PERSONALIZED MARKETING STRATEGIES: LEVERAGING ENSEMBLE METHODS AND COLLABORATIVE FILTERING FOR TARGETED PROMOTIONS

A MINI PROJECT REPORT

Submitted by

HARSAVARDHINI R (221801016)
KAVIYA S (221801024)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY IN ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

RAJALAKSHMI ENGINEERING COLLEGE
DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

ANNA UNIVERSITY, CHENNAI

NOV 2024

ANNA UNIVERSITY, CHENNAI

BONAFIDE CERTIFICATE

Certified that this Report titled "**PERSONALIZED MARKETING STRATEGIES: LEVERAGING ENSEMBLE METHODS AND COLLABORATIVE FILTERING FOR TARGETED PROMOTIONS**" is the bonafide work of **HARSAVARDHINI R (221801016), KAVIYA S (221801024)** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Dr. J.M. Gnanasekar
Professor and Head
Department of Artificial Intelligence
and Data Science
Rajalakshmi Engineering College
Chennai – 602 105

Dr. V.Saravana Kumar
Professor
Department of Artificial Intelligence
and Data Science
Rajalakshmi Engineering College
Chennai – 602 105

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman Mr. S. MEGANATHAN, B.E, F.I.E., our Vice Chairman Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S., and our respected Chairperson Dr. (Mrs.) THANGAM MEGANATHAN, Ph.D., for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution.

Our sincere thanks to Dr. S.N. MURUGESAN, M.E., Ph.D., our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to Dr. J.M. GNANASEKAR., M.E., Ph.D., Professor and Head of the Department of Artificial Intelligence and Data Science for her guidance and encouragement throughout the project work. We are glad to express our sincere thanks and regards to our supervisor Dr. V. SARAVANA KUMAR., M.Tech., PhD, Professor, Department of Artificial Intelligence and Data Science and coordinator, Dr. P. INDIRA PRIYA., M.E., Ph.D., Professor, Department of Artificial Intelligence and Data Science, Rajalakshmi Engineering College for his valuable guidance throughout the course of the project.

Finally, we express our thanks for all teaching, non-teaching, faculty and our parents for helping us with the necessary guidance during the time of our project.

ABSTRACT

In the competitive retail landscape, personalized targeting has become a crucial strategy for companies aiming to enhance customer engagement and increase sales. This research study leverages the Global Data Superstore Data set to develop a personalized marketing model that tailored promotions and recommendations to individual customers based on their historical purchase behavior. By combining collaborative filtering, Random Forest classifiers, and ensemble methods, the proposed system offers targeted product recommendations and dynamic discounts that align with customers' preferences. The system utilizes both user-based and item-based collaborative filtering techniques to deliver personalized promotions at multiple levels, accurately predicting customers' subsequent purchases and corresponding discounts. Predictive analysis further enriches the model by forecasting buying behaviors, allowing for the timely design of promotional messages that resonate with customers. The project is structured into key components: data preprocessing, where raw data is cleaned and prepared; predictive modeling, which identifies purchasing patterns and preferences; and promotional personalization, which assigns discounts and creates tailored promotional strategies. The efficiency and impact of this system extend beyond improving sales figures, as it also significantly boosts customer satisfaction by delivering more relevant offers and fostering long-term customer loyalty. Additionally, the study provides insights into the limitations of traditional segmentation methods, advocating for the adoption of machine learning techniques to optimize marketing strategies. By analyzing current practices and introducing data-driven methods, this project demonstrates how retailers can achieve more effective, timely, and targeted engagements with consumers, ultimately enhancing the overall customer experience while optimizing promotional efforts.

ANNEXURE I

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 GENERAL

The project focuses on improving marketing strategies in the retail sector by utilizing personalized promotions and discounts. Traditional retail marketing often relies on broad and uniform promotional tactics, which do not cater to the specific needs and preferences of individual customers. This approach can lead to lower engagement rates and missed opportunities for sales growth.

The aim is to revolutionize how promotions are delivered by developing a system that analyzes customer purchase history and preferences to offer tailored recommendations. By leveraging data analytics and machine learning techniques, the project seeks to better understand customer behavior and predict future purchases, allowing for targeted marketing efforts that enhance customer satisfaction and increase sales.

1.2 NEED FOR THE STUDY

The need for this study arises from the limitations of traditional marketing approaches in the retail industry, which often fail to effectively engage customers or drive sales growth. Conventional promotional strategies typically use blanket discounts or generalized offers that do not consider individual customer preferences. This lack of personalization can lead to lower customer satisfaction, reduced promotion effectiveness, and missed revenue opportunities.

With the growing availability of customer data and advances in data analytics, there is an opportunity to transform marketing strategies by adopting a more targeted approach. By analyzing customer purchase history and behavior, retailers can identify specific preferences and buying patterns, enabling them to deliver personalized promotions. This targeted approach not only enhances customer engagement but also optimizes marketing resources, making promotional efforts more efficient and impactful.

Thus, the study aims to address these gaps by developing a personalized marketing system that leverages machine learning algorithms to predict future purchases and customize promotions accordingly. This will help improve customer satisfaction and drive sales in a competitive retail environment.

1.3 OVERVIEW OF THE PROJECT

The project aims to develop a personalized marketing system for a retail chain, focusing on enhancing customer engagement and boosting sales through targeted promotions and discounts. Traditional marketing methods, which often involve generalized promotions, fail to cater to the unique preferences and behaviors of individual customers. This project seeks to address these limitations by implementing a data-driven approach that utilizes machine learning techniques.

The system leverages customer purchase history and preferences to predict future buying behavior and deliver personalized recommendations. Key machine learning algorithms such as Random Forest, Gradient Boosting, and Collaborative Filtering are employed to segment customers, recommend products, and optimize promotional strategies. The overall goal is to create a dynamic and adaptive marketing model that improves promotion effectiveness, increases redemption rates, and ultimately drives sales growth while enhancing customer satisfaction.

The project encompasses several modules, including data preprocessing, predictive modeling, recommendation engines, and promotion personalization, all working together to build a comprehensive solution for personalized marketing in the retail industry.

1.4 OBJECTIVES OF THE STUDY

The project aims to revolutionize retail marketing strategies by implementing a personalized marketing system that utilizes machine learning techniques to enhance customer engagement and drive sales growth. Traditional retail promotions often rely on generalized discounts that do not account for the unique preferences and buying behavior of individual customers. This approach can result in lower customer satisfaction and inefficient use of marketing resources. The project addresses these limitations by using data-driven methods to analyze customer purchase history and preferences, allowing for the prediction of future buying behavior and the delivery of tailored promotions.

The main objectives are to increase sales and customer satisfaction by providing customized promotions that resonate with each customer's interests. By predicting the next likely purchase, the system aims to make promotional efforts more relevant and timely. Another objective is to enhance customer engagement by delivering personalized recommendations that encourage repeat purchases. The project also seeks to optimize the allocation of marketing resources, ensuring that promotional efforts target customers who are most likely to respond positively. Additionally, it aims to improve the redemption rates of discounts and offers by making them more appealing to customers based on their past behavior and preferences. Overall, the study aims to transition the retail chain from a traditional, one-size-fits-all marketing approach to a dynamic, data-driven model that leverages customer insights for more effective marketing strategies.

CHAPTER II

REVIEW OF LITERATURE

2.1 INTRODUCTION

The literature survey explores various approaches and techniques for enhancing retail marketing through personalized recommendations and data-driven methods. Traditional customer segmentation often relies on the RFM (Recency, Frequency, Monetary) model, combined with methods such as K-means clustering for grouping customers based on their purchase behavior. While effective to some extent, these techniques face limitations in capturing complex customer behavior patterns and adapting to changing preferences.

Several studies highlight the advantages of collaborative filtering for product recommendations in retail, showing that memory-based approaches like k-NN outperform model-based techniques in certain cases. Recent research has introduced hybrid models that integrate collaborative filtering, RFM analysis, and association rules mining, improving recommendation accuracy and addressing issues like data sparsity and cold-start problems. Additionally, advanced algorithms such as boosting trees, SVM, and sequential pattern mining have been utilized to refine customer segmentation and predict sales more accurately. These approaches demonstrate the potential to enhance retail marketing through better personalization and targeted promotions, setting the foundation for the project's proposed system.

2.2 FRAMEWORK OF LCA

Traditional Methods of Customer Segmentation: Traditional approaches to customer segmentation, such as the RFM (Recency, Frequency, Monetary) model, categorize customers based on their purchasing behavior using three key metrics. This model is often paired with techniques like K-means clustering to group customers into segments. However, these methods have limitations, as they rely on a narrow set of features and often result in static customer segments. This can lead to outdated representations of customer behavior, reducing the effectiveness of marketing efforts in capturing evolving preferences and trends.

Collaborative Filtering for Recommendations: Collaborative filtering techniques, commonly used in retail for generating product recommendations, focus on predicting a customer's interest based on similar users or items. Memory-based methods, such as k-Nearest Neighbors (k-NN), and model-based techniques, like Singular Value Decomposition (SVD), have been widely adopted. Research indicates that memory-based approaches can be particularly effective in offline retail scenarios. However, collaborative filtering faces challenges such as the cold-start problem, where recommendations may not be accurate for new users or items with insufficient data.

Hybrid and Enhanced Techniques: To improve recommendation accuracy and address the limitations of single-method approaches, hybrid techniques that combine multiple methods have emerged. For instance, integrating collaborative filtering with RFM analysis and association rules mining can enhance the recommendation system's performance. Additionally, advanced methods like sequential pattern mining and ontology-based models have been used to refine customer segmentation. These hybrid models effectively address data sparsity and cold-start issues by leveraging diverse data sources and algorithms.

