PROJECT REPORT

Introduction

A credit card allows you to spend money you don't have at the time that you can then pay back at a later date. Even if you do have the money available, you can often benefit from using your credit card anyway and then paying back the balance in full before the interest properly kicks in. Credit Cards are used by a large section of the society. It’s essential that a customer segmentation must be done so as to formulate a Marketing strategy on how to increase the demand for Credit Cards with attractive financial features.

Problem Statement

This case requires trainees to develop a customer segmentation to define marketing strategy. The sample dataset summarizes the usage behavior of about 9000 active credit card holders during the last 6 months. The file is at a customer level with 18 behavioral variables.

Data

Understanding of data is the very first and important step in the process of finding solution of any business problem. Here in our case our company has provided a data set with following features, we need to go through each and every variable of it to understand and for better functioning.

- Size of Dataset Provided: - 8950 rows, 18 Columns
- Missing Values: Yes
- Outliers Presented: Yes

Below mentioned is a list of all the variable names with their meanings:

- CUST_ID- Credit card holder ID
- BALANCE- Monthly average balance (based on daily balance averages)
- BALANCE_FREQUENCY- Ratio of last 12 months with balance
- PURCHASES- Total purchase amount spent during last 12 months
- ONEOFF_PURCHASES -Total amount of one-off purchases
- INSTALLMENTS_PURCHASES- Total amount of installment purchases
- CASH_ADVANCE Total cash-advance amount
- PURCHASES_ FREQUENCY-Frequency of purchases (percentage of months with at least on purchase)
- ONEOFF_PURCHASES_FREQUENCY- Frequency of one-off-purchases
- PURCHASES_INSTALLMENTS_FREQUENCY- Frequency of installment purchases
- CASH_ADVANCE_ FREQUENCY- Cash-Advance frequency
- CASH_ADVANCE_TRX -Average amount per cash-advance transaction
- PURCHASES_TRX -Average amount per purchase transaction
- CREDIT_LIMIT- Credit limit
- PAYMENTS-Total payments (due amount paid by the customer to decrease their statement balance) in the period
- MINIMUM_PAYMENTS -Total minimum payments due in the period.
- PRC_FULL_PAYMENT- Percentage of months with full payment of the due statement balance
- TENURE- Number of months as a customer

Methodology

Pre-Processing

When we required to build a predictive model, we require to look and manipulate the data before we start modelling which includes multiple preprocessing steps such as exploring the data, cleaning the data as well as visualizing the data through graph and plots, all these steps is combined under one shed which is Exploratory Data Analysis, which includes following steps:

- Data Exploration and Cleaning
- Missing Value Analysis
- Deriving new Key Performance Indicators(KPI's)
- Outlier Treatment
- Insights from New KPI's and Data Visualisation

Applying Machine Learning(ML) Algorithms

Once all the Pre-Processing steps has been done on our data set, we will now further move to our next step which is applying ML Algorithms. ML Algorithms plays an important role to find out the good inferences from the data. Choice of models depends upon the problem statement and data set. As per our problem statement and dataset, we will try some models on our preprocessed data and post comparing the output results we will select the best suitable model for our problem. As per our data set following models need to be tested on K-Means Clustering

- Principal Component Analysis(PCA)
- Silhouette Coefficient

Checking Performance Metrics

We have to check the validating performance with 2 metrics namely

- Calinski Harabaz Score
- Silhouette Score

Data Pre Processing

Data exploration and Cleaning (Missing Values and Outliers)

The very first step which comes with any data science project is data exploration and cleaning which includes following points as per this project:

1. Drop the CUST_ID column as it is not relevant

2. As we know we have some missing values in CREDIT_LIMIT and MINIMUM_PAYMENTS, we have to remove those missing values and replace them with median.

Deriving new Key Performance Indicators(KPI)

1. Monthly average purchase and cash advance amount
2. Purchases by type (one-off, installments)
3. Average amount per purchase and cash advance transaction,
4. Limit usage (balance to credit limit ratio),
5. Payments to minimum payments ratio etc.
6. Advanced reporting: Use the derived KPIs to gain insight on the customer profiles.
7. Identification of the relationships/ affinities between services.

Creating some new variables

We created new variables from our existing variables in our Dataset

- Monthly Average Purchases

  Monthly Average Purchases = Purchases/Tenure

- Monthly Cash Advance

Monthly Cash Advance = Cash Advance/Tenure

- Purchase Type

  We combine all forms of Purchase given in our Dataset and form this new variable of Purchase Type comprising of:

  1. Both Oneoff Installment
  2. Installment
  3. None
  4. Oneoff

- Limit Usage

  Limit Usage = Balance/Credit Limit

- Payment Minpay

  Payment Minpay = Payments/Minimum Payments

Outlier Treatment

Since there are variables having extreme values, I am doing log-transformation on the dataset to remove outlier effect.

