PGP - Data Science and Business Analysis - Jan’ 21

**Data Mining Project Report**

Sarang Manohar

horizontal line

# 

# Introduction

This project report is in response to the Data mining module. In this module, we learned both Supervised and Unsupervised machine learning. Under Supervised learning, we learn techniques like CART(Classification and Regression Trees), Random Forest, and Artificial Neural Networks. We also learned various clustering techniques such as Hierarchical and K-Means clustering which unsupervised machine learning problems.

## 

## 

## 

## 

## **Problem 1**

A leading bank wants to develop a customer segmentation to give promotional offers to its customers. They collected a sample that summarizes the activities of users during the past few months. You are given the task to identify the segments based on credit card usage.

### Data dictionary

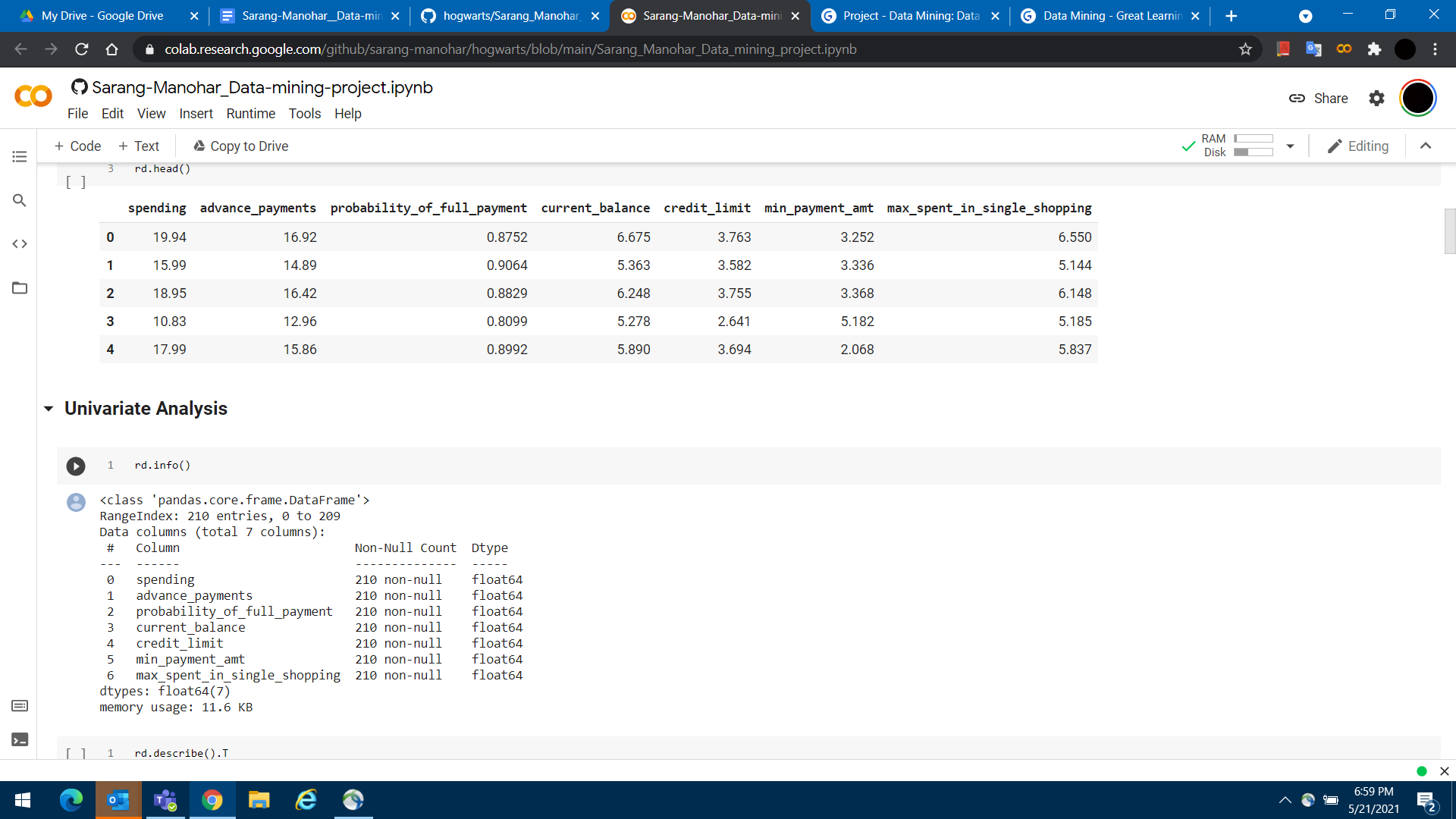
1. spending: Amount spent by the customer per month (in 1000s)
2. advance\_payments: Amount paid by the customer in advance by cash (in 100s)
3. probability\_of\_full\_payment: Probability of payment done in full by the customer to the bank
4. current\_balance: Balance amount left in the account to make purchases (in 1000s)
5. credit\_limit: Limit of the amount in credit card (10000s)
6. min\_payment\_amt: minimum paid by the customer while making payments for purchases made monthly (in 100s)
7. max\_spent\_in\_single\_shopping: Maximum amount spent in one purchase (in 1000s)

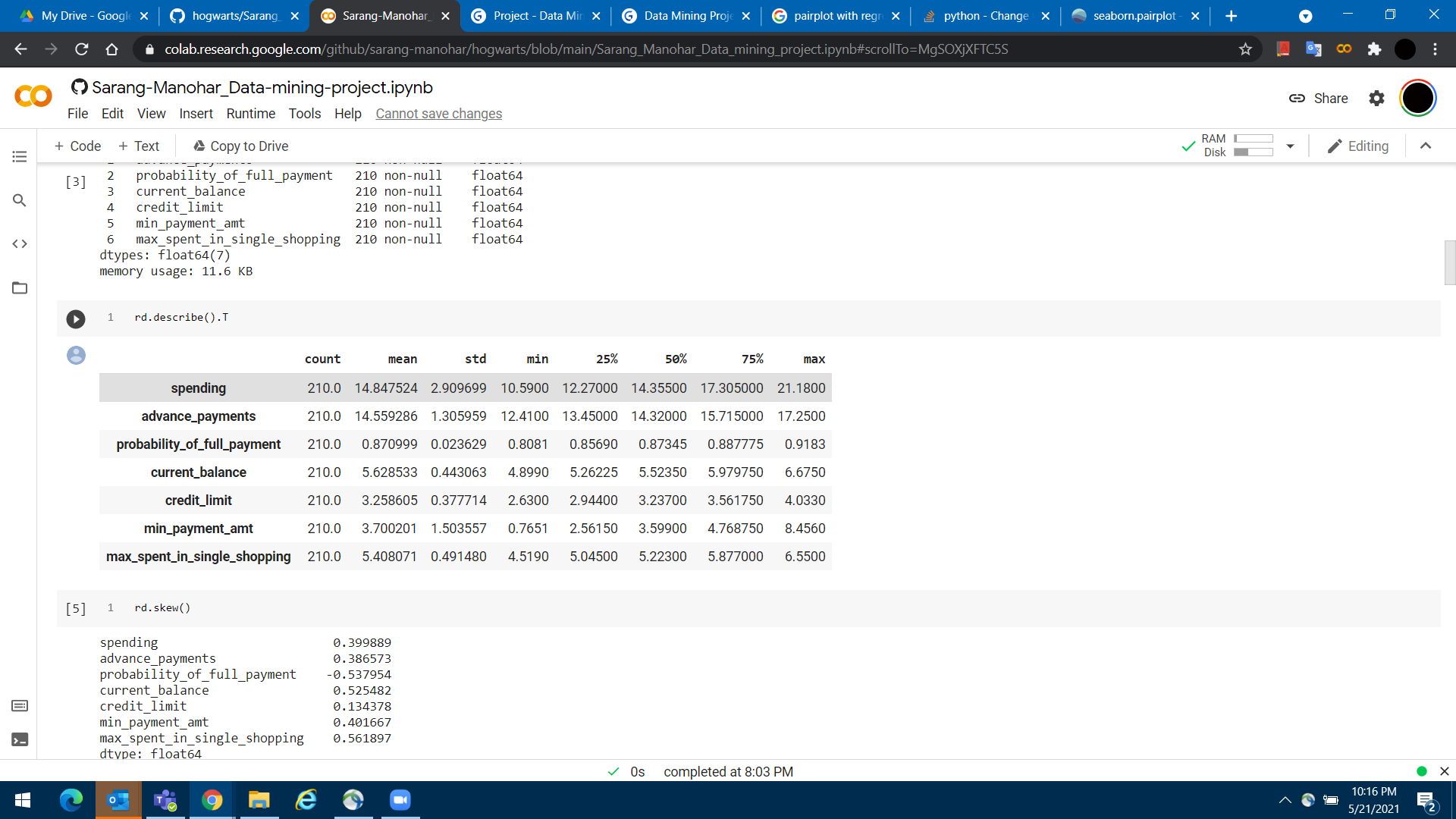
### 1.1 Read the data, do the necessary initial steps, and exploratory data analysis (Univariate, Bi-variate, and multivariate analysis)

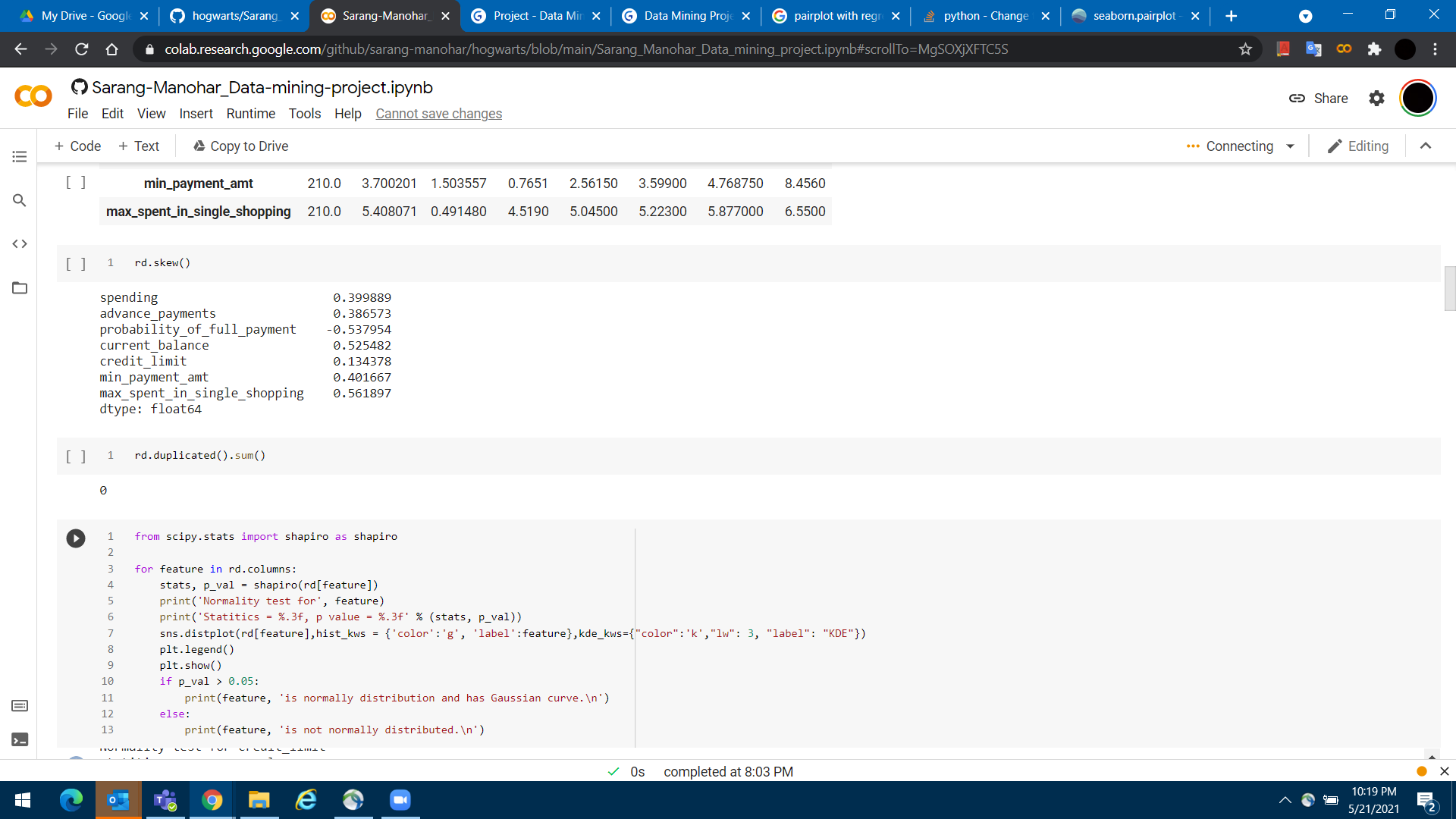
### 

Looking at the first 5 records following observations can be made

1. There are 7 independent variables
2. All of them are expected to have float datatype
3. There are no categorical variables, all of them are continuous







