High Value Customer Identification

Individual Report

Anuradha Tidke

Data Mining

Department of Data Science

George Washington University

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 work was divided between four of us as follows:

- Data cleaning: Priya
- EDA: Naveen and Rakshith
- Clustering: Anuradha (me)

All these code chunks were compiled and integrated by me to form the GUI for this project.

K-means algorithm

To process the learning data, the K-means algorithm in data mining starts with a first group of randomly selected centroids, which are used as the beginning points for every cluster, and then performs iterative (repetitive) calculations to optimize the positions of the centroids
It halts creating and optimizing clusters when either:

- The centroids have stabilized — there is no change in their values because the clustering has been successful.
- The defined number of iterations has been achieved.

Experimental Setup and my work

After the data was cleaned and the EDA graphs were plotted by my teammates, I studied those graphs to determine which variables I need to consider to run the clustering algorithm on. Initially the dataset had 9 variables which came down to 8 after cleaning. But I determined that using all 8 separately in the clustering algorithm won't provide any useful results. My teammates who had worked on the EDA had formed some grouped variables according to the "CustomerID". Following piece of code is done by my teammates:

I merged the following variables into a single dataframe for clustering:
df1[['Total Expenditure']], transactions and unique_items.
These variables were created by my teammates. I added one new variable “Avg. value per purchase” into this dataframe and called it cluster_df1. All the following pieces of codes are my individual work.

```
cluster_df1 = df1[['Total Expenditure']].merge(transactions, how='left', on='CustomerID')
cluster_df1['Avg. value per purchase'] = cluster_df1['Total Expenditure']/cluster_df1['Total Purchases']
cluster_df1 = cluster_df1.merge(unique_items, how='left', on='CustomerID')
```

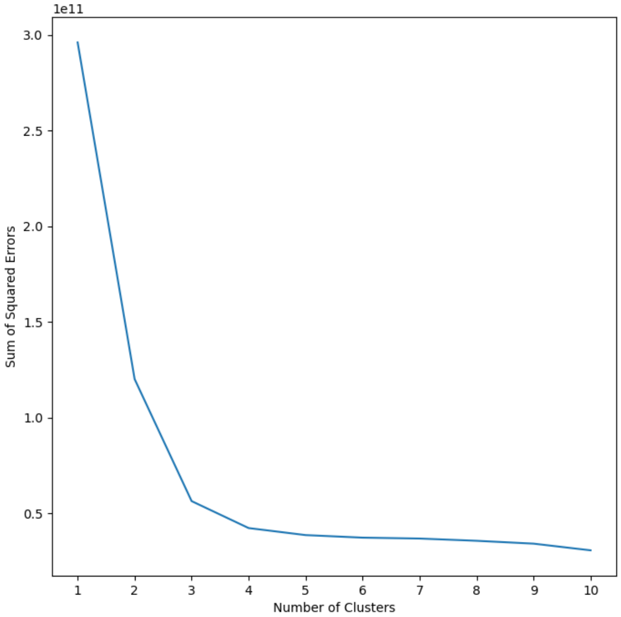
I created an array X from cluster_df1 to run my KMeans algorithm on.

```
X = cluster_df1[['Total Expenditure', 'Total Purchases', 'Avg. value per purchase','No. of unique items']]
```

To determine the best value of K for my algorithm, I plotted Sum of Squared Errors (SSE) for a range of values of K (1-11).

```
sum_sqEr = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42, init='random', n_init=10, max_iter=10)
    kmeans.fit(X)
    sum_sqEr.append(kmeans.inertia_)
```

I plotted sum_sqEr against k and following is the plot I obtained:

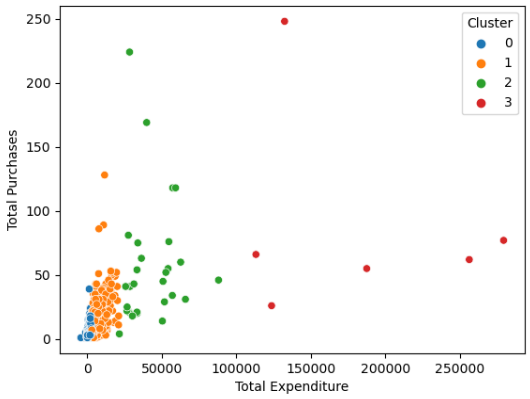


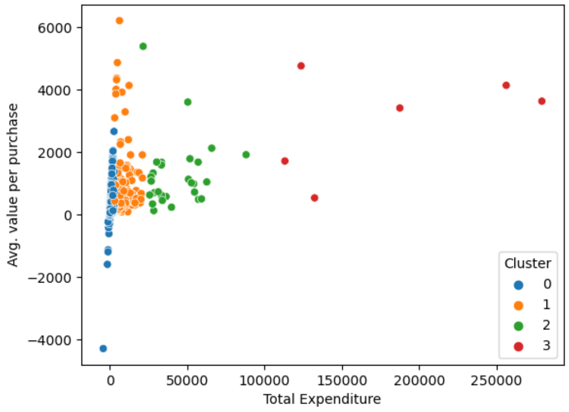
Using the Elbow method I decided to take k = 4 and run the algorithm to determine the cluster for each datapoint in our dataset:

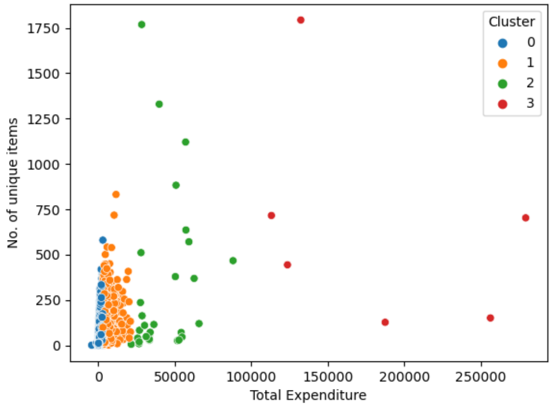
```
kmeans = KMeans(n_clusters=4, random_state=42, init='random', n_init=10, max_iter=10)
kmeans.fit(X)
cluster_df1['Cluster']=kmeans.predict(X)
```

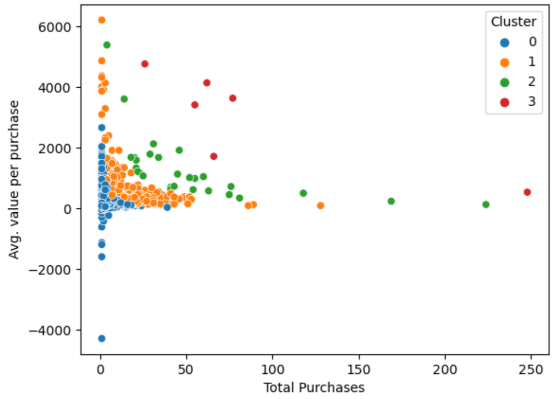
Using the “Cluster” column added to the cluster_df1 dataframe, I plotted the scatterplots between each pair of variables with legends set according to the cluster number.

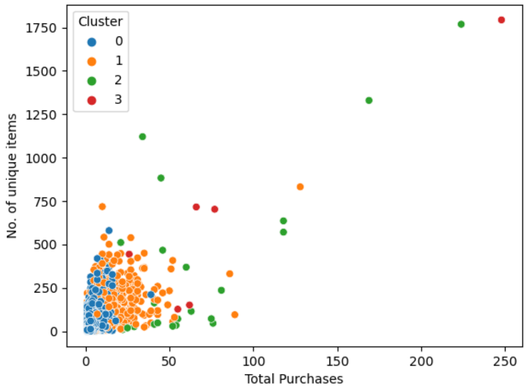
```
ar= ['Total Expenditure', 'Total Purchases', 'Avg. value per purchase','No. of unique items']
for i in range(3):
    for j in range(i+1,4):
        sns.scatterplot(x=ar[i], y=ar[j], hue='Cluster', data=cluster_df1, palette='tab10')
        plt.show()
```

