Tech Saksham

Capstone Project Report

“E-Commerce Analysis”

“University College Of Engineering – Arni”

NM ID	NAME
aut513321105702	SUBHASHINI E G

Ramar Bose

Sr. AI Master Trainer

ABSTRACT

The E-Commerce Analysis project offers a comprehensive approach to analyzing retail sales data and providing actionable insights for optimized operations. By leveraging data from sales records, the project explores monthly and hourly sales trends, city-specific sales performance, popular product demand, and common product combinations. These insights guide businesses in tailoring their sales strategies, managing inventory efficiently, and targeting marketing efforts more effectively. Additionally, the project calculates the probability of selling specific products annually and monthly, enabling businesses to optimize stock levels and promotions. The visualizations and reporting generated from the analysis support strategic decision-making across various business areas, leading to increased efficiency, profitability, and customer satisfaction. The project establishes a foundation for ongoing performance monitoring and continuous improvement, positioning businesses for success in a competitive market. With the potential to integrate additional data sources and advanced analytics in the future, the project provides a scalable solution for retail businesses aiming to enhance their operations and achieve long-term growth.

1. Problem statement
2. Data collection
3. Existing solution
4. Proposed solution with used models
5. Result

INDEX

Sr. No.	Table of Contents	Page No.
1	Chapter 1: Introduction	1
2	Chapter 2: Services and Tools Required	3
3	Chapter 3: Project Architecture	6
4	Chapter 4: Modeling and Project Outcome	12
5	Conclusion	22
6	Future Scope	23
7	References	24
8	Links	25

CHAPTER 1

INTRODUCTION

1.1 Problem Statement

- Conduct an exploratory data analysis to gain insights into sales performance, sales trends
- Customer purchasing patterns across different cities, products, and time periods.

1.2 Proposed Solution

- Load the CSV files from the specified folder and combine them into a single DataFrame.
- Clean the data by excluding rows with headers and missing values.
- Extract features such as year, month, hour, and city from the data.
- Analyze the data with histograms, boxplots, and bar charts.
- Examine sales patterns across different time periods and cities.
- Calculate monthly and yearly probabilities for specific products.

1.3 Feature

Data Preprocessing and Cleaning:

- Loading, combining, and filtering CSV files to obtain a comprehensive dataset.
- Removing null values and excluding header rows to clean the data.

Summary Statistics:

- Calculating total orders, products sold, and total sales for the year 2019.
- Providing an overview of sales performance.

Visualizations:

- Plotting monthly sales trends to identify seasonal variations.
- Visualizing hourly sales trends to identify peak sales hours.

1.4 Advantages

The project offers several advantages for businesses and data analysts working with e-commerce sales data:

- **Comprehensive Insights:**Provides a detailed analysis of sales data, including total orders, products sold, and total sales, which can inform strategic decisions and performance evaluation.
- **Data-Driven Decision-Making:**Offers data visualizations and statistical analyses that enable businesses to make informed decisions based on real data, such as optimizing inventory management and adjusting sales strategies.
- **Identification of Trends and Patterns:**Visualizations such as monthly and hourly sales charts reveal trends and patterns in sales, aiding in planning and forecasting.
- **Location-Based Analysis:**City-specific sales data helps identify high-performing locations, which can guide targeted marketing efforts and regional sales strategies.

1.5 Scope

The scope of the project can be defined in terms of the extent of data analysis, potential use cases, and avenues for further exploration and application. The project covers the following areas:

Sales Trends Analysis:

- The project analyzes monthly and hourly sales trends to identify peak sales periods and adjust strategies accordingly.
- This analysis helps in predicting future sales and optimizing resource allocation.

City-Specific Performance:

- The project examines city-specific sales performance to tailor marketing and inventory strategies to the unique needs of each location.
- It includes visualizations and analyses of sales across different cities.

Product Combinations Analysis:

- The project explores common product combinations purchased together.
- This analysis can inform bundling strategies and cross-selling opportunities.

Yearly and Monthly Probabilities:

- The project calculates the probability of selling specific products annually and monthly.
- These probabilities help in optimizing inventory management and sales promotions.

CHAPTER 2

SERVICES AND TOOLS REQUIRED

2.1 LR - Exiting Models

- **Standard Logistic Regression:**The basic form of logistic regression, available in many machine learning libraries (e.g., scikit-learn, TensorFlow, PyTorch).Suitable for binary classification tasks.
- **Regularized Logistic Regression:**Variants of logistic regression that include regularization techniques such as L1 (Lasso) and L2 (Ridge) regularization.Helps to prevent overfitting by penalizing large coefficients in the model.
- **Multinomial Logistic Regression:**Extends logistic regression to handle multiple classes (multiclass classification).Predicts the probability of each class and can be implemented using one-vs-rest or softmax approaches.
- **Stochastic Gradient Descent (SGD) Logistic Regression:**Uses stochastic gradient descent as the optimization technique for training the logistic regression model.Useful for handling large datasets and can be found in libraries such as scikit-learn.

2.1 Required – System config | Cloud computing

- **Compute Resources:**Virtual Machines: Choose virtual machines with suitable CPU and memory configurations to handle data processing and analysis tasks.
- **Managed Services:** Consider using managed services for data processing (e.g., AWS Glue, Google Dataflow) to simplify infrastructure management.

Storage:

- **Cloud Storage**: Use cloud storage services such as Amazon S3, Google Cloud Storage, or Azure Blob Storage for storing datasets.

- **Data Warehousing:** Consider using a managed data warehouse service (e.g., Amazon Redshift, Google BigQuery, or Azure Synapse Analytics) for efficient data querying and analysis.

