

DATA VISUALIZATION OF CITY ECONOMY

A Project

Presented to the faculty of the Department of Computer Science

California State University, Sacramento

Submitted in partial satisfaction of
the requirements for the degree of

MASTER OF SCIENCE

in

Computer Science

by

Tejal Rajiv Bijwe

SPRING
2023

© 2023

Tejal Rajiv Bijwe

ALL RIGHTS RESERVED

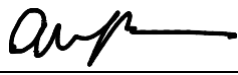
DATA VISUALIZATION OF CITY ECONOMY

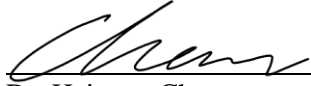
A Project

by

Tejal Rajiv Bijwe

Approved by:


_____, Committee Chair
Dr. Anna Baynes


_____, Second Reader
Dr. Haiquan Chen

4/30/23

Date

Student: Tejal Rajiv Bijwe

I certify that this student has met the requirements for the format contained in the University format manual, and this project is suitable for electronic submission to the library and credit is to be awarded for the project.

_____, Graduate Coordinator 5/1/2023_____

Dr. Haiquan Chen

Date

Department of Computer Science

Abstract
of
DATA VISUALIZATION OF CITY ECONOMY
by
Tejal Rajiv Bijwe

A city's rapid growth comes from multiple factors like understanding demography, the pattern of daily life, and financial health, which directly impact the city's economy. Every city should develop a strategy for managing urban performance, which can be more effective by monitoring all events in the city [1]. Visualizing this data can provide insights into the variables that impact growth.

This project proposes a literal visualization approach that rationalizes big data attributes into smaller blocks of information for visualization, intending to create a smart city dashboard website. The dashboard will analyze and visualize city data, providing insights into where city improvement grants should be invested or not. The development of this dashboard implies a significant investment in the city's human and financial resources from the city [2], including data on participants' behavior, such as the places they visit, their spending habits, and their daily routines. Long-term factors such as overall wages and living expenses also impact citizens.

In this project, I propose a study that predicts the future trend of city revenue, and thus it will help in efficient investment in the city. Also, I am utilizing data analysis techniques to assess the city's financial performance and forecast its economic outlook for the upcoming year by implementing this project in a web context. These results can be easily shared and refined, facilitating unified decision-making for city management and investment purposes.



_____, Committee Chair

Dr. Anna Baynes

4/30/23

Date

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude and appreciation to Dr. Anna Baynes for providing me with invaluable guidance, insides, and support in completing this project and report. Her knowledge and experience encouraged me to understand the objective, design, and process through the project's timeline.

I would like to thank my second reader, Dr. Haiquan Chen, for providing valuable advice and feedback on my project.

I am always thankful to my family and friends for their constant support and inspiration. Finally, I would like to thank the computer science department at California State University, Sacramento, for providing me with academic assistance.

TABLE OF CONTENTS

	Page
Acknowledgements	vii
List of Tables.....	x
List of Figures	xi
Chapter	
1. INTRODUCTION.....	1
1.1 Overview	1
1.2 Objective	2
1.3 Dataset.....	3
1.4 Methodology	4
1.5 Technical Stack	5
1.6 Report Organization	5
2. BACKGROUND.....	7
2.1 Literature Review	7
2.2 Challenges	12
2.2.1 Visualization Obstacles.....	12
2.2.2 Machine Learning Obstacles.....	13
3. SYSTEM ANALYSIS AND ARCHITECTURE	15
3.1 Dataset Summary	15
3.1.1 Attributes.....	15
3.1.2 Journals	16
3.1.3 Activity Logs.....	16

3.2 Data Pre-processing.....	17
3.2.1 Financial Journal and Participants.....	18
3.2.2 Participant Logs	20
4. IMPLEMENTATION	23
4.1 Dependency Management	23
4.1.1 pandas.....	23
4.1.2 NumPy.....	23
4.1.3 Matplotlib	23
4.1.4 Scikit-learn	24
4.1.5 Seaborn.....	24
4.1.6 Plotly	24
4.2 System Blueprint.....	24
4.2.1 Data Preparation.....	24
4.2.2 Model Building	25
4.2.3 Data Visualization.....	25
4.2.4 Dashboard Creation.....	25
4.3 Dashboard Pipeline	25
4.3.1 Data Numerical Statistic.....	25
4.3.2 Correlation Heatmap Of The Data	27
4.3.3 Machine Learning Models	29
4.3.4 Curating Data- Generating Future Data	38
4.3.5 Travel Journal Visualizations.....	52

5. FUTURE WORK	55
6. CONCLUSION	57
References	59

LIST OF TABLES

Tables	Page
1. Technology used in the project.....	5
2. Numerical statistic on dataset.....	26
3. Unique count before.....	32
4. Unique count after.....	33
5. Null count for financial status.	37

LIST OF FIGURES

Figures	Page
1. Financial health of all participants in a quarterly manner.....	3
2. Stages of experiment.....	7
3. 3 Tier architecture.....	8
4. Deep learning models.....	10
5. Relationship among datasets.....	17
6. Drop duplicate before creating a new dataset.	18
7. Timestamp conversion.....	19
8. Education level conversion.....	20
9. Data manipulation over financial journal and participants.....	20
10. ParticipantStatusLog conversion.....	22
11. Heatmap code snippet.....	28
12. Heatmap of the data	29
13. Predicting joviality code snippet.....	31
14. Predicting joviality score.....	31
15. Heatmap to check null values.....	34
16. Predicting financial status code snippet.....,	35
17. Predicting financial status.....	36
18. Unique counts of financial status.....	38
19. Curated data along with the calculated CAGR.....	39
20. Wage vs. Expenses Plot Combined.....	40

21.	Wage vs. Expenses scatter plot (Education Level).....	41
22.	Wage vs. Expenses scatter plot (Education Level) with Lasso Select.....	42
23.	Wage vs. Expenses scatter plot (Age).....	43
24.	Wage vs. Expenses scatter plot (Age) by specific age group.....	44
25.	Wage vs. Expenses scatter plot (Have kids).....	45
26.	Wage vs. Expenses scatter plot (House hold size).....	45
27.	Wage vs. Expenses scatter plot (interest group).....	46
28.	Code for Wage vs. Expense plots on the various categories.....	47
29.	Plot showing the Wages vs. Expenses of original and Curated Data.....	48
30.	Average Joviality (both previous and predicted) of the participants by age group.....	49
31.	Average Joviality (both previous and predicted) of the participants by education Level.....	50
32.	Stable and Unstable counts of participants (Last and This year).....	51
33.	Stable vs. Unstable counts of participants code snippet.....	51
34.	Average travel time by purpose of travel over time of dataset.....	53
35.	Average spends by purpose of travel over time of dataset.....	53

Chapter 1: INTRODUCTION

1.1. Overview

A city metaphor has become a popular method of visualizing the properties of program code [3]. The growth of computer graphics shaped modern visualization. As said, data visualization is visually presenting information with computer software support. Data can be in either static or interactive visualization form. Except for static data visualization, interactive data visualization allows the user to select the format in which data needs to be displayed and analyzed. The survey of the city data visualization needs to operate on scales beyond the normal cognitive scope. It can predict the near future result of economic growth or shrinkage. In this project, I proposed a method to build multiple visual features which are both static and interactive based on the dataset.

Interactive data visualization allows users to operate on a graphical plot to modify the relationship between various data points. It emphasizes graphic representations of data, which improves the connectivity of the information provided. These interactive visualizations are commonly used as interactive dashboards, providing an intuitive and effective way to communicate statistics. To demonstrate the relationship between data and the visual, interactive visualizations require human/bot input, such as clicking a button, moving a slider, and a quick response time [4].

Data Visualizations allow users to interactively explore, modify, and connect with the data [5]. Many visualization tools, such as Power BI and Tableau [5], and frameworks like Python libraries and D3.js [6], created interactive visualizations. As a result, data visualizations extensively communicate data as an appropriate way to convey information.

The participant's life patterns over time will be the basis for establishing the primary analysis and determining whether investing in the improvement grant is valid. The research involves data interaction to explore the dynamic nature of the city, including the daily activity of participants and the city's financial health over time, encompassing the wages and expenses of each participant. The utilization of visualizations in city planning will serve as an effective means to analyze resources and gain insights into city infrastructure, thereby facilitating data-driven decision-making and urban planning strategies. Furthermore, Machine learning models are used to predict the city's financial health.

1.2. Objective

There are many scholarly articles and studies that talk about the city layout in this domain. However, a gap was identified in the literature, as only a limited number of these studies focused primarily on visualization techniques and city policies related to city revenue. The project is centered on the comprehensive analysis of various attributes within the dataset to calculate different aspects of the city.

These aspects include the study of social networks, revenue generated by different types of businesses, monthly income, total employment within the city, job opportunities, wages concerning the overall cost of living, engagement levels, and activity calendar. The dataset also includes the personal information of all 1000 individual participants, such as their living expenses, business turnover, and health and financial status.

The project's final output will be a website or dashboard that presents the results in multiple featured visualizations. For instance, one of the specific analyses conducted on the dataset assesses the financial health of the 1000 participants over time(quarterly). This

analysis will expand further to include various aspects, such as examining wages, calculating each participant's spending ratio, and assessing the impact of these factors on their health, among others.

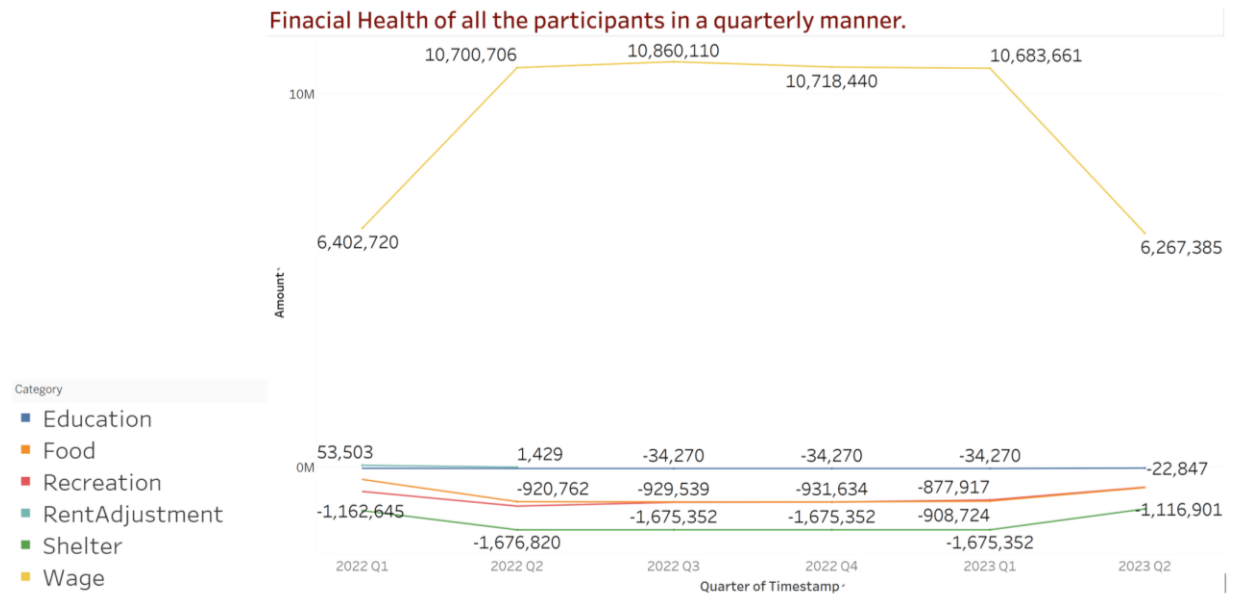


Figure 1: Financial health of all participants in a quarterly manner.

1.3. Dataset

The dataset, presented by the VAST challenge, encompasses comprehensive information on the entire city of Engagement, Ohio, USA. This dataset comprises data from 1000 representatives of modest-sized city residents who have provided inputs through urban planning. It includes activity logs of participants throughout the year and data pertaining to various factors such as participant health, finance, joviality, business types, business turnover, and building data, among others. The datasets are available in CSV or text format. Chapter 3 contains detailed explanations of data preparation and data sources.

1.4. Methodology and Implementation

The workflow approach for this project is as follows:

Step 1: Dataset Analysis and Data Enrichment

- Utilizing pandas plotting techniques to analyze the dataset.
- Performing data cleaning using the NumPy library on the CSV file based on planned, customized features.
- Implementing trifacta for further data manipulation.

Step 2: Visualization of Featured Data on Cleaned Dataset

- After researching the datasets, I analyzed features that provide an overview of city life.
- Designing these features using Python programming language and its libraries.
- Utilizing Label Encoder for data categorization and conversion to numeric.
- Using Matplotlib for interactive visualizations.
- Applying machine learning models, such as Random Forest Regression for numeric analysis and prediction and Random Forest Classifier for finance prediction.

Step 3: Dashboard implementation using featured visualizations.

- Dividing explored features into sections to get a comprehensive overview of city life.
- Planned to integrate and visualize components using the Streamlit framework for a web-based dashboard.

- By analyzing the year's revenue data, I also predicted the city's financial health for the following years using researching various machine learning models such as Regression, LSTM, and Neural Networks.
- After conducting research, I integrated the finalized ML models into the dashboard for data analysis. Predicting the city's financial health in the coming years provided insights into the trend, aiding in deciding whether to invest in the city improvement grant.

1.5. Technical Stack

Data pre-processing	Trifacta, MS Excel, Label Encoder
Framework/ Library	Scikit-learn, Streamlit, pandas, Plotly
Programming Language	Python, CSS, HTML
Tools	Visual Studio Code

Table 1: Technology used in the project.

1.6. Report Organization

The rest of the report is organized as follows:

Chapter 2: Background and Literature Review

This chapter comprehensively reviews the background and literature related to the research area. It includes a literature review, related research, and projects to establish the context and importance of the project.

Chapter 3: System Analysis and Architecture

This chapter focuses on the system analysis and architecture of the project. It discusses datasets, data pre-processing techniques, system design, and architecture. It also describes the overall project layout and the technologies and tools employed in building it.

Chapter 4: Implementation

This chapter delves into the implementation details of the project. It covers the dashboard implementation, tools and technologies used and developed, as well as the results and analysis of the project. This chapter highlights how the scheme works and the required technical aspects.

Chapter 5: Future Work

This chapter discusses potential future improvements to the project, including the current system's limitations and suggestions for future enhancements. It also explores potential applications of the project in real-world scenarios.

Chapter 6: Conclusion

This chapter wraps up the project report by summarizing the key findings, contributions made, and significance of the project. It also discusses the project's results and provides recommendations for future research. The overall evaluation of the project is also presented in this chapter, highlighting its importance and potential impact on the field.

Chapter 2: BACKGROUND

2.1 Literature Review

Numerous researchers have extensively contributed to data visualization, developing theories, and publishing scholarly articles. These articles primarily focus on identifying the best techniques for effective data visualization and establishing visual connectivity. The subsequent sections explore a variety of such reports and draw significant conclusions.

Data visualization aims to visually represent critical data for effective communication, with the understanding that images have a superior influence on memorization. For instance, the research paper "The Persuasive Power of Data Visualization" [5] provides insights into how data visualization can make a message more convincing and facilitate qualitative data analysis. Additionally, researchers have created a user interface, as illustrated in Figure 2, which outlines the aspects considered during visualization.