Machine Learning for Predictive Modeling: The use of advanced machine learning algorithms for predictive modeling in retail has shown significant potential for enhancing marketing strategies. Algorithms such as Random Forest, Gradient Boosting, and Support Vector Machines (SVM) are employed to predict customer behavior, segment customers, and anticipate future purchases. By using data mining and classification techniques for customer profiling, these methods offer more sophisticated segmentation and better-targeted marketing efforts. This shift towards data-driven strategies enables marketers to improve sales through highly personalized and timely promotions.

CHAPTER III

SYSTEM OVERVIEW

3.1 EXISTING SYSTEM

The existing system for customer segmentation and marketing in retail primarily relies on traditional methods such as the RFM (Recency, Frequency, Monetary) model, which segments customers based on their purchasing history. This model helps categorize customers into different groups by considering how recently they purchased, how often they make purchases, and the monetary value of their transactions. Techniques like K-means clustering is used alongside RFM to further classify customers into segments, allowing for basic targeted marketing.

Moreover, techniques such as Association Rule Mining (e.g., Apriori or FP-Growth) are employed to identify relationships between products frequently bought together, which can be used for cross-selling or bundling promotions. Additionally, Naive Bayes and other basic classification algorithms are applied to predict the likelihood of customers responding to promotions based on their historical purchasing patterns or demographic information.

However, these traditional approaches have several limitations. The RFM model is constrained by its use of only three metrics, potentially overlooking other important aspects of customer behavior and preferences. As a result, customer segments may not be nuanced enough to reflect the diversity of individual preferences. Traditional methods also create static customer segments that may not adapt to changing customer behaviors, leading to outdated marketing strategies. Furthermore, Association Rule Mining techniques can struggle with scalability and memory issues when handling large datasets, reducing their effectiveness in big data scenarios. These limitations underscore the need for more sophisticated, data-driven approaches that can offer dynamic and personalized marketing solutions.

3.2 PROPOSED SYSTEM

The proposed system aims to enhance retail marketing strategies by implementing a personalized marketing framework that leverages advanced data analytics and machine learning techniques. Unlike traditional methods, which rely on static customer segmentation and generalized promotions, the proposed system focuses on delivering dynamic, tailored promotions based on individual customer preferences and purchasing behavior. The goal is to improve customer satisfaction and drive sales by accurately predicting future purchases and offering personalized recommendations.

Key components of the proposed system include predictive modeling and recommendation engines. For predictive modeling, machine learning algorithms such as the Random Forest classifier are used to forecast the next likely product or category a customer will purchase. This prediction is based on analyzing historical purchase data to identify patterns in customer behavior. The recommendation engine employs both user-based and item-based collaborative filtering techniques to further personalize promotions. By analyzing similarities among customers and products, it generates tailored offers that match each customer’s preferences. Discounts are assigned based on purchase probabilities derived from the model's predictions, ensuring that promotions are both relevant and appealing.

In addition, a promotion personalization module is integrated to customize discount rates for different product categories. Using a Gradient Boosting Regressor, the system predicts discount rates based on features such as total sales and purchase quantities for each customer, providing an additional layer of personalization. The final output includes the recommended products, associated discounts, and predicted probabilities of customer engagement, allowing for targeted and effective marketing.

Overall, the proposed system offers a dynamic and adaptive marketing approach that not only increases sales but also optimizes marketing resources by focusing on the most relevant and personalized offers for each customer. This transition from traditional to data-driven marketing enables the retail chain to achieve a more customer-centric approach to promotions.

3.3 FEASIBILITY STUDY

Technical Feasibility: The proposed system leverages established machine learning algorithms such as Random Forest, Gradient Boosting, and Collaborative Filtering, which are well-suited for predictive modeling and recommendation tasks. The use of these algorithms is technically feasible, given their proven effectiveness in handling high-dimensional data and their ability to generate accurate predictions based on historical purchase behavior. Additionally, the necessary tools and frameworks for implementing these algorithms, such as Python libraries (e.g., scikit-learn, Surprise), are widely available and supported. The data preprocessing steps, including handling missing values, feature selection, and normalization, can be effectively managed with standard data processing techniques, ensuring the system's technical soundness.

Operational Feasibility: The implementation of the personalized marketing system aligns with the retail chain's operational goals of increasing sales and customer engagement. By integrating the system with existing data sources, such as customer purchase history and transaction records, it can seamlessly fit into the current marketing workflow. The modules for predictive modeling, recommendation engines, and promotion personalization are designed to be modular and adaptable, allowing the retail chain to gradually adopt the system without significant disruptions. Furthermore, the approach of using data-driven insights to guide promotional strategies ensures that the system can continuously improve and adapt to changing customer behaviors, making it operationally viable.

Economic Feasibility: From an economic standpoint, the project has the potential to deliver significant returns on investment by increasing the effectiveness of promotional campaigns. The personalized approach to marketing is expected to boost sales and improve customer satisfaction, leading to higher revenue and potentially reducing marketing costs by optimizing resource allocation.

CHAPTER IV

SYSTEM REQUIREMENTS

4.1 HARDWARE REQUIREMENTS

Server/Workstation Specifications:

- Processor: Intel Core i7 or higher (for development), Xeon processors (for production environment) to handle high computational tasks.
- RAM: Minimum of 8 GB (development) since data processing can also be done in Colab.
- Storage: SSD with at least 512 GB (development) and 1 TB or higher (production) to store datasets and model artifacts.
- GPU: Use Google Colab's provided GPU (e.g., Tesla K80, T4, P100, or V100) for training deep learning models and accelerating machine learning tasks.

Network Requirements:

- High-speed internet connection for accessing cloud services (if needed) and data transfer.
- Secure local network for on-premises deployment.

4.2 SOFTWARE REQUIREMENTS

- Operating System: Windows 10/11 (Ubuntu 20.04 or higher) for development environments.
- Programming Languages: Python (version 3.7 or higher): Primary programming language for developing machine learning models and data processing scripts.
- Machine Learning Libraries: scikit-learn for building predictive models.
- Data Processing Libraries: Pandas, NumPy, SciPy for data manipulation and preprocessing.
- Visualization Tools: Matplotlib, Seaborn for creating plots and visualizations.

CHAPTER V

SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

The model architecture explores a system for personalized marketing in the retail sector, leveraging Random Forest Classifiers and Collaborative Filtering to generate personalized promotions based on customer purchase history and preferences. The system, referred to as Promotions and Recommendations, is designed to optimize marketing resources while increasing customer satisfaction. The architecture includes several key components such as data preprocessing, predictive modeling, and recommendation engine modules. These components work together to process raw retail data, predict customer behavior, and deliver personalized offers to individual customers.

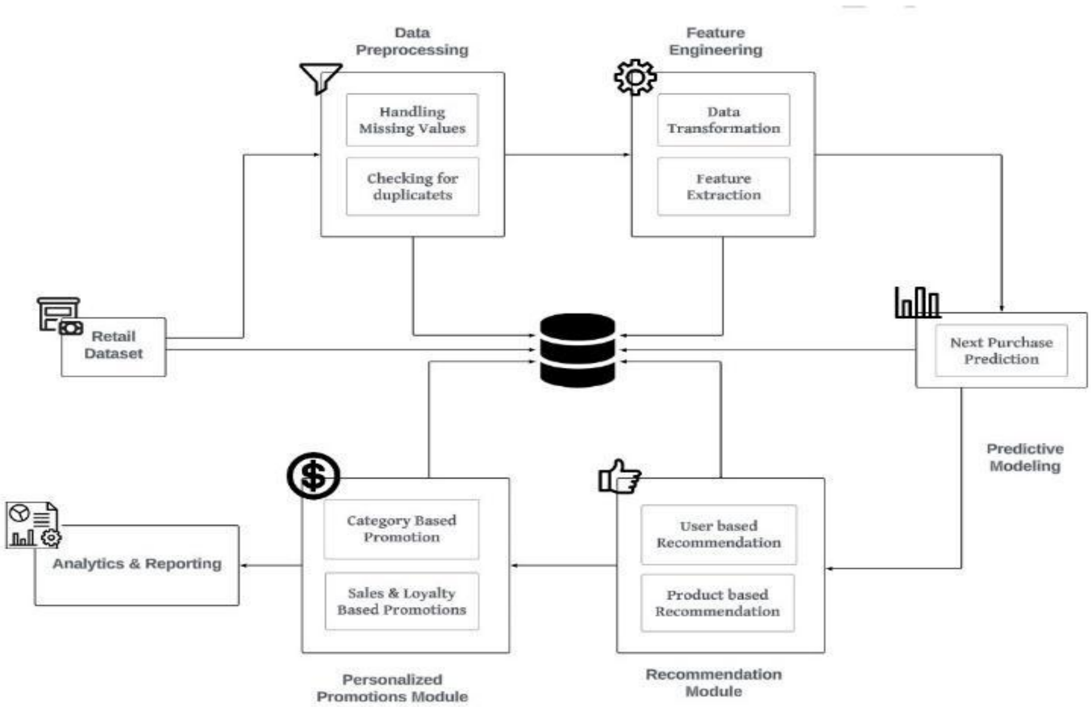


Figure 1: System Architecture

5.2 MODULE DESCRIPTION

5.2.1 PREPROCESSING MODULE

The Data Preprocessing module therefore has a significant task of preparing the dataset for analysis and modeling for increased levels of data analysis. This refers to several processes that are basic to the operation of the model mentioned above. Such steps include dealing with the missing data, how to select the features to be used, data pre- processing among others. Handling Missing Values: The first operation to perform before applying the actual data preparation steps is handling missing values within data. There is always a problem with missing data as this is likely to introduce bias and compromise the quality of various algorithm techniques in machine learning. To calculate the percentage of the moving average, the system employs df.isnull().sum()/df.shape[0]*100 on every column. Accordingly, any features that include substantially large numbers of missing values are either eliminated or dealt with through simple imputations of missing values. Example: Sometimes, Attributes such as Postal Code or Discount may have to be deleted since most of the time their values are missing and do not help the model.

