jims
Sector-5, Rohini, Delhi

PROJECT REPORT
ON

DATA ANALYSIS USING PYTHON AND TABLEAU

SUBMITTED IN PARTIAL FULFILLMENT FOR THE AWARD OFTHE DEGREE OF BACHELOR OF COMPUTER APPLICATIONS 2020-2023

SUBMITTED TO:
Dr. Isha Singh
(Assistant Professor, JIMS)

SUBMITTED BY:
Parul Mehra
Roll No: 03114002020

Jagan Institute of Management Studies, Rohini Sector 5

Jagan Institute of Management Studies

Sector-5, Rohini, Delhi-110085

Certificate

This is to certify that Parul Mehra, student of BCA 1st Shift bearing Enrollment No 03114002020 has undertaken a Major Project titled "Data analysis using python and tableau" completed under my supervision and guidance in partial fulfilment ofthe requirement for the award of degree of Bachelor of Computer Applications(BCA) as a part of the curriculum in Guru Gobind Singh Indraprastha University, New Delhi-110078.

To the best of my knowledge and belief, the data and information presented bythe student in the Major Project has not been submitted earlier.

Signature of the Faculty Guide:

Name : ……

Date : ……

ACKNOWLEDGEMENT

I Parul Mehra the student of BCA (3rd year), am extremely grateful to the Jagan Institute of Management Studies, Rohini for the confidence bestowed in me and entrusting our Practical file of Major Project.

At this juncture, I feel deeply honored in expressing my sincere thanks to my faculty Dr. Isha Singh for her valuable input, her guidance, encouragement, whole-hearted cooperation, and constructive criticism throughout the preparation of this practical file.

Parul Mehra (BCA 1st Shift)

Date: ……

Jagan Institute of Management Studies

INDEX

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Objective of the System

1.2 Justification and need for the system

1.3 Advantage of the system

1.4 Previous work or related systems, how they are used.

Chapter 1- Introduction

1.1 OBJECTIVE OF THE SYSTEM

This project aims to deduce results from a given data sets of 3 different corporations namely , value inc , blue bank and blog me. Where there would be a sales analysis for the Value inc corporation , and a loan analysis for blue bank and sentiment analysis for the company blog me

1.2 JUSTIFICATION AND NEED FOR THE SYSTEM

1. As famously said by Michael palmer in year 2006, “Data is the new oil”. Data is the new age’s boon as well as bane. In today’s world 3.5 quintillion bytes of data is generated in every world, so we need an effective and efficient way to manage, store and deduce results from the data.

2 Data analytics is the science of analyzing raw data to make conclusions about that information. Many of the techniques and processes of data analytics have been automated into mechanical processes and algorithms that work over raw data for human consumption.

3 Data analytics is a broad term that encompasses many diverse types of data analysis. Any type of information can be subjected to data analytics techniques to get insight that can be used to improve things. Data analytics techniques can reveal trends and metrics that would otherwise be lost in the mass of information. This information can then be used to optimize processes to increase the overall efficiency of a business or system

1.3 ADVANTAGES OF THE SYSTEM

1 **Powerful Data Processing**: Python is a versatile programming language with extensive libraries and frameworks for data analysis and manipulation. It provides powerful tools like NumPy, Pandas which enable efficient data processing and manipulation.

2 **Data Cleaning and Transformation:** Python offers a wide range of libraries for data cleaning and transformation, allowing you to handle missing values, perform data imputation, merge datasets, and transform data into the desired format. This ensures that your data is clean and ready for analysis.

3 **Advanced Analytics and Machine Learning:** Python has a rich ecosystem of libraries such as PyTorch, which provide advanced analytics and machine learning capabilities. You can perform tasks like predictive modeling, clustering, classification, and regression to gain deeper insights from your data.

4 **Customization and Extensibility:** Python is highly customizable and allows you to create custom functions, algorithms, and models tailored to your specific requirements. You can leverage the vast open-source community and access a wide range of pre-built solutions to accelerate your data analytics workflows.

5 **Interactive Data Visualization:** Tableau is a powerful data visualization tool that enables you to create interactive and visually appealing dashboards, reports, and charts. You can connect Python scripts directly to Tableau, allowing you to integrate Python's analytical capabilities with Tableau's visualization strengths.

6 **Real-time Analytics:** Python and Tableau can be used together to perform real-time

analytics. Python can process and analyze streaming data, and the results can be visualized in real-time using Tableau, providing up-to-date insights for making timely decisions.

7 **Scalability and Performance:** Python's scalability and performance make it suitable for handling large datasets and complex analytical tasks. With optimized libraries like Dask and Apache Spark, you can distribute computations across clusters, enabling efficient processing of big data.

8 **Collaboration and Sharing**: Both Python and Tableau offer collaboration and sharing capabilities. Python code can be easily shared and collaborated upon using version control systems like Git, while Tableau allows you to publish interactive dashboards and reports for sharing with stakeholders.

9 **Reproducibility:** Python's scripting capabilities enable reproducible data analysis. You can document and share your analysis workflows as code, ensuring that others can reproduce your results and providing transparency in your analytics process.

10 **Cost-Effective Solution:** Python is an open-source language, and Tableau offers a range of licensing options, including a free version. This combination makes data analytics using Python and Tableau a cost-effective solution compared to some other proprietary software

1.4 PREVIOUS WORK OR RELATED SYSTEMS, HOW THEY ARE USED

1.4.1 Data Collection: Analysts would manually collect data from various sources, such as surveys, paper forms, spreadsheets, and databases. This process involved

physically gathering or requesting data from different departments or individuals.

1.4.2 Data Entry: The collected data would then be manually entered into spreadsheets or databases. This step required meticulous attention to detail to ensure accurate data entry, as errors could lead to misleading or incorrect analysis.

1.4.3 Data Cleaning: Data cleaning involved identifying and correcting errors, inconsistencies, and missing values in the collected data. Analysts would manually review the data, identify anomalies, and make necessary adjustments to ensure data quality.

1.4.4 Data Exploration: Analysts would manually review and explore the data using spreadsheets, calculators, and statistical tables. They would calculate summary statistics, compute averages, perform simple calculations, and visually inspect the data to identify patterns or outliers.

1.4.5 Statistical Analysis: Statistical analysis was performed manually using mathematical formulas and statistical techniques. Analysts would calculate measures of central tendency, dispersion, and correlation coefficients. They would also conduct hypothesis testing and interpret results based on their knowledge of statistical theory.

1.4.6 Visualization: Visualization was often done by manually creating charts, graphs, and diagrams using tools like graph paper, rulers, compasses, and colored pencils.

1.4.7 Analysts would carefully plot data points, draw trend lines, and label axes to visually represent the findings.

CHAPTER 2

PROJECT DESCRIPTION

2.1 Analysis Study

2.2 User Requirements

2.3 Final Requirements

Chapter 2- REQUIREMENT ANALYSIS

2.1 ANALYSIS STUDY

Topic 1: Sales Analysis for Value Inc

Value Inc is a retail store that sells household items all over the world by bulk. The Sales Manager has no sales reporting but he has a brief idea of current sales. He also has no idea of the monthly cost, profit and top selling products. He wants a dashboard on this and says the data is currently stored in an excel sheet.

Topic 2: Blue Bank Loan Analysis

Blue Bank is a bank in USA that has a loan department which is currently understaffed. They supply loans to individuals and don't have much reporting on how risky these borrowers are. Using Python and Tableau, they'd like to see a report of borrowers who may have issues paying back the loan

Topic 3: BlogMe Sentiment and Keyword Analysis

BlogMe, a famous blogging business has a dataset of news articles that they need further analysis on. Firstly, they'd like keywords to be extracted from headlines of the article. Secondly, they would need to determine the sentiment of the news articles. The data is in an excel sheet and they would like to see a dashboard outlying sentiment, top articles etc

2.2 USER REQUIREMENTS

Topic 1: Sales Analysis for Value Inc

The Sales Manager has no sales reporting but he has a brief idea of current sales. He also has no idea of the monthly cost, profit and top selling products. He wants a dashboard on this and says the data is currently stored in an excel sheet.

1 To find profit according to the corresponding item code

2 To find monthly sales taking place inside the corporation.

3 To find profit according to the country the goods are sold to.

4 To find profit according to the diversity of clients Value Inc has

5 To find profit according to the diversity of client's age Value Inc has

6 Generate an interactive dashboard for the above aims

Topic 2: Blue Bank Loan Analysis

They supply loans to individuals and don't have much reporting on how risky these borrowers are. Using Python and Tableau, they'd like to see a report of borrowers who may have issues paying back the loan.

1. To find loans which have high interest rates and low interest rates
2. To find tentative number of loans that are given out by Blue Bank according to the credit score of the person

3. To find number of loans according to the extent till which they have been utilized
4. To find number of loans by the income group. For example people having income <10k, between 10k and 50k, etc.
5. Generate an interactive dashboard for the above aims

Topic 3: BlogMe Sentiment and Keyword Analysis

Firstly, they'd like keywords to be extracted from headlines of the article. Secondly, they would need to determine the sentiment of the news articles. The data is in an excel sheet and they would like to see a dashboard outlying sentiment, top articles etc

1. To find the engagement of the article according to the year in which it was published , the main of engagement is to have an overview of the comment engagement , the reaction engagement and the engagement by sharing the article.
2. To find the total engagement of articles that mention " murder" keyword.
3. To find total engagement according to the sources
4. To find the top negative sentiment titles that have maximum engagement
5. To find the top positive sentiment titles that have maximum engagement
6. Generate an interactive dashboard for the above aims

2.3 FINAL REQUIREMENTS OF ALL PROJECTS

- To find profit according to the corresponding item code
- To find monthly sales taking place inside the corporation.

- To find profit according to the country the goods are sold to.
- To find profit according to the diversity of clients Value Inc has
- To find profit according to the diversity of client's age Value Inc has
- Generate an interactive dashboard for the above aims
- To find loans which have high interest rates and low interest rates
- To find tentative number of loans that are given out by Blue Bank according to the credit score of the person
- To find number of loans according to the extent till which they have been utilized
- To find number of loans by the income group. For example people having income <10k, between 10k and 50k, etc.
- Generate an interactive dashboard for the above aims
- To find the engagement of the article according to the year in which it was published , the main of engagement is to have an overview of the comment engagement , the reaction engagement and the engagement by sharing the article.
- To find the total engagement of articles that mention " murder" keyword.
- To find total engagement according to the sources
- To find the top negative sentiment titles that have maximum engagement
- To find the top positive sentiment titles that have maximum engagement
- Generate an interactive dashboard for the above aims
- Above are cumulated requirements for all the three companies

CHAPTER 3

DESIGN OF THE SYSTEM

3.1 Hardware and Software Requirements

3.2 System Requirements

3.3 Detailed System Specification (Module Wise)

3.4 DFDs/Flow Chart/ER diagram/Use case/Activity Diagram/Sequence Diagram(Topic Wise)

3.5 Blueprints

Chapter 3- DESIGN OF THE SYSTEM

3.1 HARDWARE AND SOFTWARE REQUIREMENTS

3.1.1 Hardware Requirements

- Processor 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz
- Installed RAM 16.0 GB (15.8 GB usable)
- Device ID 23B1D2EB-D0A7-4AC5-A05B-A7DB63D7178C
- Product ID 00327-35936-96021-AAOEM
- System type 64-bit operating system, x64-based processor
- Pen and touchNo pen or touch input is available for this display

3.1.2 Software Requirements

- Chrome
- Spyder
- Tableau Public
- Excel
- Anaconda

3.2 SYSTEM REQUIREMENTS Operating System

- Windows 7/10/11

3.3 DETAILED SYSTEM SPECIFICATION

2.3.1 Anaconda is a popular open-source distribution platform for Python and R programming languages, widely used in data science, machine learning, and scientific computing. It provides a comprehensive package management system and an extensive collection of pre-installed libraries, making it easy to set up and manage data analysis environments. . It also includes popular data analysis libraries such as NumPy, Pandas, Matplotlib, and scikit-learn. Anaconda supports multiple operating systems and offers a user-friendly interface, Anaconda Navigator, which provides a graphical interface for managing environments, installing packages, and launching applications. With Anaconda, data scientists and analysts can quickly create reproducible and scalable data analysis environments, allowing them to focus on their analysis tasks rather than worrying about software dependencies and configurations

2.3.2 Tableau Public is a free and powerful data visualization tool that allows users to create, publish, and share interactive visualizations on the web. It is a cloud-based platform that enables individuals, organizations, and journalists to tell compelling stories with their data. With Tableau Public, users can connect to various data sources, including spreadsheets, databases, and web services, to import and analyze their data. The platform provides a user-friendly drag-and-drop interface, allowing users to easily create interactive charts, graphs, maps, and dashboards without any coding or technical expertise. Tableau Public offers a wide range of visualization options, customization features, and interactive capabilities, such as filtering, highlighting, and tooltips. Once visualizations are created, they can be published to the Tableau Public server and embedded in websites, blogs, or shared through social media. Additionally, Tableau Public encourages collaboration and exploration by allowing users to explore and interact with other publicly shared visualizations, fostering a vibrant and active community of data enthusiasts.

2.3.3 Spyder is an open-source integrated development environment (IDE) designed specifically for scientific computing and data analysis in Python. It provides a user-friendly and efficient environment for writing, debugging, and executing Python code. Spyder offers a rich set of features tailored for scientific programming, including advanced code editing capabilities, interactive console, variable explorer, and integrated help documentation. With its multi-window interface, users can easily navigate between code editors, console, and file explorers, enabling smooth workflow management. Spyder also supports various scientific libraries and tools, such as NumPy, Pandas, Matplotlib, and IPython, making it a powerful environment for data analysis tasks. It includes features like code autocompletion, syntax highlighting, and code analysis, enhancing code productivity and readability. Spyder's integration with Jupyter notebooks allows users to combine the flexibility of notebooks with the power of a full-fledged IDE. Overall, Spyder provides a versatile and efficient environment for scientific computing and data analysis with Python.

