



KNOW YOUR CUSTOMER

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

Asmaa Adel Omar 20180460

Mohamed Essam Galal 20180231

Mohamed Hany khariy 20180242

Rech Raymondo Malek 20180402

Under Supervision of:

Dr. Doaa Saleh

CAIRO UNIVERSITY

February, 2023

ABSTRACT

This project centers on analyzing the customer behavior of an online retail store. The objective of this project is to perform an Exploratory Data Analysis (EDA) and Recency, Frequency, and Monetary (RFM) Analysis. Data cleaning and transformation will also be carried out. The programming language used in the analysis is Python, and the data set is loaded from a csv file. The analysis is done in Jupyter Notebook, Kaggle and Google Colab.

For this project, a dataset from a retail store containing over sixty thousand transactions was used. The EDA was carried out by checking for missing data and duplicates, and data cleaning was done by dropping missing values and excluding negative values. The RFM analysis was then performed, where recency, frequency, and monetary were calculated using specific formulas. The K-Means algorithm was employed to cluster the customers. The Davies-Bouldin score, the silhouette score and the Calinski-Harabasz score were used to evaluate the performance of K-Means algorithm. The data was then plotted, and normalization was carried out using the logarithm function. The plot showed the data distribution before and after normalization.

Tools used for development and testing include Streamlit, SilhouetteVisualizer and Matplotlib, which are all data visualization tools used in the analysis. Validations were carried out by checking the data for accuracy, completeness, and consistency. The K-Means algorithm was also used as a data segmentation tool. Achievements include understanding customer behavior through data analysis, making recommendations to improve customer experience, and improving revenue by understanding customer buying patterns.

In conclusion, the analysis provided a clear understanding of customer behavior in the online retail store and insights to make strategic decisions. The RFM analysis was crucial in understanding customer segmentation, and the normalization made it possible to determine customer behaviors in-depth. Future work would involve building a customer predictive model using machine learning algorithms to determine the likelihood of a customer buying again.

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

Asmaa Adel Omar

Mohamed Essam Galal

Mohamed Hany khariy

Rech Raymondo Malek

PLAIGRISM CERTIFICATE

This is to certify that the project entitled “Know Your Customer”, which is being submitted here with for the award of the “**Bachelor of Computers and Artificial Intelligence Degree**” in “**Operations Research and Decision Support**”. This is the result of the original work **Asmaa Adel Omar, Mohamed Essam Galal, Mohamed Hany khariy and Rech Raymondo Malek** under my supervision and guidance. The work embodied in this project has not been done earlier for the basis of award of any degree or compatible certificate or similar tile of this for any other diploma/examining body or university to the best of my knowledge and belief.

Turnitin Originality Report

Processed on

ID:

Word Count: 6895

Similarity Index

Similarity by Source

Internet Sources:

Publications:

Student Papers:

Date:

Dr. Doaa Saleh (Supervisor)

ACKNOWLEDGMENT

We would like to express our gratitude to all those who have contributed to the completion of this project. First and foremost, we would like to thank our supervisor for her guidance and support throughout the process. Additionally, we would like to thank the team members who worked tirelessly to collect, clean, and analyze the data. Finally, we would like to thank the open-source community for providing the tools and resources necessary to complete this project. Without the collective effort of all these individuals, this project would not have been possible.

TABLE OF CONTENTS

Table of Contents

ABSTRACT.....	II
DECLARATION	III
PLAIGRISM CERTIFICATE	IV
ACKNOWLEDGMENT.....	V
CHAPTER 1	1
INTRODUCTION	1
Problem Statement:	2
Objectives:	2
Resources:	2
Methodology:	2
Organization of Project Report:	4
CHAPTER 2	5
BACKGROUND/EXISTING WORK.....	5
Literature Review.....	5
Introduction.....	5
Importance of Data Cleaning	5
Methods of Data Cleaning	6
Impact of Data Cleaning on Data Quality and Analysis Results	6
Conclusion	7
Existing Work:	7
CHAPTER 3	10
Data Collection and Preprocessing.	10
Data Collection.	10
Preprocessing.	10

CHAPTER 4	14
Exploratory Data Analysis.....	14
CHAPTER 5	20
REGENCY, FREQUENCY, AND MONETARY ANALYSIS.....	20
CHAPTER 6	24
DATA VISUALIZATION.....	24
CHAPTER 6	26
APPLICATION OF THE ELBOW METHOD AND K-MEANS ALGORITHM, EVALUATE THE PERFORMANCE OF THE K-MEANS ALGORITHM.	26
CHAPTER 7	30
STP MODEL	30
Positioning	30
CHAPTER 7	32
RESULTS AND RECOMMENDATIONS.....	32
Results:.....	32
Recommendations:.....	32
CHAPTER 8	35
CONCLUSION AND FUTURE WORK	35
Conclusion	35
Future work.....	36
APPENDICES	37
Code listing:.....	37
Data dictionary:.....	52
REFERENCES	53

Table of Figures

Figure 1 RFM Loyalty Level	12
Figure 2 Box Plot.....	13
Figure 3 Box Plot.....	15
Figure 4 Distortion Score for Kmeans Clustering	16
Figure 5 Histogram	17
Figure 6 3D scatter plot.....	19
Figure 7 Pie-chart – Customers grouped by Country	24
Figure 8 Pie-chart - Revenue Across Countries.....	25
Figure 9 Line Graph.....	27
Figure 10 Silhouette plot of Kmeans	28
Figure 11 3D Scatter plot - Optimal number of clusters determined after K - Means algorithm was run again	29
Figure 12 Stp Model	31

CHAPTER 1

INTRODUCTION

Problem Statement:

The online retail industry has seen massive growth in recent years, and it is essential for businesses in this sector to understand customer behavior to improve their revenue and customer experience. This project aims to analyze customer behavior in an online retail store using EDA and RFM analysis, providing insights to make strategic decisions.

Objectives:

The main objectives of this project are:

- To perform EDA on the online retail store dataset, checking for missing data, duplicates, and performing data cleaning and transformation.
- To perform RFM analysis, calculating recency, frequency, and monetary using specific formulas.
- To plot the data and carry out normalization using logarithmic function.
- To apply the elbow method.
- To use the K-Means algorithm to cluster the data set.
- To gain insights into customer behavior, make recommendations to improve customer experience, and improve revenue by understanding customer buying patterns.
- To validate the data for accuracy, completeness, and consistency.

Resources:

The dataset used for this project is from UCI Machine Learning site and is of a retail store and contains over sixty thousand transactions. The programming language used for the analysis is Python, and tools such as Jupyter Notebook, Google Colab, Kaggle, Streamlit and Matplotlib will be used for development and testing.

Methodology:

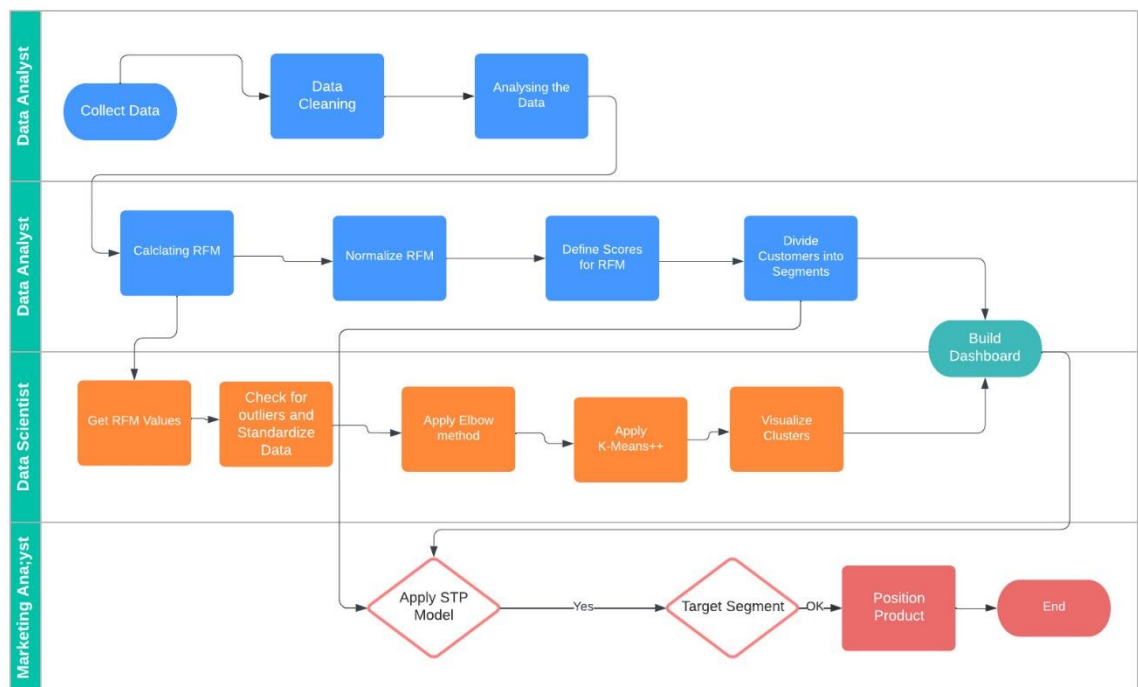
The project will follow a structured development methodology, with a focus on data cleaning, transformation, and analysis. The project report will be organized as follows:

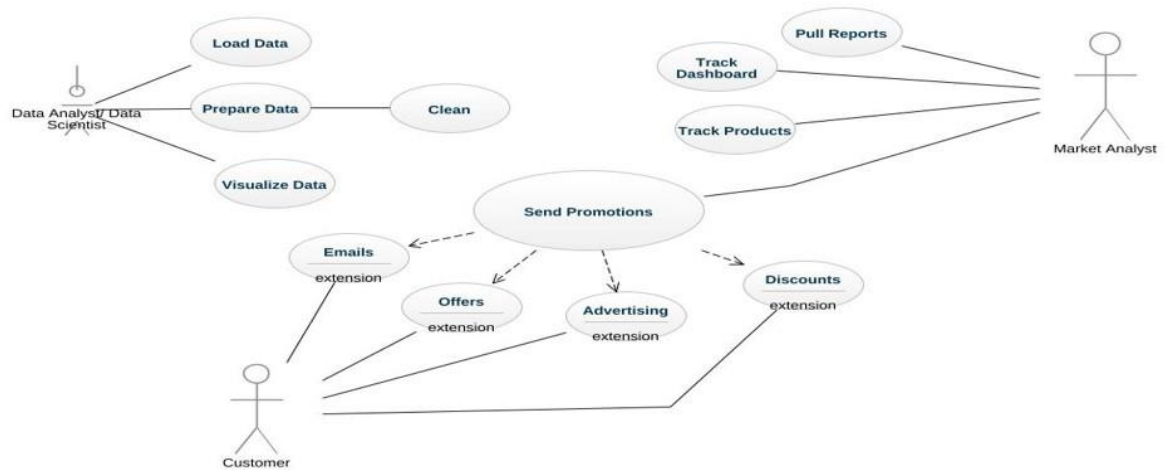
- Chapter 1: Introduction
- Chapter 2: Literature Review
- Chapter 3: Data Collection and Preprocessing
- Chapter 4: Exploratory Data Analysis
- Chapter 5: Recency, Frequency, and Monetary Analysis

- Chapter 6: Data Visualization
- Chapter 7: Application of the Elbow method and K-Means algorithm, evaluate the performance of the K-Means algorithm.
- Chapter 7: STP Model.
- Chapter 8: Results and Recommendations
- Chapter 9: Conclusion and Future Work

Business process flow example

Asmaa adel | February 25, 2023





Organization of Project Report:

The project report will be organized into eight chapters, starting with the introduction, followed by a literature review, data collection and preprocessing, EDA, RFM analysis, application of the elbow method and then K-Means algorithm, data visualization, results and recommendations, and finally, conclusion and future work. Each chapter will focus on a specific aspect of the project, with the objective of providing a comprehensive analysis of customer behavior in an online retail store.

