Title: Increasing Sales and Customer Satisfaction of Superstore using Predictive Business Analytics methods

Abstract

The retail industry is continuously evolving, driven by technological advancements and the increasing availability of data. Integration and analysis of extensive customer data to understand buying patterns and behaviors as well as predicting market trends is crucial for businesses in this sector to maintain a competitive edge.

This project explores the application of predictive business analytics in the case study of Superstore which sells household items. The goal is to gain actionable insights that can drive strategic decision-making to increase sales and customer satisfaction. Key techniques used include Market Basket Analysis to identify frequently co-purchased items, Recommender Systems to enhance the shopping experience of Superstore's customers through personalized product suggestions, and Regression Analysis to forecast sales trends and customer demand.

These methodologies are aimed at addressing common challenges faced by retailers, such as optimizing inventory levels, improving sales and marketing strategies, and enhancing customer satisfaction.

The primary objectives of this project are to:

- **Understand Customer Purchasing Behavior using Market Basket Analysis**: To Analyze transaction data to reveal patterns in customer purchases, enabling more effective cross-selling, product placements, bundling and promotions.
- **Enhance Shopping Experience using Recommender System**: To develop personalized product recommendations to improve customer engagement and satisfaction.
- **Forecast Sales Trends using Regression Models**: To predict future sales, aiding in better inventory management and demand planning.

By demonstrating the practical applications of predictive analytics, this project aims to show how data-driven insights can lead to more informed decisions and better business outcomes. The findings provide a framework that other retailers can adopt to enhance their operations and customer engagement, to help them achieve a competitive advantage in the retail sector.

1.0 PROJECT BACKGROUND

1.1 Background of the problem domain

Introduction to the Retail Industry

The retail industry is a cornerstone of the global economy, providing goods and services to consumers and generating significant employment and revenue. It includes a wide range of business models, from traditional stores to modern e-commerce platforms. The industry is characterized by its fast pace, high competition, and constant evolution driven by changing consumer behaviors and technological advancements (Hameli, 2018).

Evolution of Retail

The retail sector has undergone several transformations over the past few decades. The advent of digital technologies and the internet has revolutionized how consumers shop and interact with brands. E-commerce has become a dominant force, with platforms like Amazon, Alibaba, and eBay leading the charge. This shift has forced traditional retailers to adopt new strategies and embrace digital transformation to stay competitive (Gauri et al., 2021).

The Rise of Big Data in Retail

With the rise of digital technologies, the amount of data generated by retail transactions has exploded. This data includes information on customer demographics, purchasing patterns, inventory levels, supply chain logistics, and more. Retailers have access to data from various sources (Varma & Ray, 2023), such as:

- **Point-of-Sale Systems**: Capture transaction details at the checkout counter.
- **Customer Loyalty Programs**: Track customer purchases and preferences.
- **E-commerce Platforms**: Provide insights into online shopping behavior.
- **Social Media**: Offer valuable feedback on customer sentiment and brand perception.

Importance of Data Analytics in Retail

In this digital age, data has emerged as a crucial strategic asset for retailers. It provides insights into consumer preferences, purchasing patterns, and market trends. Retailers collect data from various sources, including point-of-sale (POS) systems, customer loyalty programs, e-commerce platforms, and social media. This data, when properly analyzed, can help retailers make informed decisions, optimize their operations, and deliver personalized experiences to customers (Varma & Ray, 2023).

The Superstore Case Study

In this project, we focus on a case study involving a retail business selling household items called Superstore. (Data source is taken from Kaggle at the following link: https://www.kaggle.com/datasets/ishanshrivastava28/superstore-sales/data).

Superstore collects extensive data on customer transactions, including information on products purchased, quantities, prices, and customer demographics. By analyzing this data, we aim to uncover patterns and trends that can inform strategic decisions. Specifically, we will use Market Basket Analysis to understand purchasing behavior, develop Recommender Systems to enhance the shopping experience, and apply Regression Analysis to forecast sales trends.

1.2 Challenges and Problem Statement

The superstore sells close to 2000 different products and continues to expand its variety to its customers. However, the growing selection of products can overwhelm customers and make it difficult for them to make purchase decisions. Furthermore, with the growth of technology, nowadays, customers can buy a variety of products from the Superstore's competitors from the comfort of their homes by simply clicking the "Buy" button on an electronic device such as a phone or computer therefore increasing the competitiveness for Superstore.

At the moment, Superstore has a huge amount of historical customer transactions data that have not been tapped and Superstore does not have any method to predict customer behaviour and understand which bundles of products would likely sell more. Furthermore, the Superstore does not know what its customers' preferences are and which segments of their customer profiles are similar. The superstore also does not know what are the factors that lead to customer satisfaction and drive sales.

1.3 Objectives of the project

Figure 1: Predictive Analytics Use Cases in Retail

The primary objective of this project is to leverage predictive business analytics to provide actionable insights that can drive strategic decision-making for a retail superstore. By addressing the key challenges identified in the problem statement, the project aims to enhance various aspects of the retail operation, from understanding customer behavior to optimizing inventory management. The specific objectives of the project are outlined below:

1. Market Basket Analysis

- **Objective**: To understand customer purchasing behavior by identifying frequently co-purchased items to optimize product placements and promotions by analyze customer transaction data to uncover patterns in customer purchases, enabling more effective product placements and promotions.
- **Approach**: Utilize association rule mining techniques such as Apriori or FP-Growth algorithms to uncover relationships between products in transaction data. Analyze the results

to provide recommendations for product bundling and cross-selling strategies (Patwary et al., 2021).

2. Develop Recommender Systems

- **Objective**: To enhance the customer shopping experience through personalized product recommendations.

- **Approach**: Implement collaborative filtering, content-based filtering, and hybrid recommendation models to provide tailored product suggestions based on customer purchase history and preferences. Evaluate the performance of these models to select the most effective approach (Jannach et al., 2021).

3. Perform Regression Analysis

- **Objective**: To forecast sales trends and customer demand, aiding in better inventory management and demand planning.
- **Approach**: Apply various regression techniques, including linear regression, polynomial regression, and time series forecasting, to predict future sales. Use these predictions to inform inventory restocking and promotional strategies (Vukovic et al., 2023)

By achieving these objectives, the project aims to demonstrate the practical applications of predictive analytics in the retail industry. The insights derived from the analysis will help the retail superstore optimize its operations, enhance the customer shopping experience, and drive better business outcomes. This project will also provide a framework that other retailers can adopt to improve their data-driven decision-making processes.

2.0 LITERATURE REVIEW

Predictive analytics has emerged as a powerful tool in the retail industry, allowing businesses to make informed decisions based on data-driven insights. This literature review examines the current state of predictive analytics in retail, focusing on the methodologies and approaches used to enhance various aspects of retail operations. The review includes key findings from accessible peer-reviewed studies, highlighting the significant impact of predictive analytics on the retail sector.

2.1 Predictive Analytics in Retail

Predictive analytics involves using statistical methods and machine learning algorithms to analyze historical data and predict future trends. In the retail industry, these predictions help address several critical areas, including demand forecasting, customer segmentation, inventory management, and personalized marketing.

- **Demand Forecasting**: Predictive analytics models are extensively used for demand forecasting, which involves predicting future product demand based on historical sales data, market trends, and customer behavior. Accurate demand forecasting helps retailers optimize inventory levels, reduce costs, and improve customer satisfaction. For example, a comprehensive review in the Journal of Big Data discusses various supervised and unsupervised learning techniques used in demand forecasting, such as regression, neural networks, and decision trees (Seyedan & Mafakheri, 2020).
- **Customer Segmentation and Personalization**: Retailers use predictive analytics to segment customers based on purchasing behavior and preferences. This segmentation enables personalized marketing campaigns and product recommendations, enhancing customer satisfaction and loyalty. A study in IGI Global emphasizes the role of predictive models in creating personalized shopping experiences, leading to increased customer engagement and sales (Dahake et al., 2024).
- **Market Basket Analysis**: This technique analyzes customer transaction data to identify associations between products frequently purchased together. Understanding these associations helps retailers optimize product placements, design effective cross-selling and upselling strategies, and improve promotional campaigns. EffectiveSoft provides a detailed analysis of how market basket analysis can be utilized to enhance sales strategies.
- **Churn Prediction**: Predictive analytics models can identify customers at risk of churn by analyzing their purchasing patterns and engagement levels. Retailers can use these insights to implement retention strategies, such as personalized offers and targeted communication, to reduce customer attrition. EffectiveSoft highlights the importance of churn prediction in maintaining customer loyalty.

2.2 Approaches and Methodologies

Predictive analytics encompasses various methodologies and techniques that can significantly enhance retail operations. This section focuses on three key approaches: Market Basket Analysis, Recommender Systems, and Regression Analysis, all of which are integral to understanding and predicting customer behavior and improving sales strategies.

Market Basket Analysis

Market Basket Analysis is a data mining technique used to discover associations between items in a transaction dataset. It helps retailers understand customer purchasing patterns, optimize product placements, and develop effective cross-selling strategies. MBA involves identifying frequent itemsets and generating association rules using algorithms like Apriori, and FP-Growth.

- **Apriori Algorithm**: This algorithm identifies frequent itemsets by iteratively combining items that appear together in transactions and pruning those that do not meet a minimum support threshold. A study by Kawale and Dahima (2018) demonstrated the effectiveness of the Apriori algorithm in analyzing customer behavior and providing insights for strategic decision-making in retail.
- **FP-Growth Algorithm**: Unlike Apriori, FP-Growth uses a divide-and-conquer approach to find frequent itemsets without candidate generation, making it more efficient for large datasets. Research by Mai Shawkat (2021) highlighted the efficiency of the FP-Growth algorithm in processing large-scale retail data to uncover valuable patterns.

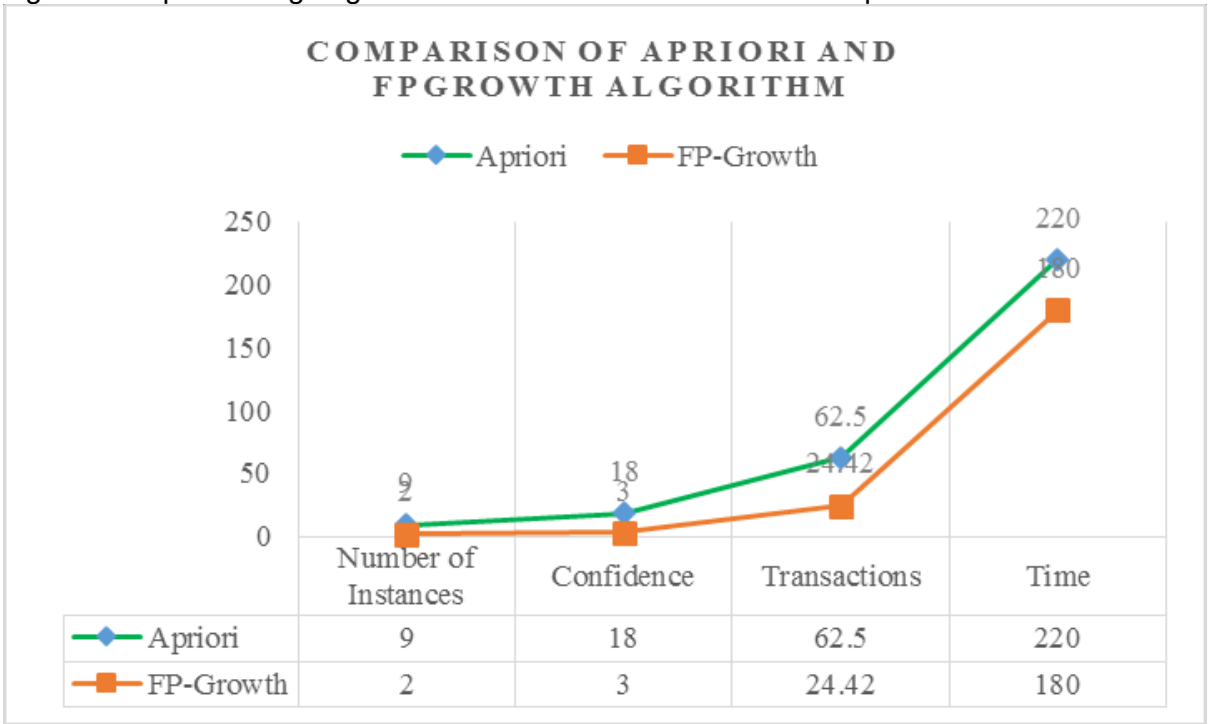

Figure 2.1: Comparison of Apriori and FP-Growth Algorithm

Recommender Systems

Recommender systems are designed to provide personalized product recommendations to customers based on their past purchases and browsing behavior. These systems enhance the shopping experience, increase customer engagement, and boost sales.

- **Collaborative Filtering**: This approach makes recommendations based on the preferences of similar users. It can be implemented using user-based or item-based techniques. Collaborative filtering has been widely used in e-commerce platforms like Amazon, where it helps in suggesting products that similar users have bought.
- **Content-Based Filtering**: This method recommends items similar to those a user has shown interest in, based on product attributes. It is particularly useful in niche markets where user data is sparse. Studies have shown that content-based filtering can effectively complement collaborative filtering by providing recommendations when user data is limited.
- **Hybrid Models**: Combining collaborative and content-based filtering, hybrid models leverage the strengths of both approaches to improve recommendation accuracy. Research by EffectiveSoft (n.d.) demonstrated the effectiveness of hybrid recommender systems in providing more accurate and diverse recommendations, leading to higher customer satisfaction.

