Credit Card Data Customer Segmentation To Define Marketing Strategy

Allison Tsang, Reshma Roy, Wangheng (Maggie) Hu, Rishik Adhikari

Basic Methods of Data Analytics CSDA1010-033-B-W21S1

Hashmat Rohian MSC, FNENG, CISSP, PMP, CFE, ACP, TOGAF, LEAN SIX SIGMA

Tuesday May 4th 2021

Introduction:

Marketing has always been a business's key tool in driving more awareness and ultimately sales. As in many other business functions, cost cutting, has forced marketing teams to be more efficient with their campaigns. An increase in globalization and online purchasing capabilities have resulted in large customer bases. Marketing teams are now challenged with only being able to target a select portion of their customer base for specific campaigns. Ultimately, targeting a specific segment of customers makes sense fiscally for the business as not everyone will respond the same to promotional messaging.

Our team will be looking at active credit card holder data from a six month time period. The goal would be to help the marketing teams cluster groups of credit card holders. Then further analysis can be done on these clusters to better understand their product needs. The following report will walk through the business problem, analytical objective, data understanding, data preparation, modelling goals, deployment, and performance assessment. Jupiter notebooks was used to clean, analyze, and model the data set. A variety of libraries within Pandas such as pandas, scipy, numpy, matplotlib, and seaborn was leveraged.

Business Understanding & Objective:

Credit card companies accumulate, track, and collect enormous amounts of data but how can internal teams leverage this to improve the business. For marketing teams the use of data is a lot more clear. Data can be end to end in marketing. Starting with identifying opportunities for promotional campaigns, understand which customers to target, how well the campaign went, etc. By leveraging data, marketing teams can use their budget more effectively to drive better results.

Transactional credit card data will sometimes need to be amalgamated in order to derive insights. This data set would have key credit card behaviour information on each customer. Important information to include would be: credit limit, min payments, cash advances, frequency of usage, etc. This project would look at using this amalgamated customer credit card data and how to cluster customers into various groups. Once these groups are created, the marketing team can then take a deeper dive and understand what promotional campaigns would interest them.

Data understanding:

The data set was acquired from Kaggle (https://www.kaggle.com/arjunbhasin2013/ccdata/). It includes usage behaviour data of about nine thousand active credit card holders from a 6 month span. Below is a data dictionary of the behavioral data set.

Attribute	Description
CUSTID	Identification of Credit Card holder (Categorical)
BALANCE	Balance amount left in their account to make purchases
BALANCEFREQUENCY	How frequently the Balance is updated, score between 0 and 1 (1 = frequently updated, 0 = not frequently updated)
PURCHASES	Amount of purchases made from account
ONEOFFPURCHASES	Maximum purchase amount done in one-go
INSTALLMENTSPURCHASES	Amount of purchase done in installment
CASHADVANCE	Cash in advance given by the user
PURCHASESFREQUENCY	How frequently the Purchases are being made, score between 0 and 1 (1 = frequently purchased, 0 = not frequently purchased)
ONEOFFPURCHASESFREQUENC	How frequently Purchases are happening in one-go (1 = frequently purchased, 0 = not frequently purchased)
PURCHASESINSTALLMENTSFREQUENCY	How frequently purchases in installments are being done (1 = frequently done, 0 = not frequently done)
CASHADVANCEFREQUENCY	How frequently the cash in advance being paid
CASHADVANCETRX	Number of Transactions made with "Cash in Advanced"
PURCHASESTRX	Number of purchase transactions made
CREDITLIMIT	Limit of Credit Card for user
PAYMENTS	Amount of Payment done by user
MINIMUM_PAYMENTS	Minimum amount of payments made by user
PRCFULLPAYMENT	Percent of full payment paid by user
TENURE	Tenure of credit card service for user

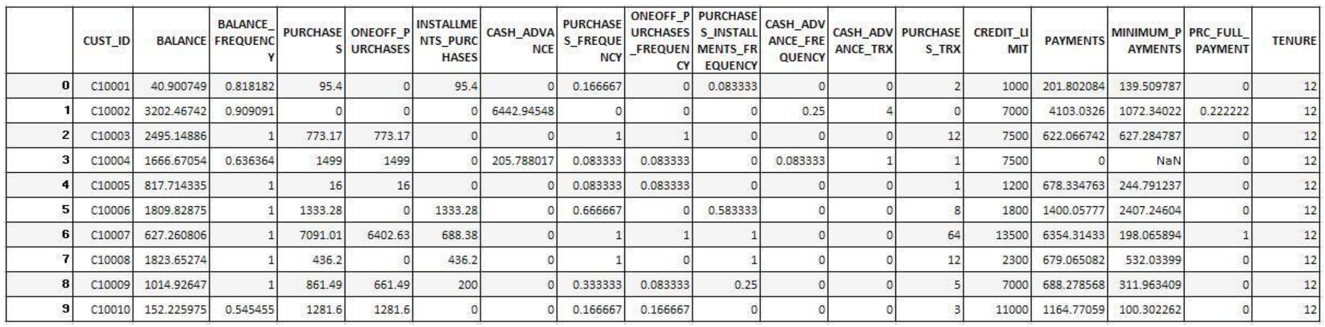
Data Preparation:

To perform the analysis, certain python libraries were used. The code below was used to load and initialize the library, then loads the data.

```
import pandas as pd
import numpy as np
from pathlib import Path
import matplotlib.pyplot as plt
from matplotlib.cbook import boxplot_stats
import seaborn as sns
```

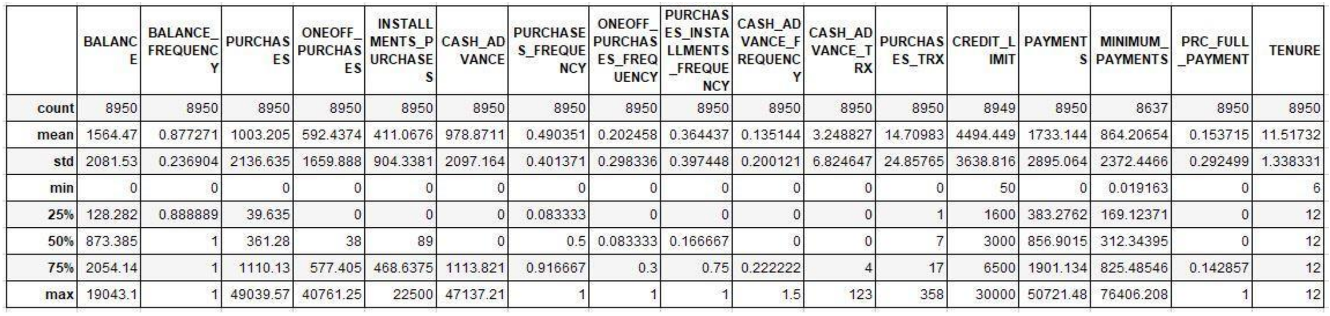
```
%matplotlib inline

df = pd.read_csv('../input/ccdata/CC GENERAL.csv')
```

Preview of the data:
Quick view of the data attributes statistics are presented in Table 2. For each attribute in the dataset the table shows min, max, mean and normal distribution 1st and 3rd quartiles values. The first 10 rows of the dataset are presented in Table . From Table 3, Mininum_Payments and credit_limit have missing values.



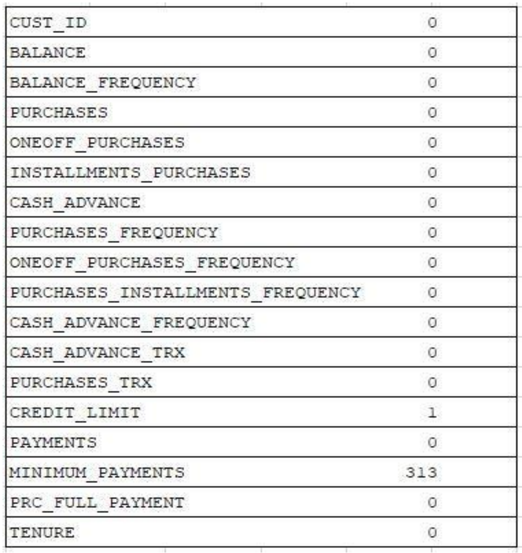
	CUST_ID	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTALLMENTS_PURCHASES	CASH_ADVANCE	PURCHASES_FREQUENCY	ONEOFF_PURCHASES_FREQUENCY	PURCHASES_INSTALLMENTS_FREQUENCY	CASH_ADVANCE_FREQUENCY	CASH_ADVANCE_TRX	PURCHASES_TRX	CREDIT_LIMIT	PAYMENTS	MINIMUM_PAYMENTS	PRC_FULL_PAYMENT	TENURE
0	C10001	40.900749	0.818182	95.4	0	95.4	0	0.166667	0	0.083333	0	0	2	1000	201.802084	139.509787	0	12
1	C10002	3202.46742	0.909091	0	0	0	6442.94548	0	0	0	0.25	4	0	7000	4103.0326	1072.34022	0.222222	12
2	C10003	2495.14886	1	773.17	773.17	0	0	1	1	0	0	0	12	7500	622.066742	627.284787	0	12
3	C10004	1666.67054	0.636364	1499	1499	0	205.788017	0.083333	0.083333	0	0.083333	1	1	7500	0	NaN	0	12
4	C10005	817.714335	1	16	16	0	0	0.083333	0.083333	0	0	0	1	1200	678.334763	244.791237	0	12
5	C10006	1809.82875	1	1333.28	0	1333.28	0	0.666667	0	0.583333	0	0	8	1800	1400.05777	2407.24604	0	12
6	C10007	627.260806	1	7091.01	6402.63	688.38	0	1	1	1	0	0	64	13500	6354.31433	198.065894	1	12
7	C10008	1823.65274	1	436.2	0	436.2	0	1	0	1	0	0	12	2300	679.065082	532.03399	0	12
8	C10009	1014.92647	1	861.49	661.49	200	0	0.333333	0.083333	0.25	0	0	5	7000	688.278568	311.963409	0	12
9	C10010	152.225975	0.545455	1281.6	1281.6	0	0	0.166667	0.166667	0	0	0	3	11000	1164.77059	100.302262	0	12

