A PROJECT REPORT

on

"ANALIZIO - A BUSINESS ANALYTICS AND VISUALIZATION WEBSITE"

Submitted to

KIIT Deemed to be University

In Partial Fulfillment of the requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER SCIENCE AND ENGINEERING

BY

SAYANTAN BANERJEE – 2005195
CHANDRALI SHYAM – 2005720
SHAUNAK CHANDRA – 2005757
TARANGINI SINHA – 2005768

UNDER THE GUIDANCE OF
DR. PRATYUSA MUKHERJEE

SCHOOL OF COMPUTER ENGINEERING
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY
BHUBANESWAR, ODISHA- 751024
MAY 2023

KIIT DEEMED to be UNIVERSITY
School of Computer Engineering
Bhubaneswar, ODISHA 751024

CERTIFICATE

This is to certify that the project entitled

“ANALIZIO - A BUSINESS ANALYTICS AND VISUALIZATION WEBSITE”

Submitted by

SAYANTAN BANERJEE – 2005195
CHANDRALI SHYAM – 2005720
SHAUNAK CHANDRA – 2005757
TARANGINI SINHA – 2005768

is a record of bona fide work carried out by them, in the partial fulfillment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science and Engineering) at KIIT Deemed to be university, Bhubaneswar. This work is done during the year 2022-2023, under our guidance.

Date: 05 / 04 / 2023

Dr. Pratyusa Mukherjee
Project Guide

ACKNOWLEDGEMENT

We would like to extend our sincere gratitude to our **Dr. Pratyusa Mukherjee** from the **School of Computer Engineering** for her invaluable guidance and unwavering support throughout the duration of this project. From the beginning to the end, this project's success has been largely attributed to her guidance and competence. We are truly fortunate to have her as our mentor, and we will always be grateful for her contributions to our success.

Sayantan Banerjee
Chandrali Shyam
Shaunak Chandra
Tarangini Sinha

ABSTRACT

Any business's ability to expand and endure depends heavily on the sales sector. Sales reports assist firms pinpoint opportunities for development and progress by offering insightful information about sales activity, trends, and patterns. For non-technical individuals, preparing these reports can be a time-consuming and difficult effort.

This project attempts to address this issue by creating an application that enables users to quickly and easily construct sales reports without the need for technical knowledge. To assist customers in analyzing their sales data, including daily sales, product-based sales, and refunds, the program will include a variety of data visualization capabilities. Users will be able to forecast upcoming sales trends and determine which product categories perform the best. This project is a significant contribution to the field of data visualization, making it easier and more accessible for people to understand complex data sets. The tool's flexibility and customizable options allow users to create visualizations that best suit their needs, making it a versatile and valuable resource for anyone who needs to analyze data.

Keywords: Sales analysis, Data visualization, Flask, Pandas, Matplotlib, Web development, Predictive analytics, Descriptive analytics, Data Preprocessing, Feature engineering.

CONTENTS

LIST OF FIGURES

Chapter 1:
Introduction

Data visualization [1] is an essential part of data analytics [3] as it allows the user to represent even complex data sets into simple visualized bar graphs, pie charts, line graphs etc. With increased amounts of data and its increasing complexity the need for developing a tool to simplify this data into simple and understandable visual representations [1] has increased.

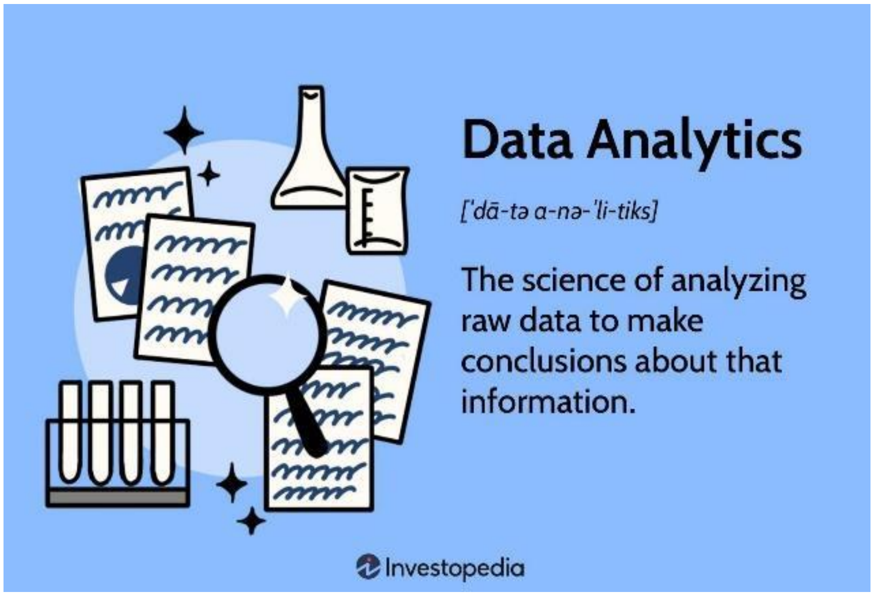

Data Analytics
['dā-tə a-nə-'li-tiks]
The science of analyzing raw data to make conclusions about that information.
Investopedia

Fig 1: Data analytics definition

In today's competitive business environment, having access to real-time data and analytics [3] is essential for making informed decisions. In order for a business to expand and succeed, sales analytics are essential. It offers information on consumer behavior, aids in the discovery of trends, and enables the optimization of sales procedures. But analyzing sales data can be difficult, particularly for non-technical staff who might not have the resources or skills to do it.

Thus Analizio is developed, it is an web application that offers organizations an interactive platform for data analysis and insight gathering, assisting them in making choices based on data.Flask and Jinja are used to create the server, with

Python libraries like NumPy, Matplotlib, and Pandas used in the backend and HTML and CSS in the frontend.

This project is aimed at developing a tool [6] to accelerate visual representation [1] and understanding of data of any given business when their data is provided in a formatted CSV file [2]. Our website inclines towards users to upload a CSV file in the specified format and represent their sales, profits, loss and highest grossing products through visual representation and a short summary.

The project has tried to integrate the concepts of Data Analysis [3-6] using python and web development to create a powerful tool in the form of a website that can help various businesses to visualize their data and have a better understanding about their sales, profits and losses. This project can further be developed to improve the model and implement more variations of visualization [1] along with detailed reports.

Chapter 2:
Basic Concepts

2.1 Background Study on Available Business Analytical Models.

To continue with this project the team had to learn about widely known and large-scale data visualization [1] applications available in the market and understand the needs and criteria for the proper execution of the model. Data Voyager [9] and Tableau [7] are the two most reputed data visualization [1] applications present in the industry as of now. They were studied thoroughly to understand and implement the needed features for a well-developed data visualization application.

2.1.1 Data Voyager:

Designed for exploratory data research, Data Voyager [9] is an interactive visualization tool. Users can upload datasets in CSV [2] format to Data Voyager, a tool created by Jeffrey Heer and his colleagues at the University of Washington, and use it to interactively explore those datasets using linked visualizations. To get insights into the underlying patterns and trends, the tool enables users to aggregate the data, filter it, and change the visualizations instantly. It's a popular option for activities requiring data exploration and analysis because of its simple, user-friendly design.

2.1.2 Tableau:

Tableau [7] is a powerful and widely-used data visualization tool that allows users to connect to and analyze data from various sources, including CSV [2] files. The tool, created by Tableau Software, offers a simple interface for building interactive dashboards and visualizations that let users explore and learn from their data. Tableau [7] offers users a variety of customization choices for structuring and decorating their visualizations. It supports a wide variety of visualization [1] styles, including maps, scatter plots, and bar charts. Tableau is a popular option for companies, analysts, and data scientists since it provides broad capabilities for data preparation, analysis, and collaboration.