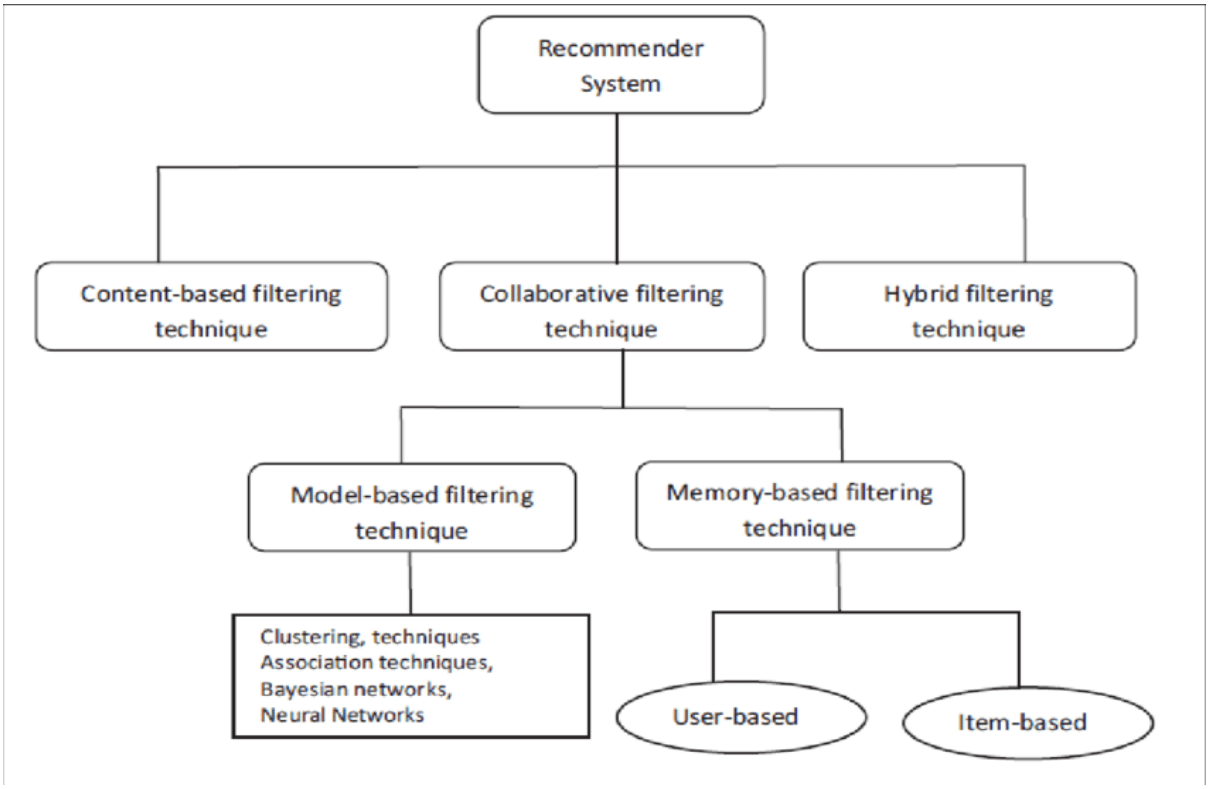

Figure 3.2: Recommender Systems

Regression Analysis

Regression analysis is a statistical method used to model the relationship between a dependent variable (e.g., sales) and one or more independent variables (e.g., price, seasonality). It is crucial for demand forecasting and sales prediction in retail.

- **Linear Regression**: This method models the linear relationship between variables, making it useful for straightforward demand forecasting scenarios. It helps identify trends and predict future sales based on historical data.
- **Multiple Regression**: Extending linear regression, this technique considers multiple independent variables, providing a more comprehensive model of sales influenced by various factors. It is particularly useful in complex retail environments where multiple factors affect sales.
- **Time Series Analysis**: Techniques like ARIMA and exponential smoothing are used to forecast future sales by analyzing past sales data. These models capture trends, seasonality, and cyclic patterns in sales, making them highly effective for long-term forecasting. A comprehensive review by the Journal of Big Data (2020) emphasized the importance of time series analysis in enhancing the accuracy of demand forecasts in retail.

2.3 Key Findings from Peer-Reviewed Studies

Enhanced Decision-Making

Predictive analytics significantly enhances decision-making in the retail industry by providing data-driven insights into potential risks and outcomes. Retailers can leverage these insights to make more informed decisions about inventory management, marketing strategies, and customer engagement. For instance, a study in the Journal of Big Data highlights how predictive models, such as neural networks and decision trees, can improve the accuracy of

demand forecasting, leading to better inventory planning and reduced stockouts. Similarly, IGI Global's comprehensive review emphasizes the role of predictive analytics in shaping strategic decisions, particularly in areas like personalized marketing and customer segmentation.

Reduced Risks

By accurately forecasting outcomes, predictive analytics helps retailers mitigate risks associated with understocking and overstocking. This proactive approach enables the development of effective risk management strategies. The Journal of Big Data provides evidence that advanced machine learning algorithms can significantly enhance the precision of sales forecasts, thereby reducing the risks related to inventory mismanagement. EffectiveSoft's case studies demonstrate how retailers can use predictive analytics to identify and address potential issues before they escalate, further minimizing operational risks.

Improved Customer Experience

Personalization powered by predictive analytics significantly enhances the customer shopping experience. Retailers can deliver tailored product recommendations and personalized marketing messages, leading to increased customer satisfaction and loyalty. IGI Global's review discusses how recommender systems, particularly hybrid models that combine collaborative and content-based filtering, can offer more accurate and diverse product suggestions, thereby improving the overall shopping experience. Additionally, a study by Kawale and Dahima (2018) shows how market basket analysis can provide valuable insights into customer purchasing behavior, helping retailers create more effective cross-selling and upselling strategies.

Increased Sales and Revenue

The application of predictive analytics in retail not only enhances customer experience but also drives sales and revenue growth. Market basket analysis helps retailers identify frequently co-purchased items, which can inform product placement strategies and promotional campaigns. Research by Shawkat et al. (2022) on the FP-Growth algorithm demonstrates its effectiveness in uncovering hidden patterns in large retail datasets, leading to more targeted marketing efforts and increased sales. EffectiveSoft also reported significant improvements in sales performance when retailers use predictive models to optimize their marketing and sales strategies.

Real-World Applications

Several real-world case studies illustrate the practical benefits of predictive analytics in retail. For example, Walmart uses advanced machine learning algorithms to analyze historical sales data, enabling precise demand forecasts that optimize inventory management and reduce stockouts. Amazon employs sophisticated neural network models to predict demand for its extensive product catalog, enhancing its ability to meet customer needs promptly. These examples underscore the transformative potential of predictive analytics in driving operational efficiency and business success in the retail sector.

3.0 METHODOLOGY

3.1 Project Framework

CRISP-DM is the most widely used data mining process model today, which guiding us through the entire phases of planning, organizing, and implementing data mining project. Figure shows a brief description of six-step CRISP-DM process and Table provides detailed information about the overall framework of the project.

Figure 3.1: The Six-step CRISP-DM Process

Table 3.1: Detail Information About the Project Framework

Steps	Content
Understanding the business	Understand project goals and requirements and develop data mining objectives from a business perspective.
Understanding the data	Initial data is collected, and data exploration is used to identify the quality and characteristics of the data.
Preparation of data	The data is cleaned, transformed, selected and formatted to make it suitable for model building after the source is fully identified.
Modelling	Select and apply appropriate data modeling techniques and adjust parameters to validate model quality
Evaluation	Evaluate the effectiveness of the model in the context of business intent to ensure that it is effective in meeting the business needs and that new models, objectives may emerge
Setting out	Deploy the model to a production environment, monitor and maintain it, and present it to stakeholders in a usable way

3.2 Dataset and Preparation

The dataset used for this analysis is taken from Kaggle at the following link: https://www.kaggle.com/datasets/ishanshrivastava28/superstore-sales/data. This dataset represents the Superstore's sales records and contains detailed information about individual transactions. It includes 9994 rows and 21 columns, each representing various aspects of the sales process. Below is a detailed description of each column in the dataset:

- **Row ID**: An integer that uniquely identifies each row in the dataset. This is primarily used for indexing and does not contain any meaningful information related to the transactions.
- **Order ID**: A unique identifier for each order. This string field allows tracking of multiple items purchased in a single transaction.
- **Order Date**: The date on which the order was placed. This field is crucial for time-based analysis, such as identifying trends over time or analyzing seasonal variations in sales. It is formatted as a string.
- **Ship Date**: The date on which the order was shipped to the customer. Similar to the order date, this field can be used to analyze shipping delays and logistics efficiency. It is formatted as a string.
- **Ship Mode**: Describes the mode of shipping chosen for the order. Categories include "Standard Class," "Second Class," "First Class," and "Same Day." This field helps in analyzing customer preferences and shipping performance.
- **Customer ID**: A unique identifier for each customer. This field allows for tracking customer purchasing behavior and loyalty.
- **Customer Name**: The name of the customer. While this field is less useful for quantitative analysis, it can be important for qualitative insights and customer relationship management.
- **Segment**: The market segment to which the customer belongs. Categories include "Consumer," "Corporate," and "Home Office." This field is useful for segmenting the analysis based on customer types.
- **Country**: The country where the customer is located. Since the dataset only contains "United States," this field has limited variability.
- **City**: The city where the customer is located. This field is useful for geographic analysis of sales.
- **State**: The state where the customer is located. Like the city field, it aids in geographic segmentation.
- **Postal Code**: The postal code of the customer's location. This field can be used for more granular geographic analysis.
- **Region**: The region of the customer within the United States. Categories include "East," "West," "Central," and "South." This field helps in regional analysis.
- **Product ID**: A unique identifier for each product. This field is essential for tracking product-level sales and inventory.
- **Category**: The high-level category of the product. Categories include "Furniture," "Office Supplies," and "Technology." This field helps in analyzing sales by product category.
- **Sub-Category**: A more detailed classification within each product category. This field allows for more detailed product-level analysis.
- **Product Name**: The name of the product. While this field is less useful for quantitative analysis, it can be important for inventory management and qualitative insights.
- **Sales**: The sales amount for the transaction. This numerical field is critical for revenue analysis and forecasting.
- **Quantity**: The number of units of the product sold. This numerical field helps in analyzing sales volume.
- **Discount**: The discount applied to the product. This numerical field is important for understanding pricing strategies and their impact on sales.

- **Profit**: The profit earned from the transaction. This numerical field is essential for profitability analysis and financial performance assessment.

This dataset provides a rich source of information for analyzing various aspects of retail sales, including customer behavior, product performance, and geographic trends. By leveraging this data, we can gain valuable insights into the business operations and identify opportunities for improvement and growth. The dataset's comprehensive nature allows for detailed and multifaceted analysis, making it a powerful tool for predictive business analytics.

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	...	Postal Code	Region	Product ID	Category	Sub-Category	Prod Na
0	1	CA-2013-152156	09-11-2013	12-11-2013	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-BO-10001798	Furniture	Bookcases	B Somer Collect Bookc
1	2	CA-2013-152156	09-11-2013	12-11-2013	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	...	42420	South	FUR-CH-10000454	Furniture	Chairs	Hon Del Fal Upholste Stack Chair
2	3	CA-2013-138688	13-06-2013	17-06-2013	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	...	90036	West	OFF-LA-10000240	Office Supplies	Labels	S Adhes Addr Labels Typewrit
3	4	US-2012-108966	11-10-2012	18-10-2012	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	33311	South	FUR-TA-10000577	Furniture	Tables	Bretf CR4 Series S Rectangu Ta

Figure 3.1: Dataset Details

3.3 Data Pre-processing

Data pre-processing is important to obtain maximum accuracy from the data analysis model before model building.