2.3.4 Excel is a widely used spreadsheet software developed by Microsoft. It provides a comprehensive set of tools for data organization, calculation, analysis, and visualization. Excel offers a user-friendly interface with a grid-like layout where users can input data, perform calculations, and create formulas to automate processes. It supports a wide range of functions and formulas for statistical analysis, mathematical calculations, and data manipulation. Excel's powerful features include sorting, filtering, pivot tables, and charts that enable users to explore and visualize their data effectively. Additionally, Excel supports add-ins and macros, allowing users to extend its functionality and automate repetitive tasks. With Excel's extensive data analysis capabilities, users can perform tasks such as data cleaning, data modeling, trend analysis, and forecasting. Excel's popularity and widespread use make it a versatile tool for data analysis, financial modeling, and reporting across various industries and domain

3.4 DFDs/Flow Chart/ER diagram/Use case diagram/Activity Diagram/Sequence Diagram(Topic Wise)

FLOW CHART– Value Inc

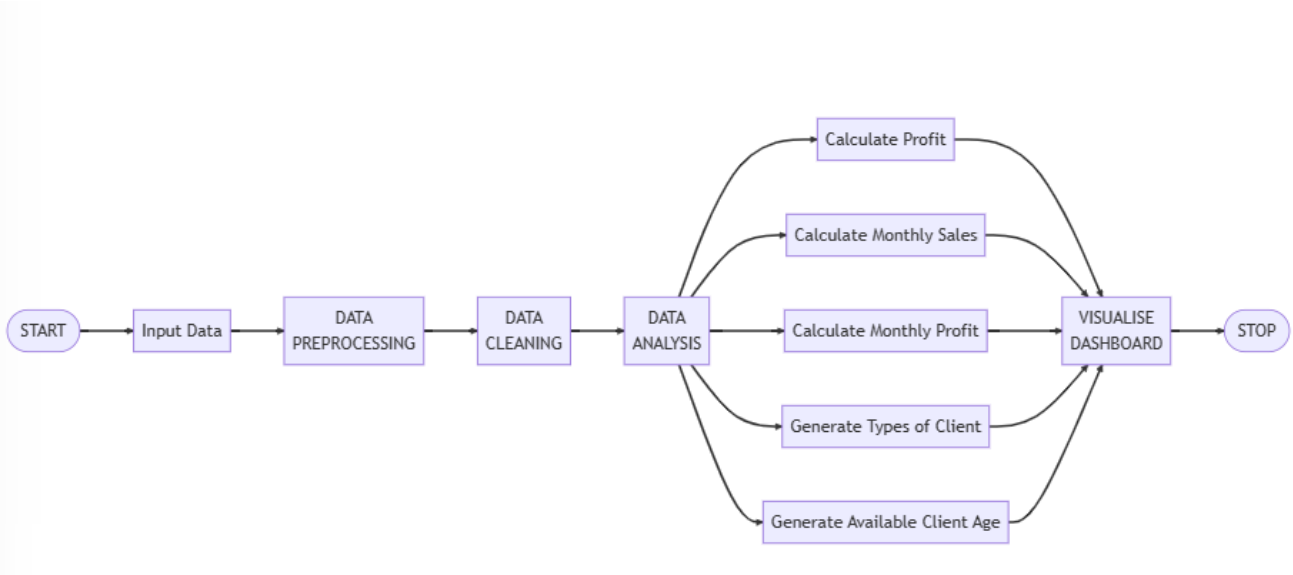

Figure 1. : Flow Chart (Value Inc)

DATA FLOW DIAGRAM – Value Inc

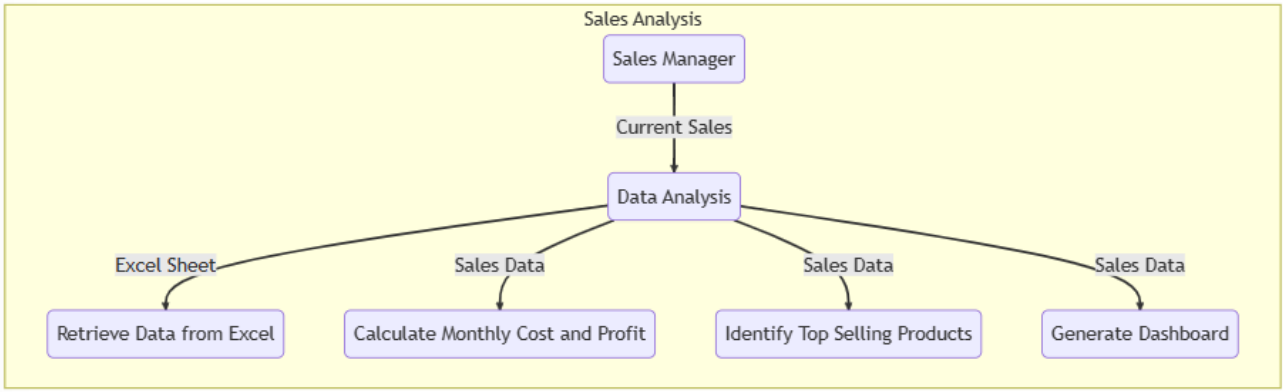

Figure 2 : DFD (Value Inc)

ER DIAGRAM– Value Inc

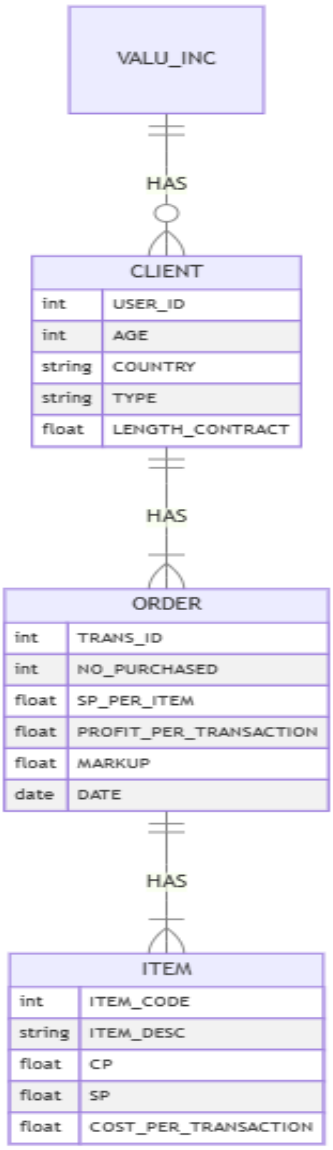

Figure 3 : ER Diagram (Value Inc)

USE CASE DIAGRAM – Value Inc

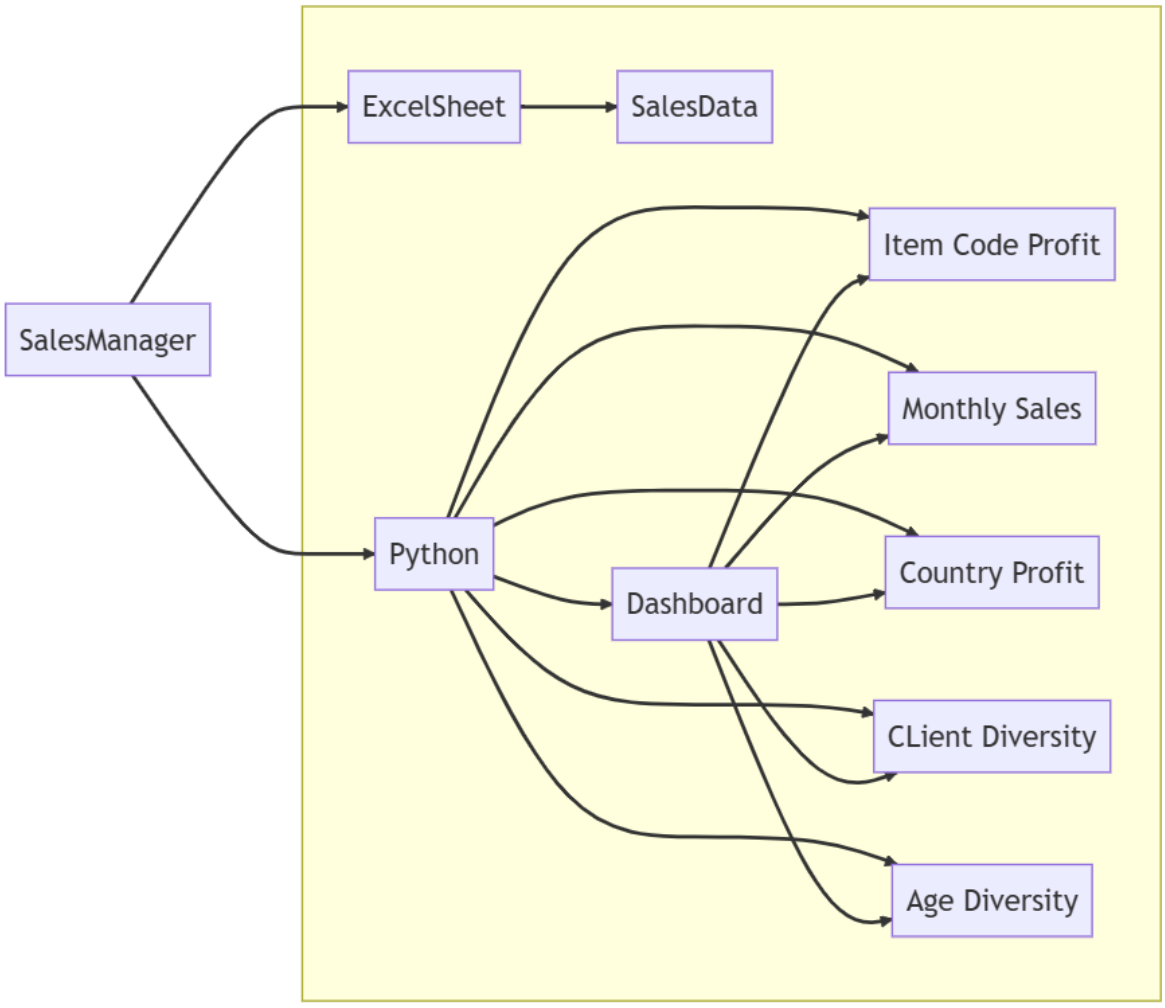

Figure 4: Use Case (Value Inc)

ACTIVITY DIAGRAM – Value Inc

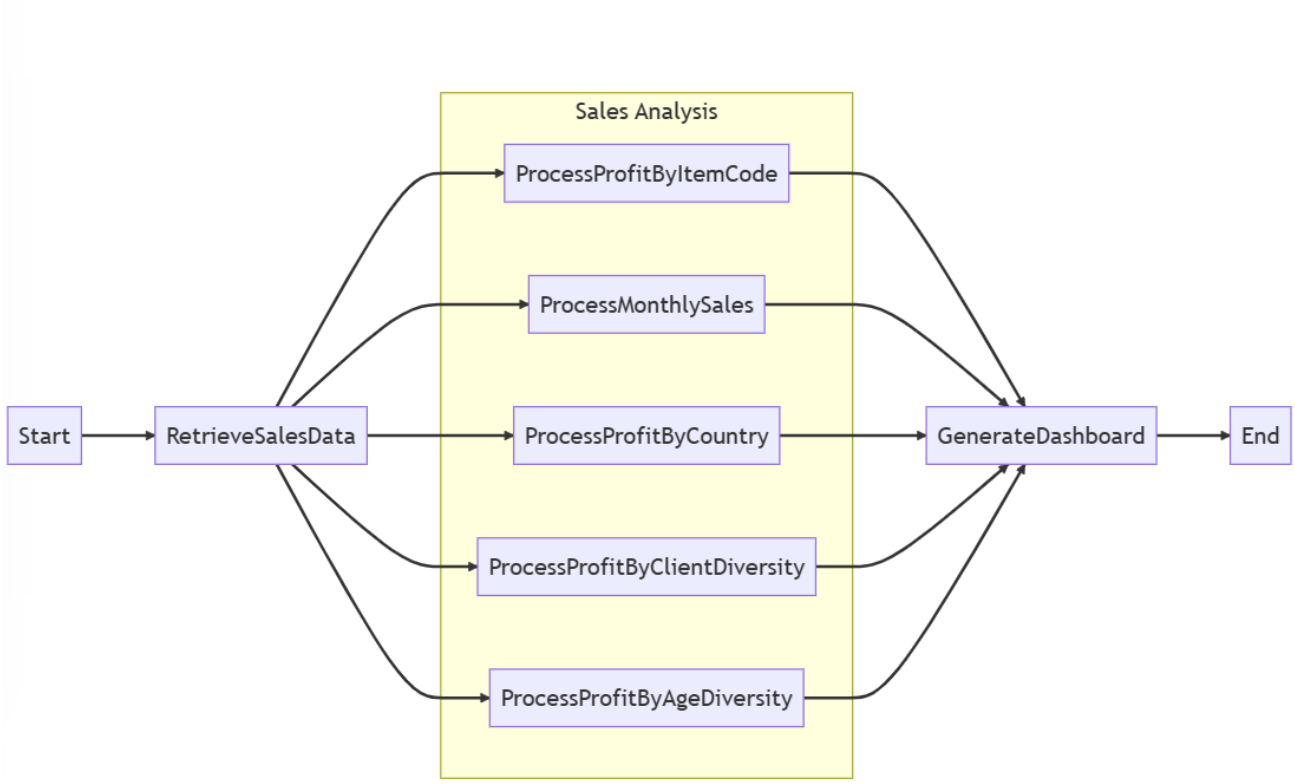

Figure 5 : Activity Diagram (Value Inc)