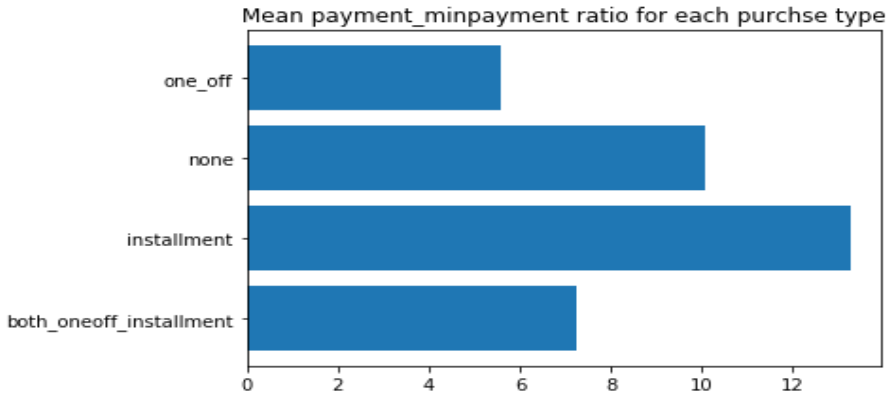
Log Transformation: This method is often used to reduce the variability of data including outlying observation.
It's often preferred when the response variable follows exponential distribution or is right-skewed**.**

Newly Extracted Variables

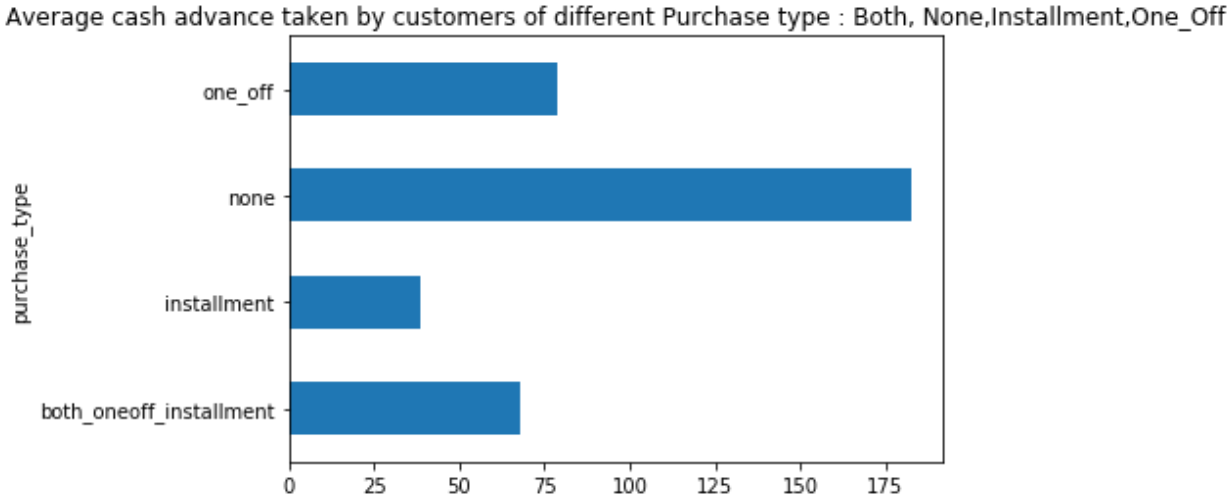
- BALANCE
- BALANCE_FREQUENCY
- PURCHASES
- ONEOFF_PURCHASE
- INSTALLMENTS_PURCHASES
- CASH_ADVANCE
- PURCHASES_FREQUENCY
- ONEOFF_PURCHASES_FREQUENCY
- PURCHASES_INSTALLMENTS_FREQUENCY
- CASH_ADVANCE_FREQUENCY
- CASH_ADVANCE_TRX
- PURCHASES_TRX
- CREDIT_LIMIT
- PAYMENTS
- MINIMUM_PAYMENTS
- PRC_FULL_PAYMENT
- TENURE
- Monthly_avg_purchase
- Monthly_cash_advance
- limit_usage
- payment_minpay

Insights from New KPI's and Data Visualization

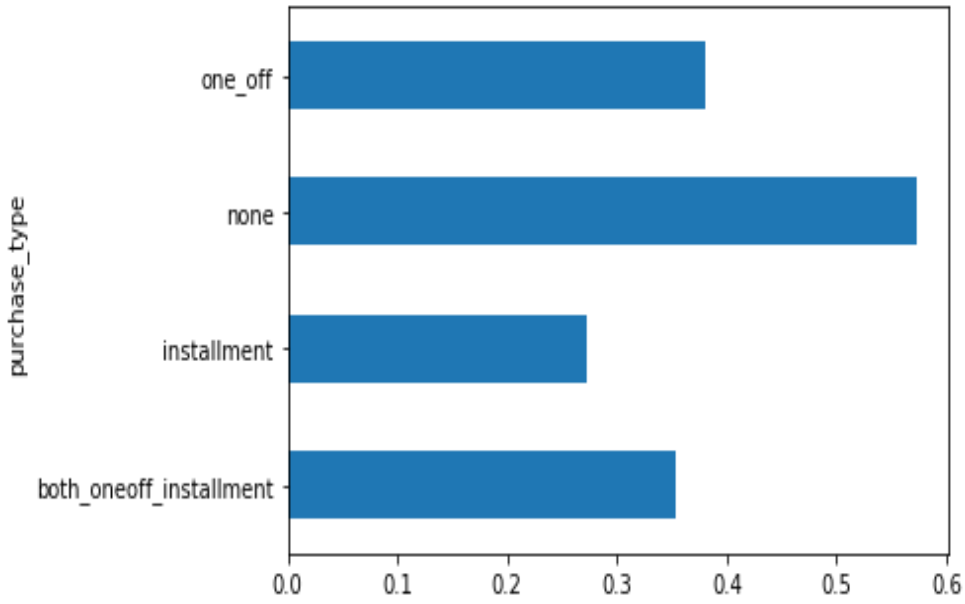
We will derive the relations between the indicators by means of visualization plots:



Customers with installment purchases are paying dues



Customers who don't do either one-off or installment purchases take more cash on advance



Customers with installment purchases have good credit score

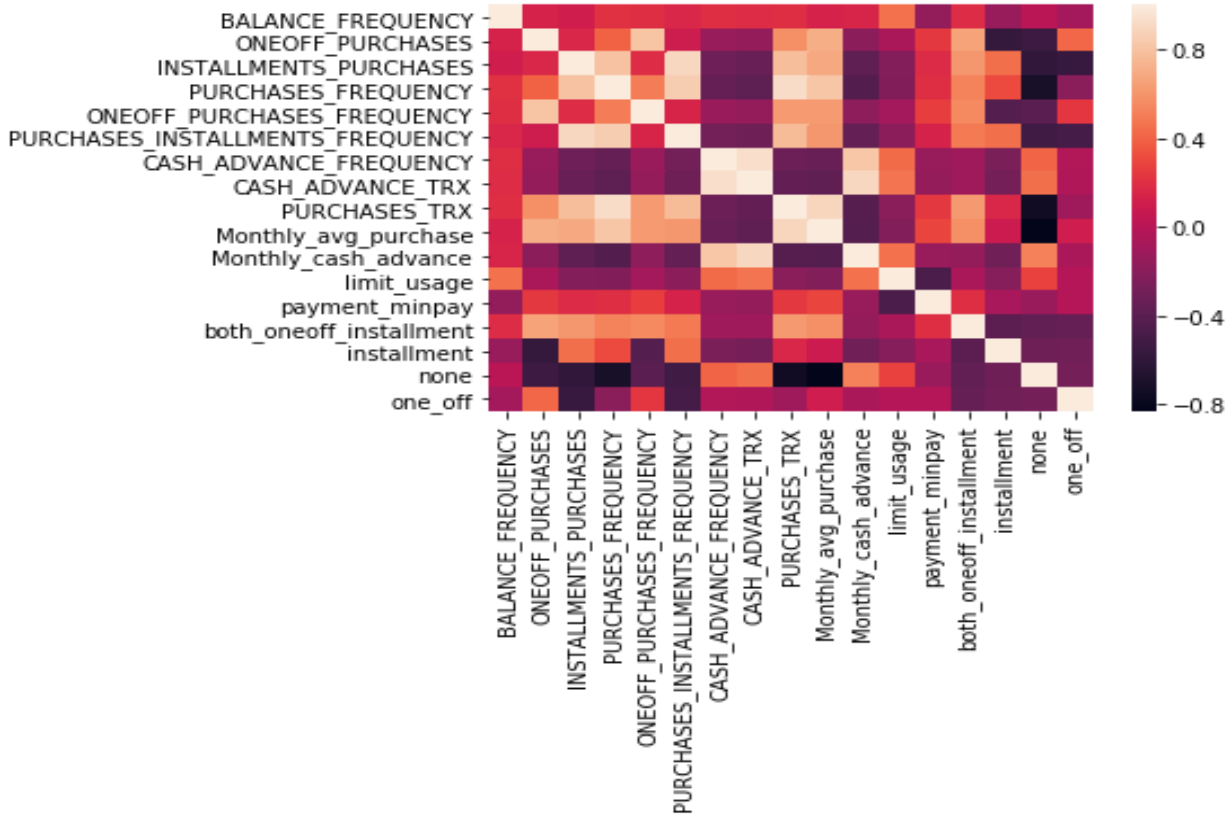
Applying Machine Learning Algorithms

The first approach would be to create a dummy with the Purchase Type variable that collectively includes:

- Both One Off Installment
- Installment
- None
- One off

After creating the dummy needs to be merged with the original Data Frame. One needs to check whether there are any missing values or not.

As there are no missing values, the next step would be to see whether there is any correlation or not by plotting the correlation graph:



Standardize the data in order to avoid effect of scale on our result. Centering and Scaling will make all features with equal weight.

Standardization

Standardization is about making sure that data is internally consistent; that is, each data type has the same content and format. Standardized values are useful for tracking data that isn't easy to compare otherwise.

Standardization typically rescales data to have a mean of 0 and a standard deviation of 1.

In this dataset , Standardization must be done in order to avoid effect on scale on our result. Scaling will make all features of equal weight.

Standardization

In [61]:
```
from sklearn.preprocessing import  StandardScaler
```

In [62]:
```
sc=StandardScaler()

```

In [63]:
```
cr_scaled=sc.fit_transform(cr_dummy)

```

In [64]:
```
cr_scaled
```

Out[64]:
```
array([[-0.14875746, -0.98708958,  0.39447984, ...,  1.72051649,
        -0.54369045, -0.514625  ],
       [ 0.17961568, -0.98708958, -1.08745376, ..., -0.58122082,
         1.83928189, -0.514625  ],
       [ 0.49271003,  1.06202168, -1.08745376, ..., -0.58122082,
        -0.54369045,  1.94316249],
       ...,
       [-0.09290575, -0.98708958,  0.52779444, ...,  1.72051649,
        -0.54369045, -0.514625  ],
       [-0.09290575, -0.98708958, -1.08745376, ..., -0.58122082,
         1.83928189, -0.514625  ],
       [-0.73437135,  1.16861854, -1.08745376, ..., -0.58122082,
        -0.54369045,  1.94316249]])
```

Principal Component Analysis(PCA)

Principal component analysis (PCA) is a statistical procedure that uses an transformation to convert a set of observations of possibly correlated variables (entities each of which takes on various numerical values) into a set of values of linearly uncorrelated variables called principal components. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors (each being a linear combination of the variables and containing n observations) are an uncorrelated set. PCA is sensitive to the relative scaling of the original variables.