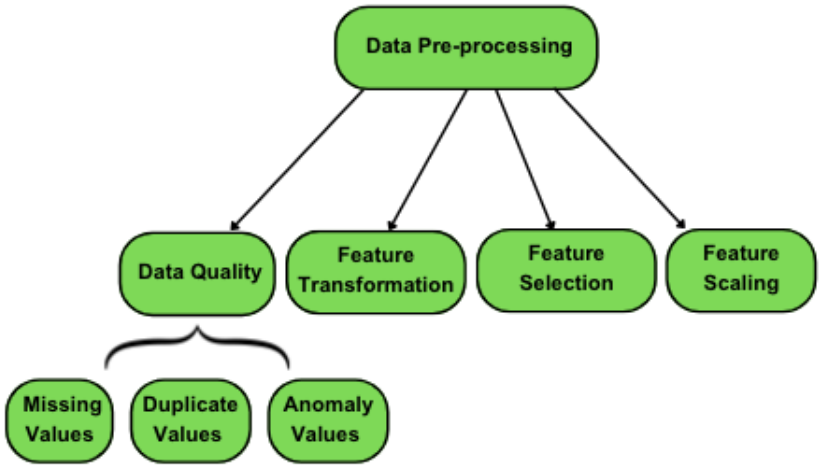

Figure 3.2: Data Pre-processing Flow Chart

The basic structure and information of the dataset were inspected using the `.info () ` method, which provided the following details:

```
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Row ID         9994 non-null   int64  
 1   Order ID       9994 non-null   object 
 2   Order Date     9994 non-null   object 
 3   Ship Date      9994 non-null   object 
 4   Ship Mode      9994 non-null   object 
 5   Customer ID    9994 non-null   object 
 6   Customer Name  9994 non-null   object 
 7   Segment        9994 non-null   object 
 8   Country        9994 non-null   object 
 9   City           9994 non-null   object 
 10  State          9994 non-null   object 
 11  Postal Code    9994 non-null   int64  
 12  Region         9994 non-null   object 
 13  Product ID     9994 non-null   object 
 14  Category       9994 non-null   object 
 15  Sub-Category   9994 non-null   object 
 16  Product Name   9994 non-null   object 
 17  Sales          9994 non-null   float64
 18  Quantity       9994 non-null   int64  
 19  Discount       9994 non-null   float64
 20  Profit         9994 non-null   float64
```

Figure 3.3: Basic Information about the Dataset

3.3.1 Data Quality

No missing values were found in the dataset, as all columns have non-null counts equal to the total number of entries. We will remove any duplicate entries in the dataset to ensure the analysis is based on unique transactions.

```
duplicates = data.duplicated().sum()
print(f'\nTotal duplicate entries: {duplicates}')
```

```
Total duplicate entries: 0
```

Figure 3.4: Check duplicate values

We check for negative values in columns where they shouldn't exist and filter them out. This step involves identifying and correcting any errors in the dataset.

```
# Correcting errors (example: ensuring no negative values in Sales, Quantity, Profit)
data = data[(data['Sales'] >= 0) & (data['Quantity'] >= 0) & (data['Profit'] >= 0)]
```

Figure 3.5: Correcting errors

Figure 3.6 shows the distribution of sales and identifies some significant outliers.

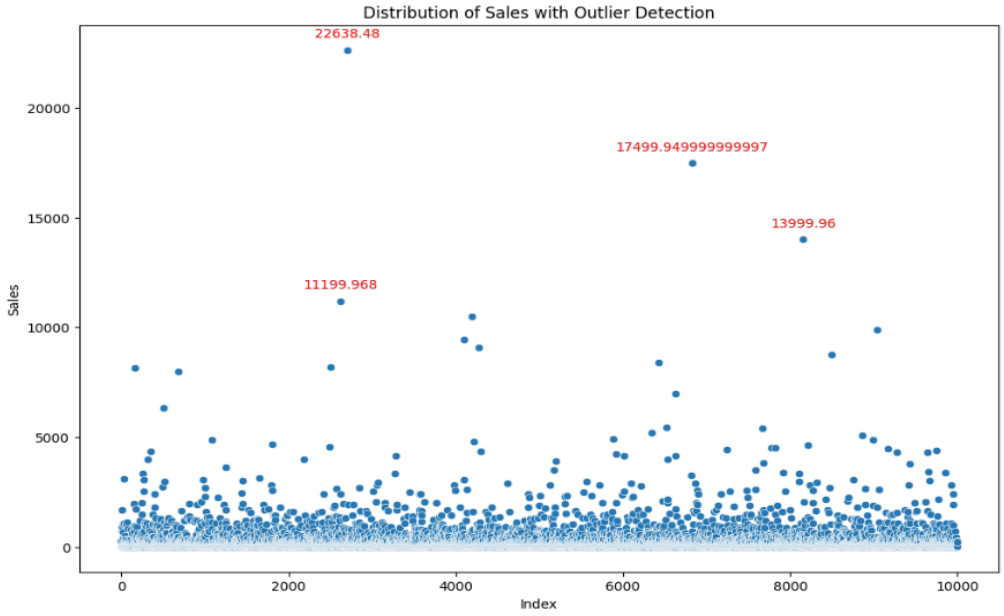

Figure 3.6: Distributed of Sales with Outlier Detection

To further ensure data quality and check for any anomalies, the distributions of `Discount`, `Quantity`, `Profit`, `Operating Expenses`, `Net Profit`, and `Order Day` are visualized.

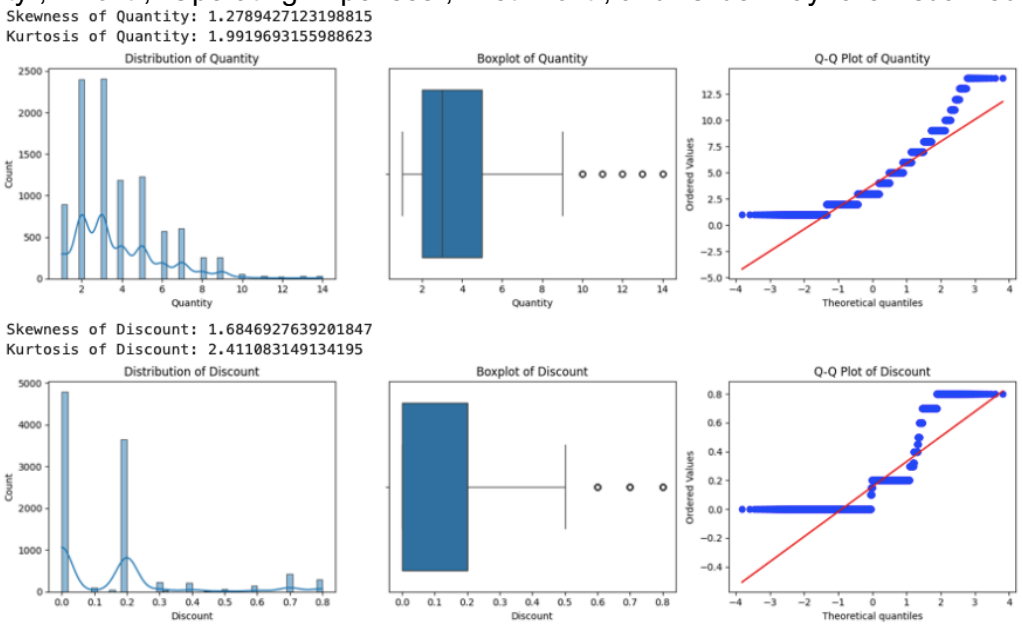

Figure 3.7: Visualizing Quantity and Discount Distributions

The `Quantity` variable is notably right-skewed with high skewness and kurtosis, indicating outliers and a heavy-tailed distribution. The box plot shows multiple outliers, and the Q-Q plot reveals a deviation from normal distribution, suggesting possible data transformation. Similarly, the `Discount` variable exhibits strong right-skewness with high skewness and kurtosis. Both the box plot and Q-Q plot indicate significant non-normal distribution characteristics, necessitating further investigation and potential data transformation.

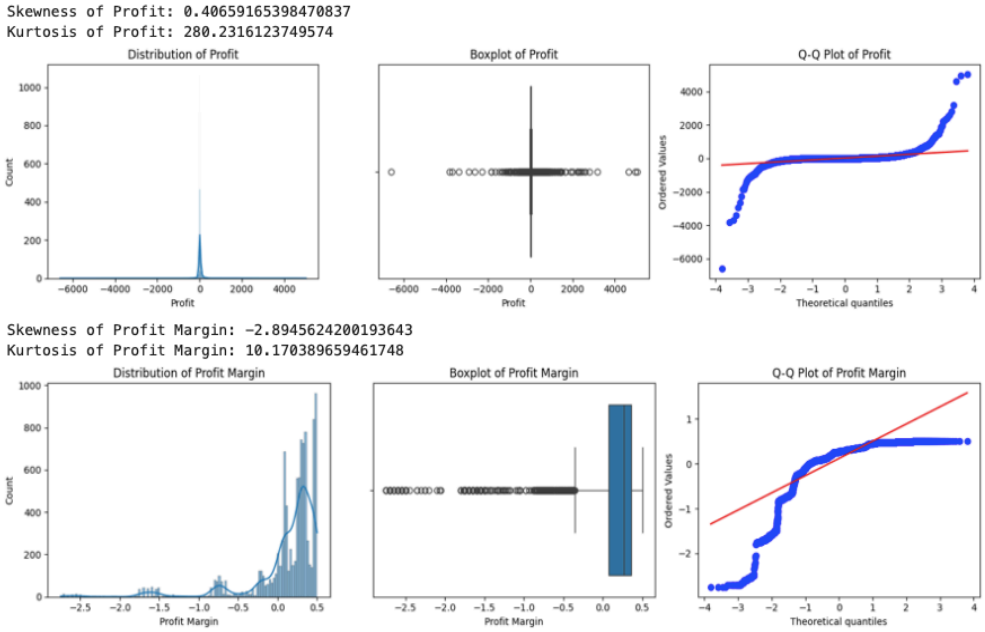

Figure 3.8: Visualizing Profit Distribution

The `Profit` distribution is characterized by bimodal peaks, with more positive and negative values. The high skewness and Kurtosis indicate the strong asymmetry of the distribution. The box plots and Q-Q plots further verify the non-normal characteristics of the profit distribution.

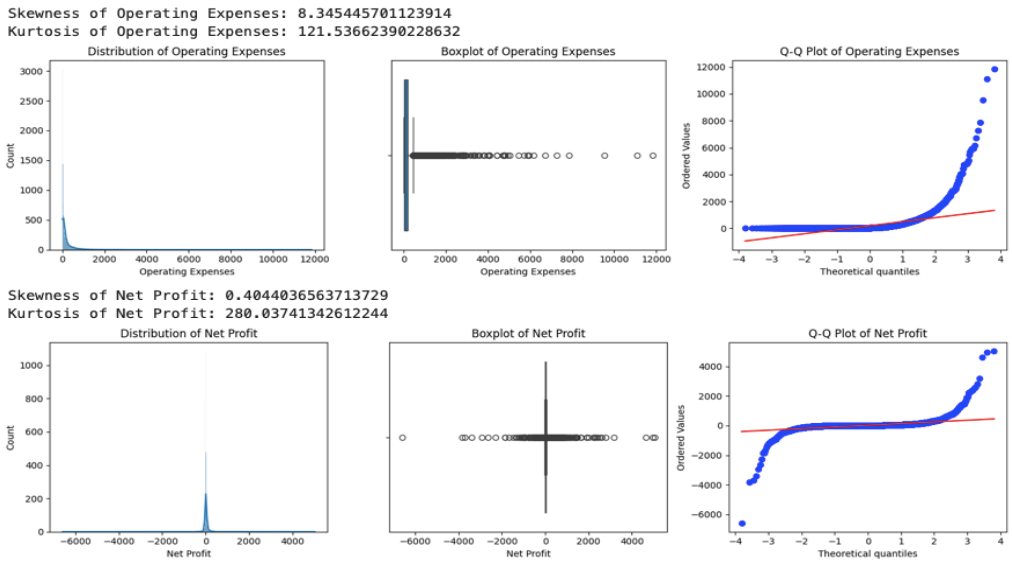

Figure 3.9: Visualizing Operating Expenses and Net Profit Distributions

The distribution characteristics of the variables Operating Expenses and Net Profit are similar to those of the previous variables, with right skewness, high peaks and high skewness.

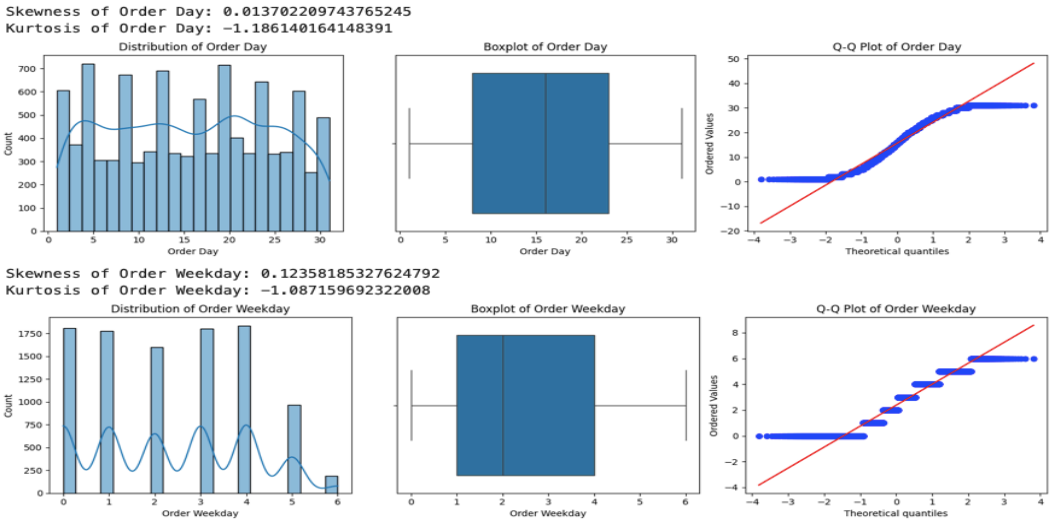

Figure 3.10: Visualizing Order Distributions

Order Day shows a periodic multi-peaked distribution, while Order Weekday is smoother.

3.3.2 Feature Selection

Feature selection is a critical step in machine learning that helps to improve the performance and interpretability of a model. In this report, we present a comprehensive analysis of the dataset through three different feature selection methods Ridge Regression Model Feature Selection, Random Forest Regression Model Feature Selection, and Recursive Feature Elimination (RFE).

Ridge Regression Model Feature Selection: `Discount`, `Profit`, `Operating Expenses`, `Net Profit`.

Random Forest Regression Model Feature Selection: `Profit`, `Discounted Profit`, `Discount Percentage`, `Operating Expenses`, `Net Profit`.