2.2 Background studies on technologies used

2.2.1 Data Analytics

Data analytics [3-6] entails analyzing data using statistical and computational methods to derive insightful knowledge. To find patterns, trends, and correlations that can guide corporate decisions and strategies, a huge volume of data must be gathered, processed, and analyzed. Businesses can understand their consumers, market trends, and operational performance better by implementing data analytics approaches. This data can be utilized to enhance decision-making, uncover areas for development and innovation, and optimize corporate operations. Data analytics [3-6] are becoming increasingly important as a tool [6] for gaining insights and promoting company performance as a result of the continued growth in the amount of data collected by firms.

2.2.2 Data Visualization

Data visualization [1] is the act of employing charts, graphs, and other visual components to visually depict large data sets in order to make the information easier to understand for viewers. Data visualization [1] aims to convey insights and patterns in the data in a way that is understandable, entertaining, and educational. Analysts and decision-makers can use data visualization techniques to spot trends, connections, and outliers that may not be obvious from a straightforward tabular format. Users may quickly understand complex information, spot trends and anomalies, and share insights with others in a way that is clear and understandable by visualizing data.

2.2.3 Web Development

The process of creating and maintaining websites and web applications is referred to as web development. Designing the user interface and user experience, building code to provide functionality and interactivity, and interacting with backend systems like databases and APIs are all included in this. A range of computer languages, frameworks, and tools are used by web developers to build secure, scalable, and adaptable websites and applications. The aim of web development is to build a website or application that satisfies the client's or end user's needs, is usable across a variety of platforms and browsers, and offers a smooth user experience.

2.2.4 Python

Python [8] is a high-level, interpreted programming language that places a strong emphasis on the simplicity, readability, and adaptability of its code. It is frequently used in a number of industries, including artificial intelligence, data analysis [3], web development, and scientific computing. Python is an approachable language for novices and a useful tool for seasoned programmers due to its syntax's straightforward design and ease of reading. It contains a large ecosystem of libraries and frameworks, supports object-oriented, functional, and procedural programming paradigms, and can be used for a variety of applications. Many developers and organizations prefer Python [8] because of its widespread use and simplicity.

2.2.5 Matplotlib and Pandas

A common Python data visualization library called Matplotlib [1] is used to produce various kinds of graphs, charts, and plots. It supports numerous data formats, including CSV [2], Excel, and SQL databases, and offers a broad range of customization options. Users can produce excellent data visualizations with Matplotlib [1] that are simple to understand and communicate insights from data.

Python's powerful data analysis library, Pandas [8], is used to manipulate and analyze huge, complex datasets. For data cleaning, exploration, and transformation, it offers simple data structures and tools for data analysis [3-6]. Pandas allows users to easily filter, sort, and summarize data as well as carry out a number of statistical operations on the data. Additionally, it supports a broad range of data formats, making it the go-to library for data scientists and analysts.

2.3 Motivation of our work

Data visualization [1] tools such as Tableau [7] and Data Voyager [9] have been around for a while now and are widely used by businesses of all sizes to analyze their data. However, these tools can be difficult to use and understand, for someone who is unfamiliar with data science, and their need for sizable datasets to produce useful insights can be a bit problematic. These products can also be pricey, which limits their availability to small firms. This project offers small businesses a compact, easy-to-use data visualization solution. This system will give small companies the ability to submit their sales data in CSV [2] format and produce a basic report on their items as well as a visualized graph [1] of their overall sales. Without the need for pricey software or in-depth data science expertise, this will assist small firms in making data-driven decisions and gaining insights into their sales patterns.

2.4 Objectives of this Project:

- To develop a user-friendly interface that allows users to select a dataset. Create a backend system that can handle different types of datasets [2].
- To provide a range of visualizations that are suitable for different types of datasets.
- To ensure the accuracy and reliability of the visualizations [1] produced by the model.
- To evaluate the performance of the application in terms of efficiency and usability.

Chapter 3:
Problem Statement

Any organization's decision-making process depends heavily on sales analysis, especially when developing plans to increase sales and profitability. Due to their lack of resources, knowledge, and access to appropriate tools, small businesses in particular may find it difficult to analyze sales data. Additionally, it can be challenging to spot trends and opportunities because many organizations may have multiple product lines, each with distinct sales characteristics. As a result, businesses might pass up opportunities for potential growth, which would hurt their overall performance in the market. Therefore, there is a need for a solution that can assist small businesses in effectively and efficiently analyzing their sales data in order to pinpoint important insights and growth prospects.

3.1 PROJECT PLANNING

(1) Define project scope and objectives

The scope of this project is to develop a sales analysis tool using Python libraries such as Matplotlib and Pandas to provide insights into a company's sales data. The objective is to assist small businesses in identifying sales trends and opportunities for growth across various product lines.

(2) Requirement Gathering

The project will be developed using a python background using Flask and other python libraries like Pandas, Matplotlib and Seaborn. Markup languages like HTML and CSS will be used.

(3) Create project plan

- Develop a user-friendly interface that allows users to select a dataset. Create a backend system that can handle different types of datasets.
- Provide a range of visualizations that are suitable for different types of datasets.
- Ensure the accuracy and reliability of the visualizations produced by the model.

- Evaluate the performance of the application in terms of efficiency and usability.

(4) Application Development

Code was written to create the application and integrate all the required functionalities. Appropriate frameworks and libraries were chosen based on the project requirements. A user-friendly interface for the application was also created.

(5) Application Testing

The application was tested based on several datasets of similar kind and it was ensured that it meets all the requirements. Few bugs were also caught in the process which was rectified later.

(6) Document the project

Document the project including its purpose, functionality, design, testing and evaluation. This document can serve as a reference for future work and can be used to improve the application.

3.2 PROJECT ANALYSIS

Functional Requirements: -

(1) User Interface

- The application should have a user-friendly interface that allows users to select a dataset and generate visualizations.
- The user interface should be intuitive and easy to navigate.
- The user should be able to upload a dataset in CSV format.
- The user should be able to select the type of visualization to be generated.

(2) Backend System

- The backend system should be able to handle different types of datasets.
- The backend system should be able to process the dataset and generate the selected visualization.
- The backend system should ensure the accuracy and reliability of the visualizations produced.

(3) Visualization

- The application should provide a range of visualizations suitable for different types of datasets.
- The visualizations should be accurate and reliable.
- The user should be able to customize the visualization (e.g., change the color scheme, adjust the scale).

(4) Data Preprocessing

The application should preprocess the data by handling missing data, imputing outliers, and normalizing the data before generating visualizations.

(5) Exporting

The user should be able to export the generated visualizations in various formats (e.g., PNG, PDF).

<u>Non-Functional Requirements</u>: -

1. Performance

- The application should be efficient and responsive.
- The application should generate visualizations within a reasonable amount of time.

2. Security

- The application should be secure and protect user data.
- The application should prevent unauthorized access to the backend system and data.

3. Usability

- The application should be user-friendly and easy to use.
- The application should provide clear instructions and feedback to the user.

3.3 SYSTEM DESIGN

3.3.1 Design Constraints

The constraints of the applications are as follows:

(1) Technology Stack

- ➢ The application should be developed using Python and its libraries only (e.g., NumPy, Pandas, Matplotlib, Flask).
- ➢ The frontend should be developed using HTML, CSS and JavaScript.
- ➢ The application should be hosted on a cloud platform (e.g., Heroku).

(2) Dataset Size

- ➢ The application should be able to handle datasets of up to 10,000 rows and 20 columns.

