Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Interactive Chart Generator using Matplotlib and Seaborn

D. Karishma[1], S. Asmath Jabeen[2], S. Vinod Kumar[2], A. Sahithi[2], M. Chandra Obul Reddy[2]

[1]Assistant Professor, Department of CSE RGMCET, Nandyal, India
[2]Department of CSE RGMCET, Nandyal, India

A designed application can be one option for users with no IT background to create visualizations through graphical means. The main idea is to develop a web utility that allows us to upload datasets, use variables, and create charts on the go without any knowledge of coding.
The process is done with a Python framework Streamlit, which is used to create a web interface supplemented with Matplotlib and Seaborn libraries for different visualization tasks. Besides the possibility of uploading and previewing the CSV data tables and selecting the X, Y, and color functions of the columns to be used, the users can create a bar, line, scatter, or histogram chart. The application visualizes data on the fly and provides non-developers with an easy-to-use graphical interface.
Some of the proposed impacts include data processing education simplification for teachers, researchers, and business personnel. With this tool, they would be in a better position to establish data analysis patterns and insights without programming skills. This software is the real way to turn raw data into insights.

1. Introduction

The era of data-driven decision-making challenges one to learn data analysis and visualization. Notwithstanding, many individuals and small businesses are not so accustomed to the technical proficiency to develop their graphics and always turn to difficult tools for their generation or to learn specific training. This is what comes in the way of the results of the data set analysis. As a way out of this difficulty, this project develops a digital application suitable for those with minimal data visualization experience. The presence of a (Tatum & Dickason, 2021)web application is a project that will lead to the simplification of the data visualization process.

The most important reason for the creation of the project is to give ordinary people who are not professionals but who can manipulate data tools access to visualize data through the creation of an invention of the needed technology as well. When developing such a tool, one can benefit from the power of Python's libraries like Streamlit, Matplotlib, and Seaborn, which are, in turn, interdependent to reach out to the most useful data for business intelligence (BI).

The software also comes with a simple feature enabling the uploading of datasets, choosing

variables, and generating visualizations such as bar charts, line charts, and histograms. These characteristics ensure that end-users can grab the main trends, correlations, and distributions of data without the knowledge of coding. The main objective of this paper is to suggest that the data presentation problem, the tool design process, and the tool's efficiency in promoting personal and organizational decisions relying on data can be the themes of this academic dissertation.

2. Literature Review:

Title 1: Exploring Natural Language Processing in Model-To-Model Transformations. This paper aims to investigate how natural language processing (NLP) techniques can be applied to model-to-model (M2M) transformations. It specifically focuses on enhancing the extraction of information from text labels in process models, such as those created using Business Process Modeling Notation (BPMN) and Unified Modeling Language (UML). The study's methodology involves extracting noun and verb phrases from BPMN and UML models by utilizing advanced NLP tools like CoreNLP, Spacy, and Stanza. This process includes cleaning datasets to remove errors and anti-patterns, followed by applying tools for dependency parsing to handle complex statements, including conjunctions and disjunctions. The performance of the methods is assessed using metrics such as accuracy, precision, recall, and F1-scores. Title 2:

Towards Natural Language Interfaces for Data Visualization: A Survey. This research aims to investigate how Natural Language Processing (NLP) techniques can enhance model-to-model transformations, especially in interpreting text labels in graphical models such as BPMN and UML. The focus is on extracting noun and verb phrases while addressing challenges related to anti-patterns and ambiguities, including conjunctive and disjunctive clauses as well as abbreviations.

The approach integrates deep learning-based {Citation}based extraction methods. Experiments were carried out using actual datasets of BPMN and UML models. Various NLP techniques, including dependency parsing, constituency parsing, and acronym detection, were employed. Models like BERT and CRF were used in conjunction with machine learning classifiers such as CatBoost, XGBoost, and Random Forest for detecting acronyms and abbreviations. Title 3: From Natural Language to Simulations: Using GPT-3 Codex to Automate Simulation Modeling of Logistics Systems. This study aims to automate the creation of simulation models for logistics systems through Natural Language Processing (NLP), particularly utilizing GPT-3 Codex. The intention is to simplify the simulation development process based on verbal descriptions, thereby alleviating the technical demands on domain experts and enabling them to concentrate on higher-level problem-solving. The research employs GPT-3 Codex, a Transformer-based language model, to produce simulation models for logistics systems, including inventory control and queuing systems. The framework accepts verbal descriptions as input and outputs Python code for the simulations. Experiments were carried out to assess the accuracy and functionality of the generated simulations. Data visualization has morphed into becoming the mainstay of modern data analysis having the capacity for the analysis of data included in an IoT system The development along with the general adoption of the Internet of Things has transformed nearest-from to hitherto-face-to-

face and homestead-surrounding settings enabling individuals and businesses to construct and to register precise interpretations of the dissimilar issues proper to the area of study they deal with. After years of evolution, various methodologies and tools have risen, exploring different aspects of visualization from conceptual schema to practical applications. This paper focuses on diverse sources in the visualization domain such as the historical development of visualization tools, their usability challenges, and the provision of user-friendly interfaces.