+

SEQUENCE DIAGRAM – Value Inc

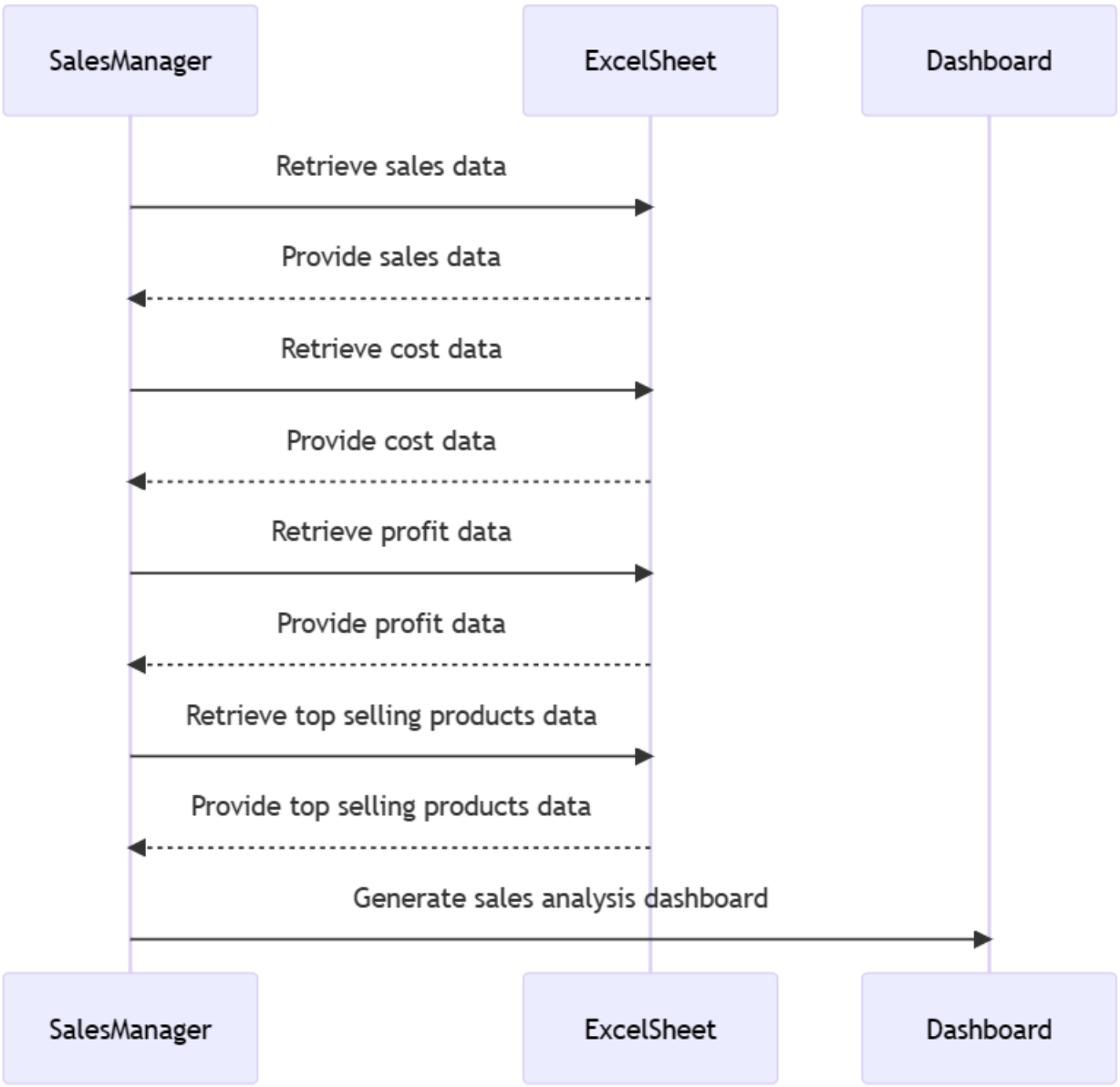

Figure 6 : Sequence Diagram (Value Inc)

STATE CHART DIAGRAM – Value Inc

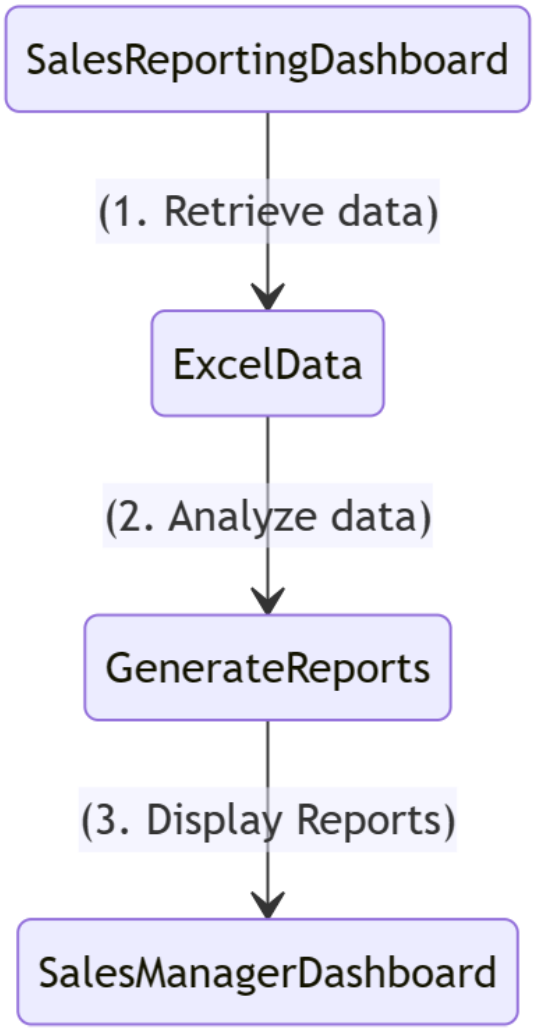

Figure 7 : State Chart (Value Inc)

FLOW CHART– Blue Bank

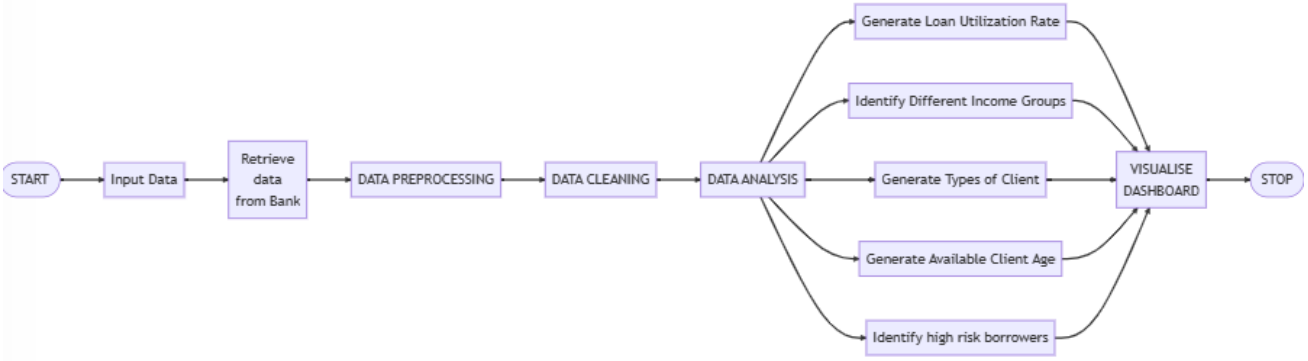

Figure 8 : Flow Chart (Blue Bank)

DATA FLOW DIAGRAM – Blue Bank

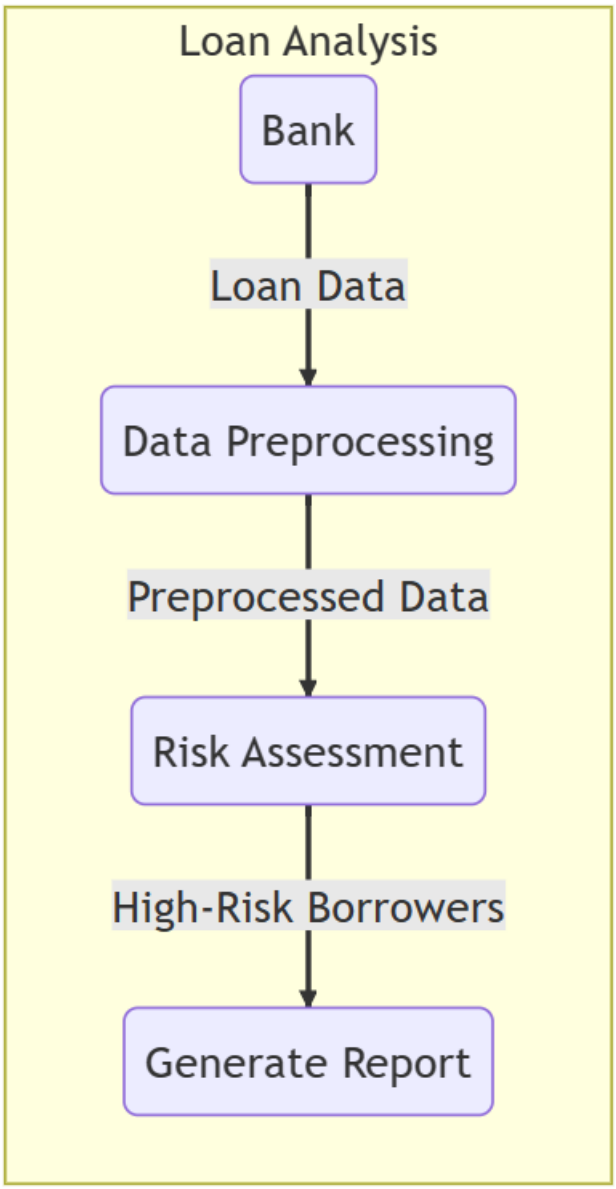

Figure 9 : DFD (Blue Bank)

ER DIAGRAM – Blue Bank

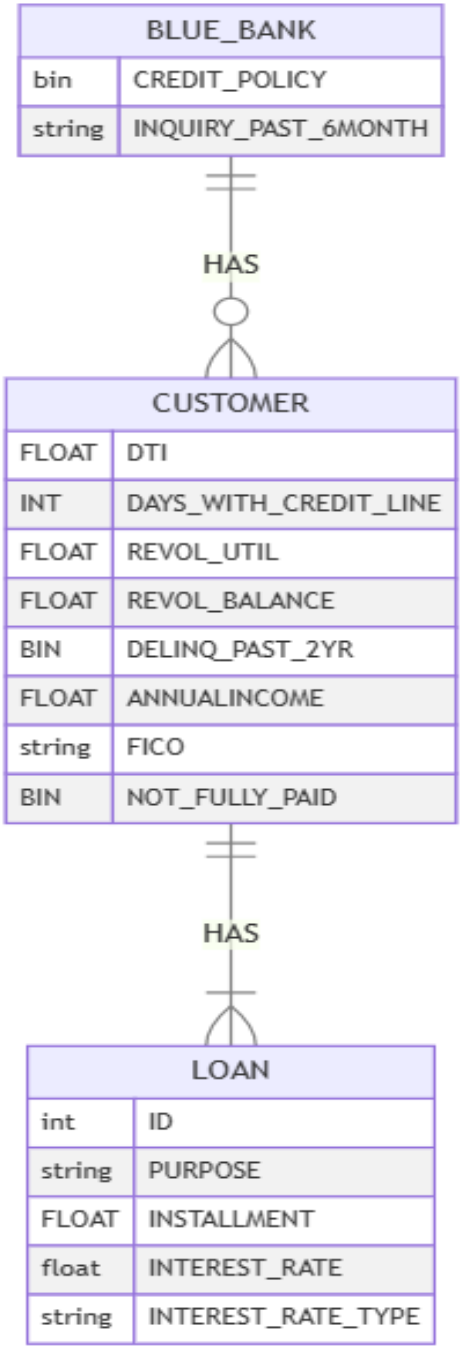

Figure 10 : ER Diagra (Blue Bank)

USE CASE DIAGRAM – Blue Bank

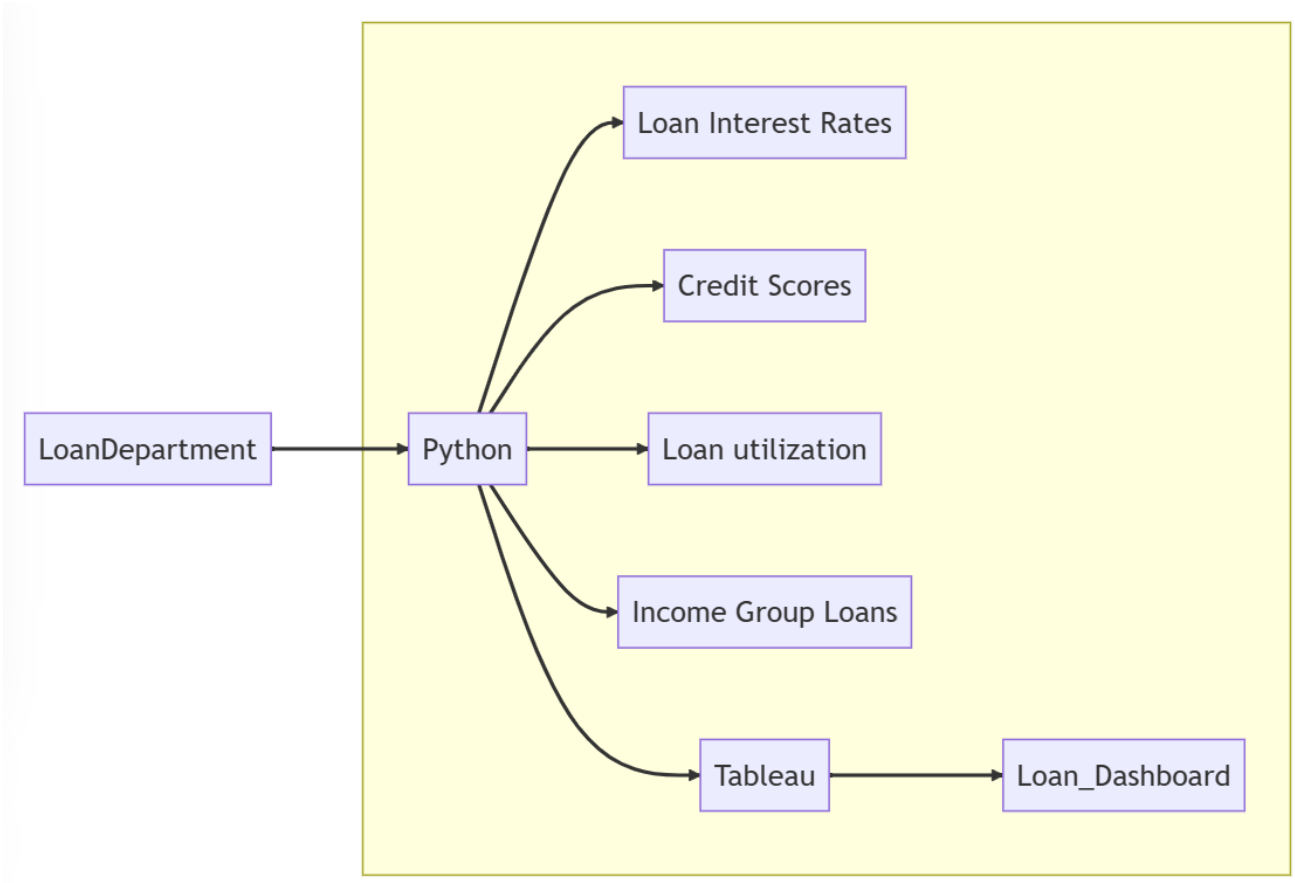

Figure 11: Use Case (Blue Bank)