PCA is mostly used as a tool in exploratory data analysis and for making predictive models. It is often used to visualize genetic distance and relatedness between populations.

Data Observations

In this report we are clustering the customers as per their spending methods so as to prepare a marketing strategy for the upcoming business year. So here we have a data set of 8950 observations with 17 variables.

The dataset has 17 variables we have to take 17 as our n component.

Find the variance of scaled Data, and after checking the variance it has been found that there are 6 components that has 90% Variance.

Now taking the components we have to use dimensionality reduction so that we don't overfit out variables. There are 5 variables for clustering purposes

	0	1	2	3	4	5
0	-0.242841	-2.759668	0.343061	-0.417359	-0.007100	0.019755
1	-3.975652	0.144625	-0.542989	1.023832	-0.428929	-0.572463
2	1.287396	1.508938	2.709966	-1.892252	0.010809	-0.599932
3	-1.047613	0.673103	2.501794	-1.306784	0.761348	1.408986
4	-1.451586	-0.176336	2.286074	-1.624896	-0.561969	-0.675214

New Variables from Dummy set

- BALANCE_FREQUENCY
- ONEOFF_PURCHASES
- INSTALLMENTS_PURCHASES
- PURCHASES_FREQUENCY
- ONEOFF_PURCHASES_FREQUENCY
- PURCHASES_INSTALLMENTS_FREQUENCY
- CASH_ADVANCE_FREQUENCY
- CASH_ADVANCE_TRX
- PURCHASES_TRX
- Monthly_avg_purchase
- Monthly_cash_advance
- limit_usage
- payment_minpay
- both_oneoff_installment
- installment
- none
- one_off

Eigen Vector

The eigenvectors and eigenvalues of a covariance (or correlation) matrix represent the core of a PCA.The eigenvectors (principal components) determine the directions of the new feature space, and the eigenva lues determine their magnitude. In other words, the eigenvalues explain the variance of the data along t he new feature axes. The Eigen Vectors of the columns are given below:

	PC_0	PC_1	PC_2	PC_3	PC_4	PC_5
BALANCE_FREQUENCY	0.029707	0.240072	-0.263140	-0.353549	-0.228681	-0.693816
ONEOFF_PURCHASES	0.214107	0.406078	0.239165	0.001520	-0.023197	0.129094
INSTALLMENTS_PURCHASES	0.312051	-0.098404	-0.315625	0.087983	-0.002181	0.115223
PURCHASES_FREQUENCY	0.345823	0.015813	-0.162843	-0.074617	0.115948	-0.081879
ONEOFF_PURCHASES_FREQUENCY	0.214702	0.362208	0.163222	0.036303	-0.051279	-0.097299
PURCHASES_INSTALLMENTS_FREQUENCY	0.295451	-0.112002	-0.330029	0.023502	0.025871	0.006731
CASH_ADVANCE_FREQUENCY	-0.214336	0.286074	-0.278586	0.096353	0.360132	0.066589
CASH_ADVANCE_TRX	-0.229393	0.291556	-0.285089	0.103484	0.332753	0.082307
PURCHASES_TRX	0.355503	0.106625	-0.102743	-0.054296	0.104971	-0.009402
Monthly_avg_purchase	0.345992	0.141635	0.023986	-0.079373	0.194147	0.015878
Monthly_cash_advance	-0.243861	0.264318	-0.257427	0.135292	0.268026	0.058258
limit_usage	-0.146302	0.235710	-0.251278	-0.431682	-0.181885	0.024298
payment_minpay	0.119632	0.021328	0.136357	0.591561	0.215446	-0.572467
both_oneoff_installment	0.241392	0.273676	-0.131935	0.254710	-0.340849	0.294708
installment	0.082209	-0.443375	-0.208683	-0.190829	0.353821	-0.086087
none	-0.310283	-0.005214	-0.096911	0.245104	-0.342222	-0.176809
one_off	-0.042138	0.167737	0.472749	-0.338549	0.362585	-0.060698

Factor Analysis

Factor analysis is a technique that is used to reduce a large number of variables into fewer numbers of factors. This technique extracts maximum common variance from all variables and puts them into a common score. As an index of all variables, we can use this score for further analysis. The outputs of the Factor Analysis is given below:

PC_0 0.402058
PC_1 0.180586
PC_2 0.147294
PC_3 0.081606
PC_4 0.065511
PC_5 0.041594

K-Means Clustering

K-means clustering is a type of unsupervised learning, which is used with unlabeled dataset. The goal of this algorithm is to find K groups in the data. The algorithm works iteratively to assign each data point to one of K groups based on the features that are provided. Data points are clustered based on feature similarity. The results of the K-means clustering algorithm are:

- The centroids of the K clusters, which can be used to label new data
- Labels for the training data (each data point is assigned to a single cluster)

K-means works by defining spherical clusters that are separable in a way so that the mean value converges towards the cluster center. Because of this, K-Means may underperform sometimes.

In this Dataset based on the type of purchases made by customers and their distinctive behavior exhibited based on the purchase type (as visualized above in Insights from KPI), we have to start off with 4 clusters.