Table 1: Credit Card Data (first 10 rows)



	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTALLMENTS_PURCHASES	CASH_ADVANCE	PURCHASES_FREQUENCY	ONEOFF_PURCHASES_FREQUENCY	PURCHASES_INSTALLMENTS_FREQUENCY	CASH_ADVANCE_FREQUENCY	CASH_ADVANCE_TRX	PURCHASES_TRX	CREDIT_LIMIT	PAYMENTS	MINIMUM_PAYMENTS	PRC_FULL_PAYMENT	TENURE
count	8950	8950	8950	8950	8950	8950	8950	8950	8950	8950	8950	8950	8949	8950	8637	8950	8950
mean	1564.47	0.877271	1003.205	592.4374	411.0676	978.8711	0.490351	0.202458	0.364437	0.135144	3.248827	14.70983	4494.449	1733.144	864.20654	0.153715	11.51732
std	2081.53	0.236904	2136.635	1659.888	904.3381	2097.164	0.401371	0.298336	0.397448	0.200121	6.824647	24.85765	3638.816	2895.064	2372.4466	0.292499	1.338331
min	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0.019163	0	6
25%	128.282	0.888889	39.635	0	0	0	0.083333	0	0	0	0	1	1600	383.2762	169.12371	0	12
50%	873.385	1	361.28	38	89	0	0.5	0.083333	0.166667	0	0	7	3000	856.9015	312.34395	0	12
75%	2054.14	1	1110.13	577.405	468.6375	1113.821	0.916667	0.3	0.75	0.222222	4	17	6500	1901.134	825.48546	0.142857	12
max	19043.1	1	49039.57	40761.25	22500	47137.21	1	1	1	1.5	123	358	30000	50721.48	76406.208	1	12

Table 2: Credit Card Data Attributes Summary



CUST_ID	0
BALANCE	0
BALANCE_FREQUENCY	0
PURCHASES	0
ONEOFF_PURCHASES	0
INSTALLMENTS_PURCHASES	0
CASH_ADVANCE	0
PURCHASES_FREQUENCY	0
ONEOFF_PURCHASES_FREQUENCY	0
PURCHASES_INSTALLMENTS_FREQUENCY	0
CASH_ADVANCE_FREQUENCY	0
CASH_ADVANCE_TRX	0
PURCHASES_TRX	0
CREDIT_LIMIT	1
PAYMENTS	0
MINIMUM_PAYMENTS	313
PRC_FULL_PAYMENT	0
TENURE	0

Table 3: Number of Missing Values in each Attributes

Observations:

Based on the initial cursory look at the data, some insights were identified. Some of these insights might be useful to revisit once customer segments have been created.

- BALANCE : There are no negative values. The mean is about 1564 whereas the median is about 873, suspect this variable for having a distribution skewed to the right. This measure is a challenge to decide whether it has value or not as it is based on 'time' this data is captured. For example, a customer could have a high balance but that's simply because their payment date has not come yet. On the other hand, there could be a high balance as the customer has not been paying off their amount owing.
- For ONEOFFPURCHASES , INSTALLMENTSPURCHASES ,PURCHASES, CASHADVANCE : The mean is higher compared to the median, suspect this variable for having a distribution skewed to the right.
- PURCHASESFREQUENCY : The range of values is from 0 to 1 with a mean of 0.5. The credit card holders in this data set is well balance on this measure.
- ONEOFFPURCHASESFREQUENCY : The range of values is from 0 to 1 with a mean of 0.08.
- PURCHASESINSTALLMENTSFREQUENCY : The range of values is from 0 to 1 with a mean of 0.16. This indicates that generally these credit card users do not make purchase installments. Looking at ONEOFF PURCHASE with this measure, one can loosely conclude generally these credit card holders are relatively in a good financial situation.
- CASHADVANCEFREQUENCY : The range of values is from 0 to 1 with a mean of 0.22.
- CASHADVANCETRX : Maximum value at 123, where the Q3 equals to 4.
- PURCHASESTRX : Maximum value at 358, where the Q3 equals to 17.

