# Customer Segmentation

Bhavesh Khanchandani

# Problem Statement

- A company that sells some of the product, and you want to know how well the selling performance of the product. You have the data that we can analyze, but what kind of analysis can we do? Well, we can segment customers based on their buying behavior on the market.
- Keep in mind that the data is really huge, and we can not analyze it using our bare eyes. We will use machine learning algorithms and the power of computing for it.
- This project will show you how to cluster customers on segments based on their behavior using the clustering algorithm in Python.

# Proposed Solution

Data Preprocessing : This step performs all pre-processing steps such as data manipulation, data filling, converting categorical into numeric, and all processes.

The EDA process involves performing

1. Univariate Analysis
2. Bivariate analysis
3. Removing Missing values if any / Outlier treatment
4. Machine Learning : Build a clustering model to segment the customer-based similarity.Also fine-tune the hyper parameters & compare the evaluation metrics of various classification algorithms

# Descriptive Analysis

| | 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 |

```
df.shape
```

```
(541909, 8)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541909 entries, 0 to 541908
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    541909 non-null  object
 1   StockCode    541909 non-null  object
 2   Description  540455 non-null  object
 3   Quantity     541909 non-null  int64
 4   InvoiceDate  541909 non-null  datetime64[ns]
 5   UnitPrice    541909 non-null  float64
 6   CustomerID   406829 non-null  float64
 7   Country      541909 non-null  object
dtypes: datetime64[ns](1), float64(2), int64(1), object(4)
memory usage: 33.1+ MB
```

```
df.isnull().sum().sort_values(ascending=False)
```

```
CustomerID     135080
Description      1454
InvoiceNo           0
StockCode           0
Quantity            0
InvoiceDate         0
UnitPrice           0
Country             0
dtype: int64
```

```
df.describe()

# we can see that quantuity has negative values
```

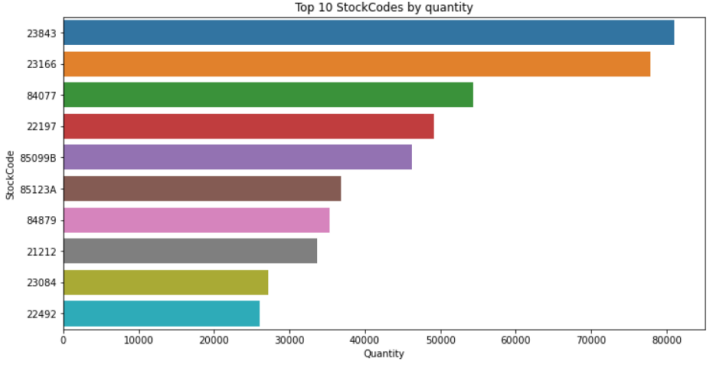
| | Quantity | UnitPrice | CustomerID |
|---|---|---|---|
| count | 406829.000000 | 406829.000000 | 406829.000000 |
| mean | 12.061303 | 3.460471 | 15287.690570 |
| std | 248.693370 | 69.315162 | 1713.600303 |
| min | -80995.000000 | 0.000000 | 12346.000000 |
| 25% | 2.000000 | 1.250000 | 13953.000000 |
| 50% | 5.000000 | 1.950000 | 15152.000000 |
| 75% | 12.000000 | 3.750000 | 16791.000000 |
| max | 80995.000000 | 38970.000000 | 18287.000000 |

