# High Value Customer Identification

Venkata Gangadhar Naveen Palaka

George Washington University

Data Mining

Department of Data Science

## Introduction

A UK-based online retail store has captured the sales data for different products for the period of one year (Nov 2016 to Dec 2017) and updated the dataset (in 2020). The organization sells gifts primarily on the online platform. The customers who make a purchase, consume directly for themselves. There are small businesses that buy in bulk and sell to other customers through the retail outlet channel. The project objective is to find significant customers for the business who make high purchases of their favorite products. The organization wants to roll out a loyalty program to high-value customers after the identification of segments. The Data mining algorithm used is K-means clustering and it is in standard form (see Appendix A for in depth knowledge). The packages used to implement the network are NumPy, Pandas, Seaborn, Matplotlib, Sklearn and Scipy. For the K-means clustering method, the most common approach for metric evaluation is the so-called elbow method. It involves running the algorithm multiple times over a loop, with an increasing number of cluster choices, and then plotting a clustering score as a function of the number of clusters. I have worked on Exploratory Data Analysis part in this project.

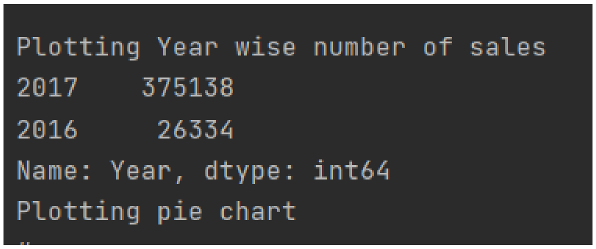
## Description of Work

I have done Exploratory data analysis in this project, I have used various resources to understand how to explore the data. In  this project I have used pandas and matplotlib for EDA.

```
#%%
print("EXPLORATORY DATA ANALYSIS: \n")
```

I have taken value_counts() function get number of sales year wise,

```
#%%
print("Plotting Year wise number of sales")
years = df['Year'].value_counts()
print(years)
```



```
Plotting Year wise number of sales
2017    375138
2016     26334
Name: Year, dtype: int64
Plotting pie chart
```

Then plotted a pie chart with figure size(7,5), labels as year wise, with two different colours to show variation, used explode for better chart, titled with Sales percentage by year.

```
pie, ax = plt.subplots(figsize=[7,5])
labels = ['2017', '2016']
colors = ['#ff9999', '#ffcc99']
plt.pie(x = years, autopct='%.1f%%', explode=[0.05]*2, labels=labels, pctdistance=0.5, colors = colors)
plt.title('Sales percentage by year')
print("Plotting pie chart")
```

```
plt.show()
print('#',50*"-")
```

Now plotting monthly sales by year, I have taken two different variables sales_16 and sales_17 and selected the data accordingly using  conditional operator from the dataframe. Then I have taken value-counts for two separate years to get monthly sales

```
#%%
print("Plotting monthly sales by year")
sales_16 = df[df['Year'] == 2016]
sales_17 = df[df['Year'] == 2017]
monthly_16 = sales_16['Month'].value_counts()
monthly_17 = sales_17['Month'].value_counts()
```

Now I have plotted the bar graph, with orange color with figsize(8,5), titled ‘Monthly number of sales for 2016’ , with xlabel as ‘Month’ and ylabel as’ number of sales’, also used grid function.

```
plt.figure(figsize=(8,5))
monthly_16.sort_index().plot(kind='bar', color='orange')
plt.title('Monthly number of sales for 2016')
plt.xlabel('Month')
plt.ylabel('number of sales')
plt.grid()
plt.show()
```

Now I have plotted the bar graph, with blue color with figsize(8,5), titled ‘Monthly number of sales for 2017’ , with xlabel as ‘Month’ and ylabel as’ number of sales’ , also used grid function.

```
plt.figure(figsize=(8,5))
monthly_17.sort_index().plot(kind='bar', color='blue')
plt.title('Monthly number of sales for 2017')
plt.xlabel('Month')
plt.ylabel('number of sales')
plt.grid()
plt.show()
print('#',50*"-")
```

I have taken aggregate groupby function to get customerID with quantity and sorted values in descending order to get top 10 customers with highest number of quantities bought.