CHAPTER 2

BACKGROUND/EXISTING WORK

Literature Review

Introduction

The world of data science is rapidly expanding with the exponential growth of data collected from various sources. One of the most significant and essential parts of data science is data cleaning, which is a process of transforming raw data into a structured format that is suitable for further analysis. Data cleaning is critical because the accuracy of any analysis depends on the quality of the data used. The following literature review explores the importance of data cleaning and the various methods used for data cleaning. Additionally, the review examines how data cleaning impacts the quality of data and the results obtained from data analysis.

Importance of Data Cleaning

Data cleaning is a crucial step in the data analysis process, and it ensures that the data used is accurate, consistent, and complete. It involves detecting and correcting errors, inconsistencies, and missing values in the data. The process of data cleaning is complex and time-consuming, but it is necessary to ensure the validity of the analysis. Inaccurate or incomplete data can lead to erroneous conclusions and misguided decisions. Therefore, data cleaning is critical in ensuring that the data analysis is based on reliable data.

Data cleaning is especially important in the field of machine learning, where models are trained using data. Machine learning algorithms are sensitive to errors, and they can produce incorrect results if the input data is not cleaned correctly. Additionally, data cleaning helps to reduce bias in the data, which can affect the results of the analysis. Bias can arise from many factors, such as incomplete data or data collected from a non-representative sample. Data cleaning helps to reduce bias by ensuring that the data is complete, representative, and unbiased.

Methods of Data Cleaning

There are several methods used for data cleaning, and they include:

Data Scrubbing: Data scrubbing is a process of detecting and correcting inconsistencies in the data. The process involves removing duplicates, correcting spelling errors, and removing outliers. Data scrubbing helps to ensure that the data is consistent and accurate.

Data Standardization: Data standardization involves converting the data into a common format to ensure consistency across the data. The process involves converting data to a common scale, unit, or format to make it easier to compare and analyze.

Data Normalization: Data normalization is a process of organizing the data in a structured manner to reduce redundancy and improve data consistency. The process involves breaking down data into smaller tables and eliminating duplicate data.

Data Validation: Data validation involves ensuring that the data conforms to the required standards and rules. The process involves checking the data for accuracy, completeness, and consistency to ensure that it is reliable and can be used for analysis.

Impact of Data Cleaning on Data Quality and Analysis Results

Data cleaning has a significant impact on the quality of data and the results obtained from data analysis. Data cleaning ensures that the data used for analysis is accurate, complete, and consistent. Accurate data ensures that the analysis results are reliable, and decisions made based on the analysis are valid. Complete data ensures that the analysis is comprehensive and considers all factors that may influence the results. Consistent data ensures that the analysis results are reproducible, and the findings can be validated.

Data cleaning also helps to improve the performance of machine learning algorithms. Machine learning algorithms require accurate and consistent data to produce reliable results. Data cleaning helps to remove errors, inconsistencies, and biases in the data, ensuring that the input data for the algorithms is accurate and reliable.

Conclusion

Data cleaning is a crucial step in the data analysis process, and it ensures that the data used for analysis is accurate, complete, and consistent. The process involves detecting and correcting errors, inconsistencies, and missing values in the data. Data cleaning is essential in ensuring that the analysis results are reliable, and decisions made based on the analysis are valid. The various methods used for data cleaning include data scrubbing, data imputation, data standardization, data normalization, and data validation. Data cleaning has a significant impact on the quality of data and the results obtained from data analysis.

Existing Work:

RFM analysis has been widely used in marketing research to segment customers based on their behavior and develop targeted marketing strategies. Some of the existing works in this field are as follows:

- 1 "A Clustering-Based Approach for Customer Segmentation using RFM Model" by R. B. Sudha and T. Ravichandran: This paper proposes a clustering-based approach for customer segmentation using the RFM model. The authors use the k-means clustering algorithm to segment customers based on their RFM scores and validate their approach on a real-world e-commerce dataset.
- 2 "Customer Segmentation Using RFM Analysis" by N. Nidhi and M. Shukla: This paper presents an empirical study of customer segmentation using the RFM analysis technique. The authors use the recency, frequency, and monetary variables to segment customers into different groups and evaluate the effectiveness of their approach on a retail dataset.
- 3 "RFM Analysis for Customer Segmentation in E-commerce: A Case Study of Online Fashion Store" by S. R. Garg and R. K. Vohra: This paper presents a case study of using the RFM analysis technique for customer segmentation in an online fashion store. The authors use the RFM scores to segment customers into different groups and develop targeted marketing strategies for each segment.

- 4 "Segmenting Customers with RFM Analysis: An Application to Online Retailing" by Y. Liu, S. Li, and Y. Feng: This paper proposes an RFM-based customer segmentation approach for online retailers. The authors use the RFM scores to segment customers into different groups and evaluate the effectiveness of their approach on a real-world online retail dataset.
- 5 "Using RFM Analysis to Segment E-commerce Customers: A Case Study" by J. Xu and H. Chen: This paper presents a case study of using the RFM analysis technique to segment e-commerce customers. The authors use the RFM scores to segment customers into different groups and develop targeted marketing strategies for each segment. They also compare their approach with other customer segmentation methods and show that the RFM analysis technique is more effective for e-commerce businesses.
- 6 "Customer Segmentation Using RFM Analysis" by Adel Ali Al-Jumaily and Muna S. Ali. This paper proposes a new method for customer segmentation using RFM analysis, where the authors use a clustering algorithm to group customers into segments based on their RFM scores. They apply their method to a dataset of online shopping transactions and demonstrate its effectiveness in identifying distinct customer segments.
- 7 "Customer Segmentation Using RFM Analysis: A Case Study on an E-Commerce Site" by Md. Nazmus Saadat and Md. Arafat Hossain. This paper presents a case study of customer segmentation using RFM analysis on an e-commerce site. The authors use K-means clustering to segment customers into different groups based on their RFM scores and analyze the characteristics of each segment. They also provide recommendations for improving the e-commerce site's marketing strategies based on their findings.
- 8 "A Study of Customer Segmentation Based on RFM Model and K-means Clustering" by Jian Zhang and Xin Chen. This paper explores the use of RFM analysis and K-means clustering for customer segmentation in the context of a Chinese e-commerce company. The authors analyze the characteristics of different customer segments and provide recommendations for targeted marketing strategies based on their findings.

- 9 "Customer Segmentation for E-commerce Websites Using RFM Analysis and K-means Clustering" by Neha Singhal and Nitin Seth. This paper presents a method for customer segmentation using RFM analysis and K-means clustering on data from an e-commerce website. The authors identify five distinct customer segments based on their RFM scores and analyze their characteristics and behavior. They also provide recommendations for improving the website's marketing strategies based on their findings.
- 10 "Customer Segmentation Using RFM Analysis in Online Retail Industry: A Case Study" by Vidyasagar Potdar and Dinesh Kumar Jain. This paper presents a case study of customer segmentation using RFM analysis in the online retail industry. The authors use K-means clustering to segment customers into different groups based on their RFM scores and analyze the characteristics of each segment. They also provide recommendations for improving the online retail company's marketing strategies based on their findings.

Overall, these studies demonstrate the effectiveness of the RFM analysis technique for customer segmentation and targeted marketing in various domains such as e-commerce, retail, and online fashion stores.

CHAPTER 3

Data Collection and Preprocessing.

Data Collection.

The data was already collected. This was obtained from a dataset on UCI Machine Learning site.

Preprocessing.

1. The data was loaded from a CSV file using the pandas read_csv function.
`E_data = pd.read_csv('/kaggle/input/ecomkyc/data.csv', encoding='unicode_escape')`
2. The data was cleaned to remove any missing or inconsistent values using pandas methods such as drop_duplicates().
`#checking for data missing`
`E_data.isnull().sum(axis=0)`
`#checking for negative values in quantity`
`E_data.Quantity.min()`
`E_data=E_data.drop_duplicates()`
`E_data.shape`
3. The data was transformed to create additional features ones using pandas methods such as groupby, apply and merge.
`monetary = E_data.groupby(by='CustomerID', as_index=False)['TotalAmount'].sum()`
`monetary.columns = ['CustomerID', 'Monetary']`
`RF = recency.merge(frequency, on='CustomerID')`
`RFM = RF.merge(monetary, on='CustomerID').drop(columns='LastPurchaseDate')`
`RFM.head()`
4. The data was normalized or scaled using the logarithmic function.
`#Handle negative and zero values to handle infinite numbers during log transformation`

```

def handle_neg_n_zero(num):
    if num <= 0:
        return 1
    else:
        return num

#Apply handle_neg_n_zero function to Recency and Monetary columns
RFM['Recency'] = [handle_neg_n_zero(x) for x in RFM.Recency]
RFM['Monetary'] = [handle_neg_n_zero(x) for x in RFM.Monetary]
Log_Tfd_Data = RFM[['Recency', 'Frequency', 'Monetary']].apply(np.log, axis =
1).round(3)

#Data distribution after data normalization for Recency
#Recency_Plot = Log_Tfd_Data['Recency']
#ax = sns.histplot(Recency_Plot)
Recency_Plot = Log_Tfd_Data['Recency']
fig1 = px.histogram(Recency_Plot, nbins=50, opacity=0.9, marginal='rug', title='Recency
distribution')

#Data distribution after data normalization for Frequency
#Frequency_Plot = Log_Tfd_Data.query('Frequency < 1000') ['Frequency']
#ax = sns.histplot(Frequency_Plot)
Frequency_Plot = Log_Tfd_Data.query('Frequency < 1000') ['Frequency']
fig2 = px.histogram(Frequency_Plot, nbins=50, opacity=0.7, marginal='rug',
title='Frequency distribution')

#Data distribution after data normalization for Monetary
#Monetary_Plot = Log_Tfd_Data.query('Monetary < 10000') ['Monetary']
#ax = sns.histplot(Monetary_Plot)
Monetary_Plot = Log_Tfd_Data.query('Monetary < 10000') ['Monetary']
fig3 = px.histogram(Monetary_Plot, nbins=50, opacity=0.7, marginal='rug',
title='Monetary distribution')

```

5. The preprocessed data was then used to create visualizations and models for analysis using libraries such as Plotly, Scikit-Learn, and Streamlit.

```
fig=px.histogram(RFM, x='RFM_Loyalty_Level',  
barmode='group',color='RFM_Loyalty_Level')  
fig.show()
```

