Project Name – Credit Card Segmentation

Contents

Chapter 1

Introduction

1.1 Problem Statement

The aim of this project is to develop a customer segmentation to define marketing strategy, by looking through their behavior/profile while using Credit Card. It helps in deploying effective marketing campaign or sales promotion to the targeted costumer.

Segmentation in marketing is a technique used to divide customers or other entities into groups based on attributes such as behaviour or demographics. It is useful to identify segments of customers who may respond in a similar way to specific marketing techniques such as email subject lines or display advertisements. As it gives businesses the ability to tailor marketing messages and timing to generate better response rates and provide improved consumer experiences.

1.2 Data

We have a data set called “credit-card-data.csv” which gives us 18 behavioural variables at customer level. Here CUST_ID is the identification of the customer, and it’s a categorical variable. Rest all are numerical variable.

The list of variable in the dataset are described below.

- CUST_ID : Identification of Credit Card holder (Categorical)
- BALANCE : Balance amount left in their account to make purchases
- BALANCE_FREQUENCY : 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
- ONEOFF_PURCHASES : Maximum purchase amount done in one-go
- INSTALLMENTS_PURCHASES : Amount of purchase done in installment
- CASH_ADVANCE : Cash in advance given by the user
- PURCHASES_FREQUENCY : How frequently the Purchases are being made, score between 0 and 1 (1 = frequently purchased, 0 = not frequently purchased)
- ONEOFFPURCHASESFREQUENCY : 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”
- PURCHASES_TRX : Number of purchase transactions made
- CREDIT_LIMIT : 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

1.3 Software Requirement

- R 3.6.1 for 64 bit
- Anaconda3 2019.07 for 64bit

Chapter 2

Methodology

2.1 Exploratory Data Analysis

Exploratory data analysis (EDA) is an approach to analyzing data sets to summarize their main characteristics, often with visual methods. EDA is for seeing what the data can tell us beyond the formal modelling or hypothesis testing task.

All the required libraries are installed and loaded in both the environments. The working directory is set. The data set given in the csv format, is loaded. We get all 18 variables. CUST_ID is factor in R and object in Python. Rest all variables are float / numerical in nature. Initial descriptive analysis of data was done.

2.2 Missing Value Analysis

Missing data (or missing values) is defined as the data value that is not stored for a variable in the observation of interest. The problem of missing data is relatively common in almost all research and can have a significant effect on the conclusions that can be drawn from the data. It is created due to human error, refusal to answer while surveying or faulty survey questions etc. The missing value can be either imputated or the observations containing the missing values can be ignores depending upon the percentage of missing value present in the data. We can use the central tendencies to fill the missing values like mean or median, or we can use KNN imputation. KNN imputation finds the nearest neighbours based on existing attributes using Euclidean or Manhattan distance.

To know which method should be used, we convert one of the known values to NA and apply all the methods. Whichever method gives the closest value is selected for missing value analysis.

The missing value percentage table is created to determine the amount of missing value in the variables. Variable MINIMUM_PAYMENTS and CREDIT_LIMIT are found to have missing values in them. In R, KNN imputation method is selected and in case of python, median method is selected.

2.3 Deriving new KPI's

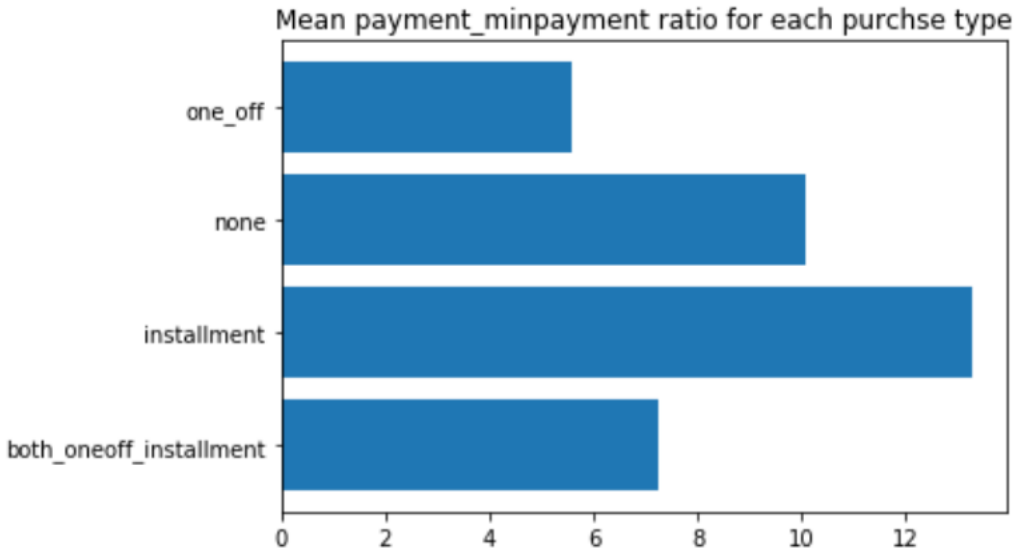
A Key Performance Indicator (KPI) is a measurable value that demonstrates how effectively a company is achieving key business objectives.