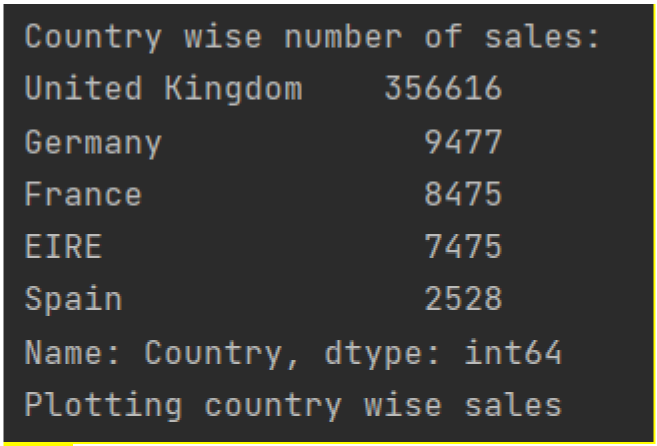
```
#%%
customers = df.groupby('CustomerID')['Quantity'].sum()
print("Plotting top 10 customers after summing their purchase quantities")
top_customers = customers.sort_values(ascending=False).head(10)
```

I plotted bar graph with fig size(8,8), color indigo, titled as 'Top 10 customers by quantity bought', xlabel as ‘CustomerID’ and ylabel as ‘Quantity’

```
plt.figure(figsize=(8,8))
top_customers.plot(kind='bar', color='indigo')
plt.title('Top 10 customers by quantity bought')
plt.xlabel('Customer ID')
plt.ylabel('Quantity')
plt.show()
print('#',50*"-")
```

I have taken country wise number of sales, by using value_counts()

```
#%%
print("Country wise number of sales:")
print(df['Country'].value_counts().head())
```



```
Country wise number of sales:
United Kingdom    356616
Germany             9477
France              8475
EIRE                7475
Spain               2528
Name: Country, dtype: int64
Plotting country wise sales
```

```
#%%
countries = df['Country'].value_counts()[1:]
```

Again I used here bar graph, subplots with figsize(12,10), color as purple, set the x-axis xticklabels with 90 degree rotation for better visualization and titled as 'Number of customers by country (excluding UK)'. We already know UK has the highest with major difference so we are not taking it into the graph.

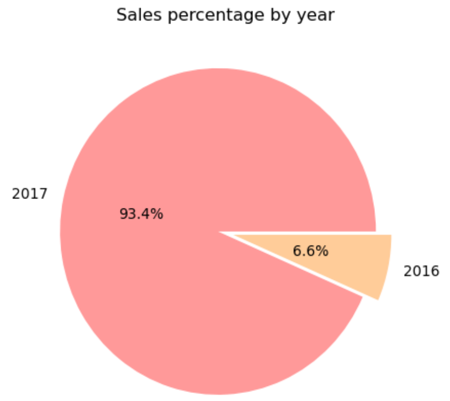
```
fig, ax = plt.subplots(figsize = (12,10))
ax.bar(countries.index, countries, color= 'purple')
ax.set_xticklabels(countries.index, rotation = 90)
ax.set_title('Number of customers by country (excluding UK)')
print("Plotting country wise sales")
plt.show()
print('#',50*"-")
```

This is a sample pyqt5, for GUI development I made a try for simple plot.

```
import sys
import matplotlib
matplotlib.use('Qt5Agg')

from PyQt5 import QtCore, QtWidgets

from matplotlib.backends.backend_qt5agg import FigureCanvasQTAgg
from matplotlib.figure import Figure


class MplCanvas(FigureCanvasQTAgg):

    def __init__(self, parent=None, width=5, height=4, dpi=100):
        fig = Figure(figsize=(width, height), dpi=dpi)
        self.axes = fig.add_subplot(111)
        super(MplCanvas, self).__init__(fig)


class MainWindow(QtWidgets.QMainWindow):

    def __init__(self, *args, **kwargs):
        super(MainWindow, self).__init__(*args, **kwargs)

        # Create the maptlotlib FigureCanvas object,
        # which defines a single set of axes as self.axes.
        sc = MplCanvas(self, width=5, height=4, dpi=100)
        sc.axes.plot([0,1,2,3,4], [10,1,20,3,40])
        self.setCentralWidget(sc)

        self.show()


app = QtWidgets.QApplication(sys.argv)
w = MainWindow()
app.exec_()
```