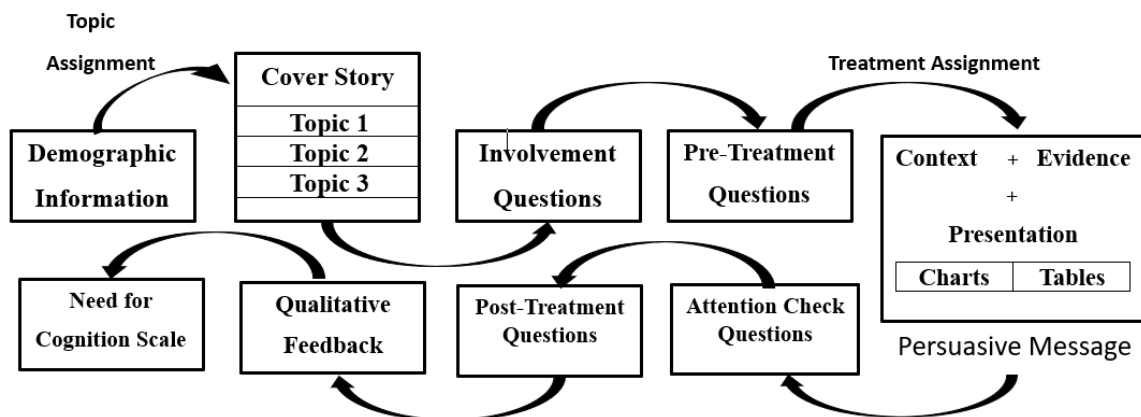


Figure 2: Stages of experiment [5].

The importance of visualizing urban planning lies in evaluating and organizing diverse data sets provided by the city. It is crucial to provide users with a sense of locality to simplify their understanding of the program [7]. Specifically, the content and operations performed on the dataset are the initial requirements for urban planning, followed by dividing the dataset into smaller data collections based on visualization techniques [8]. Visualizing smart city development involves understanding the flow of city planning and how technological advancements contribute to improving the quality of life [9]. Figure 3 depicts the web application architecture, which outlines the layout of communicating with a database to create data visualizations across multiple layers.

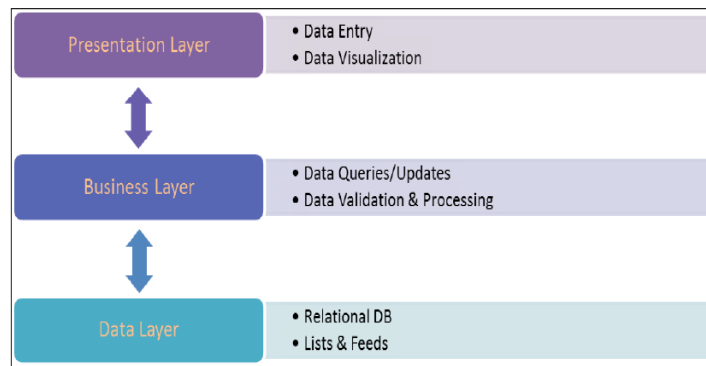


Figure 3: 3 Tier architecture [10].

Data visualization for cities has utilized various tools and strategies, including interactive and dynamic visualizations that enable real-time data exploration and static representations such as maps, charts, and graphs [6]. The book "Data Visualization: A Successful Design Process" details different visualization forms and considerations for creating them.

Many visualization tools, such as PowerBI and Tableau [5], have been developed to provide a user-friendly experience while creating visuals. These tools can create both static and interactive visuals. Another approach to representing data is using technologies like d3.js [6] and Python libraries like Matplotlib[10].

The dashboard is constructed utilizing Python, a widely used and powerful programming language renowned for its applications in machine learning and data science. Python provides a rich ecosystem of libraries and tools essential for developing sophisticated data-driven applications like dashboards. To implicitly manipulate and analyze data, the dashboard leverages the Pandas[11] and NumPy[12] libraries, known for their high-performance data structures and array programming capabilities. These libraries enable seamless data processing, transformation, and analysis, ensuring efficient handling of large datasets.

To initiate the machine learning section of this project, the book "Machine Learning: Concepts, Tools and Data Visualization" [13] was consulted. It explains how data visualization and machine learning algorithms can work together.

Machine learning is a method of analysis that uses algorithms to identify patterns in large data sets and make accurate predictions [14]. This research leverages machine learning models to enhance data visualization and expedite the data discovery process, providing valuable insights into their potential advantages. Machine learning encompasses a variety of deep learning models [15], as depicted in Figure 4.

As in the project, I intend to use deep learning models for prediction and classification. This paper specifies the data types and information required for these

models. These are vital considerations that students or visual designers should review when creating visuals, including potential ethical challenges[15] . The following section discusses the challenges that may arise when considering these factors.

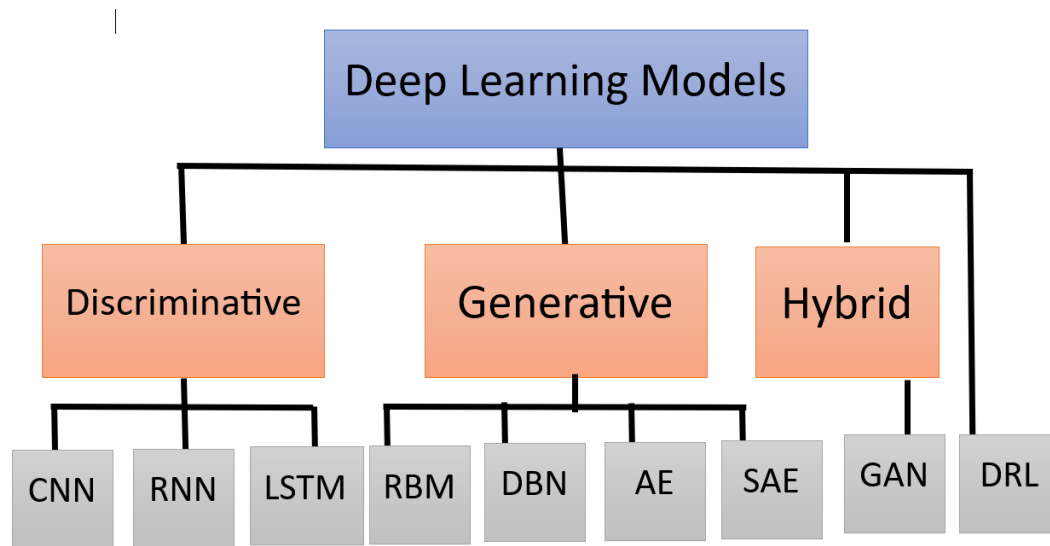


Figure 4: Deep learning models [16].

As described above, the dashboard employs machine learning algorithms for advanced analytics after data manipulation. The Random Forest Classifier [16] and the prediction model [17] are implemented using the scikit-learn library. Scikit-learn[18] is a widely used and comprehensive machine-learning library in Python that provides a rich set of tools for model training, evaluation, and selection.

To visually represent data in an appealing and informative manner, the dashboard utilizes the Matplotlib and Plotly libraries. Matplotlib is a popular plotting library in Python[10] that offers a wide range of customizable visualizations. Plotly[14] is a powerful

interactive plotting library that allows for creating interactive and dynamic visualizations, enhancing the dashboard's interactivity and user engagement.

For efficient coding and debugging, the development environment chosen for this project is PyCharm, an integrated development environment (IDE) that supports Python. PyCharm provides a comprehensive set of tools for coding, debugging, and testing Python applications, facilitating a smooth development process. PyCharm[19] .

Lastly, the dashboard is developed using Streamlit, a Python library that enables the rapid creation of web applications and interactive data visualizations. Streamlit[20] simplifies building interactive dashboards by providing a user-friendly interface and streamlined development workflow.

Integrating research findings, advanced machine learning algorithms, and data visualization techniques in dashboard development results in a valuable tool for urban planning. This dashboard provides an overview of city life, aids resource allocation, and facilitates evidence-based decision-making. By leveraging cutting-edge techniques, like machine learning algorithms, and utilizing data visualization tools, the dashboard empowers urban planners with accurate insights and transparent recommendations, enhancing the efficiency of urban planning processes.

Incorporating research findings ensures the dashboard is built on a solid foundation of empirical evidence and best practices in urban planning. It enables the dashboard to incorporate up-to-date knowledge and insights, enhancing its accuracy and relevance in addressing the complexities of urban planning challenges regarding financial health.

Integrating machine learning models and data visualizations in urban planning poses various technical challenges. The following section provides an overview of these challenges and their implications.

2.2 Challenges

Visualization and machine learning are powerful tools for analyzing and interpreting complex urban data. However, the use of unethical methods in these areas is a frequently cited concern. While many computer scientists believe that visualization programmers may consciously use unethical practices, ethical problems can arise due to a lack of knowledge about this area [24]. This section will discuss obstacles related to my project, specifically focusing on visualization and machine learning in the context of data analysis and explore strategies to overcome these ethical challenges.

1. Visualization Obstacles
2. Machine Learning Obstacles

2.2.1 Visualization Obstacles:

Visualizing a city can be challenging due to various obstacles that hinder the process. Some of the vital visualization obstacles are as follows:

- 1) **Manipulation of Data:** The quality and integrity of data are critical in visualization. Concerns arise when data from different sources are manipulated or cleaned before visualization. Additionally, when combining a small set of data with an extensive collection of data, there may be null and invalid values, which can significantly impact the integrity and accuracy of visualizations. Improper data handling can result in distorted visuals, leading to misleading interpretations and decisions.

- 2) **Type of Visuals:** Numerous visualization tools are available, such as PowerBI, Tableau, and Splunk, as well as Python libraries like Matplotlib and frameworks like D3.js and ReactJS, which support creating various graphs and charts. While these tools and frameworks make the work of visual developers more accessible, it is ultimately up to the developers to decide which tool to use and how to represent the data visually. However, developers may need a more comprehensive understanding of how these dashboards are created and may not strongly influence design decisions or data understanding. Therefore, it is essential to carefully consider the ethical implications of visualization design decisions for their intended audience to avoid moral dilemmas like misinterpretations and miscommunications.

It is crucial to ensure proper data handling and manipulation to overcome these visualization obstacles, including data cleaning, validation, and transformation. It is also essential for visual developers to have a good understanding of design principles, data literacy, and the intended audience to create effective and meaningful visualizations.

2.2.2 Machine Learning Obstacles:

Machine learning algorithms are often used to analyze and interpret complex urban data in visualization projects. However, there are several ethical obstacles associated with machine learning in data visualization:

- 1) **Dependence on Vast and Varied Datasets:** Machine learning algorithms rely on vast and diverse datasets for accurate predictions and observations. However, urban data can be incomplete, inconsistent, and distorted, affecting the accuracy and reliability of

- machine learning algorithms and the visualizations that depend on them. Ensuring data completeness and consistency is critical to prevent biases and inaccuracies in visualizations derived from machine learning.
- 2) Complexity and Opacity of Machine Learning Algorithms: Many machine learning algorithms are highly complex and considered "black box" models, challenging to understand how they arrive at their assumptions or decisions. This lack of transparency and interpretability can impact the integrity and interpretability of visualizations and observations derived from these algorithms. It is essential to use interpretable and transparent machine learning algorithms and explain the decisions made to ensure the results are interpretable and can be validated by other researchers and stakeholders.
 - 3) Ethical Concerns in Machine Learning: Applying machine learning in city data visualization can raise ethical concerns, such as unintended biases in data, which can result in distorted visualizations that propagate social, economic, or environmental imbalances in urban contexts. It is essential to review ethical considerations, such as equality, transparency, and responsibility, to ensure that machine learning-based visualizations do not propagate biases or exclusion.

In conclusion, ethical considerations play a crucial role in visualization and machine learning for data analysis. Overcoming obstacles related to data manipulation, visualization design decisions, data quality, transparency, and fairness in machine learning algorithms is crucial to ensure the integrity, reliability, and ethical use of visualization and machine learning in urban contexts. Researchers and practitioners should consider these ethical challenges carefully and strive for ethical best practices.

Chapter 3: SYSTEM ANALYSIS AND ARCHITECTURE

3.1 Dataset Summary

The VAST challenge dataset used in this research contains comprehensive information about the city of Engagement, located in Ohio, USA. The dataset includes data submitted by 1000 urban participants through urban planning, covering various aspects such as participant health, finances, joviality, business types, business turnover, and building data. The dataset has three sections: Attributes, Journals, and Activity Logs. Each section provides unique information for analysis and visualization.

3.1.1 Attributes:

This section includes static information about the urban participants, such as their demographics, socio-economic characteristics, and other relevant details. It provides a snapshot of the features of the urban population in Engagement, Ohio, and serves as a foundation for further analysis. The scope of the attribute file is as follows:

- 1) Apartments.csv: apartmentId, rentalCost, maxOccupancy, numberOfRooms, location, buildingId.
- 2) Buildings.csv: buildingId, location, buildingType, maxOccupancy, units.
- 3) Employers.csv: employerId, location, buildingId.
- 4) Jobs.csv: jobId, employerId, hourlyRate, startTime, endTime, daysToWork, educationRequirement.
- 5) Participants.csv: participantId, householdSize, haveKids, age, educationLevel, interestGroup, joviality.
- 6) Pubs.csv: pubId, hourlyCost, maxOccupancy, location, buildingId.

7) Restaurants.csv: restaurantId, foodCost, maxOccupancy, location, buildingId.

8) Schools.csv: schoolId, monthlyCost, maxEnrollment, location, buildingId.

3.1.2 Journals:

The scope of the attribute file is as follows: This section contains dynamic data that captures the daily activities and interactions of the participants. It includes daily routines, travel patterns, social interactions, and contextual information. This data provides insights into the daily lives and behaviors of the urban population, which can be analyzed to identify patterns and trends. The following is the journal file content:

- 1) CheckinJournal: participantId, timestamp, venueId, venueType.
- 2) FinancialJournal: participantId, timestamp, category, amount.
- 3) SocialNetwork: timestamp, participantIdFrom, participantIdTo.
- 4) TravelJournal: participantId, travelStartTime, travelStartLocationId, travelEndTime, travelEndLocationId, purpose, checkInTime, checkOutTime, startingBalance, endingBalance.

3.1.3 Activity Logs:

This section contains detailed logs of specific activities performed by the urban participants, such as financial transactions, business activities, health-related activities, and other relevant events. This data provides a detailed view of the activities and behaviors of the urban population, allowing for in-depth analysis and visualization. The following gives the inside of the activity field: ParticipantStatusLogs: timestamp, currentLocation, participantId, currentMode, hungerStatus, sleepStatus, apartmentId, availableBalance, jobId, financialStatus, dailyFoodBudget, weeklyExtraBudget.

3.2 Data Pre-processing

Data pre-processing, or data preparation or cleaning, is a fundamental step in data analysis. It involves a series of operations to transform raw data into a well-organized, clean, and structured state suitable for analysis. The quality of the data used for analysis directly impacts the accuracy and reliability of the analysis results, making data pre-processing an essential step in ensuring data reliability, accuracy, and consistency, which leads to superior analysis results and discoveries. Typical data pre-processing tasks include data cleansing, data integration, handling outliers, data transformation, feature selection, and data validation. These tasks ensure that the data is free from errors, inconsistencies, and redundancies and is ready for analysis. Figure 5 illustrates the relationship between datasets that I manipulated to perform the analysis for the visuals.