Data Processing:

- **Batch Processing**: If the dataset is large, you might consider using distributed data processing frameworks such as Apache Spark on cloud platforms (e.g., AWS EMR, Google Dataproc).
- **Serverless Functions:** Consider using serverless functions (e.g., AWS Lambda, Google Cloud Functions, Azure Functions) for event-driven data processing and analysis tasks.

2.1 Services Used

Data Processing:

- **Data Factory:** Managed data integration service for data transformation and preparation.
- **HDInsight:** Managed service for running big data processing tasks with Apache Spark and other frameworks.

Machine Learning:

- **Azure Machine Learning:** Managed service for building and deploying models.

Analytics:

- **Azure Synapse Analytics:** Data warehousing and analytics service for efficient querying and analysis.

Monitoring and Logging:

- **Azure Monitor:** Monitoring service for tracking resource usage and application performance.

  Tools and software used:
- **Python:**A versatile programming language commonly used for data analysis and machine learning**.**

  Libraries and packages:

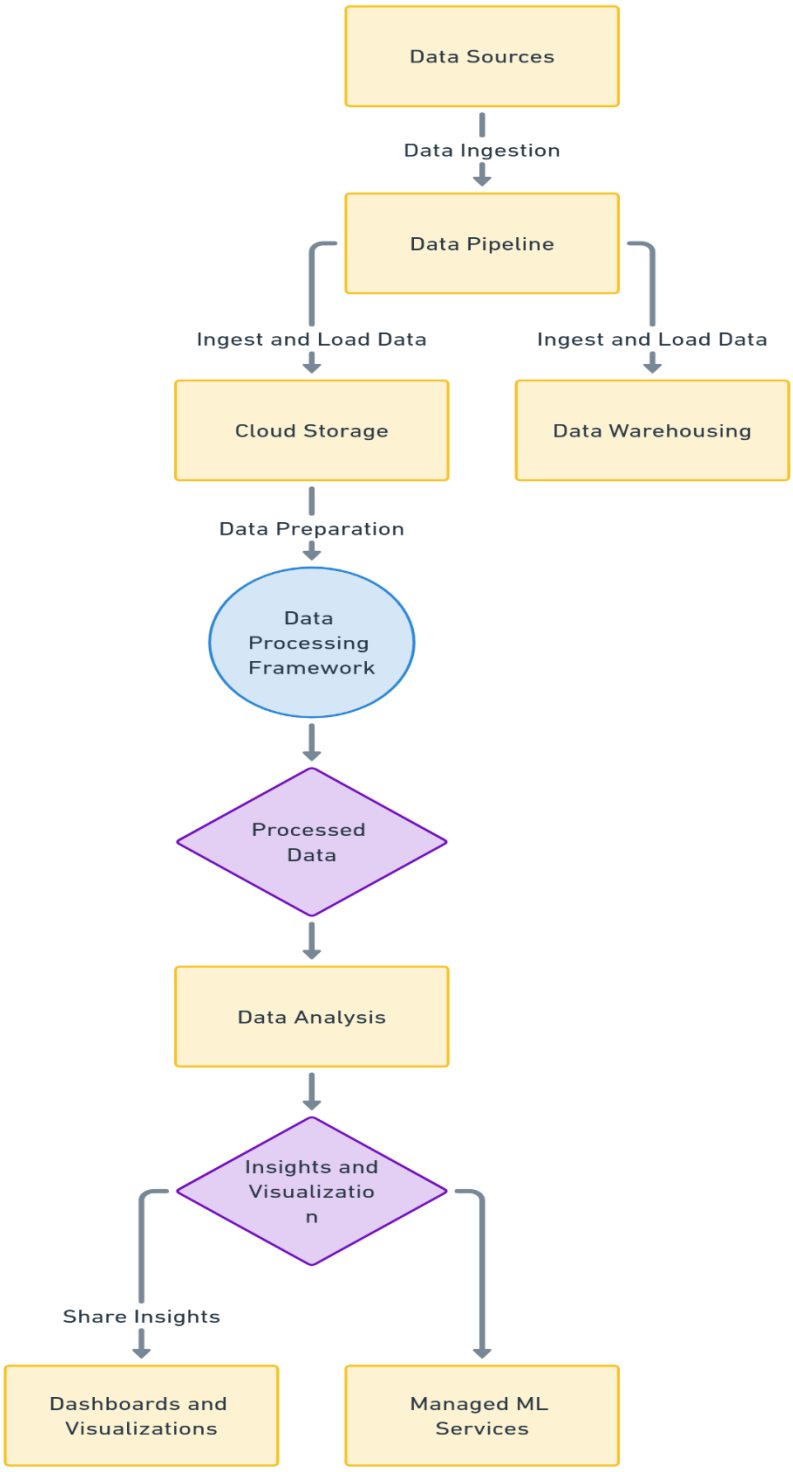
- **Pandas:** For data manipulation, cleaning, and analysis.
- **NumPy**: For numerical computations and array handling.
- **Jupyter Notebooks**:An interactive computing environment for writing and executing code, visualizing data, and documenting the analysis.
- **R:**An alternative language for statistical analysis and data manipulation.
- Popular libraries include dplyr, tidyr, and ggplot2.

