greatlearning
Learning for Life

Model Solution

Data Mining- DSBA

PROPRIETARY

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

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.

A. Read the data and do exploratory data analysis. Describe the data briefly.

B. Do you think scaling is necessary for clustering in this case? Justify

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

D. Apply K-Means clustering on scaled data and determine optimum clusters. Apply elbow curve and silhouette score.

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

Dataset for Problem 1: bank_marketing_part1_Data.csv

Data Dictionary for Market Segmentation:

a. spending: Amount spent by the customer per month (in 1000s)

b. advance_payments: Amount paid by the customer in advance by cash (in 100s)

c. probability_of_full_payment: Probability of payment done in full by the customer to the bank

d. current_balance: Balance amount left in the account to make purchases (in 1000s)

e. credit_limit: Limit of the amount in credit card (10000s)

f. min_payment_amt: minimum paid by the customer while making payments for purchases made monthly (in 100s)

g. max_spent_in_single_shopping: Maximum amount spent in one purchase (in 1000s)

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

To start with the analysis lets look at the sample data, and perform basic checks.

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
0	19.94	16.92	0.88	6.68	3.76	3.25	6.55
1	15.99	14.89	0.91	5.36	3.58	3.34	5.14
2	18.95	16.42	0.88	6.25	3.76	3.37	6.15
3	10.83	12.96	0.81	5.28	2.64	5.18	5.18
4	17.99	15.86	0.90	5.89	3.69	2.07	5.84

Top 5 Rows of the Dataset

PROPRIETARY

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Inferences-

1. The dataset has a total of seven independent variables - All are continuous
2. Shape (dimension) of the Dataset is (210, 7).
3. No NULL values & duplicate values are present in the dataset.

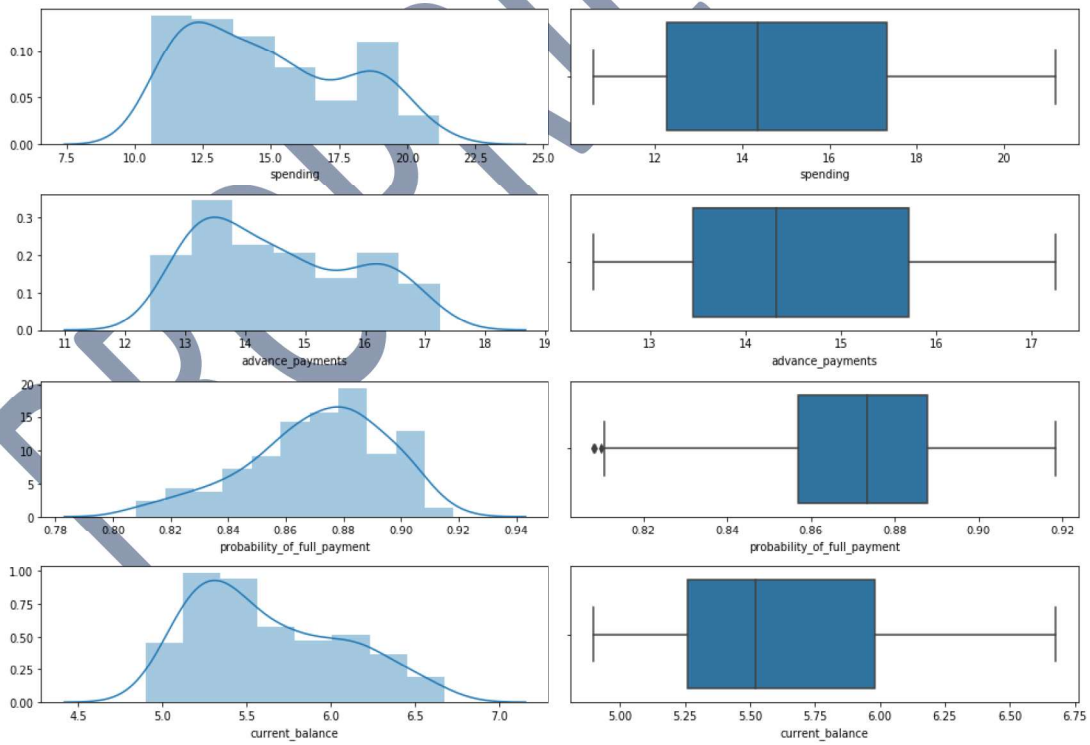
Univariate analysis -

To perform Univariate analysis on continuous variables, let us start with looking at the summary statistics of the dataset.

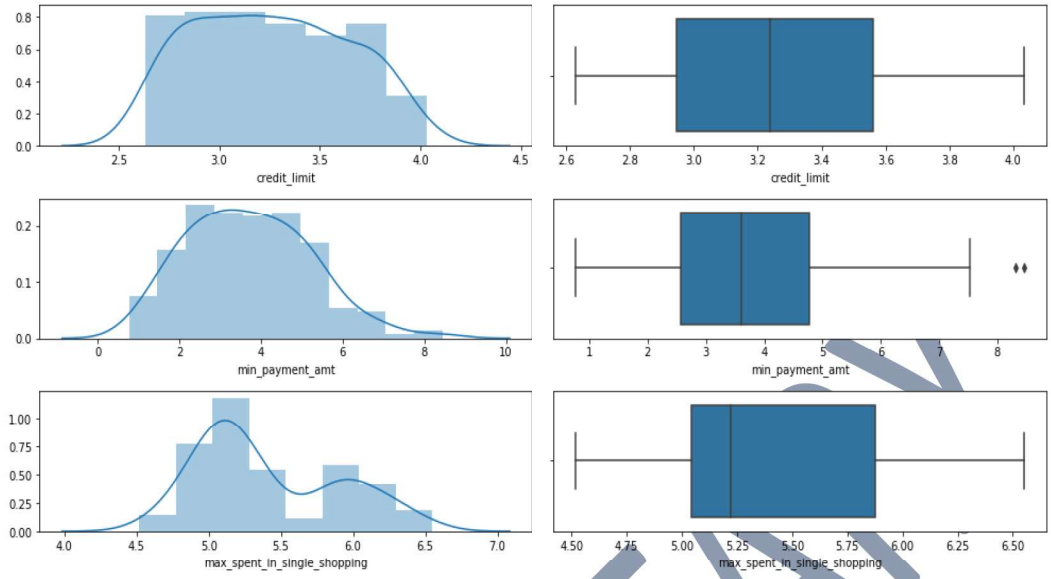
	count	mean	std	min	25%	50%	75%	max
spending	210.0	14.85	2.91	10.59	12.27	14.36	17.30	21.18
advance_payments	210.0	14.56	1.31	12.41	13.45	14.32	15.72	17.25
probability_of_full_payment	210.0	0.87	0.02	0.81	0.86	0.87	0.89	0.92
current_balance	210.0	5.63	0.44	4.90	5.26	5.52	5.98	6.68
credit_limit	210.0	3.26	0.38	2.63	2.94	3.24	3.56	4.03
min_payment_amt	210.0	3.70	1.50	0.77	2.56	3.60	4.77	8.46
max_spent_in_single_shopping	210.0	5.41	0.49	4.52	5.04	5.22	5.88	6.55

Summary Statistics of the Dataset

Box plots & Distribution plots –



Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.



Distribution & Box Plots for the variables

Percentage of outliers -

	Outlier%
spending	0.00
advance_payments	0.00
probability_of_full_payment	1.43
current_balance	0.00
credit_limit	0.00
min_payment_amt	0.95
max_spent_in_single_shopping	0.00

Only two columns have outlier values that too with less than 1.5 Percentage. For treating these outliers, we shall be using the flooring and capping technique in which the outlier value will be changed to the nearest whisker.

Note – *In the current dataset, the proportion of outliers is very small. Hence, replacement of the outliers by the outer limit of the whiskers is acceptable. However, if the proportion of outliers is very large, e.g. 20%, then alternative methods of treating them are required.*

Kurtosis & Skewness in Dataset –

	Kurtosis	Skewness
spending	-1.08	0.40
advance_payments	-1.11	0.39
probability_of_full_payment	-0.19	-0.52
current_balance	-0.79	0.53
credit_limit	-1.10	0.13
min_payment_amt	-0.22	0.36
max_spent_in_single_shopping	-0.84	0.56

PROPRIETARY

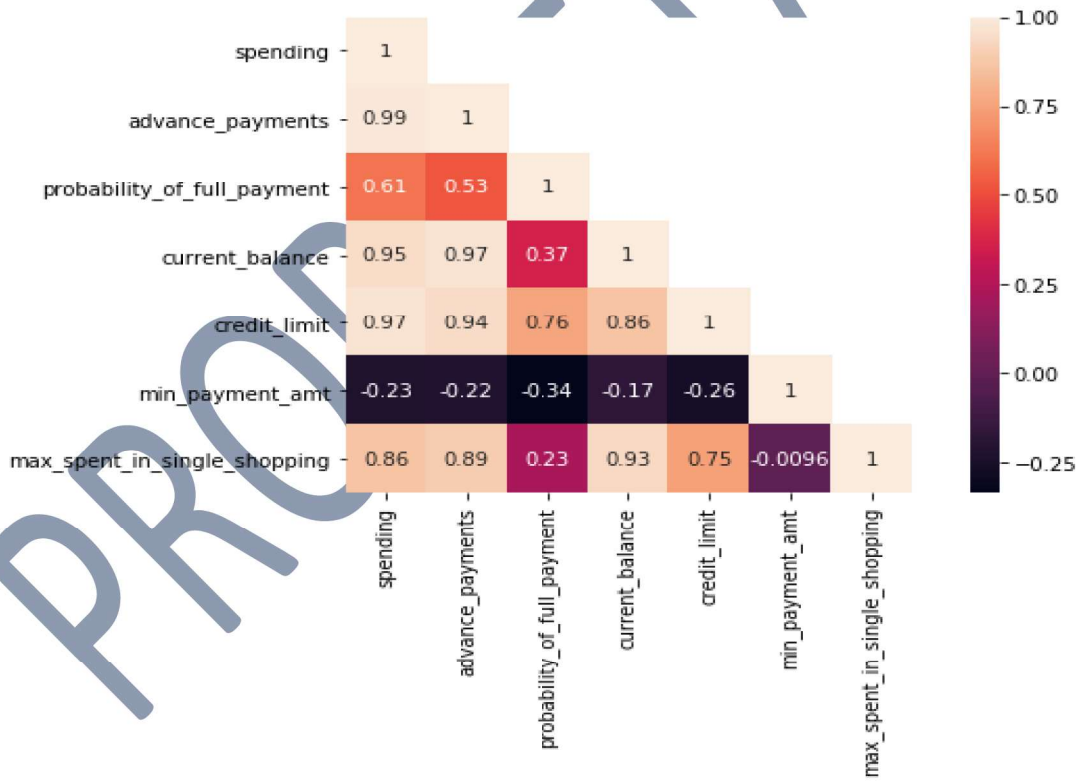
Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Inferences on Univariate analysis-

- From the Box plots we can conclude that there are few outliers present in min_payment_amt in the positive side and probability of full payment. The outliers were treated and were made equal to the whisker values i.e Q1 - 1.5* IQR or Q3 + 1.5* IQR whichever is nearer.
- Variables Spending, Advance_payments and max_spent_in_single_shopping have bi-modal distributions.
- Variables probability_of_full_payment & min_payment_amt have very low kurtosis values, near normal distributions.
- All variables except probability_of_full_payment have positive skewness i.e. to a longer on the left side of the distribution.
- All variables have negative kurtosis i.e. the distributions are flatter than a normal curve with the same mean and standard deviation.

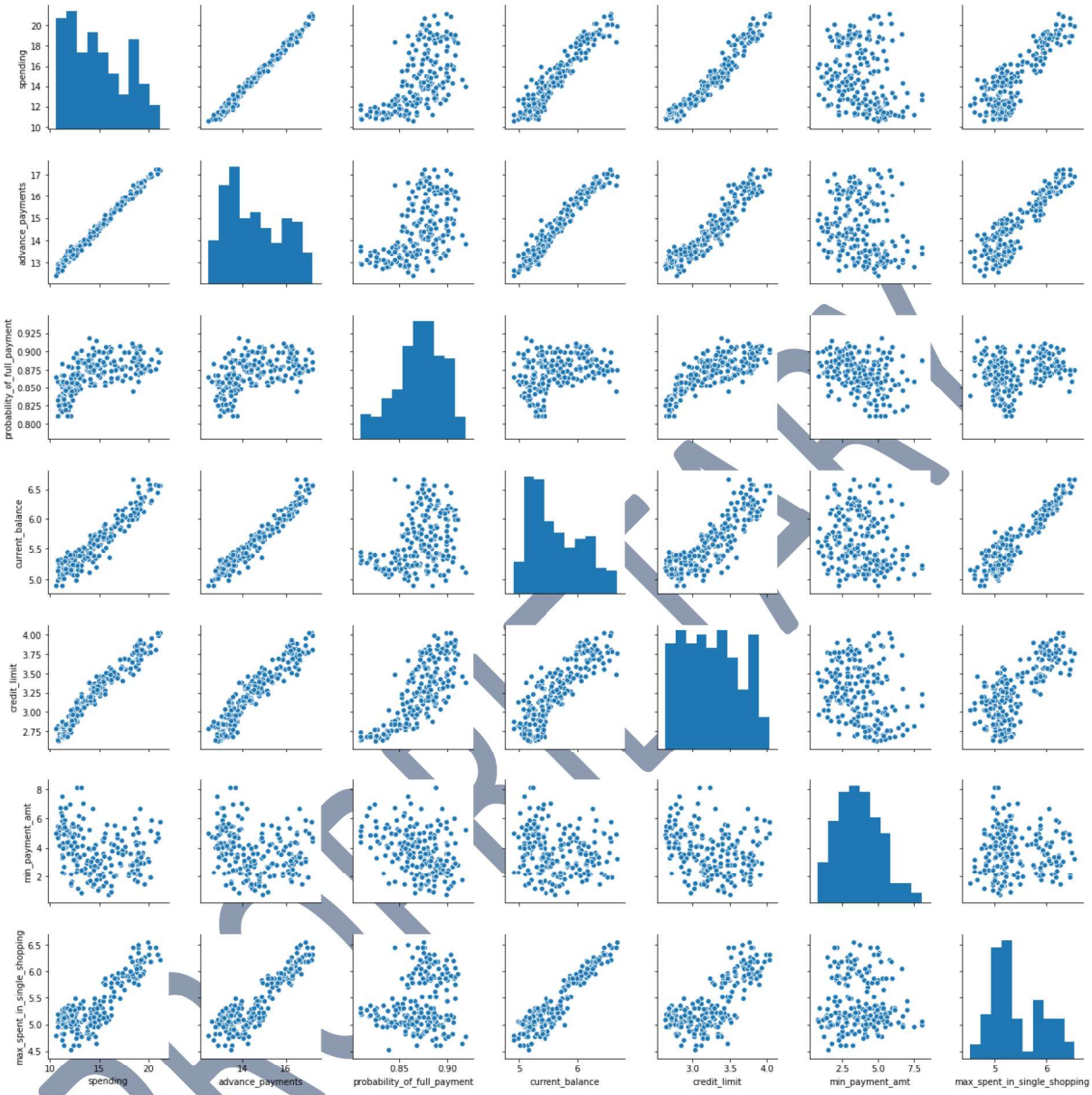
Multi-variate Analysis-

Analyzing the relationship among continuous variables by using Pair plot and Correlation Heatmap.



