TABLE OF NOTATIONS AND ABBREVIATIONS

Abbreviation	Meaning
RFM segmentation	The segmentation of customer based on the ranking of three characteristics: Recency, Frequency and Monetary.
K-means clustering	A method of vector quantization, originally from signal processing, that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

ABSTRACT

The trend of using data science to help improve the quality of work has profoundly affected every profession around the world. The e-commerce industry is no exception to that trend. E-commerce is a segment that generates and operates based on a large volume of data. Thus, for e-commerce businesses, being able to utilize and extract information from such data can be vital to improving efficiency, profitability, and competitiveness.

In this report, we will study a specific E-Commerce dataset, develop questions, and draw insights to make suggestions for the corresponding company. The report will tackle two main problems: Sales Analysis and Customer Segmentation, of which results can help improve the sales, marketing, and customer service operations of the business.

CHAPTER 1: INTRODUCTION

I. About the research problem

The dataset was created by a UK-based and registered non-store online business. The company mainly sells unique all-occasion gifts, and many of its customers are wholesalers. This enterprise has had purchases made by 4000 customers over a period of one year, in 37 countries around the world. With such a large scale, the needs of improving marketing quality, increasing sales, and expanding business are not easy problems to solve. After this one year, the organization realized that its marketing campaign was not effective and that its investments and strategies to boost sales had not been met with adequate returns. Consequently, its revenues did not experience any upward trend in the period.

Our research was made with the mission to solve this problem for the company. With applications of descriptive and predictive data analytics, our team aims to analyze the sales operation of the company and build a model that can categorize individuals into certain customer segments.

II. About the dataset

The “E-commerce Data” dataset that we analyze is taken from the Kaggle dataset repository. This is a transnational data set which contains all the transactions occurring between 01/12/2010 and 09/12/2011. Each entry in the dataset describes the purchase of a product by a particular customer on a given date.

Data Set Characteristics:	Multivariate, Sequential, Time-Series	**Number of Instances:**	541909
Attribute Characteristics:	Integer, Real	**Number of Attributes:**	8
Associated Tasks:	Classification, Clustering	**Missing Values?**	N/A

- InvoiceNo: Invoice number. Nominal, a 6-digit integral number uniquely assigned to each transaction. If this code starts with the letter "c”, it indicates a cancellation.

- StockCode: Product (item) code. Nominal, a 5-digit integral number uniquely assigned to each distinct product.

- Description: Product (item) name. Nominal.

- Quantity: The quantities of each product (item) per transaction. Numeric.

- InvoiceDate: Invoice Date and time. Numeric, the day and time when each transaction was generated.

- UnitPrice: Unit price. Numeric, Product price per unit in sterling.

- CustomerID: Customer number. Nominal, a 5-digit integral number uniquely assigned to each customer.

- Country: Country name. Nominal, the name of the country where each customer resides.

III. Motivation and purpose of the research

For the two topics of research above, the purpose of this study is to help this business draw insights, thereby radically improving the shortcomings in its operations, especially in customer service and marketing.

Several benefits and improvements that the firm can create from the results of this research include:

- Understanding customers' behaviors.

- Developing different customer types and personas.

- Promoting targeted, effective marketing campaigns.

- Developing more relevant policies for sales and pricing.

- Improving customer service.

IV. Main questions

- Sales Analysis:

- Which country consumed goods the highest and the lowest?
- What are the most consumed products of the company?
- At what time of a day did the company experience the highest/lowest sales?
- What is the sales situation of each month of the year?
- What are the different segments of products at the company?

- Customer Segmentation:

- What are the segments of customers that the company has?
- What are the characteristics of each type?
- What policies should the company gives to each customer segment?

CHAPTER 2: ANALYSIS METHODOLOGY

For this report, our team will use the Python programming language to perform different descriptive analytics, build 1 model for product segmentation and 2 models for customer segmentation: the RFM model and the K-means clustering model.

