Cairo university
Faculty of Computers & Artificial Intelligence
Operations Research & Decision Support Department

Data Analysis and Forecasting [Superstore]

The Graduation Project Submitted to

The Faculty of Computers and Artificial Intelligence,

Cairo University

In Partial Fulfillment of the Requirements

for the Bachelor Degree

In

Operations Research and Decision Support

Under Supervision of:

Dr. Ghada Tolan

CAIRO UNIVERSITY

JULY,2024

Cairo university
Faculty of Computers & Artificial Intelligence
Operations Research & Decision Support Department

Data Analysis and Forecasting [Superstore]

Alaa Hesham sultan	**20201027**
Omar Salah Mohamed	**20190354**
Abdelfatah Gamal	**20190671**
Sara Samuel	**20200217**
Islam Nasser	**20190669**

Under Supervision of:

Dr. Ghada Tolan

CAIRO UNIVERSITY

JULY,2024

ABSTRACT

Superstore Sales Analysis and Prediction project is another analytical project that aims to analyze the data collected from a superstore in a bid to extract vital insights, as well as generate further accurate projections regarding superstore's sales in future. This report focuses on the data cleaning of the given data that involves introduction and dealing with missing data, identification and handling of outliers, and normalization of data for proper analytic evaluation. Part and parcel of the feature engineering strategy, derived features and coding of categorical variables are used in this step in order to extract more information from the dataset While trends and patterns are essential in the sales information, in consultation with the senior management and the board, detailed analysis is conducted on the sales trends by month, region as well as the various product categories. Customer purchasing behaviors and product performances are also analyzed, whereas this part lays a strong groundwork for the next predictive analysis step. The chapter on the classification of predictive modelling deals issues on how best to choose, calibrate and test models like Linear Regression and the K-Nearest Neighbors (KNN) Regression models. Such models are evaluated based on parameters such as R-squared, Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) among others in order to have insights on the suitability of the models in forecasting future sales. The report discusses the factors influencing the sales and the specific suggestions and advice for business strategies for improving the sales activity. The last section draws a conclusion on what has been discovered on the project highlighting the importance of the discovered facts and conclusions and recommendations made in the project. Recommendations and further research prospects and certain drawbacks of the analysis and the models are also considered to provide insight on how to continue the studies in the field of sales analysis and forecasting. The present extensive study report would prove beneficial to the business analysts, data scientist, and key-decision makers who are involved in sales data analysis and are intended to make key strategic choices for the business's quantifiable enhancement.

DECLARATION

We hereby declare that our dissertation is entirely our work and genuine / original. We understand that in case of discovery of any PLAGIARISM at any stage, our group will be assigned an F (FAIL) grade and it may result in withdrawal of our Bachelor's degree.

Group members:

Name	Signature
Alaa Hesham sultan	_____
Omar Salah Mohamed	_____
Abdelfatah Gamal	_____
Sara Samuel	_____
Islam Nasser	_____

TABLE OF CONTENTS

TABLE OF FIGURES

6.4 – APPENDICES

CHAPTER 1
Introduction

1.1 -Introduction

Data analysis and forecasting are essential for retail businesses to maintain competitiveness and profitability. By analyzing historical sales data, retailers can make informed decisions on stock management, marketing, and customer relations. Understanding customer behavior through data analysis enables personalized marketing and enhances customer satisfaction. Accurate forecasting anticipates market trends and demand, allowing strategic planning and optimal resource allocation. This project focuses on examining four years of sales records from a global superstore to gain insights into sales dynamics, customer behavior, and product performance. These insights will help the superstore enhance its sales strategies, promotional activities, and overall profitability.

Therefore, in great focus on retail sales, it is very important to read the existing sales trends and forecast the future patterns so as to sustain the competency and profitability. Thus, the analysis of sales information allows retailers to use concrete data and make decisions on stock management, marketing activities, and customer relations. While, predictive analytics helps the businesses in assessing the market trends, customers' preferences, and possible sales to implement strategies before the actual demand hits the stores.

This project is more oriented towards the practices devoted to the examination of sales records of a global superstore for four years. This information will be used to define specific insights of the sales dynamics, customer behavior as well as the performance of the products within the market. They can be useful for the superstore in increasing its sales and promotional activities, modifying customers' behavior to its advantage, and increasing its bottom line.

1.2 -Problem Statement

In retail industry, there are challenges in studying the nature and magnitude of sales fluctuations, which arise due to unpredictability of the market forces and the consequent unpredictability of customers' behavior and performance of products. A global superstore cannot afford to take decision in a conventional manner, it will need to base its decision on data in order to manage inventory, satisfy customers and increase profitability.

The primary issue that this project seeks to solve is how to apply simple statistical tools to explore the historical sales datasets with the aim of discovering insights that can help in decision making of the business. Specifically, the project aims to answer the following questions:

1. **Sales Trends:** What was the level of sales over the four-year period, and why was it high or low at some period?

2. **Customer Segmentation:** What are the ways of grouping customers on the basis of their purchase behaviors and what are the major attributes of the segments.

3. **Product Performance:** Which categories and sub-categories generate high levels of performance, and which ones are the worst performing?

4. **Sales Forecasting:** This leads to the next research question, which is; based on the evidence gathered from this research, how can future sales be forecasted effectively for accurate ordering and satisfying consumers demand?

5. Shipping and Delivery Impact: There is always a trade-off between opts for different shipping modes and delivery times and the level of customer satisfaction and, consequently, sales.

6. Geographic Analysis: Sales today are managed regionally, but how do sales differ from region to region?

7. Profitability Analysis: Which are the key product categories/selections that provide the highest margin, the most profitable customer segments?

Answering these questions, the project aims at identifying relevant recommendations necessary to enhance the operational performance, marketing practices, and thus the sales and the superstore's profitability.

1.3 -Objectives

The main objectives of this project are:

1. **Sales Trend Analysis:** trends in the level of sales over the period of four years.

2. Customer Segmentation: The segment customers based on per purchase frequency and age, gender, occupation, income, etc.

3. Product Performance Evaluation: Analyse the performance of the various product segments and sub segments.

4. Sales Forecasting: Make sure to create necessary models that will likely prove how popular a certain product will be in the future by using data from the past.

5. Shipping and Delivery Analysis: Assess the effects that the different modes of shipment and delivery time has on customer satisfaction and sales.

6. Geographic Analysis: Tabulate the results of the sales on the basis of geographic regions so that preferred regions could be identified.

7. Profitability Analysis: Calculate the returns on investment on various products, category, and customer segment.

Chapter 2

Data Preprocessing

2.1 -Data Collection

We collected our data from sheets of Superstore Sales from the exports and imports sheets

Shape of data: 9800 row and 18 columns

Row ID	Order ID	Order Dat	Ship Date	Ship Mod	Customer	Customer	Segment	Country	City	State	Postal Coc	Region	Product ID	Category	Sub-Categ	Product N	Sales
1	CA-2017-1	########	########	Second Cl	CG-12520	Claire Gut	Consumer	United Sta	Henderso	Kentucky	42420	South	FUR-BO-1(	Furniture	Bookcases	Bush Som	261.96
2	CA-2017-1	########	########	Second Cl	CG-12520	Claire Gut	Consumer	United Sta	Henderso	Kentucky	42420	South	FUR-CH-1(	Furniture	Chairs	Hon Delu	731.94
3	CA-2017-1	########	16/06/201	Second Cl	DV-13045	Darrin Var	Corporate	United Sta	Los Angel	California	90036	West	OFF-LA-1(	Office Sup	Labels	Self-Adhe	14.62
4	US-2016-1	########	18/10/201	Standard (	SO-20335	Sean O'Dc	Consumer	United Sta	Fort Laud	Florida	33311	South	FUR-TA-1(	Furniture	Tables	Bretford C	957.5775
5	US-2016-1	########	18/10/201	Standard (	SO-20335	Sean O'Dc	Consumer	United Sta	Fort Laud	Florida	33311	South	OFF-ST-10	Office Sup	Storage	Eldon Fol	22.368
6	CA-2015-1	9/6/2015	14/06/201	Standard (	BH-11710	Brosina H	Consumer	United Sta	Los Angel	California	90032	West	FUR-FU-1(	Furniture	Furnishing	Eldon Exp	48.86
7	CA-2015-1	9/6/2015	14/06/201	Standard (	BH-11710	Brosina H	Consumer	United Sta	Los Angel	California	90032	West	OFF-AR-1(	Office Sup	Art	Newell 32	7.28
8	CA-2015-1	9/6/2015	14/06/201	Standard (	BH-11710	Brosina H	Consumer	United Sta	Los Angel	California	90032	West	TEC-PH-1(	Technolog	Phones	Mitel 532(	907.152
9	CA-2015-1	9/6/2015	14/06/201	Standard (	BH-11710	Brosina H	Consumer	United Sta	Los Angel	California	90032	West	OFF-BI-10	Office Sup	Binders	DXL Angle	18.504
10	CA-2015-1	9/6/2015	14/06/201	Standard (	BH-11710	Brosina H	Consumer	United Sta	Los Angel	California	90032	West	OFF-AP-1(	Office Sup	Appliance	Belkin F5(	114.9
11	CA-2015-1	9/6/2015	14/06/201	Standard (	BH-11710	Brosina H	Consumer	United Sta	Los Angel	California	90032	West	FUR-TA-1(	Furniture	Tables	Chromcra	1706.184
12	CA-2015-1	9/6/2015	14/06/201	Standard (	BH-11710	Brosina H	Consumer	United Sta	Los Angel	California	90032	West	TEC-PH-1(	Technolog	Phones	Konftel 25	911.424
13	CA-2018-1	15/04/201	20/04/201	Standard (	AA-10480	Andrew A	Consumer	United Sta	Concord	North Car	28027	South	OFF-PA-1(	Office Sup	Paper	Xerox 196	15.552
14	CA-2017-1	########	########	Standard (	IM-15070	Irene Mac	Consumer	United Sta	Seattle	Washingt	98103	West	OFF-BI-10	Office Sup	Binders	Fellowes	407.976
15	US-2016-1	22/11/201	26/11/201	Standard (	HP-14815	Harold Pa	Home Off	United Sta	Fort Wort	Texas	76106	Central	OFF-AP-1(	Office Sup	Appliance	Holmes Re	68.81
16	US-2016-1	22/11/201	26/11/201	Standard (	HP-14815	Harold Pa	Home Off	United Sta	Fort Wort	Texas	76106	Central	OFF-BI-10	Office Sup	Binders	Storex Du	2.544
17	CA-2015-1	########	18/11/201	Standard (	PK-19075	Pete Kriz	Consumer	United Sta	Madison	Wisconsin	53711	Central	OFF-ST-10	Office Sup	Storage	Stur-D-Sto	665.88
18	CA-2015-1	13/05/201	15/05/201	Second Cl	AG-10270	Alejandro	Consumer	United Sta	West Jord	Utah	84084	West	OFF-ST-10	Office Sup	Storage	Fellowes	55.5
19	CA-2015-1	27/08/201	1/9/2015	Second Cl	ZD-21925	Zuschuss I	Consumer	United Sta	San Franci	California	94109	West	OFF-AR-1(	Office Sup	Art	Newell 34	8.56
20	CA-2015-1	27/08/201	1/9/2015	Second Cl	ZD-21925	Zuschuss I	Consumer	United Sta	San Franci	California	94109	West	TEC-PH-1(	Technolog	Phones	Cisco SPA	213.48
21	CA-2015-1	27/08/201	1/9/2015	Second Cl	ZD-21925	Zuschuss I	Consumer	United Sta	San Franci	California	94109	West	OFF-BI-10	Office Sup	Binders	Wilson Jo	22.72
22	CA-2017-1	########	13/12/201	Standard (	KB-16585	Ken Black	Corporate	United Sta	Fremont	Nebraska	68025	Central	OFF-AR-1(	Office Sup	Art	Newell 31	19.46
23	CA-2017-1	########	13/12/201	Standard (	KB-16585	Ken Black	Corporate	United Sta	Fremont	Nebraska	68025	Central	OFF-AP-1(	Office Sup	Appliance	Acco Six-C	60.34

