SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CUSTOMER SEGMENTATION USING RFM MODEL AND K-MEANS CLUSTERING ALGORITHM

PREDICTIVE ANALYTICS (CSE3047)

FINAL REPORT

submitted by

MYLIE MUDALIYAR- 20BCE2661

SWETA CHANDRASEKHAR - 20BCE2625

In partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

ACKNOWLEDGEMENT

This is to state on the record that a report of this stature was made possible because of a concerted effort put in by the group members who succoured and motivated throughout and due to the availability of resources that helped us in forming the contents. In this regard, we would like to thank our Professor Dr. W.B. Vasantha**, an eminent faculty member of this reverent institution who has incidentally also been our professor for the course Predictive Analytics (CSE3047) in the Fall Semester 2022. It is her way of teaching and guidance that helped us to work on our project. Her suggestions made us to improve our project in each phase. All the group members have helped whenever someone encountered difficulties and this acknowledgement would be incomplete without them. Last but not the least, this institution, which provided us with sufficient resources and allocating the course Predictive Analytics.**

1) Abstract

In carrying out successful E-Commerce, the most important things are innovation and understanding what customer wants. According to business, different **CRM (customer relationship management)** strategies are brought forward to gain a high level of customer satisfaction. A company may create three segments like **High** (group who buys often, spends more and visited the platform recently), **Medium** (group which spends less than high group and is not that much frequent to visit the platform) and **Low** (group which is on the verge of churning out). In this project, we plan to propose a customer segmentation concept in which the customers based of an establishment are divided into segments based their characteristics, activities and attributes. This idea can be used by the B2C companies to outperform the competition by developing uniquely appealing products and services and make it reach to potential customers. This approach will be implemented using a **RFM (recency, frequency, and monetary)** model and **K-means clustering** algorithm.

Literature Review

YEAR OF RELEASE	STUDIES	CONTEXT, RESEARCH DESIGN AND ANALYSIS	PURPOSE AND KEY FINDINGS
2016	**Sarvari et al.**	• A data from a global pizza restaurant chain. • RFM analysis + K means clustering + Association rules	• Aim to determine the best approach to customer segmentation. • Different types of scenarios were designed, performed and evaluated under test condition. • They showed that having an appropriate

			segmentation approach is vital if there are to be strong association. Also, the weights of RFM attributes affected rule association performance positively.
2016	**Dursun & Caber**	• A sample of 369 from the population 5939 • Hotel customers • RFM analysis + K means clustering	• Aim to segment hotel customers • Eight clusters were obtained according to their RFM score • Loyal customers, loyal summer season customers, collective buying customers, winter season customers, lost customers, high potential customers, new customers and winter season high potential customers were identified. • Customers' card types were compared with new segmentation.
2016	**Abirami & Pattabira man**	• Retailing sector • RFM analysis + K means clustering + Association rules	• They suggested a approach of customer classification. • RFM model to analyze and estimate customer behavior using

			clustering algorithms and data mining techniques.
2015	**You et al.**	• A real data from a Chinese company • RFM analysis + K means clustering + Decision tree	• To propose a model to accurately predict monthly supply quantity, using the RFM approach to select attributes to cluster customers into different groups. • This framework helped managers to identify the latent characteristics of different customer categories. • The model was also helpful to predict marketing strategies, which can greatly reduce inventory for every customer category
2014	**Zalaghi & Varzi**	• RFM analysis + K means clustering + Genetic algorithm	• A method used to obtain the behavioral traits of customers using the RFM approach. • For their suggested approach, the customers' records clustered and then the RFM model items were specified through selecting the effective properties

			on the customers' loyalty rate • Customer scores regarding to their loyalty for each cluster was calculated.

2) Literature Survey

A. Customer Classification

Over the years, the commercial world has become more competitive, as organizations such as these have to meet the needs and wants of their customers, attract new customers, and thus improve their businesses. The task of identifying and meeting the needs and requirements of each customer in the business is a very difficult task. This is because customers may vary according to their needs, wants, demographics, shapes, taste and taste, features and so on. As it is, it is a bad practice to treat all customers equally in business. This challenge has led to the adoption of the concept of customer segmentation or market segmentation, where consumers are divided into subgroups or segments where members of each subcategory exhibit similar market behaviors or features. Accordingly, customer segmentation is the process of dividing the market into indigenous groups.

B. Big Data

Recently, Big Data research has gained momentum. defines big data as - a term that describes a large number of formal and informal data, which cannot be analyzed using traditional methods and algorithms. Companies include billions of data about their customers, suppliers, and operations, and millions of internally connected sensors are sent to the real world on devices such as mobile phones and cars, sensing, creating, and communicating data. the ability to improve forecasting, save money, increase efficiency and improve decisionmaking in various fields such as traffic control, weather forecasting, disaster prevention, finance, fraud control, business transactions, national security, education, and healthcare. Big data is seen mainly in the three Vs namely: volume, variability and speed. There are other 2Vs available - authenticity and value, thus making it 5V.

C. Data Collection

Data collection is the process of collecting and measuring information against targeted variations in an established system, enabling one to answer relevant questions and evaluate results. Data collection is part of research in all fields of study including physical and social sciences, humanities and business. The purpose of all data collection is to obtain quality evidence that allows analysis to lead to the creation of convincing and misleading answers to the questions submitted. We collected data from the UCI Machine Learning Repository.

D. Clustering data

Clustering is the process of grouping the information in the dataset based on some similarities. There are a number of algorithms which can be chosen to be applied on a dataset based on the situation provided. However, no universal clustering algorithm exists that's why it becomes important to opt for appropriate clustering techniques. In this paper, we have implemented three clustering algorithms using python sk-learn library.

E. **K-Mean**
K- means that an algorithm is one of the most popular classification algorithms. This clustering algorithm depends on the centroid where each data point is placed in one of the overlapping K clusters pre-programmed into the algorithm, The clusters are created that correspond to the hidden pattern in the data that provides the information needed to help decide the execution process. There are many ways to make k-means assembling; we will use the elbow method.

3) Methodology

The data used in this paper were collected from the UCI Machine Learning Repository. This is a set of geographic data containing all transactions occurring between 1/1/2/10 and 9/12/2011 in an unregistered and unregistered UK broker. The company mainly sells unique gifts all together. Many of the company's customers are shopkeepers. The database contains 8 attributes. These attributes include:
InvoiceNo: Invoice number. By default, a 6-digit aggregate number is assigned separately for each transaction. If this code starts with the letter 'c', it indicates the cancellation.
StockCode Code: Product (item). Name, a 5-digit number assigned only to each unique product.
Definition: Product name (item). By name. ”
Price: The value of each product (item) made. Number. "
InvoiceDate: Invitation Date and Time. In terms of numbers, the date and time of each transaction.
UnitPrice: Price is a unit. Prices, product price per unit of measurement. "
Customer: Customer number. Name, 5-digit number assigned to each customer.
Country: Country name. Name, the name of the country where each customer lives.

In this project several steps were taken to obtain an accurate result. It involves the addition of a feature alongside the first step of the centroids, the allocation step and the update step, which are the most common steps k-means algorithms.

STEPS:

1. **Collect data**
This is a data preparation phase. The feature usually helps to refine all data items at a standard rate to improve the performance of the clustering algorithm.

Each data point changes from grade 2 to +2. Integration techniques that include Min-max, decimal and z-points The standard z-signing strategy are used to make things unequal before applying the k-Means algorithm to a dataset.

2. Customer Classification Methods

There are many ways to perform segmentation, which vary in severity, data requirements, and purpose. The following are some of the most commonly used methods, but this is not an incomplete list. There are papers that discuss artificial neural networks, particle fixation, and complex types of ensemble, but are not included due to limited exposure. In future articles, I may go into some of these alternatives, but for now, these more common methods should be sufficient.

Each subsequent section of this article will include a basic description of the method, as well as a code example for the method used. If you don't have the expertise, well, just skip the code and you'll still have to get a good handle on each of the 4 sub-sections we include in this article.

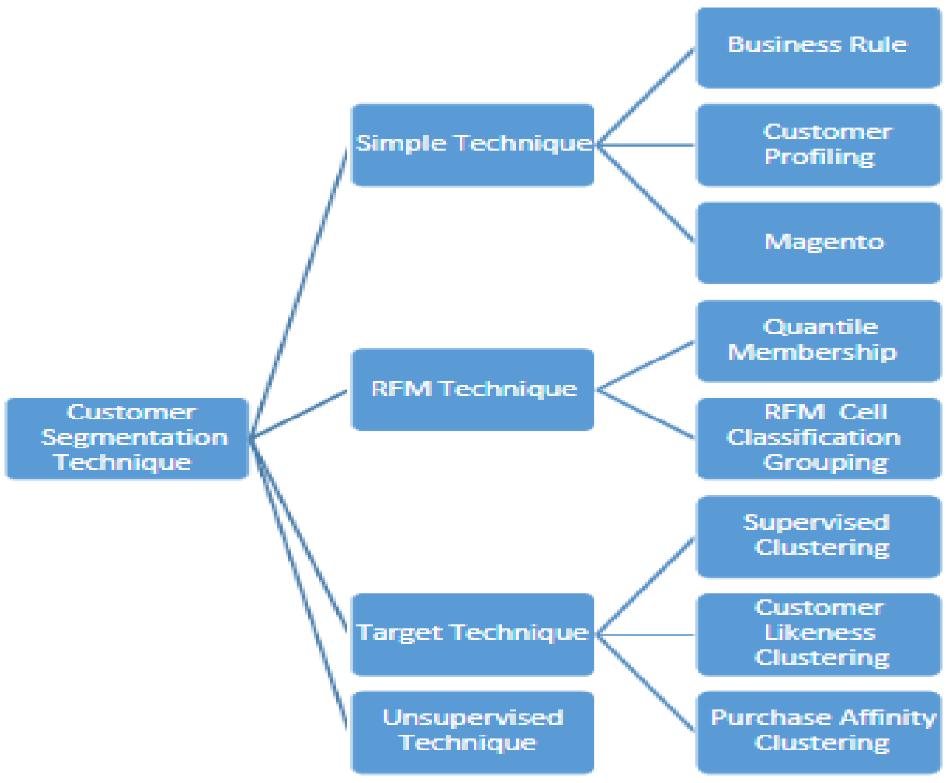


Figure 1. Customer Segmentation Classification

3. Group Analysis

Group analysis is a unifying, or unifying, approach for consumers based on their similarities.