Correlation Analysis: After that, when it comes to dealing with missing values, the system performs correlation heatmap analysis. In this process the company finds out how various features are related to one variable of interest such as frequency of purchases. Similar features, which are those with low correlation coefficients are removed, and those with high correlation coefficients are retained. This step makes it easier to reduce the database to a set of the most significant vectors, which makes the work of models more efficient. Example: Domain knowledge and decision-making abilities informs measures like Customer Age or Region might have very low correlation with purchase frequency and hence are discarded.

Feature Scaling and Encoding: Once these features needed are selected, scaling and encoding are carried out. Numerical features are also scaled to fall within the same range so that we can effectively address the scale and feature importance issue, where one feature will have a dominating effect on the model. Example:

Some of the product categories such as Electronics, Clothing are then transformed into binary variables (1 for purchased and 0 for not purchased) to fit the predictive modeling.

Data Splitting: Upon preprocessing, the obtained plain and structured dataset is then split into training and testing datasets. This split lets the model be tested using unique data and hence gives a better understanding of the model's performance. In most cases, the dataset is divided where 80% of data is used for model training and 20% data are used for model testing and validation. At the end of data preprocessing phase, data cleansing is performed on the dataset, extraneous information is removed and what the Predictive Modeling module will feed to the machine learning algorithms is meaningful data.

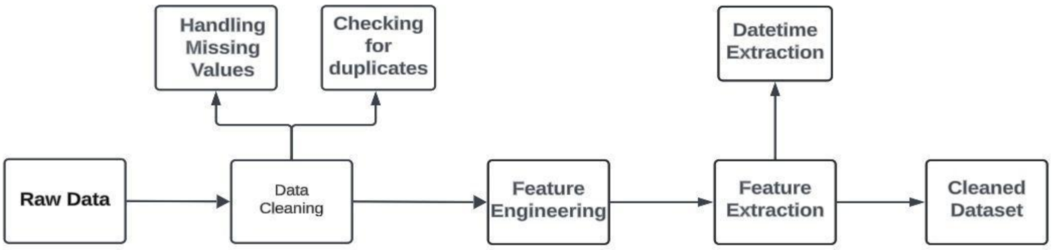


Figure 2: Data Preprocessing

5.2.2 PREDICTIVE MODELING MODULE

The Predictive Modeling sub-module is another important part in the system where it is envisaged to predict future customer purchasing patterns from transaction history. The principal goal of this module, therefore, is to increase the accuracy of targeted advertisements by determining what the next product a customer is likely to buy. There is so much speculation before a quantitative data texture analysis; the entropy of the number of purchasers is computed first. This entropy calculation assists in choosing the features with higher information by pointing out those options that bring about the highest entropy. For instance, product type, purchase rate, and client information is assessed for change prediction capability.

Another crucial part in the system is the Predictive Modeling sub-module where it is planned to predict customer purchasing trends in the future in transactions. The main purpose of this module hence is to enhance probability of accurate targeted advertisements by finding out what the customer is likely to buy next. Before a quantitative data texture analysis there is much anticipations; first the entropy of the number of purchasers is determined. This entropy calculation helps in selecting the features with higher information by highlighting the options with highest entropy.

$$Entropy(s) = \sum_{i=1}^{n} p_i \log_2(p_i)$$

For example, when measuring the change prediction capacity of a reference artifact, product type, purchase rate, and client information are examined.

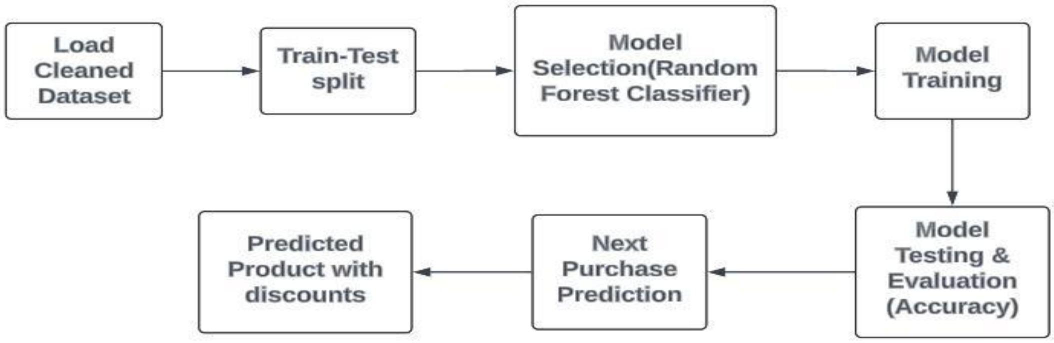

Load Cleaned Dataset
Train-Test split
Model Selection(Random Forest Classifier)
Model Training
Model Testing & Evaluation (Accuracy)
Next Purchase Prediction
Predicted Product with discounts

Figure 3: Predictive Modeling

5.2.3 RECOMMENDATION ENGINE MODULE

The Recommendation Engine employs Collaborative Filtering algorithms to create a product recommendation list based on user’s activity, and similarity among the products. This engine is divided into two main approaches: These two strategies include; User-Based Collaborative Filtering and Item-Based Collaborative Filtering. User based collaborative filtering is a technique where the system groups customers who have similar buying patterns and then suggests to a particular user items that the similar user has bought but other similar users haven’t bought. This process employs the K-nearest Neighbors (KNN) algorithm in an effort to create groups of customers, in view of their purchasing histories, in order to recommend products that will be relavant to the particular customer. For instance, in the case of two customers, having similar past purchase patterns, products bought by the one but not the other are suggested together with a calculated Purchase Probability.

$$\mathbf{r}_{\mathrm{ui}} = \frac{\sum_{v \in N(u)} w_{uv} . r_{vi}}{\sum_{v \in N(u)} | w_{uv} |}$$

On the other hand, Item-Based Collaborative Filtering focuses on suggesting products similar to those already purchased by the customer. The system identifies similarities between products based on features such as purchase frequency, customer ratings, and product categories. Using cosine similarity and the KNN algorithm, the system predicts ratings for unpurchased products and converts these ratings into recommendation probabilities.

$$\text{Recommendation Probability} = \frac{r_{ui}}{MaxQuantity} \times 100$$

Products with the highest probabilities are recommended, and the system further enhances customer engagement by assigning personalized discounts to the recommended products. For instance, products with a lower likelihood of purchase are offered at a higher discount to incentivize the customer, while those with higher purchase probabilities receive smaller discounts. This approach ensures that recommendations are not only relevant but also strategically incentivized to maximize conversions.

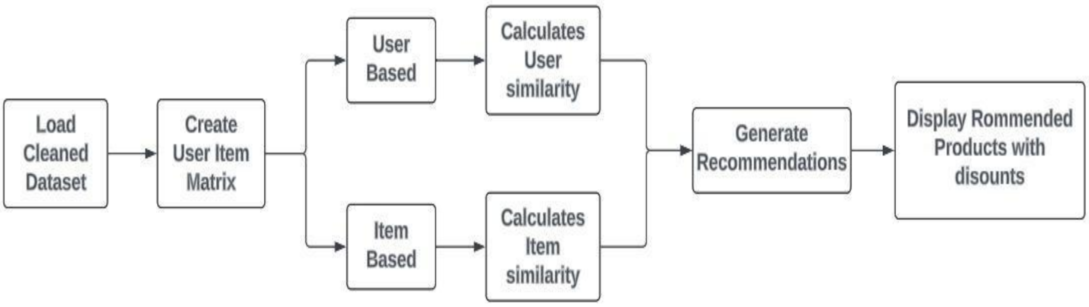

Load Cleaned Dataset
Create User Item Matrix
User Based
Item Based
Calculates User similarity
Calculates Item similarity
Generate Recommendations
Display Rommended Products with disounts

Figure 4: Recommendation Engine

5.2.4 PROMOTION PERSONALIZATION MODULE

The Promotion Personalization module provides offers and discounts personalized to each purchasing customer looking at their preferences in the product. To start with, this module computes the total sale and the total quantity bought for each customer for different product types. All these metrics form feature inputs for the model chosen as the Gradient Boosting Regressor to forecast the best discount rate for every customer. Gradient Boosting applies the concept of machine learning by building decision tree consecutively: the following tree tries to minimize them is takes which the previous tree made, thus arrives at a correct value of the discount rate which has high likelihood to lead to customers' purchase.

$$Discount\ Amount = \frac{\text{Predicted Discount } \times \text{Total Sales}}{100}$$

The Promotion Personalization module gives out offer and discounts in participation to the customer who owns the product with respect to the customer's profile in the product. First of all, this module calculates the total sale, and total quantity customers bought for each customer for the products which belongs to the different types. All these metrics create feature inputs for the model which was selected as the Gradient Boosting Regressor to predict the best discount rate for each customer. Gradient Boosting applies the concept of machine learning by building decision tree consecutively: the following tree tries to avoid such mistakes which the previous tree made and ends up with a right value of the discount rate which hold real potentiality to influence the customers into purchasing.