When we get the info of the data set we can confirm the following

1. There 210 observations in the data set
2. There are no null values across any of the 7 variables
3. All of them are of float64 data type which is in line with the initial expectations
4. The mean and 50 percentile values of all the variables have variations, suggesting skewness in the data
5. The min value is non-zero across all the variables suggesting there are no missing values that have been defaulted to zero. Hence, overall the data capture is of good quality with no missing values
6. A quick check on the duplicates confirms all the observations are unique and there are no duplicates

Since all of the 7 variables are continuous variables we can create histograms for each to check the distribution of the observations. We would run the following checks and see if the variables are normally distributed

* Shapiro-Wilk test
* Distribution curve
* Skewness stats
* Boxplot

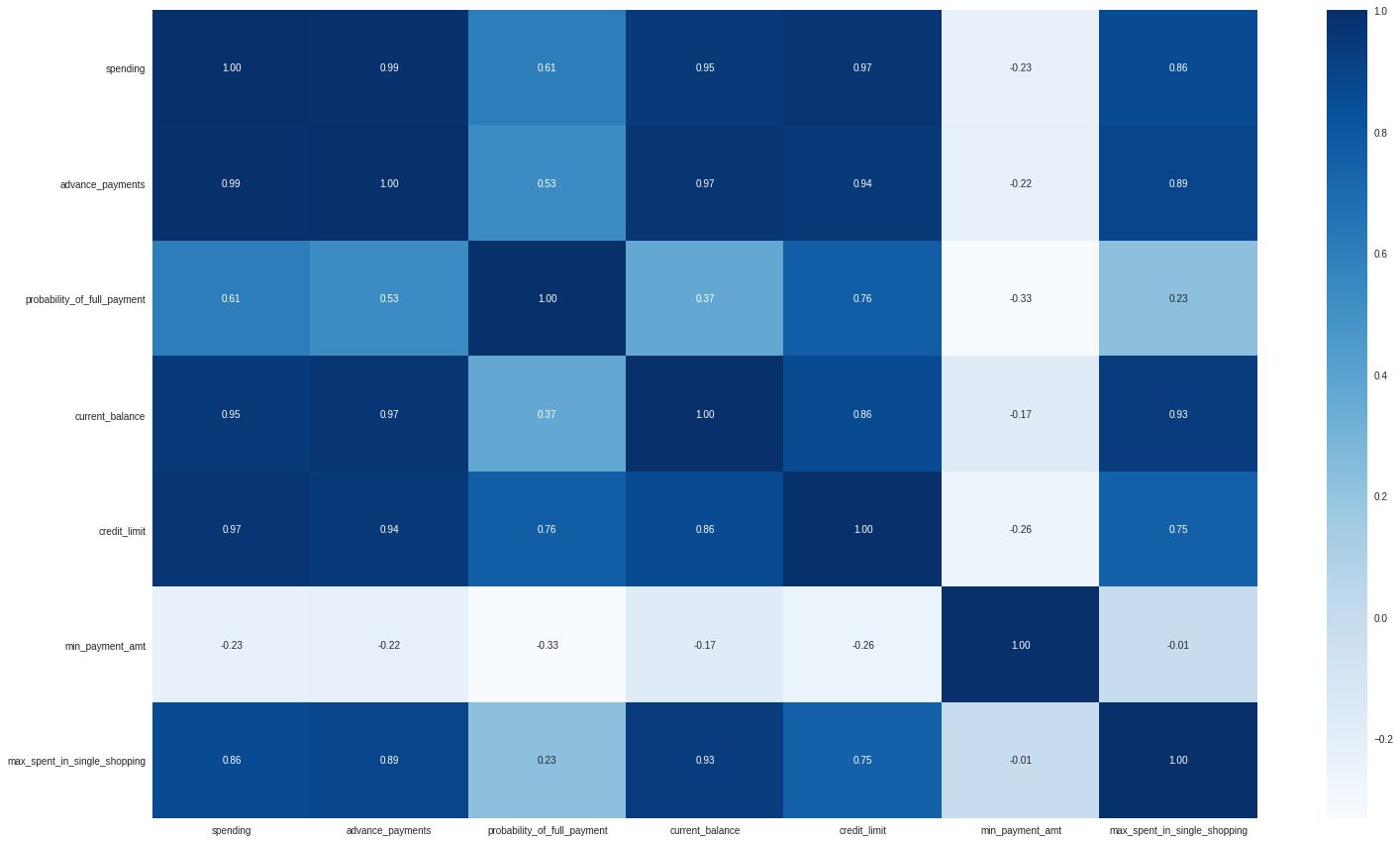
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Variable name** | **Shapiro-Wilk Stats** | **Skewness** | **Histogram and KDE plot** | **Boxplot** |
| spending | Statistics = 0.933  P-value = 0.000 | 0.399889 |  |  |
| advance\_payments | Statistics = 0.936  P-value = 0.000 | 0.386573 |  |  |
| probability\_of\_full\_payment | Statistics = 0.973  P-value = 0.000 | -0.537954 |  |  |
| current\_balance | Statistics = 0.944  P-value = 0.000 | 0.525482 |  |  |
| credit\_limit | Statistics = 0.961  P-value = 0.000 | 0.134378 |  |  |
| min\_payment\_amt | Statistics = 0.984  P-value = 0.000 | 0.401667 |  |  |
| max\_spent\_in\_single\_shopping | Statistics = 0.925  P-value = 0.000 | 0.561897 |  |  |

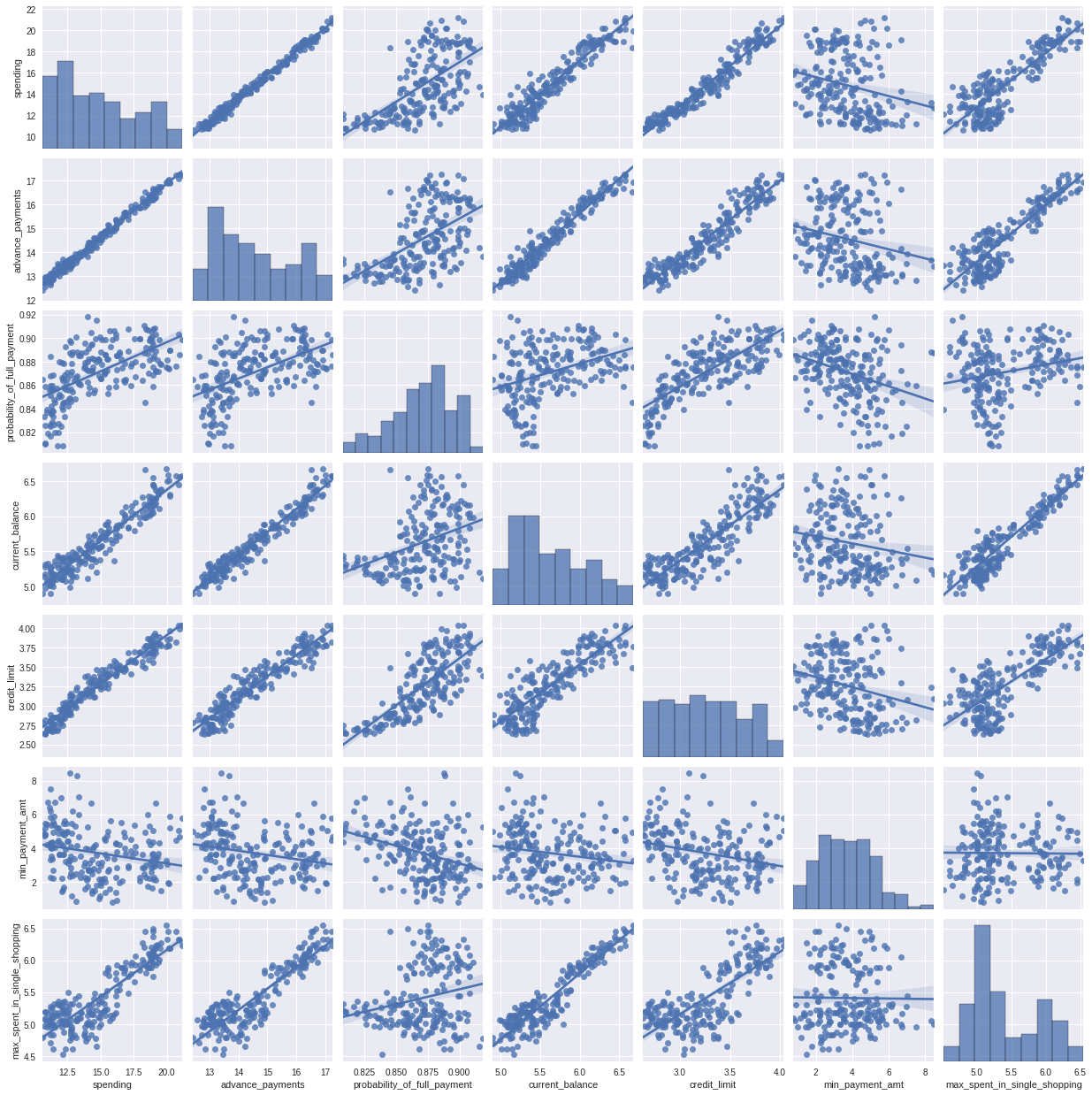
Following inferences can be made from the results of the tests executed above.

* P-Values of the Shapiro-Wilk test are less than 0.05 for all the variables, meaning none of them are normally distributed
* Similarly, the variation of skewness is also on the higher side with credit\_limit has the value nearest to zero i.e. ~0.134.
* All the variables are right-skewed except probability\_of\_full\_payment which is the only variable with left-skewness
* For neither of the variables, the KDE and plot and histogram plots follow the Gaussian curve
* We can observe that there aren’t any major outliers in the data except for probability\_of\_full\_payment & min\_payment\_amt with a boxplot. We can ignore these outliers for clustering as long any value is between 0 and 1, it should be an expected range for probability which is true in this case. In the case of min\_payment\_amt, the outlier value is very close to the max value on the boxplot.