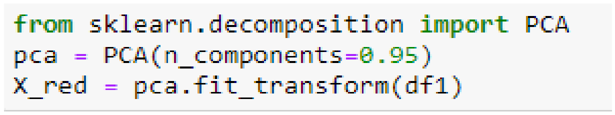
- CREDITLIMIT : The range of values is from $50 to $30000 which is very large. Looking at the mean of PURCHASE and CREDITLIMIT we can see that customers are utilizing about 25% of their credit limit
- PAYMENTS : Maximum value at 50721, where the Q3 equals to 1901.
- MINIMUM_PAYMENTS : Maximum value at 76406, where the Q3 equals to 825. With the mean min payment well above $10 (which is typically the min payment one must make), it may indicate a financially sound customer base.
- PRCFULLPAYMENT : Most of users are not paying fully.
- TENURE : Most users have a value at 12 meaning these are long time / loyal customers to our business. Few users have a value at 6.
- Overall on the measures which were rated between 0 and 1, it is seen that customers are generally not that active in leveraging these credit card tools.

Modelling

With this data and the goal of customer segmentation, a Kmeans cluster model was the most appropriate choice. The team worked on two methods of data cleaning to find the a strong option. The first (which will be referred to as Strategy 1) was a Principal Component Analysis. The second was a Kmeans model (which will be referred to as Strategy 2).

Strategy 1:

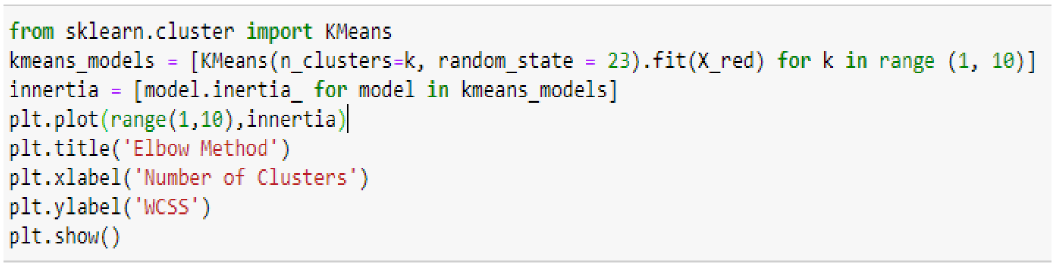
A Principal Component Analysis (PCA) was used for dimensionality reduction. PCA was used for dimensionality through projecting each data point onto only the first few principal components. This will results in obtaining lower-dimensional data while preserving as much of the data's variation as possible. The dataset was processed so that data in the lower dimensions explains 95% of variation of our original data. Below is our code to accomplish this.



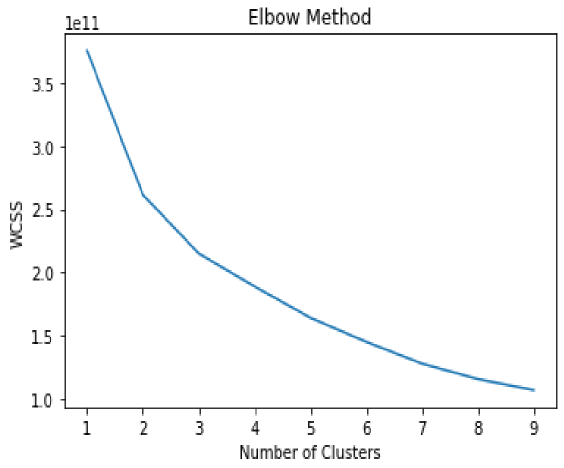
```
from sklearn.decomposition import PCA
pca = PCA(n_components=0.95)
X_red = pca.fit_transform(df1)
```

Cluster Analysis:

The Kmeans clustering algorithm was used to define clusters of information from our dataset. This algorithm identifies 'k' number of centroids, and then allocates every data point to the nearest cluster, while keeping the centroids as small as possible. ***n*** inertia metric is not considered because with the increase in the number of clusters inertia keeps on increasing. Plotting the graph of n versus inertia we can observe an elbow after which the inertia declines at a much lower rate. Based on this model, it seems that the elbow is approximately 3 or 4 (see below).