CHAPTER 3

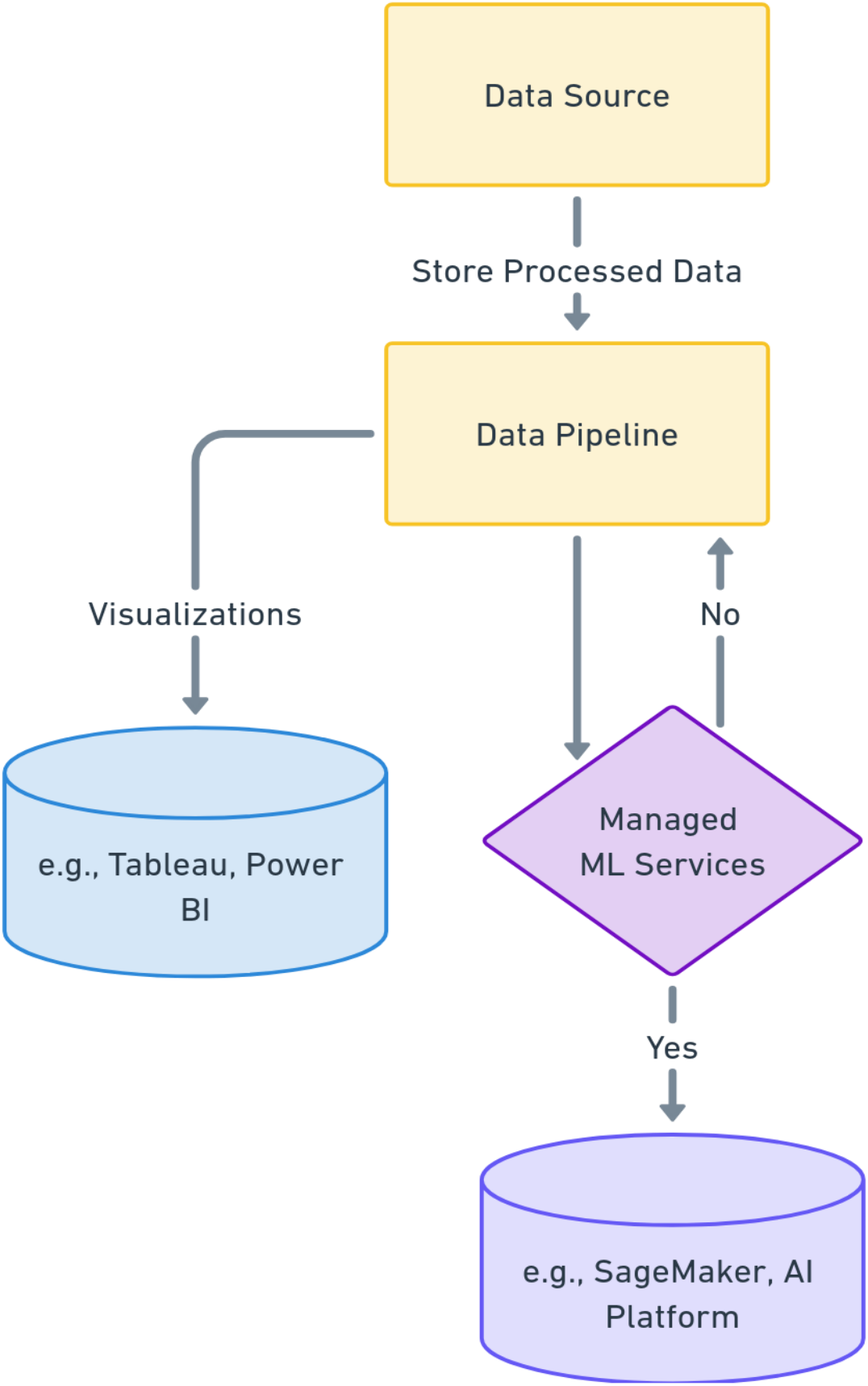
PROJECT ARCHITECTURE

3.1 Architecture

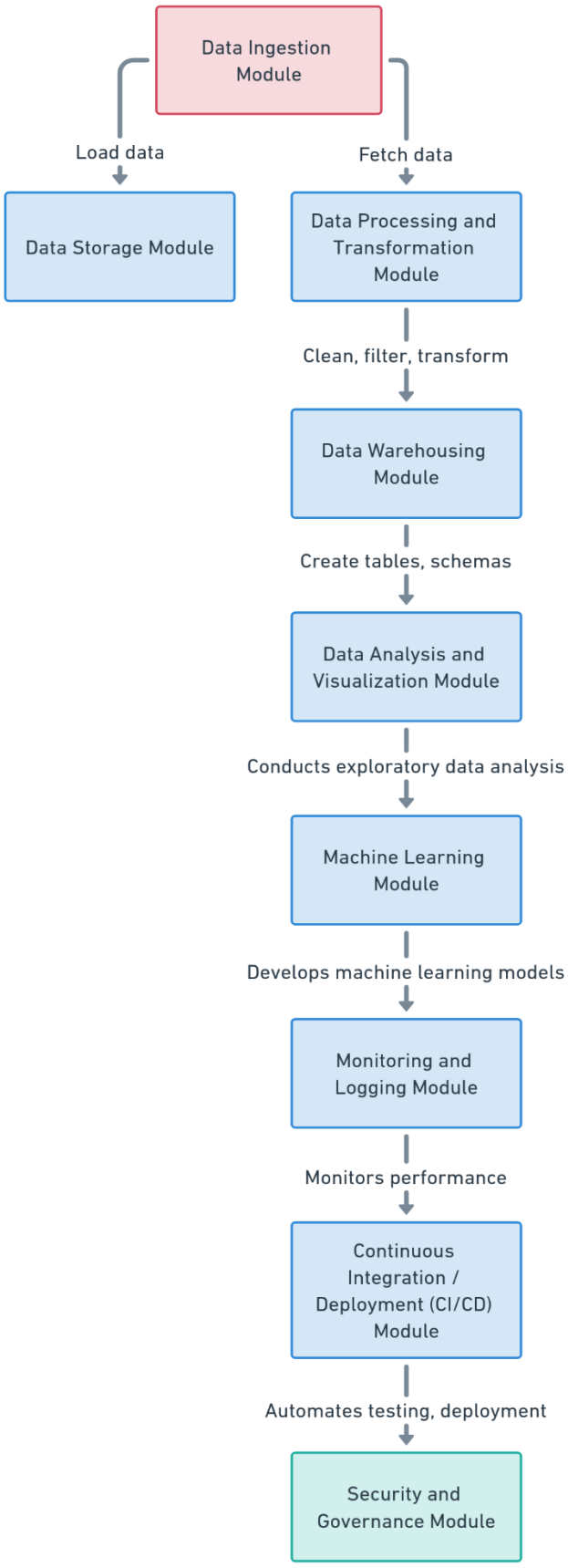
1. System flow diagram



2. Data flow diagram

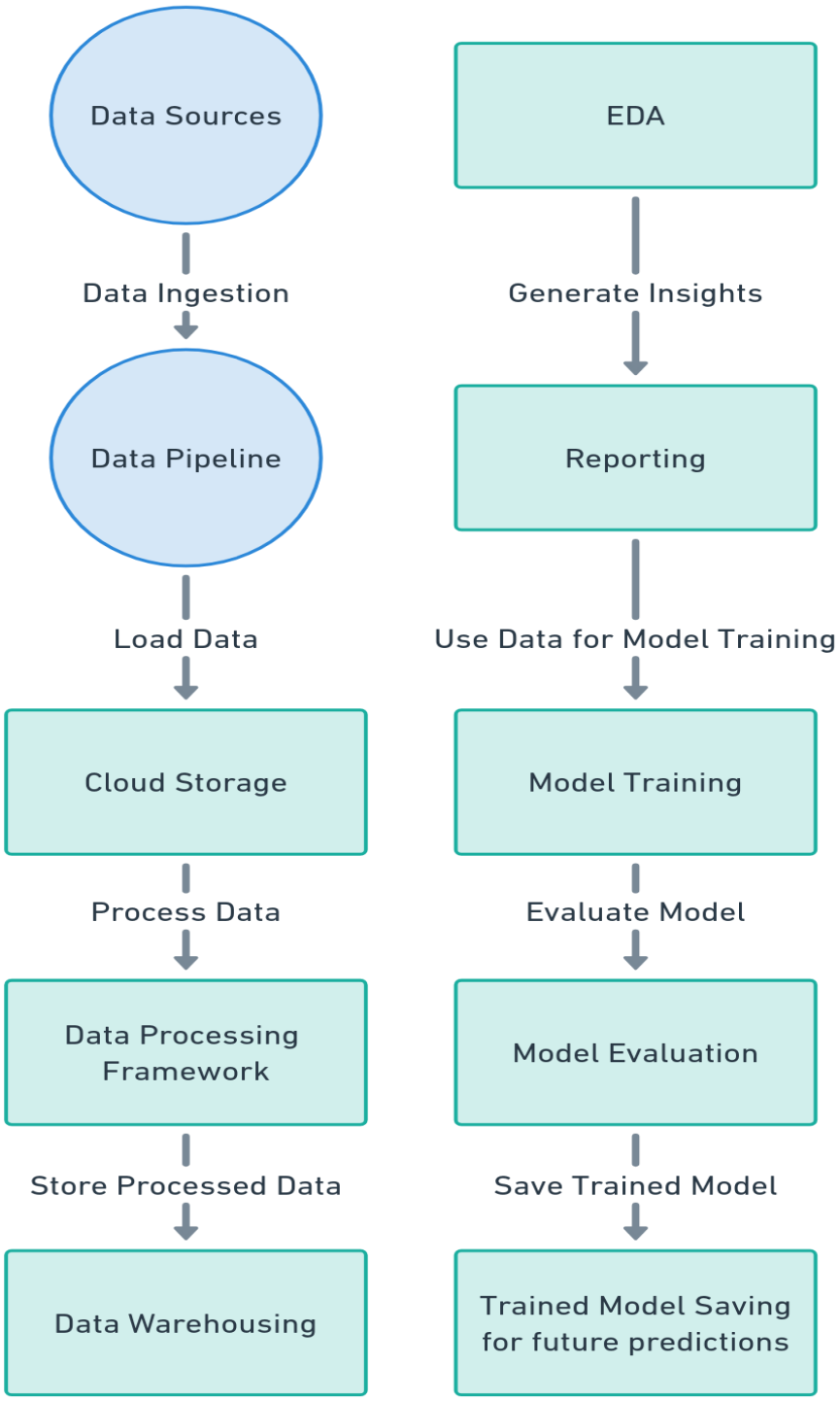


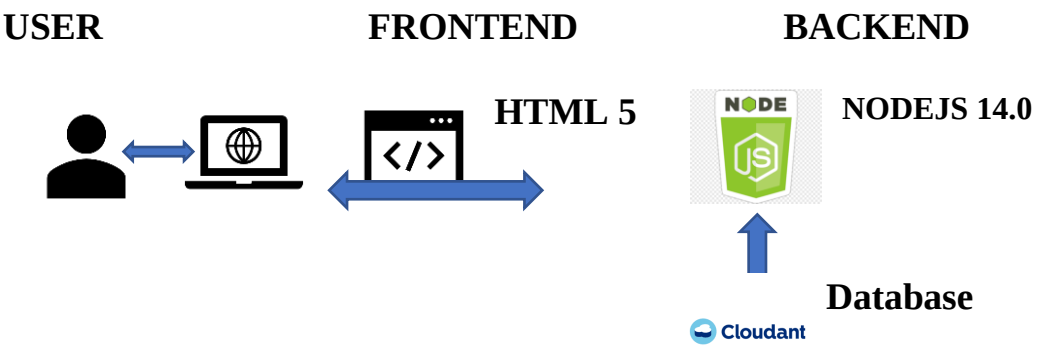
3.Module



4. User interface

- Next Module (EDA) flow diagram
- Training model diagram
- Predicting model's diagram
- Model Performance evaluation models





Here's a high-level architecture for the project:

Data Ingestion and Storage:

- Data is collected from various sources such as online store transactions and customer information.
- The data is ingested and stored in cloud storage solutions like Amazon S3, Google Cloud Storage, or Azure Blob Storage.