With correlation and regression plots across variables, we adjudge the following

1. **spending**: Has a strong positive correlation with all other features except probability\_of\_full\_payment and min\_payment\_amt
2. **advance\_payments**: Similar to spending, it has a strong positive correlation with all other features except probability\_of\_full\_payment and min\_payment\_amt
3. **probability\_of\_full\_payment**: In general, has a weak but positive correlation with all features except min\_payment\_amt with which it has the strongest negative correlation amongst all the features
4. **current\_balance**: Similar to spending, it has a strong positive correlation with all other features except probability\_of\_full\_payment and min\_payment\_amt
5. **credit\_limit**: In general, it has a strong positive correlation with all other features except min\_payment\_amt
6. **min\_payment\_amt**: In general, it has a weak negative correlation with all other features except max\_spent\_in\_single\_shopping, with which it has almost no correlation
7. **max\_spent\_in\_single\_shopping** : Has strong positive correlation with all other features except probability\_of\_full\_payment and min\_payment\_amt





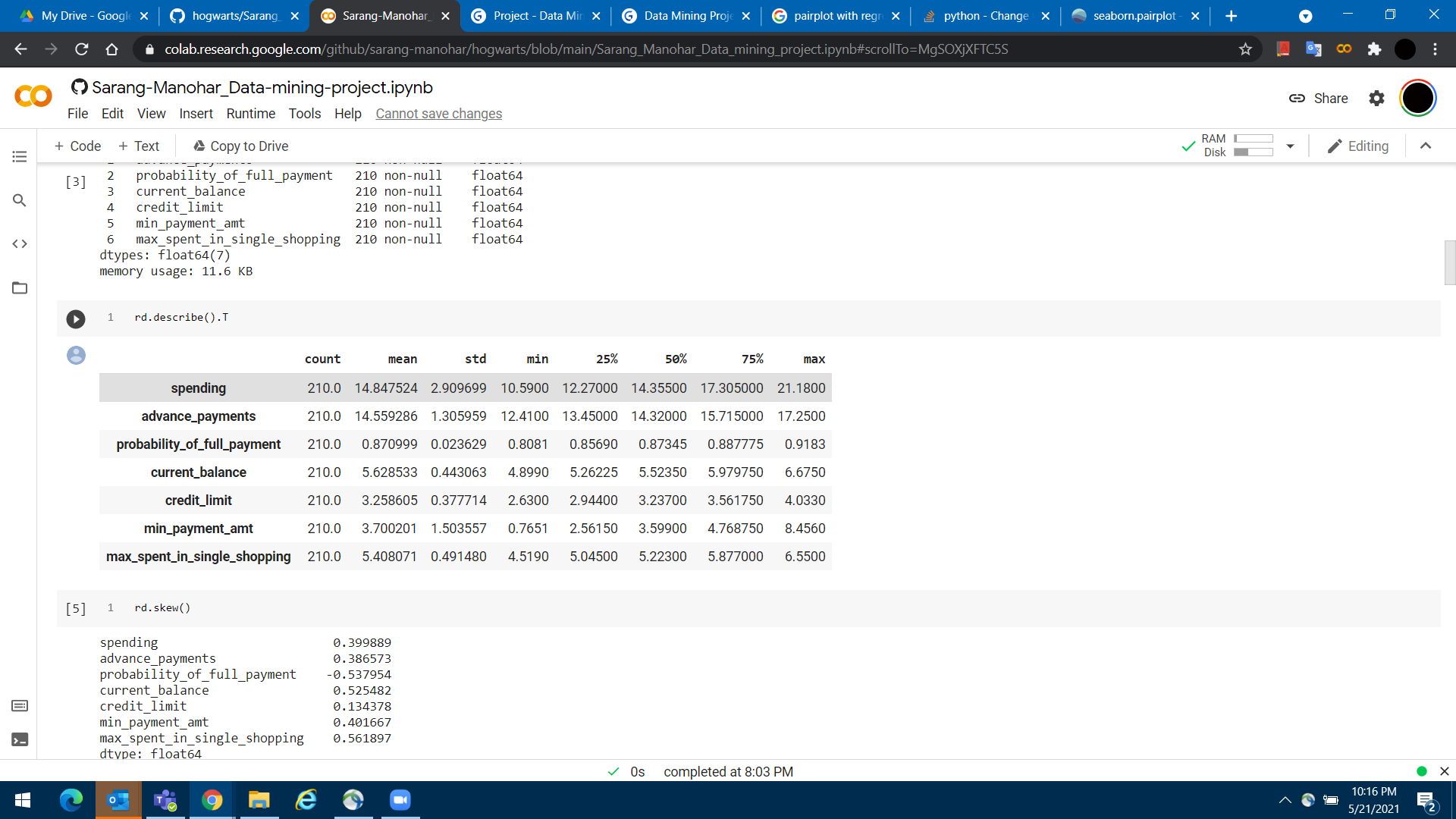
### 1.2 Do you think scaling is necessary for clustering in this case? Justify

Per the “Clustering in Machine Learning” course from Google.

“In clustering, you calculate the similarity between two examples by combining all the feature data for those examples into a numeric value. Combining feature data requires that the data have the same scale.”

[*https://developers.google.com/machine-learning/clustering/prepare-data*](https://developers.google.com/machine-learning/clustering/prepare-data)

The data set provided for this project is scaled down to a certain level from 100’s, 1000’s, and 10000’s to unity, and tens. Even after this normalization of the data, we can observe that the mean and standard deviation of all the features are not in a similar range. If this data is used as-is, and since clustering is distance-based algorithms, it will lead to biased output in favour of features like spending and advance\_payments.

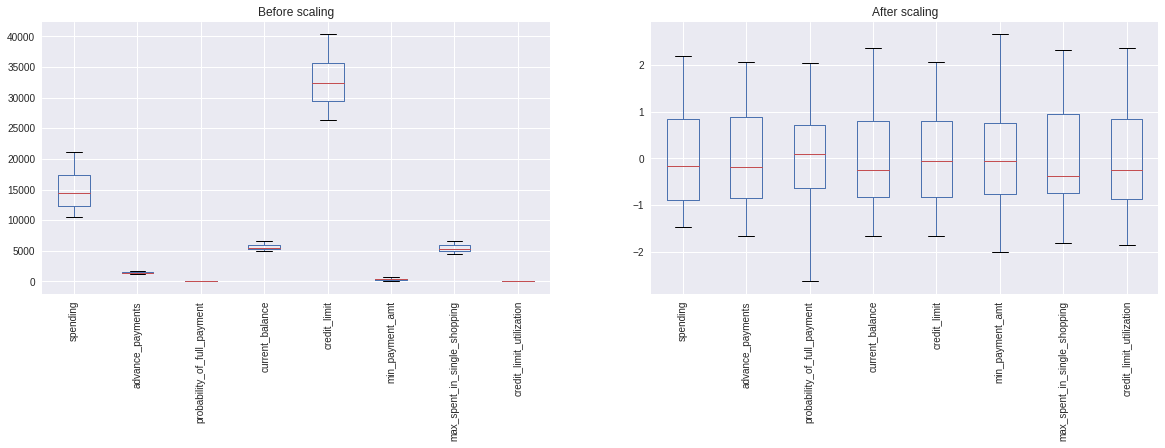


Also, using spending and credit\_limit, I have created a new feature named credit\_limit\_utlilization using the following formulae

*credit\_limit\_utilization = spending / credit\_limit*

This feature is a common way to look at the financial health of the customers and also gauge their dependency on credits for their regular spending. A high credit\_limit\_utlization infers a higher risk for the lenders. A low credit\_limit\_utlization means lower risk for the lenders and financially disciplined customers.

This feature is introduced in the data set right before the scaling of the data which we will discuss next.

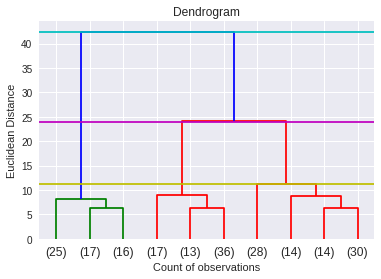
To solve the problem at hand, I have up-scaled the data to the factors they have been scaled down from and then applied zscore for Hierarchical clustering and StandardScaler for K-Means clustering to scale down the data. In doing so, the intent was to scale down the data from the actual values. This is to avoid any mathematical bias with the original state of the data where it was misleading, for example, spending and advance\_payments were of the same scale whereas, in reality, the spending is 10 times the scale of advance\_payments. Scaling down transforms all the features with a mean of 0 and the standard deviation of 1.

### 1.3 Apply hierarchical clustering to scaled data. Identify the number of optimum clusters using Dendrogram and briefly describe them

Analyzing the output of dendrograms and applying various linkage methods, we got the following output.

|  |  |  |  |
| --- | --- | --- | --- |
| **Linkage** | **Is monotonic?** | **Max distance bw. Non-singleton clusters** | **Dendrogram** |
| Single | TRUE | 1.558283 |  |
| Complete | TRUE | 8.793922 |  |
| Average | TRUE | 4.915263 |  |
| Weighted | TRUE | 5.180676 |  |
| Centroid | FALSE | 4.454246 |  |
| Median | FALSE | 5.504185 |  |
| Ward | TRUE | 42.43271 |  |

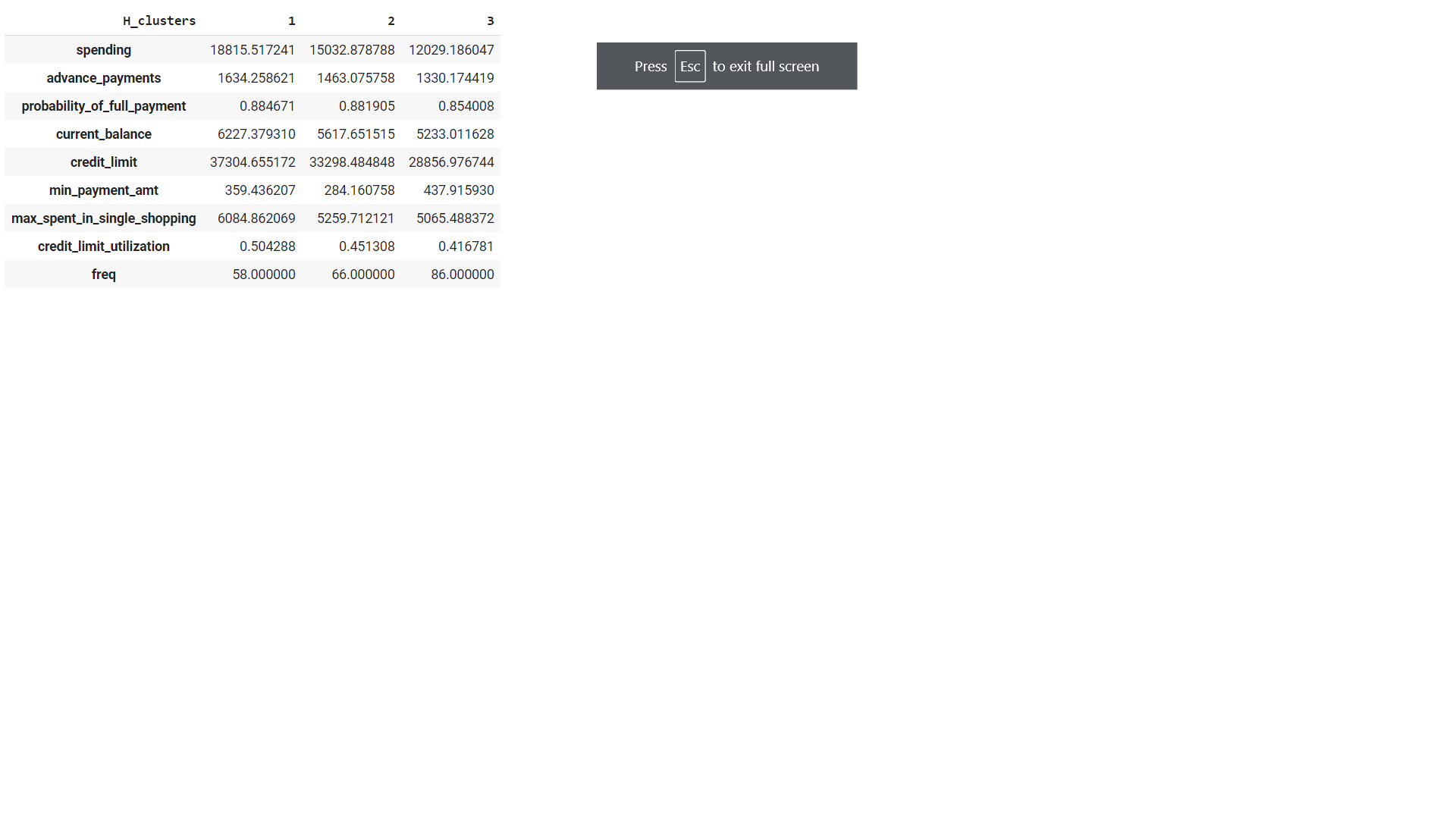
Since with the Ward linkage method, the vertical distance between various cluster nodes is maximum, I choose to use this methodology. By default, for Ward Euclidean distance(affinity) gets applied. Looking at the Dendrogram below we can infer that if we cut the tree between 11(yellow line) and 24(magenta line) then it would result in 3 distinct clusters with the size of (58, 66, 86)



Next, the Hierarchical clustering was executed on the scaled data, via the Fclusters([criterion = ‘maxcluster’](https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.hierarchy.fcluster.html?highlight=fcluster#scipy.cluster.hierarchy.fcluster)) to create 3 clusters out of the data. The silhouette scores were calculated for various cluster sizes. We can see that a cluster size of 3 has the best silhouette score of ~0.383 and a minimum silhouette sample value of ~ -0.198.