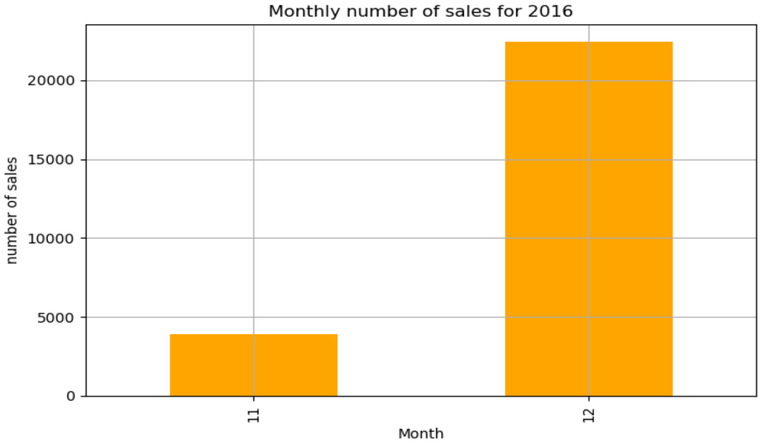
## Results

- **Plotting Year wise number of sales**

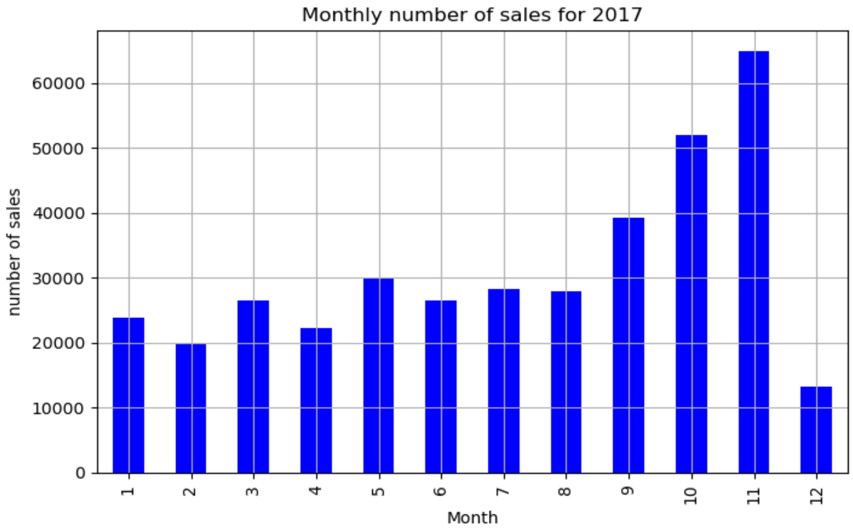


We can observe in the pie chart, in the year 2016 it records 6.6% which is very low as compared to the sales in the year 2017 which is 93.4%

- **Plotting monthly sales by year**

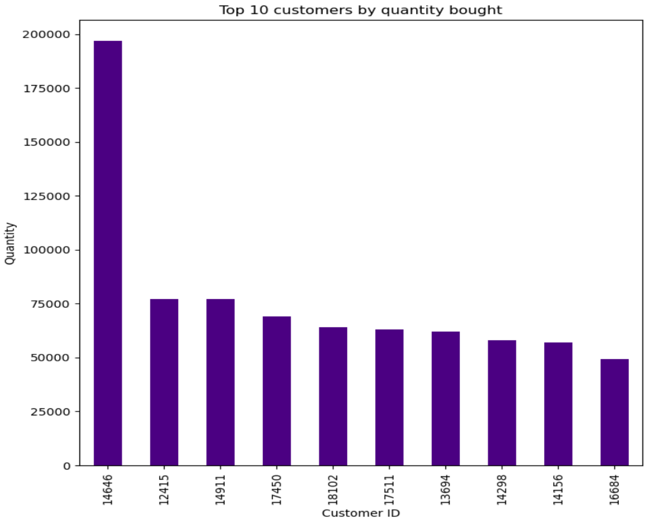


From the above bar graph, we can see in the year 2016, November has less than 5000 sales whereas in December it has more 20000 sales.