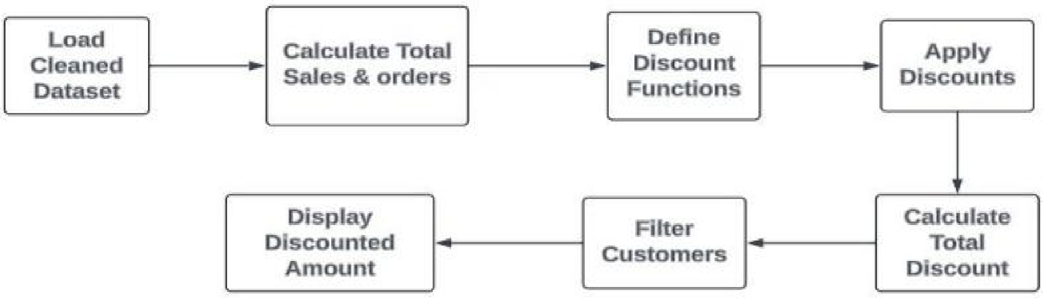

Load Cleaned Dataset
Calculate Total Sales & orders
Define Discount Functions
Apply Discounts
Calculate Total Discount
Filter Customers
Display Discounted Amount

Figure 5: Promotion Personalization

CHAPTER VI

RESULT AND DISCUSSION

Compared to traditional marketing systems where promotion is done through the rule-based or demographic segmentation, this machine learning-based system offered certain distinct benefits. It becomes apparent that most traditional marketing & communication systems are unable to obtain detailed customer behavioral patterns, resulting in mass and comparatively ineffective efforts in marketing communication. On the other hand, using Random Forests for prediction, and Collaborative for recommendation allowed the client to specify the promotional actions more accurately due to better understanding of the customers’ preferences and activities. Furthermore, by continuously updating the customers’ profile and behavior in real time, the promotion offered are also valid as the customers change over time. This was much better than the typical fixed segmentation approaches used in traditional marketing that may well soon become obsolete. With the proposed system, the levels of promotion relevance were significantly higher and delivered ina continuous update and real time personalization the levels of relevance were at 99%.

CHAPTER VII

CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION

The Promotion Personalization module identified as the module using the Gradient Boosting Regressor provided the best accuracy in the task of recommending the optimal discount rates for each client. Hence the features like total target sales and purchase quantity, the model proposed in this paper was able to predict the right discounts with an accuracy of 64.94 percent it created a way for higher customer compensation and retail returns as well. The above observation motivates the use of personalization in promotions since the customers who were targeted bought more than the other customers who were not targeted through personalized promotions specific communication. On average, the customers bought more under the personalized discounting system by 20% during the testing period of this study. It also disclosed superior cost savings since discounts were offered in terms of a probability distribution of purchase and therefore, did not squander an abundance of costless discounts on sizes that were most likely to be purchased.

7.2 FUTURE ENHANCEMENT

Future enhancements for the project could include incorporating more advanced machine learning algorithms, such as deep learning models for improved personalization and prediction accuracy. Additionally, integrating real-time data processing could enable dynamic updates to recommendations and promotions, making them even more relevant. Expanding the system to include multi-channel marketing, such as social media and mobile notifications, would further enhance customer engagement. Moreover, incorporating customer feedback loops to refine models based on user responses could continuously optimize the effectiveness of promotions.

APPENDIX

A1.1 SAMPLE CODE

1.DATA PREPROCESSING MODULE

```
!pip install --upgrade xlrd
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
file_path = '/content/Global Data Superstore.xls'

# Try to read the file with a different engine
try:
    df = pd.read_excel(file_path, engine='xlrd')
except:
    # If xlrd fails, try openpyxl
    df = pd.read_excel(file_path, engine='openpyxl')
```

```
df.head()
```

```
df.tail()
```

```
df.shape
```

```
df.info()
```

```
# Checking null values
df.isna().sum()
df.isnull().sum()/df.shape[0]*100
```

```
#Checking for duplicates
df.duplicated().sum()
```

```
#identifying Garbage values
for i in df.select_dtypes(include='object').columns:
  print(df[i].value_counts())
  print("***"*10)
```

```
#Exploratory Data Analysis
df.describe()
```

```
#Finding Correlation
df.select_dtypes(include='number').corr()
```

```
#Data Transformation
df['Unit Price']=df['Sales']/df['Quantity']
df.head()
```

```
#Visualizing Correlation in heatmap
sns.heatmap(df.select_dtypes(include='number').corr(),annot=True)
```

```
#Missing value treatments
df.drop('Postal Code', axis=1, inplace=True)
df.drop('Discount', axis=1, inplace=True)
```

```
# Feature Extraction
df['Order Year'] = df['Order Date'].dt.year
df['Order Month'] = df['Order Date'].dt.month
df['Order Day'] = df['Order Date'].dt.day
print(df.head())
```

```
#Save the cleaned dataset
df.to_csv('New_retail_dataset.csv', index=False)
# Download the file
from google.colab import files
files.download('New_retail_dataset.csv')
```

2.PREDICTIVE MODELING MODULE

Next Purchase prediction with discounts using Random Forest Classifier

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

# Load the dataset
data = pd.read_csv('/content/New Retail Dataset.csv',
encoding='latin-1')

data = data.sort_values(by=['Customer ID', 'Order Date'])

data['Next Product Name'] = data.groupby('Customer ID')['Product
Name'].shift(-1)
data['Next Category'] = data.groupby('Customer
ID')['Category'].shift(-1)

data = data.dropna(subset=['Next Product Name', 'Next Category'])

# Select features and target for modeling
features = ['Order Year', 'Order Month']
X = data[features]
y_product = data['Next Product Name']
y_category = data['Next Category']

X_train, X_test, y_product_train, y_product_test = train_test_split(
    X, y_product, test_size=0.3, random_state=42
)

# Random Forest Classifier for predicting the next product
rf_classifier_product = RandomForestClassifier(n_estimators=100,
random_state=42)
rf_classifier_product.fit(X_train, y_product_train)
```

```
# Predict next product
y_product_pred = rf_classifier_product.predict(X_test)

# Evaluate the product prediction
product_accuracy = accuracy_score(y_product_test, y_product_pred)
print(f'Product Prediction Accuracy: {product_accuracy * 100:.2f}%')

# Similarly for Category Prediction
X_train, X_test, y_category_train, y_category_test =
train_test_split(
    X, y_category, test_size=0.3, random_state=42
)

rf_classifier_category = RandomForestClassifier(n_estimators=100,
random_state=42)
rf_classifier_category.fit(X_train, y_category_train)

# Predict next category
y_category_pred = rf_classifier_category.predict(X_test)

# Evaluate the category prediction
category_accuracy = accuracy_score(y_category_test, y_category_pred)
print(f'Category Prediction Accuracy: {category_accuracy * 100:.2f}%')

# Function to assign promotions and discounts
def assign_discount(product, category):
    # Example discount assignment logic based on product or category
    if category in ['Electronics', 'Accessories']:
        return "20% off", "Special deal on electronics and
accessories!"
    elif product in ['Product A', 'Product B']:  # Replace with real
product names
        return "15% off", "Limited-time offer on selected products!"
    else:
        return "10% off", "Enjoy a general discount on your next
purchase!"

# Predict the next purchase for a specific customer
def predict_next_purchase(customer_id, model_product, model_category,
data, features):
    # Filter customer data
    customer_data = data[data['Customer ID'] == customer_id][features]

    # Predict using the RandomForest model
    predicted_product = model_product.predict(customer_data.head(1))
    predicted_category = model_category.predict(customer_data.head(1))

    # Assign discounts and promotions
    discount, promotion = assign_discount(predicted_product[0],
predicted_category[0])

    return predicted_product[0], predicted_category[0], discount,
promotion

# Example: Predict next purchase for a specific customer
customer_id = 'RH-19495'  # Replace with the actual customer ID
next_product, next_category, discount, promotion =
predict_next_purchase(
    customer_id, rf_classifier_product, rf_classifier_category, data,
features
)

# Output the prediction, discount, and promotion
print(f"Next predicted product for Customer {customer_id}:
{next_product} in Category: {next_category}")
print(f"Assigned Discount: {discount}")
print(f"Promotion: {promotion}")
```

3.RECOMMENDATION ENGINE MODULE

i)Personalized Promotions and Discounts Using User Based Collaborative Filtering

```
# Import necessary libraries
from surprise import Dataset, Reader, KNNBasic
from surprise.model_selection import train_test_split
import pandas as pd

# Load the dataset
df = pd.read_csv('/content/New_retail_dataset.csv', encoding='latin-
1')  # Update with your file path

# Filter the relevant columns for collaborative filtering
# Added 'Quantity' as a proxy for rating
df_filtered = df[['Customer ID', 'Product ID', 'Quantity']]

# Define a reader object for Surprise
reader = Reader(rating_scale=(1, df['Quantity'].max()))

# Load data into Surprise dataset
data = Dataset.load_from_df(df_filtered, reader)

# Split the dataset into train and test sets
trainset, testset = train_test_split(data, test_size=0.2)

# User-based Collaborative Filtering
sim_options = {
    'name': 'cosine',
    'user_based': True  # User-based collaborative filtering
}
user_based_model = KNNBasic(sim_options=sim_options)
user_based_model.fit(trainset)

# Function to calculate discounts based on predicted ratings
def calculate_discount(rating, max_rating):
    """Calculate discount percentage based on predicted rating."""
    if rating > 0.8 * max_rating:
        return 30  # 30% discount for highly recommended products
    elif rating > 0.6 * max_rating:
        return 20  # 20% discount for moderately recommended products
    else:
        return 10  # 10% discount for less recommended products

# Function to recommend products and provide discounts for a given
customer
def get_user_based_recommendations_with_discounts(customer_id, k=15):
    # Get customer name from Customer ID
    customer_name = df[df['Customer ID'] == customer_id]['Customer
Name'].values[0]

    # Find all products the customer has not yet interacted with
    customer_purchases = df[df['Customer ID'] ==
customer_id]['Product ID'].unique()
    all_products = df['Product ID'].unique()
    products_to_recommend = list(set(all_products) -
set(customer_purchases))

    # Predict ratings for all products the customer has not
interacted with
    predictions = []
    for product_id in products_to_recommend:
        pred = user_based_model.predict(customer_id, product_id)
        predictions.append((product_id, pred.est))  # Product ID and
estimated rating

    # Sort by predicted rating in descending order
    predictions.sort(key=lambda x: x[1], reverse=True)

    # Get top k recommendations
    top_k_recommendations = predictions[:k]
    recommendations = []
    max_quantity = df['Quantity'].max()  # Use this for scaling
discount
    for product_id, rating in top_k_recommendations:
        product_name = df[df['Product ID'] == product_id]['Product
Name'].values[0]
```