RFE: `Postal Code`, `Quantity`, `Discount`, `Profit`, `Profit Margin`, `Discounted Profit`, `Discount Percentage`, `Operating Expenses`, `Net Profit`, `Order Day`.

Combining the selection results of the three methods, we find that certain features such as 'Profit' and 'Net Profit' are selected in all the methods, which indicates that these features are important for model prediction. The following 11 features were finally selected:

```
final_features = ['Quantity', 'Profit', 'Profit Margin', 'Discounted Profit','Discount',
       'Discount Percentage', 'Operating Expenses', 'Net Profit',
       'Order Day', 'Ship Day', 'Ship Weekday']
```

Figure 3.11: Final Features

3.3.3 Feature Transformation

Converting the `Order Date` and `Ship Date` columns in a DataFrame from `string` format to `datetime` format makes subsequent time series operations more convenient and intuitive.

```
df['Order Date'] = pd.to_datetime(df['Order Date'])
df['Ship Date'] = pd.to_datetime(df['Ship Date'])
```

Figure 3.12: Time series data conversion

Data transformations were performed on columns in the dataset that were considered to have skewed distributions in the expectation that the skewness and kurtosis would be reduced to meet the assumptions of the subsequent analysis and modeling. Based on the transformed skewness and kurtosis metrics, the optimal transformation method for each column is selected in Figure 3.12. We perform logarithmic conversions for `Net Profit`, `Discounted Profit`, `Profit`, `Ship Day`, `Order Day`, square root conversions for `Discount`, `Order Month`, and power conversions for `Quantity`, `Profit Margin`, `Discount Percentage`, `Operating Expenses`, `Order Weekday`.

	Column	Transformer	Before Skewness	Before Kurtosis	After Skewness	After Kurtosis	Score
2	Quantity	power	1.278943	1.991969	0.013357	-0.506032	0.519390
3	Discount	sqrt	1.684693	2.411083	0.455860	-0.988861	1.444721
7	Profit	log1p	0.406592	280.231612	0.359537	0.484427	0.843964
11	Profit Margin	power	-2.894562	10.170390	-0.280131	-0.747027	1.027157
13	Discounted Profit	log1p	13.084781	320.660127	-0.000491	1.214361	1.214852
17	Discount Percentage	power	10.826595	188.805137	0.945853	-0.742109	1.687962
20	Operating Expenses	power	8.345446	121.536624	0.052678	-0.895320	0.947998
22	Net Profit	log1p	0.404404	280.037413	0.407298	0.228207	0.635504
25	Order Day	log1p	0.013702	-1.186140	-0.954409	0.090293	1.044702
29	Order Weekday	power	0.123582	-1.087160	-0.152827	-1.134366	1.287193
30	Order Month	sqrt	-0.432853	-0.984427	-0.855301	-0.144528	0.999829
34	Ship Day	log1p	-0.012122	-1.206964	-1.007131	0.233591	1.240722

Figure 3.13: The best conversion method for each column

To enhance our machine learning model's performance, we apply specific transformations to selected features based on insights from feature selection and Figure 3.13.

```
transformer = ColumnTransformer(transformers=[
    ('log_transform',FunctionTransformer(np.log1p),['Discounted Profit','Net Profit','Profit','Order Day','Ship Day']),
    ('sqrt_transform',FunctionTransformer(np.sqrt),['Discount']),
    ('power_transform',PowerTransformer(),['Quantity','Discount Percentage','Operating Expenses','Profit Margin'])
],remainder='passthrough')

transformer
```

Figure 3.14: Apply transformation

The converted DataFrame `final_X_test` has some missing values. The interpolation process fills in the missing values with the median of each column.

```
print('Train data')
print(final_X_train.isnull().sum()[final_X_train.isnull().sum()>1])
print('Test data')
final_X_test.isnull().sum()[final_X_test.isnull().sum()>1]
```

```
Train data
Quantity         1174
Profit           1307
Profit Margin    1293
dtype: int64
Test data

Quantity         494
Profit           546
Profit Margin    542
dtype: int64
```

```
for col in final_X_train.columns:
    imputer = SimpleImputer(strategy='median')
    final_X_train[col] = imputer.fit_transform(final_X_train[[col]])

for col in final_X_test.columns:
    imputer = SimpleImputer(strategy='median')
    final_X_test[col] = imputer.fit_transform(final_X_test[[col]])
```

```
final_X_train.isnull().sum()
```

```
Quantity              0
Profit                0
Profit Margin         0
Discounted Profit     0
Discount              0
Discount Percentage   0
Operating Expenses    0
Net Profit            0
Order Day             0
Ship Day              0
Ship Weekday          0
dtype: int64
```

Figure 3.15: Deal with Missing Values

3.3.4 Feature Scaling

Standardized scaling of `final_X_train` and `final_X_test` using `StandardScaler`, a common normalization method that scales the features by subtracting the mean and dividing by the standard deviation so that the features have zero mean and unit variance.

```
scaler = StandardScaler()
features = final_X_train.columns
final_X_train = scaler.fit_transform(final_X_train)
final_X_train = pd.DataFrame(final_X_train,columns=features)
final_X_test = scaler.transform(final_X_test)
final_X_test = pd.DataFrame(final_X_test,columns=features)
final_X_train.head()
```

Quantity	Profit	Profit Margin	Discounted Profit	Discount	Discount Percentage	Operating Expenses	Net Profit	Order Day	Ship Day	Ship Weekday
0.867561	0.830624	0.826407	-2.749708	-0.604093	-0.961354	-0.838451	-0.757745	0.995166	-0.385224	-0.006000
-0.454287	-0.468778	-0.451967	-1.760367	-0.956226	0.604659	1.088977	1.188020	-0.595375	0.361574	0.476275
-1.093101	-1.221680	-1.175006	-0.602911	0.014988	0.604659	-0.146819	0.968039	-0.225907	-0.645927	0.958550
-0.290615	-0.425751	-0.426658	-0.316490	0.287033	-0.961354	-0.838451	-0.757745	-1.009947	1.370823	-0.970550
-0.704700	-0.874945	-0.874668	0.126200	0.514748	-0.961354	-0.146819	-0.757745	-1.502531	1.522159	-1.452824

Figure 3.16: Normalized Output

These pre-processing steps ensure that the dataset is clean, well-structured, and ready for subsequent analysis.

3.4 Approach and Algorithm

This project employed three distinct analytical approaches to derive insights and make predictions based on the data:
1. **Market Basket Analysis:** Using the Apriori algorithm to discover association rules within transactional data.
2. **Recommender System Development:** Implementing a hybrid recommendation system that combines collaborative and content-based filtering methods.
3. **Regression Analysis:** Evaluating multiple regression models to determine the best fit for predicting our target variable based on the given features. Each method was carefully chosen and implemented to address specific analytical needs and to enhance the overall understanding and utility of the dataset.

3.4.1 Market Basket Analysis

Association rule mining is a task in the field of data mining that aims to discover association relationships between sets of items in a dataset. It is mainly used for mining transactional data such as shopping basket data. The goal of association rule mining is to find the frequent itemsets in the data and generate rules based on the association between them. Association rules are usually presented in the form of 'if...then...', where the 'if' part is the precondition and the 'then' part is the conclusion. The 'if' part is the precondition and the 'then' part is the conclusion. For example, a correlation rule could be, 'If a customer buys milk and bread, then they will probably also buy butter.' This rule can help merchants increase sales by recommending butter to customers based on their purchase history.

Apriori algorithm is a classical approach in association rule mining that mines association rules by generating frequent itemsets. A frequent itemset is the frequency of occurrence of an itemset in a transaction record for a certain support threshold. In this project, we use the Apriori algorithm to identify the strong association rules in our dataset. The Apriori workflow is shown in Figure 3.17.

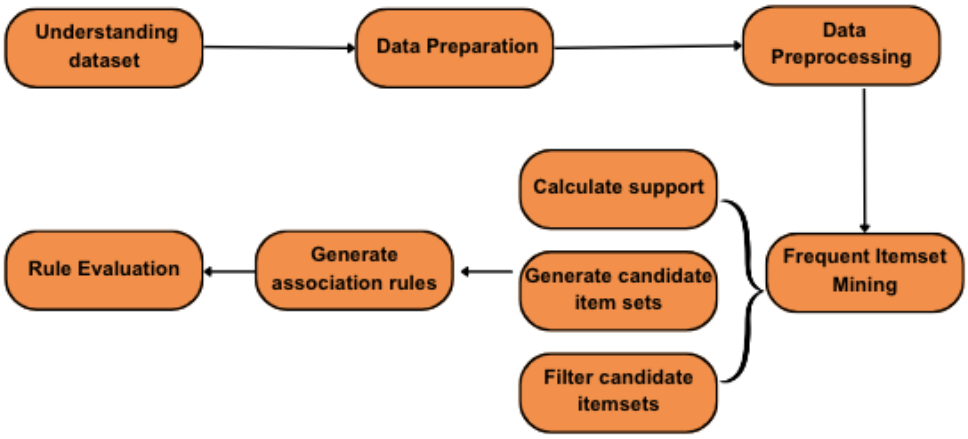

Figure 3.17: Apriori Workflow

3.4.2 Recommender System

This report presents the development and evaluation of a hybrid recommendation system designed to suggest products to users based on their previous interactions and preferences. The system leverages both collaborative filtering and content-based filtering to enhance recommendation accuracy and relevance.

Collaborative filtering is a recommendation method based on user behaviour, which recommends products by analysing the similarities between users. There are two main types:

User-based collaborative filtering: find other users with similar interests to the target user and then recommend products that these users like.

Item-based collaborative filtering: find other products that are similar to the target product and then recommend them to users who like those products.

Singular Value Decomposition (SVD) is a mathematical technique for dimensionality reduction and feature extraction. In collaborative filtering, SVD can be used to discover potential features of users and products to improve the accuracy of recommendations.

Content-based filtering is a product feature-based recommendation method that analyses the features of products that users liked in the past to recommend products with similar features.

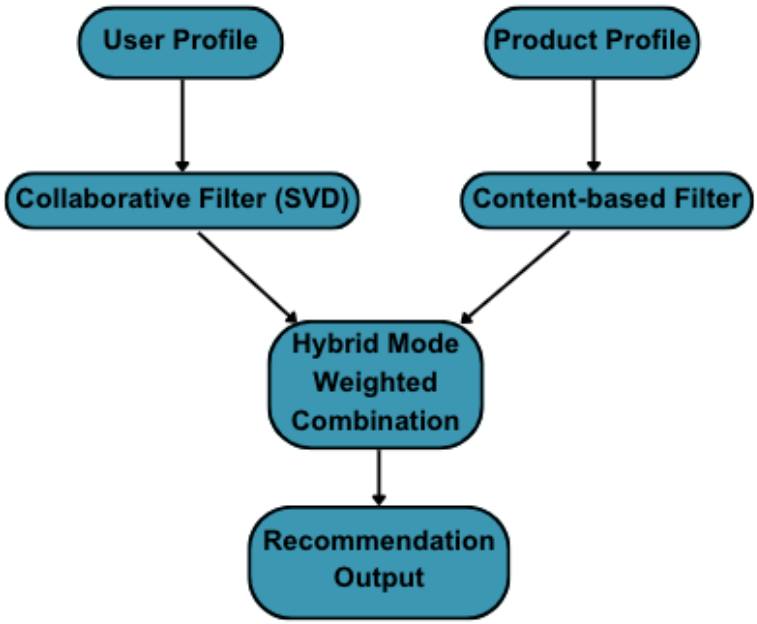

Figure 3.18: Hybrid Recommendation system Flow Chart

The system first utilizes the user's historical interaction data and product feature information input into the collaborative filtering and content-based filtering modules. In collaborative filtering, the user-item matrix is processed by SVD technique to extract low-dimensional user and item representations, while content-based filtering analyses the product feature vectors. The outputs obtained from these two approaches are subsequently weighted and combined in a hybrid model to generate comprehensive recommendation results. Finally, the system ranks and selects based on these results and outputs a list of recommended products that best match the user's preferences.

3.4.3 Regression Analysis

Regression analysis is used to study the relationship between a dependent variable (target variable) and one or more independent variables (predictor variables), and is mainly used for prediction and causal analysis.

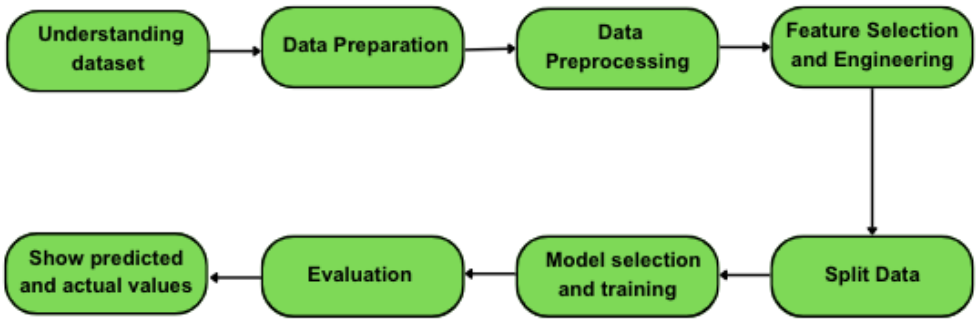

Figure 3.19: Regression Analysis Flow Chart

In this report, we evaluated a series of regression models to determine their performance on specific data sets. The following is a brief description of these models:

Random Forest Regressor: as an integrated learning technique, it improves the accuracy and robustness of the model by constructing multiple decision trees and combining their predictions.