There are 2 main types of group analysis categorized into market policy: Hierarchical group analysis, and classification (Miller, 2015). In the meantime, we will discuss how to classify clusters called k-methods.

4. **Key Metrics and othe KPIs¶**

   Monthly Revenue & Revenue Rate
   Monthly Active Customer
   Monthly Order Count
   Average Revenue Per order
   Monthly Order average
   Revenue per month for New and Existing customers
   New Customer Ratio

5. **K-means encounter**

   The k-means clustering algorithm is an algorithm that is frequently used to draw insights into the formats and differences within a database. In marketing, it is often used to build customer segments and to understand the behavior of these unique segments. Let's get into building assembly models in the python environment.

6. **Centroids Initiation**

   Selected cents or initials were selected. Figure 1 introduces the start of graduation centers. Four selected centers shown in different shapes were selected using the Forgy method. In Forgy's method of using k (in this case k = 4) data points are randomly selected as cluster centroids.

- **Technical Introduction**

  The code below was created in the Jupyter manual using Python 3.x and a few Python packages for editing, processing, analyzing, and visualizing information. The open-source data used in the following code comes from Cost Irvine's Machine Learning Repository.

4) Code Implementation

Import packages and data

We import all the necessary packages needed to do our analysis and then import the xlsx (excel spreadsheet) data file.

```
# Import key Libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from datetime import datetime
```

```
retail.head()
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom

Data Analysis, pre-processing & Feature engineering

Before proceeding with some more EDA and processing, lets start with engineering some of the new features identified in the Initial Data Review above, which can be quite straightforward off the back of the existing dataset:

1. Break down InvoiceNo to
   - InvoiceNumber - new feature
   - InvoiceCode - new feature

2. Break down InvoiceDate to
   - InvoiceYearMonth - new feature
   - InvoiceYear - new feature
   - InvoiceMonth - new feature

3. The rest of the Data Analysis and pre-processing will involve:
   - Definining Active population based on feature rules
   - Creating Masks for filtering out bad data (empty or Null Values, negative prices, etc...) that don't contribute towards the active population
   - Visualising some of our data to get a better idea of the population

Construct New features

Invoice

```
In [15]: # Breaking down Invoice Number into Code and Number
         retail['InvoiceNumber'] = retail.InvoiceNo.astype(str).str[-6:]
         retail['InvoiceCode'] = retail.InvoiceNo.astype(str).str[-7:-6]
```

Dates

```
In [16]: # Transforming InvoiceDate column to datetime type and mapping to a new columne as InvoiceDateTime
         retail['InvoiceDate'] = pd.to_datetime(retail['InvoiceDate'])
```

```
In [17]: # Construct Year, Month and YearMonth from Invoice Date field
         retail['Year'], retail['Month'] = retail['InvoiceDate'].dt.year, retail['InvoiceDate'].dt.month
         retail['YearMonth'] = retail['InvoiceDate'].map(lambda x: 100*x.year + x.month)
```

```
In [18]: # Create "Date" column in datetime format to use for index
         retail['Date'] = pd.to_datetime(retail.InvoiceDate.dt.date)
         retail.set_index('Date', inplace=True)
```

Processing and EDA on new features

```
In [23]: # Counting types of Invoices
         retail.InvoiceCode.value_counts()

Out[23]:      532618
         C      9288
         A         3
         Name: InvoiceCode, dtype: int64
```

```
In [24]: # Unique values
         retail.InvoiceCode.unique()

Out[24]: array(['', 'C', 'A'], dtype=object)
```

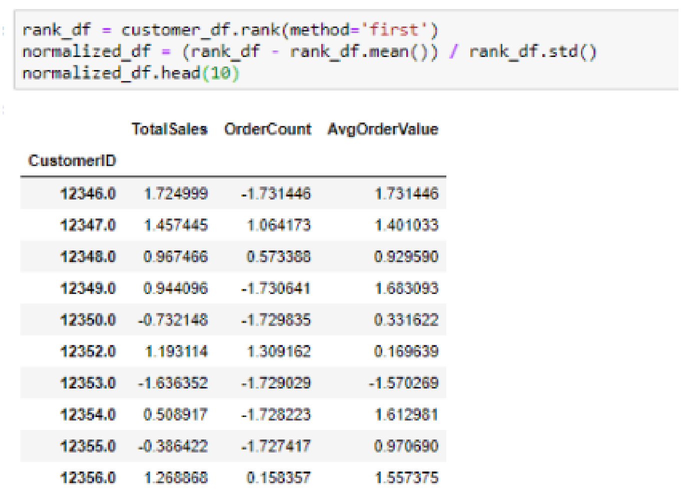
```
In [25]: # Replacing '' with 'N', to reflect Normal transactions
         retail.InvoiceCode.replace({'': 'N'}, inplace = True)
         retail.InvoiceCode.unique()

Out[25]: array(['N', 'C', 'A'], dtype=object)
```

```
In [26]: # Inspecting type A and C invoices
         retail[retail.InvoiceCode == "A"].head()
```

Masks for Data Processing

Creating various Masks to implement the rules per feature for narrowing down the active population

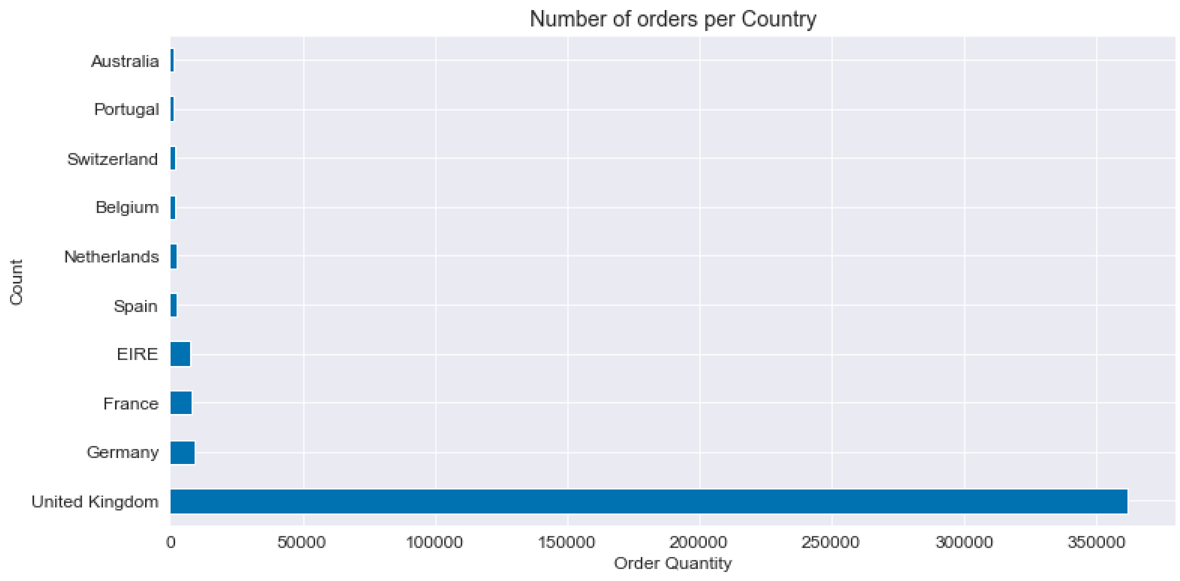
```
# Create a Masks Summary Dataframe to store and view the different count across masks
df_mask = pd.DataFrame(retail.columns)
df_mask.columns = ['Features']

df_mask['Valid Prices'] = retail[valid_price].count().values
df_mask['Valid Description'] = retail[valid_desc].count().values
df_mask['Valid Customer IDs'] = retail[valid_CID].count().values
df_mask['Invoice Normal'] = retail[inv_N].count().values
df_mask['Invoice Cancellation'] = retail[inv_C].count().values
df_mask['Invoice Adjustment'] = retail[inv_A].count().values
df_mask['Negative Quantities'] = retail[q_neg].count().values
df_mask['Positive Quantities'] = retail[q_pos].count().values
```



