

A PROJECT REPORT
on
“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

NAME	ROLL NUMBER
SAYANTAN BANERJEE	2005195
CHANDRALI SHYAM	2005720
SHAUNAK CHANDRA	2005757
TARANGINI SINHA	2005768

UNDER THE GUIDANCE OF
PROF. PRATYUSA MUKHERJEE MAM



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
“BUSINESS ANALYTICS AND VISUALIZATION WEBSITE”
Submitted by

NAME	ROLL NUMBER
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: / /

Prof. Pratyusa Mukherjee
Project Guide

ACKNOWLEDGEMENT

We would like to extend our sincere gratitude to our **Professor Pratyusa Mukherjee** mam from the **Department of Computer Science and 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.

Our 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.

Keywords: Sales analysis, Data visualization ^[5], Flask ^[4], Pandas ^[2], Matplotlib ^[5], Web development, Predictive analytics ^[2], Descriptive analytics, Data Preprocessing ^[2], Feature engineering

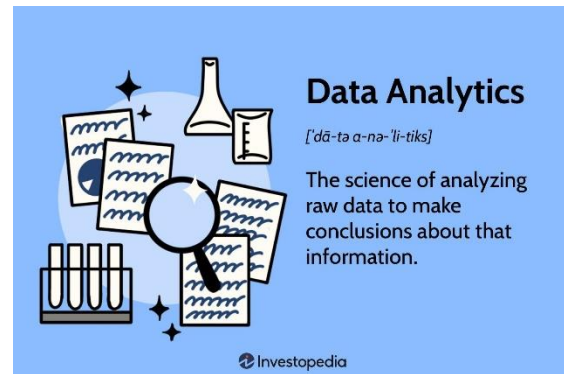
-

CONTENTS

SL No.	Title	Page No.
1	Introduction	5
2	Basic Concepts/ Literature Review	6-7
	2.1 Data Analytics	6
	2.2 Data Visualization	6
	2.3 Web Development	6
	2.4 Python	7
	2.5 Matplotlib and Pandas	7
3	Problem Statement / Requirement Specifications	8-11
	3.1 Project Planning	8
	3.2 Project Analysis	9
	3.3 System Design	11
	3.3.1 Design Constraints	11
	3.3.2 System Architecture/Block Diagram	11
4	Implementation	12-17
	4.1 Introduction	12
	4.2 Data Collection and Exploration	12
	4.3 Data Preprocessing	13
	4.4 Data Visualization	13
	4.5 Coding	15
	4.6 Integration	16
	4.7 Conclusion	17
5	Standard Adopted	18
	5.1 Design Standards	18
	5.2 Coding Standards	18
	5.3 Testing Standards	18
6	Conclusion and Future Scope	20
	6.1 Conclusion	20
	6.2 Future Scope	20
	References	22-21
	Individual Contribution	23-26
	Plagiarism Report	27

1.INTRODUCTION

Data visualization is an essential part of **data analytics** 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 has increased.



In today's competitive business environment, having access to real-time data and analytics 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, our team aimed to develop a tool to accelerate visual representation and understanding of data of any given business when their data is provided in a formatted CSV file. Our website inclines towards users to upload a CSV^[1] file in the specified format and represent their sales, profits, loss and highest grossing products through visual representation and a short summary.

We have tried to integrate the concepts of Data Analysis^[2] 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 along with detailed reports.

2. Basic Concepts/Literature Review

2.1 Data Analytics

Data analytics ^[2] 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 are becoming increasingly important as a tool for gaining insights and promoting company performance as a result of the continued growth in the amount of data collected by firms.

2.2 Data Visualization

Data visualization ^[5] 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 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.

2.3 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.

2.4 Python

Python 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, 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 because of its widespread use and simplicity.

2.5 Matplotlib and Pandas

A common Python data visualization library called Matplotlib ^[5] is used to produce various kinds of graphs, charts, and plots. It supports numerous data formats, including CSV ^[1], Excel, and SQL databases, and offers a broad range of customization options. Users can produce excellent data visualizations with Matplotlib that are simple to understand and communicate insights from data.

Python's powerful data analysis library, Pandas, 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. Pandas allows users to easily filter, sort, and summarise 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.