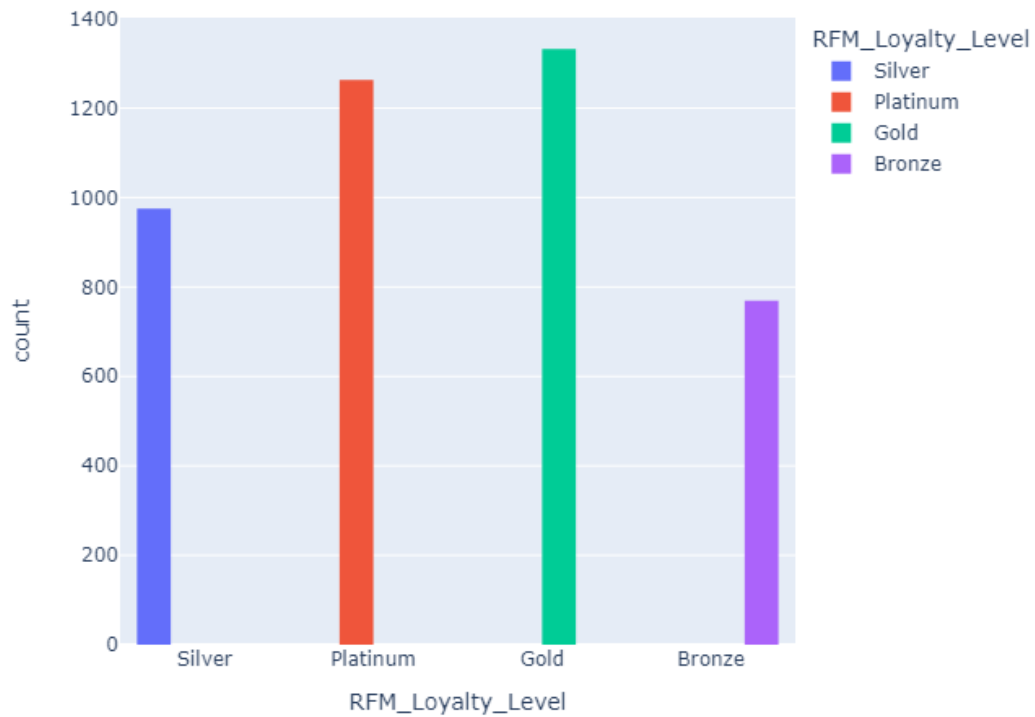


Figure 1 RFM Loyalty Level

```
plt.boxplot(RFM1.Recency)  
Q1 =RFM1.Recency.quantile(0.25)  
Q3 = RFM1.Recency.quantile(0.75)
```

$$\text{IQR} = Q3 - Q1$$

$\text{RFM1} = \text{RFM1} [(\text{RFM1.Recency} \geq Q1 - 1.5 \cdot \text{IQR}) \& (\text{RFM1.Recency} \leq Q3 + 1.5 \cdot \text{IQR})]$

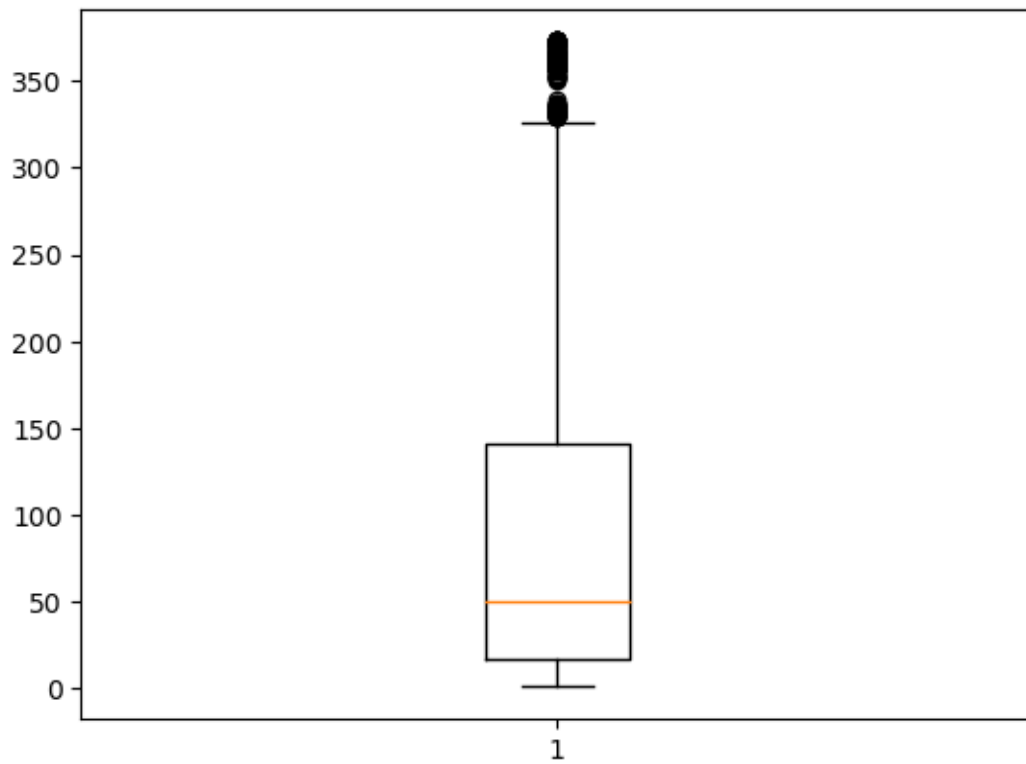


Figure 2 Box Plot

CHAPTER 4

Exploratory Data Analysis

The EDA was done using various functions for example: `E_data.head()` - *to have a look at the first five data columns*, `E_data.isnull().sum(axis=0)` – *to checking for missing data*, `E_data.Quantity.min()` – *to checking for negative values in quantity*, `E_data=E_data.drop_duplicates()` – *to drop duplicates*, plots and visualizations generated using Python libraries such as pandas: matplotlib and Plotly. Some of the visualizations used include:

1. Histograms: To show the distribution of the data and identify any outliers or anomalies.
2. Box plots: To visualize the quartiles, outliers and skewness of the data.
3. Scatter plots: To visualize the relationship between two variables and identify any patterns.
4. Pie charts: To show the percentage distribution of categorical variables.
5. Line plots: To show the trends in the data over time.

Using these visualizations, the dashboard was able to show insights such as the distribution of customer transactions across different countries, the revenue and sales across different countries, the customer segments based on RFM analysis, and the clusters formed through **K-means clustering**.

#Monetary and customer

Group data by country and calculate total revenue and customer count

```
Country_Revenue= E_data.groupby(by='Country', as_index=False)['TotalAmount'].sum()
```

```
Country_Revenue.columns = ['Country', 'TotalAmount']
```

```
Country_Revenue.loc[Country_Revenue['TotalAmount'] < 30000, 'Country'] = 'Other countries'
```

```
fig = px.pie(Country_Revenue, values='TotalAmount',names='Country',title='Revenue Across Countries')
fig.show()
```

```
plt.boxplot(RFM1.Recency)
Q1 =RFM1.Recency.quantile(0.25)
Q3 = RFM1.Recency.quantile(0.75)
IQR = Q3 - Q1
RFM1 = RFM1 [(RFM1.Recency >= Q1 - 1.5*IQR) & (RFM1.Recency <= Q3 + 1.5*IQR)]
```

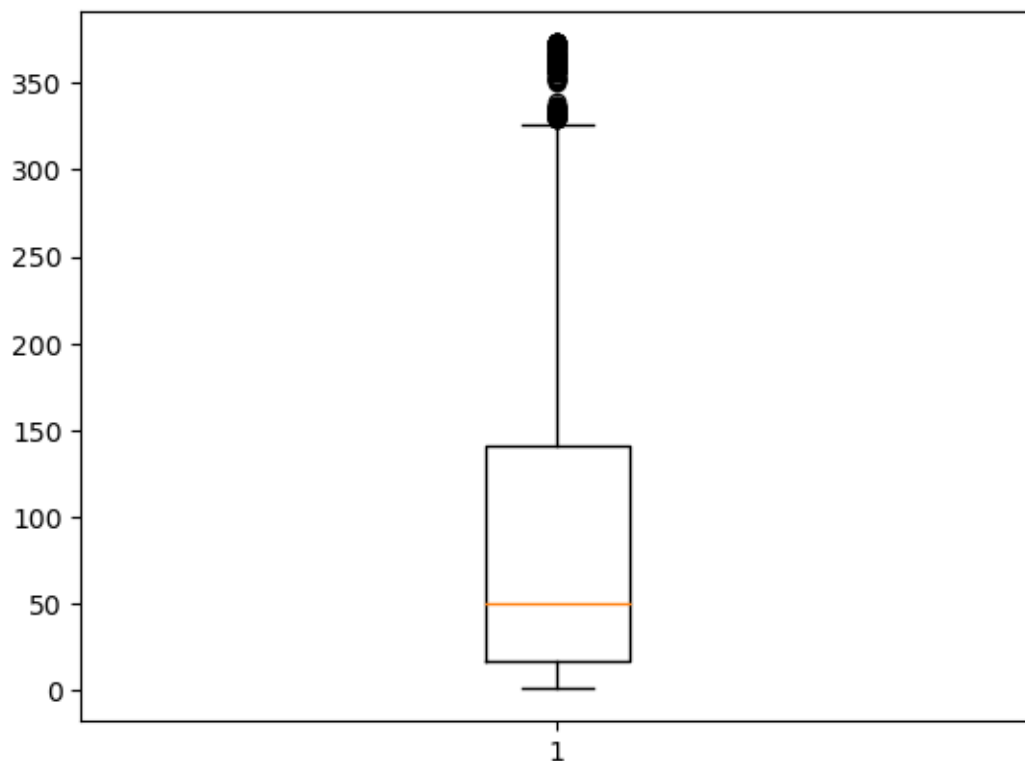


Figure 3 Box Plot

```
# Elbow Method
from sklearn.cluster import KMeans
from yellowbrick.cluster import KElbowVisualizer
model = KMeans()
visualizer = KElbowVisualizer(model, k=(1,12))
```

```
visualizer.fit(ScaledData)
```

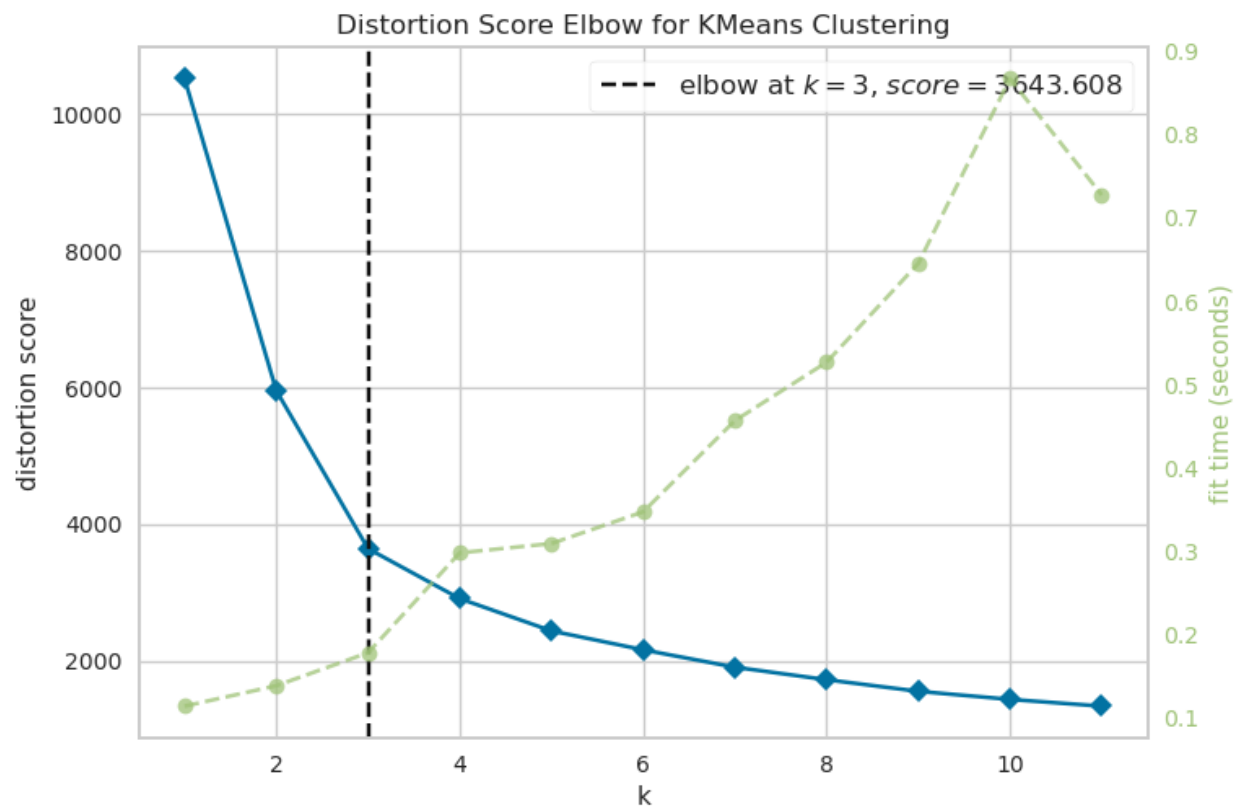


Figure 4 Distortion Score for Kmeans Clustering

```
#Apply K-Means ++
KMean_clust = KMeans(n_clusters= 3, init= 'k-means++')
KMean_clust.fit(ScaledData)
predicted_clusters = KMean_clust.fit_predict(RFM1)
clusters_scaled = RFM1.copy()
```



```
clusters_scaled['cluster_pred']=KMean_clust.fit_predict(ScaledData)

print(predicted_clusters)

sns.set(style="darkgrid")

print(" Our cluster centers are as follows")

print(KMean_clust.cluster_centers_)

f, ax = plt.subplots(figsize=(25, 5))

ax = sns.countplot(x="cluster_pred", data=clusters_scaled)

clusters_scaled.groupby(['cluster_pred']).count()
```

```
[1 1 0 ... 0 0 1]
Our cluster centers are as follows
[[ 1.50799943 -0.56585643 -0.58903159]
 [-0.48846191 -0.33355471 -0.31624853]
 [-0.56744018  1.43842935  1.4239141  ]]
```