```
rank_df = customer_df.rank(method='first')
normalized_df = (rank_df - rank_df.mean()) / rank_df.std()
normalized_df.head(10)
```

CustomerID	TotalSales	OrderCount	AvgOrderValue
12346.0	1.724999	-1.731446	1.731446
12347.0	1.457445	1.064173	1.401033
12348.0	0.967466	0.573388	0.929590
12349.0	0.944096	-1.730641	1.683093
12350.0	-0.732148	-1.729835	0.331622
12352.0	1.193114	1.309162	0.169639
12353.0	-1.636352	-1.729029	-1.570269
12354.0	0.508917	-1.728223	1.612981
12355.0	-0.386422	-1.727417	0.970690
12356.0	1.268868	0.158357	1.557375

Data Inspection and visualisations



3 Key Performance Indicators - KPIS

Before proceeding into modelling, the following categories of KPIs will be examined in more detail:

- **Transaction KPIs (Revenue, Order count, Active Customers, etc...)**

1. **Monthly Revenue**
   Revenue = $\sum_{Month} UnitPrice\ x\ Quantity$∑MonthUnitPrice x Quantity (Across all active customers)

   - Group by YearMonth
   - Calculate Order Value for each row by multiplying Price with Quantity
   - Sum Order Values grouped by YearMonth

2. **Monthly Revenue Growth**
    - Month by month percentage comparison of Revenue change

3. **Active Customers**
    - Group: by YearMonth
    - Unique count of Customer IDs per grouping

4. **Monthly Order Count**
    - Group: by YearMonth
    - Count Invoices per grouping

5. **Average revenue per order**
    - Average monthly revenue above

- **Customer KPIs (New vs Old Customers, Retention, etc...)**
- **Product KPIs (Top selling product, etc...)**

RFM Modelling

In order to do Customer Segmentation, the RFM modelling technique will be used.

RFM stands for Recency - Frequency - Monetary Value with the following definitions:

1. **Recency** - Given a current or specific date in the past, when was the last time that the customer made a transaction
2. **Frequency** - Given a specific time window, how many transactions did the customer do during that window
3. **Monetary Value** or **Revenue** - Given a specific window, how much did the customer spend

Based on the values above, one could construct various segments that classify the customers to:

Low value Segment

- R - Less active
- F - Not very frequent
- M - Low spend

Mid value Segment

- R - Medium or inconsistent activity
- F - infrequent or frequent at specific times

- M - varied spending but overall in the mid of the scale

High value Segment

- R - Very active
- F - Very frequent
- M - Great spending profile

Approach

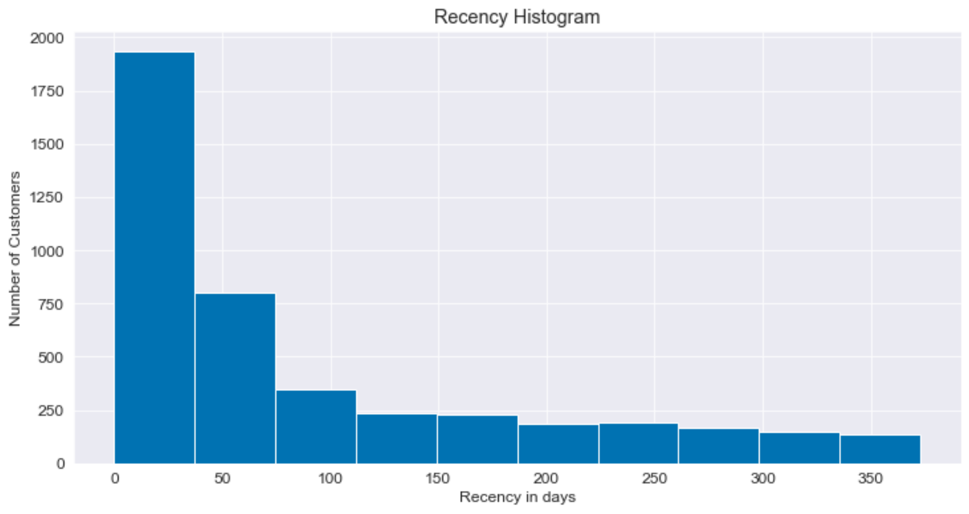
Going forward the approach for deriving customer segments would be:

- Calculate RFM scores individually and plot them to get an idea about their values and ranges
- Use K-Means clustering unsupervised learning algorithm to create these 3 clusters above
- Use techniques to optimise the number of clusters based on Silhouette and Inertia scores

Recency Score

```
In [91]: # Generate new data frame based on latest Invoice date from retail_ppp dataframe per Customer (groupby = CustomerID)
         recency = retail_ppp.groupby('CustomerID').InvoiceDate.max().reset_index()
         recency.columns = ['CustomerID','LastPurchaseDate']

         # Set observation point as the last invoice date in the dataset
         LastInvoiceDate = recency['LastPurchaseDate'].max()

         # Generate Recency in days by subtracting the Last Purchase date for each customer from the Last Invoice Date
         recency['Recency'] = (LastInvoiceDate - recency['LastPurchaseDate']).dt.days
```

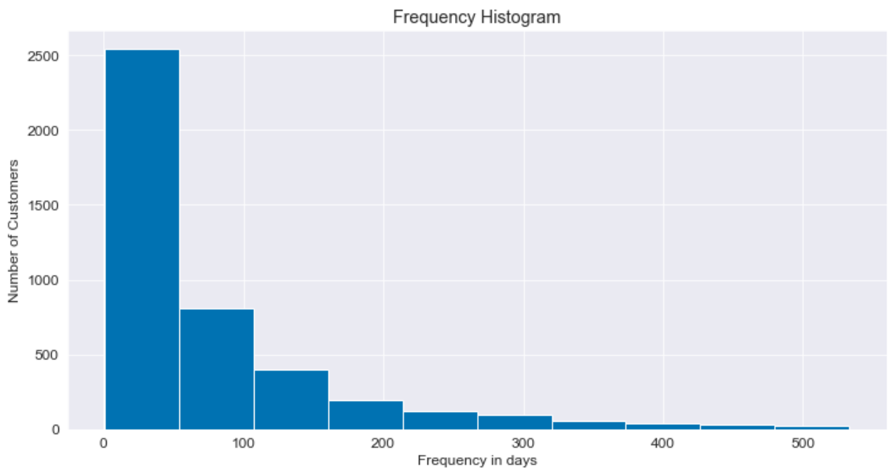
```
In [94]: # Plot Recency
         customer.Recency.plot.hist();
         plt.xlabel("Recency in days")
         plt.ylabel("Number of Customers")
         plt.title("Recency Histogram");
```



Frequency Score

Frequency metric reflects the number of orders per Customer, so a simple count of the invoices grouped per Customer ID would do

```
In [99]: # Plot Frequency
         # Frequency seems to have some outliers, with vey high frequency, but very few in numbers
         # In order to plot effectively and not have a skewed diagram, we've sorted the frequencies
         # and cropped the top 72 values in our diagram
         customer.Frequency.sort_values().head(4300).plot.hist();
         plt.xlabel("Frequency in days")
         plt.ylabel("Number of Customers")
         plt.title("Frequency Histogram");
```



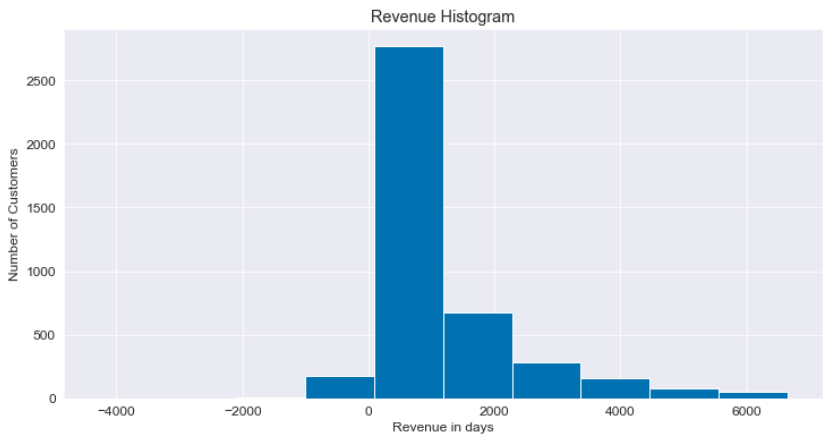
Monetary Value Score (Revenue)

```
In [100]: # Revenue per transaction has already been calculated as per KPIs section
          # Grouping revenue per Customer ID
          revenue = retail_ppp.groupby('CustomerID').Revenue.sum().reset_index()

          # Consolidate Revenue to existing Customer DataFrame
          customer = pd.merge(customer, revenue, on='CustomerID')
          customer.head()
```

Out[100]:

	CustomerID	Recency	Frequency	Revenue
0	17850.0	301	312	5288.63
1	13047.0	31	196	3079.10
2	12583.0	2	251	7187.34
3	13748.0	95	28	948.25
4	15100.0	329	6	635.10

```
In [103]: # Plot Revenue
          customer.Revenue.sort_values().head(4200).plot.hist();
          plt.xlabel("Revenue in days")
          plt.ylabel("Number of Customers")
          plt.title("Revenue Histogram");
```



```
print('Start date:' , df['InvoiceDate'].min())
print('End date:' , df['InvoiceDate'].max())
```

```
Start date: 2009-12-01 07:45:00
End date: 2010-12-09 20:01:00
```

```
# Create revenue colummn
df['Revenue'] = df['Price'] * df['Quantity']
# Convert to show date only
from datetime import datetime
df["InvoiceDate"] = df["InvoiceDate"].dt.date
```

```
import datetime as dt
snapshot_date = max(df.InvoiceDate) + dt.timedelta(days=1)
```

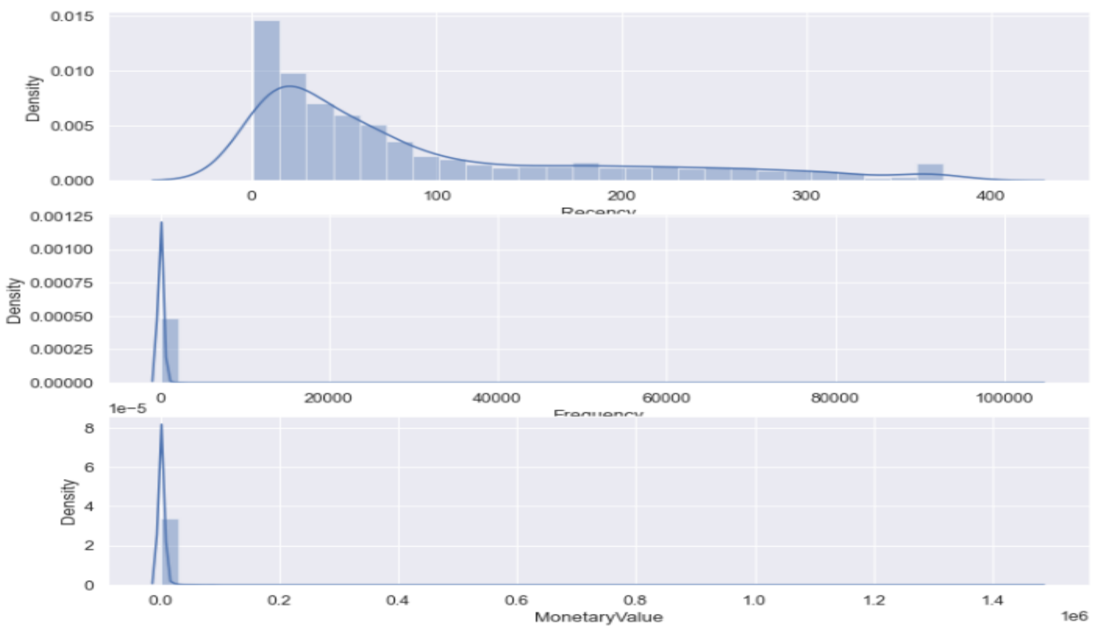
```
# RFM table
# Aggregate data by each customer
rfm = df.groupby('Customer ID').agg({'InvoiceDate': lambda x: (snapshot_date - x.max()).days,
                                     'Invoice': lambda x: len(x),
                                     'Revenue': lambda x: x.sum()}).reset_index()
rfm['InvoiceDate'] = rfm['InvoiceDate'].astype(int)