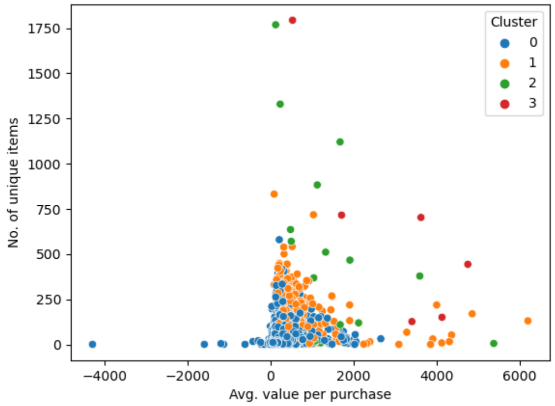






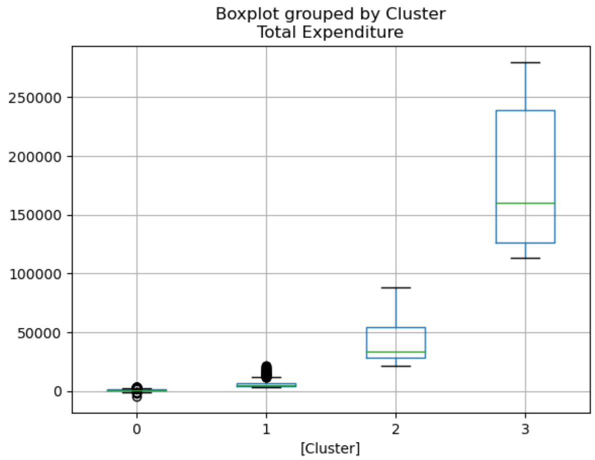


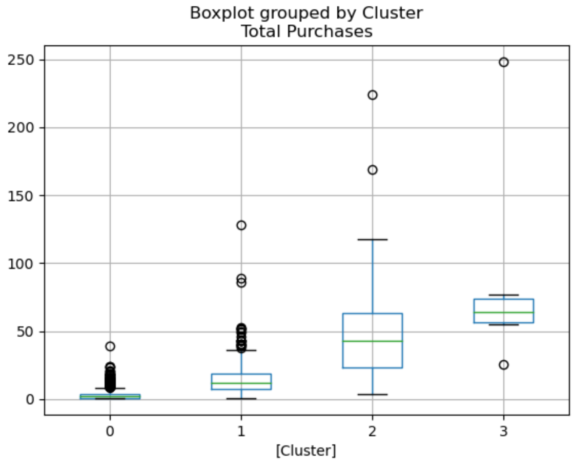


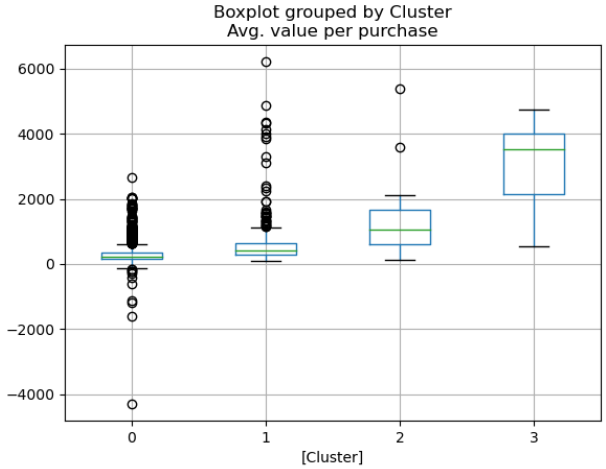


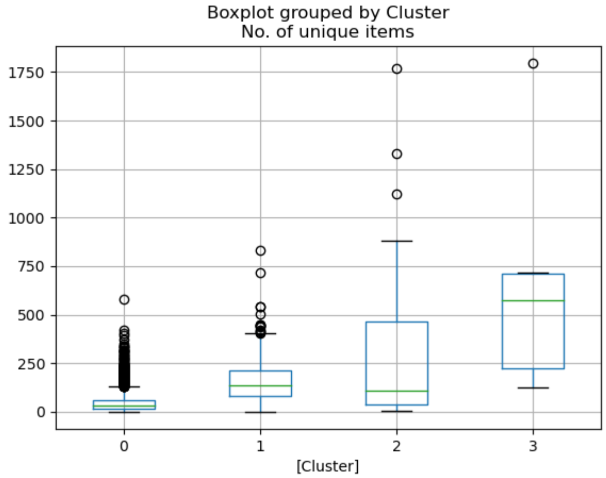
To verify the results I obtained from the above graphs, I plotted boxplots as well for these 4 variables as follows:

```
for i in range(4):
    cluster_df1.boxplot(column=ar[i], by=["Cluster"])
    plt.show()
```









All the above chunks of code I wrote are uploaded in my individual folder “Anuradha-Tidke-Individual-project”

Graphical User Interface:

I used the EDA codes written by my teammates and compiled them into a GUI using PyQt. I studied the demo and tutorial files uploaded by Professor and edited them for my project. I defined multiple classes for the EDA and Clustering part of our code. I worked on dimensioning the canvas and the main window of the GUI and added the dropdown menus and subplots in them.

The ProjectGUI.py file uploaded in the main "Code" folder is the one I fully worked on by myself without any help from my teammates.

Observations:

From the scatter plots showing different clusters, I concluded a few things and assigned some characteristics to the clusters.

Cluster 0: Total expenditure is the least of all, total purchases are minimal and a minimal number of unique items are bought by them.

Cluster 1: Least total expenditure is observed. Total purchases are moderate. These customers have bought a large number of unique items.

Cluster 2: Total expenditure is more than clusters 0 and 1, number of unique items bought is on a higher side as compared to the first two clusters.

Cluster 3: Total expenditure is the maximum though the total purchases don't vary much.

Project Conclusions:

Based on these results from the cluster feature graphs and the box plots, we can categorize the clusters as follows:

Cluster 0 and **1** need to be focussed **more** on, in terms of **discounts** and other **offers** in order to increase their purchase numbers.

