a, Sisense Inc 2004a, Tableau Software 2007a, Microsoft 2015a] and the web [Luo *et al.* 2018a, Database Group 2018], there are many tools to create dashboards. Such tools provide efficient interfaces to edit manually a dashboard: this consists of selecting which data to visualize, then which visualizations to use, how to organize them on the screen, and what dependencies should exist between them like linking or brushing. However, this series of choices is not easy to resolve. In the academic domain, many researchers agree that creating

a single visualization is not straightforward. So creating a set of visualizations is indeed even more complex.

For several decades (see Table 2.4 and(see Table 2.2.2), researchers have studied how to automatically generate data visualizations, to fulfill the user's objectives. Such generative tools are being integrated into the most advanced [Mackinlay *et al.* 2007a, Microsoft 2015a, QlikTech International 2013a, Sisense Inc 2004a, Tableau Software 2007a, Microsoft 2015a] commercial software that suggests visualizations and dashboards such as Tableau, PowerBI, Qlik Sense, or Excel. More recently, academic researchers have also studied how to automatically recommend not a single visualization, but rather a set of visualizations, i.e. a dashboard. Multiple visualizations help users make better decisions [Costa & Aparício 2019] because the dashboard contains multiple views in contrast to a single visualization. Many viewpoints can be defined in this process of dashboard recommendation: data visualizations, generative algorithms, layout, interfaces, user feedback, and finally the validation of such approaches with user evaluations.

## 1.2 Statement of the research problem

A dashboard (DBo) is a tool commonly used in many human activities. Dashboards can be considered as a special case of multiple view (MpV) visualizations, a set of approaches that makes use of several coordinated visualizations to explore data. The distinction between them is not obvious [Wu *et al.* 2021a]. One difference is probably the target users and tasks. While a MpV visualization can include systems with advanced and specific visualizations intended to be used by domain experts in a data mining context, the dashboard could consider simpler and commonly used visualizations and a broader audience, i.e., novice users like citizens or business experts, who would like to explore or monitor multidimensional data. However, the recent progress made in commercial software for instance is such that dashboard creation/management systems are becoming more and more complex, thus making the difference between dashboards and other MpV approaches thinner.

Automatic dashboard recommendation systems (ADRSs) are the focus of this thesis and can be defined as follows: an ADRS is a system that generates a dashboard with the use of an automatic suggestion mechanism. It is a tool that assists users in the conception and implementation of a dashboard by formulating recommendations. Such recommendations can occur at any of the major steps of dashboard creation: selection of data, selection of visualizations with their data mapping, and layout of the visualizations on the screen. Systems that rely only on manual editing, even with user-friendly interfaces, are not considered as ADRSs. A suggestion mechanism has to be used. Indeed, many ADRSs will also include the possibility to edit the generated dashboard. Many of them can thus be considered as semi-interactive or semi-automatic systems, i.e., with the use of a tight coupling between a suggestion mechanism and a user interface. ADRSs follow different methods of dashboard

recommendations like predefined dashboard templates, machine learning and deep learning algorithms, statistical properties of the dataset, and direct matching.

The problem we study in this thesis is how to help users, possibly novice users, to create an efficient DBo with as less human intervention as possible. This problem implies selecting a set of visualizations, including the mapping with data attributes, and proposing a layout of these visualizations. This thesis focuses on the first problem (selection of visualizations). Selecting an efficient set of visualizations to create a DBo is a difficult problem: selected visualizations have to be adapted to the data, and to the user's goals and skills. These visualizations are also interdependent: it is the whole set of visualizations that provides the most powerful DBo.

## 1.3 Objectives of the thesis

In the scientific literature, several works are devoted to the generation or suggestion of one visualization, like for instance ExploroBOT [McAuley *et al.* 2019], Foresight [Demiralp *et al.* 2017], ManyEyes [Viegas *et al.* 2007], Data2vis [Dibia & Demiralp 2019], Google fusion tables [Gonzalez *et al.* 2010], ViewSeeker [Zhang *et al.* 2019], Visclean [Luo *et al.* 2020], Polaris [Stolte *et al.* 2002], Autovis [Wills & Wilkinson 2010], Lyra [Satyanarayan & Heer 2014], SeeDB [Vartak *et al.* 2015], and DVMS [Wu *et al.* 2014]. We concentrate our study on ADGS which can suggest a complete DBo, with several visualizations.

Considering the industry, many software propose semi-interactive tools to select visualizations in a DBo, like Tableau [Mackinlay *et al.* 2007b, TABLEAU SOFTWARE 2007b], Qlik Sense [QlikTech International 2013b], PowerBI [Microsoft 2015b], Quick Sight [Amazon Web Services 2016a], Sisense [Sisense Inc 2004b], Draco [Moritz *et al.* 2018a], Exploration View [Elias & Bezerianos 2011a], VizDeck [Key *et al.* 2012a], Voyager 2 [Wongsuphasawat *et al.* 2017a], "Engaging Dashboards" [Aksu *et al.* 2019a]. Most of them propose visualizations one by one that is incrementally added to the DBo being constructed. In general, they do not generate at once a complete set of co-adapted visualizations. Hence, the suggestion mechanism does not take into account the global structure of the DBo being created.

Such tools can be completely automatic like Scagexplorer [Dang & Wilkinson 2014a], Deepeye [Luo *et al.* 2018b], and DashBot [Da Col *et al.* 2021a], or semi-automatic like Vizdeck [Key *et al.* 2012a], and Zenvisage [Siddiqui *et al.* 2016a], or manually-customized DynSpace [El Meseery *et al.* 2018a], MultiVision [Wu *et al.* 2021b], "CC" [Chen *et al.* 2020a]. However, there is no existing tool for novices that supports complete automation for the multiple visualizations generation problem.

The aim of the thesis is to develop a tool with the following objectives:

- data visualization and dashboard analysis tool for novices,
- automatic dashboard recommendation, with a set of co-adapted visualizations,




Figure 1.1: Proposed an order for reading the manuscript. The black arrows represent a sequence between chapters.

- minimum user's manual interventions,
- tools take user preferences into account for better and customized dashboard recommendations,
- simple and fast with a user-friendly interface,

## 1.4 Research methodology and thesis overview

In this thesis, we state the need for automatic dashboard recommendation systems. The research is supported by the state of the art with 19 multi-view generation systems. And also provides a complete process of design and development of the system to recommend the best dashboards based on user preferences. We validated the system's aim and authenticity with user evaluation.

The thesis structured presented in Figure 1.1 is as follows: Chapter 2 explores the current state of the art in automatic dashboard recommendation systems. This includes a comprehensive review of existing methodologies, technologies, and their applications, highlighting key advancements and identifying existing gaps in the research. Chapter 3 delves into the intricacies of genetic algorithm and its experimental results. This chapter covers the fundamental concepts of our proposed algorithms. It also includes detailed illustrations of the selection process used for operators (such as crossover and mutation) and operand selection. Subsequently, we implemented these algorithms to optimize dashboard recommendations effectively.

In chapter 4, we provide an in-depth overview of the user interface for dashboard recommendation named SimpleVis with the system's design rationales behind them. This interface began with the system's architecture, schema diagrams, design patterns, and the decision-making process that guided the development of the system. During the system development, we ensure it meets the specified requirements and performance criteria of different user profiles.

Chapter 5 focuses on the user evaluation of the system. This chapter details the methodologies used for system testing by users, the feedback received, and the overall user experience. It also includes empirical data gained from system evaluation by users and performance metrics. Finally, we end this section with a statistical analysis of user evaluation feedback based on the comparative study of both the tools SimpleVis and Power BI.

Finally, chapter 6 concludes the thesis with a thorough conclusion and future perspectives. This chapter synthesizes the key findings, reflects on the implications of the research, and suggests potential directions for future work to advance the field of automatic dashboard recommendation systems.

This manuscript also contains two appendixes. In Appendix A we have presented five use cases. There are three use cases for novices that show the system's adaptability and evolving complexity as users become more experienced and interact more with its features, from fully automated dashboard generation to more refined, customized, and comparative approaches. There are two for experts, both use cases (EUC1 and EUC2) showcase the system's versatility and its capacity to support expert users, from simple feature exploration with an in-built dataset to a more involved workflow using new data and domain knowledge for customization.

Appendix B contains system feedback form. This form allows users to give their feedback and share experiences. In addition user's feedback and experiences also help us to improve our tool to meet their expectations.

Chapter 2

# State of the art

## Contents


**2.1 Definition for visualizations and dashboards . . . . . 8**
**2.2 Survey scope and methodology . . . . . 9**
2.2.1 Research methodology . . . . . 9
2.2.2 Related previous surveys . . . . . 11
2.2.3 Specific inclusion or exclusion of systems . . . . . 11
**2.3 Dimensions used to describe ADRSs . . . . . 13**
2.3.1 Year . . . . . 14
2.3.2 Considered visualizations . . . . . 14
2.3.3 Type of visualization suggestion method . . . . . 14
2.3.4 Utility functions . . . . . 16
2.3.5 Layout . . . . . 16
2.3.6 User feedback . . . . . 17
2.3.7 User interface and user evaluation . . . . . 17
2.3.8 License . . . . . 17
**2.4 A classification of ADRSs . . . . . 18**
2.4.1 Overview . . . . . 18
2.4.2 Year . . . . . 19
2.4.3 Included visualizations . . . . . 19
2.4.4 Suggestion mechanisms . . . . . 23
2.4.5 Utility functions . . . . . 26
2.4.6 User interfaces and evaluation . . . . . 27
2.4.7 Market accessibility . . . . . 27
2.4.8 Transverse categorization of systems based on methodologies and features . . . . . 29
**2.5 Discussion . . . . . 31**
2.5.1 About this survey . . . . . 31
2.5.2 Analysis of system launched year . . . . . 32
2.5.3 No dominant methodology and lack of experimental comparison 32
2.5.4 Novice users . . . . . 32
2.5.5 Dashboard interface . . . . . 33
2.5.6 User feedback . . . . . 33

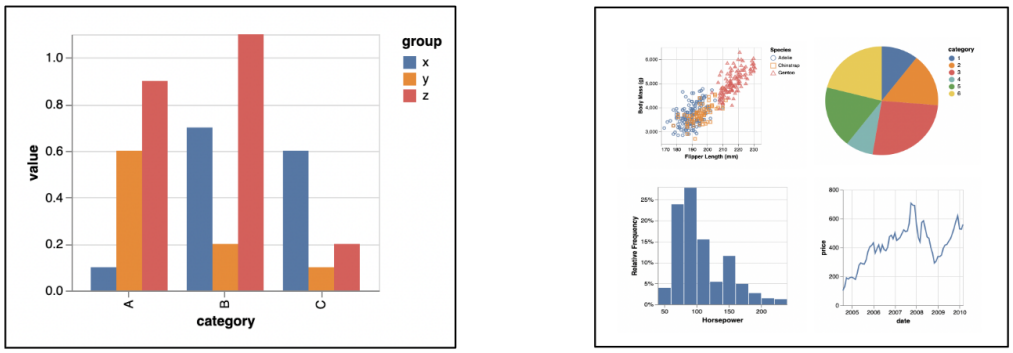



Figure 2.1: On the left side, a single visualization with one graph (bar chart) is presented while on the right side, multiple visualizations with four graphs (scatterplot, histogram, pie chart, line chart) are presented.

2.5.7 Lack of open source and freeware ADRS . . . . . 34
2.5.8 Thesis goals and objective . . . . . 34

This chapter 2 presents the state of the art on automatic or semi-automatic recommendation systems that help users create dashboards. It starts by showing the important role that dashboards play in data science and gives an informal definition of dashboards, i.e., a set of visualizations possibly with linkage, a screen layout, and user feedback. We are mainly interested in systems that use a fully or partially automatic mechanism to recommend dashboards to users. This automation includes the suggestion of data and visualizations, the optimization of the layout, and the use of user feedback. We position our work with respect to existing surveys. Starting from a set of over 1000 papers, we have selected and analyzed 19 papers/systems along several dimensions. The main dimensions were the set of considered visualizations, the suggestion method, the utility/objective functions, the layout, and the user interface. In this thesis, we have highlighted the key achievements in this field and concluded by proposing the perspectives and goals we have studied.

## 2.1 Definition for visualizations and dashboards

**Single visualization vs multiple visualizations.** When there is only one visualization on screen is called a single visualization. Single visualizations focus on a certain number of data attributes of a dataset. When there is more than one on-screen we call multiple visualizations. Multiple visualizations give a collective analysis and insights into the whole dataset. Multiple visualization are also called dashboard.

**What is a dashboard.** Even if multiple definitions can be proposed for DBo [Brouns *et al.* 2015a], there is no formal definition of a dashboard [Sarikaya *et al.* 2018]. We will consider that a DBo is a set of visualizations with interactions, intended to visualize a dataset that users must explore, analyze or predict, in order to make decisions. Indeed many authors give their own definitions, depending on the studied domain: Stephen Few defined a dashboard as "a visual display of the most important information needed to achieve one or more objectives; consolidated and arranged on a single screen so the information can be monitored at a glance" [Few 2006]. Brouns et al. described dashboards as "an easy to read, often single page, real-time user interface, showing a graphical presentation of the current status (snapshot) and historical trends of an organization's key performance indicators (KPIs) to enable instantaneously and informed decisions to be made at a glance" [Brouns *et al.* 2015b]. Others referred to dashboards as "a container of indicators" [Ji *et al.* 2014] or as "an emerging performance management system, for example, to monitor productivity, analyze cost-effectiveness and improve customer satisfaction" [Park & Jo 2015]. Dashboards can be considered as a special case of multiple view (MpV) visualizations, a set of approaches that make use of several coordinated visualizations to explore data. The distinction between them is not obvious [Wu *et al.* 2021a]. One difference is probably the target users and tasks. While a MpV visualization can include systems with advanced and specific visualizations intended to be used by domain experts in a data mining context, a dashboard could consider simpler and commonly used visualizations and a broader audience, i.e., novice users like citizens or business experts, who would like to explore or monitor multidimensional data. However, the recent progress made in commercial software for instance is such that dashboard creation/management systems are becoming more and more complex, thus making the difference between dashboards and other MpV approaches thinner.

**Manual dashboard vs automatic dashboard.** In a manual dashboard there is a maximum of human intervention to generate the dashboard. There is no use of certain methods/algorithms for dashboard recommendation in manual dashboard generation. In an automatic dashboard there is no (minimal) human intervention There is/are the use of certain methods/algorithms for automatic dashboard recommendation.

## 2.2 Survey scope and methodology

### 2.2.1 Research methodology

To find a relevant set of papers, we performed a systematic literature review following the guidance proposed in [Schwendimann *et al.* 2016, Toasa *et al.* 2018, Vázquez-Ingelmo *et al.* 2019, Abduldaem & Gravell 2019]. Figure 2.2 illustrates our process of identifying papers from different resources. It finally results in selecting 50 papers from which 19 are relevant to ADRSs. We considered the following

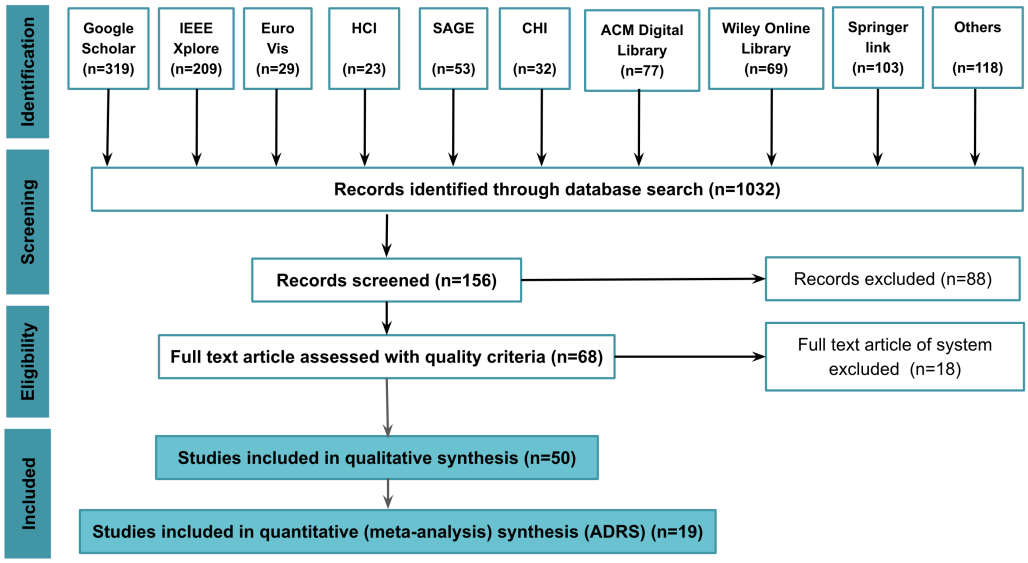



Figure 2.2: PRISMA flow adapted from [Moher *et al.* 2009].

sources: Google Scholar, IEEE Xplore, ACM Digital Library, Wiley Online Library, Springer Link, Science Direct, Semantic Scholar, CHI, SAGE, EuroVis, HCI, and others. We were also interested in other online articles/videos to gather information and knowledge. For querying those sources, we used the following main keywords: “dashboard automatic generation”, “visualization recommendation”, “dashboard generation”, “dashboard visualization”, “multiple view visualizations”, “business intelligence dashboard”, “multiple visualization tools”, “multi-vision dashboard”, “dashboard guideline and challenges", “need of dashboard". Our last access to the resources was on 15th September 2021 with the latest and updated versions of tools/papers/systems.

Initially, we identified 1032 papers using different keywords from different resources. We screened them on the basis of different dimensions like number of citations, publishers, and year of publication, and considered around 156 interesting papers. We discarded more papers based on new dimensions like abstract reading, a quick overview of sections, figures, dashboard interface, layout, methodology, evaluation, and user feedback. With these dimensions, we got the basic idea of each of these papers which helped us to consider papers for full-text reading. Finally, we came up with 68 papers for full-text reading. We categorized these papers into four categories: survey papers, ADRS, single-view systems, and miscellaneous (needs and challenges in ADRS, critics, pros and cons in ADRS, common algorithms in ADRS, guidelines to develop ADRS, features, and components of ADRS). We excluded the single view systems (18) because our survey is specific to MpV visualizations. We finally included 50 papers of three categories, 16 miscellaneous, 15 surveys, and 19 ADRSs.

### 2.2.2 Related previous surveys

We reviewed several surveys or state of the art papers that deal with dashboards and ADRSs. Table 2.2.2 splits them into three parts. In the first part (top of the table), several papers are devoted to dashboards in general, and their use in specific domains or in the industry. Such papers contribute to clarifying the context, domain, and usage of dashboards. In the second part (middle of the table), papers provide surveys with a more specific focus that can be related to ADRSs. In the third part of the table (bottom), papers present recommendation systems, and they can be compared to those of our survey. [Kaur & Owonibi 2017] reviewed different recommendation strategies but for single visualizations, while we concentrate on complete dashboards. [Law *et al.* 2020] analyzed 20 papers dealing with automated insights along 16 different dimensions. These 16 dimensions are majorly divided into two categories, auto-insights and purposes

[Qin *et al.* 2020] performed a survey about making data visualization more effective and efficient. [Zhu *et al.* 2020] performed a survey for an automatic recommendation of infographics and visualizations. First, they broadly classified all the considered systems into two categories: data-driven and knowledge-based. Then they further classified them within these two categories with different dimensions: statistical automatic visualization, automatic storytelling visualization, automatic annotation, etc. Both surveys [Qin *et al.* 2020, Zhu *et al.* 2020] included all types of automatic visualization generation systems independent of view types (single view systems and MpV systems).

In our survey, we concentrate on ADRSs, the core recommendation method for dashboards and its integration in a user interface, and provide a more detailed analysis of such systems along 8 specific dimensions.

### 2.2.3 Specific inclusion or exclusion of systems

Our survey includes three systems that do not generate dashboards exactly but rather MpV structures. They use recommendation methods of interest: ScagExplorer [Dang & Wilkinson 2014b], DataShot [Wang *et al.* 2019] and Calliope [Shi *et al.* 2020].

We have excluded also three systems. Qualdash [Elshehaly *et al.* 2020] offers design principles and a process to generate dashboards from the analysis of tasks and requirements expressed by users. It uses a Metric Specification Structure, which, once defined, can be automatically turned into a Vega-lite representation of the dashboard. However, this specification structure is determined by human experts, so we decided not to include Qualdash in the detailed analysis, even if it represents valuable work.

LADV considers the same principle of using initial user specifications but with sketches or images [Ma *et al.* 2020]: the user provides the system with an image that represents the future dashboard, and LADV, using image recognition methods, turns the graphical input into a dashboard. The user must then determine the

Table 2.1: The different survey or state-of-the-art papers considered in our survey, with three categories: on top, general surveys, and information about dashboards. In the middle, surveys focused on specific features in dashboards. On the bottom, are surveys related to visualization recommendations. P denotes the number of studied papers, when relevant.

| Reference | P | Scope | Dimensions of analysis |
|---|---|---|---|
| [Schwendimann *et al.* 2016] | 55 | Learning dashboard | Context, contribution, users, settings, data, eval. |
| [Toasa *et al.* 2018] | | Real-time information | Visualization techniques and platforms |
| [Diamond & Mattia 2017] | 4 | Software tools | Excel, Power BI, Tableau, IBM Watson Analytics |
| [Kubernátová *et al.* 2018] | 15 | Recommender systems | Data, task, domain Knowledge, user preferences |
| [Gowthami & Kumar 2017] | 5 | BI Tools for enterprise | License, trail, ease of use, training, platform, Integration, price |
| [Vázquez-Ingelmo *et al.* 2019] | 23 | Tailored dashboards | Year, author, type of publication, domain, factors, target, stage, method, test |
| [Pauwels *et al.* 2009] | | Dashboards as a service | Purpose, user, effectiveness, development, adoption |
| [Abduldaem & Gravell 2019] | 18 | Design of dashboards | Accuracy, completeness, representation, objectivity |
| [Sarikaya *et al.* 2018] | | About dashboards | Purpose, audience, visual features, data semantics |
| [Shen *et al.* 2021] | 45 | Nat. lang. interfaces | Query interpretation, data transformation, mapping, view, interaction, dialogue, presentation |
| [Elias & Bezerianos 2012] | | Annotating dashboards | Needs, prototype, design |
| [Zhu *et al.* 2020] | 22 | Recommendations | Data-driven, knowledge-based, hybrid |
| [Law *et al.* 2020] | 20 | Automated data insights | Types, purposes |
| [Kaur & Owonibi 2017] | | Recommendation | Data, task, knowledge, user preferences |
| [Qin *et al.* 2020] | 29 | Data visualization | Specifications, approaches, recommendation |

data to visualization mapping, or possibly edit the dashboard. Again, the initial specification is given by the user and not by a suggestion algorithm. Hence we have not included LADV in our survey.

Voder [Srinivasan *et al.* 2018] is a system that helps users discover and highlight

data facts in a visualization. Voder has a simple suggestion mechanism for proposing alternative visualizations to represent data facts. However, Voder is more intended to generate one visualization rather than an MpV visualization, so we did not include it in our survey.

In our survey, we also do not include the following MpV systems, DIVE [Hu *et al.* 2018], VizML[Hanrahan 2006], and VizQL[Hu *et al.* 2019] because they are neither automatic nor semi-automatic.

Finally, several system exist that suggest single visualizations to users, and not dashboards. Hence we have excluded such systems in our survey.

Table 2.2: The 8 studied dimensions for the 19 systems included in our survey.

| Important step/features in ADRS | Studied dimensions |
|---|---|
| Work origin | System's name, year, and reference |
| Suggestion of visualizations | Considered visualizations<br>Suggestion method<br>Utility functions |
| Layout | Layout method |
| User's viewpoint | User feedback<br>User interface and evaluation<br>License |

## 2.3 Dimensions used to describe ADRSs

To describe the 19 collected papers/systems, we have defined 8 dimensions that represent key characteristics of ADRS (see Table 2.2). Those dimensions were derived both from what is observed in existing systems and also from conceptual analysis of the main steps/features that should be present in an ADRS:

- a recommendation method, that will consider a set of visualizations, among

which selected visualizations will be proposed by a suggestion method, possibly using utility functions,

- a layout of the suggested visualizations, with several possible methods,
- a user's viewpoint, with the possibility of providing feedback to the system, the features of the user interface and its evaluation, and the software license of the system which might limit its use.

In the following sections, we give more details about each dimension.

### 2.3.1 Year

This is the year the paper was published. This gives an idea of the currentness of the work. Besides that, we added the paper/system reference.

### 2.3.2 Considered visualizations

The visualizations that can be integrated into the dashboard belong to a predefined set. With this dimension, we want to characterize this set. Such visualizations can be:

- standard and typical visualizations, such as scatter plot (SP), table (T), bar chart (BC), pie chart (PC), line chart (LC), map (M), histogram (HT), area chart (AC), gantt chart (GC), radar chart (RC), gauge (G), frequency polygon (FP),
- mark type (MT) like cross, plus, circle, etc. Some systems propose the choice of a mark type to generate visualizations. These mark types are not considered directly as visualization types but ultimately they are the main component of visualization (like a circle in the case of a scatter plot),
- many visualizations (MV), a case in which many visualizations are proposed (including the standard ones and many others).

In this dimension, we also mention the total number of visualizations available in the system. If we did not find this number of visualizations, we indicate it by "not found" (NF).

### 2.3.3 Type of visualization suggestion method

This is an important dimension as it is a key component of an ADRS. Here, we want to classify the different methods that are used to propose visualizations to the user. First, we make a distinction between ADRSs according to the overall functioning of the suggestion method:

- single and independent (SI), in which the suggestion of the dashboard is done by proposing visualizations one by one, whatever the current content of the dashboard, i.e. the previously selected visualizations,

- single and dependent (SiD), in which the suggestion method also proposes visualizations one by one but taking into account the current state of the dashboard,
- global optimization (GO), in which the method proposes a complete set of co-adapted visualizations, thus performing a global optimization of the suggested visualizations.

.

We now make distinction between methods according to the core algorithm used to make suggestions. Some methods use simple suggestion algorithms with straightforward heuristics that do not involve learning or specific adaptation to the data distribution:

- direct matching (DM) is one of the simplest methodologies that matches data columns (categorical, numerical, temporal, etc) to visual attributes of a simple visualization (in general BC, PC, SP) with simple and fixed rules. It might use a basic decision model designed by hand from expert knowledge,
- Predefined dashboard templates (T) can be used to propose a set of visualizations with its associated layout. Templates can be determined with expert knowledge or past user behavior.

We distinguish a second set of methods that try to adapt the suggested set of visualizations according to statistical information about data or users, or according to expert knowledge. We call them advanced suggestion methods because they are not strictly fixed algorithms but methods whose behavior depends on data characteristics and user objectives. These methods use:

- design from User's Initial Actions (UIAs) is a process that allows users to initially choose particular columns (dimensions or measures) of the given data. Then the system automatically generates and recommends the visualizations relevant to selected data using heuristics often inspired by expert or domain knowledge. This process typically supports user interaction with different system features (drag and drop, selection mechanism) to further manually edit the dashboard,
- visualization query language (VQL) and associated knowledge expressed as constraints. The current situation (data, objectives) is turned into a query and a method such as a constraint solver can output a set of matching visualizations,
- statistical properties of the dataset (SD). Such a method evaluates statistical properties (i.e entropy, variance, probability distribution, etc) of the dataset to rank and recommend visualizations,
- behavior-based (BB) approach that considers users' current or past behavior as inputs, and possibly from a large usage history. Such an approach infers

knowledge about visualizations, or users' intended tasks, so as to recommend useful visualizations. They involve simple statistical properties or machine learning (ML) models.

Finally, a third set of complex suggestion methods can be distinguished, and it uses:

- ML models trained on a large set of behavioral observations. Many ML methods are considered in this context: decision trees, learning to rank, neural networks, reinforcement learning, long short-term memory, and deep learning.

### 2.3.4 Utility functions

During the optimization process of a dashboard, the system might use objective functions to evaluate the possible options before proposing them to the user. These functions are evaluation functions that automatically evaluate a criterion. It can be the expressiveness or effectiveness of the visualization [Mackinlay 1986], the coverage (are all important data attributes represented in the dashboard), etc. We consider the following cases for utility functions:

- insights (IN), i.e. the potential interest of the discoveries that can be done with this visualization. For instance, to suggest a potentially interesting scatter plot, a utility function can be the correlation between the two axes. In this case, the system will prefer visualizations of correlated attributes,
- "none", when systems do not involve utility functions,
- "not found" (NF), for systems that do not give a clear idea whether they used utility functions or not.

### 2.3.5 Layout

A dashboard is a set of visualizations that have to be organized on one or more computer screen(s). This organization raises the problem of finding a correct layout of the visualizations. It means that both the location on the screen as well as the size of each graphic have to be determined. In addition, the layout can be spread on several slides or pages. The possible options for this dimension are:

- manual size (MS) or manual layout (MLa), in which the user has to edit the size and location of the visualizations,
- imposed on a grid (IG) or a specific layout, or with an imposed size (IS) for each visualization,
- optimized Layout (OL) or optimized size (OS), in which an optimization algorithm is used to compute the layout,
- predefined templates (PT), in which the user has to select a template that fits the given dataset,

- “none", if the system does not support any kind of layout.

### 2.3.6 User feedback

We distinguish two main possibilities for user feedback:

- direct feedback (DF), in which the ADRS suggests choices and then lets the user immediately tell the ADRS if this suggestion is interesting or not,

- behavioral feedback (BF), in which the ADRS can record what choices the user has accepted or rejected, on a longer history, and can use the recorded behavior to improve its suggestions. A typical case of BF consists of collecting and labeling hundreds or thousands of observations so as to form a learning set that can be used to derive statistical properties or to train ML models,

- “none", if the system does not support any kind of user feedback.

### 2.3.7 User interface and user evaluation

The user interface plays a key role in ADRS. Some ADRS provide features such as:

- drag and drop (DD) of visualizations filters for dimensions selection, icons, etc., in the control panel to design or modify suggested visualizations in a dashboard. These features allow users to edit a dashboard,

- selection mechanisms (SM), such as mark type selection, data dimensions, color combination, and global and local filters, at the local (particular visualization) and global (for all the visualization in a dashboard) levels. It makes the dashboard more user-friendly and appealing.

We also summarize in section 2.4.6 the evaluation performed for each interface, and we use the following notations:

- “Sole" when the evaluation includes the studied system only,

- “Comp." when a comparative evaluation with at least one other system is conducted.

### 2.3.8 License

ADRSs have different types of licenses to come into the market: commercial (C), open-source (OS), and freeware (F). If we did not find the license of the system we indicate it as “not found" (NF).

## 2.4 A classification of ADRSs

### 2.4.1 Overview

In Table 2.4, we present the 19 systems included in our survey, with the main values for the 8 studied dimensions. Before analyzing each dimension, we present the main design rationale for each system, starting with the academic work. For the easier understanding of Table 2.4 we have presented the list of acronyms in Table 2.3 used in Table 2.4.

ScagExplorer [Dang & Wilkinson 2014b] is a system that focuses on scatter plots evaluated by utility functions, and that performs global clustering and optimization to produce a MpV visualization that summarizes such plots. DashBot [Da Col *et al.* 2021b] generates a dashboard considering user immediate feedback and reinforcement learning. [Chen *et al.* 2020b] (denoted by CCP in the following for “Composition and Configuration Patterns") uses the combination of a behavior-based approach and statistical knowledge for proposing a dashboard layout. Exploration Views [Elias & Bezerianos 2011b] proposes initial dashboard templates and a rich user interface to help novice users. DynSpace [El Meseery *et al.* 2018b] is a system designed for novice users so as to ease the user exploration process with multiple workspaces. VizDeck [Key *et al.* 2012b] involves the user intervention as direct feedback to choose the vizlet to add to the dashboard from the recommended hand thumbnails, which in turn helps the system to improve a decision tree used for recommendations. MultiVision [Wu *et al.* 2021a] uses two deep learning models, one siamese network for ranking visualizations with their data mapping, and one LSTM for assessing the quality of single or multiple charts. Then a simple incremental heuristic is used to add the chart to the dashboard, maximising the assessment computed by deep neural networks. DeepEye [Luo *et al.* 2018a] performs an exhaustive search of visualizations and uses ML to keep the good ones, and then it defines a partial order on visualizations and selects the top-k to form a dashboard [Database Group 2018]. Draco [Moritz *et al.* 2018b] uses VQL specification for visualization recommendation with the use of logic programming and a constraint solver, but it does not provide a user interface. Likewise, Zenvisage [Siddiqui *et al.* 2016b] also supports VQL for visualization recommendation but unlike Draco, it provides a user interface with different features. Voyager 2 [Wongsuphasawat *et al.* 2017b] is an advanced version of Voyager. Voyager 2 uses the concept of wildcards for visualization recommendations. “Engaging dashboards" [Aksu *et al.* 2019b] suggests the visualizations with a predefined simple decision model established with expert knowledge. Datashot [Wang *et al.* 2019] generates all interesting facts from a data table, each fact being evaluated with utility functions. Then it aggregates facts to form relevant fact sheets, using a simple heuristic and a decision tree to turn facts into visualizations. Calliope [Shi *et al.* 2020] has a similar behavior to generate data stories, but to scan the huge search space of facts and stories, it uses a more advanced optimization algorithm.

There are numerous commercial tools for recommending dashboards, and

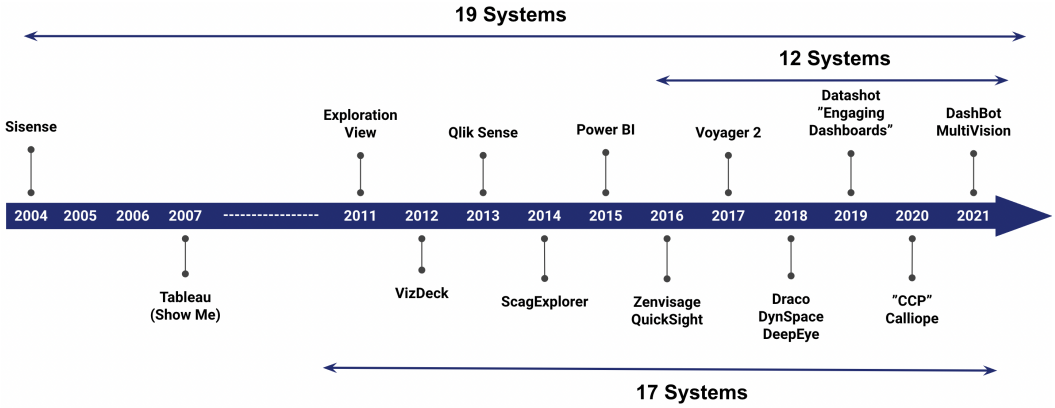



Figure 2.3: The timeline representing the systems studied in our survey.

we selected well-known tools in our survey: Tableau [Mackinlay *et al.* 2007a], Power BI [Microsoft 2015a], Qlik Sense [QlikTech International 2013a], QuickSight [Amazon Web Services 2016b], and Sisense [Sisense Inc 2004a]. They use almost the same approach for visualization recommendation, with a one-by-one suggestion of visualizations. These tools support imposed and manual layout optimization with a lot of customization features in the user interface. More importantly, these systems are designed for expert users.

### 2.4.2 Year

In Figure 2.3, we analyze the year of publication for each system. After the pioneering work of [Mackinlay *et al.* 2007a], this domain is active mainly for a decade. It does not represent so many papers compared to other domains. It has been preceded by many works on single visualization recommendation systems, starting in the 90's. During this last decade, an important contribution has been made every year, with a slight increase in recent years.

### 2.4.3 Included visualizations

Through our analysis, we find that the minimum number of visualizations supported by a system is 1, in the case of ScagExplorer, while the maximum can be hard to evaluate, several tens for commercial software, because of the many possible variants. In this section, we consider the systems ordered by increasing the number of proposed visualizations, and we distinguish two kinds of systems: those with a small or medium number of proposed visualizations (8 or less), and those with a large number of proposed visualizations.

The first group of systems starts with ScagExplorer which supports only one visualization, i.e. scatter plot. This system concentrates on this visualization and is able to perform a precise evaluation of the visualizations with 9 utility functions devoted to scatter plots. Defining such functions for other visualizations could be

possible but was beyond the scope of ScagExplorer. Dynspace supports only 3 visualizations, histogram, pie chart, and scatter plot, because this system is clearly designed for novice users. DeepEye supports the 4 most frequent visualizations, bar chart, line chart, pie chart, and scatter plot because it requires a learning set and expert rules, which take time to establish. It does not specifically mention its target audience, but it can be used by both novices and experts.

Multi-vision supports 5 visualizations, scatter plots, bar chart, line chart, pie chart, and area charts, because these are considered the most commonly used visualizations. DashBot supports many visualizations, some of them are: bar charts, pie charts, scatter plots and tables. VizDeck supports 7 visualizations, i.e. scatter plot, histogram, bar chart, pie chart, time-series plot, line charts and maps, and their implementation as vizlets. Zenvisage supports bar charts, scatter plots, heat maps, histograms and line charts because this system does not have yet a complete user interface. In [Elias & Bezerianos 2011b], the number of visualizations supported by Exploration Views is not clearly mentioned, but it seems to be around 8. Exploration views is intended to be used by novice users and therefore it integrate common visualizations only.

The second group consists of systems that can propose many visualizations (>8). Voyager 2 supports many visualizations, i.e. histogram, scatter plot, line chart, and mark types, by making use of and generalizing the vega-lite specifications, as well as Draco. The number of visualizations in such systems increases the effort needed to update the knowledge base. “Engaging Dashboards" supports multiple visualization types: bullet graph, heat map, highlighted table, dot plot, sparkline, line graph, slope graph, stacked bar graph, variance graph, boxplot, histogram, frequency polygon, and spatial map. It is designed to propose the most frequent visualizations used in dashboards. CCP [Chen *et al.* 2020b] supports 14 visualizations, 12 types of information visualization as suggested in [Borkin *et al.* 2013], plus SciVis and Panel. DataShot and Calliope support many visualizations especially those used for storytelling.

Table 2.3: Different acronyms and their meaning.

| *Acronyms* | Meaning | *Acronyms* | Meaning |
|---|---|---|---|
| CCP | “Composition and Configuration Patterns" | ML | Machine Learning |
| SP | scatter plot | SiD | Single and dependent |
| HT | histogram | T | Predefined dashboard templates |
| AC | Area Chart | DM | Direct Matching |
| GC | gantt chart | UIAs | User’s Initial Actions |
| RC | Radar Chart | VQL | Visualization Query Language |
| G | Gauge | SI | Single and Independent |
| FP | Frequency Polygon | IN | Insights |
| T | Table | NF | Not Found |
| BC | Bar Chart | OL | Optimized Layout |
| PC | Pie Chart | OS | Optimized Size |
| LC | Line Chart | IS | Imposed Size |
| M | Map | IG | Imposed on a Grid |
| MT | Mark Type | MS | Manual size |
| MV | Many Visualizations | MLa | Manual Layout |
| GO | Global Optimization | DF | Direct Feedback |
| BB | Behavior-Based | BF | Behavioral Feedback |
| SD | Statistical Properties of the Dataset | SM | Selection Mechanism |
| DD | Drag and Drop | C | Commercial |
| OS | Open Source | F | Freeware |

Table 2.4: Overview of the studied ADRSs.

| System | Year and Reference | Considered vis. | Suggestion method | Utility functions | Layout | User feedback | User Interface | License |
|---|---|---|---|---|---|---|---|---|
| ScagExplorer | 2014 [Dang & Wilkinson 2014b] | SP (1) | SD, GO | 9, IN | OL, OS | None | SM | NF |
| DashBot | 2021 [Da Col *et al.* 2021b] | T, BC, SP, PC (4) | BB, SD, ML, SiD | 3, IN | IS, IG | DF | SM | NF |
| CCP | 2020 [Chen *et al.* 2020b] | T, LC, BC, M, MV (14) | BB, T | None | PT, MS, ML | BF | DD, SM | NF |
| Exploration Views | 2011 [Elias & Bezerianos 2011b] | PC, BC, T, MV (NF) | T, DM | None | PT, MS, MLa | None | DD, SM | C |
| DynSpace | 2018 [El Meseery *et al.* 2018b] | HT, BC, SP (3) | DM | None | IS, IG, MS, MLa | None | DD, SM | NF |
| VizDeck | 2012 [Key *et al.* 2012b] | HT, BC, SP, LC, PC, M (7) | SD, BB, ML | None | IS, IG | DF | SM | C |
| MultiVision | 2021 [Wu *et al.* 2021a] | SP, LC, BC, AC, PC (5) | BB, ML, SiD | 4 | IS, IG, MS, MLa | None | DD, SM | OS |
| Tableau (Show Me) | 2007 [Mackinlay *et al.* 2007a] | T, BC, SP, GC, MV (14) | UIAs, SI | (2) NF | IS, IG, MS, MLa | None | DD, SM | C |
| Power BI | 2015 [Microsoft 2015a] | LC, T, BC, PC, M, SP, MV (21) | UIAs, SI | NF | IS, IG, MS, MLa | None | DD, SM | C |
| DeepEye | 2018 [Luo *et al.* 2018a] | BC, LC, PC, SP (4) | BB, ML | 3, IN | IS, IG | None | SM | F |
| Draco | 2018 [Moritz *et al.* 2018b] | MT (NF) | VQL, ML | None | None | None | None | C |
| Zenvisage | 2016 [Siddiqui *et al.* 2016b] | HM, SP, HT, LC, MV (NF) | VQL | None | IS, IG | None | DD, SM | OS |
| Qlik Sense | 2013 [QlikTech International 2013a] | LC, G, PC, PT, HT, MV (NF) | UIAs | NF | IS, IG, MS, MLa | None | DD, SM | C |
| Voyager 2 | 2017 [Wongsuphasawat *et al.* 2017b] | HT, SP, LC, MT, MV (NF) | VQL | 2 | IS, IG, MS, MLa | None | DD, SM | C |
| QuickSight | 2016 [Amazon Web Services 2016b] | BC, T, SP, HT, MV (NF) | UIAs | NF | IS, IG, MS, MLa | None | DD, SM | C |
| Sisense | 2004 [Sisense Inc 2004a] | SP, PC, HT, LC, MV (NF) | UIAs | NF | IS, IG, MS, MLa | None | DD, SM | C |
| "Engaging Dashboards" | 2019 [Aksu *et al.* 2019b] | SP, HT, HM, FP, MV (13) | DM | None | IS, IG | None | SM | C |
| Datashot | 2019 [Diamond & Mattia 2017] | BC, AC, M, SD, PC (14) | BB, SD, ML | 2, IN | IS, IG | None | SM | OS |
| Calliope | 2020 [Zeng *et al.* 2021] | BC, LC, PC, AC, BP (12) | BB, SD, GO | 1, IN | IS, IG | None | SM | OS |

Commercial tools studied in this paper, namely Tableau, PowerBI, Qlik Sense, Sisense, and Quicksight, form a sub-group in this category. They propose many visualizations. Users of such systems are BI experts, data analysts, and researchers, who might have a high level in data visualization, so they can work with complex visualization in addition to common ones. Another reason is that such systems charge users for their services, and in return, users expect to benefit from more advanced features and services.

In the first group, systems are more focused on a limited set of standard visualizations. Their design is more oriented towards novice users, who can more easily understand common visualizations than others. In addition, these systems might not have a fully developed user interface so the number of proposed visualizations is limited. In the second group, systems include common visualizations but also advanced visualizations that experts can understand and configure. They address different users. The inclusion of complex visualizations also requires an important effort of design and development that other academic works cannot afford.

### 2.4.4 Suggestion mechanisms

Here we discuss the automatic or semi-automatic algorithms and methods used for the recommendation of dashboards as mentioned in Table 2.4, ordered by increasing complexity of suggestion mechanisms (see Section 2.3.3):

**Simple suggestion mechanisms** The systems, in this category, use simple predefined suggestion methods, like direct matching or predefined templates. DynSpace [El Meseery *et al.* 2018b] focused on the management of multiple workspaces to ease the user exploration process. DynSpace uses SD and DM to propose visualizations and mapping, with simple heuristics: “histograms are used for single dimensions, scatter plots for two numerical dimensions, and bar charts are created when one dimension is categorical and the second is numerical". Once the user has selected a data attribute, DynSpace automatically provides visualizations for this dimension, according to the DM simple strategy.

“Engaging Dashboards" [Aksu *et al.* 2019b] uses a similar strategy but slightly more advanced: a simple predefined decision model for automatic visualization recommendations is used. This model for KPIs visualization is established from experts’ recommendations. It takes the form of a decision tree, in which leaves are visualizations. Then, considering a description of KPIs, it generates dashboards automatically using the decision model.

Exploration Views [Elias & Bezerianos 2011b] also uses DM but in a different context. Initially, Exploration Views proposes predefined dashboard templates to users. Users can choose one of the templates from the set of given templates based on their domain knowledge and previous experience. Then users can customize the initial dashboard by selecting data and visualization. DM is used with a simple heuristic to provide data mapping. The user interface has interesting features for novice users, like undo or visual feedback.

To summarize, in this category, the target users are novices. The suggestion mechanism has to be autonomous without requiring any complex actions from users. Furthermore, except for “Engaging dashboards" which do not seem to have any user interface, DynSpace and Exploration views have rich user interfaces with many specific features for novice users. These features compensate for the simplicity of the suggestion method and the limited number of proposed visualizations.

**Advanced suggestion mechanisms** These systems use more complex statistical models and methods involving statistical or knowledge-based approaches. CCP [Chen *et al.* 2020b] uses a BB approach. Authors have analyzed visualizations layouts proposed in many MpV visualization papers. They have created co-occurrence matrices of visualizations that appear often together in the same MpV interface, as well as representative layout templates with their associated frequency. Then, they use this statistical knowledge to help users determine the content and layout of their dashboard.

ScagExplorer is designed to explore data through a huge collection of scatter plots. ScagExplorer is based on two suggestion mechanisms: GO and SD. The system generates all possible scatter plots from the data and evaluates each of them with nine scagnostics measures which are utility functions. Then ScagExplorer uses a clustering approach to produce groups of scatter plots. Such a group is a list of plots with a leader that is used as a group prototype. To visualize the clusters, a forced-directed layout technique is used on leader scatter plots. About the interaction, users can select a leader to see all similar plots in that cluster or filter scatter plots by their features. During this process, the system uses statistical properties, i.e. density distribution, for filtering scatter plots.

Several systems are using visual query languages to search and propose visualizations. Voyager 2 is an advanced version of the original Voyager [Wongsuphasawat *et al.* 2015] with a set of two new important extensions, automating visualization suggestions and avoiding redundant suggestions, as discussed in [Wongsuphasawat *et al.* 2017b]. In Voyager 2, automatic visualization suggestions are supported by the wildcards concept and CompassQL [Wongsuphasawat *et al.* 2016] query. Wildcards in the dashboard panel allow users to change the properties of a specification given in the CompassQL query, thus letting the system suggest visualizations. It uses two standard utility functions, expressiveness, and effectiveness, to rank the proposed visualizations. Zenvisage uses VQL to specify the desired visual patterns to be present in the visualizations. Zenvisage query language lets the user formulate a query, and then the system returns visualizations. Using a small number of queries, users can explore trends, patterns, and insights of the given data.

In this category of systems, one subgroup is formed with commercial software, namely Tableau, Power BI, Qlik Sense, QuickSight, and Sisense. In general, the user starts by selecting data of interest. Then these systems recommend visualizations for the considered data. They use a one-by-one strategy to suggest visualizations, and

it seems that they do not take into account the current state of the dashboard and the previously selected visualizations. This one-by-one strategy is highly guided by the user who must make crucial decisions. Visualizations can be deeply customized within a rich interface, both from the visual and interactive points of view. So such systems require in general skilled users. In addition, they do not disclose how their suggestion mechanism works.

**Complex suggestion mechanisms** These are systems that use ML-based suggestion mechanisms. The general idea in such systems is to acquire expert knowledge about visualization and dashboards through the learning of a model trained on empirical visualization data. Such data can be for instance the labeling of existing visualizations or the comparison of several couples of visualizations with the selection of the best one for each couple. This train set can be augmented as users interact with the system.

Draco [Moritz *et al.* 2018b] is based on a VQL used to query a knowledge-based model with constraints, hard and soft. Draco compiles a user's query including the dataset, the partial specification, and the task, into a set of facts and combines them with the existing knowledge base in the Answer Set Programming framework. Draco then calls a solver (.clingo) to resolve the constraints and obtain a feasible answer set. Finally, Draco translates the answer set into a Vega-lite specification. To establish the weights associated with soft constraints, Draco uses a support vector machine to learn such weights from empirical data, with a learning-to-rank approach that is empirically more efficient than the use of expert weights.

DashBot [Da Col *et al.* 2021b] generates a dashboard on a one-by-one visualization basis using reinforcement learning. DashBot selects the most relevant panel/visualization, i.e. data attributes, using SD (i.e. entropy, variance, and coverage). It takes into account the panels already selected in the dashboard. The user can provide immediate feedback, such as "yes" in which case the panel is added to the dashboard, or "no with explanations", in which the user can tune the panel, or "no without explanations" in which case a reinforcement learning algorithm is used to generate a new panel. DashBot supports two strategies, $\varepsilon$-greedy, and soft-max, each of them providing different exploitation and exploration trade-offs for panel generation.

VizDeck [Key *et al.* 2012b] uses the combination of two concepts, statistical function and vizlet model. A vizlet is defined as a 4-tuple that includes two render functions (visualization producing thumbnails and interactive version of the vizlet), a predicate (indicating compatibility with dataset), and a scoring function (one of several signals used for ranking). For recommendation purposes, a decision tree is trained to relate statistical properties, like distinct values, entropy, coefficient of variation, kurtosis, or periodicity, to a particular vizlet. User actions of promoting or discarding vizlet allow the system to improve its learned model by collecting positive and negative examples. VizDeck involves also the use of heuristics based on human perception and these heuristics control vizlets recommendations.

DataShot [Wang *et al.* 2019] uses a large history of visualization usage in spreadsheets to learn what models are often used and in what conditions. It derives also a decision tree from this model to determine what visualizations to use for a given data fact. It uses an exhaustive search of facts, followed by a clustering/aggregation approach that takes into account the topic covered by facts. Then it generates the corresponding fact sheet with the use of the decision tree to turn facts into visualizations.

Calliope [Shi *et al.* 2020] also uses as a starting point a large database of storytelling videos that have been manually analyzed and annotated. It does not scan the whole search space but it uses rather a tree-based optimization method to select to most relevant facts and their organization into a story (nodes of the tree are facts, a story being a path from root to leaf). It provides a rich user interface to edit stories.