PROPRIETARY

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.



Inferences –

1. Top 3 Positive correlation pairs (Descending order of Correlation) - spending & advance_payments (0.99) , credit_limit & spending (0.97) , current balance & advance_payments (0.97)

2. Top 3 Negative correlation pairs (Descending order of Correlation) - probability_of_full_payment & min_payment_amount (-0.34) , min_payment_amount & credit_limit (-0.26) , min_payment_amount & spending (-0.23)

Note - *For practical purposes correlations in the range of [-0.4, 0.4] are not considered to be important.*

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

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

Scaling is required because most of the variables vary in magnitude; for example, advance_payments has a Std. Deviation = 1.31 while probability_of_full_payment has a Std. Deviation = 0.02. Central tendencies Mean & Median vary significantly as well.

For the current dataset, we will be using Min-Max scaler because the Variance of the variables vary a lot, and we would like to preserve that. Standard scaling on the other hand equates all variable means to 0 and Std. Deviations to 1 (converting all the variables to have unit variance).

However, for business insights and recommendation we will use the unscaled data due to better interpretability.

Summary Stats of the scaled data used for clustering –

	count	mean	std	min	25%	50%	75%	max
spending	210.0	0.40	0.27	0.0	0.16	0.36	0.63	1.0
advance_payments	210.0	0.44	0.27	0.0	0.21	0.39	0.68	1.0
probability_of_full_payment	210.0	0.56	0.22	0.0	0.43	0.58	0.72	1.0
current_balance	210.0	0.41	0.25	0.0	0.20	0.35	0.61	1.0
credit_limit	210.0	0.45	0.27	0.0	0.22	0.43	0.66	1.0
min_payment_amt	210.0	0.40	0.20	0.0	0.25	0.39	0.55	1.0
max_spent_in_single_shopping	210.0	0.44	0.24	0.0	0.26	0.35	0.67	1.0

Summary Statistics of the Scaled Data

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

Before training the model, let us discuss two major model parameters of hierarchical clustering (sklearn. AgglomerativeClustering) - Linkage and Metric.

Distance Metric-

The method used to calculate the distance between data points is specified using distance metric.

Types of Distance metrics are –

1. Euclidean Distance

The shortest distance between two points. For example, if x=(a,b) and y=(c,d), the Euclidean distance between x and y is $\sqrt{(a-c)^2+(b-d)^2}$

PROPRIETARY

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

2. Manhattan Distance

The distance between two points measured along axes at right angles. If x=(a,b) and y=(c,d), the Manhattan distance between x and y is $|a-c|+|b-d|$

3. Cosine Distance -

Cosine distance is a measure to find distance between two non-zero vectors of an inner product space. It is defined to equal the cosine of the angle between them.

Linkage -

The linkage criterion determines which distance to use between sets of observation. The algorithm will merge the pairs of cluster that minimize this criterion.

Linkage methods -

1. Single-Linkage

Single-linkage (nearest neighbor) is the shortest distance between a pair of observations in two clusters. It can sometimes produce clusters where observations in different clusters are closer together than to observations within their own clusters. These clusters can appear spread-out.

2. Complete-Linkage

Complete-linkage (farthest neighbor) is where distance is measured between the farthest pair of observations in two clusters. This method usually produces tighter clusters than single-linkage, but these tight clusters can end up very close together. Along with average-linkage, it is one of the more popular distance metrics.

3. Average-Linkage

Average-linkage is where the distance between each pair of observations in each cluster are added up and divided by the number of pairs to get an average inter-cluster distance. Average-linkage and complete-linkage are the two most popular distance metrics in hierarchical clustering.

4. Centroid-Linkage

PROPRIETARY

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

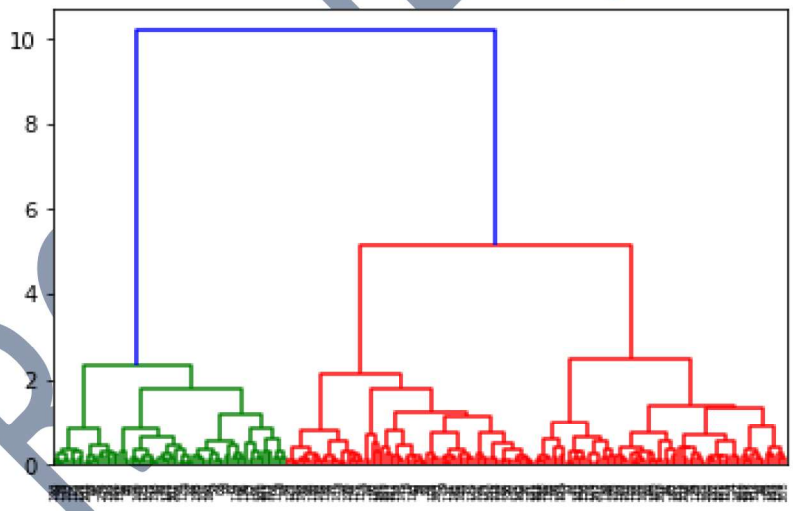
Centroid-linkage is the distance between the centroids of two clusters. As the centroids move with new observations, it is possible that the smaller clusters are more similar to the new larger cluster than to their individual clusters causing an inversion in the dendrogram. This problem doesn't arise in the other linkage methods because the clusters being merged will always be more similar to themselves than to the new larger cluster.

5. Ward Linkage -

Ward uses the Ward variance minimization algorithm. Ward linkage performs well even if noise is present between the clusters. We shall be using Ward linkage in the current study.

The choice of Linkage, and distance metric depends on the Distribution of the data. Thus for best results, the data must be thoroughly inspected and the most optimal linkage method, and metric must be chosen. For the current case study, let's use Ward Linkage method, and Euclidean distance metric as these are among the most commonly used techniques.

Let's plot the Dendrogram to visualize the probable clusters present in the dataset. We will not truncate the Dendrogram in the first go.



Steps to find the Optimum number of clusters -

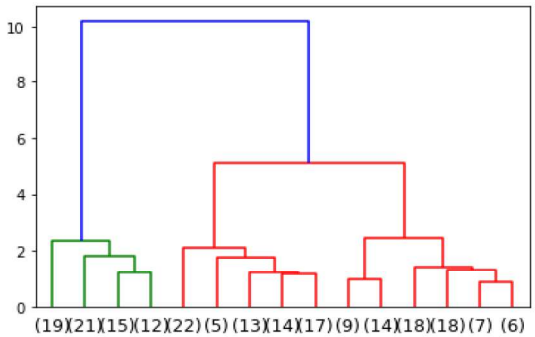
1. Determine the largest vertical distance that does not intersect any of the other clusters

2. Draw a horizontal line at both extremities

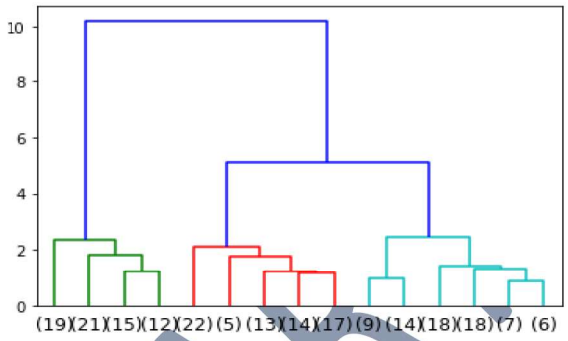
3. The optimal number of clusters is equal to the number of vertical lines going through the horizontal line

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

From the above Dendrogram, we can see two good candidates for optimum clusters at color threshold 8 and 4 mark (Y-axis). Let's plot a truncated Dendrogram with these two levels for better visualization.



2 Clusters @Color Threshold = 8



3 Clusters @Color Threshold = 4

For Color threshold 8, we get Two clusters whereas for 4 we get Three clusters in the dataset. Generally analyzing Two clusters is not very useful to the business hence we will choose Three clusters as the optimum. In situations where the business needs to make a Binary decision say whether to give a loyalty bonus or not, two clusters can be utilized.

Lets compare the Silhouette Scores for different clusters.

	Silh_Scores	Cluster Distribution
2	0.49	[143, 67]
3	0.39	[72, 67, 71]
4	0.3	[67, 49, 71, 23]
5	0.25	[71, 49, 48, 23, 19]
6	0.22	[48, 49, 49, 23, 19, 22]
7	0.22	[49, 49, 27, 23, 19, 22, 21]
8	0.23	[49, 44, 27, 23, 19, 22, 21, 5]
9	0.22	[31, 44, 27, 23, 19, 22, 21, 5, 18]

Index = Number of Clusters

As discussed earlier, choosing two as optimum clusters is not very beneficial for the business so we will be selecting the clusters with second best scores.

From the Silhouette scores we see that choosing Three as the optimum clusters is good as the Silhouette score is very high (second only after the Two clusters) and the distribution of data-points among the three clusters is fairly even as well.

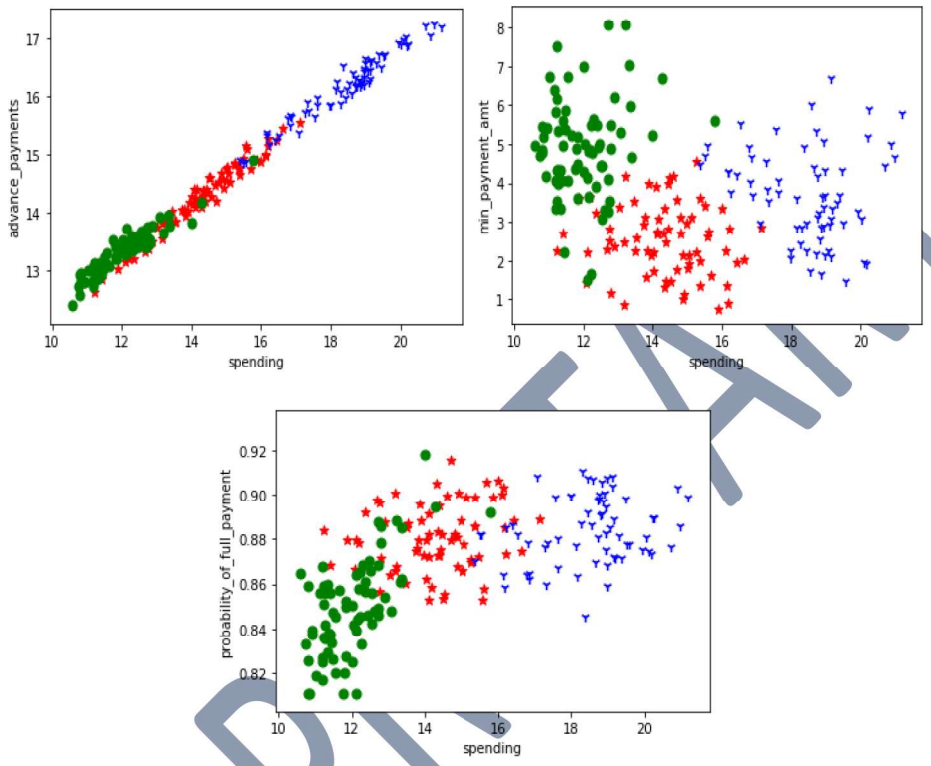
Lets proceed with building Hierarchical clustering model (Agglomerative Clustering) with Three clusters for labelling the Data points. Again, we will use the 'ward' linkage and Euclidean metric to find linkages.

PROPRIETARY

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Post the modelling, lets visualize the Three clusters using scatter plot between two variables. Different set of variables can lead to different set of inferences regarding segmentation, thus it can be helpful to visualize scatter plot for all the pairs.

Let's look at three variable pairs -



Scatter plots between two variables with cluster label as Hue element

Inferences from the above plots -

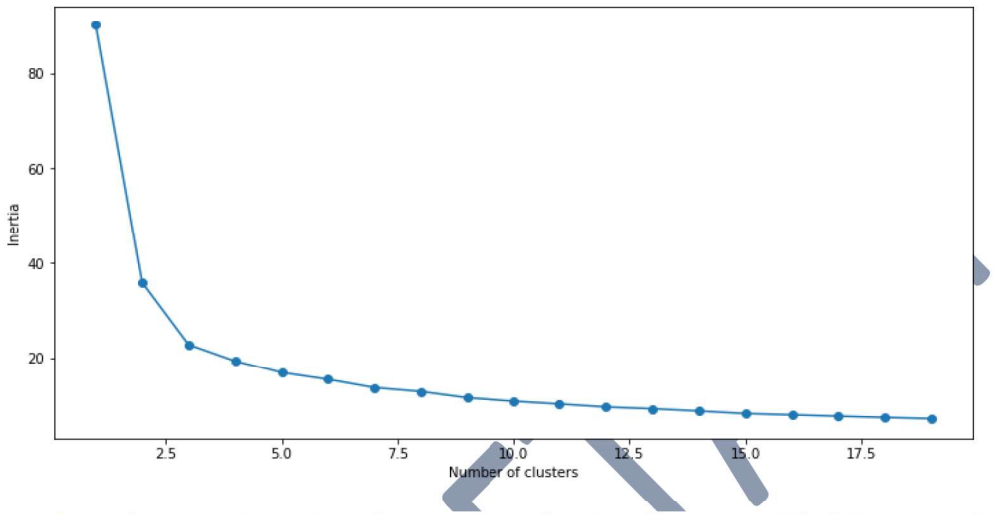
- We are able to achieve distinguishable Clusters among almost all the pairs.
- These plots can help in arriving at decisions when the business is interested in a particular pair of variables.
- A common order of Clusters is Green, Red and Blue. In most of the plots the Green cluster has low values on both the axes, whereas the Blue cluster has max values.

  Detailed description of the Three Clusters in the subsequent sections.

1.4 Apply K-Means clustering on scaled data and determine optimum clusters. Apply elbow curve and silhouette score. Interpret the inferences from the model.

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

We will use the KMeans class from sklearn to perform K-Means clustering. As the KMeans model needs us to predefine the 'K' value, so we will need to find the optimum k value by using an elbow curve and silhoutte scores for a range of k values. We will use the default parameters for the model.



To determine the optimal number of clusters, we have to select the value of k at the “elbow” ie the point after which the distortion/inertia starts decreasing in a linear fashion. Thus for the given data, we conclude that the optimal number of clusters is 3.