3.PROBLEM STATEMENT/ REQUIREMENT SPECIFICATION

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^[5] 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.^[5] Markup language 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^[5] 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 ^[1] 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 ^{[2][5]}.
- 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).^{[2][5]}

(4) Data Preprocessing

- The application should preprocess the data by handling missing data, imputing outliers, and normalizing the data before generating visualizations.^{[2][5]}

(5) Exporting

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

Non-Functional Requirements: -

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:

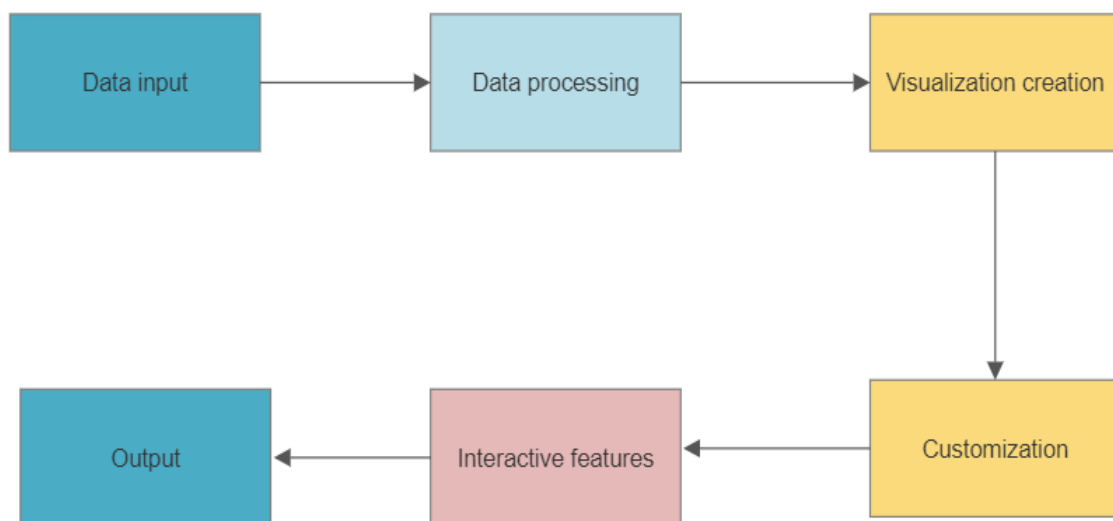
(1) Technology Stack

- The application should be developed using Python and its libraries ^{[2][5]} 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



4.IMPLEMENTATION

4.1 Introduction

Let us now discuss the project's overall flow. Typically, both the front-end and back-end aspects are to be developed first. JavaScript and Flask^[4] have been used for the front-end part. The back end is completely made by python^[2] using in-built libraries like NumPy, pandas, seaborn and matplotlib^[5] and testing data has been sourced from Kaggle^[1]. The back end is responsible for processing of the data and creation of the visualizations, which are subsequently saved in a static folder and later shown on the display page to the user. Before this happens, the data is to be collected and cleansed first.

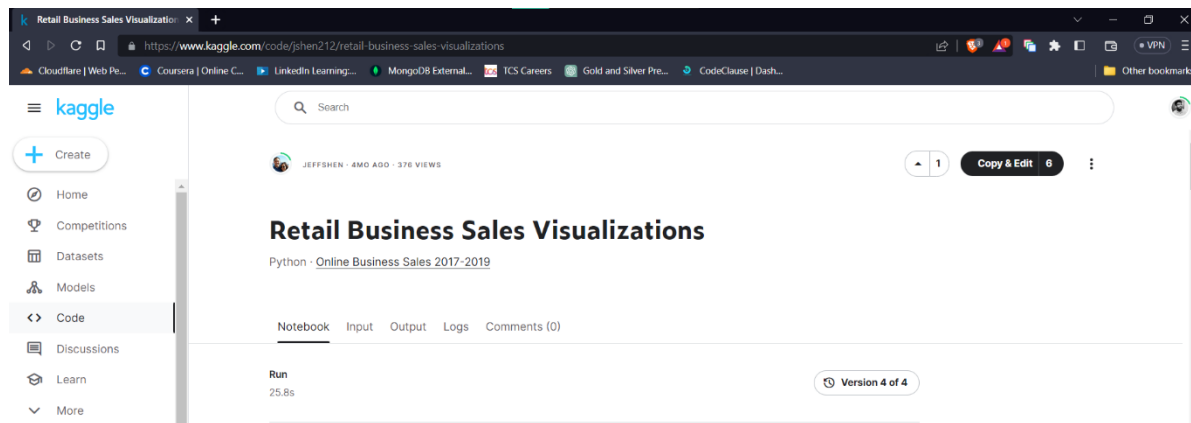
Although the normalization technique is frequently used to take into account data variations, it has not been required in this case due to the small amount of variation present. Finally, we come to the end result which is a cohesive and informative display of data that has been cleaned, processed and visualized by the back-end functions, and displayed to the user through the front-end interface.