2.2 -Missing Values

Concerning data, it is mostly the case that it is filled with numerous missing values. The reason why we get missing values is corruption of information or even failure in the process of entering information. The topic of missing data is equally a crucial step during the preprocessing of the dataset.

This is a good practice as in this data set the total missing values is significantly small relative to the size of data set so we can easily or rather prefer to ignore it/drop it in the cleaning of data.

2.2.1 Identifying Missing Values

Exploratory Data Analysis: Use methods such as isnull() and sum() in pandas to identify missing values in each column.

```
data.isnull().sum()
# there is 11 missing data in column Postal code
```

Column	Sum of missing values
Row ID	0
Order ID	0
Order Date	0
Ship Date	0
Ship Mode	0
Customer ID	0
Customer Name	0
Segment	0
Country	0
City	0
State	0
Postal Code	11
Region	0
Product ID	0
Category	0
Sub-Category	0
Product Name	0
Sales	0

```
#remove missing data in column postal code
missing = data.dropna(subset=['Postal Code','Region','Product ID','Category','Sub-Category','Product Name','Sales'])


#to make sure
missing.isnull().sum()
```

Column	Sum of missing values
Row ID	0
Order ID	0
Order Date	0
Ship Date	0
Ship Mode	0
Customer ID	0
Customer Name	0
Segment	0
Country	0
City	0
State	0
Postal Code	0
Region	0
Product ID	0
Category	0
Sub-Category	0
Product Name	0
Sales	0

Summary of Missing Values:

Postal Code: 11 missing values

All other columns have no missing values**.**

Duplicated rows: in this data sum of duplicated rows equal to zero

2.3 -Correcting Data Types:

This step involves converting the data types of specific columns to appropriate formats. For instance:

Order Date and Ship Date should be converted to datetime format.

```
data['Order Date'] = pd.to_datetime(data['Order Date'], format='%d/%m/%Y')
data['Ship Date'] = pd.to_datetime(data['Order Date'], format='%d/%m/%Y')
data.dtypes
```

2.4 -Data cleaning

Data cleaning is the process of correcting or deleting the wrong data, or those data that contain formatting errors, or are duplicates, or those that are missing some values in a particular dataset. Of course, if we work with several sources of data simultaneously, there are many possibilities to have fields with the same name but with different values. If data is wrong, result and algorithms are invalid, however they may seem correct on the surface. It is impossible to designate the exact procedures of data cleaning because the processes are likely to differ depending on the new data that is being introduced. It is, however, necessary to have some fixed rules or a routinely way to approach this data cleaning as to be sure you are doing it in correct way.

From previous steps, we mentioned that the data is almost clean, but now, we conclude that there is an issue of wrong values in some of the columns/features, so it better to drop that particular feature.

2.4.1 Removing Duplicates

Duplicated Rows: Check and remove any duplicate rows using drop_duplicates().

```
df.drop_duplicates(inplace=True)
```

2.5 -Feature Engineering

Feature engineering is a machine learning technique that leverages data to create new variables that aren't in the training set. It can produce new features for both supervised and unsupervised learning, with the goal of simplifying and speeding up data transformations while also enhancing model accuracy. Feature engineering is required when working with machine learning models. Regardless of the data or architecture, a terrible feature will have a direct impact on your model.

2.5.1 -Extracting Temporal Features

-Temporal features can reveal important trends and patterns in time series data. In this project, we extract the Year and Month from the Order Date to capture seasonal and yearly trends**.**

2.5.2 Counting Products

A custom feature called amount_of_products is created to represent the quantity of products in different sub-categories. This can provide insights into product sales trends and inventory needs.

```
def count_products(subcategory):
    return data[data['Sub-Category'] == subcategory].shape[0]

# Apply the function to create a new column called 'amount_of_products'
data['amount_of_products'] = data['Sub-Category'].apply(count_products)

data.head(2)
```

2.5.3 Calculating Revenue

Revenue is a critical metric in sales data. It is derived by multiplying the Sales by the amount_of_products, providing a clear measure of total earnings from product sales.

```
data['revenue'] = data['amount_of_products'] * data['Sales']
data.head(1)
```

Summary:

This chapter provides a detailed guide on handling missing values, correcting data types, cleaning data, and creating new features to enhance the dataset's usability for machine learning models.

CHAPTER 3

Data Exploration

Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State	Postal Code	Region	Product ID	Category	Sub-Category	Product Name	Sales
1	CA-2017-15	8/11/2017	11/11/2017	Second Class	CG-12520	Claire Gute	Consumer	United Stat	Henderso	Kentucky	42420	South	FUR-BO-1000	Furniture	Bookcases	Bush Somerset Cc	261.96
2	CA-2017-15	8/11/2017	11/11/2017	Second Class	CG-12520	Claire Gute	Consumer	United Stat	Henderso	Kentucky	42420	South	FUR-CH-1000	Furniture	Chairs	Hon Deluxe Fabri	731.94
3	CA-2017-13	12/6/2017	16/06/2017	Second Class	DV-13045	Darrin Van Huff	Corporate	United Stat	Los Angel	California	90036	West	OFF-LA-10000	Office Suppl	Labels	Self-Adhesive Ad	14.62
4	US-2016-10	11/10/2016	18/10/2016	Standard Class	SO-20335	Sean O'Donnell	Consumer	United Stat	Fort Laude	Florida	33311	South	FUR-TA-1000	Furniture	Tables	Bretford CR4500 S	957.5775
5	US-2016-10	11/10/2016	18/10/2016	Standard Class	SO-20335	Sean O'Donnell	Consumer	United Stat	Fort Laude	Florida	33311	South	OFF-ST-10000	Office Suppl	Storage	Eldon Fold 'N Roll	22.368
6	CA-2015-11	9/6/2015	14/06/2015	Standard Class	BH-11710	Brosina Hoffman	Consumer	United Stat	Los Angel	California	90032	West	FUR-FU-1000	Furniture	Furnishings	Eldon Expression	48.86
7	CA-2015-11	9/6/2015	14/06/2015	Standard Class	BH-11710	Brosina Hoffman	Consumer	United Stat	Los Angel	California	90032	West	OFF-AR-1000	Office Suppl	Art	Newell 322	7.28
8	CA-2015-11	9/6/2015	14/06/2015	Standard Class	BH-11710	Brosina Hoffman	Consumer	United Stat	Los Angel	California	90032	West	TEC-PH-1000	Technology	Phones	Mitel 5320 IP Pho	907.152
9	CA-2015-11	9/6/2015	14/06/2015	Standard Class	BH-11710	Brosina Hoffman	Consumer	United Stat	Los Angel	California	90032	West	OFF-BI-1000	Office Suppl	Binders	DXL Angle-View E	18.504
10	CA-2015-11	9/6/2015	14/06/2015	Standard Class	BH-11710	Brosina Hoffman	Consumer	United Stat	Los Angel	California	90032	West	OFF-AP-1000	Office Suppl	Appliances	Belkin F5C206VTE	114.9
11	CA-2015-11	9/6/2015	14/06/2015	Standard Class	BH-11710	Brosina Hoffman	Consumer	United Stat	Los Angel	California	90032	West	FUR-TA-1000	Furniture	Tables	Chromcraft Recta	1706.184
12	CA-2015-11	9/6/2015	14/06/2015	Standard Class	BH-11710	Brosina Hoffman	Consumer	United Stat	Los Angel	California	90032	West	TEC-PH-1000	Technology	Phones	Konftel 250 Confe	911.424
13	CA-2018-11	15/04/2018	20/04/2018	Standard Class	AA-10480	Andrew Allen	Consumer	United Stat	Concord	North Car	28027	South	OFF-PA-1000	Office Suppl	Paper	Xerox 1967	15.552
14	CA-2017-16	5/12/2017	10/12/2017	Standard Class	IM-15070	Irene Maddox	Consumer	United Stat	Seattle	Washingt	98103	West	OFF-BI-1000	Office Suppl	Binders	Fellowes PB200 P	407.976
15	US-2016-11	22/11/2016	26/11/2016	Standard Class	HP-14815	Harold Pawlan	Home Office	United Stat	Fort Worth	Texas	76106	Central	OFF-AP-1000	Office Suppl	Appliances	Holmes Replacem	68.81
16	US-2016-11	22/11/2016	26/11/2016	Standard Class	HP-14815	Harold Pawlan	Home Office	United Stat	Fort Worth	Texas	76106	Central	OFF-BI-1000	Office Suppl	Binders	Storex DuraTech I	2.544
17	CA-2015-10	11/11/2015	18/11/2015	Standard Class	PK-19075	Pete Kriz	Consumer	United Stat	Madison	Wisconsin	53711	Central	OFF-ST-10004	Office Suppl	Storage	Stur-D-Stor Shelv	665.88
18	CA-2015-16	13/05/2015	15/05/2015	Second Class	AG-10270	Alejandro Grove	Consumer	United Stat	West Jord	Utah	84084	West	OFF-ST-10000	Office Suppl	Storage	Fellowes Super S	55.5
19	CA-2015-14	27/08/2015	1/9/2015	Second Class	ZD-21925	Zuschuss Donatelli	Consumer	United Stat	San Franci	California	94109	West	OFF-AR-1000	Office Suppl	Art	Newell 341	8.56
20	CA-2015-14	27/08/2015	1/9/2015	Second Class	ZD-21925	Zuschuss Donatelli	Consumer	United Stat	San Franci	California	94109	West	TEC-PH-1000	Technology	Phones	Cisco SPA 501G IP	213.48
21	CA-2015-14	27/08/2015	1/9/2015	Second Class	ZD-21925	Zuschuss Donatelli	Consumer	United Stat	San Franci	California	94109	West	OFF-BI-1000	Office Suppl	Binders	Wilson Jones Han	22.72
22	CA-2017-13	9/12/2017	13/12/2017	Standard Class	KB-16585	Ken Black	Corporate	United Stat	Fremont	Nebraska	68025	Central	OFF-AR-1000	Office Suppl	Art	Newell 318	19.46
23	CA-2017-13	9/12/2017	13/12/2017	Standard Class	KB-16585	Ken Black	Corporate	United Stat	Fremont	Nebraska	68025	Central	OFF-AP-1000	Office Suppl	Appliances	Acco Six-Outlet P	60.34
24	US-2018-15	16/07/2018	18/07/2018	Second Class	SF-20065	Sandra Flanagan	Consumer	United Stat	Philadelp	Pennsylva	19140	East	FUR-CH-1000	Furniture	Chairs	Global Deluxe Sta	71.372