3.3.2 Flow of user data

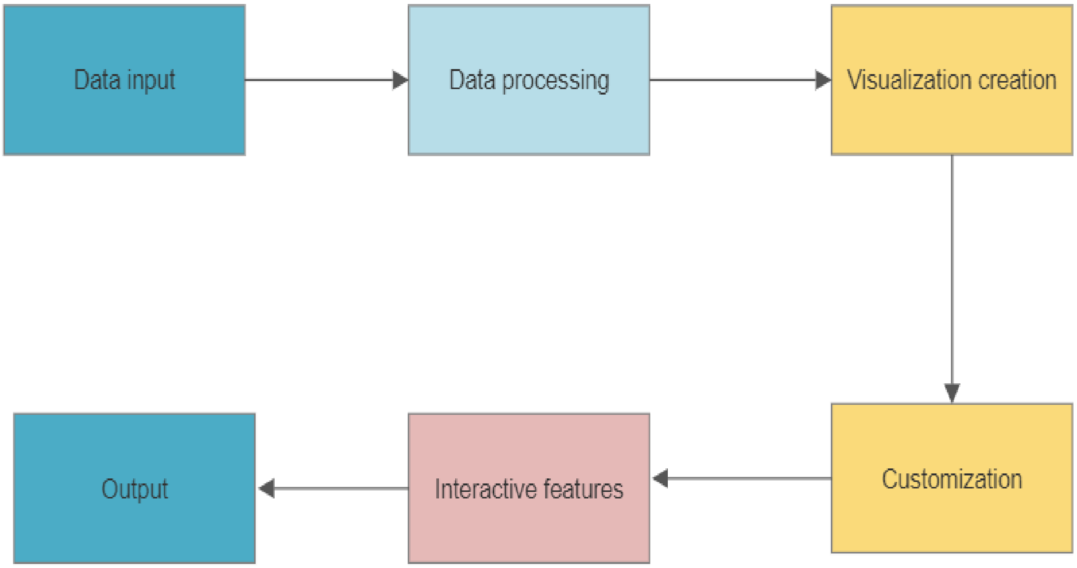
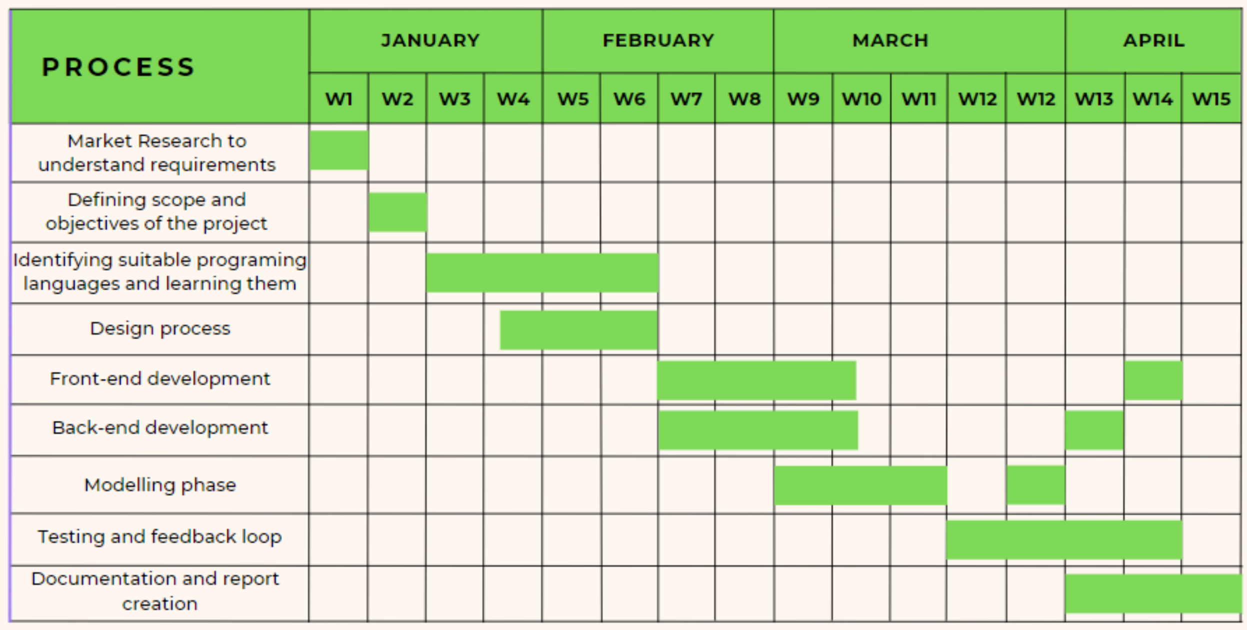


Fig 2: Flow of data on user side

3.3.3 Gantt Chart



PROCESS	JANUARY				FEBRUARY				MARCH					APRIL		
	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W12	W13	W14	W15
Market Research to understand requirements	🟩															
Defining scope and objectives of the project		🟩														
Identifying suitable programing languages and learning them			🟩	🟩	🟩	🟩										
Design process				🟩	🟩	🟩										
Front-end development							🟩	🟩	🟩	🟩					🟩	
Back-end development							🟩	🟩	🟩	🟩				🟩		
Modelling phase									🟩	🟩	🟩		🟩			
Testing and feedback loop												🟩	🟩	🟩	🟩	
Documentation and report creation														🟩	🟩	🟩

Fig 3: Gantt Chart for development of application

Chapter 4: IMPLEMENTATION

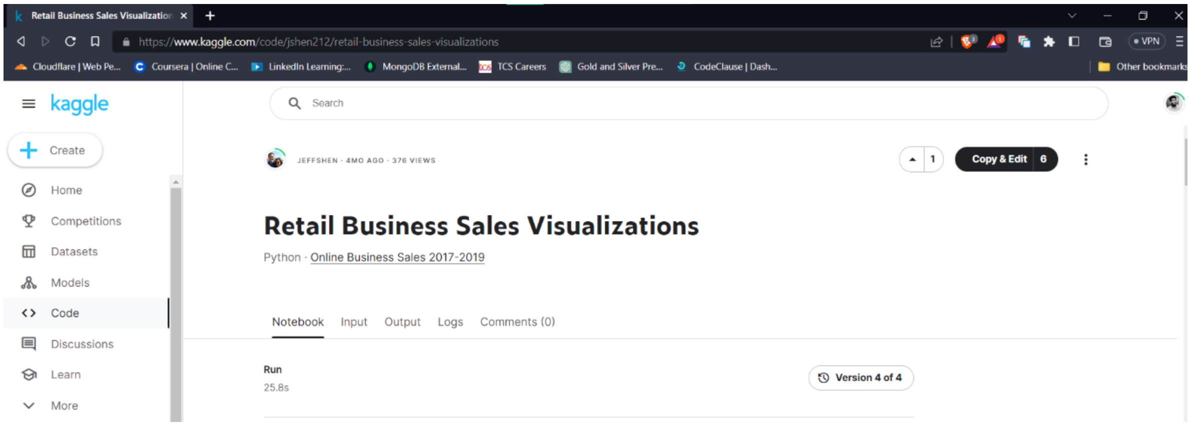
The implementation was carried out in these basic steps: -

4.1 Data Collection and Exploration

The well-known website Kaggle provides access to a wide variety of datasets from a variety of industries, including banking, healthcare, and social media. Before users can start downloading data from Kaggle, they must register and log in to the platform. After logging in, they can browse the datasets that are available or use keywords to search for specific ones.

The dataset used is a business retail dataset that contains information on Product Type, Net Quantity, Gross Sales, Discounts, Returns and Total Net Sales. The data types used here are numeric and categorical values and there are a total of 1777 data entries with 8 missing data in the Product type column.

Descriptive statistics and data visualizations are to be used to study and analyze a dataset to draw conclusions. These methods enable the detection of patterns and trends as well as the extraction of insightful information for decision-making.