4.2 Data Collection and Exploration

The well-known website Kaggle^[1] 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^[1]. 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.



4.3 Data Preprocessing ^[2]

Any type of processing done on raw data ^[1] 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. We also analyzed the data and saw relations between each data column and drew inferences according to the need. ^[1]

```
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
```

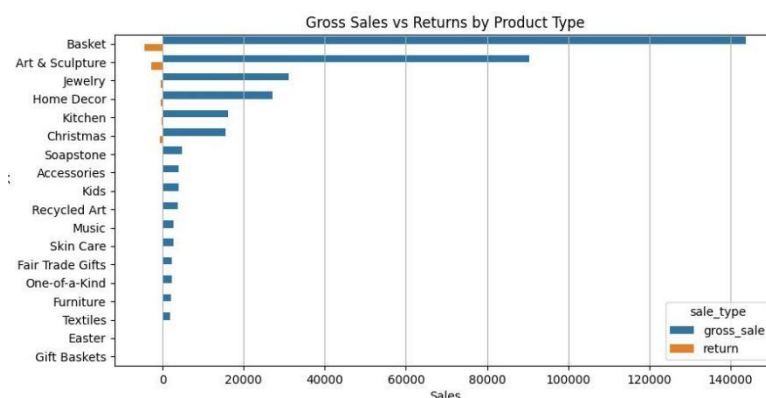
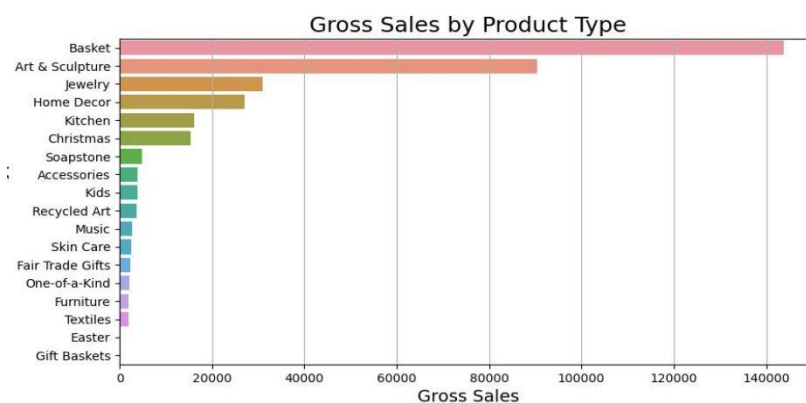
4.4 Data Visualization ^{[2][5]}

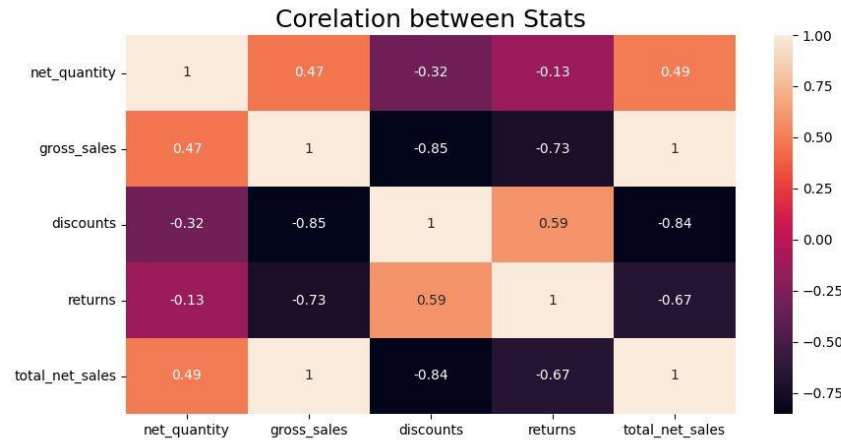
The provided information shows the net quantity, gross sales, discounts, returns, and total net sales for a range of goods, including baskets, Christmas decorations, artwork, and sculpture. Several data visualization techniques, including bar charts, scatter plots, stacked bar charts, pie charts, heatmaps,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

Python ^[2] 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 ^[5].