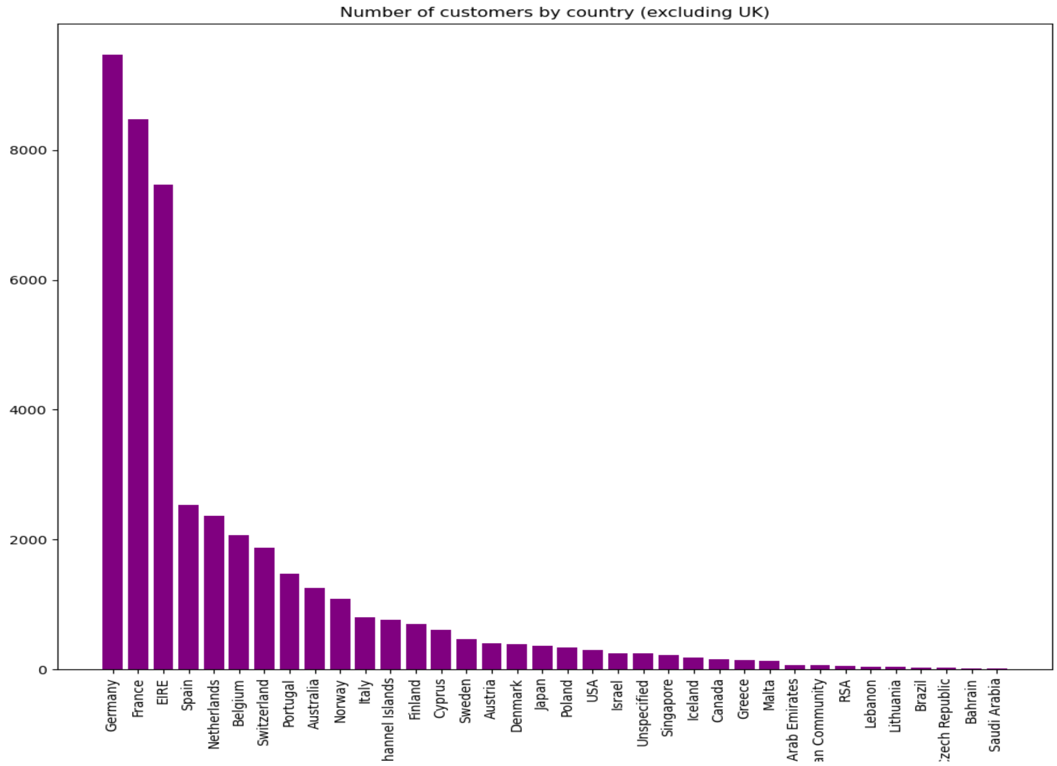
For the year 2017, The first six months sales fluctuate up and down not crossing more than 30k sales, but from July we can observe it starts increasing drastically till November which records highest number of sales of more than 60k, then suddenly sales drop in December to lowest of all.

- **Plotting top 10 customers after summing their purchase quantities**



From the above plot, We have taken top 10 customers who bought more, The customerID with14646 has bought highest number of quantities, who can be considered as the highest valued customer.

- **Plotting country wise sales (Except UK)**



We see in the above bar chart, Germany records second the highest sales (9,477) then coms France. We already know from above code UK has the first highest sales (3,56,616) with major difference.

## Summary and conclusions

In conclusion, I have done exploratory data analysis to understand better insights in the dataset, the year 2017 have more number of sales than 2016. The last six months has good increase in number of sales. Also, I have explored top 10 high valued customers in the dataset. Along with this, UK has highest number of sales then comes Germany. If we get to know more number of parameters in the dataset will be helpful for deep insights to determine high valued customers.

## Percentage of the code

lines of code from internet – 40
Modified – 10
Added – 26
Percentage = (40 – 10 / 40+26) x 100 = (30/66)x100 = 45%

# References

**https://www.kaggle.com/vik2012kvs/high-value-customers-identification**
**https://pandas.pydata.org/**
**https://matplotlib.org/**