Retail Business Sales Visualizations

Python · Online Business Sales 2017-2019

JEFFSHEN · 4MO AGO · 376 VIEWS

Run

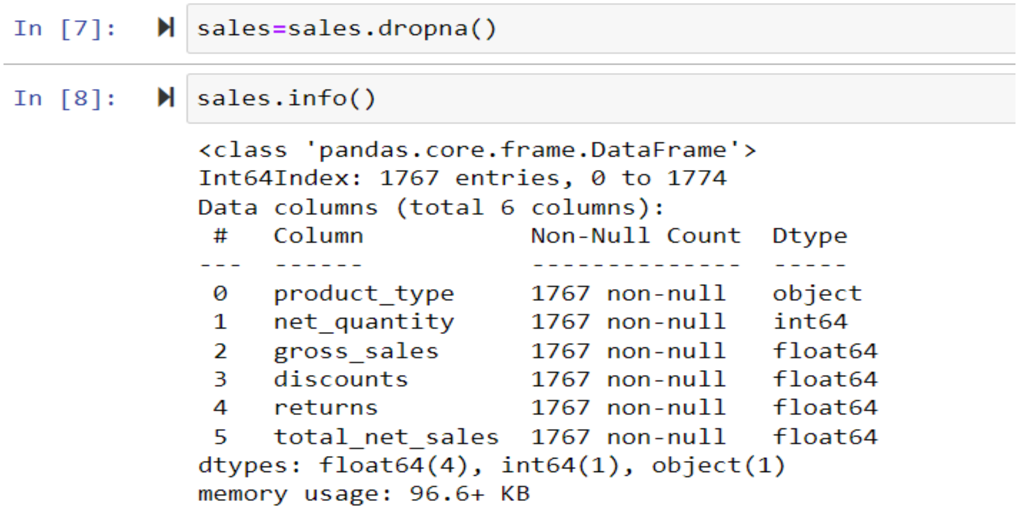
25.8s

Fig 4: Snapshot of the dataset from Kaggle

4.2 Data Preprocessing

Any type of processing done on raw data to get it ready for another data processing operation is referred to as data preprocessing, which is a part of data preparation. It has historically been a crucial first stage in the data mining process.

In the data set used, there were 8 missing values within one column. This was rectified by preprocessing the data using the removal of the missing values. Although the normalization technique is frequently used to take into account data variations, it has not been required in this case due to the small amount of variation present. As part of the project analysis, examination of the data and establishing relations between the different data columns is done, drawing relevant inferences as required.



```
In [7]:   sales=sales.dropna()

In [8]:   sales.info()

          <class 'pandas.core.frame.DataFrame'>
          Int64Index: 1767 entries, 0 to 1774
          Data columns (total 6 columns):
           #   Column           Non-Null Count  Dtype
          ---  ------           --------------  -----
           0   product_type     1767 non-null   object
           1   net_quantity     1767 non-null   int64
           2   gross_sales      1767 non-null   float64
           3   discounts        1767 non-null   float64
           4   returns          1767 non-null   float64
           5   total_net_sales  1767 non-null   float64
          dtypes: float64(4), int64(1), object(1)
          memory usage: 96.6+ KB
```

Fig 5: Data set columns after cleaning

4.3 Data Visualization

The provided information shows the net quantity, gross sales, discounts, returns, and total net sales for a range of goods, including baskets, Christmas decorations, artwork, and sculpture. Several data visualization techniques, including bar charts, scatter plots, stacked bar charts, pie charts, heatmaps, line graphs, and area graphs, can be used to effectively analyze and present this data.

Python has a variety of data visualization libraries that can be used to create bar charts, scatter plots, pie charts, heatmaps, line charts and area charts. Two of the most common libraries in python include Matplotlib and Seaborn.

The Matplotlib module is popularly used to produce Python visualizations. Bar graphs, scatter plots, pie graphs, line graphs, and area graphs are just a few of the graphs and charts that may be created using the wide range of tools that are available. Matplotlib is an excellent choice for creating sophisticated and polished visualizations because of its flexibility.

Seaborn is a popular Python data visualization library. It builds on Matplotlib and provides a higher-level interface for creating visualizations. The visualizations look more polished and professional by using one of the many pre-installed themes and color schemes in Seaborn. Using the pie function and the heatmap function, respectively, you can produce pie charts and heatmaps. Area and line charts can be made using the plot function.

Utilizing these tools to visualize different parts of the sales data, including daily sales, product-by-product sales, returns, and product categories, is a key component of the project's data visualization phase. These visualizations will allow anyone to find patterns and trends in the data, which can guide their decision-making and help them enhance their sales approach.