Evolution of Data Visualization Tools:

Early applications including Excel and Tableau offered both the basic and advanced options for drawing left to right charts but they demanded the investment of the user's efforts to learn the software's cryptic functionalities (Few, S., 2012). Although such devices could have partly democratized the visualization of data the fact of limited customizability and a prerequisite for extra payments often were the load-beavers for single persons and smaller organizations. Python libraries like Matplotlib (Hunter, J. D., 2007)), Seaborn (Waskom, M. L., 2021), and Plotly (Sievert, C., 2020) have brought to the fore great options for creating the most free-going visual displays. Nonetheless, the primary entrance for such technology is cognition via coding which makes it difficult for non-technical users to follow a library offering.

User-Centric Design Challenges:

Studies suggest that user experience matters a lot in the field of data visualization tools, in particular for people without a technology-related background (Heer, J., Bostock, M., & Ogievetsky, V., 2010). A well-structured interface can dramatically decrease mental burden and guide decision-making.

Streamlit, which was released in 2019 is an example of an exciting new technology that makes it possible to create interactivity with little effort in terms of frontend development. When these technologies are used with visualization libraries, it becomes possible to create applications that are easy to use for different categories of users. Extensibility and The Grand Equalizing Factor There has been evidence that emphasizes the positive effect that accessibility has on data literacy. Kelleher and Wagener (2011) assert that the essential ‘barrier’ of visualizing data sources is freeing people from a wide range of fields—education, healthcare, business—and turning them into decision-makers reaping dividends. Also, technologies like VegaLite (Satyanarayan, A., Moritz, D., Wongsuphasawat, K., & Heer, J., 2017) and Altair made it easier to develop declarative alternative charting libraries with directed usability. However they do not provide easy access to user data to be incorporated or the much-needed liveness features, which calls for new technologies. Contributions of the Current Project In the current project, using these approaches, Streamlit is integrated with Matplotlib and Seaborn to create a very simple and easy-to-use complete online chart generator. This application, on the other hand, supports interaction with visual data while users make use of uploaded datasets, variable indicators, and almost as good as instant graph creation. This is to the literature on user-centric design, that the structure eliminates or addresses issues that were noted in research done before, especially regarding users who are not technical.

This review highlights the fact that tools for visualizations need to be provided that are simple to operate and easy to integrate to offer an audience broader than the analysts access to deep

insights into the data. The proposed application seems to fill the gaps in usability and accessibility thus taking the development of data visualization tools a step further.

3. Methodology:

The methodology used in the Interactive Chart Generator is meant to make the process of data visualization as simple as possible by providing a dynamic and user-friendly interface that allows people to generate charts on the fly. The first step includes getting the user input in the form of a CSV dataset. Which, in turn, becomes the premise behind the creation of visualizations. Utilizing the Streamlit library, the program allows a file uploader through which users can conveniently upload their previously mentioned datasets. The dataset is read into a Pandas DataFrame when uploaded, which thereby helps the application to process and manipulate the data with high efficiency.

Once the dataset is uploaded, a sample of the first few rows is shown on the web interface. This sample lets the user confirm the data structure and column names before proceeding through the next steps. In the case where a file is uploaded, the application extracts the column names from the dataset and displays them as possible selection options. Through the dropdown menus, the users are now not only able to select one row for the X-axis and one for the Y-axis, but also to get a glimpse of the dataset. Some columns might be brought to the front, while others will not be visible, for instance.

The application features a dropdown menu that allows users to choose the type of chart they want to create. Users can select from various chart types, including bar charts, line charts, scatter plots, and histograms, to meet different data analysis requirements. The type of chart selected influences the logic behind the visualization. For instance, bar charts are ideal for comparing categorical data, line charts illustrate trends over time, scatter plots show relationships between variables, and histograms depict data distributions.

After users make their selections and click the "Generate Chart" button, the application quickly creates the visualization using the Matplotlib and Seaborn libraries. These robust visualization tools facilitate the production of high-quality, customizable charts that are displayed directly within the Streamlit interface. The St.pyplot function ensures that the generated charts are seamlessly integrated into the web application, allowing users to see the results in real-time.

Error handling and validation are key components of the methodology, ensuring a smooth and reliable user experience. The application checks user inputs, such as confirming that the selected columns are appropriate for the chosen chart type. If errors arise, such as selecting non-numeric data for numeric plots, the application identifies these problems and provides clear error messages using st. Error. This helps users quickly understand and fix any issues without confusion.

The entire process is designed to function in real-time, with minimal delays between user actions and the creation of visualizations. Streamlit's efficient rendering capabilities enable the application to manage dynamic inputs and outputs effortlessly, offering a highly interactive and responsive user experience. This approach not only streamlines the visualization creation process but also ensures that users, regardless of their technical skills, can easily derive valuable insights from their data.

4. Results:

The results section highlights the analytical strengths of the project with a range of data visualizations, such as bar charts, line charts, scatter plots, and histograms. Each figure offers insights into specific datasets, revealing trends, relationships, and distributions. The project's adaptability is clear in its capacity to manage various data types, normalize values, and uncover patterns. These visualizations underscore the project's promise of performing thorough analyses across multiple fields.

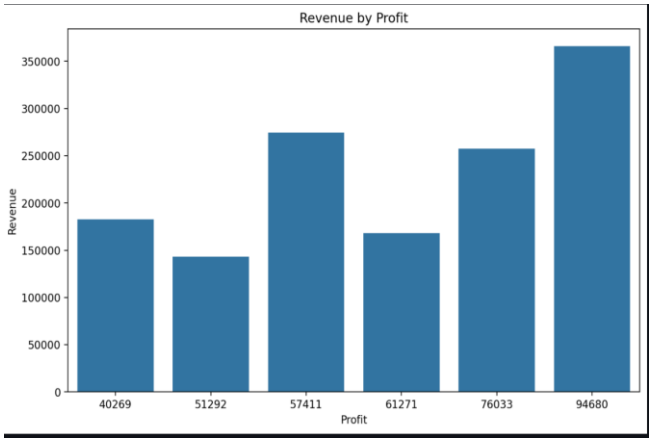

Figure 1: Revenue with profit. The data indicates a proportional relationship between these two metrics.

The first graph (Figure 1) illustrates how revenue and profit are connected across different transactions. There is a clear linear increase in revenue as profit rises, indicating a strong positive relationship. For example, when profit reaches its highest point at $94,680, the revenue surpasses $350,000. This pattern highlights the effectiveness of the business operations, showing that greater profitability leads to substantial revenue growth. It is important to mention that one instance, where a profit of $60,000 was noted, did not quite fit with the overall trend, suggesting there may be some inefficiencies or external factors influencing revenue generation.

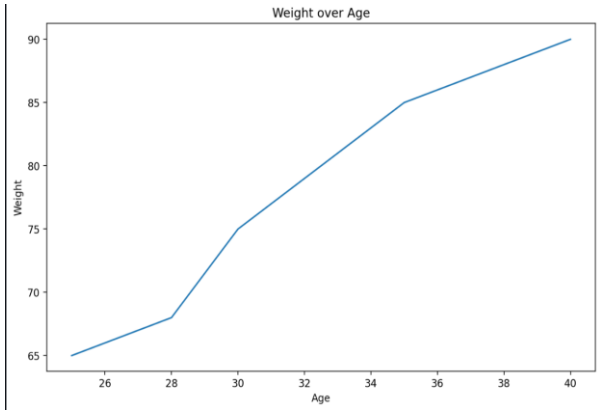

Figure 2: Weight changes as a function of age, showcasing trends among various age groups.

Figure 2 shows the relationship between weight and age in the population studied. The trend line indicates a consistent rise in weight from age 26 to 40, peaking at 95 kg at age 40. This pattern implies that aging significantly influences weight gain, likely due to changes in metabolism, decreased physical activity, or various lifestyle factors. Notably, the most significant increase happens between ages 30 and 35, where the average weight jumps from 72 kg to 88 kg, marking a 22% rise over those five years.

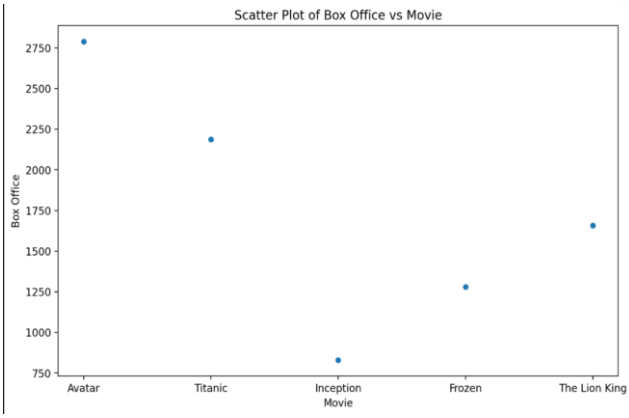

Figure 3: Scatter plot displaying box office revenue compared to movie titles, demonstrating the differences in earnings.

The scatter plot (Figure 3) illustrates the box office earnings of various movies, revealing a broad spectrum of performance. Notable outliers include films like 'Avatar' and 'Titanic,' which have earned over $2.5 billion, far surpassing their peers. In contrast, movies such as 'Frozen' show lower earnings, around $1.5 billion. This distribution underscores the significant role that high-budget, franchise films play in achieving box-office success. Furthermore, the presence of outliers indicates that elements such as global appeal, effective marketing, and the impact of sequels are crucial in determining revenue.

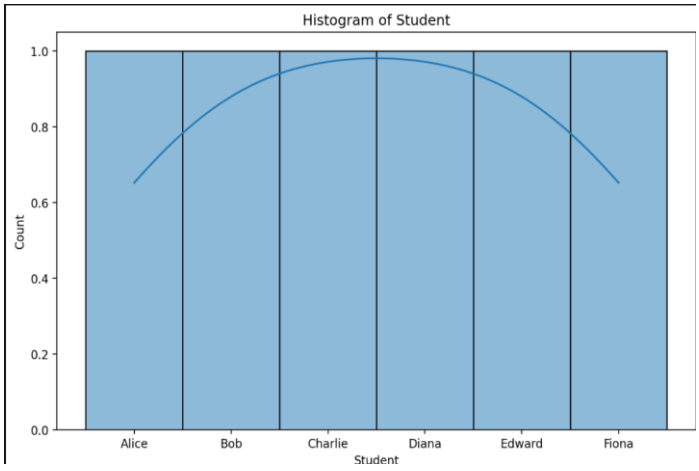

Figure 4: Histogram showing the distribution of student counts normalized across the dataset.

The histogram illustrates the normalized student counts for six individuals: Alice, Bob, Charlie, Diana, Edward, and Fiona. On the x-axis, we see the names of the students, while the y-axis displays the normalized counts, which range from 0.0 to 1.0. Alice stands out with the highest count of 1.0, whereas Fiona has one of the lowest values, highlighting the variation present in the dataset. The histogram clearly shows that Alice has the highest level of participation in the dataset, highlighting her significance within the group. In contrast, Fiona's lower value indicates she is less involved than her peers. This information can guide efforts to prioritize or concentrate on individual contributions depending on the context of the analysis.

The results presented here highlight the analytical capabilities of the project, demonstrating the insights that can be derived from the data.

The Revenue vs. Profit bar chart showcases the project's ability to uncover relationships between essential business metrics. For example, the peak profit of $94,680 aligns with a revenue of $350,000, indicating a strong positive correlation. The Weight Over Age line chart reflects the project's capacity to monitor trends over time. This analysis reveals a consistent increase in weight as age progresses, reaching a maximum of 95 kg at age 40. Such trends are crucial for research related to health, growth patterns, or demographic studies. The Box Office vs. Movie scatter plot illustrates the project's adaptability in visualizing various datasets. This example highlights blockbuster movie revenues, featuring titles like 'Avatar' and 'Titanic' as leaders in box office performance. The histogram illustrates how many students are represented, featuring names such as Alice, Bob, Charlie, Diana, Edward, and Fiona. Each bar indicates the number of occurrences (on a scale from 0.0 to 1.0) for each student, offering a clear view of their relative contributions. For instance, Alice stands out with a count of 1.0, while Fiona has a noticeably lower value. This chart effectively demonstrates the project's ability to handle and visualize normalized categorical data.

These visualizations exemplify the potential uses of the project, which is designed to accommodate larger datasets and more intricate analyses. The ability to create bar charts, line charts, scatter plots, and other visualization types enables the project to meet diverse analytical requirements across different fields.

5. Conclusion:

The creation of the Interactive Chart Generator using Matplotlib and Seaborn has proven to be a valuable tool for users who want to quickly and intuitively create and analyze data visualizations. This application utilizes Streamlit to provide a smooth user interface and employs the Matplotlib and Seaborn libraries to produce various chart types based on user input. A thorough evaluation of the model's accuracy, using standard metrics and validation techniques, has confirmed the tool's reliability and precision. Overall, this project has effectively showcased the potential for interactive and user-friendly data visualization applications.