KPI's such as Monthly average purchase, Cash advance amount, balance to credit limit ratio(limit usage), Payment to minimum payments Ratio, Purchase type (To find what type of purchases customers are making on credit card). There are four types of purchase behaviour in the data set

(i)People who only do One-Off Purchases. (ii) People who only do Installments Purchases. (iii) People who do both. (iv) People who do none. In balance to credit limit ratio, lower value implies cutomers are maintaing thier balance properly. Lower value means good credit score.

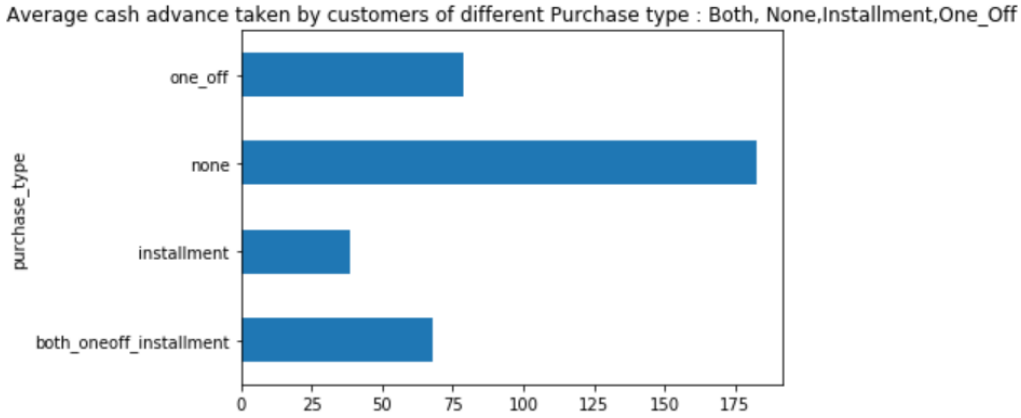
Insights from the KPI's:

Plot 1: Payment to minimum payment ratio for each purchase type



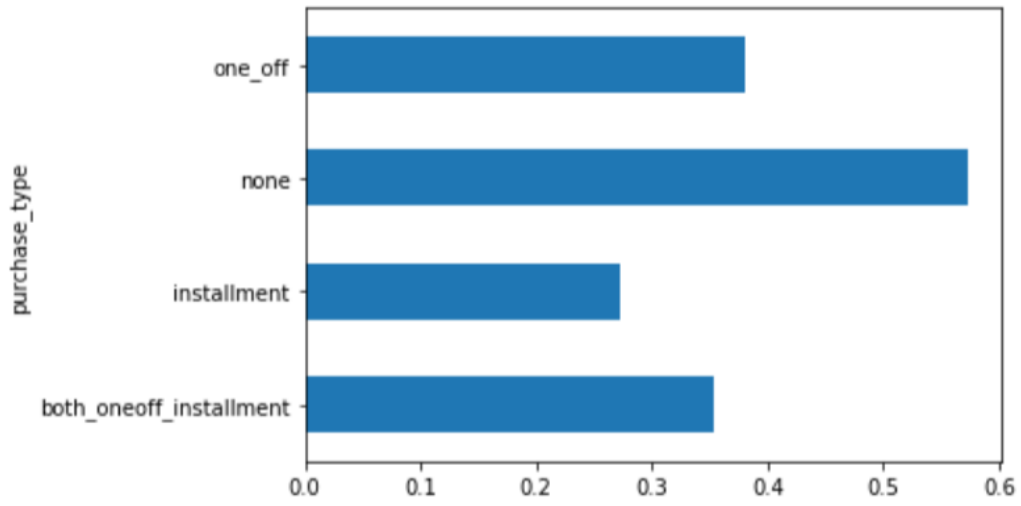
Insight 1: Customers With Installment Purchases are Paying Dues

Plot 2: Average cash advance taken by customers of different Purchase type : Both, None, Installment, One_Off



Insight 2: Customers who don't do either one-off or instalment purchases take more cash on advance.

Plot 3: limit usage (balance to credit limit ratio) for each purchase type



Insight 3: Customers with installment purchases have good credit score.