Lets also check the Silhouette scores.

	Silh_Scores	Cluster Distribution
2	0.5	[133, 77]
3	0.42	[77, 64, 69]
4	0.34	[63, 46, 51, 50]
5	0.3	[51, 27, 48, 48, 36]
6	0.28	[51, 27, 48, 22, 36, 26]
7	0.28	[16, 25, 24, 47, 41, 27, 30]
8	0.28	[37, 12, 44, 36, 23, 22, 16, 20]
9	0.26	[15, 24, 24, 33, 15, 30, 24, 21, 24]
10	0.27	[24, 35, 21, 28, 15, 22, 25, 17, 15, 8]
11	0.26	[34, 9, 22, 20, 24, 22, 15, 16, 14, 18, 16]
12	0.26	[23, 9, 22, 20, 24, 22, 14, 12, 14, 18, 16, 16]
13	0.24	[16, 18, 20, 15, 12, 17, 26, 13, 21, 6, 24, 7,...
14	0.25	[15, 18, 21, 18, 11, 17, 14, 12, 17, 6, 27, 7,...
15	0.25	[6, 21, 21, 16, 14, 13, 18, 10, 18, 17, 18, 16...
16	0.25	[6, 21, 16, 16, 14, 6, 18, 10, 15, 20, 18, 16,...
17	0.25	[6, 21, 16, 15, 14, 6, 16, 10, 15, 20, 18, 16,...
18	0.24	[6, 17, 16, 15, 13, 6, 16, 10, 15, 20, 17, 14,...
19	0.25	[16, 6, 12, 18, 14, 11, 13, 9, 7, 17, 7, 16, 1...

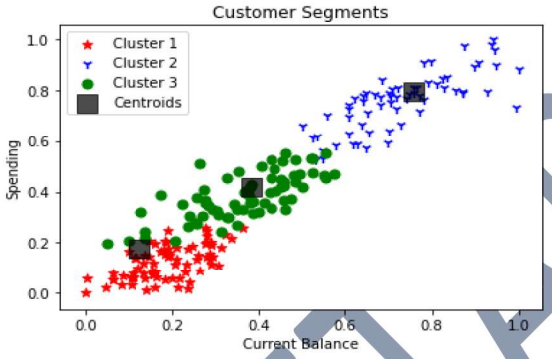
Index = Number of Clusters

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

As discussed earlier, choosing two as optimum clusters is not very beneficial for the business so we will be selecting the clusters with second best scores. Lets build the final K Means model with No of clusters as 3.

Visualizing the Clustering results –

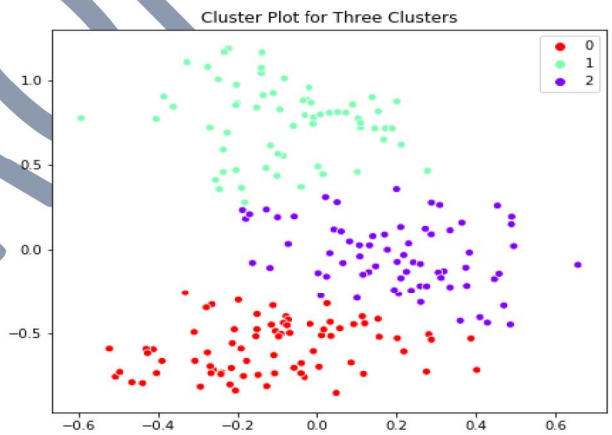
We can visualize the clusters by plotting a 2D scatter plot between two variables, and using the cluster labels as Hue parameter.



There were only 7 variables used to Cluster the data in this problem. However, in other cases there may be considerably more number of variables. Hence, it will not be possible to look at all pairs of variables to check proper separation among the clusters.

Often it is a good idea to create PCs and look at data point separations on PC axes.

In the plot shown below, the points are shown in a Scatter Plot where x-axis is PC1 and y-axis is PC2 , and Hue as Cluster labels.



Cluster Plot using the two PC's

The above plots shows three separated clusters, which implies that the clustering performed was appropriate.

PROPRIETARY

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

1.5 Describe cluster profiles for the clusters defined (2.5 pts). Recommend different promotional strategies for different clusters in context to the business problem in-hand (2.5 pts)

Let's use the results from KMeans clustering, and analyze the mean of the variables to describe the clusters –

Labels_KMeans	0	1	2
spending	11.90	18.61	14.65
advance_payments	13.26	16.25	14.44
probability_of_full_payment	0.85	0.88	0.88
current_balance	5.23	6.20	5.55
credit_limit	2.86	3.71	3.29
min_payment_amt	4.59	3.59	2.80
max_spent_in_single_shopping	5.09	6.06	5.17
Labels_Agg	1.74	0.97	0.19

Cluster Description –

Cluster membership of each data point in agglomerative and k-means is not identical; but overall characteristics of the clusters are the same.

From the above Data Frame we observe that Mean of Labels_Agg for the Three Clusters 0, 1 and 2 (obtained from K Means) are 1.74, 0.97 & 0.19 respectively. Thus indicating the following relationship.

Cluster 0 of K Means = Cluster 2 of Agg. Clustering
Cluster 1 of K Means = Cluster 1 of Agg. Clustering
Cluster 2 of K Means = Cluster 0 of Agg. Clustering

Cluster 0 - High Spending, High Advance Payments, High Probability of Full Payment, High Current Balance, High Credit Limit, Medium Min Payment Amount, High Max spent in a Single Shopping.
(Cluster 2 in Agg. Clustering)

Cluster 1 - Medium Spending, Medium Advance Payments, Low Probability of Full Payment, Medium Current Balance , Medium Credit Limit, Low Min Payment Amount , Medium Max spent in a Single Shopping. (Cluster 1 in Agg. Clustering)

Cluster 2 - Low Spending, Low Advance Payments, Medium Probability of Full Payment, Low Current Balance , Low Credit Limit, High Min Payment Amount , Low Max spent in a Single Shopping.
(Cluster 0 in Agg. Clustering)

PROPRIETARY

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Recommendations -

- **Cluster 0** customers are clearly High spenders thus very valuable to the business. The business should target to make them a loyal customer & hold them for maximum duration, continuing to use services from the Bank.
- **Cluster 1** customers should be the target of an Exclusive cashback/reward program on purchases to try to shift them to the Ideal customers cluster (i.e. cluster 0).
- **Cluster 3** customers have shown a bit reluctance to use services, the high value of min_payment_amt could imply that these customers use the services only for some particular purchases. Also, these could be the customers who are using this Bank as a secondary banking provider. Efforts must be made to convert them to frequent users.

Problem 2: CART-RF-ANN

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 provide recommendations to management. Use CART, RF & ANN and compare the models' performances in train and test sets.

2.1 Data Ingestion: Read the dataset. Do the descriptive statistics and do null value condition check, write an inference on it.

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

2.3 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score for each model

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

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

Dataset for Problem 2: insurance_part2_data-1.csv

Attribute Information:

1. Target: Claim Status (Claimed)

2. Code of tour firm (Agency_Code)

3. Type of tour insurance firms (Type)

4. Distribution channel of the tour insurance agencies (Channel).

5. Name of the tour insurance products (Product)

6. Duration of the tour (Duration)

PROPRIETARY

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

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 Data Ingestion: Read the data, do the necessary initial steps and exploratory data analysis (Univariate, Bi-variate and multivariate analysis)

To start with the analysis lets look at the sample data, and perform basic checks.

	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
0	48	C2B	Airlines	No	0.70	Online	7	2.51	Customised Plan	ASIA
1	36	EPX	Travel Agency	No	0.00	Online	34	20.00	Customised Plan	ASIA
2	39	CWT	Travel Agency	No	5.94	Online	3	9.90	Customised Plan	Americas
3	36	EPX	Travel Agency	No	0.00	Online	4	26.00	Cancellation Plan	ASIA
4	33	JZI	Airlines	No	6.30	Online	53	18.00	Bronze Plan	ASIA

Sample Data

	count	mean	std	min	25%	50%	75%	max
Age	3000.0	38.09	10.46	8.0	32.0	36.00	42.00	84.00
Commision	3000.0	14.53	25.48	0.0	0.0	4.63	17.24	210.21
Duration	3000.0	70.00	134.05	-1.0	11.0	26.50	63.00	4580.00
Sales	3000.0	60.25	70.73	0.0	20.0	33.00	69.00	539.00

	count	unique	top	freq
Agency_Code	3000	4	EPX	1365
Type	3000	2	Travel Agency	1837
Claimed	3000	2	No	2076
Channel	3000	2	Online	2954
Product Name	3000	5	Customised Plan	1136
Destination	3000	3	ASIA	2465

Summary Stats on Continuous & Categorical Features

Dependent variable - 'Claimed'

Independent variables - 'Age', 'Agency_Code', 'Type', 'Commision', 'Channel', 'Duration', 'Sales', 'Product Name', 'Destination'

Inferences-

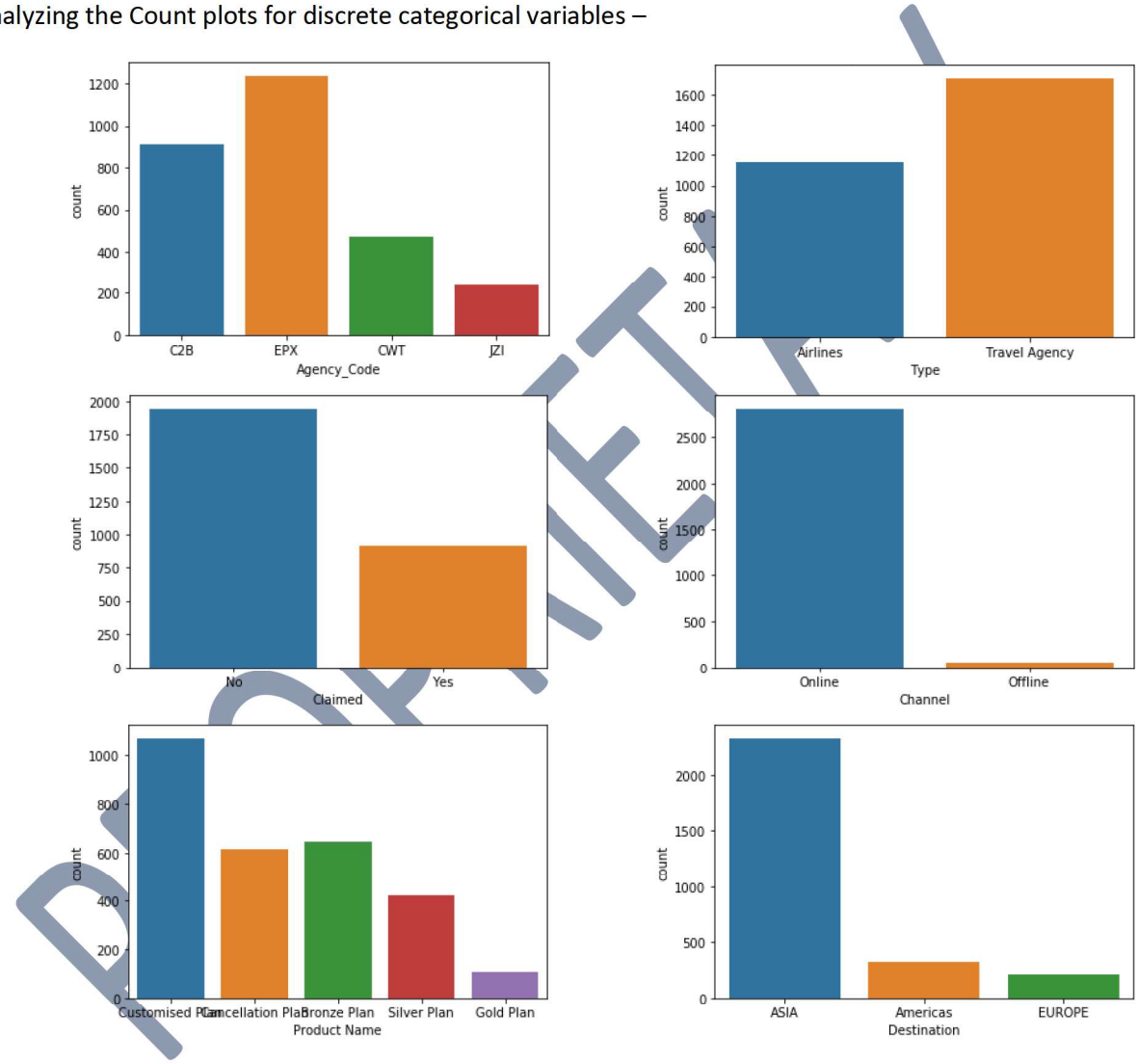
- The dependent variable (target) "Claimed" has two categories Yes (count = 924) and No (count = 2076). The minority class is approx. 45% of the majority class so the variable can be considered as balanced. A percentage of 20% or less can be treated as an unbalanced dataset.
- The dataset has a total of Nine independent variable - Four continuous (Age, Commission ,Duration, Sales) and Five discrete (Agency_Code ,Type ,Channel ,Product Name ,Destination)

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

- Shape of the Dataset is (3000, 10).
- No NULL values present in the dataset.
- There are 139 duplicate rows out of a total of 3000 rows. Duplicate rows dropped for further analysis.