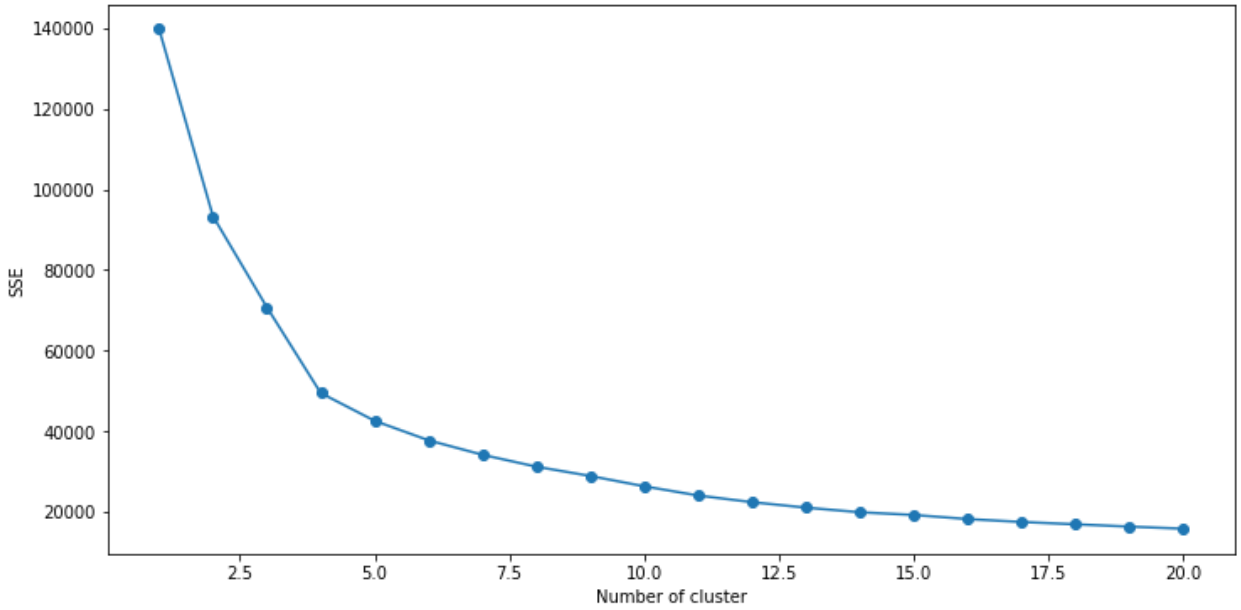
As we do not have known k value so we will find the K. To do that we need to take a cluster range between 1 and 21.

Elbow Criterion Method

As we do not know the K value we will use Elbow Criterion method between the range 1 to 21. The idea behind elbow method is to run k-means clustering on a given dataset for a range of values of k (e.g k=1 to 10), for each value of k, calculate sum of squared errors (SSE).

Calculate the mean distance between data points and their cluster centroid. Increasing the number of clusters(K) will always reduce the distance to data points, thus decrease this metric, to the extreme of reaching zero when K is as same as the number of data points. So the goal is to choose a small value of k that still has a low SSE.

We run the algorithm for different values of K(say K = 10 to 1) and plot the K values against SSE(Sum of Squared Errors). And select the value of K for the elbow point.



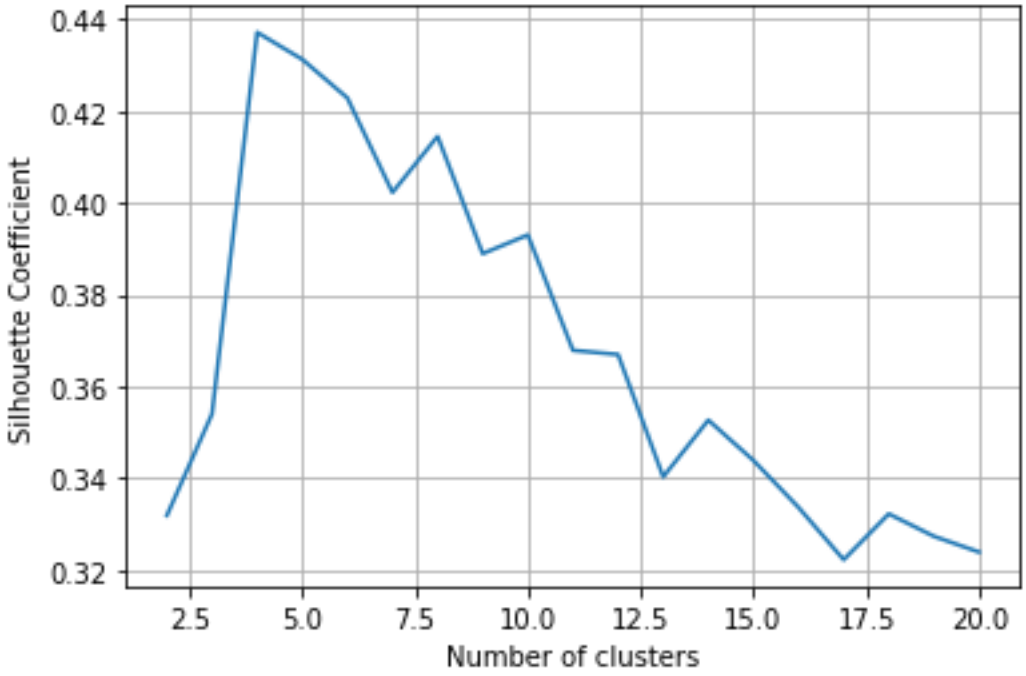
Silhouette Coefficient Method

A higher Silhouette Coefficient score relates to a model with better-defined clusters. The Silhouette Coefficient is defined for each sample and is composed of two scores:

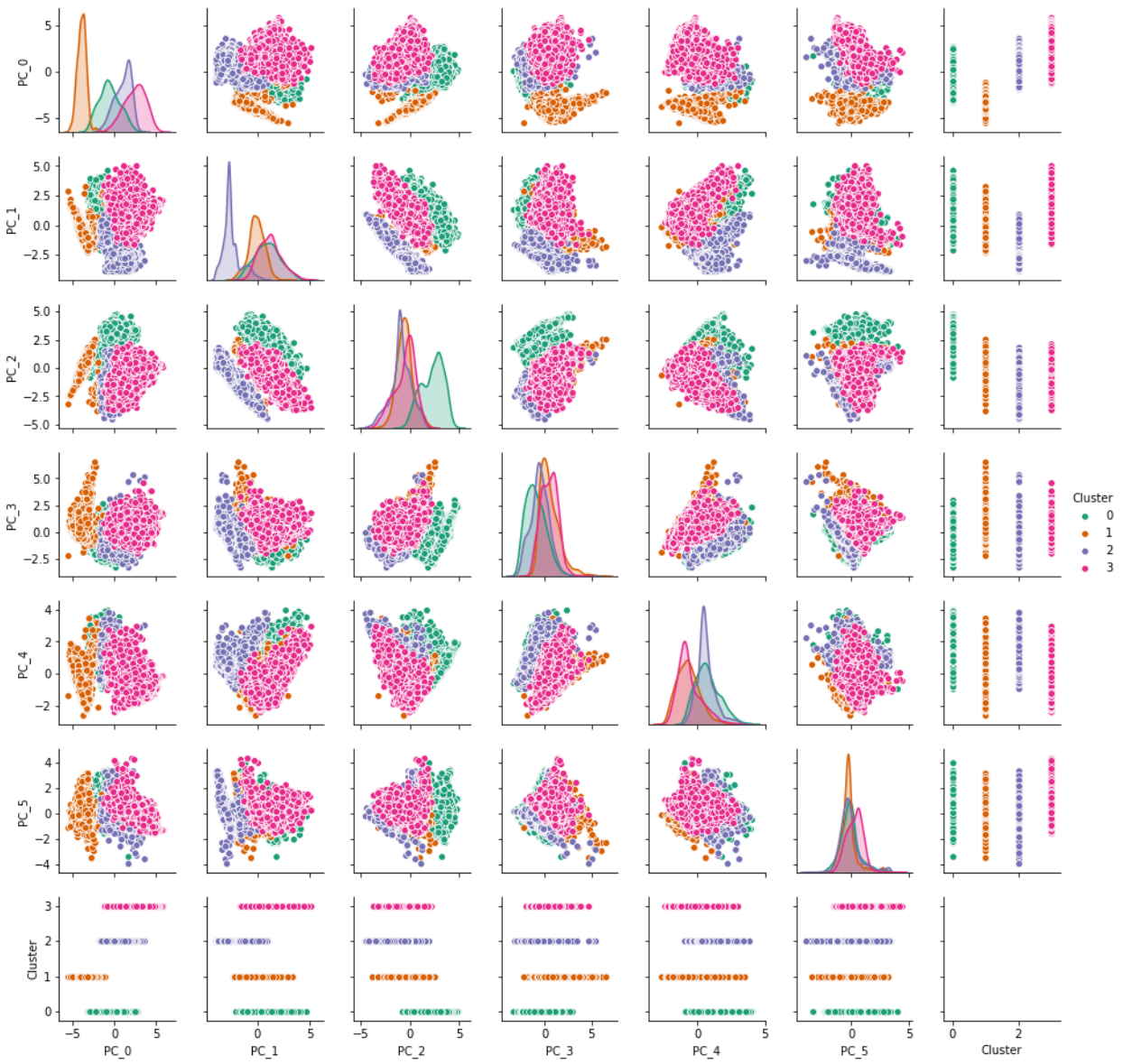
- The mean distance between a sample and all other points in the same class.
- The mean distance between a sample and all other points in the next nearest cluster.
- The Silhouette Coefficient is for a single sample is then given as:

$$s = \frac{b - a}{\max(a, b)}$$

Here in this Dataset we will do a Silhouette Coefficient between range 1 and 21.



We will use Pair plot to provide us all graph in a single frame. At the same time we have to add the Cluster column as well:



It shows that first two components are able to identify clusters

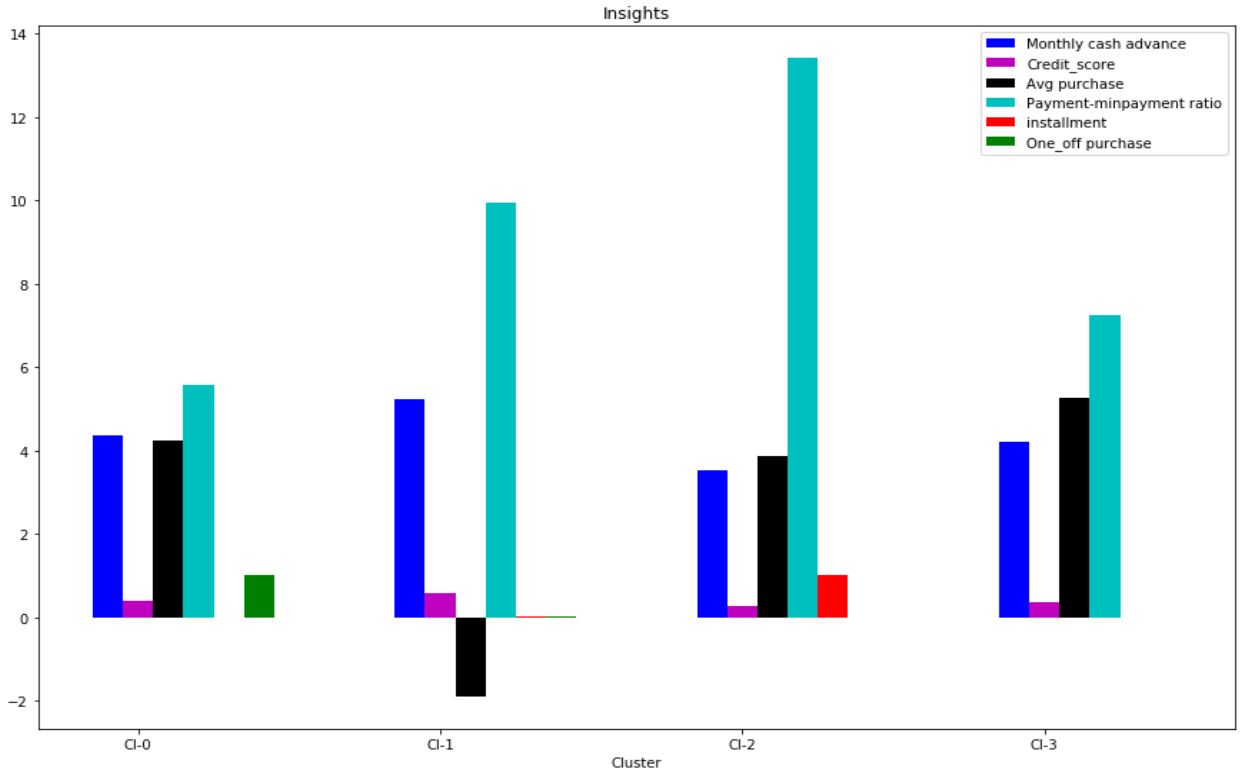
Now we have done here with principle component now we need to come bring our original data frame and we will merge the cluster with them.

To interpret result we need to use our data frame

Now we will find the Mean value of each variable of each cluster:

Cluster_4	0	1	2	3
PURCHASES_TRX	7.127341	0.043582	12.062050	33.013723
Monthly_avg_purchase	69.875917	0.148297	47.626256	193.008043
Monthly_cash_advance	78.098613	186.281319	33.550080	67.466910
limit_usage	0.379761	0.576076	0.264745	0.353591
CASH_ADVANCE_TRX	2.881220	6.540230	1.021133	2.804261
payment_minpay	5.573672	9.936617	13.422420	7.245651
both_oneoff_installment	0.000535	0.001916	0.000000	1.000000
installment	0.000000	0.017241	1.000000	0.000000
one_off	0.999465	0.002874	0.000000	0.000000
none	0.000000	0.977969	0.000000	0.000000
CREDIT_LIMIT	4519.708481	4055.156450	3338.270406	5736.732730

Graphical Representation:



Clusters are clearly distinguishing behavior within customers

Now out of 8950 observations let us cluster the customers and find out their percentages

```
# 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))
```

```
Cluster_4
0          0    1869
1          1    2088
2          2    2224
3          3    2769
Name: Cluster_4, dtype: int64
Cluster -4
   Size  Percentage
0  1869   20.882682
1  2088   23.329609
2  2224   24.849162
3  2769   30.938547
```

Group	Number	Percentage
0	1869	21%
1	2088	23%
2	2224	25%
3	2769	31%

Now let’s check the pattern of 5 Cluster

We would find the mean and derive new insights

Cluster_5	0	1	2	3	4
PURCHASES_TRX	7.096670	34.587759	0.032196	27.703746	11.905537
Monthly_avg_purchase	68.917645	210.536468	0.086126	141.584086	47.369817
Monthly_cash_advance	74.517541	4.040708	185.038534	249.942101	20.636870
limit_usage	0.377959	0.258931	0.576110	0.600096	0.250011
CASH_ADVANCE_TRX	2.697637	0.152757	6.448823	10.384790	0.550489
payment_minpay	5.562287	8.675499	9.963172	3.651686	13.783426
both_oneoff_installment	0.002148	1.000000	0.000000	0.900114	0.000000
installment	0.000000	0.000000	0.015858	0.088536	1.000000
one_off	0.997852	0.000000	0.002883	0.011351	0.000000
none	0.000000	0.000000	0.981259	0.000000	0.000000
CREDIT_LIMIT	4497.951209	5722.970627	4046.692295	5873.041998	3228.949923

Conclusion With 5 clusters :

We have a group of customers (cluster 1) having highest avergae purchases but there is Cluster 3 also having highest cash advance & secong highest purchase behaviour but their type of purchases are same.