Looking at the mean stats of variables across 3 clusters, we can see that the data has been distinctly split into relevant segments.



The 3 clusters can be broadly classified as below segments:

1. High income
2. Medium to high income
3. Medium to low income

##### High income

Following are the observations made for the high income segment

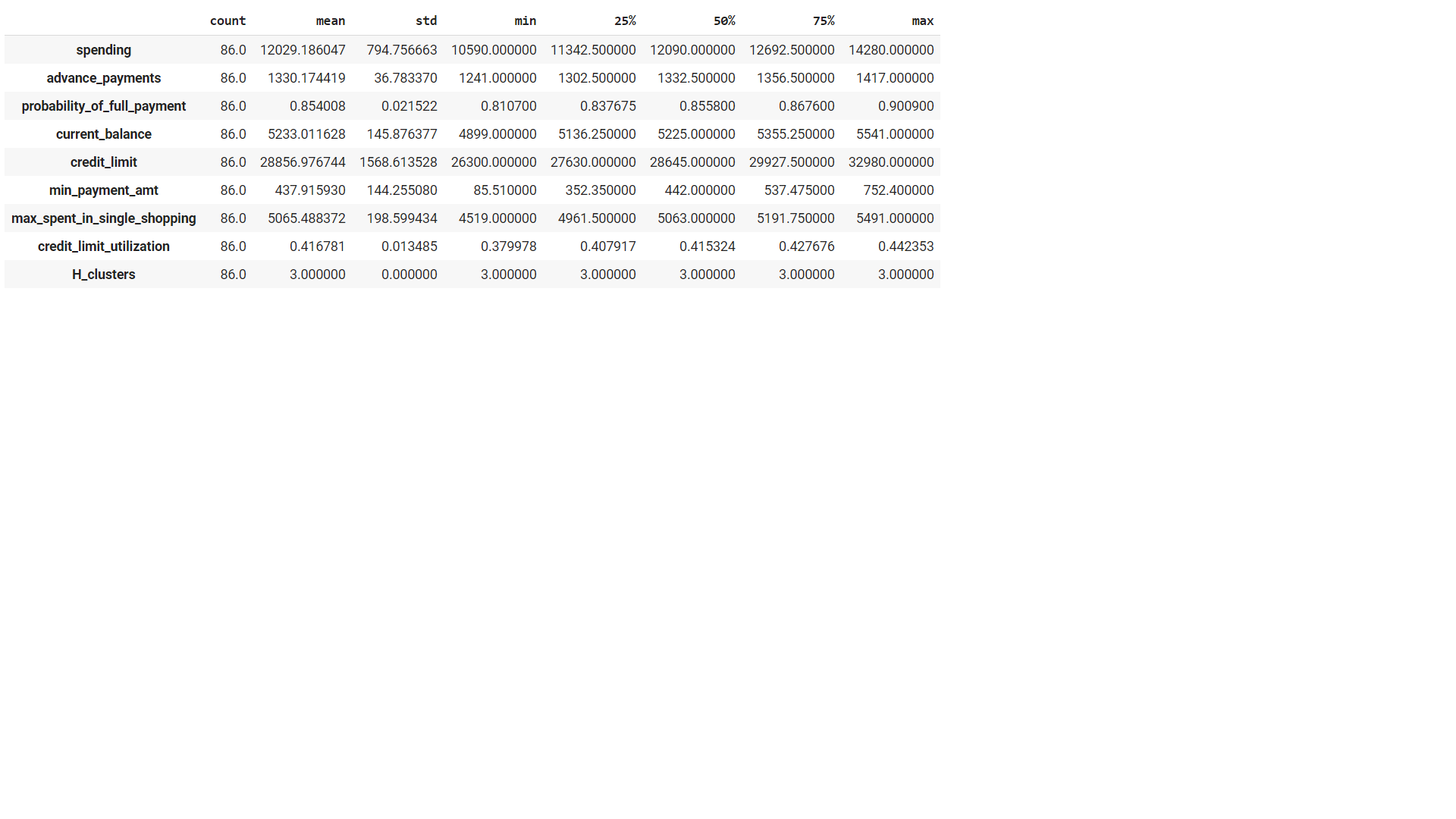
1. The average spending of this segment is almost ~18,815.5, with a maximum observation of 21,180 and a minimum of 16,770
2. This segment is doing worst in terms of advance\_payments as it pays almost ~8.68% of overall spending in advance which is the least amongst all the 3 segments
3. The probability of full payment is decent with almost 88.46%
4. Both current\_balance and credit\_limit provided by the bank is high as compared to other segments
5. The min\_payment\_amt is less than the medium to low-income segment
6. This segment is utilizing almost ~50% of its’ credit limit, which is the highest amongst the 3 segments
7. max\_spending \_in\_single shopping is highest for this segment

##### Medium to high income

Following are the observations made for the medium to high income segment

1. The average spending of this segment is almost ~15,032.88, with a maximum observation of 17,120 and a minimum of 13,500
2. This segment is doing better than the high-income segment in terms of advance\_payments as it pays almost ~9.73% of overall spending in advance
3. The probability of full payment is decent with almost 88.19% which is very close to the high income segment
4. Interestingly, the min\_payment\_amt is least amongst all the segments
5. This segment is utilizing almost ~45% of its’ credit limit, which is better than the high-income segment, suggesting this segment is more disciplined with its spending

##### Medium to low income



Following are the observations made for the medium to low income segment

1. This segment has the maximum number of customers amongst all the segments i.e 86
2. The average spending of this segment is least and almost ~12,029.19, with a maximum observation of 14,280 and a minimum of 10,590
3. This segment is doing best in terms of advance\_payments as it pays almost ~11% of overall spending in advance
4. The probability of full payment is decent with almost 85.4% which is the least amongst all the 3 segments
5. The min\_payment\_amt for this segment is the highest
6. This segment is utilizing almost ~41.7% of its’ credit limit, which is best of all the segments, suggesting this segment is most disciplined with its spending

The clusters created from the Hierarchical approach can be visualized as below

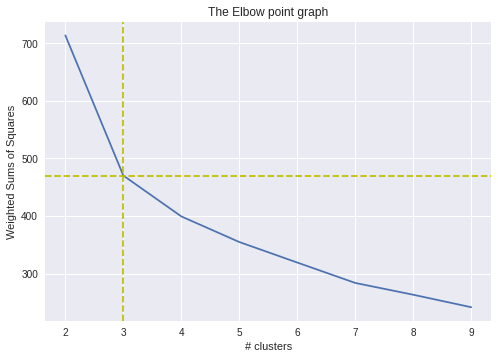


We can see that all the 3 clusters are well-defined majority of features including the engineered feature credit\_limit\_utilization.

### 

### 1.4 Apply K-Means clustering on scaled data and determine optimum clusters. Apply elbow curve and silhouette score. Explain the results properly. Interpret and write inferences on the finalized clusters.

For K-Means clustering, the first step was to create an Elbow point graph using weighted sums of squares



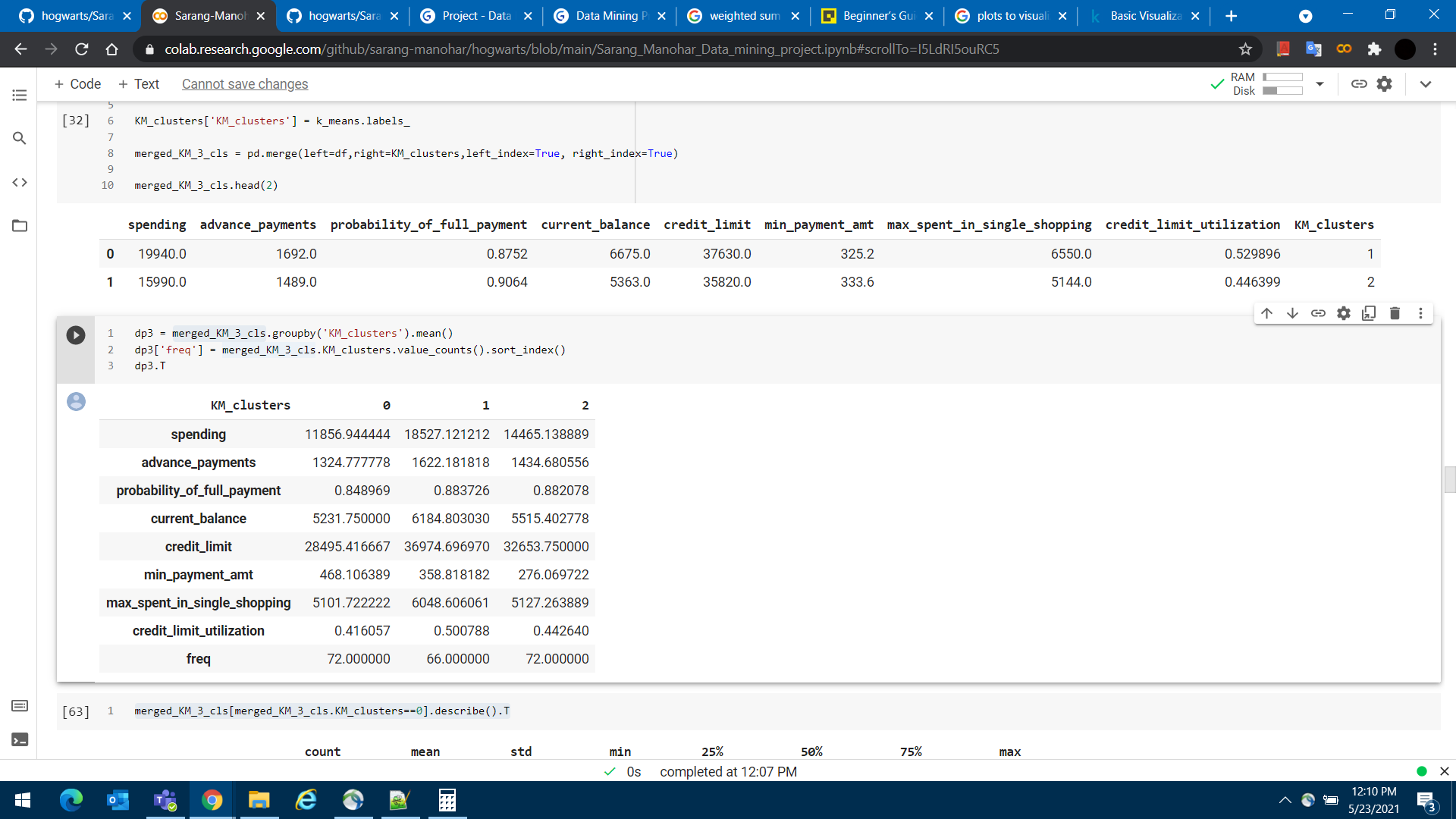
The elbow point graph shows a sharp bend at 3 clusters. Further, The silhouette score and minimum silhouette samples for various clusters were calculated as below. We can see that the 3 clusters have the best Silhouette score and however, the minimum silhouette samples value is farther away from that of the 4 clusters.



Let us investigate further to determine the optimal # clusters. The silhouette analysis of 4 clusters shows cluster ‘0’ overlaps with both cluster ‘2’ and ‘3’ to a great extent(marked by arrows). Also, clusters ‘2’ and ‘3’ are almost touching each other and cluster ‘0’ is in a way spread across the two centroids. This isn’t a clear demarcation of clusters. Hence, 3 clusters appear to be the optimal choice for this data set.