```
        percentage = round((rating / max_quantity) * 100, 2)  #
Convert rating to percentage
        discount = calculate_discount(rating, max_quantity)  #
Calculate discount based on rating
        recommendations.append({
            'Product ID': product_id,
            'Product Name': product_name,
            'Recommendation Probability (%)': percentage,
            'Discount (%)': discount
        })

    return customer_name, recommendations

# Example: Enter a customer ID to get recommendations and discounts
customer_id = 'AH-465'  # Replace with the desired Customer ID
customer_name, recommended_products =
get_user_based_recommendations_with_discounts(customer_id)

# Output the result
print(f"Customer Name: {customer_name}")
print("Recommended Products with Discounts:")
for rec in recommended_products:
    print(f"Product ID: {rec['Product ID']}, Product Name:
{rec['Product Name']}, "
          f"Probability: {rec['Recommendation Probability (%)']}%,
Discount: {rec['Discount (%)']}%")

# Save Recommendations and Discounts for All Customers
unique_customers = df['Customer ID'].unique()
recommendations_list = []

for customer_id in unique_customers:
    # Get recommendations and discounts for each customer
    customer_name, recommended_products =
get_user_based_recommendations_with_discounts(customer_id)

    # Collect the recommended product IDs and discounts as a comma-
separated string
    recommended_product_ids = ', '.join([f"{rec['Product ID']}
(Discount: {rec['Discount (%)']}%)"
                                         for rec in
recommended_products])

    # Append the customer ID and the recommended products with
discounts to the list
    recommendations_list.append({
        'Customer ID': customer_id,
        'Recommended Products with Discounts':
recommended_product_ids
    })

# Convert the list of dictionaries to a DataFrame
recommendations_df = pd.DataFrame(recommendations_list)

# Save the DataFrame to a CSV file
recommendations_df.to_csv('recommended_products_with_discounts.csv',
index=False)
print("Recommendations with discounts have been saved to
'recommended_products_with_discounts.csv'")
```

ii) Personalized Promotions and Discounts Using Item Based Collaborative Filtering

```
# Import necessary libraries
from surprise import Dataset, Reader, KNNBasic
from surprise.model_selection import train_test_split
import pandas as pd

# Load the dataset
df = pd.read_csv('/content/New_retail_dataset.csv', encoding='latin-
1')

# Filter the relevant columns for collaborative filtering
df_filtered = df[['Customer ID', 'Product ID', 'Quantity']]

# Define a reader object for Surprise
reader = Reader(rating_scale=(1, df['Quantity'].max()))

# Load data into Surprise dataset
data = Dataset.load_from_df(df_filtered, reader)

# Split the dataset into train and test sets
trainset, testset = train_test_split(data, test_size=0.2)

# Item-based Collaborative Filtering
sim_options = {
    'name': 'cosine',
    'user_based': False  # Item-based collaborative filtering
}
item_based_model = KNNBasic(sim_options=sim_options)
item_based_model.fit(trainset)

# Function to generate discounts based on recommendation probability
def get_discount(probability):
    if probability > 70:
        return "20% off"
    elif 50 <= probability <= 70:
        return "10% off"
    else:
        return "5% off"

# Function to recommend products and provide personalized
discounts/promotions
def get_item_based_recommendations_with_promotions(customer_id, k=20):
    # Get customer name from Customer ID
    customer_name = df[df['Customer ID'] == customer_id]['Customer
Name'].values[0]

    # Find all products the customer has interacted with
    customer_purchases = df[df['Customer ID'] ==
customer_id]['Product ID'].unique()

    # Predict ratings for products the customer has already
interacted with (item similarity-based recommendations)
    recommendations = []
    for product_id in customer_purchases:
        try:
            # Convert raw product ID to inner ID
            inner_id =
item_based_model.trainset.to_inner_iid(product_id)

            neighbors = item_based_model.get_neighbors(inner_id,
k=k)  # Get k nearest neighbors (similar items)

            for neighbor in neighbors:
                # Convert inner ID back to raw ID
                neighbor_raw_id =
item_based_model.trainset.to_raw_iid(neighbor)

                # Avoid recommending already purchased products
                if neighbor_raw_id not in customer_purchases:
                    predicted_rating =
item_based_model.predict(customer_id, neighbor_raw_id).est
                    product_name = df[df['Product ID'] ==
neighbor_raw_id]['Product Name'].values[0]

                    # Calculate recommendation probability
                    percentage = round((predicted_rating /
df['Quantity'].max()) * 100, 2)

                    # Get the appropriate discount based on the
recommendation probability
                    discount = get_discount(percentage)
                    # Add the product, its discount, and
recommendation probability
                    recommendations.append({
                        'Product ID': neighbor_raw_id,
                        'Product Name': product_name,
                        'Recommendation Probability (%)': percentage,
                        'Discount': discount
                    })
        except ValueError:
            pass  # Handle products not in training set

    # Sort by predicted rating in descending order and get top k
recommendations
    recommendations = sorted(recommendations, key=lambda x:
x['Recommendation Probability (%)'], reverse=True)[:k]

    return customer_name, recommendations

# Example: Enter a customer ID to get recommendations with promotions
customer_id = 'AH-465'  # Replace with the desired Customer ID
customer_name, recommended_products =
get_item_based_recommendations_with_promotions(customer_id)

# Output the result
print(f"Customer Name: {customer_name}")
print("Recommended Products with Promotions:")
for rec in recommended_products:
    print(f"Product ID: {rec['Product ID']}, Product Name:
{rec['Product Name']}, "
          f"Probability: {rec['Recommendation Probability (%)']}%,
Discount: {rec['Discount']}")
```

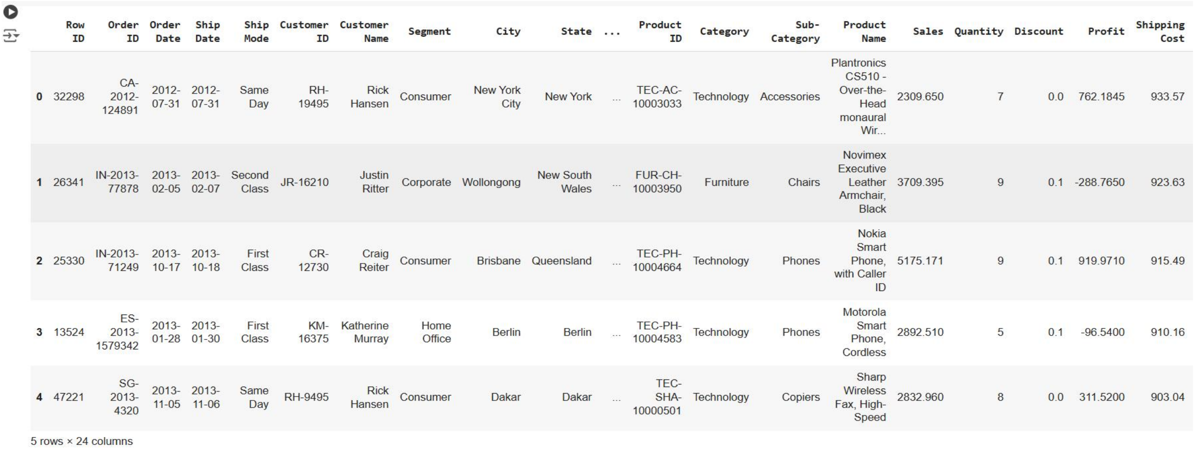
4.PERSONALIZED PROMOTION & DISCOUNT MODULE

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.metrics import mean_squared_error

# Load your dataset
df = pd.read_csv('/content/New_retail_dataset.csv', encoding='latin-
1')

# Ensure 'Order Date' is in datetime format
df['Order Date'] = pd.to_datetime(df['Order Date'])

# Calculate total sales per customer and per category
```

```
category_sales = df.groupby(['Customer ID', 'Customer Name',
'Category']) \
    .agg(Total_Sales=('Sales', 'sum'), Total_Quantity=('Quantity',
'sum')) \
    .reset_index()

# Convert 'Customer ID' to string
category_sales['Customer ID'] = category_sales['Customer
ID'].astype(str)

# Define discount rules for training based on categories
# We define different types of discounts (percentage, flat rate,
loyalty)
category_discount_map = {
    'Technology': {'Type': 'percentage', 'Value': 15},  # 15%
discount for Technology
    'Furniture': {'Type': 'flat', 'Value': 20},         # Flat $20
discount for Furniture
    'Office Supplies': {'Type': 'loyalty', 'Value': 5}  # Loyalty: $5
off for Office Supplies
}

# Assign discount types and values to the dataset
category_sales['Discount_Type'] =
category_sales['Category'].map(lambda x:
category_discount_map[x]['Type'])
category_sales['Discount_Value'] =
category_sales['Category'].map(lambda x:
category_discount_map[x]['Value'])

# Define features and target for training
X = category_sales[['Total_Sales', 'Total_Quantity']]  # Input
features
y = category_sales['Discount_Value']  # Target variable (Discount
Value)

# Split the dataset into training and testing sets (70% training, 30%
testing)
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=42)

# Initialize the Gradient Boosting Regressor model
gbr = GradientBoostingRegressor(n_estimators=100, learning_rate=0.1,
max_depth=3, random_state=42)

# Train the model
gbr.fit(X_train, y_train)
```

```
# Predict on the test set
y_pred = gbr.predict(X_test)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
print(f'Mean Squared Error: {mse:.2f}')

# Predict the discount for all customers in the dataset
category_sales['Predicted_Discount_Value'] = gbr.predict(X)

# Calculate the discount amount based on predicted discount values
and discount types
def calculate_discount(row):
    if row['Discount_Type'] == 'percentage':
        return row['Total_Sales'] * row['Predicted_Discount_Value'] /
100
    elif row['Discount_Type'] == 'flat':
        return row['Predicted_Discount_Value']
    elif row['Discount_Type'] == 'loyalty':
        return min(row['Total_Sales'],
row['Predicted_Discount_Value'])  # Cap loyalty discount to total
sales
    else:
        return 0

category_sales['Discount_Amount'] =
category_sales.apply(calculate_discount, axis=1)

# Print available customer IDs for user reference
print("Available Customer IDs:")
print(category_sales['Customer ID'].unique())

# Input customer ID from the user
customer_id_input = input("\nEnter Customer ID: ").strip()

# Filter data for the specified customer ID
customer_data = category_sales[category_sales['Customer ID'] ==
customer_id_input]

# Check if customer data exists
if not customer_data.empty:
    # Print results for the specified customer
    print(f"\nCustomer ID: {customer_id_input}")
    for index, row in customer_data.iterrows():
        print(f" - Name: {row['Customer Name']}")
        print(f" - Category: {row['Category']}")
        print(f" - Total Sales: ${row['Total_Sales']:.2f}")
        print(f" - Total Quantity: {row['Total_Quantity']}")
        print(f" - Predicted Discount Type: {row['Discount_Type']}")
        print(f" - Predicted Discount Value:
{row['Predicted_Discount_Value']:.2f}")
        print(f" - Discount Amount: ${row['Discount_Amount']:.2f}\n")
else:
    print(f"No data found for Customer ID: {customer_id_input}")
```