Cluster 0 and Cluster 2 are behaving similar in terms of Credit_limit and Cluster 2 and Cluster 3 have cash transactions is on higher side

There is no distinguishable characters with 5 clusters

Observation with 6 clusters

Like earlier we would derive insights by looking at the mean value of each variables of the clusters

Cluster_6	0	1	2	3	4	5
PURCHASES_TRX	34.663789	5.967143	0.030347	11.905537	7.760575	27.919908
Monthly_avg_purchase	211.196582	54.091602	0.088891	47.369817	78.585295	140.374727
Monthly_cash_advance	4.027720	205.502536	184.829434	20.636870	3.603272	242.856971
limit_usage	0.258206	0.605930	0.575724	0.250011	0.245772	0.600654
CASH_ADVANCE_TRX	0.150838	7.642857	6.434971	0.550489	0.125212	10.000000
payment_minpay	8.702974	3.257979	9.976487	13.783426	6.911822	3.616973
both_oneoff_installment	1.000000	0.000000	0.000000	0.000000	0.006768	0.911899
installment	0.000000	0.000000	0.016378	1.000000	0.000000	0.088101
one_off	0.000000	1.000000	0.000000	0.000000	0.993232	0.000000
none	0.000000	0.000000	0.983622	0.000000	0.000000	0.000000
CREDIT_LIMIT	5735.293514	4577.649351	4047.527296	3228.949923	4471.701020	5834.610984

Conclusion with 6 clusters

Nothing much could be inference from 6 clusters

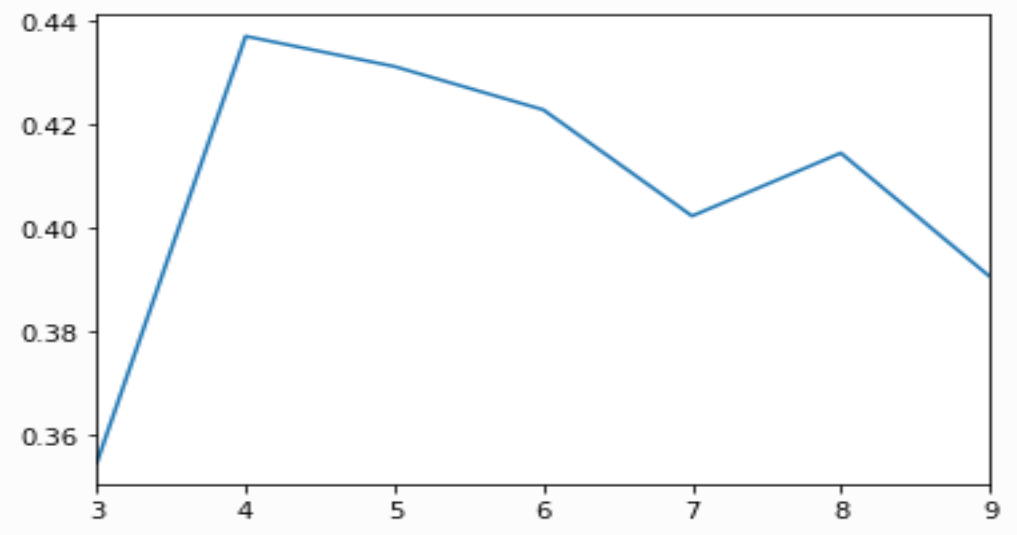
We have a group of customers (cluster 0) having highest average purchases but there is Cluster 5 also having highest cash advance & second highest purchase behaviour but their type of purchases are same.

Cluster 0 and Cluster 5 are behaving similar in terms of Credit_limit and Cluster 1 and Cluster 2 have cash transactions is on higher side

Checking Performance Matrix

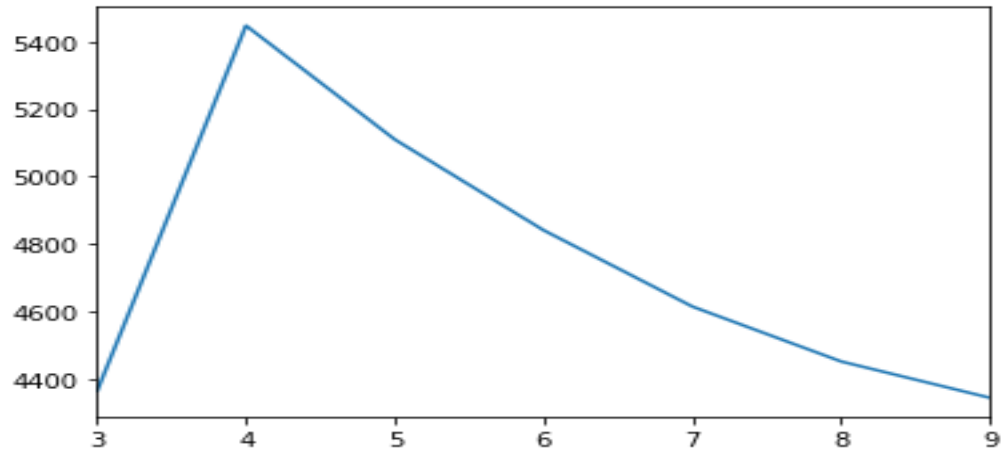
Calinski-Harabasz

Calinski-Harabasz is a kind of estimate that can help us chooses the proper clustering number before performing the algorithm.



Silhouette Score

Silhouette Score refers to a method of interpretation and validation of consistency within clusters of data. The technique provides a succinct graphical representation of how well each object has been classified.



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

Insights with 4 clusters

1. Cluster 3 is the group of customers who have highest Monthly_avg purchases and doing both one_off installment purchases, have comparatively good credit score. *This group is about* 31% of the total customer base.

2. Cluster 1 is taking maximum advance_cash and is paying comparatively less minimum payment and poor credit_score & doing no purchase transaction. This group is about 23% of the total customer base.

3. Cluster 0 customers are doing maximum One_Off transactions and least payment ratio and credit_score on lower side This group is about 21% of the total customer base.

4. Cluster 2 customers have maximum credit score and are paying dues and are doing maximum installment purchases. This group is about 25% of the total customer base

The End