Data Processing and Transformation:

- Distributed processing frameworks such as Apache Spark or cloud-native solutions are used to clean, preprocess, and transform the data.
- The processed data is then loaded into a data warehouse such as Amazon Redshift, Google BigQuery, or Azure Synapse Analytics for analysis.

Exploratory Data Analysis (EDA) and Visualization:

- Analysts use tools such as Python, R, or business intelligence platforms to explore and visualize data.
- Visualizations like charts and dashboards communicate findings and insights.

Model Development and Deployment:

- Machine learning models are developed, trained, and validated using the processed data.
- Trained models are deployed as web services or APIs for real-time predictions.

Monitoring and Reporting:

- Models in production are monitored and maintained to ensure consistent performance.
- Insights and reports are shared with stakeholders for strategic decision-making and planning.

CHAPTER 4

MODELING AND PROJECT OUTCOME

Data load:

Code:

```
folder_path = '/content/sample_data/dataset'  # Replace
'your_folder_name' with your folder's name
files = [os.path.join(folder_path, f) for f in os.listdir(folder_path)
if f.endswith('.csv')]
for file_path in files:
    df = pd.read_csv(file_path)
df.head()
```

Ouput:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
0	236670	Wired Headphones	2	11.99	08/31/19 22:21	359 Spruce St, Seattle, WA 98101
1	236671	Bose SoundSport Headphones	1	99.99	08/15/19 15:11	492 Ridge St, Dallas, TX 75001
2	236672	iPhone	1	700.0	08/06/19 14:40	149 7th St, Portland, OR 97035
3	236673	AA Batteries (4-pack)	2	3.84	08/29/19 20:59	631 2nd St, Los Angeles, CA 90001
4	236674	AA Batteries (4-pack)	2	3.84	08/15/19 19:53	736 14th St, New York City, NY 10001

EDA – analysis report:

1. Missing Code:

```
    # Check for data types

    df.info()
    # Check for null values

    df.isna().sum()
    # Check rows with null values

    df[df.isna().any(axis=1)]
```


Output:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
310	NaN	NaN	NaN	NaN	NaN	NaN
1220	NaN	NaN	NaN	NaN	NaN	NaN
2639	NaN	NaN	NaN	NaN	NaN	NaN
2675	NaN	NaN	NaN	NaN	NaN	NaN
3109	NaN	NaN	NaN	NaN	NaN	NaN
3300	NaN	NaN	NaN	NaN	NaN	NaN
4277	NaN	NaN	NaN	NaN	NaN	NaN
4293	NaN	NaN	NaN	NaN	NaN	NaN
4443	NaN	NaN	NaN	NaN	NaN	NaN
4667	NaN	NaN	NaN	NaN	NaN	NaN
5508	NaN	NaN	NaN	NaN	NaN	NaN
5649	NaN	NaN	NaN	NaN	NaN	NaN
6063	NaN	NaN	NaN	NaN	NaN	NaN
6428	NaN	NaN	NaN	NaN	NaN	NaN
6588	NaN	NaN	NaN	NaN	NaN	NaN
6779	NaN	NaN	NaN	NaN	NaN	NaN
7492	NaN	NaN	NaN	NaN	NaN	NaN
7928	NaN	NaN	NaN	NaN	NaN	NaN
7935	NaN	NaN	NaN	NaN	NaN	NaN
8665	NaN	NaN	NaN	NaN	NaN	NaN
8800	NaN	NaN	NaN	NaN	NaN	NaN
8910	NaN	NaN	NaN	NaN	NaN	NaN
9400	NaN	NaN	NaN	NaN	NaN	NaN

2. Data Visualizations

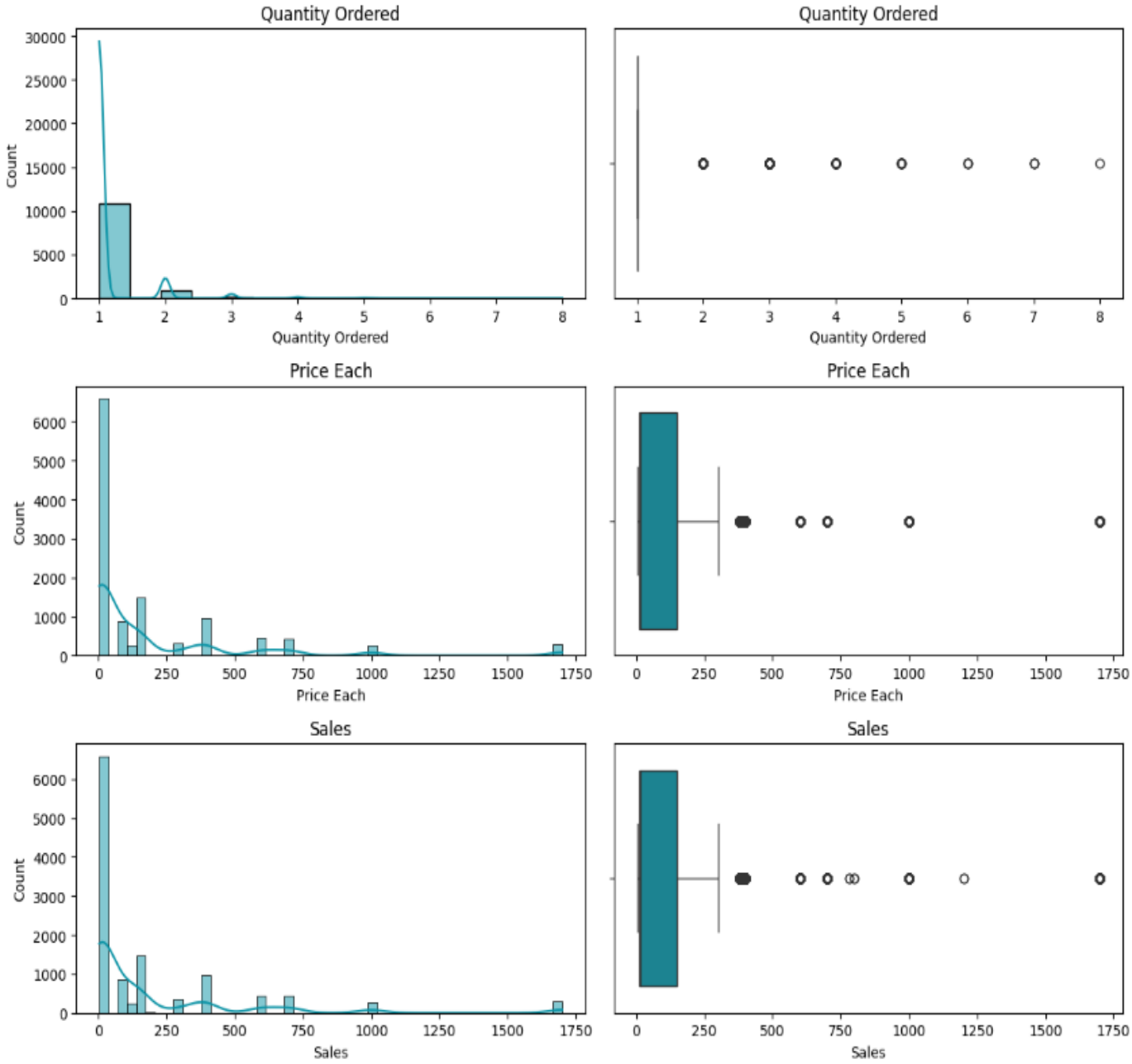
Code:

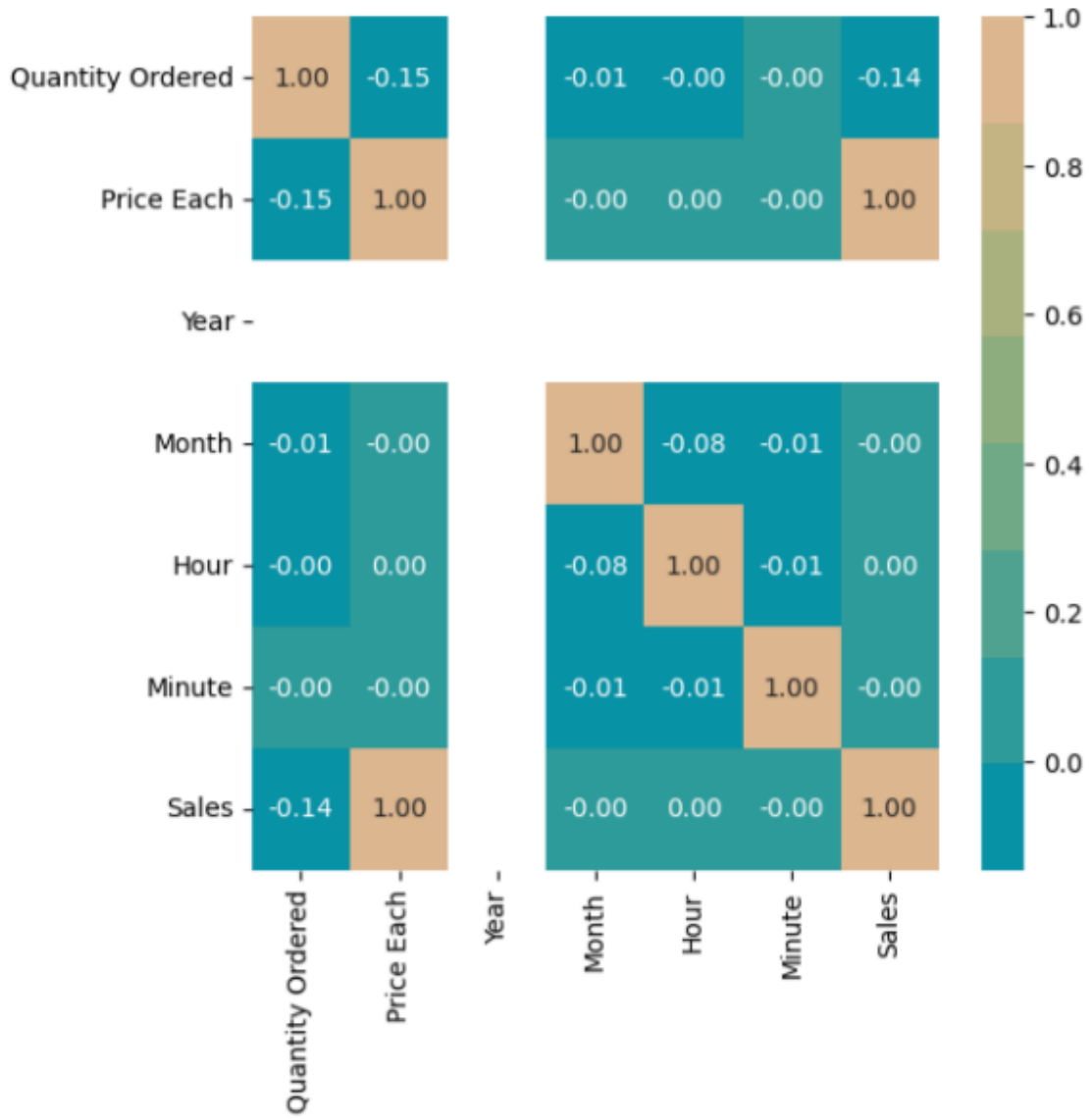
```
fig, ax = plt.subplots(3, 2, figsize=(12, 10))
sns.histplot(data=df, x='Quantity Ordered', kde=True, ax=ax[0, 0])
sns.histplot(data=df, x='Price Each', kde=True, ax=ax[1, 0], bins=50)
sns.histplot(data=df, x='Sales', kde=True, ax=ax[2, 0], bins=50)

ax[0, 0].set_title('Quantity Ordered')
ax[1, 0].set_title('Price Each')
ax[2, 0].set_title('Sales')

sns.boxplot(data=df, x='Quantity Ordered', ax=ax[0, 1])
sns.boxplot(data=df, x='Price Each', ax=ax[1, 1])
sns.boxplot(data=df, x='Sales', ax=ax[2, 1])

ax[0, 1].set_title('Quantity Ordered')
ax[1, 1].set_title('Price Each')
ax[2, 1].set_title('Sales')

plt.tight_layout()
plt.show()

plt.figure(figsize=(10, 6))
df['Cities'].value_counts().plot(kind='bar',
color=['#0892a5','#2e9b9b','#50a290','#6fa985','#8dad7f','#a9b17e','#c4b
383','#dbb68f'])
# sns.countplot(df['Cities'])
plt.title('Cities orders distribution', weight='bold', fontsize=20,
pad=20)
plt.ylabel('Orders', fontsize=12)
plt.xlabel('City', fontsize=12)
plt.show()

plt.figure(figsize=(10, 6))
df['Month'].value_counts().sort_index().plot(kind='bar',
color=['#0892a5','#2e9b9b','#50a290','#6fa985','#8dad7f','#a9b17e','#c4b
383','#dbb68f'])
plt.title('Month orders distribution', weight='bold', fontsize=20,
pad=20)
plt.ylabel('Orders', fontsize=12)
plt.xlabel('Month', fontsize=12)
plt.show()
```



```
plt.figure(figsize=(6, 6))
sns.heatmap(numeric_df.corr(), annot=True, fmt='.2f', cmap=['#0892a5',
'#2e9b9b', '#50a290', '#6fa985', '#8dad7f', '#a9b17e', '#c4b383',
'#dbb68f'])
plt.show()
```