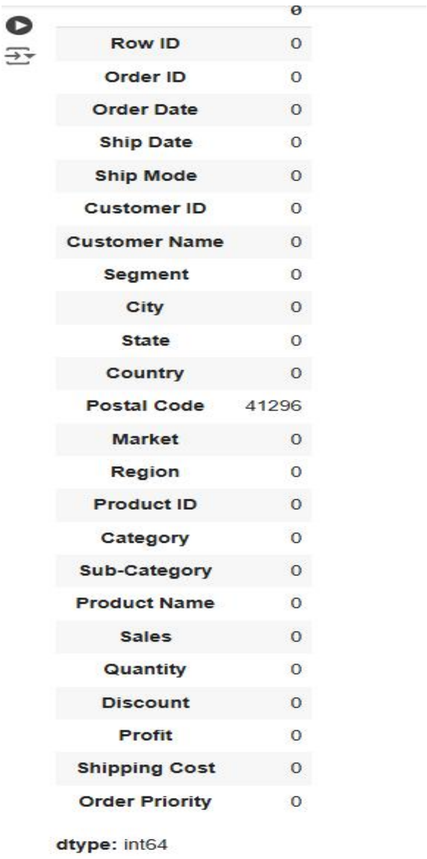
A1.2 SCREENSHOTS



	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	City	State	...	Product ID	Category	Sub-Category	Product Name	Sales	Quantity	Discount	Profit	Shipping Cost
0	32298	CA-2012-124891	2012-07-31	2012-07-31	Same Day	RH-19495	Rick Hansen	Consumer	New York City	New York	...	TEC-AC-10003033	Technology	Accessories	Plantronics CS510 - Over-the-Head monaural Wir...	2309.650	7	0.0	762.1845	933.57
1	26341	IN-2013-77878	2013-02-05	2013-02-07	Second Class	JR-16210	Justin Ritter	Corporate	Wollongong	New South Wales	...	FUR-CH-10003950	Furniture	Chairs	Novimex Executive Leather Armchair, Black	3709.395	9	0.1	-288.7650	923.63
2	25330	IN-2013-71249	2013-10-17	2013-10-18	First Class	CR-12730	Craig Reiter	Consumer	Brisbane	Queensland	...	TEC-PH-10004664	Technology	Phones	Nokia Smart Phone, with Caller ID	5175.171	9	0.1	919.9710	915.49
3	13524	ES-2013-1579342	2013-01-28	2013-01-30	First Class	KM-16375	Katherine Murray	Home Office	Berlin	Berlin	...	TEC-PH-10004583	Technology	Phones	Motorola Smart Phone, Cordless	2892.510	5	0.1	-96.5400	910.16
4	47221	SG-2013-4320	2013-11-05	2013-11-06	Same Day	RH-9495	Rick Hansen	Consumer	Dakar	Dakar	...	TEC-SHA-10000501	Technology	Copiers	Sharp Wireless Fax, High-Speed	2832.960	8	0.0	311.5200	903.04

5 rows × 24 columns

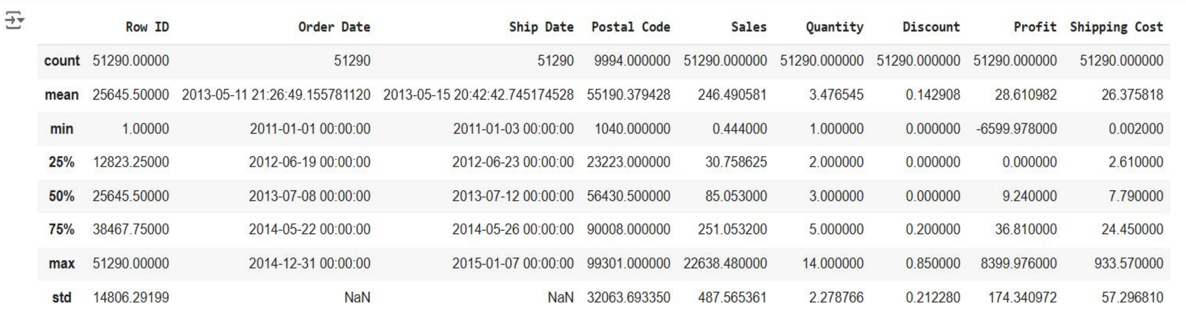
Figure 6: Head of the dataset



	0
Row ID	0
Order ID	0
Order Date	0
Ship Date	0
Ship Mode	0
Customer ID	0
Customer Name	0
Segment	0
City	0
State	0
Country	0
Postal Code	41296
Market	0
Region	0
Product ID	0
Category	0
Sub-Category	0
Product Name	0
Sales	0
Quantity	0
Discount	0
Profit	0
Shipping Cost	0
Order Priority	0

dtype: int64

Figure 7: Sum of missing values in rows



	Row ID	Order Date	Ship Date	Postal Code	Sales	Quantity	Discount	Profit	Shipping Cost
count	51290.00000	51290	51290	9994.000000	51290.000000	51290.000000	51290.000000	51290.000000	51290.000000
mean	25645.50000	2013-05-11 21:26:49.155781120	2013-05-15 20:42:42.745174528	55190.379428	246.490581	3.476545	0.142908	28.610982	26.375818
min	1.00000	2011-01-01 00:00:00	2011-01-03 00:00:00	1040.000000	0.444000	1.000000	0.000000	-6599.978000	0.002000
25%	12823.25000	2012-06-19 00:00:00	2012-06-23 00:00:00	23223.000000	30.758625	2.000000	0.000000	0.000000	2.610000
50%	25645.50000	2013-07-08 00:00:00	2013-07-12 00:00:00	56430.500000	85.053000	3.000000	0.000000	9.240000	7.790000
75%	38467.75000	2014-05-22 00:00:00	2014-05-26 00:00:00	90008.000000	251.053200	5.000000	0.200000	36.810000	24.450000
max	51290.00000	2014-12-31 00:00:00	2015-01-07 00:00:00	99301.000000	22638.480000	14.000000	0.850000	8399.976000	933.570000
std	14806.29199	NaN	NaN	32063.693350	487.565361	2.278766	0.212280	174.340972	57.296810