DeepEye proposes the use of two ML models. First, it generates all possible visualizations, and a first decision is made to discard “bad" visualizations and to keep only the “good" ones. This ML model is trained on a learning set where users have established a ground truth. Then, a partial order is established between the selected visualizations. A heuristic based on expert knowledge, or alternatively a learning-to-rank approach, is used to select the top-k visualizations that can be proposed to the user. In the evaluation of these two methodologies, authors found the first mechanism based on expert knowledge to perform better than the second one.

MultiVision [Wu *et al.* 2021a] learns deep learning models to guide a recommender algorithm. The deep neural network are trained with labeled data or data from the logs of the system. These networks can assess the quality of a visualization and its data mapping, or the quality of a MV visualization. The recommender is a simple heuristic that adds visualization one by one so as to maximize the evaluation provided by the deep networks.

### 2.4.5 Utility functions

Systems that integrate utility functions use either advanced or complex suggestion mechanisms for visualization generation. This is due to the fact that utility functions are used to automate the evaluation of the generated visualizations, opening the possibility to use optimization or ML algorithms. Hence such functions are involved in either completely automatic systems like ScagExplorer, MultiVision, DeepEye, DataShot, and Calliope or systems that require minimal human intervention like Tableau or Voyager 2. DashBot, which is a more interactive system than the previous ones, uses utility functions to improve the suggestions proposed to the user. The literature for commercial systems [QlikTech International 2013a, Amazon Web Services 2016b, Sisense Inc 2004a, Microsoft 2015a] does not reveal anything about the use of utility functions, and the other remaining systems do not seem to use utility functions at all.

A special case of feedback is immediate feedback, but as far as we know only

Dashbot really implements this strategy. Hence, this system is able to take into account the user's advice given to the system. A second type of feedback is the behavioral feedback. Here a large usage history is considered to learn or improve an ML model. A slight distinction can be made between VizDeck or Multivision and the other systems. In VizDeck, the systems learn from the interaction between the user and the suggestion strategy. Recommendations that are kept (respectively not kept) by the user are considered as positive (respectively negative) examples that can be added to a training set. The system strategy can be regularly updated and could be adapted to the current user. Other systems consider the large usage history of external systems (e.g. from Excel), in order to learn knowledge about visualization design and efficiency. Here the behavioral feedback is obtained from other systems than the current recommendation strategy. Hence adaptation is not specific to the current users.

### 2.4.6 User interfaces and evaluation

In this section, we have ranked the studied systems from the simplest to the most complex user interface. We start with Draco because it does not provide any interface, but it outputs a Vega-lite code. Then some systems do not provide any editing interface, but instead a direct visualization of the suggested dashboard, as in "Engaging Dashboards". Zenvisage and Voyager 2 provide minimal user interfaces.

Then a second group of systems provides advanced user interfaces, often with a focus on specific features. Dynspace concentrates on multiple workspaces and other features for novice users, as does Exploration Views. Other systems provide rich user interfaces with detailed views of the proposed dashboards (ScagExplorer, MultiVision, DeepEye, DataShot, Calliope), advanced interactions devoted to user feedback (DashBot, VizDeck) or interactive options selection (CCP).

Finally, commercial systems [Mackinlay *et al.* 2007a, Gowthami & Kumar 2017, QlikTech International 2013a] support the most advanced user interfaces with many customization features compared to other scientific tools. These commercial tools propose a highly versatile dashboard interface. In return, the user has to be familiar with complex technical terminology and interface setup.

About the evaluation of interfaces, commercial systems do not provide evaluation results, except for Tableau. For Draco, DataShot, and ScagExplorer, no user was involved. The evaluation was a demonstration of functionalities on use cases. For DeepEye, a machine-based evaluation was performed. For the remaining 11 systems, a user evaluation was performed, as summarized in Table 2.5. 7 studies test the functionalities of the systems, while only 4 perform a comparative study involving up to 3 different systems.

### 2.4.7 Market accessibility

Some systems [Dang & Wilkinson 2014b, Da Col *et al.* 2021b, Chen *et al.* 2020b, El Meseery *et al.* 2018b] are not available practically to be tested and used. ADRSs

Table 2.5: A summary of the user evaluations performed in ADRSs. U denotes the number of users involved in the evaluation.

| System | Type | U | Mode | Findings |
|---|---|---|---|---|
| CCP | Sole | 20 | Online | Completion time, Ease of use |
| Expl.View | Sole | 15 | Offline | Ease of use, chart creation and customization |
| DynSpace | Sole | 8 | Offline | Selecting dimensions preferred |
| VizDeck | Comp. | 32 | Offline | Completion time |
| MultiVision | Sole | 12 | | Useful, convenient, easy to learn |
| Zenvisage | Comp. | 12 | Offline | Completion time, accuracy, usability and satisfaction |
| Voyager 2 | Comp. | 16 | Offline | Supports focused analysis, related views + wildcards helpful |
| Eng. Dash. | Sole | 3 | Online | Dashboard improvement |
| Tableau | Sole | 10 | Online | Show Me + Add to Sheet useful |
| DataShot | Sole | 10 | Offline | Effective data fact extraction, visual design, interactions |
| Calliope | Comp. | 16 | Offline | Logicality, memorability, comprehension, engagement, dissemination |

have different types of licenses to come into the market:

- Commercial: systems [Mackinlay *et al.* 2007a, Gowthami & Kumar 2017, Elias & Bezerianos 2011b, QlikTech International 2013a, Aksu *et al.* 2019b, Amazon Web Services 2016b, Sisense Inc 2004a] are commercial, they do not allow users to modify their source code, and users have to pay to use their services,
- Open Source: only two systems [Siddiqui *et al.* 2016b, Wu *et al.* 2021a] are open source. This kind of system allows users to make modifications to their source code for personal customization. MultiVision and Zenvisage share their source code on GitHub. MultiVision is under development,
- Freeware: only one system [Luo *et al.* 2018a] is a freeware that provides its service free of cost and without time constraints. Other commercial systems provide a free trial for a few days with time constraints.

### 2.4.8 Transverse categorization of systems based on methodologies and features

In complement to the previous analysis of descriptive dimensions, we propose in this section different and complementary viewpoints on the analyzed systems.

**Commercial systems** These systems form a group with similar features. They all propose many visualizations and advanced interfaces. They are in general designed for expert users and are more difficult to use by novice users. They generate visualizations in a dashboard with a one-by-one strategy that relies on user choices to select the proposed options. They let the user do a lot of customization in a user interface with many complex features. They do not seem to use immediate feedback and in general, they do not disclose their suggestion method.

**Proposing dashboards templates and layout** Several systems are proposing predefined templates, either according to expert knowledge [Elias & Bezerianos 2011b] or usage history like CCP. This is helpful for users. However, users need some expertise to select the right template that fits the data, and manually edit the dashboard to finalize it. These two limitations are the main difficulties for novices in using this kind of system. In other systems, the layout either relies on user manual editing or is imposed, usually with a grid or tiled format.

**ML and Deep Learning algorithms** Several systems [Da Col *et al.* 2021b, Wu *et al.* 2021a, Luo *et al.* 2018a] use ML methods for visualization recommendation. These systems reduce human efforts and suggest precise visualizations related to data. They avoid formulating complex knowledge bases as in Draco for instance. The authors of Draco also admit this fact as they use a support vector machine to

set up weights that are too difficult to be humanly determined. They also show that human settings are experimentally less efficient. However, these systems involve complex algorithms that do not explicitly acquire knowledge. They can not explain why a suggestion is interesting, or why a visualization is better than another one. They work as black boxes, with an abstraction of the whole process, and users are only aware of input and output. Another drawback is that these systems require high computation costs in the learning phase as well as data acquisition, labeling, and preprocessing.

In our survey, we have only one system [Wu *et al.* 2021a] which uses deep learning, with two different architectures (Siamese network and LSTM). As for other ML approaches, Deep Learning has the same drawbacks and advantages. It is not clear whether the actual quantity of available labeled data is in favor of Deep Learning compared to standard ML. Yet it is difficult to compare Deep Learning and ML algorithms because this would require the same data and evaluation procedure.

**Statistical algorithms** Many systems focus on the statistical properties of the dataset to improve the quality of suggestions. They look for insights, relevant information or common behavior in the data [Dang & Wilkinson 2014b, Da Col *et al.* 2021b, El Meseery *et al.* 2018b, Key *et al.* 2012b]. VizDeck [Key *et al.* 2012b] uses 5 statistical properties to generate top-k visualization. DashBot [Da Col *et al.* 2021b] uses aggregation functions (min, max, count, sum, avg) to compute other statistical properties like entropy and variance in the data prepossessing phase (offline process) of dashboard generation. ScagExplorer [Dang & Wilkinson 2014b] uses statistical properties for filtering the scatter plots. Similar to ML algorithms these systems also involve high computation of datasets to find different statistical properties. The algorithms behind calculating SD are not as complex as those of ML methods.

**Query based systems** Several systems [Da Col *et al.* 2021b, Luo *et al.* 2018a, Moritz *et al.* 2018b, Siddiqui *et al.* 2016b] use knowledge-based approaches with query languages. Those systems, in contrast to ML-based approaches, have an explicit representation of knowledge. Formulating such knowledge can only be done by experts, it requires a lot of work, and such knowledge bases might be difficult to maintain. However, once established and validated, it can be used to propose visualizations and to explain the results. Such systems do not require high computation to acquire knowledge, however, the inference they perform might require a long time.

**Completely automatic systems** Complete automation of dashboard creation has two main advantages: for novice users, it is easier to obtain a complete dashboard than to edit it, a task that requires knowledge of visualizations and an understanding of the user interface. Another advantage of such an approach is that it can cope with the huge search space involved in dashboard creation. However,

such systems are not so numerous, even if their presence has increased over recent years: ScagExplorer [Dang & Wilkinson 2014b], "Engaging dashboards", DeepEye [Luo *et al.* 2018a], DataShot [Wang *et al.* 2019] or Calliope [Shi *et al.* 2020] support complete automation for dashboard generation. These systems only ask users to upload a dataset as an input and generate dashboards as the output. These systems generate, recommend, and optimize the dashboard on the basis of the dimensions and attributes of the dataset. It is interesting to see that these systems however do not support yet user interfaces with features like drag and drop, resize, move, add, and delete the visualization. During the whole process of dashboard recommendation, these systems do not allow any kind of user intervention.

**User technical skills** In the evaluation section of the studied papers [Elias & Bezerianos 2011b, El Meseery *et al.* 2018b, Luo *et al.* 2018a], are defined different levels (beginner, advanced, expert) of expertise for different users (novice and expert) to understand BI tools. Technical skills (in data analysis, data modeling, visualizations pros and cons, terminology) possessed by users help them to decide their preference for a given system. [Elias & Bezerianos 2011b] state that it is difficult for novices to understand the technical terminology used in the system interface to define different elements and components of the dashboard. On the other hand, experts are familiar with these technical terms. Similarly, [Saket *et al.* 2018] states that after mathematical experiment and calculation, authors find that a scatter plot is the best option to find anomalies in data because of its good performance (in terms of time, precision, and accuracy). But users equally preferred bar charts instead of scatter plots for the same task because users are more familiar with bar charts and they can easily interpret bar charts in contrast to scatter plots. Systems [Elias & Bezerianos 2011b, El Meseery *et al.* 2018b] are more preferred by novice users because their terminology is easier to understand in comparison to commercial systems [Mackinlay *et al.* 2007a, Gowthami & Kumar 2017, QlikTech International 2013a] and open-source systems [Wu *et al.* 2021a] whose major users are experts and researchers.

## 2.5 Discussion

### 2.5.1 About this survey

In this state-of-the-art, we have focused on automatic dashboard recommendation systems. This survey contributes to researchers and BI experts, acting as a resource and guideline for the design and development of ADRSs. It also helps readers to categorize the level of considered visualizations and complexity of suggestions methods used in ADRSs.

We found only 19 ADRSs starting from a pool of 1036 papers. It shows the scarcity of such systems in the field of data visualization. We identify 8 dimensions to analyze these systems. The results of this analysis show that there are many challenges that could be addressed in the future of ADRS.

### 2.5.2 Analysis of system launched year

Regarding previous surveys [Shen *et al.* 2021, Zhu *et al.* 2020] which showed the evolution of automatic visualization recommendation systems (from 2001 to 2021 and from 2012 to 2020 respectively) and with our analysis presented in Table 2.4, we can conclude that in the last decade (2011-2021) there is an increase in the number of ADRS. In addition to this growth, there is also a transformation in technology used to develop such ADRSs, from simple and manual to advanced and automatic suggestion mechanisms respectively, with the use of ML. It attracts researchers, data scientists, and business intelligence experts to develop new and advanced ADRSs. Based on this fact, we think that the work in this domain will continue to increase over the years.

### 2.5.3 No dominant methodology and lack of experimental comparison

Various methodologies are used in ADRS. Indeed ML, Deep Learning, and Statistics are the most used and recent trends for developing ADRSs. However, in our opinion, knowledge-based approaches are still competitive to other ML models, and this will continue as long as collected data about visualization usage is not informative and voluminous enough, and as long as explanations of ML models will not be clear for users.

In addition, it is hard to determine which approach is the best. Yet the only possibility to compare these systems is to study their evaluation methodology and obtain results, but due to different users, data, and hardware, this could be biased. Every system uses different evaluation methodologies and dimensions to evaluate its performance and user experience. The most valuable results of the evaluation are found through users' feedback. However, it would be more beneficial to evaluate different ADRSs with the same datasets, tasks, participants, and hardware. In other words, to compare different systems' performances, there is a need to perform experiments with all systems in similar conditions, including quantitative results and users' feedback. Zeng et al. [Zeng *et al.* 2021] have proposed an evaluation-focused framework to contextualize and compare visualization recommendation algorithms. They have demonstrated the proposed framework with 5 visualization recommendation systems only. Alternatively, it would be interesting to develop a testing platform that could use the same grounds for comparing all systems.

So there is no approach that dominates the others, and studied systems are in a way complementary. This shows that we are still at an early stage of the research about ADRS.

### 2.5.4 Novice users

Recommendation systems are useful for all users, but they play a more important role for novice users who need more guidance and help than for others. System users (novices/experts) and the terminology of the system directly affect

each other. If users of the system are experts, in such a case technical words used in the dashboard interface are fine, like in the case of commercial systems [Mackinlay *et al.* 2007a, Gowthami & Kumar 2017, QlikTech International 2013a]. But on the other hand, if users are novices in this case it can be very difficult for them to deal with the dashboard interface. [Elias & Bezerianos 2011b] highlighted this issue during the evaluation through user feedback after the experiment. In addition, systems that propose a complete dashboard without user intervention can be interesting for novices who will not have to make difficult decisions. On the other hand, expert users are more likely to make good suggestions to recommendation systems through user feedback or manual editing.

### 2.5.5 Dashboard interface

Dashboards can be extended and improved with specific visual and interactive features. Such possibilities are a bonus for users because they help them understand the data story quickly and improve user interactions with the system.

Linking and brushing are common techniques in many systems, but additional and more specific features can be found, like dashboard annotation, animation of visualizations, color combination, drag and drop, local and global filters, undo/redo, and visualization caption. [L'Yi *et al.* 2019] explains and compares how the effects of annotation and animation help users understand visualization quickly. [Yuan *et al.* 2021, Few 2003] use a selection of the right color pallet for different view types (widgets), which also helps users to quickly understand insights of the dashboard. We observe that the commercial dashboard interface supports many such features (pulse feature in Sisense, narrate feature in Qlik Sense, story feature in Tableau, etc) compared to other systems of Table 2.4. Hence the dashboards designed with commercial tools are very appealing.

### 2.5.6 User feedback

The integration of user feedback is still rare but it is a major challenge for future ADRSs. Only 4 systems [Da Col *et al.* 2021b, Chen *et al.* 2020b, Key *et al.* 2012b, Aksu *et al.* 2019b] support user feedback. More specifically, the most popular commercial systems (i.e. Tableau, Power BI, Sisense, Qlik Sense, QuickSight) do not propose user feedback features. In addition, comparative evaluation of user feedback methods is difficult, because of the different testing conditions of each ADRS.

User feedback, either immediate or from logs recording visualizations usage, represents an important feature to help systems generate better suggestions. Immediate user feedback involves the users actively during the process of visualization recommendation and it can help the system to dynamically understand the user's aim and objectives. It can improve the quality of suggested visualizations. User feedback based on large usage history concerns any kind of users and domains. It would be interesting to collect richer data and more information about users and their context. The collection of such enriched data will open the way to more efficient ML

models, that will not be general to all users but more specific to some users. It could be interesting to know about user context and level to make suggestions that are better personalized.

From this point of view, reinforcement learning could play an important role because it is a learning paradigm that can take into account immediate feedback. Also, another paradigm such as semi-supervised learning or active learning could be useful to personalize models to a given user.

### 2.5.7 Lack of open source and freeware ADRS

There are more commercial ADRSs in comparison to open-source and freeware systems. Indeed ADRSs attract researchers but also major investors and enterprises. However, there is still a scarcity of open-source and freeware systems. Due to this scarcity, common users are deprived of analyzing household data using ADRSs, as these common users do not want to use enterprise versions (commercial systems) and invest money to perform their domestic tasks (electricity bills, daily sales analysis for street vendors, monthly expenses, etc). Even in the last decade, we have observed that there is less focus on open source and freeware ADRS development in contrast to commercial tools.

### 2.5.8 Thesis goals and objective

During this survey, we found a huge gap between current needs and existing ADRSs. To fulfill the gap we come up with the following goals:

1. Support Novice users: Most of the existing ADRSs are not specifically designed for novices' uses. Only Exploration View and CCP are systems designed for novice users. But they do not meet our other thesis goals. Existing ADRSs like PowerBI, Tableau, Qlik Sesne, QuickSight, MultiVision, and Sisense, are quite complex to use for novice users and even for users who do not have a computer science background. Our goal is to design a simple system that can be easily used by novices.

2. Support other user types (beginner and expert): All the existing systems like PowerBI, Tableau, MultiVision, Qlik Sense, and ScagExplorer are either for beginners or experts. Our objective is to design a system that supports all three types of users: novices, beginners, and experts. Our major focus will be on novices.

3. Provide simple visualizations: Most of the ADGSs like DeepEye, Datashot, and Calliope contain complicated visualizations in the Dashboard. In addition to them more specifically commercial tools like Tableau, PowerBI, and QlikSesne have more complex visualizations. These visualizations are very difficult for novice and beginner users to understand. Our goal is to keep a simple visualization that can be easily understood by novices.

4. Provide a simple terminologies: Most of the ADGSs (like Tableau, PowerBI, and other commercial tools) consist of technical keywords and difficult terminologies. These keywords and terminologies create a hindrance for novices and beginners users to understand and use the tool. Our goal is to bring simplicity to it using simple keywords with layman's terminologies.

5. Need of fast ADRS: Most of the existing ADRSs like Dashbot, ScagExplorer, VizDeck, and "Engaging Dashboards" take too much time to generate and recommend the dashboard. Our goal is to reduce this waiting time for users.

6. Completely automatic: There is a need for an ADRS that can generate and recommend the dashboard automatically without any user intervention except for the data upload. Systems like MultiVision, CCP, Exploration Views, DynSpace, Draco, and Zenvisage are semi-automatic. Our goal is to recommend the dashboard using certain algorithms with no user intervention in just one click after uploading the data.

7. Problem of high level manual editing: Most of the ADRSs like MultiVision, DynSpace, Exploration Views, DashBot, VizDeck, CCP, and all commercial tools support high-level manual editing. However, our goal is to minimize manual editing. If there are any manual editing possibilities in a system that should be optional so that even novices can generate the dashboard without taking manual editing into account.

8. User feedback: We found some of the systems that track user history, and previous steps and take users' inputs for better and expected dashboard generation. However, there is a need for a rating feature for the dashboard so that the system can measure user satisfaction levels to meet user expectations.

9. Problem of interactive user interface: ADGSs like DeepEye, and "Engaging dashboards" do not allow the user to interact with the interface to give their input for tailored dashboard recommendations. Systems like Draco and Zenvisage even do not have a user interface for users to interact.

10. Problem of complicated user interface and functionality: ADRSs which are commercial have very complicated user interfaces. Users like novices and beginners are lost in complex interfaces and difficult functionalities. These users were not able to find the functionalities options to implement them. Our goal is to create a simple interface and easy functionality.

We did not find any ADGS out of all 19 existing systems covered in the survey. None of them fulfill the above needs of novice users. So this thesis has the above 11 goals for an automatic dashboard recommendation tool.

Chapter 3

# Genetic algorithms

## Contents


**3.1 Model and Schema . . . . . 38**
3.1.1 Overview of our system . . . . . 38
3.1.2 Reasons for the selection of a genetic algorithm . . . . . 39
3.1.3 Data model . . . . . 40
3.1.4 User model . . . . . 40
3.1.5 Visualization model . . . . . 41
3.1.6 Data and visual attribute type mapping . . . . . 41
3.1.7 Data to visualization mapping . . . . . 43
**3.2 Genetic Algorithm . . . . . 44**
3.2.1 Overview . . . . . 44
3.2.2 Dashboard genetic representation . . . . . 46
3.2.3 Random generation . . . . . 47
3.2.4 Genetic operators . . . . . 51
3.2.5 Fitness function . . . . . 58
**3.3 Results . . . . . 62**
3.3.1 Experimental set-up . . . . . 62
3.3.2 Parameter study . . . . . 63
3.3.3 Examples of generated dashboards . . . . . 63


In this Chapter 3, we present a method for the automatic generation of dashboards (DBo) using a genetic algorithm (GA). The chapter explains the rationales for the selection of algorithms and fitness functions. Our Automatic Dashboard Generation System considers several input models for data, user, and visualizations. We propose to represent a solution (i.e. a DBo) as a variable size matrix in which rows are visualizations and columns are data attributes. This representation can be evolved with a GA. For this purpose, we define genetic operators for DBo such as random generation, crossover, and mutation. We propose a fitness function to evaluate the quality of a DBo, as well as selection and replacement schemes. Finally, we present results with a benchmark dataset and a given user scenario. We show that the GA can find DBo that maximizes the evaluation function and that can be interesting for users (novices). We also present improvements made to different GA

operations and fitness functions to get an optimized dashboard. We conclude the chapter by proposing automatic, a layout of the optimized DBo.

## 3.1 Model and Schema

### 3.1.1 Overview of our system

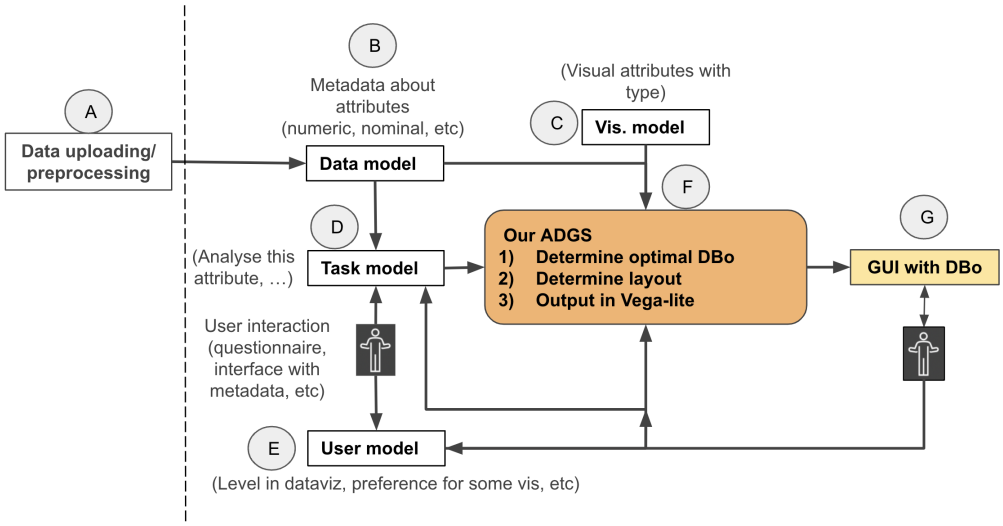



Figure 3.1: Our system takes input for (A) Preprocessed data file and four models: (B) Data Model, (C) Visulization Model, (D) Task Model, (E) User Model. With a GA, it produces an (F) optimized DBo, determines layout, and visualizes output with Vega-lite. A user can interact with the (G) user interface to recommend DBos and also regenerate DBos with modifications in the user and task model.

Our system consists of several main components (see Figure 3.1):

- Computational models for representing data (metadata), user (skills and task to be solved), and visualizations (mainly a list of visual attributes with their visual types),
- A representation of dashboards, and more precisely, for a given dashboard, the set of visualizations it includes, and for each of these visualizations, the mapping between the visual attributes and the data attributes.
- An evaluation function $F(DBo)$ that can numerically evaluate the usefulness of a dashboard,
- An optimization method based on a GA to produce good dashboards, i.e., dashboards that maximize the evaluation function. Here we preferred to implement a heuristic algorithm instead of a greedy algorithm.

### 3.1.2 Reasons for the selection of a genetic algorithm

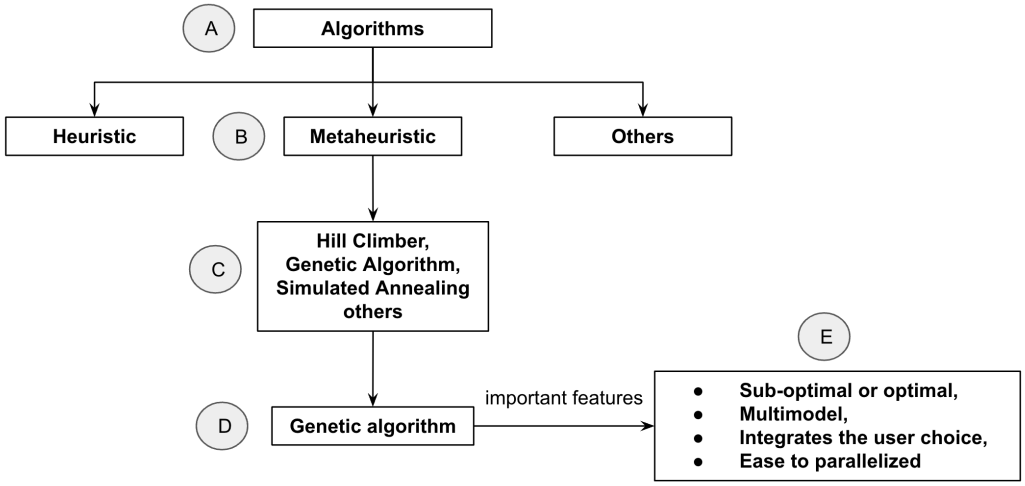



Figure 3.2: Schema for selection of Genetic Algorithm(A) Algorithms (B) metaheuristics, (C) Several types of metaheuristics, (D) Genetic Algorithm, (E) Optimisation algorithms important features.

We model our problem of finding a relevant DBo as an optimization problem. We consider:

- A search space $S$ of DBo's in which optimal DBo must be found,
- An objective function $F(DBo)$ to be optimized,

There are several methods for finding in $S$ the DBo that maximizes $F(DBo)$. Heuristic implements a local algorithm that makes use of specific knowledge about the problem. They are fast but sub-optimal and devoted to the problem (not possible with respect to $F(DBo)$). While on the other side metaheuristics are general methods that are not specialized to a given problem, and therefore, is it preferable to use metaheuristic.

We selected the GA because it meets with the following conditions as follows:

- Sub-optimal or optimal: It usually yields sub-optimal or optimal results,
- Multimodal: It provides multiple solutions not only the optimum one,
- Integrates the user choice: Users can "seed" the population by an existing solution to integrate their choice into the optimum solution,
- Ease to parallelized: Several F can be computed during the same run with different generations.

In the metaheuristic algorithms we come up with several algorithms: hill climbing, genetic algorithm, simulated annealing, and others. The classification of metaheuristics is shown in [From Wikipedia, the free encyclopedia ] with Euler diagram.

We discarded the hill climber because of local minima and maxima issues. We wanted to optimize each individual globally across the existing population and this requirement was achieved by genetic algorithm. As GA computation can be distributed instead of single solution to get much better results and better performance we preferred GA over simulated annealing. The GA crossover operation gives an optimized solution by hybridization of two parents, this is also one of the important reasons for using GA as an optimization algorithm.

To explain our proposal, we use in the following the "students' academic performance dataset (SAPS)" [Pauwels *et al.* 2009, Schwendimann *et al.* 2016]. SAPS contains 17 columns (gender, NationalITy, PlaceofBirth, StageID, GradeID, SectionID, Topic, Semester, Relation, raisedhands, VisITedResources, AnnouncementsView, Discussion, ParentAnsweringSurvey, ParentschoolSatisfaction, StudentAbsenceDays, and Class) and 480 rows.

### 3.1.3 Data model

| DA_name | gender | NationalITy | PlaceofBirth | ..... | AnnouncementsView | Discussion | Class |
|---|---|---|---|---|---|---|---|
| DA_type | Nominal | Nominal | Nominal | ..... | Numeric | Numeric | Ordinal |
| unique_value | 3 | 14 | 14 | ..... | 88 | 90 | 3 |
| user_imp | 50 | 50 | 10 | ..... | 100 | 100 | 100 |

Figure 3.3: Data Model for the SAPS dataset: each data attribute has a name, a type, a number of distinct values, and user importance.

The objective of data model is to extract and represent standard metadata about data attributes, as shown in Figure 3.3. It consists of the following information for each attribute: name, type (numeric, ordinal, nominal, and temporal), number of distinct values (some visualizations might not support too many values), and user importance.

### 3.1.4 User model

The user model serves to represent different user categories, objectives, and preferences. We have identified three user categories based on their skills and needs: novice, beginner, and expert. The user type determines the optimization function and visualization complexity to align with their specific requirements.

#### 3.1.4.1 User's skills

To represent the user's skills in data visualization, we consider three categories novice, beginner, and expert. For each of them, we propose a set of visualizations. For novices, we consider three simple visualizations: histogram, piechart, and simple bar. For beginners, we consider five visualizations histogram, piechart, simple bar, line chart, and scatterplot. For experts, we consider all six visualizations histogram, piechart, simple bar, line chart, scatterplot, and bar chart. The user can augment or

reduce the list of candidate visualizations to personalize this list. Then, to build the DBo, the GA (see Section 3.2) will only pick visualizations from the list approved by the user.

#### 3.1.4.2 User's task

The objective of the task model is to understand as well as possible the user's objective with respect to the data. We have created a list of questions to be asked to users:

- The number of visualizations in the DBo can be between 2 and 9. The system will pick any value randomly between this range (both bounds are inclusive) to generate the dashboard.
- Task to be solved: the user indicates an importance score between 0 to 100 to each data attributes. There are six possible options for selecting user importance value: Exclude (0), Less Importance (25), average (50), More Importance (75), Mandatory (100), and manually (0 to 100). In the case of manual values users have the possibility to assign any integer value between 0 to 100. If the user skips this step, by default the importance value is 50 for each data attribute. These importance values will guide the system towards interesting data attributes that correspond to the user's analysis tasks.

Users may skip questions, in that case, predefined default values are proposed.

### 3.1.5 Visualization model

To represent a single visualization (see Figure 3.4), we consider a list of visual attributes. For each such attribute, we define its type (position, length/height, angle, slope, ...), and the type of data attribute it can be matched to. This information will be useful when the system will try to match a visualization with the data (see figure 3.9). The value "1" in *mandatory* column of figure 3.4 shows that the mapping of that particular visual attribute with respected data attribute is compulsory for the feasibility of respected visualization. If the value is "0" in *mandatory* column in that case we can skip the mapping of that visual attribute for respected visualization.

### 3.1.6 Data and visual attribute type mapping

To evaluate the quality of the data to visualization mapping (see the next paragraph), we use the J. Mackinlay Matrix [Mackinlay *et al.* 2007b] as modeled in figure 3.5 to assign an efficiency score $M(VAT, DT)$ to the mapping between a visual attribute of type $VAT$ and a data attribute of type $DT$ (see the imp_value column in Figure 3.5).

We also encode the fact that a visual attribute is mandatory or not for the visualization: for instance, in a scatter plot as shown in figure 3.7, the three-axis

| visualization | vis_ID | VA | VAT | CorresDT | mandatory | Max#Val | VA_imp |
|---|---|---|---|---|---|---|---|
| Histogram | HT | X | Position | Numeric | 1 | 5000 | 100 |
| SimpleBarChartNominal | SBCN | H | Height | Nominal | 1 | 50 | 60 |
| PieChartNominal | PCO | C | ColorHue | Nominal | 1 | 50 | 95 |
| PieChartNominal | PCO | A | Angle | Nominal | 1 | 50 | 55 |
| BarChartNominal | BCO | X | Position | Numeric | 1 | 5000 | 100 |
| BarChartNominal | BCO | C | ColorHue | Nominal | 1 | 50 | 95 |
| BarChartxOffNominal | BCxN | H | Height | Nominal | 1 | 50 | 60 |
| BarChartxOffNominal | BCxN | C | ColorHue | Nominal | 1 | 50 | 95 |
| LineChartTemporal | LCT | Y | Position | Numeric | 1 | 5000 | 100 |
| LineChartTemporal | LCT | T | TimeAxis | Temporal | 1 | 2000 | 100 |
| ScatterplotNominal | SPO | X | Position | Numeric | 1 | 5000 | 100 |
| ScatterplotNominal | SPO | Y | Position | Numeric | 1 | 5000 | 100 |
| ScatterplotNominal | SPO | C | ColorHue | Nominal | 1 | 50 | 95 |
| ..... | ..... | ..... | ..... | ..... | ..... | ..... | ..... |

Figure 3.4: Visualizations model: a visualization has a name ("visualization" column), an identifier ("vis_ID"), and one or several visual attribute(s). If several visual attributes are necessary to define a visualization, we use several rows (one for each attribute). For each visual attribute, we represent its name ("VA"), type ("VAT"), the corresponding data attribute type it can be matched to ("CorresdDT"), the efficiency of the mapping with the data attribute ("VA_imp"), and the maximum number of distinct values supported by this visual attribute ("Max#Val").

X, Y, and C are mandatory to define the ordinal scatterplot (visualization with legend). The X-axis and Y-axis represent the position visual attribute type with corresponding data attributes 'Discussion' and 'raisedhands' respectively which are numeric. C represents the colorHue and corresponds to the 'class' data attribute which is categorical (ordinal).

In addition, we consider that visual attributes cannot represent too many values, and we define a maximum threshold for this as shown in column $Max\#Val$ of figure 3.4. The visualization model of an visualization having more than one visual attribute mapped with same corresponding data attribute we have consider the visual attribute with higher 'VA_imp' for matching score calculation in fitness function (see section 3.2.5).

To understand this case consider a piechart, as shown in figure 3.8, colorHue C is mandatory to define the ordinal piechart (visualization with legend). C corresponds to the 'parentsschoolSatishfaction' data attribute which is categorical (ordinal). In pie chart we have angle (A/$\theta$) (it signifies the percentage of same type legend in the particular data attribute) with 'VA_imp' 55 and ColorHue (C) with 'VA_imp' 95. Both the visual attributes are mapped to same data attribute 'ParentschoolSatishfaction'. For matching score (parameter of fitness function) calculation of dashboard we have considered 'VA_imp' 95 as it is higher than 55 for data attribute 'ParentschoolSatishfaction'.

Our model of visualization represents the complexity of each visualization (from a novice user's point of view). For this purpose (see figure 3.6), we assign a complex-

| imp_value | Numeric | Ordinal | Nominal | Time |
|---|---|---|---|---|
| 100 | Position | Position | Position | TimeAxis |
| 95 | Length/Height | Density | ColorHue | |
| 90 | Angle | ColorSaturation | Texture | |
| 85 | Slope | ColorHue | Connection | |
| 80 | Area/Size | Texture | Containment | |
| 75 | Volume | Connection | Density | |
| 70 | Density | Containment | ColorSaturation | |
| 65 | ColorSaturation | Length/Height | Shape | |
| 60 | ColorHue | Angle | Length/Height | |
| 55 | Texture | Slope | Angle | |
| 50 | Connection | Area/Size | Slope | |
| 45 | Containment | Volume | Area/Size | |
| 40 | Shape | Shape | Volume | |

Figure 3.5: Mapping efficiency scores ("imp_value" column) derived from Mackinlay's matrix for evaluating the quality of a mapping between a data attribute and a visual attribute.

ity score to each visualization with respect to user type. For instance, we consider that for a novice user histograms, bar charts, and pie charts are easier to understand than a line chart, scatter plot, and bar chart. However expert users can easily understand these complex visualizations also line charts, scatter plots, and bar charts. In terms of complexity and coverage scatter plots cover the maximum number of data attributes at a time than any other visualization. But at the same time have a higher complexity due to the high level of interpretation difficulties for novices and beginners.

### 3.1.7 Data to visualization mapping

The data and visualization models determine how a given visualization can be matched with the dataset. First, we must check if a visualization is compatible with the dataset:

1. The data attribute types required by mandatory visual attributes must be present in the dataset, and

2. The number of distinct values of data attributes must be less or equal to the maximum number of values supported by visual attributes. If those two conditions are met, the matching between the considered visualization and the dataset is feasible.

For the detailed illustration of data to visualization mapping, we considered three visualizations Pie Chart, Bar Chart, and Scatterplot with one visualization table (matrix) as shown in figure 3.9:

| visualization | VL_vis_mark | vis_ID | NV_Complex | BE_Complex | EX_Complex |
|---|---|---|---|---|---|
| Histogram | bar | HT | 10 | 10 | 1 |
| SimplebarChartOrdinal | bar | SBCO | 10 | 10 | 1 |
| SimplebarChartNominal | bar | SBCN | 10 | 10 | 1 |
| PieChartNominal | arc | PCN | 10 | 10 | 1 |
| PieChartOrdinal | arc | PCO | 10 | 10 | 1 |
| LineChart | line | LC | 20 | 20 | 1 |
| LineChartNominal | line | LCN | 20 | 20 | 1 |
| LineChartOrdinal | line | LCO | 20 | 20 | 1 |
| LineChartTemporal | line | LCT | 20 | 20 | 1 |
| Scatterplot | point | SP | 40 | 30 | 1 |
| ScatterplotNominal | point | SPN | 40 | 30 | 1 |
| ScatterplotOrdinal | point | SPO | 40 | 30 | 1 |
| BarChartNominal | bar | BCN | 40 | 30 | 1 |
| BarChartOrdinal | bar | BCO | 40 | 30 | 1 |
| BarChartxOffNominal | bar | BCxN | 40 | 30 | 1 |
| BarChartxOffOrdinal | bar | BCxO | 40 | 30 | 1 |

Figure 3.6: Complexity score assigned to each visualization. The complexity score of the visualization changes when the user type also changes as shown in columns NV_Complex, BE_Complex, and EX_Complex for novices, beginners, and experts respectively.

1. To generate a given pie chart at least one ordinal data attribute (column $CorresDT$ and row in yellow box) in the given dataset. The given data attribute $StageID$ is ordinal,

2. The bar chart is generated with at least one ordinal and one numeric data attribute (column $CorresDT$ and rows in green box) in the given dataset. The given data attributes $StageID$ and $raisedhands$ are ordinal and numeric respectively present in a bar chart, and

3. For scatterplot, at least one ordinal and two numeric data attributes (column $CorresDT$ and rows in red box) in the given dataset. The given data attributes $class$ is ordinal and $raisedhands$ and $Discussion$ are numeric as present in a scatterplot.

If there is more than one data attribute with the same type, in that case, it will choose randomly anyone of them.

## 3.2 Genetic Algorithm

### 3.2.1 Overview

A GA is a search metaheuristic that aims at finding optimal solutions in search space within a limited time. This metaheuristic evolves a population of candidate solutions (chromosomes) with operators such as selection, crossover, and mutation.

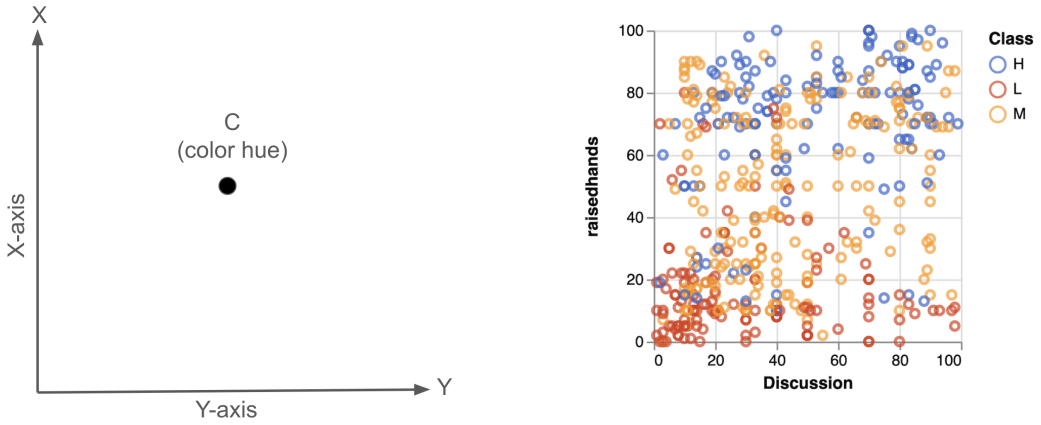



Figure 3.7: Both the figures collectively show the mapping between visual attributes and data attributes. On the left three visual attributes X, Y, and C are presented on the cartesian plane on the X-axis, Y-axis, and legend. On the right, a scatter plot is presented. These three visual attributes X, Y, and C correspond to data attributes 'Discussion', 'raisedhands', and 'class' on X-axis, Y-axis, and legend axis respectively.

It uses a fitness value for each solution, and this fitness represents the function to be optimized.

In our GA, a solution will be a DBo. Let $P(t)$ represents the population of solutions at generation $t$. We use a steady-state GA that generates one individual at each generation. The overall optimization algorithm is as follows:

1. Generate an initial population $P(0)$ of $n$ solutions (i.e. DBo) using a Random generation operator,
2. For a given number of generations $T_{max}$, evolve the population:
   (a) Select two parents $P_1$ and $P_2$ from P(t) using a binary tournament selection
   (b) Apply a crossover operator to $P_1$ and $P_2$ in order to generate two offspring $O_1$ and $O_2$,
   (c) Keep (arbitrarily) $O_1$ and apply a mutation operator to $O_1$ in order to obtain a mutated offspring $O'_1$,
   (d) Evaluate its fitness $F(O'_1)$
   (e) Insert possibly $O'_1$ in P(t): if $O'_1$ is better than the worst solution in $P(t)$, then $O'_1$ will replace that solution
   (f) $t \leftarrow t + 1$

We selected this steady state GA because it evolves smoothly the population. It always keeps in $P(t)$ the best individual found from the initial generation. The

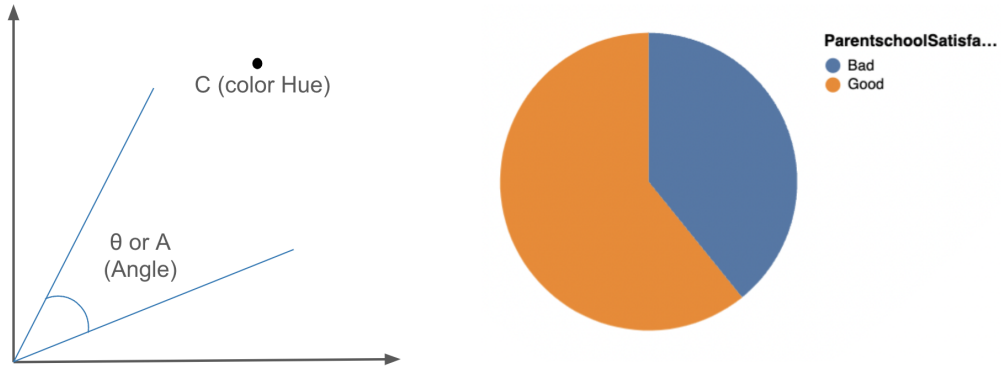



Figure 3.8: Both the figures collectively show the mapping between visual attributes and data attributes. On the left visual attribute Angle (Θ) is presented on the Cartesian plane as an angle. On the right, a pie chart is presented. While other visual attribute ColorHue(C) corresponds to legends in data attribute 'parentsschoolSatishfaction'.

binary tournament selection consists in selecting randomly two individuals, and in keeping the best of them, so as to put selective pressure on the better individuals. This operator is called twice to obtain the two parents $P_1$ and $P_2$.

### 3.2.2 Dashboard genetic representation

The representation we adopted for dashboard genetic representation (DGR) is a variable size matrix as represented in figure 3.11. The rows of this matrix are in variable numbers. Each row represents a visualization that will be included in the DBo. Columns represent the data attributes. Values of the matrix represent how each visualization is mapped to the data attributes. Those values are thus the visual attributes of the considered visualization (on the row).

As shown in figure 3.10, we consider a dashboard with four visualizations: scatterplot, bar chart, pie chart, and histogram. Each individual visualization contains an equal number of visual attributes (column $VA$ in figure 3.4) and corresponding data attributes (column $CorresDT$ in Figure 3.4) as shown in figure 3.4. The scatterplot contains three (in red box), the bar chart two (in green box), the pie chart contains one (in yellow box) and the histogram also contains only one (in violet box).

To understand the mapping between the visual attributes and data attributes we take the reference of the genetic representation of the dashboard in the form of a matrix as shown in figure 3.11. Iterating each of the following rows of visualization column $vis_name$ from top to bottom in the given matrix:

1. The scatterplot contains VisITedResources, Announcements view, and class data attributes corresponding to X, Y, and C visual attributes.

| visualization | vis_ID | VA | VAT | CorresDT | mandatory | Max#Val | VA_imp |
|---|---|---|---|---|---|---|---|
| Histogram | HT | X | Position | Numeric | 1 | 5000 | 100 |
| SimpleBarChartNominal | SBCN | H | Height | Nominal | 1 | 50 | 60 |
| 1 PieChartOrdinal | PCO | C | ColorHue | Ordinal | 1 | 50 | 85 |
| 2 PieChartOrdinal | PCO | A | Angle | Ordinal | 1 | 50 | 60 |
| 1 BarChartOrdinal | BCO | X | Position | Numeric | 1 | 5000 | 100 |
| 2 BarChartOrdinal | BCO | C | ColorHue | Ordinal | 1 | 50 | 85 |
| BarChartxOffNominal | BCxN | H | Height | Nominal | 1 | 50 | 60 |
| BarChartxOffNominal | BCxN | C | ColorHue | Nominal | 1 | 50 | 95 |
| LineChartTemporal | LCT | Y | Position | Numeric | 1 | 5000 | 100 |
| LineChartTemporal | LCT | T | TimeAxis | Temporal | 1 | 2000 | 100 |
| 1 ScatterplotOrdinal | SPO | X | Position | Numeric | 1 | 5000 | 100 |
| 2 ScatterplotOrdinal | SPO | Y | Position | Numeric | 1 | 5000 | 100 |
| 3 ScatterplotOrdinal | SPO | C | ColorHue | Ordinal | 1 | 50 | 85 |
| ..... | ..... | ..... | ..... | ..... | ..... | ..... | ..... |

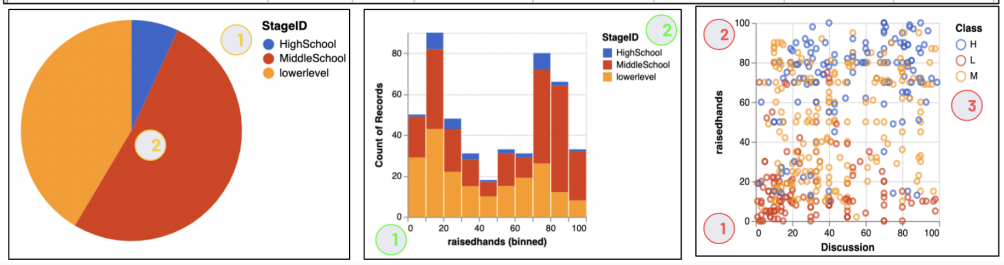



Figure 3.9: The table on top of the figure presents the visualization model with "visual attributes" and their type in columns $VA$ and $VAT$ respectively. Both columns are used to calculate the "visual attributes importance" given in column $VA_imp$. At the bottom of the figure, three visualizations (pie chart, bar chart, and scatterplot) are generated from this table. The feasibility of these visualizations is found based on a mapping between "visual attributes types" and corresponding "data attributes types" given in columns $VAT$ and $CorresDT$ respectively. Finally "visual attributes" are mapped with "data attributes" of the data set as indicated with a yellow, green, and red box in the table and numbering on visualizations.

2. Bar chart contains raisedhands and StageID data attributes corresponding to X and C visual attributes.

3. Pie chart contains ParentsSchoolSatishfaction data attributes corresponding to C visual attribute.

4. Histogram contains VisITedResources data attributes corresponding to X visual attribute.

This representation can encode any dashboard.

### 3.2.3 Random generation

This random generation operator generates DBo from scratch but follows many constraints derived from the data, user, and visualization models. Let $D$ represent the DBo to generate. This operator works as follows:

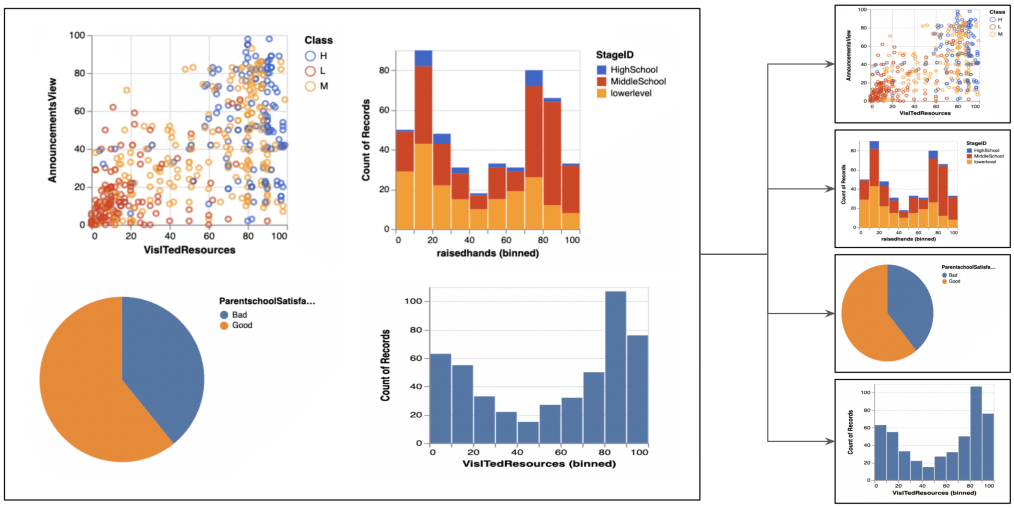



Figure 3.10: The figure showed how the dashboard is generated from 4 individual visualizations (Scatterplot, Bar chart, Pie chart, and Histogram). Each visualization row shown in figure 3.11 is converted to a graphical chart with the help of Vega-lite encoding.

| DBo | vis_name | gender | ..... | StageID | GradeID | ..... | raisedhands | VisITedResources | AnnouncementsView | ..... | ParentschoolSatisfaction | ..... | Class |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DBo_8 | ScatterplotNominal | | | | | | | x | Y | | | | color |
| DBo_8 | BarChartNominal | | | color | | | x | | | | | | |
| DBo_8 | PieChartNominal | | | | | | | | | | color/angle | | |
| DBo_8 | Histogram | | | | | | | x | | | | | |

Figure 3.11: The DBo representation used in our GA: the four rows, are visualizations present in the DBo. Columns represent the data attributes. Values of the matrix are the visual attribute names and represent the mapping between the data attributes and the visual attributes.