Univariate analysis -

Analyzing the Count plots for discrete categorical variables –



Count Plots for the Categorical variables

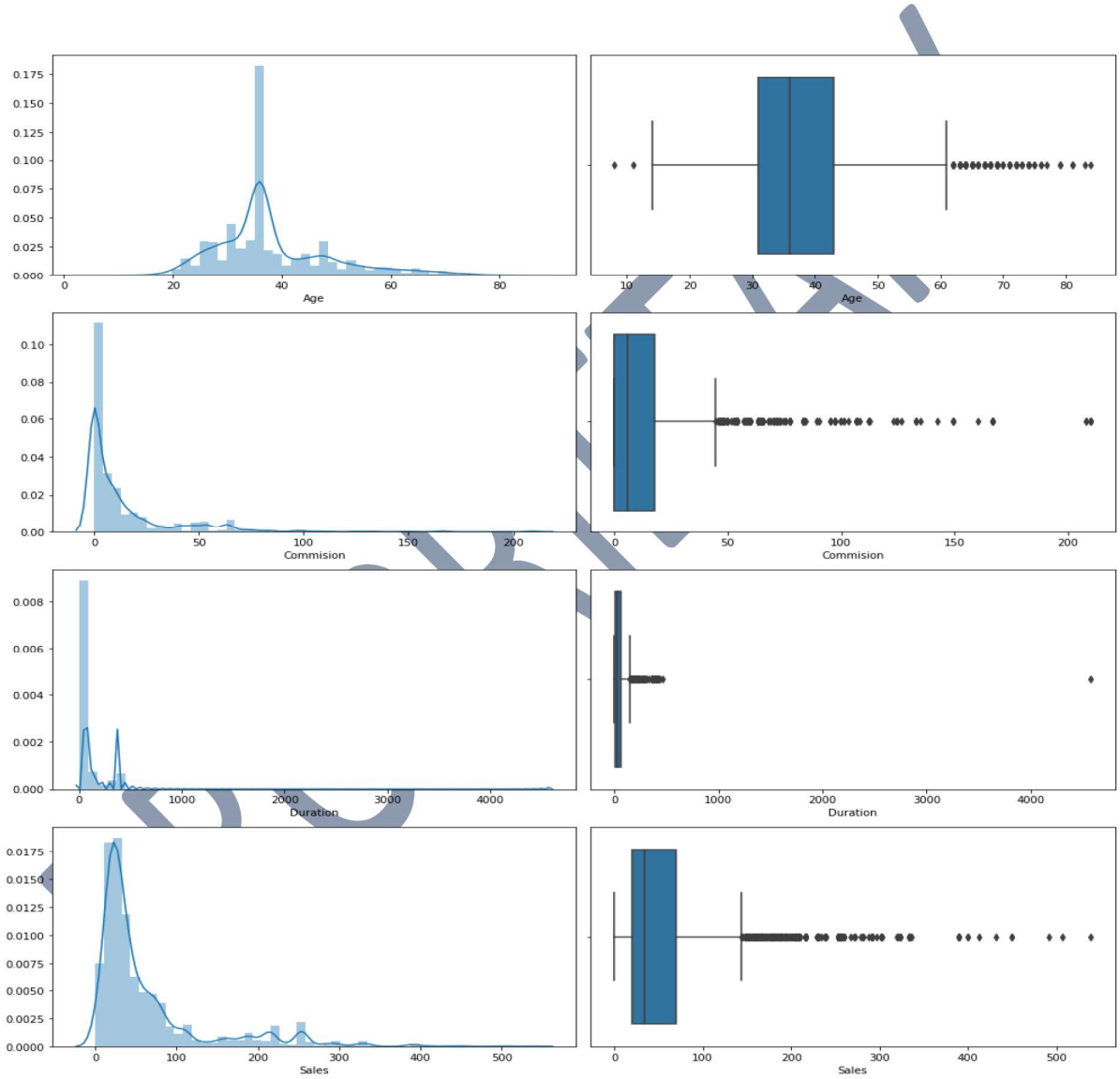
Categories in the variables (in ascending order of count) -

- Destination: Europe < Americas < Asia. Asia is the most popular destination.
- Product Name: Gold Plan < Silver Plan < Cancellation < Bronze Plan < Customized. Majority of the customers choose customized or Bronze plans. We observe significant number of cancellations as well (Approx. 20%).

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

- Channel: Offline < Online A very few customers used offline channel.
- Claimed: Yes < No. Claim percentage is very high approx. 30% of the total observations.
- Agency Code: JZI < CWT < C2B < EPX. Most customers used EPX agency.
- Type: Airlines < Travel Agency.

Analysing the Box plots and Distribution plots for Continuous variables –



Distribution & Box Plots for the Continuous variables

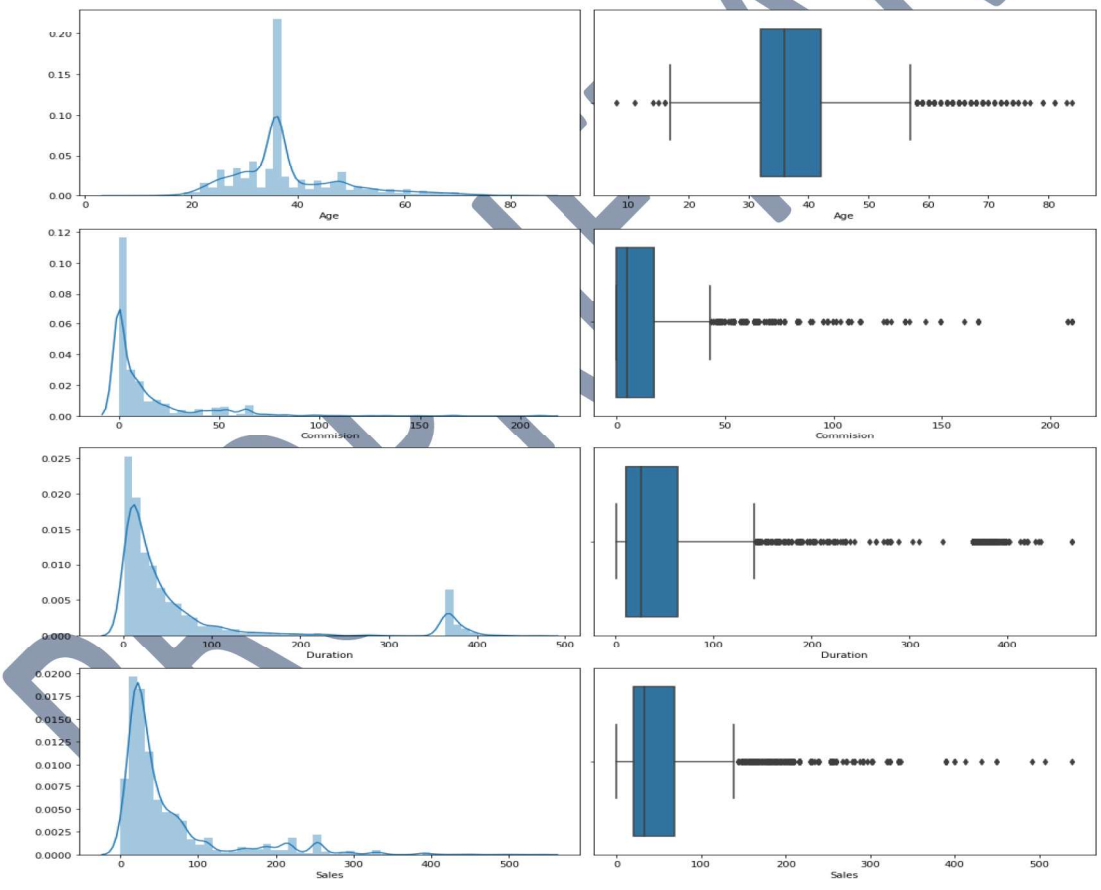
On closer inspection of the variable Duration, we observe that there is one observation with a negative value(-1) and two rows have 0 duration. In general, we would never expect a variable in Time units to have a value less than or equal to zero so this can be considered as an anomaly unless specified otherwise

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

by the responsible team. In addition, the duration boxplot clearly depicts that the observation 4580 is way beyond the normal data points, the next value being close to 500. This indicates that the value 4580 could possibly be generated due to Data entry issues. For the analysis, let's replace the -1 & 0 Duration with 1, and 4580 with the next smallest value (i.e. 466).

Note – These values are outright anomaly thus we are treating them before outlier detection is perfromed on the dataset.

This will be a very good place to note that the Data may not always be perfect, despite performing checks like outliers detection, or Null Value detection few anomalies might go undetected. Thus, apart from performing the Run of the Mill checks, one must be very vigilant to find such nuances. In addition, domain knowledge comes in handy in scenarios like these. Depending upon the context, the decision to handle such observations may change.



Distributions after treating the anomalies

Post removing the anomalies, we achieve a distribution on a good scale for the Duration variable.

Percentage of Outliers –

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

	Outliers %
Age	4.54
Duration	12.65
Sales	12.09
Commision	12.37

Duration variable has the max number of outliers with approximately 13% outlier values. CART and Random Forrest algorithms are unaffected by presence of outliers whereas ANN model is affected only when the percentage is greater than 15% as per the prevalent thumb rule. Hence, no treatment of outliers will be carried out for the current dataset.

Kurtosis & Skewness in Dataset –

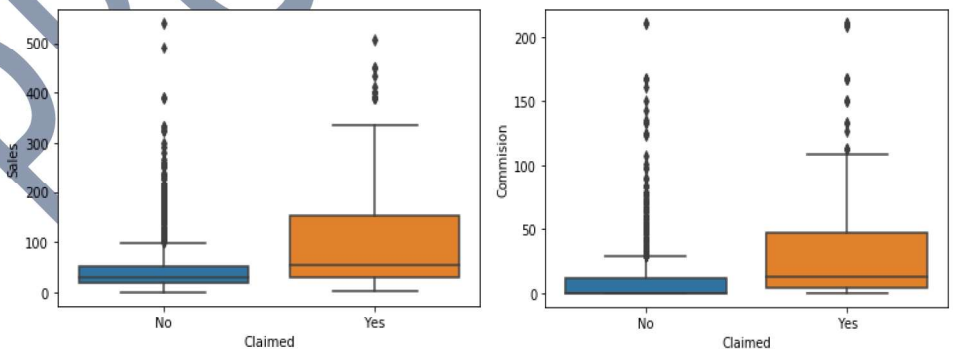
	Kurtosis	Skewness
Age	1.44	1.10
Commision	13.59	3.10
Duration	3.48	2.19
Sales	5.97	2.34

Inferences on Distribution of Continuous variables-

- All variables have of outlier values (observed as points beyond the whiskers in the Box Plots).
- Variable Age has multimodal distribution, composed of three distributions with the means at around 25, 38 and 50 years mark(approximation).
- Variables Commission, Duration and Sales have highly skewed (positively) distribution.
- All variables have Positive Kurtosis i.e. the distributions are peaked and much of the values are located in the tails of the distribution rather than around the mean.
- There are some instances where Sales, Commission values are zero.

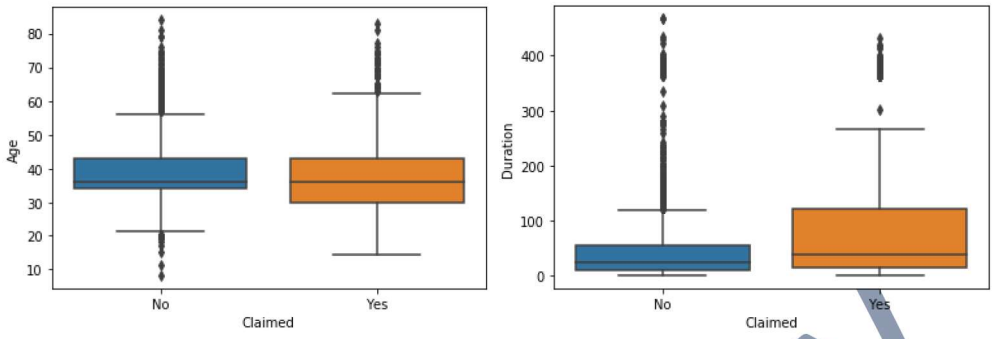
Bi-variate Analysis-

For the Bi-variate analysis of Target vs continuous variables, we shall use Box Plots.



PROPRIETARY

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.



Box Plots of Target variable vs Continuous variables

	Claimed	No	Yes
Age	**std**	10.244338	11.534705
	median	36.000000	36.000000
Commision	**std**	20.268408	32.351966
	median	0.000000	12.250000
Duration	**std**	78.348101	142.254581
	median	24.000000	38.000000
Sales	**std**	50.743972	93.238267
	median	29.000000	54.500000

Std. Deviation and Median Values grouped by Target variable 'Claim'

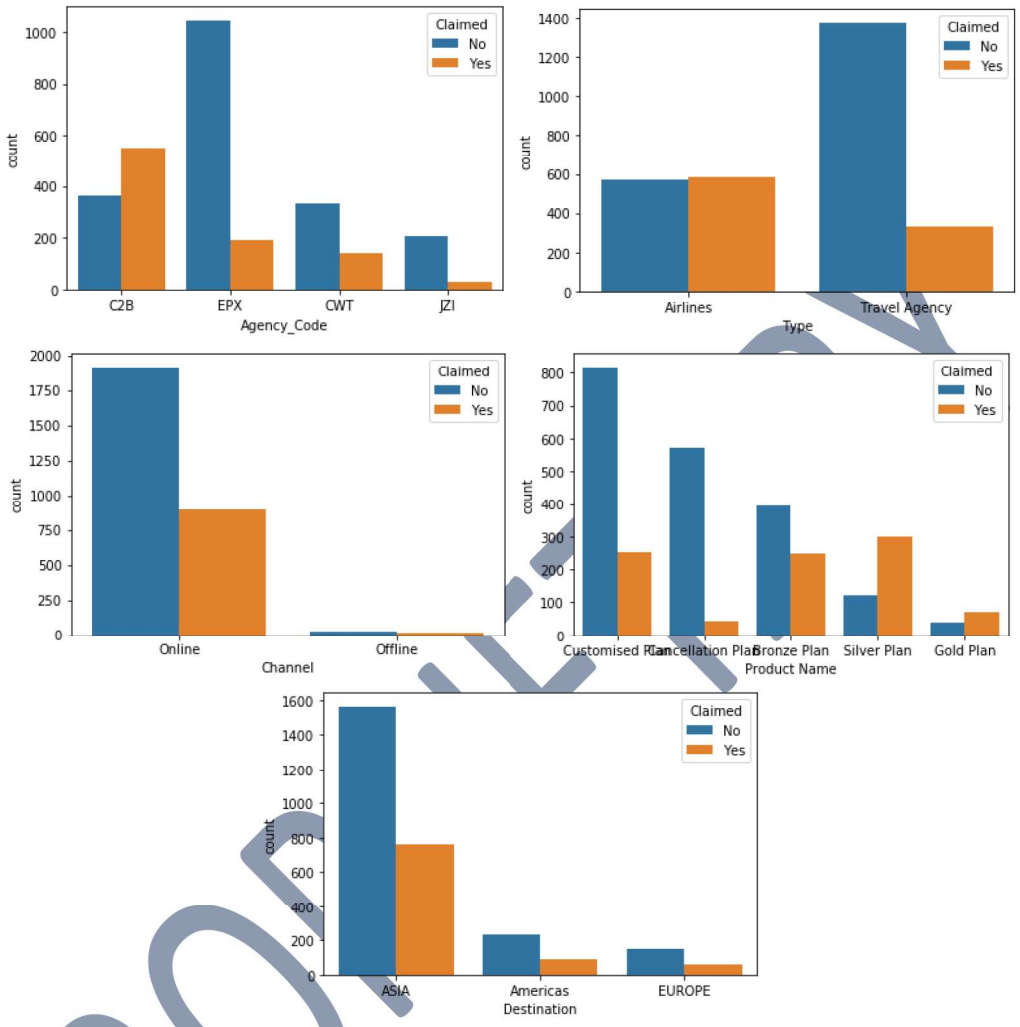
Inferences-

- Customers who have claimed for Insurance have similar maximum and minimum values compared to those who did not claim across the four continuous variables.
- Median ages are same for both Yes and No Claimant 1. Customers who have claimed for Insurance have similar maximum and minimum values compared to those who did not claim.
- IQR of variables commission, duration & sales is higher for Yes compared to the and No Claim group.
- Median ages are same for both Yes and No Claim groups. Std Dev is very similar as well.
- People who have claimed have higher median commission, duration & sales compared to those who did not claim.