2.4 Preparing for machine learning

2.4.1 Feature selection

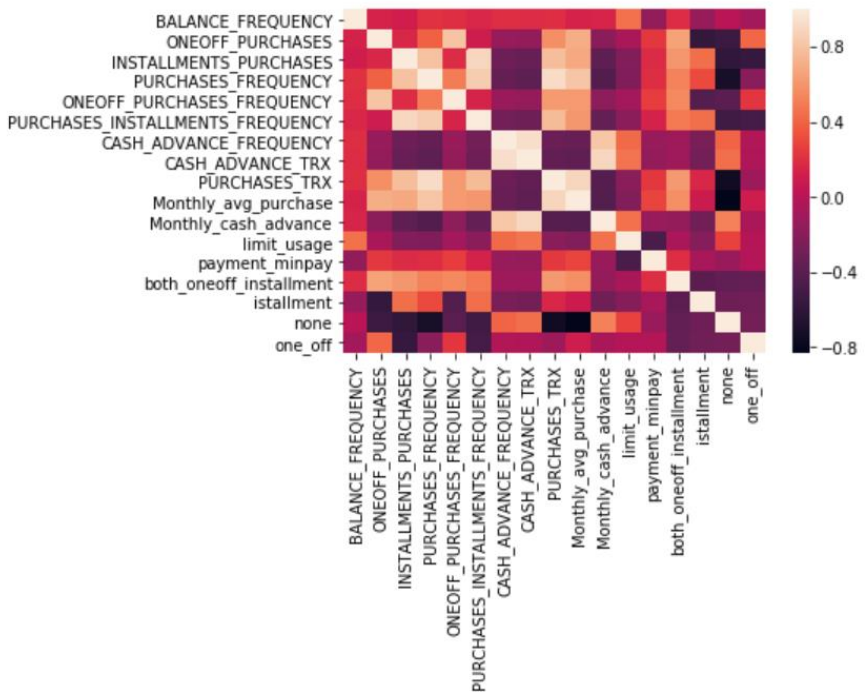
Feature selection means selecting a subset of relevant feature(Variables, predictors) for use in model construction.

Correlation Analysis - Correlation tells you the association between two continuous variables. It ranges from -1 to 1. Measures the direction and strength of the linear relationship between two quantitative variables.

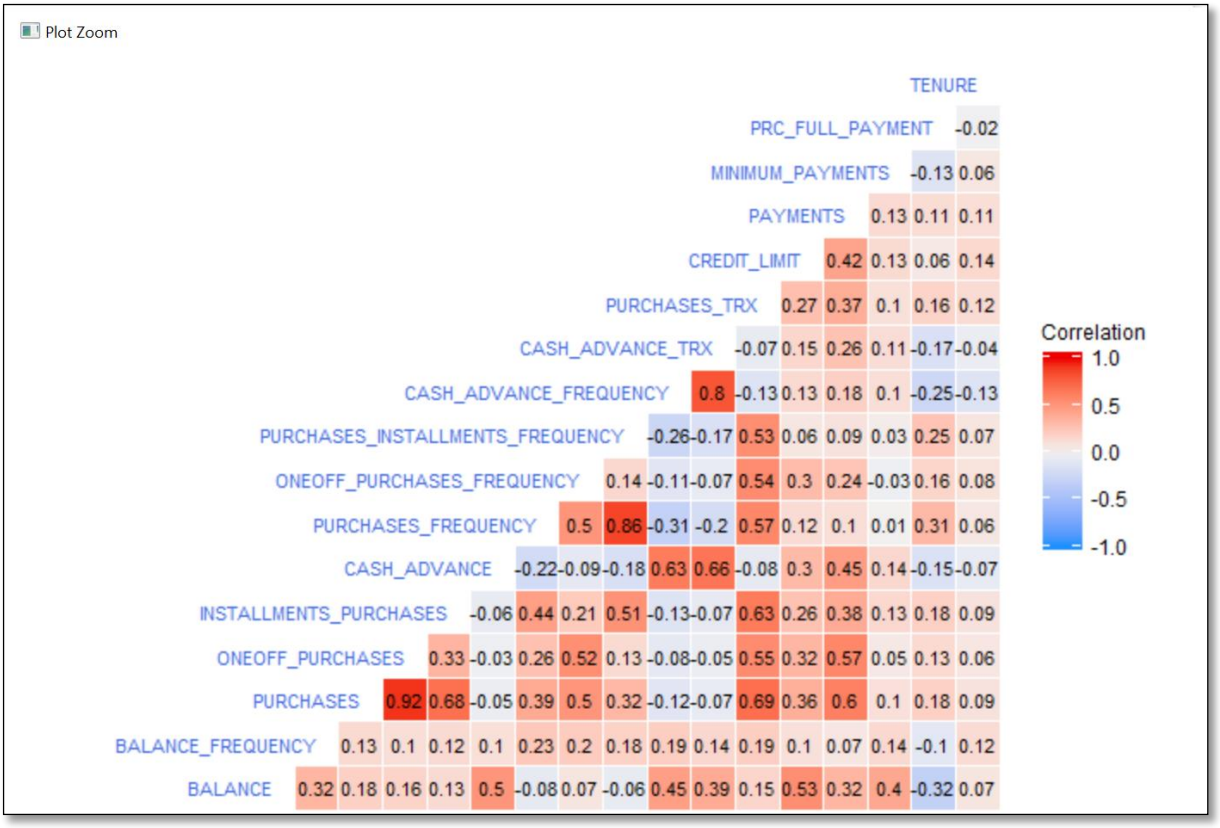
CUST_ID was a unique variable and we can't get further information from it.

The scale on most of the variables is different, but before scale it, we will take a closer look at the data, we will deal with the scaling at PCA Session.

- pairwise correlation of all columns in the dataframe



Correlation Between Variables using R:



- Since all of the variables are numerics, it will easier to see their correlation by using correlation matrix.

- Most of the variables are having positive correlation rather than negative one.

- Some of them even strongly correlated, just like PURCHASES and ONEOFF_PURCHASES, indicating most customers spend their purchase in one-go, the same thing goes to PURCHASES_FREQUENCY and PURCHASESINSTALLMENTSFREQUENCY. It indicates that there some strong multicollinearity between them. But since we are not going regression model, it won't be a problem.