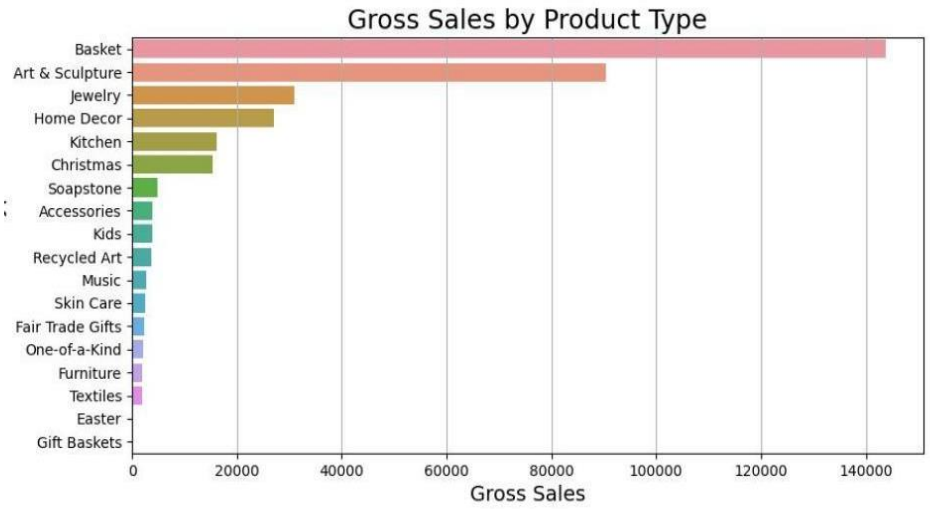


Fig 6: Bar graph of Gross Sales by Product type

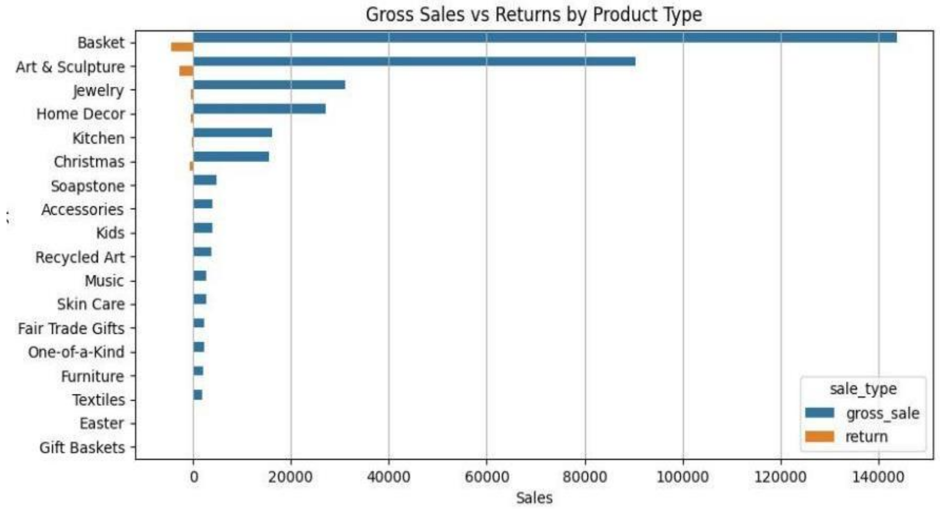


Fig 7: Bar graph of Gross sales vs returns by product type

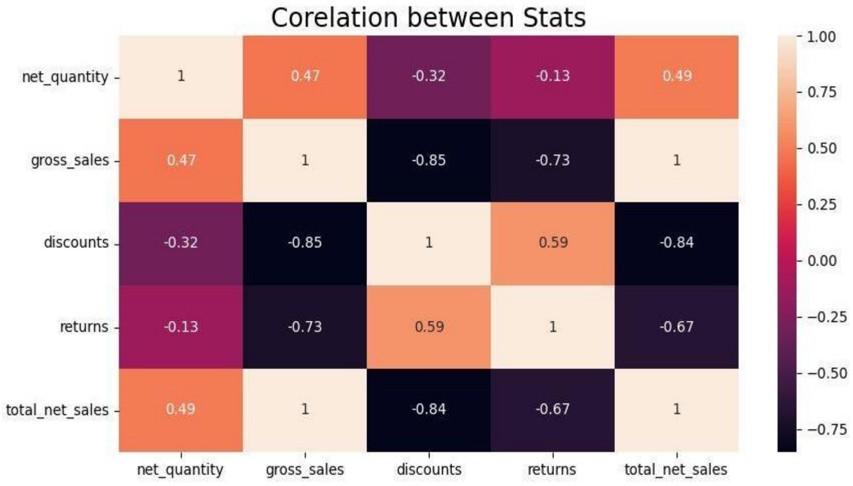
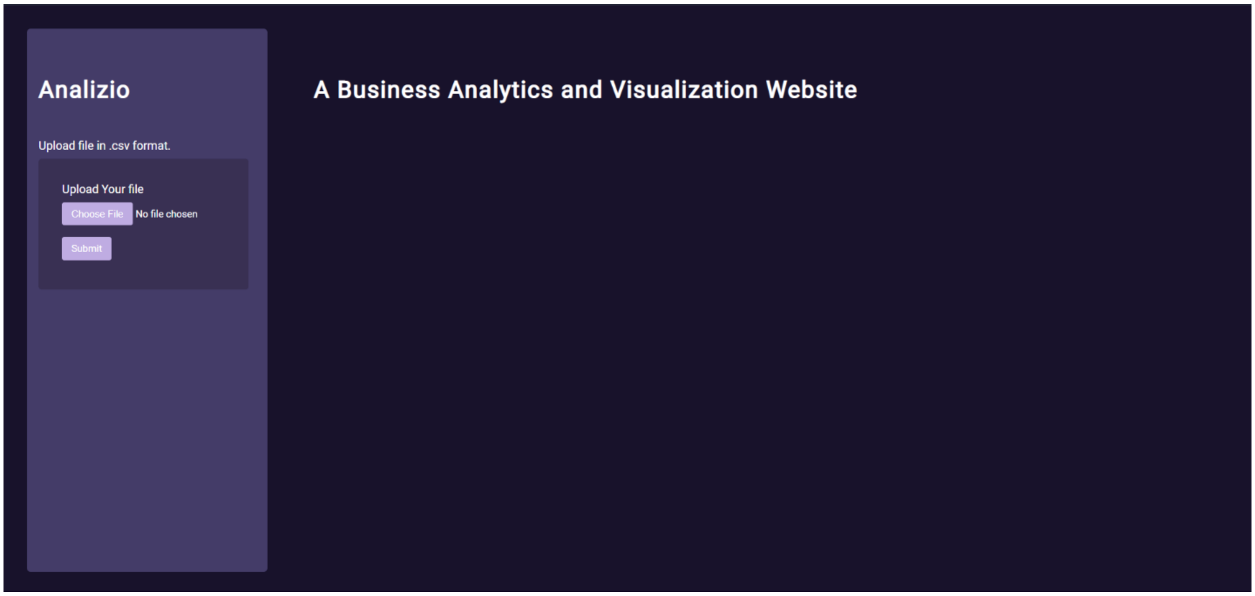


Fig 8: Correlation matrix between the different attributes of data set

4.4 Coding

For the frontend, HTML, CSS, and JavaScript is used with the Bootstrap framework for responsive design. It also uses Jinja2 templating engine to dynamically render HTML pages with data from the backend. The user can select a dataset from the frontend and the corresponding charts are displayed on the same page using AJAX calls to the backend.



Analizio

A Business Analytics and Visualization Website

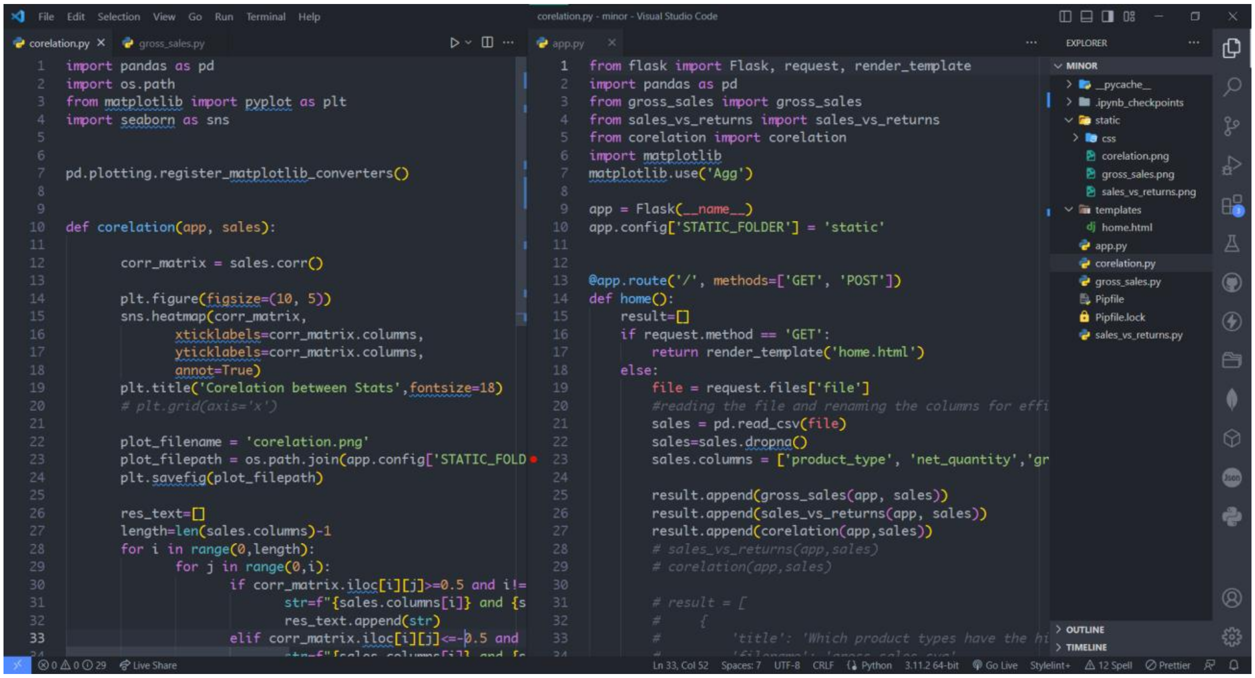
Upload file in .csv format.

Upload Your file

Fig 9: Snapshot of Website

The app's backend is in charge of handling frontend requests and delivering the necessary responses. It is created with Flask, a Python web framework that is small and lightweight. The Flask application defines a number of endpoints that deal with various requests, such as getting data from a database, processing it, and returning it in a particular format.

In this project, Flask is responsible for handling requests coming from front-end which generally consists of the data set provided by the user. After this the Flask server calls appropriate functions for chart generation and then returns the result in HTML format.