Based on the above inferences we can split the observations into 3 broad segments. They are very similar to hierarchical clusters but with observations more evenly spread across the segments i.e. (66,72,72)

1. High income
2. Medium to high income
3. Medium to low income



##### High income

##### 

Following are the observations made for the high income segment

1. The average spending of this segment is almost ~18,527.12, with a maximum observation of 21,180 and a minimum of 15,770
2. This segment is doing worst in terms of advance\_payments as it pays almost ~8.75% of overall spending in advance which is the least amongst all the 3 segments
3. The probability of full payment is decent with almost 88.37%
4. Both current\_balance and credit\_limit provided by the bank is high as compared to other segments
5. The min\_payment\_amt is less than the medium to low income segment
6. This segment is utilizing almost ~50% of its’ credit limit, which is the highest amongst the 3 segments
7. max\_spending \_in\_single shopping is highest for this segment

##### 

##### Medium to high income

Following are the observations made for the medium to high income segment

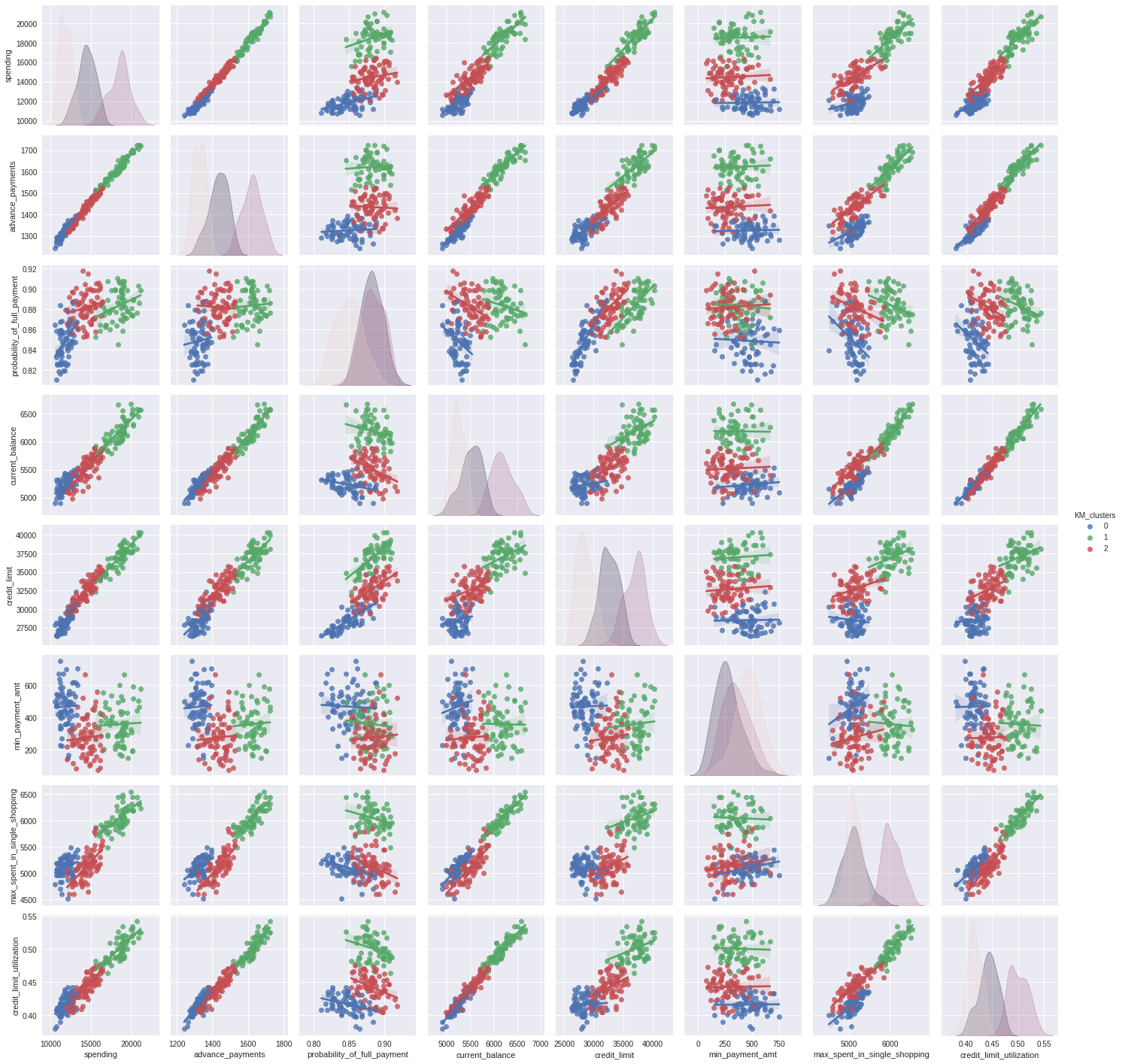
1. The average spending of this segment is almost ~14,465.14, with a maximum observation of 16,440 and a minimum of 12,080
2. This segment is doing better than high income segment in terms of advance\_payments as it pays almost ~9.91% of overall spending in advance
3. The probability of full payment is decent with almost 88.2% which is very close to the high income segment
4. Interestingly, the min\_payment\_amt is least amongst all the segments
5. This segment is utilizing almost ~44.2% of its’ credit limit, which is better than the high income segment, suggesting this segment is more disciplined with its spending

##### Medium to low income

Following are the observations made for the medium to low income segment

1. This segment has the maximum number of customers amongst all the segments i.e 86
2. The average spending of this segment is least and almost ~11,856.94, with a maximum observation of 13,440 and a minimum of 10,590
3. This segment is doing best in terms of advance\_payments as it pays almost ~11.16% of overall spending in advance
4. The probability of full payment is decent with almost 84.9% which is the least amongst all the 3 segments
5. The min\_payment\_amt for this segment is the highest
6. This segment is utilizing almost ~41.6% of its’ credit limit, which is best of all the segments,
7. suggesting this segment is most disciplined with their spending

The clusters created from the K-Means approach can be visualized as below.

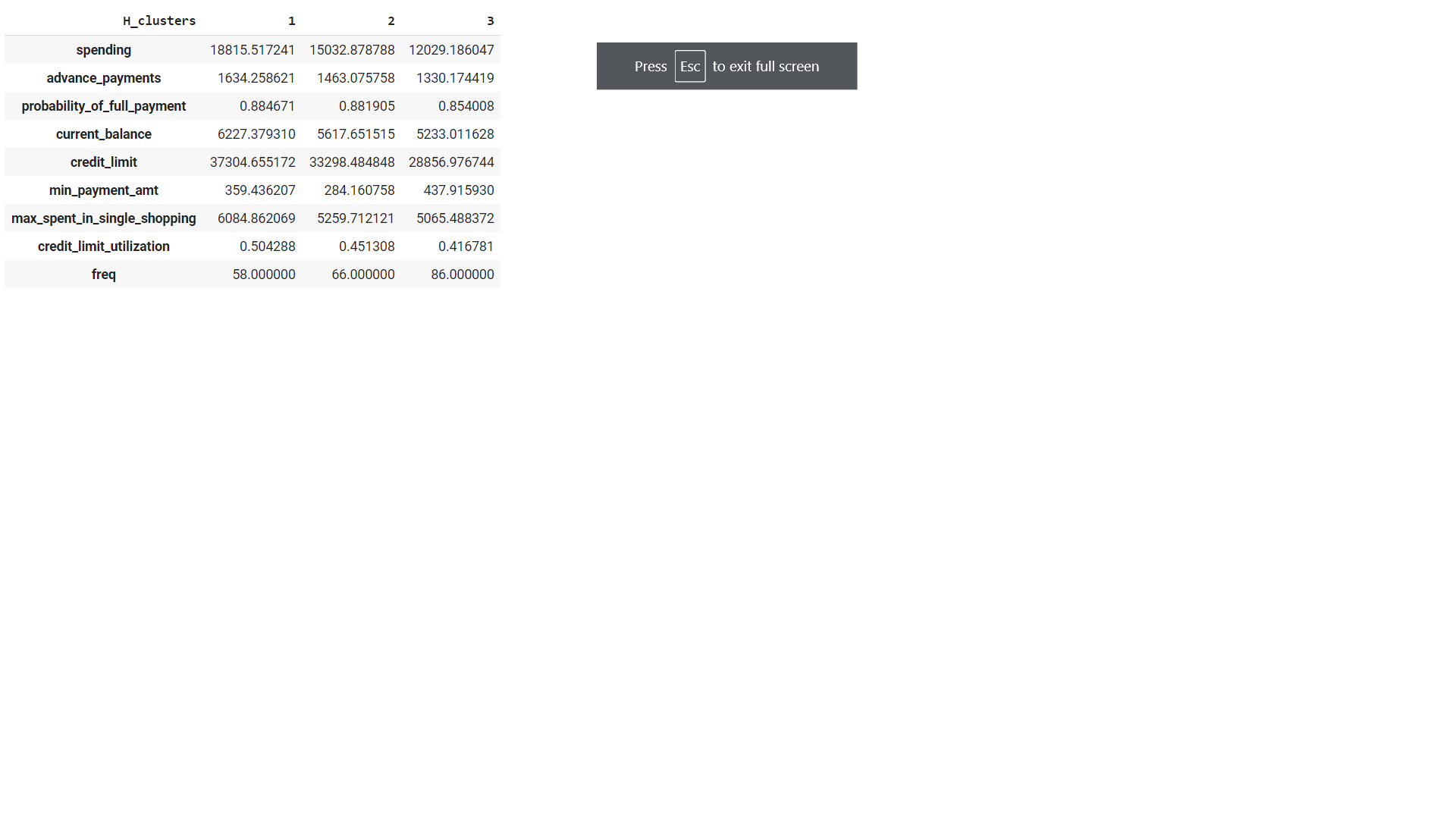


We can see that all the 3 clusters are well-defined majority of features including the engineered feature credit\_limit\_utilization.

### 1.5 Describe cluster profiles for the clusters defined. Recommend different promotional strategies for different clusters.

The output of Hierarchical clustering, helped us split the customers into 3 broad segments as below

1. High income
2. Medium to high income
3. Medium to low income



High income segment are the **high valued customers** but have **poor financial discipline**. This group is at **high risk** for the bank. They

1. are the **highest spender** but have the **lowest advance payments** amongst all the segments
2. have **highest credit utilization** even though they have the **best credit limit** amongst the group
3. tend to **spend huge amounts in single shopping transactions**
4. The **probability of full payment is highest** possibly because they are financially stable

**Promotional strategy:** We should look to retain/grow this segment by providing value-added services but at the same time ensuring not to increase any liabilities for the customers, given the high-risk profile for this segment is high. Offering upgrades to premium credit cards, with perks like

* discounts on restaurant/hotel bills
* priority check-in at airports/lounges
* attached health, accidental, and travel insurance plans

Medium to high income is a segment with a lot of potentials that **can drive growth for the bank.** They have **moderate financial discipline,** which the bank can tap in to increase the revenue by influencing their spending patterns. They

1. are the **moderate spenders** and also does **advance payments with good discipline** amongst all the segments
2. have a **moderate credit utilization**
3. not huge **spenders in single shopping transactions**
4. The **probability of full payment is at par with high income segment** possibly because they are financially stable

**Promotional strategy:** As this segment has good financial discipline, stimulating this group to spend more would be the key to the success of the bank. Co-brand credit cards would be the ideal product for this segment.

* discounts/cashback on spending on e-commerce websites
* offering to convert outstanding amounts to EMI’s

Medium to low income is the **most disciplined in financial terms**. This group is of the lowest risk and the bank can look to promote additional products to this segment but with caution. They

1. have the **best advance payments to spending ratio**
2. have the **lowest probability of full payment** amongst all the segments
3. have the **lowest credit limit utilization**, suggesting less reliance on credits

**Promotional strategy:** This is a low-spending segment. These could be customers who have recently joined the workforce or have financial instability and hence refrain from a lot of spending. The bank should look to expand their services that give support to such customers with additional products such as

* personal loans
* overdraft accounts
* offers on long-term investments accounts

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## **Problem 2**

An Insurance firm providing tour insurance is facing higher claim frequency. The management decides to collect data from the past few years. You are assigned the task to make a model which predicts the claim status and provides recommendations to management. Use CART, RF & ANN and compare the models' performances in train and test sets.