- The PURCHASES and PURCHASES_TRX has strong correlation, indicating that the amount of purchase comes along with the transaction numbers.

- Some interesting finding can be also found between CASH_ADVANCE and the BALANCE. It seems that customer who has bigger balance will tend to have payment with cash in advance.

- Customer with bigger BALANCE tends to has a bigger CREDIT_LIMIT, since their correlation is positive (0.54).

- TENURE seems has weak correlation with other variables, it seems TENURE did not affected by the customer's behavior. We will check it again upon the PCA analysis.

2.4. PCA (Principal component analysis)

Principal Component Analysis (PCA) is an unsupervised, non-parametric statistical technique primarily used for "dimensionality reduction" in machine learning. It is basically a statistical procedure to convert a set of observation of possibly correlated variables into a set of values of linearly uncorrelated variables.

High dimensionality means that the dataset has a large number of features. The primary problem associated with high-dimensionality in the machine learning field is model overfitting, which reduces the ability to generalize beyond the examples in the training set. Richard Bellman described this phenomenon in 1961 as the Curse of Dimensionality where "Many algorithms that work fine in low dimensions become intractable when the input is high-dimensional."

Now the data is going to be scaled to a more normal distribution and has had any necessary outliers removed, we can now apply PCA to the data to discover which dimensions about the data best maximize the variance of features involved. In addition to finding these dimensions, PCA will also report the explained variance ratio of each dimension — how much variance within the data is explained by that dimension alone. Note that a component (dimension) from PCA can be considered a new "feature" of the space, however it is a composition of the original features present in the data.

2.4.1 Standardizing the data

PCA is a unsupervised learning algorithm that's affected by scale. We standardize data to avoid effect of scale on our result. We standardize the dataset's variables onto unit scale (mean=0 and variance=1) which is a requirement for the optimal performance of many machine learning algorithms. Centering and Scaling will make all features with equal weight. Basically we do this before applying PCA, to put data on the same scale.

2.4.2 Applying PCA

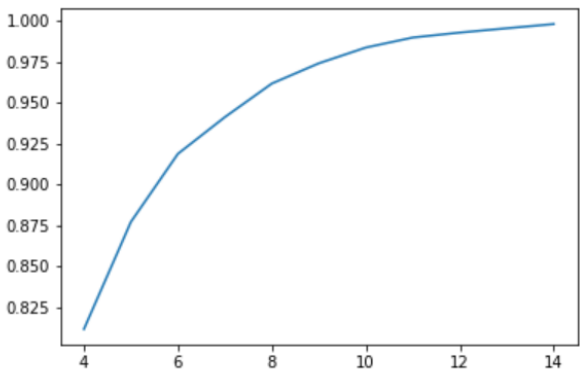
To apply PCA in python, we import PCA from sklearn.decomposition. It is used for linear dimensionality reduction using Singular Value Decomposition of the data to project it to a lower dimensional space. In R, we use prcomp function. Prcomp performs a principal components analysis

Explained Variance: It tells us how much information (variance) can be attributed to each of the principal component. This is important as while reducing the dimension, we lose some of the variance (information). By using the attribute explained_variance_ratio_, we can check the percentage of variance each principal component contains.

We had to find the best number of principal components for the dataset. pca.explained_variance_ratio_ helps us in determining this.

We tried to find the best no. of components and checked for no. of component = 4 to 15. We saw that 5 components are explaining about 87% variance so we selected 5 components. Here is a plot of no. of components vs variance.

```
In [178]: var_ratio
Out[178]: {4: 0.811544276235126,
           5: 0.8770555795291433,
           6: 0.9186492443512616,
           7: 0.9410925256030128,
           8: 0.9616114053683068,
           9: 0.9739787081990643,
           10: 0.9835896584630709,
           11: 0.9897248107341954,
           12: 0.9927550009135228,
           13: 0.9953907562385418,
           14: 0.9979616898169594}

In [179]: pd.Series(var_ratio).plot()
Out[179]: <matplotlib.axes._subplots.AxesSubplot at 0xe316748>
```



- Factor Analysis was also done to see variance explained by each component-

```
In [45]: # Factor Analysis : variance explained by each component
         pd.Series(pc_final.explained_variance_ratio_,index=['PC_'+ str(i) for i in range(5)])

Out[45]: PC_0    0.402058
         PC_1    0.180586
         PC_2    0.147294
         PC_3    0.081606
         PC_4    0.065511
         dtype: float64
```

Sum of variance explained by each principal component here comes out to almost 87%

2.5 Clustering

Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups.

Clustering in Machine Learning - It is basically a type of unsupervised learning method. An unsupervised learning method is a method in which we draw references from datasets consisting of input data without labeled responses. Generally, it is used as a process to find meaningful structure, explanatory underlying processes, generative features, and groupings inherent in a set of examples.

Clustering Algorithms:

K-means clustering algorithm – It is the simplest unsupervised learning algorithm that solves clustering problem. K-means algorithm partition n observations into k clusters where each observation belongs to the cluster with the nearest mean serving as a prototype of the cluster .