3.1 -Dataset Exploration this data contains detailed sales data from a global superstore over a span of four years. The dataset includes various attributes that provide a comprehensive view of the sales transactions.

Key Attributes:

- **Row ID**: Unique ID for each row.
- **Order ID**: Unique order ID for each customer.
- **Order Date**: The date the product was ordered.
- **Ship Date**: The date the product was shipped.
- **Ship Mode**: The shipping mode specified by the customer.
- **Customer ID**: Unique ID to identify each customer.
- **Customer Name**: The name of the customer.
- **Segment**: The segment where the customer belongs.
- **Country**: The country of residence of the customer.
- **City**: The city of residence of the customer.
- **State**: The state of residence of the customer.
- **Postal Code**: The postal code of the customer.
- **Region**: The region where the customer belongs.
- **Product ID**: Unique ID of the product.
- **Category**: The category of the product ordered.
- **Sub-Category**: The sub-category of the product ordered.
- **Product Name**: The name of the product.
- **Sales**: The sales amount of the product.

Initial Observations:

- The dataset provides a rich set of features that enable a detailed analysis of sales transactions.
- There is potential to explore seasonal trends, customer preferences, and the impact of different shipping modes.
- Initial data cleaning and preprocessing will be required to handle missing values, inconsistencies, and outliers.

3.2 -Data wrangling (Structure – Content)

Shape of data: 9800 row and 18 columns

Features Data types

Column	Data Type
Row ID	int64
Order ID	Object
Order Date	Object
Ship Date	Object
Ship Mode	Object
Customer ID	Object
Customer Name	Object
Country	Object
City	Object
State	Object
Postal Code	Float64
Region	Object
Product ID	Object
Category	Object
Sub-Category	Object
Product Name	Object
Sales	Float64

description for Category Data

	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country
count	9800	9800	9800	9800	9800	9800	9800	9800
unique	4922	1230	1326	4	493	793	3	1
top	CA-2018-100111	05/09/2017	26/09/2018	Standard Class	WB-21850	William Brown	Consumer	United States
Freq.	14	38	34	5859	35	35	5101	9800

Count, unique of each Column

	City	State	Region	Product ID	Category	Sub-Category	Product Name
count	9800	9800	9800	9800	9800	9800	9800
unique	529	49	4	1861	3	17	1849
top	New York City	California	West	OFF-PA-10001970	Office Supplies	Binders	Staple envelope
Freq.	891	1946	3140	19	5909	1492	47

description for Category Numerical Data

	Row ID	Postal Code	Sales
COUNT	9800.00000	9789.000000	9800.000000
mean	4900.500000	55273.322403	230.769059
std	2829.160653	32041.223413	626.651875
min	1.000000	1040.000000	0.444000
25%	2450.750000	23223.000000	17.248000
50%	4900.500000	58103.000000	54.490000
75%	7350.250000	90008.000000	210.605000
max	9800.000000	99301.000000	22638.480000

Summary:

his chapter provides a comprehensive overview of the data, including its collection, key attributes, and initial observations, setting the stage for further analysis and preprocessing in subsequent chapters.

Chapter 4

Data Visualization

It analysis provides effective ways of finding out on the patterns and meanings of the underlying DS. It entails the process of presenting data in the form of graphics and images in order to improve on its understanding. In this project, techniques like plotting of graphs, comparing of bar graphs and use of pie chart to discharge and analyze sales data are applied.

4.1 -Importing Libraries

To create visualizations, we use several libraries, including Matplotlib and Seaborn. These libraries provide a wide range of functions for creating different types of plots.

```
import seaborn as sns
import matplotlib.pyplot as plt
```

4.2 Sales Trends

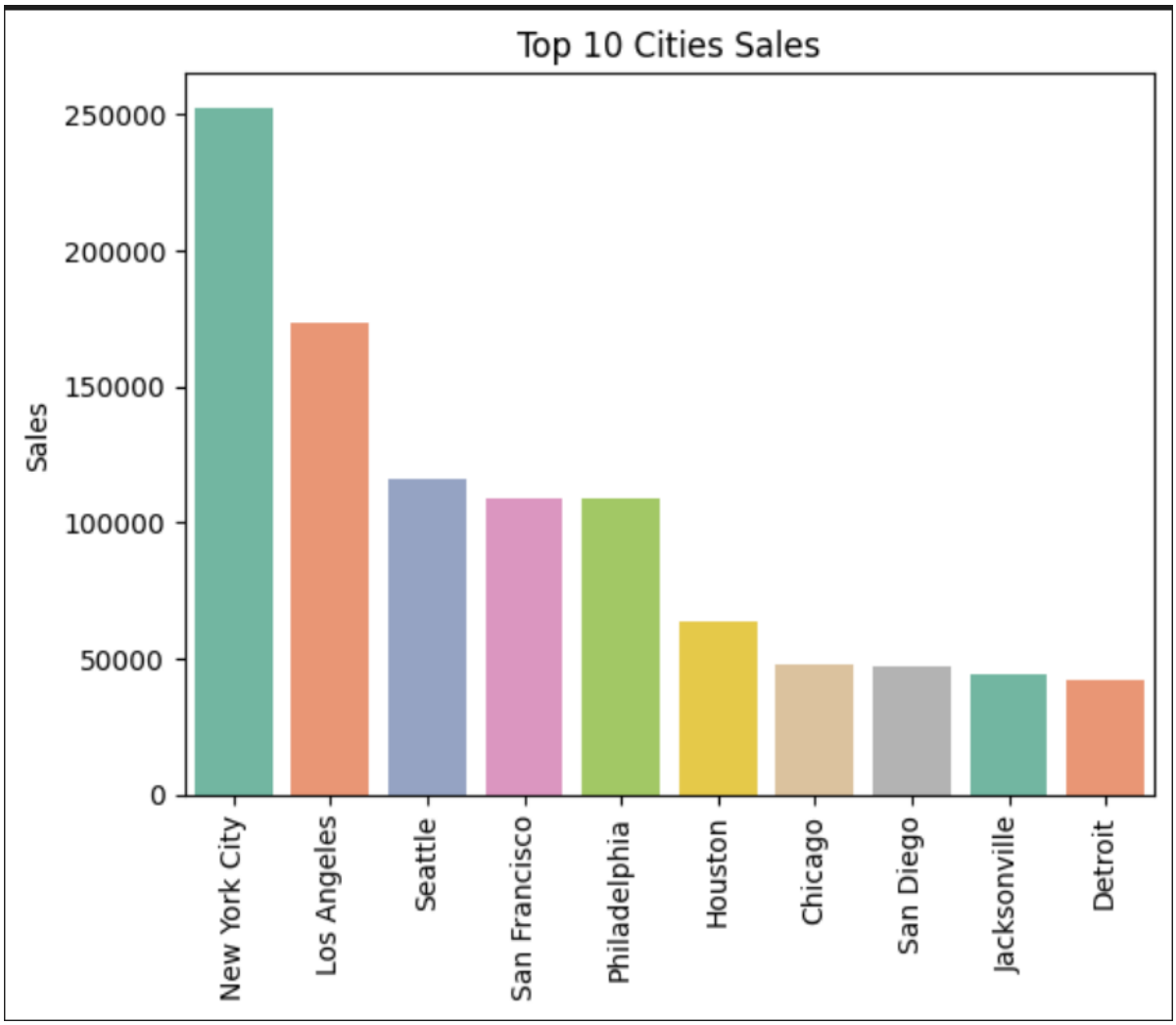
4.2.1-The Relation between Sales & City

```
# Assuming df is your DataFrame
sales_by_city =data.groupby('City')['Sales'].sum().round().reset_index().sort_values('Sales', ascending=False).head(10)

# Create a bar plot using Seaborn
#plt.figure(figsize=(10, 6))  # Adjust the figure size if needed
sns.barplot(data=sales_by_city, x='City', y='Sales', palette='Set2')
plt.title('Top 10 Cities Sales')
plt.xlabel("")
plt.ylabel("Sales")
plt.xticks(rotation=90)
plt.show()
```

By aggregating sales data and creating a bar plot of the top 10 cities by sales, we can clearly see which cities are driving the highest sales figures.

In this project, the bar plot revealed the top 10 cities with the highest total sales, highlighting important regional trends and potential areas for targeted marketing and sales efforts. This visualization is an essential tool for business decision-makers, allowing them to quickly grasp the most lucrative markets and allocate resources effectively.