Cluster 2 consists of **regular** value but **loyal** customers who visit the store pretty often.

Cluster 3 can be considered as **high-valued customers** for whom a loyalty program should be rolled out. These customers are loyal as well seeing the number of purchases

Summary:

Kmeans method worked well for our dataset as we obtained some good promising results in the form of the clusters. After interpreting the clusters, we could categorize the customers in 4 different groups which was a success.

I learned to create useful data variables from the feature variables already in the dataset. I tried using the whole dataset first but it didn't create any useful plots and results. So, I had to group the data by "CustomerID" and use those newly formed features.

Regarding the KMeans algorithm, I learned how it works and how different parameters can affect the result, which was a really important thing to understand. Also, I learnt the limitations of the same which is that it doesn't work well if the data can't be clustered in spheres.

Percentage of copied code:

Main code (excluding the GUI code) (Project.py):

Copied from internet: 8

Modified: 3

Added by myself: 12

Copy percentage = 25%

GUI code (ProjectGUI.py): I used the same code (Main.py) that Professor has uploaded at Data-Mining-master\Demo\PyQt5\Demo. I am considering it as "used from the internet". I modified and added some of my own as well.

Copied from internet: 367

Modified: 126

Added by myself: 145

Copy percentage = 47%

Overall copy percentage: 46.2%

References

https://www.kaggle.com/vik2012kvs/high-value-customers-identification

https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1

https://www.kdnuggets.com/2019/11/customer-segmentation-using-k-means-clustering.html

https://en.wikipedia.org/wiki/K-means_clustering

https://pandas.pydata.org/

https://matplotlib.org/