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.





4.5 Coding

For the backend, the Flask^[4] framework in Python^[2] was used to handle incoming requests and call appropriate Python functions to process the data and generate charts. We have also used Pandas and Matplotlib^{[2][5]} libraries for data processing and visualization respectively. The generated charts are stored in the static folder for later retrieval.

For the frontend, you have used HTML, CSS, and JavaScript with the Bootstrap framework for responsive design. You have also used Jinja2^[3] 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.

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8" />
5 <meta
6   name="viewport"
7   content="width=device-width, initial-scale=1
8 </meta>
9 <title>Form with side bar</title>
10 <link rel="stylesheet" href="{
11   url_for('static', filename='/css/style.css') }}">
12 <link rel="preconnect" href="https://fonts.google
13 <link rel="preconnect" href="https://fonts.gstatic
14   href="https://fonts.googleapis.com/css2?family=
15   rel="stylesheet" />
16 </head>
17 <body>
18 <main>
19 <section id="upload-section">
20 <div>
21 <h1>Business Analyzer</h1>
22 </div>
23 <div>
24 <label for="form">Upload file in .
25 <form
26   action="/"
27   id="form"
28   method="post"
29   enctype="multipart/form-data"
30 <label for="file">Upload Your
31 <input type="file" name="file"
32 <button type="submit">Submit<

```

```

1 from flask import Flask, request, render_template
2 import pandas as pd
3 from gross_sales import gross_sales
4 from sales_vs_returns import sales_vs_returns
5 from correlation import correlation
6 import matplotlib
7 matplotlib.use('Agg')
8
9 app = Flask(__name__)
10 app.config['STATIC_FOLDER'] = 'static'
11
12 @app.route('/', methods=['GET', 'POST'])
13 def home():
14     result = []
15     if request.method == 'GET':
16         return render_template("home.html")
17     else:
18         file = request.files["file"]
19         #reading the file and renaming the columns for effi
20         sales = pd.read_csv(file)
21         sales=sales.dropna()
22         sales.columns = ['product_type', 'net_quantity', 'gr
23
24         result.append(gross_sales(app, sales))
25         result.append(sales_vs_returns(app, sales))
26         result.append(correlation(app,sales))
27         # sales_vs_returns(app,sales)
28         # correlation(app,sales)
29
30     # result = [
31     # {
32     #     'title': 'Which product types have the hi
33     #     'gross_sales': gross_sales,
34     #     'sales_vs_returns': sales_vs_returns,
35     #     'correlation': correlation
36     # }
37     # ]
38     return render_template("home.html", result=result)