For analysing relationship between Categorical variables & Target variable, we shall use Count Plots with Target Variable as Hue element.

PROPRIETARY

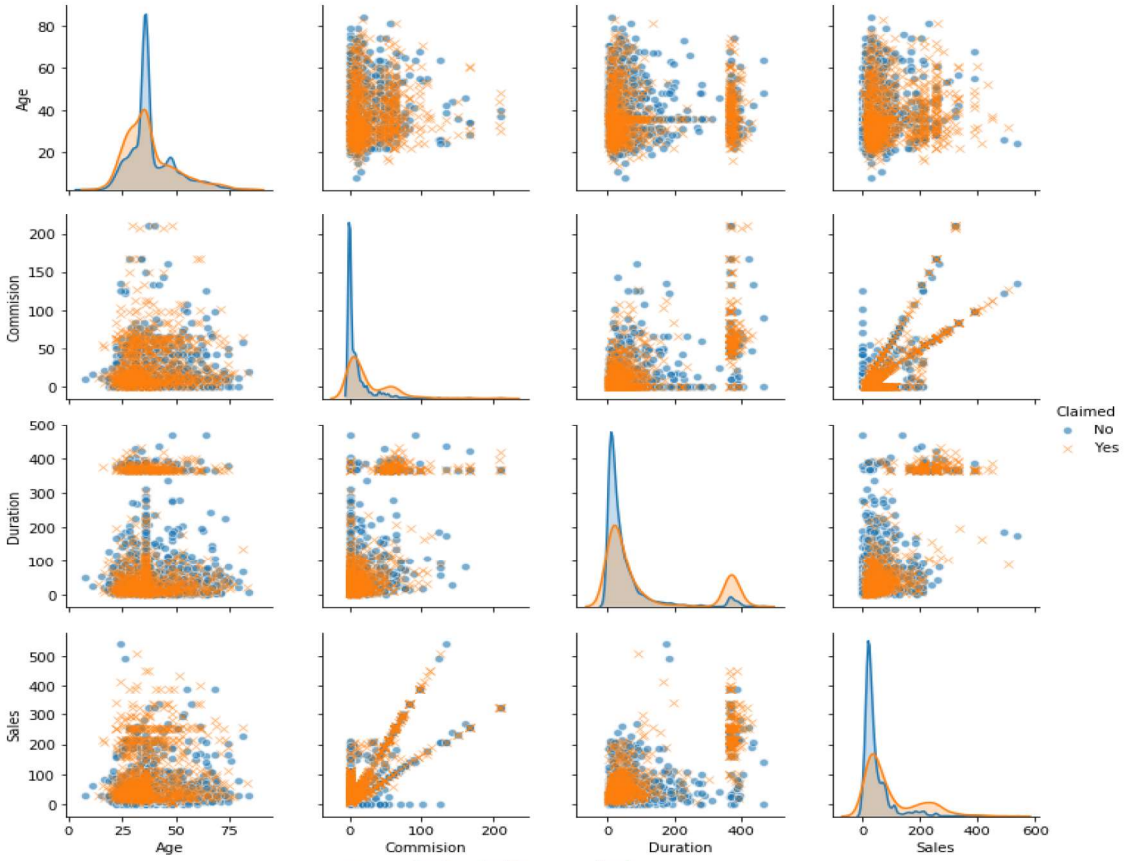
Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.



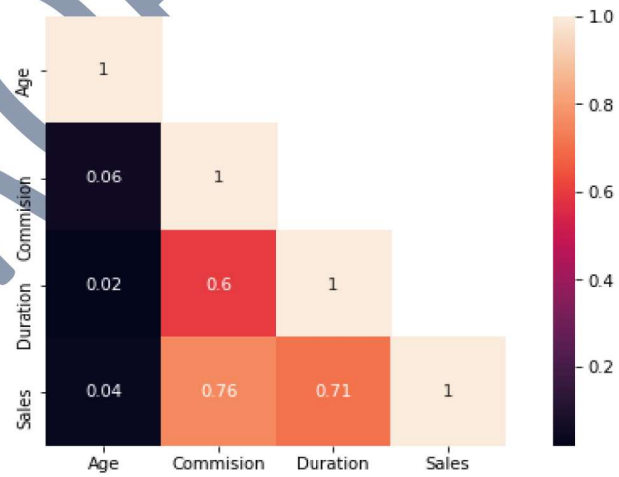
Count Plots of Categorical variables with Target as Hue element

Analysing the relationship among continuous variables by using Pair plot and Correlation Heat map-

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.



Pairplot of the dataset with Target as Hue element



Correlation Heat map for the Dataset

Inferences -

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

- Variables Sales and Commission have the highest positive correlation (0.76), followed by Sales and Duration (0.71), Duration and Commission (0.60).
- Rest of the variable pairs have close to zero correlations.
- No additional insights on adding the Target variable as a Hue argument in the Pair plot.

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

Data Split: We will be using a 70:30 Train Test split for the current case study.

CART Model -

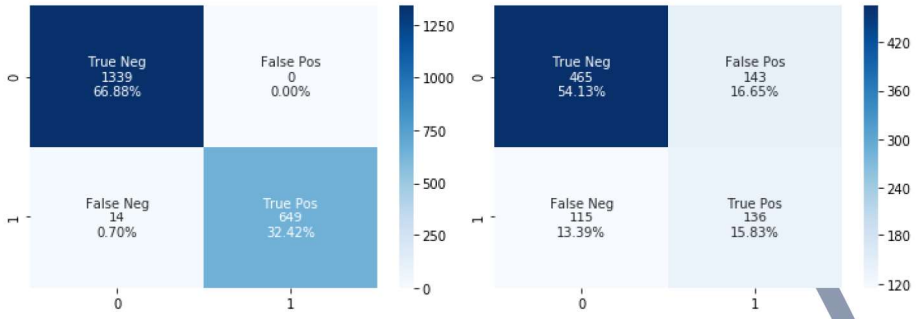
For the first iteration, let us build the CART model with default Hyperparameters. Depending upon the nature of fit i.e. over fit or under fit, we will go ahead with Hyperparameter tuning to arrive at better-fitted model.

Results from the first iteration –

```
Accuracy on Training Set is : 0.99

Accuracy on Test Set is : 0.7

Train set ROC-AUC score is : 1.0

Test set ROC-AUC score is : 0.66


Classification report for train set -
              precision    recall  f1-score   support

          No       0.99      1.00      0.99      1339
         Yes       1.00      0.98      0.99       663

    accuracy                           0.99      2002
   macro avg       0.99      0.99      0.99      2002
weighted avg       0.99      0.99      0.99      2002


Classification report for test set -
              precision    recall  f1-score   support

          No       0.80      0.76      0.78       608
         Yes       0.49      0.54      0.51       251

    accuracy                           0.70       859
   macro avg       0.64      0.65      0.65       859
weighted avg       0.71      0.70      0.70       859
```

PROPRIETARY

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.



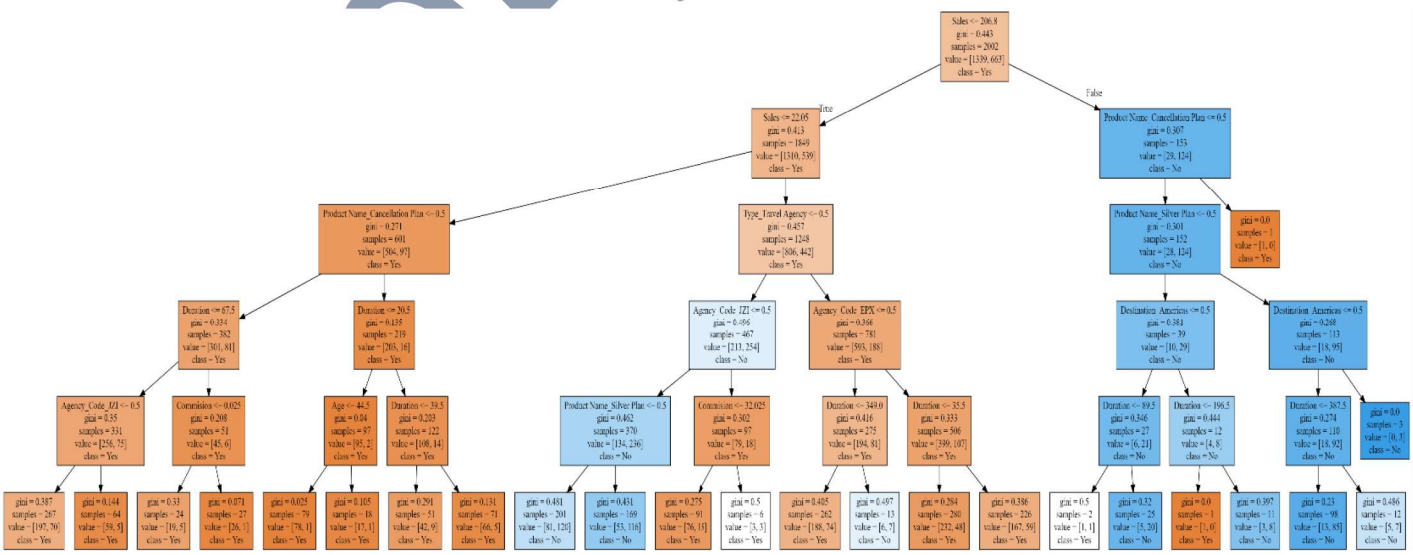
Confusion Matrices for Train & Test Set

Negative Class = No, Positive Class = Yes

From the above results it is evident that the model has clearly over-fit (Accuracy on Train is ~1 and Test is ~0.70). Now we must choose the set of Hyperparameters that will prune the decision tree i.e. prevent it from over-fitting the train data.

Hyperparameters chosen for tuning-

- Max_Depth : This parameters controls the depth a DT can grow to, by default this parameter is set to None (the tree grows until each leaf node has only one observation).
- Min_samples_split: The minimum number of samples required to split an internal node, greater is this value, lesser overfit the model would be.
- Max_features : Max features to be considered for each split. As a thumb rule, a good value is the log or square root of total features.
- Criterion: Function to consider for the split.



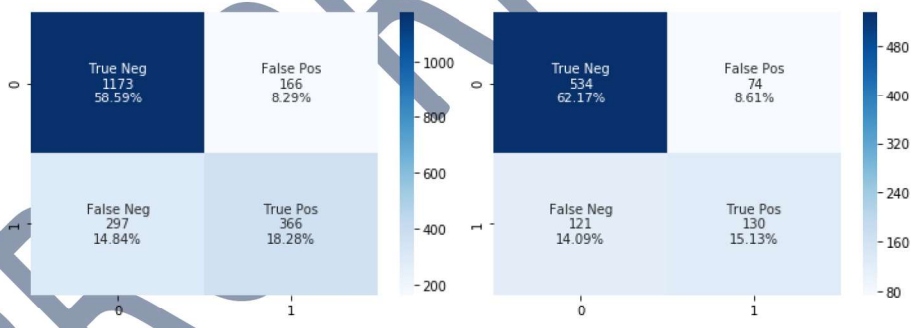
Decision Tree plotted using tree.plot_tree from sklearn

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Note – Categorical variables were One Hot encoded so the decision looks like Type_Travel_Agency<=0.5.

Results from the Tuned Model –

```
Train Accuracy is : 0.77

Test Accuracy is : 0.77

Train Set ROC-AUC score is : 0.78

Test Set ROC-AUC score is : 0.79


Classification report for train set -
              precision    recall  f1-score   support

          No       0.80      0.88      0.84      1339
         Yes       0.69      0.55      0.61       663

    accuracy                           0.77      2002
   macro avg       0.74      0.71      0.72      2002
weighted avg       0.76      0.77      0.76      2002


Classification report for test set -
              precision    recall  f1-score   support

          No       0.82      0.88      0.85       608
         Yes       0.64      0.52      0.57       251

    accuracy                           0.77       859
   macro avg       0.73      0.70      0.71       859
weighted avg       0.76      0.77      0.77       859
```

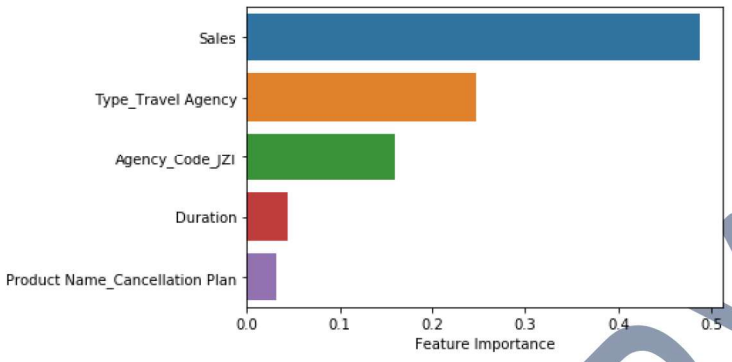


Confusion Matrices for Train & Test Set

Negative Class = No, Positive Class = Yes

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Feature Importance from tuned CART model -



Top 5 Features from the Tuned CART model

Inferences -

- The best parameters are - {'criterion': 'gini', 'max_depth': 5, 'max_features': 'log2', 'min_samples_split': 2, 'random_state': 100}
- From the results it is evident that we have greatly reduced the overfit of the CART model, with almost each metric performing similarly on train and test set.
- We achieved an accuracy of 0.77 , ROC AUC of 0.78 & f1 score of 0.57 on test set.
- Five most important predictors are -
  I. Sales
  II. Type_Travel_Agency
  III. Agency_Code_JZI
  IV. Duration
  V. Product Name_Cancellation Plan

Random Forrest -

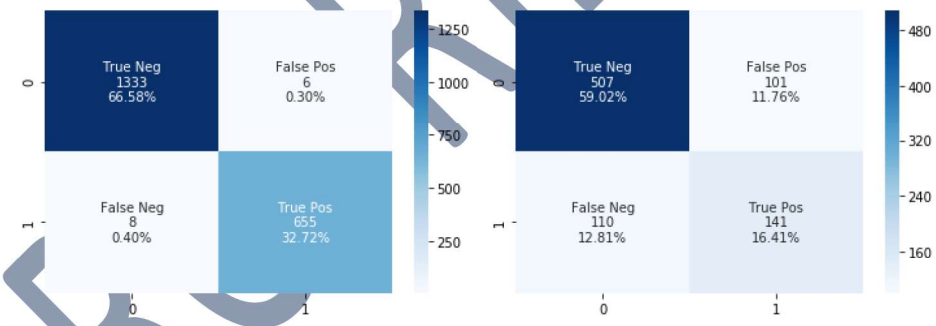
For the first iteration lets build the RF model with default Hyperparameters. Depending upon the nature of fit i.e over fit or under fit, we will go ahead with Hyperparameter tuning to arrive at better-fitted model.