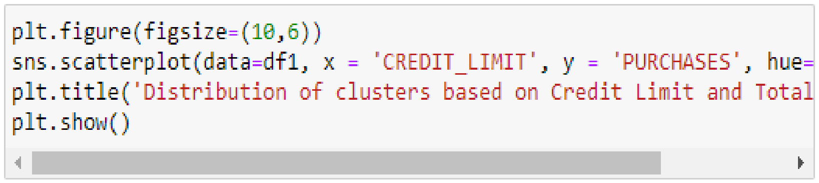
```
from sklearn.cluster import KMeans
kmeans_models = [KMeans(n_clusters=k, random_state = 23).fit(X_red) for k in range (1, 10)]
innertia = [model.inertia_ for model in kmeans_models]
plt.plot(range(1,10),innertia)
plt.title('Elbow Method')
plt.xlabel('Number of Clusters')
plt.ylabel('WCSS')
plt.show()
```



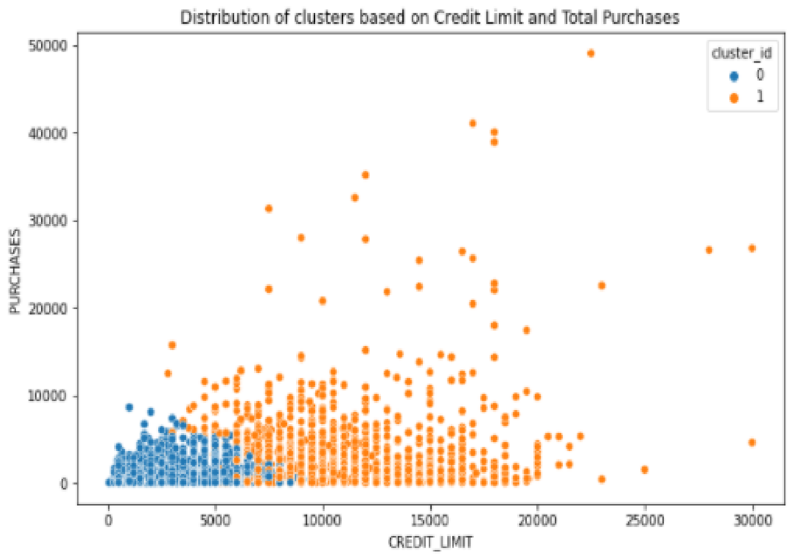
Another method is to use silhouette score as a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The silhouette score ranges from -1 to +1, where a high value implies that the object is well matched to its cluster and poorly matched to neighboring clusters. When this was run in against our model, a value of 0.5208 indicates a moderate silhouette score which is not too bad. Possibly with other data preparations techniques before model the score could be higher.

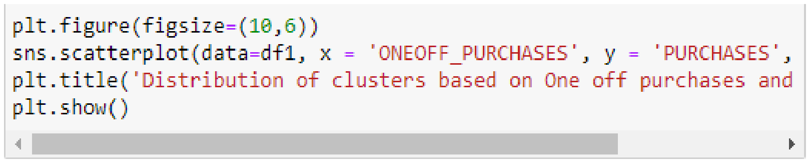
These two graphs demonstrate that Strategy 1 has clustered customers with low usage or credit card in one cluster and with higher usage in the other cluster.

Strategy 2:

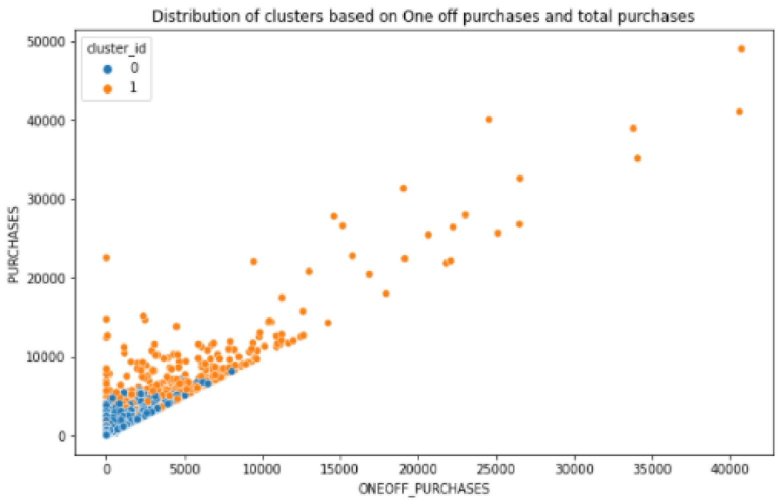


```
plt.figure(figsize=(10,6))
sns.scatterplot(data=df1, x = 'CREDIT_LIMIT', y = 'PURCHASES', hue=
plt.title('Distribution of clusters based on Credit Limit and Total
plt.show()
```





```
plt.figure(figsize=(10,6))
sns.scatterplot(data=df1, x = 'ONEOFF_PURCHASES', y = 'PURCHASES',
plt.title('Distribution of clusters based on One off purchases and
plt.show()
```



Determining optimal number of clusters:

Based on the Elbow Method, it appears k=2 seems to have higher silhouette scores than k = 3 and k = 4 (see graph on 6).