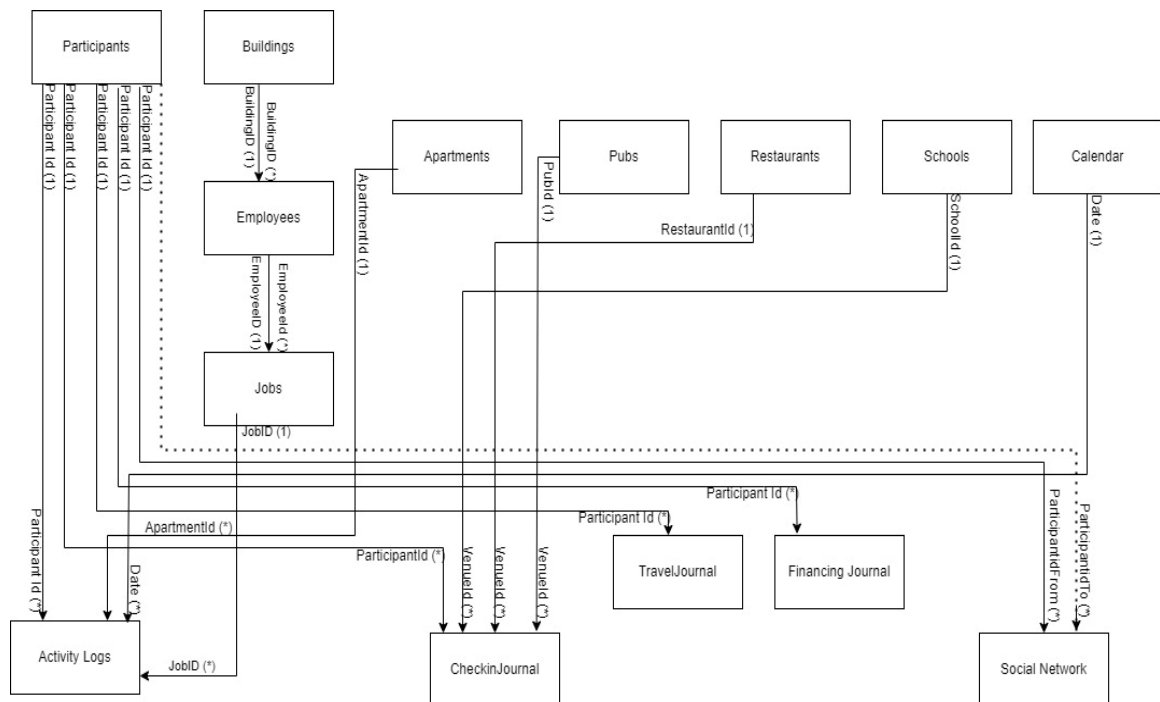


Figure 5: Relationship among datasets.

In this, data pre-processing is performed on the Vast challenge data, encompassing all the elements in the city as attributes, journals, and activity logs for all the participants. I analyze all the components to operate on the city's financial health.

Two central Data manipulations are done in this dashboard apart from other null value cleanings. I utilized the Trifacta tool and Python libraries like pandas to facilitate the data manipulation tasks.

Dataset manipulated:

1. Financial Journal and Participants.
2. Participant Status Logs

3.2.1 Financial Journal and Participants:

Specifically, two CSV files, "FinancialJournal" and "Participants," are combined to create a new dataset. The process involves removing duplicates from the combined dataset to ensure data cleaning.

Figure 6 exemplifies the use of the `drop_duplicates()` function on two CSV files for identifying and removing duplicate records based on specific criteria.

```
def getdata(self):
    fj = pd.read_csv(os.path.join('Data', 'Journals', 'FinancialJournal.csv'))
    partcpnts = pd.read_csv(os.path.join('Data', 'Attributes', 'Participants.csv'))
    fj.drop_duplicates(inplace=True)
    partcpnts.drop_duplicates(inplace=True)
    return fj, partcpnts
```

Figure 6: Drop duplicate before creating new dataset.

Before combining the operations performed on both the datasets are as follows:

- Financial Journal: Enhancing visual accuracy for viewers. The timestamp is separated into date format in the Participant's dataset, As shown in Figure 7, with the help of `datetime()` and `strftime()` functions. Such as year, month, and day, for improved readability and analysis. For example, "02022023" is converted into "20230202" in the desired format.

```
def splitdata(self):
    fj, partcpnts = self.getdata()
    category = fj["category"].unique()
    partid = fj["participantId"].unique()
    newfj = pd.pivot_table(fj, values='amount', index = ['participantId', 'timestamp'], columns=['category']).
    reset_index()
    nefj = newfj.fillna(0)
    newfj['timestamp'] = pd.to_datetime(newfj['timestamp']).dt.strftime('%Y-%m-%d')
    newfj['timestamp'] = newfj['timestamp'].str.replace('-', '').astype('int')
    # print(newfj)
    newfj = newfj.rename(columns =({'timestamp':'date'}))
    newfj = newfj.groupby(['participantId', 'date'])['Education', 'Food', 'Recreation', 'RentAdjustment', 'Shelter',
    'Wage'].sum().reset_index()
```

Figure 7: Timestamp conversion.

- Participants: This dataset replaces the education level attribute with numerical codes for better analysis. For example, "HighschoolorCollege" is replaced with 2, "Low" with 1, "Bachelors" with 4, and "Graduate" with 3. This transformation enables better quantitative analysis of the education level attribute.
- The code for performing the manipulation is visually depicted in Figure 8.
- Furthermore, the "Have Kids" section has been enhanced with modifications. If the participant indicates they have a child, a selected button will be displayed in the corresponding column, denoting the affirmative response.
- The resulting dataset, after these manipulations, is presented in Figure 9.


```
def janitor(self,newfj):
    fj, partcpnts = self.getdata()
    fj = pd.merge(newfj, partcpnts, how='left', on='participantId')
    fj['educationLevel'] = fj['educationLevel'].str.replace('HighSchoolOrCollege','2').str.replace('Low','1').str.replace('Bachelors','4').str.replace('Graduate','3')
    return fj
```

Figure 8: Education level conversion.

Dataset														
	participantId	date	Education	Food	Recreation	RentAdjustment	Shelter	Wage	householdSize	haveKids	age	educationLevel	interestGroup	joviality
439	0	20230514	0	-11.019	0	0	0	0	3	<input checked="" type="checkbox"/>	36	2	H	0.0016
440	0	20230515	0	-8.1733	0	0	0	411.28	3	<input checked="" type="checkbox"/>	36	2	H	0.0016
441	0	20230516	0	-8.1733	0	0	0	411.28	3	<input checked="" type="checkbox"/>	36	2	H	0.0016
442	0	20230517	0	-8.1733	0	0	0	411.28	3	<input checked="" type="checkbox"/>	36	2	H	0.0016
443	0	20230518	0	-8.1733	0	0	0	411.28	3	<input checked="" type="checkbox"/>	36	2	H	0.0016
444	0	20230519	0	-8.1733	0	0	0	411.28	3	<input checked="" type="checkbox"/>	36	2	H	0.0016
445	0	20230520	0	-9.5095	0	0	0	0	3	<input checked="" type="checkbox"/>	36	2	H	0.0016
446	0	20230521	0	-9.5095	0	0	0	0	3	<input checked="" type="checkbox"/>	36	2	H	0.0016
447	0	20230522	0	-8.1733	0	0	0	411.28	3	<input checked="" type="checkbox"/>	36	2	H	0.0016
448	0	20230523	0	-8.1733	0	0	0	411.28	3	<input checked="" type="checkbox"/>	36	2	H	0.0016
449	0	20230524	0	-8.1733	0	0	0	411.28	3	<input checked="" type="checkbox"/>	36	2	H	0.0016
450	1	20220301	-38.0054	-10.5923	-30.8237	0	54.9886	2,407.9736	3	<input checked="" type="checkbox"/>	25	2	B	0.3281
451	1	20220302	0	-10.2059	-22.1148	0	0	361.4114	3	<input checked="" type="checkbox"/>	25	2	B	0.3281
452	1	20220303	0	-9.5095	-35.1388	0	0	361.4114	3	<input checked="" type="checkbox"/>	25	2	B	0.3281
453	1	20220304	0	-10.2059	-43.8086	0	0	361.4114	3	<input checked="" type="checkbox"/>	25	2	B	0.3281

Figure 9 : Data manipulation over Financial Journal and Participants.

3.2.2 Participant Logs:

In addition to the above, Participant Logs data also requires cleaning and integration. There are 72 CSV files for Participant logs, which consist of participants' activity over one year. These files are combined into a CSV file named "ParticipantStatusLogs" to facilitate further analysis. Additionally, the data in this file is updated to determine participants' stable, unstable, and unknown statuses. After careful

analysis, the data from five ParticipantStatusLog CSV files are combined to create a unified dataset for predicting financial health, as elaborated in the subsequent section of this report.

The code flow in Figure 10 is as follows:

- Creating a copy of a DataFrame 'df1' and storing it in a new DataFrame 'df_jov' using the 'copy()' function from the 'copy' module.
- Defining the file path for CSV files containing activity logs data.
- Creating an empty dictionary 'all_als' to store DataFrames read from CSV files.
- Looping through a list of integers [1,2,6,7,72], and for each integer: Reading the corresponding CSV file using 'pd.read_csv()' function and appending it to the 'all_als' dictionary with a key of "al_" followed by the integer value.
- I am creating an empty DataFrame 'cdf' to store selected datasets. Also, I am initializing an empty list 'mls' to store the keys of the selected DataFrames.
- Looping through the keys ['al_1','al_2','al_6','al_7','al_72'] in the 'all_als' dictionary, and for each key: Checking if unique values number in the 'financialStatus' column of the DataFrame is greater than 1, indicating the presence of both stable and unstable categories. If the condition is met, append the DataFrame to 'cdf' and add the key to 'mls' list.

The financial status prediction classification model incorporates details outlined in Section 4, "Machine Learning Models," of this report. These details are utilized as inputs in the classification model to facilitate the accurate prediction of financial status.

```

df_jov = copydf1.copy()
path = r"C:\Users\Owner\Desktop\Project\DataSet\Activity Logs\ParticipantStatusLogs"
all_als = {}
for i in [1,2,6,7,72]:
    df = pd.read_csv(path + str(i)+".csv")
    all_als["al_"+str(i)] = df
st.write("The activity logs of the people have the data of their activities within a fixed duration.
It records thier financial status after each activity to see if they are financially stable or not.")
cdf = pd.DataFrame()
st.write('>> Out of 72 datasets having records of their financial status we will select only the
datasets that contain stable and unstable both categories. \n Here are the Suffixes of each dataframe
that satisfies abovesaid condition.')
mls = []
for key in ['al_1','al_2','al_6','al_7','al_72']:
    if len(all_als[key]['financialStatus'].unique()) > 1:
        cdf = cdf.append(all_als[key])
        mls.append(key)
st.write("df suffixes are : ",str(mls))

```

Figure 10: ParticipantStatusLog conversion.

Data pre-processing is an imperative step in the data analysis process. It ensures data reliability, accuracy, and consistency, leading to superior analysis results and insights. Python libraries and various data manipulation techniques facilitate the cleaning, integration, and transformation of raw data into a structured and organized form, essential for meaningful analysis and interpretation of research results.

Chapter 4: IMPLEMENTATION

Streamlit, an open-source Python toolkit, is utilized in this project for constructing interactive web applications for machine learning and data science projects simply and efficiently. Streamlit empowers developers to quickly build online applications with interactive user interfaces (UIs) to display data, explore machine learning models, and gather user feedback. The potential applications of Streamlit include constructing web-based dashboards, interactive reports, data visualization tools, and more.

4.1 Dependency Management

The dashboard implementation begins with importing the necessary Python libraries, which mainly consist of plotting and machine-learning models. The following is the description of the libraries used in this project:

4.1.1. pandas:

pandas are a powerful library used for data manipulation and analysis. It provides a data structure format known as DataFrame, which efficiently manages data in a tabular form, making it suitable for analysis.

4.1.2. NumPy:

NumPy, also known as Numerical Python, is a popular library used for numerical computing. It offers various tools that enable users to perform mathematical operations on matrices and arrays, making it a fundamental tool for data analysis.

4.1.3. Matplotlib:

Matplotlib is a widely used library for data visualization. It provides a plethora of tools for creating plots, graphs, and charts, making it an invaluable resource for visualizing data.

4.1.4. Scikit-learn:

Scikit-learn, commonly referred to as sklearn, is a comprehensive library that offers a wide range of tools for model evaluation, model selection, and feature extraction, making it indispensable for developing machine learning models.

4.1.5. Seaborn:

Seaborn is a powerful Python data visualization toolkit built on Matplotlib. It provides a high-level interface for constructing visually appealing and informative statistical visuals. Seaborn is designed explicitly for displaying complex information and is particularly effective for researching and visualizing correlations between variables in large databases.

4.1.6. Plotly:

Plotly is a popular Python data visualization framework that allows users to create dynamic web-based plots and dashboards. It is an ideal choice for developing interactive data visualizations for data presentation, data analysis, and exploratory data analysis (EDA), as it provides a flexible and powerful means to generate interactive visualizations that can display in web browsers.

4.2 System Blueprint

The implementation of the dashboard is divided into different sections, including:

4.2.1. Data Preparation:

This section involves importing and manipulating data using pandas and NumPy libraries to prepare it for further analysis and visualization.

4.2.2. Model Building:

In this section, machine learning models are developed using the Scikit-learn library. Model evaluation and selection techniques are also applied to choose the best-performing model.

4.2.3. Data Visualization:

Matplotlib, Seaborn, and Plotly libraries are utilized in this section to create various visualizations, such as plots, graphs, and charts, to represent the analyzed data and model results effectively.

4.2.4. Dashboard Creation:

Streamlit creates an interactive web-based dashboard that incorporates the data visualizations and machine learning models developed earlier. The dashboard provides a user-friendly interface for users to explore the data and interact with the machine learning models, allowing them to gain insights and provide feedback.

Overall, implementing the dashboard involves seamlessly integrating various Python libraries, including pandas, NumPy, Scikit-learn, Matplotlib, Seaborn, Plotly, and Streamlit, to construct an interactive web-based application for data analysis and machine learning model exploration.

4.3 Dashboard Pipeline

4.3.1. Data Numerical Statistic:

In implementing the dashboard, a required step is the analysis of numerical data statistics. It involves performing modifications on the dataset, such as data cleaning, pre-processing, and feature engineering, to ensure that it is in a suitable format for analysis.

Once appropriately prepared, the dataset created a new file named "Dataset.csv" to store the modified data.

Next, load the dataset into a pandas dataframe, a powerful data manipulation and analysis tool in Python. The "description()" function, provided by the pandas library, is then applied to the dataframe to generate a comprehensive summary of the data statistics. It includes measures such as count, mean, minimum, maximum values, standard deviation, and percentiles at 25%, 50%, and 75%. These statistics offer valuable insights into the data's distribution, central tendency, and spread.

	count	mean	std	min	25%	50%	75%	max
Participant Id	396,700	481.327	294.280	0	225	463	725	1,010
Education	396,700	-0.439	5.297	-91.143	0	0	0	0
Food	396,700	-11.501	2.9722	-20	-13.76	-10.198	-9.1132	0
Recreation	396,700	-12.536	26.8614	-198.182	-8.2484	0	0	0
Rent Adjustment	396,700	0.1109	13.1806	0	0	0	0	3,865.798
Shelter	396,700	-21.222	121.229	-3,469	0	0	0	0
Wage	396,700	140.239	181.54	0	0	122.99	195.078	4,749.594
Household Size	396,700	1.8986	0.8081	1	1	2	3	3
Age	396,700	39.1299	12.3919	18	29	39	50	60
Education Level	396,700	2.6555	0.9381	1	2	2	4	4
Joviality	396,700	0.466	0.287	0.0002	0.217	0.448	0.697	0.992

Table 2: Numerical statistics on dataset.

Table 2 visually illustrates the numerical data statistics. This table aids as a reference for the reader to understand the dataset's characteristics, including the summary

statistics of each attribute. It facilitates detailed data analysis, identifying trends, patterns, and outliers that may influence the subsequent steps in implementing the dashboard.