Poisson Regressor: a regression model specialized for count data, based on the Poisson distribution, suitable for predicting the number of events.

Decision Tree Regressor: Inferring target values from features by learning simple decision rules. The tree structure of the model makes it intuitive and easy to understand but can lead to overfitting.

KNeighbours Regressor: an instance-based learning algorithm that predicts outcomes by finding the K nearest neighbours of a test data point, simple but computationally expensive.

Stochastic Gradient Descent Regressor (SGDRegressor): a linear regression model that combines a stochastic gradient descent optimization algorithm, suitable for large-scale datasets, but may require careful tuning of parameters.

Lasso regressor (Lasso): feature selection via L1 regularization, reduces model complexity and helps prevent overfitting, but may cause some coefficients to become zero.

Ridge regression (Ridge): uses L2 regularization to reduce the risk of overfitting and, unlike Lasso, tends to make the coefficients close to zero but not exactly zero.

Linear Regression: the most basic regression model, suitable for cases where there is a linear relationship between the features and the target variable, simple and effective.

4.0 EXPERIMENT AND ANALYSIS

4.1 Market Basket Analysis

Market Basket Analysis (MBA) is a powerful tool for understanding customers' purchasing preferences and habits. By analyzing associations and co-occurrences between products or categories of products, MBA reveals hidden patterns in buying behavior so that strategic improvements can be made about various aspects such as product placement, promotions, inventory management, store layout which can lead to increased sales customer satisfaction.

In this section, MBA is performed on the Superstore dataset. The python code snippet in Figure 4.1 is used to find out how many unique Product ID is in the Superstore dataset. The output shows that there are 1,862 unique products.

```
# Calculate the number of unique product IDs
num_unique_product_ids = df['Product ID'].nunique()

The number of unique product IDs is: 1862
```

Figure 4.1: Python snippet to find the number of unique Product ID in Superstore dataset

Applying MBA on such a large number of products might cause problems in terms of computational power and output may be hard to interpret. Figure 4.2 below shows that Superstore dataset only has 17 different sub-categories compared to Product ID with 1862 different categories. Therefore, this study focuses on sub-categories. The top 3 best-selling sub-categories are Binders followed by Paper and Furnishings. Despite knowing this, there is not enough knowledge to know what are the combinations of sub-categories that are frequently bought together or a good chance for cross-selling.

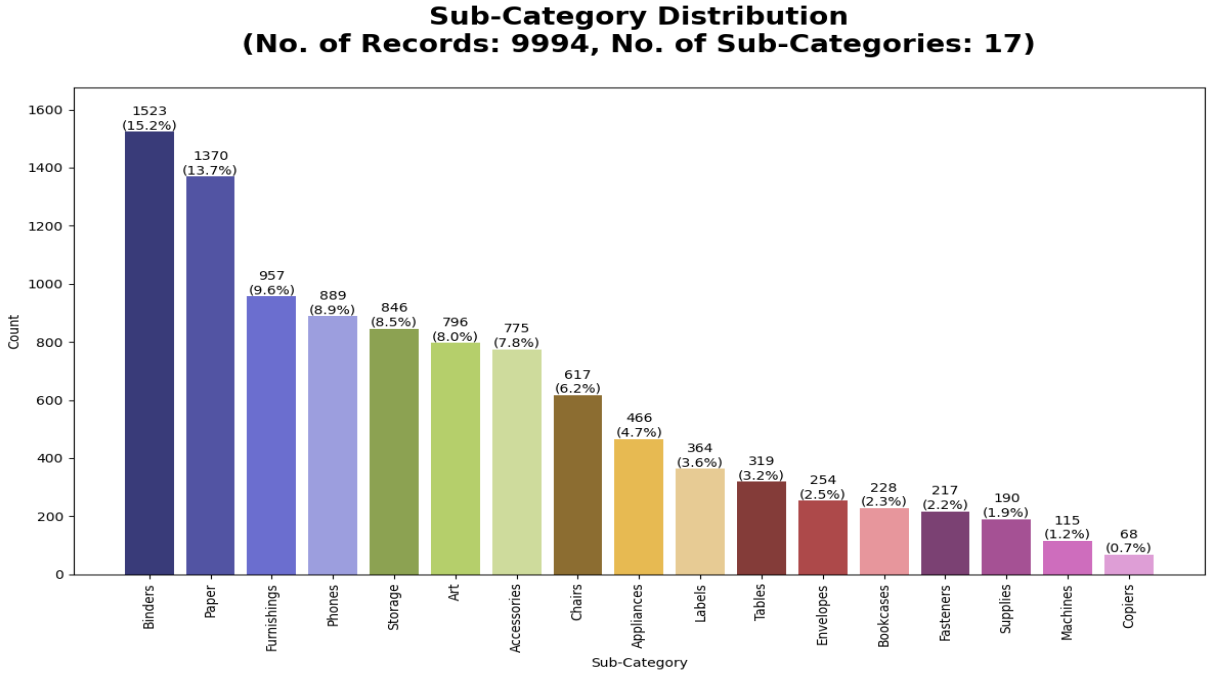

Figure 4.2: Superstore dataset Sub-category distribution

In order to discover what are the combinations of sub-categories are frequently purchased together and what are the sub-categories with strong associations, the associations rules for these sub-categories are determined using the Apriori algorithm.

```python
# Import the apriori and association_rules functions from the mlxtend.frequent_patterns module
from mlxtend.frequent_patterns import apriori, association_rules

# Generate frequent itemsets
frequent_itemsets = apriori(df_transformed, min_support=0.01, use_colnames=True)

# Generate association rules
rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1)
```

Figure 4.3: Python code snippet to identify frequent itemsets and generate association rules.

Figure 4.3 above shows Apriori algorithm Python code snippet is used to identify frequent itemsets with support of at least 0.01 and lift of at least 1 to generate association rules. Table 4.1 shows the that 68 Frequent itemsets is identified and 42 association rules is generated shown in Table 4.2.

Table 4.1 Frequent itemsets

	support	itemsets
0	0.143342	(Accessories)
1	0.090038	(Appliances)
2	0.145937	(Art)
3	0.262727	(Binders)
4	0.044720	(Bookcases)
..	...	...
63	0.023358	(Phones, Storage)
64	0.010381	(Tables, Phones)
65	0.010182	(Furnishings, Binders, Paper)
66	0.010781	(Binders, Phones, Paper)
67	0.010581	(Binders, Storage, Paper)

[68 rows x 2 columns]

Table 4.2 Association rules generated using Apriori algorithm

Antecedents	**Consequents**	**Support**	**Confidence**	**Lift**
(Binders)	(Appliances)	0.025953	0.098784	1.09714
(Appliances)	(Binders)	0.025953	0.288248	1.09714
(Furnishings)	(Appliances)	0.015971	0.09122	1.013129
(Appliances)	(Furnishings)	0.015971	0.177384	1.013129
(Appliances)	(Paper)	0.021761	0.241685	1.016458
(Paper)	(Appliances)	0.021761	0.09152	1.016458
……	……	……	……	……
(Binders, Paper)	(Storage)	0.010581	0.192727	1.242434
(Storage, Paper)	(Binders)	0.010581	0.297753	1.133316
(Binders)	(Storage, Paper)	0.010581	0.040274	1.133316
(Storage)	(Binders, Paper)	0.010581	0.068211	1.242434
(Paper)	(Binders, Storage)	0.010581	0.0445	1.120114

In this study, only the support, confidence and lift are considered. Further steps are taken to study these metrics in more detail. To narrow down further on the list of 41 association rules shown in Table 4.2 so that useful rules for strategic business decision making is selected, only the association rules in Table 4.2 that meet the criteria of top 10 lift values is chosen.

According to recent studies (Smith et al., 2020), lift serves as a superior metric for identifying meaningful product associations compared to confidence and support [17]. Lift quantifies the strength of relationships between the sub-categories based on their co-occurrence in transactions. The Python code in Figure 4.4 below picks the Top 10 association rules by highest lift values from Table 4.2.

```
# Sort rules by lift in descending order and select top 10
rules_sorted = rules.sort_values(by='lift', ascending=False).head(10)

# Display the sorted association rules
print(rules_sorted)
```

Figure 4.4: Python code snippet to sort highest to lowest lift and pick top 10

The generated output is 10 association rules with the highest lift values shown in Table 4.3 shown below.

Table 4.3 Association rules selected based on Top 10 lift values

Antecedents	Consequents	Support	Confidence	Lift
(Storage)	(Binders, Paper)	0.010581	0.068211	1.242434
(Binders, Paper)	(Storage)	0.010581	0.192727	1.242434
(Phones)	(Binders, Paper)	0.010781	0.066339	1.208336
(Binders, Paper)	(Phones)	0.010781	0.196364	1.208336
(Binders)	(Phones, Paper)	0.010781	0.041033	1.174494
(Phones, Paper)	(Binders)	0.010781	0.308571	1.174494
(Fasteners)	(Paper)	0.011779	0.274419	1.154125
(Paper)	(Fasteners)	0.011779	0.049538	1.154125
(Paper)	(Binders, Phones)	0.010781	0.04534	1.141248
(Binders, Phones)	(Paper)	0.010781	0.271357	1.141248

To view the association rules with top 10 lift values in Table 4.3 visually, the Lift vs Confidence 2D chart is plotted and shown on Figure 4.5 below. This 2D chart is split into 4 quadrants by the dotted lines of median lift and median confidence.

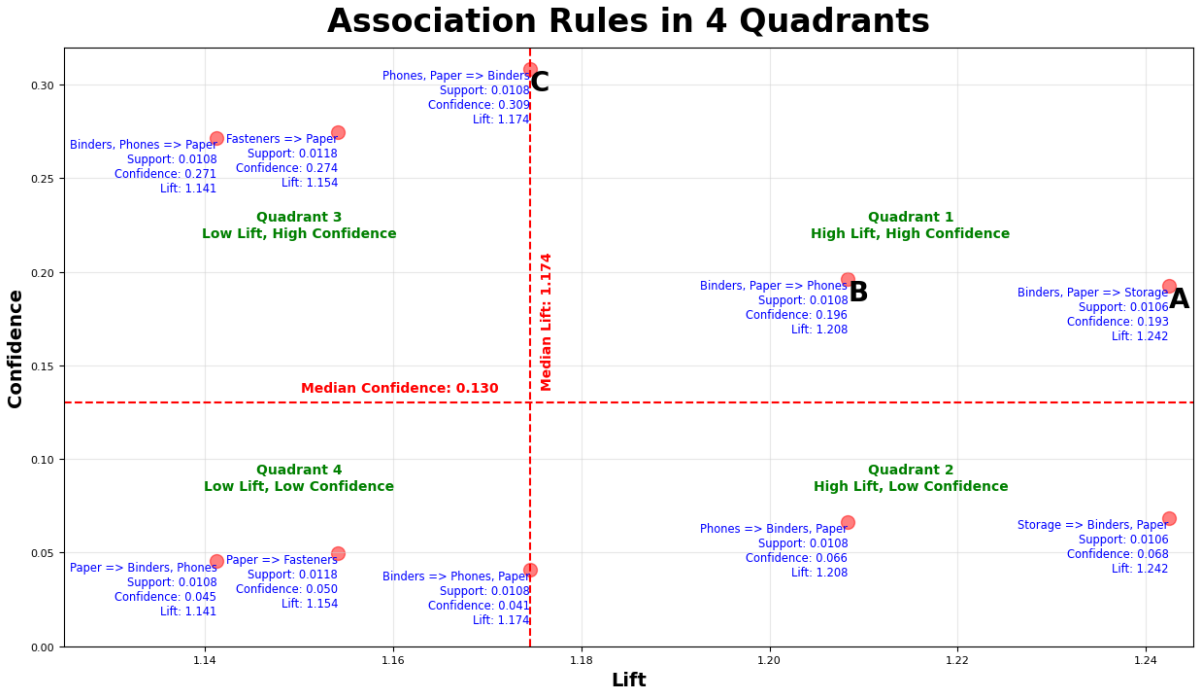

Figure 4.5: Lift vs Confidence 2D chart for top 10 association rules by largest Lift values

The 2D plot in Figure 4.5 incorporates association rules within the framework of customer segmentation emphasized by retail literature. This segmentation approach enables personalized marketing strategies tailored to each customer segment's preferences and needs for more targeted customer engagement to increase customer loyalty, and drive sales growth.

Although there are 4 regions in Figure 4.5, the most important region is Quadrant 1 which is located at the top right-hand side of the 2D plot because it is the region with high lift and high confidence. Association rules (plotted as red nodes) falling within this region would be most preferred in terms of identifying statistically significant purchase patterns and customer preferences.

There are 3 association rules represented by Labels A, B and C in Quadrant 1. Label A is the association rule of (Binders, Paper) => (Storage) has the highest combination of lift value of 1.242 and confidence value of 0.193 followed by Label B with the association rule (Binders, Paper) => (Phones) with the second highest combination of lift value of 1.208 and confidence value of 0.196, followed by Label C with association rule (Phones, Paper) => (Binders) with third highest combination of lift value of 1.174 and confidence value of 0.309.

By prioritizing association rules labelled A, B and C in Quadrant 1 shown in Figure 4.5, Superstore can strategically place Binders, Papers, Phones and Storage near each other to enhance shopping experience to its customers. For example, promoting Paper, Binders, Storage and Phones in marketing efforts to cross-sell these products or bundle them together has a higher potential of increasing revenue for Superstore. Superstore is also able to optimize its inventory management knowing that if one product runs low on inventory for example Papers, it could indicate Binders could be running low on inventory also, therefore make necessary purchases ahead of time to avoid stockout.