# Rename columns
rfm.rename(columns={'InvoiceDate': 'Recency',
                    'Invoice': 'Frequency',
                    'Revenue': 'MonetaryValue'}, inplace=True)
```

```
rfm.head()
```

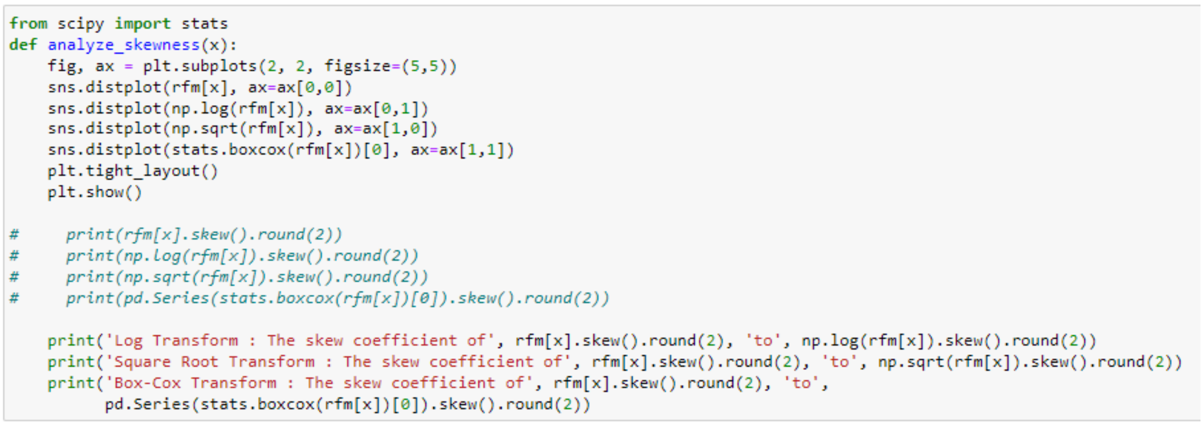
	Customer ID	Recency	Frequency	MonetaryValue
0	12346.0	165	33	372.86
1	12347.0	3	71	1323.32
2	12348.0	74	20	222.16
3	12349.0	43	102	2671.14
4	12351.0	11	21	300.93

Manage Skewness and Scaling

```
plt.figure(figsize=(12,10))

# Plot recency distribution
plt.subplot(3, 1, 1); sns.distplot(rfm['Recency'])

# Plot frequency distribution
plt.subplot(3, 1, 2); sns.distplot(rfm['Frequency'])

# Plot monetary value distribution
plt.subplot(3, 1, 3); sns.distplot(rfm['MonetaryValue'])

# Show the plot
plt.show()
```

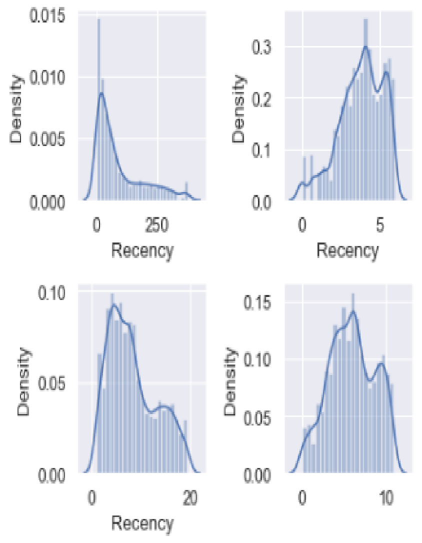


- log transformation
- square root transformation
- box-cox transformation Note: We can use the transformation if and only if the variable only has positive values.



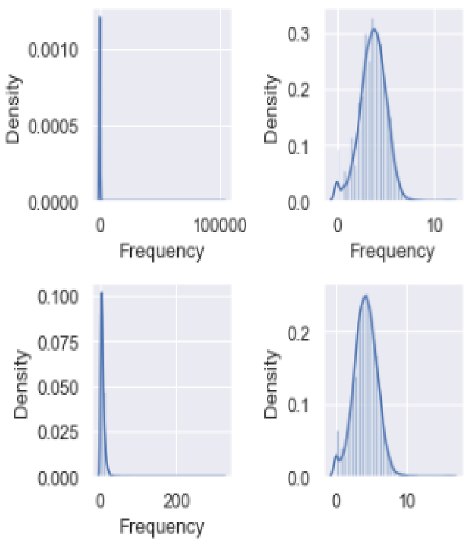
```
from scipy import stats
def analyze_skewness(x):
    fig, ax = plt.subplots(2, 2, figsize=(5,5))
    sns.distplot(rfm[x], ax=ax[0,0])
    sns.distplot(np.log(rfm[x]), ax=ax[0,1])
    sns.distplot(np.sqrt(rfm[x]), ax=ax[1,0])
    sns.distplot(stats.boxcox(rfm[x])[0], ax=ax[1,1])
    plt.tight_layout()
    plt.show()

#     print(rfm[x].skew().round(2))
#     print(np.log(rfm[x]).skew().round(2))
#     print(np.sqrt(rfm[x]).skew().round(2))
#     print(pd.Series(stats.boxcox(rfm[x])[0]).skew().round(2))

    print('Log Transform : The skew coefficient of', rfm[x].skew().round(2), 'to', np.log(rfm[x]).skew().round(2))
    print('Square Root Transform : The skew coefficient of', rfm[x].skew().round(2), 'to', np.sqrt(rfm[x]).skew().round(2))
    print('Box-Cox Transform : The skew coefficient of', rfm[x].skew().round(2), 'to',
          pd.Series(stats.boxcox(rfm[x])[0]).skew().round(2))
```

```
analyze_skewness('Recency')
```



```
Log Transform : The skew coefficient of 1.28 to -0.62
Square Root Transform : The skew coefficient of 1.28 to 0.59
Box-Cox Transform : The skew coefficient of 1.28 to -0.07
```

```
analyze_skewness('Frequency')
```



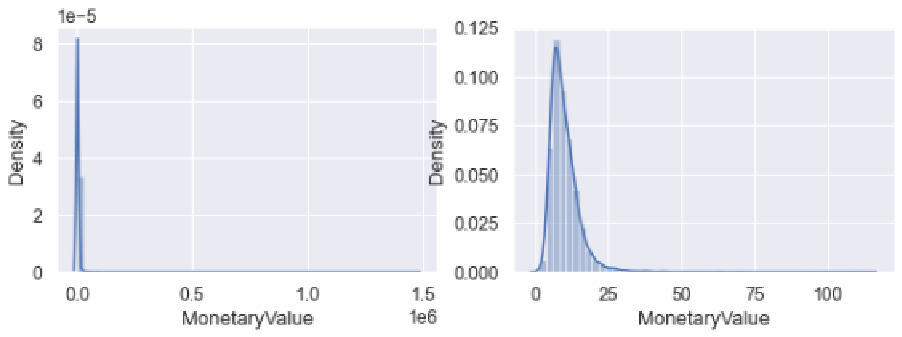
```
Log Transform : The skew coefficient of 64.17 to -0.27
Square Root Transform : The skew coefficient of 64.17 to 19.51
Box-Cox Transform : The skew coefficient of 64.17 to 0.02
```

Based on that visualization, it shows that the variables with box-cox transformation shows a more symmetrical form rather than the other transformations. To make sure, we calculate each variable using the skew function. The result:

Skew Coefficient	Recency	Frequency
Log Transform	1.28 to -0.62	64.17 to -0.27
Square Root Transform	1.28 to 0.59	64.17 to 19.51
Box-Cox Transform	1.28 to -0.07	64.17 to 0.02

Based on that calculation, we will utilize variables that use box-cox transformations. Except for the MonetaryValue variable because the variable includes negative values. To handle this variable, we can use cubic root transformation to the data.

```
fig, ax = plt.subplots(1, 2, figsize=(10,3))
sns.distplot(rfm['MonetaryValue'], ax=ax[0])
sns.distplot(np.cbrt(rfm['MonetaryValue']), ax=ax[1])
plt.show()
print(rfm['MonetaryValue'].skew().round(2))
print(np.cbrt(rfm['MonetaryValue']).skew().round(2))
```



```
53.9
4.25
```

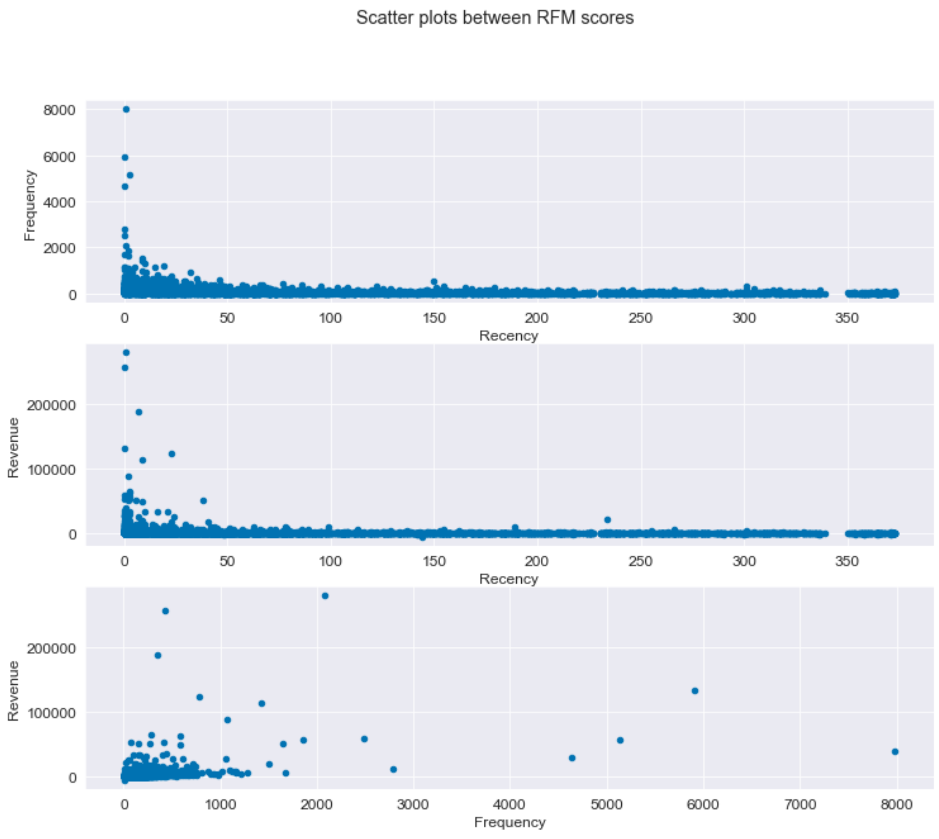
By using the transformation, we will have data that less skewed. **The skewness value declines from 16.63 to 1.16**. Therefore, we can transform the RFM table with this code,