1. Determine the number $N$ of visualizations to create to $D$. $N$ is randomly generated from 2 to the maximum number specified by the user, and let $D \leftarrow \emptyset$,
2. Pick randomly (with replacement) a visualization $Vis$ in the list approved by the user,
3. Try to match $Vis$ with the data. For each visual attribute of $Vis$:
   (a) List the possible data attributes that can be matched with this visual attribute. This list $L$ is established with several constraints: the data attribute type must be compatible with the visual attribute type (see the visualization model). The number of distinct values of the data attribute must also be compatible with the maximum number of values that can be visualized with the visual attribute,
   (b) If $L = \emptyset$, then no match is possible with this visualization and Go to 2 (to try with another one),

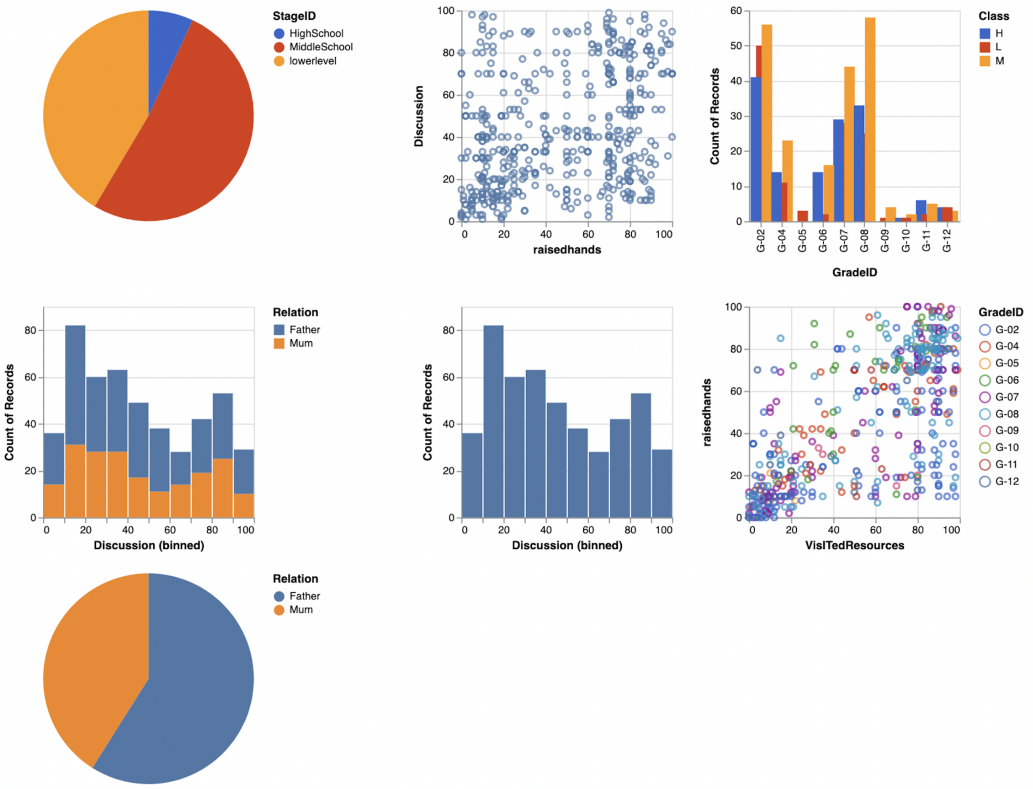
(c) Else match the visual attribute to a randomly selected data attribute in $L$,

4. Add $Vis$ to $D$

5. If $N$ visualizations have been added to $D$ Then Stop, Else Go to 2

To understand the random generation we can consider 2 examples. Let's consider the dashboards DBo1 and DBo2 are generated randomly.

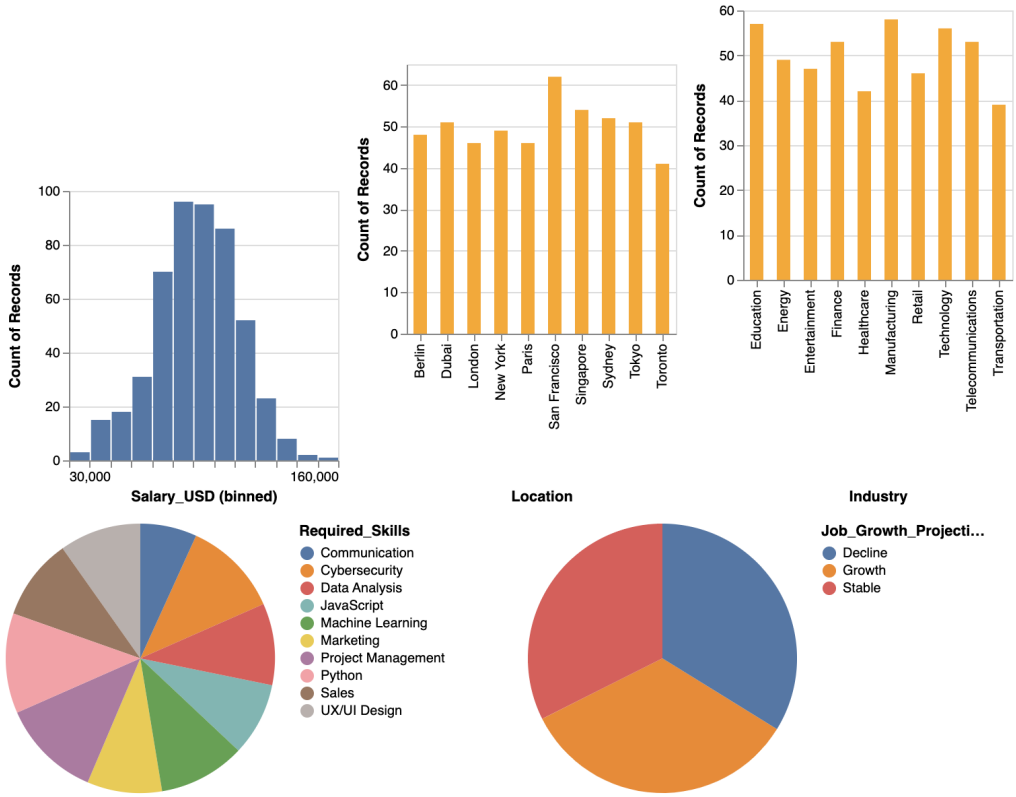
**Example 1 for randomly generated dashboard having 7 visualizations** To generate the DBo1, we consider the system to choose a number 7 randomly, between 2 to 9. During this example 1, we use the student dataset.




Now we need to select 7 visualizations for DBo1 one by one. The system will pick the visualization randomly from the list of visualizations as shown in figure 3.6. Let's consider the first selected visualization is a pie chart then next the system will use the visualization model to check the feasibility of the pie chart with respect to the given dataset. If visualization is feasible in that case pie chart will be added to DBo1. Again system selects a second visualization this time we consider a line chart, but the line chart is not feasible because the given dataset does not contain at least one 'Temporal' data attribute type which is mandatory to generate a line chart.

This random selection of visualization and their feasibility check will continue until the system finds 6 more visualizations to create a DBo1 with 7 visualizations. Finally to generate the DBo2 system the feasibility of the following visualizations: Pie chart, ScatterPlot, Bar chart, and Histogram.

**Example 2 for randomly generated dashboard having 6 visualizations** To generate DBo2, the system randomly selects a number, in this case, 6, from a range between 2 and 9. In Example 2, we use the "CAI-Powered Job Market Insights" dataset [Laksika Tharmalingam 2024].




The process begins with the system selecting a visualization, starting with a Histogram. The system then checks whether the Histogram is feasible given the dataset. If the Histogram is appropriate, it is added to DBo2. The system then randomly selects another visualization, for instance, a scatterplot. However, if the scatterplot is not feasible because the dataset lacks at least two numerical attributes, which are necessary for generating a scatterplot, it is not added. Similarly, a line chart would also be deemed infeasible if the dataset does not include any temporal data attributes.

This process of randomly selecting visualizations and checking their feasibility continues until the system successfully identifies four additional feasible visualizations, resulting in a DBo2 with a total of five visualizations. In the end, the system

confirms the feasibility of the following visualizations for DBo2: Histogram, Simple Bar Chart, and Pie Chart.

### 3.2.4 Genetic operators

Genetic operators evolve the chromosomes of an individual (i.e. dashboard). We use two genetic operators crossover and mutation. The detailed application and implementation of both the operators are illustrated in a further sections 3.2.4.1 and 3.2.4.2 for crossover and mutation respectively.

#### 3.2.4.1 Crossover operators

The crossover operator consists of exchanging visualizations from the two parents according to a randomly selected cutting point.

The crossover operator gives two new offspring. These new offspring are optimized dashboards that contain visualizations of both the parent's dashboards. The new offspring have better fitness value (see section 3.2.5) than parents' dashboards.

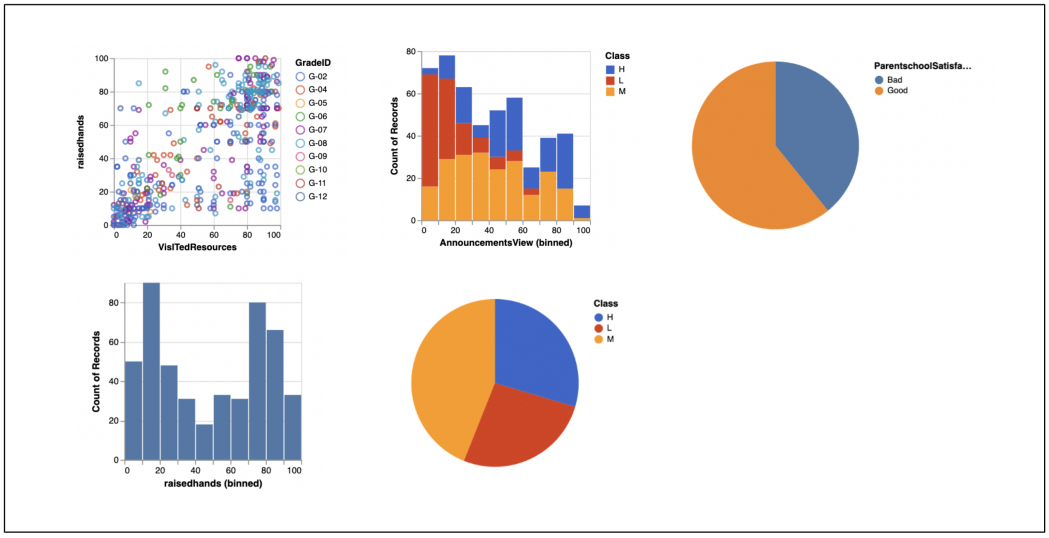



Figure 3.12: The figure presents parent 1 as a dashboard. It contains five visualizations: Scatterplot, Bar chart, Histogram, and two Pie charts

For the visual explanation of the crossover operator, we considered the example of two parents as shown in figure 3.12 for $parent-1$ and figure 3.13 for $parent-2$. $Parent-1$ contains five visualizations: scatterplot, bar chart, two pie chart, and histogram. $Parent-2$ contains four visualizations: bar chart, scatterplot, histogram, and pie chart. Figure 3.14 presents the genetic representation of both dashboards.

The crossover is performed between both parent's dashboards to generate two new offspring dashboards as shown in figure 3.15 and 3.16 graphically and figure 3.17

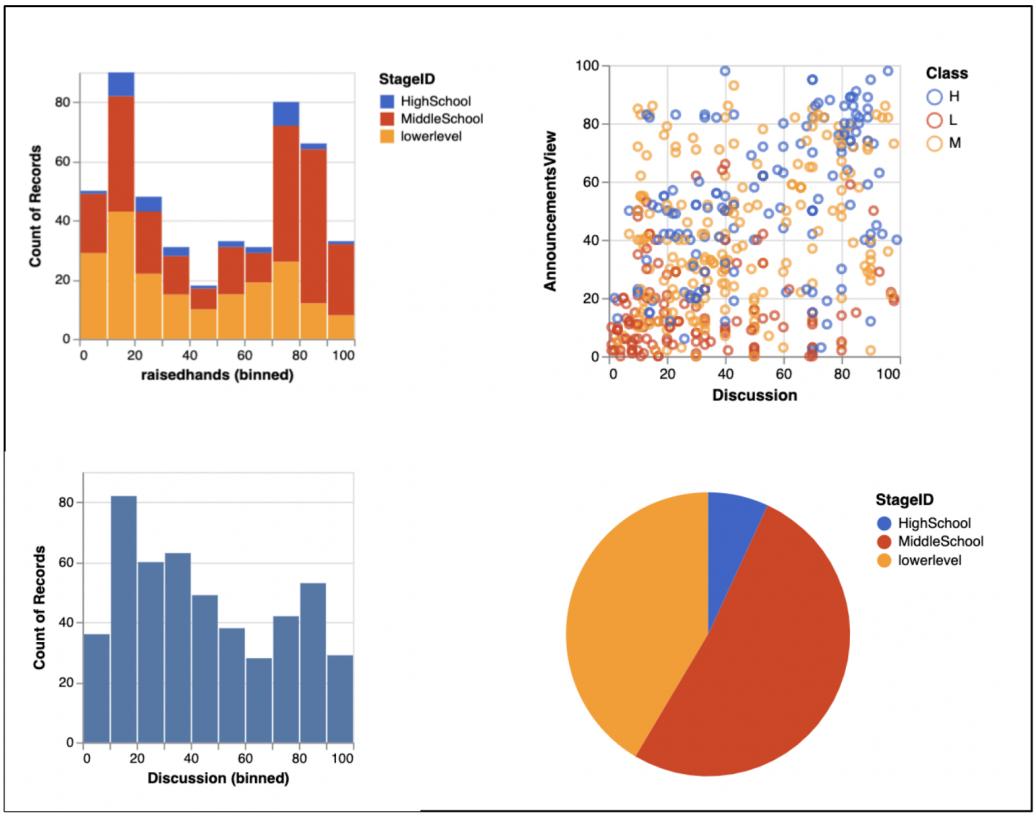



Figure 3.13: The figure presents parent 2 as a dashboard. It contains four visualizations: Bar chart, Scatterplot, Histogram and Pie chart

with DGR. The red broken line in all the three figures 3.15, 3.16 and 3.17 represents the cutting point. These cutting points divide both parents into two parts.

As shown in figure 3.18 for $offspring-1$ and figure 3.19 for $offspring-2$. Both the newly generated offspring dashboards contain some visualization of $parent-1$ and $parent-2$. The $offspring-1$ contains three visualizations (scatterplot, bar chart, and pie chart) of $parent-1$ (part-1 of parent-1) and two visualizations (pie chart and histogram) of $parent-2$ (part-2 of parent-2). While on the other side, the $offspring-2$ contains two visualizations (bar chart, and scatterplot) of $parent-2$ (part-1 of parent-2) and two visualizations (histogram and pie chart) of $parent-1$ (part-2 of parent-2). The figure 3.20 represents DGR for both offsprings.

#### 3.2.4.2 Mutation operator

The mutation operators implement the following three operation on the dashboard: add visualization, remove visualization, and visualization mapping.

- Add visualization operator add a new visualization to the given DBo,

**Parent 1**

| DBo | vis_name | gender | StageID | GradeID | raisedhands | VisITedResources | AnnouncementsView | Discussion | ParentschoolSatisfaction | Class |
|---|---|---|---|---|---|---|---|---|---|---|
| DBo_11 | ScatterplotOrdinal | | | C | Y | X | | | | |
| DBo_11 | BarChartOrdinal | | | | | | X | | | C |
| DBo_11 | PieChartNominal | | | | | | | | C/A | |
| DBo_11 | Histogram | | | | X | | | | | |
| DBo_11 | PieChartOrdinal | | | | | | | | | C/A |

**Parent 2**

| DBo | vis_name | gender | StageID | GradeID | raisedhands | VisITedResources | AnnouncementsView | Discussion | ParentschoolSatisfaction | Class |
|---|---|---|---|---|---|---|---|---|---|---|
| DBo_55 | BarChartOrdinal | | C | | X | | | | | |
| DBo_55 | ScatterplotNominal | | | | | | Y | X | | C |
| DBo_55 | PieChartOrdinal | | C/A | | | | | | | |
| DBo_55 | Histogram | | | | | | | X | | |

Figure 3.14: The figure presents both the parent's dashboard (parent-1 and parent-2) in the form of genetic representation.

- Remove visualization operator remove an existing visualization from the given DBo,
- Visualization mapping operator is perform by changing the mapping between visual attribute and data attribute keeping data attribute type constant.

The mutation operator optimizes the dashboard by making small and big modifications as needed. The mutation operation avoids the complete iteration of GA to save time.

After crossover, out of these two offspring, we have selected the $offspring-1$ randomly to apply all three mutation operators (see figure 3.24) as follows:

1. As shown in figure 3.21, a new visualization is added to $offspring-1$. The new visualization is selected randomly from the list of given visualization figure 3.6. If the selected visualization is feasible concerning given data attributes of a dataset then only this mutation (add visualization) will take place. Here bar chart present in the red box is feasible and added to $offspring-1$ as shown in Figure 3.21.

2. Remove visualization mutation is simple. We will select randomly any one visualization out of five visualizations present in $offspring-1$ (see figure 3.18). The randomly selected visualization is scatterplot which is deleted from the dashboard. The newly mutated dashboard after applying mutation (remove visualization) will contains only four visualizations, as shown in figure 3.22.

3. Unlike the add and remove visualization mutation, the mapping mutation is completely different. In mapping mutation, we have randomly mapped each visual attribute of each visualization of the dashboard with all possible data attributes of the same type. As the dashboard is shown in figure 3.23 after mapping mutation. The three visualizations (scatterplot, histogram, and pie chart in the second row) as shown in the red box are mapped with new data

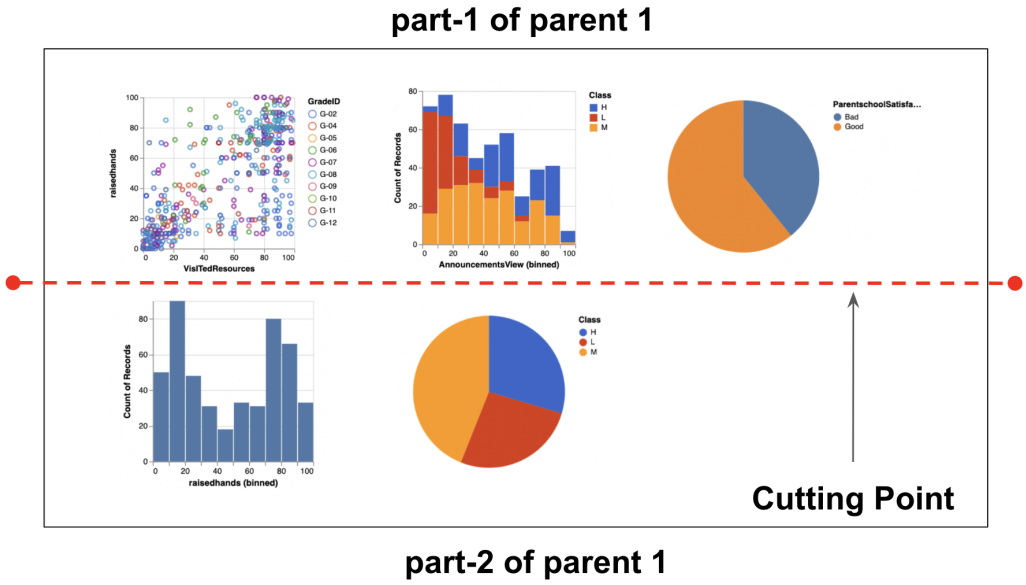



Figure 3.15: The figure presents the cutting points (with red broken line) of the parent 1 dashboards for the crossover operator. This cutting point divides the parent 1 into two parts. The visualizations below this cutting point are exchanged.

attributes. But two (bar chart and pie chart in the second row) remains constant.

To understand the genetic representation of all these three mutated dashboards we have shown in figure 3.24:

- On the top the of figure figure 3.24 the matrix represents DGR for mutated dashboard with added new visualization. Here BarchartNominal is added with two data attributes VisITedResources and gender with corresponding X and C visual attributes. This visualization is added as a row at the end of the DGR.

- In the middle of figure 3.24 the matrix represents DGR for mutated dashboard with remove visualization. Here ScatterPlotOrdinal row is removed for DGR. This visualization is removed as a row between the data attributes row and the BarchartOrdinal row.

- At the bottom of figure 3.24 the matrix represents DGR for mutated dashboard with mapping the visualization's visual attributes.

    1. The scatter plot, present in the second row of the matrix is mapped to the new data attribute as follows: X visual attribute is mapped with AnnouncementsView (indicated by orange arrow) instead of VisITedResources (indicated by oragne box). Y visual attribute is mapped with Discussion (indicated by red arrow) instead of raisedhands (indicated by

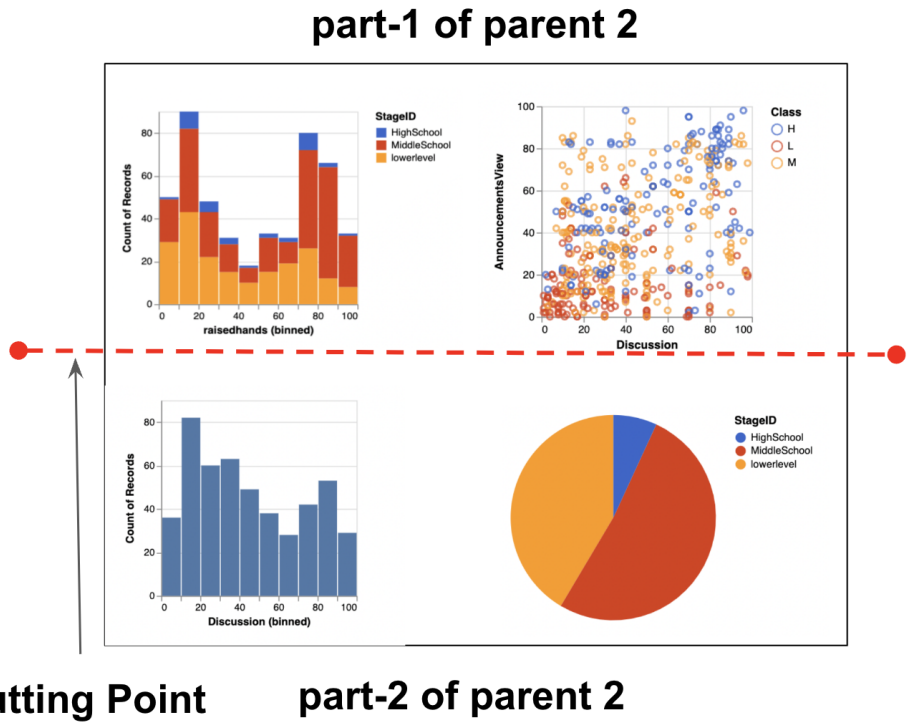



Figure 3.16: The figure presents the cutting points (with red broken line) of the parent 2 dashboards for the crossover operator. This cutting point divides the parent 2 into two parts. The visualizations below this cutting point are exchanged.

the red box). C visual attribute is mapped with gender (indicated by green arrow) instead of GradeID (indicated by green box).

2. The PieChartOrdinal, present in the fifth row of the matrix is mapped to new data attribute as follows: C/A (color/angle) visual attribute is mapped with Class (indicated by green arrow) instead of StageID (indicated by green box).
3. The Histogram, present in the sixth row of the matrix, is mapped to new data attribute as follows: X visual attribute is mapped with raisedhands (indicated by the orange arrow) instead of Discussion (indicated by the orange box).

#### 3.2.4.3 Probability of execution genetic operators

The probability of execution of both the operator mutation and crossover is depicted in figure 3.25. The two pairs of individuals are randomly picked (A) from the population. (B) We selected 2 parents based on a binary tournament (P1 and P2). We have performed the binary tournament to select the best of two individuals to

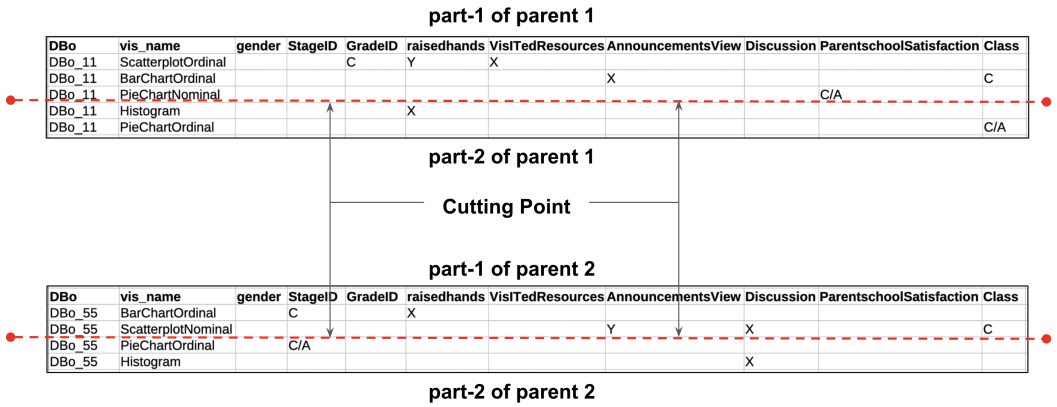



Figure 3.17: The genetic representation of both dashboards for crossover operator. The red broken line presents the cutting point. The visualizations below this cutting point are exchanged.

ensure that the newly generated DBo will have a better fitness value than the worst individual in the population.

The three possibilities for the next step are crossover, mutation, and direct replacement. The probability of executing crossover and mutation is as follows:

- The crossover (C) executes with a probability of 0.35 when the number of data attributes is more than 5. Otherwise, it will execute with a probability of 0.01. The probability of crossover is low when the number of data attributes is less than 5 because in that particular scenario, the chromosomes between two parents are not able to be exchanged effectively due to less sample space. The generated dashboard after the crossover has almost the same or sometimes less fitness value than parents.

- The mutation (D) executes with a probability of 0.65 when the number of data attributes is more than 5. Otherwise, it will execute with a probability of 0.99. The mutation works effectively because it has three levels of degree of modification:

  Further, the mutation operators are executed with different levels as follows:

  1. The add new visualization (E) will execute with a probability of 0.8. This operator is moderate neither it completely destroys the exitance of the current dashboard nor keeps the current dashboard as it is with the same visualization. We count this operator between the crossover and the other two mutations (mapping and removing visulization) in terms of dashboard fitness value and assign a high probability of execution. This operator has a higher probability of increasing the fitness value of the dashboard as compared to others.
  2. The remove visualization (F) will execute with a probability of 0.1. his operator has a high probability of decreasing the fitness value of the

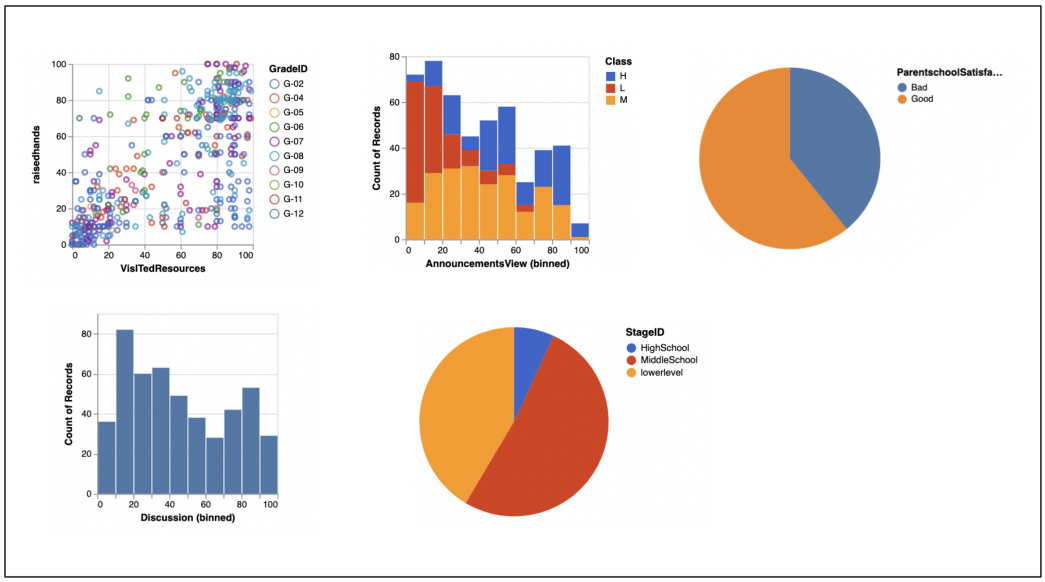



Figure 3.18: Offspring 1, generated after crossover operator. This dashboard contains three visualizations in the first row from parent 1 and two visualizations in the second row from parent 2.

dashboard as compared to add visulization. But sometimes it removes the visulization from the dashboard which has high complexity and increases overall dashboard fitness value.

3. The mapping (G) of visualization's visual attributes will execute with a probability of 0.1. This operator does not make any modifications in the dashboard at the visulization level. This operator is only helpful when we want to cover specific data attributes for certain visualization in the dashboard based on user importance.

In an ideal case, there is a possibility that neither crossover nor mutation operator get executed. In that scenario, GA will execute the replacement (H) directly (considering the fitness value condition is false) after the selection and binary tournament of dashboard individuals. To understand such an ideal scenario. Consider the data set which has only one numeric, one temporal, and one nominal data attribute. For a certain GA iteration after the binary tournament, the mutation operator executes with 0.65 probability by skipping crossover. Next, we consider within mutation, the mapping operator executes with 0.1 probability by skipping add and remove visualization. But for instance, the mapping cannot be performed because there is no alternative for each visual attribute to map with any other data attribute. In this case, the individual dashboard will remain unmutated with the same fitness value. But in the end, the replacement operator will be executed if its fitness value is greater then the worst individual present in the population. Likewise, this ideal scenario can be also possible when there are only two visualizations in

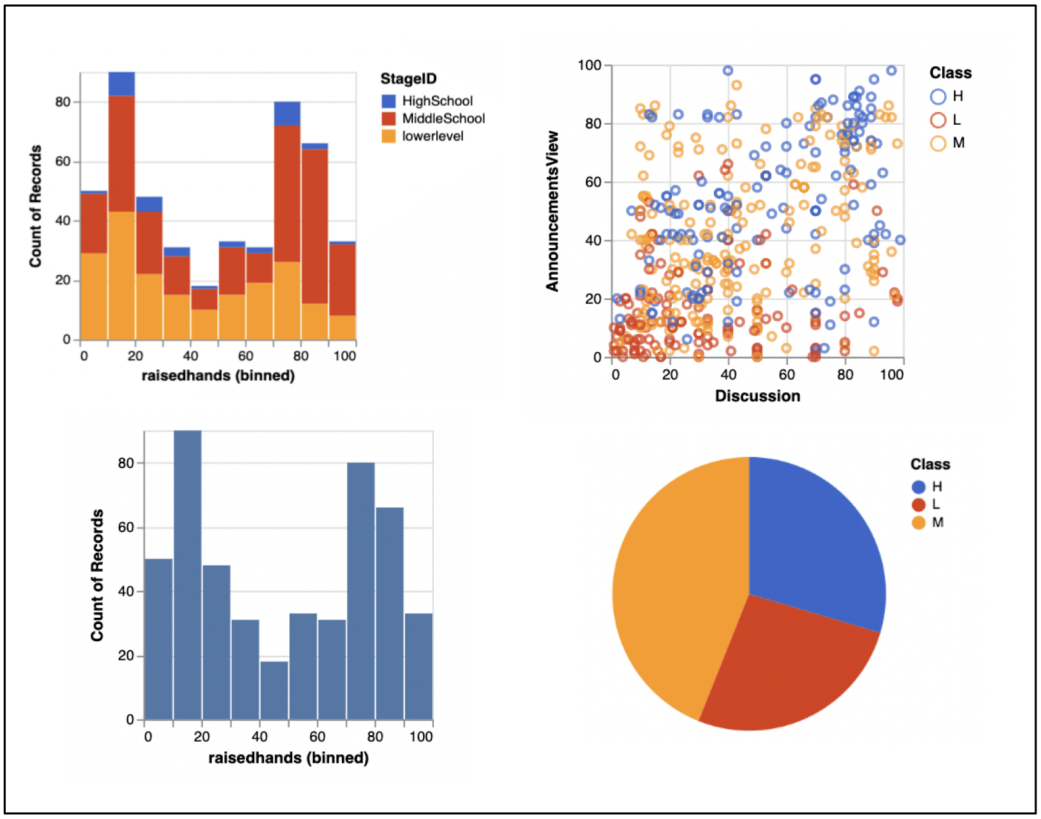



Figure 3.19: Offspring 2, generated after crossover operator. This dashboard contains two visualizations in the first row from parent 2 and two visualizations in the second row from parent 1.

the individual dashboard and mutation is performed with the remove visualization mutation operator.

### 3.2.5 Fitness function

For the mathematical evaluation of a dashboard, we defined a combination of some parameters, coefficients, and mathematical operators in the form of an equation. This equation is known as the fitness function. Fitness function $F(D)$ is maximal when selected visualizations are simple, cover many important attributes for the user, and match efficiently the data attributes. The higher the value of the fitness function the better the quality of the dashboard.

The quality of the dashboard depends on three parameters: Complexity score, Coverage importance score, and Matching score. Then each of the parameter is defined as following:

- Complexity score: to define the simpler visualization in the DBo,
- Coverage importance score: to cover all important attribute in DBo,

**offspring 1**

| DBo | vis_name | gender | StageID | GradeID | raisedhands | VisITedResources | AnnouncementsView | Discussion | ParentschoolSatisfaction | Class |
|---|---|---|---|---|---|---|---|---|---|---|
| DBo_11 | ScatterplotOrdinal | | | C | Y | X | | | | |
| DBo_11 | BarChartOrdinal | | | | | | X | | | C |
| DBo_11 | PieChartNominal | | | | | | | | C | |
| DBo_55 | PieChartOrdinal | | C/A | | | | | | | |
| DBo_55 | Histogram | | | | | | | X | | |

**offspring 2**

| DBo | vis_name | gender | StageID | GradeID | raisedhands | VisITedResources | AnnouncementsView | Discussion | ParentschoolSatisfaction | Class |
|---|---|---|---|---|---|---|---|---|---|---|
| DBo_55 | BarChartOrdinal | | C | | X | | | | | |
| DBo_55 | ScatterplotNominal | | | | | | Y | X | | C |
| DBo_11 | Histogram | | | | X | | | | | |
| DBo_11 | PieChartOrdinal | | | | | | | | | C/A |

Figure 3.20: The genetic representation of both offspring dashboards after crossover operator.

- Matching score: to have a good data to visualization mapping.

For an ideal dashboard, the value of the Coverage importance score and the matching score should be 1. For Coverage importance, it means the DBo covers all the data attributes present in dataset. And for matching score it signifies the mapping between data to visualization is best based on Mackinley's matrix (as modeled in figure 3.5). But on the other side the Complexity score should be approaching 0. It indicates that the given dashboard contains simplest visualizations considering certain user type.

But the value of the fitness equation for an ideal dashboard varies because it also depends on all these three parameters coefficients $\alpha$, $\beta$, and $\gamma$. These coefficient values are defined based on user type.

To understand the fitness function and its parameters. We consider a given dashboard $D$, and the fitness function $F$ aims at evaluating the interest of $D$. $F(D)$ is used to guide the genetic search. To define $F$, we have used several scores:

- Complexity score $Cplx \in [0, 1]$: the mean of the complexity values of each visualization present in $D$, divided by 100. The lower this score, the easier the dashboard should be for the user to understand. The complexity value of visualization is based on user type as shown in Figure 3.6,

- Coverage importance score $Cov_{imp} \in [0, 1]$: the sum of the data attributes importance for the data attributes that are represented in $D$ (if an attribute is present several times, it accounts only for once), divided by the sum of all data attributes an importance in a given dataset. This score is to be maximized, in order to propose dashboards that cover all important attributes for the user. We avoid doubles otherwise the GA would recommend the DBo having the same visualizations several times with the same data attributes,

- Matching score $Match \in [0, 1]$: the mean of the individual matching scores (from Figure 3.5) for each couple of data-visual attributes, divided by 100.

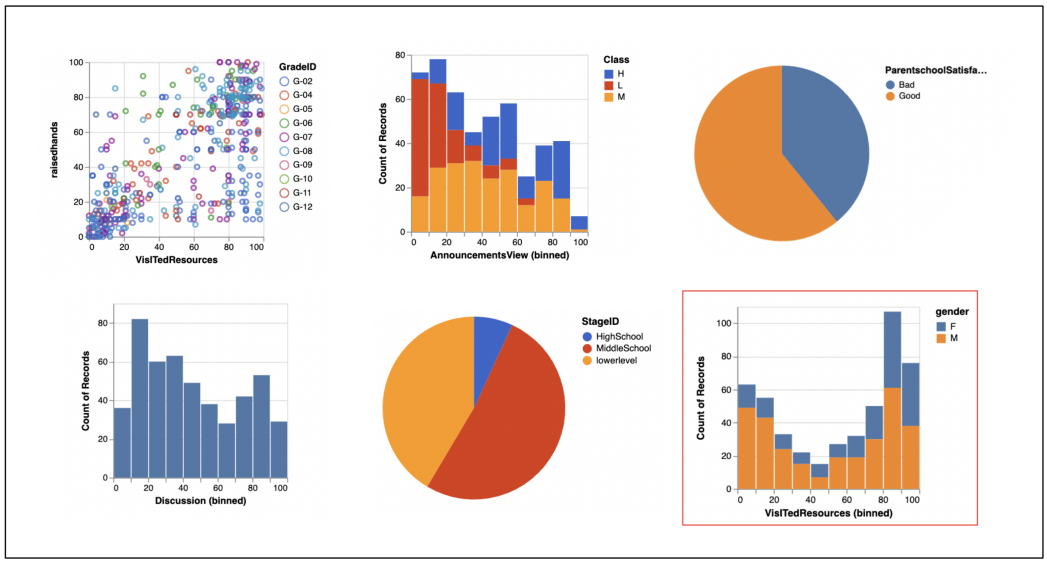



Figure 3.21: Add visualization mutation: Mutation is implemented on offspring 1. A new visualization "Bar Chart" is added to the second row of dashboard (present in red box). After mutation, there are six visualizations in the dashboard.

The final fitness value of $D$ is computed as follows:

$$F(D) = \alpha Match + \beta Cov_{imp} - \gamma Cplx$$

$F(D)$ reaches its maximum when the selected visualizations are simple, capture a wide range of important user attributes, and align efficiently with the data characteristics. The value of coefficients changes with respect to selected user types as follows:

Table 3.1: Coefficients values as per user types

| $UserType$ / $Coefficients$ | $\alpha$ | $\beta$ | $\gamma$ |
|---|---|---|---|
| Novice | 1 | 2 | 1 |
| Beginner | 1 | 2 | 0.5 |
| Expert | 1 | 2 | 0.1 |

The values of $\alpha$ and $\beta$ are the same for all three types of users. But the $\gamma$ values change in each case. The value of $\gamma$ is corresponding to visualization complexity. We considered all three user types novice, beginner, and expert, to have different levels of

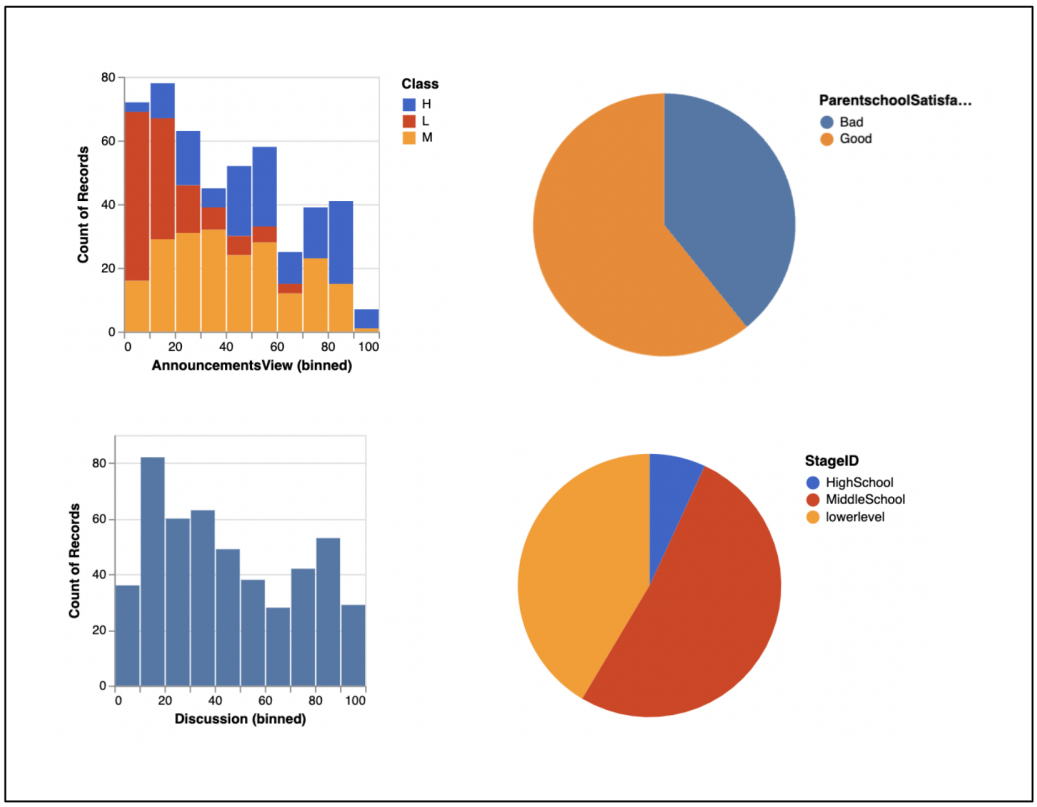



Figure 3.22: Remove visualization mutation: Mutation is implemented on offspring 1. An existing visualization "scatterplot" is removed from the first row of the dashboard. After mutation, there are four visualizations in the dashboard.

understanding and interpretation skills for different visualizations. For novices and beginner, visualization complexity plays a very important role in the interpretation recommended dashboard. For experts, complexity does not matter much like novices and beginners. For experts, we consider the same complexity value (that is 1) for all visualizations as shown in Figure 3.6. The goal of considering the complexity value of visualizations in one digit is to minimize (neglect) the intervention of selected visualization types during dashboard recommendations. We consider the expert to be familiar with all visualizations and can easily interpret all the visualizations present in the recommendation dashboard.

The values of $\alpha$, $\beta$, and $\gamma$ are having different weights. Because $\alpha$ corresponds to the Matching score. The match score is important for individual visualization not for the whole dashboard. Match score is internally defined (with Mackinley's Matrix as modeled in figure 3.5) without taking user input into account. In the case of $\beta$, it corresponds to the Coverage importance score which determines the presence of data attributes in the dashboard. The ideal dashboard contains all data attributes. The Coverage importance score also depends on user inputs (user importance value) so we want to keep it higher than $\alpha$ and $\gamma$. As $\gamma$ corresponds to the complexity

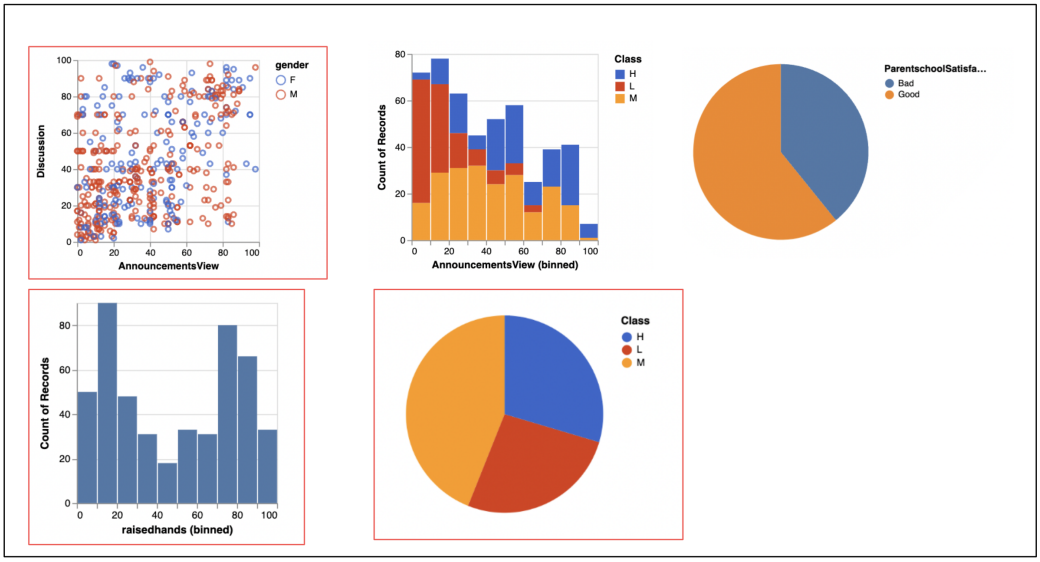



Figure 3.23: Mapping between "visual attributes" and "data attributes" for mutation: Mapping mutation is implemented on offspring-1. An existing visualization remains in the dashboard but the mapping between "visual attributes" and "data attributes" is changed for a few visualizations. The mapping is changed for "Scatterplot", "Histogram", and "Pie Chart" as shown in the red box.

score and the complexity score is determined based on the user type.

We always keep the probability of mutation higher than crossover because crossover distorts individuals with big changes by exchanging or adding or removing at least one or more visualizations from the dashboard.

## 3.3 Results

### 3.3.1 Experimental set-up

In order to test the performance of our approach, we have used several scenarios inspired by the SAPS dataset. In this dataset, data attributes are classified into four major categories: (1) Demographic (2) Academic background, (3) Behavioral attributes, and (4) Performance attributes. We concentrate on the following user scenario (inspired by behavioral attributes): the user puts an emphasis on Raised hands-on class, Visited resources, Viewing announcements, Discussion groups, Absence days, and Class. These attributes have an importance of 100. The user considers that other attributes could also be included: Gender, Nationality, StageID have an importance of 50. All other data attributes have an importance of 10.

With this scenario, we wanted to check the ability of the GA to propose interesting visualizations. All runs were performed on a MacBook Pro (M1 processor, 8 cores, 3.2 GHz) equipped with 16 GB of RAM and an SSD drive.

| DBo | vis_name | gender | StageID | GradeID | raisedhands | VisITedResources | AnnouncementsView | Discussion | ParentschoolSatisfaction | Class |
|---|---|---|---|---|---|---|---|---|---|---|
| DBo_11 | ScatterplotOrdinal | | | C | Y | X | | | | |
| DBo_11 | BarChartOrdinal | | | | | | X | | | C |
| DBo_11 | PieChartNominal | | | | | | | | C/A | |
| DBo_55 | PieChartOrdinal | | C/A | | | | | | | |
| DBo_55 | Histogram | | | | | | | X | | |
| DBo_A1 | BarChartNominal | C | | | | | X | | | |

| DBo | vis_name | gender | StageID | GradeID | raisedhands | VisITedResources | AnnouncementsView | Discussion | ParentschoolSatisfaction | Class |
|---|---|---|---|---|---|---|---|---|---|---|
| DBo_11 | BarChartOrdinal | | | | | | X | | | C |
| DBo_11 | PieChartNominal | | | | | | | | C/A | |
| DBo_55 | PieChartOrdinal | | C/A | | | | | | | |
| DBo_55 | Histogram | | | | | | | X | | |

| DBo | vis_name | gender | StageID | GradeID | raisedhands | VisITedResources | AnnouncementsView | Discussion | ParentschoolSatisfaction | Class |
|---|---|---|---|---|---|---|---|---|---|---|
| DBo_11 | ScatterplotOrdinal | C | | | | | X | Y | | |
| DBo_11 | BarChartOrdinal | | | | | | X | | | C |
| DBo_11 | PieChartNominal | | | | | | | | C/A | |
| DBo_55 | PieChartOrdinal | | | | | | | | | C/A |
| DBo_55 | Histogram | | | | X | | | | | |

Figure 3.24: The genetic representation of all three mutated dashboards. On the top, "add visualization mutation", where "BarChartNominal" row is added at the end of DBo. In the middle "remove visualization mutation", where "ScatterPlotOrdinal" row is removed between data attribute names, and "BarCharOrdinal". At the bottom Mapping between "visual attributes" and "data attributes" mutation. The mapping is changed only for the first and last two visualizations shown in the red box. The boxes (green, red, and orange) represent the old position of visual attributes while the arrows (green, red, and orange) represent the new position.

### 3.3.2 Parameter study

We have performed a simple grid search to evaluate which parameters we should be using for our GA.

- We tested several values of population size with $n = 10, 20, 30, 50, 100$. We did not go beyond $n = 100$ because results show that when $n = 50$ has better fitness value than $n = 100$
- Number of generations $T_{max} = 1000, 2000, 5000, 10000, 20000$, for several scenarios. We did not go beyond $T_{max} = 20000$ because results show that when $T_{max} = 20000$ have a better fitness value than $T_{max} > 20000$.

Figure 3.26 shows a typical run for the studied scenario: the GA convergence can be observed with the increasing of the three curves. Table 3.2 shows the results of the grid search. After analyzing the results from several scenarios, we concluded that the best parameters for the GA were $T_{max} = 20000$ and $n = 50$. The running times of our GA are short: a run for 20000 generations takes around 6.4s. The user does not have to wait long until the GA proposes a DBo.

### 3.3.3 Examples of generated dashboards

A sample of the obtained results is presented in Figure 3.27. The fitness value and data attribute analysis of the sample dashboards is given in Table 3.3 and Table 3.4 respectively. In Figure 3.27(a), the presented DBo comes from the initial population.

Table 3.2: Sample of results obtained during the grid search: values represent the best fitness values obtained at each run and averaged over ten runs. Values in parenthesis represent the standard deviations and the best result is highlighted. For each run, the running time is less than 10 seconds.

| $T_{max}/n$ | 10 | 20 | 30 | 50 | 100 |
|---|---|---|---|---|---|
| 1000 | 0.467 | 0.527 | 0.520 | 0.483 | 0.508 |
| | (0.057) | (0.056) | (0.066) | (0.052) | (0.016) |
| 2000 | 0.444 | 0.581 | 0.563 | 0.573 | 0.554 |
| | (0.043) | (0.047) | (0.055) | (0.062) | (0.041) |
| 5000 | 0.609 | 0.594 | 0.607 | 0.608 | 0.638 |
| | (0.057) | (0.053) | (0.041) | (0.043) | (0.039) |
| 10000 | 0.618 | 0.639 | 0.638 | 0.640 | 0.647 |
| | (0.067) | (0.049) | (0.022) | (0.019) | (0.022) |
| 20000 | 0.623 | 0.655 | 0.633 | **0.699** | 0.681 |
| | (0.038) | (0.038) | (0.031) | **(0.027)** | (0.033) |

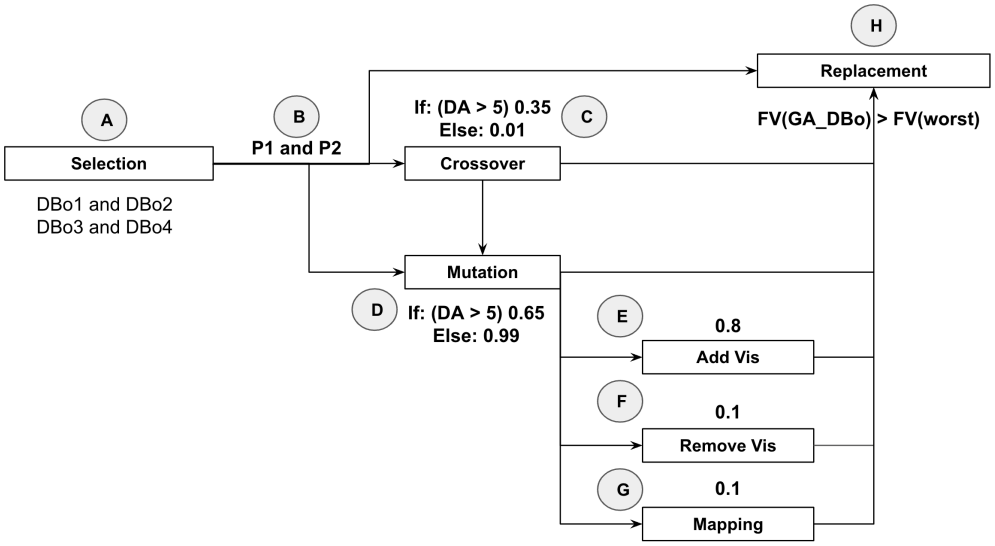



Figure 3.25: GA Operators Probability. (A) The selection of two pairs of DBo was done randomly from the population. (B) Selection of two parents (P1 and P2) for crossover based on binary tournament. (C) The crossover will be performed with a probability of 0.35 if a number of data attributes in the dataset is more than 5 else with a probability of 0.01. (D) The mutation will be performed with a probability of 0.65 if a number of data attributes in the dataset is more than 5 else with a probability of 0.99. (E) Add Visualization mutation will be performed with a probability of 0.8. (F) Remove Visualization mutation will be performed with a probability of 0.1. (G) Mapping mutation will be performed with a probability of 0.1. (H) Replacement can be possible in three ways: through crossover, mutation, or directly after binary tournament

It has a low fitness value (0.099) because it was generated randomly. It contains diverse visualizations including three scatter plots whose complexity (Figure 3.6) is high. In Figure 3.27(b), the DBo is the best solution found after 20000 generations. It has a much higher fitness value (0.799, compared to 0.099) as shown in Table 3.3 last column. This shows that the GA has the ability to quickly optimize the fitness function and produce better individuals in the population.

Taking a closer look at each DBo, we can notice that:

- The optimized DBo contains simpler visualizations (complexity score of 0.171) compared to the initial DBo (complexity score of 0.571) as shown in Table 3.3 complexity column. Scatter plots have been removed in the final DBo, and the selected visualizations are easier to interpret and understand for novice users (as expressed by the complexity weights),

- the initial DBo considers 8 data attributes and has a coverage score of 0.687, and the optimized DBo considers 11 data attributes with a coverage score of

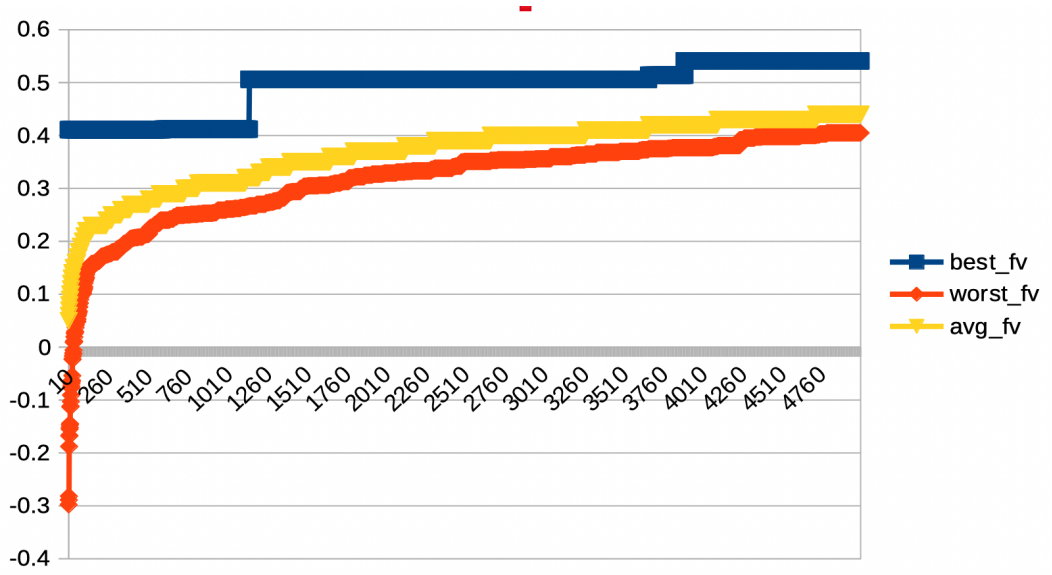



Figure 3.26: A typical run of the GA (population size of 50 and over 5000 generations): each curve represents respectively, the best (in blue), the average (yellow) and the worst (red) fitness values of the population. This run takes around 1.4s.

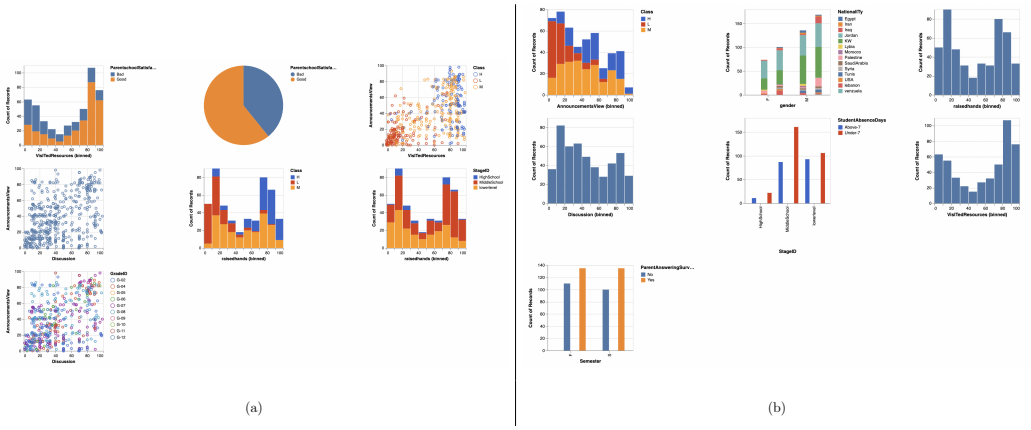
Table 3.3: Fitness value analysis: DBo (Initial Population) vs DBo (after GA run)

| *DBo/GAparams* | Match Score | Coverage | Complexity | Fitness Value |
|---|---|---|---|---|
| DBo (Initial pop) | 0.977 | 0.687 | 0.571 | 0.099 |
| DBo (After GA) | 0.970 | 0.928 | 0.171 | 0.729 |

0.928. This optimized DBo covers all 6 important data attributes ( Raised hands, VisITed resources, Announcements view, Discussion, StudentAbsenceDays, Class) and includes all the 3 secondary attributes (Gender, Nationality, StageID), and 2 of the low-importance attributes were selected. In contrast, the initial DBo considers only 5 important data attributes (Raised hands, VisITed resources, Announcements view, Discussion, Class), 1 secondary attribute (StageID) and 2 low importance attributes. The comparative analysis of covered data attributes is presented in Table 3.4.

- The final DBo also has a matching score (0.970) that is similar to the initial DBo (0.977). as shown in Table 3.3 second column.

Indeed those results should be extended and improved. We have studied the automatic generation of DBo. We have proposed a complete framework with a GA



(a) (b)

Figure 3.27: Two examples of DBo generated with our tool and for behavior analysis scenario. In (a), a DBo generated during the initial population with a fitness value of 0.099, and in (b) a final DBo generated after 20000 generations with a fitness value 0.729. This run takes around 6.4 seconds.

Table 3.4: Data attribute analysis: DBo (Initial Population) vs DBo (after GA run)

| $DAcovered/DBo$ | DBo (Initial pop) | DBo (After GA) |
|---|---|---|
| Total DA | 8 | 11 |
| Primary DA | 5 | 6 |
| Secondary DA | 1 | 3 |
| Low DA | 2 | 2 |

that can optimize a set of co-adapted visualizations: we have defined several models to represent data, users, and visualizations. Several genetic operators were defined, as well as a fitness function. Starting from a set of visualizations, we could use again an optimization method to find a relevant layout for the DBo. Finally, our GA is integrated into a platform devoted to Business Intelligence for novice users as presented in the next chapter 4.

Chapter 4

# User interface for dashboard recommendation

## Contents


**4.1 System design . . . . . 70**
4.1.1 System requirements and goals . . . . . 70
4.1.2 System Architecture . . . . . 71
4.1.3 Tech Stack Selection . . . . . 72
**4.2 Data upload and in-build dataset . . . . . 72**
**4.3 System manual editing and parameter setting . . . . . 74**
4.3.1 Metadata . . . . . 74
4.3.2 User data . . . . . 75
4.3.3 User preferences setting . . . . . 76
**4.4 Data visualization and dashboard recommendation . . . . . 78**
4.4.1 Main Dashboard and its control features . . . . . 78
4.4.2 Parameter Setting . . . . . 83
4.4.3 Five best recommended dashboards . . . . . 85
4.4.4 Personalized dashboard . . . . . 86
4.4.5 Dashboard Rating System and Additional Help . . . . . 87
4.4.6 Miscellaneous features . . . . . 90
**4.5 Conclusion . . . . . 95**


The system 'SimpleVis' contributes to the automatic recommendation of dashboards. It aims to be easily used by novice users. The system supports three types of users: novices, beginners, and experts.

As mentioned in the state-of-the-art chapter 2, we found 19 ADRSs. However, none of them met the needs of novice users for easy and quick data analysis with automatic dashboard recommendations. This motivation led us to develop the new automatic dashboard recommendation tool. The tool's persona should cater to novice users, providing support for data visualization and analysis with minimal user intervention. Novices do not have much domain knowledge to perform manual editing so automatic dashboard recommendation will be helpful for them with some optional selection features.

'SimpleVis' consists of distinguished features were with respect to existing ADRSs. The features configured based on user type selection. The system is completely automatic but also supports interactive features like parameter setting, user preferences selection and manual editing. These interactive features are very helpful for users. In case of novice users system recommends dashboards in three simple steps which takes less than a minute. 'SimpleVis' allows parameter setting and user preferences selection such as visualization selection, setting range of visualizations, user type, assigning user importance for data attributes. Considering the expert users system manual editing features: changing data attribute type keeping visualization constant, changing visualization keeping data attribute type constant, manually assigning user importance for data attributes. Expert user can also configure GA operators and fitness function coefficients as well as operators. Dashboard rating system, creating personalized DBos, and detailed DBo information features make 'SimpleVis' more elegant and user friendly.

## 4.1 System design

Following the literature gap in ADRSs for novices as mentioned in section 2.5.8 of chapter 2, we set the system design requirements and goals, architecture and tech stack for the system development. The system designs are set considering novice users.

User can visualize the data with dashboard recommendation within three simple steps as follows.

- Step 1 is data upload or select from in-build dataset (screen 1).
- Step 2 is metadata editing and parameter setting with user preferences (screen 2).
- Last one is step 3 to visualize the recommended dashboards (screen 3).

Above three steps need to be perform by user automatic data visualization and dashboards generation are based on the systems design: requirements and goals, architecture and tech stack as mentioned in section 4.1.1, section 4.1.2, and section 4.1.3.

### 4.1.1 System requirements and goals

We set the following design requirements and goals for system development:

1. Simple and fewer efforts: The system should be very simple. It must take just three simple steps for dashboard recommendation and data visualization.
2. Fast: The system should take less than a minute to recommend dashboards.
3. Recommendations of several DBos: It should recommend several possible dashboards to the user.

4. Completely Automatic: The System should be completely automatic. It does not require any necessary user interventions except uploading the data file.

5. Manual editing: The system should support the basic manual editing feature for metadata, visualizations, and DBo. User should be able to edit the recommended DBo so as to adapt them to user needs.

6. Parameter setting: User can change different parameters as per their need for better DBo recommendations.

7. Personalized DBo: System should support personalized DBo feature. So that users can create desired and personalized DBo from the recommended DBo. User should be able to save them so that they can rerun GA for more recommendations.

8. Feedback and suggestions: The user can show satisfaction by rating the recommended DBo. The system will recommend feedback and suggestions to user and implement them. This will allows the system to recommended expected DBos.

9. Interactive user interface: The complete system, must support a few interactive features like zoom-in, hover, double-click, drop-down, etc. These features can be user type and data attribute importance selection. The system must support basic CRUD (create, remove, update, and delete) features on DBo and visualization.

10. Multiple user types support: The system should recommend a dashboard based on user type selection. The system should support different types of users.

11. Dashboard Rating: User can do manual editing for correction, ask for help, and follow the system suggestions by rating the DBo for different parameter setting.

### 4.1.2 System Architecture

We began with the system wireframe and its workflow architecture. We created the system wire frame using a professional tool "Figma". The design part includes the selection of the color palette, icon, bottoms, images, and tool content. We have focused on the placement of components on each screen. The first screen of the tool contains 4 main components (data upload, system features, publications, and authors), the second screen contains 3 main components (metadata, user data, and system preferences) and the third screen contains 3 main components (main dashboard, parameters setting and other dashboards).

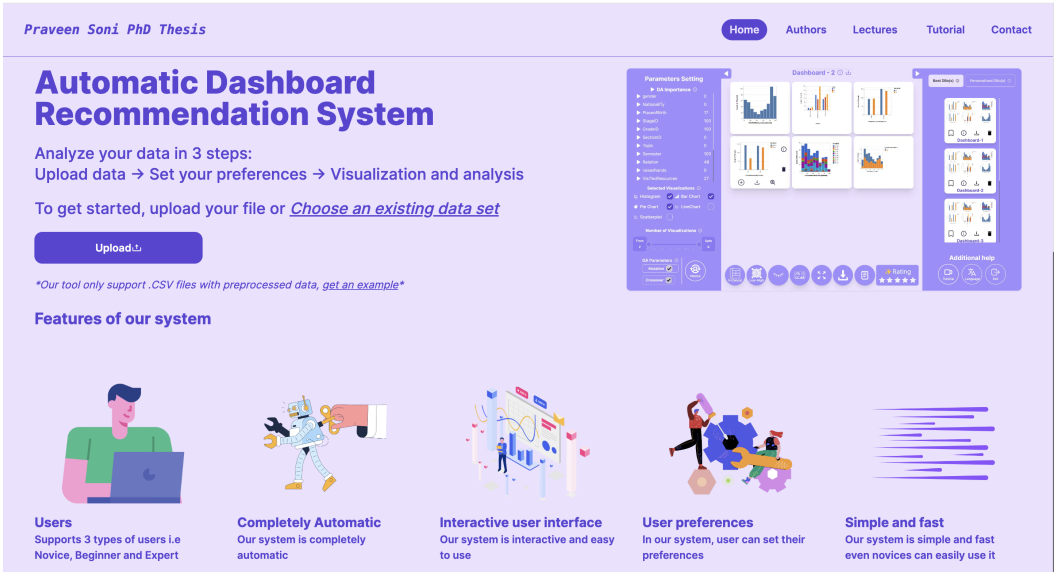



Figure 4.1: SimpleVis system welcome screen

### 4.1.3 Tech Stack Selection

Considering the features of our system we have focused on a tech stack that can handle multiple users and fast computation. Our selection is as follows:

- Frontend: The complete frontend of software is developed in the Next.js framework of JavaScript. Next.js is fast because it supports server-side rendering.
- Backend: The backend of the software is developed in Node.js. Node.js is an asynchronous, event-driven, and non-blocking programming language. It is a better option for speeding up the development process.
- Vega Lite: For the data visualization we have used vega lite framework. Vega Lite act as an bridge between Node.JS and Next.JS.
- Database: We do not use a specific database for storing user-uploading files. We have applied all operations to the CSV file uploaded by the user. We store the file storing as a 2D array. We used the MySQL database to store user feedback and contact forms.
- Server and hosting: The complete code both backend and frontend is hosted and deployed on the AWS server.

## 4.2 Data upload and in-build dataset

The data upload feature of the system has the following goals:

1. User can choose any one of 5 inbuild datasets for demo

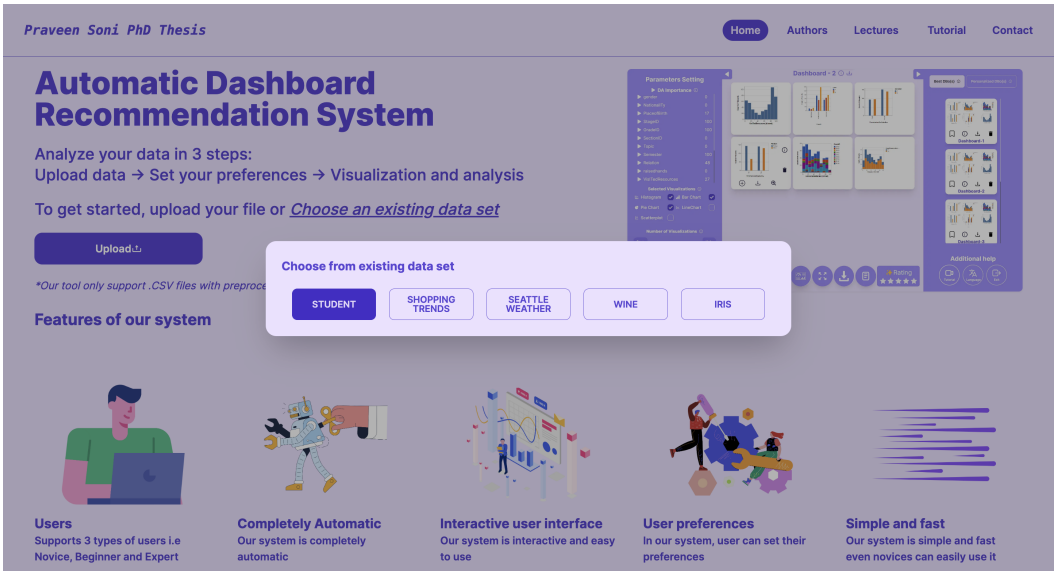



Figure 4.2: Data upload: There are 5 in-build datasets for testing SimpleVis tool

2. User can upload his/her own dataset

The system allows the user to analyze the data and use it in two ways. One way is to simply select any one of the prepossessed datasets. Users can select any dataset from 5 in-build datasets for testing. The datasets can be selected by clicking on the button "Choose and exiting data set", as shown in Figure 4.1. When the user clicks on this button a pop-up will appear with 5 different datasets as shown in Figure 4.2. Users can select any one of them to go to step 2. Another way is by uploading their own new dataset. Users can click on the button "Upload" and then they can select any dataset from the local machine, as shown in Figure 4.3. The data file must be in .csv format.

There are five in-build datasets Student, Shopping Trends, Seattle Weather, Wine, and Iris:

- Student [Pauwels *et al.* 2009, Schwendimann *et al.* 2016], that contains 432 rows and 17 columns with 3 data attribute types (Ordinal, Nominal, Numeric).
- Shopping Trends [Sourav Banerjee 2023], that contains 3000 rows and 10 columns with 4 data attribute types (Ordinal, Nominal, Numeric, Temporal).
- Seattle Weather [ANANTH R 2021], that contains 1462 rows and 6 columns with 3 data attribute types (Ordinal, Nominal, Numeric).
- Wine [Sebastian Raschka 2014], that contains 178 rows and 14 columns with 3 data attribute types (Nominal, Numeric).
- Iris [MathNerd 2018], that contains 150 rows and 5 columns with 2 data attribute types (Nominal, Numeric).

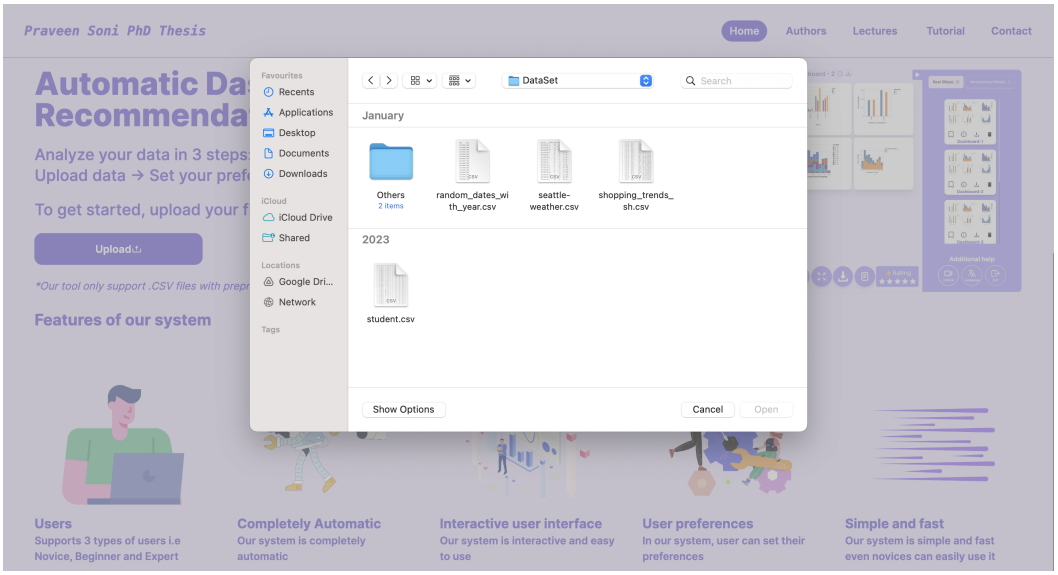



Figure 4.3: Data upload: uploading new dataset from local machine (the format of uploaded file must be .csv)

## 4.3 System manual editing and parameter setting

In the second step, the screen as shown in Figure 4.4 is divided into three parts: (A) Metadata, (B) user data preview, and (C) User preference setting. The second screen of the system has the following goals:

1. User can perform manual editing of user importance in metadata,
2. user can see and verify the uploaded data,
3. design and selection features for metadata editing,
4. design and selection features for user dashboard preference.

### 4.3.1 Metadata

Metadata is the statistical and semantic information extracted from the uploaded data for dashboard recommendations. The metadata contains the data attribute name, unique value, data attribute type, and user importance as shown in Figure 4.5. Each of the extracted properties is explained as follows:

- Data attributes name: These are data attributes present in the uploaded dataset. Users can only view them but cannot edit them.
- Unique value: It represents the total number of unique values present in the particular column of each data attribute. Users can also only view them but cannot edit them.

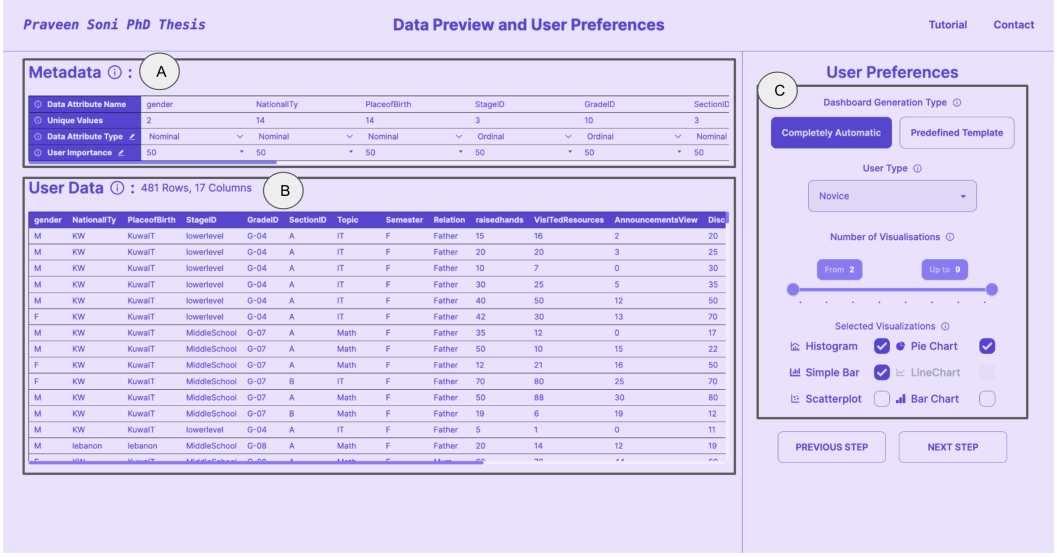



Figure 4.4: On the left top (A) Metadata and on the left bottom (B) User data preview, with (C) user preferences setting on the right.

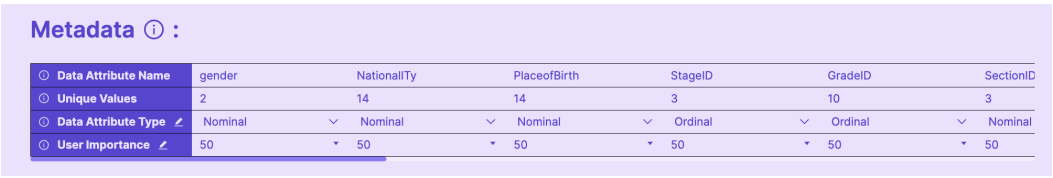



Figure 4.5: The first two rows of metadata are fixed: Data attribute name and Unique values. User can edit the last two rows: Data attribute types and user importance.

- Data attributes type[1]: It represents the type of each data attribute. Users can edit them based on domain knowledge. There are four possible options to select data attribute types: Numeric, Nominal, Ordinal, and Temporal.
- User importance: The user importance value represents the importance of particular data attributes during dashboard recommendations. There are five possible options to select user importance value: Exclude (0), Less Importance (25), average (50), More Importance (75), and Mandatory (100). Users can also have the possibility to enter any integer value between 0 to 100.

### 4.3.2 User data

The user data is the preprocessed dataset uploaded by the user. It represents the total number of rows and columns of uploaded data as shown in figure 4.6. The

[1]The "automatic selection of data attribute type for newly uploaded data" in metadata section, was supposed to be done by other work package of the project BI4PEOPLE.

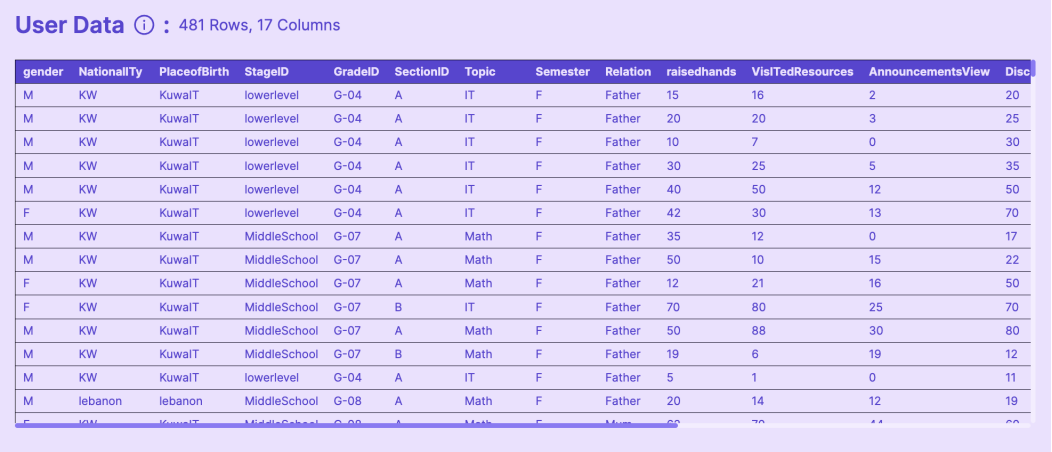



Figure 4.6: User dataset uploaded by user. It represents rows and columns in a dataset.

system allows users to preview and check the uploaded dataset.

### 4.3.3 User preferences setting

The user can set user preferences (see figure 4.7). These user preferences allow users to get customized and preferred dashboard recommendations. There are four sets of user preferences: (A) Dashboard Generation Type, (B) User Type, (C) Number of Visualizations, and (D) Selected Visualizations.

- (A) Dashboard generation type (automatic and predefined templates): The automatic dashboard recommendation is based on user preference and genetic algorithm. In predefined templates, the user preference will not be taken into account to recommend the dashboard. Users will able to choose one of the templates based on uploaded data (this function is not implemented yet, see the perspective in Chapter 6).

- (B) User type: Based on previous knowledge and experience in data analysis a user can select his/her type. There are three types of users: novices, beginners, and experts.

- (C) Number of visualizations: Users can set the range of the number of visualizations to be present in a recommended dashboard. Both bounds are inclusive.

- (D) Selected visualizations: Users can select the desired visualizations to be present in the recommended dashboard. The system allows the multiple visualization selection.

At the bottom of screen 2, there are two buttons. Users can go to the next step or the previous step.

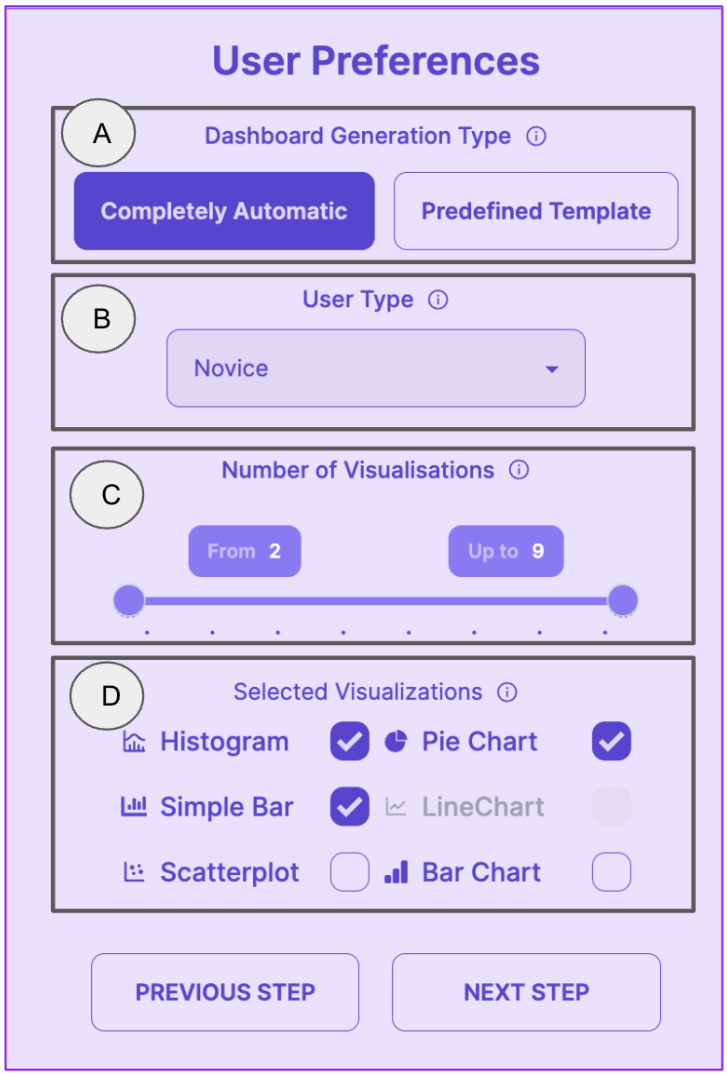



Figure 4.7: Four sets of user preferences: (A) Dashboard Generation Type, (B) User Type, (C) Number of Visualizations, and (D) Selected Visualizations

## 4.4 Data visualization and dashboard recommendation

In our system, the user can visualize the uploaded dataset and analyze it with the help of a recommended and personalized dashboard. The data visualization and dashboard recommendation screen of the system has the following goals:

1. Recommend best five dashboards, and present the best among the 5 in the center of the screen with the larger space.
2. Enable the user to do basic manual editing of some parameters like Data attributes importance, Selected Visualizations, Number of Visualizations, and user type selection likewise in step 2. Then the user can rerun the GA.
3. Offer advanced features for expert users like genetic algorithm configuration and User type selection with fitness function configurations.
4. User can create a personalized dashboard with manual editing.
5. Include buttons for implementing features on the main dashboard and its controls.
6. Provide users with the ability to give suggestions and feedback for improving dashboard generation, based on their rating of the existing dashboard.
7. Incorporate additional features such as help resources, a system evaluation form, and options for contact and support.

In the third step of dashboard visualization, the "SimpleVis" screen is divided into 11 components as shown in Figure 4.8: (A) Main Dashboard as generated by GA (B) Data attributes and their importance, (C) Selected Visualizations to be appear in the generated DBos(D) Number of Visualizations, (E) GA Parameters, (F) User type configuration (G) the five best Dashboards, (H) Personalized Dashboards, (I) Main DBo features to control, and (J) Rating system for feedback and (K) Additional Help.

### 4.4.1 Main Dashboard and its control features

**(A) Main dashboard.** This is the best dashboard recommended by the system based on user preference and genetic algorithm. This DBo is present in the center of the screen. The main dashboard contains a set of visualizations in a range of between 2 to 9. In this case, it is 9 visualization as shown in Figure 4.8. On the top of the main dashboard, it shows the dashboard name and two icons: the info icon and the download icon. These two have the following interactive features for debugging or expert user:

- Info Icon: When the user clicks on the info icon, it comes up with a pop-up window and shows two insights as shown in Figure 4.9. First, it shows detailed information about the dashboard fitness evaluation used for dashboard




Figure 4.8: Screen of step 3 contains 11 components: (A) Main Dashboard as generated by GA (B) Data attributes and their importance, (C) Selected Visualizations to be appear in the generated DBos (D) Number of Visualizations, (E) GA Parameters, (F) User type configuration (G) the five best Dashboards, (H) Personalized Dashboard, (I) Main DBo features control, and (J) Rating system for feedback and (K) Additional Help.

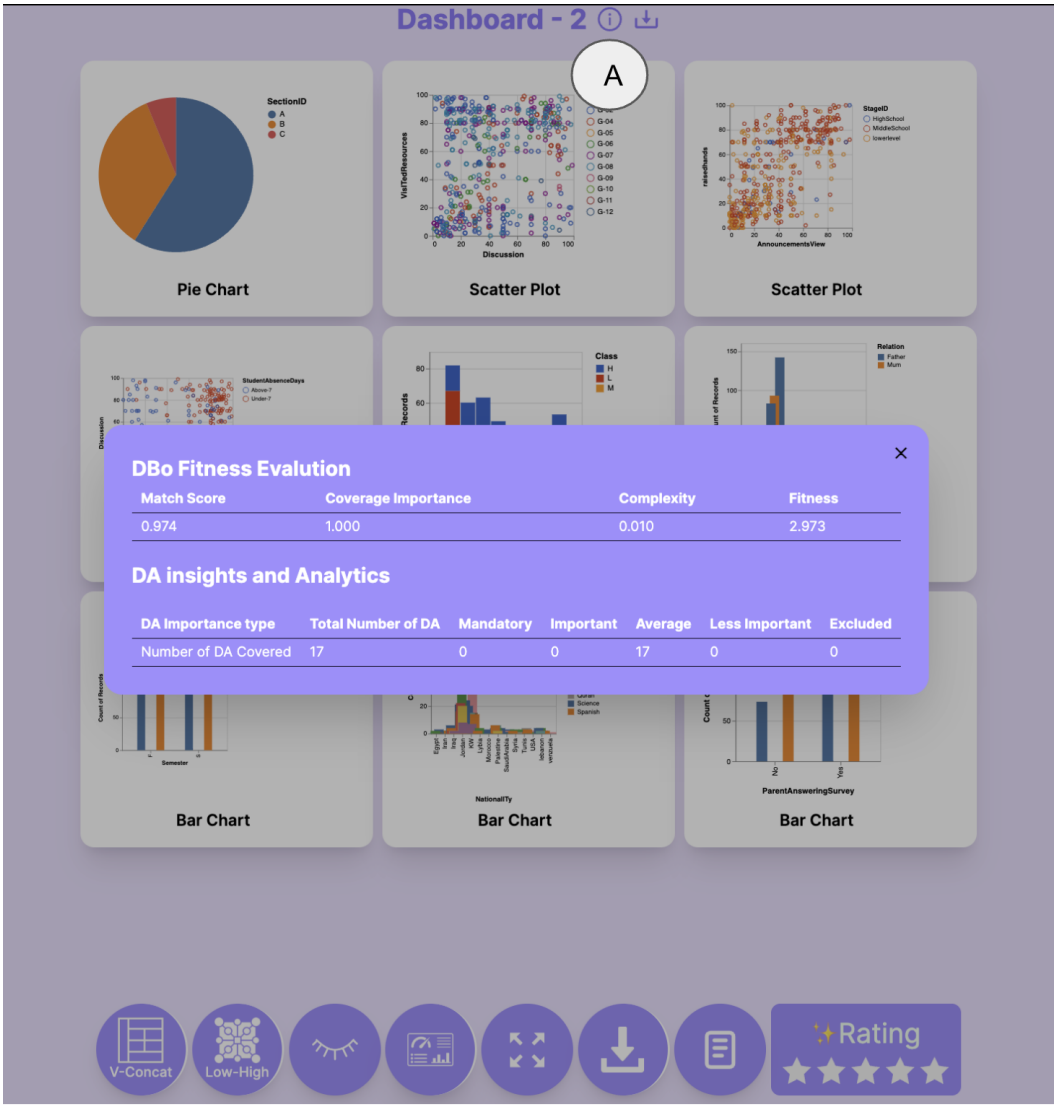



Figure 4.9: When a user clicks on (A) Info Icon it shows pop up with information: Detailed information about the dashboard fitness evaluation and data attributes information.

evaluation. In the second line of the pop-up, it shows the 'data attributes information' for that particular dashboard.

- Download icon: The download button allows the user to get the GA population and dashboard evaluation files in Excel format. This feature is only for expert users.

Each visualization of the main dashboard contains the following 5 individual icons as shown in figure 4.10. These icons allow users to interact with visualization and apply different functionality. There are two additional functionalities when the user hovers and single clicks on each visualization. The functionalities of each icon are detailed below:

- (A) Add: When the user clicks on the add button the visualization will be moved to personalized DBo.
- (B) Download: You can download the visualization on a local machine in SVG format.
- (C) Full Screen: The user can zoom in on the individual visualization. It also allows user to edit the data attributes present in a selected visualization individually.
- (D) Delete: You can delete that particular Visualization from the dashboard.
- (E) Information: The user can see other detailed information and learn about that particular visualization (link will be given in pop-up to learn more details)
- Hover: The user can hover on visualization to see interactive options.
- Single click: The user can make a single click to visualization to modify/look into its covered data attributes and their user importance present on the left side of the screen as shown in figure 4.11. When user click on the scatter plot (present in third column of first row), all three data attributes (AnnouncementsView, VisITedResources, and Relation) present in the scatter plot will be appear on the left side of screen user 'DA Importance 'to edit the 'user importance' of it.

**(I) Main dashboard features control.** At the bottom of the main dashboard. There are six icons for implementing interactive functionality to the main dashboard. Each of them is as follows:

- Notes: The user can take manual notes for each of the dashboards.
- Download: The user can download the dashboard on a local machine in SVG format.




Figure 4.10: Each visualization of the main dashboard has 5 interactive icons with different functions. (A) Add visualization to personalized DBo, (B) Download this visualization, (C) View in full screen and edit, (D) Delete from this DBo, (E) Information about this visualization for detail study. There are two additional functionalities when the user hovers and single clicks on each visualization.




Figure 4.11: Each visualization of the main dashboard has 5 interactive icons with different functions. There are two additional functionalities when the user hovers and single clicks on each visualization.

- Full Screen: The user sees all the visualizations only in the dashboard in full screen for better analysis.
- Re-run GA: The user can re-run the genetic algorithm after making necessary and desired modifications on the 'parameters setting'.
- Show/Hide icons: The user can click on the show icon to see all the icons present on each visualization for user interaction as a toggle button.
- Complexity: The user can arrange the visualization of the dashboard in two ways according to complexity values: 'High to Low' and 'Low to High' as a toggle button.
- Layout: It will suggest the best and most optimized dashboard layout for the screen. This feature is not implemented yet, so for the moment we keep a 3*3 matrix layout. The width and height of all visualizations are the same.

### 4.4.2 Parameter Setting

Likewise in step 2, we have also given a manual editing feature for the user in step 3. Users can perform manual editing for the data attributes' importance, visualization selection, number of visualizations, genetic algorithm parameters configuration, and user type selection with fitness function configuration. Each of the following features is described below (see figure 4.8):

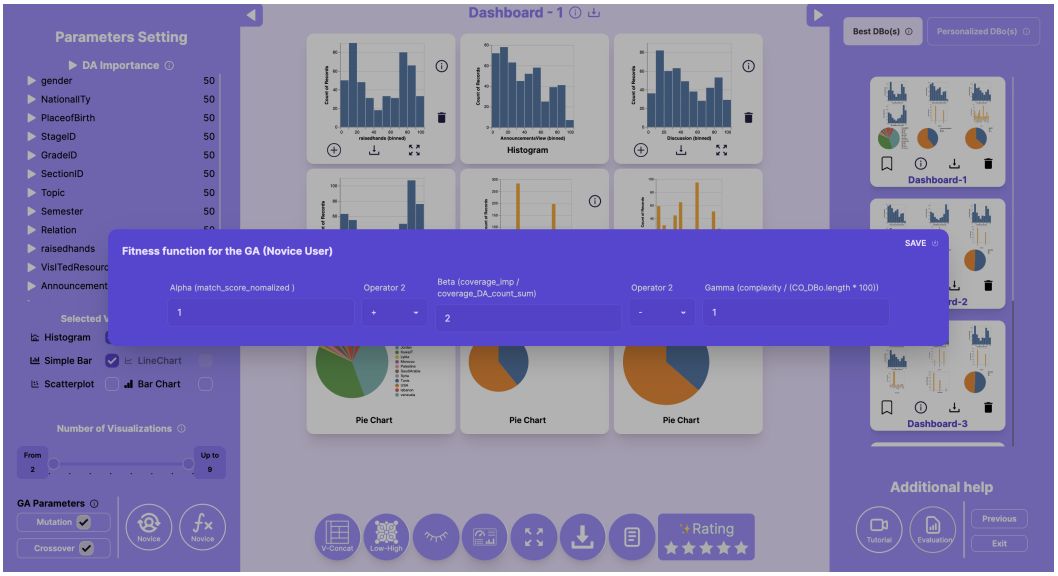



Figure 4.12: There are 5 best dashboards that are automatically recommended by the system. It contains interactive features like saving, viewing information, downloading, and deleting.

**(B) Data attributes importance.** The user importance values represent the importance of particular data attributes during dashboard recommendations. It lets the user to specify which data attribute should/ must/ must not appear in the DBo.

**(C) Selected visualizations.** Users can select the desired visualizations to be present in the recommended dashboard. There are 6 visualizations: Histogram, Pie Chart, Simple Bar, Line Chart, Scatter Plot, and Bar Chart.

**(D) Number of visualizations.** Users can set the range of the number of visualizations to be present in a recommended dashboard (both bounds are inclusive).

**(E) Genetic algorithm parameters.** It represents the selection of different GA operators (mutation and crossover) that are used to optimize dashboard recommendations. Users can select or skip different GA operators as per their choice for automatic dashboard recommendation.

**(F) User type selection with fitness function configuration.**

- User type: Based on previous knowledge and experience in data analysis. A user can select his/her type. The user types are three types Novices, beginners, and experts.

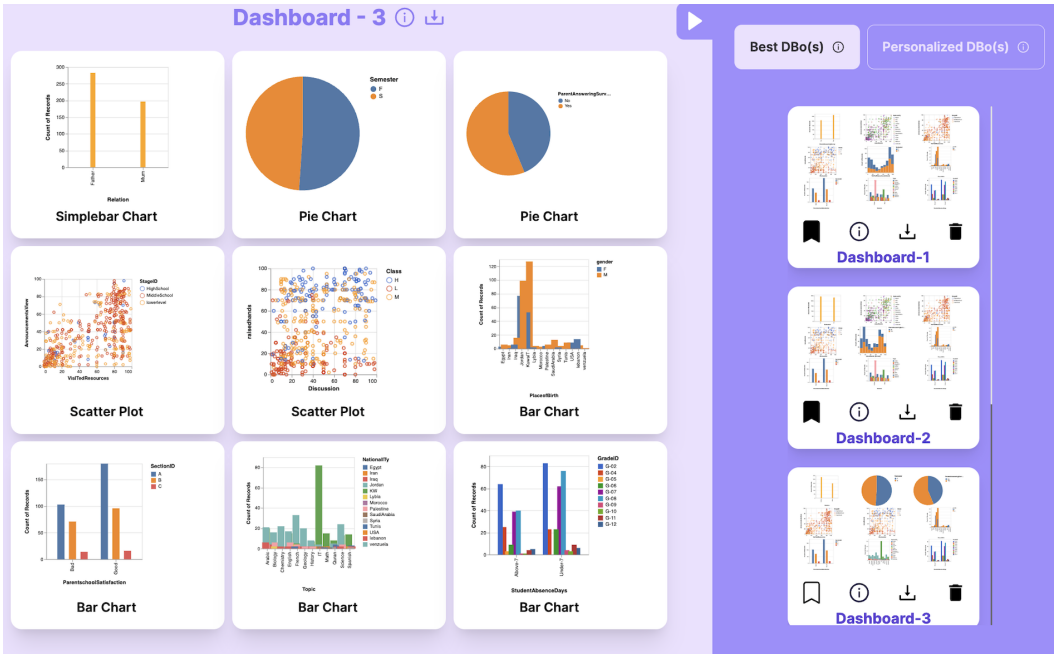



Figure 4.13: There are 5 best dashboards that are automatically recommended by the system. It contains interactive features like saving, viewing information, downloading, and deleting.

- Fitness function configuration: The user can change fitness function coefficients and operators between 2 fitness parameters as needed. There are four possible parameters subtraction, add, multiplication, and division as shown in figure 4.12.

### 4.4.3 Five best recommended dashboards

The system recommends five best dashboards based on genetic algorithm (as presented in the previous chapter 3). It is possible with the multimodelity property of GA. The personalized dashboard feature allows users to create customized dashboards from visualizations present in the best dashboard. Both best and personalized dashboard features are shown on the right side of Figure 4.8.

**(G) Best dashboard.** These are 5 dashboards that are initially recommended by the system using metadata and user preference as shown in Figure 4.13. Each best dashboard has the following interactive features:

- Save: The user can save the particular Visualization from the dashboard using the save icon.

- Information: The user can see other detailed information about the fitness function used for dashboard evaluation for that particular dashboard.

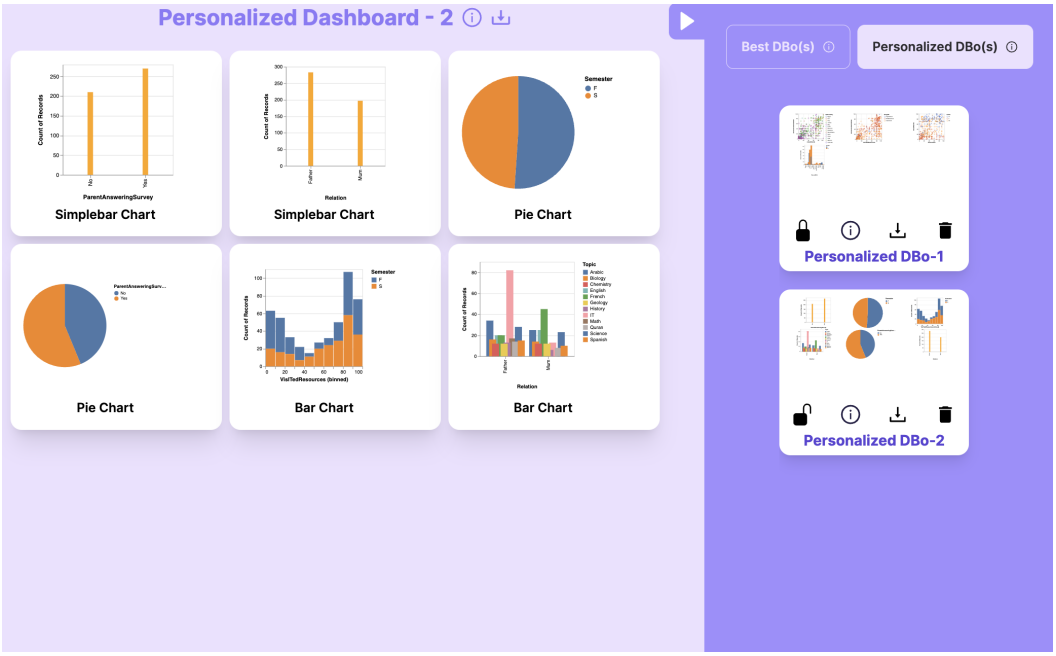



Figure 4.14: Personalized dashboard can be generated from 5 best-recommended dashboards. It contains interactive features like locking/unlocking the dashboard, viewing information, downloading, and deleting.

- Download: The user can download the dashboard on a local machine in SVG format.
- Delete: The user can delete that particular dashboard from the list of best-recommended dashboards.

### 4.4.4 Personalized dashboard

**(H) Personalized dashboard.** The user can create these personalized dashboards as per their needs. Users can create a maximum of 5 personalized dashboards. When the user click on any of the dashboards present on the right panel (under 'Personalized DBo' tab), it will appear at the center. Each personalized dashboard has the following interactive features icons as shown in figure 4.14 (for the detail explanation of creating the personalized dashboard see section A.2).