Strategy 2:

In the second strategy, only Standard Scaler was used to prepare data for PCA as the thinking was that PCA worked better on scaled data.

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
```

```
scaler.fit(data)
scaled_data = scaler.transform(data)
scaled_data
```

```
array([[-0.74462486, -0.37004679, -0.42918384, ..., -0.30550763,
        -0.53772694,  0.35518066],
       [ 0.76415211,  0.06767893, -0.47320819, ...,  0.08768873,
         0.21238001,  0.35518066],
       [ 0.42660239,  0.50540465, -0.11641251, ..., -0.09990611,
        -0.53772694,  0.35518066],
       ...,
       [-0.75297728, -0.29709491, -0.40657175, ..., -0.32957217,
         0.30614422, -4.22180042],
       [-0.75772142, -0.29709491, -0.47320819, ..., -0.34081076,
         0.30614422, -4.22180042],
       [-0.58627829, -1.09958965,  0.03129519, ..., -0.32709767,
        -0.53772694, -4.22180042]])
```

Then, the whole data set was divided into two Principal Component Analysis because it would be easier to plot the clusters later.

```
from sklearn.decomposition import PCA
pca = PCA(n_components = 2)
```

```
pca.fit(scaled_data)
x_pca=pca.transform(scaled_data)
```

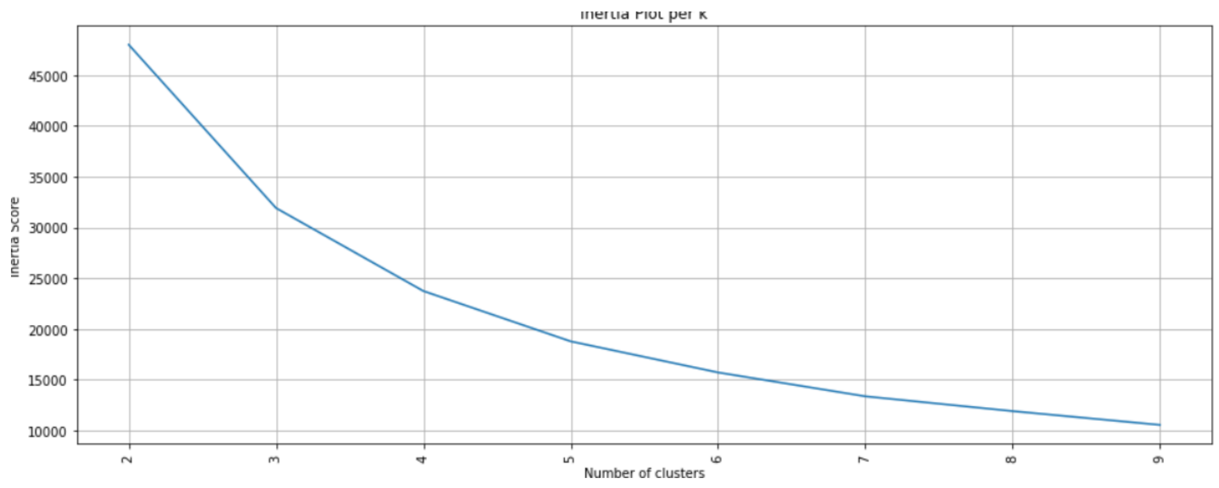
Finally, the model was run multiple times with different numbers of clusters to choose the best option through Elbow Method.

```
no_of_clusters = range(2,10)
inertia = []


for f in no_of_clusters:
    kmeans = KMeans(n_clusters=f, random_state=2)
    kmeans = kmeans.fit(x_pca)
    u = kmeans.inertia_
    inertia.append(u)
    print("The innertia for :", f, "Clusters is:", u)
```