```
# Set the Numbers
from scipy import stats
customers_fix = pd.DataFrame()
customers_fix["Recency"] = stats.boxcox(rfm['Recency'])[0]
customers_fix["Frequency"] = stats.boxcox(rfm['Frequency'])[0]
customers_fix["MonetaryValue"] = pd.Series(np.cbrt(rfm['MonetaryValue'])).values
customers_fix.tail()
```

	Recency	Frequency	MonetaryValue
4308	6.419299	3.657323	7.728829
4309	10.213088	2.662843	7.530248
4310	7.612872	4.731003	10.903929
4311	3.844546	5.032893	13.286934
4312	0.000000	16.140682	113.794050

Each variable don’t have the same mean and variance. We have to normalize it. To normalize, we can use **StandardScaler** object from scikit-learn library to do it. The code will look like this,

```
In [105]: # Finally lets review the scatter plots between the different scores

          fig, (ax1, ax2, ax3) = plt.subplots(3)
          fig.suptitle('Scatter plots between RFM scores')
          customer.plot.scatter(x = 'Recency', y = 'Frequency', ax = ax1, figsize = (12,10));
          customer.plot.scatter(x = 'Recency', y = 'Revenue', ax = ax2);
          customer.plot.scatter(x = 'Frequency', y = 'Revenue', ax = ax3);
```



K-Means Clustering

To make segmentation from the data, we can use the K-Means algorithm to do this.

K-Means algorithm is an unsupervised learning algorithm that uses the geometrical principle to determine which cluster belongs to the data. By determine each centroid, we calculate the distance to each centroid. Each data belongs to a centroid if it has the smallest distance from the other. It repeats until the next total of the distance doesn’t have significant changes than before.

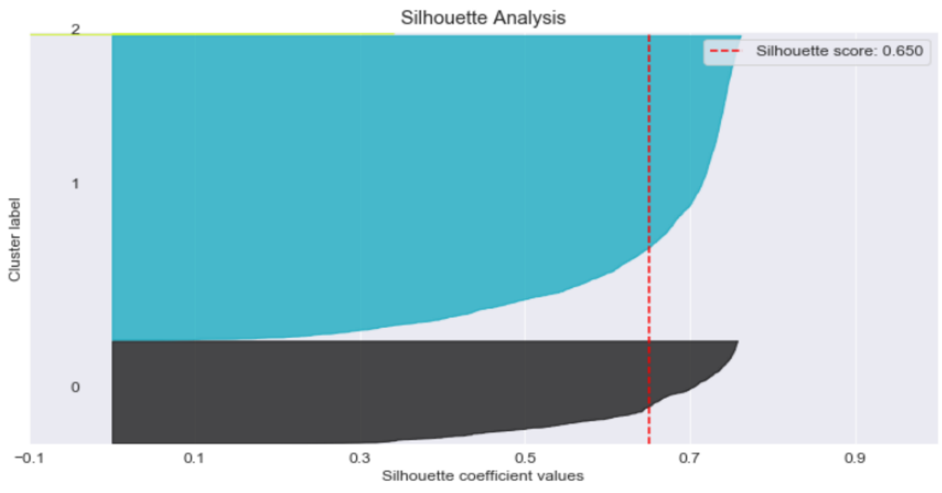
Determine the Optimal K

Evaluation Metrics - Silhouette and Elbow method scores

To make our clustering reach its maximum performance, we have to determine which hyperparameter fits to the data. To determine which hyperparameter is the best for our model and data, we can use the elbow method or silhouette method to decide.

Silhouette (Clustering):
Silhouette means how to interpret and verify consistency within data structures. This method provides a picture showing how well each item is organized. [1]
The value of a silhouette is a measure of how something is similar in its collections (combinations) compared to other clusters (divisions). **The silhouette goes from –1 to +1, where a higher value indicates that an item is properly matched to its collection and compared to neighbouring clusters**. If multiple objects have a high value, then the integration configuration is appropriate. If most points have a value or a negative value, then the coordinate system may have too many or too few clusters.
The silhouette can be calculated with any distance metric, such as Euclidean distance or Manhattan distance.
Now that we know a whole lot more of the silhouette, let's go in and use the

```
In [115]: from sklearn import metrics
          metrics.silhouette_score(Xstd, cluster_labels, metric='euclidean')

Out[115]: 0.6501286458518293

In [116]: model.inertia_

Out[116]: 5408.404670803728

In [117]: # Plotting Silhouette Score
          from scikitplot.metrics import plot_silhouette
          plot_silhouette(Xstd, cluster_labels);
```

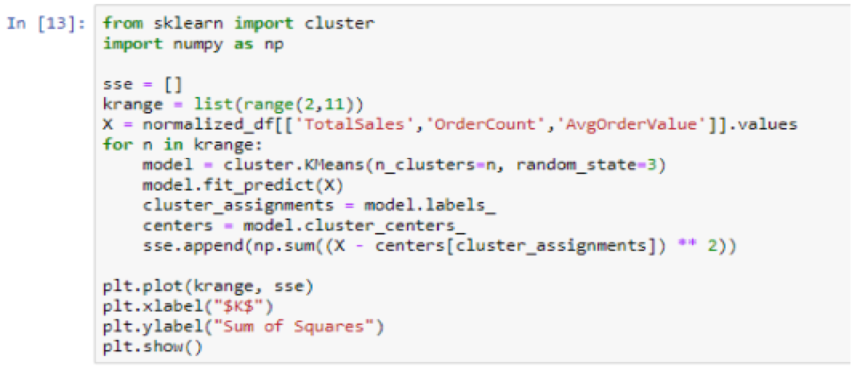


Cluster 4 had the most complete silhouette fit, indicating that 4 could be the best number of clusters. But we'll look at that twice with the elbow way.

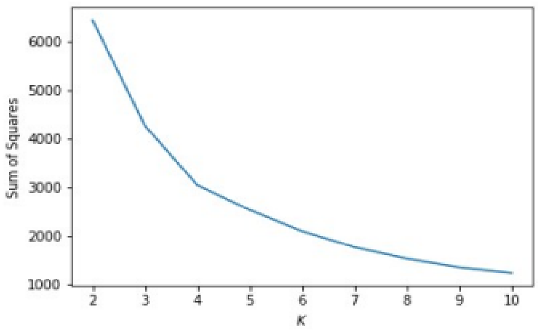
Elbow Criterion Method(with the Sum of Squared Errors (SSE)):

The idea behind the elbow method is to run the k-mean correlation in the given data for k values (num_clusters, e.g. k = 1 to 10), and for each k value, to calculate the sum of squared errors (SSE).
After that, adjust the SSE line for each k value. If the line graph looks like an

arm - a red circle below the line of the line (as an angle), the "elbow" on the arm is the correct price (collection value). Here, we want to reduce the SSE. SSE usually drops to 0 as we go up k (and SSE is 0 where k equals the number of data points, because where each data point is its own set, and there is no error between it and its trunk).
Therefore the purpose is to select a small value of k that still has a low SSE, and the cone usually represents where it starts to have a negative return with increasing k.



```
In [13]: from sklearn import cluster
         import numpy as np

         sse = []
         krange = list(range(2,11))
         X = normalized_df[['TotalSales','OrderCount','AvgOrderValue']].values
         for n in krange:
             model = cluster.KMeans(n_clusters=n, random_state=3)
             model.fit_predict(X)
             cluster_assignments = model.labels_
             centers = model.cluster_centers_
             sse.append(np.sum((X - centers[cluster_assignments]) ** 2))

         plt.plot(krange, sse)
         plt.xlabel("$K$")
         plt.ylabel("Sum of Squares")
         plt.show()
```



Based on the graph above, it looks like K = 4, or 4 clusters is the correct number of clusters in this analysis. Now let's translate the customer segments provided by these components

Fit the model - 3 cluster

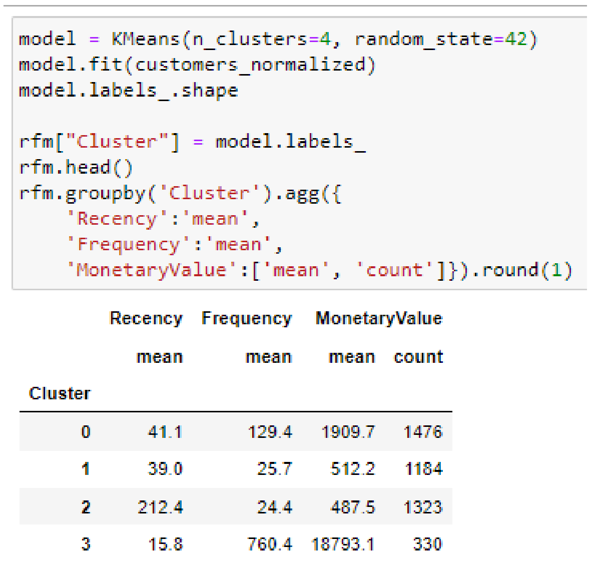
```
rfm["Cluster"] = model.labels_
rfm.head()
```

	Customer ID	Recency	Frequency	MonetaryValue	Cluster
0	12346.0	165	33	372.86	1
1	12347.0	3	71	1323.32	2
2	12348.0	74	20	222.16	1
3	12349.0	43	102	2671.14	2
4	12351.0	11	21	300.93	2

```
rfm.groupby('Cluster').agg({
    'Recency':'mean',
    'Frequency':'mean',
    'MonetaryValue':['mean', 'count']}).round(1)
```

	Recency	Frequency	MonetaryValue	
	mean	mean	mean	count
Cluster				
0	21.8	449.6	9983.6	748
1	181.9	20.4	433.9	1606
2	43.2	69.2	1075.6	1959

Fit the model - 4 cluster



```
model = KMeans(n_clusters=4, random_state=42)
model.fit(customers_normalized)
model.labels_.shape

rfm["Cluster"] = model.labels_
rfm.head()
rfm.groupby('Cluster').agg({
    'Recency':'mean',
    'Frequency':'mean',
    'MonetaryValue':['mean', 'count']}).round(1)
```

	Recency	Frequency	MonetaryValue	
	mean	mean	mean	count
Cluster				
0	41.1	129.4	1909.7	1476
1	39.0	25.7	512.2	1184
2	212.4	24.4	487.5	1323
3	15.8	760.4	18793.1	330

From the above table, we compared the distribution of mean values of recency, frequency, and monetary metrics across 3 and 4 cluster data. It seems that we get a more detailed distribution of our customer base using k=4.

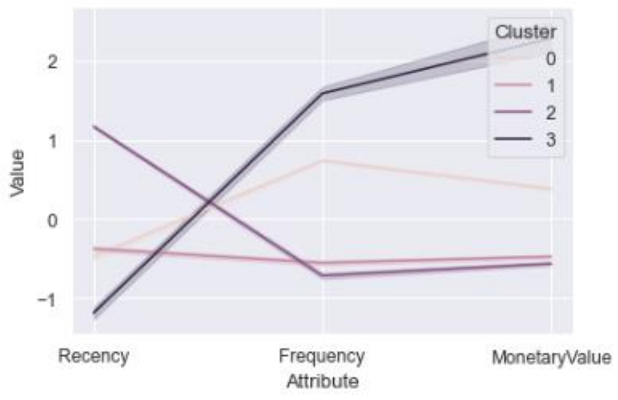
Another commonly used method to compare the cluster segments is Snakeplots. They are commonly used in marketing research to understand customer perceptions.

We built a snake plot for our data with 4 clusters.

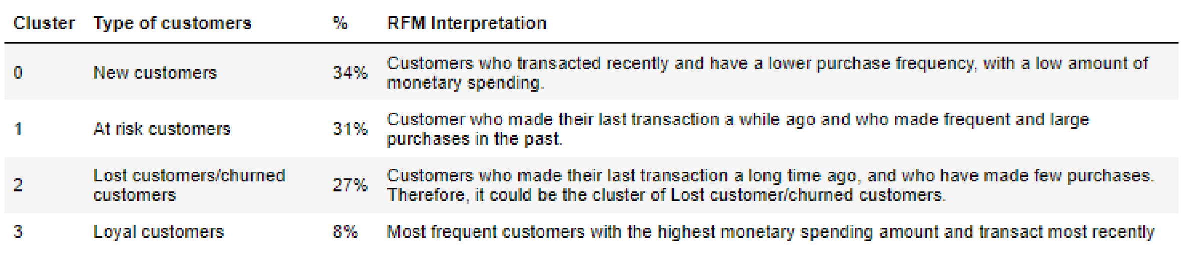
Cluster Exploration and Visualization

Snake Plots

Besides that, we can analyze the segments using snake plot. It requires the normalized dataset and also the cluster labels. By using this plot, we can have a good visualization from the data on how the cluster differs from each other.

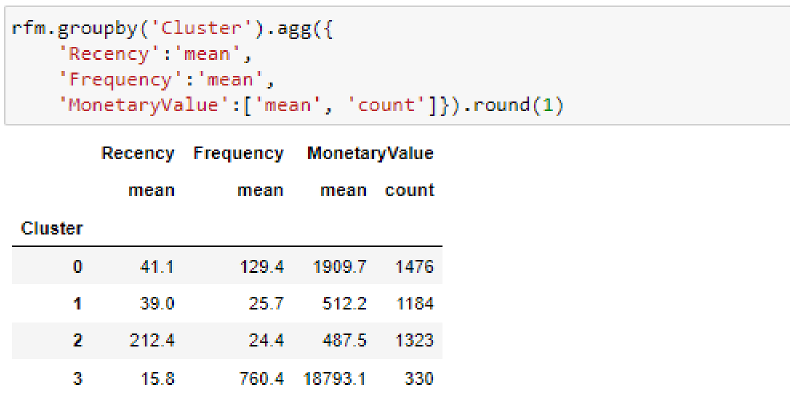


From the above snake plot, we can see the distribution of recency, frequency, and monetary metric values across the four clusters. The four clusters seem to be separate from each other, which indicates a good heterogeneous mix of clusters.



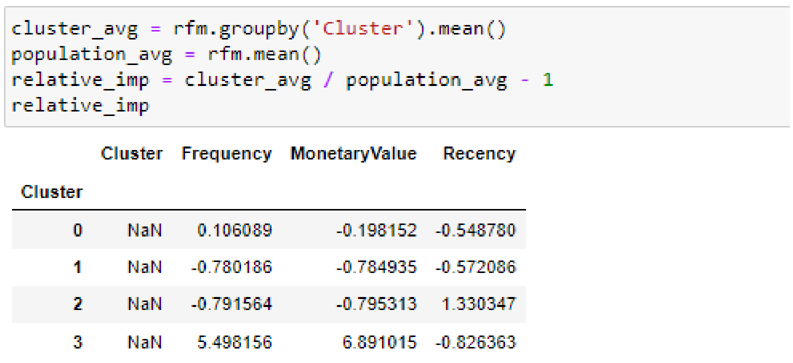
Cluster	Type of customers	%	RFM Interpretation
0	New customers	34%	Customers who transacted recently and have a lower purchase frequency, with a low amount of monetary spending.
1	At risk customers	31%	Customer who made their last transaction a while ago and who made frequent and large purchases in the past.
2	Lost customers/churned customers	27%	Customers who made their last transaction a long time ago, and who have made few purchases. Therefore, it could be the cluster of Lost customer/churned customers.
3	Loyal customers	8%	Most frequent customers with the highest monetary spending amount and transact most recently

Comparing the different of rfm between population and clusters



```
rfm.groupby('Cluster').agg({
    'Recency':'mean',
    'Frequency':'mean',
    'MonetaryValue':['mean', 'count']}).round(1)
```

```
        Recency  Frequency  MonetaryValue
           mean       mean     mean  count
Cluster
      0    41.1      129.4   1909.7   1476
      1    39.0       25.7    512.2   1184
      2   212.4       24.4    487.5   1323
      3    15.8      760.4  18793.1    330
```



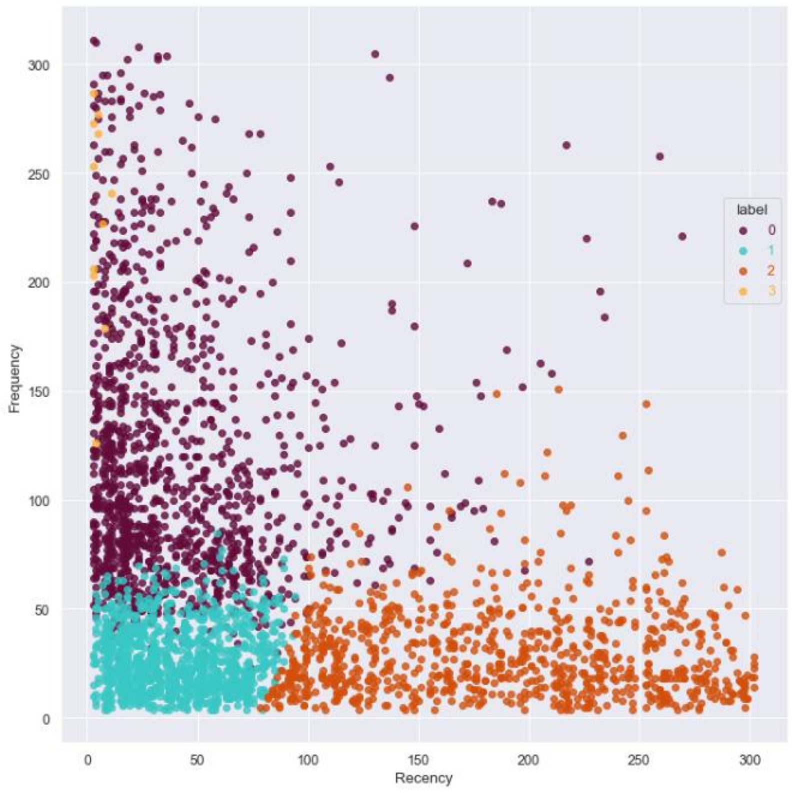
```
cluster_avg = rfm.groupby('Cluster').mean()
population_avg = rfm.mean()
relative_imp = cluster_avg / population_avg - 1
relative_imp
```

```
        Cluster  Frequency  MonetaryValue   Recency
Cluster
      0     NaN   0.106089      -0.198152 -0.548780
      1     NaN  -0.780186      -0.784935 -0.572086
      2     NaN  -0.791564      -0.795313  1.330347
      3     NaN   5.498156       6.891015 -0.826363
```

Scatter Plot

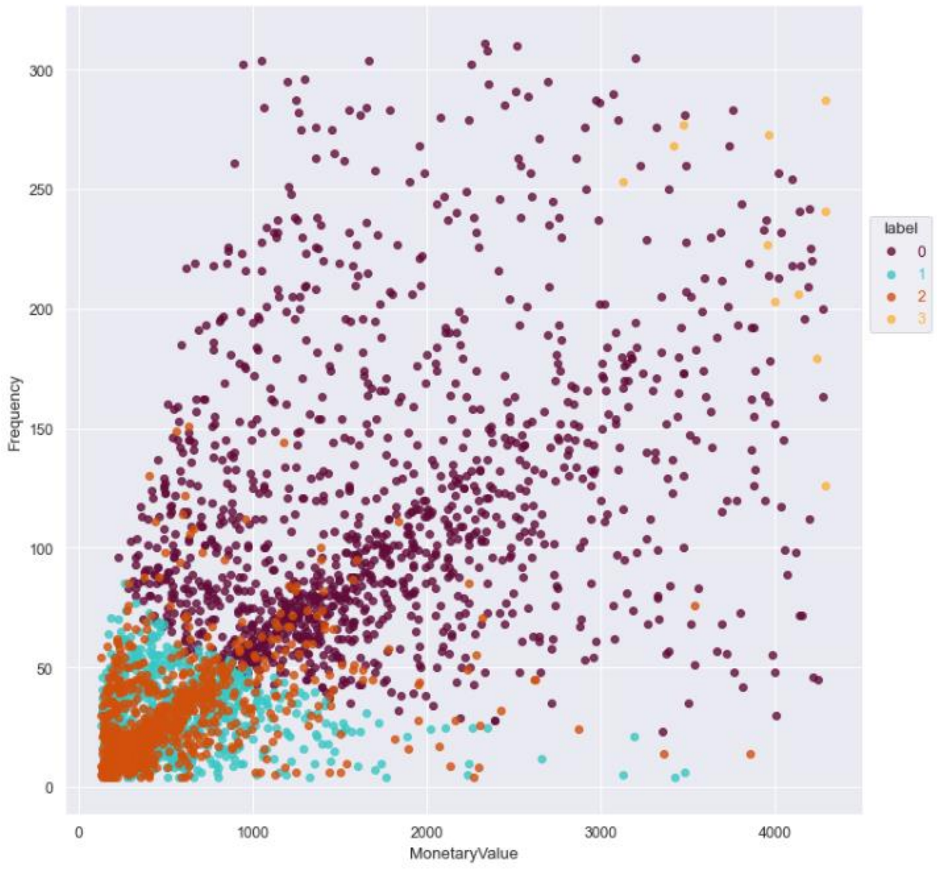
The scatter plot is the data analysis method we use when we have more than two variables. Remove the outlier from the plot to create a clear visualization result. Those outliers are taken into consideration in the model development. Exclude them only for visualization purposes.

Recency Vs frequency



A high frequency is found with customers who have a recent purchase within a month.

Frequency Vs Monetary



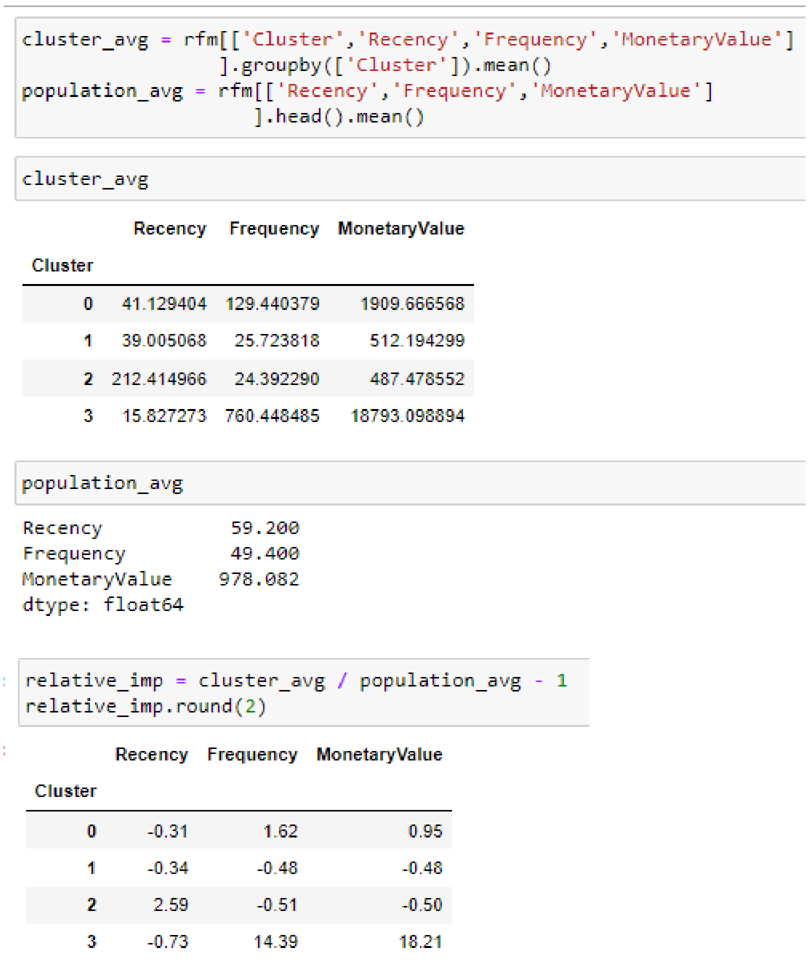
Customers who buy frequently spend less money.

Recency Vs Frequency Vs Monetary

In the above plot, the color specifies Cluster. From the above plot, we can see how the customers are spread among Recency, Frequency and Monetary dimension. Customers in Cluster 1 have made recent purchases with a high frequency, but with lower amounts. The reason for this could be that the customer frequently purchase Accessories that are not so expensive.
We can also use the following method to understand the relative importance of segments in the data. To do this, we will complete the following steps:

1. Calculate average values of each cluster
2. Calculate average values of population
3. Calculate importance score by dividing them and subtracting 1 (ensures 0 is returned when cluster average equals population average)

As the final step in this analysis, we can extract this information now for each customer that can be used to map the customer with their relative importance by the company:



```
cluster_avg = rfm[['Cluster','Recency','Frequency','MonetaryValue']
                  ].groupby(['Cluster']).mean()
population_avg = rfm[['Recency','Frequency','MonetaryValue']
                     ].head().mean()
```

```
cluster_avg
```

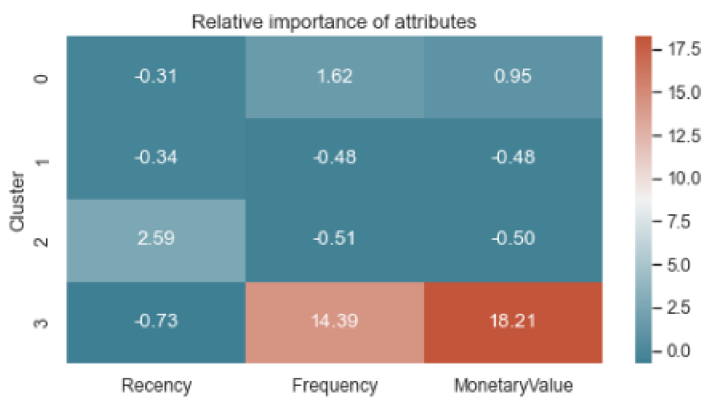
Cluster	Recency	Frequency	MonetaryValue
0	41.129404	129.440379	1909.666568
1	39.005068	25.723818	512.194299
2	212.414966	24.392290	487.478552
3	15.827273	760.448485	18793.098894

```
population_avg
```

```
Recency           59.200
Frequency         49.400
MonetaryValue    978.082
dtype: float64
```

```
relative_imp = cluster_avg / population_avg - 1
relative_imp.round(2)
```

Cluster	Recency	Frequency	MonetaryValue
0	-0.31	1.62	0.95
1	-0.34	-0.48	-0.48
2	2.59	-0.51	-0.50
3	-0.73	14.39	18.21



Result:

From the above analysis, we can see that there should be 4 clusters in our data. To understand what these 4 clusters mean in a business scenario, we should look back the table comparing the clustering performance of 3 and 4 clusters for the mean values of recency, frequency, and monetary metric.

Below is the table giving the RFM interpretation of each segment and the points that a company is recommended to keep in mind while designing the marketing strategy for that segment of customers.

Cluster	Type of customers	%	RFM Label	RFM Interpretation	Actioinable insight
0	New customers	34%	45% Loyal customers	Customers who transacted recently and have a lower purchase frequency, with a low amount of monetary spending. According to the RFM segmentation, half of them are loyal customers.	To convert them into Champion customers, the Online Store should engage them more frequently
1	At risk customers	31%	28% About to Sleep; 33% Potential loyalists	Customer who made their last transaction a while ago and who made frequent and large purchases in the past. RFM segmentation shows that 30% of them are Potential Loyalists and 30% are About to Sleep.	The online store should offer the customer popular products at a discount or reconnect the customer
2	Lost customers/churned customers	27%	70% Hibernating	Customers who made their last transaction a long time ago, and who have made few purchases. Therefore, it could be the cluster of Lost customer/churned customers. Additionally, RFM segmentation shows that most customers are Hibernating in this cluster	A marketing campaign should be launched to revive interest in online stores
3	Loyal customers	8%	67% Champions	Most frequent customers with the highest monetary spending amount and transact most recently	The online store should upsell higher value products and solicit reviews from this group of customers to provide better service and products

Using the RFM segmentation to identify the type of customer according to RFM score

Getting the individual RFM scores

Getting the individual RFM score can be done in several ways. You could use your own business expertise and heuristics to make rankings that suit your customer base. For this case, we are going to go the statistical route and rank our customer using quartiles.

The ranking of the individual RFM scores is done by dividing each of the RFM values into quartiles which creates four more or less equal buckets. We then rank each bucket from one to four; four being the best.

Calculate the overall RFM score

This step can be done in two ways:

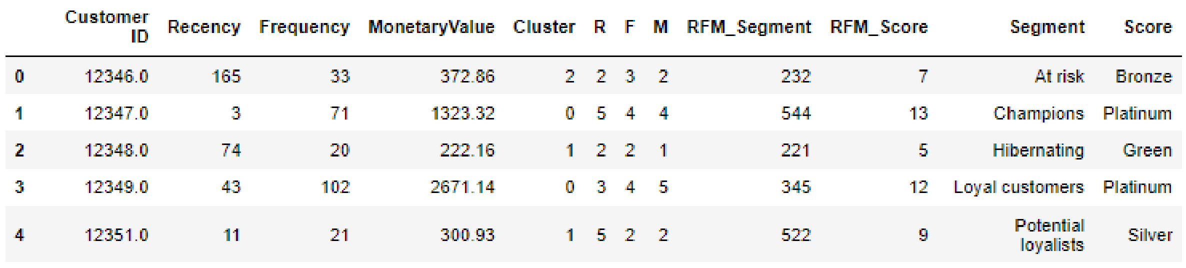
- Concatenation: creates segments Here we just concatenate (not add) the individual RFM score like strings and get labeled segments in return. Our best segment will be 444 and our worst will be 111 — signifying the lowest score on all three of the RFM categories.
- Addition: creates a score Here we add the individual RFM scores like numbers and get a number in return indicating the customer score. The score will range from 3 to 12 and we can use this to create more human friendly labelled categories.

After calculations on the RFM data we can create customer segments that are actionable and easy to understand — like the ones below:

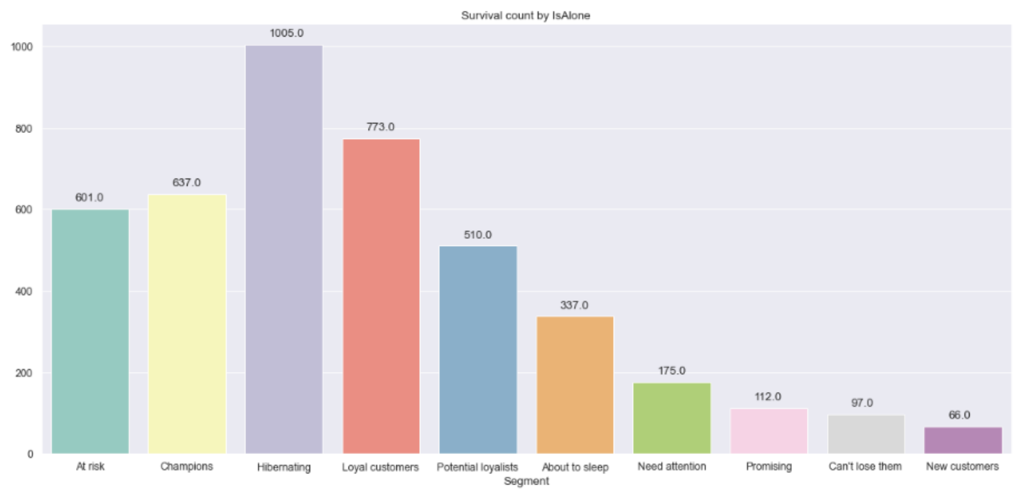
- Champions: Bought recently, buy often and spend the most
- Loyal customers: Buy on a regular basis. Responsive to promotions.
- Potential loyalist: Recent customers with average frequency.
- Recent customers: Bought most recently, but not often.
- Promising: Recent shoppers, but haven’t spent much.
- Needs attention: Above average recency, frequency and monetary values. May not have bought very recently though.
- About to sleep: Below average recency and frequency. Will lose them if not reactivated.
- At risk: Some time since they’ve purchased. Need to bring them back!
- Can’t lose them: Used to purchase frequently but haven’t returned for a long time.
- Hibernating: Last purchase was long back and low number of orders. May be lost.

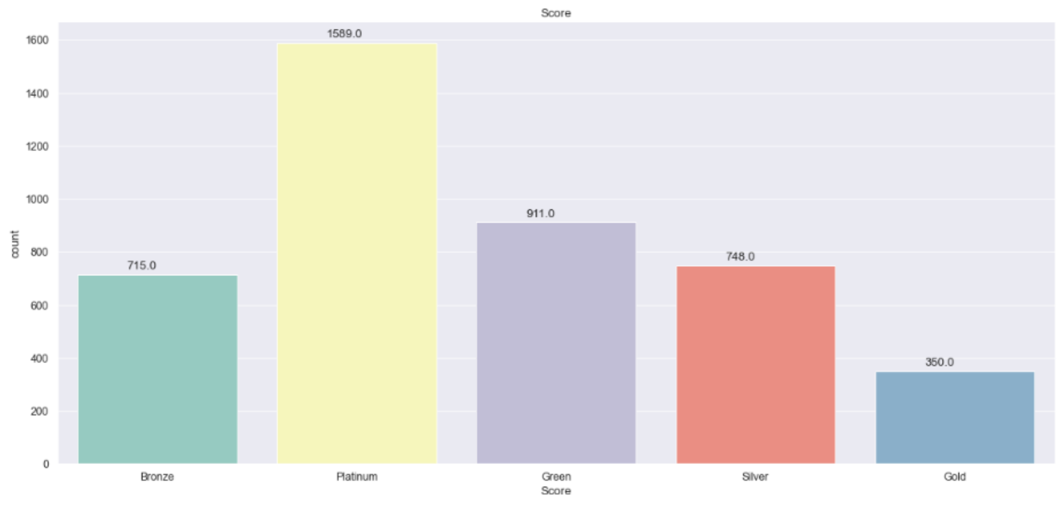
Grouping and labelling with RFM label

For the RFM segment we are going to use the most common naming scheme, as outlined above.



	Customer ID	Recency	Frequency	MonetaryValue	Cluster	R	F	M	RFM_Segment	RFM_Score	Segment	Score
0	12346.0	165	33	372.86	2	2	3	2	232	7	At risk	Bronze
1	12347.0	3	71	1323.32	0	5	4	4	544	13	Champions	Platinum
2	12348.0	74	20	222.16	1	2	2	1	221	5	Hibernating	Green
3	12349.0	43	102	2671.14	0	3	4	5	345	12	Loyal customers	Platinum
4	12351.0	11	21	300.93	1	5	2	2	522	9	Potential loyalists	Silver





	Segment	Count	percent
0	About to sleep	337	7.8
1	At risk	601	13.9
2	Can't lose them	97	2.2
3	Champions	637	14.8
4	Hibernating	1005	23.3

	Score	Count	percent
0	Bronze	715	16.6
1	Gold	350	8.1
2	Green	911	21.1
3	Platinum	1589	36.8
4	Silver	748	17.3

Using RFM segment to Interpret the result

	Cluster	Segment	Count	percent
0	0	About to sleep	2	0.1
1	0	At risk	156	10.6
2	0	Can't lose them	88	6.0
3	0	Champions	411	27.8
4	0	Hibernating	4	0.3
5	0	Loyal customers	664	45.0
6	0	Need attention	45	3.0
7	0	Potential loyalists	106	7.2
8	1	About to sleep	330	27.9
9	1	At risk	55	4.6
10	1	Champions	5	0.4
11	1	Hibernating	72	6.1
12	1	Loyal customers	19	1.6
13	1	Need attention	130	11.0
14	1	New customers	66	5.6
15	1	Potential loyalists	395	33.4
16	1	Promising	112	9.5
17	2	About to sleep	5	0.4
18	2	At risk	386	29.2
19	2	Can't lose them	4	0.3
20	2	Hibernating	928	70.1
21	3	At risk	4	1.2
22	3	Can't lose them	5	1.5
23	3	Champions	221	67.0
24	3	Hibernating	1	0.3
25	3	Loyal customers	90	27.3
26	3	Potential loyalists	9	2.7

FUTURE WORK

The proposed basic cluster model given the lack of details about the changes in customer behaviours. Therefore different rules and strategies are necessary to find the hidden patterns and shopping trends of the customers. RFM and K-means helped to find clusters of potential customers. In addition to this Cross Selling and Market Basket Analysis techniques can be used to analyse and offer additional products to customers as a suggestion in the hope that they would buy benefiting the customer and the retail establishment. Addition of new variables like Tenure: The number of days since the first transaction by each customer. This will tell us how long each customer has been with the system. Conducting deeper segmentation on customers based on their geographical location, and demographic and psychographic factors.

CONCLUSION

Based on RFM analysis, there are 8% of loyal customers who tend to spend big amount of money while buying. Also there are groups of customers who are already lost and who are going to be lost in near future.

Specific actions can be planned across the Business (Operations, Marketing, Product, etc...) to address any potential issues;

1. High Value:
- Improve Retention of these customers as they are the most valuable asset

2. Mid Value:
- Increase Rention and Frequency and bring them closer to the brand and the product so eventually they become High Value

3. Low Value:
- Increase Frequency and understand if there are any potential issues around the product or service

REFERENCES

- Blanchard, Tommy. Bhatnagar, Pranshu. Behera, Trash. (2019). Marketing Analytics Scientific Data: Achieve your marketing objectives with Python's data analytics capabilities. S.l: Packt printing is limited
- Griva, A., Bardaki, C., Pramatari, K., Papakiriakopoulos, D. (2018). Sales business analysis: Customer categories use market basket data. Systems Expert Systems, 100, 1-16.
- Hong, T., Kim, E. (2011). It separates consumers from online stores based on factors that affect the customer's intention to purchase. Expert System Applications, 39 (2), 2127-2131.
- Hwang, Y. H. (2019). Hands-on Advertising Science Data: Develop your machine learning marketing strategies… using python and r. S.l: Packt printing is limited ISSN: 0975-5853. Volume 12 Issue 1.
- T.Nelson Gnanaraj, Dr.K.Ramesh Kumar N.Monica. AnuManufactured

cluster analysis using a new algorithm from structured and unstructured data. International Journal of Advances in Computer Science and Technology. 2007. Volume 3, No.2.

- McKinsey Global Institute. Big data. The next frontier is creativity, competition and productivity. 2011. Jean Yan. - Big Data, Big Opportunities- Domains of Data.gov: Promote,
  lead, contribute, and collaborate in the big data era. 2013. A.K. Jain, M.N. Murty and P.J. Flynn. Data Integration: A Review . ‖ Global Journal of Management and Business Publisher ‖ Global Journal of Management and Business Publisher
  ACM Computer Research. 1999. Vol. 31, No. 3.
- Vishish R. Patel1 and Rupa G. Mehta. MpImpact for External Removal and Standard Procedures for JCSI International International Science Issues Issues, Vol. 8, Appeals 5, No 2, September 2011 ISSN (Online): 1694-0814