Figure 8: Summary of the data

```
<Axes: >
```

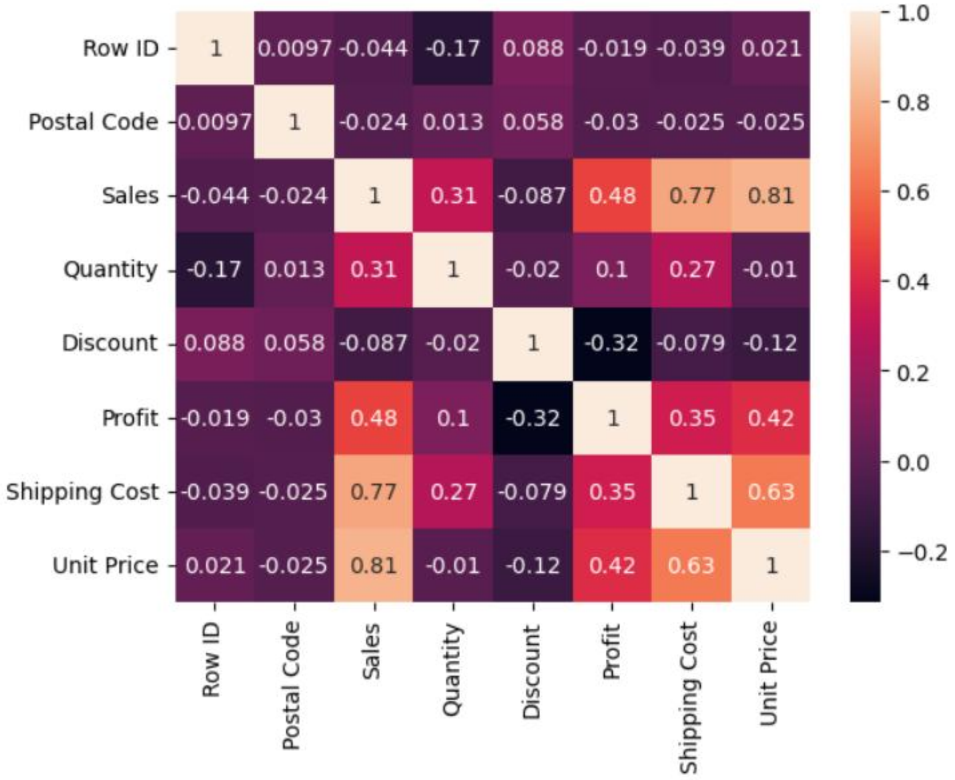
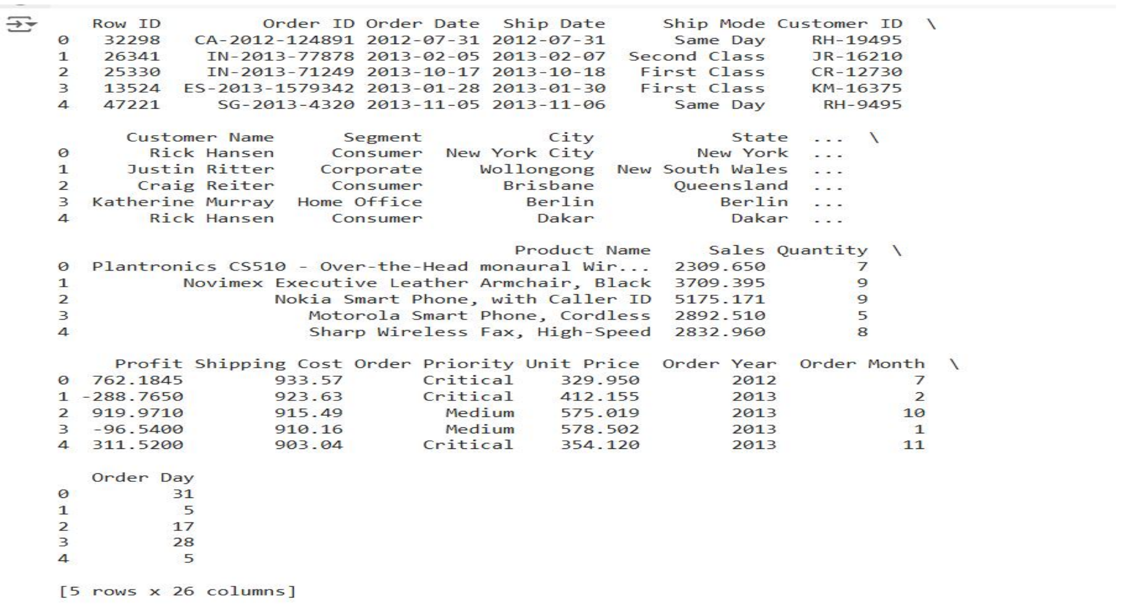
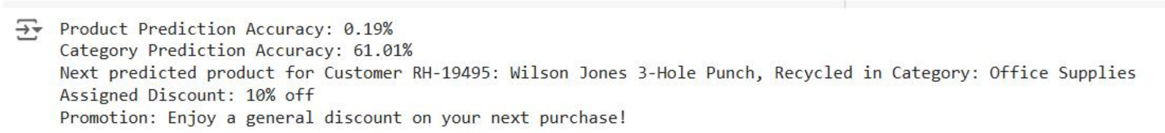


Figure 9: Correalation Heatmap for distribution of features



```
   Row ID         Order ID Order Date  Ship Date     Ship Mode Customer ID  \
0   32298   CA-2012-124891 2012-07-31 2012-07-31      Same Day    RH-19495   
1   26341    IN-2013-77878 2013-02-05 2013-02-07  Second Class    JR-16210   
2   25330    IN-2013-71249 2013-10-17 2013-10-18   First Class    CR-12730   
3   13524  ES-2013-1579342 2013-01-28 2013-01-30   First Class    KM-16375   
4   47221     SG-2013-4320 2013-11-05 2013-11-06      Same Day     RH-9495   

      Customer Name      Segment           City            State  ...  \
0       Rick Hansen     Consumer  New York City         New York  ...   
1     Justin Ritter    Corporate     Wollongong  New South Wales  ...   
2      Craig Reiter     Consumer       Brisbane       Queensland  ...   
3  Katherine Murray  Home Office         Berlin           Berlin  ...   
4       Rick Hansen     Consumer          Dakar            Dakar  ...   

                                    Product Name     Sales Quantity  \
0  Plantronics CS510 - Over-the-Head monaural Wir...  2309.650        7   
1          Novimex Executive Leather Armchair, Black  3709.395        9   
2              Nokia Smart Phone, with Caller ID  5175.171        9   
3                 Motorola Smart Phone, Cordless  2892.510        5   
4                 Sharp Wireless Fax, High-Speed  2832.960        8   

     Profit Shipping Cost Order Priority Unit Price  Order Year  Order Month  \
0  762.1845        933.57       Critical    329.950        2012            7   
1 -288.7650        923.63       Critical    412.155        2013            2   
2  919.9710        915.49         Medium    575.019        2013           10   
3  -96.5400        910.16         Medium    578.502        2013            1   
4  311.5200        903.04       Critical    354.120        2013           11   

   Order Day  
0         31  
1          5  
2         17  
3         28  
4          5  

[5 rows x 26 columns]
```

Figure 10: Feature Extraction (Order day,Month,Year)



```
Product Prediction Accuracy: 0.19%
Category Prediction Accuracy: 61.01%
Next predicted product for Customer RH-19495: Wilson Jones 3-Hole Punch, Recycled in Category: Office Supplies
Assigned Discount: 10% off
Promotion: Enjoy a general discount on your next purchase!
```

Figure 11: Next purchase prediction

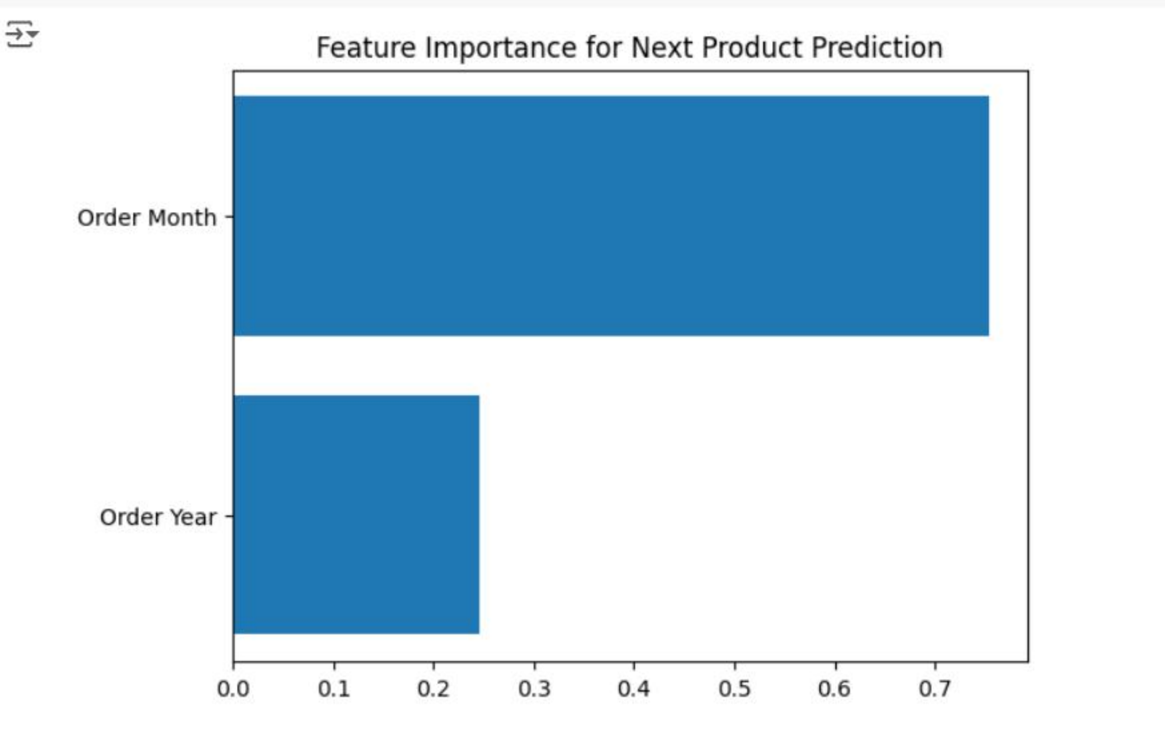
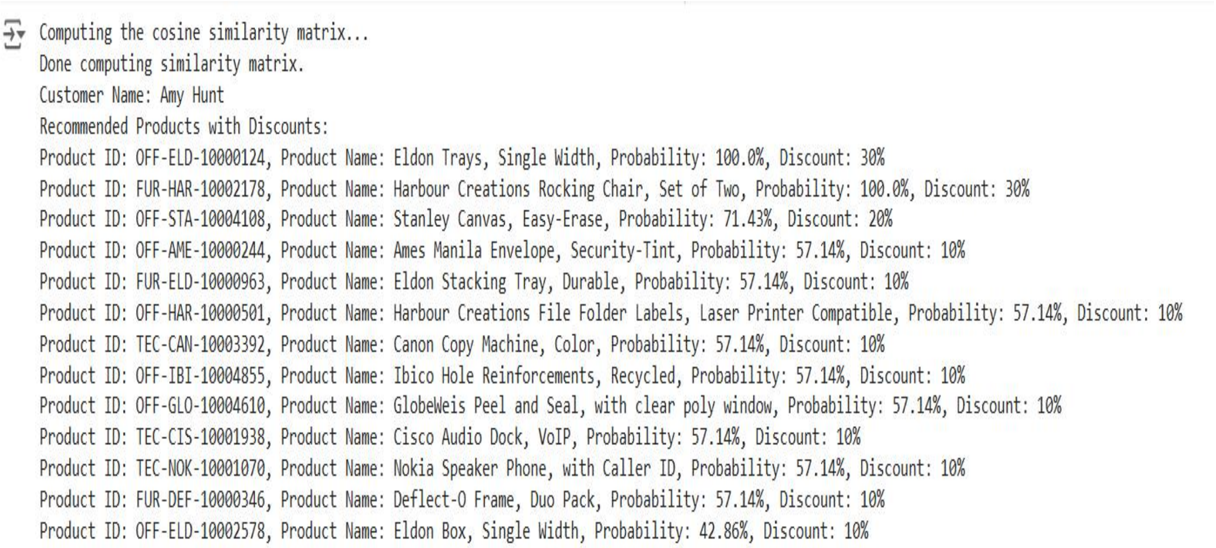
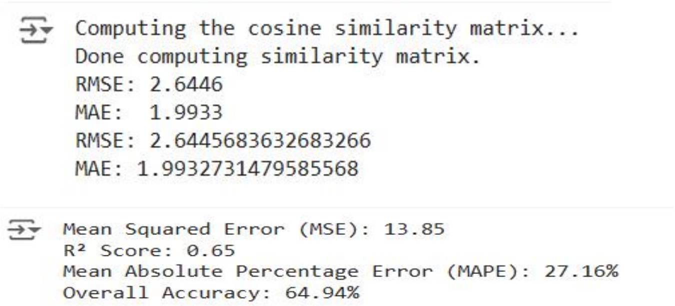