```
The innertia for : 2 Clusters is: 48011.929716576764
The innertia for : 3 Clusters is: 31903.348590860947
The innertia for : 4 Clusters is: 23720.107847338197
The innertia for : 5 Clusters is: 18755.912071489067
The innertia for : 6 Clusters is: 15688.887467383627
The innertia for : 7 Clusters is: 13333.91584444013
The innertia for : 8 Clusters is: 11874.06533033619
The innertia for : 9 Clusters is: 10515.587475744276
```

Determining optimal number of clusters:

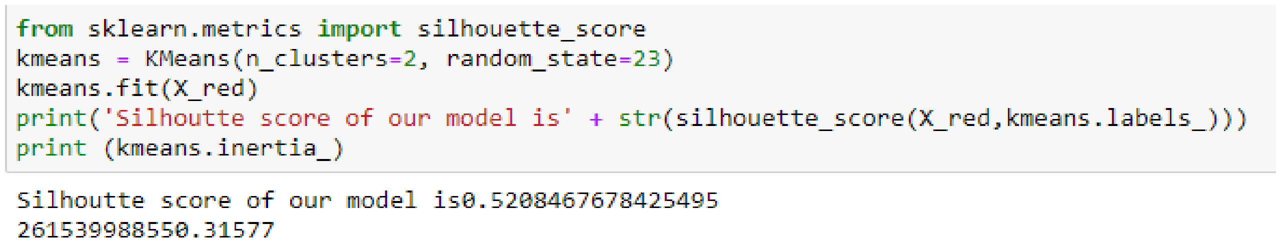
The inertia calculated (see page 8) was plotted to get the optimum number of clusters. Using the graphical chart below, 3 clusters was chosen as the optimum number of clusters for results.



Evaluation:

Strategy 1: Clustering using K-means method & Elbow

A mild silhouette score of 0.521 implies that this strategy is not a bad clustering model. It's right in the middle of the thresholds. However the Kmeans algorithm clusters data is an attempt to separate a dataset in k groups of equal variance, minimizing a criterion known as the inertia. Therefore, from this standpoint, this strateagy did not perform well in minimizing the inertia metric with a high value of 261,539,988,550.

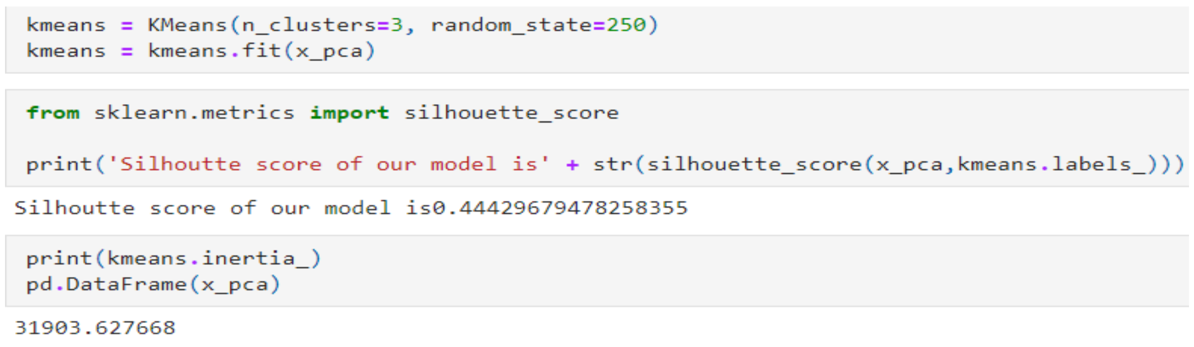


```
from sklearn.metrics import silhouette_score
kmeans = KMeans(n_clusters=2, random_state=23)
kmeans.fit(X_red)
print('Silhoutte score of our model is' + str(silhouette_score(X_red,kmeans.labels_)))
print (kmeans.inertia_)
```

```
Silhoutte score of our model is0.5208467678425495
261539988550.31577
```

<u>Strategy 2:</u> Clustering using K-means method, Standard Scaler & Elbow

After running the KMeans model with 3 numbers of clusters, it's inertia was 31904 and Silhouette score was 0.45.



```
kmeans = KMeans(n_clusters=3, random_state=250)
kmeans = kmeans.fit(x_pca)
```