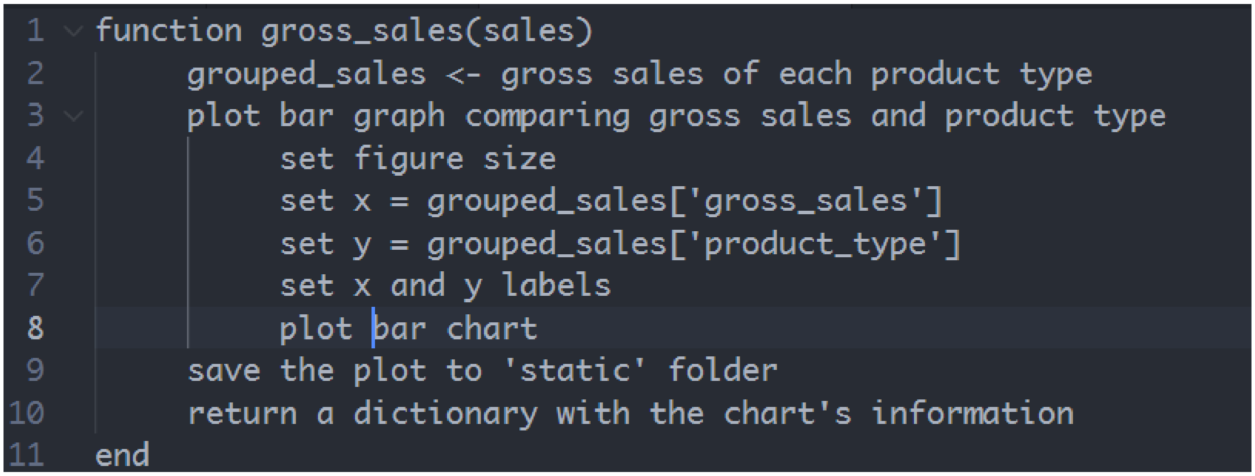
```
import pandas as pd
import os.path
from matplotlib import pyplot as plt
import seaborn as sns


pd.plotting.register_matplotlib_converters()


def corelation(app, sales):

        corr_matrix = sales.corr()

        plt.figure(figsize=(10, 5))
        sns.heatmap(corr_matrix,
                xticklabels=corr_matrix.columns,
                yticklabels=corr_matrix.columns,
                annot=True)
        plt.title('Corelation between Stats',fontsize=18)
        # plt.grid(axis='x')

        plot_filename = 'corelation.png'
        plot_filepath = os.path.join(app.config['STATIC_FOLD
        plt.savefig(plot_filepath)

        res_text=[]
        length=len(sales.columns)-1
        for i in range(0,length):
                for j in range(0,i):
                        if corr_matrix.iloc[i][j]>=0.5 and i!=
                                str=f"{sales.columns[i]} and {s
                                res_text.append(str)
                        elif corr_matrix.iloc[i][j]<=-0.5 and
```

```
from flask import Flask, request, render_template
import pandas as pd
from gross_sales import gross_sales
from sales_vs_returns import sales_vs_returns
from corelation import corelation
import matplotlib
matplotlib.use('Agg')

app = Flask(__name__)
app.config['STATIC_FOLDER'] = 'static'


@app.route('/', methods=['GET', 'POST'])
def home():
    result=[]
    if request.method == 'GET':
        return render_template('home.html')
    else:
        file = request.files['file']
        #reading the file and renaming the columns for effi
        sales = pd.read_csv(file)
        sales=sales.dropna()
        sales.columns = ['product_type', 'net_quantity','gr

        result.append(gross_sales(app, sales))
        result.append(sales_vs_returns(app, sales))
        result.append(corelation(app,sales))
        # sales_vs_returns(app,sales)
        # corelation(app,sales)

        # result = [
        #     {
        #         'title': 'Which product types have the hi
```

Fig 10: Snippet of code in the backend part

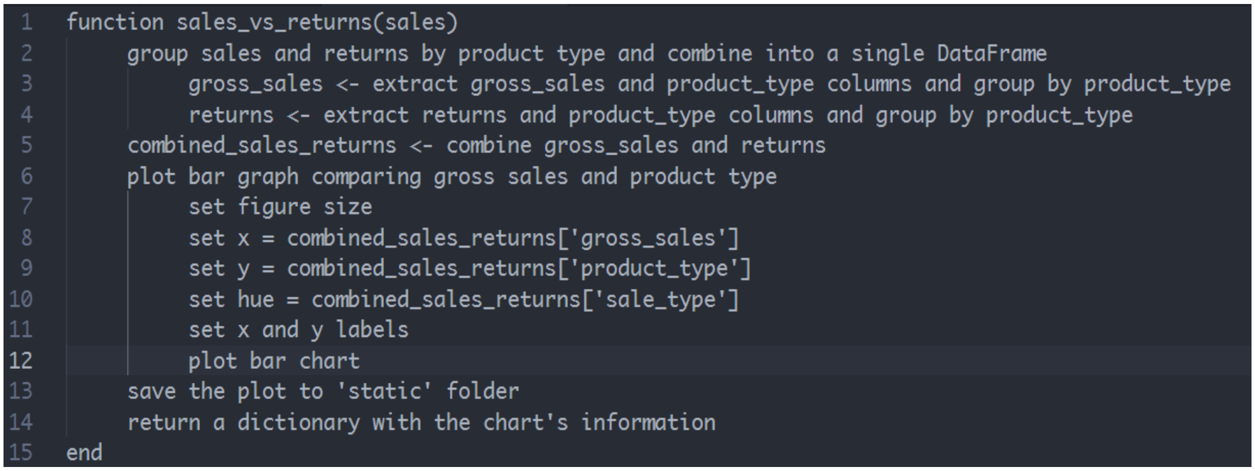
In the backend of the application few function calls were made which can generate charts and then they are saved at a folder named “static”. This is a special folder which is used to keep static files such as images, stylesheets, and JavaScript files. This is done for performance reasons, as serving static files directly is usually faster than processing a route handler function.

The pseudo codes of the functions that generate the charts are given below:



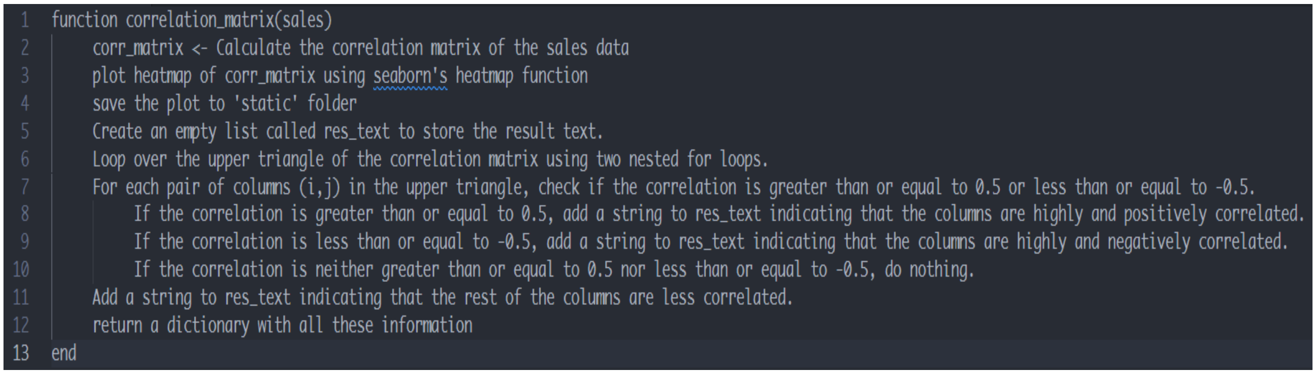
```
function gross_sales(sales)
    grouped_sales <- gross sales of each product type
    plot bar graph comparing gross sales and product type
        set figure size
        set x = grouped_sales['gross_sales']
        set y = grouped_sales['product_type']
        set x and y labels
        plot bar chart
    save the plot to 'static' folder
    return a dictionary with the chart's information
end
```

Fig 11: Pseudocode of gross_sales function



```
function sales_vs_returns(sales)
    group sales and returns by product type and combine into a single DataFrame
        gross_sales <- extract gross_sales and product_type columns and group by product_type
        returns <- extract returns and product_type columns and group by product_type
    combined_sales_returns <- combine gross_sales and returns
    plot bar graph comparing gross sales and product type
        set figure size
        set x = combined_sales_returns['gross_sales']
        set y = combined_sales_returns['product_type']
        set hue = combined_sales_returns['sale_type']
        set x and y labels
        plot bar chart
    save the plot to 'static' folder
    return a dictionary with the chart's information
end
```

Fig 12: Pseudocode of sales_vs_returns function



```
function correlation_matrix(sales)
    corr_matrix <- Calculate the correlation matrix of the sales data
    plot heatmap of corr_matrix using seaborn's heatmap function
    save the plot to 'static' folder
    Create an empty list called res_text to store the result text.
    Loop over the upper triangle of the correlation matrix using two nested for loops.
    For each pair of columns (i,j) in the upper triangle, check if the correlation is greater than or equal to 0.5 or less than or equal to -0.5.
        If the correlation is greater than or equal to 0.5, add a string to res_text indicating that the columns are highly and positively correlated.
        If the correlation is less than or equal to -0.5, add a string to res_text indicating that the columns are highly and negatively correlated.
        If the correlation is neither greater than or equal to 0.5 nor less than or equal to -0.5, do nothing.
    Add a string to res_text indicating that the rest of the columns are less correlated.
    return a dictionary with all these information
end
```