ACTIVITY DIAGRAM – Blue Bank

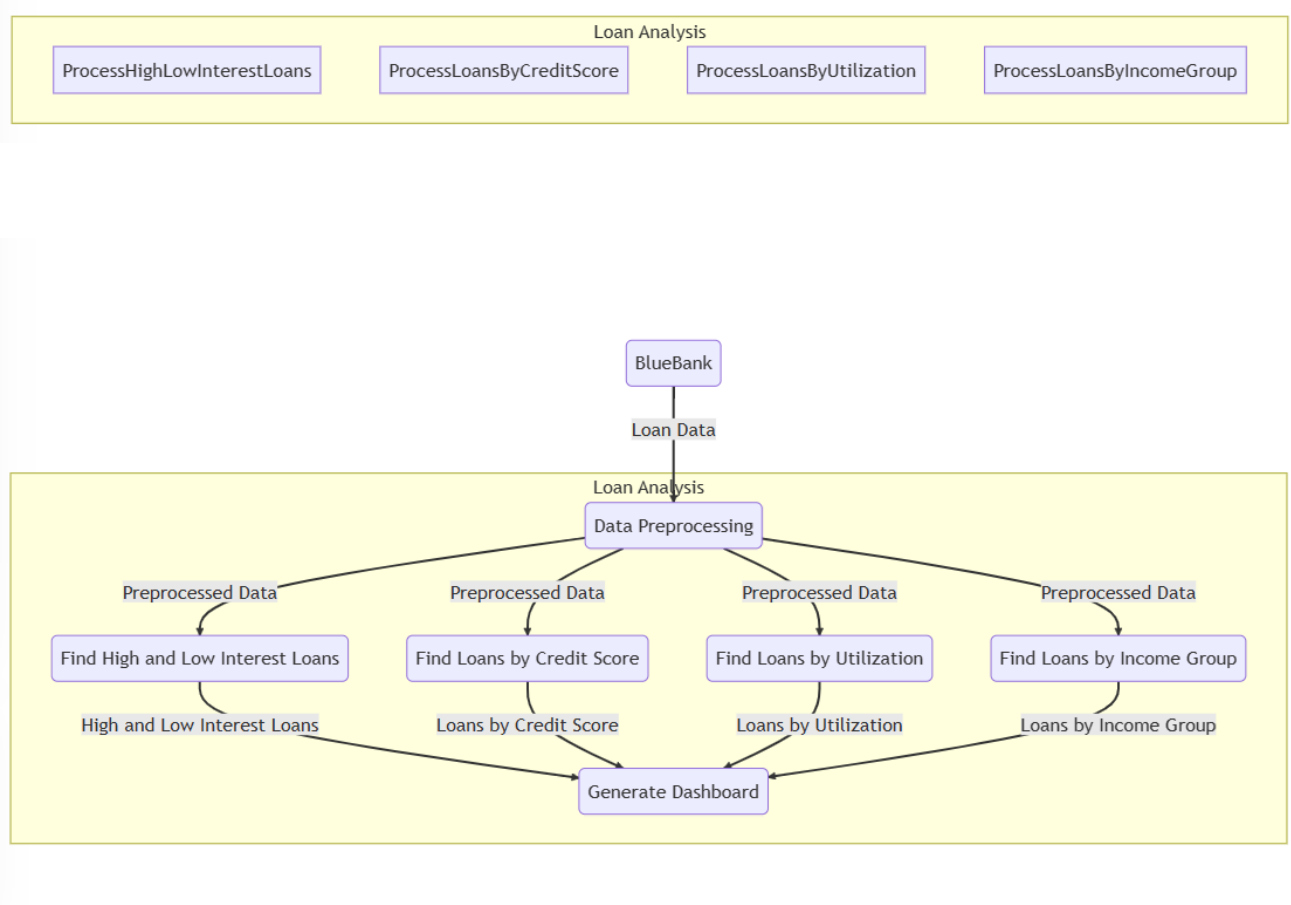

Figure 12 : Activity Diagram (Blue Bank)

SEQUENCE DIAGRAM – Blue Bank

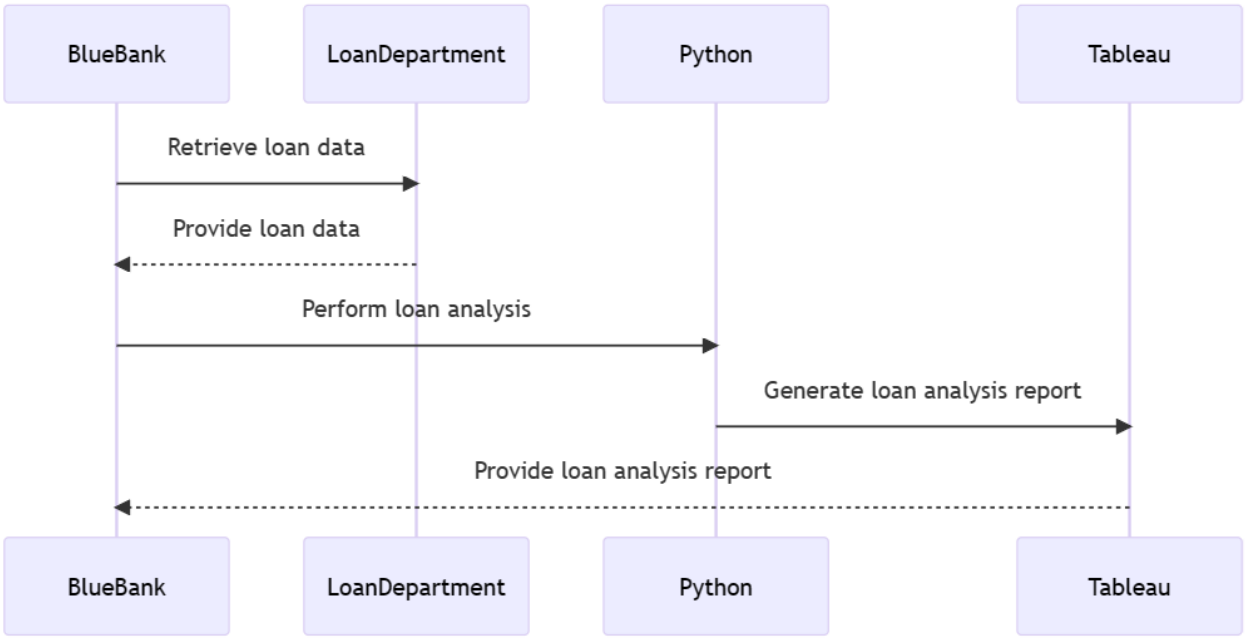

Figure 13 : Sequence Diagram (Blue Bank)

STATE CHART DIAGRAM – Blue Bank

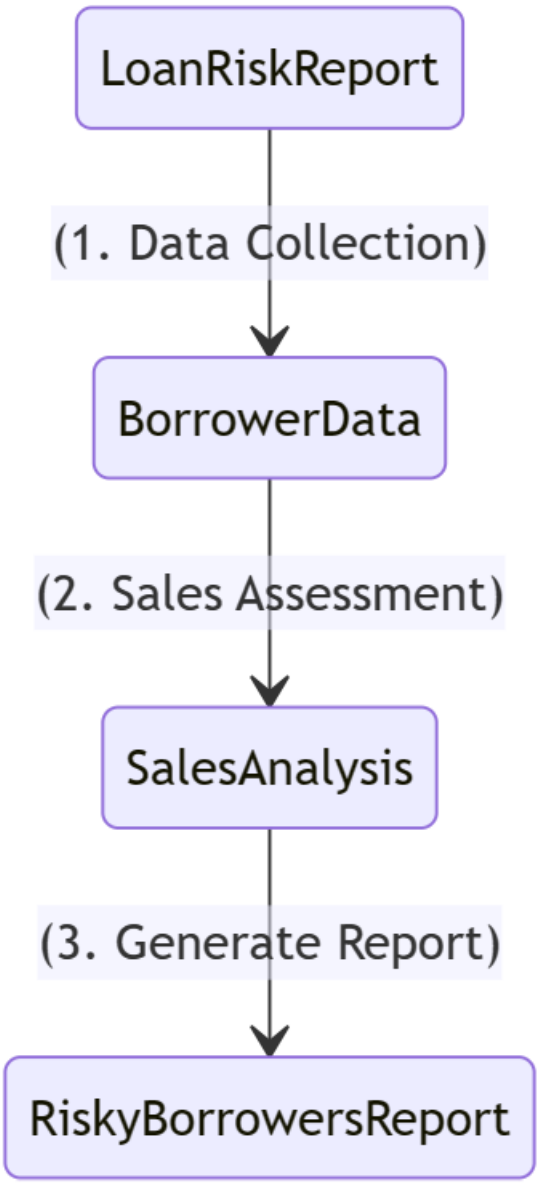

Figure 14 : State Chart (Blue Bank)

Flow chart – Blog Me

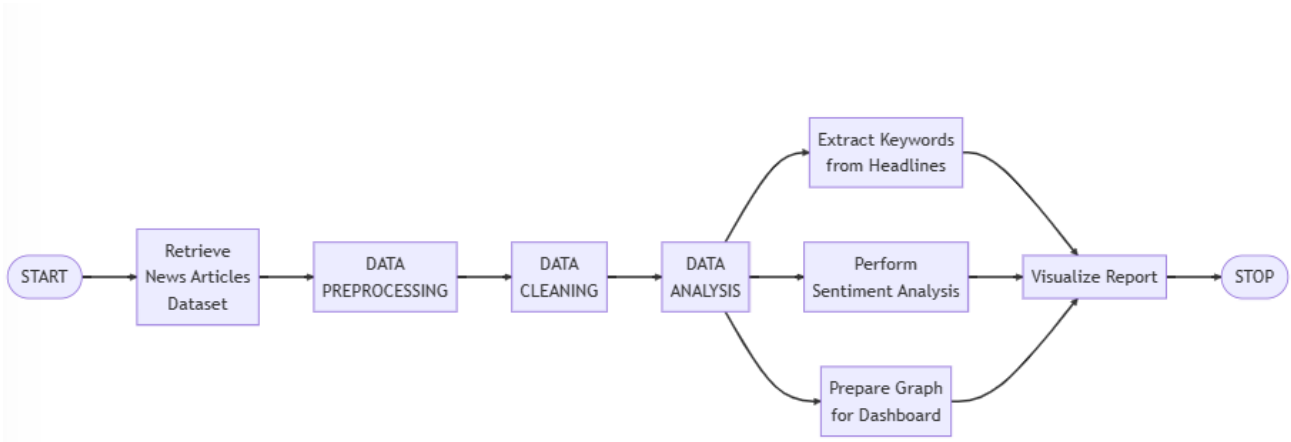

Figure 15 : Flow Chart (Blog Me)

DATA FLOW DIAGRAM – Blog Me

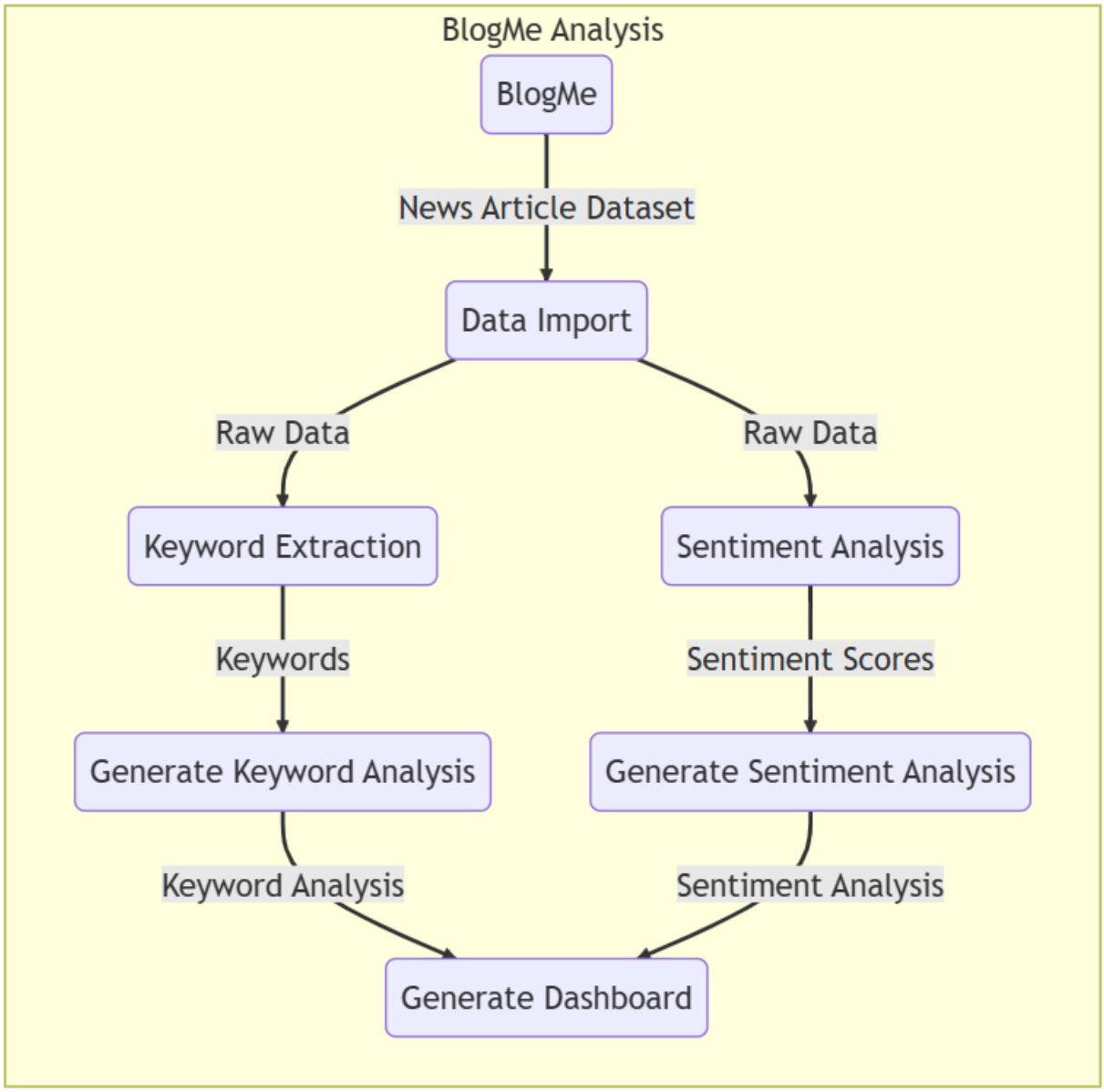

Figure 16 : DFD (Blue Bank)