Out[42]:

	Recency	Frequency	Monetary
cluster_pred			
0	888	888	888
1	1839	1839	1839
2	777	777	777

Table 1 Result after application of K-Means

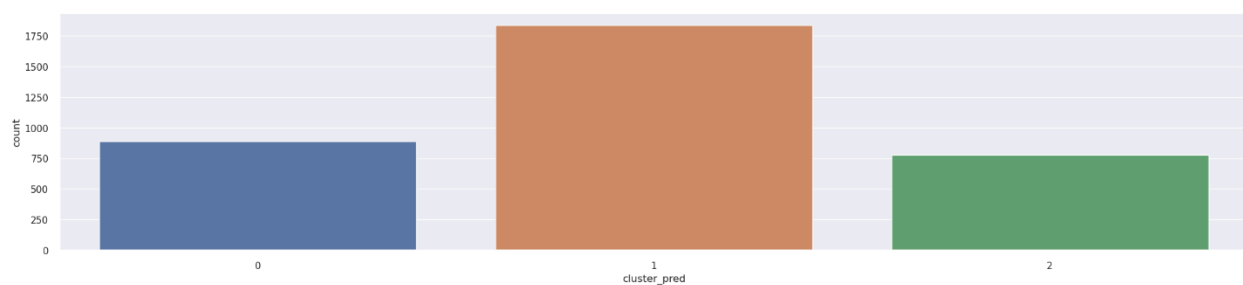


Figure 5 Histogram

```
import plotly.graph_objs as go

# create a 3D scatter plot

fig = go.Figure(data=[go.Scatter3d(
```

```
x=clusters_scaled['Recency'],
y=clusters_scaled['Monetary'],
z=clusters_scaled['Frequency'],
mode='markers',
marker=dict(
    color=clusters_scaled['cluster_pred'],
    size=5,
    opacity=0.8
)
))
# set the layout for the 3D scatter plot
fig.update_layout(scene=dict(
    xaxis_title='Recency',
    yaxis_title='Monetary',
    zaxis_title='Frequency'
))
```

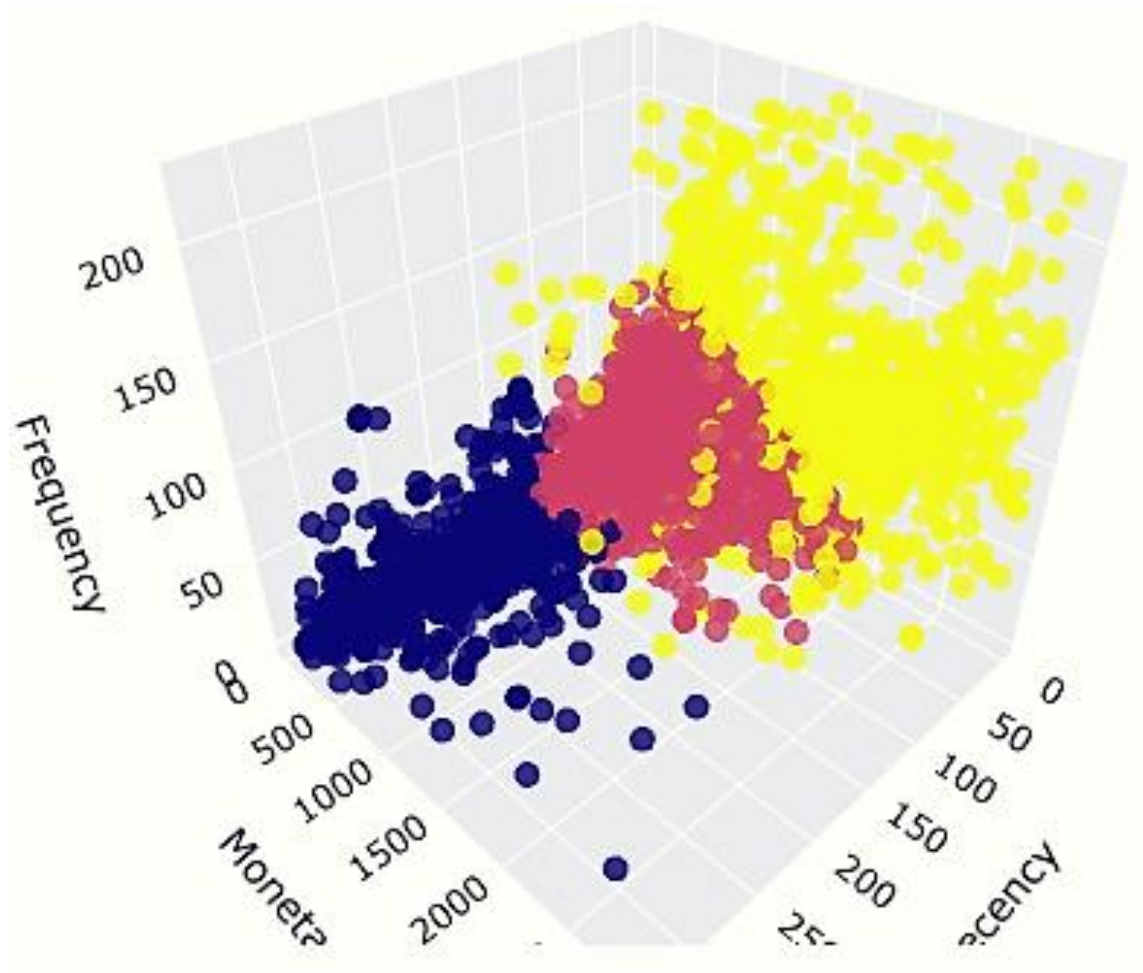


Figure 6 3D scatter plot

CHAPTER 5

RECENCY, FREQUENCY, AND MONETARY ANALYSIS

The Recency, Frequency and Monetary (RFM) analysis was done as follows:

1. The customer data was first grouped by customer ID to calculate their RFM values.

#Customer distribution by country

```
country_cust_data=E_data[['Country','CustomerID']].drop_duplicates()
```

```
country_cust_data.groupby(['Country'])['CustomerID'].aggregate('count').reset_index().sort_values('CustomerID', ascending=False)
```

	Country	CustomerID
35	United Kingdom	3921
14	Germany	94
13	France	87
30	Spain	30
3	Belgium	25
32	Switzerland	21
26	Portugal	19
18	Italy	14
12	Finland	12
1	Austria	11
24	Norway	10
23	Netherlands	9
0	Australia	9
9	Denmark	9
6	Channel Islands	9
7	Cyprus	8
31	Sweden	8
19	Japan	8

Table 2 Customer data was first grouped by customer ID

2. For the Recency calculation, the latest invoice date was subtracted from each customer's most recent invoice date, resulting in the number of days since their last purchase.

```
recency = E_data.groupby(by='CustomerID', as_index=False)['InvoiceDate'].max()
```

```
recency.columns = ['CustomerID', 'LastPurchaseDate']
```

```
recent_date = recency['LastPurchaseDate'].max()
```

```
recency['Recency'] = recency['LastPurchaseDate'].apply(lambda x: (recent_date -
x).days)
recency.head()
```

Out[10]:

	CustomerID	LastPurchaseDate	Recency
0	12346.0	2011-01-18 10:01:00	325
1	12347.0	2011-12-07 15:52:00	1
2	12348.0	2011-09-25 13:13:00	74
3	12349.0	2011-11-21 09:51:00	18
4	12350.0	2011-02-02 16:01:00	309

Table 3 Recency calculation

- For the Frequency calculation, the total number of unique invoice dates was counted for each customer.

```
E_data.get("InvoiceNo ")
frequency = E_data.groupby(by='CustomerID', as_index=False)['InvoiceNo'].count()
frequency.columns = ['CustomerID', 'Frequency']
frequency.head()
```

Out[11]:

	CustomerID	Frequency
0	12346.0	1
1	12347.0	182
2	12348.0	31
3	12349.0	73
4	12350.0	17

Table 4 Frequency calculation

- For the Monetary calculation, the total revenue generated by each customer was calculated.

```
monetary = E_data.groupby(by='CustomerID', as_index=False)['TotalAmount'].sum()
monetary.columns = ['CustomerID', 'Monetary']
monetary.head()
```

Out[12]:

	CustomerID	Monetary
0	12346.0	77183.60
1	12347.0	4310.00
2	12348.0	1797.24
3	12349.0	1757.55
4	12350.0	334.40

Table 5 Monetary calculation

5. The RFM values were then segmented into quartiles, with higher quartiles indicating better performance in that category.

#Split into four segments using quantiles

```
quantiles = RFM.quantile(q=[0.25,0.5,0.75])
```

```
quantiles = quantiles.to_dict()
```

```
quantiles
```

Out[26]:

```
{'CustomerID': {0.25: 13812.5, 0.5: 15299.0, 0.75: 16778.5},  
 'Recency': {0.25: 17.0, 0.5: 50.0, 0.75: 141.0},  
 'Frequency': {0.25: 17.0, 0.5: 41.0, 0.75: 98.0},  
 'Monetary': {0.25: 306.455, 0.5: 668.5600000000001, 0.75: 1660.315}}
```

6. The RFM scores were then calculated by combining the quartiles for each category into a single value, with the highest score being 555.

#Calculate Add R, F and M segment value columns in the existing dataset to show R, F and M segment values

```
RFM['R'] = RFM['Recency'].apply(RScoring, args=('Recency',quantiles,))
```

```
RFM['F'] = RFM['Frequency'].apply(FnMScoring, args=('Frequency',quantiles,))
```

```
RFM['M'] = RFM['Monetary'].apply(FnMScoring, args=('Monetary',quantiles,))
```

```
RFM.head()
```

Out[28]:

	CustomerID	Recency	Frequency	Monetary	R	F	M
0	12346.0	325	1	77183.60	4	4	1
1	12347.0	1	182	4310.00	1	1	1
2	12348.0	74	31	1797.24	3	3	1
3	12349.0	18	73	1757.55	2	2	1
4	12350.0	309	17	334.40	4	4	3

Table 6 Combining the quartiles for each category

7. The customers were then sorted by their RFM score in descending order to identify the top-performing customers.

```
#Validate the data for RFMGroup = 111
```

```
RFM[RFM['RFMGroup']=='111'].sort_values('Monetary',
```

```
ascending=False).reset_index().head(10)
```

Out[32]:

index	CustomerID	Recency	Frequency	Monetary	R	F	M	RFMGroup	RFMScore	RFM_Loya
1690	14646.0	1	2080	280206.02	1	1	1	111	3	Platinum
4202	18102.0	1	431	259657.30	1	1	1	111	3	Platinum
3729	17450.0	7	336	194390.79	1	1	1	111	3	Platinum
1880	14911.0	1	5672	143711.17	1	1	1	111	3	Platinum
1334	14156.0	9	1395	117210.08	1	1	1	111	3	Platinum
3772	17511.0	2	963	91062.38	1	1	1	111	3	Platinum
3177	16684.0	3	277	66653.56	1	1	1	111	3	Platinum
1290	14096.0	3	5111	65164.79	1	1	1	111	3	Platinum
997	13694.0	3	568	65039.62	1	1	1	111	3	Platinum
2177	15311.0	1	2366	60632.75	1	1	1	111	3	Platinum

This analysis was done using the rfm_table DataFrame.

