Project: Credit Card Segmentation

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.

Number of attributes:

● 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

● AVERAGE\_PURCHASE\_TRX Average amount per purchase transaction

● 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

**Exploratory Data Analysis:**

In exploring the data, we have

• CUST\_ID variable which was an index to data doesn’t carry much information but later required for analysis so stored in one variable.

**Missing Value Analysis:**

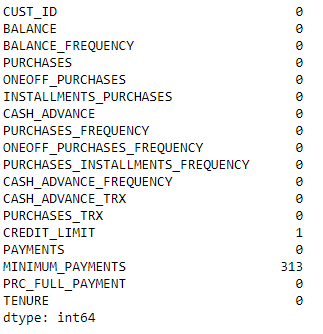
Missing value analysis is done to check is there any missing value present in given

dataset. There missing values can be treated by replacing it by using several

imputation techniques like mean, median and KNN imputation for continuous

variable and mode for categorical variable. These techniques are used to impute

missing value.



I found two variables with missing values present in them out of which one variable had 1 value missing and other had 313 values missing.

I treated this with taking the median value of the given variable.

**The reason behind going with median rather than mean was that it should not have any effect of outliers if there were present in the given variable.**

**Feature Creation**:

In this clustering analysis for profiling, we need some KPI’s key performing indices so that they will help in developing profiling of customers and mainly helps in to select the suitable marketing strategy as from KPI’s we can know customer behaviors The KPIs are

1. Monthly average purchases

2. Monthly cash advance

3. Limit usage

4. Payment to min payment ratio

5. Purchase type

Purchase type: These variables was created based on the instalment and one-off

purchases we categorized data into 4 types using logic as follows:

dat1[(dat1['ONEOFF\_PURCHASES']==0) &

(dat1['INSTALLMENTS\_PURCHASES']==0)].shape

dat1[(dat1['ONEOFF\_PURCHASES']>0) &

(dat1['INSTALLMENTS\_PURCHASES']>0)].shape

dat1[(dat1['ONEOFF\_PURCHASES']>0) &

(dat1['INSTALLMENTS\_PURCHASES']==0)].shape

dat1[(dat1['ONEOFF\_PURCHASES']==0) &

(dat1['INSTALLMENTS\_PURCHASES']>0)].shape

So purchase type variable defines how the data categorized as only one off , only instalment , both one off instalment and none After categorizing we used that variable in analysis by converting it in the form of dummies

cr\_pre['purchase\_type']=dat1.loc[:,'purchase\_type'] pd.get\_dummies(cr\_pre['purchase\_type']).head()

**Outlier Analysis:**

In data there are many extreme values and even the data was in skewed format so to treat this kind of data we use logarithm transformation. It’s a kind of Extreme value analysis.

# log tranformation cr\_log=dat1.drop(['purchase\_type'],axis=1).applymap(lambda x: np.log(x+1))

**Feature Selection**

Feature selection analysis is done to Select subsets of relevant features (variables, predictors) to be in model construction. As all variable are continuous so we can only go for correlation check. As chisquare test is only for categorical variable.

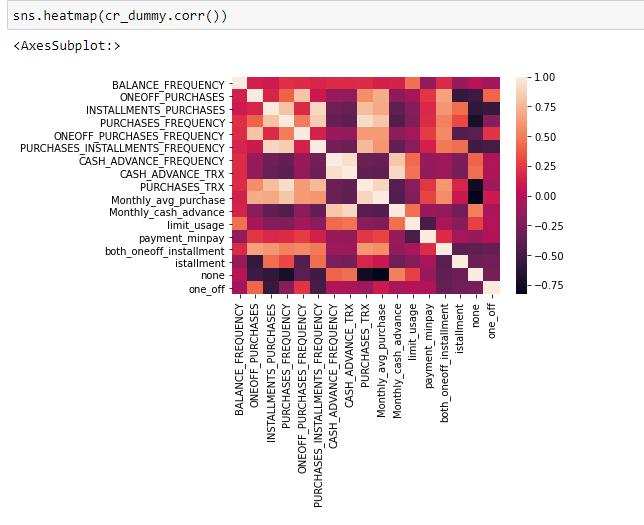


Figure show a correlation plot for all numeric variable present in dataset in above correlation plot the two independent variables [temp, atemp] are highly correlated which are carrying same weightage of data this will leads to multicollinearity problem. They both has correlation value near to 0.99. So, I decided to remove atemp variable from dataset.

As we can see there are more number of variable which are correlated to each other so i consider PCA analysis for variable reduction

PCA is widely used to simplify high-dimensional datasets to lower dimensional ones. It lower number of variables, that are not correlated, without losing information contained in the dataset

To reduce the dimensionality, the number of components are lower than the original columns Optimal ‘k’:

- Top ‘k’ components capture what proportion of the variance?

-We look at a scree plot, similar to the plot we used to choose number of clusters in k-means clustering

-Against the #principal components, we look at the cumulative proportion of variance captured - Choose k after which incremental benefit negligible

- Common choices: 80%, 90%, 95% variance

from sklearn.decomposition import PCA

pca = PCA(svd\_solver='randomized', random\_state=42)

#Performing the PCA pca.fit(dat2)

#Let's check the variance ratios pca.explained\_variance\_ratio\_ It better to prefer n number of components which covers explained variance at least 85- 95 percentage

#Plotting the scree plot

%matplotlib inline

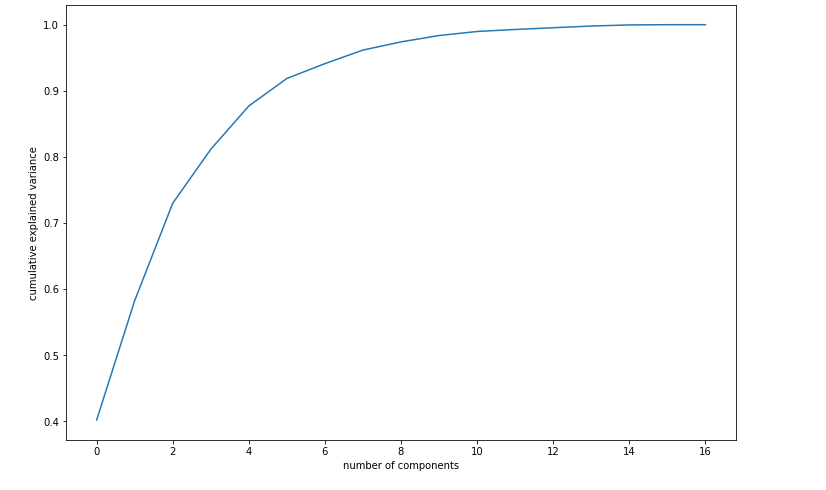
fig = plt.figure(figsize = (12,8))

plt.plot(np.cumsum(pca.explained\_variance\_ratio\_))

plt.xlabel('number of components')

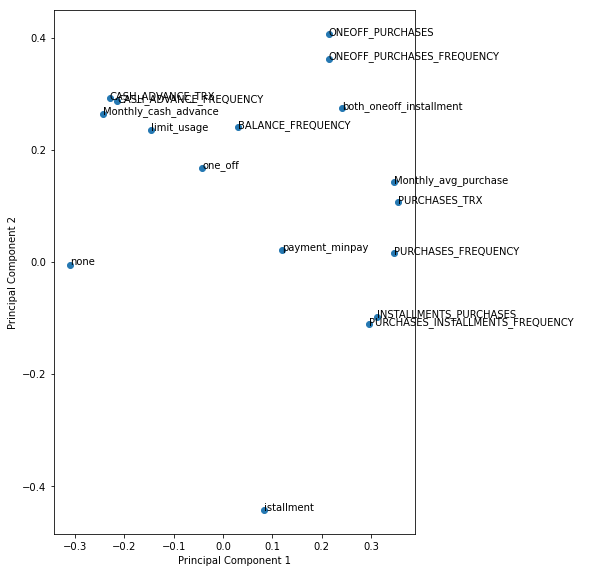
plt.ylabel('cumulative explained variance')

plt.show()



As from above scree plot and cumulative summation of variance of each PC component we get to know that the 5 P components are explaining 87% of variance

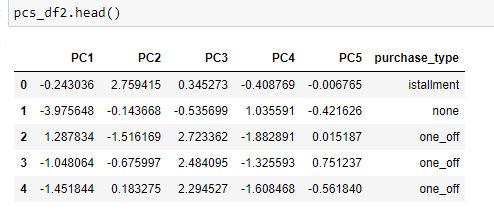
By plotting graph between PC1 and PC2 components we can see how the features are loaded



#Finally let's go ahead and do dimenstionality reduction using the five Principal Components from sklearn.decomposition import IncrementalPCA

pca\_final = IncrementalPCA(n\_components=5)

From the variance we get to know 5 PC’s are enough now let’s we implement dimensionality reduction using principal components



Here we able to reduce the higher dimensionality of data 18 variables to lower dimensionality of data 5 dimensions without losing much information.

**Feature Scaling**

Feature scaling includes two functions normalization and standardization. It is done reduce unwanted variation either within or between variables and to bring all of the variables into proportion with one another.

**Standardrizing data**

We use standardizing of data when the data was normally distributed The main purpose of standardizing was to make the variables unit less.

**Clustering Modelling**

**Model Implementation**

The main objective of this case was to develop a customer segmentation to define marketing strategy. Here, we can apply several clustering models. After applying many clustering models, we will do profiling of customers based on their purchasing behavior. The two clustering models that I decided to apply are as follows.

1. K-means clustering

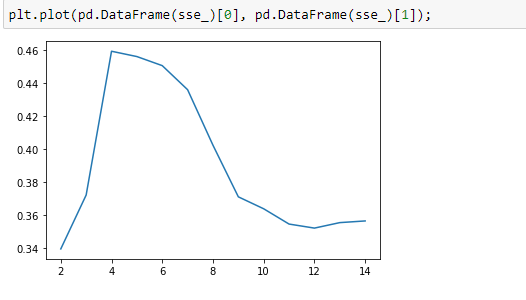
2. Hierarchical clustering

**K-means clustering**

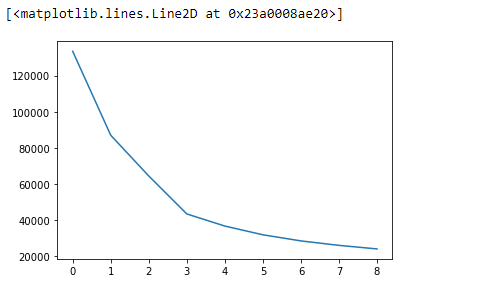
The K-Means algorithm uses the concept of the centroid to create K clusters. In simple terms, a centroid of n points on an x-y plane is another point having its own x and y coordinates and is often referred to as the geometric center of the n points.

For example, consider three points having coordinates (x1, y1), (x2, y2) and (x3, y3). The centroid of these three points is the average of the x and y coordinates of the three points, i.e. (x1 + x2 + x3 / 3, y1 + y2 + y3 / 3).

Calculating Hopkins statistics, it’s a kind of hypothesis testing and was a way to measuring the central tendency of a data set. After checking Hopkins, we got value as 0.93 as it is greater than 0.5 so we can conclude that the data has adequate data for clustering purpose. After calculating silhouette analysis score, we plotted a graph which it was suggesting us to prefer 4 clusters.



Even elbow curve suggested us to prefer cluster range from 3 to 5. Here also we're seeing a distinct bend at around 4 clusters. Hence it seems a good K to choose. From three kind of estimations we choose k value as 4.



#Let's perform K means using K=4 model\_clus2 = KMeans(n\_clusters = 4, max\_iter=50,random\_state = 50) model\_clus2.fit(dat3\_1)

1 2757

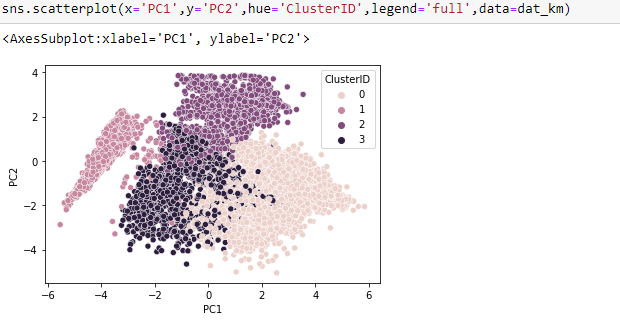
2 2229

0 2090

3 1874

Name: ClusterID, dtype: int64

Plotting the graph after creating clusters is as follows. sns.scatterplot(x='PC1',y='PC2',hue='ClusterID',legend='full',data=dat\_km)



From above graph we can see how accurately it clustered and we didn’t find any ambiguity to identify cluster of each observation.

**Hierarchical clustering**

Hierarchical clustering is type of clustering analysis that groups the similar objects into groups called clusters. Finally, it creates n number of clusters where each cluster is distinct to each other and object in each cluster are broadly similar to each other



Above Figure represents the clusters in colour format as we can see it formed four clusters represented in different colours. clusterCut = pd.Series(cut\_tree(mergings, n\_clusters = 4).reshape(-1,)) The above code assigns cluster ids to each observation.

Random forest functions in below way

1: Draws a bootstrap sample from training data.

2: For each sample grow a decision tree and at each node of the tree

a. Randomly draws a subset of n-try variable and p total of features that are available

b. Picks the best variable and best split from the subset of n-try variable

c. Continues until the tree is fully grown.

Code in R and Python to create Random Forest model R-script:

RF\_model=randomForest(cnt~., data=train,importance=TRUE,ntree=200) predictions\_RF=predict(RF\_model,test[,-12]) Python: RFmodel = RandomForestRegressor(n\_estimators = 200).fit(train.iloc[:,0:11], train.iloc[:,11]) RF\_Predictions = RFmodel.predict(test.iloc[:,0:11])

**Conclusion**:

**Analysis of clusters**

Insights with 4 Clusters

1: Cluster 3 customers are doing maximum One\_Off transactions and has least payment ratio amongst all the cluster.

2 : Cluster 0 is the group of customers who have highest Monthly cash advance and doing both installment as well as one\_off purchases, have comparatively good credit score but have poor average purchase score.

3: Cluster 1 customers have maximum Average Purchase and good Monthly cash advance but this cluster doesn't do installment or one\_off purchases.

4 : Cluster 2 is doing maximum installment, has maximum payment too min\_payment ratio and doesn't do one-off purchases

**Marketing Strategy**

**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.**