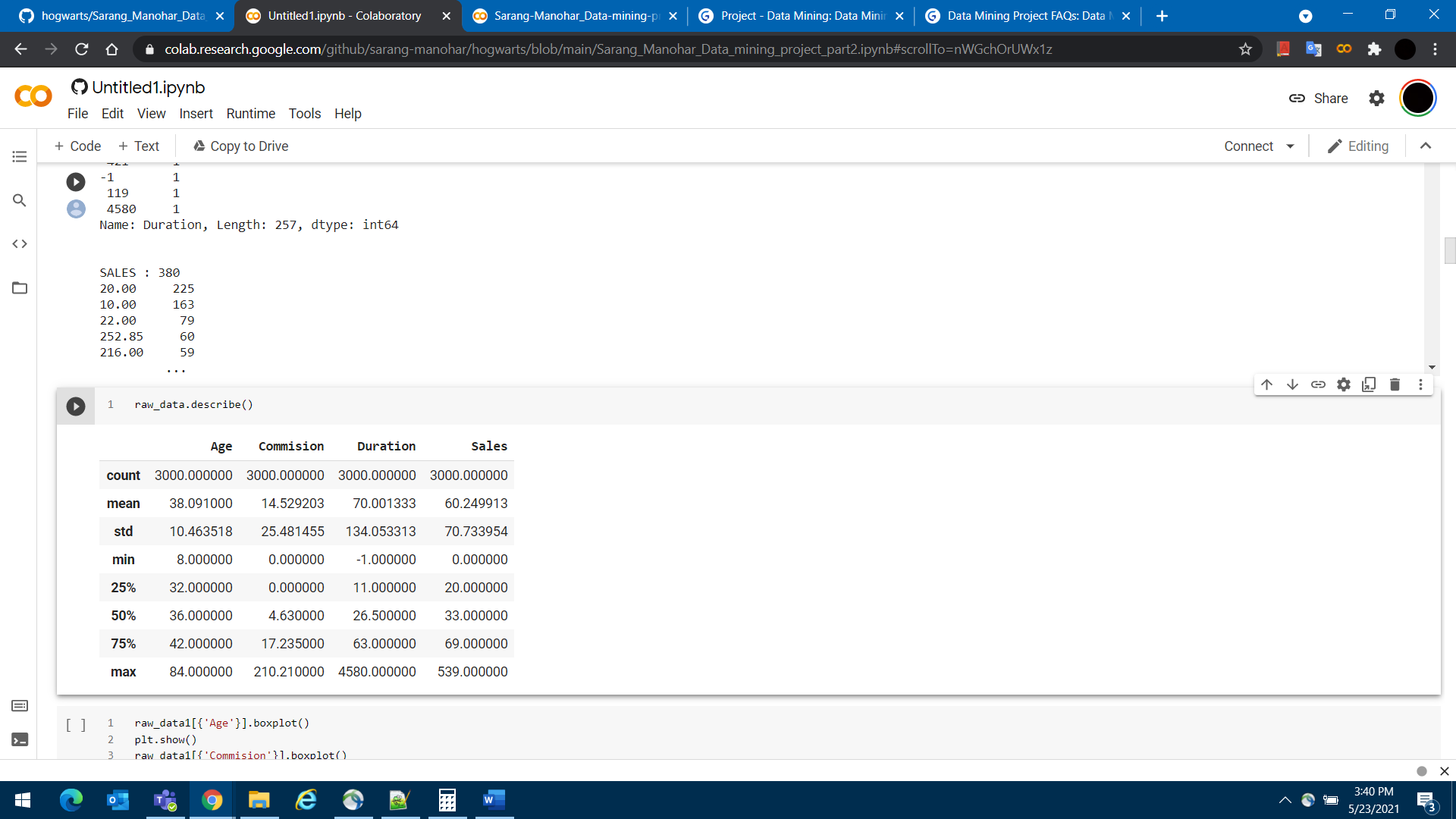
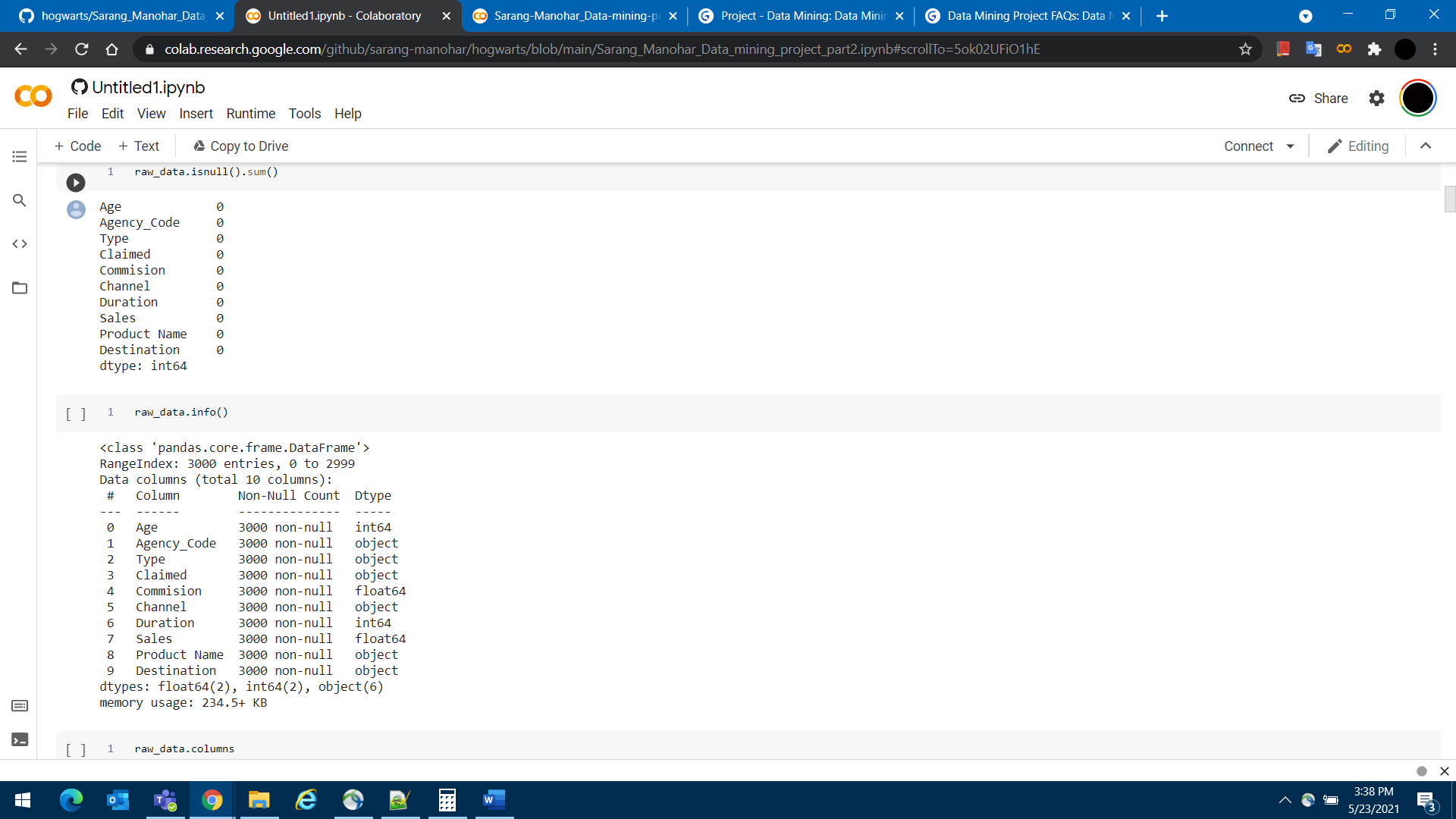
### Data dictionary

1. Target: Claim Status (Claimed)
2. Code of tour firm (Agency\_Code)
3. Type of tour insurance firms (Type)
4. Distribution channel of tour insurance agencies (Channel)
5. Name of the tour insurance products (Product)
6. Duration of the tour (Duration)
7. Destination of the tour (Destination)
8. Amount of sales of tour insurance policies (Sales)
9. The commission received for tour insurance firm (Commission)
10. Age of insured (Age)

### **2.1** Read the data, do the necessary initial steps, and exploratory data analysis (Univariate, Bi-variate, and multivariate analysis).

Looking at the first 5 records following observations can be made

1. There are 9 independent variables and 1 dependent variable
2. Age, Commission, Duration, and Sales are continuous independent variables
3. Agency\_Code, Type, Channel, Product Name, and Destination are categorical independent variables
4. Claimed is the dependent variable



When we get the info of the data set we can confirm the following

1. There are 3,000 observations in the data set
2. There are no NULL values across any of the 9 independent and 1 dependent variable
3. The continuous variables are either int or float which as per the expectation
4. All the categorical variables are of object data type
5. Categorical variables have the following characteristics
   1. Agency\_Code: 4 distinct values, EPX has a maximum of 1,365 occurrences, JZI has a minimum of 239 occurrences
   2. Type: 2 distinct values, Travel Agency with 1,837 occurrences, Airlines with 1,163 occurrences
   3. Channel: 2 distinct values, Online with 2,954 occurrences, Offline with 46 occurrences
   4. Product name: 5 distinct values, Customised Plan with maximum 1,136 occurrences and Gold Plan with minimum 109 occurrences
   5. Destination : 3 distinct values, Asia was the most preferred destination with 2,465 occurrences, followed by Americas with 320 occurrences and then Europe with 215 occurrences
6. The continuous variables have the following characteristics

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Variable name** | **Skewness** | **Outliers %** | **Histogram and KDE plot** | **Boxplot** |
| Age | 1.149713 | 6.8% |  |  |
| Commision | 3.148858 | 18.87% |  |  |
| Duration | 13.784681 | 31.6% |  |  |
| Sales | 2.381148 | 43.37% |  |  |

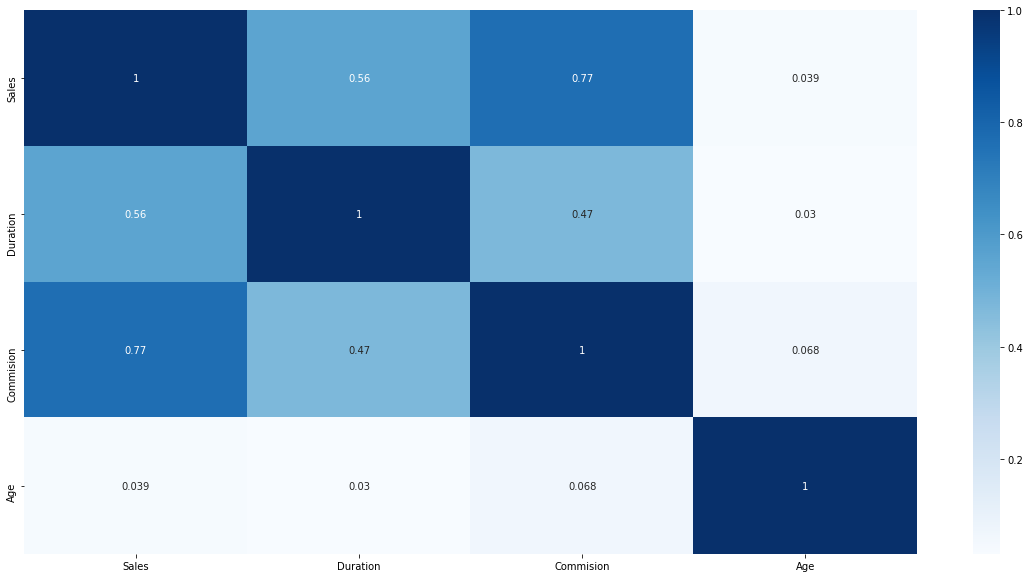
* 1. All the continuous variables are right-skewed
  2. The KDE plots suggest they are don’t follow the Gaussian curve
  3. Boxplot shows there are many outliers in the data, on further calculation Sales and Duration values are 43.37% and 31.6% outside the inter-quartile range respectively

The above observations suggest that none of the continuous variables are normally distributed. However, there are other observations with continuous variables worth noting

1. There are 9 observations where the age of customers is less than 18 years, of which 1 record with age as low is 8 years
2. There are 3 records with a Duration less than 1, but all of them have Claimed as No(class 0), hence we should be good not to treat them any further
3. There 53 records with Sales equal to 0, but all of them have Claimed as No(class 0), hence we should be good not to treat them any further
4. There appear 139 records that may be duplicates, but given the collection of variables, these could be mere coincidences that all the observations exactly match with another observation. This would need further investigation with additional variables, but for, this project we will retain these records for now.