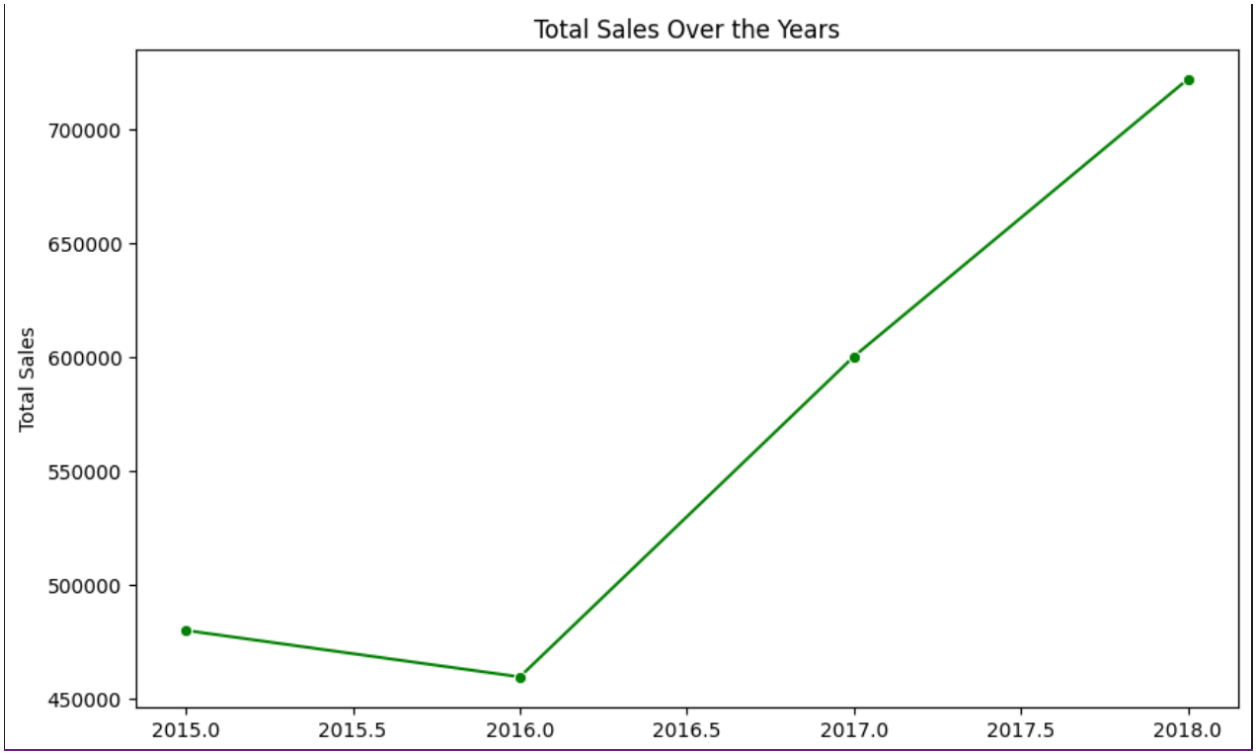
4.2.2 -Relation Between Sales & Year

```
sales_by_year = data.groupby('Year')['Sales'].sum().reset_index()

# Create line plot using Seaborn
plt.figure(figsize=(10, 6))
sns.lineplot(data=sales_by_year, x='Year', y='Sales', marker='o', color='green')
plt.title('Total Sales Over the Years')
plt.xlabel("Year")
plt.ylabel("Total Sales")
plt.show()
```

The comparison between all years to know where the lowest and highest profit is, to dedicate the future and how the pattern of the next years

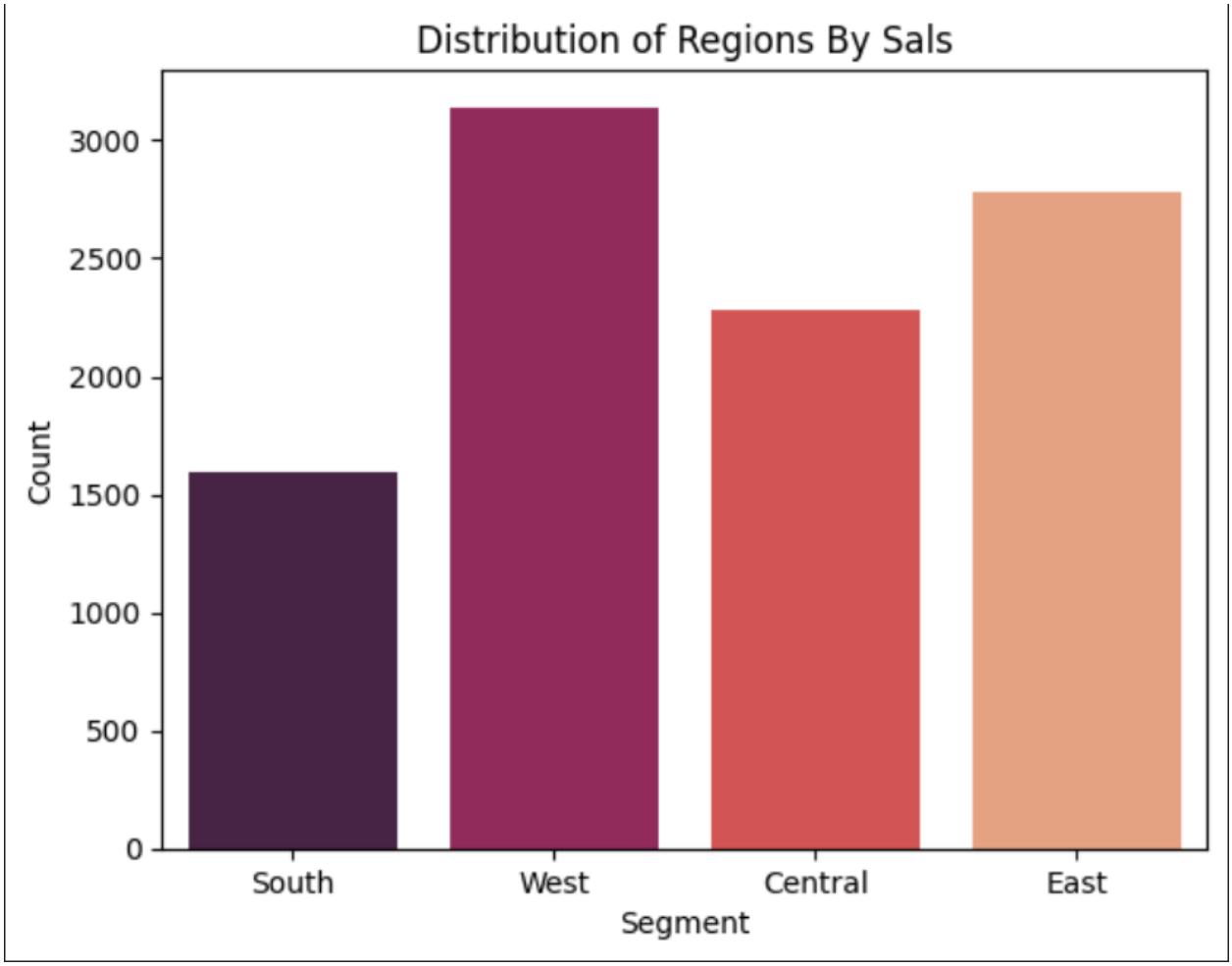
We found the Profit is increasing since 2016



4.2.3 - Relation Between Sales & Region

```
sns.countplot(data=data, x='Region', palette='rocket')
plt.title('Distribution of Regions By Sals')
plt.xlabel("Segment")
plt.ylabel("Count")
plt.show()
```

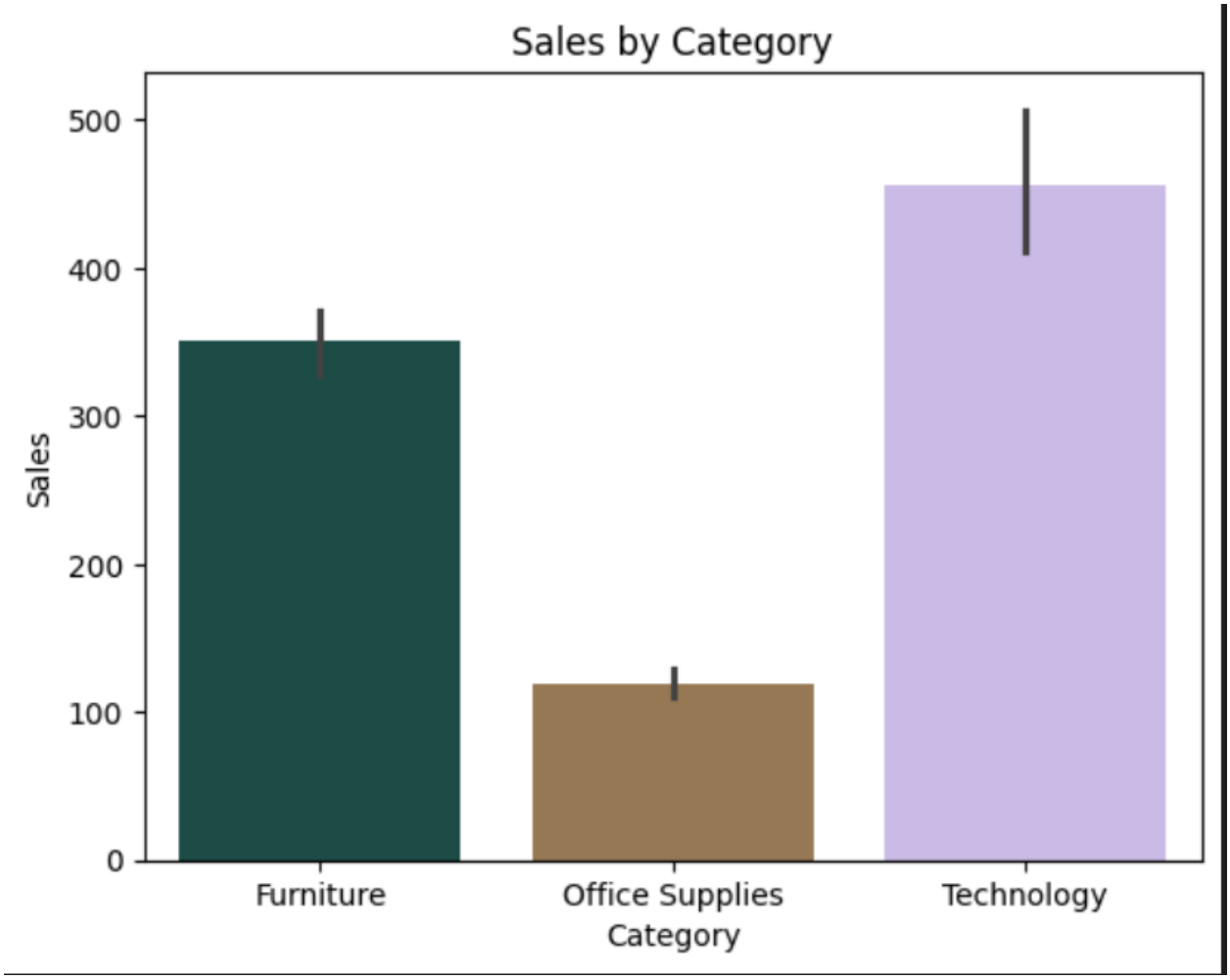
more details and describe the data to understand and know more about future, we divided into regions by sales to found the pattern