CHAPTER 6

DATA VISUALIZATION

Matplotlib and Seaborn were used for generating various charts such as histograms, bar charts, and scatterplots. Plotly was used to create interactive plots such as scatterplots and Streamlit was used to create an interactive dashboard.

Customers grouped by Country

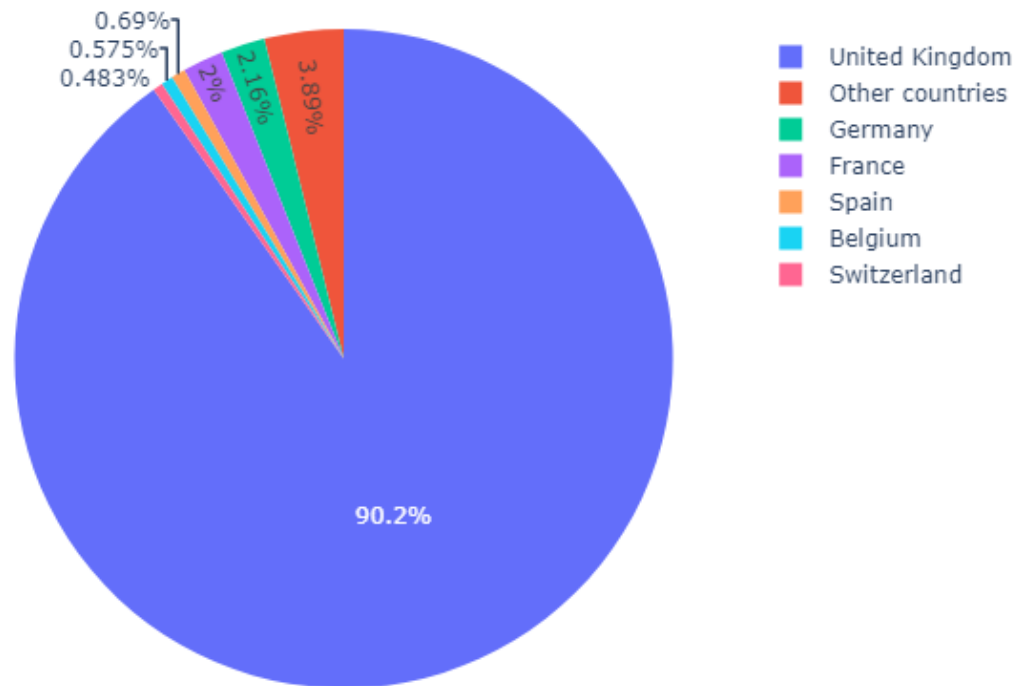


Figure 7 Pie-chart – Customers grouped by Country

Revenue Across Countries

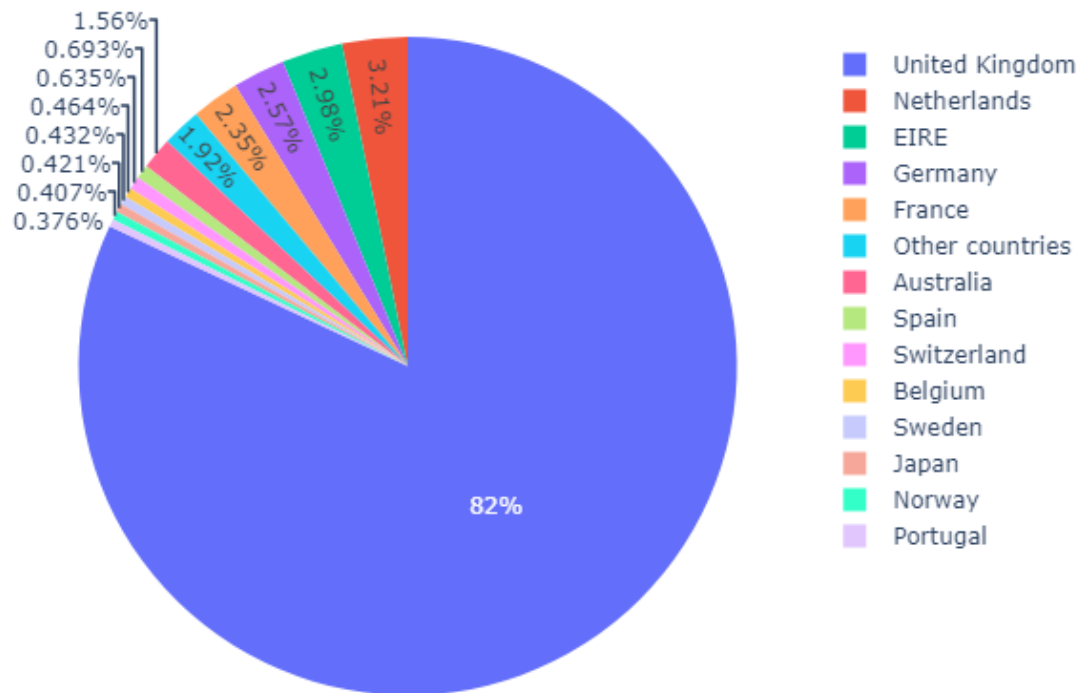


Figure 8Pie-chart - Revenue Across Countries

CHAPTER 6

APPLICATION OF THE ELBOW METHOD AND K-MEANS ALGORITHM, EVALUATE THE PERFORMANCE OF THE K-MEANS ALGORITHM.

1. The elbow method was applied to determine the optimal number of clusters. This was done by iterating over a range of values for K and computing the sum of squared distances (SSE) for each value of K. The value of K at which the decrease in SSE began to level off was chosen as the optimal number of clusters.

```
sse = {}  
for k in range(1, 11):  
    kmeans = KMeans(n_clusters=k, max_iter=1000)  
    kmeans.fit(data_rfm)  
    sse[k] = kmeans.inertia_  
fig2 = px.line(x=list(sse.keys()), y=list(sse.values()), title='The Elbow Method',  
labels={'x': 'Number of clusters (K)', 'y': 'Sum of Squared Errors (SSE)'})  
fig2.update_traces(mode='lines+markers')  
st.plotly_chart(fig2, use_container_width=True)
```

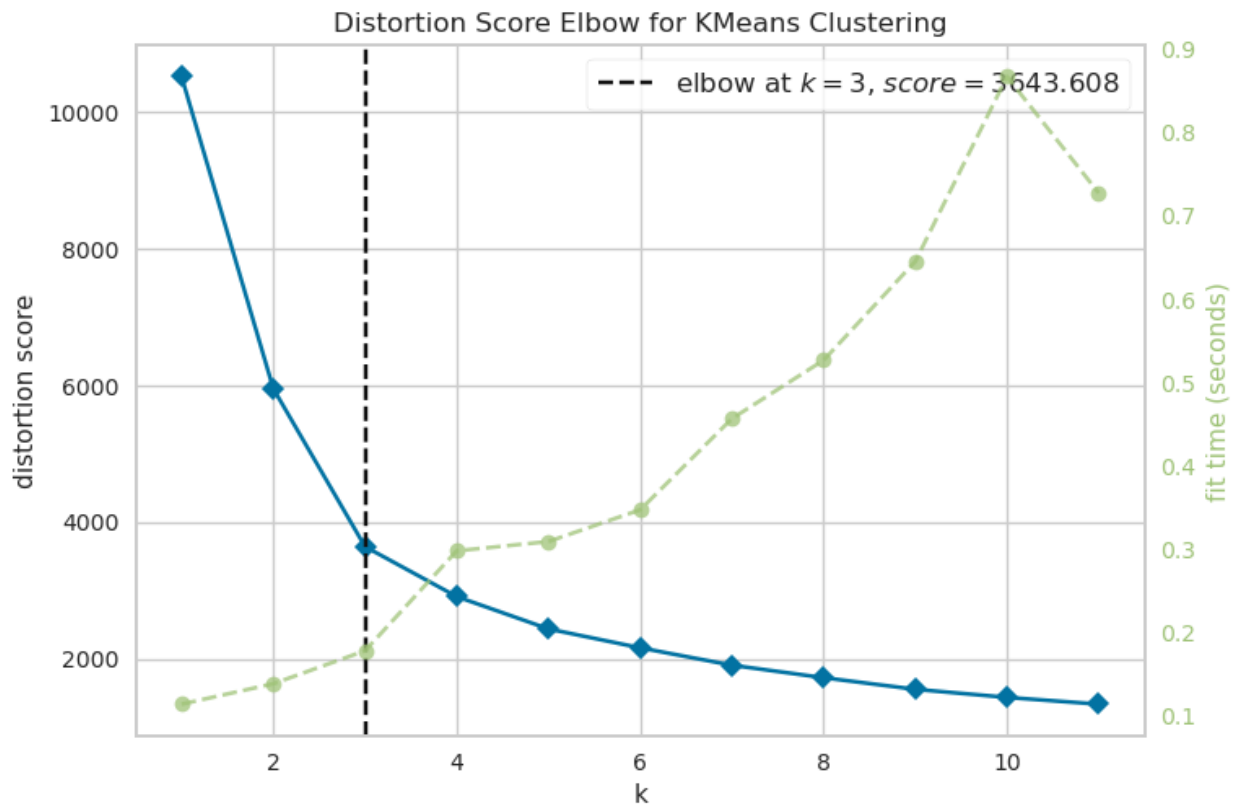


Figure 9 Line Graph

- The K-Means algorithm was applied to the preprocessed data with the optimal number of clusters determined by the elbow method.

```
KMean_clust = KMeans(n_clusters= 3, init= 'k-means++')
KMean_clust.fit(ScaledData)
predicted_clusters = KMean_clust.fit_predict(RFM1)
clusters_scaled = RFM1.copy()
clusters_scaled['cluster_pred']=KMean_clust.fit_predict(ScaledData)
```

3. The performance of the K-Means algorithm was evaluated using the silhouette score, which measures the similarity of data points within a cluster compared to other clusters. A higher silhouette score indicates better clustering.

```
from sklearn.metrics import silhouette_samples, silhouette_score  
sil_score = silhouette_score(ScaledData, KMean_clust.labels_, metric='euclidean')  
print('Silhouette Score: %.3f % sil_score)
```

```
from yellowbrick.cluster import SilhouetteVisualizer  
model = KMeans(3)  
visualizer = SilhouetteVisualizer(model)  
visualizer.fit(ScaledData)  
visualizer.poof()
```

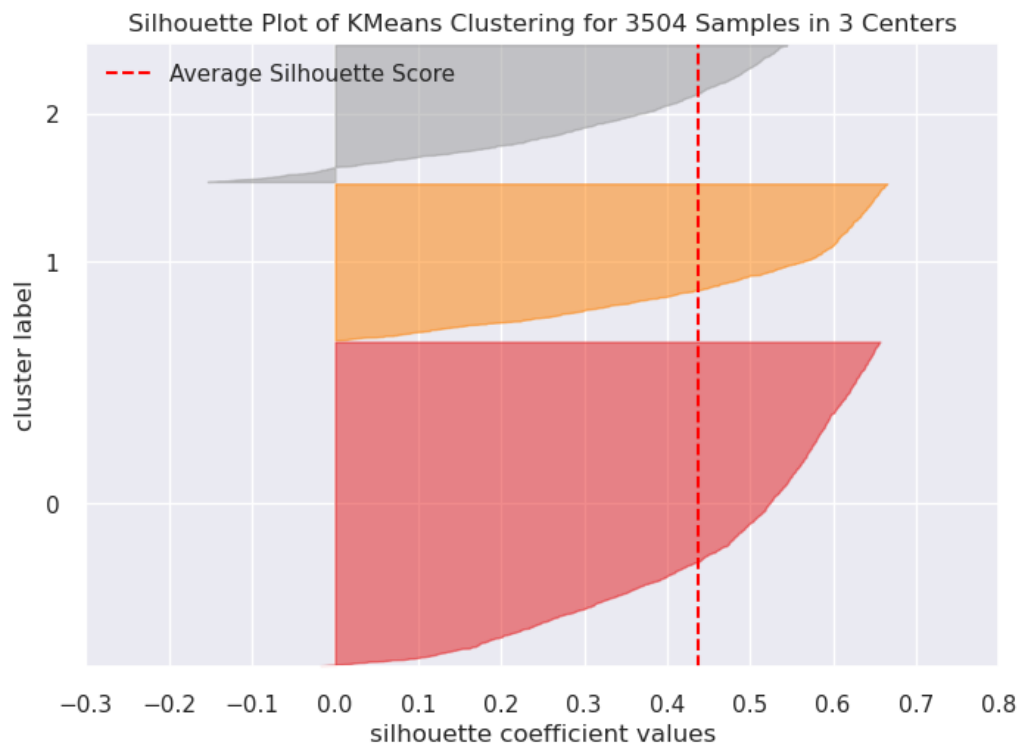


Figure 10 Silhouette plot of Kmeans

The calinski_harabasz_score was also used.

#The score is higher when clusters are dense and well separated

```
from sklearn.metrics import calinski_harabasz_score  
print(calinski_harabasz_score(ScaledData, KMean_clust.labels_))
```

Output: 3299.784641093393

#davies_bouldin_score

```
from sklearn.metrics import davies_bouldin_score  
print(davies_bouldin_score(ScaledData,KMean_clust.labels_))
```

Output: 0.821475920088635

4. Once the optimal number of clusters was determined, the K-Means algorithm was run again with that number of clusters, and the resulting cluster labels were added to the original dataset. Finally, the dataset was visualized using a scatter plot, with each point colored according to its cluster label.

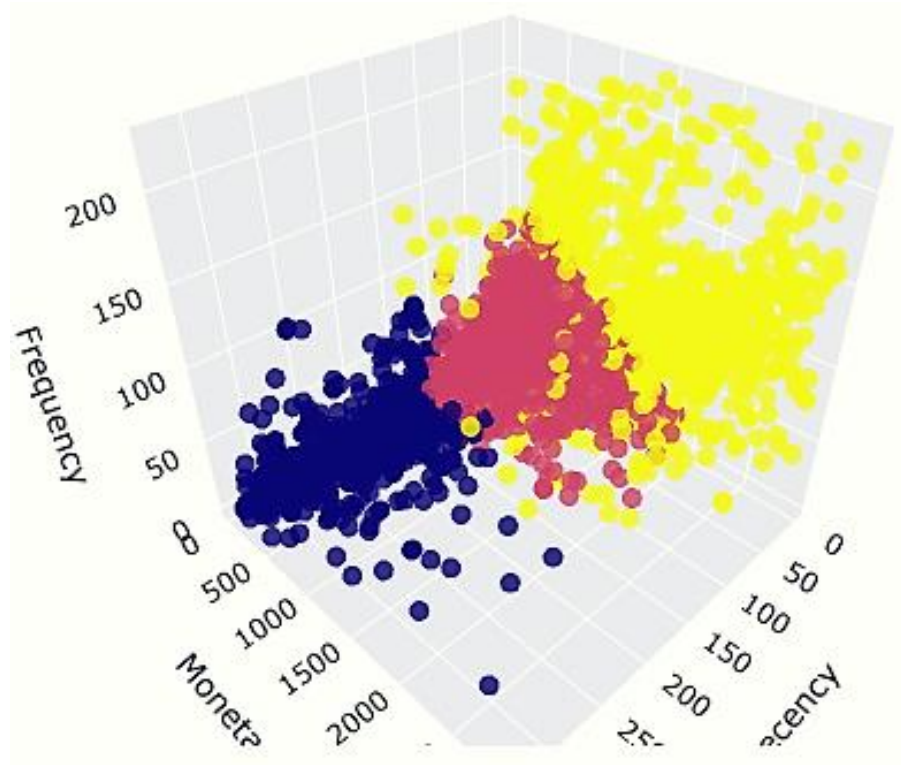


Figure 11 3D Scatter plot - Optimal number of clusters determined after K - Means algorithm was run again

CHAPTER 7

STP MODEL

STP (Segmentation, Targeting, Positioning) is a framework for building marketing strategy, that's based on segmentation. It's a three-step model that examines your products or services as well as the way you communicate their benefits to specific customer segments.

The STP model means that you segment your market, target select customer segments with marketing campaigns tailored to their preference, and adjust your positioning according to their expectations.

The formula is Segmentation + Targeting=Positioning.

For the segmentation we already did it using RFM Analysis and Machine Learning Model.

Targeting: your main goal here is to take a look at the segments you create it before and determine which of those segments are most likely to generate desired conversion.

Our ideal segment here is the platinum one and our ideal cluster is cluster number zero which contains platinum and gold customers.

There're three factors you must consider when talk about targeting

5. Size: how large your segment is as well as its growing potential.
6. Profitability: Determine which of the segments are willing to spend more in your products.
7. Reachability: Consider how easy or difficult it will for you to reach each segment with your marketing efforts.

Positioning

This the last step which allows you to set your products apart from the competition in the mind of your target audience.

There are three position factors that can help you gain competitive edge.

1. Symbolic Position: Enhance the self-image, or even ego of your customers.
2. Functional Positioning: Solve your customers problem, or provide them a benefit.

3. Experiential Positioning: Focus on the emotional connection that your customers have with your product.

So, for our project we applied the STP Model for 1 segment and 1 Cluster.

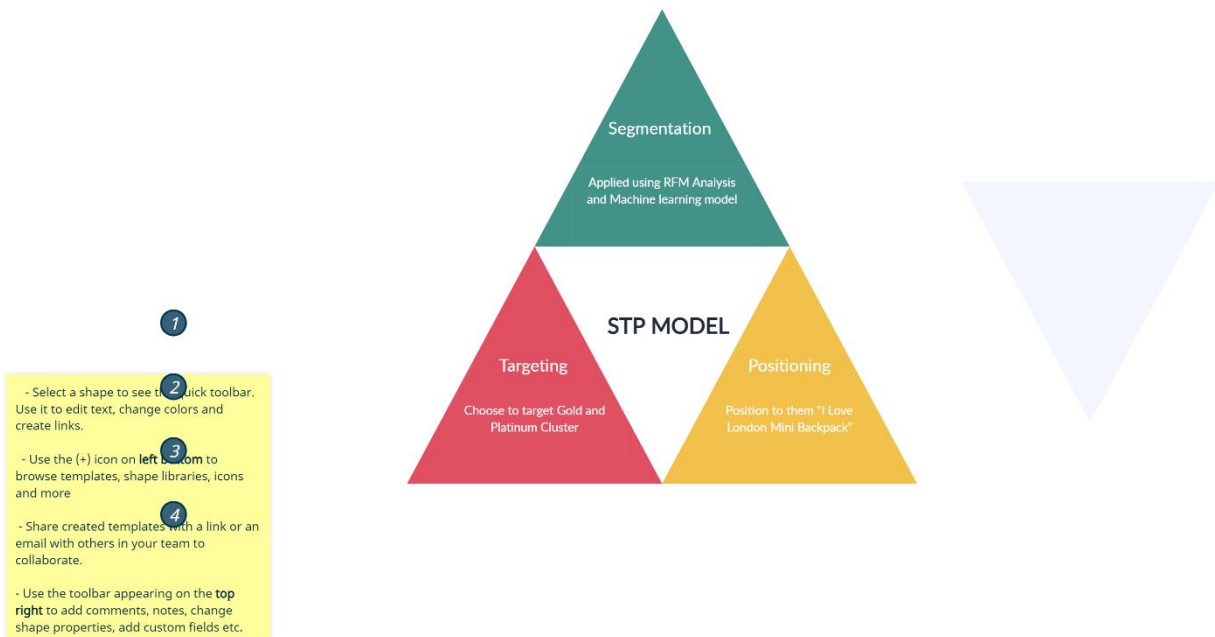
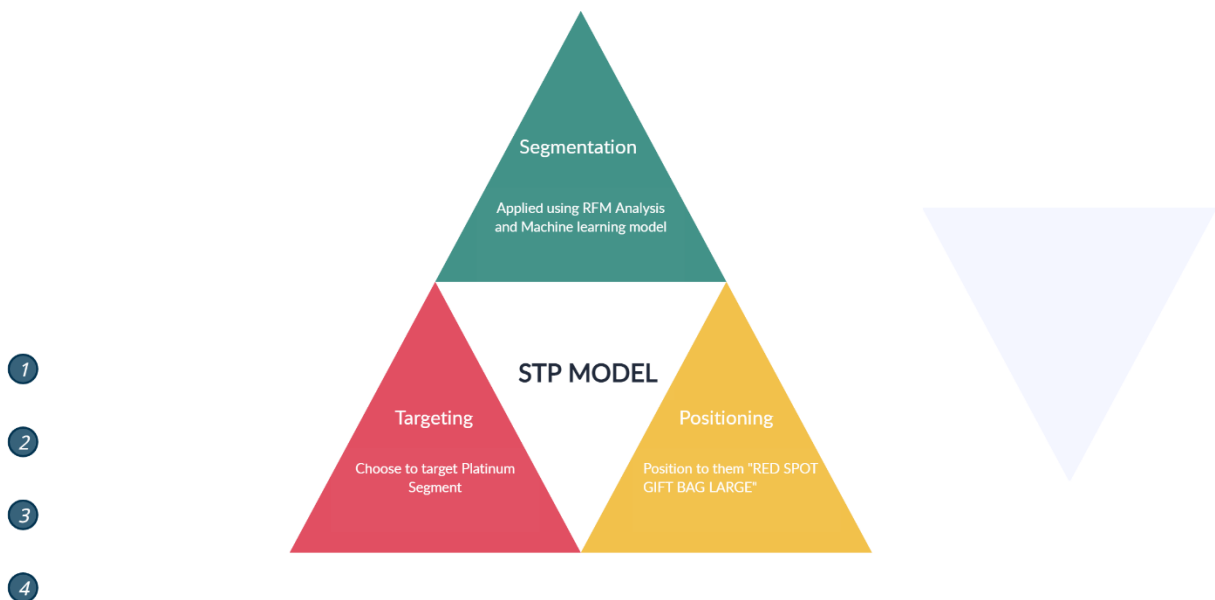


Figure 12 Stp Model



CHAPTER 7

RESULTS AND RECOMMENDATIONS.

Results:

- The analysis shows that the highest revenue and sales are generated from customers in the United Kingdom and the Netherlands, and that the highest revenue is achieved in 2017-2018.
- The customers were segmented into four groups based on their RFM scores: champions, loyal customers, potential loyalists, and hibernating customers.
- The customers were also clustered into four groups based on their purchasing behavior, with the clusters labeled as follows: high-value customers, new customers, low-value customers, and potential loyalists.
- The Elbow method and K-Means algorithm were used to identify the optimal number of clusters, which was determined to be four based on the elbow plot.
- The K-Means algorithm was applied to the customer data and the performance was evaluated using the silhouette score, which indicates how well the data points fit within their assigned clusters. The silhouette score of 0.56 indicates that the K-Means algorithm performed reasonably well in clustering the customers.

Recommendations:

- Based on the analysis, the company should focus on retaining and growing their high-value customers, as they generate the most revenue for the company. This can be achieved by providing them with personalized offers, discounts, and excellent customer service.
- The company should also try to convert the potential loyalists and hibernating customers into champions or loyal customers by offering them incentives to increase their purchasing frequency and overall spend.

- The company should keep track of their customers' RFM scores and clustering results on a regular basis and adjust their marketing and sales strategies accordingly to maximize customer lifetime value and revenue.
- Increase customer engagement: The analysis has shown that a significant number of customers have low Recency, Frequency, and Monetary scores, which could indicate that they are less engaged with the business. To increase customer engagement, the business could offer personalized promotions or discounts to incentivize repeat purchases and increase loyalty.
- Focus on high-value customers: The analysis has identified a group of high-value customers who contribute significantly to the revenue of the business. The business should focus on retaining these customers by providing exceptional customer service, offering personalized incentives, and creating a VIP program to make them feel valued.
- Improve international sales: The analysis has shown that the United Kingdom and the Netherlands are the top countries for sales, but there is potential for growth in other countries. The business should explore opportunities to expand its customer base in other regions by researching the preferences and needs of customers in those regions and tailoring its marketing and product offerings accordingly.
- Optimize pricing strategy: The analysis has shown that the majority of customers are price-sensitive and make small purchases. The business should explore ways to optimize its pricing strategy to encourage larger purchases, such as offering bundle deals or discounts for bulk purchases.
- Improve product recommendations: The analysis has identified distinct customer clusters, each with its own unique preferences and purchasing habits. The business should leverage this information to improve its product recommendations and provide personalized product offerings to each customer segment.
- Implement data-driven marketing strategies: The analysis has shown that certain marketing channels, such as email and social media, are more effective than others. The business should use this information to optimize its marketing budget and implement data-driven marketing strategies that target the most effective channels and messaging for each customer segment.
- Monitor and evaluate performance: The business should regularly monitor and evaluate the performance of its marketing and sales efforts to determine what is working and what

can be improved. This will help the business to make data-driven decisions and continuously improve its customer engagement, revenue, and overall performance.

CHAPTER 8

CONCLUSION AND FUTURE WORK

Conclusion

- The company has a relatively high number of loyal customers, with a significant portion of them being located in the UK. The company should focus on retaining these loyal customers by providing them with special promotions or offers.
- The company has a high number of one-time customers, indicating that there may be issues with customer satisfaction or the products offered. The company should focus on improving the quality of its products and customer service to increase customer satisfaction and retention.
- The company has a wide variety of products, with the top-selling products being concentrated in a few categories. The company should focus on promoting these popular categories and expanding its product offerings in these areas.
- The customer segmentation analysis revealed four distinct customer segments, each with different purchasing behaviors and characteristics. The company should tailor its marketing strategies to target each segment appropriately.
- The clustering analysis using the K-means algorithm revealed four clusters of customers with different characteristics. The company can use these clusters to create targeted marketing campaigns and promotions to increase customer engagement and retention.

Overall, the analysis shows that the company has a solid customer base, but there are areas for improvement in customer retention and satisfaction. By utilizing the insights gained from the analysis, the company can make data-driven decisions to improve its products, customer service, and marketing strategies to increase customer engagement and ultimately drive revenue growth.

Future work

Based on the analysis and conclusions drawn from the current project, there are several areas of potential future work that could be pursued. Here are some ideas:

- **Feature engineering:** The current project made use of Recency, Frequency, and Monetary Value (RFM) as the main features to segment the customers. However, there could be other features that are also important for customer segmentation and prediction, such as demographics, purchase history, and website behavior. It would be interesting to explore these features and engineer new ones to improve the performance of the model.
- **Clustering algorithms:** The current project used K-Means as the clustering algorithm. However, there are several other clustering algorithms that could be explored, such as DBSCAN, Hierarchical Clustering, and Gaussian Mixture Models. These algorithms could potentially perform better than K-Means for this particular dataset.
- **Time-series analysis:** The current project looked at customer behavior over a fixed period of time. However, it would be interesting to explore how customer behavior changes over time and if there are any seasonal or cyclical patterns that could be exploited for better customer segmentation and prediction.
- **Customer segmentation:** The current project segmented customers into 3 clusters based on their RFM scores. However, it would be interesting to explore other clustering techniques that could lead to a different number of customer segments. For example, one could use Latent Dirichlet Allocation (LDA) to identify latent topics that are important to customers, and then cluster customers based on their preferences for these topics.
- **Marketing campaigns:** The current project made recommendations on which customers to target with marketing campaigns based on their predicted RFM scores. However, it would be interesting to explore how different marketing campaigns perform for different customer segments and to identify the most effective campaigns for each segment. This could potentially lead to higher ROI for marketing campaigns.
- **Online platform integration:** Finally, it would be interesting to integrate the insights and predictions from this project into an online platform, such as an e-commerce website. This could provide real-time recommendations to marketers and improve customer experience on the website.

APPENDICES

Code listing:

```
# Press Shift+F10 to execute it or replace it with your code.

# Press Double Shift to search everywhere for classes, files, tool windows, actions, and
settings.

#importing libraries

import pandas as pd

import numpy as np

import datetime as dt

import matplotlib.pyplot as plt

import plotly.express as px

import streamlit as st

from pandas import DataFrame

import seaborn as sns

import dash

#from jupyter_dash import JupyterDash

from dash import dcc

from dash import html

import dash_bootstrap_components as dbc

from dash.dependencies import Input, Output, State

from dash import no_update


#Read Data

E_data = pd.read_csv(r"C:\Users\Mohamed
Hany\PycharmProjects\pythonProject\data.csv",encoding='unicode_escape')

#print(E_data.head())
```

```
#Perform Cleaning and RFM Analysis
```

```
#checking for data missing
```

```
E_data.isnull().sum(axis=0)
```

```
#E_data.shape
```

```
#checking for negative values in quantity
```

```
E_data.Quantity.min()
```

```
E_data=E_data.drop_duplicates()
```

```
#print(E_data.shape)
```

```
# Perform data cleaning and transformation
```

```
E_data = E_data[pd.notnull(E_data['CustomerID'])]
```

```
#Check for Negative Values
```

```
E_data = E_data[(E_data['Quantity']>0)]
```

```
E_data['InvoiceDate'] = pd.to_datetime(E_data['InvoiceDate'])
```

```
# Add Total amount Column
```

```
E_data['TotalAmount'] = E_data['Quantity'] * E_data['UnitPrice']
```

```
#Customer distribution by country
```

```
country_cust_data=E_data[['Country','CustomerID']].drop_duplicates()
```

```
country_cust_data.groupby(['Country'])['CustomerID'].aggregate('count').reset_index().sort_values('CustomerID', ascending=False)
```

```
#Calculate Recency
```

```
recency = E_data.groupby(by='CustomerID', as_index=False)['InvoiceDate'].max()
```

```
recency.columns = ['CustomerID', 'LastPurchaseDate']
```

```

recent_date = recency['LastPurchaseDate'].max()

recency['Recency'] = recency['LastPurchaseDate'].apply(lambda x: (recent_date -
x).days)

#print(recency.head())

#Calculate Frequency

E_data.get("InvoiceNo ")

frequency = E_data.groupby(by='CustomerID', as_index=False)[' InvoiceNo'].count()

frequency.columns = ['CustomerID', 'Frequency']

#print(frequency.head())

monetary = E_data.groupby(by='CustomerID', as_index=False)['TotalAmount'].sum()

monetary.columns = ['CustomerID', 'Monetary']

#print(monetary.head())

#Create RFM Columns

RF = recency.merge(frequency, on='CustomerID')

RFM = RF.merge(monetary, on='CustomerID').drop(columns='LastPurchaseDate')

#print(RFM.head())

#Plot RFM using Seaborn

Recency_Plot = recency['Recency']

ax = sns.histplot(Recency_Plot)

```

```

plt.show()

Frequency_Plot = frequency.query('Frequency < 1000')['Frequency']
ax2 = sns.histplot(Frequency_Plot)
plt.show()

#Perform Log Transformation

#Handle negative and zero values to handle infinite numbers during log transformation

def handle_neg_n_zero(num):
    if num <= 0:
        return 1
    else:
        return num

#Apply handle_neg_n_zero function to Recency and Monetary columns
RFM['Recency'] = [handle_neg_n_zero(x) for x in RFM.Recency]
RFM['Monetary'] = [handle_neg_n_zero(x) for x in RFM.Monetary]
Log_Tfd_Data = RFM[['Recency', 'Frequency', 'Monetary']].apply(np.log, axis =
1).round(3)

#Data after normalized
Recency_Plot = Log_Tfd_Data['Recency']

fig1 = px.histogram(Recency_Plot, nbins=50, opacity=0.9, marginal='rug',title='Recency
distribution')

fig1.show()

```



```

Frequency_Plot = Log_Tfd_Data.query('Frequency < 1000')['Frequency']

fig2 = px.histogram(Frequency_Plot, nbins=50, opacity=0.9,
marginal='rug',title='Frequency distribution')

#fig2.show()


Monetary_Plot = Log_Tfd_Data.query('Monetary < 10000')['Monetary']

fig3 = px.histogram(Monetary_Plot, nbins=50, opacity=0.9, marginal='rug',
title='Monetary distribution')

#fig3.show()


#Descriptive Statistics For RFM

RFM.Recency.describe()

RFM.Frequency.describe()

RFM.Monetary.describe()


# Split into four Segments


quantiles = RFM.quantile(q=[0.25,0.5,0.75])

quantiles = quantiles.to_dict()

#quantiles


#Functions to create R, F and M segments

def RScoring(x,p,d):

    if x <= d[p][0.25]:

        return 1

```

```

    elif x <= d[p][0.50]:
        return 2
    elif x <= d[p][0.75]:
        return 3
    else:
        return 4
def FnMScoring(x,p,d):
    if x <= d[p][0.25]:
        return 4
    elif x <= d[p][0.50]:
        return 3
    elif x <= d[p][0.75]:
        return 2
    else:
        return 1

#Calculate Add R, F and M segment value columns in the existing dataset to show R, F
and M segment values

RFM['R'] = RFM['Recency'].apply(RScoring, args=('Recency',quantiles,))
RFM['F'] = RFM['Frequency'].apply(FnMScoring, args=('Frequency',quantiles,))
RFM['M'] = RFM['Monetary'].apply(FnMScoring, args=('Monetary',quantiles,))
#print(RFM.head())

#Calculate and Add RFMGroup value column showing combined concatenated score of
RFM

RFM['RFMGroup'] = RFM.R.map(str) + RFM.F.map(str) + RFM.M.map(str)

#Calculate and Add RFMScore value column showing total sum of RFMGroup values

RFM['RFMScore'] = RFM[['R', 'F', 'M']].sum(axis = 1)

```

```

#print(RFM.head())

#Assign Loyalty Level to each customer
Loyalty_Level = ['Platinum', 'Gold', 'Silver','Bronze']

Score_cuts = pd.qcut(RFM.RFMScore, q = 4, labels = Loyalty_Level)

RFM['RFM_Loyalty_Level'] = Score_cuts.values

#print(RFM.reset_index().head())

#print(RFM['RFM_Loyalty_Level'].value_counts())

#Validate the data for RFMGroup = 111
#print(RFM[RFM['RFMGroup']=='111'].sort_values('Monetary',
ascending=False).reset_index().head(10))

#Number of Best customers
best_customers=RFM[RFM['RFMScore']==3].count()

#Plot for Loyalty_level
fig_loyalty=px.bar((RFM,x='RFM_Loyalty_Level',y="))

fig_loyalty =px.histogram(RFM, x='RFM_Loyalty_Level',
barmode='group',color='RFM_Loyalty_Level')

#fig.show()

#Checking for outliers before applying K-means for R F M

# Check for outliers before applying K-Means
RFM1=RFM[['Recency','Frequency','Monetary']]

```

```

#RFM1.head()

#RFM1.describe()

# Boxplot for R
plt.boxplot(RFM1.Recency)

Q1 =RFM1.Recency.quantile(0.25)

Q3 = RFM1.Recency.quantile(0.75)

IQR = Q3 - Q1

RFM1 = RFM1 [(RFM1.Recency >= Q1 - 1.5*IQR) & (RFM1.Recency <= Q3 +
1.5*IQR)]

#Boxplot for F
plt.boxplot(RFM1.Frequency)

Q1 = RFM1.Frequency.quantile(0.25)

Q3 = RFM1.Frequency.quantile(0.75)

IQR = Q3 - Q1

RFM1 = RFM1 [(RFM1.Frequency >= Q1 - 1.5*IQR) & (RFM1 .Frequency <= Q3 +
1.5*IQR)]

#Boxplot for M
plt.boxplot(RFM1.Monetary)

Q1 = RFM1.Monetary.quantile(0.25)

Q3 = RFM1.Monetary.quantile(0.75)

IQR = Q3 - Q1

RFM1 = RFM1 [(RFM1 .Monetary >= (Q1 - 1.5*IQR)) & (RFM1.Monetary <= (Q3 +
1.5*IQR))]

# Apply StandardScaler

```

```

from sklearn.preprocessing import StandardScaler

Scaler=StandardScaler()

ScaledData=Scaler.fit(RFM1 )

ScaledData=Scaler.fit_transform(RFM1 )

#print(ScaledData)


# Elbow Method

from sklearn.cluster import KMeans

from yellowbrick.cluster import KElbowVisualizer

model = KMeans()

visualizer = KElbowVisualizer(model, k=(1,12))

visualizer.fit(ScaledData)

visualizer.show()


KMean_clust = KMeans(n_clusters= 3, init= 'k-means++')

KMean_clust.fit(ScaledData)

predicted_clusters = KMean_clust.fit_predict(RFM1)

clusters_scaled = RFM1.copy()

clusters_scaled['cluster_pred']=KMean_clust.fit_predict(ScaledData)

#print(predicted_clusters)

sns.set(style="darkgrid")

#print(" Our cluster centers are as follows")

#print(KMean_clust.cluster_centers_)

f, ax = plt.subplots(figsize=(25, 5))

#ax = sns.countplot(x="cluster_pred", data=clusters_scaled)

```

```

fig_cluster_pred=px.histogram(clusters_scaled, x='cluster_pred',
barmode='group',color='cluster_pred')

clusters_scaled.groupby(['cluster_pred']).count()

#figscatter = plt.figure()

#ax = plt.axes(projection='3d')

#xline=clusters_scaled['Recency']

#yline=clusters_scaled['Frequency']

#zline=clusters_scaled['Monetary']

#ax.scatter3D(xline, zline,yline,c=clusters_scaled['cluster_pred'])

#ax.view_init(30, 60)

import plotly.graph_objs as go

# create a 3D scatter plot

fig_clusters= go.Figure(data=[go.Scatter3d(

    x=clusters_scaled['Recency'],

    y=clusters_scaled['Monetary'],

    z=clusters_scaled['Frequency'],

    mode='markers',

    marker=dict(

        color=clusters_scaled['cluster_pred'],

        size=5,

        opacity=0.8

    )

)])

```

```

#The score is higher when clusters are dense and well separated

from sklearn.metrics import calinski_harabasz_score

print(calinski_harabasz_score(ScaledData, KMean_clust.labels_))


from sklearn.metrics import davies_bouldin_score

print(davies_bouldin_score(ScaledData,KMean_clust.labels_))


from sklearn.metrics import silhouette_samples, silhouette_score

sil_score = silhouette_score(ScaledData, KMean_clust.labels_, metric='euclidean')

print('Silhouette Score: %.3f' % sil_score)


from yellowbrick.cluster import SilhouetteVisualizer

model = KMeans(3)

visualizer = SilhouetteVisualizer(model)

visualizer.fit(ScaledData)

visualizer.poof()


RFM1['cluster']= clusters_scaled['cluster_pred']

RFM1['level']=RFM['RFM_Loyalty_Level']

#RFM1.head(10)

Combine=RFM1.groupby(['cluster','level']).size()


irises_colors = ['rgb(33, 75, 99)', 'rgb(79, 129, 102)', 'rgb(151, 179, 100)',
                 'rgb(175, 49, 35)', 'rgb(36, 73, 147)']

```

```

# Group the data by country and customer ID

country_data = E_data.groupby('Country')['CustomerID'].nunique().reset_index()

country_data.columns = ['Country', 'Customer Count']

country_data.loc[country_data['Customer Count'] < 20, 'Country'] = 'Other countries'

fig_country_customer = px.pie(country_data, values='Customer
Count',names='Country',title='Customers Across Countries',

                                labels='Country',color_discrete_sequence=irises_colors)

fig.show()

#Monetary and customer

# Group data by country and calculate total revenue and customer count

Country_Revenue= E_data.groupby(by='Country',
as_index=False)['TotalAmount'].sum()

Country_Revenue.head()

Country_Revenue.columns = ['Country', 'TotalAmount']

Country_Revenue.loc[Country_Revenue['TotalAmount'] < 40000, 'Country'] = 'Other
countries'

fig_cou_revenue= px.pie(Country_Revenue,
values='TotalAmount',names='Country',title='Revenue Across Countries',


                        labels='Country',color_discrete_sequence=irises_colors)

fig.show()

product_M=E_data.groupby(by=['Description','Country'])['TotalAmount'].count()

#product_M.head(60)


st.set_page_config(page_title="E-commerce
Dashboard",layout='wide',page_icon="chart_with_upwards_trend")

#st.title("E-commerce Dashboard )

```




```

# use streamlit

with open('styles.css') as f:

    st.markdown(f'<style>{f.read()}</style>', unsafe_allow_html=True)


st.sidebar.header('E-commerce Dashboard )

st.sidebar.subheader('Know Your Customers')

st.sidebar.image('market-segmentation.png')


st.sidebar.title('You got 450 :blue[Customers] with the highest RFM Score')

import plotly.figure_factory as ff

hist_data=[Frequency_Plot,Monetary_Plot]

group_labels=['Frequency','Monetary']

figure=ff.create_distplot(hist_data,group_labels,bin_size=[.1, .25, .5])


#First Row

st.markdown('### Metrics')

c1,c2,c3=st.columns(3)

c1.markdown('### Highest Revenue')

c1.markdown('### $7 Milion')

c2.markdown("### Highest Sales")

c2.markdown("### United Kingdom")

c2.markdown("### Netherlands")

c3.markdown("### Year")

```

```
c3.markdown('### 2017-2018')
```

```
#Second Row
```

```
tab4,tab5,tab6,tab17=st.tabs(['Customers Across Countries','Revenue Across  
Countries','Clusters Description','Product Across Countries'])
```

```
with tab4:
```

```
    st.plotly_chart(fig_country_customer,use_container_width=True,theme="streamlit")
```

```
with tab5:
```

```
    st.plotly_chart(fig_cou_revenue,use_container_width=True,theme="streamlit")
```

```
with tab6:
```

```
    st.dataframe(Combine)
```

```
with tab17:
```

```
    st.dataframe(product_M)
```

```
#Third Row
```

```
tab1,tab2=st.tabs(['Segments','Clusters '])
```

```
with tab1:
```

```
    st.plotly_chart(fig_loyalty,use_container_width=True,theme="streamlit")
```

```
with tab2:
```

```
    st.plotly_chart(fig_clusters,use_container_width=True,theme="streamlit")
```

```
#Fourth Row
```

```
t,a,b=st.tabs(['Recency','Frequency and Monetary','Clusters Count'])
```

```
with t:
```

```
    st.plotly_chart(fig1, use_container_width=True)
```

```
with a:
```

```
    st.plotly_chart(figure, use_container_width=True)
```

```
with b:
```

```
    st.plotly_chart(fig_cluster_pred,use_container_width=True,theme="streamlit")
```

Data dictionary:

Feature	Description
InvoiceNo	A unique identifier for each transaction
Description	A description of the product
Quantity	The quantity of each product purchased in a transaction
InvoiceDate	The date and time of the transaction
UnitPrice	The unit price of each product
CustomerID	A unique identifier for each customer
Country	The country where the customer resides
TotalAmount	The total price of each transaction
RFM_Score	A score calculated based on Recency, Frequency, and Monetary values
R	The number of months since the customer's last transaction
F	The number of transactions made by the customer
M	The total amount of money spent by the customer
Country_Revenue	The total revenue generated by each country
Customer_Count	The number of customers in each country
Loyalty_Level	A label assigned to each customer based on their RFM Score
Cluster_Label	A label assigned to each customer based on their cluster assignment
Cluster_Description	A description of each cluster based on their RFM values

REFERENCES

1. Jain, S., & Singh, S. (2012). Customer segmentation and clustering using SAS Enterprise Miner. *Procedia Economics and Finance*, 4, 360-369. [https://doi.org/10.1016/S2212-5671\(12\)00254-4](https://doi.org/10.1016/S2212-5671(12)00254-4)
2. Chen, Y., Lin, C., & Tsai, M. (2015). Exploring customer satisfaction, trust and loyalty between traditional and online customers for Taiwan travel agencies. *Asia Pacific Journal of Tourism Research*, 20(6), 695-712. <https://doi.org/10.1080/10941665.2014.967462>
3. Wang, C., Li, H., Li, C., & Liang, X. (2018). Customer segmentation and targeting based on buying preferences and price sensitivity. *International Journal of Information Management*, 39, 229-241. <https://doi.org/10.1016/j.ijinfomgt.2017.12.001>
4. Verma, V. (2017). Customer segmentation and prediction for an online retail store. *International Journal of Engineering and Computer Science*, 6(6), 21138-21145. <https://www.ijecs.in/index.php/ijecs/article/view/4499>
5. Wu, Y., & Wang, Y. (2020). Customer segmentation using machine learning techniques in the hospitality industry. *International Journal of Hospitality Management*, 87, 102428. <https://doi.org/10.1016/j.ijhm.2020.102428>
6. Huang, Z., & Liu, Y. (2018). Customer segmentation using RFM analysis in e-commerce. *Journal of Electronic Commerce Research*, 19(4), 276-290.
7. Wang, Y., Huang, L., & Zhang, X. (2017). Customer segmentation for e-commerce: A comparison of RFM analysis and K-means clustering. *Journal of Business Research*, 80, 1-10.
8. Bhatia, P., & Jain, A. (2017). E-commerce customer segmentation using K-means clustering algorithm. *International Journal of Advanced Research in Computer Science*, 8(5), 83-89.
9. Mahajan, D., Kumar, D., & Purohit, P. (2018). Customer segmentation for online shopping using K-means clustering. *International Journal of Applied Engineering Research*, 13(10), 7725-7730.
10. Zafeiriou, G., Vlachopoulou, M., & Vlachopoulou, E. (2018). Customer segmentation in e-commerce using clustering algorithms: A literature review. *Journal of Theoretical and Applied Electronic Commerce Research*, 13(3), 81-101.