Output:





Model output:

Code:

```
def proba_prod(product, df):
    """
    Calculate the monthly and yearly probabilities for a given product.

    Arguments:
    product -- The name of the product to calculate probabilities for.
    df -- The DataFrame containing the data.

    Returns:
```

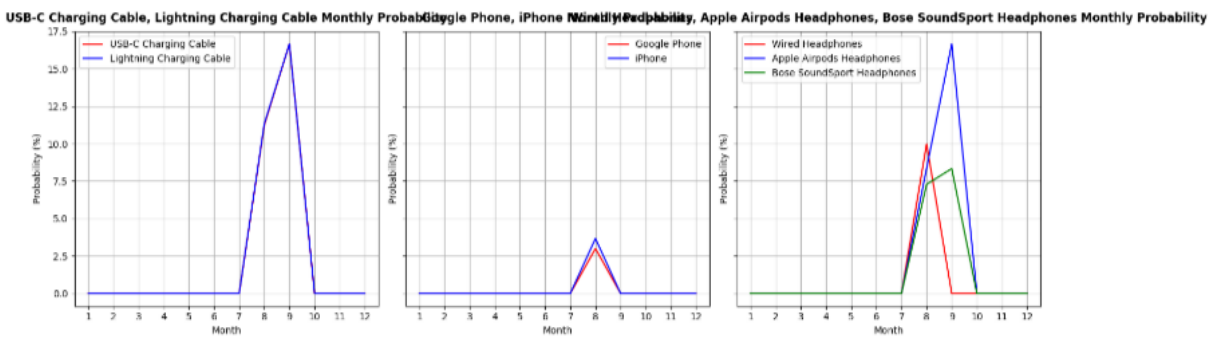


```
    A tuple (prob_month, prob_year):
    prob_month -- A numpy array of monthly probabilities.
    prob_year -- The yearly probability.
    """
    # Total number of rows in the DataFrame
    total_rows = df.shape[0]

    # Filter the DataFrame for the specified product
    product_df = df[df['Product'] == product]
    product_rows = product_df.shape[0]

    # Calculate yearly probability for the specified product
    prob_year = round(product_rows / total_rows * 100, 2)

    # Initialize lists for monthly calculations
    monthly_probabilities = []

    # Calculate monthly probabilities
    for month in range(1, 13):
        # Total rows in the current month
        monthly_total = df[df['Month'] == month].shape[0]

        # Rows for the current product in the current month
        monthly_product = product_df[product_df['Month'] ==
month].shape[0]

        # Calculate monthly probability
        if monthly_total == 0:
            monthly_probabilities.append(0)  # Handle division by zero
gracefully
        else:
            monthly_probability = round((monthly_product /
monthly_total) * 100, 3)
            monthly_probabilities.append(monthly_probability)

    # Convert list of monthly probabilities to a numpy array
    prob_month = np.array(monthly_probabilities)

    return prob_month, prob_year
# Define the list of products to analyze
products = [
    'USB-C Charging Cable', 'Lightning Charging Cable', 'Google Phone',
'iPhone',
    'Wired Headphones', 'Apple Airpods Headphones', 'Bose SoundSport
Headphones'
]

# Create a row of three subplots with shared y-axis
```



```
fig, axes = plt.subplots(1, 3, figsize=(15, 5), sharey=True)

# Define the colors for plotting
colors = ['r', 'b', 'g']

# Define the pairs (or triplets) of products for each subplot
subplots_products = [
    ['USB-C Charging Cable', 'Lightning Charging Cable'],
    ['Google Phone', 'iPhone'],
    ['Wired Headphones', 'Apple Airpods Headphones', 'Bose SoundSport
Headphones']
]

# Iterate through each subplot and each set of products
for i, (products, ax) in enumerate(zip(subplots_products, axes)):
    for j, product in enumerate(products):
        prob_month, prob_year = proba_prod(product, df)

        # Plot monthly probabilities
        ax.plot(range(1, 13), prob_month, label=product,
color=colors[j])

    # Set plot title, labels, and grid
    ax.set_title(f'{", ".join(products)} Monthly Probability',
weight='bold', fontsize=12, pad=10)
    ax.set_xticks(range(1, 13))
    ax.set_xlabel('Month')
    ax.set_ylabel('Probability (%)')
    ax.grid(True)
    ax.legend()

# Adjust layout and display the plots
plt.tight_layout()
plt.show()
```