4.3.2. Correlation Heatmap Of The Data:

A correlation heatmap is a visual depiction of a dataset's correlation matrix that uses color to indicate the relationship between each pair of variables. A statistical measure is known as correlation quantifies the direction and degree of the relationship between two variables. A positive correspondence means that when one variable rises, the other tends to rise. In contrast, a negative correlation means that as one variable rises, the other tends to fall. The variables have no linear link when the correlation value is 0.

The intensity of the connection is often depicted on a correlation heatmap using a color gradient. A color scale goes from white to purple to signify a positive association. The color scale from purple to black represents a negative correlation. The magnitude or strength of the correlation is represented by the color's intensity, with darker or more vivid hues denoting stronger correlations.

The correlation matrix is first computed to create a correlation heatmap by calculating the correlation coefficients between pairs of variables in the dataset, typically using a method such as Pearson's correlation coefficient. The correlation matrix is then visualized as a heatmap using graphical plotting libraries or software tools, with the colors representing the strength and direction of the correlations.

Below Figure 11 is the code snippet for the correlation heatmap of the data:

- Here '`corr_mat = df.corr()`' line calculates the correlation matrix (`corr_mat`) of the data frame (`df`) using the `.corr()` function. `df` contains the dataset file.


```
st.subheader('Correlation heatmap of the data')
corr_mat = df.corr()
fig, ax = plt.subplots()
sns.heatmap(corr_mat, ax=ax)
st.write(fig, use_container_width=True)
```

Figure 11: Heatmap code snippet.

- 'fig, ax = plt.subplots()' line of code creates a Matplotlib figure (fig) and axes (ax) object, which will be used to plot the correlation.
- 'sns.heatmap(corr_mat, ax=ax)' uses the Seaborn library to create a heatmap of the correlation matrix (corr_mat) on the ax axes object. The sns.heatmap() function in Seaborn generates a heatmap with colors representing the strength and direction of correlations.
- st.write(fig, use_container_width=True): This line of code is using Streamlit to display the heatmap figure (fig) created by Seaborn in the web application. The use_container_width=True argument ensures that the figure is displayed within the width of the container in the Streamlit app, making it responsive and visually appealing.

Overall, this code generates a correlation heatmap of the data in the data frame (df) using Seaborn and displays it in the dashboard using Matplotlib and Streamlit functions. The heatmap can help visualize the strength and direction of relationships between variables in the data frame, providing insights into the data's correlation structure. Figure 12 shows the outcome of the heatmap. None of the features are highly correlated, making our data suitable for Model Training.

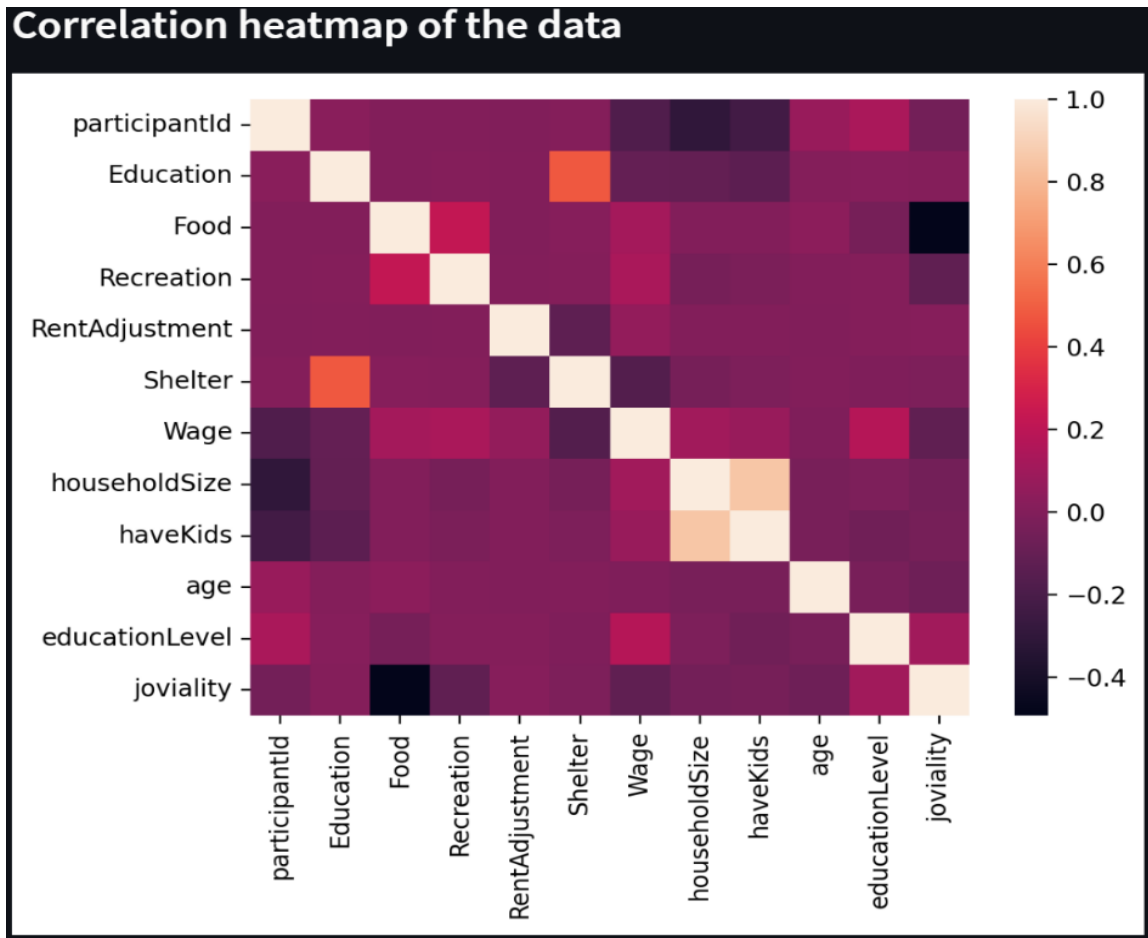


Figure 12: Heatmap of the data.

4.3.3. Machine Learning Models:

1) Regression Model – Predicting Joviality

Regression modeling is a statistical method for simulating the relationship between an independent variable (known as the predictor variable) and a dependent variable (known as the response variable). A regression model's objective is to estimate the model's parameters, which may then be used for predicting outcomes or inferring correlations between variables based on the observed data.

Label encoding is an approach used in machine learning and data pre-processing to turn categorical or textual data into mathematical labels that machine learning algorithms can easily interpret. It is a type of feature transformation or feature encoding. In label encoding, a categorical variable's distinct categories or labels are each given a unique numerical label.

For example, Label Encoder is applied to "interestGroup" and "haveKids" assigned a numerical label. After that, I dropped the participantid and date column for better prediction over joviality.

- The test and train split for this prediction is 0.2 and 0.8.
- To train our model, I used Random Forest Regressor with 40 estimators.

Random Forest Regressor is a machine learning ensemble approach that combines many decision tree models to produce a solid and accurate prediction model. The number of estimators is a Random Forest method hyperparameter that defines the number of decision trees to include in the ensemble. For the Random Forest ensemble following is the formula:

$$y = f1(x) + f2(x) + \dots + fn(x)$$

where,

- $f1(x)$, $f2(x)$, ..., $fn(x)$ is each individual decision tree in the decision function.
- n is the number of decision trees in total.
- y is the dependent variable (i.e., predicted target variable).
- x is the independent variable (i.e., the input feature vector).

The target variable y is predicted using Random Forest Regression by averaging the forecasts of many ensemble decision trees, hence the name "random forest."

For example, I set the `n_estimators=40`. It means the number of estimators in a Random Forest Regressor is set to 40. The algorithm will generate 40 unique decision tree models during training, each using a random subset of the training data. The predictions from these different trees will be integrated to get the final forecast for a particular input. Visuals for joviality and predicted joviality are shown in section 4, "Generating Future Data."

Figure 13 provides a comprehensive overview of the code implementation for the model and the operations described previously. It visually presents the key components, functions, and algorithms utilized in the model, along with their interconnections and relationships. And Figure 14 gives the final score of the model.

```
copydf1 = copydf.copy()
copydf1['date'] = copydf1['date'].astype('str')
copydf = copydf.drop(columns= ['participantId','date'])
st.write()
df_y = copydf['joviality']
X_train, X_test, y_train, y_test = train_test_split(copydf.drop('joviality',axis=1),df_y,test_size=0.2)

st.write()
model_reg = RandomForestRegressor(n_estimators=40)
model_reg.fit(X_train, y_train)

st.write('The score of our model is : ',model_reg.score(X_test,y_test))
y_predicted = model_reg.predict(X_test)
```

Figure 13: Predicting joviality code snippet.

The score of our model is : 0.9938268022261887

Figure 14: Predicting joviality score.

1) Classification Model – Predicting Financial Status

A classification model uses features or attributes to predict the class or category of a given input data item. Classification is an automated learning job in which the model is trained on labeled data consisting of examples of inputs and their matching class labels. Once trained, the classification model can predict the class labels of new, previously unknown data pieces. The activity logs of the people have the data of their activities within a fixed duration. It records their financial status after each activity to determine their stability. Here I performed prediction over "ParticipantStatusLogs." Data manipulation is done on this file, as mentioned in section 3.

Out of 72 datasets having records of their financial status, we will select only the datasets that contain stable and unstable both categories. Here are the Suffixes of each dataframe that satisfies the above-said condition. "df suffixes are: ['al_1', 'al_2', 'al_6', 'al_7', 'al_72']." The length of the activity log data is manipulated and printed, in this case, "6913218," where the unique count of stable vs. unstable is shown in Table 3.

Financial Status	Count
Stable	6,744,350
Unstable	127,970
Unknown	40,896

Table 3: Unique count before.

There is a third category in financial status, Unknown. So, we are Putting this Unknown category of financial status into the 'Unstable' category. Now the unique counts of stable vs. unstable are shown in Table 4.

Financial Status	Count
Stable	6,744,350
Unstable	168,866

Table 4: Unique count after.

In line with previous procedures, a Heatmap was generated to assess null values within the dataset, explicitly focusing on the "financialStatus" attribute. Figure 15 provides a comprehensive overview of the heatmap, with the following key features:

- The y-axis represents the attributes with null value present in the dataset.
- The x-axis showcases the number of records available, ranging from 0 to 7 million.
- The heatmap in question uses a color scale to represent the presence of null values in a dataset visually. The heatmap uses dark colors to indicate that there are no null values (i.e., a false condition) and light colors to represent the presence of one null value (i.e., a true condition). Upon conducting a detailed analysis of the dataset, it was observed that all attributes, except for the "financialStatus" attribute, contained no null values. Null values are typically missing or unknown data points in a dataset, and they can adversely affect the accuracy and reliability of data analysis and modeling.

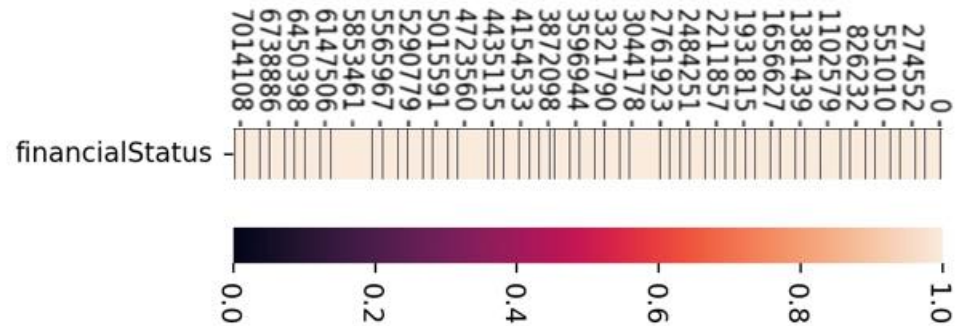


Figure 15: Heatmap to check null values.

- To address this issue, the dataset was filtered to eliminate the null values in the "financialStatus" attribute. This was done to ensure that the dataset used for training or further analysis was free of any missing data points in the "financialStatus" attribute, making it suitable for accurate and reliable analysis.
- After applying the filtering process, I observed that of the total 7 million records in the dataset, approximately 70,000 values in the "financialStatus" attribute were found to be non-null. It indicates that most of the data in the "financialStatus" attribute was complete and contained no missing values. The availability of a significant portion of data in the "financialStatus" attribute can contribute to more accurate and reliable results in subsequent analyses or modeling tasks, as the dataset is now enriched with comprehensive data in this attribute. This finding highlights the importance of data quality and integrity in ensuring the robustness of data-driven analyses and modeling efforts.

```

mgdf_nn = mgdf[mgdf.financialStatus.notnull()].reset_index().drop(columns=['index'])
mgdf_in = mgdf[mgdf.financialStatus.isnull()].reset_index().drop(columns=['index'])

mgdf_nn_1 = mgdf_nn.copy()
mgdf_nn = mgdf_nn.drop(columns=['participantId', 'date', 'joviality']).dropna()
mgdf_nn = mgdf_nn.dropna()

y_dat = mgdf_nn['financialStatus']
X_train, X_test, y_train, y_test = train_test_split(mgdf_nn.drop(columns=['financialStatus']), y_dat,
test_size=0.2)
st.write('>>2. The train test split is : Train size = 0.8 , Test size = 0.2 \n')
'>>3. To train our model, we are using RandomForestClassifier with 2 estimators. The reason to use
less estimators is to prevent our model from overfitting.')
model_cla = RandomForestClassifier(n_estimators=2)
model_cla.fit(X_train, y_train)
st.write('The score of the model is : ', model_cla.score(X_test, y_test))

```

Figure 16: Predicting financial status code snippet.

After filtering the dataset to remove null values from the "financialStatus" attribute, the filtered data was used to train a classifier model.

In brief, the code in Figure 16 executes the subsequent technical operations:

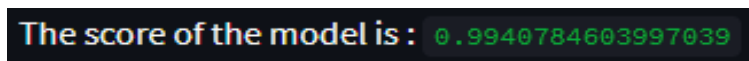
- Creates two new dataframes, `mgdf_nn`, and `mgdf_in`, by indexing and resetting the original dataframe `mgdf` based on whether the 'financialStatus' column is not null or null, respectively. The 'index' column is dropped from both dataframes.
- Creates a copy of `mgdf_nn` as `mgdf_nn_1`, and then drops unnecessary columns ('participantId', 'date', 'joviality') and rows with missing values from `mgdf_nn`, preparing the data for model training.
- Extracts the target variable 'financialStatus' from `mgdf_nn` as `y_dat` and performs feature extraction by splitting it into training and testing sets using the split function with a test size of 0.2 (20%).
- Defines a `RandomForestClassifier` model with 2 estimators to prevent overfitting.

- Fits the RandomForestClassifier model to the training data using the 'fit' method, training the model on the features (input variables) and corresponding target variable (output variable).
- Calculates and displays the accuracy score of the trained model on the testing data using the 'score' method, which measures the model's performance in predicting the target variable on unseen data.

A RandomForestClassifier model was chosen for this task, with two estimators. RandomForestClassifier is an ensemble learning method that combines multiple decision trees to make predictions. This division allows for evaluating the model's performance on unseen data, which helps to assess its generalization capability.

Using multiple trees in a random forest mitigates the risk of overfitting. It occurs when a model is too complex to learn and memorize the training data instead of generalizing it.

Figure 17 displays the output of the classification model, presenting the predicted score for the financial status.



```
The score of the model is : 0.9940784603997039
```

Figure 17: Predicting financial status.

In order to ensure that the "financialStatus" attribute in the dataset does not contain any null or missing values, a thorough analysis was conducted. This verification process

involved visualizing the dataset's heatmap (Figure 15), which visually represents the presence or absence of null values in the "financialStatus" attribute.

To further validate the absence of null values, Table 5 is generated, which displays the total count of null values in the "financialStatus" attribute. After curating the dataset, the count of null values in the "financialStatus" attribute becomes null, indicating that the curation process involved dropping the rows with null values, as illustrated in Figure 16.

To facilitate the analysis and comparison of the original and curated data, a new DataFrame was created using the Pandas library. This DataFrame includes two columns labeled 'Original null count for financial status' and 'Curated null count for financial status.' The values in these columns are obtained from the original and curated data. In this case, the original null count in the "financialStatus" attribute was reported to be 369,685, indicating a significant number of null values in the original dataset.

A dataset without missing data ensures that the model is trained on complete and reliable data, which can lead to more accurate and trustworthy results in subsequent analyses or modeling tasks.

Financial Status	Null Count
Original	369685
After curating	0

Table 5: Null count for financial status.

Figure 18 provides a visual representation of the unique count of "financialStatus" records in the filtered data, offering insights into the distribution of Stable and Unstable

records in the dataset. This information can be valuable for further analysis or modeling tasks that require understanding the characteristics of different "financialStatus" categories within the dataset.

	financialStatus
Stable	395,610
Unstable	1,091

Figure 18: Unique counts of financial status.

4.3.4. Curating Data- Generating Future Data:

Data curation, which includes data collection, organization, cleaning, and preparation, is a critical stage in data analytics and machine learning workflow. The accuracy and dependability of the insights or models obtained from the data directly depend on the quality and reliability of the data used for analysis.

To curate future data, I have chosen to use CAGR (Cumulative Annual Growth Rate) as the basis for calculating the particulars for the following year. Although there may be other ideas for data curation, CAGR is generally applicable to all participants and is expected to yield good results. I have also calculated CMGR (Cumulative Monthly Growth Rate) as an additional step, as the dataset contains various monthly data points and yearly data. This allows us to capture and account for any monthly variations in the data, ensuring a more comprehensive and accurate analysis of future data. Following Figure 19 is the curated data along with the calculated CAGR.

	participantid	householdSize	haveKids	age	educationLevel	interestGroup	date	Education	Food	Recreation	RentAdjustment	Shelter	Wage
0	0	3	1	36	2	7	202203	38.0054	263.8075	338.3596	0	554.9886	11,558.228
1	0	3	1	36	2	7	202204	38.0054	261.3673	212.9114	0	554.9886	8,366.365
2	0	3	1	36	2	7	202205	38.0054	260.1483	371.6325	0	554.9886	8,764.7633
3	0	3	1	36	2	7	202206	38.0054	252.6337	451.8496	0	554.9886	8,764.7633
4	0	3	1	36	2	7	202207	38.0054	265.6506	1,037.7758	0	554.9886	8,366.365
5	0	3	1	36	2	7	202208	38.0054	257.4178	304.8108	0	554.9886	9,163.1616
6	0	3	1	36	2	7	202209	38.0054	251.7946	285.8944	0	554.9886	8,764.7633
7	0	3	1	36	2	7	202210	38.0054	262.2317	24.2719	0	554.9886	8,366.365
8	0	3	1	36	2	7	202211	38.0054	256.6662	366.2314	0	554.9886	8,764.7633
9	0	3	1	36	2	7	202212	38.0054	261.5631	346.1286	0	554.9886	8,764.7633
10	0	3	1	36	2	7	202301	38.0054	260.5679	203.5272	0	554.9886	8,764.7633
11	0	3	1	36	2	7	202302	38.0054	235.079	310.4557	0	554.9886	7,967.9666
12	0	3	1	36	2	7	202303	38.0054	256.217	141.8722	0	554.9886	9,163.1616
13	0	3	1	36	2	7	202304	38.0054	253.7768	114.7176	0	554.9886	7,967.9666
14	0	3	1	36	2	7	202305	38.0054	204.3429	215.2541	0	554.9886	7,171.17
15	1	3	1	25	2	1	202203	38.0054	283.1713	969.6006	0	554.9886	10,042.6699
16	1	3	1	25	2	1	202204	38.0054	251.5082	476.8226	0	554.9886	7,357.8591
17	1	3	1	25	2	1	202205	38.0054	252.9865	331.7499	0	554.9886	7,708.2334
18	1	3	1	25	2	1	202206	38.0054	255.7766	624.901	0	554.9886	7,708.2334
19	1	3	1	25	2	1	202207	38.0054	265.1855	467.3968	0	554.9886	7,357.8591
20	1	3	1	25	2	1	202208	38.0054	265.4036	405.3896	0	554.9886	8,058.6076

Figure 19: Curated data along with the calculated CAGR.

Wages vs. Expenses Plot

I have thoroughly examined and visualized various categories in analyzing the wages vs. expenses plot. These visualizations have been generated using previously manipulated data, and the section is segmented into Education Level, Age Group, Have Kids, Household Size, and Interest Group for comprehensive analysis.

The first visual, Figure 20, showcases the combined data of wages vs. expenses, aggregated according to the date. An interactive panel allows for data exploration for each month from March 2023 to May 2024. The date intervals are set based on the Cumulative Monthly Growth Rate (CMGR) for greater accuracy. Specifically, the months of March,

May, July, September, and November in 2023 and January, March, and May in 2024 are visible in the visual. Furthermore, hovering over the data points in the visual provides additional details, revealing the expenses and wages of the total number of participants. It allows for a more in-depth data analysis, enabling insights into the participants' spending patterns and income levels at different time points.

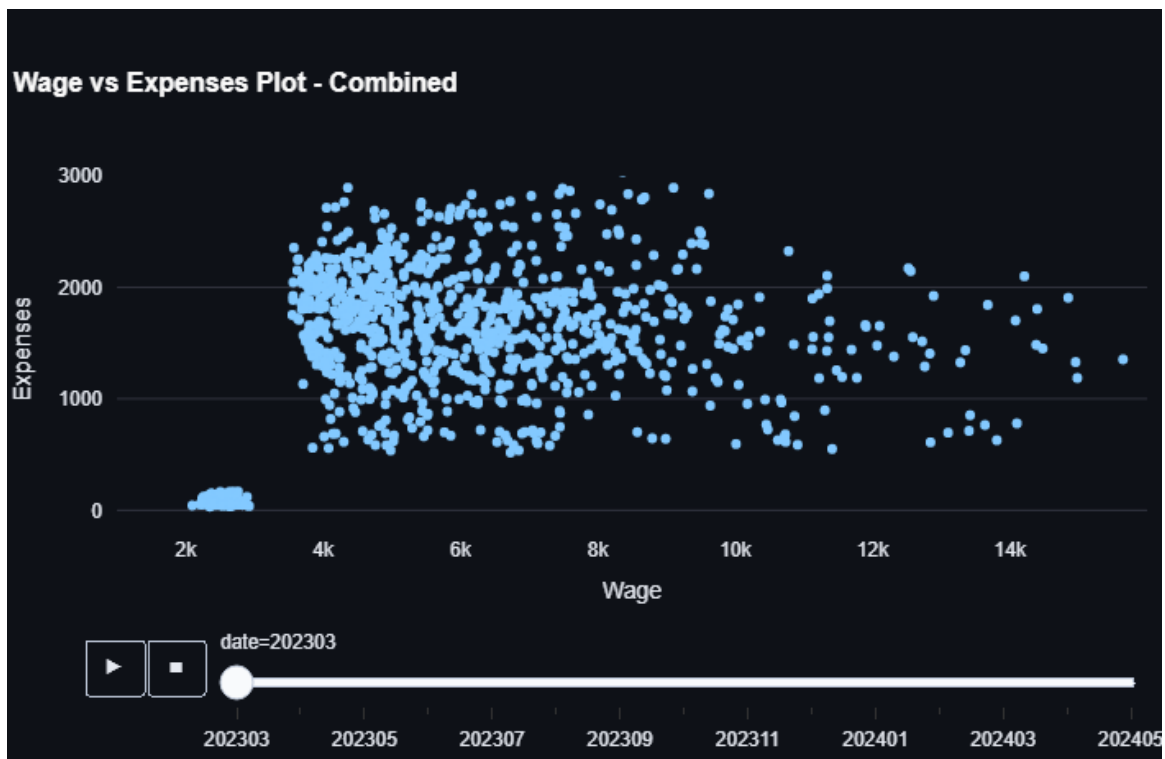


Figure 20: Wage vs. Expenses Plot Combined.

Overall, this meticulous analysis and visualization of the wages vs. expenses data, along with the interactive features and detailed information, aim to provide a comprehensive understanding of the trends and patterns in the data for the specified time, enhancing the accuracy and reliability of the analysis.

Figure 21 presents a scatter plot that illustrates the correlation between wages and expenses among participants, with the data points segmented by their educational background. The educational levels are categorized into four distinct groups, namely "High School or College," "Graduate," "Bachelors," and "Low," as per the predefined categories in the dataset. The scatter plot is plotted using the date format previously specified in the data. This graphical representation supplies valuable insights into the participant's financial health, visually analyzing how their wages and expenses are distributed across different education levels. By examining the scatter plot, patterns, trends, or anomalies in the data can be identified, enabling further analysis of the participants' financial behaviors and financial well-being based on their educational attainment.

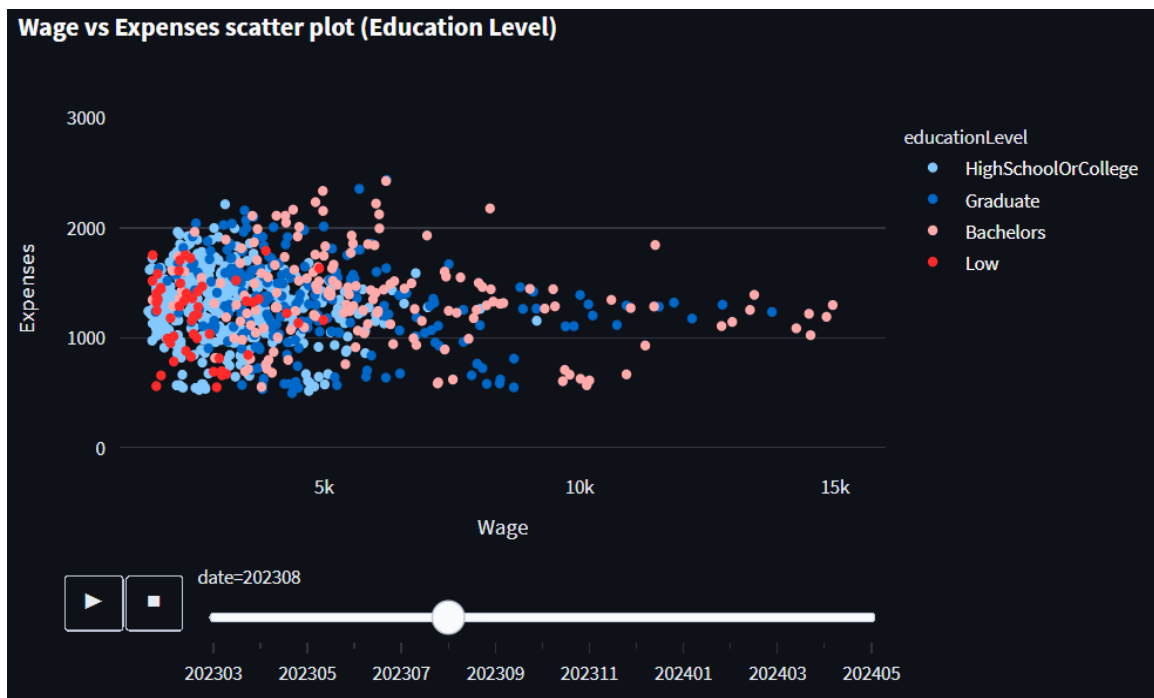


Figure 21: Wage vs. Expenses scatter plot (Education Level).

Furthermore, Figure 22 showcases a tool feature utilized by Streamlit. The specific tool feature employed in the analysis is detailed in the figure, highlighting the software or functionality employed to generate the scatter plot and visualize the data in a user-friendly and interactive manner. This further demonstrates the utilization of advanced technical tools in the data analysis process, adding rigor and precision to the findings presented in the figures.

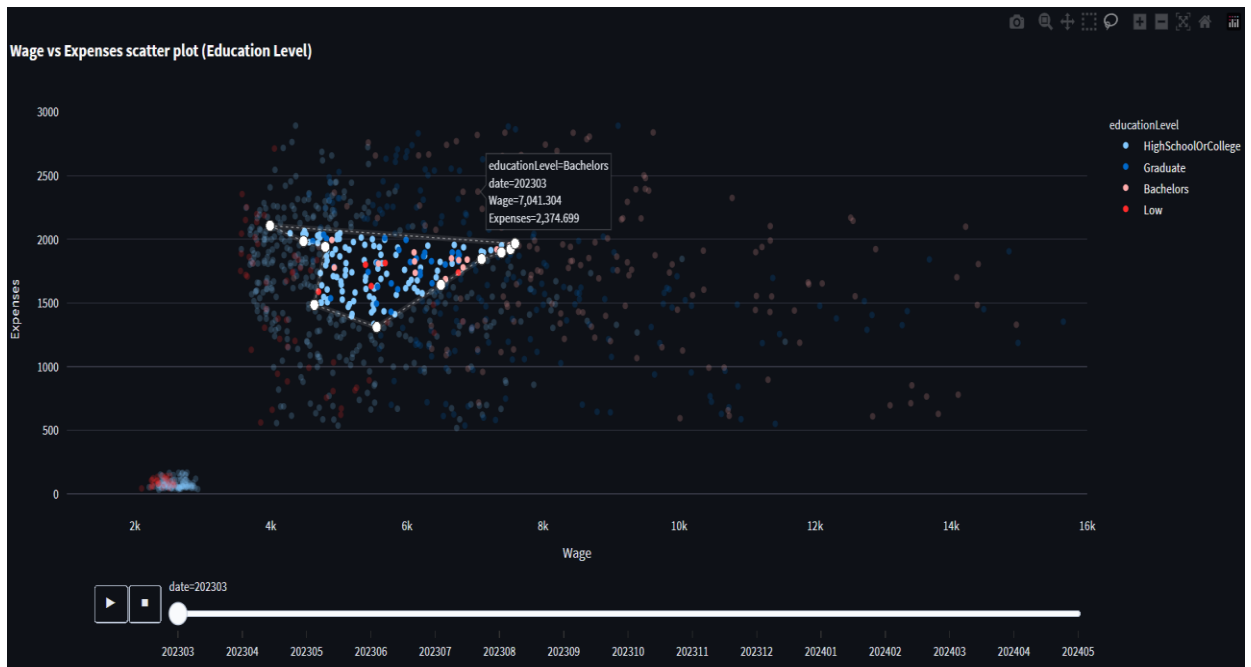


Figure 22: Wage vs. Expenses scatter plot (Education Level) with Lasso Select.

Figure 23 presents a scatter plot depicting participants' wage vs. expenses relationship, stratified by age groups. The age groups are defined as 0-18, 19-25, 26-35, 35-50, and 50-70, providing a categorical segmentation based on age. The x-axis represents wages in a specific time frame, while the y-axis represents expenses. The scatter plot is

generated using a data set including participant data with age categorization per the specified age groups.

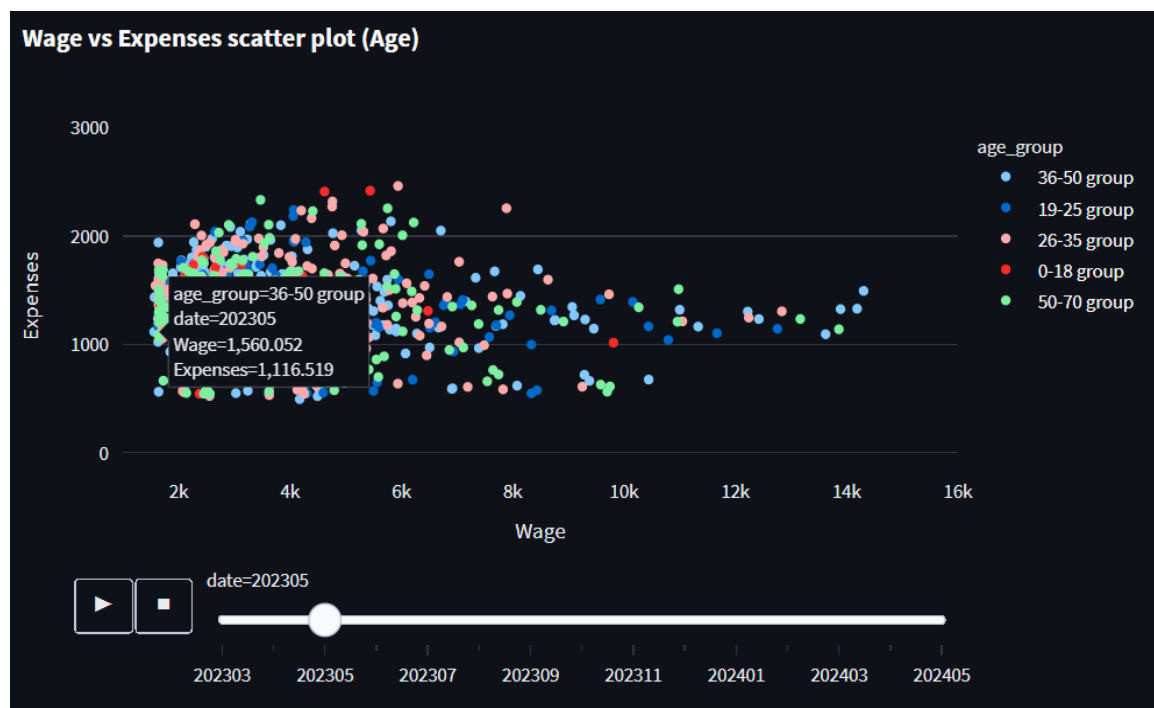


Figure 23: Wage vs. Expenses scatter plot (Age).

Figure 24 showcases an interactive data visualization feature, likely implemented using a web-based framework such as Streamlit, allowing users to select different age groups to visualize the data dynamically.

The ability to select different age groups and visualize the data dynamically allows users to gain insights into how wages and expenses vary across different age cohorts. This feature can be especially useful in identifying trends or patterns that may need to be apparent when looking at the data. By providing a more granular view of the data, users can conduct more detailed analyses.

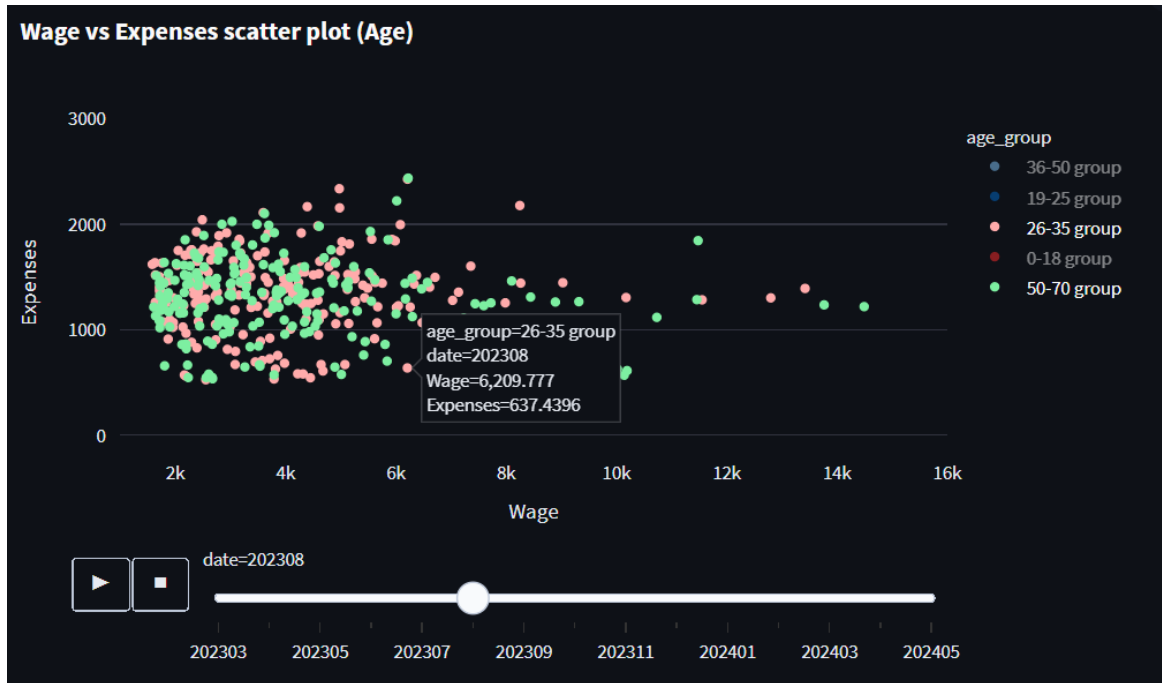


Figure 24: Wage vs. Expenses scatter plot (Age) by specific age group.

Having kids or not, this factor also affects the city's financial health as expenses increase. Calculating the expenses concerning wages is important in this scenario. Figure 25 gives the inside of this factor. It is visualized using a color scheme, with dark blue denoting participants with children and light blue denoting participants without children.

This visualization provides insights into the relationship between wages, expenses, and the presence of children, offering a comparative analysis between the two groups. In conclusion, participants with kids usually have more wages and expenses.

Figure 26 displays the scatter plot differentiating the household size of the participants. The data is scattered across all household sizes and color-coded to distinguish different dataset contents. Over time, variations in the data become apparent.

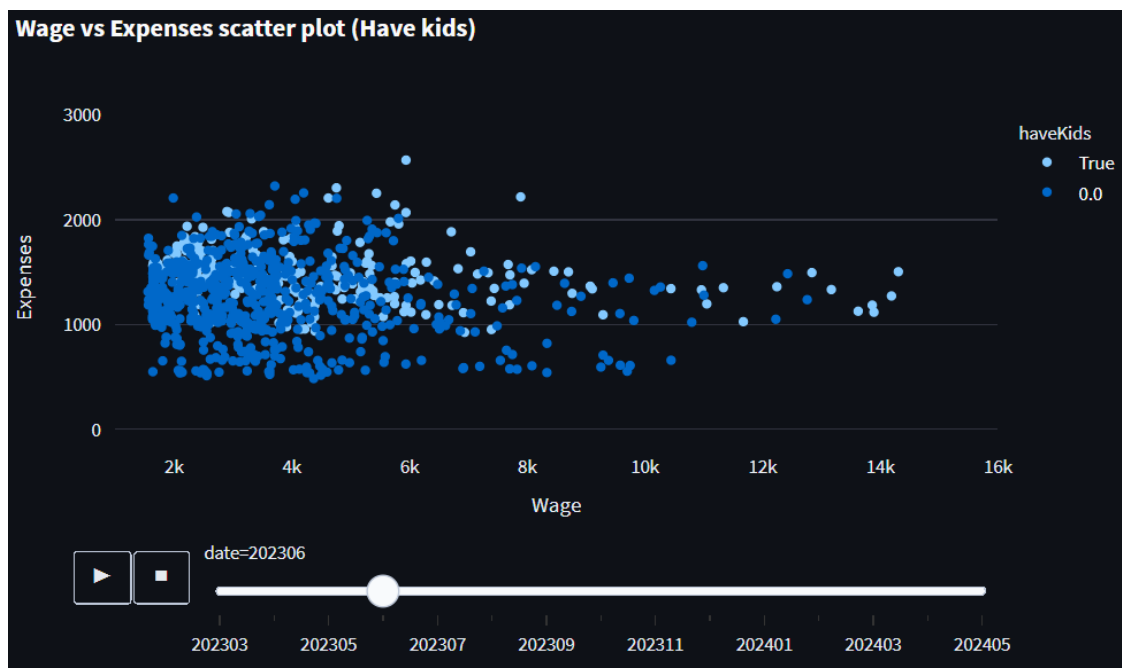


Figure 25: Wage vs. Expenses scatter plot (Have kids).

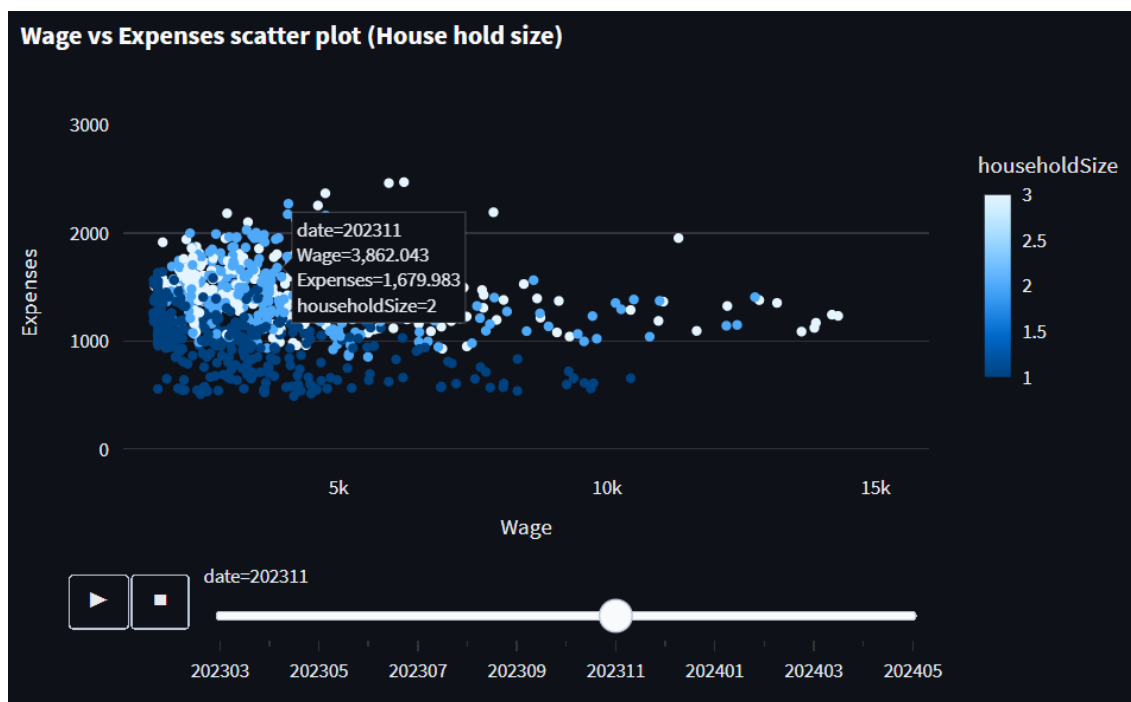


Figure 26: Wage vs. Expenses scatter plot (Household size).

Interest groups also affect the finance of any individual as the financial rules for each interest differ from others. The dataset provided information on what interest group each participant was enrolled in. In data pre-processing, I manipulated the interest group information as the data was not in numerical format. Figure 27 shows the scatter plot according to the interest group, and the sidebar shows the differentiation of each interest group from 0 to 8.

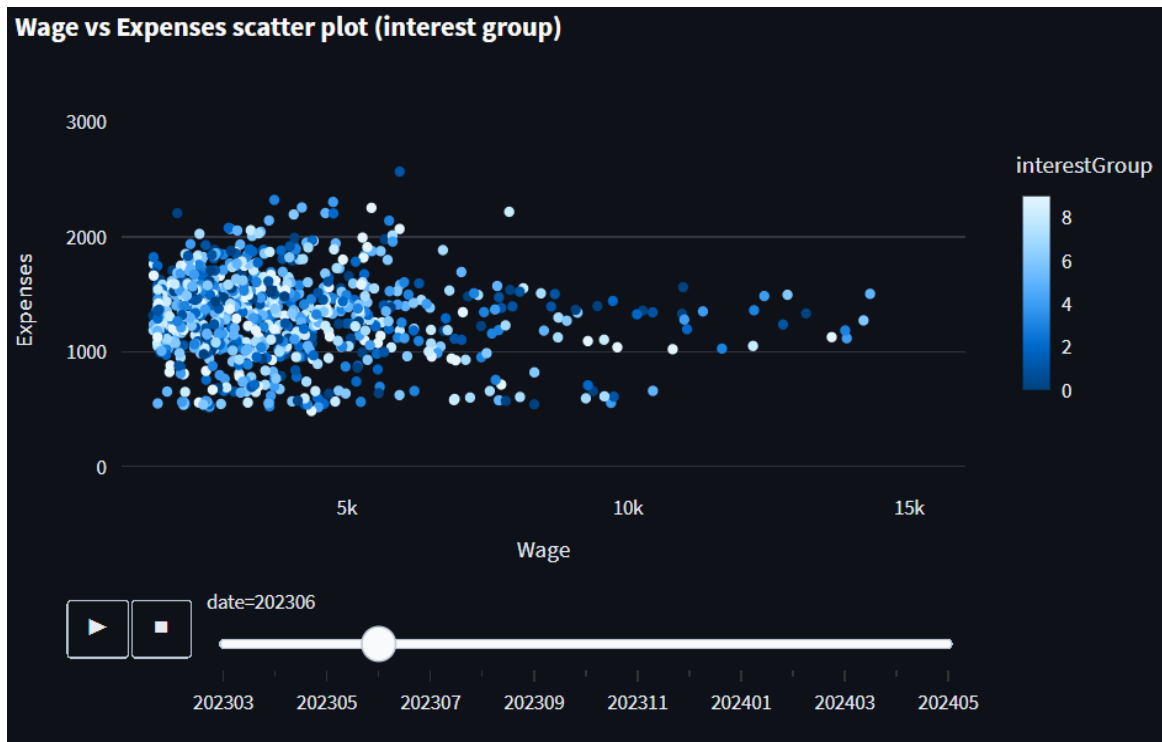


Figure 27: Wage vs. Expenses scatter plot (interest group).

Here we see that all the plots are scattered in a legitimate way of living. There are a few things to note from each plot, as below:

1. Education level - The lower the education level, the lower will be the wages, so as the expenses. However, the expenses vary to the full range regardless of their low wages.