Figure 12: Feature importance for Next purchase prediction



```
Computing the cosine similarity matrix...
Done computing similarity matrix.
Customer Name: Amy Hunt
Recommended Products with Discounts:
Product ID: OFF-ELD-10000124, Product Name: Eldon Trays, Single Width, Probability: 100.0%, Discount: 30%
Product ID: FUR-HAR-10002178, Product Name: Harbour Creations Rocking Chair, Set of Two, Probability: 100.0%, Discount: 30%
Product ID: OFF-STA-10004108, Product Name: Stanley Canvas, Easy-Erase, Probability: 71.43%, Discount: 20%
Product ID: OFF-AME-10000244, Product Name: Ames Manila Envelope, Security-Tint, Probability: 57.14%, Discount: 10%
Product ID: FUR-ELD-10000963, Product Name: Eldon Stacking Tray, Durable, Probability: 57.14%, Discount: 10%
Product ID: OFF-HAR-10000501, Product Name: Harbour Creations File Folder Labels, Laser Printer Compatible, Probability: 57.14%, Discount: 10%
Product ID: TEC-CAN-10003392, Product Name: Canon Copy Machine, Color, Probability: 57.14%, Discount: 10%
Product ID: OFF-IBI-10004855, Product Name: Ibico Hole Reinforcements, Recycled, Probability: 57.14%, Discount: 10%
Product ID: OFF-GLO-10004610, Product Name: GlobeWeis Peel and Seal, with clear poly window, Probability: 57.14%, Discount: 10%
Product ID: TEC-CIS-10001938, Product Name: Cisco Audio Dock, VoIP, Probability: 57.14%, Discount: 10%
Product ID: TEC-NOK-10001070, Product Name: Nokia Speaker Phone, with Caller ID, Probability: 57.14%, Discount: 10%
Product ID: FUR-DEF-10000346, Product Name: Deflect-O Frame, Duo Pack, Probability: 57.14%, Discount: 10%
Product ID: OFF-ELD-10002578, Product Name: Eldon Box, Single Width, Probability: 42.86%, Discount: 10%
```

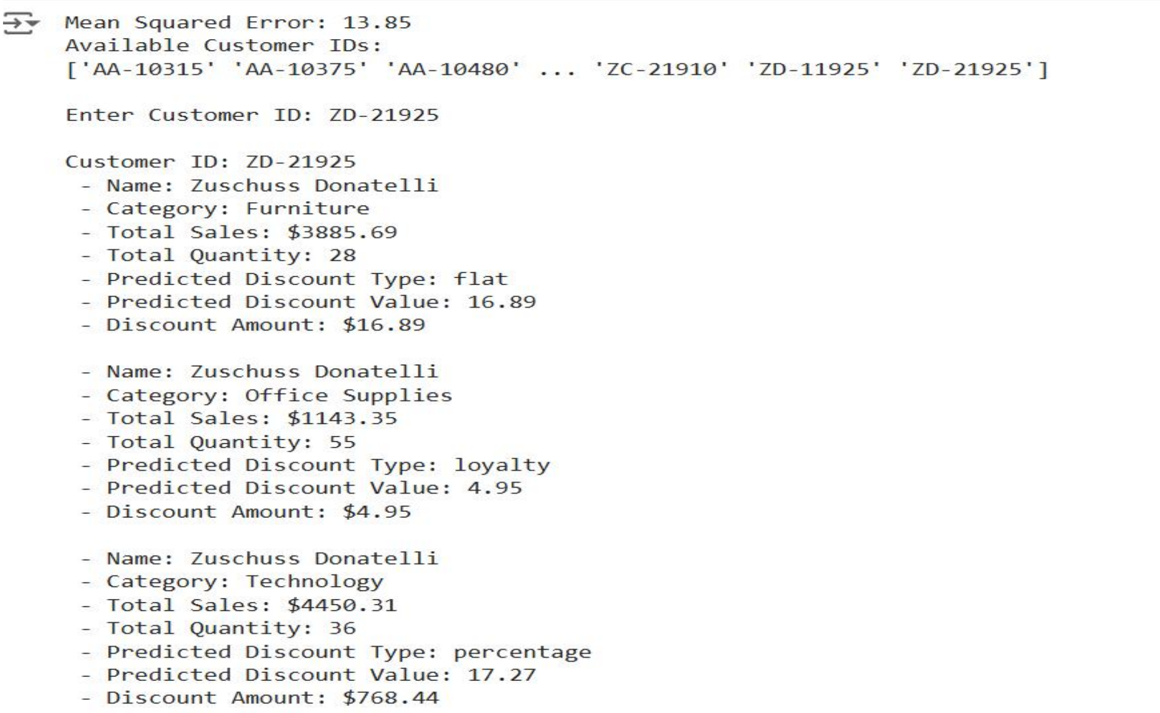
Figure 13: Recommended products with discounts



```
Computing the cosine similarity matrix...
Done computing similarity matrix.
RMSE: 2.6446
MAE:  1.9933
RMSE: 2.6445683632683266
MAE: 1.9932731479585568
```

```
Mean Squared Error (MSE): 13.85
R² Score: 0.65
Mean Absolute Percentage Error (MAPE): 27.16%
Overall Accuracy: 64.94%
```

Figure 14: Evaluation Metrics for Collaborative Filtering



```
Mean Squared Error: 13.85
Available Customer IDs:
['AA-10315' 'AA-10375' 'AA-10480' ... 'ZC-21910' 'ZD-11925' 'ZD-21925']

Enter Customer ID: ZD-21925

Customer ID: ZD-21925
  - Name: Zuschuss Donatelli
  - Category: Furniture
  - Total Sales: $3885.69
  - Total Quantity: 28
  - Predicted Discount Type: flat
  - Predicted Discount Value: 16.89
  - Discount Amount: $16.89

  - Name: Zuschuss Donatelli
  - Category: Office Supplies
  - Total Sales: $1143.35
  - Total Quantity: 55
  - Predicted Discount Type: loyalty
  - Predicted Discount Value: 4.95
  - Discount Amount: $4.95

  - Name: Zuschuss Donatelli
  - Category: Technology
  - Total Sales: $4450.31
  - Total Quantity: 36
  - Predicted Discount Type: percentage
  - Predicted Discount Value: 17.27
  - Discount Amount: $768.44
```

Figure 15: Promotions and Discounts based on their purchase

REFERENCES

[1] Pratama, B. Y., Budi, I., & Yuliawati, A. (2020). Product recommendation in offline retail industry by using collaborative filtering. International Journal of Advanced Computer Science and Applications, 11(9), 635-643.

[2] G. Mustafa et al., "OntoCommerce: Incorporating Ontology and Sequential Pattern Mining for Personalized E-Commerce Recommendations, (2024)" in IEEE Access, vol. 12, pp. 42329-42342, doi: 10.1109/ACCESS.2024.3377120.

[3] Kasem, Mahmoud SalahEldin, Mohamed Hamada and Islam Taj-Eddin. "Customer profiling, segmentation, and sales prediction using AI in direct marketing." Neural Computing and Applications (2023): 1-11.

[4] M. C. Cohen, N. H. Z. Leung, K. Panchamgam, G. Perakis, and A. Smith (2017), "The impact of linear optimization on promotion planning," Operations Research, vol. 65, no. 2, pp. 446–468. http://dx.doi.org/10.1287/opre.2016.1573

[5] R. Fildes, P. Goodwin, and D. Önkal (2019)"Use and misuse of information in supply chain forecasting of promotion effects," International Journal of Forecasting, vol. 35, no. 1, pp. 144–156. http://dx.doi.org/10.1016/j.ijforecast.2017.12.006

[6] K. H. Van Donselaar, J. Peters, A. De Jong, and R. Broekmeulen (2016), "Analysis and forecasting of demand during promotions for perishable items," International Journal of Production Economics, vol. 172, pp. 65–75. http://dx.doi.org/10.1016/j.ijpe.2015.10.022

[7] J. R. Trapero, N. Kourentzes, and R. Fildes (2015), "On the identification of sales forecasting models in the presence of promotions," Journal of the Operational Research Society, vol. 66, no. 2, pp. 299–307. http://dx.doi.org/10.1057/jors.2013.174

[8] J. Henzel and M. Sikora, (2020), "Gradient Boosting Application in Forecasting of Performance Indicators Values for Measuring the Efficiency of Promotions in FMCG Retail," 2020 15th Conference on Computer Science and

Information Systems (FedCSIS), Sofia, Bulgaria, pp. 59-68, doi: 10.15439/2020F118.

[9] Dr.Sunil Bhutada, Dr.V.Saravana Kumar et al., (2019), “DATA ANALYTICS USED IN BOOSTING THE RETAIL BUSINESS”, International Journal of Advanced Science and Technology, Vol 29, no 5, pp. 2776-2790, April 2019.