4.2 Recommender System

Using the `surprise` library, a SVD model was implemented.

```
df_cf = df[['Customer ID', 'Product ID', 'Sales']]

reader = Reader(rating_scale=(df_cf['Sales'].min(), df_cf['Sales'].max()))
data = Dataset.load_from_df(df_cf, reader)

trainset, testset = train_test_split(data, test_size=0.2)
```

```
# Use Singular Value Decomposition (SVD) for collaborative filtering
algo = SVD()
algo.fit(trainset)

predictions = algo.test(testset)
```

Figure 4.6: SVD Algorithm

The Content-Based Filtering (CBF) approach to calculating similarity between products relies heavily on the characteristics of the product rather than the historical behaviour of the user.

Unique coding of Category and Sub-Category for categorical features. Key features such as `Product ID`, `Category`, `Sub-Category`, `Sales`, `Quantity`, `Discount`, and `Profit` were used to compute item similarities using cosine similarity.

```
df_cb = df[['Product ID', 'Category', 'Sub-Category', 'Sales', 'Quantity', 'Discount', 'Profit']]

# One-hot encode categorical features
df_cb_encoded = pd.get_dummies(df_cb, columns=['Category', 'Sub-Category'])

scaler = StandardScaler()
df_cb_encoded[['Sales', 'Quantity', 'Discount', 'Profit']] = scaler.fit_transform(df_cb_encoded[['Sales', 'Quan
```

```
# Compute cosine similarity matrix
similarity_matrix = cosine_similarity(df_cb_encoded.drop('Product ID', axis=1))
product_idx = pd.Series(df_cb_encoded.index, index=df_cb_encoded['Product ID']).drop_duplicates()
```

```
similarity_matrix
```

```
array([[ 1.        ,  0.5408671 ,  0.3641069 , ...,  0.16500063,
         0.16169589,  0.37995612],
       [ 0.5408671 ,  1.        ,  0.13750351, ...,  0.0394667 ,
         0.06902485,  0.2847916 ],
       [ 0.3641069 ,  0.13750351,  1.        , ...,  0.15784517,
         0.53780733,  0.66475299],
       ...,
       [ 0.16500063,  0.0394667 ,  0.15784517, ...,  1.        ,
        -0.09241492,  0.1619919 ],
       [ 0.16169589,  0.06902485,  0.53780733, ..., -0.09241492,
         1.        ,  0.49926279],
       [ 0.37995612,  0.2847916 ,  0.66475299, ...,  0.1619919 ,
         0.49926279,  1.        ]])
```

Figure 4.7: Similarity between products

As shown in Figure 4.8, the system captures the user's historical behavioural data through the `get_cf_recommendations` function, filters the predicted ratings corresponding to a specific user ID, and extracts the top N recommended products with the highest ratings by sorting them in order of high or low ratings. The `get_cb_recommendations` function calculates the similarity between products based on their feature information and recommends the N other products that are most similar to them for each input product ID. In the `hybrid_recommendations` function, the system combines the results of collaborative filtering and content-based filtering by merging the recommendation lists and removing duplicates to ensure that the recommended products are both consistent with the user's historical preferences and novel. In addition, the system excludes products that have already been rated by the user to avoid duplicate recommendations, thus generating a final recommendation list containing up to N unique products.

```python
def get_cf_recommendations(user_id, top_n=10):
    # Get all predictions for the user
    user_predictions = [pred for pred in predictions if pred.uid == user_id]
    # Sort predictions by estimated rating
    user_predictions.sort(key=lambda x: x.est, reverse=True)
    # Get top N recommendations
    cf_recommendations = [pred.iid for pred in user_predictions[:top_n]]
    return cf_recommendations

def get_cb_recommendations(product_ids, top_n=10):
    cb_recommendations = []
    for product_id in product_ids:
        if product_id in product_idx:
            idx = product_idx[product_id]
            similar_products = list(enumerate(similarity_matrix[idx]))
            # Ensure we handle scalar values properly
            similar_products = [(i, sim.flatten()[0] if not np.isscalar(sim) else sim) for i, sim in similar_products]
            similar_products = sorted(similar_products, key=lambda x: x[1], reverse=True)
            cb_recommendations.extend([df_cb_encoded['Product ID'].iloc[i[0]] for i in similar_products[1:top_n+1]])
    return cb_recommendations


def hybrid_recommendations(user_id, top_n=10):
    user_data = df[df['Customer ID'] == user_id]
    user_rated_products = user_data['Product ID'].unique()

    # Collaborative filtering recommendations
    cf_recommendations = get_cf_recommendations(user_id, top_n=top_n)

    # Content-based filtering recommendations
    cb_recommendations = get_cb_recommendations(user_rated_products, top_n=top_n)

    # Combine recommendations and get unique products
    combined_recommendations = list(set(cf_recommendations + cb_recommendations))

    final_recommendations = [prod for prod in combined_recommendations if prod not in user_rated_products]

    return final_recommendations[:top_n]
```

Figure 4.8: Hybrid Recommender System

Figure 4.9 shows that Recommended Products are generated for user `CG-12520` and shows the details of the recommended products. These products are available in various categories and subcategories like `Office Supplies`, `Furniture`, `Technology`, covering subcategories like `Labels`, `Tables`, `Storage`, `Decorative`, `Art`, `Phones` and `Appliances`.

```python
user_id = 'CG-12520'
recommended_products_id = hybrid_recommendations(user_id, top_n=10)
print(f"Recommended products for user {user_id}: {recommended_products_id}")
```

```
Recommended products for user CG-12520: ['FUR-TA-10001539', 'FUR-TA-10000577', 'OFF-AP-10002892', 'OFF-AR-10002833', 'OFF-LA-10000240', 'TEC-PH-10002275', 'FUR-FU-10001487', 'OFF-ST-10000760']
```

```python
recommended_products_details = Product_info_df[Product_info_df['Product ID'].isin(recommended_products_id)]
print(f'Recommended products for user {user_id}:')
recommended_products_details
```

```
Recommended products for user CG-12520:
```

	Product ID	Product Name	Category	Sub-Category
2	OFF-LA-10000240	Self-Adhesive Address Labels for Typewriters b...	Office Supplies	Labels
3	FUR-TA-10000577	Bretford CR4500 Series Slim Rectangular Table	Furniture	Tables
4	OFF-ST-10000760	Eldon Fold 'N Roll Cart System	Office Supplies	Storage
5	FUR-FU-10001487	Eldon Expressions Wood and Plastic Desk Access...	Furniture	Furnishings
6	OFF-AR-10002833	Newell 322	Office Supplies	Art
7	TEC-PH-10002275	Mitel 5320 IP Phone VoIP phone	Technology	Phones
9	OFF-AP-10002892	Belkin F5C206VTEL 6 Outlet Surge	Office Supplies	Appliances
10	FUR-TA-10001539	Chromcraft Rectangular Conference Tables	Furniture	Tables

Figure 4.9: Recommended products by user 'CG-12520'

With this approach, the recommender system is able to combine collaborative filtering and content-based filtering to generate personalized and enriched recommendations and improve user satisfaction.

4.3 Regression Analysis

We split 30% of the samples from the dataset into a test set to evaluate the performance and generalisation ability of the model. The remaining 70% of the samples are used as a training set for learning the parameters and fitting the model, as shown in Figure 4.10.

```
X = df1.drop('Sales',axis=1)
y = df1['Sales']
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

print('X_train: ',X_train.shape)
print('X_test: ',X_test.shape)
print('y_train: ',y_train.shape)
print('y_test: ',y_test.shape)
```

```
X_train:  (6993, 63)
X_test:  (2997, 63)
y_train:  (6993,)
y_test:  (2997,)
```

Figure 4.10: Spit training and testing dataset

Visualizing the correlation between different features is very helpful to quickly understand the relationship between data features in Figure 4.11.

We retained 4 continuous feature variables: `Quantity`, `Discount`, `Profit`, and `Postal Code` to measure their impact on the target variable (sales) using a heatmap to show the autocorrelation of the variables. Specifically, we noted a significant positive correlation between `Sales` and `Profit`, indicating that an increase in sales is often accompanied by an increase in profit\. `Quantity'' also shows a positive correlation with `Profit`. However, we also found a slight negative correlation between `Discount` and `Profit`, which may imply that higher discounts may have some negative impact on profit.

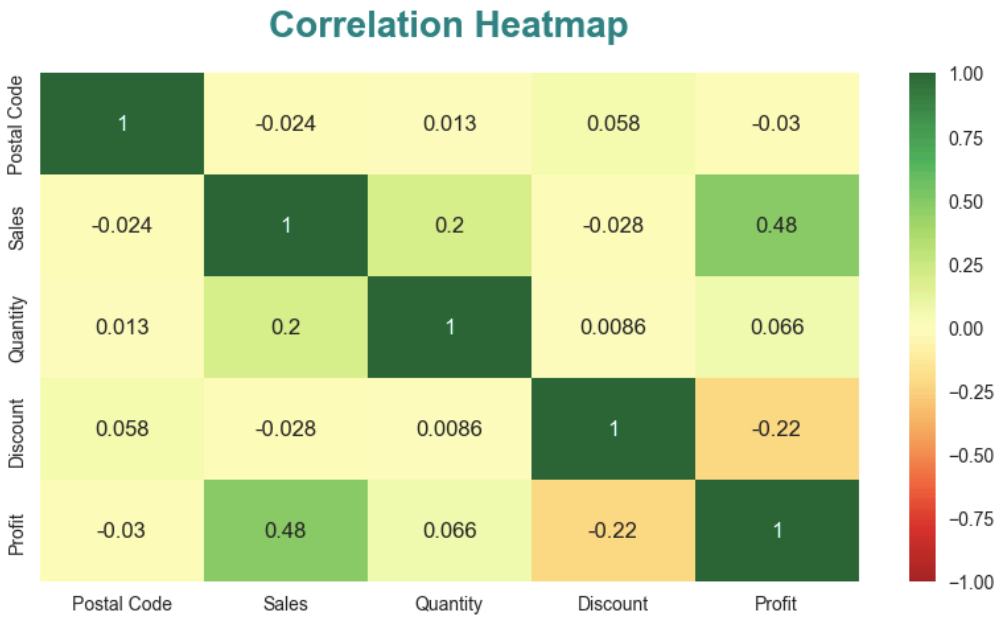

Figure 4.11: Correlation Heatmap

In this regression analysis, we applied a variety of regression models including Random Forest Regressor, Poisson Regressor, Decision Tree Regressor, K Nearest Neighbor Regressor, Stochastic Gradient Descent Regressor, Lasso Regressor, Ridge Regressor, and Linear Regression Model. By using multiple models, we can achieve a more comprehensive model comparison and selection and more reliable prediction results from Figure 4.12 to Figure 4.19.