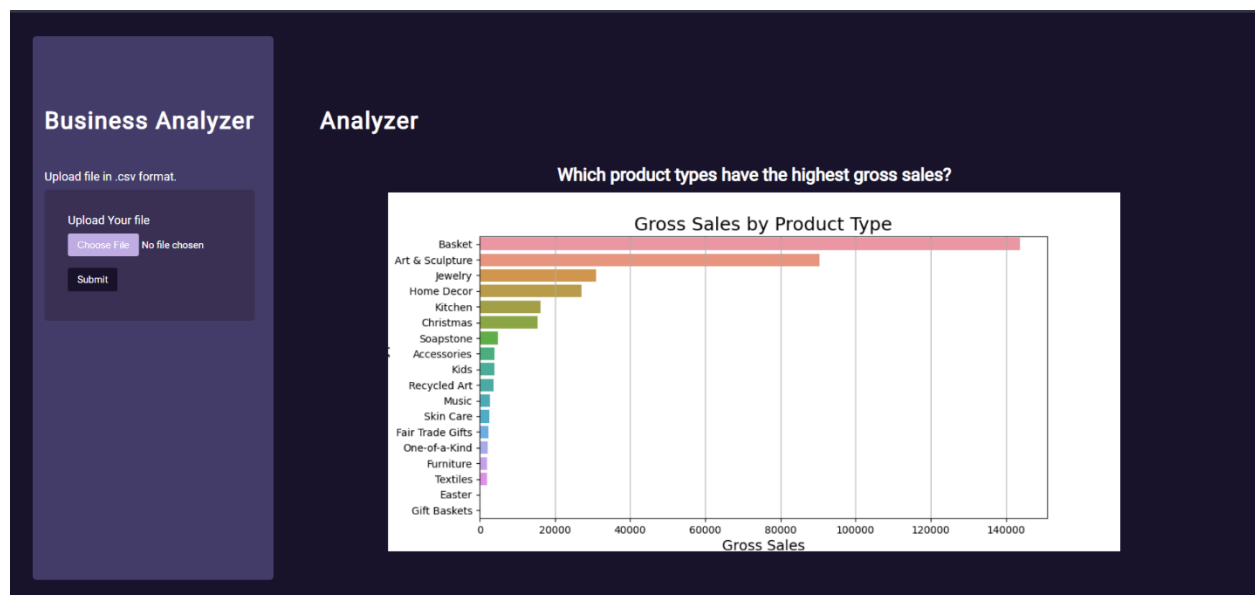
```


4.7 Integration

To ensure that our application runs smoothly, we have given utmost importance 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 ^{[2][5]} depending on the input data.

We have used the Jinja ^[3] templating engine and the Flask ^[4] web framework 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.

We create Flask routes 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. We can develop a seamless and effective user experience that enables users to meaningfully interact with their data by linking the frontend and backend in this way.



4.5 Conclusion

After completing this project, we have come to the conclusion 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.

5.STANDARDS ADOPTED

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

5.1 Design Standards

5.1.1 Data Visualization ^{[2][5]} Standards:

To effectively communicate insights and trends in the data, we used the best practices 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:

We implemented "JavaScript Coding Standards" for our 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, we made sure that our JavaScript code complied with best practices and was simple for other programmers to understand.

5.2.2 Python Standards:

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

5.2.3 CSV ^[1] Standards:

To ensure interoperability with a variety of data management and visualization applications, we embraced the CSV file standards 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 our target market, we conducted user acceptability testing

(UAT). 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, we adhered to accepted quality assurance (QA) processes. 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 our target audience could easily navigate and use our data visualization website, we selected these standards. 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 for us, particularly in identifying the proper column headings and data types for specific datasets. However, through iterative design and planning, we were able to overcome these difficulties. The adopted standards were not altered or modified during the project.

6.CONCLUSION AND FUTURE SCOPE

6.1 Conclusion:

After this project, we come to the conclusion 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 scope:

Every day, new tools and methodologies are being introduced in the ever-growing discipline of Business analysis and visualization.

Here are a few future scopes of this discipline:

(1) Natural language processing:

NLP is a technology that is developing quickly and has the potential to change how people interact with websites. Business analysis and visualization tools might be useful in integrating the same to websites. For example, it can be applied to analyze emails, social media posts, and reviews from customers to determine their preferences, grievances, and sentiments. This data can be used to enhance customer satisfaction and improve goods and services.

(2) Big Data visualization:

The amount of data produced grows every day and accordingly, the requirement of visualization increases. Business analysis and visualization tools may be helpful to create innovative ways for visualizing massive datasets consisting of complex data.

(3) Integration of machine learning with artificial intelligence:

By offering predictive analytics and automated decision-making capabilities, machine learning (ML) and artificial intelligence (AI) can support your project for sales analysis. By utilizing ML and AI techniques, you can create algorithms that can automatically analyze large amounts of sales data and spot patterns and trends that human analysts might not notice right away.

7.REFERENCES

- [1] <https://www.kaggle.com/code/jshen212/retail-business-sales-visualizations>
- [2] <https://youtu.be/GPVsHOIRBBI>
- [3] <https://jinja.palletsprojects.com/en/3.1.x/>
- [4] <https://flask.palletsprojects.com/en/2.3.x/>
- [5] <https://builtin.com/data-science/data-visualization-tutorial>

INDIVIDUAL CONTRIBUTION REPORT:

BUSINESS ANALYTIC AND VISUALIZATION WEBSITE

Sayantana 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.

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.

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.

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.