CHAPTER 3: PROCESSING STEPS AND RESULTS

I. Sales Analysis

1. Data Preparation

As the dataset is fairly clean, int this part the data preparation process only includes removing missing, duplicate values and negative quantity values, as well as changing the datatype of the "InvoiceDate" column to datetime.

```
df_initial.dropna(axis = 0, subset = ['CustomerID'], inplace = True)
print('Dataframe dimensions:', df_initial.shape)
#-----------------------------------------------------------
# gives some infos on columns types and numer of null values
tab_info=pd.DataFrame(df_initial.dtypes).T.rename(index={0:'column type'})
tab_info=tab_info.append(pd.DataFrame(df_initial.isnull().sum()).T.rename(index={0:'n
ull values (nb)'}))
tab_info=tab_info.append(pd.DataFrame(df_initial.isnull().sum()/df_initial.shape[0]*1
00).T.
                         rename(index={0:'null values (%)'}))
display(tab_info)
```

2. Descriptive Analytics

a. Which country consumed goods the highest, lowest?

In the dataset, there are 37 countries that show sales record. However, the distribution of orders from countries is markedly different. Through a simple command, we can see which country the dominance is coming from.

```
temp = df_initial[['CustomerID', 'InvoiceNo', 'Country']].groupby(['CustomerID', 'Inv
oiceNo', 'Country']).count()
temp = temp.reset_index(drop = False)
countries = temp['Country'].value_counts()
print('Nb. de pays dans le dataframe: {}'.format(len(countries)))
```

```
Nb. de pays dans le dataframe: 37
```

```
countries = df['Country'].value_counts(normalize=True)
countries[:5]
```

```
United Kingdom    0.890513
Germany           0.022720
France            0.020963
EIRE              0.018186
Spain             0.006243
Name: Country, dtype: float64
```

It is easy to realize that the UK is accounting for 89% of the orders, showing the superiority of this country over other countries in this dataset. This is understandable as the business is headquartered in the UK.

b. What are the most consumed products of the company?

Now, we will visualize the sales of each product and the top popular products the company sold:

```
most_sold = df.groupby(['StockCode'])['Quantity'].sum().reset_index()
most_sold = most_sold.sort_values(by=["Quantity"], ascending=False)
```

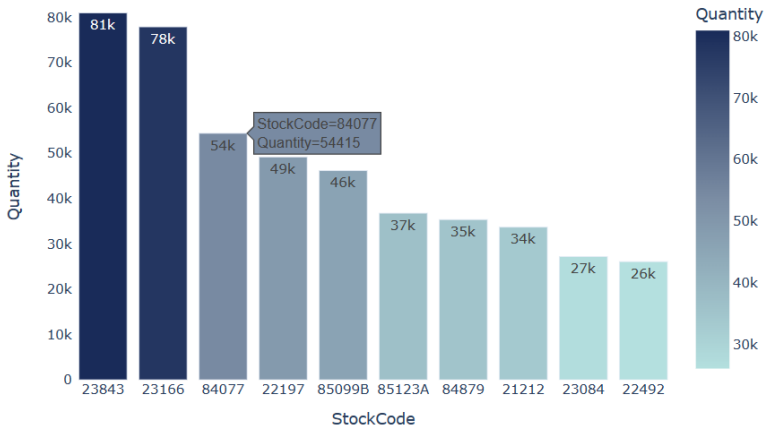


Figure 1 - Aggregate sales of individual products

From the figure, it can be concluded that the top 3 items with the highest sales are: “PAPER CRAFT, LITTLE BIRDIE” (23843), “MEDIUM CERAMIC TOP STORAGE JAR” (23166) and “WORLD WAR 2 GLIDERS ASSTD DESIGNS” (84077).

c. What is the daily sales trend for the company?

Here, we will try to see if customers tend to purchase items at a more particular time of the day.

- Set up a new dataframe that include the month and hour information in transactions and count the number of occurrences of each hour to compare and find the most popular buying hour of customer:

```
df_temp = df.groupby([df.CustomerID,df.InvoiceDate]).Quantit
y.sum()
df_temp = pd.DataFrame(df_temp).reset_index()
```

```
df_temp["Hour"]= df_temp["InvoiceDate"].dt.hour
df_temp["Month"]= df_temp["InvoiceDate"].dt.to_period('M')
```

```
count_hours = pd.DataFrame(columns=range(1,25))
count_hours["Month"] = ""

for month in sorted(df_temp["Month"].unique()):
    row = []
    for hour in range(1,25):
        freq = len(df_temp[(df_temp["Month"]==month)&(df_tem
p["Hour"]==hour)])
        row.append(float(freq))
    row.append(month)
    count_hours.loc[len(count_hours)] = row

count_hours = count_hours.set_index("Month")
```

- Then we plot the results and find that the majority of transactions appear to be completed between 11am and 12 am. There are hardly many transactions conducted in the early morning between 5 and 8, however. Additionally, hardly any transactions take place at night.

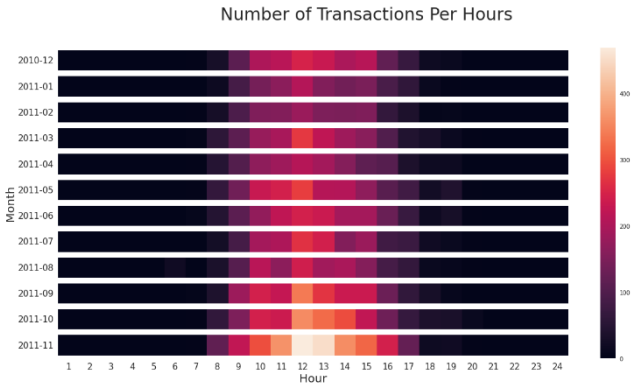


Figure 2 – Number of transactions in each hour of a day (from 12/2010 to 11/2011)

d. What is the daily sales trend for the company?

To have a clear view about the sales department of this organization, we consider the revenue of each month.

- First of all, because in the dataset, there is no revenue column, we have to build our revenue by multiplying the “Unit price” and the “Quantity”:

```
# Create revenue column
df["Revenue"] = df["UnitPrice"] * df["Quantity"]
```

- To make the information easier to grasp, we represent it with the following diagram:

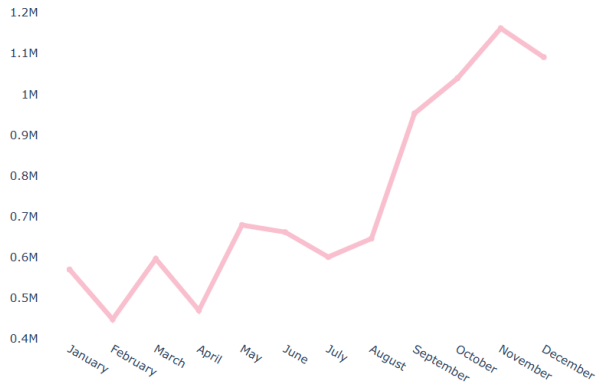


Figure 3 - Aggregated sales by months

- As we can see, the best-selling months are October, November, and December.

- The graphic indicates that the revenue is completely unstable throughout the year, with a differential of more than 3 times between months with lowest and highest revenue.
- Since most individuals in the UK consider themselves to be "Christian," this increase in sales during the end of the year might be connected to Christmas, when “holiday season” come and goods are purchased more.

3. Predictive Analytics

Question: What are the different segments of products at the company?

For an e-commerce company that sells unique all-occasion gifts, it is a fact that several products might fall in the same segment and be purchased together at a certain period of a year. Thus, we are motivated to categorize the products into different segments in this part, using the K-means clustering approach.

- Prior to building the model, we use the silhouette score to define the number of clusters that best represent the data:

```
matrix = X.as_matrix()
for n_clusters in range(3,10):
    kmeans = KMeans(init='k-means++', n_clusters = n_clusters, n_init=30)
    kmeans.fit(matrix)
    clusters = kmeans.predict(matrix)
    silhouette_avg = silhouette_score(matrix, clusters)
    print("For n_clusters =", n_clusters, "The average silhouette_score is :", silhou
ette_avg)
```

```
For n_clusters = 3 The average silhouette_score is : 0.10071681758064248
For n_clusters = 4 The average silhouette_score is : 0.12208239761153944
For n_clusters = 5 The average silhouette_score is : 0.1470081849157512
For n_clusters = 6 The average silhouette_score is : 0.14389841472426354
For n_clusters = 7 The average silhouette_score is : 0.15212220110144017
For n_clusters = 8 The average silhouette_score is : 0.1558201267218184
For n_clusters = 9 The average silhouette_score is : 0.11656173409117862
```

In practice, the scores obtained above can be considered equivalent since, depending on the run, scores of 0.1±0.05 will be obtained for all clusters with n_clusters > 3 (we obtain slightly lower scores for the first cluster). On the other hand, we found that beyond 5 clusters, some clusters contained very few elements. Our team therefore choose to separate the dataset into 5 clusters. In order to ensure a good classification at every run of the notebook, we iterate until we obtain the best possible silhouette score, which is, in the present case, around 0.15.

- After running the model with 5 clusters, we now can have a look at the types of objects that each cluster represents and output the result as word clouds:

```
def random_color_func(word=None, font_size=None, position=None,
                      orientation=None, font_path=None, random_state=None):
    h = int(360.0 * tone / 255.0)
    s = int(100.0 * 255.0 / 255.0)
    l = int(100.0 * float(random_state.randint(70, 120)) / 255.0)
    return "hsl({}, {}%, {}%)".format(h, s, l)
#_____________________________________________________________________________
def make_wordcloud(list, increment):
    ax1 = fig.add_subplot(4,2,increment)
    words = dict()
    trunc_occurences = list[0:150]
    for s in trunc_occurences:
        words[s[0]] = s[1]
    #_____________________________________________________
    wordcloud = WordCloud(width=1000,height=400, background_color='lightgrey',
                          max_words=1628,relative_scaling=1,
                          color_func = random_color_func,
                          normalize_plurals=False)
    wordcloud.generate_from_frequencies(words)
    ax1.imshow(wordcloud, interpolation="bilinear")
    ax1.axis('off')
    plt.title('cluster nº{}'.format(increment-1))
#_____________________________________________________________________________
fig = plt.figure(1, figsize=(14,14))
color = [0, 160, 130, 95, 280, 40, 330, 110, 25]
for i in range(n_clusters):
    list_cluster_occurences = occurence[i]

    tone = color[i] # define the color of the words
    list = []
    for key, value in list_cluster_occurences.items():
        list.append([key, value])
    list.sort(key = lambda x:x[1], reverse = True)
    make_wordcloud(list, i+1)
```

Figure 4 – Word cloud of different product segments

From this illustration, we can see, for instance, that one of the clusters contains items that may be related to seasonal gifting (keywords: Christmas, packaging, card, etc.). A different cluster would choose to have expensive goods and jewelry (keywords: necklace, bracelet, lace, silver, etc.).

4. Findings and suggestions

Generally, with the result coming from this analysis, the firm can be more informed when conducting capacity, material, purchasing, inventory, and production planning:

- Based on the time series record of the above online store, we observed a dramatic increase in shopping volume in the final months of the year, as well as a regular pattern of transactions between 11 and 12AM. This demonstrates that in order to give customers the ideal purchasing experience, the store needs to arrange online employee to quickly inform consumers at peak hours and set up a stable shopping platform.

- The rise in sales during the final months of the year shows that the firm should constantly update its selection of Christmas, Thanksgiving, New Year's, etc.- related merchandise to satisfy client demand.

- Concentrate marketing efforts on the UK market. More precisely, companies can air advertisements between the hours of 11 a.m. and 12 a.m., create advertising programs, or offer specials to encourage holiday buying.

- The model in this analysis can help company predict the categories of goods that are often bought together to increase the average order value (AOV) and manage inventory.

II. Customer Segmentation

1. RFM Segmentation

a. Introduction

To capture and provide better customer service for different customer types, the RFM model could be applied. The fundamental of the RFM segmentation technique is that by examining three quantitative characteristics, marketers may acquire a thorough picture of their clients. These characteristics are:

- **Recency:** Number of days since the last purchase. In most cases, the more recently a customer has interacted or transacted with a brand, the more likely that customer will be responsive to communications from the brand.

- **Frequency:** How frequently did a client interact with the brand over a specific time period? Customers who participate in activities frequently are undoubtedly more interested and devoted than those who do so infrequently. In this case, frequency is the number of transactions made over a given period

- **Monetary:** This variable shows the total amount of money a consumer has spent with the brand over a specific time frame. Customers who spend a lot deserve different treatment than those who spend little.

b. Data Preparation

- In the Sales analysis part, we already know that more than 90% of the customers in the data are from the United Kingdom. There's some research indicating that customer clusters vary by geography, so here we will restrict the data to the United Kingdom only.

```
#Keep only United Kingdom data
Rtl_data = Rtl_data.query("Country=='United Kingdom'").reset_index(drop=True)
```

- Next, we will clean our data like in the previous section and also create the Total Amount field, which is the result of Quantity multiple to Unit Price.

```
#Remove missing values from CustomerID column, can ignore missing values in description column
Rtl_data = Rtl_data[pd.notnull(Rtl_data['CustomerID'])]
```

- Additionally, our team will validate if there are any negative values in the Quantity and the Unit Price column (these 2 attributes should not be negative). If there are any, our group will filter them out.

```
#Validate if there are any negative values in Quantity column
Rtl_data.Quantity.min()
```

```
-80995
```

```
#Validate if there are any negative values in UnitPrice column
Rtl_data.UnitPrice.min()
```

```
0.0
```

```
#Filter out records with negative values
Rtl_data = Rtl_data[(Rtl_data['Quantity']>0)]
```

After the cleaning process, the dataset has only 354345 rows and 9 columns.

```
Rtl_data.shape
```

```
(354345, 9)
```

```
Rtl_data.head()
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	TotalAmount
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom	15.30
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	20.34
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom	22.00
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	20.34
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	20.34

- Because the last invoice date is 9th December 2011, we set the latest date to 10th December 2011. By doing this, we could calculate the number of days from recent purchases for the Recency Score.

```
#Set Latest date 2011-12-10 as last invoice date was 2011-12-09.
#This is to calculate the number of days from recent purchase
Latest_Date = dt.datetime(2011,12,10)
```

c. Building the model

- Our group will sort the customers into quartiles based on these three metrics, and calculating their RFM score. To do this, we use the aggregate function. Based on the definition of RFM:

- Recency = Latest Date - Last Invoice Data
- Frequency = Count of Invoice No. of transaction(s)
- Monetary = Sum of Total Amount. These three variables will be grouped by CustomerID.

```
#Create RFM Modelling scores for each customer
RFMScores = Rtl_data.groupby('CustomerID').agg({'InvoiceDate': lambda x: (Latest_Date - x.max()).days,
                                                'InvoiceNo': lambda x: len(x),
                                                'TotalAmount': lambda x: x.sum()})

#Convert Invoice Date into type int
RFMScores['InvoiceDate'] = RFMScores['InvoiceDate'].astype(int)

#Rename column names to Recency, Frequency and Monetary
RFMScores.rename(columns={'InvoiceDate': 'Recency',
                          'InvoiceNo': 'Frequency',
                          'TotalAmount': 'Monetary'}, inplace=True)

RFMScores.reset_index().head()
```

	CustomerID	Recency	Frequency	Monetary
0	12346.0	325	1	77183.60
1	12747.0	2	103	4196.01
2	12748.0	0	4596	33719.73
3	12749.0	3	199	4090.88
4	12820.0	3	59	942.34

- In the next step, we will subdivide our entire data set into four groups based on the Recency, Frequency and Monetary that we calculated earlier using quantiles. In our case, the quantile values are 0.25, 0.5, and 0.75.

```
#Split into four segments using quantiles
quantiles = RFMScores.quantile(q=[0.25,0.5,0.75])
quantiles = quantiles.to_dict()
```

```
quantiles
```

```
{'Recency': {0.25: 17.0, 0.5: 50.0, 0.75: 142.0},
 'Frequency': {0.25: 17.0, 0.5: 41.0, 0.75: 99.0},
 'Monetary': {0.25: 300.03999999999996, 0.5: 651.8199999999999, 0.75: 1575.89}}
```

```
#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
```

- Our team divides the customer list into tiered groups for each of the three dimensions (R, F and M), in which a score of 1 is the best and a score of 4 is the worst.

Table 1 – Table of RFM scores

Recency	**Frequency**	**Monetary**
R-Tier-1 (Most recent)	F-Tier-1 (Most Frequent)	M-Tier-1 (Highest Spend)
R-Tier-2	F-Tier-2	M-Tier-2
R-Tier-3	F-Tier-3	M-Tier-3
R-Tier-4 (Least recent)	F-Tier-4 (Small number of transactions)	M-Tier-4 (Lowest Spend)

- In the last step, we will assign RFM Score for each Customer and create a RFMGroup column:

```
#Calculate Add R, F and M segment value columns in the existing dataset to show R, F and M segment values
RFMScores['R'] = RFMScores['Recency'].apply(RScoring, args=('Recency',quantiles,))
RFMScores['F'] = RFMScores['Frequency'].apply(FnMScoring, args=('Frequency',quantiles,))
RFMScores['M'] = RFMScores['Monetary'].apply(FnMScoring, args=('Monetary',quantiles,))
```

```
#Calculate and Add RFMGroup value column showing combined concatenated score of RFM
RFMScores['RFMGroup'] = RFMScores.R.map(str) + RFMScores.F.map(str) + RFMScores.M.map(str)
```

CustomerID	Recency	Frequency	Monetary	R	F	M	RFMGroup
12346.0	325	1	77183.60	4	4	1	441
12747.0	2	103	4196.01	1	1	1	111
12748.0	0	4596	33719.73	1	1	1	111
12749.0	3	199	4090.88	1	1	1	111
12820.0	3	59	942.34	1	2	2	122

d. Results Communication and Recommendation

Here, we will manually define different types of customers based on the RFM score. With this, businesses (more specifically the customer service department) can have better strategies and provide suitable customer service for each segment:

Table 22 – Customer segments and corresponding strategies

Customer Type	RFM Score	Customer Characteristic	Business Strategy
Best Customer - Cannot lose them	111	Highly engaged customers who have bought the most recent, the most often, and generated the most revenue.	Focus on loyalty programs and new product introductions. These customers have proven to have a higher willingness to pay, so don't use discount pricing to generate incremental sales. Instead, focus on value-added offers through product recommendations based on previous purchases.
Loyal - Most loyal customer	X1X	Customer who buys the most from the store	Loyalty programs are effective for these repeat visitors. Advocacy programs and reviews are also common X1X strategies. Lastly, consider rewarding these customers with Free Shipping or other benefits.
Whales - Highest Paying Customers	XX1	Customers who have generated the most revenue for the business	These customers have demonstrated a high willingness to pay. Consider premium offers, subscription tiers, luxury products, or value add cross/up-sells to increase AOV. Business should not waste margin on discounts.

Promising - Faithful customers	X13, X14	Customers who return often, but do not spend a lot	In this case, business already succeeded in creating loyalty. Focus on increasing monetization through product recommendations based on past purchases and incentives tied to spending thresholds (pegged to the store's AOV).
Rookies - Newest Customers	14X	First time buyer from the store	Most customers never graduate to be loyal. Having clear strategies in place for first time buyers such as triggered welcome emails will give free gift or voucher.
Slipping - Once Loyal, Now Gone	44X	Great past customers who have not bought in awhile	Analyzing the customers' journey. to have a clear picture of all customer interactions across the website and touch points throughout every stage of the customer lifecycle, and be present with the right strategy at the right place and time.

- From the graph, we can clearly see that:

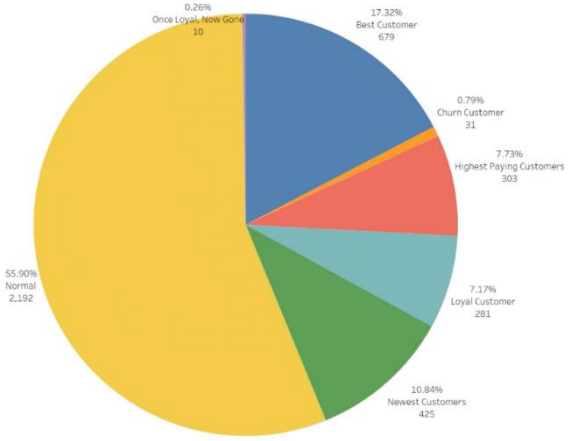


Figure 5 – Distribution of customer segments with RFM

- **Normal Customers** make up most of the store's customers. Meanwhile, **Best customer** accounts for about 17%. **Churn customer** and **One Loyal, Now gone** make up a small part.
- For C**hurn Customers**, they bought a long time ago, they also buy few and spend little. For this type of customer, the store should not spend too much to re-acquire.
- In addition, it should be noted that this business also has a surprisingly good customer base: **Loyal Customer, Best Customer, High spending Customer**. Businesses should have strategies like the one in the table above to continue to maintain their loyalty.
- For **Normal Customer**, which accounts for more than half of the customer type, the store needs to have customized strategies such as: Provide better customer service to

increase Customer Satisfaction, diversify products and increase product quality, as well as having strategies to stimulate shopping demand to gradually turn normal customers into **Loyal customers, Best customers.**

2. K-Means Clustering

a. Introduction

Now, we will be approaching the segmentation using K-Means Clustering, a popular unsupervised learning algorithm. With this method we can separate customers into k-number of groups that have similarities based on their recency, frequency and monetary. This time, the categories are more flexible and not based on a grade like when using the RFM model.

b. Data Preparation

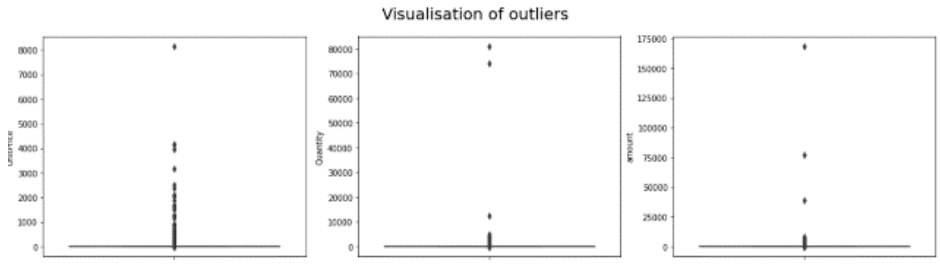
- Just like in the previous session, we will create a new dataframe that includes the recency, frequency and monetary values for each customer and drop the columns containing null and negative values.

```
kmeans = data_all.groupby(['CustomerID'])\
                 .agg({'InvoiceDate': lambda x: (analysis_date - x.max()).days,
                       'InvoiceNo': 'count',
                       'amount': 'sum'})\

#rename each column to denote the R,F,M Values
kmeans.rename(columns = {'InvoiceDate': 'Recency',
                                    'InvoiceNo': 'Frequency',
                                    'amount': 'Monetary'}, inplace=True)
```

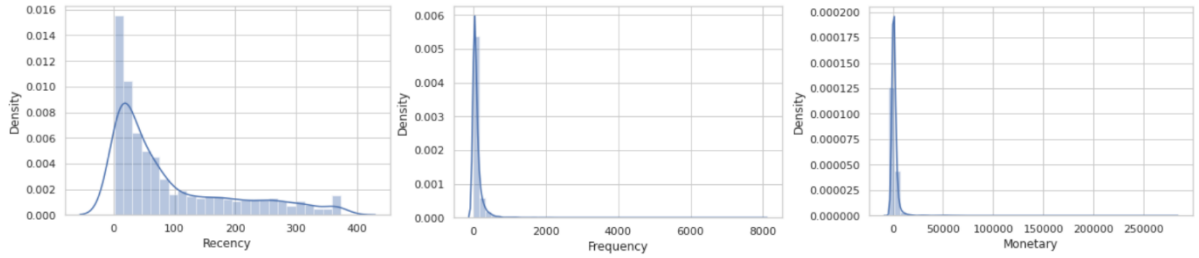
- Prior to building the model, the data must be processed using some techniques to make it more suitable for the K-Means model.

- K-Means is sensitive toward outliers so we should check and delete these outliers.




```
# remove values that are out of the 99,7% range
data = data[(np.abs(stats.zscore(data[["amount"]])) < 3).all(axis=1)]
```

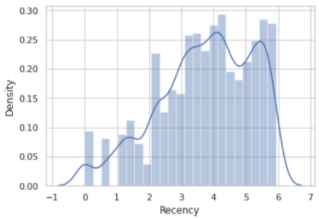
- K-Means assumes that the variables are not skewed. Thus, we will test our R, F, M values. To do this, we will plot the distribution of these three attributes.



Since they are skewed, we will use logarithmic transformation to eliminate the skewness.

```
#log transformation of recency metric
recency_log= np.log(df_kmeans['Recency'])

#plot the transformed variable
sns.distplot(recency_log)
plt.show()
```



- K-Means assumes that all the variables have a similar mean and variance. Therefore, we will check the range and mean of each of the variables.

	Recency	Frequency	Monetary
count	4372.000000	4372.000000	4372.000000
mean	92.047118	93.053294	1898.459701
std	100.765435	232.471608	8219.345141
min	1.000000	1.000000	-4287.630000
25%	17.000000	17.000000	293.362500
50%	50.000000	42.000000	648.075000
75%	143.000000	102.000000	1611.725000
max	374.000000	7983.000000	279489.020000

Because they are dissimilar, we will be using the Standard Scalar to normalize them.

```
rfm = grouped[['Recency','Frequency','Monetary']]

#making all values in Monetary positive
rfm['Monetary']=rfm['Monetary']+1

#applying logarithmic transformation
for c in ['Recency', 'Frequency']:
    rfm[c]= np.log(rfm[c])


#Normalization of variables
from sklearn.preprocessing import StandardScaler

ecomm_standardized= StandardScaler().fit_transform(rfm)
rfm[['Recency','Frequency','Monetary']]=ecomm_standardized.round(2)
```

CustomerID	Recency	Frequency	Monetary
12346.0	1.40	-2.23	-0.23
12347.0	-2.08	1.13	0.29
12348.0	0.40	-0.19	-0.01
12349.0	-0.54	0.45	-0.02
12350.0	1.37	-0.63	-0.19

c. Building the Model

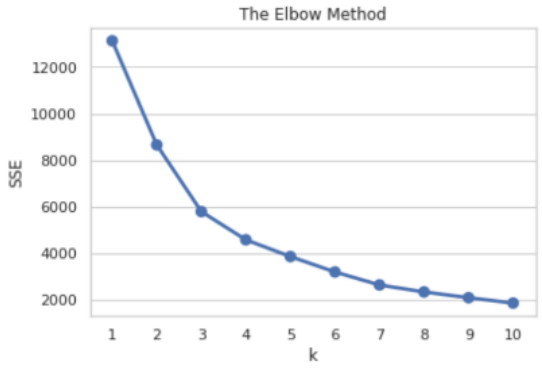
- First, we will use the elbow criterion method to find the optimum number of clusters. Let's set a limit for the number of clusters at 10 and plot the SSE value of each number.

```
#start k-means clustering
from sklearn.cluster import KMeans

sse = {}

#find the optimum number of clusters from 1 to 10
for k in range(1, 11):
    model = KMeans(n_clusters=k, random_state=1)
    model.fit(rfm)
    sse[k] = model.inertia_

# Plot SSE for each value of k
plt.title('The Elbow Method')
plt.xlabel('k');
plt.ylabel('SSE')
sns.pointplot(x=list(sse.keys()), y=list(sse.values()))
plt.show()
```



- Based on the SSE result on the plot, we will fit k-means with 4 clusters. Next, a table is created to compare these clusters based on recency, frequency and monetary (using mean value).

```
#fit k-means with 4 clusters
kmeans = KMeans(n_clusters=4, random_state=1)
kmeans.fit(rfm)
```

Cluster	Recency	Frequency	Monetary
0	179.32	13.60	368.33
1	70.96	73.91	1211.68
2	7.88	1527.19	88707.14
3	9.10	215.13	3786.62

- In the last step, we will assign each customer to these clusters with the Group column and see the distribution of each cluster with an illustration:

```
# Classify the customer based on the clusters
df_kmeans['Group']= model.labels_
df_kmeans.head()
```

CustomerID	Recency	Frequency	Monetary	Group
12347.0	2	182	4310.00	3
12348.0	75	31	1797.24	1
12349.0	19	73	1757.55	1
12350.0	310	17	334.40	0
12352.0	36	85	2506.04	1

d. Results Communication and Recommendation

- From the K-means clustering, we can sort every customer into 4 different clusters that seem to have different behaviors:

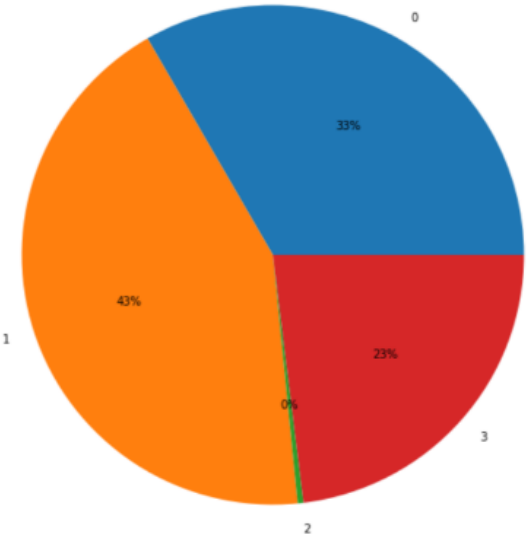


Figure 6 - Distribution of customer segments with RFM

- Cluster 0: "Hibernating customers" - Those are the customers that buy at the lowest frequency, the lowest recently, and that spend the least money.
- Cluster 1: "Normal customers" - Those are the customers that occasionally buy and spend a moderate amount on the website.
- Cluster 2: "Exceptional customers" - Those are the customers that we want to keep, that buy at the highest frequency, the most recent, and that spend the most money.
- Cluster 3: " Recent customers" - Those are customers that have been active quite recently that might be interesting to keep stimulated.

- According to the segmentation:

- There is no discernible sign of exceptional customers.
- There are almost 50% of normal customers for whom we would like to continue providing incentives to encourage their spending.
- There are also 33% of the customers that are not really active and don't spend much.
- New buyers account for a quarter of the total customers. Policies should be made to turn these new buyers into normal customers of the business.