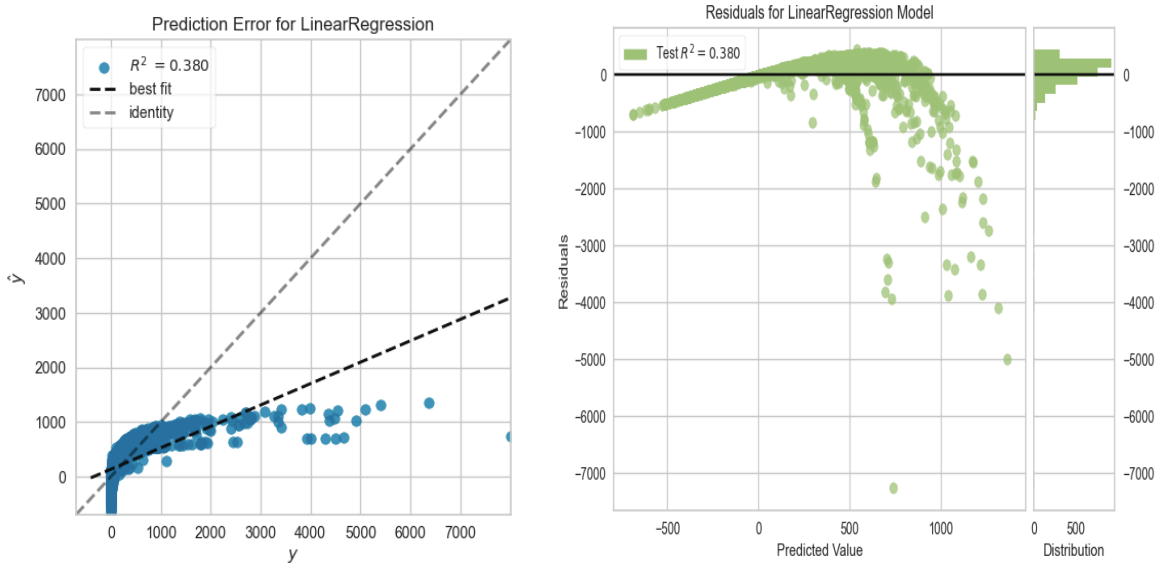

Figure 4.12: Prediction error and Residual plots for linear regression model

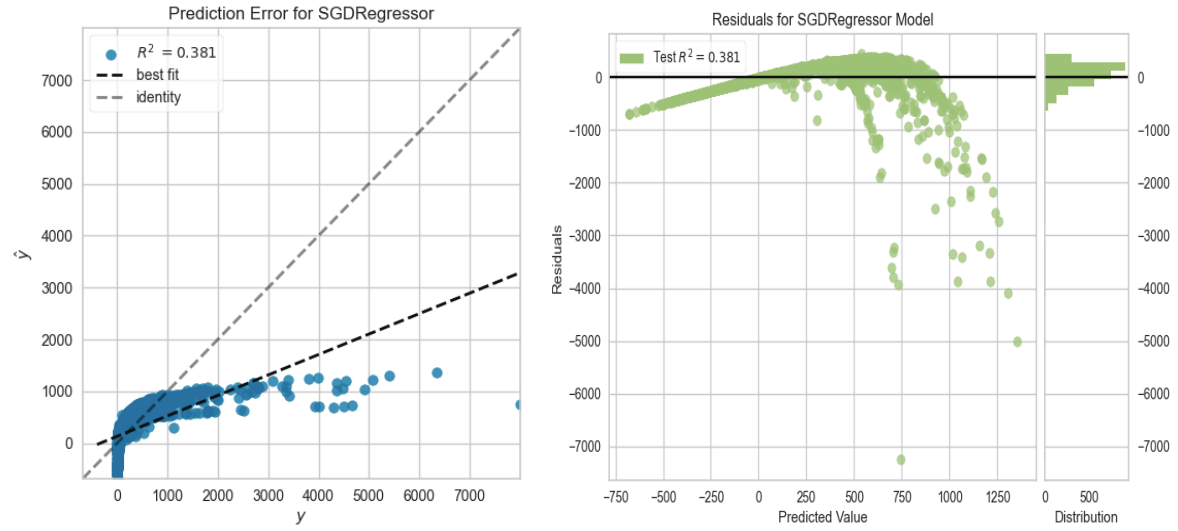

Figure 4.13: Prediction error and Residual plots for SGDRegressor model

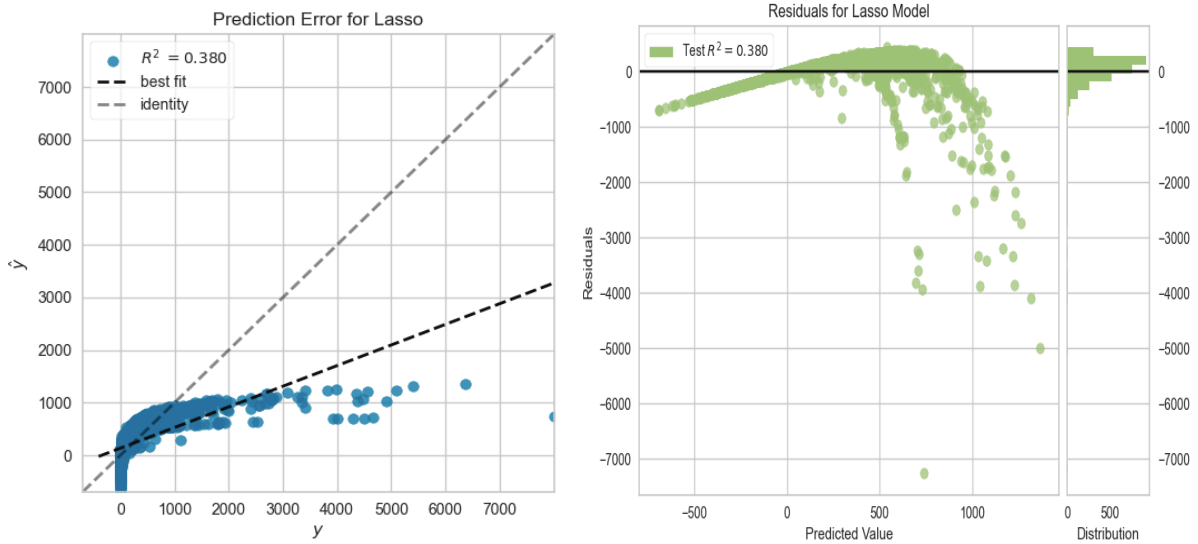

Figure 4.14: Prediction error and Residual plots for Lasso model

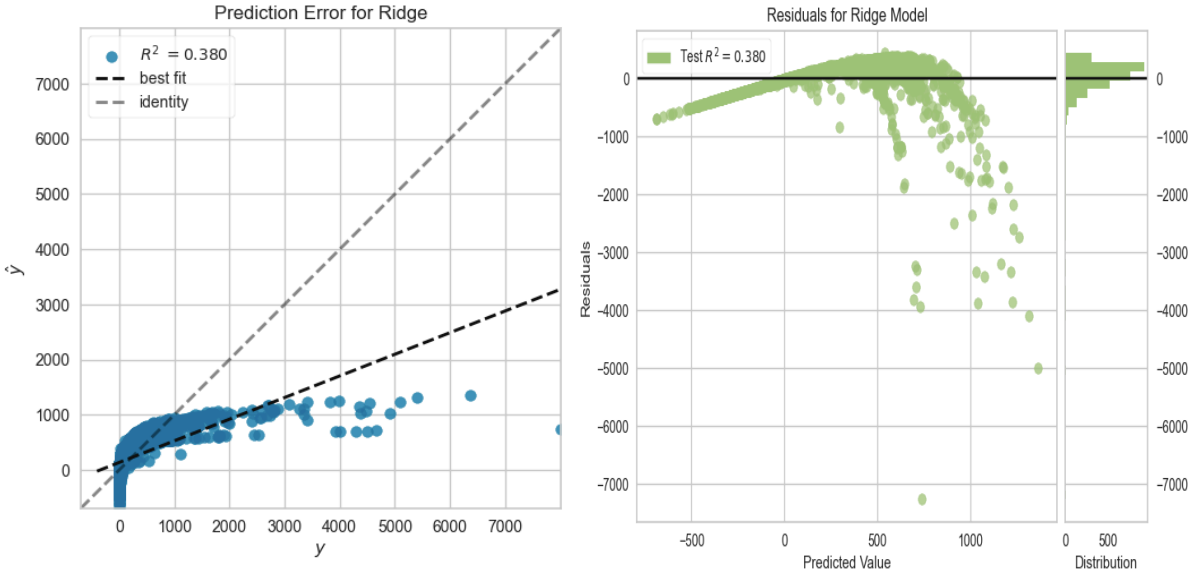

Figure 4.15: Prediction error and Residual plots for Ridge model

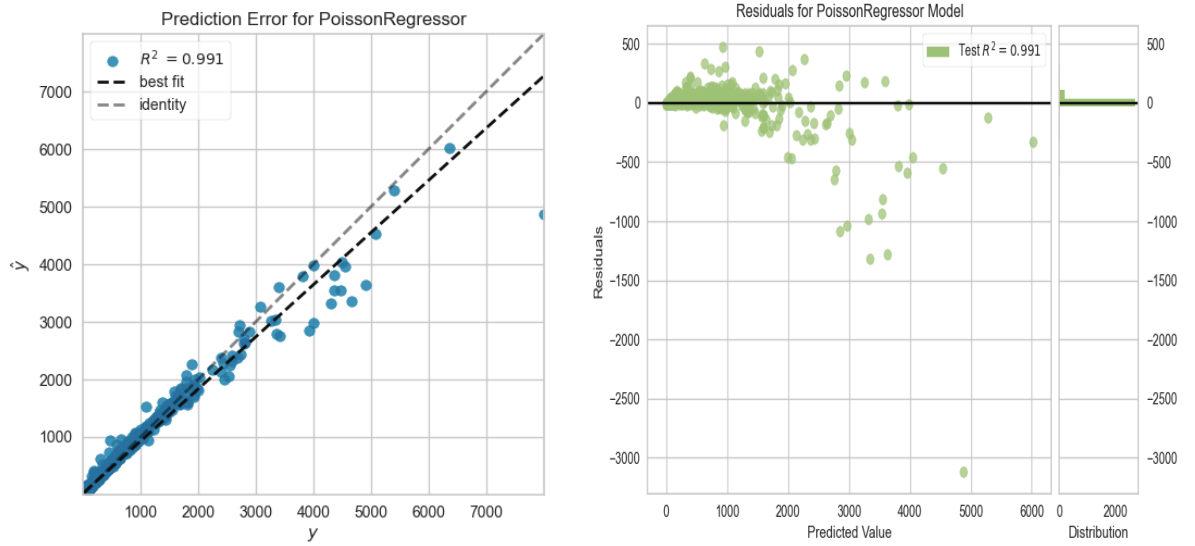

Figure 4.16: Prediction error and Residual plots for Poisson Regressor model

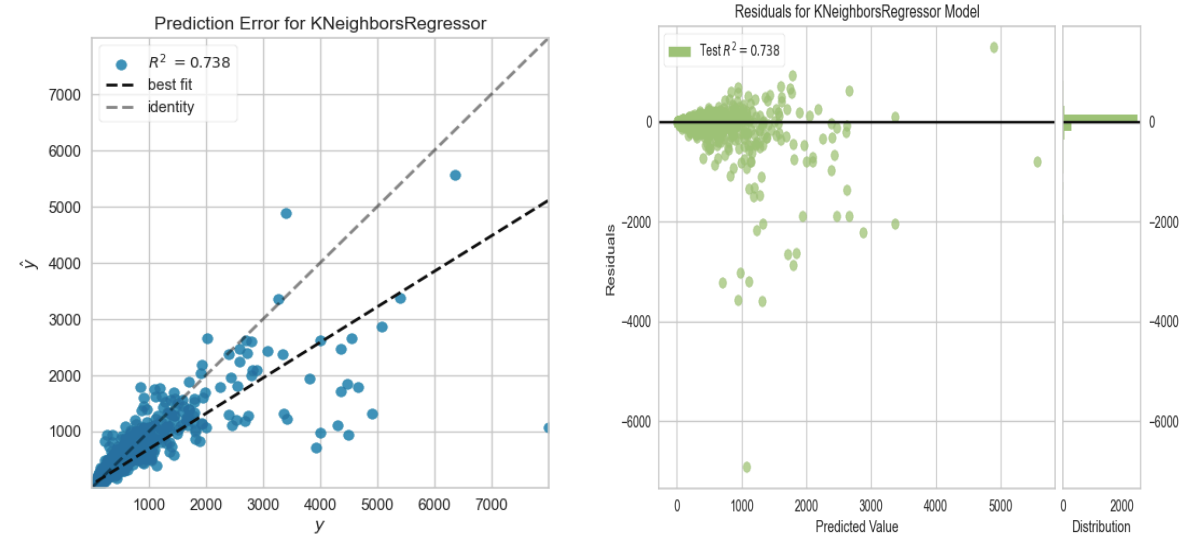

Figure 4.17: Prediction error and Residual plots for KNeighbors Regressor model

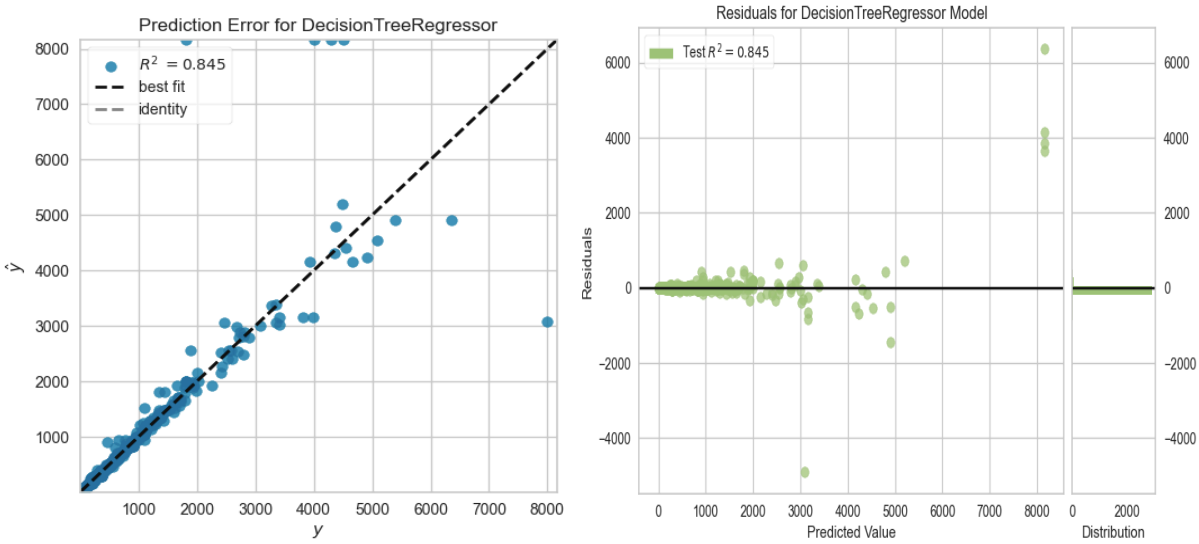

Figure 4.18: Prediction error and Residual plots for Decision Tree Regressor mode