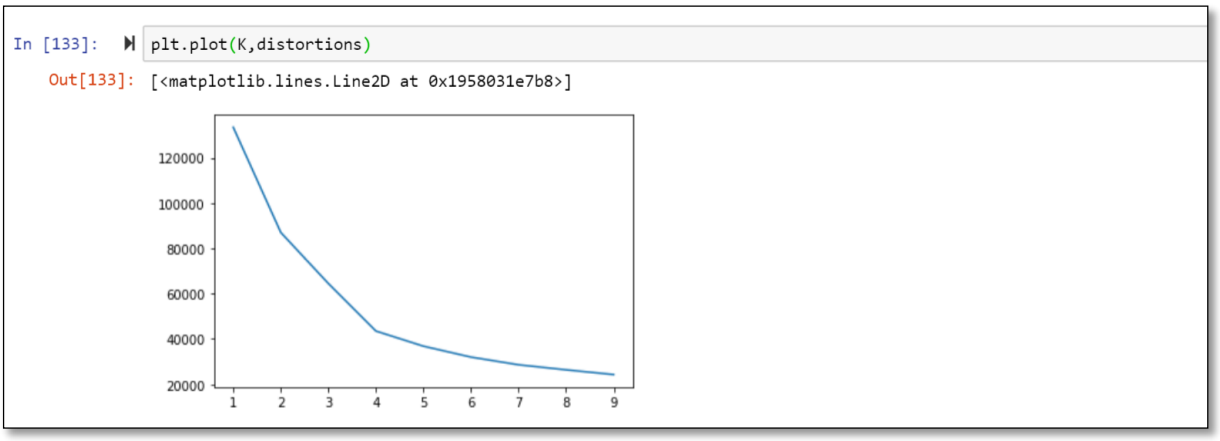
- After defining which dimensions that are going to used in Clustering, now we will use K-Means to determine how many Clusters do we need to divide Customers, which may represents their profile and hopefully we can determine what kind of tretment should be given to them.

2.5.1 Elbow method

The “elbow” method helps select the optimal number of clusters by fitting the model with a range of values for K. If the line chart resembles an arm, then the “elbow” (the point of

inflection on the curve) is a good indication that the underlying model fits best at that point. In the visualizer “elbow” will be annotated with a dashed line.

Taking a range for K from 1 to 10 for the data set, we plot K vs distortions, where distortions are defined by the sum of squared distances between each observation and its closest centroid.

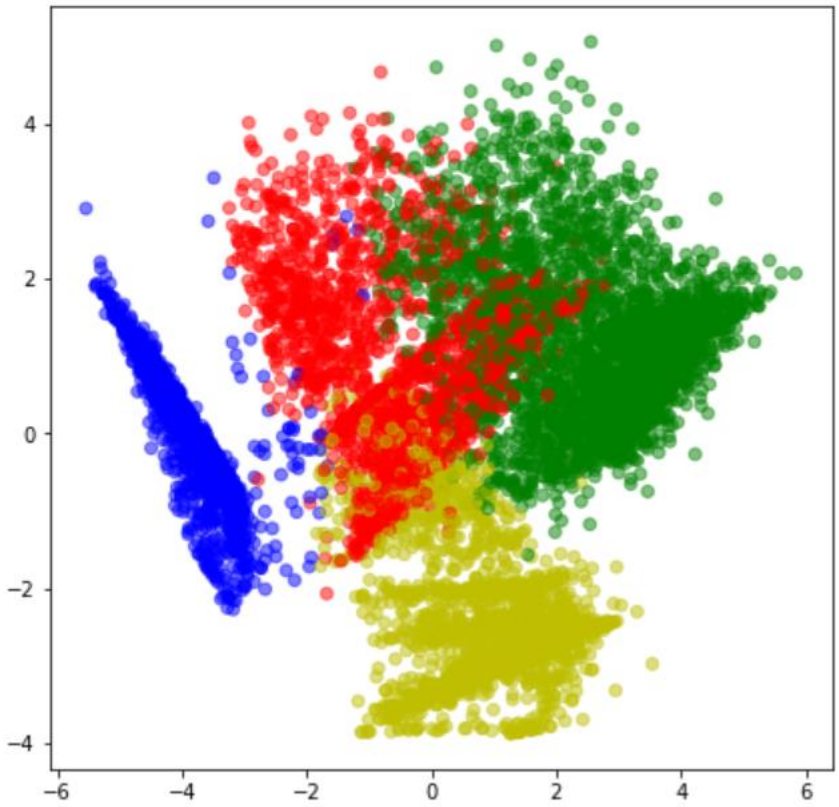


Based on the Elbow Method, we can say that the potential number of cluster (k) that may represents the customer segmentation is lies on 4-6. Actually based on the picture we can see at the 4th K, the line already steadily declined, but since on the 6th K the line is declined in quite serious numbers, we will take it along. We will test for k = 4,5 and 6