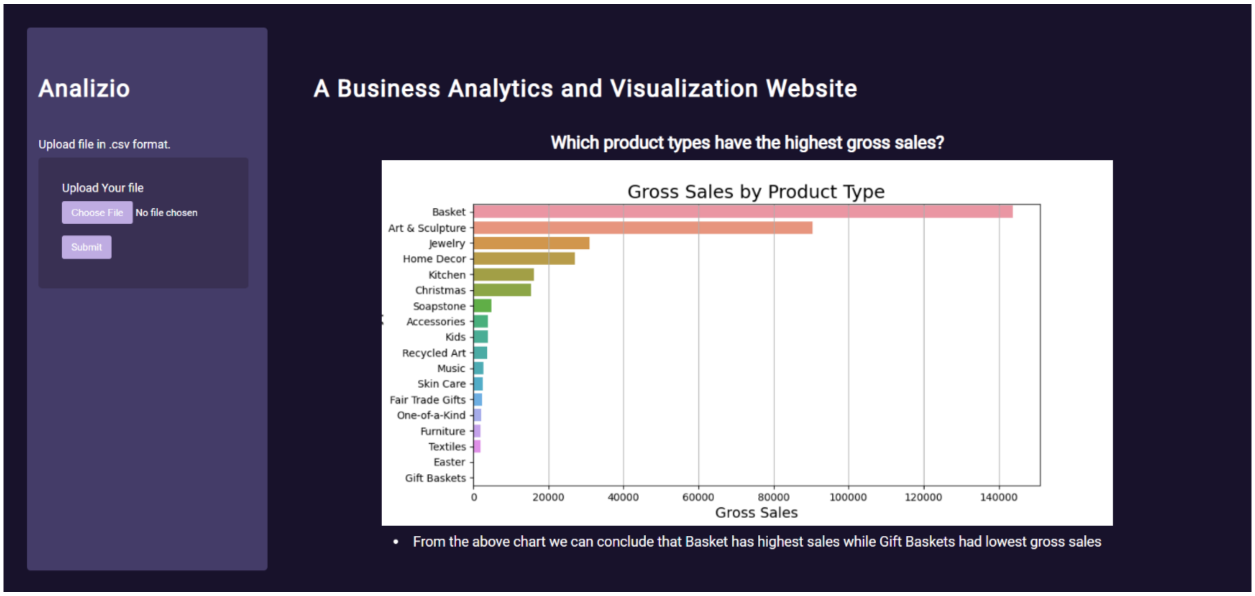
Fig 13: Pseudocode of correlation_matrix function

4.5 Integration

To ensure that this application runs smoothly, utmost importance is given to the integration of backend with the frontend in the project. Connecting the data sources, backend operations, and frontend visualization tools are all parts of the integration process. It also involves putting into practice certain python functions that produce the visualizations depending on the input data.

The Jinja templating engine and the Flask web framework has been used to integrate the project's frontend and backend. Flask offers a straightforward and adaptable solution to create web applications, whereas Jinja enables us to develop dynamic HTML templates that can be filled with information from the backend.

Flask routes are created that correlate to various frontend pages or activities. For our project, the backend can make use of Python routines to generate the charts and save them to the static subdirectory. When needed, Flask can deliver these static files straight to the user's browser. A seamless and effective user experience can be developed that enables users to meaningfully interact with their data by linking the frontend and backend.



Analizio

Upload file in .csv format.

Upload Your file

A Business Analytics and Visualization Website

Which product types have the highest gross sales?

- From the above chart we can conclude that Basket has highest sales while Gift Baskets had lowest gross sales

Fig 14: The Application returns the necessary analysis along with visual representations

4.6 Documentation

This HTML and CSS-based data visualization tool project is a robust and easy-to-use tool for visualizing data. It was created with Python and flask framework in the backend and HTML and CSS in the frontend. The application provides output in the form of various graphs, pie charts, heatmaps, and other visualization options by allowing users to input data in the form of CSV or Excel files. The tool also offers insightful data insights and patterns, enabling users to base their judgements on their analyses.

Overall, this initiative makes an important contribution to the field of data visualization by facilitating people's understanding of complex data sets and increasing accessibility. Anyone who needs to analyze data can use the tool because of its adaptability and configurable features, which let users design visualizations that best meet their needs.

After completing this project, the conclusion has been drawn that in order to make data-driven decisions and keep up with the competition, every organization needs to invest in a business analysis and visualization tool project. Organizations may learn vital information about their operations, clients, and rivals by utilizing the newest tools and approaches in the field of business research and visualization.

For successful business analysis and visualization tool projects, business analysts and visualization specialists should collaborate to ensure that the tool satisfies the unique requirements of the enterprise. To meet shifting business requirements and data sources, the tool must be simple to use, adaptable, and extensible.

Chapter 5:
STANDARDS ADOPTED

The following principles and best practices have been adopted to direct the creation and implementation of our data visualization website:

5.1 Design Standards

5.1.1 Data Visualization Standards:

To effectively communicate insights and trends in the data, the best practices were used, recommended by the industry, including the use of the proper chart kinds, color schemes, and labeling.

5.2 Coding Standards

5.2.1 JavaScript (JS) Coding Standards:

"JavaScript Coding Standards" was implemented for this project to ensure excellent and uniform code. These standards, which promoted effective development, covered name conventions, formatting, commenting, and error prevention. By following these guidelines, it was made sure that the JavaScript code compiled with best practices and was simple for other programmers to understand.

5.2.2 Python Standards:

To make sure that the Python code was consistent, readable, and simple to maintain, the Python Software Foundation's PEP 8 style and structural requirements were adhered to.

5.2.3 CSV Standards:

To ensure interoperability with a variety of data management and visualization applications, the CSV file standards were embraced, created by the Internet Engineering Task Force (IETF).

5.3 Testing Standards

5.3.1 User acceptability Testing Standards:

In order to make sure that the website satisfies the requirements and expectations of the target market, user acceptability testing (UAT) was conducted. This involved evaluating the website's usability, functionality, and performance as well as getting user input on the data visualization tools.

5.3.2 Standards for Quality Assurance:

To make sure the website was dependable, secure, and free of bugs or errors, accepted quality assurance (QA) processes were adhered to. This included following the best practices for software testing and bug tracking, using automated testing tools, and conducting code reviews.

In order to make sure that the target audience could easily navigate and use this data visualization website, these standards were selected. The website's success in efficiently visualizing data from uploaded CSV files was largely due to the continuous use of these criteria throughout the project.

The use of the csv standards presented several difficulties throughout the project, particularly in identifying the proper column headings and data types for specific datasets. However, through iterative design and planning, these difficulties were successfully overcome. The adopted standards were not altered or modified during the project.

Chapter 6:
CONCLUSION AND FUTURE SCOPE

6.1 Conclusion:

This HTML and CSS-based data visualization tool project is a robust and easy-to-use tool for visualizing data. It was created with Python and flask framework in the backend and HTML and CSS in the frontend. The application provides output in the form of various graphs, pie charts, heatmaps, and other visualization options by allowing users to input data in the form of CSV or Excel files. The tool also offers insightful data insights and patterns, enabling users to base their judgements on their analyses.

Overall, this initiative makes an important contribution to the field of data visualization by facilitating people's understanding of complex data sets and increasing accessibility. Anyone who needs to analyze data can use the tool because of its adaptability and configurable features, which let users design visualizations that best meet their needs.

After working with this project, an important conclusion was drawn that each organization needs to invest in a business analysis and visualization tool project in order to make data-driven decisions and maintain its competitiveness. By applying the latest tools and techniques in the field of business research and visualization, organizations can gain crucial insights about their operations, customers, and competitors.

Business analysts and visualization experts should work together on successful business analysis and visualization tool projects to ensure that the tool fulfills the specific requirements of the enterprise. The tool needs to be easy to use, versatile, and extendable in order to accommodate changing business requirements and data sources.