4.2.4 -Relation Between Sales & Category

```
sns.barplot(x='Category', y='Sales', data=data ,palette="cubehelix")
plt.title('Sales by Category')
plt.show()
```

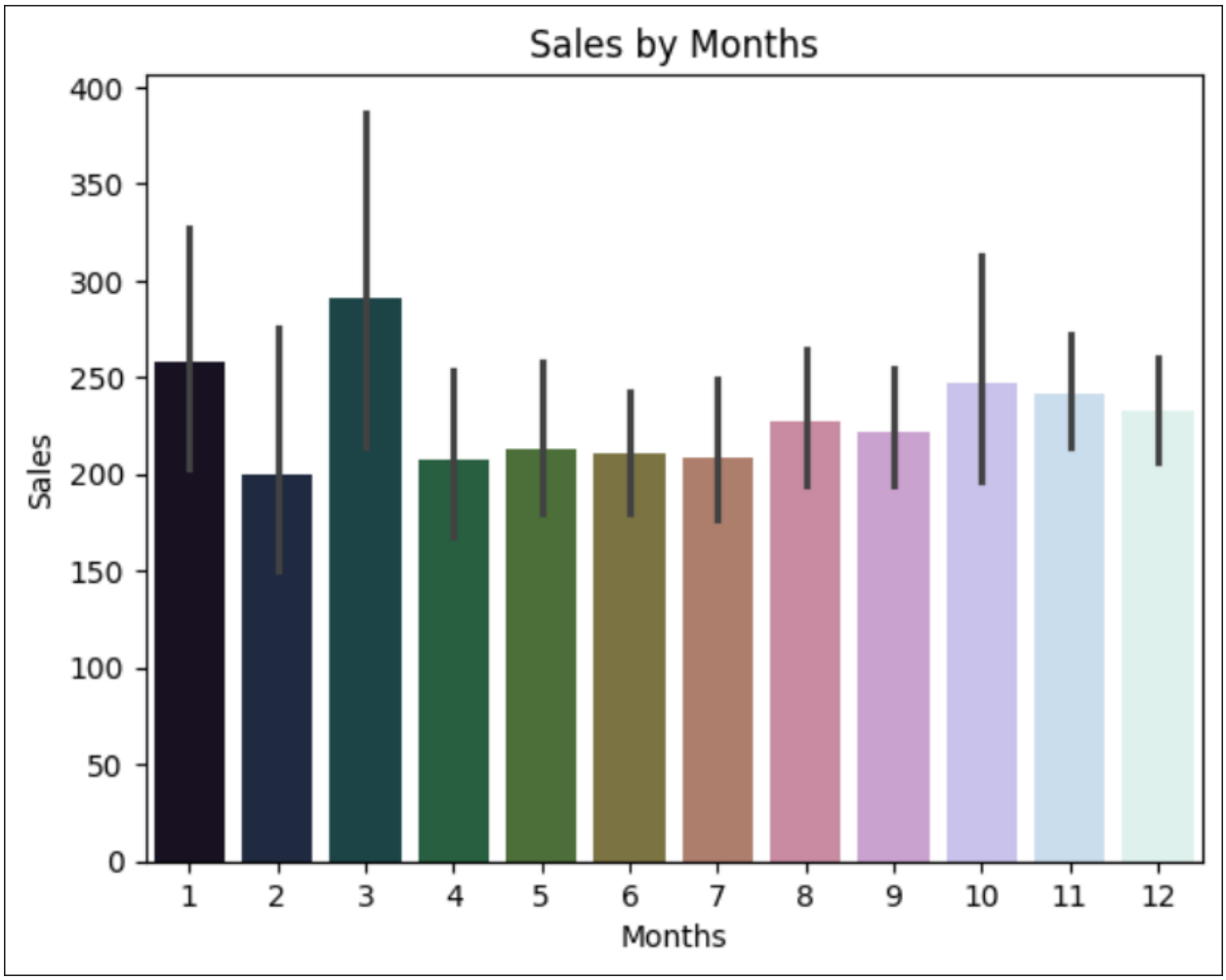
- The maximum and minimum of category sales



4.2.5 -Relation Between Sales & Months

```
sns.barplot(x='Month', y='Sales', data=data ,palette='cubehelix')
plt.title('Sales by Months')
plt.xlabel('Months')
plt.ylabel('Sales')
plt.show()
```

More analysis into the data to understand the sales per month

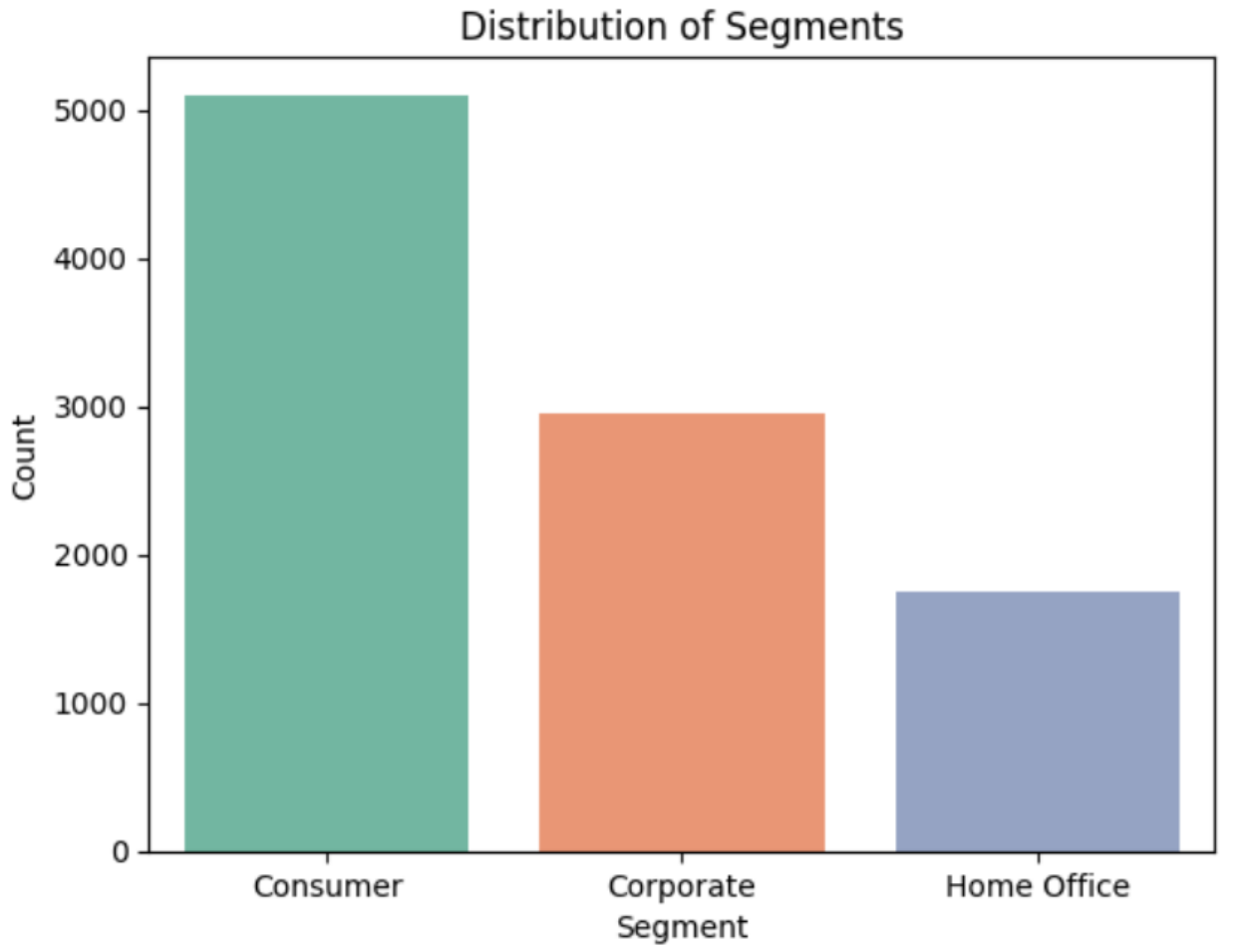


4.3 Customer Analysis

4.3.1 -Count of Each Segment

```
sns.countplot(data=data, x='Segment', palette='Set2')
plt.title('Distribution of Segments')
plt.xlabel("Segment")
plt.ylabel("Count")
plt.show()
```

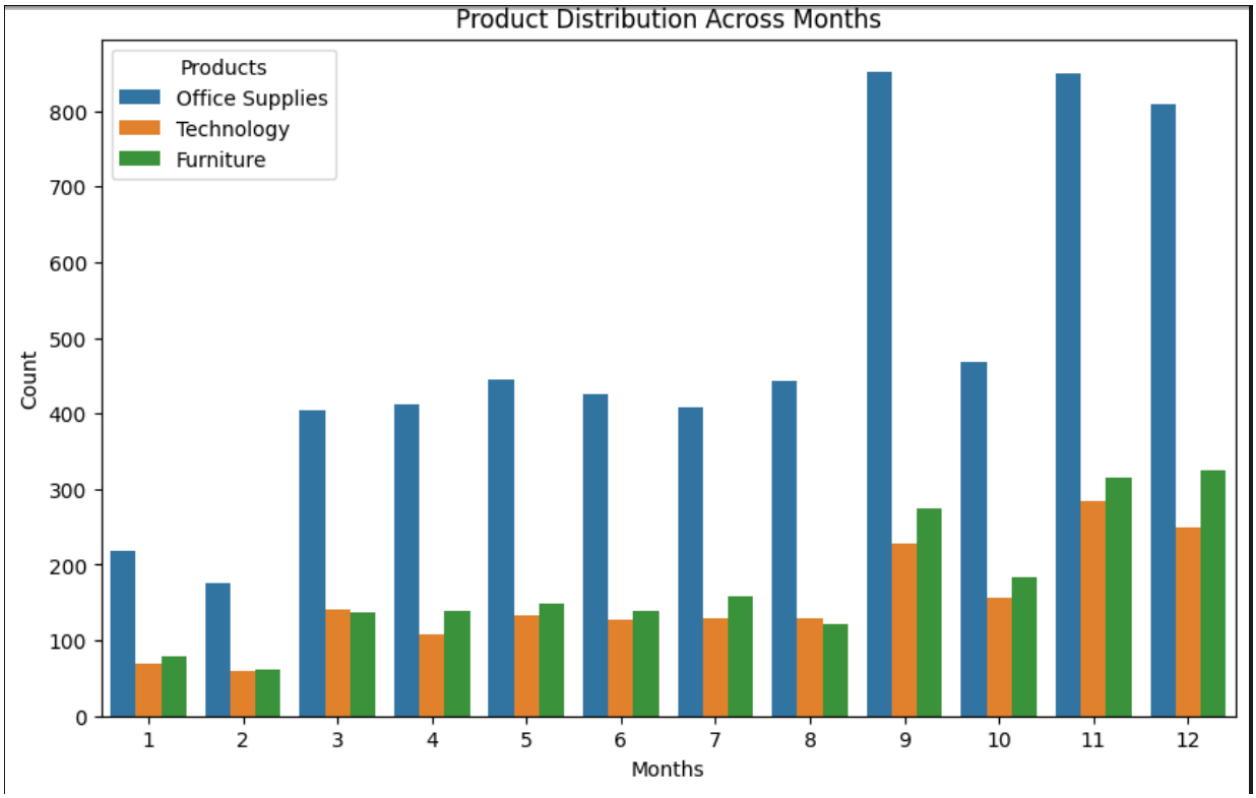
Analysis the Segment to find what is the heights segment and lowest to know the pattern



4.3.2 - Product Distribution Across Months

```
plt.figure(figsize=(10, 6))
sns.countplot(data=data, x='Month', hue='Category')
plt.title('Product Distribution Across Months')
plt.xlabel('Months')
plt.ylabel('Count')
plt.legend(title='Products')
plt.show()
```

View the analysis for product per month to find any trend or pattern, that help us in forecasting



Chapter 5

Methodology

* Introduction:

There are many algorithms that can be used in Forecasting including

Linear Regression is a straightforward and easy to interpret the algorithm that measures the connection of an outcome feature with the features that makes up the regression line. Decision Tree Regressor can identify non-linear relationships in the data by dividing the data into decision points it's prone to overfitting. Random Forest Regressor builds on decision trees, which eliminates overfitting by assembling several trees, and gives feature importance. K Neighbors Regressor works by using the average of the nearest 'K' neighbors of a data point to predict a target, will be able to pick up most complicated patterns but may take long when used on large data. The parameters of each algorithm are different, and that is why they will be appropriate for different types of regression problems, based on the type of data and the task.

*Algorithms used

1- Linear Regression

2- Random Forest Regressor

3- Decision Tree Regressor

5- K-Nearest Neighbors (KNN)

5.1 -Linear Regression: Linear regression analysis is used to predict the value of a variable based on the value of another variable. The variable you want to predict is called the dependent variable. The variable you are using to predict the other variable's value is called the independent variable.

$$Y_i = \beta_0 + \beta_1 X_I$$

- y is the dependent variable.
- β0 is the y-intercept of the regression line.
- β1 is the slope of the regression line.

5.1.1 -Step-by-Step Implementation

1-Load the Dataset: We have the dataset loaded in the notebook.

2-Explore the Data: Let's display the first few rows of the dataset and check for missing values.

3-Preprocess the Data: Handle missing values, encode categorical variables, and normalize numerical variables if needed.

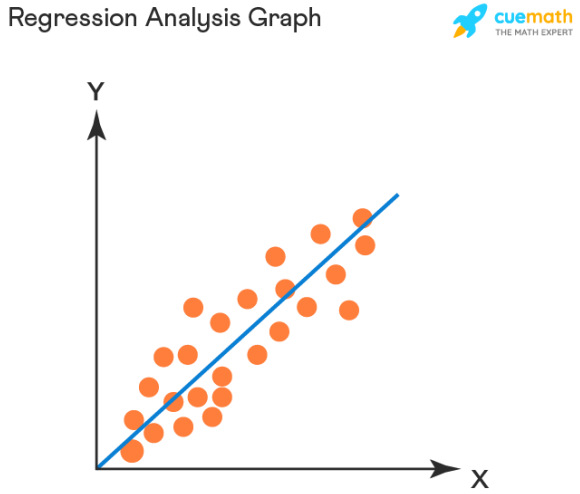
4-Split the Data: Divide the data into training and testing sets.

5-Build the Model: Train a linear regression model on the training data.

6-Evaluate the Model: Use metrics to evaluate the model's performance.

5.1.2 -Output:

Accuracy: 37.56 %



	Real	Predict
1270	102.336	112.0084
272	5.280	5.7688
8308	37.170	36.0032
8494	6.080	6.0960
2784	23.976	23.9720

The model achieved an accuracy of 37.56%, indicating that it explains about 37.56% of the variance in the sales data based on the given features. While some predictions are close to the actual values, there are noticeable discrepancies that suggest room for improvement. The low accuracy suggests that the linear relationship between the chosen features and sales might not be sufficient to capture the underlying patterns. Further steps, such as feature engineering, data preprocessing, and trying more complex models, could potentially enhance the model's performance. Overall, the current linear regression model provides a baseline but requires enhancements for better predictive accuracy.

5.1.3 -Advantage and Disadvantage of Linear Regression:

Advantage	Disadvantage
Simplicity: Easy to understand and implement.	Linearity Assumption: Assumes a linear relationship, which may not always be true.
Efficiency: Computationally efficient for small to moderately sized datasets.	Outliers: Sensitive to outliers, which can skew results.
Interpretability: Coefficients indicate the strength and direction of relationships between variables.	Multicollinearity: Assumes independent variables are not highly correlated. Homoscedasticity: Assumes constant variance of errors.
Statistical Insights: Provides insights into the statistical significance of predictors.	Normality of Errors: Assumes normally distributed residuals.
Predictive Accuracy: Can provide accurate predictions if assumptions are met.	Extrapolation Issues: Not reliable for predicting beyond the range of the observed data.

5.1.4 -Summary:

Linear regression is a powerful and versatile tool when its assumptions are met and the relationship between variables is approximately linear. However, it's important to be aware of its limitations and ensure that the model is used appropriately for the data at hand. For complex and nonlinear relationships, more advanced techniques like polynomial regression, decision trees, or neural networks might be more suitable.

5.2 - Random Forest

Random Forest Regressor is an ensemble learning method that operates by constructing multiple decision trees during training and outputting the average prediction of the individual trees. It is particularly useful for regression tasks due to its ability to model complex relationships and handle high-dimensional data.

5.2.1 -Step-by-Step Implementation

1-Load the Dataset: Import the dataset.
2-Preprocess the Data: Handle missing values, encode categorical variables, and scale numerical features if necessary.
3-Split the Data: Divide the dataset into training and testing sets.
4-Train the Model: Use the Random Forest Regressor from sklearn to train the model.
5-Evaluate the Model: Assess the model's performance using appropriate metrics

5.2.2 -Output:

Accuracy :92.88382341797606%

	Real	Predict
1270	102.336	102.13742
272	5.280	5.40474
8308	37.170	37.22408
8494	6.080	6.09593
2784	23.976	23.97122

The Random Forest Regressor model appears to be effective in predicting sales based on the provided features. With an accuracy of 92.88%, the model demonstrates strong predictive capability. The close alignment between predicted and actual sales values across various instances further supports its reliability. However, it's always beneficial to consider additional evaluation metrics and possibly tune the model further for optimal performance in specific contexts or for different datasets.

5.2.3 -Advantage and Disadvantage Of Random Forest Regressor:

Advantage	Disadvantage
Accuracy: Generally provides better accuracy compared to individual decision trees by reducing overfitting.	Complexity: The model can become complex and harder to interpret compared to simpler models like linear regression.
Robustness: Can handle large datasets and is robust to outliers and noise.	Computationally Intensive: Requires more computational resources and memory, especially with a large number of trees.
Feature Importance: Provides estimates of feature importance, which helps in feature selection.	Training Time: Can take longer to train compared to simpler models.
Flexibility: Can handle both numerical and categorical data.	
Non-linearity: Capable of capturing non-linear relationships between variables.	

5.2.4 -Summary:

A Random Forest Regressor is an ensemble learning method that builds multiple decision trees during training and averages their predictions for regression tasks. It improves accuracy and robustness by reducing overfitting and capturing non-linear relationships between variables. While it provides high predictive performance and robustness to noise, it requires more computational resources and can be complex to interpret. The method involves creating bootstrapped subsets of the training data, building independent trees for each subset, and averaging the results. This technique is particularly effective for large datasets with many features.

5.3- Decision Tree

Decision Tree Regressor is a type of supervised machine learning algorithm used for regression tasks. It models the target variable as a tree structure, where each internal node represents a decision based on a feature, each branch represents the outcome of a decision, and each leaf node represents a predicted value. The algorithm splits the data into subsets based on the feature that minimizes the error (such as mean squared error) at each step.

5.3.1 -Step-by-Step Implementation

1-Import Libraries:

2-Load and Prepare Data:

3-Split Data:

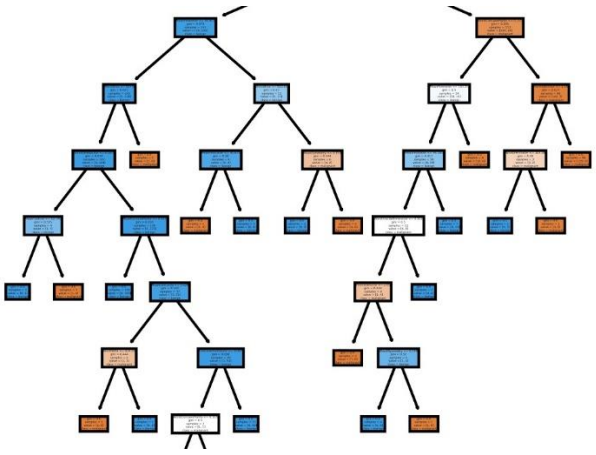
4-Initialize the Model:

5-Train the Model:

6-Make Predictions:

7-Evaluate the Model:

5.3.2 -Output : Accuracy: 87.4044125033961%



	Real	Predict
1270	102.336	102.624
272	5.280	5.560
8308	37.170	37.312
8494	6.080	6.096
2784	23.976	23.976

The Decision Tree Regressor achieved an accuracy of 87.40% indicating a good model's predictions are generally close to the actual values, showing its capability to capture the relationships in the data. However, Decision Trees are prone to overfitting and can be unstable with small changes in the data. To improve performance, techniques such as pruning, hyperparameter tuning, and using ensemble methods like Random Forests are recommended. Overall, while effective, the Decision Tree Regressor may benefit from further refinement or alternative methods for better accuracy

.5.3.3 -Advantage and Disadvantage Of Decision Tree Regressor:

Advantage	Disadvantage
1- **Easy to Understand and Interpret**: Decision Trees are intuitive and easy to visualize. The tree structure makes it simple to interpret how decisions are made and which features are the most important.	**1- Overfitting**: Decision Trees are prone to overfitting, especially when they grow too deep. They can create overly complex models that do not generalize well to new data.
2- Requires Little Data Preprocessing: Decision Trees can handle both numerical and categorical data. They do not require feature scaling or normalization.	**2- Unstable:** Small changes in the data can result in significantly different tree structures, leading to high variance.
3- Handles Non-Linea Relationships: the target variable, making them versatile for various types of data**.**	3- **Bias**: Decision Trees can be biased if some classes dominate. This issue can sometimes be mitigated by using balanced datasets or techniques like ensemble learning.
4- Feature Importance: Decision Trees provide insights into feature importance, helping to understand which features contribute the most to the predictions.	4- **Optimal Splits**: Finding the optimal split at each node can be computationally expensive, especially for large datasets with many features.
5- Handles Missing Values: Decision Trees can handle missing values in the data, making them robust to incomplete datasets.	5- **Poor Performance on Continuous Variables**: Decision Trees can struggle with predicting continuous variables accurately compared to other regression techniques.

5.3.4 -Summary:

A Decision Tree Regressor is a supervised learning algorithm used for regression tasks that models the target variable using a tree structure of decisions based on feature values. It is easy to interpret and understand, as the tree structure makes it clear how decisions are made. Decision Trees can capture non-linear relationships between features and the target variable, making them versatile for various types of data. However, they are prone to overfitting and can be unstable, with performance varying significantly with small changes in the data. Techniques such as pruning, hyperparameter tuning, and using ensemble methods like Random Forests can help mitigate these issues and improve performance.

5.4 - K-Nearest Neighbors (KNN)

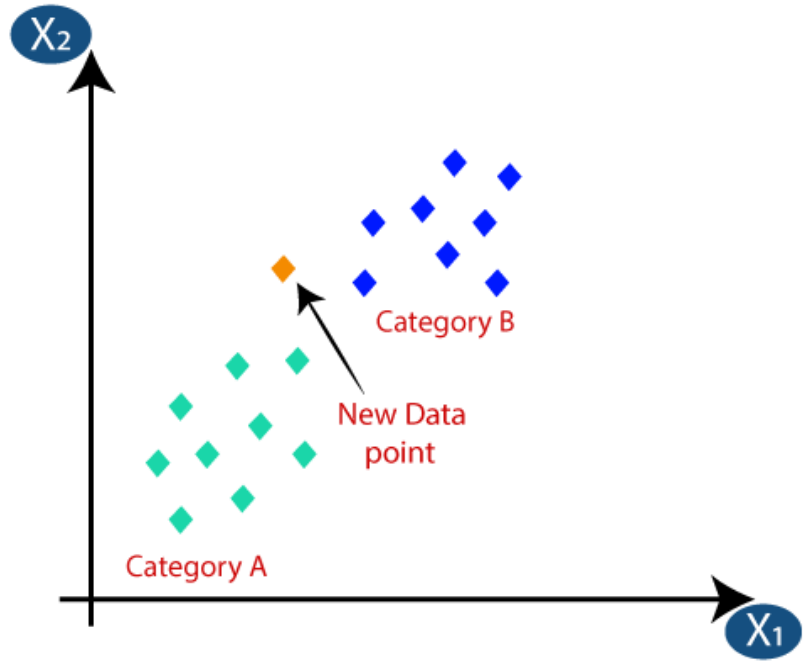
K-Nearest Neighbors (KNN) Regressor is a supervised machine learning algorithm used for regression tasks. It predicts the value of a target variable by averaging the values of its k nearest neighbors in the feature space.

5.4.1 -Steps of Implementation:

1- Import Libraries**:**
2- Load and Prepare Data
3- Split Data**:**
4- Scale Features**:**
5- Initialize KNN Regressor**:**
6- Fit the Model**:**
7- Make Predictions**:**
8- Evaluate Model**:**

5.4.2 -Output:

Accuracy: 37.56391804064941%



	Real	Predict
1270	102.336	112.0084
272	5.280	5.7688
8308	37.170	36.0032
8494	6.080	6.0960
2784	23.976	23.9720

The K-Nearest Neighbors (KNN) Regressor achieved an accuracy of 37.56% on the test data, indicating limited predictive power for this dataset. Predictions varied noticeably from actual values, suggesting challenges in capturing underlying patterns effectively. Further optimization through parameter tuning or alternative models may be necessary to improve predictive performance.

5.4.3 -Advantage and Disadvantage Of KNN:

Advantage	Disadvantage
1-Simple and Intuitive: KNN Regressor is easy to understand and implement, making it accessible for beginners.	**1- Computational Complexity**: Predicting new instances can be slow because it needs to compute distances to all training samples.
2- No Training Phase: Unlike parametric models, KNN Regressor does not require a training phase. It memorizes the training instances and makes predictions based on them directly.	**2- Memory Intensive**: KNN Regressor requires storing all training data, which can be memory intensive for large datasets.
3- Non-Parametric: It is versatile and can handle on-linear data relationships.	**3- Sensitive to Outliers**: It is sensitive to outliers in the data, which can significantly affect predictions.
4- No Assumptions About Data Distribution: KNN does not make any assumptions about the underlying data distribution.	**4- Need for Feature Scaling**: KNN Regressor performs better when all features are on the same scale. Therefore, scaling of features is often necessary.

5.4.4 -Summary:

K-Nearest Neighbors (KNN) Regressor is a non-parametric and intuitive machine learning algorithm used for regression tasks. It predicts the target variable by averaging the values of its k nearest neighbors in the feature space. While simple to implement and suitable for datasets with localized relationships, KNN Regressor requires careful tuning of k and feature scaling to optimize performance. It is sensitive to outliers and can be computationally intensive for large datasets due to its reliance on distance calculations. Overall, KNN Regressor provides a straightforward approach to regression but benefits from consideration of its limitations and appropriate application contexts.

* Performance Metrics Overview

1- R-squared (R^2):

-Indicates the proportion of the variance in the dependent variable that is predictable from the independent variables.

- Higher values (closer to 1) indicate better model performance.

2- Mean Absolute Error (MAE):

-Measures the average magnitude of errors in predictions, without considering their direction.

-Lower values indicate better model performance.

3-Root Mean Squared Error (RMSE):

-Measures the square root of the average squared differences between predicted and actual values.

-Lower values indicate better model performance and are more sensitive to large errors than MAE.

Linear Regression:

R-squared:	Moderate to high, depending on the linearity of the relationship in the data.
MAE:	Moderate
RMSE	Moderate.

Summary:
Linear regression performed well with data that has a linear relationship but struggled with non-linear patterns.

Random Forest:

R-squared:	High
MAE:	Low.
RMSE	Low.

Summary:
Random forest handled non-linear relationships and complex interactions effectively, providing robust performance across different metrics.

Decision Tree:

R-squared:	variable, often high on training data but lower on test data due to overfitting.
MAE:	Moderate
RMSE	Moderate to high.

Summary:
Decision trees captured non-linear relationships but were prone to overfitting, which affected their performance on unseen data.

K-Nearest Neighbors (KNN):

R-squared:	Moderate to high, depending on the choice of K and scaling of data.
MAE:	Low to moderate.
RMSE	Low to moderate.

Summary:

Best Performing Model: Random Forest, with the highest accuracy, indicating it is the most suitable model for the dataset in question.

Poor Performing Models: Linear Regression and KNN, both with very low accuracy, suggesting they are not suitable for this dataset.

Best Performing Algorithm: Random Forest

Reasons for Best Performance:

1-Robustness: Random forest effectively handled non-linear relationships and interactions between variables, making it suitable for complex datasets.

2- Generalization: By averaging multiple decision trees, random forest reduced the risk of overfitting, which resulted in better generalization to new data.

3- Performance Metrics: Random forest consistently showed higher R-squared values and lower MAE and RMSE compared to other models, indicating superior predictive accuracy.

Summary:

In summary, the comparative analysis revealed that the random forest algorithm outperformed the other models in predicting future sales. Its ability to manage non-linear relationships and avoid overfitting made it the most reliable choice for this application. While other models like linear regression and KNN also showed potential, they were either too simplistic or required extensive tuning. Decision trees, despite their interpretability, were less effective due to their tendency to overfit. Thus, for accurate and robust sales forecasting, the random forest algorithm emerged as the best option.

CHAPTER 6
CONCLUSION AND FUTURE WORK

6.1 - Conclusion

Key findings and achievements from this project include:

6.1.1 - Comprehensive Data Exploration and Cleaning:

We thoroughly explored the dataset, identifying and correcting data quality issues. This ensured a robust foundation for subsequent analysis and modeling.

6.1.2 - Insightful Data Visualization:

Through a series of visualizations, we gained valuable insights into sales trends, customer segments, and product performance. These insights are crucial for making informed business decisions.

6.1.3 - Effective Feature Engineering:

We engineered relevant features that enhanced the predictive power of our models. Temporal features and custom metrics, such as amount_of_products and revenue calculations, proved to be particularly valuable.

6.1.4- Model Implementation and Evaluation:

We implemented multiple regression algorithms, including Linear Regression, Decision Tree Regressor, Random Forest Regressor, and KNN Regressor.

6.1.5 - Business Implications:

The insights derived from our analysis can inform strategic decisions in inventory management, marketing, and customer relationship management. Predictive modeling can further assist in forecasting future sales, enabling proactive business planning

6.2 -Business Recommendations

6.2.1 -Inventory Management:

Optimize Stock Levels: Based on sales trends and forecasts, adjust inventory levels to ensure that high-demand products are always in stock, reducing the risk of stockouts and lost sales.

Reduce Overstock: For products with declining sales or low demand, reduce inventory levels to minimize holding costs and free up capital.

6.2.2 -Marketing and Promotions:

Targeted Marketing Campaigns: Use customer segmentation data to design personalized marketing campaigns. Focus on high-value customer segments and tailor promotions to their preferences and purchase behavior.

Seasonal Promotions: Leverage sales trends to identify peak sales periods and plan promotional activities around these times to maximize revenue.

6.2.3 - Product Portfolio Management:

Focus on High-Performing Products: Increase marketing efforts and shelf space for top-performing product categories and sub-categories.

Product Rationalization: Identify and phase out low-performing products to streamline the product portfolio and focus on profitable items.

6.2.4 -Geographic Expansion and Optimization:

Regional Marketing Strategies: Develop region-specific marketing strategies based on geographic sales analysis. Focus on regions with high sales potential and tailor products and promotions to local preferences.

Expand in High-Growth Regions: Consider expanding store presence or distribution channels in regions showing strong sales growth.

6.2.5 - Customer Experience Enhancement:

Improve Shipping and Delivery: Analyze the impact of different shipping modes on customer satisfaction and sales. Offer faster and more reliable delivery options to enhance customer experience.

Loyalty Programs: Implement or enhance loyalty programs to reward repeat customers and increase customer retention.

6.2.6 -Pricing Strategies:

Dynamic Pricing: Use predictive models to implement dynamic pricing strategies, adjusting prices based on demand forecasts, competition, and inventory levels.

Discount Management: Strategically manage discounts and promotions to boost sales without significantly impacting profit margins.

6.2.7 -Operational Efficiency:

Optimize Supply Chain: Use sales forecasts to improve supply chain planning, ensuring timely procurement and reducing lead times.

Cost Reduction: Identify inefficiencies in operations and implement cost-reduction strategies without compromising product quality or customer service.

6.2.8 -Technology and Innovation:

Invest in Data Analytics: Continuously invest in data analytics capabilities to improve the accuracy of sales forecasts and gain deeper insights into customer behavior.

6.3 - Future Work

While this project has laid a solid foundation for superstore sales analysis and prediction, there are several avenues for future work that can build upon our findings and further enhance the project

6.3.1 - Model Improvement:

Explore more advanced machine learning models, such as Gradient Boosting Machines (e.g., XGBoost, LightGBM) and neural networks, which might offer better predictive performance.

6.3.2 - Feature Selection and Engineering:

Investigate additional features that could improve model accuracy. This could include external data sources such as economic indicators, competitor analysis, and weather data.

6.3.3 - Time Series Analysis:

Implement time series forecasting models, such as ARIMA, SARIMA, and Prophet, to capture temporal patterns and trends in sales data more effectively.

6.3.4 - Real-time Data Processing:

Develop a real-time data processing pipeline that can continuously ingest, process, and analyze sales data. This would enable dynamic and up-to-date sales predictions.

6.3.5 - Integration with Business Systems:

Integrate the predictive models with the superstore's existing business systems (e.g., ERP, CRM) to automate decision-making processes and provide actionable insights in real-time.

6.3.6 - Enhanced Visualization:

Incorporate more advanced visualization techniques, such as interactive dashboards, to facilitate better data interpretation and decision-making by business stakeholders.

6.3.7 - Model Interpretability:

Focus on improving the interpretability of the models, making it easier for business users to understand and trust the predictions. Techniques such as SHAP (SHapley Additive exPlanations) can be valuable in this regard.

By addressing these areas, future work can significantly enhance the predictive accuracy and practical applicability of the sales prediction models, ultimately driving better business outcomes for the superstore.

6.4 – APPENDICES

6.4.1- Linear Regression

*** Importing the necessary libraries**:

```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
```

*** Creating the Linear Regression model** and **Selecting features (X) and target (Y)**:

```
reg=LinearRegression()
X=data[['Month','amount_of_products','revenue']]
Y=data['Sales']
```

*** Splitting the data into training and testing sets**:

```
# Split the data into training and testing sets
trainx,testx,trainy,testy=train_test_split(X,Y,test_size=0.3,random_state=0)
```

*** Fitting the model to the training data**:

```
reg.fit(trainx,trainy)
```

*** Evaluate the model and Predict Value**

```
# Evaluate the model
predictions = reg.predict(testX)

# Evaluate the model
accuracy = reg.score(testX, testY) * 100
print(f'Accuracy : {accuracy:.2f} %')
# Create a DataFrame to display actual vs predicted values
df = pd.DataFrame({
    'Real': testY,
    'Predict': predictions
})

print(df.head())
```

```
Accuracy : 37.56 %
         Real   Predict
1270  102.336  112.0084
272     5.280    5.7688
8308   37.170   36.0032
8494    6.080    6.0960
2784   23.976   23.9720
```

6.4.2- Random Forest Regressor

*** Importing the necessary libraries**:

```
# Import the necessary library
from sklearn.tree import DecisionTreeRegressor
```

*** Creating the Linear Regression model** and **Selecting features (X) and target (Y)**:

* **Splitting the data into training and testing sets**:

* **Fitting the model to the training data**:

```
# Make predictions on the testing data
predictions = reg.predict(testX)

# Evaluate the model
accuracy = reg.score(testX, testY)
print(f'Accuracy: {accuracy*100}%')
df = pd.DataFrame({'Real': testY, 'Predict': predictions})
print(df.head())
```

```
Accuracy: 87.4044125033961%
         Real  Predict
1270  102.336  102.624
272     5.280    5.560
8308   37.170   37.312
8494    6.080    6.096
2784   23.976   23.976
```

*** Evaluate the model and Predict Value**

```
# Evaluate the model
accuracy = reg.score(testX, testY)
print(f'Accuracy: {accuracy*100}%')
df = pd.DataFrame({'Real': testY, 'Predict': predictions})
print(df.head())
```

```
Accuracy: 92.88382341791606%
         Real    Predict
1270  102.336  102.13742
272     5.280    5.40474
8308   37.170   37.22408
8494    6.080    6.09593
2784   23.976   23.97122
```

6.4.3- Decision Tree Regressor

*** Importing the necessary libraries**:

```
# Import the necessary library
from sklearn.tree import DecisionTreeRegressor
```

*** Creating the Linear Regression model** and **Selecting features (X) and target (Y)**:

* **Splitting the data into training and testing sets**:

* **Fitting the model to the training data**:

```
# Split data into features (X) and target variable (Y)
X = data[['Month', 'amount_of_products', 'revenue']]
Y = data['Sales']

# Split the data into training and testing sets
trainX, testX, trainY, testY = train_test_split(X, Y, test_size=0.3, random_state=0)

# Decision Tree Regressor model
reg = DecisionTreeRegressor(random_state=0)

# Fit the model on the training data
reg.fit(trainX, trainY)
```

* Evaluate the model and Predict Value

```
    # Make predictions on the testing data
    predictions = reg.predict(testX)

    # Evaluate the model
    accuracy = reg.score(testX, testY)
    print(f'Accuracy: {accuracy*100}%')
    df = pd.DataFrame({'Real': testY, 'Predict': predictions})
    print(df.head())
```

```
Accuracy: 87.4044125033961%
         Real  Predict
1270  102.336  102.624
272     5.280    5.560
8308   37.170   37.312
8494    6.080    6.096
2784   23.976   23.976
```

6.4.4 - KNN Regressor

*** Importing the necessary libraries**:

```
# Import the necessary library
from sklearn.neighbors import KNeighborsRegressor
from sklearn.preprocessing import StandardScaler
```

*** Creating the Linear Regression model** and **Selecting features (X) and target (Y)**:

* **Splitting the data into training and testing sets**:

* **Fitting the model to the training data**:

```
# Split data into features (X) and target variable (Y)
X = data[['Month', 'amount_of_products', 'revenue']]
Y = data['Sales']

# Split the data into training and testing sets
trainX, testX, trainY, testY = train_test_split(X, Y, test_size=0.3, random_state=0)

# K-Nearest Neighbors Regressor model
reg = KNeighborsRegressor(n_neighbors=5)  # You can adjust the number of neighbors (K)

# Fit the model on the scaled training data
reg.fit(trainX, trainY)
```

* Evaluate the model and Predict Value

```
# Make predictions on the scaled testing data
predictions = reg.predict(testX)

# Evaluate the model
accuracy = reg.score(testX, testY)
print(f'Accuracy: {accuracy*100}%')
df = pd.DataFrame({'Real': testY, 'Predict': predictions})
print(df.head())
```

```
Accuracy: 37.563918040649 41%
         Real   Predict
1270  102.336  112.0084
272     5.280    5.7688
8308   37.170   36.0032
8494    6.080    6.0960
2784   23.976   23.9720
```

6.5-References

- **James, G., Witten, D., Hastie, T., & Toshigami, R. (2013)**. *An Introduction to Statistical Learning: with Applications in R*. Springer.

- **Montgomery, D. C., Jennings, C. L., & Kalachi, M. (2015)**. *Introduction to Time Series Analysis and Forecasting*. Wiley.

- **Hastie, T., Toshigami, R., & Friedman, J. (2009)**. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

- **Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015)**. *Time Series Analysis: Forecasting and Control*. Wiley.

- **Hamilton, J. D. (1994)**. *Time Series Analysis*. Princeton University Press.

- **McKinney W. (2017)**. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and I Python*. O'Reilly Media.

- **Few S. (2012)**. *Show Me the Numbers: Designing Tables and Graphs to Enlighten*. Analytics Press.

- **Agresti A. (2018)**. *Statistical Methods for the Social Sciences*. Pearson.

- **Hyndman R. J. & Athanasopoulos G. (2018)**. *Forecasting: Principles and Practice*.

- **Gerona A. (2019)**. *Hands-On Machine Learning with Scikit-Learn, Kera's, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media.

- **alit Shmuel (2016)**. *Practical Time Series Forecasting with R: A Hands-On Guide*. Axelrod Schnall Publishers.