Output:



Manage relationship

Managing relationships is an essential aspect of any e-commerce sales data analysis project. The project involves various stakeholders, such as data scientists, business analysts, IT teams, and business executives, as well as third-party vendors and service providers. Here are some key aspects of managing relationships in this project:

Stakeholder Engagement and Communication:

- Maintain clear and open lines of communication with all stakeholders to ensure project goals are understood and aligned.
- Provide regular updates on project progress, milestones, and potential challenges.

Cross-Functional Collaboration:

- Foster collaboration between different teams, such as data science, IT, and business units, to leverage diverse expertise and perspectives.
- Encourage teamwork and shared ownership of project outcomes.

Expectation Management:

- Clearly define project scope, deliverables, and timelines to set realistic expectations for stakeholders.
- Manage expectations regarding the potential impact and limitations of the analysis and models.

Vendor and Service Provider Management:

- Establish strong relationships with cloud service providers, data vendors, and other third-party partners.
- Negotiate contracts and service level agreements (SLAs) that align with project needs and business objectives.

Feedback and Continuous Improvement:

- Gather feedback from stakeholders on project deliverables and processes to identify areas for improvement.

- Use feedback to make iterative adjustments to the project approach and deliver better results.

Data Governance and Compliance:

- Work closely with legal and compliance teams to ensure data privacy and security regulations are followed.
- Establish clear data governance policies to manage data access, usage, and storage.

Change Management:

- Prepare stakeholders for changes resulting from the project, such as new data-driven strategies or technology implementations.
- Provide training and support to help stakeholders adapt to changes effectively.

Performance Measurement and Reporting:

- Measure and report on key performance indicators (KPIs) to demonstrate the project's value and impact.
- Use performance data to justify ongoing investment and support for the project.

By effectively managing relationships with stakeholders, teams, and vendors, the project can achieve its objectives and deliver meaningful insights that drive business success. Strong relationships contribute to a collaborative and productive project environment, enabling efficient execution and optimal outcomes.

Project result:

```
Probability in year for Wired Headphones: 9.97%
Probability in year for Apple Airpods Headphones: 8.36%
Probability in year for Bose SoundSport Headphones: 7.28%
Total orders in 2019: 11,957 orders
Total products sold in 2019: 13,442 items
Total sales in 2019: 2,244,412.3099999996 USD
```


CONCLUSION

The project serves as a valuable tool for retail businesses seeking to optimize their operations and improve performance. By leveraging sales data, the project generates actionable insights that inform decision-making in various areas, including sales strategies, inventory management, marketing, and customer service. Insights into monthly and hourly sales trends enable businesses to tailor their strategies to peak demand periods, resulting in improved sales and profitability. Analysis of popular products and city-specific sales helps businesses manage inventory more effectively, ensuring that in-demand products are consistently available and avoiding overstock. City-specific performance analysis allows businesses to target marketing campaigns and promotions to specific regions, increasing the chances of success. Identifying common product combinations purchased together provides opportunities for cross-selling and bundling strategies, enhancing average order value. The project offers a foundation for ongoing performance monitoring, allowing businesses to track sales performance and adapt strategies as needed to stay competitive. Additionally, calculating the probability of selling specific products annually and monthly helps businesses optimize inventory levels and sales promotions. Overall, the project provides data-driven insights that empower businesses to make informed decisions, leading to increased efficiency, profitability, and customer satisfaction. By continuing to monitor and analyze sales data, businesses can maintain a competitive edge and adapt to changing market dynamics.

FUTURE SCOPE

The future scope of the project encompasses several areas that can further enhance its capabilities and impact. One potential development is the integration of additional data sources such as customer reviews, demographic data, and competitive market data, providing a more comprehensive view of sales performance and customer behavior. Machine learning algorithms can be employed to predict future sales trends and identify potential areas for growth, enabling more precise and proactive decision-making. Advanced analytics can also be used to uncover deeper insights into customer preferences and emerging market trends. Additionally, real-time data processing and visualization can offer up-to-date insights, allowing businesses to make agile decisions in response to dynamic market conditions. Lastly, expanding the project to incorporate sentiment analysis and customer segmentation can help businesses tailor their strategies to specific target markets, fostering stronger customer relationships and brand loyalty. By exploring these avenues, the project can continue to evolve and deliver even greater value to retail businesses.

REFERENCES

1. Project Github link (https://github.com/SUBHASHINI299/E-Commerce/tree/main), Subhashini E G, 2024
2. Project video recorded link (https://www.youtube.com/watch?v=lhjn4qI28eg), Subhashini E G, 2024
3. Project PPT & Report github link (https://github.com/SUBHASHINI299/E-Commerce/tree/main/Report%20and%20PPT), Subhashini E G, 2024
4. Project Dataset Github link (https://github.com/SUBHASHINI299/E-Commerce/tree/main/Dataset), Subhashini E G 2024

GIT Hub Link of Project Code:

https://github.com/SUBHASHINI299/E-Commerce/tree/main/Code

THANK YOU