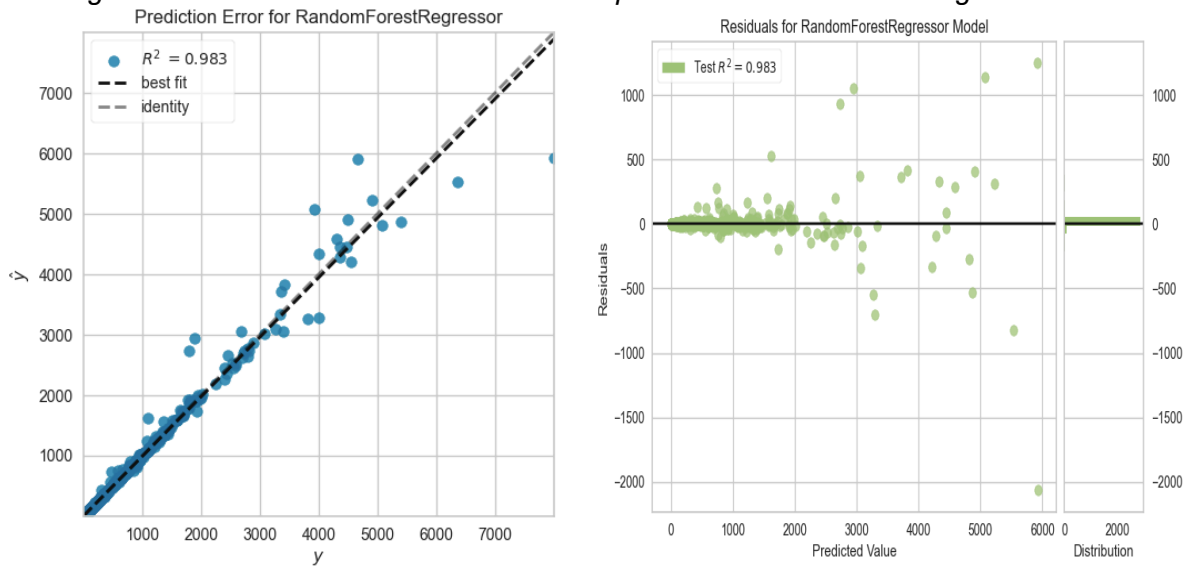

Figure 4.19: Prediction error and Residual plots for Random Forest Regressor model

In this regression analysis, the random forest regressor performed the best. Its R² value was 0.983, indicating that the model was able to explain 98.3% of the variability of the variables. In addition, its MAE, MSE and RMSE values are 7.92, 4353.78 and 65.98, respectively, showing high prediction accuracy and low error. In contrast, other models such as Poisson regressor, Decision tree regressor, and K-nearest neighbour regressor are inferior to random forest regressor in terms of explanatory power and prediction accuracy.

Table 4.4 Comparison of Regression Models Performance Metrics

Model	R2	MAE	MSE	RMSE
RandomForestRegressor	0.983018	7.924408	4353.784587	65.983214
Poisson Regressor	0.968298	20.509232	8127.607600	90.153245
DecisionTreeRegressor	0.845301	17.166289	39661.415549	199.151740
KNeighorsRegressor	0.738266	56.583806	67102.921825	259.042317
SGDRegressor	0.380694	212.694202	158776.357892	398.467512
Lasso	0.380164	213.651781	158912.186541	398.637914
Ridge	0.380153	213.656787	158915.011474	398.641457
Linear Regression	0.380153	213.656802	158915.013338	398.641460

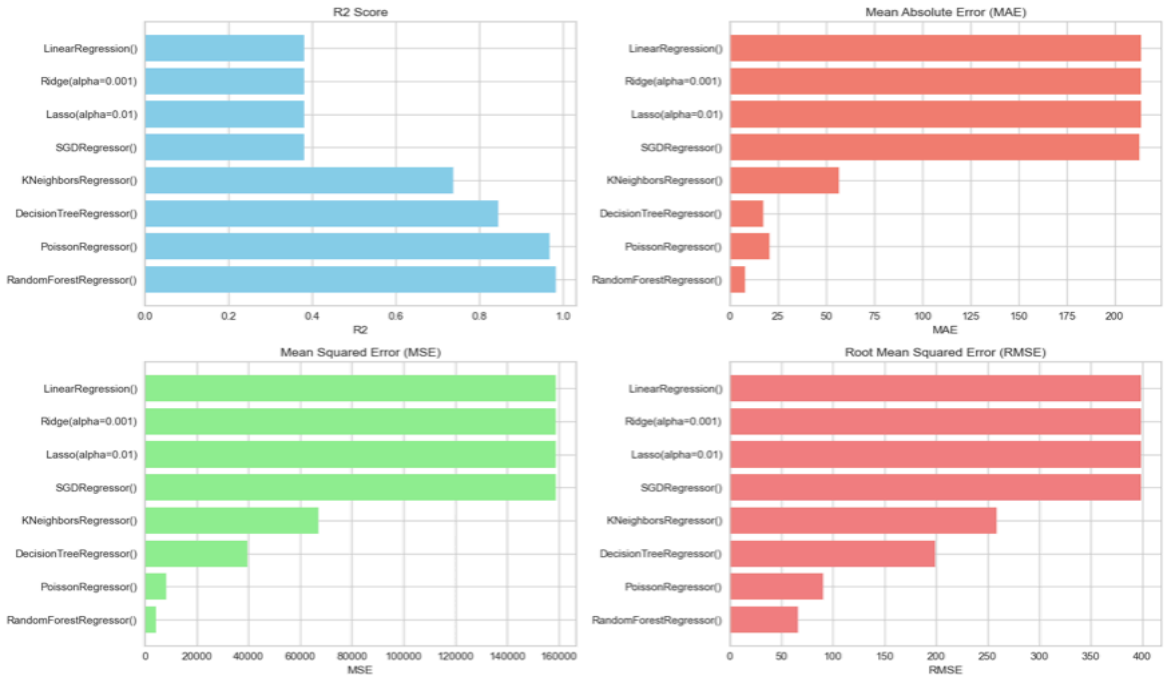

Figure 4.20: Model Evaluation

In Figure 4.21, we used `GridSearchCV` to perform grid search on `n_estimators` (number of trees), `min_samples_split` (minimum number of samples required to split a node), and `min_samples_leaf` (minimum number of samples required to leaf a node) and evaluated the model performance based on 5-fold cross validation. The model performance is evaluated on the basis of 5-fold cross validation. The best combination of parameters was found: {`min_samples_leaf`: 1, `min_samples_split`: 2, `n_estimators`: 300}. After the hyperparameter tuning, the performance of the random forest regression model is greatly improved. the values of MAE, MSE and RMSE are low, and the R^2 score is very high, close to 1, which indicates that the model can fit the data well and has high prediction accuracy.

```
from sklearn.model_selection import GridSearchCV
# then tune the parameter
param_grid = {
    'n_estimators': [100, 200, 300],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}
rf = RandomForestRegressor()
grid_search = GridSearchCV(estimator=rf, param_grid=param_grid, cv=5, n_jobs=-1, verbose=1)
grid_search.fit(final_X_train, y_train.ravel())

best_rf = grid_search.best_estimator_
y_pred = best_rf.predict(final_X_test)
```

```
Fitting 5 folds for each of 27 candidates, totalling 135 fits
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages/numpy/ma/core.p
d value encountered in cast
  _data = np.array(data, dtype=dtype, copy=copy,
```

```
print("Best parameters found by GridSearchCV:")
print(grid_search.best_params_)
```

```
Best parameters found by GridSearchCV:
{'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 300}
```

```
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
r2 = r2_score(y_test, y_pred)

print("Mean Absolute Error:", mae)
print("Mean Squared Error:", mse)
print("Root Mean Squared Error:", rmse)
print("R2 Score:", r2)
```

```
Mean Absolute Error: 7.530318304971641
Mean Squared Error: 3861.446849788698
Root Mean Squared Error: 62.14054111277675
R2 Score: 0.9849384568097486
```

Figure 4.21: Tuning Random Forest Regression Models

5.0 CONCLUSION

This study employed Market Basket Analysis (MBA) on the Superstore dataset to uncover associations between product sub-categories, aiming to improve product placement, promotions, inventory management, and store layout. The analysis revealed 1,862 unique products, prompting a focus on 17 sub-categories to manage computational complexity.

Using the Apriori algorithm, we identified 68 frequent itemsets and generated 42 association rules. To refine the results for strategic decision-making, we prioritized rules with the highest lift values, resulting in 10 association rules. The top three rules indicated strong associations between (Binders, Paper) and Storage, (Binders, Paper) and Phones, and (Phones, Paper) and Binders. These insights can guide product placement and promotional strategies, enhancing customer satisfaction and sales.

A hybrid recommender system combining collaborative filtering and content-based filtering was also developed. This system leverages user historical data and product characteristics to generate personalized recommendations, further improving the shopping experience.

In the regression analysis, multiple models were compared to predict sales. The Random Forest Regressor outperformed others, explaining 98.3% of the variability in the data and exhibiting high prediction accuracy. Hyperparameter tuning further improved the model's performance.

Future Research

Future research can build upon this study by exploring several avenues:

1. **Enhanced Data Integration**: Incorporate additional data sources such as customer demographics, online browsing behavior, and social media interactions to enrich the dataset. This can provide deeper insights into customer preferences and enhance predictive accuracy.
2. **Real-Time Analytics**: Implement real-time analytics to dynamically update recommendations and inventory management based on live data streams. This can help in adapting to changing customer behaviors and market trends more swiftly.
3. **Advanced Algorithms**: Experiment with more advanced algorithms like deep learning and ensemble methods to further improve the accuracy of MBA, recommendation systems, and regression models.
4. **User Feedback Loop**: Integrate a feedback mechanism to continuously refine recommendation models based on user interactions and satisfaction levels. This can help in maintaining the relevance and accuracy of recommendations over time.
5. **Broader Applications**: Apply the developed models to other retail sectors to test their robustness and adaptability. Comparative studies across different retail environments can provide valuable insights for further improvement.

By pursuing these research directions, the potential of data-driven strategies in retail can be fully harnessed, leading to more personalized customer experiences and optimized business operations.

References

Dahake, P. S., Bagaregari, P., & Dahake, N. S. (2024). Shaping the Future of Retail: A Comprehensive Review of Predictive Analytics Models for Consumer Behavior. *Entrepreneurship and Creativity in the Metaverse*, 143-160. https://doi.org/10.4018/979-8-3693-1734-1.ch011

EffectiveSoft. (n.d.). Using predictive analytics in retail: benefits, examples, and best practices. https://www.effectivesoft.com/blog/predictive-analytics-in-retail.html

Gauri, D. K., Jindal, R. P., Ratchford, B., Fox, E., Bhatnagar, A., Pandey, A., ... & Howerton, E. (2021). Evolution of retail formats: Past, present, and future. *Journal of Retailing*, *97*(1), 42-61. https://doi.org/10.1016/j.jretai.2020.11.002

Hameli, K. (2018). A literature review of retailing sector and business retailing types. *ILIRIA International Review*, *8*(1), 67-87.

Jannach, D., Pu, P., Ricci, F., & Zanker, M. (2021). Recommender systems: Past, present, future. *Ai Magazine*, *42*(3), 3-6. https://doi.org/10.1609/aimag.v42i3.18139

Kawale, N. M., & Dahima, S. (2018). Market basket analysis using apriori algorithm in r language. *nternational Journal of Trend in Scientific Research and Development*, *2*(4), 2628-2633.

Marzolf, M. J. (2023). *Exploring Challenges and the Evolution of the Retail Industry: A Consumer Perspective*. Michigan State University.

Nur, M. F., & Siregar, A. (2024). Exploring the Use of Cluster Analysis in Market Segmentation for Targeted Advertising. *IAIC Transactions on Sustainable Digital Innovation (ITSDI)*, *5*(2), 158-168. https://doi.org/10.34306/itsdi.v5i2.665

Patwary, A. H., Eshan, M. T., Debnath, P., & Sattar, A. (2021, July). Market Basket Analysis Approach to Machine Learning. In *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)* (pp. 1-9). IEEE. https://doi.org/10.1109/ICCCNT51525.2021.9580175

ProGlobal Business Solutions. (n.d.). Six steps in CRISP-DM - The standard data mining process. Retrieved from https://www.proglobalbusinesssolutions.com/six-steps-in-crisp-dm-the-standard-data-mining-process/

Rana, S., & Mondal, M. N. (2021). A Seasonal and Multilevel Association Based Approach for Market Basket Analysis in Retail Supermarket. *European Journal of Information Technologies and Computer Science*, *1*(4), 9-15. https://doi.org/10.24018/compute.2021.1.4.31

Seyedan, M., & Mafakheri, F. (2020). Predictive big data analytics for supply chain demand forecasting: methods, applications, and research opportunities. *Journal of Big Data*, *7*(1), 53.

Shawkat, M., Badawi, M., El-ghamrawy, S., Arnous, R., & El-desoky, A. (2022). An optimized FP-growth algorithm for discovery of association rules. *The Journal of Supercomputing*, *78*(4), 5479-5506.

Srivastava, V., Kishore, S., & Dhingra, D. (2021). Technology and the future of customer experience. In *Crafting customer experience strategy* (pp. 91-116). Emerald Publishing Limited. https://doi.org/10.1108/978-1-83909-710-220211006

Varma, A., & Ray, S. (2023). Big Data and Analytics in Retailing: Transforming the Customer Experience.

Vukovic, D. B., Spitsina, L., Gribanova, E., Spitsin, V., & Lyzin, I. (2023). Predicting the performance of retail market firms: Regression and machine learning methods. *Mathematics*, *11*(8), 1916. https://doi.org/10.3390/math11081916

Wei, Y., Tran, S., Xu, S., Kang, B., & Springer, M. (2020). Deep learning for retail product recognition: Challenges and techniques. *Computational intelligence and neuroscience*, *2020*(1), 8875910. https://doi.org/10.1155/2020/8875910