- Lock/Unlock: The user can 'lock' and 'unlock' the dashboard. Users can start adding visualizations from any of the 5 best dashboards. Once the desired personalized dashboard is done user can lock it. If the user wants to add a new visualization in the same dashboard, it needs to unlock it.
- Information: The user can see other detailed information about the fitness function used for dashboard evaluation for that particular dashboard.

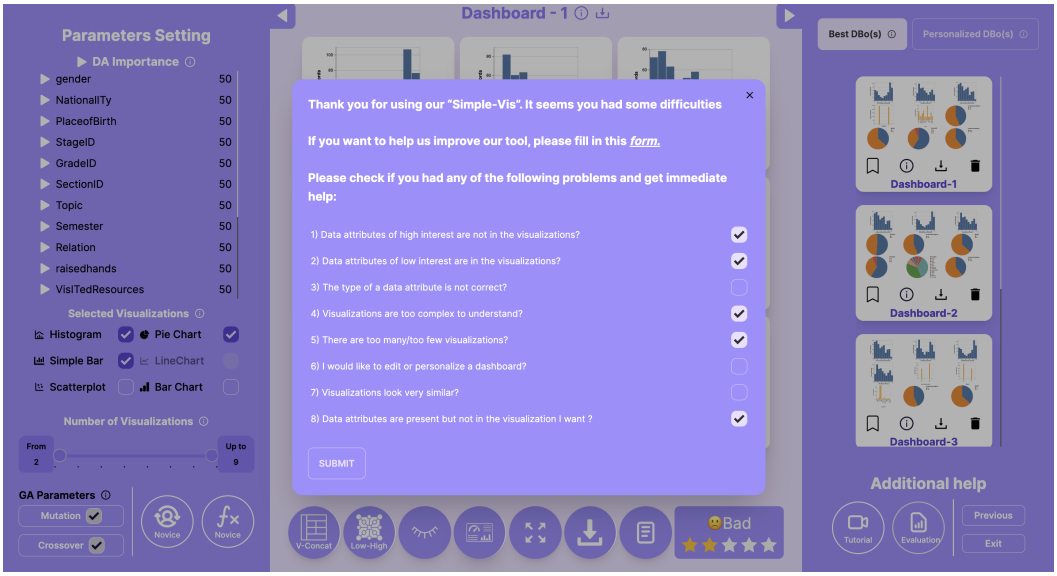



Figure 4.15: Rating system for dashboard shows that user selected problems 1, 2, 4, 5, and 8 out of a total 8 problems for dashboard recommendation improvements

- Download: The user can download the dashboard on a local machine in SVG format.
- Delete: The user can delete that particular dashboard from the list of personalized dashboards.

### 4.4.5 Dashboard Rating System and Additional Help

The system rating feature helps the user to improve the dashboard recommendation. The additional help contains tutorials, evaluation, and system navigation, (see figure 4.8 for all features).

**(K) Rating system for dashboard.** The rating system for the dashboard contains 8 problems. Users can select multiple problems and the system will give suggestions for dashboard recommendation improvements. In Figure 4.15 user selected the problems 1, 2, 4, 5, and 8 out of a total of 8 problems for dashboard recommendation improvements. In Figure 4.16 the rating system suggested solutions for the problem 1, 2, 4, 5, and 8. The rating system for dashboard improvements has the following problems and their respective solutions:

**Problem 1:** Data attributes of high interest are not in the visualizations?
**Solution 1:** Please try to adjust the attributes' importance in the "Data attributes importance" panel. Increase the importance to 100 for those attributes

**Problem 2:** Data attributes of low interest are in the visualizations?

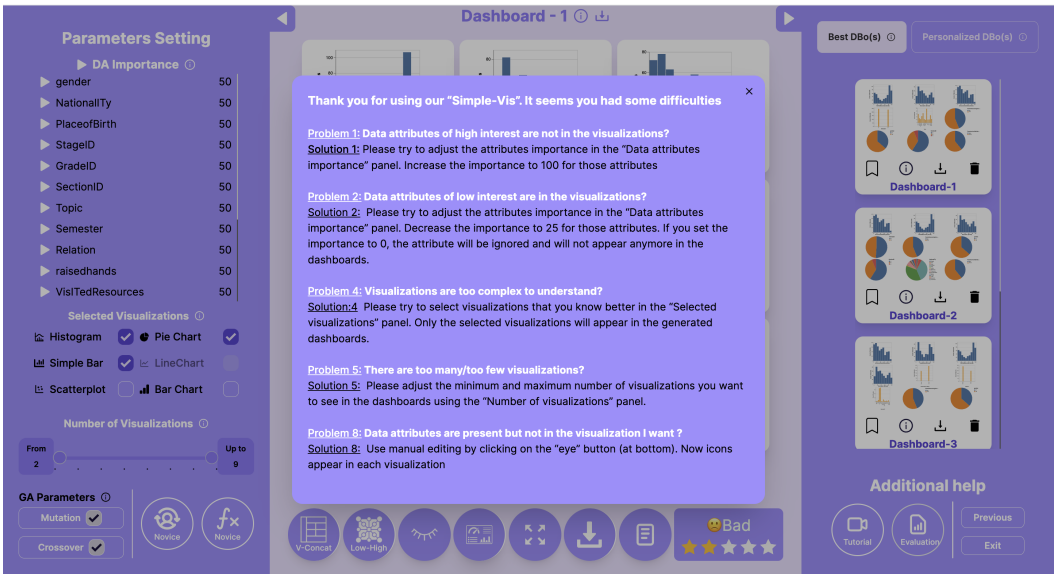



Figure 4.16: Rating system for dashboard shows the suggested solutions for problems 1, 2, 4, 5, and 8 for dashboard recommendation improvements.

**Solution 2:** Please try to adjust the importance of the attributes in the "Data attributes importance" panel. Decrease the importance to 25 for those attributes. If you set the importance to 0, the attribute will be ignored and will not appear anymore in the dashboards.

**Problem 3:** The type of a data attribute is not correct?
**Solution 3:** Please go to the home page, and reload your data. You will have the possibility to change the metadata about data attributes.

**Problem 4:** Visualizations are too complex to understand.
**Solution 4:** Please try to select visualizations that you know better in the "Selected visualizations" panel. Only the selected visualizations will appear in the generated dashboards.

**Problem 5:** There are too many/too few visualizations.
**Solution 5:** Please adjust the minimum and maximum number of visualizations you want to see in the dashboards using the "Number of visualizations" panel.

**Problem 6:** I would like to edit or personalize a dashboard.
**Solution 6:** Please click on the "eye" button. Icons now appear in each visualization. Delete: You can remove any visualization from the dashboard. Change: You can change the data attributes present in a visualization Add +: You can add the visualization to a personalized dashboard (see the upper right corner of the interface)

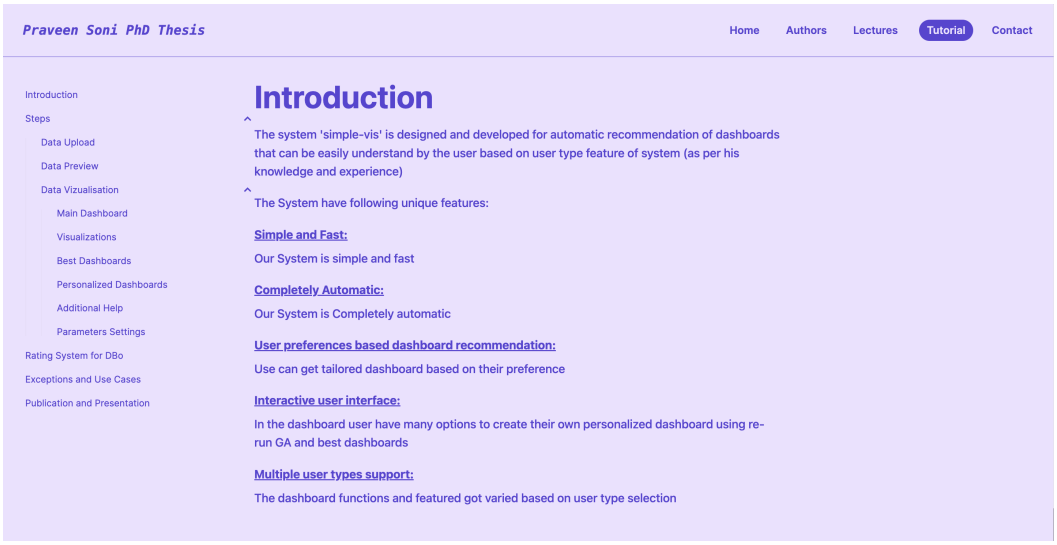



Figure 4.17: The system tutorial acts as the complete guide for the new users to make them familiar with system uses and features.

**Problem 7:** Visualizations look very similar?
**Solution 7:** Please adjust the minimum and maximum number of visualizations you want to see in the dashboards using the "Number of visualizations" panel.

**Problem 8:** Data attributes are present but not in the visualization I want.
**Solution 8:** Use manual editing by clicking on the "eye" button (at the bottom). Now icons appear in each visualization

**(H) Additional Help.** The Additional Help helps users with navigation. It contains a system tutorial and system evaluation. The system tutorial acts as the complete guide for the new user as shown in Figure 4.17. Users can also evaluate the system in this section.

- Tutorial: A new user can learn the tool with the help of a tutorial. The complete tutorial is divided into a total of 5 main sections as shown in Figure 4.17. The 'steps' section and 'data visualization' section have further subsections.
- Evaluation: The user can evaluate our tool and tell us their experience with the evaluation form. This form also helps us improve the tool and meet user expectations. As shown in Figure 4.18 and 4.19 the evaluation form is divided into two sections. The first section contains basic information about the user: Full name, Organization, email address, and user type. The second section contains 11 questions related to tool experience and feedback. For details see the Appendix B.
- Previous: The user can go back to the previous step, i.e. screen 2.

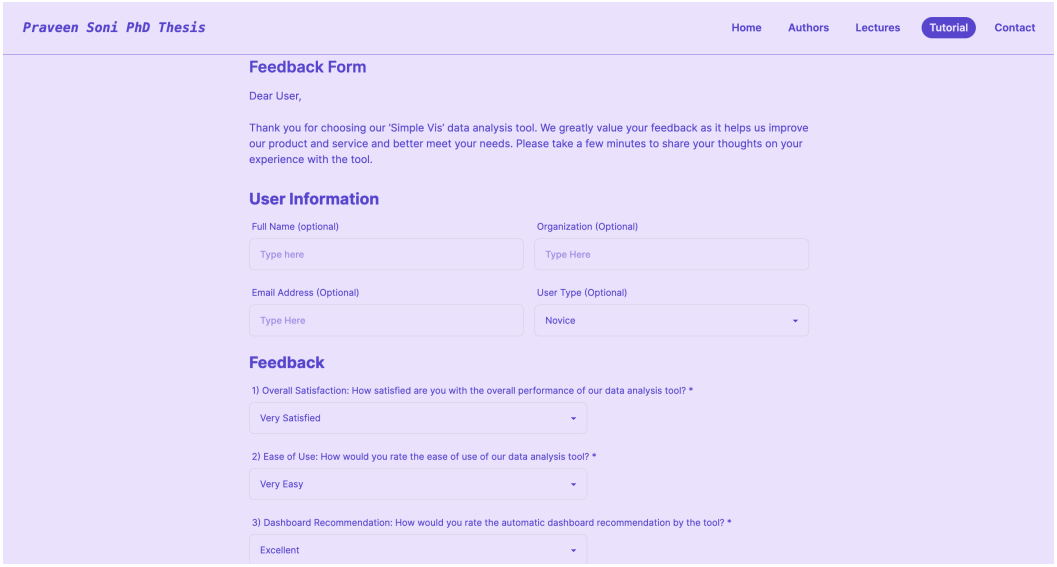



Figure 4.18: System evaluation form with section 1 question and section 2 question from Q1 to Q3.

- Exit: The user exits from the process of data analysis and goes back to step one i.e. first screen.

- Contact Us: If the user needs any help and has a suggestion can use the "Contact Us" form as shown in Figure 4.20. Users can reach to Contact Us form by clicking to tutorial button.

### 4.4.6 Miscellaneous features

The system supports some miscellaneous features which are as follows:

**Visualization manual editing in full-screen mode.** Users can zoom in on each visualization to perform manual editing. These features allows uses to 'select alternative data attribute keeping visualization constant' and 'select alternative visualization keeping data attribute constant'.

1. Alternative data attribute keeping visualization constant: This feature allows user to selects the 'alternative data attributes keeping visualization constant'. This feature is supported by all the six visualizations. To understand this feature consider and example of scatter plot as shown in Figure 4.21. The scatter plot contains the 'edit icon' in red box. When user click on 'edit icon' , three visual attributes (X, Y and color) will appear on top of scatter plot as shown in green box. These visual attributes have corresponding data attribute for possible mapping as shown in Figure 4.22. Like in case of 'color' (i.e. legend)

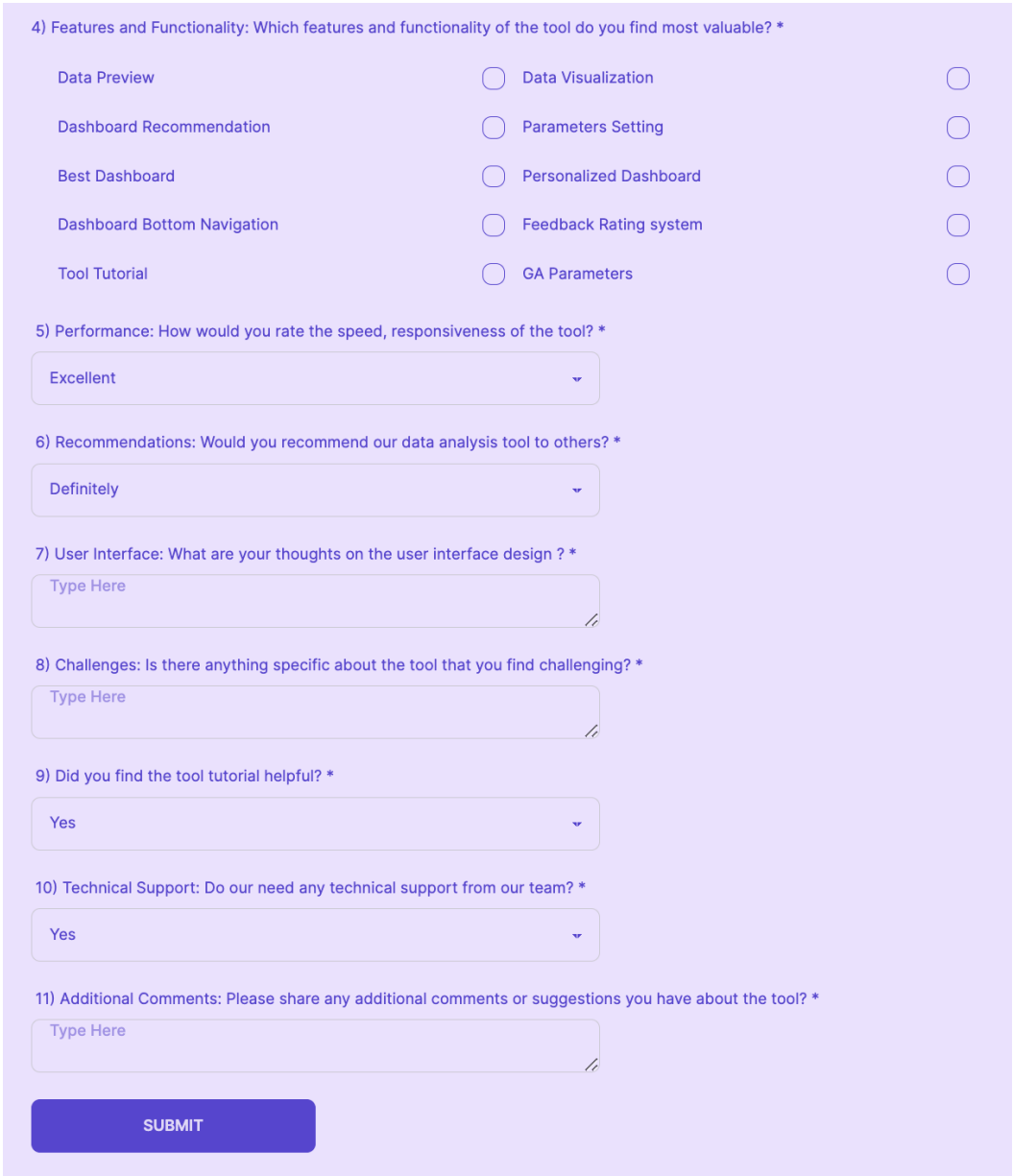



Figure 4.19: System evaluation for with question from Q4 to Q11 of section 2

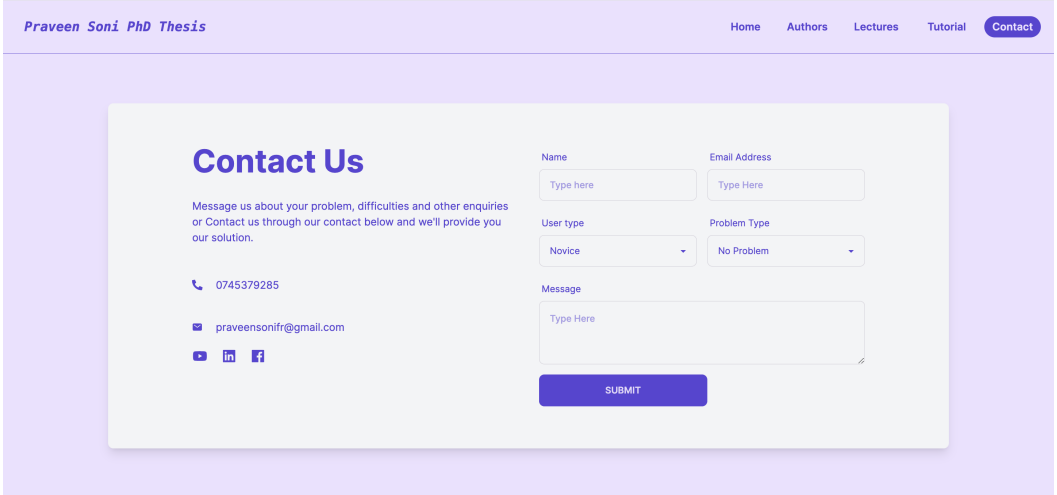



Figure 4.20: Contact Us form for user help and support

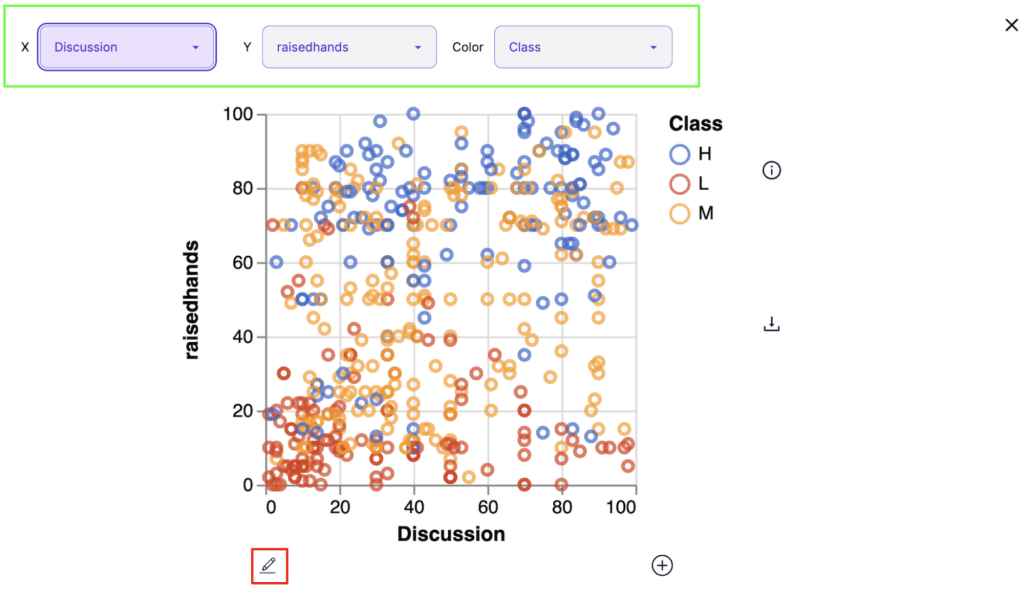



Figure 4.21: Scatter plot with alternative data attribute features keeping visualization constant. When user clicks on edit icon (shown in red box), an window will appear at top of visualization with data attributes corresponding to visual attribute (shown in green box). When user clicks on drop down of respective data attribute all alternative data attribute of same type will appear.

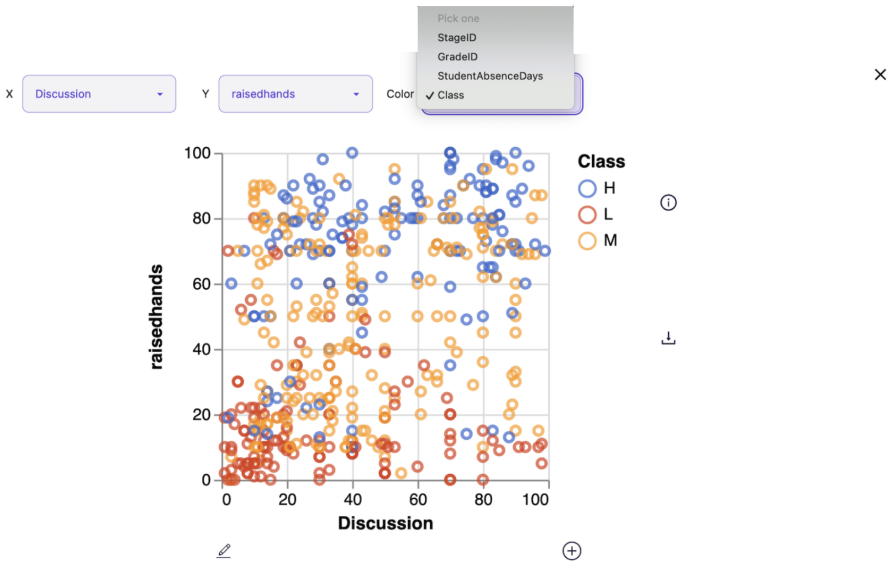



Figure 4.22: In this Scatter plot, the visual attribute 'color' have total 4 data attributes options for mapping.

user have 4 data attributes options: Stage ID, Grade ID, StudentAbsenceDays, and Class.

2. Alternative visualization keeping data attribute constant: This feature allows user to selects the alternative 'visualization' keeping 'data attributes' constant. This feature is supported by only the two visualizations: SimpleBar chart and Pie chart. To understand this feature consider and example of SimpleBar chart as shown in Figure 4.23. The SimpleBar contains the 'change icon' in red box. When user click on 'change icon', the process of generating alternative visualization will began as pop up alert shown in black box. The alternative visualization 'Pie Chart' is generated as shown Figure 4.24 keeping data attribute constant i.e. 'ParentAnsweringSurvey'.

**Exceptions and special cases.** There are few exception and special cases which block the GA run. To overcome from these blocking points we have defined the exceptions and special cases. Which are as follows:

- The selection of one chart only when only one visualization is possible in the dashboard concerning data attribute types in a given dataset. In this case, the system will give you an alert message: "The Selected ... Visualisation Name... is/are not sufficient to generate a dashboard considering the given dataset, please select more visualizations to generate a dashboard".

- When the selected visualization is not possible in the dashboard concerning data attribute types in a given dataset. In this case, the system will give

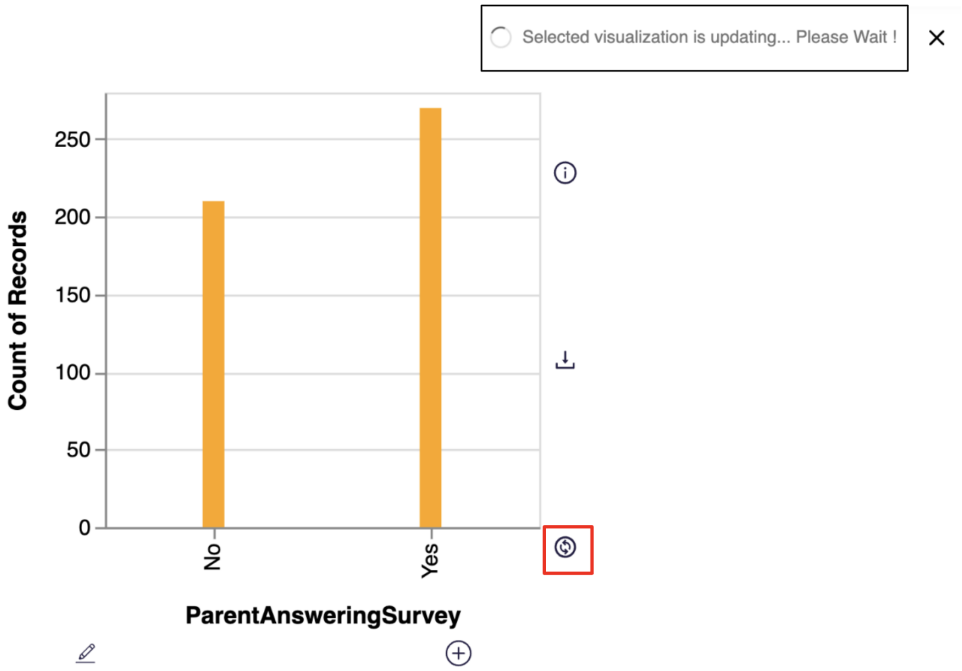



Figure 4.23: Generation of alternative visualization of keeping data attributes constant. This feature is only in the case of simple bar charts and pie charts.

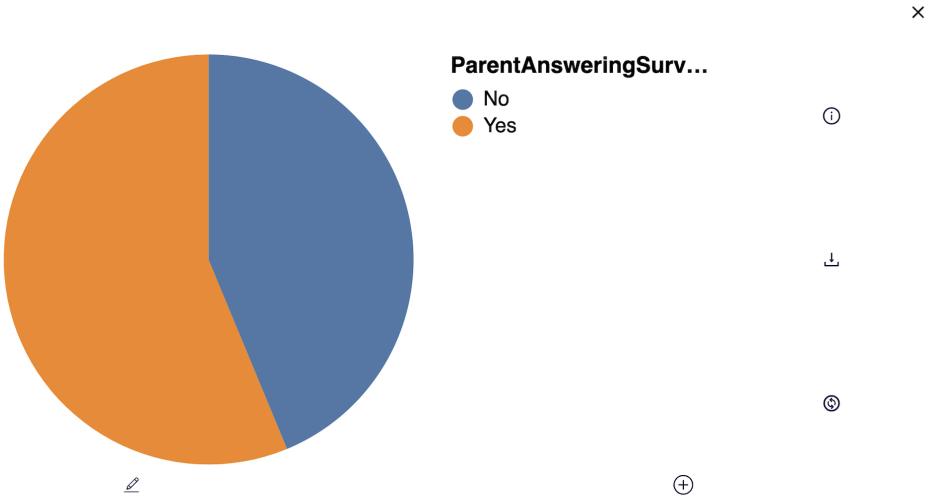



Figure 4.24: SimpleBar Chart is replaced with alternative 'Pie Chart' keeping data attributes constant

you a suggestion message (left side of 'Selected Visualization'): "The following selected visualizations will not generate considering the given dataset: ... Visualisation Names... Please select other visualizations to generate a dashboard".

Minimum requirement of data attribute types to generate the following visualizations concerning data attributes present in the given dataset:

1. Histogram: Numeric >= 1
2. Pie Chart: Categoric (Nominal or Ordinal) >= 1
3. Simple Bar Chart: Categoric (Nominal or Ordinal) >= 1
4. Line Chart: Numeric >= 1 and Temporal >= 1
5. Scatter Plot: Numeric >= 2
6. Bar Chart: Categoric (Nominal or Ordinal) >= 1 and Numeric >= 1

## 4.5 Conclusion

In summary, the system user interface design for the Automatic Dashboard Recommendation System (ADRS) has been carefully crafted to meet the needs of novice users through a simple, intuitive, and efficient interface. The system allows users to visualize data and receive dashboard recommendations in just three straightforward steps: data upload, metadata and parameter setting, and dashboard visualization. With a strong focus on user-friendliness, the system supports minimal manual intervention while enabling customization and personalized dashboards.

The design requirements and goals emphasize speed, simplicity, automation, and interactivity. The architecture and tech stack, including the use of Next.js, Node.js, and Vega Lite, ensure that the system is fast, scalable, and capable of handling multiple users. Furthermore, AWS hosting provides a reliable and robust environment for deployment. Overall, the ADRS system is well-equipped to provide users with dynamic and personalized dashboard recommendations, while continuously improving through user feedback and interactive features.

Chapter 5

# User Evaluation: Comparative study

## Contents


**5.1 Participants and Dataset . . . . . 99**
**5.2 Evaluation protocol . . . . . 100**
**5.3 Results and Statistical Analysis . . . . . 102**
5.3.1 Task 1 with Power BI and SimpleVis . . . . . 103
5.3.2 Task 2 with Power BI and SimpleVis . . . . . 107
**5.4 Questionnaire Analysis . . . . . 111**
**5.5 User type based analysis of results . . . . . 115**
5.5.1 Novices vs Experienced users: Task 1 with Power BI and SimpleVis . . . . . 115
5.5.2 Novices vs Experienced users: Task 2 with Power BI and SimpleVis . . . . . 121
**5.6 System evaluation analysis summary . . . . . 127**


We have evaluated SimpleVis (SV) using a comparative study with PowerBI (PBI). We have conducted this study online with global users (India and France). In this chapter, we discuss participants, datasets, tools, tasks, results, and statistical analysis of participants' feedback. The goal of this comparative study is to compare both tools on the following dimensions:

- Quality of generated dashboard (expected visualizations),
- time taken to perform a particular task (fast/slow),
- ease of use and learning the tool,
- level of manual editing,
- ease of interacting with user interface, and
- user preference for the tool.

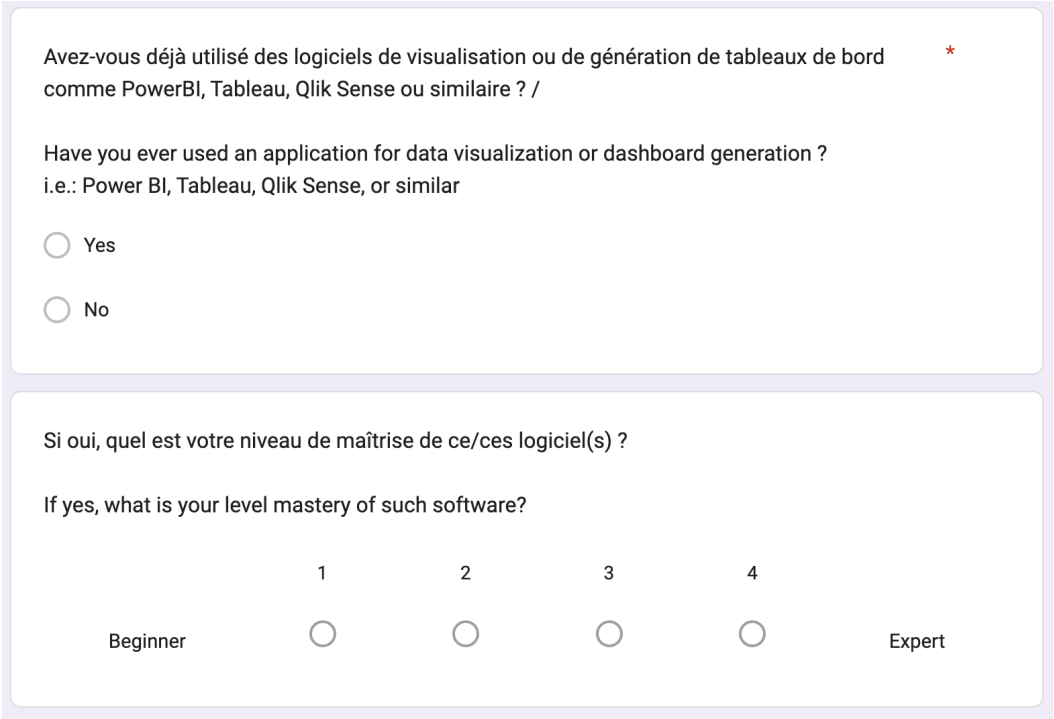



Figure 5.1: Level of Mastery: a couple of questions to participant related to their level of mastery in using the application for data visualization or dashboard generation. The participant needs to answer first if he/she is already familiar with such applications (Power BI, Tableau, Qlik Sense, or similar) or not. If the participant chooses no for the first question, we consider them novice users. In case of yes they need to choose their mastery level from 1 to 4 as follows: 1 for beginner, 2 for learners, 3 for advanced, and 4 for expert.

## 5.1 Participants and Dataset

All participants filled out the form with their basic details. The participant's information analysis is as follows:

- Our study has 25 participants (18 males and 7 females).
- The age group is between 20-29 with a mean of 21.48 and a standard deviation of 2.45.
- Level of study: Participants have different levels of study 18 bachelors, 2 masters, 1 doctorate, and 4 are corporate employees in the computer science field.
- Level of tools mastery: We have asked a couple of questions to participants related to their level of mastery in using the applications (Power BI, Tableau, Qlik Sense, or similar) for data visualization or dashboard generation as shown in figure 5.1. The participants need to answer first if they are already familiar with such an application or not. If the participant chooses no for the first question (see figure 5.1), we consider them novice users. But in case of yes they need to choose their mastery level from 1 to 4 as follows: 1 for beginner, 2 for learners, 3 for advanced, and 4 for expert. We consider the beginner who began to acquire skills less than a month ago. But for learners, we consider students who already/currently have some courses in data visualization.
- Based on responses, the level of mastery for all 25 participants for such tools is as follows: we found 11 are novices (new to such tools) and 14 participants are already familiar with data visualization or dashboard generation tools. These 14 participants are categorized as: 5 are learners, 6 are beginners, 3 are advanced and no one is an expert.

To perform this study we have chosen 5 datasets. One for training and four for testing. The Iris dataset [MathNerd 2018] is used for training. Two couples of datasets are used for testing: One is 'Car' [Riley Chang 2016] and 'Weather' [ANANTH R 2021], and the other is 'Wine' [Sebastian Raschka 2014] and 'Crop' [Atharva Ingle 2020]. Each couple of datasets has similar constraints with the same number of attributes and the same type of attributes. To apply the constraints on the testing dataset we have deleted a few columns from actual datasets. All the 5 datasets are given below.

The training dataset is:

- Iris: This contains 150 rows and 5 columns with 2 data attribute types (1 Nominal and 4 Numeric).

The first couple of datasets are car and weather used for the exploratory tasks (T1):

- Car: This contains 407 rows and 5 columns with 2 data attribute types (1 Nominal and 4 Numeric).
- Weather: This contains 1462 rows and 5 columns with 2 data attribute types (1 Nominal and 4 Numeric).

A second couple datasets is wine and crop used for the directed task (T2):

- Wine: This contains 178 rows and 8 columns with 2 data attribute types (1 Nominal and 7 Numeric).
- Crop: This contains 2201 rows and 8 columns with 2 data attribute types (1 Nominal and 7 Numeric).

From the above datasets Iris dataset is used for training. Car, Weather, Wine, and Crop are used for testing. We have chosen these datasets because each of the couples has a similar type and number of data attributes.

## 5.2 Evaluation protocol

**Tools** To perform the comparative study of SimpleVis, we have chosen the PowerBI tool. The choice of PowerBI as the tool for the comparative study in evaluating SimpleVis is motivated by several key factors that make it an ideal benchmark for assessing the effectiveness and innovation of the SimpleVis system. These factors are: Widespread adoption and industry standard, Comprehensive feature set, Automation and manual customization, Strong focus on user experience, Analytical capabilities, Commercial and academic relevance, and market leader since many years.

By selecting PowerBI for the comparative study, the evaluation of SimpleVis is placed in direct contrast with a powerful, well-known, and feature-rich commercial tool that is trusted by a large global user base. This choice not only provides a high bar for performance but also ensures that the findings are practical, relevant, and significant for both industry and academic audiences. The comparative study will thus offer deeper insights into the unique strengths and innovations SimpleVis brings to the field of automatic dashboard recommendation systems, especially in terms of automation, customization, simplicity, manual editing and user experience.

**Tasks** We have given two types of tasks to participants: Exploratory tasks and Directed tasks.

1. Exploratory task (T1): Participants need to generate a dashboard with all data attributes of the dataset. All these data attributes should be present the generated dashboard, whatever the visualizations.
2. Directed task (T2): Generate the dashboard with at least 2 scatter plots. If there are no scatterplots in the recommended dashboard, the participant can generate them by manual editing.

The participant needs to perform both tasks with each tools. T1 with PowerBI and SimpleVis, T2 with PowerBI and SimpleVis. For each Task, participants need to take a screenshot of the generated dashboard and record the time taken to complete that task in seconds/minutes. The order of the systems and datasets was randomized.

**Training** First, we have trained the participants with the Iris dataset. During training first, we perform exploratory tasks with each tool 'PowerBI' and 'SimpleVis' and then participants practice both tasks (task-1 and task-2). Later we train participants for directed tasks with each tool 'PowerBI' and 'SimpleVis' and then participants practice both tasks.

The time for training and practice of exploratory tasks with both tools is 3 minutes each. In total, it takes 12 minutes. Similarly, the time for training and practice of directed tasks is also 3 minutes each. In total, it takes 12 minutes. The other miscellaneous task during training takes 4 minutes, which includes dataset and dashboard creation explanation. The total training session is expected to be 28 minutes.

**Testing and feedback** To record the participant details and feedback, we have created a five-step Google form:

- Step-1: In the first step we ask users to fill in personal details like name, age, gender, level of study, and the domain of Study. Whether they are familiar with tools for data visualization or dashboards, and their level of mastery of such data visualization tools. This basic information session lasts for 5 minutes.

- Step-2: In the second step we have given the training to the participant with both the tools (SimpleVis and PowerBI) and both the tasks (exploratory and directive). In this step, participants confirm whether they perform each of the four tasks taught during the training session. This training session lasts for 28 minutes.

- Step-3: In the third step participant performs the task for exploratory analysis. There are two sub-tasks in the third step: T1-PBI and T1-SV. Participants performed the tasks with each of the tools 'Power BI' and 'SimpleVis' using 2 datasets 'car' and 'weather'. They have to choose any one dataset randomly for 'PowerBI' and then use another remaining dataset for 'SimpleVis'. Participants can start with any one of the tools randomly. This session lasts for 12 minutes with 5 minutes maximum for each task and 2 minutes to fill out the form.

- Step-4: In the fourth step user performs the task for directed analysis. There are two sub-tasks in the fourth step: T2-PBI and T2-SV. Participants performed the tasks with each of the tools 'PowerBI' and 'SimpleVis' using 2 datasets 'wine' and 'crop'. They have to choose any one dataset randomly for

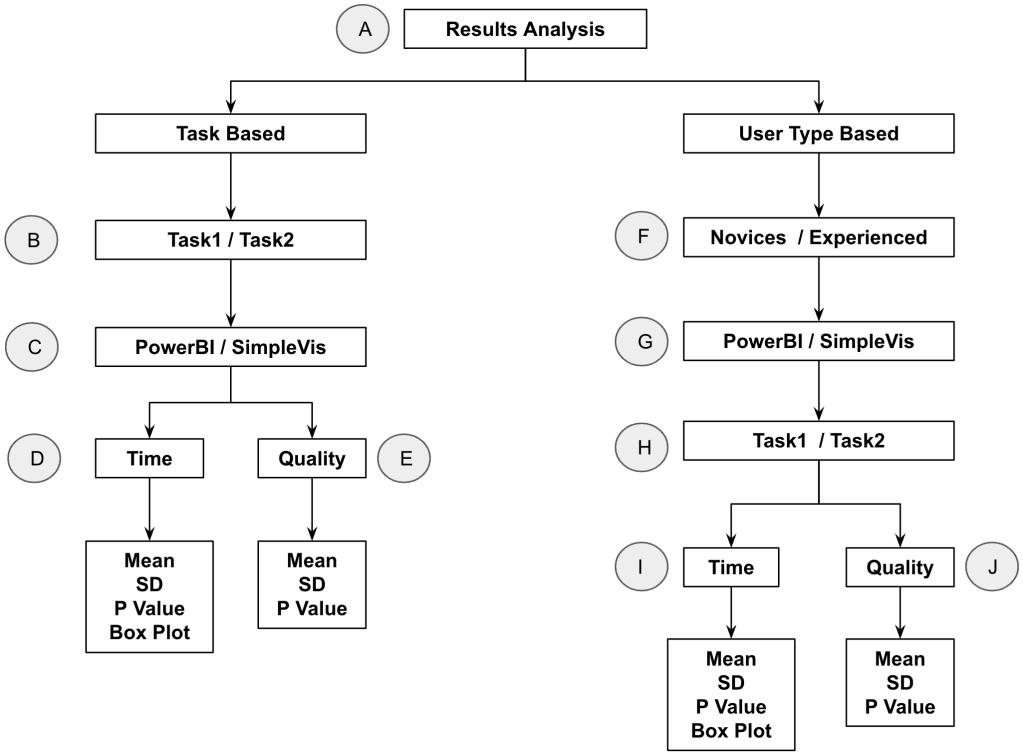



Figure 5.2: Results analysis schema based on tasks and user types: on the left workflow of task based results analysis and on the right workflow of user type based results analysis.

'Power BI' and then use another remaining dataset for 'SimpleVis'. Participants can start with any one of the tools randomly. This session is expected to be lasts for 12 minutes with 5 minutes maximum for each task and 2 minutes to fill out the form.

- Step-5: In the fifth step, participants will open the form again, and give comparative feedback on their experience with both tools during all 4 tasks. There are a total of 11 questions in the fifth step. This session lasts for 10 minutes.

The total time by this study is expected to be 72 minutes.

## 5.3 Results and Statistical Analysis

After receiving the feedback from all the participants, we have analyzed them on two dimensions: time and quality. The participants response time analysis was performed with statistical tests: mean, standard deviation, and p-value for t-test for both tasks T1 and T2 as mentioned in Table 5.1 and Table 5.3. Similarly participants

task quality was performed with statistical tests: mean, standard deviation, and p-value for t-test for both tasks T1 and T2 as mentioned in Table 5.2 and Table 5.4.

In Figure 5.2 on the left side the task based workflow for results analysis is presented. (B) We have considered Task 1 and Task 2 one by one, (C) we have chosen tool Power BI and SimpleVis for each task (D) and, (E) the statistical analysis is performed on 2 dimensions time and quality.

### 5.3.1 Task 1 with Power BI and SimpleVis

The analysis of task 1 with Power BI and SimpleVis is performed based on two dimensions: Time and Quality.

#### 5.3.1.1 Task 1 response time analysis

We found the following outcome as mentioned in Table 5.1 for task 1 and response time.

The datasets will not affect the time taken by participants because both the datasets used for T1-PBI and T1-SV have the same number and type of dimensions and measures.

Table 5.1: Statistical Analysis of time taken in seconds by users for each sub-task of T1 (with a p_value of 0.00273 during statistical t-test).

| $TimeTaken/TaskTool$ | Mean | SD | Min | Max |
|---|---|---|---|---|
| T1-PBI | 101.48 | 85.40 | 15 | 364 |
| T1-SV | 42.72 | 27.14 | 15 | 109 |
| p-value | 0.00273 | | | |

The analysis for task 1 in Table 5.1 and Figure 5.3 is as follows:

- The mean of time taken by SimpleVis for T1-SV is less than that of T1-PBI for Task 1. It shows that SimpleVis is fast for simple dataset having small number of columns (i.e. 5). It is easier to get the desired visualization and data attributes in the dashboard in SimpleVis than in PowerBI,
- The standard deviation of time taken by SimpleVis for T1-SV is less than that of T1-PBI for Task 1. It indicates data are clustered more tightly around the mean for the task T1 performed by SimpleVis. While a large standard deviation in the case of T1-PBI indicates data are more spread out for PowerBI,

- We have performed the qualitative test with two groups of quantitative values. These two groups are: one is with participants who performed task 1 with power BI (T1-PBI) and the other one is those who performed task 1 with SimpleVis (T1-SV). Each group contains 25 observations generated by 25 participants. For task 1 the p-value for t-test is 0.00273 (0.0027257102961144 to be exact). As the p-value is less than 0.05, it means that there are significant differences in the observations of each group. The p-value t-test was performed using BiostaTGV [iPLesp, INSERM et Sorbonne Université ].

- Finally, figure 5.3 shows the analysis of task 1 response time visually. We have chosen the box plot, because it represents a distribution of quantitative data through a set of summary statistics. In this box plot, the box represent summary statistics for task 1 response time done with Power BI while on the other side red box represent task 1 done using SimpleVis. The white horizontal bar in centre of both bar represents the median of observations, which is 66 for T1-PBI and 33 for T1-SV. The both boxes show the interquartile range (IQR). IQR for T1-PBI in blue box are Q1 and Q3 with value 44 and 112 respectively. And IQR for T1-SV in red box are Q1 and Q3 with value 22 and 57 respectively. The blue and red dots presents the outlier which shown that these observations are extremely unexpected and unusual. There are 2 outlines for T1-PBI in blue and one for T1-SV in red. Untimately the lower value for T1-SV with respect to T1-PBI for different parameters indicates that participant perform the T1 with SimpleVis is more faster than PowerBI for exploratory task.

#### 5.3.1.2 Task 1 quality analysis

We found all the participants performed the tasks except one participant who has wrongly uploaded T1-PBI DBo screen copy. To analyze the quality of task 1 performed by participants we count the number of data attributes present in dashboards they have generated using the respective dataset. The below Table 5.2 shows the quality of task 1 performed by participants. Where column 1 contains the participants' names. Column 2 and column 3 indicate the number of data attributes covered by dashboards participants have generated with respective tools Power BI and SimpleVis. As both datasets (car and weather) contain five columns so maximum number of data attributes covered by the dashboard can be 5.

The quality analysis for task 1 in Table 5.2 is as follows:

- The mean of T1-PBI (4.84) is slightly greater than T1-SV (4.80) and standard deviation for T1-PBI is 0.37 and T1-SV is 1. This is because participant 5 submitted the wrong screen copy of T1-SV dashboard.

- The standard deviation for T1-PBI is 0.37 and T1-SV is 1. It indicates observations are more stick to mean in case of T1-PBI than that of T1-SV.

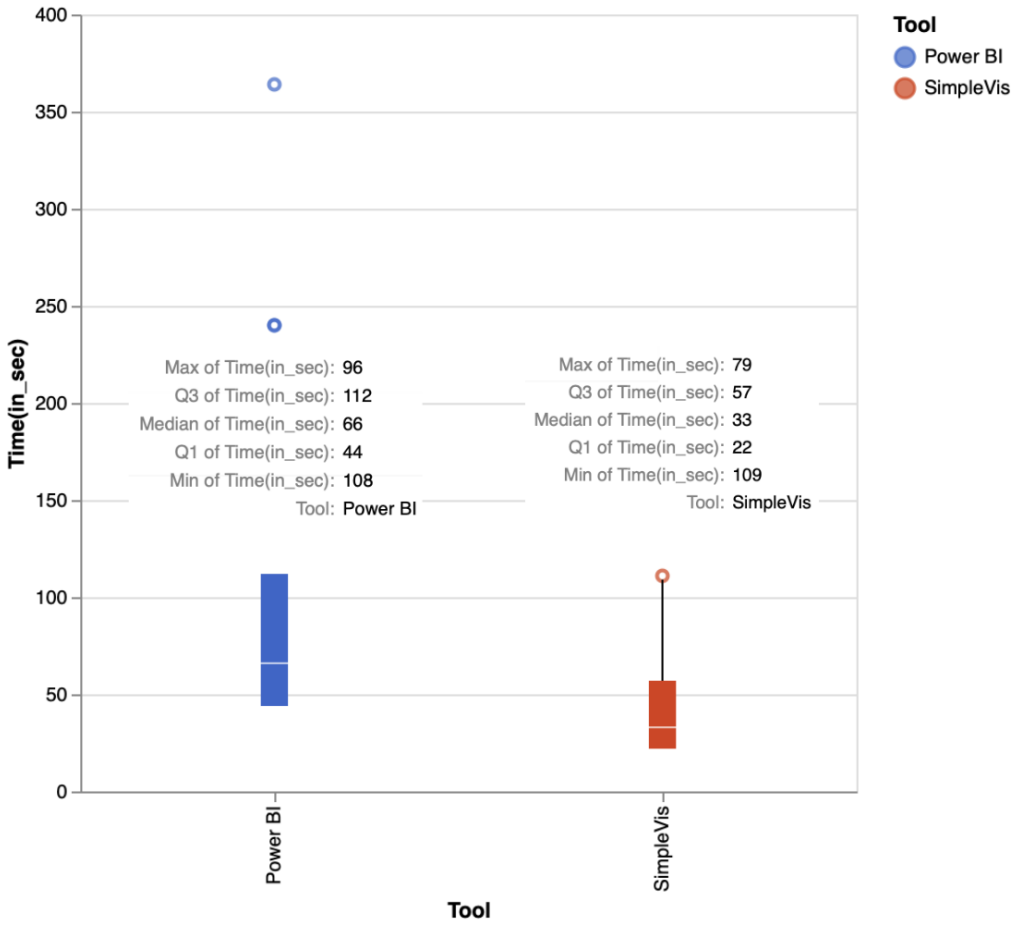



Figure 5.3: The box represent summary statistics for task 1 done with Power BI while on the other side red box represent task 1 done using SimpleVis. The lower value for T1-SV with respect to T1-PBI for different parameters indicates that participant perform the T1 with SimpleVis is more faster than PowerBI.