ER DIAGRAM – Blog Me

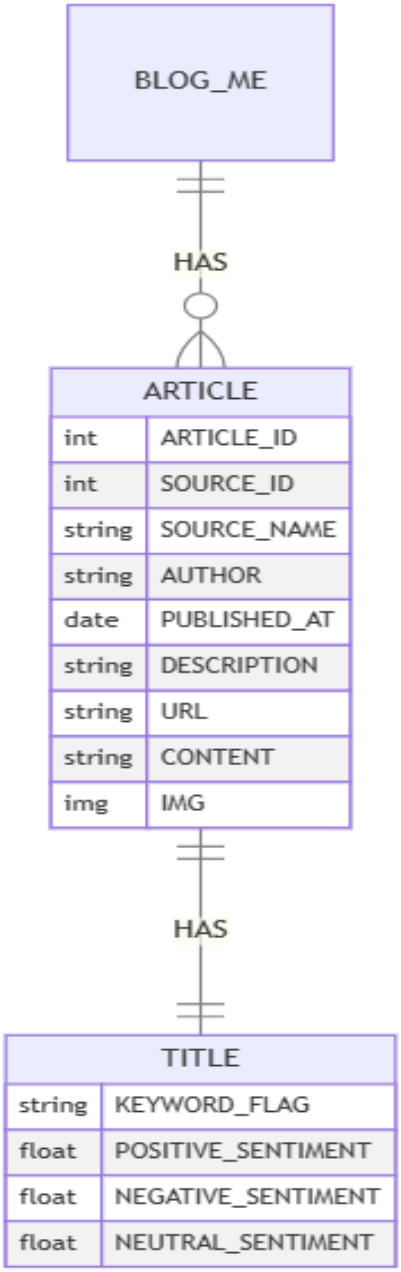

Figure 17 : ER Diagram (Blog Me)

USE CASE DIAGRAM – Blog me

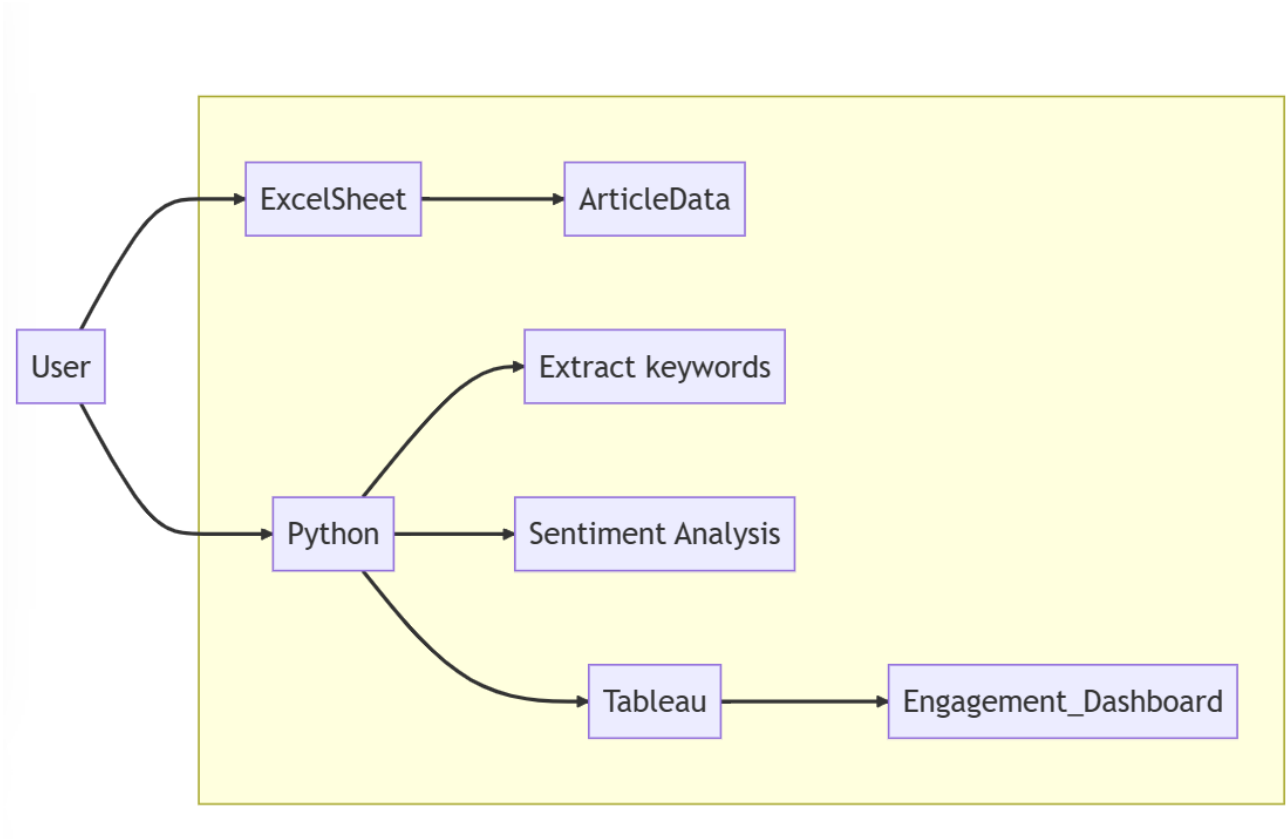

Figure 18: Use Case (Blog Me)

ACTIVITY DIAGRAM – Blog Me

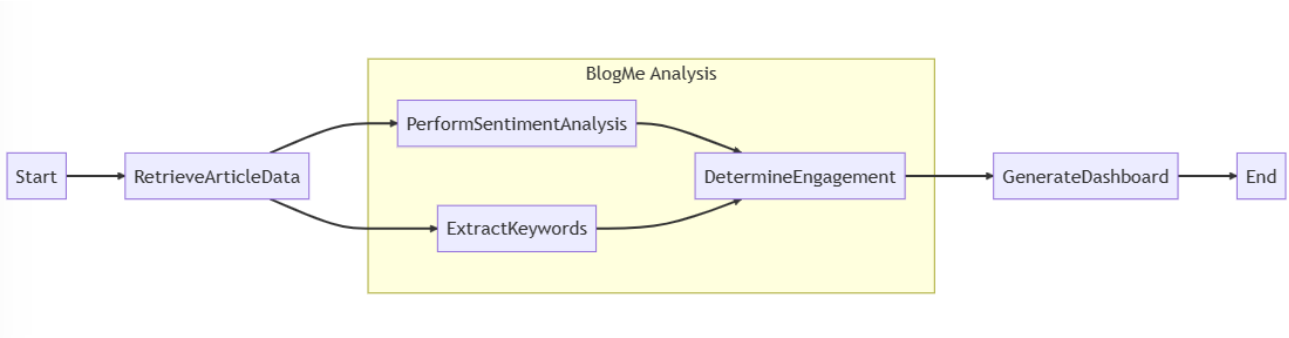

Figure 19 : Activity Diagram

SEQUENCE DIAGRAM – Blog Me

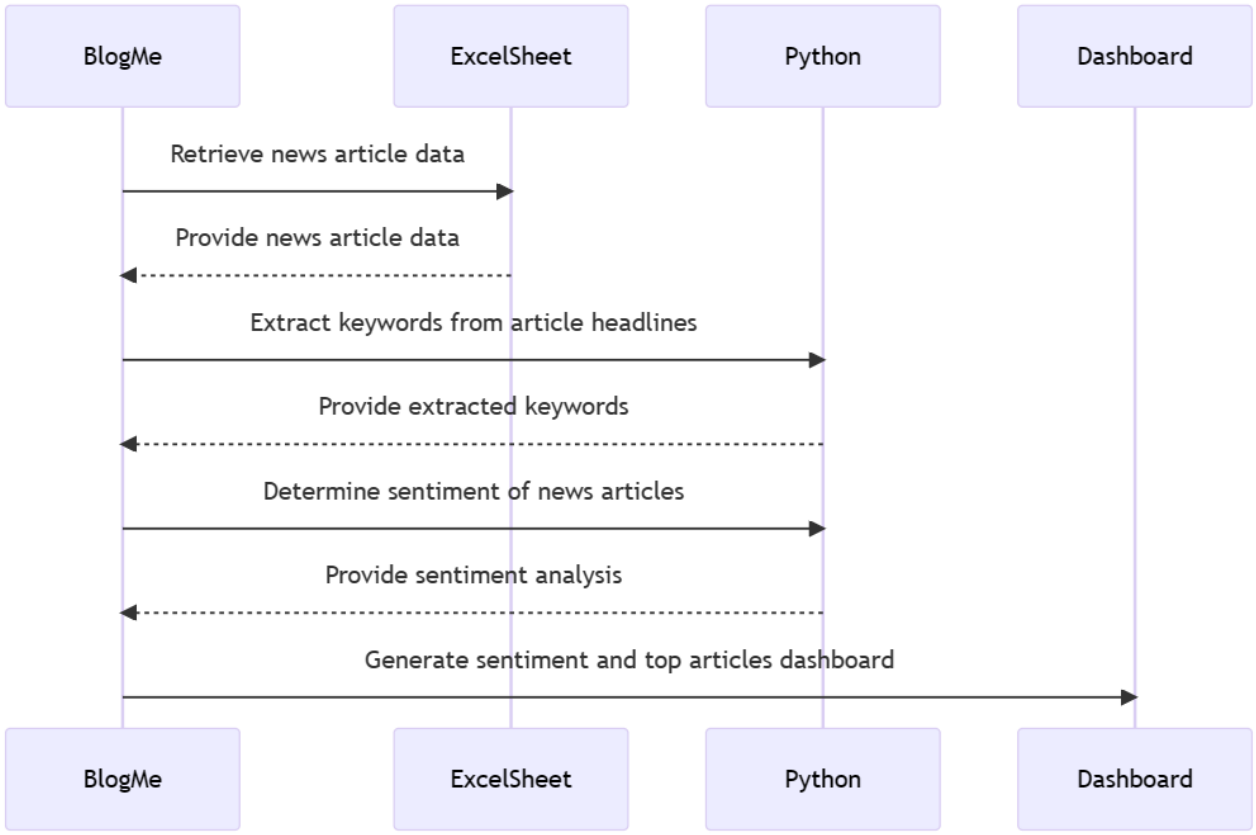

Figure 20 : Sequence Diagram (Blog Me)

STATE CHART DIAGRAM – Blog Me

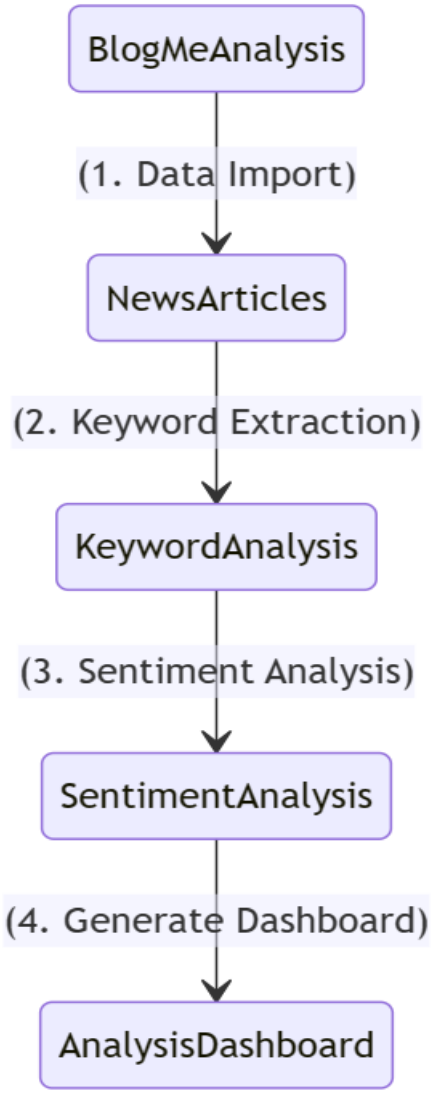

Figure 21 : State Chart (Blog Me)

3.5 BLUEPRINTS: -

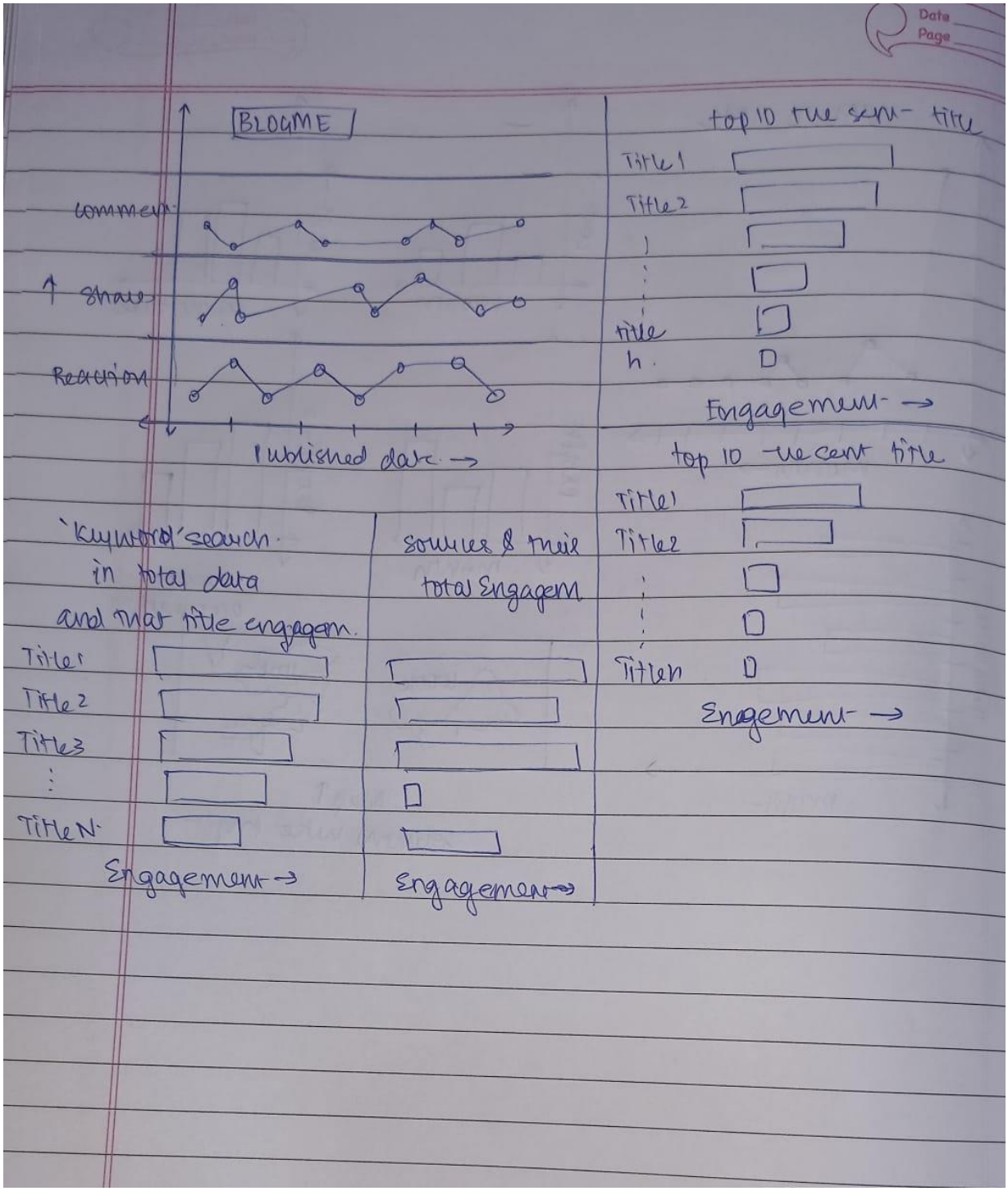

Figure 22 : Blog Me BluePrint

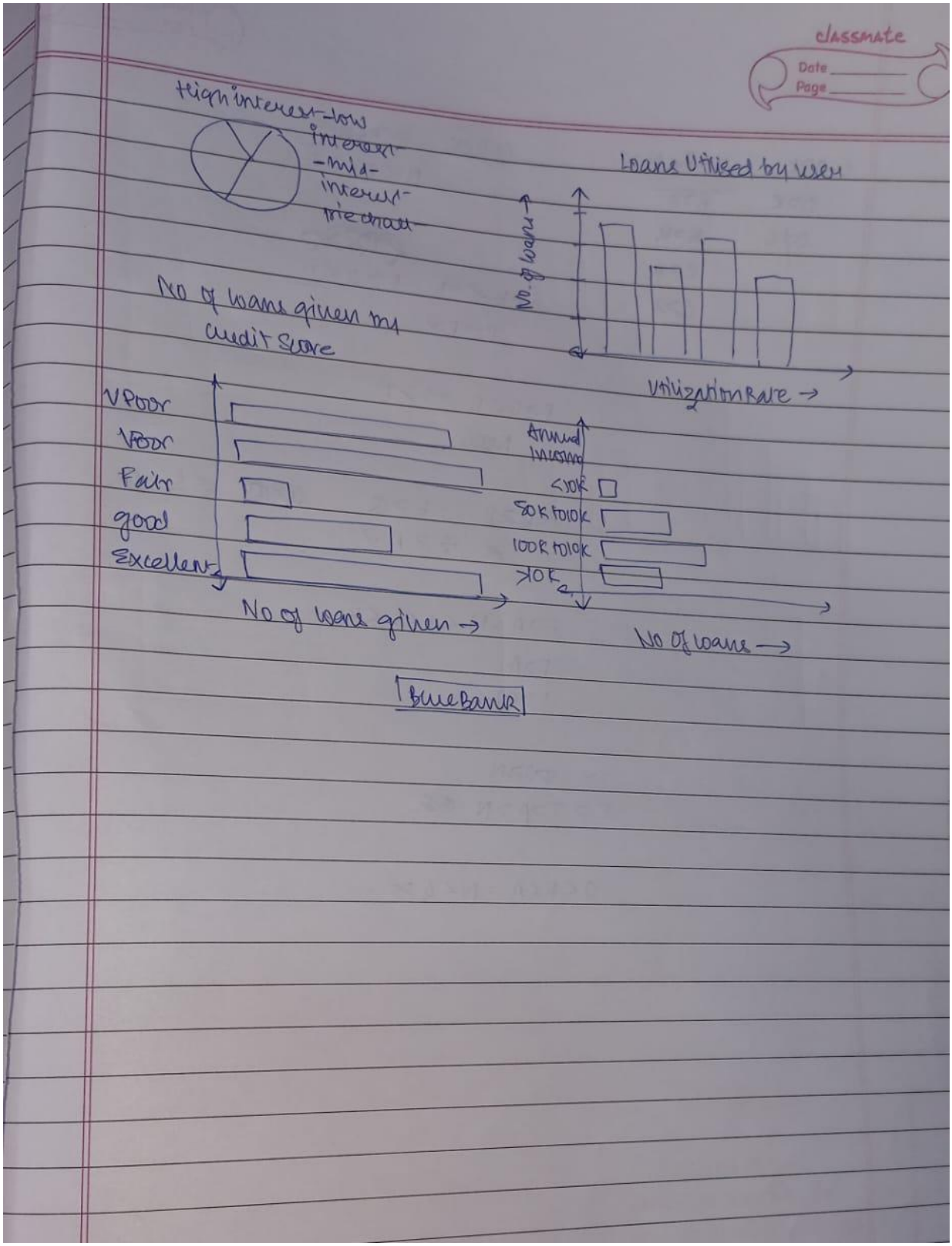

Figure 23 : Blue Bank BluePrint

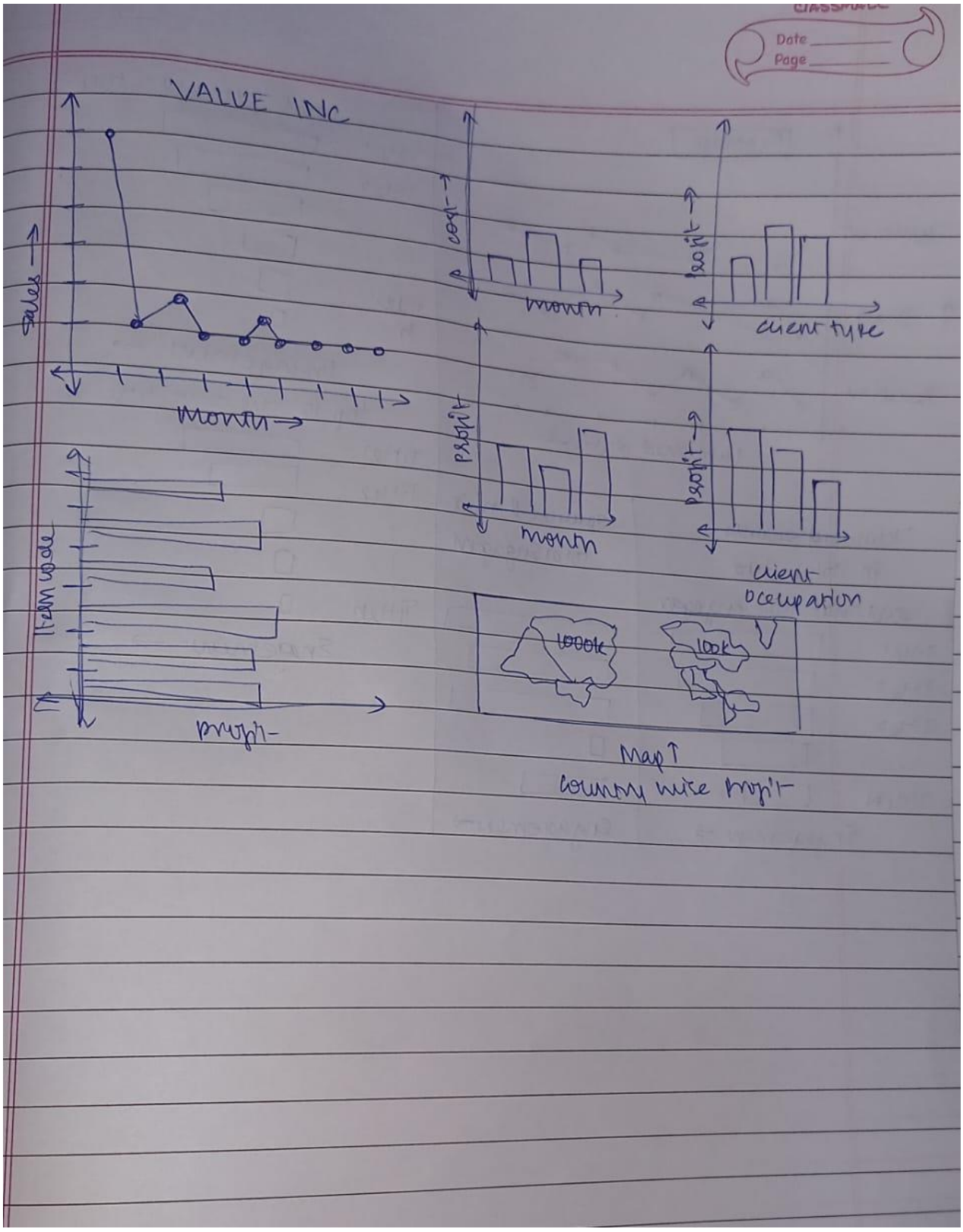

Figure 24 : Value Inc BluePrint

CHAPTER 4

IMPLEMENTATION & CODING

4.1 Operating System

4.2 Languages

4.3 S/W Tools

Chapter 4- IMPLEMENTATION & CODING

4.1 OPERATING SYSTEM

- Windows 8.1/10/11

4.2 LANGUAGE

4.2.1 PYTHON: Python is a high-level, versatile, and easy-to-learn programming language that has gained immense popularity in various fields, including data analysis, web development, artificial intelligence, scientific computing, and automation. Developed by Guido van Rossum in the late 1980s, Python emphasizes code readability and simplicity, making it an ideal choice for beginners and experienced developers alike.

Python boasts a vast ecosystem of libraries and frameworks that enable developers to accomplish a wide range of tasks efficiently. Its standard library provides a comprehensive set of modules for file I/O, string manipulation, networking, and more. Additionally, Python's third-party libraries like NumPy, Pandas, TensorFlow, and Django offer powerful tools for scientific computing, data analysis, machine learning, and web development.

With its clean and concise syntax, Python promotes code readability and reduces development time. Its object-oriented programming approach allows for modular and reusable code, while dynamic typing provides flexibility and faster prototyping. Python supports multiple programming paradigms, including procedural, functional, and object-oriented programming, allowing developers to choose the most suitable approach for their projects.

Furthermore, Python's cross-platform compatibility allows code to run seamlessly on different operating systems. Its extensive community support and active development ensure a wealth of resources, tutorials, and community-driven modules, making Python a flexible and robust language for a wide range of application

4.3 S/W TOOLS:

- Chrome
- Spyder
- Tableau Public
- Excel
- Anaconda

CHAPTER 5

TEST CASES AND THEIR RESULTS

5.1 TEST CASE FOR VALUE INC. UNCLEANED DATA

5.2 TEST CASE FOR BLUE BANK UNCLEANED DATA

5.3 TEST CASE FOR BLOG ME UNCLEANED DATA

TABLE 1 : TEST CASE FOR VALUE INC. UNCLEANED DATA

TEST CASE	TEST CASE DESCRIPTION	RESULT
TEST CASE 1	RETRIEVING SALES PER TRANSACTION	NEGATIVE
TEST CASE 2	RETRIEVING DATE AS STRING	POSITIVE
TEST CASE 3	RETRIEVING CLIENT AGE	NEGATIVE
TEST CASE 4	RETRIEVING CLIENT TYPE	NEGATIVE
TEST CASE 5	RETRIEVING CLIENT KEYWORDS	POSITIVE
TEST CASE 6	RETRIEVING ITEM DESCRIPTION	POSITIVE

TABLE 2 : TEST CASE FOR BLUE BANK UNCLEANED DATA

TEST CASE	TEST CASE DESCRIPTION	RESULT
TEST CASE 1	DESCRIBING INTEREST RATE COLUMN	POSITIVE
TEST CASE 2	DESCRIBING FICO CATEGORY COLUMN	POSITIVE
TEST CASE 3	RETRIEVING DEBT TO INCOME RATIO	POSITIVE
TEST CASE 4	RETRIEVING CATEGORY WISE LOAN DATA	NEGATIVE
TEST CASE 5	RETRIEVING DATA FRAME	POSITIVE
TEST CASE 6	RETRIEVING INTEREST RATE TYPE	NEGATIVE

TABLE 3 :TEST CASE FOR BLOG ME UNCLEANED DATA

TEST CASE	TEST CASE DESCRIPTION	RESULT
TEST CASE 1	RETRIEVING COUNT OF ARTICLES PER SOURCES	NEGATIVE
TEST CASE 2	RETRIEVING SUM OF ENGAGEMENT COUNT	POSITIVE
TEST CASE 3	FINDING SEGREGATED DATA BASED ON KEYWORD	NEGATIVE
TEST CASE 4	RETRIEVING TITLE	POSITIVE
TEST CASE 5	RETRIEVING DATA FRAME	POSITIVE
TEST CASE 6	RETRIEVING SENTIMENT	NEGATIVE

CHAPTER 6

MODULES AND SUB MODULES WITH DESCRIPTION

6.1 MODULES WITH DESCRIPTION

6.2 TABLEAU AND PROBLEMS FACED

6.3 GITHUB LINKS FOR PYTHON SCRIPTS

6.1 Modules & sub-modules

Blue Bank

1.4.7.1.1 JASON FILE MODULE

Brief: Initially the data is in json file format which need to be embedded inside the code in order to extract the main data frame. So we have to import json python library in order to do so

Sub Module:

- Pd.dataframe(jsonfile)
- The above function converts json file to a data frame using a , where dataframe function is a predefined function present inside pandas python library, which is imported under the alias pd.

II. DESCRIBE MODULE

Brief: Describe module is used to generate a statistical overview of a particular column. Describe module consists of a function describe() which yields count , mean , std , min 25% , 75%, and max including the inter quartile ranges of a particular column.

The syntax for the above mentioned function is dataframe[‘columnName’].describe

III. EXPONENT MODULE

Brief: Exponent module is used to convert exponented value of annual income into a readable floating point value. Exponent module is a part of numpy python library which is imported under the alias np. The syntax for the above mentioned function is as follows:

Newcol=np.exp(dataframe[‘columnName’])

IV. FICO MODULE

Brief: FICO module describes the credit score of the person who has been given loan, and using the score we segregate the credit scores according to the desired categories using the following syntax :

```
for x in range(0,len(loandata)):
    category = loandata['fico'][x]
    try:
        if category>=300 and category<400:
            cat='Very Poor'
        elif category>=400 and category<600:
            cat='Poor'
        elif category>=601 and category<660:
            cat='Fair'
        elif category>=660 and category<700:
            cat='Good'
        elif category>=700:
            cat='Excellent'
        else:
            cat='Unknown'
    except:
```

```
        cat='errror'


    ficocat.append(cat)
ficocat = pd.Series(ficocat)


loandata['fico.category']=ficocat
```

V. INTEREST MODULE

Brief: This module derives the interest rates from the dataframe and using that data we segregate the interest rates according to the high interest rate and low interest rates that each user has. And to do the followingwe use loc() python predefined function.

VI. FICO MODULE LOANS NUMBER

Brief: We segregate the number of loans according to the fico category that was earlier provided by us , the categories included very poor , poor , fair , good and excellent.

VII. PLOTTING MODULE (OPTIONAL)

Brief: The plotting module is a module that enables us to plt different type of graphs and plots using the python library named as matplotlib which is imported under the alias plt.

The syntax for plotting bar graph & scatterplot is as follows:

```
ypoint = loandata['annualincome']
```

```
xpoint = loandata['dti']

plt.scatter(xpoint,ypoint,color='red')

purposecount=loandata.groupby(['purpose']).size()

purposecount.plot.bar(color='blue',width=0.9)
```

VIII. EXPORTING MODULE

Brief : We export the cleaned , transformed and integrated data to a new file using the syntax : data.to_csv('BlueBank_Cleaned.csv',index=False)

6.1 Modules & sub-modules

Value Inc

I. CSV FILE MODULE

i. Brief: Initially the data is in CSV file format which need to be embedded inside the code in order to extract the main data frame. So we have to import read_csv pandas library's function in order to do so

ii. Sub Module:

b. data = pd.read_csv('transaction.csv' , sep=';')

i. the above function reads the csv file and saves it as a dataframe inside our code , here the first parameter represents filename and second parameter represents the separator

II. CHANGING VALUES

i. Brief: This module is used to change the values of certain fields according to the desired values by applying computational formulas , for example sales per transaction is the product of selling price of each item and number of the items purchased , similary profit per transaction is the submission of sales per transaction and cost per transaction

ii. The syntax for the above is :

b. #sales per transaction

c. data['SalesPerTransaction'] = data['SellingPricePerItem'] * data['NumberOfItemsPurchased']

d. #ProfitPerTransaction = sales - cost

e. data['ProfitPertransaction'] = data['SalesPerTransaction'] - data['CostPerTransaction']

III. MARKUP MODULE

i. Brief: Markup is the the amount added to the cost price of goods to cover overheads and profit. In which we use the basic formula

ii. markup = (sales-cost)/cost

iii. The syntax for the above is :

iv. data['Markup'] = (data['SalesPerTransaction'] - data['CostPerTransaction'])/data['CostPerTransaction']

v. data['Markup'] = (data['ProfitPertransaction'])/data['CostPerTransaction']

vi. #rounding markup

vii. roundmarkup = round(data['Markup'],2)

viii. data['Markup'] = round(data['Markup'],2)

IV. DATE MODULE

i. Brief: day, date and month are three independent columns present inside uncleaned data which needs to be segregated under one column so we perform simple concatenation of the columns and change the final result's data type to string.

V. SPLIT MODULE

i. Brief: This module derives the individual values from a concatenated column named client keywords which describes the information about client type , client occupation and time for which client has been associated with the company. To perform the split function we use the following syntax:

ii. #using split to split client keyword field

b. #new_var = column.str.split('sep' , expand = True)

c. split_col = data['ClientKeywords'].str.split(',',expand = True)

d. data['ClientAge'] = split_col[0]

e. data['ClientType'] = split_col[1]

f. data['LengthOfContract'] = split_col[2]

g. where “,”represents the separator.

VI. MERGING MODULE

i. Brief: We merge two csv files , in order to do that we first read the file to be merged using pd.read_csv() function. We use the following syntax :

ii. #merging files: merge_df = pd.merge(df_old, df_new , on = 'key')

b. data = pd.merge(data, seasons, on='Month')

c. where month is the common key.

VII. EXPORTING MODULE

Brief : We export the cleaned , transformed and integrated data to a new file using the syntax : data.to_csv('BlueBank_Cleaned.csv',index=False)

6.1 Modules & sub-modules

Blog Me

VIII. EXCEL FILE MODULE

i. Brief: Initially the data is in EXCEL file format which need to be embedded inside the code in order to extract the main data frame. So we have to import read_excel pandas library’s function in order to do so

ii. Syntax for the above is as follows:

iii. data = pd.read_excel("articles.xlsx")

iv. the above function reads the excel file and saves it as a dataframe inside our code , here the first parameter represents filename.

IX. DESCRIBE MODULE

Brief: Describe module is used to generate a statistical overview of a particular column. Describe module consists of a function describe() which yields count , mean , std , min 25% , 75%, and max including the inter quartile ranges of a particular column.

The syntax for the above mentioned function is dataframe.describe()

IX. GROUPBY MODULE

i. Brief: This module is used to group certain column according to an another column. For example there are two independent fields named source name and article id ,and we need count of articles that are published by each source , so in order to do that we use groupby function.

ii. the syntax to perform the above is as follows:

```
#counting number of articles per source

#format : dataframe.groupby(['column_to_group'])['column_to_count'].count()

data.groupby(['source_id'])['article_id'].count()


#total number of reactions by publishers

data.groupby(['source_id'])['engagement_reaction_count'].sum()
```

X. KEYWORD MODULE

i. Brief: I will be using a function named keyword flag which will accept a keyword as a parameter and if that keyword is present in either of all the rows that are present inside the heading column then it will append flag value 1 inside a series which will generate the number of times that keyword occurs inside the whole data frame.

ii. The syntax for the above is :

```
def keywordflag(keyword):
    length = len(data)
    keyword_flag = []
    for x in range(0,length):
        heading = data['title'][x]
        try:
            if keyword in heading:
                flag=1


            else:
                flag=0
        except:
            flag=0
        keyword_flag.append(flag)
    return keyword_flag
```

keywordflag = keywordflag('murder')

XI. SENTIMENT MODULE

i. Brief: sentiment module consists of a loop that traverses through each value present inside title field and calculate the sentiment's intensity and generates a polarity score for the same. The intensity is calculated as a floating point value representing positive , neutral and negative sentiment and it's intensity. Here we create a new column for each title that describes the intensity of the corresponding title. We take up each title and pass it as a parameter to a function of a function that is SentimentIntensityAnalyser.polarityscores(), which returns a floating point value. An append that value according to the intensity inside titles with negative sentiment , inside titles with positive sentiment and title with neutral sentiment

XII. EXPORTING MODULE

i. **Brief :** We export the cleaned , transformed and integrated data to a new excel file using the syntax : data.to_excel('blogme_clean.xlsx',sheet_name='blogmedata',index=False)

6.2 Tableau and problems faced

Tableau is a powerful and widely used data visualization tool that allows users to create interactive and visually appealing dashboards, reports, and charts. It provides a user-friendly interface and intuitive drag-and-drop functionality, making it accessible to users with varying levels of technical expertise. Tableau is designed to help individuals and organizations make sense of complex data by transforming raw data into meaningful visual representations.

Tableau's applications span across various industries and domains. In business settings, Tableau enables data analysts, business intelligence professionals, and decision-makers to explore, analyze, and present data in a visually compelling manner. It allows users to connect to a wide range of data sources, including databases, spreadsheets, cloud services, and big data platforms, and quickly generate interactive visualizations without the need for complex coding or programming.

With Tableau, organizations can gain valuable insights and derive actionable intelligence from their data. It facilitates data discovery, trend analysis, and pattern identification, helping users identify correlations and outliers that may not be apparent in traditional data formats. Tableau's interactive features enable users to drill down into specific data points, apply filters, and manipulate visualizations in real-time, empowering them to explore data from multiple perspectives and uncover meaningful insights.

TABLEAU - VALUE INC

1. We had the values of current markup and goal markup , goal markup is a field that can be mutilated by user or can be set to specific value, so we needed a definite value for how much markup the company needs to make in order to achieve the target markup. So to solve the above problem we can either create a calculated field , but that won't be necessary since it contains a singular value not a column or rows of values so instead I made a parameter in which I set current value as 40 and it's name as Markup%Target

2. Markup%Target

target markup – current markup = goal markup

3. And to formulate the above in the form of percentage I created a calculated field in which following code was put :

4. AmountToTarget[Markup%Target/100 – AVG(Markup)]

5. We created a month wise filter for transaction dates , and applied it to all the sheets.

TABLEAU - BLUE BANK

1. We wanted a graph for number of loans according to the income groups , but income groups were not classified so to do so , I created a calculated field named as annual income group and added the following code :

2. IF [Annual Income]<10000 then "<10k"

   ELSEIF [Annual Income]>=10000 and [Annual Income]<50000 then "10k to 50k"

   ELSEIF [Annual Income]>=50000 and [Annual Income]<100000 then "50k to 100k"
   ELSEIF [Annual Income]>=100000 and [Annual Income]<250000 then "100k to 250k"
   ELSEIF [Annual Income]>=250000 and [Annual Income]<500000 then "250k to 500k"
   ELSEIF [Annual Income]>=500000 then ">500k"
   ELSE "unknown"
   END

3. Created filters were applied to all the sheets

TABLEAU - BLOG ME

1. In blogme initially we had 2 tables , one consisting of blog me sources and one consisting of all the data , so in order to make a dashboard we needed to perform a join on these two tables. So I went with left join since my major table that was blogmedata table, had all the source names and I needed to add more source information to this table so the following join was performed

2. Blogmedata leftjoin blogmesources

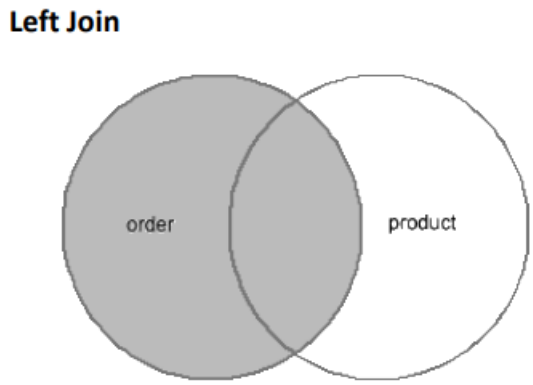

Figure 25 : Left Join Venn Diagram

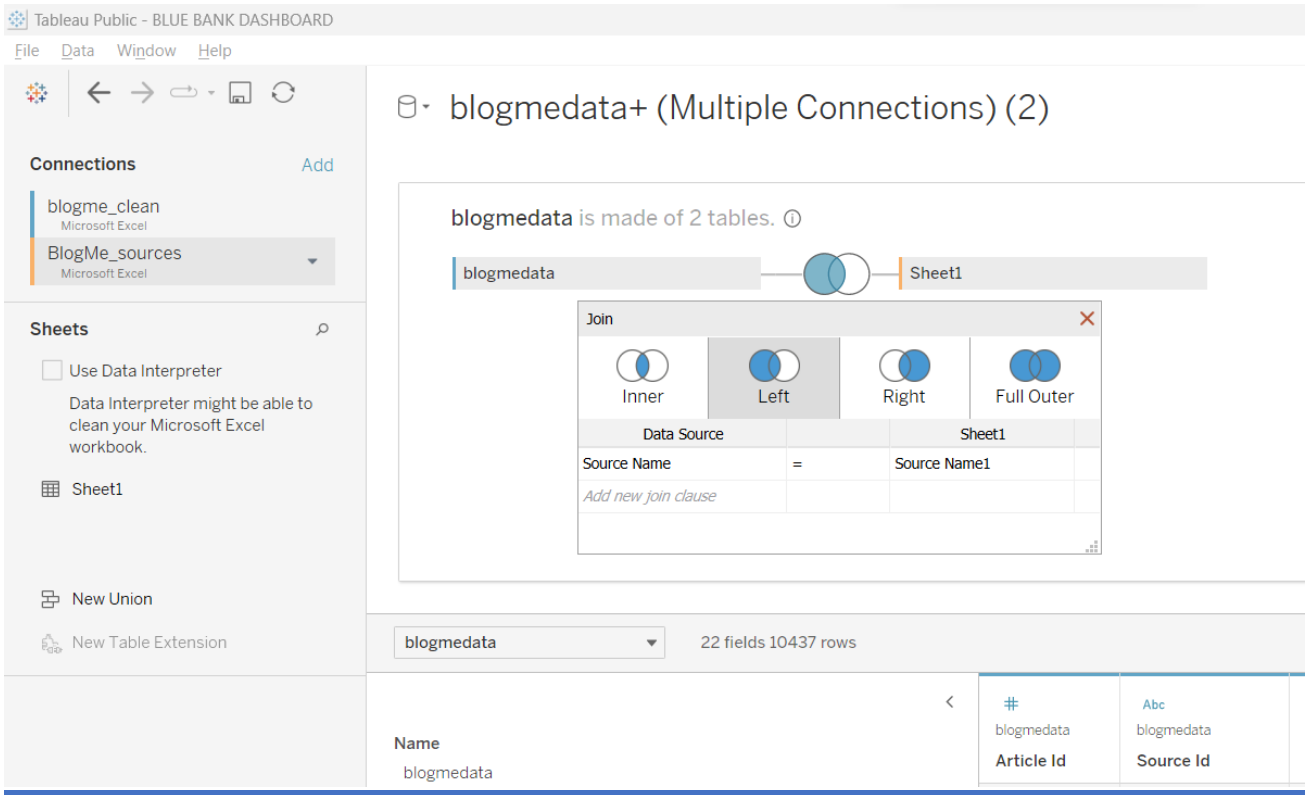

Figure 26 : Left Join Venn Diagram

3. Date consisted of date and random character so I used a string function to extract 10 letters from the left which gave me just the date. The syntax for the following was: LEFT(publishedAt,10). Where publishedAt is a filed name.