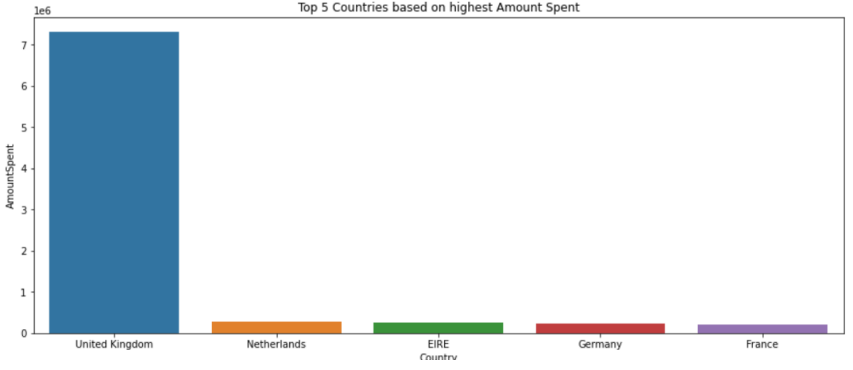
# Exploratory Data Analysis

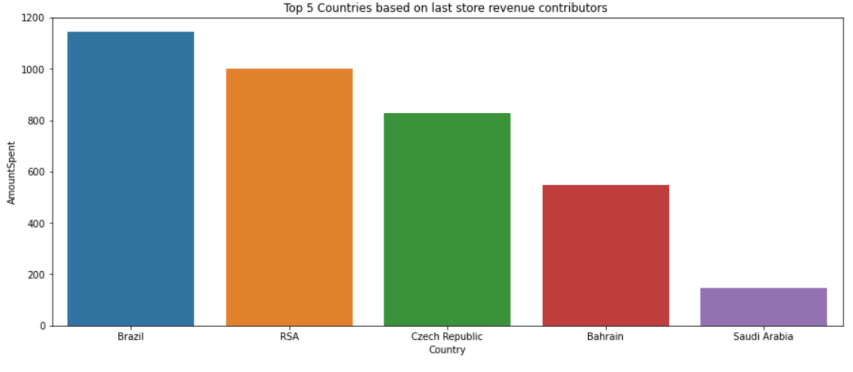
```
# Top 5 Customers with the highest no of orders

tf=df.groupby(by=['CustomerID','Country'], as_index=False)['InvoiceNo'].count()
tf.sort_values(by='InvoiceNo', ascending=False).iloc[:5]
```

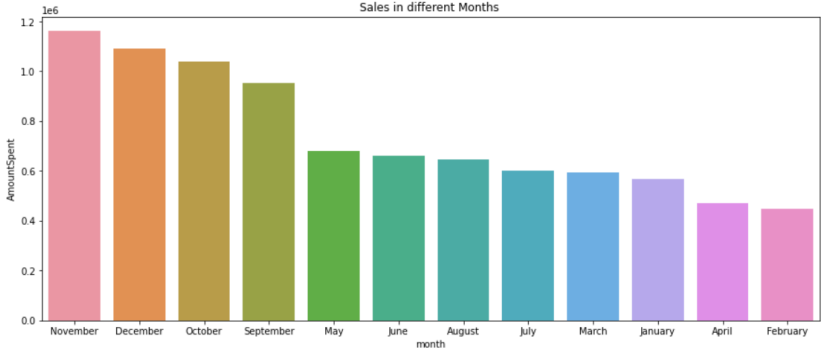
| | CustomerID | Country | InvoiceNo |
|---|---|---|---|
| **4019** | 17841.0 | United Kingdom | 7847 |
| **1888** | 14911.0 | EIRE | 5677 |
| **1298** | 14096.0 | United Kingdom | 5111 |
| **334** | 12748.0 | United Kingdom | 4596 |
| **1670** | 14606.0 | United Kingdom | 2700 |

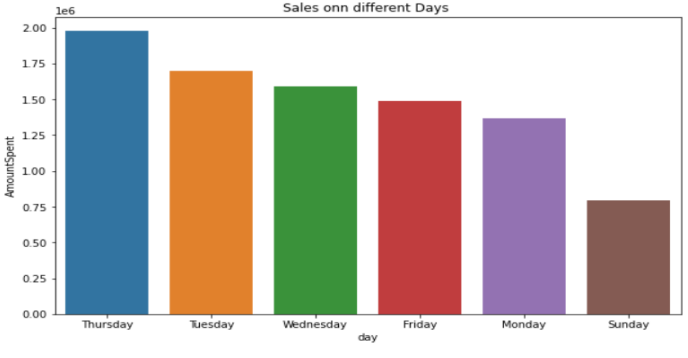


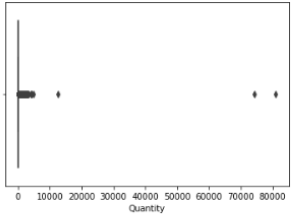



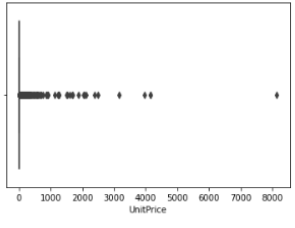
# Exploratory Data Analysis

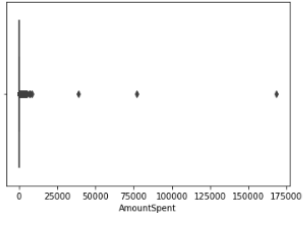



```
# outliers
for i in df[['Quantity','UnitPrice','AmountSpent']] :
    sns.boxplot(df[i])
    plt.show()
```



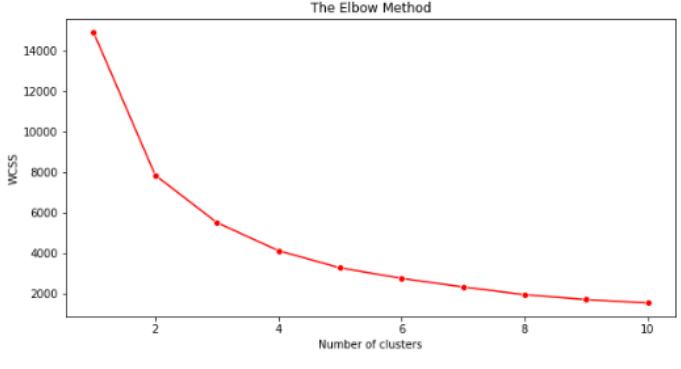



# Machine Learning Modelling

- Feature selection for algorithm

X = df[['AmountSpent','CustomerID']].values

- Normalize Data using MimMaxScaler
- Find the best value for K using Elbow Method
- Number of cluster = 2



# Optimisation

# Conclusion

# Future Scope