Looking at the correlation plot, follow can be observed

1. There is some positive correlation between Sales and Commision
2. The positive correlation between Sales and Duration is also worth noting
3. Age has no correlation with any of the other continuous variables



### **2.2** Data Split: Split the data into test and train, build classification model CART, Random Forest, Artificial Neural Network

Before we split the data for training, the categorical variables with object data types have to be converted into numeric categorical variables.

With categorical variables now converted to numeric values, the data set is now split into train and test data with a 70:30 split ratio and with a random state equal to 1.

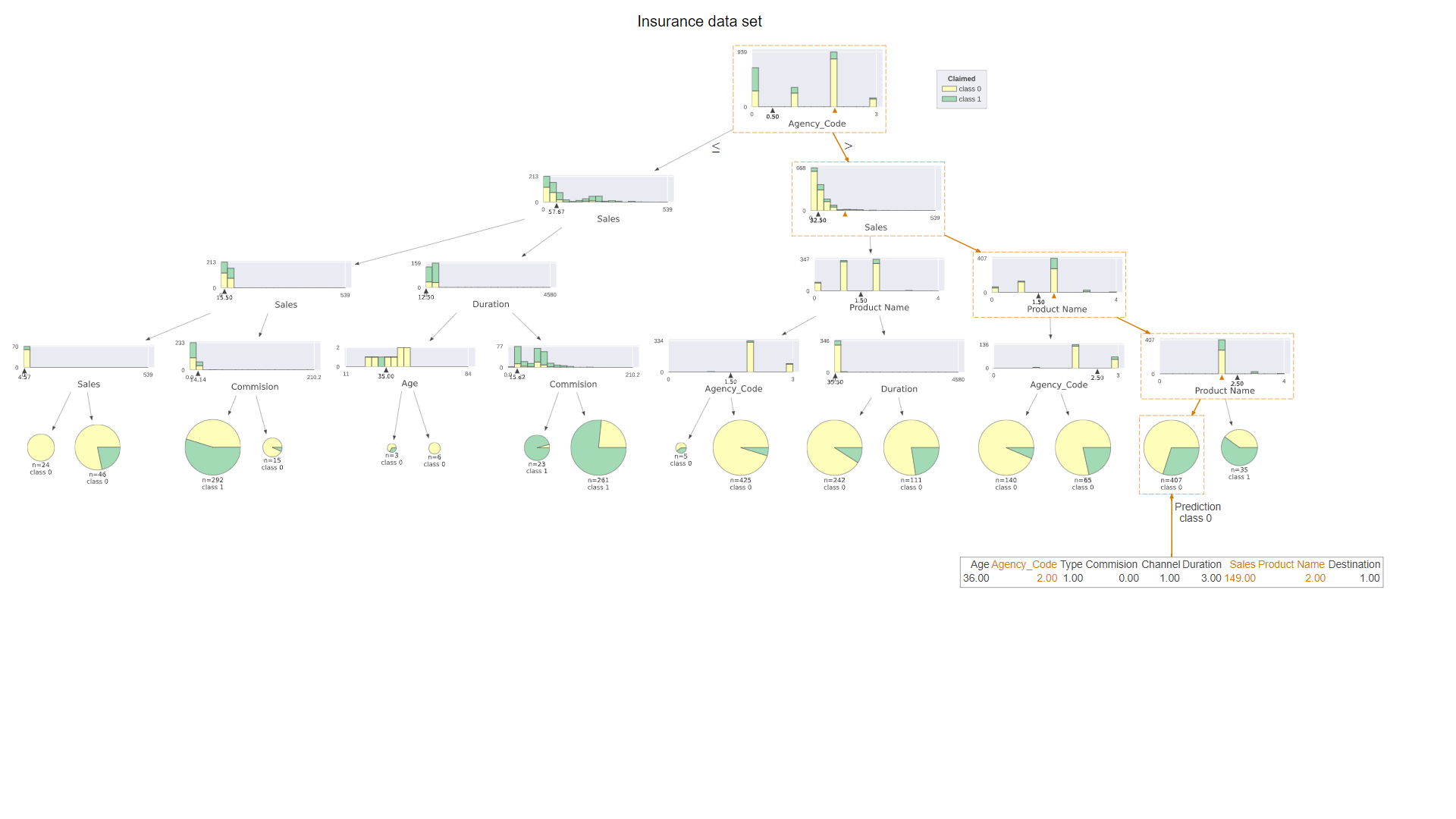
Pre and post data split between train and test samples the distribution of target variable ‘Claimed’ is as below

|  |  |  |
| --- | --- | --- |
|  | Class 0 | Class 1 |
| Original data | 69.2% | 30.8% |
| Train data | 70.1% | 29.9% |
| Test data | 67.2% | 32.8% |

##### **The initial version of models without hyperparameter tuning**

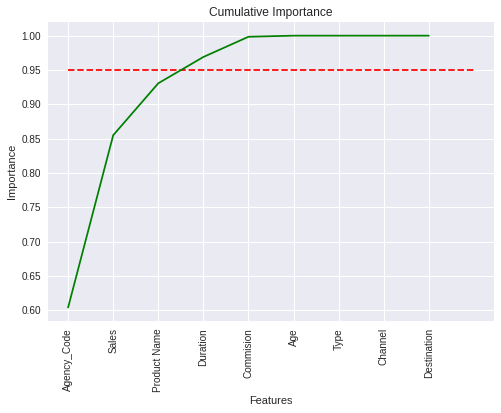
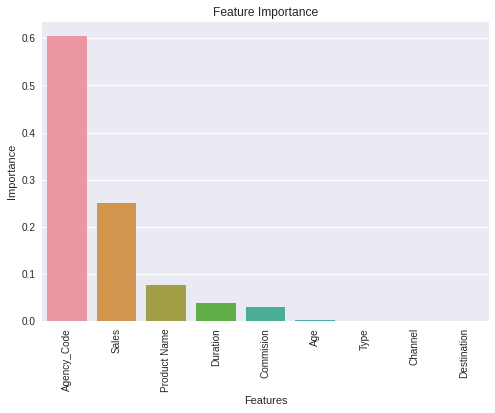
**Decision Tree**

Using the ‘gini’ index as criteria, below is a decision tree with a maximum depth of 4 with no hyperparameters.



DecisionTreeClassifier(ccp\_alpha=0.0, class\_weight=None, criterion='gini', max\_depth=4, max\_features=None, max\_leaf\_nodes=None, min\_impurity\_decrease=0.0, min\_impurity\_split=None, min\_samples\_leaf=1, min\_samples\_split=2, min\_weight\_fraction\_leaf=0.0, presort='deprecated', random\_state=None, splitter='best')

Using the attributes of the decision tree model with default parameters we can see that Agency\_Code has the highest importance influencing 60% of the decision making of the model. Variables like Age, Type, Channel, and Destination have almost no influence on the decision-making algorithm of the model. This can be referred back to EDA where we saw that Age has no correlation with other continuous variables and Type, Channel and Destination were categorical variables with the majority of observations biased towards one value.



The final set of hyperparameters used for the Decision tree

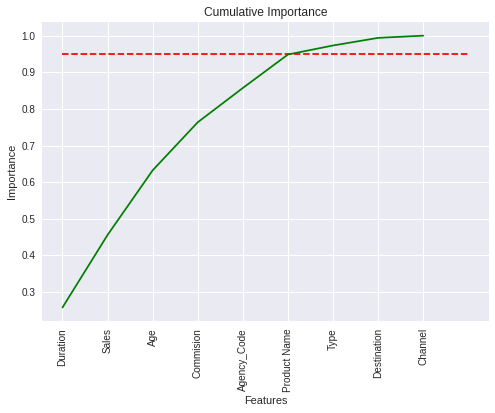
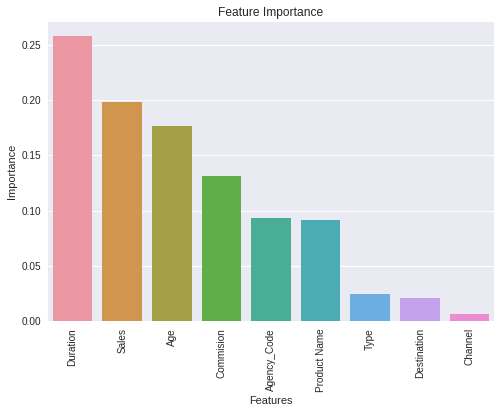
* Maximum depth = 6, any value higher than this would have caused the tree to overgrow, and values lesser than it would have resulted in biased outcomes without considering the importance of other features
* Minimum samples split = 50, typically taken as 1-2% of # observations, in our case between 30-60
* Minimum samples leaf = 15, with above two parameters, I tried various combinations of minimum samples leaf and got the best output with 15

Any values higher or lower than the above-mentioned values were resulting in lower accuracy, precision, and recall scores for class 1on test data. After trying multiple combinations, I was able to get to a difference of 2% between train and test data accuracy of the model

**Random Forest**

RandomForestClassifier(bootstrap=True, ccp\_alpha=0.0, class\_weight=None, criterion='gini', max\_depth=None, max\_features='auto', max\_leaf\_nodes=None, max\_samples=None, min\_impurity\_decrease=0.0, min\_impurity\_split=None, min\_samples\_leaf=1, min\_samples\_split=2, min\_weight\_fraction\_leaf=0.0, n\_estimators=100, n\_jobs=None, oob\_score=False, random\_state=None, verbose=0, warm\_start=False)

Looking at the importance plot of various features with default parameters, it is much different from our observations from the decision tree. With Random forest, the importance is more democratized and the majority of the features have a significant influence on the decision making. To reach 95% of importance, it took 6 out 9 features.



The final set of hyperparameters used for the Random forest

* Maximum features = 3, typically taken as the square root of the number of features, which was 9 in our case
* Minimum samples split = 50, typically taken as 1-2% of # observations, in our case between 30-60
* Number of estimators(trees) = 100, any values less or more than 100 was giving a lesser accuracy score with test data

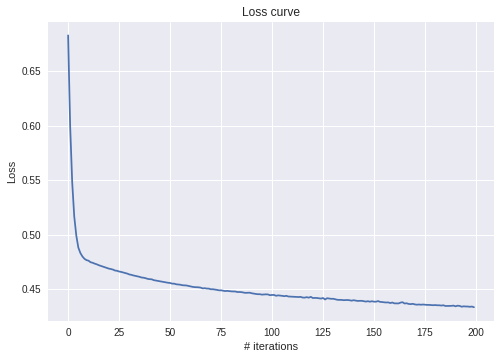
Any values higher or lower than the above-mentioned values were resulting in lower accuracy, precision, and recall score for class 1 on test data. After trying multiple combinations, I was able to get to a difference of ~5% between the train and test data accuracy of the model.

**Artificial Neural Network(ANN)**

Artificial neural networks required scaled data as a pre-processing step. Standardscaler() was used to scale down the data.

MLPClassifier(activation='relu', alpha=0.0001, batch\_size='auto', beta\_1=0.9, beta\_2=0.999, early\_stopping=False, epsilon=1e-08, hidden\_layer\_sizes=(100,), learning\_rate='constant', learning\_rate\_init=0.001, max\_fun=15000, max\_iter=200, momentum=0.9, n\_iter\_no\_change=10, nesterovs\_momentum=True, power\_t=0.5, random\_state=None, shuffle=True, solver='adam', tol=0.0001, validation\_fraction=0.1, verbose=False, warm\_start=False)

We ran the ANN model with default parameters captured above and got the following loss curve. We can see that after the first few iterations the loss rate reduces and flattens out as the number of iterations increases.