4. Sentiment analyzer generated a polarity score for the title , but did not classify it as

positive negative or neutral title, so in order to do so I created a calculated field with the name sentiment and added the following code to it :

IF [Title Neg Sentiment]>=0.5 then "Negative"

ELSEIF [Title Pos Sentiment]>=0.5 then "Positive"

ELSE "Unknown"

6.3 GITHUB LINKS FOR PYTHON SCRIPTS

1. **VALUE INC**
   pythonTableau/valueinc_sales.py at main · Parul0103/pythonTableau (github.com)

2. **BLUE BANK**
   pythonTableau/bluebank.py at main · Parul0103/pythonTableau (github.com)

3. **BLOGME**
   pythonTableau/blogme.py at main · Parul0103/pythonTableau (github.com)

CHAPTER 7

RESULTS & CONCLUSION

7.1 FINAL DASHBOARD RESULTS

7.2 PERSONAL REFLECTION

7.3 FUTURE SCOPE

7.4 CONCLUSION

Chapter 6- RESULTS & CONCLUSION

7.1 FINAL DASHBOARD RESULTS

1. TABLEAU DASHBOARD – BLOG ME

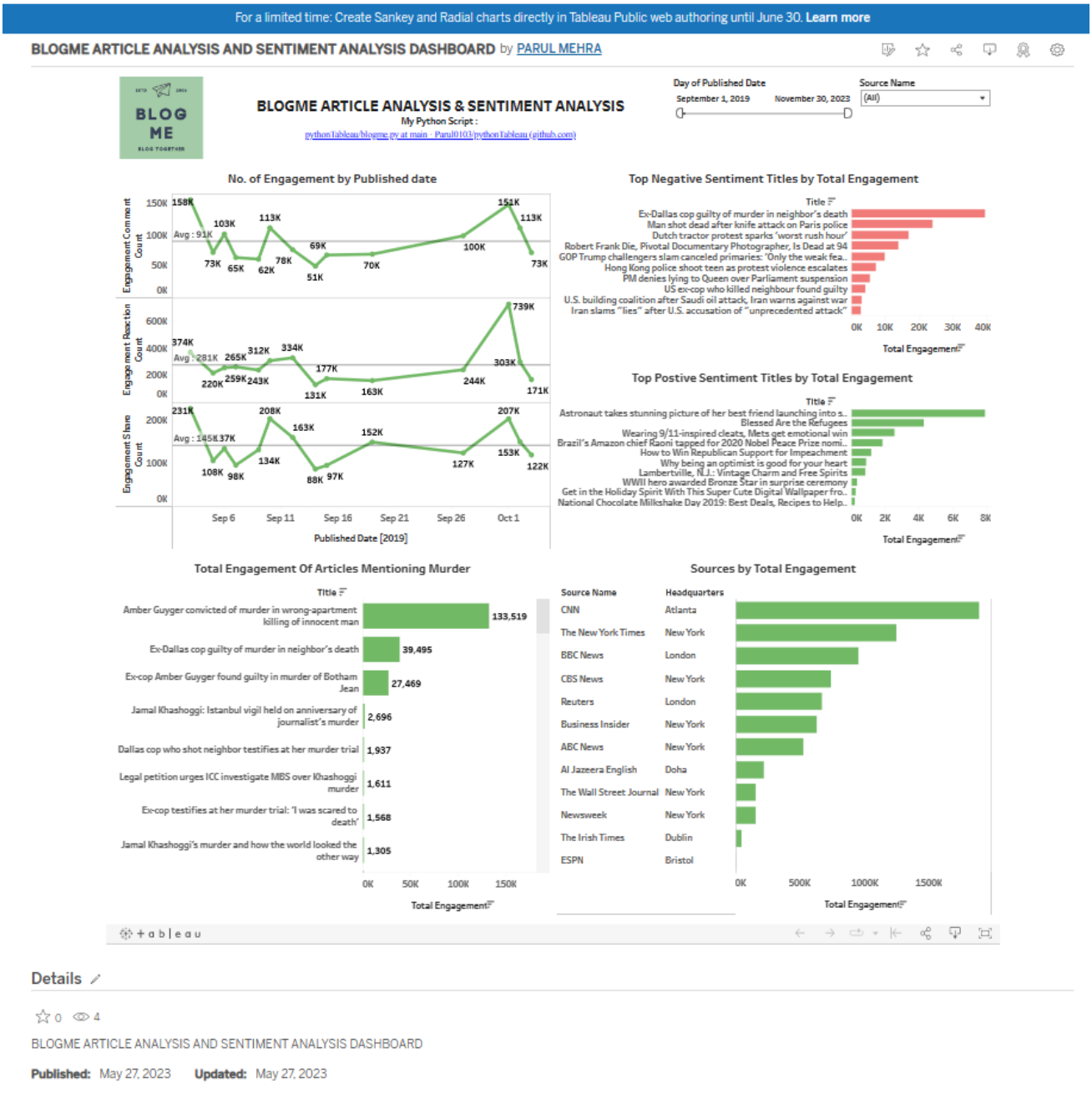

Figure 27 : Blog Me Dashboard

2. TABLEAU DASHBOARD – BLUE BANK

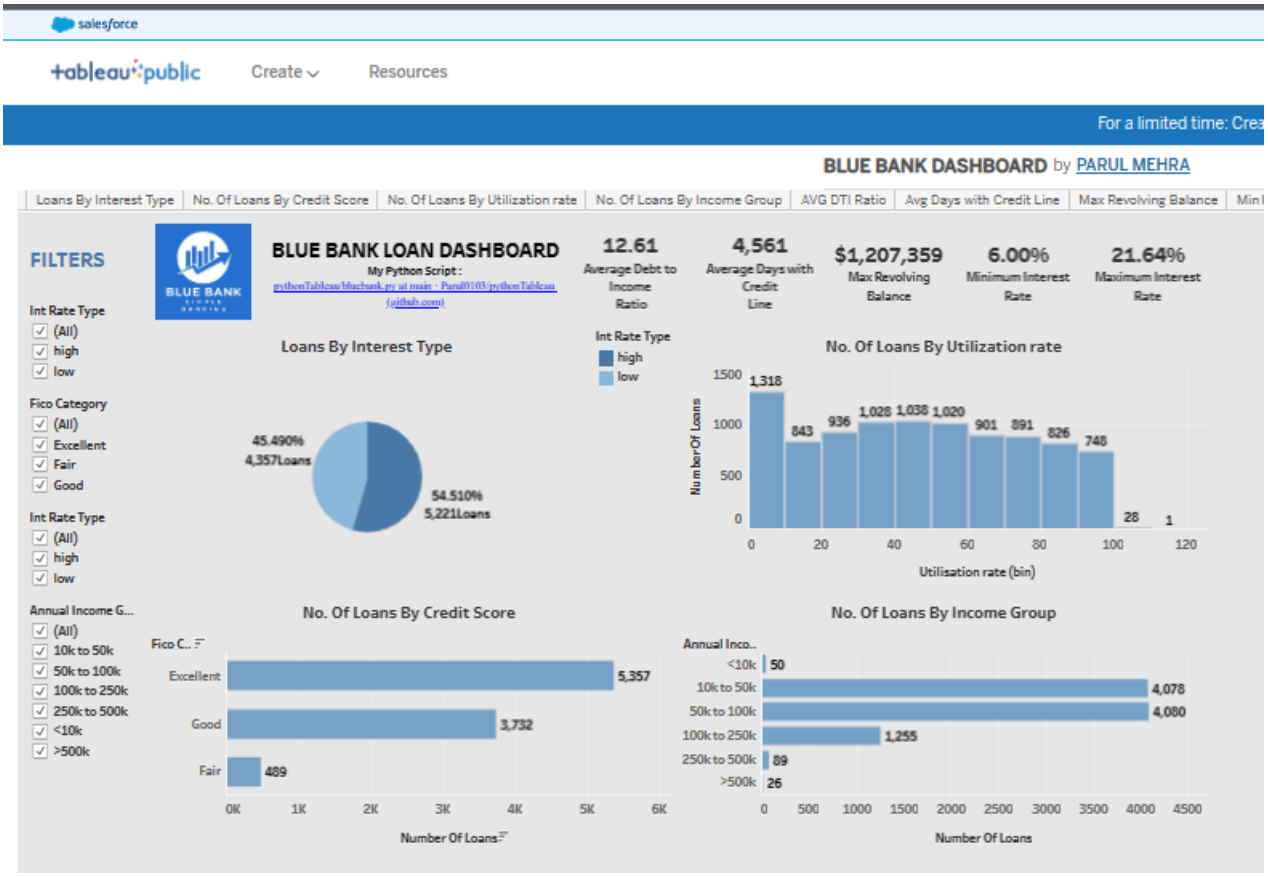

Figure 28 : Blue Bank Dashboard

3. TABLEAU DASHBOARD – VALUE INC

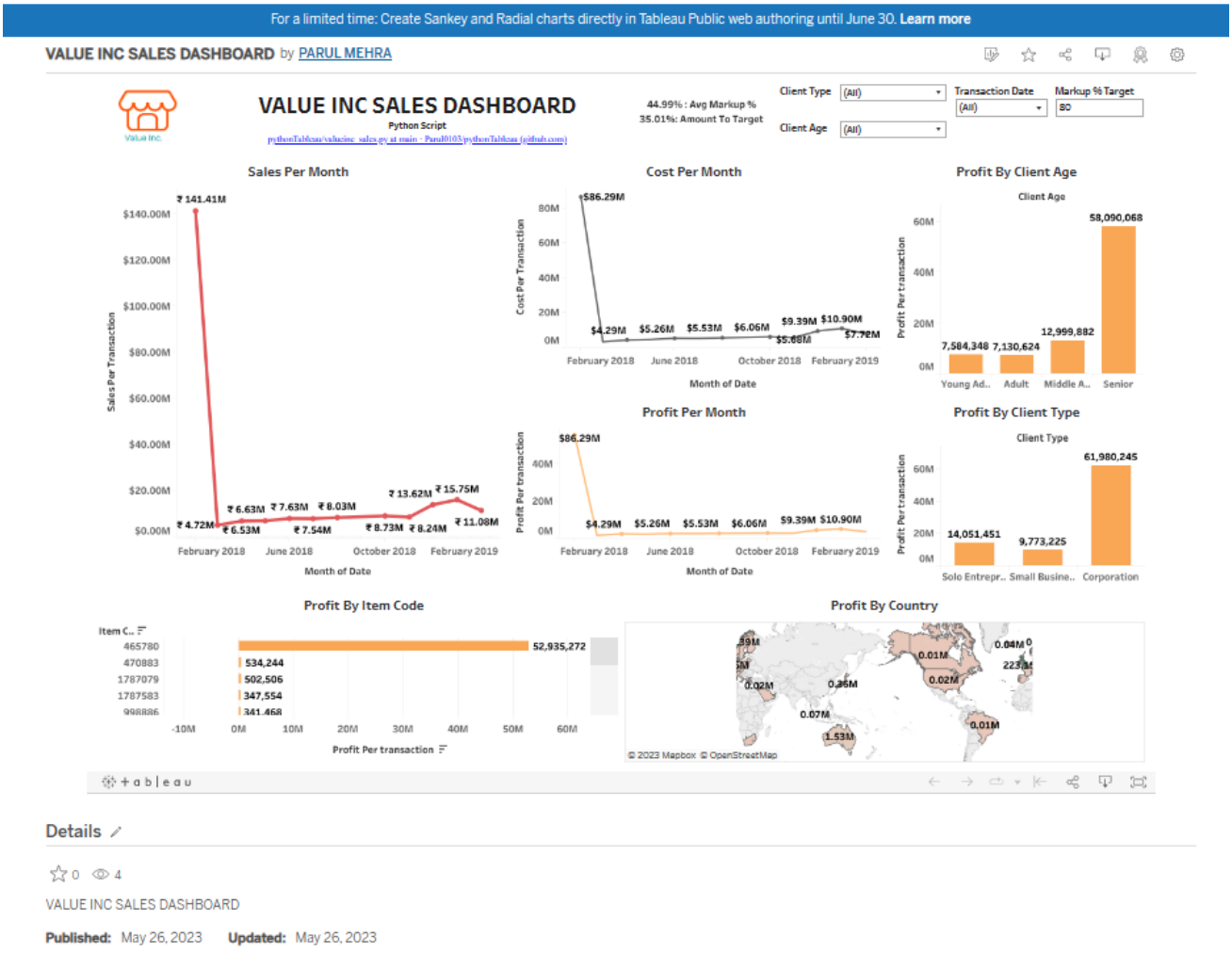

Figure 29 : Value Inc Dashboard

7.2 Personal Reflection

I learnt a lot upon completing this project. It's the very first project that I need to start from project specification and designing until documentation. It somehow gives us an image of how analytics on data is performed according to aim from beginning to end in the real industry. And by debugging some bugs that we had never encountered before also increased programming experience and self-value in the programming market.

And by completing this project, I had gained programming knowledge and faced some problems as well and to resolve those problems we obtained help from our mentors as well as the trainer who supported us in every part of the project. They provided some valuable opinions, and their knowledge helped us.

7.3 Future Scope

- **Advanced Analytics Techniques:** As data analytics continues to evolve, there is a constant need to explore and implement advanced analytics techniques. This can involve incorporating machine learning algorithms, predictive modeling, natural language processing, or deep learning techniques into the project. By expanding the scope of analytics techniques, you can uncover more insights and provide more valuable recommendations.

- **Real-Time Data Analytics:** With the increasing availability of real-time data streams from various sources such as IoT devices, social media platforms, and online transactions, there is a growing need to develop real-time data analytics capabilities. Integrating real-time data processing and analysis into the project can enable businesses to make faster and more informed decisions based on up-to-date information.

- **Big Data Analytics:** The growth of big data has created new challenges and opportunities for data analytics projects. Incorporating big data technologies, such as Apache Hadoop or Apache Spark, can enable the processing and analysis of large volumes of data. This can involve working with structured, semi-structured, and unstructured data sources, including text, images, and videos.

- **Data Visualization and Storytelling:** While Tableau is a powerful tool for data visualization, there is always room for enhancing the visual storytelling aspect of the project. Exploring interactive and dynamic visualizations, creating compelling data stories, and incorporating interactive dashboards can help in effectively communicating insights and engaging stakeholders.

- **Cloud Computing and Scalability:** Cloud computing offers scalability, flexibility, and cost-efficiency advantages for data analytics projects. Adopting cloud-based platforms, such as Amazon Web Services (AWS) or Microsoft Azure, can provide access to a wide range of scalable computing resources, storage capabilities, and analytics tools. Leveraging cloud services can facilitate the handling of large datasets and the execution of computationally intensive tasks.

- **Data Governance and Privacy:** As data privacy and governance regulations become more stringent, ensuring compliance and protecting sensitive information are critical. Exploring methods for data anonymization, encryption, access control, and auditing can enhance the project's security and compliance measures.

- **Integration with Business Processes:** Integrating data analytics and visualization with core business processes can drive actionable insights and support data-driven decision-making across the organization. Aligning analytics projects with specific business objectives and integrating them into existing workflows and systems can maximize the project's impact and create a culture of data-driven decision-making

7.4 CONCLUSION

The development of the application has been successfully completed with the results of all the tests being very positive with minute bugs found and fixed. There have been no issues regarding the application and is serving its purposes very efficiently. This project gives an insight to how to analyze the data and derive appropriate results from the analyzed data along with how to visualize data using a powerful visualization tool – tableau. This project helped me improve basic coding as well as the usage of various libraries in python

7.5 REFERENCES

- Introducing Data Science: Big data, machine learning, and more, using Python tools
- Data Analytics Models and Algorithms for Intelligent Data Analysis
- Learning Tableau: A data visualization tool by Steven Batt ,Tara Grealis , Oskar Harmon & Paul Tomolonis
- 201 Exploratory Data Analysis using Python Kabita Sahoo, Abhaya Kumar Samal, Jitendra Pramanik, Subhendu Kumar Pani. International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8, Issue-12, October 201
- Process Mining for Python (PM4Py): Bridging the Gap Between Process- and Data Science Alessandro Berti, Sebastiaan J. van Zelst, Wil van der Aalst