2. Age Group - In this plot, wages and expenses are scattered, so we do not see any specific pattern. That also shows that people of all ages earn and spend money throughout the range.
3. Have Kids - Here, we see an interesting pattern which is the obvious thing. Those who do not have kids are shifted to lower wages and lower expenses and vice versa.
4. Household size - As well, the smaller the household size, the lower the expenses and wages will be.

```
# Drawing some plots based on the various categories available
fig = px.scatter(predicted, x='Wage', y='Expenses', animation_frame='date', animation_group='participantId', color='educationLevel', range_x=[1000,16000], range_y=[-100,3000], title='Wage vs Expenses scatter plot (Education Level)')
st.plotly_chart(fig, use_container_width=True)
fig = px.scatter(predicted, x='Wage', y='Expenses', animation_frame='date', animation_group='participantId', color='age_group', range_x=[1000,16000], range_y=[-100,3000], title='Wage vs Expenses scatter plot (Age)')
st.plotly_chart(fig, use_container_width=True)
fig = px.scatter(predicted, x='Wage', y='Expenses', animation_frame='date', animation_group='participantId', color='haveKids', range_x=[1000,16000], range_y=[-100,3000], title='Wage vs Expenses scatter plot (Have kids)')
st.plotly_chart(fig, use_container_width=True)
fig = px.scatter(predicted, x='Wage', y='Expenses', animation_frame='date', animation_group='participantId', color='householdSize', range_x=[1000,16000], range_y=[-100,3000], title='Wage vs Expenses scatter plot (House hold size)')
st.plotly_chart(fig, use_container_width=True)
fig = px.scatter(predicted, x='Wage', y='Expenses', animation_frame='date', animation_group='participantId', color='interestGroup', range_x=[1000,16000], range_y=[-100,3000], title='Wage vs Expenses scatter plot (Interest group)')
st.plotly_chart(fig, use_container_width=True)
```

Figure 28: Code for Wage vs. Expense plots on the various categories.

The code snippet in Figure 28 creates scatter plots using the Plotly library to visualize the relationship between the 'Wage' and 'Expenses' variables for different categories. It performs the following steps:

- Creates scatter plot with 'educationLevel' as the color parameter and sets the x-axis range to [1000,16000] and y-axis range to [-100,3000]. Adds animation based on 'date' and 'participantId' as the animation frame and animation group, respectively. Sets the plot title as "Wage vs. Expenses scatter plot (Education Level)."

- Displays the plot using Streamlit's 'st.plotly_chart()' function with the parameter 'use_container_width=True' for responsive width.
- Repeats the same process for different categories, including 'age_group,' 'haveKids,' 'householdSize,' and 'interestGroup,' with appropriate color parameters, x-axis range, y-axis range, and title for each plot.
- Plots are displayed using the 'st.plotly_chart()' function after creating each scatter plot, with the 'use_container_width=True' parameter for responsive width.

Plot showing the Wages vs Expenses of original and Curated Data

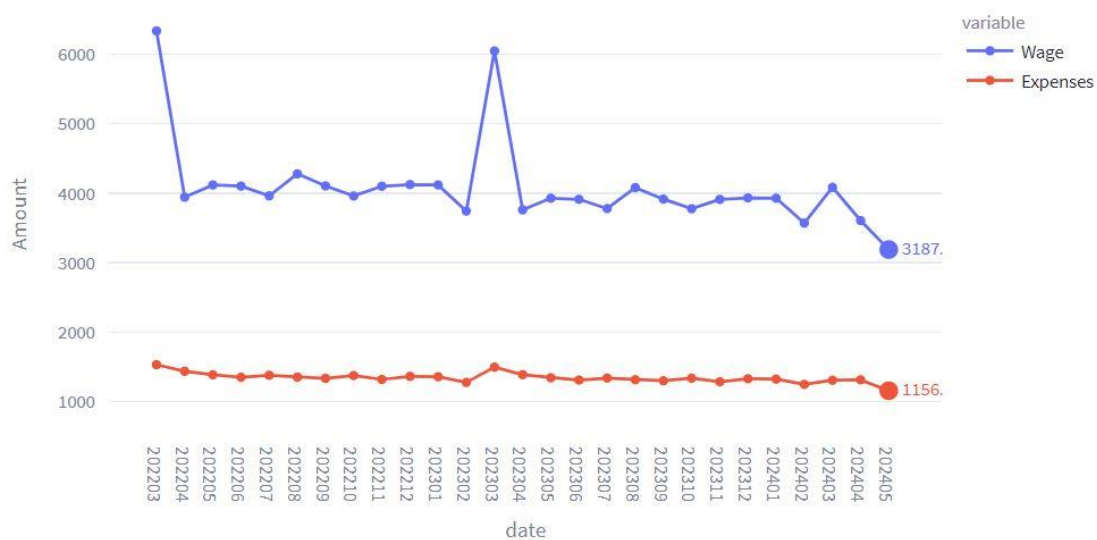


Figure 29: Plot showing the Wages vs. Expenses of original and Curated Data.

This plot gives the inside of wages vs. expenses for the data already presented in the dataset and the predicted data. All the data after 2023 is predicted, and as shown in

Figure 29, it is observed that the wages and expenses are falling in the end, which shows financial instability.

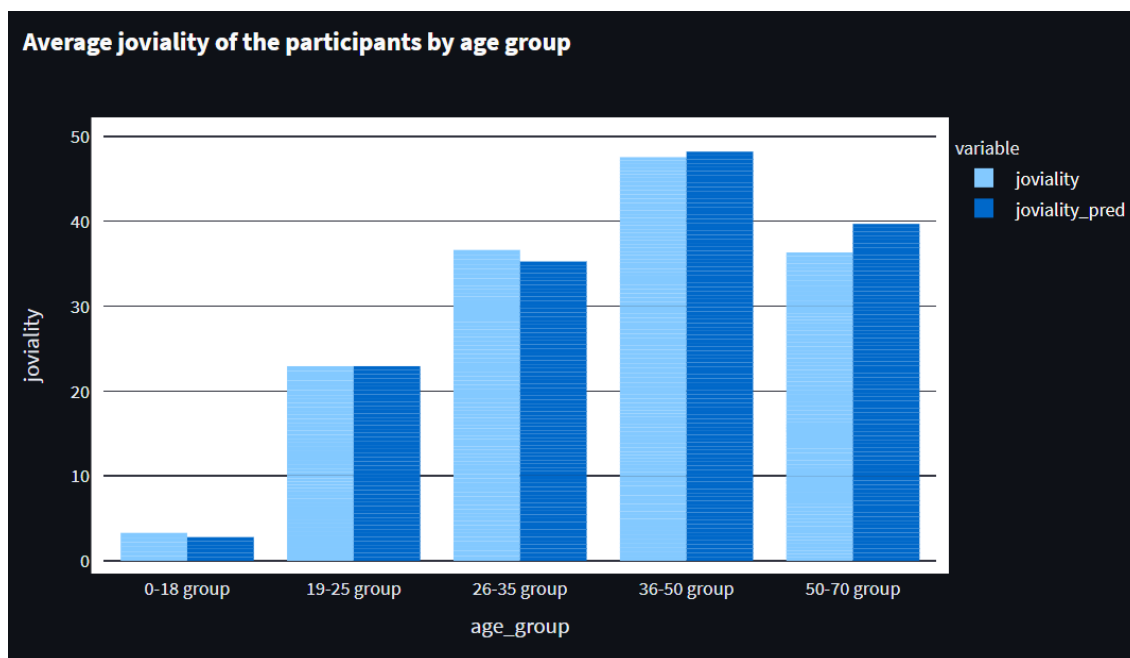


Figure 30: Average Joviality (both previous and predicted) of the participants by age group.

Figure 30 gives an inside comparison between joviality and predicted joviality. This data was obtained by the data frame created previously at the start of the cumulated section. As the above plot mentioned, the downfall of the city finance, we need to see which factors are the reason for that. By analyzing, I came across many attributes, and joviality is one of the main ones. As the participant, joviality shows if the person is happy or not. To differentiate the data more. This visual gives the inside concerning age group. As observed, age groups 0-18, 19-25 and 26-35 have a downfall the joviality, which means people are unhappy till their mid-adulthood. Age groups 36-50 and 50-70 have an up fall for joviality in the following year. So, participants tend to be happy after a certain age.

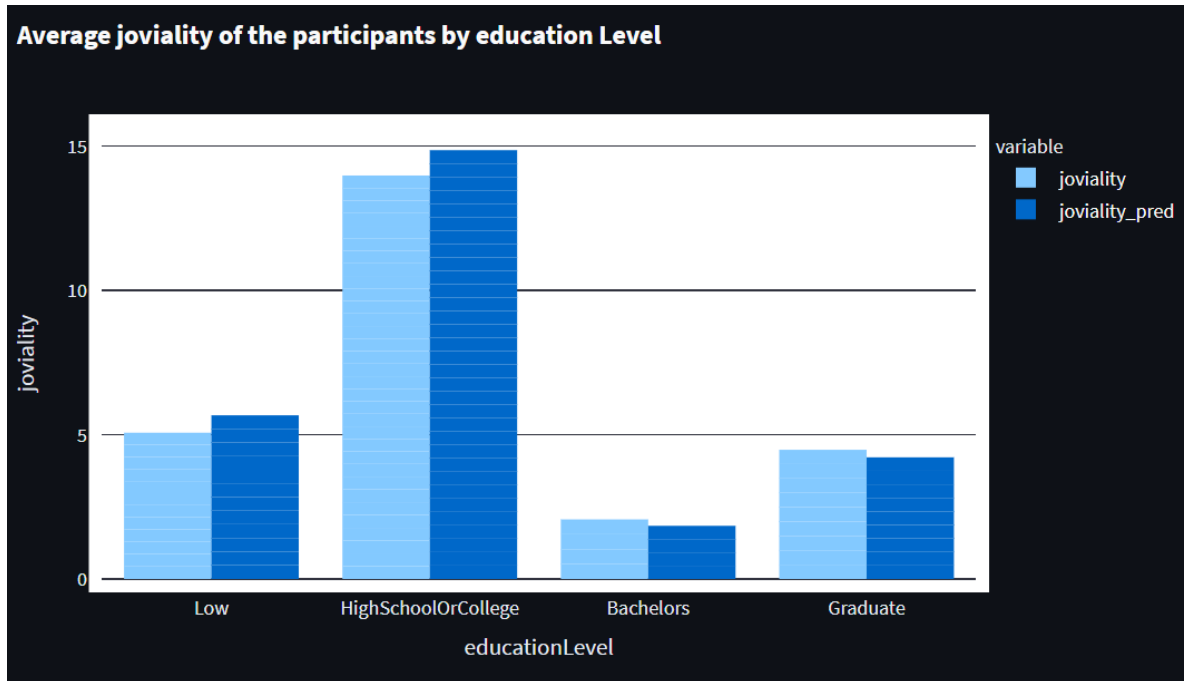


Figure 31: Average Joviality (both previous and predicted) of the participants by education Level.

Same as the previous visual, Figure 31 compares joviality and predicted joviality. The city's finance could be more stable in the following years, and many factors caused that. One of the attributes that affect this situation is the education level of the participants. As Jobs in the city are specialized for certain education levels, having fewer employees can affect the overall turnover of the city. What happened here? In the figure, it is detected that the average joviality of "Low" and "Highschool or college" are increasing as job opportunities for them are more than "Bachelor" and "Graduate."

This concludes that the education of the participants is mostly below bachelor's and graduate, which directly affects the city's growth.

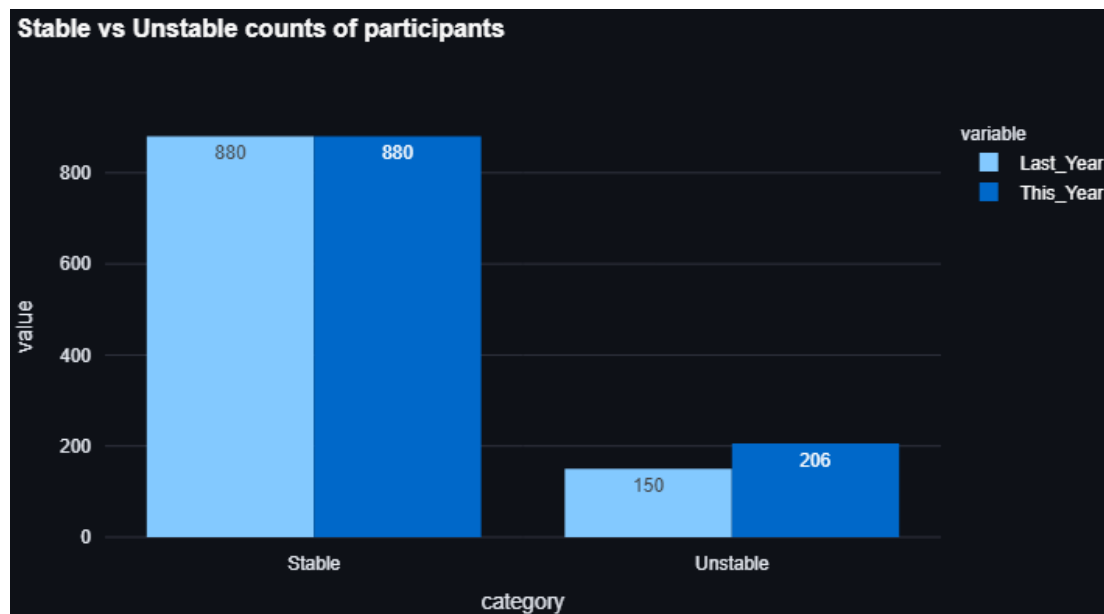


Figure 32: Stable and Unstable counts of participants (Last and This year).

```
fdm = tom_df[['participantId', 'financialStatus']].dropna()
fdm = fdm.drop_duplicates()
cnt_ly = fdm['financialStatus'].value_counts()

dds = final_old_data[['participantId', 'financialStatus']].drop_duplicates()
cnt_ty = dds['financialStatus'].value_counts()

vcfd = pd.DataFrame({'category': ['Stable', 'Unstable'], 'Last_Year': cnt_ly.to_list(), 'This_Year': cnt_ty.to_list()})
fig = px.bar(vcfd, x='category', y=['Last_Year', 'This_Year'], bar_mode='group', text_auto=True, title='Stable vs Unstable counts of participants')
st.plotly_chart(fig, use_container_width=True)
```

Figure 33: Stable vs. Unstable counts of participants code snippet.

Figure 32 plots the difference in participants' stable and unstable status last year and this year. The code snippet presented in Figure 33 plots the count data mentioned above. The dataset provides the last year's data, and This year's data is calculated by the prediction model described previously in the dashboard. The value supplied is the total count of financial status.

The project's outcome is to analyze the financial status of the participant in the following year. Moreover, in this visual, what is happening financially in the city is visible. It gives an inside stable and unstable participant prediction in the following year. The number of stable participants is the same, but the unstable number is increasing, which can predict that the city is not progressing in financial health.

4.3.5. Travel Journal Visualizations:

The final section of the dashboard provides insights into the travel journal, which includes travel time and overall spending. The travel time is measured in minutes and has consistently decreased from April 2022 to April 2023 across all aspects illustrated in Figure 34. The purpose of travel is categorized in the sidebar as Coming Back from a Restaurant, Eating, Going Back home, Recreation (Social Gathering), and Work/Home Commute. The data indicate that Work/Home Commute takes significantly longer than other purposes. Towards the end of the plot, it is evident that participants have either reduced their travel or ceased traveling altogether. The most significant change over time is observed in social gatherings and going back home, suggesting that participants may have decreased their social activities and stayed home due to financial constraints.

Visuals depicting the travel journal provide evidence of the city's financial struggles. Figure 35 displays spending in terms of travel time, with a noticeable decline in data flow over time, directly reflecting the city's financial status. Similarly, as seen in the previous visual, the spending on social gatherings has also decreased, indicating a decline in social activities. The only aspect that has shown an increase is the purpose of returning home, suggesting that participants stay home more frequently over the observed period.