Table 5.2: Quality analysis of task 1 performed by participants. NV is for novices and EXP is for experienced users.

| *Participants/Task(Tool)* | T1-PBI | T1-SV | *Participants/Task(Tool)* | T1-PBI | T1-SV |
|---|---|---|---|---|---|
| Participant 1 (NV) | 5 | 5 | Participant 2 (EXP) | 5 | 5 |
| Participant 3 (NV) | 5 | 5 | Participant 4 (NV) | 5 | 5 |
| Participant 5 (EXP) | 5 | 0 | Participant 6 (EXP) | 5 | 5 |
| Participant 7 (NV) | 5 | 5 | Participant 8 (EXP) | 5 | 5 |
| Participant 9 (NV) | 5 | 5 | Participant 10 (EXP) | 5 | 5 |
| Participant 11 (EXP) | 4 | 5 | Participant 12 (EXP) | 4 | 5 |
| Participant 13 (EXP) | 5 | 5 | Participant 14 (EXP) | 5 | 5 |
| Participant 15 (NV) | 5 | 5 | Participant 16 (EXP) | 5 | 5 |
| Participant 17 (EXP) | 5 | 5 | Participant 18 (EXP) | 5 | 5 |
| Participant 19 (EXP) | 5 | 5 | Participant 20 (EXP) | 4 | 5 |
| Participant 21 (NV) | 4 | 5 | Participant 22 (NV) | 5 | 5 |
| Participant 23 (NV) | 5 | 5 | Participant 24 (NV) | 5 | 5 |
| Participant 25 (NV) | 5 | 5 | | | |

| | T1-PBI | T1-SV |
|---|---|---|
| Mean (NV) | 4.84 | 4.80 |
| SD | 0.37 | 1 |
| p-value | 0.8522015267 | |

- Task 1, p-value (for t-test) is 0.85 (0.8522015267 to be exact). As the p-value is greater than 0.05, it means that there are no significant differences in the observations of each group. So from statistical point of view there is no difference in the quality of task 1 performed by both group of participants.

The results shown at T1 performed by SimpleVis cover all 5 data attributes except one dashboard. We found that one dashboard was wrongly uploaded by a user. In the case of PowerBI, there are four dashboards that contain only 4 data attributes, and the remaining 21 dashboards cover all 5 data attributes.

#### 5.3.1.3 Conclusions on task 1

In task 1, users succeeded in solving the exploratory task, with a much shorter time for SV than PBI. In terms of quality of task 1 both systems have almost similar outcomes from statistical point of view.

### 5.3.2 Task 2 with Power BI and SimpleVis

Similarly the analysis of task 2 with Power BI and SimpleVis is also performed based on two dimensions: Time and Quality.

#### 5.3.2.1 Task 2 time analysis

We obtained the results on response time for task 2 as shown in Table 5.3. Here, T2-PBI represents the tasks completed with Power BI, while T2-SV refers to those completed with SimpleVis. Both datasets used for T2-PBI and T2-SV contain the same number and type of dimensions and measures to avoid any dataset effect.

Table 5.3: Statistical Analysis of response time taken in seconds by users for each sub-task of T2 with a significant difference between the two means (p-value of 0.00013 during the statistical t-test).

| *TimeTaken/TaskTool* | Mean | SD | Min | Max |
|---|---|---|---|---|
| T2-PBI | 153.6 | 93.09 | 29 | 360 |
| T2-SV | 64.4 | 29.51 | 30 | 160 |
| p-value | 0.00013 | | | |

The analysis for task 2 response time in Table 5.3 and Figure 5.4 is as follows:

- The mean of time taken by SimpleVis for T2-SV and less than that of PowerBI for T2-PBI. It shows that SimpleVis is not only fast for simple datasets but also for complex datasets. It is easier to get the desired visualization in the dashboard in SimpleVis than that in PowerBI even during directed tasks.

- The standard deviation of time taken by SimpleVis for T2-SV is less than that of PowerBI for T2-PBI. It indicates data are clustered tightly around the mean also for the T2-SV task performed by SimpleVis. While a large standard deviation in the case of T2-PBI indicates data are more spread out for PowerBI.

- We conducted a statistical analysis for T2 with same two groups of participants similar to T1. The resulting p-value (for t-test) for task 2 was 0.00013 (precisely, 0.00013496502556748). Since this p-value is well below the threshold of 0.05, it indicates that there are statistically significant differences between the observations in the two groups.

- Similar to task 1, we presented an analysis of task 2 with box plot, as illustrated in Figure 5.4. The white horizontal bar in the center of each box marks the median of the observations, which is 116 for T2-PBI and 58 for T2-SV. For T2-PBI, the IQR (shown in blue) has a Q1 value of 82 and a Q3 value of 221. For T2-SV, the IQR (shown in red) has a Q1 value of 43 and a Q3 value of 88. The blue and red dots indicate outliers.

  Ultimately, the consistently lower values for T2-SV compared to T2-PBI across various parameters suggest that participants completed task 2 faster using SimpleVis than Power BI, not only for exploratory tasks but also for directed tasks, regardless of the dataset used.

#### 5.3.2.2 Task 2 quality analysis

We observed that all participants completed the tasks. To evaluate the quality of their work on Task 2, we assessed the number of scatterplots included in the dashboards they created to quantify the result quality, if participant's DBo contains more than 2 scatter plots we consider only 2 as of threshold, the task objective was "at least two scatterplot" should be presented in DBo.

Table 5.4 summarizes this evaluation, with the first column listing the participants' names, and the second and third columns showing the number of scatterplots in the dashboards produced using Power BI and SimpleVis, respectively.

The quality analysis for task 2 in Table 5.4 is as follows:

- The mean of T2-PBI (1.48) is less than T1-SV (1.76). It indicates that SimpleVis gives better quality of results than PBI. But if consider DBo's having exact 2 visualizations than there are only 15 participants in case of PBI and none for SV.

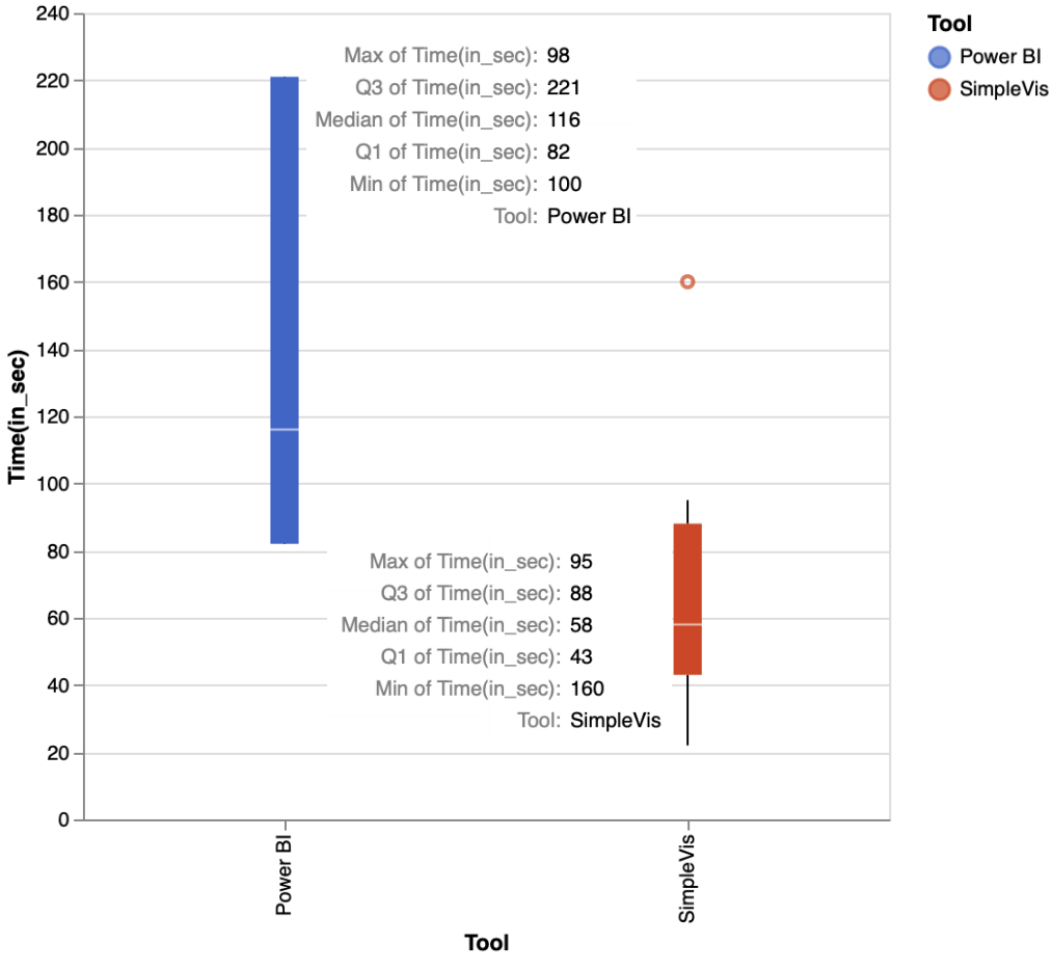



Figure 5.4: The box represents the summary statistics of the response time for task 2 completed with Power BI, while the red box represents task 2 performed using SimpleVis. The lower values for T2-SV compared to T2-PBI across various parameters indicate that participants completed task 2 faster with SimpleVis than with Power BI (and the difference is significant from a statistical point of view with a p-value of 0.00013 during the statistical test).

Table 5.4: Quality analysis of task 2 performed by participants. NV is for novices and EXP is for experienced users.

| *Participants/Task(Tool)* | T2-PBI | T2-SV | *Participants/Task(Tool)* | T2-PBI | T2-SV |
|---|---|---|---|---|---|
| Participant 1 (NV) | 1 | 2 | Participant 2 (EXP) | 2 | 2 |
| Participant 3 (NV) | 0 | 0 | Participant 4 (NV) | 2 | 2 |
| Participant 5 (EXP) | 2 | 2 | Participant 6 (EXP) | 1 | 1 |
| Participant 7 (NV) | 2 | 2 | Participant 8 (EXP) | 2 | 2 |
| Participant 9 (NV) | 2 | 1 | Participant 10 (EXP) | 2 | 2 |
| Participant 11 (EXP) | 2 | 2 | Participant 12 (EXP) | 3 | 2 |
| Participant 13 (EXP) | 2 | 2 | Participant 14 (EXP) | 2 | 2 |
| Participant 15 (NV) | 2 | 2 | Participant 16 (EXP) | 2 | 2 |
| Participant 17 (EXP) | 2 | 2 | Participant 18 (EXP) | 2 | 2 |
| Participant 19 (EXP) | 2 | 2 | Participant 20 (EXP) | 1 | 2 |
| Participant 21 (NV) | 0 | 0 | Participant 22 (NV) | 1 | 2 |
| Participant 23 (NV) | 1 | 2 | Participant 24 (NV) | 0 | 2 |
| Participant 25 (NV) | 0 | 2 | | | |

| | T2-PBI | T2-SV |
|---|---|---|
| Mean | 1.48 | 1.76 |
| SD | 0.77 | 0.60 |
| p-value | 0.15738314918967 | |

- The standard deviation for T2-PBI is 0.77 and T2-SV is 0.60. It indicates observations are more stick to mean in case of T1-SV than that of T1-PBI.

- Task 2, p-value (for t-test) is 0.15 (0.15738314918967 to be exact). As the p-value is greater than 0.05, it means that there are no significant differences in the observations of each group. So from statistical point of view there is no significant difference in the quality of task 2 performed by both group of participants. Although It indicates SimpleVis gives same quality of results that of PowerBI with less user intervention and minimum manual editing.

To be more precise the screen copies of the results (as shown in Table 5.4) for Task 2 are as follows: SimpleVis reveal that scatterplots were included in nearly all participant-generated dashboards, with only two dashboards lacking them. Specifically, one dashboard contains 9 scatterplots, five dashboards have 8 scatterplots, one has 7 scatterplots, fourteen include 4 scatterplots, one dashboard features a single scatterplot, and two dashboards contain no scatterplots. In contrast, the Power BI results show that four dashboards lack scatterplots entirely. Five dashboards include 1 scatterplot, fifteen contain 2 scatterplots, one dashboard has 3 scatterplots, and four dashboards have no scatterplots.

#### 5.3.2.3 Conclusions on task 2

In task 2 users succeeded in solving the directed task, with a much shorter time for SV than PBI. In terms of quality of task 2 both system have similar results without any significant differences from statistical point of view.

## 5.4 Questionnaire Analysis

In step 5, we asked 11 questions to participants. The first 4 are related to individual tools as shown in Figure 5.5. Participants can answer the questions based on rating. The linear scale is given to the participant from 1 to 5. Each choice of linear scale is indicated as follows: 5 for excellent, 4 for very good, 3 for good, 2 for bad, and 1 for very bad.

We asked the participants to rate the next 7 comparative questions as shown in Figure 5.6. The linear scale is given to the participant from 1 to 5. If a participant chooses less than 3 it means PowerBI is better. If a participant chooses more than 3 it means SimpleVis is better. If a participant chooses 3 it means PowerBI and SimpleVis have the same level of advantage.

The individual analysis of both tools for Q1 to Q4 is as follows:

- Tool usefulness: The mean (4.4) for Q1 is slightly higher than Q2 mean (3.92) which shows that SimpleVis is more useful than PowerBI. And SimpleVis has less SD (0.76) than PowerBI SB (0.91) which shows SimpleVis is more consistent.

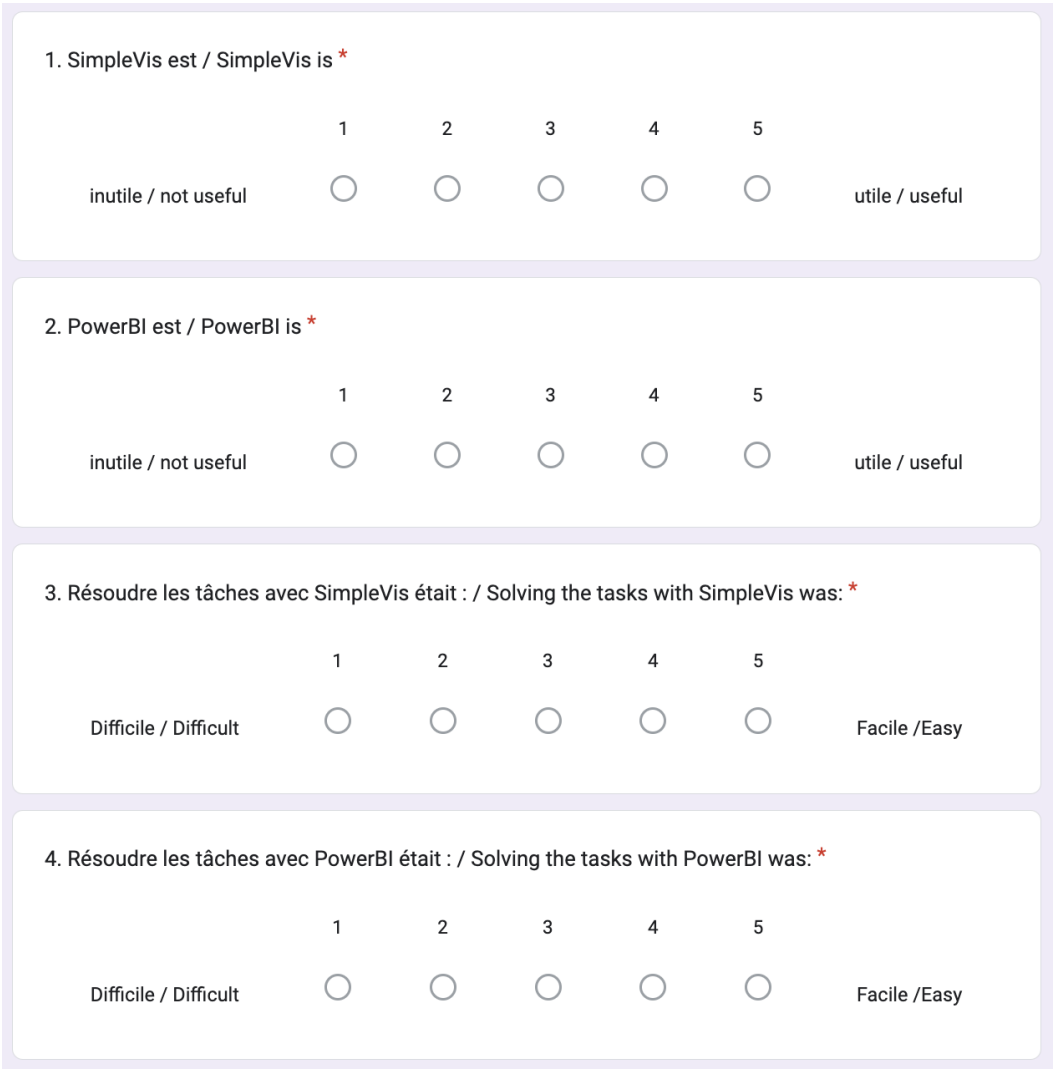



Figure 5.5: The first 4 (Q1 to Q4) are related to individual tools.

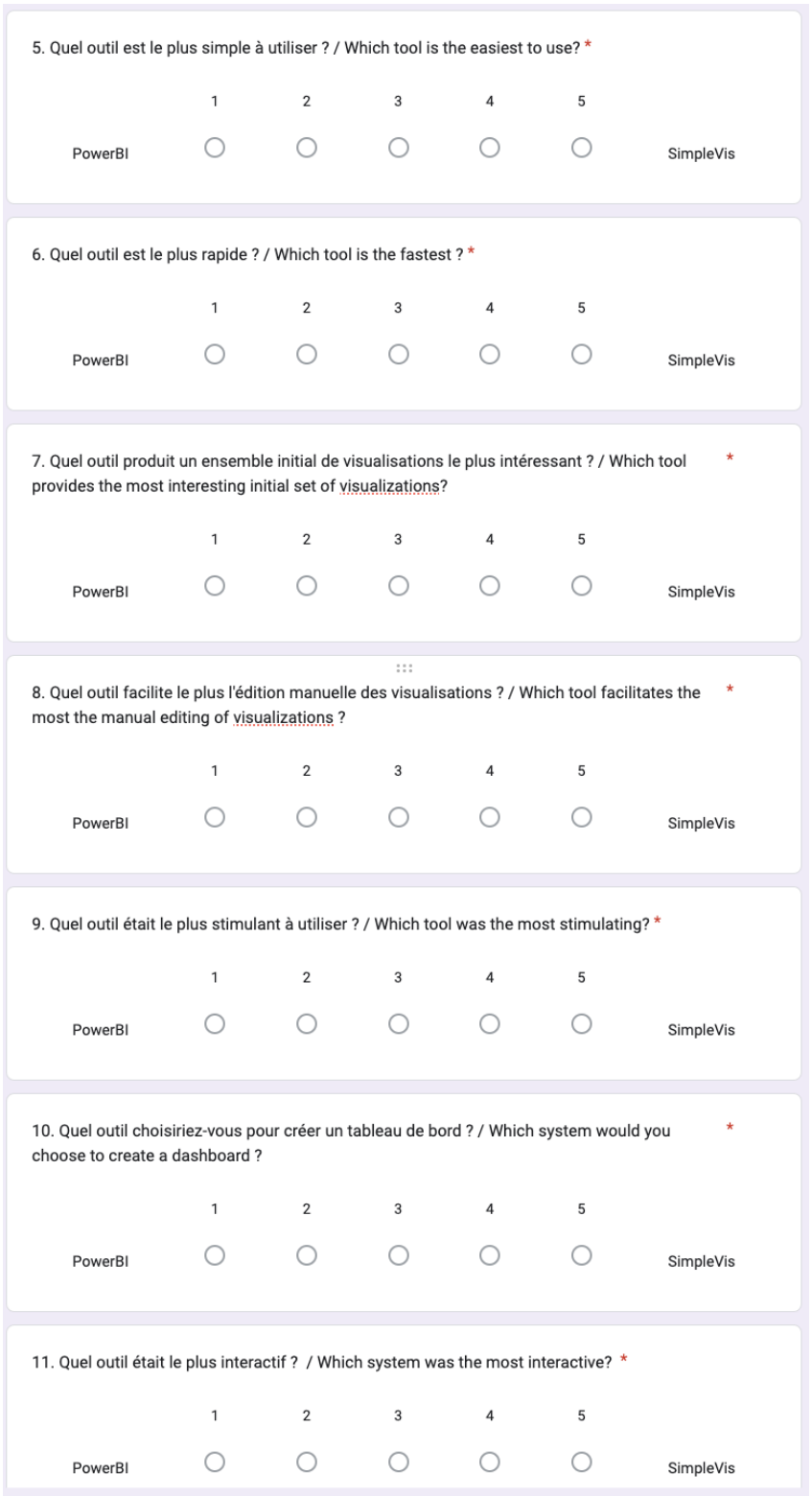



Figure 5.6: The next 7 comparative questions (Q5 to Q11) for PowerBI and SimpleVis.

Table 5.5: Post-task questionnaire mean and standard deviation for questions Q1 to Q11.

| *Responses/Question* | Mean | SD |
|---|---|---|
| Q1. SimpleVis is: not useful - useful | 4.4 | 0.76 |
| Q2. PowerBI is: not useful - useful | 3.92 | 0.91 |
| Q3. Solving the tasks with SimpleVis was: Difficult - Easy | 4.68 | 0.85 |
| Q4. Solving the tasks with PowerBI was: Difficult - Easy | 3.52 | 0.92 |
| Q5. Which tool is the easiest to use? PBI - SV | 4.68 | 0.56 |
| Q6. Which tool is the fastest? PBI - SV | 4.4 | 0.91 |
| Q7. Which tool provides the most interesting initial set of visualizations? PBI - SV | 3.56 | 1.26 |
| Q8. Which tool facilitates the most manual editing of visualizations? PBI - SV | 2.76 | 1.42 |
| Q9. Which tool was the most stimulating? PBI - SV | 3.76 | 1.13 |
| Q10. Which system would you choose to create a dashboard? PBI - SV | 3.76 | 1.20 |
| Q11. Which system was the most interactive? PBI - SV | 3.56 | 1.39 |

- Ease of task solving: The Q3 mean (4.68) is more than Q4 mean (3.52) which shows that SimpleVis is easier than PowerBI for solving the tasks. And SimpleVis have less SD (0.85) than PowerBI SD (0.92) which shows SimpleVis is more consistent.

The comparative analysis of both tools for Q5 to Q11 is as follows:

- Q 5 Ease of tool: SimpleVis is easier than PowerBI.

- Q 6 Fastest tool: SimpleVis is faster than PowerBI.
- Q 7 Interesting initial set of visualizations: SimpleVis gives slightly interesting initial set of visualizations than PowerBI.
- Q 8 Facilitates the manual editing of visualizations: PowerBI facilitates the more manual editing of visualizations than SimpleVis. To perform the manual editing user must have domain knowledge and experience of using BI tools, in our study 14 participant are already familiars with some BI tool.
- Q 9 Most stimulating: SimpleVis is more stimulating than PowerBI.
- Q 10 Preferred system to create a dashboard: SimpleVis is a Preferred system to create a dashboard over PowerBI.
- Q 11 Most interactive system: SimpleVis is a more interactive system than PowerBI.

## 5.5 User type based analysis of results

In this section we perform the comparative analysis of tasks perform by participants but we have divided the participants into 2 categories: Novice (NV) users and Experienced (EXP) users according to their answers to the questionnaire. As explained in section 5.1, all those participants who have not used before or were familiar with BI tools are considered as Novices. While those who have used before or were already familiar with BI tools are considered as experienced. We consider beginners, advanced and experts as experienced users. Out of 25 participants there are 11 novices and 14 experienced users.

In Figure 5.2 on the right side the user type based workflow for results analysis is presented. (B) We have grouped the responses of both user types novices and experienced separately, (C) each user (NV and EXP) have used the tool Power BI and SimpleVis one by one, (H) to performed the Task 1 and Task 2 with each of the tool, (I) and (J) the statistical analysis is performed on 2 dimensions time and quality.

### 5.5.1 Novices vs Experienced users: Task 1 with Power BI and SimpleVis

We conducted a comparative analysis between novice and experienced users for Task 1. The analysis was carried out separately for PowerBI and SimpleVis, based on feedback received from the same group of participants in both categories. The analysis of task 1 with Power BI and SimpleVis is performed based on two dimensions: Time and Quality.

#### 5.5.1.1 Task 1 with PowerBI time analysis

Based on analysis of results we found the following outcome as mentioned in Table 5.6. In this table T1-PBI-NV stands for Task 1 done by novices with PBI and T1-PBI-EXP stands for Task 1 done by experienced with PBI.

Table 5.6: Statistical Analysis of time taken in seconds by novices user and experienced users to complete the Task 1 with Power BI.

| $TimeTaken/TaskToolUser$ | Mean | SD | Min | Max |
|---|---|---|---|---|
| T1-PBI-NV | 110.27 | 110.87 | 15 | 364 |
| T1-PBI-EXP | 94.57 | 62.37 | 30 | 240 |
| p-value | 0.6802514168 | | | |

From Table 5.6 we come up with the following comparative analysis of task 1 with PBI based on user types (novice and experienced):

- The mean time taken by T1-PBI-NV (110.27) is greater than T1-PBI-EXP (94.57). This suggests that novices users spend more time on the task 1 with PBI, likely because they are unfamiliar with PowerBI or similar BI tools.
- The standard deviation of time for Task 1 performed T1-PBI-NV (110.87) is greater than T1-PBI-EXP (62.37).
- To study the significance of differences between the two groups, we conducted a statistical t-test using BiostaTGV [iPLesp, INSERM et Sorbonne Université ]. The analysis involved comparing two groups of quantitative data: one group consisted of 11 novices participants who completed task 1 using Power BI (T1-PBI-NV), while the other group comprised 14 experienced participants who completed the same task using a SimpleVis tool (T1-PBI-EXP). The resulting p-value for task 1 was 0.68 (precisely, 0.6802514168). Since this p-value is greater than the threshold of 0.05, it indicates that there is no statistically significant differences between the observations in the two groups from statistical point of view.
- As illustrated in Figure 5.7. In this box plot, blue and red boxes represent summary statistics for task 1 done with PowerBI by experienced and novices users respectively. The red box have more IQR than blue. It means observations in T1-PBI-NV are more spread than T1-PBI-EXP. The median value in blue T1-PBI-EXP (70) is higher than that of red T1-PBI-NV (62). It indicates

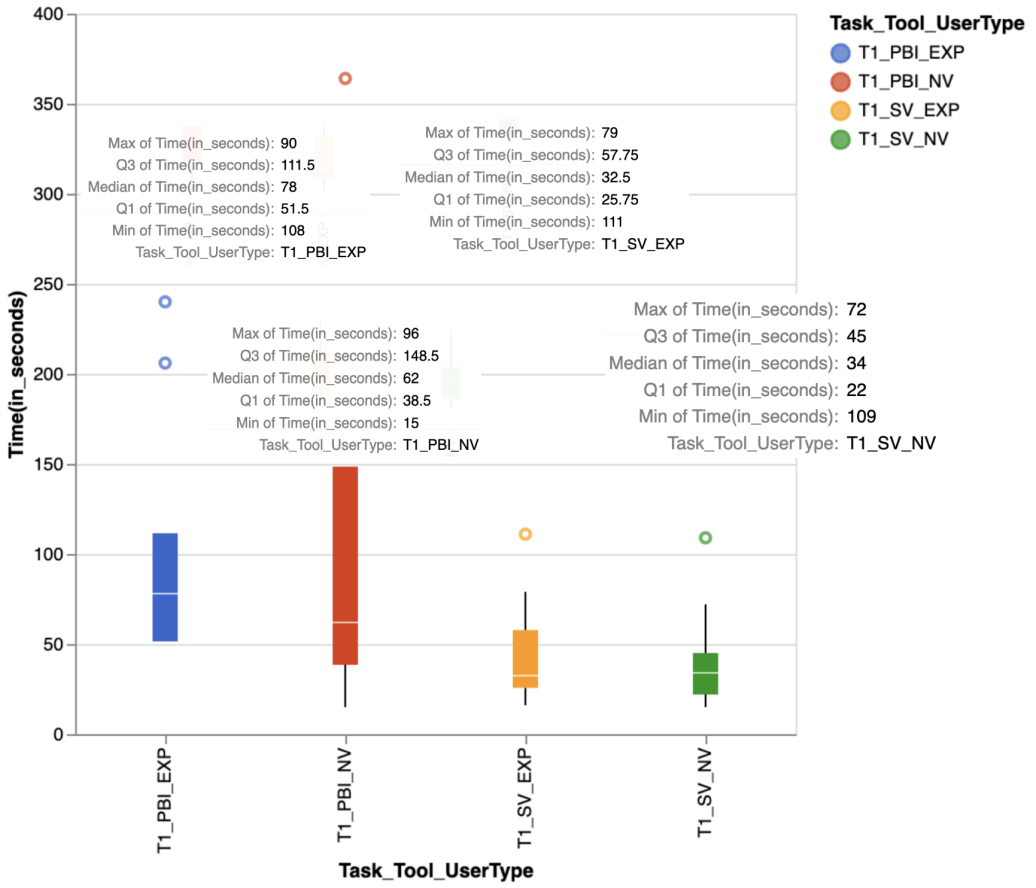



Figure 5.7: The box represent summary statistics for task 1 done with PowerBI by experienced and novices users in blue and red respectively. While on the other side yellow and green box represent task 1 done using SimpleVis by experienced and novice users respectively.

experience user spend more time than novices as experience users are already familiar with BI tools.

#### 5.5.1.2 Task 1 with PowerBI quality analysis

From Table 5.7 we come up with the following comparative analysis of task 1 with PBI is as follows:

- The mean of T1-PBI-NV (4.91) is slightly greater than T1-PBI-EXP (4.79). It indicates that almost both the types of users obtained same quality of results.

- The standard deviation for T1-PBI-NV is 0.30 and T1-PBI-EXP is 0.43. It indicates observations are almost consistent in both groups.

Table 5.7: Quality analysis of task 1 performed by novices and experienced participants with PBI.

| *Participants/Task(Tool)* | T1-PBI-NV | T1-PBI-EXP | *Participants/Task(Tool)* | T1-PBI-NV | T1-PBI-EXP |
|---|---|---|---|---|---|
| Participant 1 (NV) | 5 | | Participant 2 (EXP) | | 5 |
| Participant 3 (NV) | 5 | | Participant 4 (NV) | 5 | |
| Participant 5 (EXP) | | 5 | Participant 6 (EXP) | | 5 |
| Participant 7 (NV) | 5 | | Participant 8 (EXP) | | 5 |
| Participant 9 (NV) | 5 | | Participant 10 (EXP) | | 5 |
| Participant 11 (EXP) | | 4 | Participant 12 (EXP) | | 4 |
| Participant 13 (EXP) | | 5 | Participant 14 (EXP) | | 5 |
| Participant 15 (NV) | 5 | | Participant 16 (EXP) | | 5 |
| Participant 17 (EXP) | | 5 | Participant 18 (EXP) | | 5 |
| Participant 19 (EXP) | | 5 | Participant 20 (EXP) | | 4 |
| Participant 21 (NV) | 4 | | Participant 22 (NV) | 5 | |
| Participant 23 (NV) | 5 | | Participant 24 (NV) | 5 | |
| Participant 25 (NV) | 5 | | | | |

| | T1-PBI-NV | T1-PBI-EXP |
|---|---|---|
| Mean (NV) | 4.91 | 4.79 |
| SD | 0.30 | 0.43 |
| p-value | 0.42479303309914 | |

- p-value for Task 1 with PBI is 0.42 (0.42479303309914 to be exact). As the p-value is greater than 0.05, it means that there is no significant differences in the observations of participants of both groups from statistical point of view.

#### 5.5.1.3 Conclusions on task 1 with PowerBI

In task 1 with PBI, novices took slightly more time to complete the task than experienced. Novices have had slightly better quality of tasks. But in both cases time and quality there is no significant differences in the outcomes for both the groups from statistical point of view.

#### 5.5.1.4 Task 1 with SimpleVis time analysis

In Table 5.8 T1-SV-NV represents Task 1 performed by novice users with SimpleVis and T1-SV-EXP represents Task 1 performed by experienced users with SimpleVis. Based on analysis of results we found the following outcome as mentioned in Table 5.8 is as follows:

Table 5.8: Statistical Analysis of time taken in seconds by novices user and experienced users to complete the Task 1 with SimpleVis.

| $TimeTaken/TaskToolUser$ | Mean | SD | Min | Max |
|---|---|---|---|---|
| T1-SV-NV | 40.45 | 28.04 | 15 | 109 |
| T1-SV-EXP | 44.50 | 27.34 | 16 | 111 |
| p-value | 0.7208938488 | | | |

The user type based comparative analysis of time for task 1 with novices in Table 5.8 is as follows:

- The mean time taken by T1-SV-NV (40.45) is greater than T1-SV-EXP (44.50). This suggests that novices users spend less time on the task 1 with SV, likely because novices can easily learn and use simple BI tools.
- The standard deviation of time for Task 1 performed T1-SV-NV (28.04) is greater than T1-SV-EXP (27.34).
- p-value for Task 1 with SV is 0.72 (0.7208938488 to be exact). As the p-value is greater than 0.05, it means that there is no significant differences in the from statistical point of view.
- As illustrated in Figure 5.7, yellow and green boxes represent task 1 done using SimpleVis by experienced and novice users respectively. Yellow box have slightly more spread IQR than green. The observations in T1-SV-EXP are slightly more spread than T1-SV-NV. There are one outlier for yellow (T1-SV-EXP), and one green(T1-SV-NV).

#### 5.5.1.5 Task 1 with SimpleVis quality analysis

Table 5.9: Quality analysis of task 1 performed by novices and experienced participants with SimpleVis.

| *Participants/Task(Tool)* | T1-SV-NV | T1-SV-EXP | *Participants/Task(Tool)* | T1-SV-NV | T1-SV-EXP |
|---|---|---|---|---|---|
| Participant 1 (NV) | 5 | | Participant 2 (EXP) | | 5 |
| Participant 3 (NV) | 5 | | Participant 4 (NV) | 5 | |
| Participant 5 (EXP) | | 0 | Participant 6 (EXP) | | 5 |
| Participant 7 (NV) | 5 | | Participant 8 (EXP) | | 5 |
| Participant 9 (NV) | 5 | | Participant 10 (EXP) | | 5 |
| Participant 11 (EXP) | | 5 | Participant 12 (EXP) | | 5 |
| Participant 13 (EXP) | | 5 | Participant 14 (EXP) | | 5 |
| Participant 15 (NV) | 5 | | Participant 16 (EXP) | | 5 |
| Participant 17 (EXP) | | 5 | Participant 18 (EXP) | | 5 |
| Participant 19 (EXP) | | 5 | Participant 20 (EXP) | | 5 |
| Participant 21 (NV) | 5 | | Participant 22 (NV) | 5 | |
| Participant 23 (NV) | 5 | | Participant 24 (NV) | 5 | |
| Participant 25 (NV) | 5 | | | | |

| | T1-SV-NV | T1-SV-EXP |
|---|---|---|
| Mean | 5 | 4.46 |
| SD | 0 | 1.34 |
| p-value | 0.3867406194 | |

From the Table 5.9 we come up with the following comparative quality analysis of task 1 with SV is as follows:

- The mean of T1-SV-NV (5) is slightly greater than T1-SV-EXP (4.46). It indicates that almost both the types of users obtained same quality of results.
- Standard deviation for T2-SV-NV is 0 and T1-SV-EXP is 1.34.It shows the consistancy of observations.
- p-value for Task 1 with SV is 0.38 (0.3867406194 to be exact). As the p-value is greater than 0.05, it means that there is no significant differences in the observations of participants of both groups from statistical point of view.

#### 5.5.1.6 Conclusions on task 1 with SimpleVis

In task 1 with SV, novices took slightly less time to complete the task than experienced. And at the same time Novices have had slightly better quality of tasks. But in both cases time and quality there is no significant differences in the outcomes for both the groups from statistical point of view. It indicates that novices can easily learn and use simple tools like SimpleVis as similar to experienced users.

### 5.5.2 Novices vs Experienced users: Task 2 with Power BI and SimpleVis

In this section we have presented the results of the comparative analysis for Novices vs Experienced users for task 2. Analysis is performed for PowerBI and SimpleVis with same group of participants Novices and Experienced users.

#### 5.5.2.1 Task 2 with PowerBI time analysis

In Table 5.10, T2-PBI-NV refers to Task 2 performed by novice users using the PowerBI tool, while T2-PBI-EXP represents Task 2 performed by experienced users with PowerBI. The analysis of the results, as outlined in Table 5.10, reveals the following outcomes:

Table 5.10: Statistical Analysis of time taken in seconds by novices user and experienced users to complete the Task 2 with PowerBI.

| $TimeTaken/TaskToolUser$ | Mean | SD | Min | Max |
|---|---|---|---|---|
| T2-PBI-NV | 104 | 89.89 | 29 | 360 |
| T2-PBI-EXP | 192.57 | 77.87 | 80 | 318 |
| p-value | 0.01467808798 | | | |

The user type based comparative analysis of time for task 2 with novices in Table 5.10 is as follows:

- The mean time taken by T2-PBI-EXP (192.57) is greater than T2-PBI-NV (104). It indicates that experienced user spends more time than novices. Likely because experienced users spend time (based on screen copies submitted by participants) to perform manual editing and generating scatter plots to get the expected results. In contrast, we can say novice users avoided manual editing and relied on the automatically recommended dashboards that why they spend less time.

- The standard deviation of time for Task 2 performed T2-PBI-EXP (89.89) is slightly greater than T2-PBI-NV (77.87). It indicates observation in T2-PBI-EXP are slightly less consistent than T2-PBI-NV.

- p-value for Task 2 with PBI is 0.01 (0.01467808798 to be exact). As the p-value is less than 0.05, it means that there is significant differences between observations of two groups in the statistical point of view.

- As shown in Figure 5.8, The blue and red boxes represent SimpleVis results for experienced and novice users, respectively. The blue box has a wider IQR compared to the red, indicating that observations for T2-PBI-EXP are more spread out than for T2-PBI-NV. The median in the blue box for T2-PBI-EXP (220) is higher than in the red box for T2-PBI-NV (94). There is no observations that deviate significantly from the rest of the data because there is only one outlier for T2-PBI-NV (red).

#### 5.5.2.2 Task 2 with PowerBI quality analysis

The quality analysis for task 2 with SimpleVis in Table 5.11 is as follows:

From the Table 5.11 we come up with the following comparative quality analysis of task 2 with PBI is as follows:

- The mean of T2-PBI-NV (1) is less than T2-PBI-EXP (1.86). It indicates that experienced users have better quality of DBo results. This could be because they are already familiar with BI tools.

- Standard deviation for T2-PBI-NV is 0.89 and T2-PBI-EXP is 0.36. It shows the consistency of observations is more in T2-PBI-EXP than T2-PBI-NV.

- p-value for Task 2 with PBI is 0.003 (0.0033368251206698 to be exact). As the p-value is less than 0.05, it means that there is significant differences in the quality of task 2 with PBI performed by both groups from statistical point of view.

Table 5.11: Quality analysis of task 2 performed by novices and experienced participants with PBI.

| *Participants/Task(Tool)* | T2-PBI-NV | T2-PBI-EXP | *Participants/Task(Tool)* | T2-PBI-NV | T2-PBI-EXP |
|---|---|---|---|---|---|
| Participant 1 (NV) | 1 | | Participant 2 (EXP) | | 2 |
| Participant 3 (NV) | 0 | | Participant 4 (NV) | 2 | |
| Participant 5 (EXP) | | 2 | Participant 6 (EXP) | | 1 |
| Participant 7 (NV) | 2 | | Participant 8 (EXP) | | 2 |
| Participant 9 (NV) | 2 | | Participant 10 (EXP) | | 2 |
| Participant 11 (EXP) | | 2 | Participant 12 (EXP) | | 2 |
| Participant 13 (EXP) | | 2 | Participant 14 (EXP) | | 2 |
| Participant 15 (NV) | 2 | | Participant 16 (EXP) | | 2 |
| Participant 17 (EXP) | | 2 | Participant 18 (EXP) | | 2 |
| Participant 19 (EXP) | | 2 | Participant 20 (EXP) | | 1 |
| Participant 21 (NV) | 0 | | Participant 22 (NV) | 1 | |
| Participant 23 (NV) | 1 | | Participant 24 (NV) | 0 | |
| Participant 25 (NV) | 0 | | | | |

| | T2-PBI-NV | T2-PBI-EXP |
|---|---|---|
| Mean | 1 | 1.86 |
| SD | 0.89 | 0.36 |
| p-value | 0.0033368251206698 | |

#### 5.5.2.3 Conclusions on task 2 for PowerBI

In task 2 with PBI, experienced users take more time to complete the task than novices. Because they spend time to maintain the quality of results by performing manual editing. Also results show that quality task performed by experienced is

better than novices. In both cases p-value is less than 0.05 so from statistical point there is significant difference in the observations for both the groups.

#### 5.5.2.4 Task 2 with SimpleVis time analysis

In Table 5.12, T2-SV-NV refers to Task 2 performed by novice users using the SimpleVis tool, while T2-SV-EXP represents Task 2 performed by experienced users with SimpleVis. The analysis of the results, as outlined in Table 5.12, reveals the following outcomes:

Table 5.12: Statistical Analysis of time taken in seconds by novices user and experienced users to complete the Task 2 with SimpleVis.

| $TimeTaken/TaskToolUser$ | Mean | SD | Min | Max |
|---|---|---|---|---|
| T2-SV-NV | 62.73 | 37.60 | 22 | 160 |
| T2-SV-EXP | 65.71 | 22.71 | 30 | 95 |
| p-value | 0.8076671292 | | | |

The user type based comparative analysis of time for task 2 with novices in Table 5.10 is as follows:

- The average time taken by T2-SV-NV (62.73) is slightly lower than that of T2-SV-EXP (65.71), suggesting that novice users spent nearly the same amount of time as experienced users. Despite SimpleVis being new to all participants and novices being unfamiliar with BI tools, they didn't face significantly more challenges when completing the directed tasks compared to experienced users, who already had BI tool experience. This may be due to SimpleVis's high level of automation and minimal need for manual adjustments, which novices often prefer over manual tasks.

- The standard deviation of time for Task 2 performed T2-SV-EXP (22.71) is slightly less than T2-SV-NV (37.60). It indicates observation in T2-SV-EXP are slightly more consistent than T2-SV-NV.

- p-value for Task 2 with SV is 0.80 (0.8076671292 to be exact). As the p-value is greater than 0.05, it means that there is no significant differences between observations of two groups in the statistical point of view.

- As shown in Figure 5.8, the yellow and green boxes show SimpleVis results for experienced and novice users. The yellow box has a slightly larger IQR than the green, suggesting that observations for T2-SV-EXP are slightly more

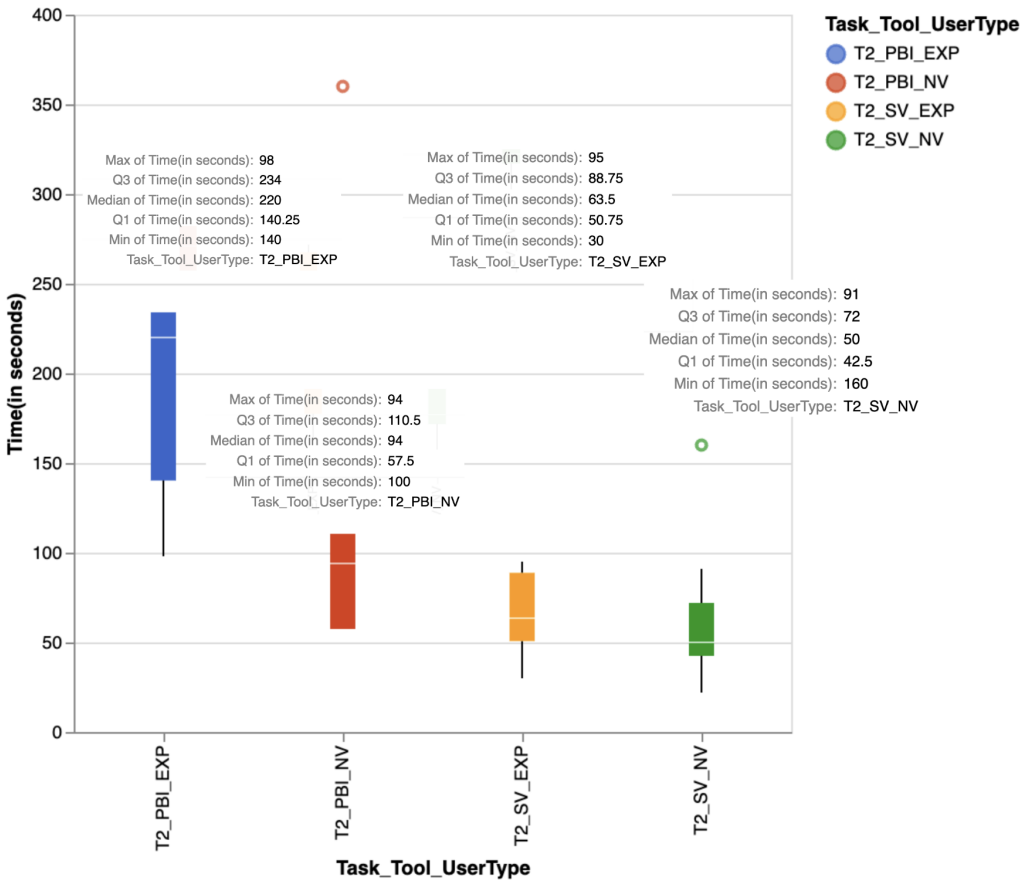



Figure 5.8: The boxes represent the summary statistics for Task 2, with SimpleVis results shown in blue for experienced users and red for novice users. On the other hand, the yellow and green boxes represent Task 2 completed using SimpleVis by experienced and novice users, respectively.

spread than for T2-SV-NV. The median for the green box representing T2-SV-NV (50) is lower than that for the yellow box representing T2-SV-EXP (63.5). There is no observations that deviate significantly from the rest of the data because there is only one outlier for T2-SV-NV (green).

#### 5.5.2.5 Task 2 with SimpleVis quality analysis

From the Table 5.13 we come up with the following comparative quality analysis of task 2 with SV is as follows:

- The mean of T2-SV-EXP (1.93) is greater than T2-SV-NV (1.55). It indicates that quality of task 2 performed by experienced is slightly better than novices using SimpleVis.

Table 5.13: Quality analysis of task 2 performed by novices and experienced participants with SV.

| *Participants/Task(Tool)* | T2-SV-NV | T2-SV-EXP | *Participants/Task(Tool)* | T2-SV-NV | T2-SV-EXP |
|---|---|---|---|---|---|
| Participant 1 (NV) | 2 | | Participant 2 (EXP) | | 2 |
| Participant 3 (NV) | 0 | | Participant 4 (NV) | 2 | |
| Participant 5 (EXP) | | 2 | Participant 6 (EXP) | | 1 |
| Participant 7 (NV) | 2 | | Participant 8 (EXP) | | 2 |
| Participant 9 (NV) | 1 | | Participant 10 (EXP) | | 2 |
| Participant 11 (EXP) | | 2 | Participant 12 (EXP) | | 2 |
| Participant 13 (EXP) | | 2 | Participant 14 (EXP) | | 2 |
| Participant 15 (NV) | 2 | | Participant 16 (EXP) | | 2 |
| Participant 17 (EXP) | | 2 | Participant 18 (EXP) | | 2 |
| Participant 19 (EXP) | | 2 | Participant 20 (EXP) | | 2 |
| Participant 21 (NV) | 0 | | Participant 22 (NV) | 2 | |
| Participant 23 (NV) | 2 | | Participant 24 (NV) | 2 | |
| Participant 25 (NV) | 2 | | | | |

| | T2-SV-NV | T2-SV-EXP |
|---|---|---|
| Mean | 1.55 | 1.93 |
| SD | 0.82 | 0.27 |
| p-value | 0.11292275650921 | |

- Standard deviation for T2-SV-NV is 0.82 and T2-SV-EXP is 0.27. It indicates observation in T2-SV-EXP are slightly more consistent than T2-SV-NV.

- p-value for Task 2 with SV is 0.11 (0.11292275650921 to be exact). As the

p-value is greater than 0.05, it means that there is no significant differences in the observations of participants of both groups from statistical point of view.

#### 5.5.2.6 Conclusions on task 2 with SimpleVis

In task 2 with SV, novices users take almost same (slightly less) time that of experienced users. Because novices can easily learn and use simple BI tool like SimpleVis which supports more automation. Also results show that quality of task performed by novices is less than experienced users. In both cases p-value is greater than 0.05, so from statistical point there is no significant difference in the observations for both the groups in terms of time and quality of task.

## 5.6 System evaluation analysis summary

In Table 5.14, and Table 5.15 we presented the task based and user type based system evaluation analysis summary respectively.

Table 5.14: Task Based Analysis Summary

| *Task* | *Dimension* | *TaskTool* | Mean | SD | p-value | Remark |
|---|---|---|---|---|---|---|
| Task 1 | Time | T1-PBI | 101.48 | 85.40 | 0.002 | SV faster than PBI |
| | | T1-SV | 42.72 | 27.14 | | |
| | Quality | T1-PBI | 4.48 | 4.80 | 0.85 | SV same quality as PBI |
| | | T1-SV | 0.37 | 1 | | |
| Task 2 | Time | T2-PBI | 153.6 | 93.09 | 0.0001 | SV faster than PBI |
| | | T2-SV | 64.4 | 29.51 | | |
| | Quality | T2-PBI | 1.48 | 1.76 | 0.15 | SV same quality as PBI |
| | | T2-SV | 0.77 | 0.60 | | |

The task based analysis summary in Table 5.14 is as follows:

- For Task 1, SV is faster than PBI in terms of time taken for completing the task. For quality of DBo, SV have same quality as that of PBI. It indicates

that participants obtain the same quality of results with SV but in a much shorter time with SV.

- For Task 2 also, SV is faster than PBI and have same quality of DBo's as that of PBI. It indicates that participants obtain the same quality of results with SV for directed task but in a much shorter time with SV.

The user type based analysis summary in present in Table 5.15 is as follows:

- For Task 1 with PBI, NV takes similar time as EXP (NV takes slightly more because EXP already familiar with BI tools). For Task 1 with PBI NV obtained the DBo's of same quality as that of EXP users.
- For Task 1 with SV, NV takes similar time as EXP. And NV obtained similar DBo's quality as that of EXP.
- For task 1 with SV, novices have similar results to experts, so this is a good result for SV. Because these participants are novices who are not familiar with BI tools like experienced users. So they can reach the same level of quality as experts within same time.
- For Task 2 with PBI, EXP takes more time as NV. For Task 2 with PBI, EXP users obtained better quality of DBo's as that of NV users. This means that novices have troubles with PBI to solve the task.
- For Task 2 with SV, NV takes similar time as EXP. And NV obtained similar DBo's quality as that of EXP.
- For Task 2, Novices have no specific troubles (or at least the same as EXP). This means that they understand SV at the same level as EXP, so again this is a good result for SV.