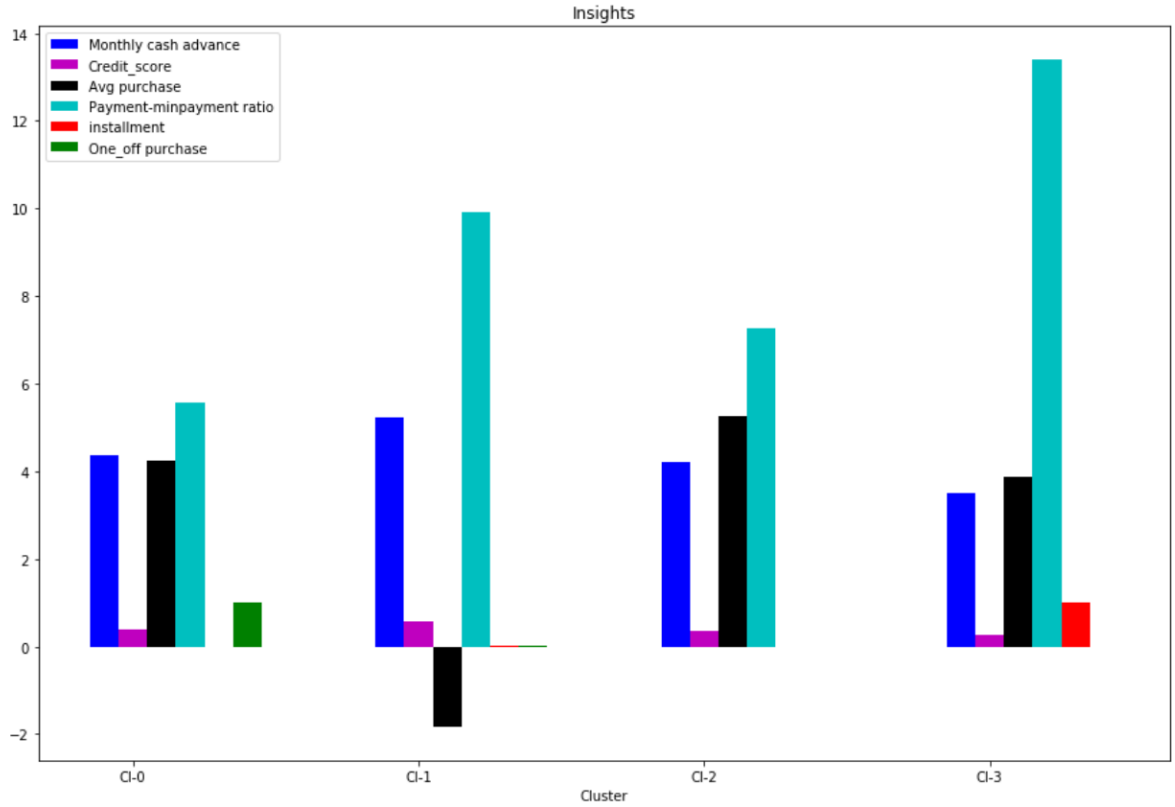
2.5.2 Applying K-means algorithm

First , the k-means algorithm was applied for k = 4 i.e., first we are tried finding behaviour with four clusters, then for 5 cluster and then for 6 clusters. The cluster obtained from each is shown below.

For k = 4



We Concatenated labels found through Kmeans with data. Since Mean value gives a good indication of the distribution of data. So we found mean value for each variable for each cluster and called it cluster_4. Then we plotted it to gain some insights about the cluster formed. This is shown below.



- Clusters are clearly distinguishing behavior within customers
- Findings through clustering is validating Insights dervied from KPI. (as shown above in Insights from KPI)
- Percentage of each cluster in the total customer base was found

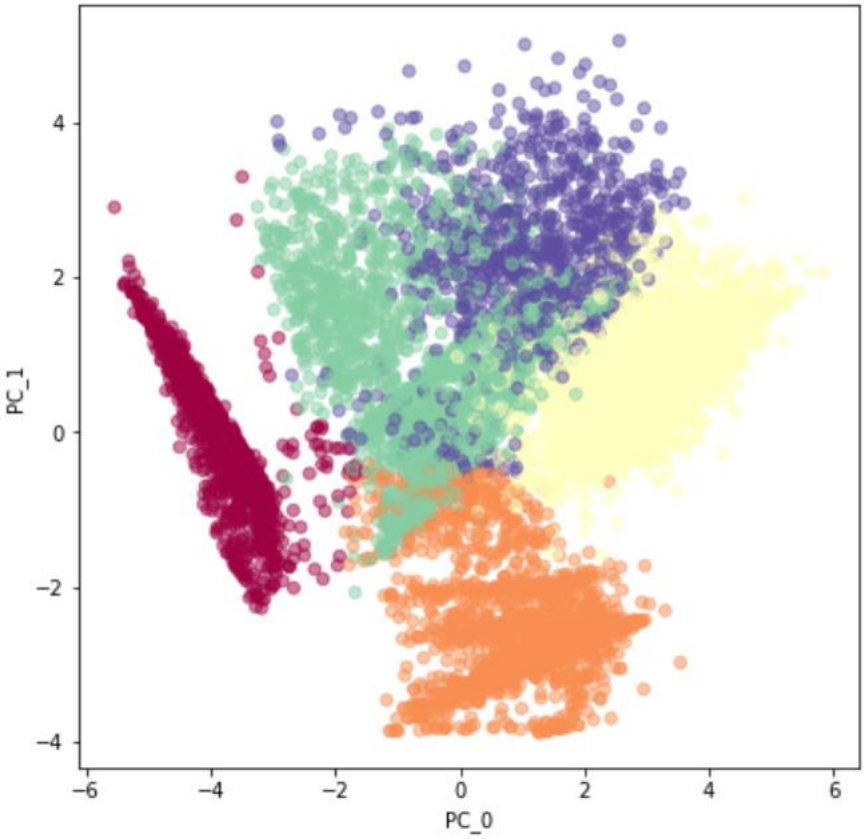
```
# Percentage of each cluster in the total customer base
s=cluster_df_4.groupby('Cluster_4').apply(lambda x: x['Cluster_4'].value_counts())
print s,'\n'

per=pd.Series((s.values.astype('float')/ cluster_df_4.shape[0])*100,name='Percentage')
print "Cluster -4 ",'\n'
print pd.concat([pd.Series(s.values,name='Size'),per],axis=1),'\n'
```

```
Cluster_4
0          0    1874
1          1    2090
2          2    2758
3          3    2228
Name: Cluster_4, dtype: int64

Cluster -4

   Size  Percentage
0  1874   20.938547
1  2090   23.351955
2  2758   30.815642
3  2228   24.893855
```