```
from sklearn.metrics import silhouette_score

print('Silhoutte score of our model is' + str(silhouette_score(x_pca,kmeans.labels_)))
```

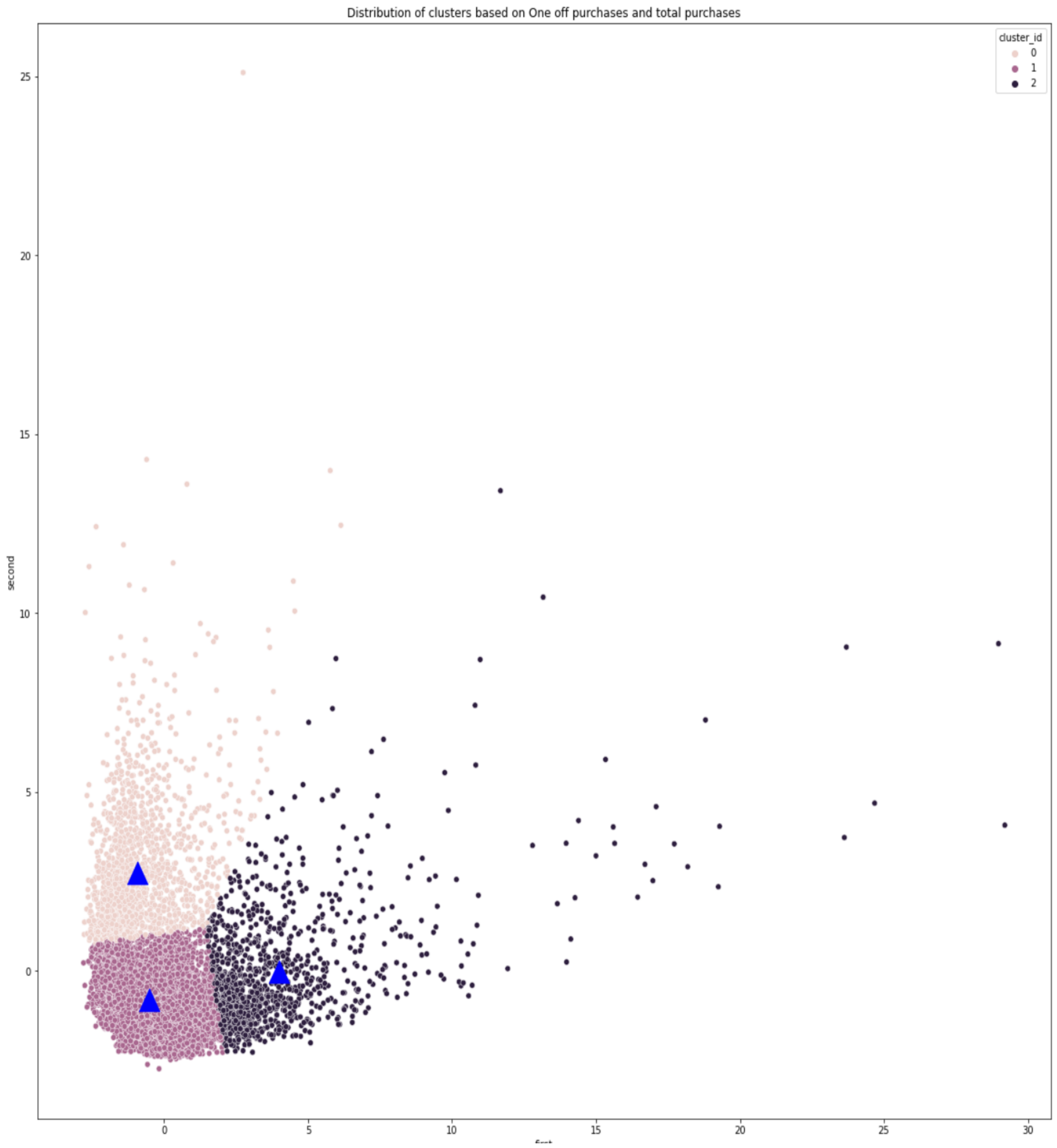
```
Silhoutte score of our model is0.44429679478258355
```

```
print(kmeans.inertia_)
pd.DataFrame(x_pca)
```

```
31903.627668
```

Although the silhouette score of the model is not as good as strategy 1, the inertia is much better than the previous model. Looking at the plotted clusters after prediction, the data were scattered very closely in a large amount which caused the silhouette score to be a little low. The silhouette score calculates the distance between centroids while inertia calculates the closeness of data points within clusters. For the dataset used in this project, having good inertia value was important. Below is a graph of clusters after the prediction by model.

```
x1 = kmeans.cluster_centers_[:,0]
y1 = kmeans.cluster_centers_[:,1]

plt.figure(figsize=(20,20))
sns.scatterplot(data=x_pca,x='first',y='second', hue ='cluster_id')
plt.scatter(x1,y1,color='blue',s=500, marker='^')
plt.title('Distribution of clusters based on One off purchases and total purchases')
plt.show()
```



Conclusion:

In review of both strategies tested, Strategy 2 would be the better option to present back to the marketing team. As mentioned, the inertia score was far better on Strategy 2 and the sellout score was relatively similar (0.08 difference). Lastly, with all models, there could always be further data cleaning and preparation to come up with models that score higher.

Deployment:

In terms of deploying this model, the project team would have worked closely with the marketing team to tighten up the model further. It is assumed, that this credit card company already has data infrastructure to store, house, and process this data. The company likely uses a direction data connection with a system like Juypter Notebooks to all for data manipulation and processing. For the marketing team specifically, it would be suggested to re-run models for new campaigns each time. The rationale behind this is that, finically situations (personally and the market) can change the profile of customers. This will ensure target customers to engage are still relevant.