Table 5.15: User Type Based Analysis Summary

| *Task* | *Dimension* | *TaskToolUser* | Mean | SD | p-value | Remark |
|---|---|---|---|---|---|---|
| Task 1 | Time | T1-PBI-NV | 101.27 | 110.87 | 0.68 | NV almost same as EXP |
| | | T1-PBI-EXP | 94.57 | 62.37 | | |
| | Quality | T1-PBI-NV | 4.91 | 4.79 | 0.42 | NV same quality as EXP |
| | | T1-PBI-EXP | 0.30 | 0.43 | | |
| | Time | T1-SV-NV | 40.45 | 28.04 | 0.72 | NV similar to EXP |
| | | T1-SV-EXP | 44.50 | 27.34 | | |
| | Quality | T1-SV-NV | 5 | 4.46 | 0.38 | NV similar to EXP |
| | | T1-SV-EXP | 0 | 1.34 | | |
| Task 2 | Time | T2-PBI-NV | 104 | 89.89 | 0.01 | EXP takes more than SV |
| | | T2-PBI-EXP | 192.57 | 77.87 | | |
| | Quality | T2-PBI-NV | 1 | 0.89 | 0.003 | EXP better than NV |
| | | T2-PBI-EXP | 1.86 | 0.36 | | |
| | Time | T2-SV-NV | 62.73 | 37.60 | 0.80 | NV similar to EXP |
| | | T2-SV-EXP | 65.71 | 22.71 | | |
| | Quality | T2-SV-NV | 1.55 | 1.93 | 0.11 | NV similar to EXP |
| | | T2-SV-EXP | 0.82 | 0.27 | | |

Chapter 6

# Conclusion and Perspective

Contents


**6.1 Conclusion . . . . . 131**
**6.2 Perspective . . . . . 132**
6.2.1 Short Term . . . . . 132
6.2.2 Long Term . . . . . 133
6.2.3 Broader and Generalize Perspective . . . . . 134


## 6.1 Conclusion

In the domain of data visualization, dashboards serve as powerful tools for simplifying and conveying complex information through structured, interactive visual representations. Despite their widespread use across various domains, creating an effective dashboard remains a challenging task. This thesis has explored the intricacies of dashboard creation, particularly focusing on the selection of visualizations and the development of Automatic Dashboard Recommendation Systems (ADRSs) to aid in the data analysis and visualization process.

The research presented here highlights the difficulties involved in selecting and organizing visualizations that are both data-appropriate and aligned with user goals. The challenge is further compounded when considering the interdependence of visualizations within a dashboard, where the overall effectiveness depends on how well these elements work together as a cohesive unit.

Through an in-depth examination of existing tools and methods, both in industry and academia, this thesis identifies the gaps in current approaches, particularly the lack of tools that fully automate the creation of dashboards for novice users. By developing a system that minimizes manual intervention and incorporates user preferences, this research aims to bridge these gaps, offering a solution that is both user-friendly and efficient.

The proposed SimpleVis tool not only recommends a complete set of co-adapted visualizations as a dashboard but also proposes individual visualizations manual editing, taking into account the global structure of the dashboard. This approach ensures that users, regardless of their expertise level, can create dashboards that are

tailored to their specific needs, thus enhancing decision-making and data exploration processes.

The primary goal of SimpleVis is to create an optimal dashboard by effectively combining visualizations and data attributes to align with user requirements. To achieve this, SimpleVis employs a genetic algorithm for the automatic generation of dashboards. Genetic algorithms are particularly well-suited to optimization problems with vast and complex solution spaces, making them an ideal choice for this task. Apart from optimized results, other distinguished GA features such as multimodality, user choice integration, and ease of parallelized computation make us choose it. The use of a GA offers several advantages, including optimization in complex spaces, adaptability, customization, and efficiency, making it a powerful tool for this application.

In conclusion, this thesis contributes to the field of data visualization by providing a novel approach to dashboard recommendation that leverages automation and user-centered design. The system developed here has the potential to democratize the use of dashboards, making them more accessible to a broader audience, and ultimately enabling more effective communication of data-driven insights. The validation of this system through user evaluations underscores its practicality and relevance, paving the way for further advancements in the automation of dashboard design.

## 6.2 Perspective

The current system does not need major improvements in terms of user interface during all three steps of dashboard recommendation and analysis. However, improvements can be made to make the system more elegant, simple, and secure for the users. These improvements can be in terms of features, functionalities, and dashboard recommendation quality that meets user expectations. We have divided our perspective into short-term goals and long-term goals for system improvement.

### 6.2.1 Short Term

**System feature and functionality** The system already contains many interesting features and functionalities but still more can be added as follows:

- In the metadata section of step 2, the user can drag and drop the data attribute into different data attribute types instead of selecting from the drop-down.
- In the user data section of step 2, the user can remove/block a particular column and generate the dashboard for a certain number of data attribute columns.
- In the user preferences section of step 2, there is synchronization between user type and selected visualization. This synchronization can be also implemented between user types and a number of visualizations. For Novices, beginners, and experts it can be 5, 7, and 9 respectively as a default setting.

**Automatic selection of data attribute type for newly uploaded data** Our system works perfectly with all five in-build datasets. But in the case of a new dataset, the user needs to choose the right data attribute type for corresponding data attributes. But by default, it selects randomly one of the four data attribute types. This random selection needs to be verified by a user based on knowledge. To avoid this user verification we need to develop an algorithm that selects the correct data attribute type for corresponding data attributes and avoid the random selection.

**Extension of visualizations by adding new visualizations** The system supports only 6 visualization: Histogram, SimpleBar, Pie Chart, LineChart, Scatterplot, and Bar Chart. More visualizations like Heat Maps, radar charts, maps, parallel coordinates, etc. can be added. This will also cover more complex datasets and data attribute types.

**Commercialisation of tool for common users** The system is available for free to all types of individual users. But in the future, we can commercialize it with more advanced and pro features. If users want to access the advanced and pro features, they need to have a paid subscription.

**System documentation and video tutorial** The system already contains a complete tutorial with documentation to guide the user from scratch. But still, we need to give some video tutorials for the system. It will be more helpful for novice users to use our tool easily and learn fast.

### 6.2.2 Long Term

**Proposition of predefined templates** The SimpleVis system only supports the automatic recommendation of dashboards. The predefined template features can be added based on the user types, domain of datasets (like sports, medicine, finance, etc.), size of dataset, choice of visualization types, and data attribute types present in the dataset. These templates come from other users who found good DBo.

**User account and previous projects** A login and logout system can be created for the users. It will help user to save all the projects within their account. These projects can be accessed from anywhere and anytime for a long time, based on system terms and conditions.

**Previous projects and history based recommendation** We can track the users expectation based on previous projects and work history, and based on this information personal DBo's recommendation can be provided.

**Integration into third-party business tools** We have not yet developed the third-party APIs for the commercial and personalized use of the tool. These APIs help any business to use our software within their CRM frameworks and business analytics tools.

**Automatic implementation of suggestions given by 'dashboard rating system'** The system rating feature allows the user to rate the main dashboard present in the center of the screen. When users rate the dashboard less than 5, the system suggests some improvement. This improvement can only be implemented manually by the user. However, it can be more interesting for users to implement these suggestions automatically and rerun GA.

**Genetic algorithm improvements** There are still possibilities to improve the genetic algorithm by improving fitness function. During the repetition of data attributes in the recommended dashboard (within certain limits), a penalty can be assigned which can minimize the fitness value of the function. Certain combinations of operators and their coefficients can be tried to check if the fitness function can be improved considering user type configuration.

**Aggregation operators** On viewing the individual visualization in full-screen. A user can manipulate visualization type and data attributes but cannot apply the aggregation operators. We can add the feature to implement aggregations like Min, Max, Sum, Avg, and Count to the data attributes of each visualization.

**User interface effects: Brushing, Linkage, Annotation** It lacks advanced features and effects like brushing, linkage (interconnection of visualization based on common data attributes), and annotation. Once these effects are implemented the system will be more friendly for precise analytics.

**Secure and robust** We have not worked much on the security of the system. However, once we implement the user account creation feature in the system, the system will be more secure and robust.

### 6.2.3 Broader and Generalize Perspective

To broaden the perspective beyond the development of specific tools like SimpleVis, it is essential to consider the evolving trends, challenges, and opportunities in the field of data visualization as a whole. These perspectives can guide future research and innovation by addressing overarching themes and fostering advancements that apply to a wide range of use cases, tools, and technologies. The future of data visualization lies in creating tools and systems that are scalable, user-centered, transparent, and ethical. Advances in interactivity, automation, AI integration, and big data visualization will continue to transform the field. By addressing these broader research perspectives, the field of data visualization can evolve to meet the

growing demands of diverse users, from novices to experts, across a wide range of industries. The goal is not only to enhance the ability to visualize data but also to ensure that visualization should be automatic that lead to actionable insights and informed decision-making, driving innovation and understanding in an increasingly data-driven world.

# Appendices

Appendix A

# System use cases

## Contents


**A.1 Novice user use cases . . . . . 140**
A.1.1 Novice use case 1 . . . . . 141
A.1.2 Novice use case 2 . . . . . 143
A.1.3 Novice use case 3 . . . . . 146
**A.2 Expert user use cases . . . . . 150**
A.2.1 Expert use cases 1 . . . . . 151
A.2.2 Expert use case 2 . . . . . 164
**A.3 Use cases summary . . . . . 171**


To strengthen the motivation behind each use case, it is essential to focus on user objectives: what the user aims to achieve through the system rather than solely on the functionalities. By aligning the use cases with clear objectives, we can provide a more user-centric perspective, demonstrating not just how the tool works but why different users interact with it in distinct ways.

To provide a detailed and step-by-step understanding of the system, we have outlined two types use cases, novices and experts as shown in Figure A.1. The chapter presents total five use cases, three tailored for novice users and two designed for expert users.

The aims for both novice and expert users reflect their distinct needs and expectations when using the system. Novice users prioritize simplifying data analysis and dashboard generation, seeking an intuitive process that allows them to easily explore and learn through the tool. They value the ability to compare different dashboard solutions to determine the most suitable one for their task, even with no data expertise. Additionally, they want to retain some control over the data analysis process to gradually refine their understanding and results.

On the other hand, expert users focus on flexibility and customization, particularly when working with complex and diverse datasets. They require the ability to fine-tune visualizations for specific insights, ensuring that the tool supports manual editing that advanced decision-making tasks. Experts value precise control over the analytics process, enabling them to tailor visualizations to meet high-level strategic goals in complex scenarios.

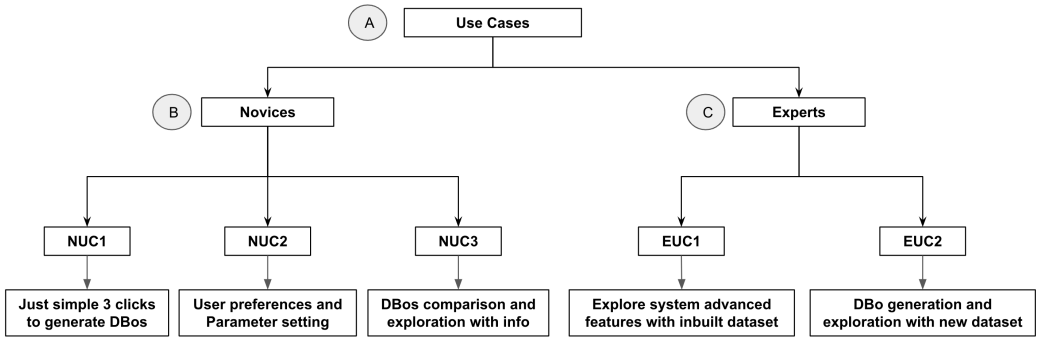



Figure A.1: We have presented (A) five use case, (B) three for novices and (C) two for experts. Each of it objective is mentioned below each of them.

By accommodating these varying objectives, the system ensures that both novices and experts can achieve their respective goals, from basic exploration to detailed and precise analytics. These use cases of the system have the following objectives:

- To propose the step-by-step exploration of the tool,
- How different user uses the tool differently,
- How do system features and functionality work when user type changes,
- Importance of user domain knowledge to understand the recommended dashboards, dashboard analytics, and decision-making.

During both use cases, we consider the same inbuilt dataset i.e. student.

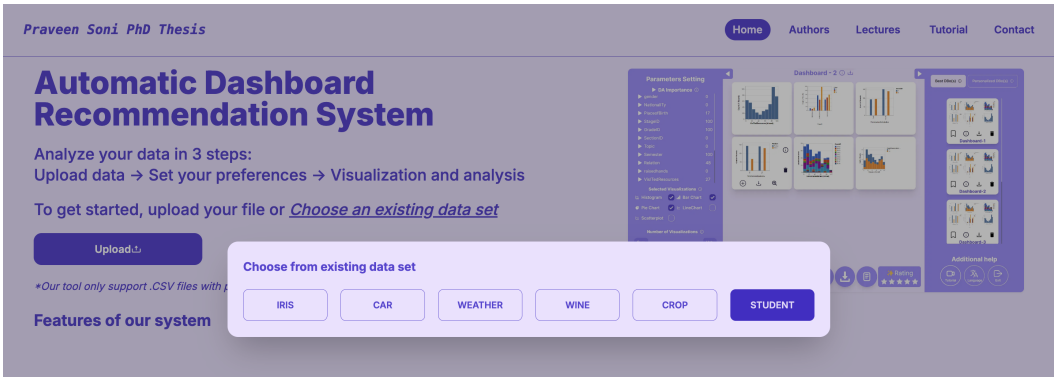
## A.1 Novice user use cases

The novice user performs the data analysis in three simple steps: dataset selection/upload, user preferences with metadata extraction, and dashboard recommendation. In the case of novices, we present three use cases:

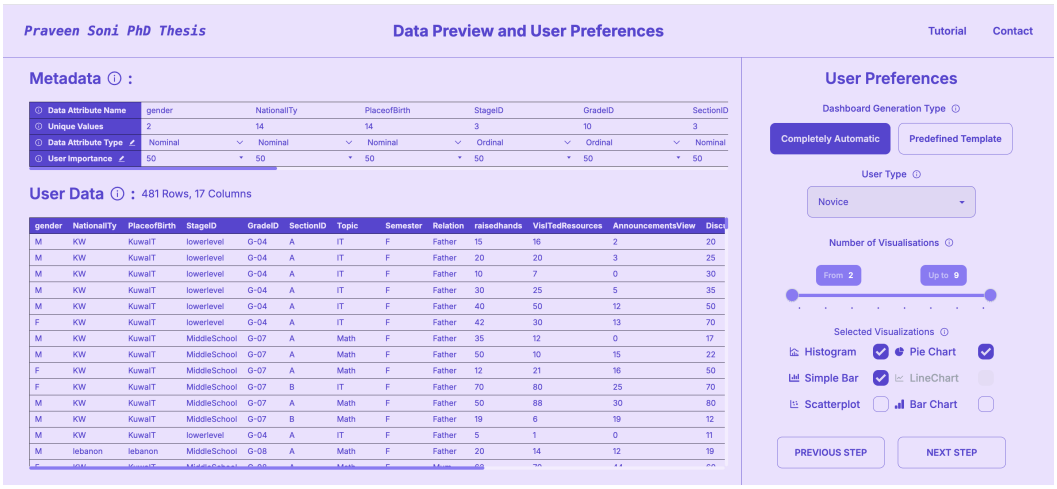
1. In novice use case 1 (NUC1) the user only clicks the next button without any change in parameters during all three steps.
2. In novice use case 2 (NUC2) the user changes user importance for data attribute in the system parameter setting. The user unselects some visualizations and changes the number of visualizations scale to appear in the dashboard during step 3.
3. In novice use case 3 (NUC3) the user changes user importance for data attributes likewise NUC2 and rerun GA. Here user compares the two DBos (i.e. DBo1 and DBo3 randomly out of 5 recommended DBos) and their information: data attributes covers, visualization present and fitness value.

### A.1.1 Novice use case 1

The user selects the student dataset from the in-build dataset as shown below:



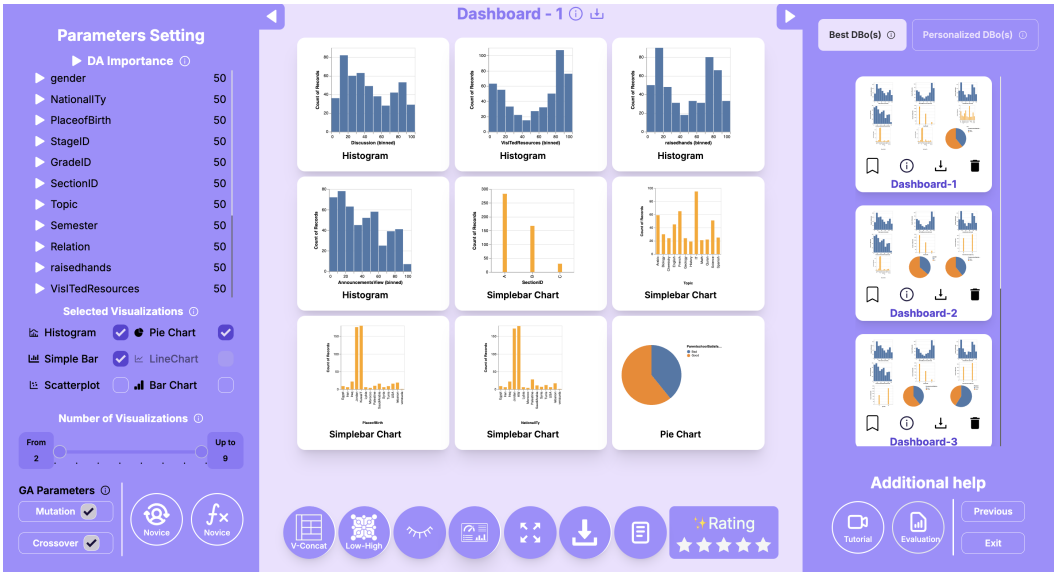

Once the user selects the dataset the next screen will appear for data preview and set user preference as shown below:




In the above picture, there are three sections: Metadata, user data, and user preferences. Users do not make any modifications and only click on the next button.

SimpleVis recommended the 5 best DBos. Then the Novice user obtains the DBos in the first run as in the picture below, DBo 1 and DBo 3:

Dashboard - 1




Dashboard - 3




The novice user have nothing more than a few clicks to obtain a complete DBo.

**Summary NUC1:** In NUC1, the user follows a straightforward approach, simply clicking the next button without adjusting any parameters. This case demonstrates how the system can generate dashboards automatically without user input or customization, relying entirely on its built-in algorithms. It serves as a baseline use case, showing the default behavior of the dashboard generation system and ensuring that even users unfamiliar with the tool can quickly produce a dashboard.

### A.1.2 Novice use case 2

In the use case of NUC2, step 1 and step 2 will be the same as that of NUC2. However, in step 3, the user used the parameter setting feature of the system to reset/change the preferences and rerun the GA. The change of preferences is done with the following three parameters:

1. User importance for data attribute in the system parameter setting,
2. Unselect some visualizations, and
3. Change the range of the number of visualizations to appear in the dashboard during step 3.

We present below each of them one by one:

a) User assigned the user importance for data attribute in the system parameter setting: The importance value for data attribute is as follows: NationallTy (100), StageID (0), GradeID (100), and SectionID (0), and Topic (100). For the rest of the data attribute default value 50 is used. The obtained DBo is shown below:



b) Unselect some visualizations: The user unselects the Simple Bar and keeps only the Histogram and Pie Chart as shown below.




c) Change the range of the number of visualizations. The user selects the lower bound 4 and upper bound 6. The system recommends the 5 DBos with 6 visualizations as shown below:



d) The DBo obtained with only a Pie Chart with 4 to 6 visualizations in DBo:




e) The given DBo was obtained with a SimpleBar and Bar Chart with 4 to 6 visualizations. The novice user can also select complex visualizations like Bar Charts as shown below:



f) The DBo in full-screen mode with SimpleBar and Bar Chart having 4 to 6 visualizations:

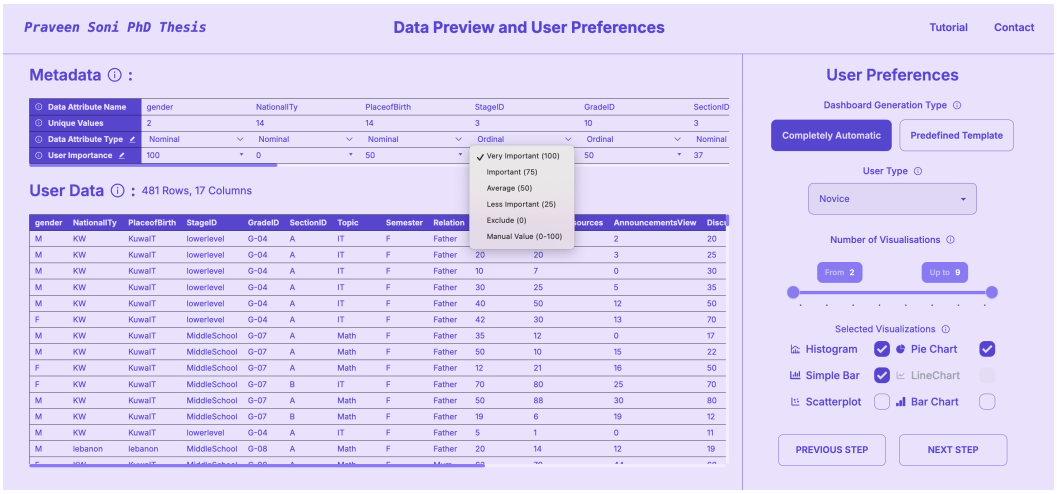



**Summary NUC2:** In NUC2, the user starts to interact more actively with the system by modifying the importance of specific data attributes and customizing the visualizations. This use case highlights how user preferences, such as selecting certain visualizations and adjusting the number of visualizations, can influence the generated dashboard. The modifications indicate that even with minor user interaction, the system remains flexible, allowing for more tailored dashboard outputs. It also showcases the system's ability to accommodate user-driven customization and deliver more relevant visualizations based on those preferences.

### A.1.3 Novice use case 3

In this novice use case user assigned importance to data attributes, selected/unselected visualizations, changed visualization number, and compared two dashboards based on information.

Here as shown in the picture below user has assigned importance to some of the data attributes:




The DBo generated are:




We reset the system parameters such as user importance (as shown for 'Placeof-Birth', 'Semester' and 'StageID' is 100, 'gender' is 49, 'SectionID' is 26, 'NationallTy' is 0, and for the rest data attribute is 50), selected visualizations (range from 2 to 9 is reassigned as from 4 to 7), and number of visualizations (Histogram, SimpleBar, and PieChart) then rerun GA to get DBo with its information as shown below:

Dashboard-1 (after GA rerun-1)




Dashboard-1 information




Again for GA rerun-2, we keep all the system parameters same as that of GA rerun-1.

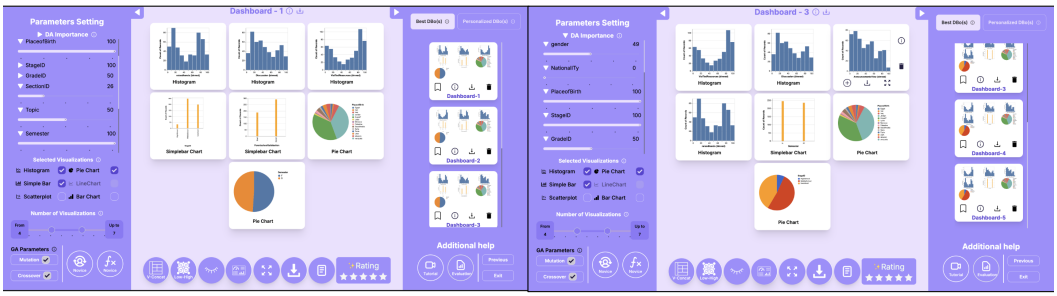
Dashboard-3 (after GA rerun-2)




Dashboard-3 information




Dashboard-1 and Dashboard-3 comparison



The comparison shows that:

1. Dashboard-1 contains 3 histograms, 2 Simple BarCharts and 2 PieCharts but other side Dashboard-2 contains 4 histograms, 1 Simple BarChart and 2 PieCharts,

2. Dashboard-1 and Dashboard-2 contains all 3 mandatory data attributes ('PlaceofBirth', 'Semester' and 'StageID') which have user importance 100 but excluded data attributes ('NationallTy' and 'SectionID') with 0 or lower importance (i.e. 26).

Dashboard-1 and Dashboard-3 information comparison



The comparison shows that:

1. For Dashboard-1 fitness value is 1.960 and Dashboard-3 fitness value is 1.963

2. For Dashboard-1 and Dashboard-2 total data attributes covered are 7

With the help of comparison of these two DBos and there information we can conclude that rerun GA makes the significant effects although keeping all system parameters constant.

**Summary NUC3:** In NUC3, the user further engages with the system by changing the importance of data attributes and rerunning the genetic algorithm (GA) to compare two dashboards (DBo1 and DBo3). This case focuses on evaluating the system's capability to support dashboard comparison based on multiple factors like data attribute coverage, the visualizations present, and fitness values. The user gains a deeper understanding of how the system balances these factors to recommend optimal dashboards. This use case demonstrates how the system enables users to assess and select the most suitable dashboard from multiple recommendations by leveraging comparative analysis, enhancing decision-making in dashboard selection.

## A.2 Expert user use cases

In the case of the expert, we presented two use cases. One with an in-build dataset and the other with uploading a new dataset. The expert user use case aims present following features of the system:

- Select the in-build dataset like the novice use case in step 1,
- How the system works when the user uploads a new dataset,
- Parameter setting for metadata and user preferences in step 2,
- Parameter setting: for user importance, visualization selection, visualization number, and user type,
- GA parameters and fitness function configuration during step 3,
- Visualization manual editing: Change in visualization data attributes keeping visualization constant and the change in visualization keeping visualization data attributes constant.

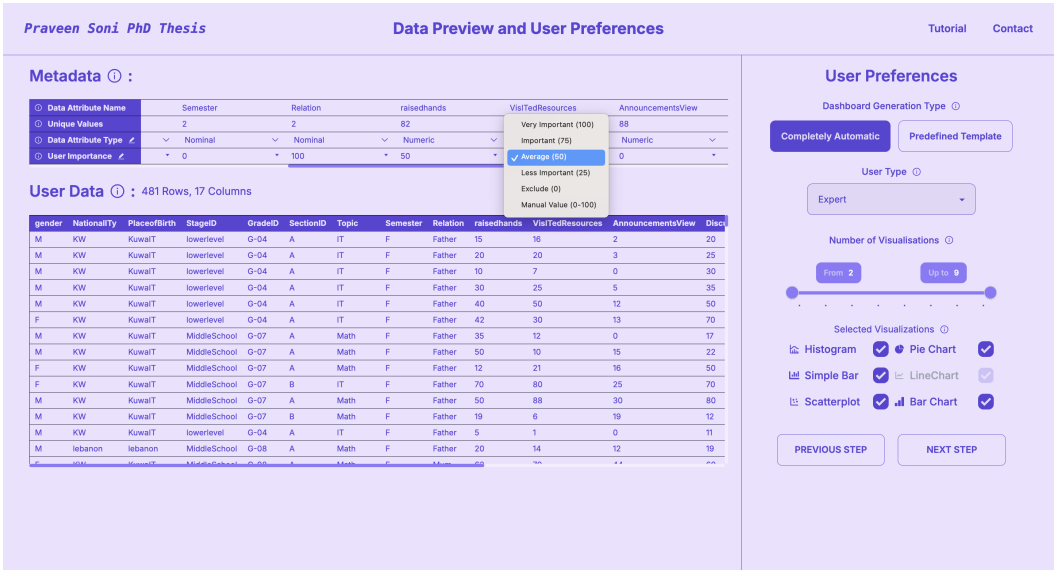
The overview of each of the use cases is presented below:

1. In expert use case 1 (EUC1) the user uses the in-build data set to explore the system features,
2. In expert use case 2 (EUC2) the user uses the new data set but uploading it from local machine and in step 2 assigned the data attributes type to each data attributes based on domain knowledge to explore the system features.

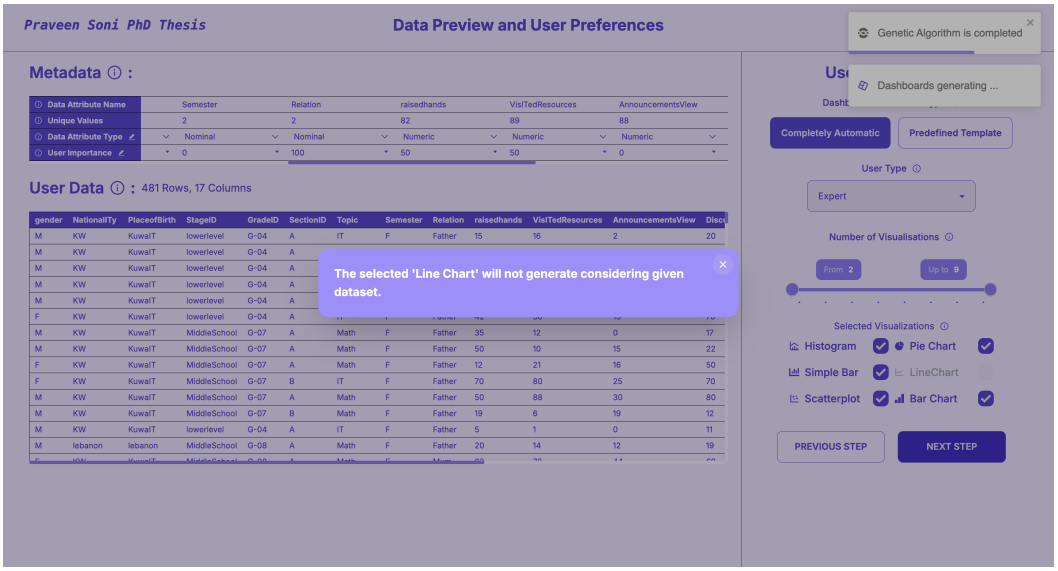
### A.2.1 Expert use cases 1

During the expert use case 1 (EUC1), the user uses the in-build dataset to explore the system features:

In step 1 for expert use cases, we select the in-build student dataset as we did for novices. In step 2 we assign some user importance for some data attributes manually as we did for novices. However here we selected the user type as expert as shown below:



When the user clicks on next, a pop-up will appear indicating that the given visualization is impossible for a given dataset.




In the next step 5 best DBos are proposed by the system automatically: Dashboard-1



Dashboard-2




Dashboard-3



Dashboard-4




Dashboard-5




With the help of these 5 dashboards, we created 3 personalized dashboards. To add visualization to personalized dashboards, the user can click on the '+' icon (it will appear when the user hovers on visualization or makes it full screen). All 3 personalized dashboards are shown below:

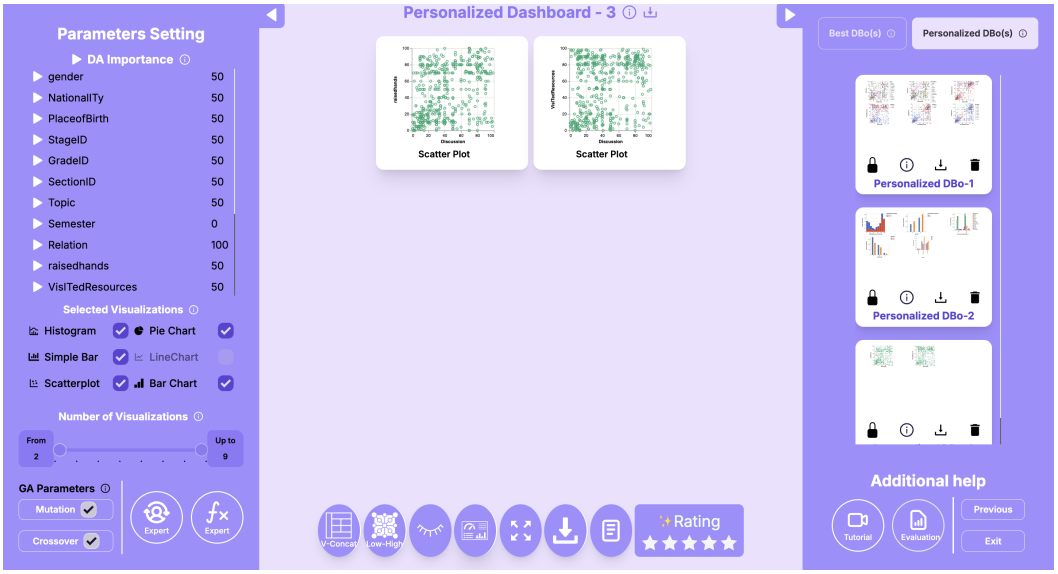
Personalized Dashboard-1: We added all scatter plots from all 5 recommended DBos.



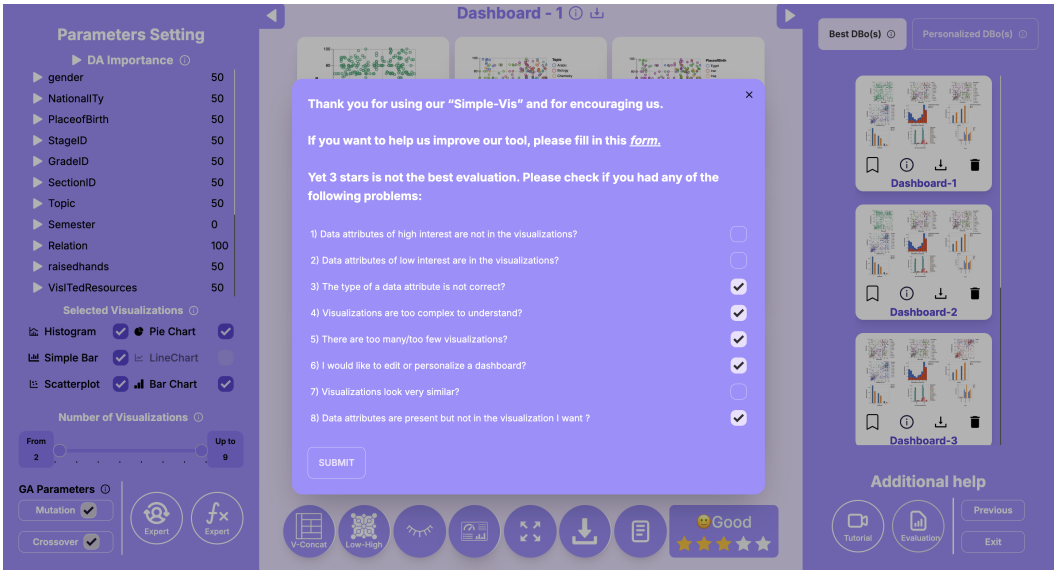

Personalized Dashboard-2: We added all bar charts from all 5 recommended DBos.



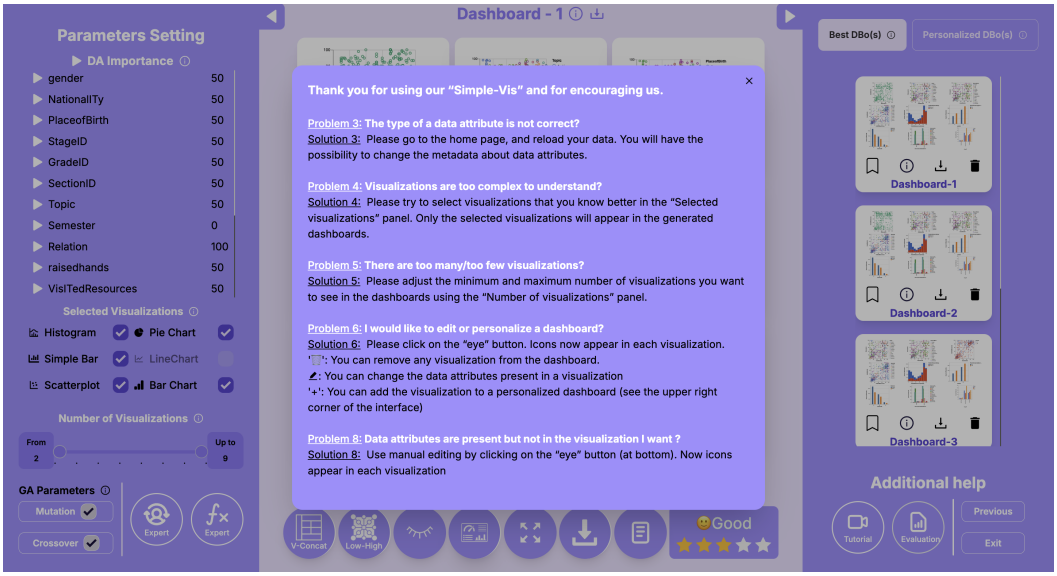
Personalized Dashboard-3: We added all simple scatter plots without legends from all 5 recommended DBos.




The user rated the DBo as 3* (three stars) and the system suggested a list of possible problems, shown below:



A user selects the problems (shown above) faced and submits the answer. The system proposed the solutions shown below:

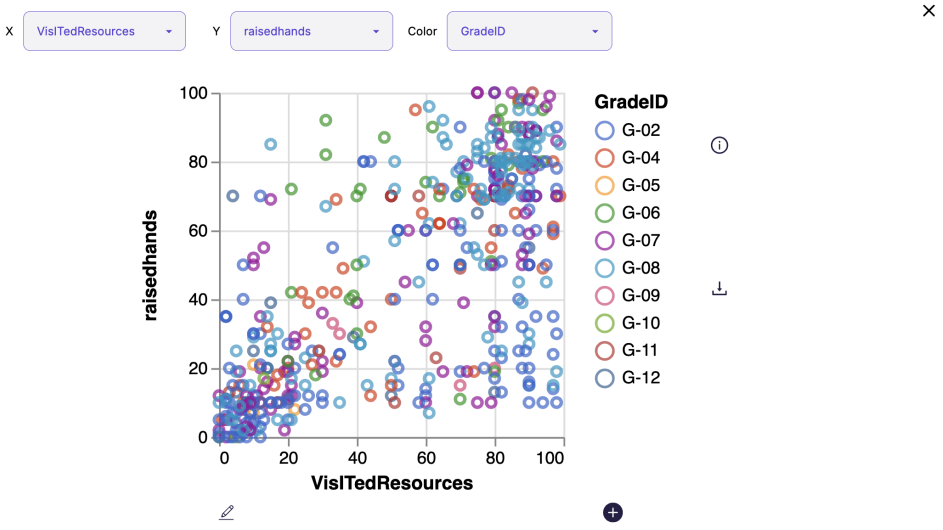



On Dashboard 3 (shown below), the user implemented the change in visualization data attributes keeping visualization constant and the change in visualization keeping visualization data attributes constant.

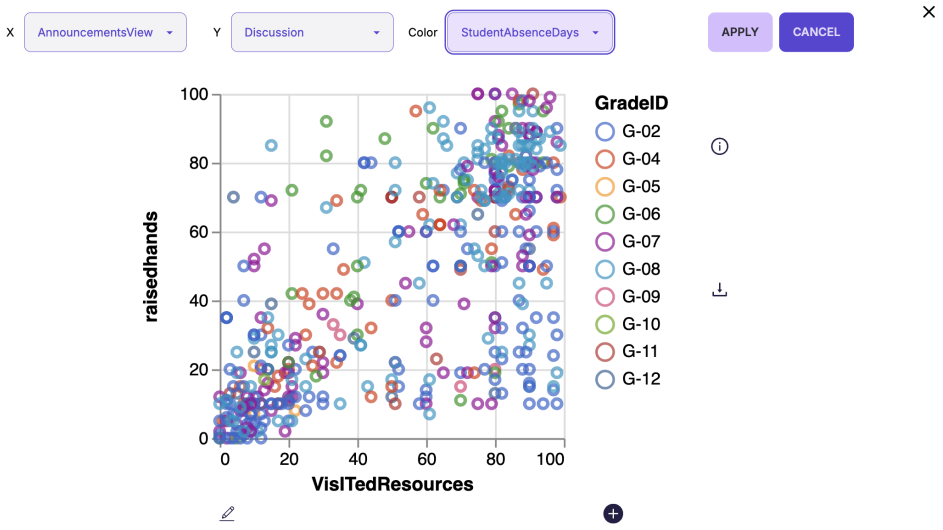
Dashboard 3



The user makes the visualization (scatterplot) present in the last row of the last column.



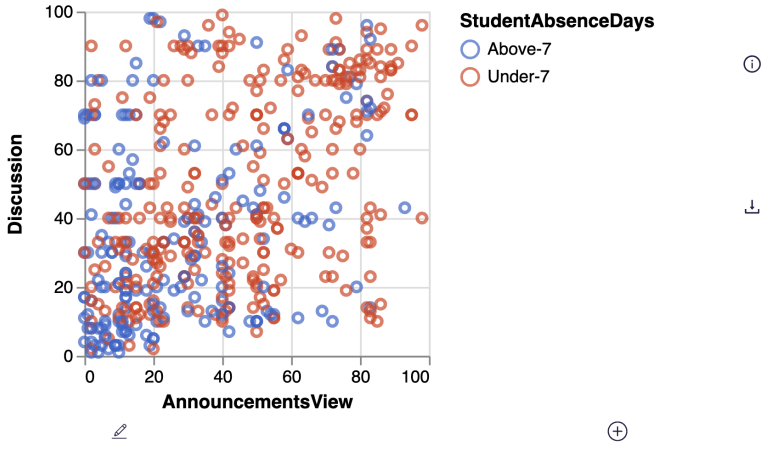

Once all three data attributes are changed user clicks on the apply button, shown below:



The visualization attributes are updated in DBo-3 shown below:



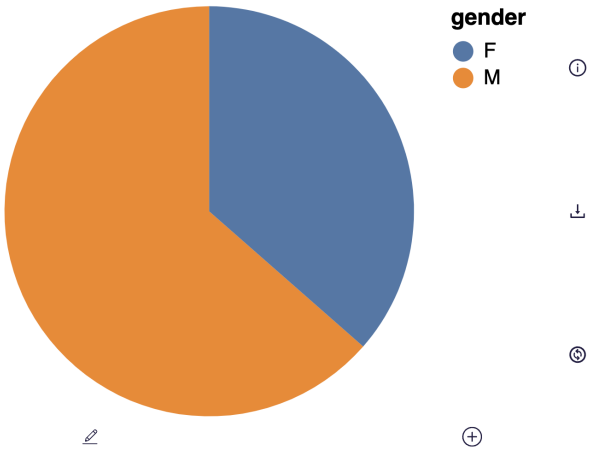

Also, see when you make it full screen shown below:



Similarly, users make changes in visualization keeping visualization data attributes constant. Again see DBo-3 having a pie chart in the first row and second column, shown below.



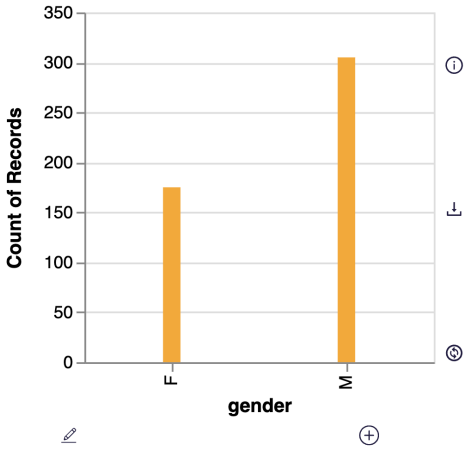

User make it full screen and click on the 'change' icon present right bottom as shown below:



The visualization (pie chart) is updated with a new visualization (simple bar) as shown below:




The user clicks on the updated visualization and makes it full screen:

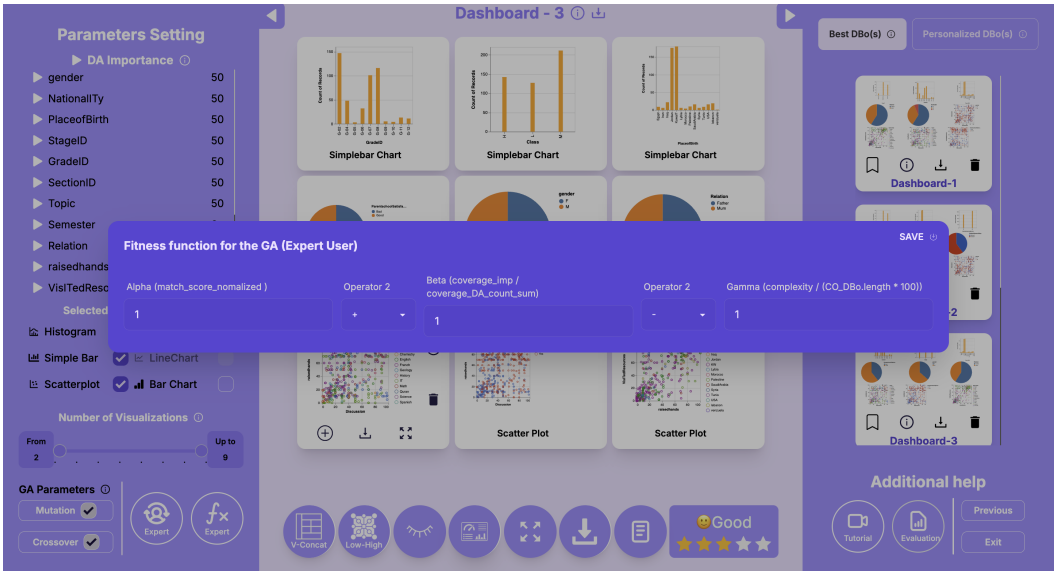


Users explore the notes feature by taking notes for DBo-3, shown below:




The user also tested the features of the fitness function configuration. The DBo-3 was generated with the default value (1, 2, 0.1 for alpha, beta, and gamma)of the fitness function coefficient.

But as shown below user changes the value of the coefficient as 1, 1, and 1.



After coefficient configuration, the user rerun the GA and came up with the following dashboards:




Users also explore the feature of GA parameter modification as shown below on the left bottom of the screen. Here user unselects the crossover and selects all three mutation operators.




Users can also explore the additional help section present on the right bottom as shown above: Tutorial, Evaluation, Previous step, and exit to the main screen.

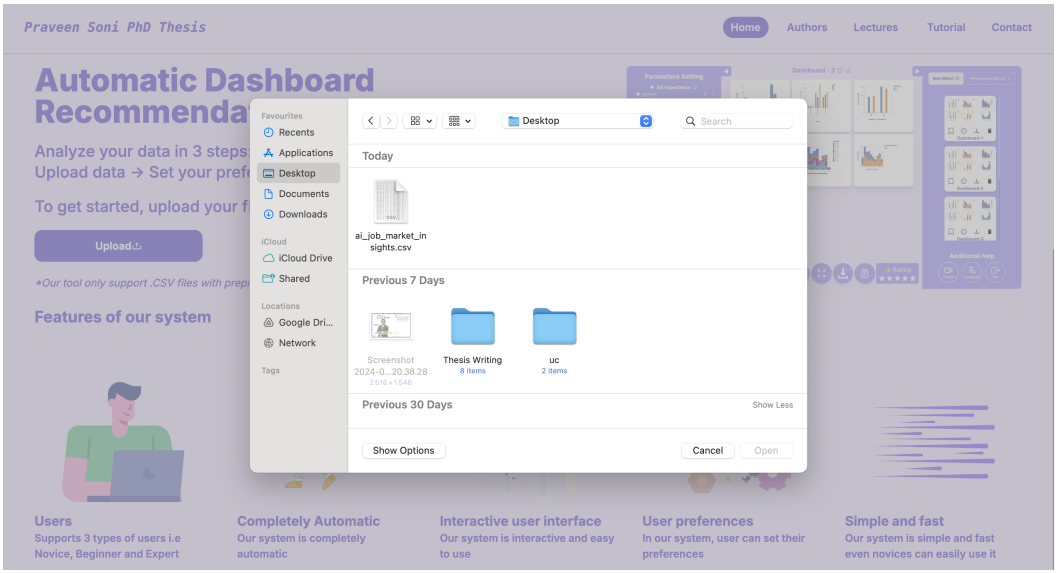
**Summary EUC1:** In EUC1, the expert user explores the system's features using the in-built dataset. This scenario highlights the system's capacity to accommodate experienced users who are more familiar with dashboard generation and data visualization. By leveraging the pre-loaded dataset, the user can focus on experimenting

with the system's advanced functionalities, evaluating the range of visualizations and customization options without the need to pre-process or upload external data. This use case shows how experts can efficiently navigate the system to explore its potential, utilizing their knowledge to maximize the tool's functionality and assess its capability for advanced use.

### A.2.2 Expert use case 2

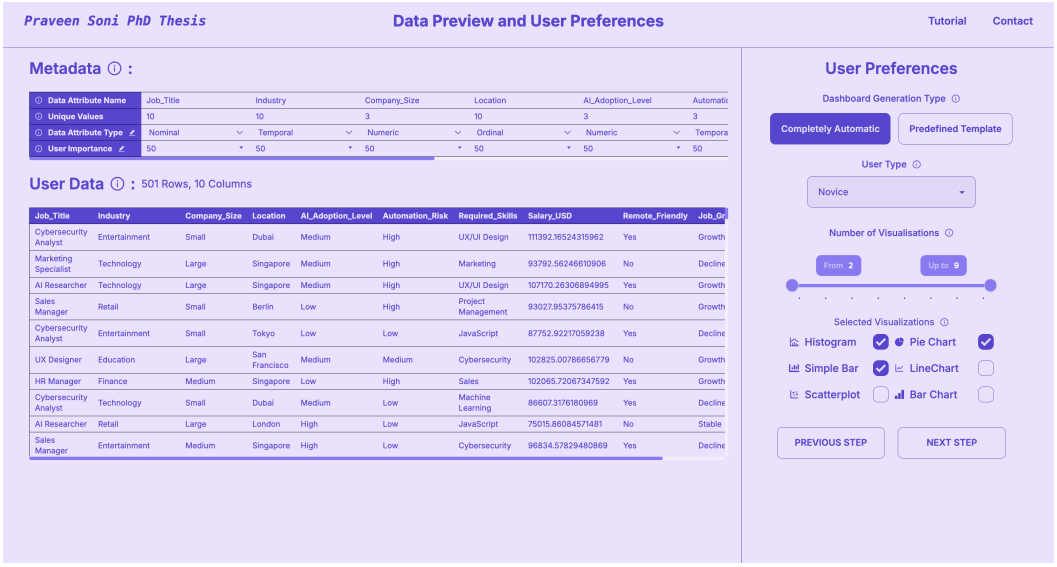
During the expert use case 2 (EUC2), the user uses the new dataset (AI-Powered Job Market Insights)[Laksika Tharmalingam 2024].