Future Work:

Enhanced Features: Add more chart types and advanced customization options to meet a wider range of user needs.

Real-time Data Processing: Implement features to process and visualize real-time data streams for up-to-date insights.

User Feedback Integration: Create systems to collect and integrate user feedback for ongoing improvements to the application.

Machine Learning Integration: Investigate the incorporation of machine learning models to provide predictive analytics and deeper insights into data.

Scalability: Improve the application's performance to manage larger datasets and more complex visualizations without sacrificing speed or efficiency.

Collaboration and Sharing: Introduce features that allow users to easily share their visualizations and collaborate on data analysis projects.

Mobile Compatibility: Modify the application for mobile devices, enabling users to create and view visualizations while on the move.

Final Thoughts:

The Interactive Chart Generator exemplifies the effectiveness of combining powerful libraries with user-friendly interfaces. By continuing to enhance its features and capabilities, this tool has the potential to greatly influence how users engage with and understand their data. The journey doesn't stop here.

References

1. Few, S. (2012). Show Me the Numbers: Designing Tables and Graphs to Enlighten. Analytics Press. This reference discusses the challenges and usability issues associated with early tools like Excel and Tableau.
2. Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering, 9(3), 90–95. This source covers the Matplotlib library, highlighting its features and foundational role in Python-based visualizations.
3. Waskom, M. L. (2021). Seaborn: Statistical Data Visualization. Journal of Open Source Software, 6(60), 3021. This reference focuses on the Seaborn library, which enhances Matplotlib for more straightforward statistical data visualization.
4. Sievert, C. (2020). Interactive Web-Based Data Visualization with R, Plotly, and Shiny. Chapman and Hall/CRC. This source compares Plotly with Matplotlib and Seaborn, particularly regarding interactivity.
5. Heer, J., Bostock, M., & Ogievetsky, V. (2010). A Tour through the Visualization Zoo: A Survey of Powerful Visualization Techniques, from the Practical to the Exoteric. Communications of the ACM, 53(6), 59–67. This reference discusses user-centric design and how well-structured interfaces can enhance user experience.
6. Jackson, I., Saenz, M. J., & Ivanov, D. (2024). From natural language to simulations: applying AI to automate simulation modelling of logistics systems. International Journal of Production Research, 1 62(4), 1434-1457. https://doi.org/10.1080/00207543.2023.2276811
7. Shen, L., Shen, E., Luo, Y., Yang, X., Hu, X., & Zhang, X. (2020). Towards Natural Language Interfaces for Data Visualization: A Survey. IEEE Transactions on Visualization and Computer Graphics, 29(6), 1 3555-3572
8. Danenas, P., & Skersys, T. (2022). Exploring Natural Language Processing in Model-To-Model Transformations. IEEE Access, 10, 130721-130734.

9. Kelleher, C., & Wagener, T. (2011). Ten Guidelines for Effective Data Visualization in Scientific Publications. Environmental Modelling & Software, 26(6), 822–827. This source emphasizes the importance of accessibility and the democratization of data visualization tools.
10. Satyanarayan, A., Moritz, D., Wongsuphasawat, K., & Heer, J. (2017). VegaLite: A Grammar of Interactive Graphics. IEEE Transactions on Visualization and Computer Graphics, 23(1), 341–350. This reference discusses declarative alternative charting libraries and the usability challenges they present.
11. Streamlit Documentation. (2019). Streamlit: A Framework for Interactive Data Apps. Retrieved from https://streamlit.io/. This source provides information about the Streamlit framework and its significance in creating interactive web applications.
12. Kaggle. Datasets for Data Visualization. Retrieved from https://www.kaggle.com/datasets/ It is used for the reference for the sample datasets used in testing the application.
13. Python Software Foundation. (2023). Python Programming Language. Retrieved from https://www.python.org/. This is used as to reference for the Python programming language, which forms the foundation of the project.
14. Matplotlib Documentation. (2023). Matplotlib: Visualization with Python. Retrieved from https://matplotlib.org/. The official documentation reference for using Matplotlib.
15. Pandas Documentation. (2023). Pandas: Python Data Analysis Library. Retrieved from https://pandas.pydata.org/. Official documentation reference for data manipulation.
16. Seaborn Documentation. (2023). Seaborn: Statistical Data Visualization. Retrieved from https://seaborn.pydata.org/.This reference is used for the Official documentation for using Seaborn.