Result set from 1st iteration –

PROPRIETARY

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

```
Accuracy on Training Set is : 0.99

Accuracy on Test Set is : 0.75

Train set ROC-AUC score is : 1.0

Test set ROC-AUC score is : 0.78


Classification report for train set -
              precision    recall  f1-score   support

          No       0.99      1.00      0.99      1339
         Yes       0.99      0.99      0.99       663

    accuracy                           0.99      2002
   macro avg       0.99      0.99      0.99      2002
weighted avg       0.99      0.99      0.99      2002


Classification report for test set -
              precision    recall  f1-score   support

          No       0.82      0.83      0.83       608
         Yes       0.58      0.56      0.57       251

    accuracy                           0.75       859
   macro avg       0.70      0.70      0.70       859
weighted avg       0.75      0.75      0.75       859
```



Confusion Matrices for Train & Test Set

Negative Class = No, Positive Class = Yes

From the above results it is evident that the model has clearly overfit (Accuracy on Train is ~1 and Test is ~0.75). Now we must choose the set of hyperparameters, which will prune the decision trees of the RF ensemble i.e. prevent it from overfitting the train data.

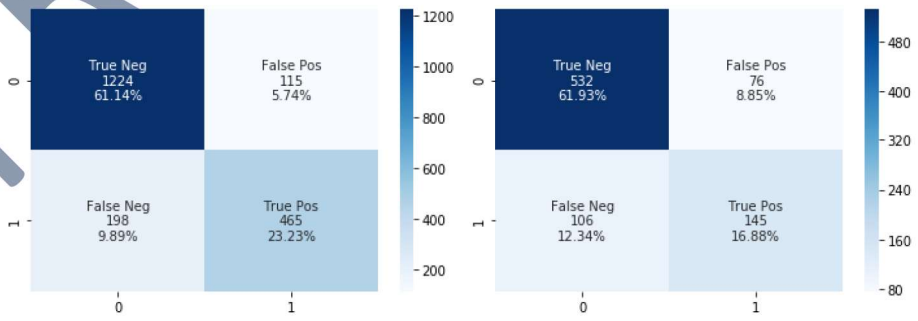
Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Hyperparameters chosen for tuning-

- Max_Depth : This parameters controls the depth a DT can grow to, by default this parameter is set to None (the tree grows untill each leaf node has only one observation).
- Min_samples_split: The minimum number of samples required to split an internal node, greater is this value, lesser overfit the model would be.
- n_estimators: Number of estimators used in the ensemble. Generally, a higher number of estimators gives better results but it uses more resources.
- Max_features : Max features to be considered for each split. As a thumb rule, a good value is the log or sqare root of total features.
- Criterion: Function to consider for the split.
- Warm_start : This parameter controls whether or not to reuse the solution of the previous call to fit.

Results from the Tuned Model –

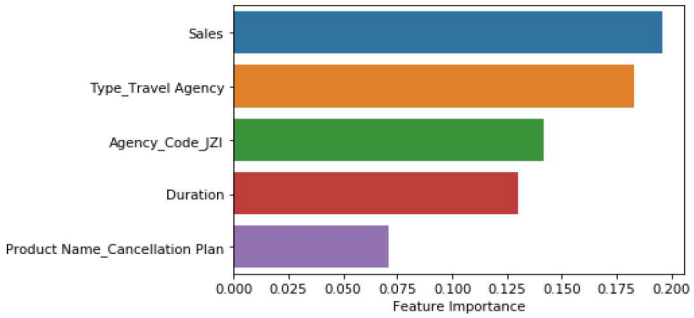
```
Train Accuracy is : 0.84

Test Accuracy is : 0.79

Train Set ROC-AUC score is : 0.93

Test Set ROC-AUC score is : 0.81


Classification report for train set -
              precision    recall  f1-score   support

          No       0.86      0.91      0.89      1339
         Yes       0.80      0.70      0.75       663

    accuracy                           0.84      2002
   macro avg       0.83      0.81      0.82      2002
weighted avg       0.84      0.84      0.84      2002


Classification report for test set -
              precision    recall  f1-score   support

          No       0.83      0.88      0.85       608
         Yes       0.66      0.58      0.61       251

    accuracy                           0.79       859
   macro avg       0.74      0.73      0.73       859
weighted avg       0.78      0.79      0.78       859
```



Confusion Matrices for Train & Test Set
Negative Class = No, Positive Class = Yes

Feature Importance from tuned RF model –

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.



Inferences -

- There exists a certain degree of overfit even after Hyperparameter tuning, it implies that the parameter grid chosen could be further improved. Model building is an iterative task and may require several trials to achieve desired results.
- The best parameters from the chosen values are -
  {'criterion': 'entropy',
  'max_depth': 10,
  'max_features': 'log2',
  'min_samples_split': 4,
  'n_estimators': 500,
  'warm_start': True}
- In terms of model performance, we were able to obtain better generalized results from CART when compared to RF.
- Five best predictors from RF are -
  I. Sales
  II. Type_Travel Agency
  III. Agency_Code_JZI
  IV. Duration
  V. Product Name_Cancellation Plan

PROPRIETARY

Artificial Neural Network

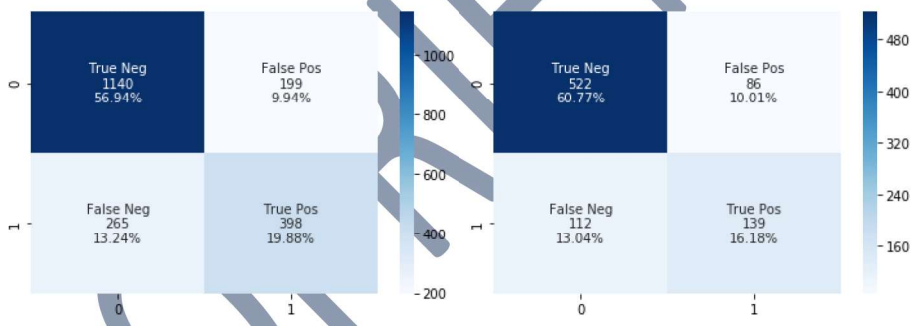
For the first iteration, let us build the ANN model with default Hyperparameters. Depending upon the nature of fit i.e over fit or under fit, we will go ahead with Hyperparameter tuning to arrive at a better-fitted model.

Scaling needs to be performed before Training a Neural Network or else the variables with greater magnitudes influence the results for some of the inputs, with an imbalance not due to the intrinsic nature of the data but merely due to higher magnitudes. We will be using the Min Max scaling for the current dataset.

Result set on 1st iteration ANN model –

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

```
Accuracy on Training Set is : 0.77

Accuracy on Test Set is : 0.77

Train set ROC-AUC score is : 0.81

Test set ROC-AUC score is : 0.82


Classification report for train set -
              precision    recall  f1-score   support

          No       0.81      0.85      0.83      1339
         Yes       0.67      0.60      0.63       663

    accuracy                           0.77      2002
   macro avg       0.74      0.73      0.73      2002
weighted avg       0.76      0.77      0.76      2002


Classification report for test set -
              precision    recall  f1-score   support

          No       0.82      0.86      0.84       608
         Yes       0.62      0.55      0.58       251

    accuracy                           0.77       859
   macro avg       0.72      0.71      0.71       859
weighted avg       0.76      0.77      0.77       859
```



Confusion Matrices for Train & Test Set
Negative Class = No, Positive Class = Yes

We observe that the default parameters have performed fairly well (similar scores on train & test). Let us see if we can improve the results further by Hyperparameter tuning.

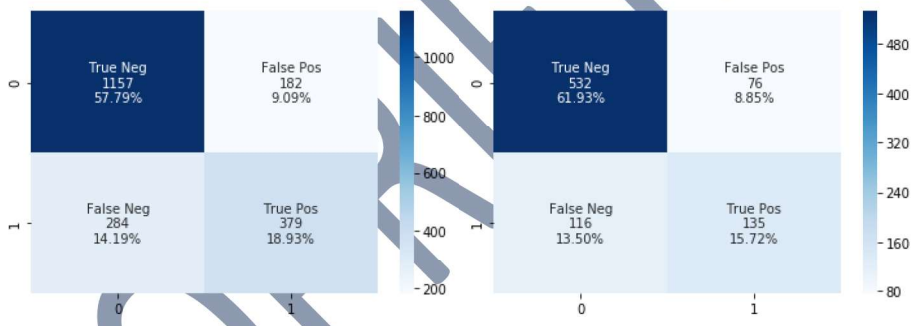
Hyperparameters chosen for tuning-

- Hidden_layer_sizes - The parameter describes the neural architecture to be used. By default it is (100,).
- Activation - Activation function to introduce non-linearity.
- Learning_rate - Parameter that governs the rate for weight updates.
- Alpha - L2 penalty (regularization term) parameter
- Max_iter - Maximum number of iterations of data unless the convergence is achieved.

PROPRIETARY

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

```
Train Accuracy is : 0.77

Test Accuracy is : 0.78

Train Set ROC-AUC score is : 0.81

Test Set ROC-AUC score is : 0.82


Classification report for train set -
              precision    recall  f1-score   support

          No       0.80      0.86      0.83      1339
         Yes       0.68      0.57      0.62       663

    accuracy                           0.77      2002
   macro avg       0.74      0.72      0.73      2002
weighted avg       0.76      0.77      0.76      2002


Classification report for test set -
              precision    recall  f1-score   support

          No       0.82      0.88      0.85       608
         Yes       0.64      0.54      0.58       251

    accuracy                           0.78       859
   macro avg       0.73      0.71      0.72       859
weighted avg       0.77      0.78      0.77       859
```



Confusion Matrices for Train & Test Set
Negative Class = No, Positive Class = Yes

Inferences –

1. Tuned ANN model performed very similar to the default ANN model.
2. Overall, we were able to achieve good generalized results from the ANN model, and Recall scores slightly better than the previous two models.
3. The best values of parameters from the chosen parameter grid are –
   {'activation': 'relu', 'alpha': 0.05,
   'hidden_layer_sizes': (10, 30, 10), 'learning_rate': 'constant', 'max_iter': 200}

2.4 Final Model - Compare all models on the basis of the performance metrics in a structured tabular manner. Describe which model is best/optimized.

PROPRIETARY

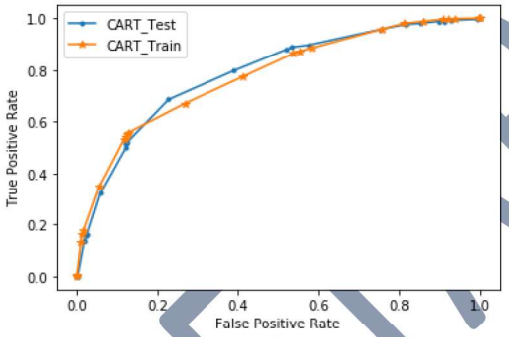
Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

	Train_Accuracy	Test_Accuracy	Train_ROC-AUC	Test_ROC-AUC	Train_Precison	Test_Precison	Train_Recall	Test_Recall	Train_f1	Test_f1
CART	0.77	0.77	0.78	0.79	0.69	0.64	0.55	0.52	0.61	0.57
RF	0.84	0.79	0.93	0.81	0.80	0.66	0.70	0.58	0.75	0.61
ANN	0.77	0.78	0.81	0.82	0.68	0.64	0.57	0.54	0.62	0.58

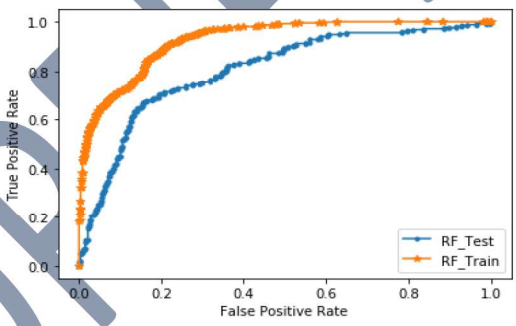
Results from the Tuned Models

Comparison of ROC curves –

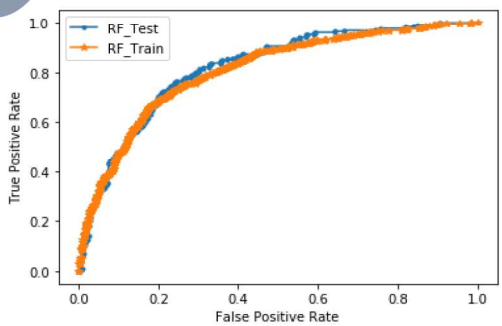
1. CART Model



2. Random Forest Model –



3. ANN Model –



From the results above, we can conclude that ANN model is best/optimized because of the lowest difference among evaluation metrics over train and test set, and slightly higher scores (better than CART, which can be considered as second best). In addition, in this particular case study, recall score is a very

PROPRIETARY

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

important metric because the business would like to find all the customers likely to claim an Insurance (mathematically minimize the FN). Good recall score for Positive class was obtained from the ANN model.

2.5 Based on your analysis and working on the business problem, detail out appropriate insights and recommendations to help the management solve the business objective.
Insights and Recommendations -

- Median Commission is much higher for people who have claimed for Insurance. Similarly, Median Sales is also higher for people who claimed for Insurance. This indicates that high budget trips are more prone to insurance claims.
- C2B agency has been responsible for highest claim percentage; detailed study must be conducted to find out the reasons, and possible solutions. Agency must be made aware of Best Practices and precautionary measures to lower the claims.
- Customers using Airlines have shown high claims; efforts must be made to improve this situation. Airline partners can be made aware of this analysis and together plan a remedial course of action.
- Sales, Type_Travel Agency and Agency_Code_JZI have been the top predictors from the both CART and RF model. (We do not get Feature Importance values for an ANN from sklearn module , thus we are considering the feature importance results from the CART & RF models only)
- We were able to create a generalized ANN model, which can be used to predict the outcome with a Recall of 0.54.

PROPRIETARY

End of Submission

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Appendix

Problem 1

Importing necessary libraries-

```
import pandas as pd
import numpy as np
```

```
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
```

Data Ingestion:

1. Data Ingestion: Read the data, do the necessary initial steps and exploratory data analysis (Univariate, Bi-variate and multivariate analysis)

```
data_bank= pd.read_csv('bank_marketing_part1_Data.csv')
```

```
data_bank.head().round(2)
```

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
0	19.94	16.92	0.88	6.68	3.76	3.25	6.55
1	15.99	14.89	0.91	5.36	3.58	3.34	5.14
2	18.95	16.42	0.88	6.25	3.76	3.37	6.15
3	10.83	12.96	0.81	5.28	2.64	5.18	5.18
4	17.99	15.86	0.90	5.89	3.69	2.07	5.84

```
data_bank.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 210 entries, 0 to 209
Data columns (total 7 columns):
spending                      210 non-null float64
advance_payments              210 non-null float64
probability_of_full_payment   210 non-null float64
current_balance               210 non-null float64
credit_limit                  210 non-null float64
min_payment_amt               210 non-null float64
max_spent_in_single_shopping  210 non-null float64
dtypes: float64(7)
memory usage: 11.6 KB
```

```
print('The Shape of the Dataset is :{}'.format(data_bank.shape))
```

```
The Shape of the Dataset is :(210, 7)
```

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

```
data_bank.describe().T.round(2)
```

	count	mean	std	min	25%	50%	75%	max
spending	210.0	14.85	2.91	10.59	12.27	14.36	17.30	21.18
advance_payments	210.0	14.56	1.31	12.41	13.45	14.32	15.72	17.25
probability_of_full_payment	210.0	0.87	0.02	0.81	0.86	0.87	0.89	0.92
current_balance	210.0	5.63	0.44	4.90	5.26	5.52	5.98	6.68
credit_limit	210.0	3.26	0.38	2.63	2.94	3.24	3.56	4.03
min_payment_amt	210.0	3.70	1.50	0.77	2.56	3.60	4.77	8.46
max_spent_in_single_shopping	210.0	5.41	0.49	4.52	5.04	5.22	5.88	6.55

```
data_bank.isnull().sum()
```

```
spending                      0
advance_payments              0
probability_of_full_payment   0
current_balance               0
credit_limit                  0
min_payment_amt               0
max_spent_in_single_shopping  0
dtype: int64
```

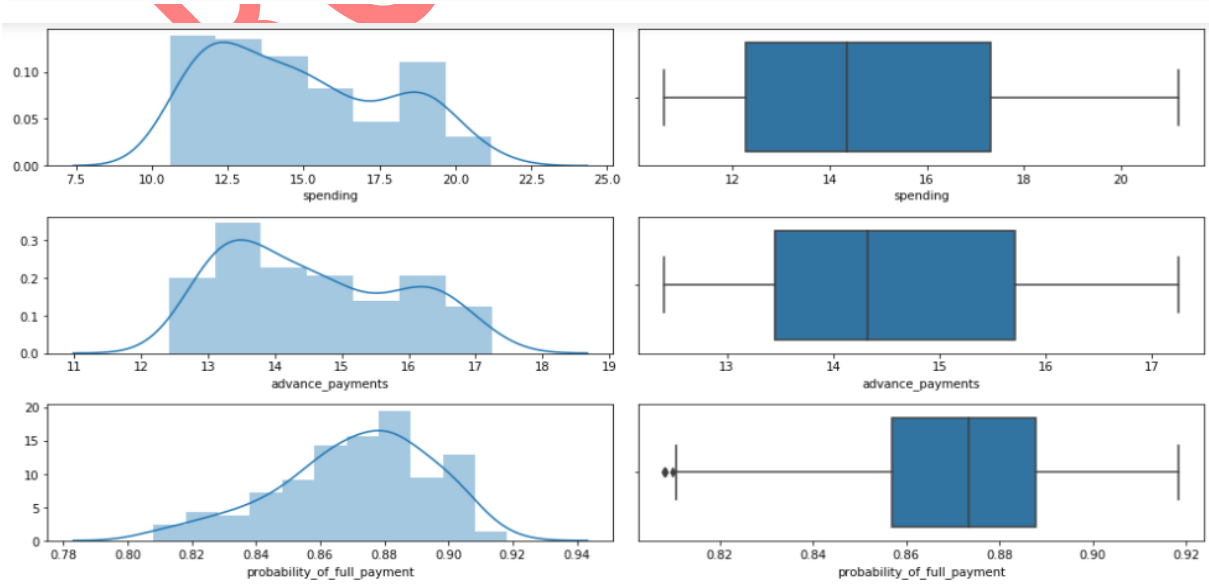
```
data_bank[data_bank.duplicated()].shape[0]
```

```
0
```

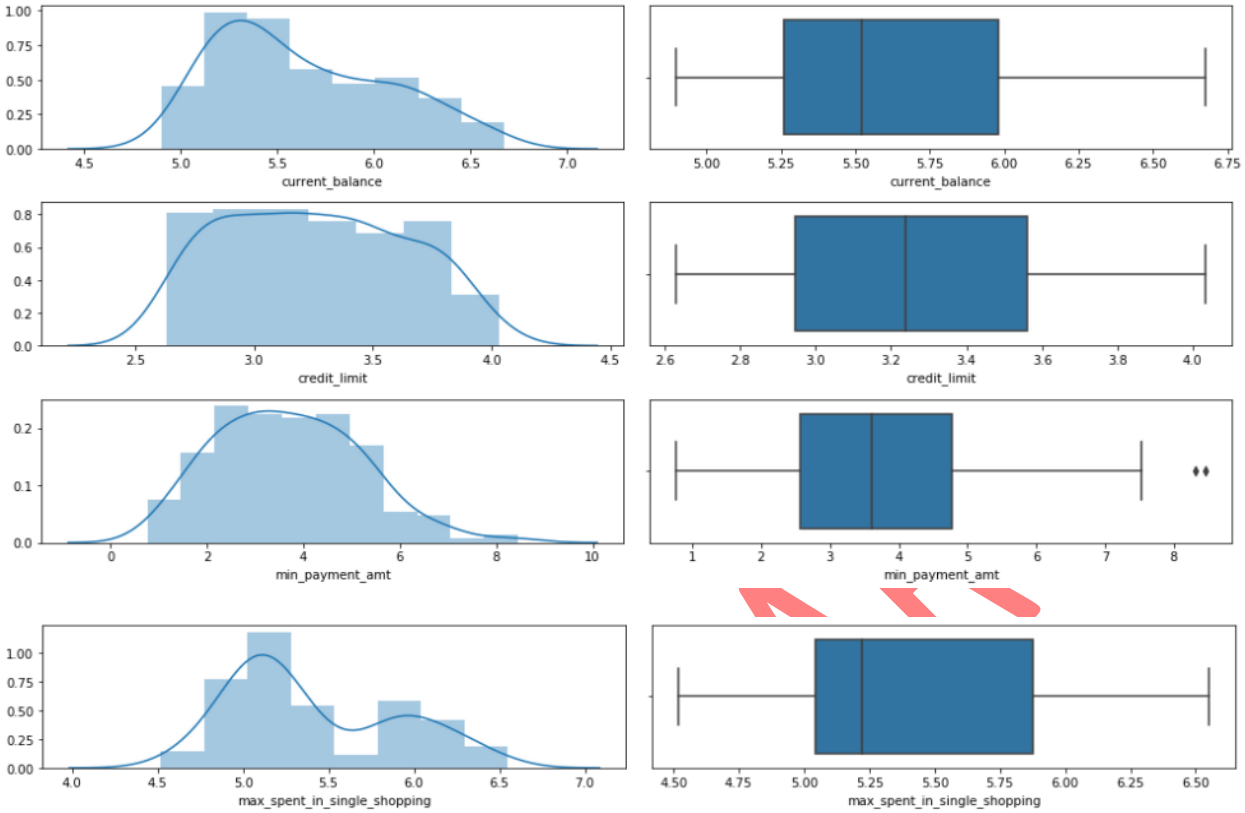
Univariate analysis -

To perform Uni-variate analysis on continuous variables, lets plot the Box plots & Distribution plots.

```
fig, axes = plt.subplots(nrows=7,ncols=2)
fig.set_size_inches(14, 16)
j=0
for i in data_bank.columns:


        a = sns.distplot(data_bank[i],ax=axes[j][0])
        a = sns.boxplot(data_bank[i] , orient = "h" , ax=axes[j][1])
        j=j+1

fig. tight_layout(w_pad=1.0)
plt.show()
```



Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.



Lets find out the Percentage of outliers in each column -

```
Q1 = data_bank.quantile(0.25)
Q3 = data_bank.quantile(0.75)
IQR = Q3 - Q1
pd.DataFrame((((data_bank < (Q1 - 1.5 * IQR)) | (data_bank > (Q3 + 1.5 * IQR))).sum()/data_bank.shape[0]*100),
             columns =['Outlier %'],index=None).round(2)
```

	Outlier %
spending	0.00
advance_payments	0.00
probability_of_full_payment	1.43
current_balance	0.00
credit_limit	0.00
min_payment_amt	0.95
max_spent_in_single_shopping	0.00

Only two columns have outlier values that too with less than 1.5 Percentage. For treating these outliers we shall be using the flooring and capping technique in which the outlier value will be changed to the nearest whisker.

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

```
Q1 = data_bank['min_payment_amt'].quantile(0.25)
Q3 = data_bank['min_payment_amt'].quantile(0.75)
IQR = Q3 - Q1
data_bank['min_payment_amt'] = np.where(data_bank['min_payment_amt'] <Q1 - 1.5 *IQR ,Q1-1.5*IQR, data_bank['min_payment_amt'])
data_bank['min_payment_amt'] = np.where(data_bank['min_payment_amt'] >Q3 + 1.5 *IQR ,Q3+ 1.5 *IQR, data_bank['min_payment_amt'])
```

```
Q1 = data_bank['probability_of_full_payment'].quantile(0.25)
Q3 = data_bank['probability_of_full_payment'].quantile(0.75)
IQR = Q3 - Q1
data_bank['probability_of_full_payment'] = np.where(data_bank['probability_of_full_payment'] <Q1 - 1.5 *IQR ,Q1 - 1.5 *IQR,
                                                    data_bank['probability_of_full_payment'])
data_bank['probability_of_full_payment'] = np.where(data_bank['probability_of_full_payment'] >Q3 + 1.5 *IQR ,Q3 + 1.5 *IQR,
                                                    data_bank['probability_of_full_payment'])
```

```
Q1 = data_bank.quantile(0.25)
Q3 = data_bank.quantile(0.75)
IQR = Q3 - Q1
((data_bank < (Q1 - 1.5 * IQR)) | (data_bank > (Q3 + 1.5 * IQR))).sum()/data_bank.shape[0]*100
```

```
spending                    0.0
advance_payments            0.0
probability_of_full_payment 0.0
current_balance             0.0
credit_limit                0.0
min_payment_amt             0.0
max_spent_in_single_shopping 0.0
dtype: float64
```

```
pd.DataFrame(data = [data_bank.kurtosis(),data_bank.skew()],index=['Kurtosis','Skewness']).T.round(2)
```

	Kurtosis	Skewness
spending	-1.08	0.40
advance_payments	-1.11	0.39
probability_of_full_payment	-0.19	-0.52
current_balance	-0.79	0.53
credit_limit	-1.10	0.13
min_payment_amt	-0.22	0.36
max_spent_in_single_shopping	-0.84	0.56

Multi-variate Analysis-

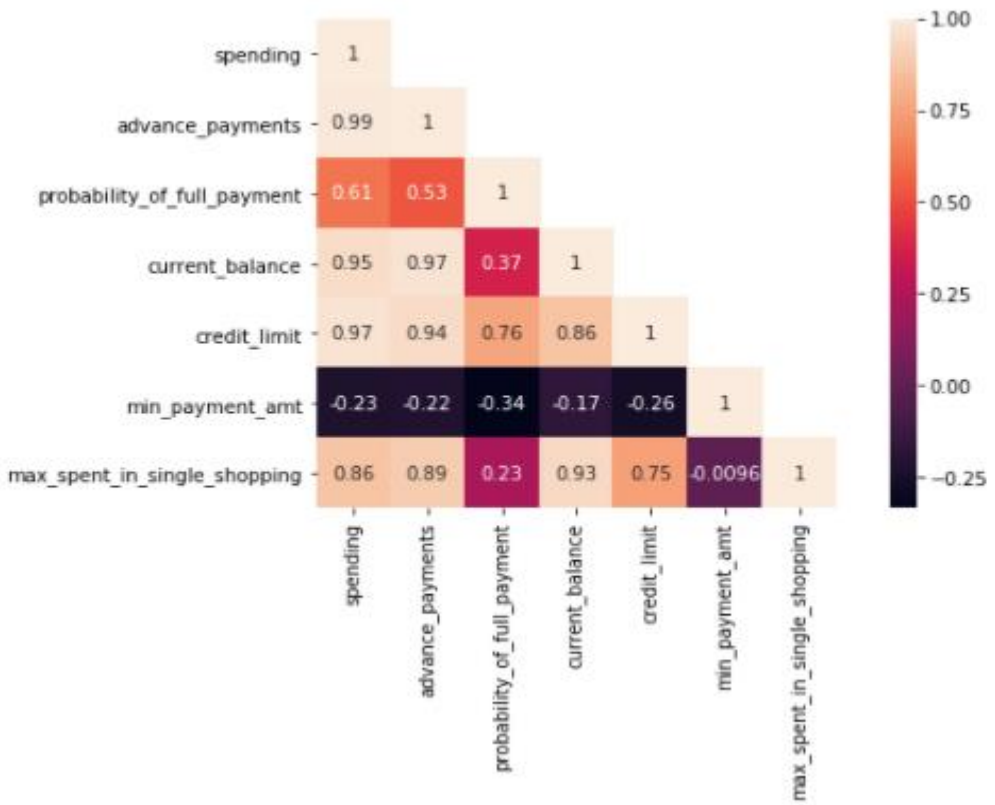
Analyzing the relationship among continuous variables by using Pair plot and Correlation Heatmap-

```
sns.pairplot(data_bank)
plt.show()
```

Correlation of Continuous variables -

```
cor = data_bank.corr()
mask = np.array(cor)
mask[np.tril_indices_from(mask)] = False
fig,ax= plt.subplots()
fig.set_size_inches(10,5)
sns.heatmap(cor, mask=mask,vmax=1, square=True,annot=True)
plt.show()
```

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.




```
data_bank.describe().T.round(2)
```

	count	mean	std	min	25%	50%	75%	max
spending	210.0	14.85	2.91	10.59	12.27	14.36	17.30	21.18
advance_payments	210.0	14.56	1.31	12.41	13.45	14.32	15.72	17.25
probability_of_full_payment	210.0	0.87	0.02	0.81	0.86	0.87	0.89	0.92
current_balance	210.0	5.63	0.44	4.90	5.26	5.52	5.98	6.68
credit_limit	210.0	3.26	0.38	2.63	2.94	3.24	3.56	4.03
min_payment_amt	210.0	3.70	1.49	0.77	2.56	3.60	4.77	8.08
max_spent_in_single_shopping	210.0	5.41	0.49	4.52	5.04	5.22	5.88	6.55

```
min_max = MinMaxScaler()
```

```
data_bank_sc =pd.DataFrame(columns= data_bank.columns, data = min_max.fit_transform(data_bank))
```

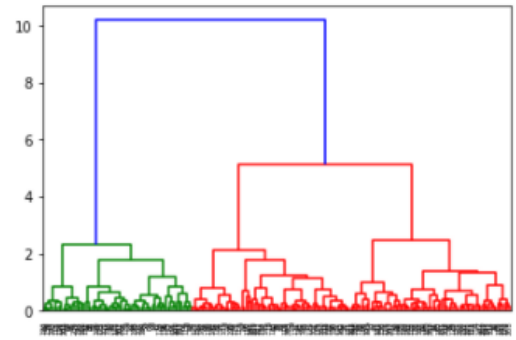
```
data_bank_sc.describe().T.round(2)
```

	count	mean	std	min	25%	50%	75%	max
spending	210.0	0.40	0.27	0.0	0.16	0.36	0.63	1.0
advance_payments	210.0	0.44	0.27	0.0	0.21	0.39	0.68	1.0
probability_of_full_payment	210.0	0.56	0.22	0.0	0.43	0.58	0.72	1.0
current_balance	210.0	0.41	0.25	0.0	0.20	0.35	0.61	1.0
credit_limit	210.0	0.45	0.27	0.0	0.22	0.43	0.66	1.0
min_payment_amt	210.0	0.40	0.20	0.0	0.25	0.39	0.55	1.0
max_spent_in_single_shopping	210.0	0.44	0.24	0.0	0.26	0.35	0.67	1.0

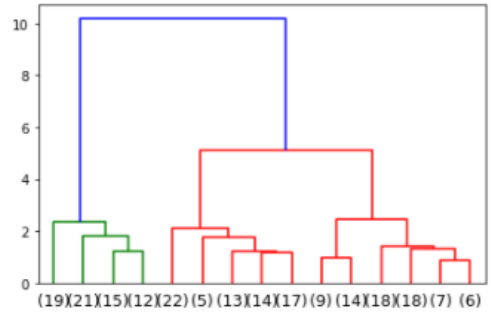
Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

```
from sklearn.cluster import AgglomerativeClustering
import scipy.cluster.hierarchy as sch
from sklearn.metrics import silhouette_score
```

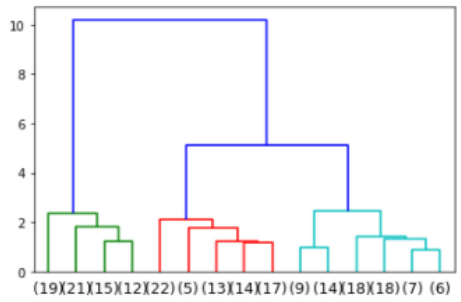
```
dendrogram = sch.dendrogram(sch.linkage(data_bank_sc, method='ward',metric='euclidean'))
```




```
dendrogram = sch.dendrogram(sch.linkage(data_bank_sc, method='ward'),p = 15 ,truncate_mode='lastp',color_threshold=8)
```




```
dendrogram = sch.dendrogram(sch.linkage(data_bank_sc, method='ward'),p = 15 ,truncate_mode='lastp',color_threshold=4)
```



For Color threshold 8, we get Two clusters whereas for 4 we get Three clusters in the dataset. Generally analyzing Two clusters is not very useful to the business hence we will choose Three clusters as the optimum. In situations where the business needs to make a Binary decision say whether to give a loyalty bonus or not, two clusters can be utilized.

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

```
sil_scores = {}   # blank dictionary to store score and number of unique values for each cluster
```

```
for i in range(2,10):
    model = AgglomerativeClustering(n_clusters=i, affinity='euclidean', linkage='ward')
    model.fit(data_bank_sc)
    labels = model.labels_
    sil_scores[i] = (silhouette_score(data_bank_sc, labels, metric='euclidean',random_state=100).round(2),
                     np.unique(labels, return_counts=True)[1])
```

```
pd.DataFrame(sil_scores,index=['Silh_Scores','Cluster Distribution']).T
```

	Silh_Scores	Cluster Distribution
2	0.49	[143, 67]
3	0.39	[72, 67, 71]
4	0.3	[67, 49, 71, 23]
5	0.25	[71, 49, 48, 23, 19]
6	0.22	[48, 49, 49, 23, 19, 22]
7	0.22	[49, 49, 27, 23, 19, 22, 21]
8	0.23	[49, 44, 27, 23, 19, 22, 21, 5]
9	0.22	[31, 44, 27, 23, 19, 22, 21, 5, 18]

```
from sklearn.cluster import KMeans
```

```
SSE = []
for cluster in range(1,20):
    kmeans = KMeans(n_jobs = -1, n_clusters = cluster, init='k-means++',random_state=100)
    kmeans.fit(data_bank_sc)
    SSE.append(kmeans.inertia_)

# converting the results into a dataframe and plotting them
frame = pd.DataFrame({'Cluster':range(1,20), 'SSE':SSE})
plt.figure(figsize=(12,6))
plt.plot(frame['Cluster'], frame['SSE'], marker='o')
plt.xlabel('Number of clusters')
plt.ylabel('Inertia')
```

```
Text(0, 0.5, 'Inertia')
```

```
pd.DataFrame(sil_scores,index=['Silh_Scores','Cluster Distribution']).T
```

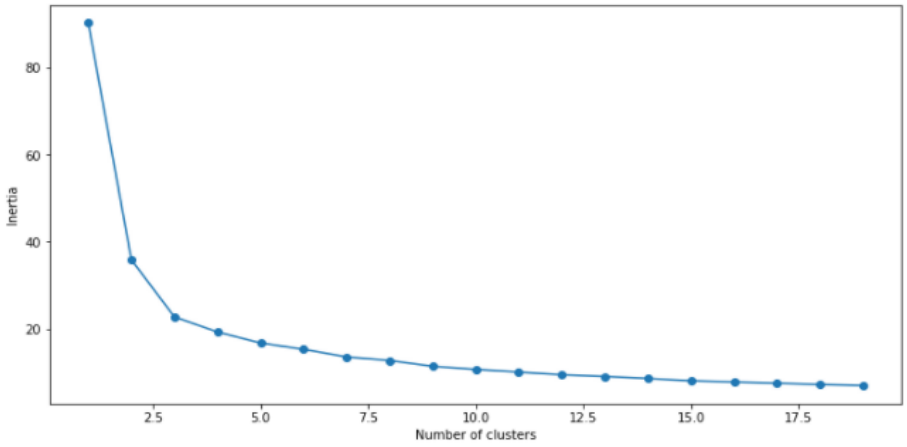
	Silh_Scores	Cluster Distribution
2	0.49	[143, 67]
3	0.39	[72, 67, 71]
4	0.3	[67, 49, 71, 23]

```
columns_list = list(data_bank_sc.columns)

for i in columns_list:
    for j in columns_list:
        if i!=j:

            X = data_bank.loc[:, [i, j]].values
            plt.scatter(x= X[labels==0, 0], y= X[labels==0, 1], s=50, marker="*", color='red')
            plt.scatter(x=X[labels==1, 0],y= X[labels==1, 1], s=50, marker="1", color='blue')
            plt.scatter(x=X[labels==2, 0],y= X[labels==2, 1], s=50, marker='o', color='green')
```

```
sil_scores_knn = {}
for i in range(2,20):
    model = KMeans(n_jobs = -1, n_clusters = i, init='k-means++',random_state=100)
    model.fit(data_bank_sc)
    labels = model.labels_
    sil_scores_knn[i] = (silhouette_score(data_bank_sc, labels, metric='euclidean',random_state=100).round(2),
                         np.unique(labels, return_counts=True)[1])
```

```
pd.DataFrame(sil_scores_knn,index =['Silh_Scores','Cluster Distribution'] ).T
```

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.



To determine the optimal number of clusters, we have to select the value of k at the "elbow" ie the point after which the distortion/inertia starts decreasing in a linear fashion. Thus for the given data, we conclude that the optimal number of clusters is 3.

PROPRIETAR

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

	Silh_Scores	Cluster Distribution
2	0.5	[133, 77]
3	0.42	[77, 64, 69]
4	0.34	[63, 46, 51, 50]
5	0.3	[51, 27, 48, 48, 36]
6	0.28	[51, 27, 48, 22, 36, 26]
7	0.28	[16, 25, 24, 47, 41, 27, 30]
8	0.28	[37, 12, 44, 36, 23, 22, 16, 20]
9	0.26	[15, 24, 24, 33, 15, 30, 24, 21, 24]
10	0.27	[24, 35, 21, 28, 15, 22, 25, 17, 15, 8]
11	0.26	[34, 9, 22, 20, 24, 22, 15, 16, 14, 18, 16]
12	0.26	[23, 9, 22, 20, 24, 22, 14, 12, 14, 18, 16, 16]
13	0.24	[16, 18, 20, 15, 12, 17, 26, 13, 21, 6, 24, 7,...
14	0.25	[15, 18, 21, 18, 11, 17, 14, 12, 17, 6, 27, 7,...
15	0.25	[6, 21, 21, 16, 14, 13, 18, 10, 18, 17, 18, 16...
16	0.25	[6, 21, 16, 16, 14, 6, 18, 10, 15, 20, 18, 16,...
17	0.25	[6, 21, 16, 15, 14, 6, 16, 10, 15, 20, 18, 16,...
18	0.24	[6, 17, 16, 15, 13, 6, 16, 10, 15, 20, 17, 14,...
19	0.25	[16, 6, 12, 18, 14, 11, 13, 9, 7, 17, 7, 16, 1...

```
kmeans = KMeans(n_jobs = -1, n_clusters = 3, init='k-means++',random_state=100)
kmeans.fit(data_bank_sc)
pred_km = kmeans.predict(data_bank_sc)
```

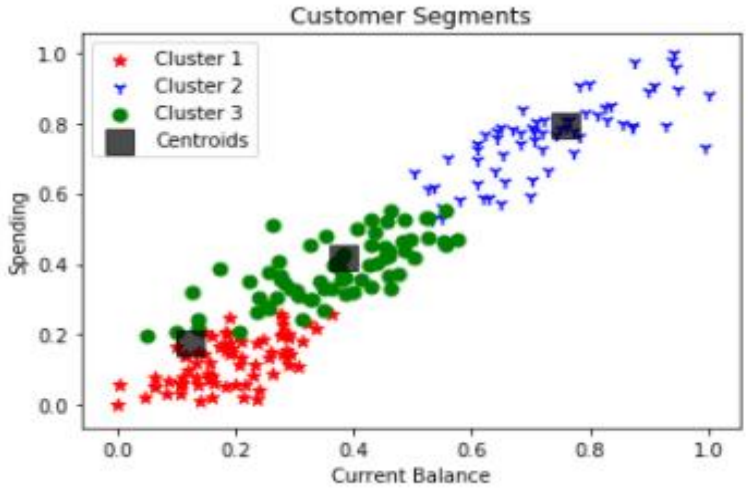
```
np.unique(pred_km,return_counts=True)
```

```
(array([0, 1, 2]), array([77, 64, 69], dtype=int64))
```

Visualizing Clusters -

```
#X = data_bank_sc[['current_balance','spending']]
X = data_bank_sc.loc[:, ['current_balance','spending']].values
plt.scatter(x= X[labels==0, 0], y= X[labels==0, 1], s=50, marker="*", color='red',label = 'Cluster 1')
plt.scatter(x=X[labels==1, 0],y= X[labels==1, 1], s=50, marker="1", color='blue',label = 'Cluster 2')
plt.scatter(x=X[labels==2, 0],y= X[labels==2, 1], s=50, marker='o', color='green',label = 'Cluster 3')


plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1],s=200,marker='s',c='black',
            alpha=0.7,label = 'Centroids')


plt.title('Customer Segments')

plt.xlabel('Current Balance')
plt.ylabel('Spending')
plt.legend()
plt.show()
```

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

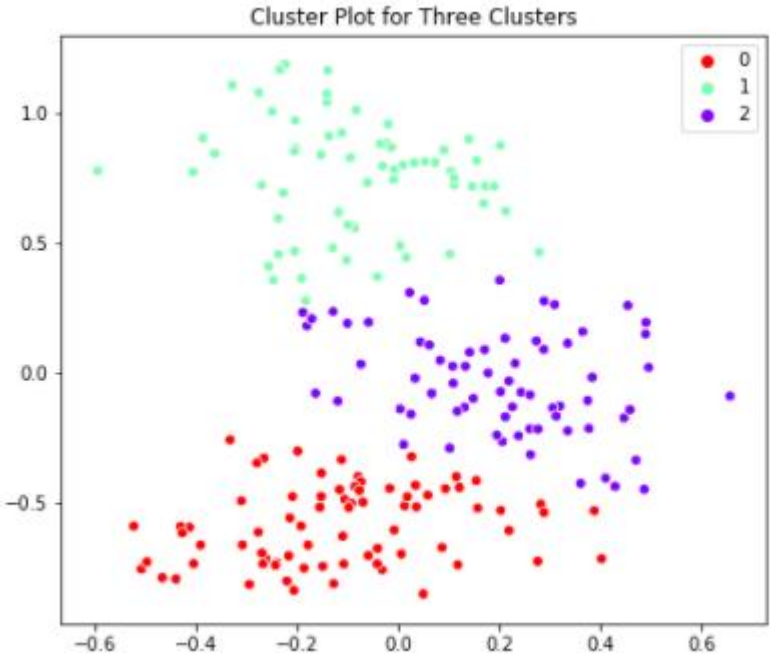


We can also visualize the clusters by using two Principal Components -

```
X = data_bank_sc.values
from sklearn.decomposition import PCA
```

```
pca = PCA(n_components=2,random_state=100)
pca_data = pca.fit_transform(X)
plt.figure(figsize=(7,6))
sns.scatterplot(x= pca_data[:,1], y =pca_data[:,0], hue= labels,palette="rainbow_r")
plt.title('Cluster Plot for Three Clusters')
```

```
Text(0.5, 1.0, 'Cluster Plot for Three Clusters')
```



Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

```
data_bank['Labels_Agg'] = labels
data_bank['Labels_KMeans'] = pred_km
```

```
print(np.unique(labels,return_counts=True))
print(np.unique(pred_km,return_counts=True))
```

```
(array([0, 1, 2], dtype=int64), array([72, 67, 71], dtype=int64))
(array([0, 1, 2]), array([77, 64, 69], dtype=int64))
```

```
data_bank.groupby('Labels_KMeans').mean().T.round(2)
```

Labels_KMeans	0	1	2
spending	11.90	18.61	14.65
advance_payments	13.26	16.25	14.44
probability_of_full_payment	0.85	0.88	0.88
current_balance	5.23	6.20	5.55
credit_limit	2.86	3.71	3.29
min_payment_amt	4.59	3.59	2.80
max_spent_in_single_shopping	5.09	6.06	5.17
Labels_Agg	1.74	0.97	0.19

PROPRIE

Proprietary content. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.