6.2 Future scopes:

Business analysis and visualization is a quickly growing field that is always developing new tools and methodologies. An important future scope is the use of natural language processing (NLP) technology, which may be used to examine customer preferences, complaints, and moods. Big data visualization is another upcoming field that will also be necessary for creative ways to visualize complex datasets. It will be feasible to examine enormous amounts of sales data and make automated judgements by integrating machine learning and artificial intelligence, identifying patterns and trends that otherwise might be missed.

Chapter 7:
REFERENCES

[1]https://builtin.com/data-science/data-visualization-tutorial
[2]https://www.kaggle.com/code/jshen212/retail-business-sales-visualizations
[3]https://youtu.be/GPVsHOlRBBI
[4]https://jinja.palletsprojects.com/en/3.1.x/
[5]https://flask.palletsprojects.com/en/2.3.x/
[6]https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7192728
[7]https://archive.nytimes.com/bits.blogs.nytimes.com/2014/06/13/tableau-software-helps-data-become-more-visual/
[8] https://www.javatpoint.com/python-pandas
[9]https://data-voyager.gitbook.io/voyager/

INDIVIDUAL CONTRIBUTION REPORT:

BUSINESS ANALYTIC AND VISUALIZATION WEBSITE
Sayantan Banerjee
2005195

Abstract: The Business Analysis and Visualization Website project provides a platform for businesses to analyze and gain insights from their data through interactive visualizations. The website's backend utilizes Python libraries such as NumPy, Matplotlib, and Seaborn for data processing, while the frontend interface is developed using React and Flask for data input and presentation. The website offers services such as data import, cleansing, and statistical analysis, along with customizable visualization tools. User verification and data sharing features make it a collaborative platform for businesses. This project provides a powerful tool for data-based decision-making with a user-friendly interface and a reliable backend.

Individual Contribution: In this project I contributed mostly during the ideation phase and preparing the workflow of events on how to proceed with the project and findings. I helped in using and finding datasets suitable for our model from Kaggle. I worked on python libraries such as NumPy and pandas to implement the coding features. I took a proactive role in finding and fixing errors in the code during the testing phase to make sure the website was functioning properly. Based on my findings, I gave the team input and suggestions that increased the project's overall quality.

Individual Contribution to the Report: Knowing how important presentation and proper documentation is necessary for the project to be successfully acknowledged, I worked on the total outline and structure of the report. I completely handled the Introduction and Standards adopted segments of the report. I studied the various concepts and industry accepted standards that could be implemented on the code for its better acceptance and understanding.

Sayantan Banerjee

INDIVIDUAL CONTRIBUTION REPORT:

BUSINESS ANALYTIC AND VISUALIZATION WEBSITE
Chandrali Shyam
2005720

Abstract: The Business Analysis and Visualization Website project provides a platform for businesses to analyze and gain insights from their data through interactive visualizations. The website's backend utilizes Python libraries such as NumPy, Matplotlib, and Seaborn for data processing, while the frontend interface is developed using React and Flask for data input and presentation. The website offers services such as data import, cleansing, and statistical analysis, along with customizable visualization tools. User verification and data sharing features make it a collaborative platform for businesses. This project provides a powerful tool for data-based decision-making with a user-friendly interface and a reliable backend.

Individual Contribution: In this project I studied and researched the various datasets and processed it using Python libraries like Pandas, NumPy and Matplotlib in detail to implement and design the various functions of the project. I overlooked in the creation of charts and the necessary criteria that would be important for generating the analytic report and also the graph functions. I also helped in integrating the backend part of the project.

Individual Contribution to the Report: Knowing how important presentation and proper documentation is necessary for the project to be successfully acknowledged, I worked on writing the Abstract, Implementation and the conclusion of the report along with the detailed understanding of the outputs along with attaching the various outcomes in the report. I also overlooked the completion of the report and the correctness of it and checked for any errors and problems if made by my teammates.

Chandrali

INDIVIDUAL CONTRIBUTION REPORT:

BUSINESS ANALYTIC AND VISUALIZATION WEBSITE
Shaunak Chandra
2005757

Abstract: The Business Analysis and Visualization Website project provides a platform for businesses to analyze and gain insights from their data through interactive visualizations. The website's backend utilizes Python libraries such as NumPy, Matplotlib, and Seaborn for data processing, while the frontend interface is developed using React and Flask for data input and presentation. The website offers services such as data import, cleansing, and statistical analysis, along with customizable visualization tools. User verification and data sharing features make it a collaborative platform for businesses. This project provides a powerful tool for data-based decision-making with a user-friendly interface and a reliable backend.

Individual Contribution: I managed the whole coding aspect of the project and helped in integrating the frontend and the python functions for visualizations. Using Python libraries like Flask, I worked on creating the backend part of the project. I created particular endpoints for API requests and directed each call to specific functions and returned the resulted data back to the user using basic routing system. I deepened my understanding of these potent Python libraries and the real-world uses for them to create beautiful and responsive applications.

Individual Contribution to the Report: Knowing how important presentation and proper documentation is necessary for the project to be successfully acknowledged, I worked on writing the Problem Statement and Requirement Specification segment and also helped in completing the Standard Adopted segment. I also overlooked the completion of the report and the correctness of it and checked for any errors and problems if made by my teammates.

Shaunak Chandra.

INDIVIDUAL CONTRIBUTION REPORT:

BUSINESS ANALYTIC AND VISUALIZATION WEBSITE
Tarangini Sinha
2005768

Abstract: The Business Analysis and Visualization Website project provides a platform for businesses to analyze and gain insights from their data through interactive visualizations. The website's backend utilizes Python libraries such as NumPy, Matplotlib, and Seaborn for data processing, while the frontend interface is developed using React and Flask for data input and presentation. The website offers services such as data import, cleansing, and statistical analysis, along with customizable visualization tools. User verification and data sharing features make it a collaborative platform for businesses. This project provides a powerful tool for data-based decision-making with a user-friendly interface and a reliable backend.

Individual Contribution: My particular contribution to this project was developing the frontend. I spent time developing a user-friendly interface that made data entry and presentation simple. I achieved this by using Jinja2 templating language to produce a dynamic website to ensure a seamless user experience. I also contributed to the development of the website's overall design and layout. Finally, I helped in integrating the frontend with the backend via API calls and appropriate data passing.

Individual Contribution to the Report: Knowing how important presentation and proper documentation is necessary for the project to be successfully acknowledged, I worked on writing the Literature Review segment along with assisting in the Project Analysis segment and the Future Scope segment. I also overlooked the completion of the report and the correctness of it and checked for any errors and problems if made by my teammates.

Tarangini Sinha

BUSINESS ANALYTICS AND VISUALIZATION WEBSITE

ORIGINALITY REPORT

11%	9%	4%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.coursehero.com Internet Source	4%
2	Submitted to National Institute of Technology Warangal Student Paper	3%
3	Submitted to University of Computer Studies Student Paper	1%
4	issuu.com Internet Source	<1%
5	repository.uwl.ac.uk Internet Source	<1%
6	Submitted to Liverpool John Moores University Student Paper	<1%
7	Submitted to University of Greenwich Student Paper	<1%
8	dailygram.com Internet Source	<1%

9	"Advances in Systems, Control and Automations", Springer Science and Business Media LLC, 2021 Publication	<1 %
10	dspace.daffodilvarsity.edu.bd:8080 Internet Source	<1 %
11	"Advances in Electronics, Communication and Computing", Springer Science and Business Media LLC, 2021 Publication	<1 %
12	Chenglong Wang, Yu Feng, Rastislav Bodik, Isil Dillig, Alvin Cheung, Amy J Ko. "Falx: Synthesis-Powered Visualization Authoring", Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, 2021 Publication	<1 %
13	Submitted to Liverpool Hope Student Paper	<1 %
14	www.icicel.org Internet Source	<1 %
15	"Advances in Power Systems and Energy Management", Springer Science and Business Media LLC, 2021 Publication	<1 %
16	"Machine Learning and Knowledge Discovery in Databases: Applied Data Science Track",	<1 %

Springer Science and Business Media LLC, 2021
Publication

17	medium.com Internet Source	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		