During EUC2, in step one user clicks on the upload button and selects the dataset file (AI-powered Job Market Insights) from the local machine as shown below:

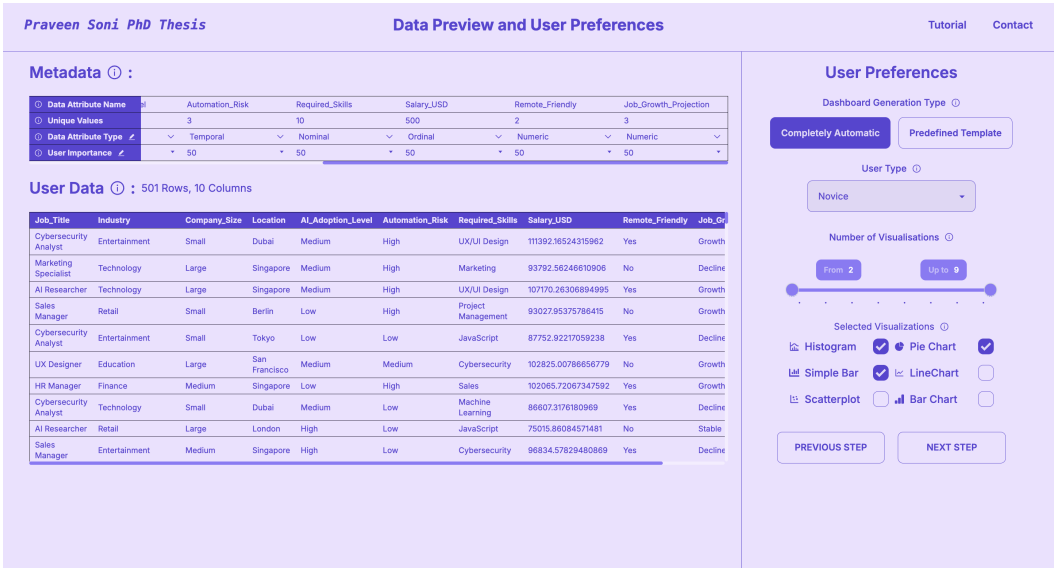



In step 2 all the data attribute types for respective data attributes are randomly picked by the system from the list of 4 types (Numeric, Nominal, Ordinal, and Temporal) as shown below. There are 510 rows and 10 columns in the dataset.

The picture shows the first 5 data attributes:

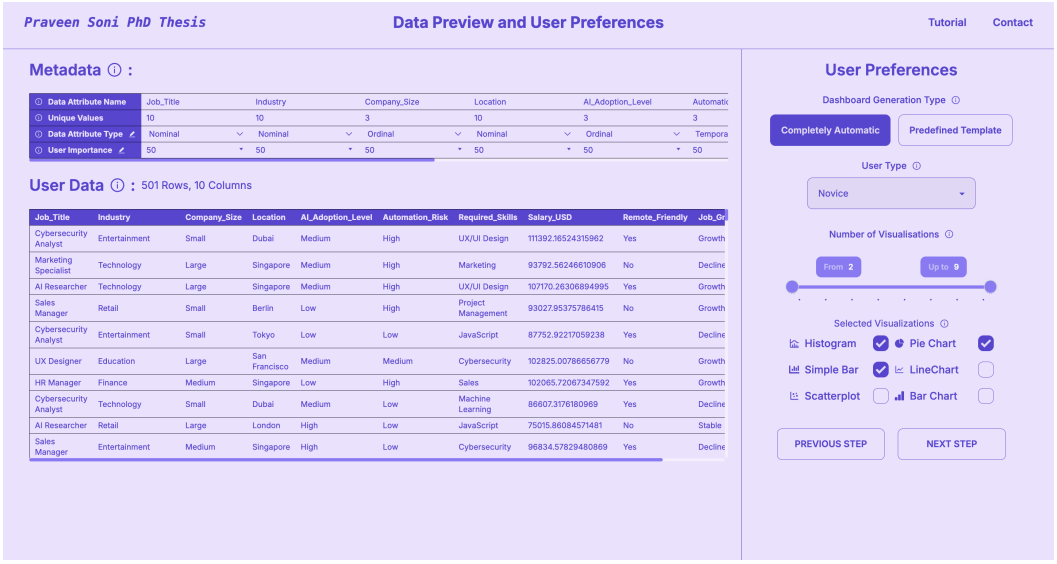



Picture the shows last 5 data attributes:

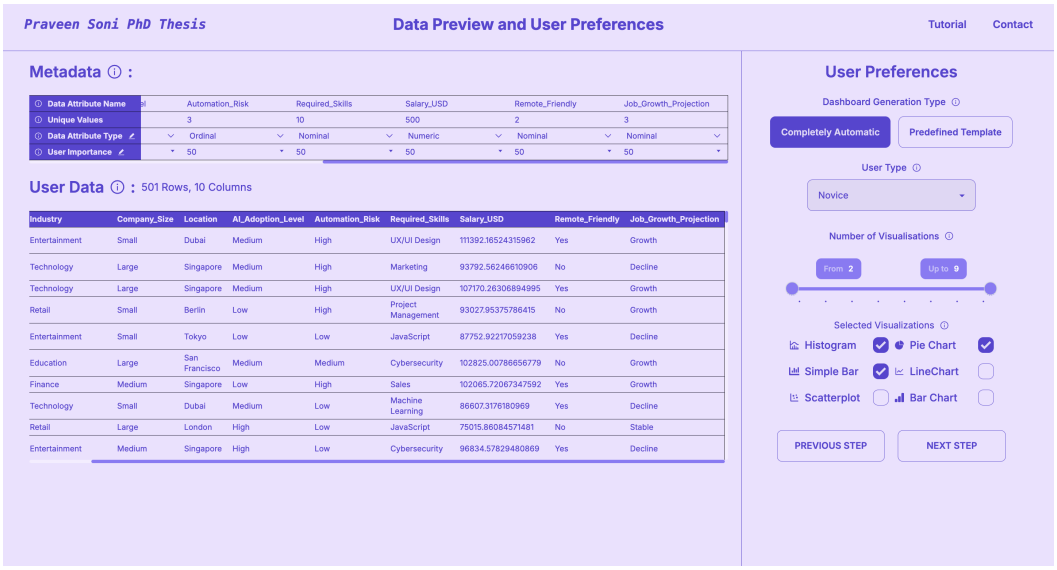



Based on user domain knowledge user reassigned the data attribute types for respective data attributes as shown below:

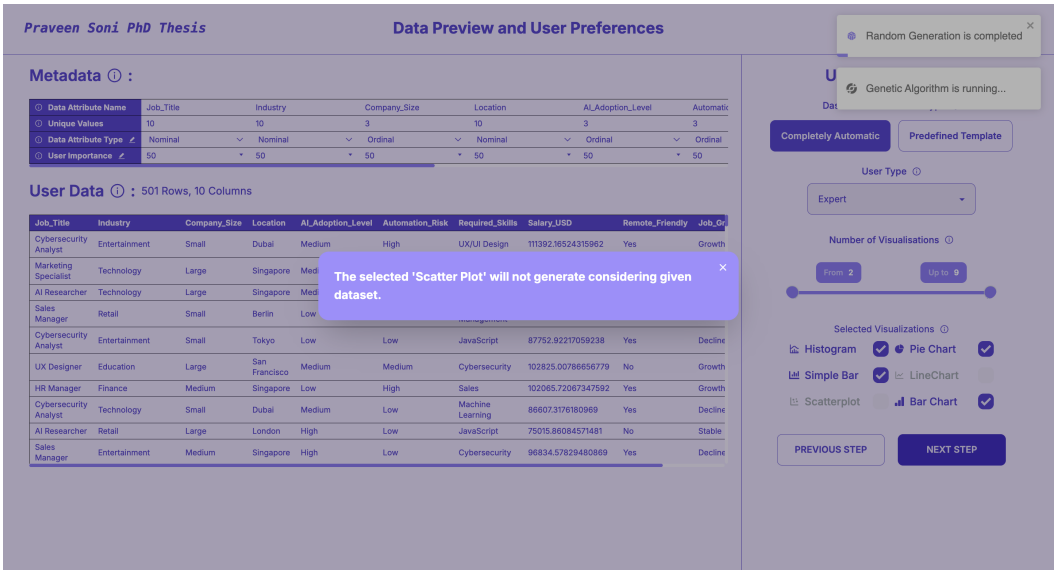
The picture shows the reassigned of data attribute types to the first 5 data attributes:


Praveen Soni PhD Thesis
Data Preview and User Preferences
Tutorial
Contact
Metadata ⓘ :
Data Attribute Name
Unique Values
Data Attribute Type
User Importance
Job_Title
Industry
Company_Size
Location
AI_Adoption_Level
Automatic
10
10
3
10
3
3
Nominal
Nominal
Ordinal
Nominal
Ordinal
Tempora
50
50
50
50
50
50
User Data ⓘ : 501 Rows, 10 Columns
Job_Title
Industry
Company_Size
Location
AI_Adoption_Level
Automation_Risk
Required_Skills
Salary_USD
Remote_Friendly
Job_Gr
Cybersecurity Analyst
Entertainment
Small
Dubai
Medium
High
UX/UI Design
111392.16524315962
Yes
Growth
Marketing Specialist
Technology
Large
Singapore
Medium
High
Marketing
93792.56246610906
No
Decline
AI Researcher
Technology
Large
Singapore
Medium
High
UX/UI Design
107170.26306894995
Yes
Growth
Sales Manager
Retail
Small
Berlin
Low
High
Project Management
93027.95375786415
No
Growth
Cybersecurity Analyst
Entertainment
Small
Tokyo
Low
Low
JavaScript
87752.92217059238
Yes
Decline
UX Designer
Education
Large
San Francisco
Medium
Medium
Cybersecurity
102825.00786656779
No
Growth
HR Manager
Finance
Medium
Singapore
Low
High
Sales
102065.72067347592
Yes
Growth
Cybersecurity Analyst
Technology
Small
Dubai
Medium
Low
Machine Learning
86607.3176180969
Yes
Decline
AI Researcher
Retail
Large
London
High
Low
JavaScript
75015.86084571481
No
Stable
Sales Manager
Entertainment
Medium
Singapore
High
Low
Cybersecurity
96834.57829480869
Yes
Decline
User Preferences
Dashboard Generation Type ⓘ
Completely Automatic
Predefined Template
User Type ⓘ
Novice
Number of Visualisations ⓘ
From 2
Up to 9
Selected Visualizations ⓘ
Histogram
Pie Chart
Simple Bar
LineChart
Scatterplot
Bar Chart
PREVIOUS STEP
NEXT STEP

The picture shows the reassigned of data attribute types to the last 5 data attributes:

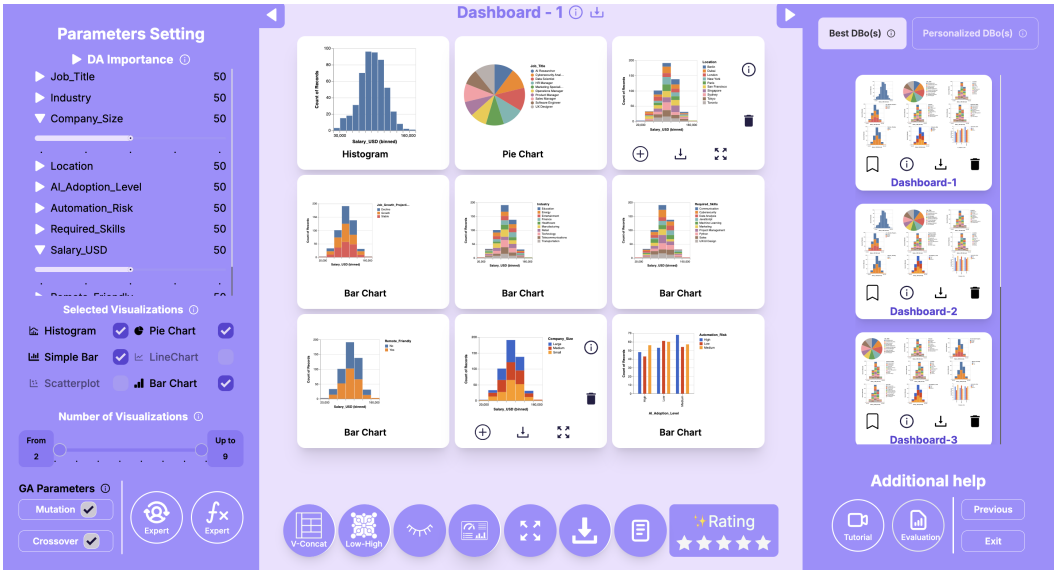


Before the user clicks on the next button to generate dashboards he changes user type to expert (shown below). The system gives a popup saying that a 'scatterplot will not be possible for the given dataset' as shown below. This is because, to generate a scatterplot at least 2 numeric data attributes are needed but the data set contains only one.




In step 3 system recommends the 5 best dashboards. We have shown below DBo-1 and DBo-4.

DBo-1



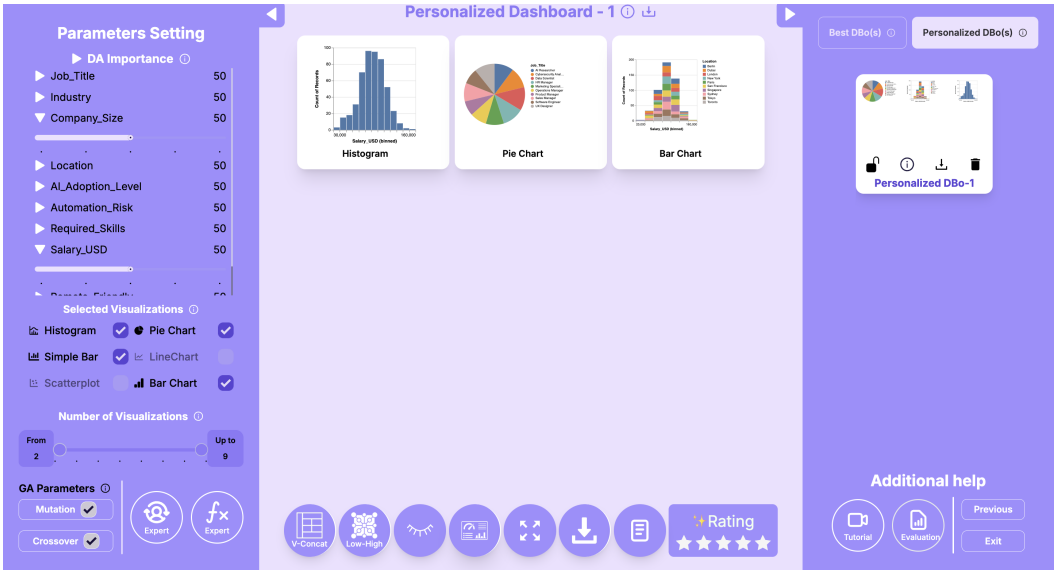
DBo-4




Users make the visualization full-screen for a close look. During full-screen system allows the user to perform manual editing on visualization data attributes, as shown below.



User also created on personalized DBo with three visualizations shown below:




The user reruns the GA by changing the user type to novice to check the difference. The generated DBos are shown below:

DBo-1



DBo-5




**Summary EUC2:** In EUC2, the expert user uploads a new dataset from a local machine and assigns data attribute types based on domain expertise. This use case underscores the system's ability to integrate external data while allowing users to apply their domain knowledge to define attributes manually. By exploring system features with a new dataset, the expert user can see how the system adapts to different data structures and requirements. The flexibility to assign data attribute types demonstrates that the system can support customized, domain-specific dashboard generation, giving experts full control over how data is handled and visualized. This case emphasizes the system's robustness in accommodating more complex, expert-driven use cases, where specialized knowledge plays a crucial role in tailoring the dashboard creation process.

## A.3 Use cases summary

The outlined use cases, targeting both novice and expert users, highlight how the system caters to different levels of user expertise and how user type influences the interaction with the tool. For novice users, the system simplifies the process into three main steps—dataset selection, user preferences with metadata extraction, and dashboard recommendation—while offering varying degrees of control (user preferences/ parameter setting) depending on the use case. Whether the user follows the default system settings (NUC1) or customizes parameters (NUC2 and NUC3), the system adapts to the user's skill level, providing automated recommendations and visualizations that require minimal domain knowledge.

For expert users, the system allows more granular control over the analysis process, from selecting or uploading datasets to configuring advanced parameters such as visualization preferences, metadata extraction, and the genetic algorithm (GA)

parameters. In both expert use cases (EUC1 and EUC2), the system demonstrates flexibility by supporting detailed adjustments (user preferences/ parameter setting and manual editing), whether using built-in datasets or uploading new ones. The system also enables users to manually edit visualizations, offering significant customization and in-depth analytics capabilities that align with the user's domain knowledge and expertise.

Overall, these use cases demonstrate the system's versatility in adapting to different user needs. Novice users benefit from an easy-to-use, automated process, while expert users can leverage a high degree of control and customization, enhancing the tool's effectiveness in supporting informed decision-making through tailored dashboard recommendations. This highlights the system's robustness in providing data-driven insights across varying levels of user proficiency and domain knowledge. All the five use cases demo is recorded and can be accessible by user by clicking on System Demo.

Appendix B

# System feedback

## B.1 System feedback

The feedback form contains the following questions:
Dear User,
Thank you for choosing our 'Simple Vis' data analysis tool. We greatly value your feedback as it helps us improve our product and service and better meet your needs. Please take a few minutes to share your thoughts on your experience with the tool.

**User Information:**
Full Name (Optional):
Organization (Optional):
Email Address (Optional):
User Type (Optional)

**Feedback form:**
1. Overall Satisfaction: How satisfied are you with the overall performance of our data analysis tool? (select one only)
a. Very Satisfied
b. Satisfied
c. Neutral
d. Dissatisfied
e. Very Dissatisfied

2. Ease of Use: How would you rate the ease of use of our data analysis tool? (select one only)
a. Very Easy
b. Easy
c. Neutral
d. Difficult
e. Very Difficult

3. Dashboard recommendation: How would you rate the automatic dashboard recommendation by the tool? (select one only)
a. Excellent
b. Good
c. Average

d. Poor
e. Very Poor

4. Features and Functionality: Which features and functionality of the tool do you find most valuable? (Multiple select)
a. Data Preview
b. Data Visualization
c. Dashboard recommendation
d. Parameters Setting
e. Best Dashboard
f. Personalized Dashboard
g. Dashboard Bottom Navigation
h. Feedback Rating system
i. Tool Tutorial
j. GA Parameters

5. Performance: How would you rate the speed and responsiveness of the tool? (select one only)
a. Excellent
b. Good
c. Average
d. Poor
e. Very Poor

6. Recommendations: Would you recommend our data analysis tool to others? (select one only)
a. Definitely
b. Probably
c. Not Sure
d. Probably Not
e. Definitely Not

7. User Interface: What are your thoughts on the user interface design? (text)

8. Challenges: Is there anything specific about the tool that you find challenging? (text)

9. Is the tool tutorial helpful?
a. Yes
b. No

10. Technical Support: Do you need any technical support? (select any one)

a. Yes
b. No

11. Additional Comments: Please share any additional comments or suggestions you have about the tool. (Text)

Message to the user after form submission:
Thank you for your valuable feedback! Your input is crucial in helping us enhance our data analysis tool. If you have any immediate concerns or issues, please contact our support team at praveensonifr@gmail.com

Sincerely,
Praveen Soni
SimpleVis
Link to form: System Demo. https://simplevis.soniconsultancy.in/

# Bibliography

[Abduldaem & Gravell 2019] Asmaa Abduldaem and Andy Gravell. *Principles for the design and development of dashboards: literature review.* Proceedings of INTCESS, pages 1307–1316, 2019. (Cited on pages 9 and 12.)

[Aksu *et al.* 2019a] Ünal Aksu, Adela del Río-Ortega, Manuel Resinas and Hajo A Reijers. *An approach for the automated generation of engaging dashboards.* Dans OTM Confederated International Conferences" On the Move to Meaningful Internet Systems", pages 363–384. Springer, 2019. (Cited on page 3.)

[Aksu *et al.* 2019b] Ünal Aksu, Adela del Río-Ortega, Manuel Resinas and Hajo A Reijers. *An approach for the automated generation of engaging dashboards.* Dans OTM Confederated International Conferences" On the Move to Meaningful Internet Systems", pages 363–384. Springer, 2019. (Cited on pages 18, 22, 23, 29 and 33.)

[Amazon Web Services 2016a] Amazon Web Services. *QuickSight.* https://aws.amazon.com/quicksight/, 2016. (Cited on page 3.)

[Amazon Web Services 2016b] Amazon Web Services. *QuickSight.* https://aws.amazon.com/quicksight/, 2016. (Cited on pages 19, 22, 26 and 29.)

[ANANTH R 2021] ANANTH R . *WEATHER PREDICTION.* https://www.kaggle.com/datasets/ananthr1/weather-prediction, 2021. (Cited on pages 73 and 99.)

[Atharva Ingle 2020] Atharva Ingle. *Crop Recommendation Dataset.* https://www.kaggle.com/datasets/atharvaingle/crop-recommendation-dataset/, 2020. (Cited on page 99.)

[Borkin *et al.* 2013] Michelle A Borkin, Azalea A Vo, Zoya Bylinskii, Phillip Isola, Shashank Sunkavalli, Aude Oliva and Hanspeter Pfister. *What makes a visualization memorable?* IEEE transactions on visualization and computer graphics, vol. 19, no. 12, pages 2306–2315, 2013. (Cited on page 20.)

[Brouns *et al.* 2015a] Francis Brouns, Marta Zorrilla, Elena Álvarez, Pedro Solana-González, Angel Cobo, Rocio Rocha, Marta Viaña, Carlos Hoyos, Mariana Silva, Carmen Lazo*et al. D2. 5 learning analytics requirements and metrics report.* Technical report, Technical Report, 2015. (Cited on page 9.)

[Brouns *et al.* 2015b] Francis Brouns, Marta Zorrilla, Elena Álvarez, Pedro Solana-González, Angel Cobo, Rocio Rocha, Marta Viaña, Carlos Hoyos, Mariana Silva, Carmen Lazo*et al. D2. 5 learning analytics requirements and metrics report.* Technical report, Technical Report, 2015. (Cited on page 9.)

[Chen *et al.* 2020a] Xi Chen, Wei Zeng, Yanna Lin, Hayder Mahdi Ai-Maneea, Jonathan Roberts and Remco Chang. *Composition and configuration patterns in multiple-view visualizations.* IEEE Transactions on Visualization and Computer Graphics, vol. 27, no. 2, pages 1514–1524, 2020. (Cited on page 3.)

[Chen *et al.* 2020b] Xi Chen, Wei Zeng, Yanna Lin, Hayder Mahdi Ai-Maneea, Jonathan Roberts and Remco Chang. *Composition and configuration patterns in multiple-view visualizations.* IEEE Transactions on Visualization and Computer Graphics, vol. 27, no. 2, pages 1514–1524, 2020. (Cited on pages 18, 20, 22, 24, 27 and 33.)

[Costa & Aparício 2019] Carlos J Costa and Manuela Aparício. *Supporting the decision on dashboard design charts.* Dans Proceedings of 254th The IIER International Conference 2019, pages 10–15, 2019. (Cited on page 2.)

[Da Col *et al.* 2021a] Sandrine Da Col, Radu Ciucanu, Marta Soare, Nassim Bouarour and Sihem Amer-Yahia. *DashBot: An ML-Guided Dashboard Generation System.* Dans Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pages 4696–4700, 2021. (Cited on page 3.)

[Da Col *et al.* 2021b] Sandrine Da Col, Radu Ciucanu, Marta Soare, Nassim Bouarour and Sihem Amer-Yahia. *DashBot: An ML-Guided Dashboard Generation System.* Dans Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pages 4696–4700, 2021. (Cited on pages 18, 22, 25, 27, 29, 30 and 33.)

[Dang & Wilkinson 2014a] Tuan Nhon Dang and Leland Wilkinson. *Scagexplorer: Exploring scatterplots by their scagnostics.* Dans 2014 IEEE Pacific visualization symposium, pages 73–80. IEEE, 2014. (Cited on page 3.)

[Dang & Wilkinson 2014b] Tuan Nhon Dang and Leland Wilkinson. *Scagexplorer: Exploring scatterplots by their scagnostics.* Dans 2014 IEEE Pacific visualization symposium, pages 73–80. IEEE, 2014. (Cited on pages 11, 18, 22, 27, 30 and 31.)

[Database Group 2018] Database Group. *DeepEye.* https://deepeye.tech/, 2018. (Cited on pages 1 and 18.)

[Demiralp *et al.* 2017] Çağatay Demiralp, Peter J Haas, Srinivasan Parthasarathy and Tejaswini Pedapati. *Foresight: Rapid data exploration through guideposts.* arXiv preprint arXiv:1709.10513, 2017. (Cited on page 3.)

[Diamond & Mattia 2017] Michael Diamond and Angela Mattia. *Data visualization: An exploratory study into the software tools used by businesses.* Journal of Instructional Pedagogies, vol. 18, 2017. (Cited on pages 12 and 22.)

[Dibia & Demiralp 2019] Victor Dibia and Çağatay Demiralp. *Data2vis: Automatic generation of data visualizations using sequence-to-sequence recurrent neural networks.* IEEE computer graphics and applications, vol. 39, no. 5, pages 33–46, 2019. (Cited on page 3.)

[El Meseery *et al.* 2018a] Maha El Meseery, Yuyao Wu and Wolfgang Stuerzlinger. *Multiple workspaces in visual analytics.* Dans 2018 International Symposium on Big Data Visual and Immersive Analytics (BDVA), pages 1–12. IEEE, 2018. (Cited on page 3.)

[El Meseery *et al.* 2018b] Maha El Meseery, Yuyao Wu and Wolfgang Stuerzlinger. *Multiple workspaces in visual analytics.* Dans 2018 International Symposium on Big Data Visual and Immersive Analytics (BDVA), pages 1–12. IEEE, 2018. (Cited on pages 18, 22, 23, 27, 30 and 31.)

[Elias & Bezerianos 2011a] Micheline Elias and Anastasia Bezerianos. *Exploration views: understanding dashboard creation and customization for visualization novices.* Dans IFIP conference on human-computer interaction, pages 274–291. Springer, 2011. (Cited on page 3.)

[Elias & Bezerianos 2011b] Micheline Elias and Anastasia Bezerianos. *Exploration views: understanding dashboard creation and customization for visualization novices.* Dans IFIP conference on human-computer interaction, pages 274–291. Springer, 2011. (Cited on pages 18, 20, 22, 23, 29, 31 and 33.)

[Elias & Bezerianos 2012] Micheline Elias and Anastasia Bezerianos. *Annotating bi visualization dashboards: Needs & challenges.* Dans Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pages 1641–1650, 2012. (Cited on page 12.)

[Elshehaly *et al.* 2020] Mai Elshehaly, Rebecca Randell, Matthew Brehmer, Lynn McVey, Natasha Alvarado, Chris P Gale and Roy A Ruddle. *QualDash: adaptable generation of visualisation dashboards for healthcare quality improvement.* IEEE Transactions on Visualization and Computer Graphics, vol. 27, no. 2, pages 689–699, 2020. (Cited on page 11.)

[Few 2003] Stephen Few. *Information dashboard design.* S. Few, 2003. (Cited on page 33.)

[Few 2006] Stephen Few. Information dashboard design: The effective visual communication of data, volume 2. O'reilly Sebastopol, CA, 2006. (Cited on page 9.)

[From Wikipedia, the free encyclopedia ] From Wikipedia, the free encyclopedia. *Metaheuristic.* https://en.wikipedia.org/wiki/Metaheuristic. (Cited on page 39.)

[Gonzalez *et al.* 2010] Hector Gonzalez, Alon Halevy, Christian S Jensen, Anno Langen, Jayant Madhavan, Rebecca Shapley and Warren Shen. *Google fusion tables: data management, integration and collaboration in the cloud.* Dans Proceedings of the 1st ACM symposium on Cloud computing, pages 175–180, 2010. (Cited on page 3.)

[Gowthami & Kumar 2017] K Gowthami and MR Pavan Kumar. *Study on business intelligence tools for enterprise dashboard development.* International Research Journal of Engineering and Technology (IRJET), vol. 4, no. 04, pages 2987–2992, 2017. (Cited on pages 1, 12, 27, 29, 31 and 33.)

[Hanrahan 2006] Pat Hanrahan. *Vizql: a language for query, analysis and visualization.* Dans Proceedings of the 2006 ACM SIGMOD international conference on Management of data, pages 721–721, 2006. (Cited on page 13.)

[Hu *et al.* 2018] Kevin Hu, Diana Orghian and César Hidalgo. *DIVE: A mixed-initiative system supporting integrated data exploration workflows.* Dans Proceedings of the workshop on human-in-the-loop data analytics, pages 1–7, 2018. (Cited on page 13.)

[Hu *et al.* 2019] Kevin Hu, Michiel A Bakker, Stephen Li, Tim Kraska and César Hidalgo. *Vizml: A machine learning approach to visualization recommendation.* Dans Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, pages 1–12, 2019. (Cited on page 13.)

[iPLesp, INSERM et Sorbonne Université ] iPLesp, INSERM et Sorbonne Université. *BiostaTGV.* https://biostatgv.sentiweb.fr/?module=tests/student. (Cited on pages 104 and 116.)

[Ji *et al.* 2014] Min Ji, Christine Michel, Elise Lavoué and Sébastien George. *DDART, a dynamic dashboard for collection, analysis and visualization of activity and reporting traces.* Dans European Conference on Technology Enhanced Learning, pages 440–445. Springer, 2014. (Cited on page 9.)

[Kaur & Owonibi 2017] Pawandeep Kaur and Michael Owonibi. *A review on visualization recommendation strategies.* Dans International Conference on Information Visualization Theory and Applications, volume 4, pages 266–273. SCITEPRESS, 2017. (Cited on pages 11 and 12.)

[Key *et al.* 2012a] Alicia Key, Bill Howe, Daniel Perry and Cecilia Aragon. *Vizdeck: self-organizing dashboards for visual analytics.* Dans Proceedings of the 2012 acm sigmod international conference on management of data, pages 681–684, 2012. (Cited on page 3.)

[Key *et al.* 2012b] Alicia Key, Bill Howe, Daniel Perry and Cecilia Aragon. *Vizdeck: self-organizing dashboards for visual analytics.* Dans Proceedings of the 2012 acm sigmod international conference on management of data, pages 681–684, 2012. (Cited on pages 18, 22, 25, 30 and 33.)

[Kubernátová *et al.* 2018] Petra Kubernátová, Magda Friedjungová and Max van Duijn. *Constructing a data visualization recommender system.* Dans International Conference on Data Management Technologies and Applications, pages 1–25. Springer, 2018. (Cited on page 12.)

[Laksika Tharmalingam 2024] Laksika Tharmalingam. *CAI-Powered Job Market Insights.* https://www.kaggle.com/datasets/uom190346a/ai-powered-job-market-insights?resource=download, 2024. (Cited on pages 50 and 164.)

[Law *et al.* 2020] Po-Ming Law, Alex Endert and John Stasko. *Characterizing automated data insights.* Dans 2020 IEEE Visualization Conference (VIS), pages 171–175. IEEE, 2020. (Cited on pages 11 and 12.)

[Luo *et al.* 2018a] Yuyu Luo, Xuedi Qin, Nan Tang and Guoliang Li. *Deepeye: Towards automatic data visualization.* Dans 2018 IEEE 34th international conference on data engineering (ICDE), pages 101–112. IEEE, 2018. (Cited on pages 1, 18, 22, 29, 30 and 31.)

[Luo *et al.* 2018b] Yuyu Luo, Xuedi Qin, Nan Tang and Guoliang Li. *Deepeye: Towards automatic data visualization.* Dans 2018 IEEE 34th international conference on data engineering (ICDE), pages 101–112. IEEE, 2018. (Cited on page 3.)

[Luo *et al.* 2020] Yuyu Luo, Chengliang Chai, Xuedi Qin, Nan Tang and Guoliang Li. *Visclean: Interactive cleaning for progressive visualization.* Proceedings of the VLDB Endowment, vol. 13, no. 12, pages 2821–2824, 2020. (Cited on page 3.)

[L'Yi *et al.* 2019] Sehi L'Yi, Youli Chang, DongHwa Shin and Jinwook Seo. *Toward understanding representation methods in visualization recommendations through scatterplot construction tasks.* Dans Computer Graphics Forum, volume 38, pages 201–211. Wiley Online Library, 2019. (Cited on page 33.)

[Ma *et al.* 2020] Ruixian Ma, Honghui Mei, Huihua Guan, Wei Huang, Fan Zhang, Chengye Xin, Wenzhuo Dai, Xiao Wen and Wei Chen. *Ladv: Deep learning assisted authoring of dashboard visualizations from images and sketches.* IEEE Transactions on Visualization and Computer Graphics, vol. 27, no. 9, pages 3717–3732, 2020. (Cited on page 11.)

[Mackinlay *et al.* 2007a] Jock Mackinlay, Pat Hanrahan and Chris Stolte. *Show me: Automatic presentation for visual analysis.* IEEE transactions on visualization and computer graphics, vol. 13, no. 6, pages 1137–1144, 2007. (Cited on pages 1, 2, 19, 22, 27, 29, 31 and 33.)

[Mackinlay *et al.* 2007b] Jock Mackinlay, Pat Hanrahan and Chris Stolte. *Show me: Automatic presentation for visual analysis.* IEEE transactions on visualization and computer graphics, vol. 13, no. 6, pages 1137–1144, 2007. (Cited on pages 3 and 41.)

[Mackinlay 1986] Jock Mackinlay. *Automating the Design of Graphical Presentations of Relational Information.* ACM Trans. Graph., vol. 5, pages 110–141, 04 1986. (Cited on page 16.)

[MathNerd 2018] MathNerd. *Iris Flower Dataset.* https://www.kaggle.com/datasets/arshid/iris-flower-dataset, 2018. (Cited on pages 73 and 99.)

[McAuley *et al.* 2019] John McAuley, Rohan Goel and Tamara Matthews. *ExploroBOT: Rapid Exploration with Chart Automation.* Dans VISIGRAPP (3: IVAPP), pages 225–232, 2019. (Cited on page 3.)

[Microsoft 2015a] Microsoft. *PowerBI.* https://powerbi.microsoft.com/, 2015. (Cited on pages 1, 2, 19, 22 and 26.)

[Microsoft 2015b] Microsoft. *PowerBI.* https://powerbi.microsoft.com/, 2015. (Cited on page 3.)

[Moher *et al.* 2009] David Moher, Alessandro Liberati, Jennifer Tetzlaff, Douglas G Altman and PRISMA Group*. *Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement.* Annals of internal medicine, vol. 151, no. 4, pages 264–269, 2009. (Cited on pages xiii and 10.)

[Moritz *et al.* 2018a] Dominik Moritz, Chenglong Wang, Greg L Nelson, Halden Lin, Adam M Smith, Bill Howe and Jeffrey Heer. *Formalizing visualization design knowledge as constraints: Actionable and extensible models in draco.* IEEE transactions on visualization and computer graphics, vol. 25, no. 1, pages 438–448, 2018. (Cited on page 3.)

[Moritz *et al.* 2018b] Dominik Moritz, Chenglong Wang, Greg L Nelson, Halden Lin, Adam M Smith, Bill Howe and Jeffrey Heer. *Formalizing visualization design knowledge as constraints: Actionable and extensible models in draco.* IEEE transactions on visualization and computer graphics, vol. 25, no. 1, pages 438–448, 2018. (Cited on pages 18, 22, 25 and 30.)

[Park & Jo 2015] Yeonjeong Park and I-H Jo. *Development of the learning analytics dashboard to support students' learning performance.* Journal of Universal Computer Science, vol. 21, no. 1, page 110, 2015. (Cited on page 9.)

[Pauwels *et al.* 2009] Koen Pauwels, Tim Ambler, Bruce H Clark, Pat LaPointe, David Reibstein, Bernd Skiera, Berend Wierenga and Thorsten Wiesel. *Dashboards as a service: why, what, how, and what research is needed?* Journal of service research, vol. 12, no. 2, pages 175–189, 2009. (Cited on pages 1, 12, 40 and 73.)

[Qin *et al.* 2020] Xuedi Qin, Yuyu Luo, Nan Tang and Guoliang Li. *Making data visualization more efficient and effective: a survey.* The VLDB Journal, vol. 29, no. 1, pages 93–117, 2020. (Cited on pages 11 and 12.)

[QlikTech International 2013a] QlikTech International. *Qlik Sense*. https://www.qlik.com/us/products/qlik-sense, 2013. (Cited on pages 1, 2, 19, 22, 26, 27, 29, 31 and 33.)

[QlikTech International 2013b] QlikTech International. *Qlik Sense*. https://www.qlik.com/us/products/qlik-sense, 2013. (Cited on page 3.)

[Riley Chang 2016] Riley Chang. *Cars Dataset*. https://github.com/vega/vega-datasets/blob/main/data/cars.json, 2016. (Cited on page 99.)

[Saket *et al.* 2018] Bahador Saket, Alex Endert and Çağatay Demiralp. *Task-based effectiveness of basic visualizations*. IEEE transactions on visualization and computer graphics, vol. 25, no. 7, pages 2505–2512, 2018. (Cited on page 31.)

[Sarikaya *et al.* 2018] Alper Sarikaya, Michael Correll, Lyn Bartram, Melanie Tory and Danyel Fisher. *What do we talk about when we talk about dashboards?* IEEE transactions on visualization and computer graphics, vol. 25, no. 1, pages 682–692, 2018. (Cited on pages 1, 9 and 12.)

[Satyanarayan & Heer 2014] Arvind Satyanarayan and Jeffrey Heer. *Lyra: An interactive visualization design environment*. Computer graphics forum, vol. 33, no. 3, pages 351–360, 2014. (Cited on page 3.)

[Schwendimann *et al.* 2016] Beat A Schwendimann, Maria Jesus Rodriguez-Triana, Andrii Vozniuk, Luis P Prieto, Mina Shirvani Boroujeni, Adrian Holzer, Denis Gillet and Pierre Dillenbourg. *Perceiving learning at a glance: A systematic literature review of learning dashboard research*. IEEE Transactions on Learning Technologies, vol. 10, no. 1, pages 30–41, 2016. (Cited on pages 9, 12, 40 and 73.)

[Sebastian Raschka 2014] Sebastian Raschka. *wine_data: A 3-class wine dataset for classification*. https://rasbt.github.io/mlxtend/user_guide/data/wine_data/, 2014. (Cited on pages 73 and 99.)

[Shen *et al.* 2021] Leixian Shen, Enya Shen, Yuyu Luo, Xiaocong Yang, Xuming Hu, Xiongshuai Zhang, Zhiwei Tai and Jianmin Wang. *Towards Natural Language Interfaces for Data Visualization: A Survey*. arXiv preprint arXiv:2109.03506, 2021. (Cited on pages 12 and 32.)

[Shi *et al.* 2020] Danqing Shi, Xinyue Xu, Fuling Sun, Yang Shi and Nan Cao. *Calliope: Automatic visual data story generation from a spreadsheet*. IEEE Transactions on Visualization and Computer Graphics, vol. 27, no. 2, pages 453–463, 2020. (Cited on pages 11, 18, 26 and 31.)

[Siddiqui *et al.* 2016a] Tarique Siddiqui, Albert Kim, John Lee, Karrie Karahalios and Aditya Parameswaran. *Effortless data exploration with zenvisage: an expressive and interactive visual analytics system*. arXiv preprint arXiv:1604.03583, 2016. (Cited on page 3.)

[Siddiqui *et al.* 2016b] Tarique Siddiqui, Albert Kim, John Lee, Karrie Karahalios and Aditya Parameswaran. *Effortless data exploration with zenvisage: an expressive and interactive visual analytics system.* arXiv preprint arXiv:1604.03583, 2016. (Cited on pages 18, 22, 29 and 30.)

[Sisense Inc 2004a] Sisense Inc. *Sisense.* https://www.sisense.com/get/free-trial/, 2004. (Cited on pages 1, 2, 19, 22, 26 and 29.)

[Sisense Inc 2004b] Sisense Inc. *Sisense.* https://www.sisense.com/get/free-trial/, 2004. (Cited on page 3.)

[Sourav Banerjee 2023] Sourav Banerjee. *Customer Shopping Trends Dataset.* https://www.kaggle.com/datasets/iamsouravbanerjee/customer-shopping-trends-dataset, 2023. (Cited on page 73.)

[Srinivasan *et al.* 2018] Arjun Srinivasan, Steven M Drucker, Alex Endert and John Stasko. *Augmenting visualizations with interactive data facts to facilitate interpretation and communication.* IEEE transactions on visualization and computer graphics, vol. 25, no. 1, pages 672–681, 2018. (Cited on page 12.)

[Stolte *et al.* 2002] Chris Stolte, Diane Tang and Pat Hanrahan. *Polaris: A system for query, analysis, and visualization of multidimensional relational databases.* IEEE Transactions on Visualization and Computer Graphics, vol. 8, no. 1, pages 52–65, 2002. (Cited on page 3.)

[Tableau Software 2007a] Tableau Software. *Tableau (ShowMe).* https://www.tableau.com/products/trial, 2007. (Cited on pages 1 and 2.)

[TABLEAU SOFTWARE 2007b] TABLEAU SOFTWARE. *Tableau (ShowMe).* https://www.tableau.com/products/trial, 2007. (Cited on page 3.)

[Toasa *et al.* 2018] Renato Toasa, Marisa Maximiano, Catarina Reis and David Guevara. *Data visualization techniques for real-time information—A custom and dynamic dashboard for analyzing surveys' results.* Dans 2018 13th Iberian Conference on Information Systems and Technologies (CISTI), pages 1–7. IEEE, 2018. (Cited on pages 9 and 12.)

[Tundo *et al.* 2020] Alessandro Tundo, Chiara Castelnovo, Marco Mobilio, Oliviero Riganelli and Leonardo Mariani. *Declarative Dashboard Generation.* Dans 2020 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), pages 215–218. IEEE, 2020. (Cited on page 1.)

[Vartak *et al.* 2015] Manasi Vartak, Sajjadur Rahman, Samuel Madden, Aditya Parameswaran and Neoklis Polyzotis. *Seedb: Efficient data-driven visualization recommendations to support visual analytics.* Proceedings of the VLDB Endowment International Conference on Very Large Data Bases, vol. 8, no. 13, page 2182, 2015. (Cited on page 3.)

[Vázquez-Ingelmo *et al.* 2019] Andrea Vázquez-Ingelmo, Francisco José García-Peñalvo and Roberto Therón. *Tailored information dashboards: A systematic mapping of the literature.* Dans Proceedings of the XX international conference on human computer interaction, pages 1–8, 2019. (Cited on pages 9 and 12.)

[Viegas *et al.* 2007] Fernanda B Viegas, Martin Wattenberg, Frank Van Ham, Jesse Kriss and Matt McKeon. *Manyeyes: a site for visualization at internet scale.* IEEE transactions on visualization and computer graphics, vol. 13, no. 6, pages 1121–1128, 2007. (Cited on page 3.)

[Wang *et al.* 2019] Yun Wang, Zhida Sun, Haidong Zhang, Weiwei Cui, Ke Xu, Xiaojuan Ma and Dongmei Zhang. *Datashot: Automatic generation of fact sheets from tabular data.* IEEE transactions on visualization and computer graphics, vol. 26, no. 1, pages 895–905, 2019. (Cited on pages 11, 18, 26 and 31.)

[Wills & Wilkinson 2010] Graham Wills and Leland Wilkinson. *Autovis: automatic visualization.* Information Visualization, vol. 9, no. 1, pages 47–69, 2010. (Cited on page 3.)

[Wongsuphasawat *et al.* 2015] Kanit Wongsuphasawat, Dominik Moritz, Anushka Anand, Jock Mackinlay, Bill Howe and Jeffrey Heer. *Voyager: Exploratory analysis via faceted browsing of visualization recommendations.* IEEE transactions on visualization and computer graphics, vol. 22, no. 1, pages 649–658, 2015. (Cited on page 24.)

[Wongsuphasawat *et al.* 2016] Kanit Wongsuphasawat, Dominik Moritz, Anushka Anand, Jock Mackinlay, Bill Howe and Jeffrey Heer. *Towards a general-purpose query language for visualization recommendation.* Dans Proceedings of the Workshop on Human-In-the-Loop Data Analytics, pages 1–6, 2016. (Cited on page 24.)

[Wongsuphasawat *et al.* 2017a] Kanit Wongsuphasawat, Zening Qu, Dominik Moritz, Riley Chang, Felix Ouk, Anushka Anand, Jock Mackinlay, Bill Howe and Jeffrey Heer. *Voyager 2: Augmenting visual analysis with partial view specifications.* Dans Proceedings of the 2017 chi conference on human factors in computing systems, pages 2648–2659, 2017. (Cited on page 3.)

[Wongsuphasawat *et al.* 2017b] Kanit Wongsuphasawat, Zening Qu, Dominik Moritz, Riley Chang, Felix Ouk, Anushka Anand, Jock Mackinlay, Bill Howe and Jeffrey Heer. *Voyager 2: Augmenting visual analysis with partial view specifications.* Dans Proceedings of the 2017 chi conference on human factors in computing systems, pages 2648–2659, 2017. (Cited on pages 18, 22 and 24.)

[Wu *et al.* 2014] Eugene Wu, Leilani Battle and Samuel R Madden. *The case for data visualization management systems: vision paper.* Proceedings of the VLDB Endowment, vol. 7, no. 10, pages 903–906, 2014. (Cited on page 3.)

[Wu *et al.* 2021a] Aoyu Wu, Yun Wang, Mengyu Zhou, Xinyi He, Haidong Zhang, Huamin Qu and Dongmei Zhang. *MultiVision: Designing Analytical Dashboards with Deep Learning Based Recommendation.* IEEE Transactions on Visualization and Computer Graphics, vol. 28, no. 1, pages 162–172, 2021. (Cited on pages 2, 9, 18, 22, 26, 29, 30 and 31.)

[Wu *et al.* 2021b] Aoyu Wu, Yun Wang, Mengyu Zhou, Xinyi He, Haidong Zhang, Huamin Qu and Dongmei Zhang. *MultiVision: Designing Analytical Dashboards with Deep Learning Based Recommendation.* IEEE Transactions on Visualization and Computer Graphics, vol. 28, no. 1, pages 162–172, 2021. (Cited on page 3.)

[Yuan *et al.* 2021] Linping Yuan, Ziqi Zhou, Jian Zhao, Yiqiu Guo, Fan Du and Huamin Qu. *Infocolorizer: Interactive recommendation of color palettes for infographics.* IEEE Transactions on Visualization and Computer Graphics, 2021. (Cited on page 33.)

[Zeng *et al.* 2021] Zehua Zeng, Phoebe Moh, Fan Du, Jane Hoffswell, Tak Yeon Lee, Sana Malik, Eunyee Koh and Leilani Battle. *An Evaluation-Focused Framework for Visualization Recommendation Algorithms.* IEEE Transactions on Visualization and Computer Graphics, vol. 28, no. 1, pages 346–356, 2021. (Cited on pages 22 and 32.)

[Zhang *et al.* 2019] Xiaozhong Zhang, Xiaoyu Ge, Panos K Chrysanthis and Mohamed A Sharaf. *ViewSeeker: An Interactive View Recommendation Tool.* Dans EDBT/ICDT Workshops, 2019. (Cited on page 3.)

[Zhu *et al.* 2020] Sujia Zhu, Guodao Sun, Qi Jiang, Meng Zha and Ronghua Liang. *A survey on automatic infographics and visualization recommendations.* Visual Informatics, vol. 4, no. 3, pages 24–40, 2020. (Cited on pages 11, 12 and 32.)

# Praveen SONI


Automatic generation of dashboard


Les Dashboard jouent un rôle crucial dans la visualisation de données complexes dans divers domaines, notamment la veille stratégique, l'éducation et l'administration publique. Ils aident les utilisateurs à surveiller les processus, à explorer les données et à tirer des enseignements grâce à des représentations visuelles structurées. Cependant, la création de tableaux de bord efficaces pose des défis importants, en particulier pour les novices ayant une expertise limitée en visualisation de données. Les difficultés courantes incluent la sélection des bonnes visualisations, la projection précise des données sur la visualisation et l'organisation des éléments dans un Dashboard cohérent. Bien que les outils existants offrent des interfaces conviviales, ils nécessitent souvent une saisie manuelle importante et des connaissances spécialisées, ce qui conduit à des tableaux de bord qui ont du mal à communiquer efficacement des données complexes.

Cette thèse présente SimpleVis, un système de recommandation automatique de Dashboard (SRAD) qui simplifie la création de tableaux de bord, en particulier pour les utilisateurs novices. SimpleVis automatise des aspects critiques tels que la sélection de visualisations, la cartographie des données et la mise en page à l'aide de techniques avancées telles que les algorithmes génétiques et l'analyse statistique. Contrairement aux outils traditionnels, SimpleVis génère un ensemble cohérent de visualisations interconnectées, formant un Dashboard structuré et perspicace.

Un examen approfondi des outils existants met en évidence leurs limites, en particulier pour aider les novices. SimpleVis comble ces lacunes en fournissant des recommandations automatisées pour une conception cohérente des tableaux de bord, tout en offrant des options de personnalisation aux utilisateurs experts. Une étude comparative des utilisateurs démontre que SimpleVis réduit la complexité de la création des tableaux de bord et améliore la qualité des visualisations. En démocratisant le processus, SimpleVis permet aux utilisateurs de tous niveaux de compétences de créer des tableaux de bord de qualité professionnelle, facilitant une meilleure prise de décision basée sur les données et des informations plus approfondies.



Mots-clés : Dashboard, algorithme génétique, multi-visualisation, automatisation, novices.



Dashboards play a crucial role in visualizing complex data across various fields, including business intelligence, education, and public administration. They help users monitor processes, explore data, and derive insights through structured visual representations. However, creating effective dashboards poses significant challenges, especially for novices with limited expertise in data visualization. Common difficulties include selecting the right visualizations, mapping data accurately, and organizing elements into a coherent dashboard. While existing tools offer user-friendly interfaces, they often require substantial manual input and specialized knowledge, leading to dashboards that struggle to communicate complex data effectively.

This thesis introduces SimpleVis, an Automatic Dashboard Recommendation System (ADRS) that simplifies dashboard creation, particularly for novice users. SimpleVis automates critical aspects like visualization selection, data mapping, and layout using advanced techniques such as genetic algorithms and statistical analysis. Unlike traditional tools, SimpleVis generates a cohesive set of interconnected visualizations, forming a structured and insightful dashboard.

A thorough review of existing tools highlights their limitations, especially in aiding novices. SimpleVis bridges these gaps by providing automated recommendations for coherent dashboard design, while also offering customization options for expert users. A comparative user study demonstrates that SimpleVis reduces the complexity of dashboard creation and enhances the quality of visualizations. By democratizing the process, SimpleVis enables users of all skill levels to create professional-quality dashboards, facilitating better data-driven decision-making and deeper insights.



Keywords: Dashboard, Genetic Algorithm, Multi-Visualization, Automation, Novices.