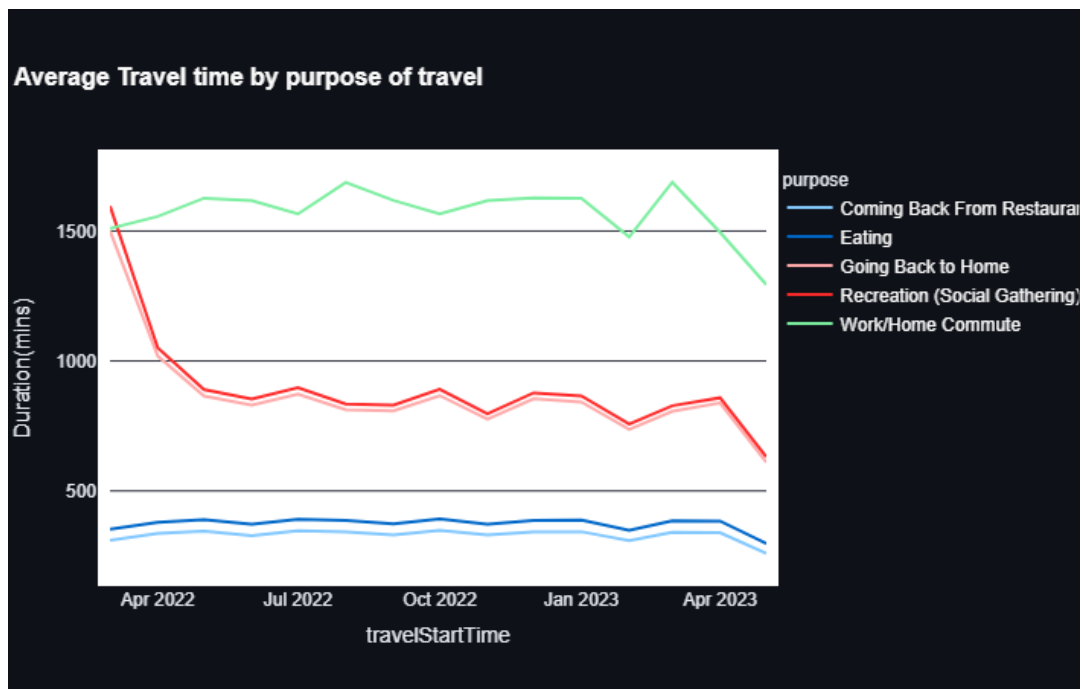


Figure 34: Average travel time by purpose of travel over time of dataset.

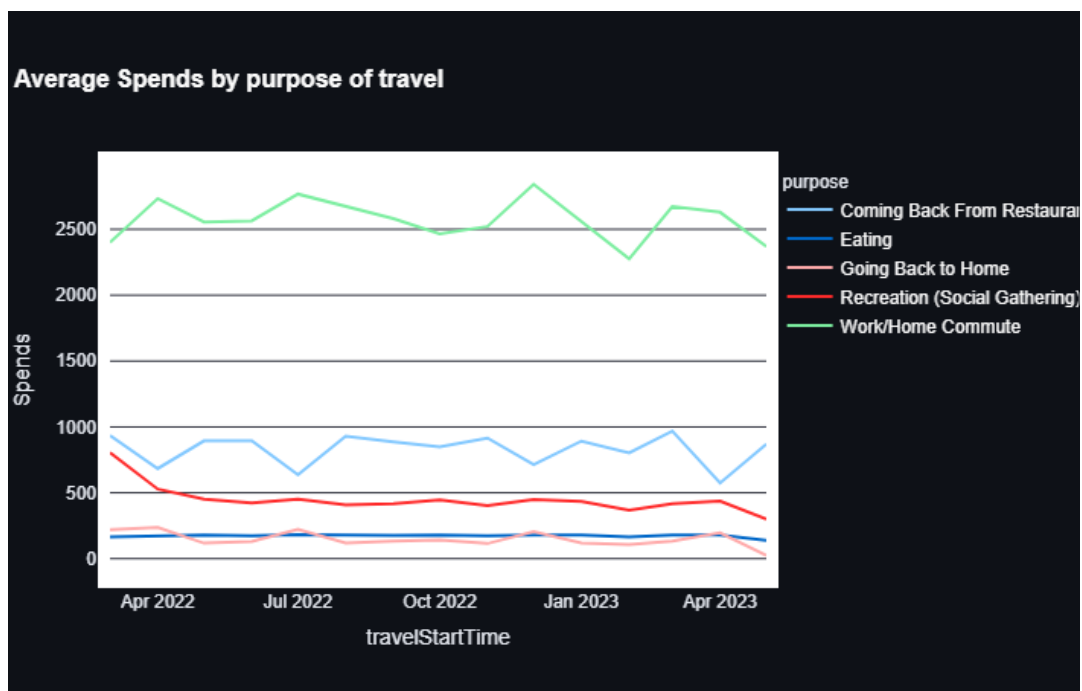


Figure 35: Average spends by purpose of travel over time of dataset.

These findings collectively support the conclusion that the city is facing financial challenges, as evident from the decreasing travel time, reduced spending on social gatherings, and increased time at home. The data suggest that participants have curtailed their travel and social activities, likely due to financial concerns, and have been spending more time at home.

Chapter 5: FUTURE WORK

In future work, several potential areas could be explored in more detail to enhance the project's contributions and findings to the field of study.

One potential avenue for further investigation is analyzing wages on a regional basis using geo-plotting techniques. It could involve obtaining additional wage data in different areas or regions and visualizing it on a map to identify spatial patterns or trends. For example, it is valuable to calculate and compare the average wages in different neighborhoods or communities to identify areas requiring more attention regarding financial status. By conducting a spatial analysis of wages, the project could provide insights into the distribution of income levels and highlight areas that may require targeting or policy interventions.

Another aspect that could be explored in future work is the issue of overfitting in machine learning models. Overfitting occurs when a model learns to perform but fails to generalize the training data well to unseen data. It can lead to inaccurate predictions and reduced model performance. Further research could be conducted to explore different techniques to mitigate overfitting. The current project has utilized Random Forest modeling for regression and classifier tasks, but numerous other machine-learning algorithms could be explored in future work. For instance, techniques such as neural networks could be implemented and compared with the current models to determine their effectiveness in predicting financial status and joviality.

It is important to record that machine learning is rapidly evolving, and staying updated with the latest research and advancements is necessary. Future work in the project could involve keeping up to date with the latest techniques, algorithms, and methodologies in machine learning and incorporating them into the project to improve the accuracy and reliability of the models.

The future work highlights potential areas for further exploration, including analyzing regional wages, addressing overfitting in machine learning models, and exploring different machine learning algorithms. It emphasizes the need for continuous learning, research, and incorporation of the latest advancements to enhance the project's outcomes and extend the field of study.

Chapter 6: CONCLUSION

In conclusion, using data visualization and machine learning models holds significant potential for predicting the financial trajectory of cities. These models can provide valuable insights by applying sophisticated algorithms and analyzing historical data. While developing various visualization, I encountered challenges and identified how each attribute could be linked to and enhance visualization. The dataset utilized in this project provides a comprehensive view of the participants' lives from the city's perspective, allowing for a holistic understanding of the city's development across various domains.

Furthermore, data visualization can play a crucial role in evaluating and interpreting the results of machine learning models, enabling investors to gain insights into the key variables influencing financial patterns and changes. The validity and reliability of the training and validation data significantly impact these models' effectiveness. Visualization techniques can also enhance the accuracy and dependability of the models by facilitating the identification of outliers, anomalies, and potential biases in the data.

During this project, I gained valuable insights into the complexities related to selecting a suitable machine-learning model, including the challenge of addressing overfitting. The accuracy of the final model was rigorously validated through comparison with other models using the same dataset. I also understand the significance of meticulous data source selection, thorough data cleaning, and thoughtful feature engineering in ensuring the reliability of predictions and subsequent decision-making processes.

This project results suggest that the city under study is not experiencing growth, and the standard of living may be declining. The city should reconsider its planning for the recently received renewal grant based on the data analysis. Integrating machine learning models with data visualization can significantly aid cities in predicting and planning for their financial future, facilitating improved resource allocation, financial management, and sustainable urban development. These models should undergo continuous research, improvement, and validation to fully realize their potential and ensure their applicability in real-world scenarios.

References:

- [1] Estrada, Elsa, et al. "Smart City Visualization Tool for the Open Data Georeferenced Analysis Utilizing Machine Learning." *International Journal of Combinatorial Optimization Problems and Informatics*, vol. 9, 2018, pp. 25-40.
- [2] Rojas, E., Bastidas, V., & Cabrera, C. "Cities-Board: A Framework to Automate the Development of Smart Cities Dashboards." *IEEE Internet of Things Journal*, vol. 7, no. 10, 2020, pp. 10128-10136. doi:10.1109/JIOT.2020.3002581.
- [3] Jeffery, Clinton. "The City Metaphor in Software Visualization." 2019, doi:10.24132/CSRN.2019.2901.1.18.
- [4] Somisetty, AG. "Inculcating Ethics in Data Visualization Dashboards." California State University, Sacramento, 2022, p. 71.
- [5] Pandey, V., Manivannan, A., Nov, O., Satterthwaite, M., & Bertini, E. "The Persuasive Power of Data Visualization." *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, 2014, pp. 2211-2220. doi:10.1109/TVCG.2014.2346419.
- [6] Kirk, A. "Data Visualization: A Successful Design Process; A Structured Design Approach to Equip You with the Knowledge of How to Successfully Accomplish Any Data Visualization Challenge Efficiently and Effectively." 1st edition, Packt Pub., 2012.

- [7] Wettel, R., & Lanza, M. "Visualizing Software Systems as Cities." Proceedings of the 2007 4th IEEE International Workshop on Visualizing Software for Understanding and Analysis, 2007, pp. 92-99.
- [8] Bao, Fan, and Jia Chen. "Visual framework for big data in d3.js." Proceedings of the 2014 IEEE Workshop on Electronics, Computer and Applications, 2014, pp. 47-50. doi:10.1109/IWECA.2014.6845553.
- [9] Baviskar, N. "Smart City Development Using Data Analytics." 2017.
- [10] Stančin, Igor, and Alan Jović. "An overview and comparison of free Python libraries for data mining and big data analysis." 2019 42nd International convention on information and communication technology, electronics and microelectronics (MIPRO). IEEE, 2019.
- [11] Harris, Charles R., et al. "Array programming with NumPy." *Nature* 585.7825 (2020): 357-362.
- [12] Sial, Ali Hassan, Syed Yahya Shah Rashdi, and Abdul Hafeez Khan. "Comparative analysis of data visualization libraries Matplotlib and Seaborn in Python." *International Journal* 10.1 (2021).
- [13] Kang, M., & Choi, E. "Machine Learning: Concepts, Tools and Data Visualization." World Scientific, 2021.
- [14] Gumelar, A. B. "An Anatomy of Machine Learning Data Visualization." Proceedings of the 2019 International Seminar on Application for Technology of Information and Communication (iSemantic), IEEE, 2019, pp. 1-6. doi:10.1109/ISEMANTIC.2019.8884340.

- [15] Kluyver, Thomas, et al. Jupyter Notebooks-a publishing format for reproducible computational workflows. Vol. 2016. 2016.
- [16] Atitallah, SB, Driss, M, Boulila, W, Ghézala, HB. "Leveraging Deep Learning and IoT big data analytics to support the smart cities development: Review and future directions." Computer science review, vol. 38, 2020, pp. 100303-. doi:10.1016/j.cosrev.2020.100303
- [17] Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." the Journal of Machine Learning research 12 (2011): 2825-2830.
- [18] Breiman, Leo. "Random forests." Machine learning 45 (2001): 5-32.
- [19] Virtanen, Pauli, et al. "SciPy 1.0: fundamental algorithms for scientific computing in Python." Nature methods 17.3 (2020): 261-272.
- [20] McKinney, Wes. "Data structures for statistical computing in python." Proceedings of the 9th Python in Science Conference. Vol. 445. No. 1. 2010.
- [21] Sudalaimani, C., et al. "Automated detection of the preseizure state in EEG signal using neural networks." biocybernetics and biomedical engineering 39.1 (2019): 160-175
- [22] Jaillot, V., Rigolle, V., Servigne, S., Samuel, J., & Gesquière, G. "Integrating Multimedia Documents and Time-Evolving 3D City Models for Web Visualization and Navigation." Transactions in GIS, vol. 25, no. 3, 2021, pp. 1419-1438. doi:10.1111/tgis.12734

- [23] J Johansson, T., Segerstedt, E., Olofsson, T., & Jakobsson, M. "Revealing Social Values by 3D City Visualization in City Transformations." *Sustainability*, vol. 8, no. 2, 2016, pp. 195. <https://doi.org/10.3390/su8020195>
- [24] Michael Correll. 2019. "Ethical Dimensions of Visualization Research." *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, Paper 188, 1–13. DOI: <https://doi.org/10.1145/3290605.3300418>
- [25] Souza, J, Francisco, A, Piekarski, C, Prado, G. "Data Mining and Machine Learning to Promote Smart Cities: A Systematic Review from 2000 to 2018." *Sustainability (Basel, Switzerland)*, vol. 11, no. 4, 2019, pp. 1077-. doi:10.3390/su11041077
- [26] Ali, M. "Big Data and Machine Intelligence in Software Platforms for Smart Cities." In: *Software Architecture*. Springer International Publishing, 2020, pp. 17-26. doi:10.1007/978-3-030-59155-7_2
- [27] Chen, Q, Wang, W, Wu, F, et al. "A Survey on an Emerging Area: Deep Learning for Smart City Data." *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 3, no. 5, 2019, pp. 392-410. doi:10.1109/TETCI.2019.2907718
- [28] Shams, S, Goswami, S, Lee, K, Yang, S, Park, SJ. "Towards Distributed Cyberinfrastructure for Smart Cities Using Big Data and Deep Learning Technologies." In: *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2018, pp. 1276-1283. doi:10.1109/ICDCS.2018.00127

- [29] Wang, Q, Chen, Z, Wang, Y, Qu, H. "A Survey on ML4VIS: Applying Machine Learning Advances to Data Visualization." *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 12, 2021, pp. 5134-5153. doi:10.1109/TVCG.2021.3106142
- [30] Steinbrückner, F, Lewerentz, C. "Representing Development History in Software Cities." In: *Proceedings of the 5th International Symposium on Software Visualization*. ACM, 2010, pp. 193-202. doi:10.1145/1879211.1879239
- [31] Nair, L, Shetty, S, & Shetty, S. "Interactive Visual Analytics on Big Data: Tableau vs D3.js." *Journal of e-Learning and Knowledge Society*, vol. 12, no. 4, 2016, pp. Italian eLearning Association. Retrieved November 12, 2021.
- [32] Ardigò, S., Nagy, C., Minelli, R., & Lanza, M. "M3triCity: Visualizing Evolving Software & Data Cities." *arXiv.org*, Cornell University Library, *arXiv.org*, 2022. doi:10.1145/3510454.3516831
- [33] Adugna, Tesfaye, Wenbo Xu, and Jinlong Fan. "Comparison of random forest and support vector machine classifiers for regional land cover mapping using coarse resolution FY-3C images." *Remote Sensing* 14.3 (2022): 574.
- [34] Rojas, E., Bastidas, V., & Cabrera, C. "Cities-Board: A Framework to Automate the Development of Smart Cities Dashboards." *IEEE Internet of Things Journal*, vol. 7, no. 10, 2020, pp. 10128-10136. doi:10.1109/JIOT.2020.3002581.
- [35] "Workflow Basics," [Online]. Available: <https://docs.trifacta.com/display/SS/Workflow+Basics>.
- [36] "D3.js Graph Gallery," [Online]. Available:

<https://d3-graph-gallery.com/>.

[37] “Getting started with Power Bi,”[Online].Available:

<https://powerbi.microsoft.com/en-us/getting-started-with-power-bi/>.

[38] “Introduction to Tensor Flow,”[Online].Available:

<https://www.tensorflow.org/learn>.