.

Enhancing Data Science Efficiency through Automated Visualization and Personalized Recommendation System

Tal Ariel Ziv, and *Arnon* Lutsky

[1] *Bar-Ilan University - Tabular Data science*

Abstract. Our final project aims to enhance and automate the data visualization process within the data science pipeline. Visualization is a critical step in understanding the data, allowing users to explore distributions, analyze relationships between features and target variables, and gain meaningful insights from different perspectives. By improving and automating this process, we seek to make data exploration more efficient, more intuitive, and accessible. Our solution is an algorithm that automatically analyzes the data for different statistical relations and interesting observations and recommends visualizations based on analysis and a recommendation system.

1 Problem description

Creating meaningful visualizations is a tedious and time-consuming task. It takes trial and error, and sometimes creating a good visualization requires familiarity with the data, which paradoxically may require visualizing it in the first place. Automating the generation of visualizations based on data characteristics and analysis can significantly enhance efficiency, accuracy, and scalability.

Data science pipelines involve a large amount of data and analysis. Automated visualization may help at various stages, from initial data exploration to model performance monitoring. This can ensure better evaluation of data quality, importance of features, and predictive model behavior. In addition, an automated visualization tool can make complex data representations accessible to non-technical users. This is particularly beneficial for business professionals and decision-makers who rely on visual analytics but lack expertise in programming or data visualization techniques.

Furthermore, increasing user satisfaction requires tailoring automatic visualizations to each user's needs. By adapting to user preferences, domain expertise, and specific analytical goals, an automated tool can provide more relevant and insightful visual representations, enhancing the overall experience and usability.

2 Solution overview

Our solution is composed of three main components, each one contributing in it's own way to create a complete solution:

- relation detection algorithm
- recommendation system
- plot generator

Each component addresses a critical need within our approach to effectively handle high correlation between features in data visualization. The relation detection algorithm identifies and quantifies the relationships between features, enabling us to reduce redundancy and highlight the most informative aspects of the data.

Then, the recommendation system provides actionable insights by suggesting the most relevant relation between features to visualizations based on the user's specific needs and the characteristics of the dataset.

Finally, the plot generator visualizes these insights in an interpretable manner, ensuring that end-users can easily understand the new, reduced feature space. This holistic approach not only enhances interpretability but also streamlines the analysis process, offering a practical solution to the challenges of high feature correlation.

In the following sections, we will take a closer look at each component of our solution.

2.1 Relation Detection Algorithm

Most datasets have a noticeable amount of relation between their features. Some features may be highly correlated and some may have non-linear relationships. Features may also have outliers that may influence the model's ability to predict a target feature. In our solution, we chose some interesting methods covered in class in order to extract valuable statistical information on the different features in a given dataset. Below, we present a detailed overview of these methods.

Correlation

Our algorithm calculates the Pearson correlation coefficients between two numerical columns and identifies highly correlated features.
The Pearson correlation formula is[1]:

$$\rho_{X,Y} = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y}$$

Pairs with correlation values exceeding a predefined threshold are recorded as potentially important relations. Identifying such relations helps detecting redundant features and potential collinearity. Therefore, providing a visualization of these detected relationships is essential.

Correlation with the Target Variable

Our algorithm calculates the correlation between numerical features and the target variable. Feature relationships with correlation values exceeding a defined threshold are considered potentially significant. Analyzing these correlations aids in identifying features with strong predictive power.

Categorical Effects

Our algorithm assesses the impact of categorical variables on a numerical target variable using Analysis of Variance (ANOVA). The ANOVA test compares the means of different groups, and if the p-value is below a predefined threshold, the relationship is considered statistically significant. These results provide insights into the importance of categorical features.

Categorical Association (Chi-Squared Test)

The chi-squared test[1] evaluates the dependence between categorical features by comparing observed and expected frequencies in contingency tables. When the p-value of the test is below a predefined threshold, the relationship between the variables is considered statistically significant and worth visualizing.

Date-Numerical Relationship

Date columns are transformed into ordinal values, allowing the algorithm to compute correlations with numerical features. Strong correlations indicate meaningful trends over time, providing valuable insights into temporal patterns.

Date-Categorical Relationship

The algorithm assesses the relationship between categorical features and time-based features using the chi-square test. Data is aggregated into monthly periods, and the test evaluates whether categorical distributions vary significantly over time, revealing potential temporal patterns.

Non-Linear Relationships

The algorithm computes mutual information scores between numerical features to detect strong relationships that may not be identified by linear correlation methods. High mutual information indicates significant non-linear dependencies, providing deeper insights into complex feature interactions.

Feature Importance

The algorithm employs a Random Forest Regressor to assess the importance of numerical features in predicting the target variable. Features with the highest importance scores are highlighted, offering insights into which variables contribute most significantly to model predictions.

Outliers

The algorithm identifies outliers using the Z-score method and analyzes how these outliers influence feature correlations. Significant shifts in correlation values in the presence of outliers indicate important relationships, highlighting features where outliers have a substantial impact.

Target Variable Analysis

Analyzing the target variable involves detecting outliers and assessing its distribution against known probability

distributions[2]. The Kolmogorov-Smirnov test is used to determine the best-fitting distribution, providing valuable insights into the statistical properties and behavior of the target variable.

The Relation Detection Algorithm uses statistical methods to identify linear and non-linear relationships, assess categorical effects, analyze temporal patterns, and evaluate feature importance and outliers. The insights gained from these analyses serve as the foundation for the next step: leveraging these insights within the **Recommendation System** to provide tailored visualization suggestions based on user needs and dataset characteristics.

2.2 Recommendation system

In addition to automatically extracting meaningful information, our solution includes a recommendation system designed to enhance user satisfaction. By employing techniques such as content filtering, the system customizes visualization suggestions to match each user's specific needs, preferences, and analytical goals.

2.2.1 User-Based Collaborative Filtering

User-Based Collaborative Filtering (UBCF) is a method used to predict a user's rating for a visualization based on the ratings of similar users. The similarity between users u and v is calculated using cosine similarity[1]:

$$Similarity(u, v) = cos(\Theta) = \frac{\sum_{i \in items}(r_{ui} \cdot r_{vi})}{\sqrt{\sum_{i \in items} r_{ui}^2} \sqrt{\sum_{i \in items} r_{vi}^2}}$$

Once similarities are established, the predicted rating for visualization p by user u is computed using the formula:

$$r_{up} = \bar{r}_u + \frac{\sum_{i \in users} Similarity(u, i) \cdot (r_{ip} - \bar{r}_u)}{\sum_{i \in users} |Similarity(u, i)|}$$

where $\bar{r}_u$ represents the average rating given by user u. This approach enables the system to suggest visualizations that align with the preferences of users with similar tastes.

2.2.2 Item-Based Collaborative Filtering

Similar to User-Based Collaborative Filtering (UBCF), Item-Based Collaborative Filtering (IBCF) predicts a user's rating by analyzing similarities between items based on ratings provided by other users. The similarity between items i and j is calculated using the cosine similarity formula[1]:

$$Similarity(i, j) = cos(\Theta) = \frac{\sum_{u \in users}(r_u i \cdot r_u j)}{\sqrt{\sum_{u \in users} r_{ui}^2} \sqrt{\sum_{u \in users} r_{uj}^2}}$$

The predicted rating for visualization p by user u is then computed using:

$$r_{ui} = \bar{r}_i + \frac{\sum_{j \in rated\_items} Similarity(i, j) \cdot (r_{uj} - \bar{r}_i)}{\sum_{j \in rated\_items} |Similarity(i, j)|}$$

Final Recommendation

After extracting meaningful relationships and predicting user ratings for all relation types, our solution selects the most suitable visualization based on the available data. To achieve this, a score is calculated for each visualization by combining the predicted ratings from User-Based Collaborative Filtering (UBCF) and Item-Based Collaborative Filtering (IBCF) with the automatically assigned importance of each relation. Relations are ranked on a scale from 1 (least likely to be meaningful) to 5 (most likely to be meaningful) based on factors such as p-values or correlation strengths. The visualization with the highest final score is presented to the user, who can then provide feedback to refine future predictions and recommendations.

To conclude, the Recommendation System uses User-Based and Item-Based Collaborative Filtering to predict personalized ratings for visualizations. By combining these predictions with the significance of detected relationships, it generates a final score for each visualization. The highest-scoring visualization is then brought to life using the **Plot Generator**, ensuring that the presented visualizations are both statistically relevant and tailored to user preferences.

2.3 Plot Generator

The Plot Generator is a crucial component of our solution, responsible for converting the statistical insights derived from the **Relation Detection Algorithm** and the ones chosen by the **Recommendation System** into clear and insightful visualizations. Utilizing advanced plotting libraries such as Matplotlib and Seaborn, it offers a wide array of visualization techniques to effectively present different types of relationships within the dataset. The Plot Generator produces the following types of visualizations:

Scatter and Regression Plots:

Highlight linear and non-linear correlations between numerical features.

Box Plots:

Illustrate the impact of categorical variables on numerical target variables, providing insights into categorical effects.

Heatmaps:

Display the results of chi-squared tests, showcasing dependencies between categorical features.

Line and Stacked Bar Charts:

Reveal trends over time in both numerical and categorical features.

Outlier Visualizations:

Identify and highlight the influence of outliers on feature relationships.

Feature Importance Bar Charts:

Demonstrate the most significant features in predicting the target variable using Random Forest analysis.

Distribution Analysis:

Analyze the target variable's distribution, including outlier ratios and best-fitting probability distributions.

In designing the Plot Generator, we referred to guidance from an insightful article on Medium called Choosing the Right Graph [8]. This article helped us refine our approach in selecting the most appropriate visualization techniques for different types of data relationships.

By creating tailored visual representations of detected relations, the Plot Generator not only aids in data analysis but also enhances the interpretability and communication of complex statistical insights.

3 Experimental Evaluation

We evaluated the effectiveness of our solution on different real datasets. The evaluation of the performance of our solution is against the naive solution of randomly chosen visualizations on randomly selected features.

3.1 System Implementation

Our system is implemented as a desktop application using the **Tkinter** library, providing a user-friendly interface for testing and evaluating the visualization recommendation system. This is important to test one of our goals of assisting non-technical users. The system consists of three main components: the **Relation Detection Algorithm**, the **Recommendation System**, and the **Plot Generator**. Each component plays a distinct role in the process of analyzing data, recommending visualizations, and displaying the results to the user.

System Workflow: The system operates through a structured workflow that ensures seamless integration of its components:

1. **Relation Detection Algorithm:** The system first executes the Relation Detection Algorithm, which analyzes the dataset to identify meaningful relationships between features.

2. **Recommendation System:** Once the relations are identified, the Recommendation System evaluates them using different methods. It calculates personalized scores for each potential visualization by combining predicted user preferences with the automatically ranked significance of each relationship. The system then selects the top-scoring visualizations for display.

3. **Plot Generator:** After selecting the recommended visualizations, the Plot Generator creates the actual visual representations using **Matplotlib** and **Seaborn**. It generates various plot types, including scatter plots, box plots, heatmaps, and line charts, based on the detected relationships and recommendation scores.

4. **User Interaction:** The application presents each plot to the user through the Tkinter interface. Users are prompted to rate the visualizations and provide comments, which are recorded alongside metrics such as time taken for evaluation. This feedback is essential for refining the recommendation algorithm and enhancing future suggestions.

5. **Data Collection and Analysis:** The system saves all user interactions, ratings, and comments to a text file for later analysis. These results will be used to measure the system's performance using metrics

such as accuracy of recommendations, user satisfaction, and efficiency.

3.2 Technical Details

The system is built with Python, leveraging the following libraries and tools:

- **Tkinter:** For building the graphical user interface (GUI) and managing user input.
- **Pandas and Numpy:** For data manipulation, processing, and statistical analysis.
- **Matplotlib and Seaborn:** For generating visualizations in the Plot Generator.
- **PIL (Python Imaging Library):** For handling and displaying plot images within the application.
- **Time and Random:** For managing the display timing of plots and randomizing the order of visualizations.

The application creates a controlled experimental environment by displaying a mix of system-recommended and randomly selected visualizations to the user. It tracks both the user's ratings and the time taken to understand each plot, providing valuable insights into the user's engagement and comprehension of the visualizations.

This comprehensive approach allows for a robust assessment of the system's ability to automatically detect meaningful relationships, recommend the most relevant visualizations, and present them in a clear and engaging manner.

3.3 Evaluation Metrics

The performance of our system is assessed using the following key metrics:

- **User Feedback:** Collects user ratings on clarity, usefulness, and insightfulness, along with qualitative feedback on what aspects of the visualization worked well and potential areas for improvement.
- **Time to Interpretation:** Measures the time required by users to accurately interpret each visualization, reflecting the clarity and intuitiveness of the visualization.

3.4 Experimental Setup

We conducted our experiments using the movie dataset[7] previously utilized in our other projects. This dataset provided a diverse set of features and relationships, allowing us to thoroughly test the system's ability to detect patterns and recommend appropriate visualizations. The evaluation was carried out under controlled conditions, where users interacted with the visualizations and provided feedback through a structured questionnaire. The collected data was then analyzed to assess the effectiveness of the recommendation algorithm and the quality of the generated visualizations.

3.5 Experiment Results

An experiment was conducted with 11 participants, all computer science students, to evaluate the performance of our developed system. Each participant was presented with 15 visualizations: 10 generated by our solution using intelligent feature and relation selection, and 5 randomly generated visualizations with arbitrary features and relations. The following sections provides a detailed analysis of the collected feedback, highlighting the system's effectiveness and comparing the results with the randomly generated visualizations.

3.5.1 Average Ratings of Plot Types

The average ratings for the two plot types, System and Random, are presented in Table 1. The results indicate a clear preference for the plots generated by our system, with an average rating of 3.65 compared to 1.8 for randomly generated plots. This demonstrates the effectiveness of our system's intelligent selection of features and relations, leading to more insightful and valuable visualizations for the users.

Table 1. Average Ratings for System vs. Random Plots

Plot Type	Average Rating
System	3.65
Random	1.80

The significant difference in average ratings suggests that the visualizations generated by our system were consistently perceived as more effective and informative by the participants. This finding supports the hypothesis that leveraging intelligent feature and relation selection can enhance the quality of data visualizations.

3.5.2 Statistical Significance of Rating Differences

To determine whether the difference in average ratings between System-generated and Random visualizations is statistically significant, a two-sample t-test was performed. The t-test is a statistical method used to compare the means

of two groups and assess whether any observed differences are unlikely to have occurred by random chance.

The results of the t-test showed a t-statistic of 9.0941 with a p-value of 0.0000. Since the p-value is well below the commonly used significance threshold of 0.05, we reject the null hypothesis that there is no difference between the ratings of the two plot types. This result provides strong evidence that the higher average rating for System-generated visualizations is statistically significant, highlighting the effectiveness of our intelligent visualization generation approach.

3.5.3 *Analyzing the Difference Between System and Random Visualizations*

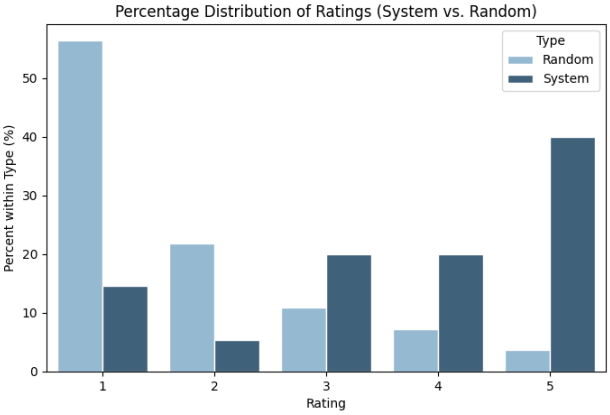

Figure 1. Distribution of Ratings for System and Random Visualizations

Figure 1 displays the distribution of ratings as a percentage of the total ratings for both randomly generated (Random) and system-generated (System) visualizations. The chart highlights a stark contrast in user preferences: over 80% of the Random visualizations received a rating of 3 or lower, whereas more than 80% of the System visualizations achieved a rating of 3 or higher. This clear difference underscores the effectiveness of our system in generating valuable and insightful visualizations.

Figure 2 illustrates the average time participants spent evaluating each plot before submitting their rating. The plot numbers reflect the sequence in which the visualizations were presented (e.g., 1 for the first plot, 5 for the fifth, etc.). As expected, the initial plots required more time as users became acquainted with the system and its features.

A clear trend of decreasing engagement time emerged over the course of the experiment. Around plot 7, a temporary increase in viewing time suggested a sign of user

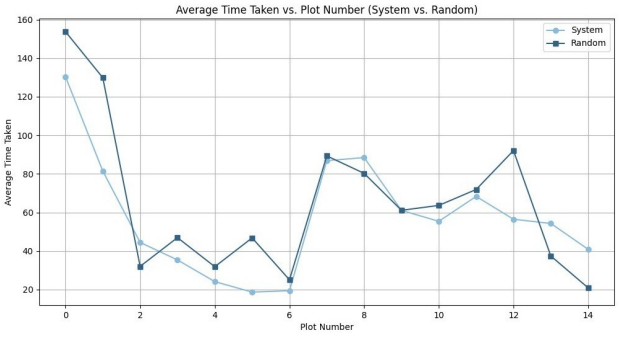

Figure 2. Average Time Spent vs. Plot Order for System and Random Visualizations

fatigue, followed by a more gradual and variable decline in time spent per plot.

Comparing System and Random visualizations, it is evident that participants generally required less time to understand and rate System-generated plots. This difference suggests that while users initially attempted to interpret the Random visualizations thoroughly, many eventually lost interest, leading to quicker but likely less thoughtful ratings. This pattern reinforces the idea that our System plots were not only more effective but also more intuitive for users.

3.5.4 *Analysis and Summary of User Comments*

For each visualization, users provided both ratings and qualitative comments describing their interpretations and insights. A clear distinction emerged between the feedback for System-generated and Random visualizations, as well as across different plot indices.

Plots 0–4: *System:* Users valued these visualizations for effectively highlighting feature importance and correlations. In particular, the insights regarding budget and vote count as predictors of revenue were well received, consistent with the SHAP analysis presented in previews parts of the project. One participant remarked, “You can clearly see that vote count and budget are really important and the others less…”. *Random:* In contrast, these visualizations were frequently characterized as confusing, uninformative, and misaligned with the underlying data, with several users expressing difficulty in discerning the relationships presented. As one comment noted, “this is not informative by any means”.

Plots 5–9: *System:* Feedback for this set was generally positive, with several users indicating that these plots revealed interesting relationships between features and noting similarities to earlier, well-received visualiza-

tions. This observation is corroborated by the decrease in evaluation time shown in Figure 2, and it further underscores the recommendation system's effectiveness in presenting follow-up plots with similar, high-quality characteristics. *Random:* These visualizations were uniformly criticized for displaying unrelated features and lacking coherent relationships, leading to an overall perception of uninformative content.

Plots 10–14: *System:* Comments in this range were mixed; some users found the visualizations interesting and meaningful, while others pointed out repetitiveness. *Random:* Feedback was predominantly negative, with remarks such as "what is this?!" and "cannot understand anything." Although a minority of users identified potential correlations, the overall criticism suggests a decline in engagement, possibly due to survey fatigue or the diminishing availability of informative relations in later visualizations.

In summary, System-generated visualizations consistently received more favorable evaluations compared to Random visualizations, with users finding them significantly more effective in conveying key feature relationships and insights. This proves that our solution improves user satisfaction dramatically.

3.5.5 Analysis of Ratings by Relation Type for System Plots

Figure 3 displays the distribution of ratings for different relation types in System-generated visualizations. The box plot provides insights into the central tendency, variability, and outliers for each relation type, offering a deeper understanding of which visualization characteristics were most effective.

The plot reveals that the *target_correlation* and *target_analysis* relation types received consistently high ratings, with median scores near 5 and minimal variability. These relation types also showed few to no outliers, indicating strong agreement among participants regarding their effectiveness.

On the other hand, relation types such as *categorical_effect* and *outlier_pattern* displayed a wider spread of ratings, including several lower scores. This suggests that while some participants found these visualizations useful, others did not perceive the same level of value. The significant spread in ratings for *high_correlation* visualizations further supports this observation.

Notably, the *date_numerical_trend* relation type consistently received the lowest ratings, with a median score below 2 and several outliers at the minimum value. This result indicates that visualizations of this type did not resonate well with participants, potentially due to difficulty in interpreting the insights or lack of perceived relevance.

Overall, this analysis demonstrates that certain relation types, particularly those focused on target variable analysis, align well with user e xpectations and needs. It also highlights opportunities for improving and refining visualizations with broader rating distributions to enhance overall user satisfaction.

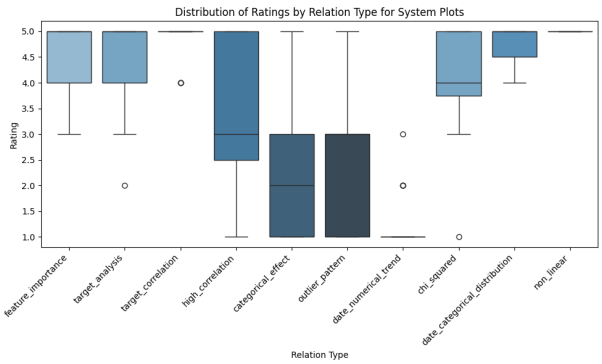

Figure 3. Distribution of Ratings by Relation Type for System Plots

3.5.6 Effectiveness of the Recommendation System

While the quantitative results do not directly measure the participants' satisfaction with the recommendation system, qualitative insights from their comments and the sequence of visualizations displayed provide strong evidence of its effectiveness. The recommendation system aimed to suggest new visualizations with similar characteristics to those previously well-received by the user.

The analysis of visualization sequences revealed a clear pattern: when participants highly rated a visualization, the subsequent visualization often shared similar features and relations. This pattern aligns with the intended behavior of the recommendation system, suggesting that it successfully adapted to user preferences. Additionally, comments from participants often reflected a sense of continuity and relevance, further supporting the system's ability to generate meaningful and tailored visualizations.

These observations, while not quantified, demonstrate the potential of our recommendation approach to enhance the user experience by delivering increasingly relevant visualizations based on individual preferences.

3.6 Evaluation on Additional Datasets

To assess the generalizability of our system, we applied it to additional datasets beyond the primary movie dataset used in our controlled experiment. The results were highly encouraging—our system successfully identified meaningful relationships within the data and generated insightful visualizations that effectively highlighted key patterns. The visualizations accurately captured significant feature interactions, demonstrating the robustness and adaptability of our approach.

Although we did not conduct a formal user study on these alternative datasets, we are confident that the system would yield similarly positive results. Given its consistent performance in identifying and visualizing important relationships, we expect that users would find the visualizations equally informative and beneficial. Future research could extend this evaluation by conducting user studies on a diverse set of datasets to further validate the system's effectiveness across different domains.

One critical consideration when applying the system to new datasets is the computational complexity introduced by larger feature spaces. Datasets with a higher number of features and complex relationships require additional processing time, particularly during the Relation Detection Algorithm phase. As the number of feature pairs and potential interactions increases, the system must perform a more extensive analysis to accurately identify the most relevant patterns. Consequently, while larger and more complex datasets may introduce longer processing times, this trade-off ensures that the system provides a comprehensive and meaningful visualization of the data.

These findings reinforce the system's potential as a versatile tool for automated data visualization, capable of adapting to a wide range of datasets while maintaining its ability to extract and present significant insights.

4 Related Work

Numerous studies have explored the automatic generation of data visualizations. Many existing solutions, such as *Show Me* [3], focus on enhancing the user experience by suggesting visualizations based on selected data and offering features to add fields to a view or generate new views from multiple fields.

SeeDB [4] takes a different approach by automatically analyzing and recommending the most "interesting" visualizations for a given query. This system evaluates all potential two-column visualizations by adding attribute aggregates and group-by clauses to the query, then eliminates less meaningful views through a pruning process.

Other tools, such as *QuickInsights* [5], prioritize speed in data mining and visualization generation. Their primary goal is to rapidly and automatically extract insights from multi-dimensional data, streamlining the analysis process for users.

A more recent development, *LIDA* [6], utilizes large language models (LLMs) and image generation models (IGMs) to generate visualizations and infographics automatically. This system employs a multi-stage approach:

- **Summarizer** – Transforms raw data into a natural language summary.
- **Goal Explorer** – Identifies visualization goals relevant to the data.
- **VisGenerator** – Produces and refines visualization code.
- **Infographer** – Creates graphics using image generation models.

In contrast to these approaches, our solution uniquely automates the extraction and evaluation of significant feature relationships within an entire dataset. It allows users to rate the generated visualizations, which not only enhances user satisfaction but also maximizes information gain compared to randomly generated visualizations. This adaptive feedback mechanism ensures that our system delivers increasingly relevant and insightful visualizations based on user preferences.

5 Conclusion

Our project demonstrated the potential of automating the data visualization process to enhance the efficiency and quality of data analysis. Through the integration of a Relation Detection Algorithm, a Recommendation System, and a Plot Generator, our solution effectively identified significant relationships within datasets and provided tailored visualization suggestions to users.

The experimental results showed a clear advantage of our system-generated visualizations over randomly generated ones, with higher average ratings, faster interpretation times, and more insightful data presentations. The user feedback and the statistical analysis confirmed the effectiveness of our approach, particularly in handling highly correlated features and adapting to user preferences.

Throughout this project, we learned the importance of combining statistical analysis with user-centric design. We

gained valuable insights into how automated systems can assist non-technical users in understanding complex data, and how adaptive recommendation systems can continuously improve the user experience. The project's success not only highlights the feasibility of automated visualization tools but also opens up possibilities for future enhancements, such as incorporating more advanced machine learning techniques for personalized visualization recommendations.

Our solution's holistic approach to data visualization can serve as a valuable tool for data scientists and business professionals alike, providing clear and relevant insights with minimal manual effort. Moving forward, further research could explore extending the system's capabilities to support more diverse datasets and visualization types, as well as enhancing the feedback loop for even more precise and adaptive recommendations.

6 Code and Data

The code for the automation process, the generated visualizations, and the dataset used in this project are available in our Git repository under the **Final Project** directory. You can access them using the following link.

References

[1] Dr. A. Somech. *Statistical Correlation Measures in Tabular Data Science*. Tabular Data Science, pages 25–117, Bar-Ilan University, Fall 2022.

[2] Dr. A. Somech. *Exploratory Data Analysis: Distributions*. Tabular Data Science, pages 2–24, Bar-Ilan University, Fall 2022.

[3] J. Mackinlay, P. Hanrahan, and C. Stolt. *Show Me: Automatic Presentation for Visual Analysis*. IEEE Transactions on Visualization and Computer Graphics, vol. 13, no. 6, December 2007.

[4] M. Vartak, S. Madden, A. Parameswaran, and N. Polyzotis. *SEEDB: Automatically Generating Query Visualizations*. Proceedings of the VLDB Endowment, 7(13), pages 1581–1584, August 2014.

[5] R. Ding, S. Han, Y. Xu, H. Zhang, and D. Zhang. *QuickInsights: Quick and Automatic Discovery of Insights from Multi-Dimensional Data*. Microsoft Research, June 25, 2019.

[6] V. Dibia. *LIDA: A Tool for Automatic Generation of Grammar-Agnostic Visualizations and Infographics using Large Language Models*. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations), pages 113–126, Toronto, Canada. Association for Computational Linguistics, July 2023.

[7] R. Banik. *The Movies Dataset*. Kaggle, July 2017.

[8] Ajinkya Dandgavhal. *Choosing the Right Graph*. Medium, June 2023.