The final set of parameters used for the ANN

* Activation = relu, default, this is fastest activation function
* Hidden layer sizes = 200,
* Tolerance = 0.0001, a higher value would have caused the loss to be higher, a lower value would have caused the model to run very long
* Solver = adam, default
* Maximum iterations = 10000, ensures training loss adhere to tolerance improvements at all times

With the above combinations, I was able to get the best results with accuracy, precision, and recall score for class 1 on training as well test data. After trying multiple combinations, I was able to get to a difference of ~4.8% between the train and test data accuracy of the model.

### **2.3** Performance Metrics: Comment and Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC\_AUC score, classification reports for each model.

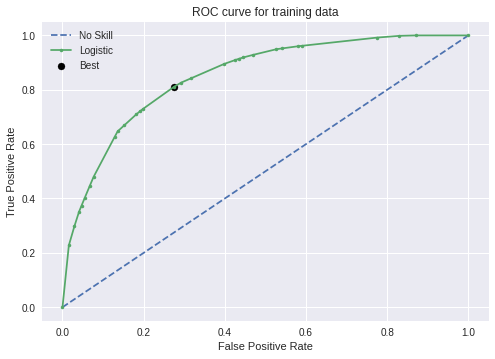
**Decision Tree**

After multiple iterations of gridsearch, I was able to determine the best possible combination of hyperparameters that gave an accuracy score of 79.9 % with train data and 77.9% with sample data. A difference of ~2% suggests a good performing model across both training and test data. Since the target variable is imbalanced it is resulting in lower accuracy scores than we would ideally hope for. The model is **neither overfitting nor underfitting** as the accuracy scores are very close. The precision has improved for the test data however, the recall and the f1-score have dropped, meaning type 2 error is high for test data.

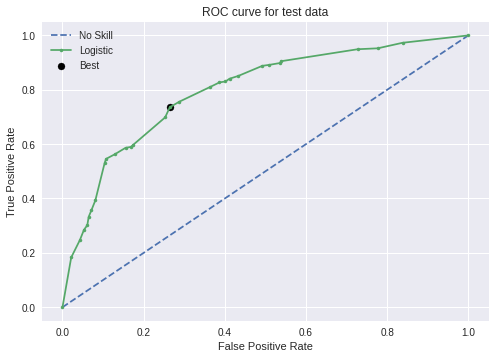
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Train data | Negative | Positive |  | Test data | Negative | Positive |
| Negative | 1271 | 200 |  | Negative | 540 | 65 |
| Positive | 222 | 407 |  | Positive | 134 | 161 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Train data | precision | recall | f1-score | support |
| Class 0 | 85% | 86% | 86% | 1471 |
| Class 1 | 67% | 65% | 66% | 629 |
| Accuracy |  |  | 79.9% | 2100 |
| Macro Average | 76% | 76% | 76% | 2100 |
| Weight Avg. | 80% | 80% | 80% | 2100 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test data | precision | recall | f1-score | support |
| Class 0 | 80% | 89% | 84% | 605 |
| Class 1 | 71% | 55% | 62% | 295 |
| Accuracy |  |  | 77.9% | 900 |
| Macro Average | 76% | 72% | 73% | 900 |
| Weight Avg. | 77% | 78% | 77% | 900 |



The area under the ROC curve for training data is 84.9%. Best Threshold=0.312883, G-Mean=0.767



The area under the ROC curve for test data is 79.7%. Best Threshold=0.270270, G-Mean=0.736

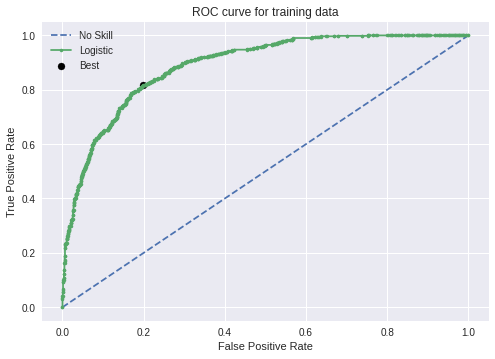
**Random Forest**

After multiple iterations of gridsearch, I was able to determine the best possible combination of hyperparameters that gave an accuracy score of 82.7% with train data and 77.2% with sample data. A difference of ~5.5% suggests a good performing model across both training and test data. Since the target variable is imbalanced it is resulting in lower accuracy scores than we would ideally hope for. The model is **slightly underfitting** as the accuracy scores have a gap of ~5% and precision, recall, and f1-scores have dropped, meaning the model’s performance on test data will be bad as compared to training data.

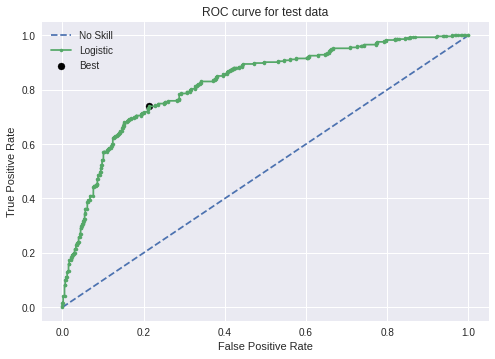
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Train data | Negative | Positive |  | Test data | Negative | Positive |
| Negative | 1346 | 125 |  | Negative | 552 | 52 |
| Positive | 238 | 391 |  | Positive | 152 | 143 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Train data | precision | recall | f1-score | support |
| Class 0 | 85% | 92% | 88% | 1471 |
| Class 1 | 76% | 62% | 68% | 629 |
| Accuracy |  |  | 82.7% | 2100 |
| Macro Average | 80% | 77% | 78% | 2100 |
| Weight Avg. | 82% | 83% | 82% | 2100 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test data | precision | recall | f1-score | support |
| Class 0 | 78% | 91% | 84% | 605 |
| Class 1 | 73% | 48% | 58% | 295 |
| Accuracy |  |  | 77.2% | 900 |
| Macro Average | 76% | 70% | 71% | 900 |
| Weight Avg. | 77% | 77% | 76% | 900 |



The area under the ROC curve for training data is 88.9%. Best Threshold=0.305293, G-Mean=0.809



The area under the ROC curve for test data is 82.2%. Best Threshold=0.266698, G-Mean=0.763

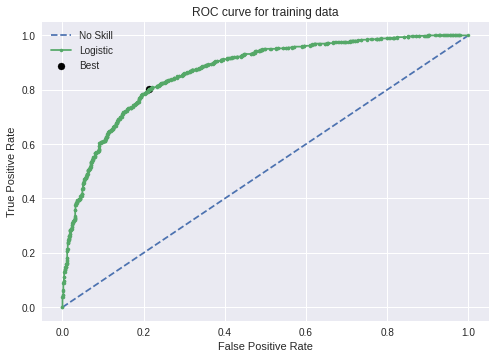
**Artificial Neural Network**

After multiple iterations of gridsearch, I was able to determine the best possible combination of hyperparameters that gave an accuracy score of 81.1% with train data and 76.3% with sample data. A difference of ~4.8% suggests a good performing model across both training and test data. Since the target variable is imbalanced it is resulting in lower accuracy scores than we would ideally hope for. The model is **slightly underfitting** as the accuracy scores have a gap of ~5% and precision, recall, and f1-score have dropped, meaning the model’s performance on test data will be bad as compared to training data.

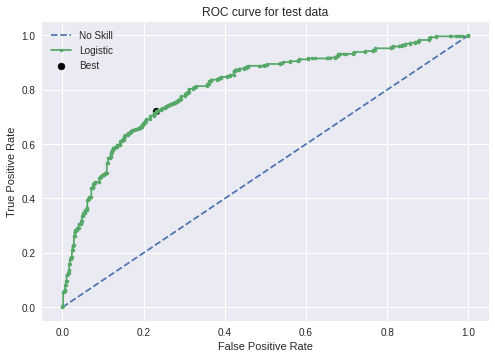
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Train data | Negative | Positive |  | Test data | Negative | Positive |
| Negative | 1316 | 155 |  | Negative | 544 | 61 |
| Positive | 241 | 388 |  | Positive | 152 | 143 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Train data | precision | recall | f1-score | support |
| Class 0 | 85% | 89% | 87% | 1471 |
| Class 1 | 71% | 62% | 66% | 629 |
| Accuracy |  |  | 81.1% | 2100 |
| Macro Average | 78% | 76% | 77% | 2100 |
| Weight Avg. | 81% | 81% | 81% | 2100 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test data | precision | recall | f1-score | support |
| Class 0 | 78% | 90% | 84% | 605 |
| Class 1 | 70% | 48% | 57% | 295 |
| Accuracy |  |  | 76.3% | 900 |
| Macro Average | 74% | 69% | 70% | 900 |
| Weight Avg. | 76% | 76% | 75% | 900 |

****

The area under the ROC curve for training data is 86.6%. Best Threshold=0.298683, G-Mean=0.795

****

The area under the ROC curve for test data is 80.8%. Best Threshold=0.261542, G-Mean=0.746

### **2.4** Final Model: Compare all the models and write an inference which model is best/optimized.

|  |  |  |  |
| --- | --- | --- | --- |
| Train Data | **Decision Tree** | **Random Forest** | **ANN** |
| Accuracy | **79.9%** | **82.7%** | **81.1%** |
| Precision | **67%** | **76%** | **71%** |
| Recall | **65%** | **62%** | **62%** |
| F1 score | **66%** | **68%** | **66%** |
| AUC ROC | **84.9%** | **88.9%** | **86.6%** |

|  |  |  |  |
| --- | --- | --- | --- |
| Test Data | **Decision Tree** | **Random Forest** | **ANN** |
| Accuracy | **77.9%** | **76.2%** | **76.3%** |
| Precision | **71%** | **73%** | **70%** |
| Recall | **55%** | **48%** | **48%** |
| F1 score | **62%** | **58%** | **57%** |
| AUC ROC | **79.7%** | **82.2%** | **80.8%** |

With the above comparison of performance metrics, we can see Random forest outperforms with training data. But with test data Decision tree does a better job especially with Recall and accuracy. The difference between train and test data accuracy is only 2% for the Decision tree. Both Random forest and ANN are slightly under-fitted.

The decision tree model, though does poor in precision metrics with the training data makes up for it in the test data and has the best recall score amongst the three models. It also has the best F1 score. With the training data, the F1 score for the decision tree was only slightly behind the Random forest model. Hence, the **Decision tree is the best-suited model for the problem at hand.**

[*https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9*](https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9)

### **2.5** Inference: Based on the whole Analysis, what are the business insights and recommendations

Following are insights and recommendations for the insurance company.

1. The target variable is imbalanced with only 30% of the data of class 0(Claimed = Yes). Having balanced data helps train the model better and predict which insurance transaction will see a claim filed

**Recommendation**: Improve the balance in the data

1. The available features in the data may not be sufficient. Secondary data required.

**Recommendation**: Include additional features like a month, year of travel, business travel vs pleasure, group travel vs individual travel, claim reasons, hotel rating, airlines, air ticket tier, country/city of origin, country/city of destination, multi-city travel, return ticket details, etc.

Gather secondary data sets like weather, to see if there were multiple flight cancellations due to weather, political incidents, etc.

1. Bookings from Agency Code C2B has almost half of the claims filed for

**Recommendation**: It may be worth checking their operations/compliance audits etc.

1. The pre-packaged plans Gold Plan, Silver Plan, Bronze Plan are statistically the worst-performing plans. ~40% of the Gold and Bronze Plans saw claims filed, and ~75% of Silver plans had claims filed. The agencies while selling these products are not doing a good job in explaining the plans to the customers or customers are buying these with some preconceived notions. This leading to customer dissatisfaction and the majority of customers filing for insurance claims

**Recommendation**: It may be worth revising the marketing strategy for these plans or even the structure of the plans.