Finding behaviour with 5 Clusters:

When we took k = 5, we saw that we don't have quite distinguishable characteristics with 5 clusters

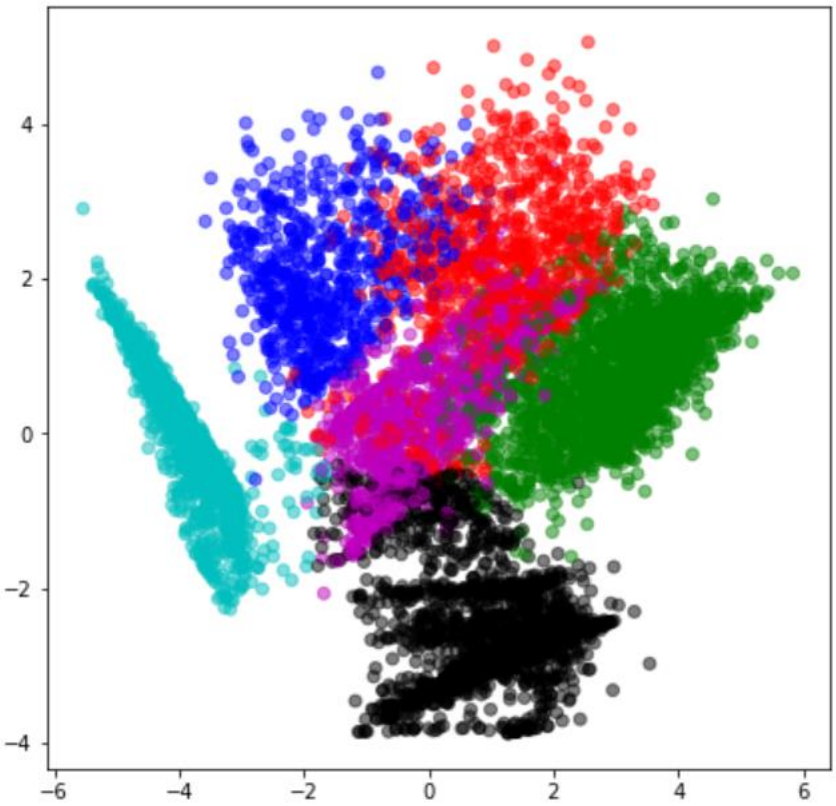
Plot:



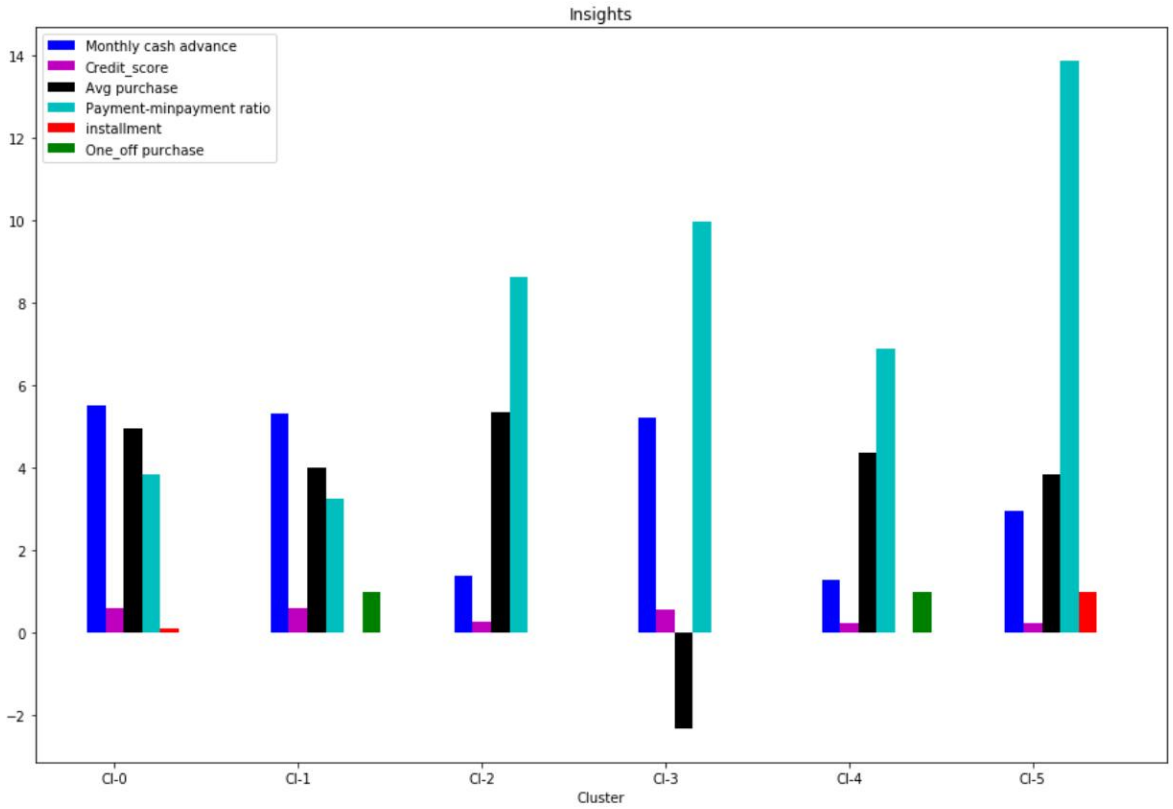
Finding Mean of features for each cluster and observing we see

- we have a group of customers (cluster 2) having highest avergae purchases but there is Cluster 4 also having highest cash advance & secong highest purchase behaviour but their type of purchases are same.
- Cluster 0 and Cluster 4 are behaving similar in terms of Credit_limit and have cash transactions is on higher side
- So we don't have quite distinguishable characteristics with 5 clusters,
- Percentage of each cluster was also found.

Finding behavior with 6 clusters:



- Here also groups are overlapping



2.5.3 Checking performance metrics for Kmeans

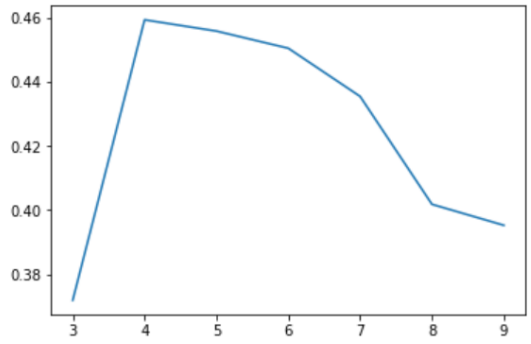
Validating performance with 2 metrics Calinski harabaz and Silhouette score

```
from sklearn.metrics import calinski_harabaz_score,silhouette_score
```

```
score={}
score_c={}
for n in range(3,10):
    km_score=KMeans(n_clusters=n)
    km_score.fit(reduced_cr)
    score_c[n]=calinski_harabaz_score(reduced_cr,km_score.labels_)
    score[n]=silhouette_score(reduced_cr,km_score.labels_)
```

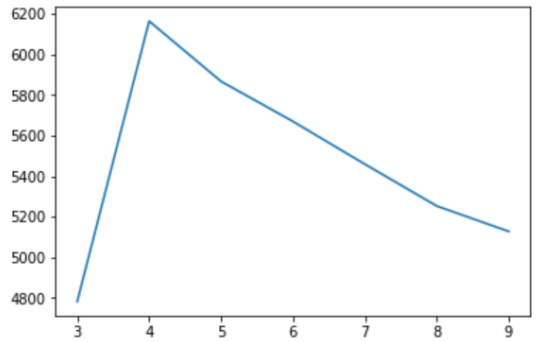
```
pd.Series(score).plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x16dbd5f8>
```




```
pd.Series(score_c).plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1b9b3940>
```



Performance metrics also suggest that K-means with 4 cluster is able to show distinguished characteristics of each cluster.

So we will have 4 segments/clusters of credit card holders. This is represented above with K=4 while applying k-means algorithm.

Result

3.1 Marketing Strategy Suggested

a. Group 2

- They are potential target customers who are paying dues and doing purchases and maintaining comparatively good credit score) -- we can increase credit limit or can lower down interest rate -- Can be given premium card /loyality cards to increase transactions

b. Group 1

- They have poor credit score and taking only cash on advance. We can target them by providing less interest rate on purchase transaction

c. Group 0

- This group is has minimum paying ratio and using card for just oneoff transactions (may be for utility bills only). This group seems to be risky group.

d. Group 3

- This group is performing best among all as cutomers are maintaining good credit score and paying dues on time. -- Giving rewards point will make them perform more purchases.

Appendix – Extra Figures

pairwise relationship of components on the data for k = 4: