DATA MINING

PGP – DSBA

Preeja Rajesh

Table of Contents

1 Clustering (Unsupervised Learning)

1.1 Problem Statement 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.

Dataset: bank_marketing_part1_Data.csv

Data Dictionary for Market Segmentation:

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 and do exploratory data analysis. Describe the data briefly.

1.1.1 Exploratory Data Analysis:

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt
0	19.94	16.92	0.8752	6.675	3.763	3.252
1	15.99	14.89	0.9064	5.363	3.582	3.336
2	18.95	16.42	0.8829	6.248	3.755	3.368
3	10.83	12.96	0.8099	5.278	2.641	5.182
4	17.99	15.86	0.8992	5.890	3.694	2.068

1.1.2 Data Pre-processing:

Data Pre-processing involves steps like data cleaning, data integration, data transformation, data reduction and data discretization

- There are total 210 rows and 7 columns in the dataset.
- There are 0 null values present in the dataset.
- Number of duplicate rows = 0

- Converting the values of dataset by its multiples.

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_sin
0	19940.0	1692.0	0.8752	6675.0	37630.0	325.2	
1	15990.0	1489.0	0.9064	5363.0	35820.0	333.6	
2	18950.0	1642.0	0.8829	6248.0	37550.0	336.8	
3	10830.0	1296.0	0.8099	5278.0	26410.0	518.2	
4	17990.0	1586.0	0.8992	5890.0	36940.0	206.8	

- 5-point summary of dataset

	count	mean	std	min	25%	50%	75%	max
spending	210.0	14847.523810	2909.699431	10590.0000	12270.0000	14355.00000	17305.000000	21180.0000
advance_payments	210.0	1455.928571	130.595873	1241.0000	1345.0000	1432.00000	1571.500000	1725.0000
probability_of_full_payment	210.0	0.870999	0.023629	0.8081	0.8569	0.87345	0.887775	0.9183
current_balance	210.0	5628.533333	443.063478	4899.0000	5262.2500	5523.50000	5979.750000	6675.0000
credit_limit	210.0	32586.047619	3777.144449	26300.0000	29440.0000	32370.00000	35617.500000	40330.0000
min_payment_amt	210.0	370.020095	150.355713	76.5100	256.1500	359.90000	476.875000	845.6000
max_spent_in_single_shopping	210.0	5408.071429	491.480499	4519.0000	5045.0000	5223.00000	5877.000000	6550.0000

Inferences:

- The variables of the data set are of different scales i.e. one variable is in thousands and other in decimals. For e.g. in our data set “credit limit” is having values in thousands and “probability of full payment” is in decimals. Since the data in these variables are of different scales, it is tough to compare these variables.

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

- Data Mining can generate effective results if normalization is applied to the dataset. It is a process used to standardize all the attributes of the dataset and give them equal weight so that redundant or noisy objects can be eliminated and there is valid and reliable data which enhances the accuracy of the result. K-Means algorithm uses Euclidean distance that is highly prone to irregularities in the size of various features.
- There are various data normalization methods like Min-Max, Z-Score etc.. The best normalization method depends on the data to be normalized. Here, we have used Min-Max normalization technique in our algorithm because our dataset is limited and has not much variability between minimum and maximum.
- Min-Max normalization technique performs a linear transformation on the data. In this method, we fit the data in a predefined boundary or in a predefined interval.

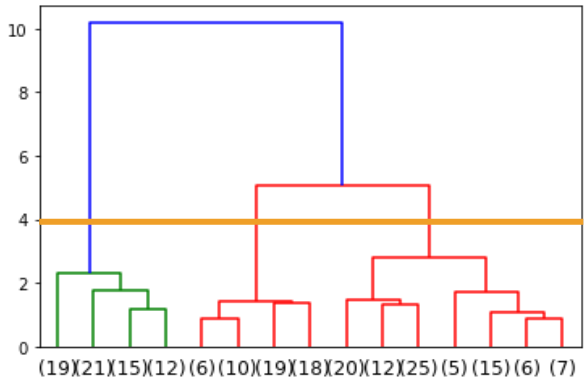
- Scaled data:

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_sin
0	0.882908	0.931818	0.608893	1.000000	0.807555	0.323356	
1	0.509915	0.512397	0.892015	0.261261	0.678546	0.334278	
2	0.789424	0.828512	0.678766	0.759572	0.801853	0.338439	
3	0.022663	0.113636	0.016334	0.213401	0.007840	0.574302	
4	0.698772	0.712810	0.826679	0.557995	0.758375	0.169408	

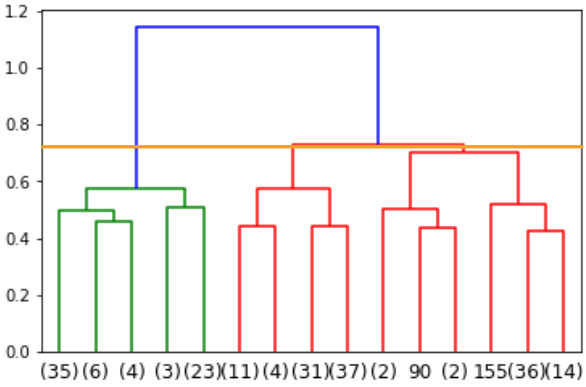
1.3 Apply hierarchical clustering to scaled data. Identify the number of optimum clusters using Dendrogram and briefly describe them

1.1.3 Creating the Dendrogram

- Choosing ward linkage method Plot the truncated dendrogram with the last 15 clusters.



- Choosing average linkage method Plot the truncated dendrogram with the last 15 clusters



- ***Inference:***

  - 2 clusters really do not make much business impact as it is kind of implicit. Cutoff 1 (at 4 of Y axis) looks to be more suitable for this type of data since the vertical line that pass through the very first cutoff are of highest length.
  - If these 3 clusters shown in 2 red and 1 green if combined into one, then it is because of agglomerative property.

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

1.1.4 K means clustering

It is one of the most popular clustering algorithms and usually the first thing practitioners apply when solving clustering tasks to get an idea of the structure of the dataset. The goal of K means is to group data points into distinct non-overlapping subgroups. One of the major applications of K means clustering is segmentation of customers to get a better understanding of them which in turn could be used to increase the revenue of the company/bank.

$$WCSS = \sum_{i \in n} (X_i - Y_i)^2$$

K-means algorithm can be summarized as follows:

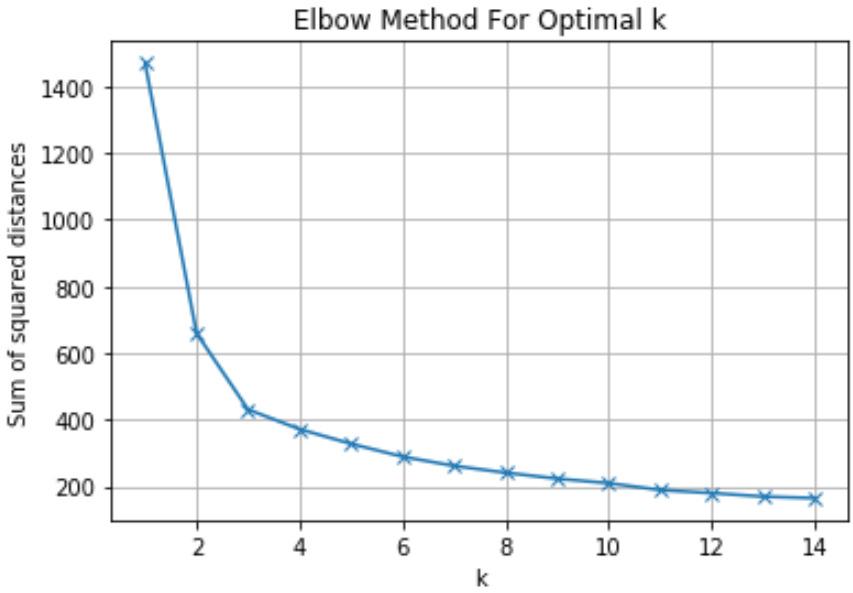
1. Specify the number of clusters (k) to be created (by the analyst)
2. Select randomly k objects from the data set as the initial cluster centres or means
3. Assigns each observation to their closest centroid, based on the Euclidean distance between the object and the centroid
4. For each of the k clusters update the cluster centroid by calculating the new mean values of all the data points in the cluster. The centroid of a k^{th} cluster is a vector of length *p* containing the means of all variables for the observations in the kth cluster; *p* is the number of variables.
5. Iteratively minimize the total within sum of square. That is, iterate steps 3 and 4 until the cluster assignments stop changing or the maximum number of iterations is reached.

1.1.5 The Elbow Method

Calculate the Within Cluster Sum of Squared Errors (WCSS) for different values of k (2-15) and choose the k for which WCSS first starts to diminish.

k

- 88.98592483911847 = 2
- 34.81326792694563 = 3
- 22.024363075666038 = 4
- 18.681577858087024 = 5
- 16.191202347675954 = 6
- 14.494564553309917 = 7
- 13.23558659174076 = 8
- 11.941783933425054 = 9
- 11.018132706956056 = 10
- 10.343795581087996 = 11
- 9.660880166057023 = 12
- 9.078827253156451 = 13
- 8.760840950267143 = 14
- 8.379514278072097 = 15



In the plot of WCSS-versus k, this is visible as an elbow. The optimal K value is found to be 3 using the elbow method.

k	silhouette_score	silhouette_samples
2	0.505	0.00106021
3	0.422	0.00460102
4	0.338	-0.01049876

Inference:

- WSS reduces as K keeps increasing
- Silhouette score: The average of sil-width for each observation of a dataset is called as silhouette score.
- Silhouette score for 2 Cluster is 0.5 which is closer to +1 than for 3 and 4 cluster (0.42 & 0.33) respectively.
- But selection 2 clusters do not give us any insights so we can say that the 3 Clusters are well separated from each other on an average.
- From 1 and 2 cluster shown in the (WSS) plot, there is a significant drop. Similarly, there is a significant drop between 2 and 3. Hence, 3 is a valuable addition in K-means algorithm.

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

1.1.6 Hierarchical clustering:

Cluster Frequency:

- Cluster 1 67
- Cluster 2 53
- Cluster 3 90

H_clusters	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	m
1	18452.537313	1618.477612	0.884042	6173.567164	36920.597015	3.696179	
2	14903.018868	1457.113208	0.881602	5595.660377	33147.547170	2.480757	
3	12131.111111	1334.222222	0.855044	5242.144444	29028.555556	4.421312	

Inference:

- Cluster 1: Customers that are big spender and pays large amount advances.
- Cluster 2: Customers that are High spenders with lowest minimum payment and has low credit limit.
- Cluster 3: Customers that are smallest spenders with highest minimum payment and lowest credit limit.

Recommendations:

- The goal was to segment the customers of a bank to give promotional offers.
- **Cluster 1:** These customers are big spender and gives large payments in advance. Customers with high average card spend and high average balance are typically those customers who are economically stable and have high spending capacity. So, can promote various expensive brand-new credit card product to these customers to increase their credit limit and so that they spend more. They are 67 customers.

- **Cluster 2:** These customers are high average spenders and has low credit limit i.e. low average balance. Economically they may not be stable but have high potential of purchasing/spending because of various reasons. So, can offer a loan to increase their spending. Offering loan to these customers will not make much difference as they have high balance anyway. They are 53 customers.

- **Cluster 3:** These customers are Smallest Spenders and has lowest credit limit., but minimum paid by the customer while making payments for purchases made monthly are highest than customers in cluster 1 & 2. So, can promote various monthly shopping schemes as well as loans to these customers to increase their monthly shopping. They are 90 customers.

K-Means clustering (k=3):

Cluster Frequency:

- Cluster 0 64
- Cluster 1 69
- Cluster 2 77

Kmeans_clusters	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_a
0	18610.156250	1625.171875	0.884600	6195.546875	37096.093750	359.2093
1	14651.159420	1444.202899	0.882067	5546.681159	32873.043478	279.6857
2	11896.103896	1325.766234	0.849775	5230.597403	28580.259740	459.9545

Inference:

- Cluster 0: Customers that are big spender and pays large amount advances.
- Cluster 1: Customers that are High spenders with lowest minimum payment and has low credit limit.
- Cluster 2: Customers that are smallest spenders with highest minimum payment and lowest credit limit.

Recommendations:

- The goal was to segment the customers of a bank to give promotional offers.

- **Cluster 0:** These customers are Big Spender and gives large payments in advance. Customers with high average card spend and high average balance are typically those customers who are economically stable and have high spending capacity. So, can promote various expensive brand-new credit card product to these customers to increase their credit limit and so that they spend more. They are 64 customers.

- **Cluster 1:** These customers are High average Spenders and has Low Credit Limit i.e. low average balance. Economically they may not be stable but have high potential of purchasing/spending because of various reasons. So, can offer a loan to increase their spending. Offering loan to these customers will not make much difference as they have high balance anyway. They are 69 customers.

- **Cluster 2:** These customers are Small Spenders and has lowest credit limit., but minimum paid by the customer while making payments for purchases made monthly are highest than customers in cluster 1 & 2. So, can promote various monthly shopping schemes as well as loans to these customers to increase their monthly shopping. They are 77 customers.

1.1.7 Conclusion:

- In this project, we went through the customer segmentation model. We developed this using a class of machine learning known as unsupervised learning. Specifically, we made use of a clustering algorithm called K-means clustering. We analysed and visualized the data and then proceeded to implement our algorithm. With the help of clustering, we can understand the variables much better, prompting us to take careful decisions. With the identification of customers, bank can release different promotional schemes that target customers based on several parameters like credit limit, payments, spending patterns, etc. Furthermore, more complex patterns like income, shopping patterns are taken into consideration for better segmentation.
- The K-means clustering algorithm is widely used for clustering huge data sets. But traditional k means algorithm does not always generate good quality results as automatic initialization of centroids affects final clusters. This analysis presents an efficient algorithm where we have first pre-processed our dataset based on normalization technique and then generated effective clusters. This is done by assigning weights to each attribute value to achieve standardization. Our algorithm has proved to be better than traditional K-means algorithm in terms of execution time and speed.

2 CART-RF-ANN (Supervised Learning)

Data Mining Phases:

a) Define the Problem
b) Identify Required Data
c) Prepare and pre-process
d) Model the Data
e) Train and Test
f) Verify and deploy

2.1 Problem Statement 2:

2.1.1 Define the Problem

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.

Dataset: **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 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)

2.1.2 Identify Required Data

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

	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

2.1.3 Exploratory Data Analysis:

- 'Claimed' is the target variable while all others are the predictors.
- There are total 3000 rows and 10 columns in the dataset
- Data types of each attribute/variables are as follows:

```
Age              float64
Agency_Code       object
Type              object
Claimed           object
Commision        float64
Channel           object
Duration         float64
Sales            float64
Product Name      object
Destination       object
```

Inference:

- Out of the 10 columns, 6 are object type, 2 are int and while remaining 2 are float type.

Summary of the data:

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Age	3000	NaN	NaN	NaN	38.091	10.4635	8	32	36	42	84
Agency_Code	3000	4	EPX	1365	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Type	3000	2	Travel Agency	1837	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Claimed	3000	2	No	2076	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Commision	3000	NaN	NaN	NaN	14.5292	25.4815	0	0	4.63	17.235	210.21
Channel	3000	2	Online	2954	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Duration	3000	NaN	NaN	NaN	70.0013	134.053	-1	11	26.5	63	4580
Sales	3000	NaN	NaN	NaN	60.2499	70.734	0	20	33	69	539
Product Name	3000	5	Customised Plan	1136	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Destination	3000	3	ASIA	2465	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Getting unique counts of all Nominal Variables

TYPE: 2		CLAIMED: 2		CHANNEL: 2	
Airlines	1163	Yes	924	Offline	46
Travel Agency	1837	No	2076	Online	2954

PRODUCT NAME: 5		AGENCY_CODE: 4		DESTINATION: 3	
Gold Plan	109	JZI	239	EUROPE	215
Silver Plan	427	CWT	472	Americas	320
Bronze Plan	650	C2B	924	ASIA	2465
Cancellation Plan	678	EPX	1365		
Customised Plan	1136				

Inference:

- There are no ? or other charecter present.
- All nominal values have 2 to 5 categories which can be included in dataset for prediction.

2.1.4 Prepare and pre-process

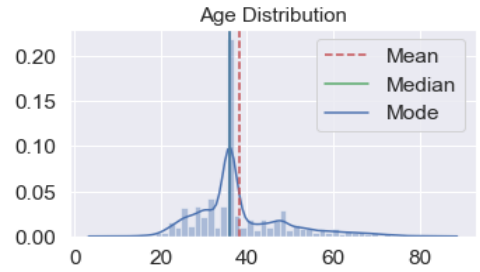
- There are 0 null values present in the dataset
- Number of duplicate rows = 139

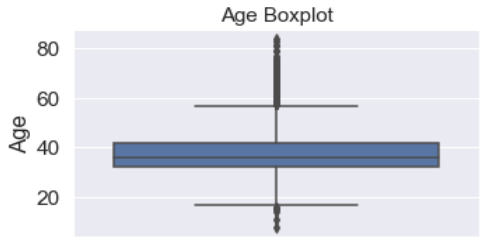
These duplicates may have different packages as it may happen that the agency is sellin g Travel Packages to similar customers in similar price tag. That way there could be du plicate in the record level, however, they are not true duplicates. Hence, we will not dro p it.

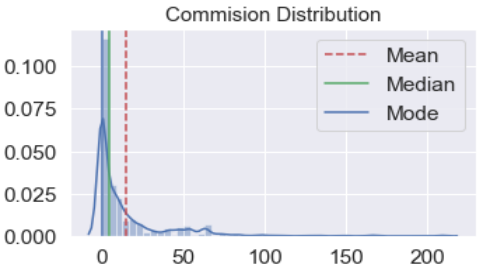
- 5-point summary of dataset

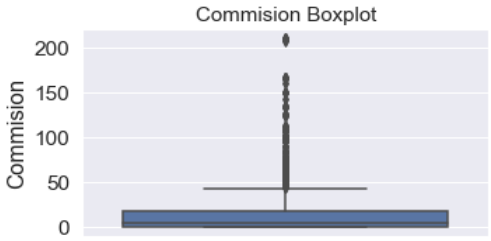
	count	mean	std	min	25%	50%	75%	max
Age	2861.0	38.204124	10.678106	8.0	31.0	36.00	43.00	84.00
Agency_Code	2861.0	1.280671	1.003773	0.0	0.0	2.00	2.00	3.00
Type	2861.0	0.597344	0.490518	0.0	0.0	1.00	1.00	1.00
Claimed	2861.0	0.319469	0.466352	0.0	0.0	0.00	1.00	1.00
Commision	2861.0	15.080996	25.826834	0.0	0.0	5.63	17.82	210.21
Channel	2861.0	0.983922	0.125799	0.0	1.0	1.00	1.00	1.00
Duration	2861.0	72.120238	135.977200	-1.0	12.0	28.00	66.00	4580.00
Sales	2861.0	61.757878	71.399740	0.0	20.0	33.50	69.30	539.00
Product Name	2861.0	1.666550	1.277822	0.0	1.0	2.00	2.00	4.00
Destination	2861.0	0.261797	0.586239	0.0	0.0	0.00	0.00	2.00

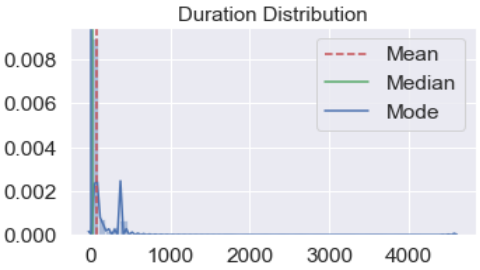
2.1.5 Univariate Analysis:

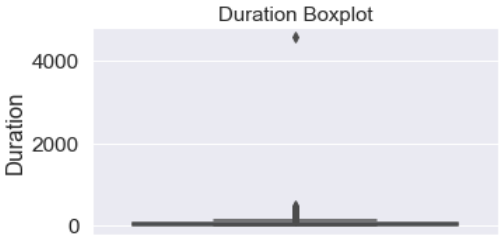


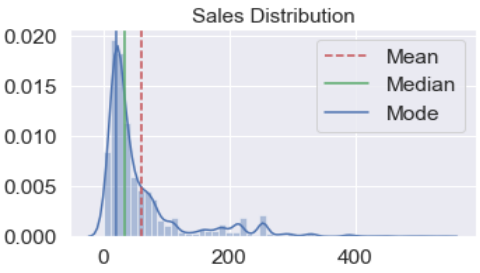


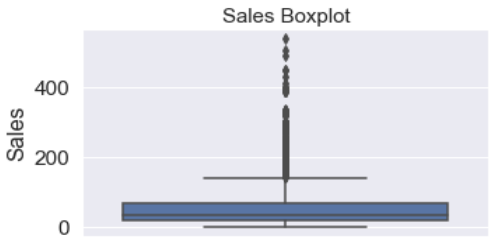












Inference:

- For all the above variables mean is greater than median, hence it is positively skewed.
- "Sales", "Commission" and "Duration" all 3 variables are positively Skewed with too many outliers
- "Age" variable is little right skewed with too many outliers.

Skewness:

- Age = 1.149
- Commision = 3.147
- Duration = 13.777 (high)
- Sales = 2.379

- There are outliers in all the variables. CART and Random Forest are robust to outliers. Neural Networks can handle outliers if there are more hidden layers and if the number of outliers is lesser. For now, we are treating the outliers using IQR.

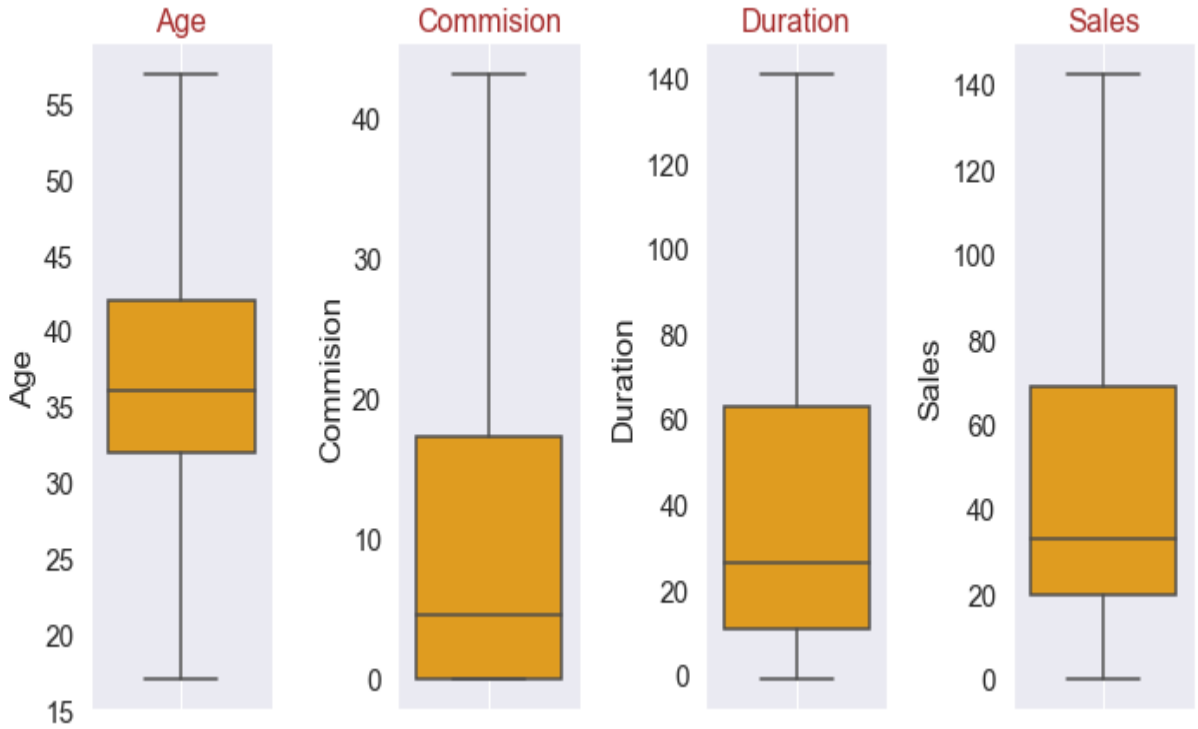
2.1.6 Outlier treatment:

Here we define a custom function in which we find the max and min value i.e.

- Min value = Q1-(1.5 * IQR)
- Max value = Q3+(1.5 * IQR)

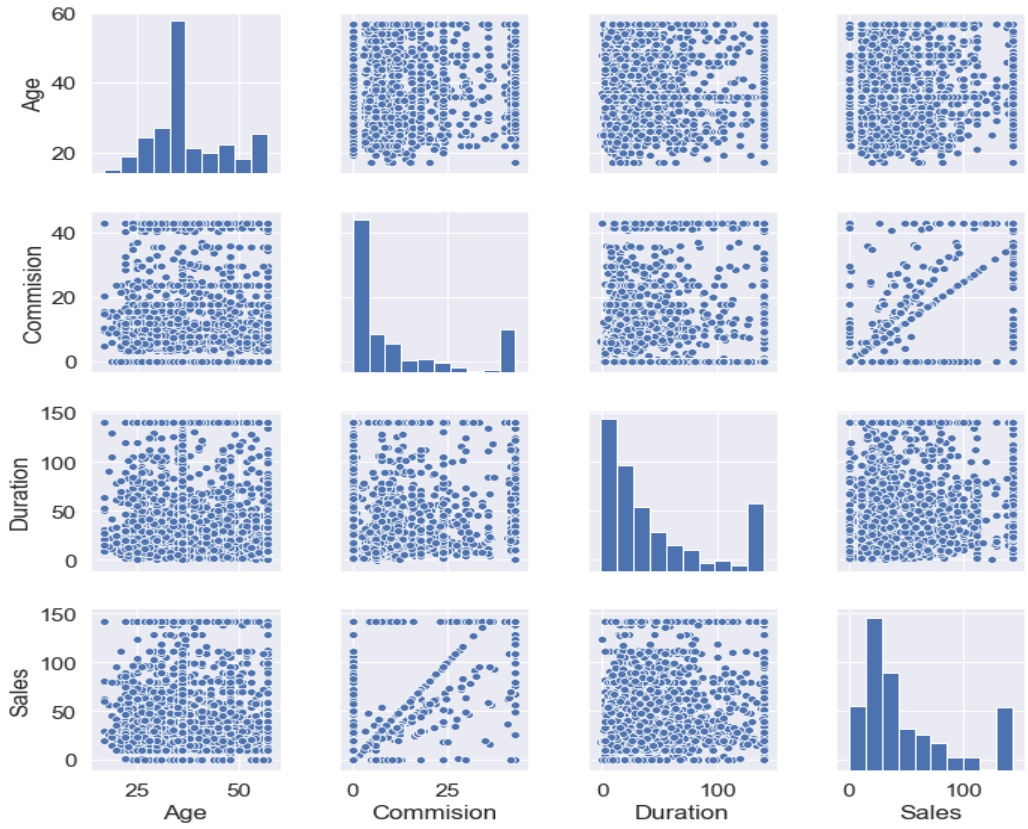
If for a column any value, is beyond the range of – 1.5 x IQR to 1.5 x IQR then that value is assigned to max value or the min value accordingly.

All the outlier is replaced by the low and high value of IQR which is nothing but one type of outlier treatment.



2.1.7 Bi-Variate Analysis:

- **Checking pairwise distribution of the continuous** variables



There are no correlations between the features

- **Checking for Correlations**

Inference:

There are mostly positive correlations between variables. Overall, the magnitude of correlations between the variables are very less.

Converting Object data type into Categorical

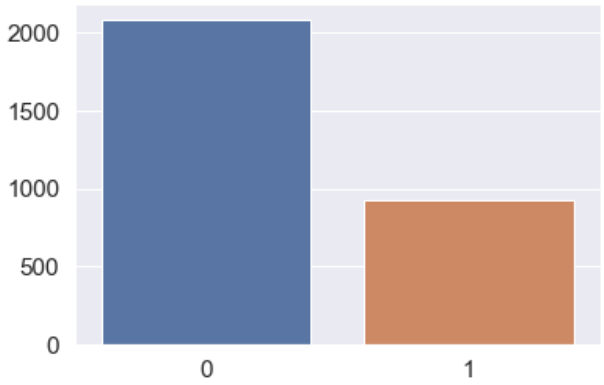
6 columns are of type object i.e. strings. These need to be converted to ordinal type.

TYPE: 2		CLAIMED: 2		CHANNEL: 2	
Airlines	0	No	0	Offline	0
Travel Agency	1	Yes	1	Online	1

PRODUCT NAME: 5		AGENCY_CODE: 4		DESTINATION: 3	
Bronze Plan	0	C2B	0	ASIA	0
Cancellation Plan	1	CWT	1	Americas	1
Customised Plan	2	EPX	2	EUROPE	2
Gold Plan	3	JZI	3		
Silver Plan	4				

	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
0	48.0	0	0	0	0.70	1	7.0	2.51	2	0
1	36.0	2	1	0	0.00	1	34.0	20.00	2	0
2	39.0	1	1	0	5.94	1	3.0	9.90	2	1
3	36.0	2	1	0	0.00	1	4.0	26.00	1	0
4	33.0	3	0	0	6.30	1	53.0	18.00	0	0

Proportion of observations in Target (Claimed) classes




```
0    69.2%
1    30.8%
```

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

2.1.8 Extract x and y

Extracting the target column into separate vectors for training set and test set.

- **Splitting the data into Train and Test set**

  - Training Dataset: The sample of data used to fit the model.
  - Test Dataset: The sample of data used to provide an unbiased evaluation of a final model fit on the training dataset.

- **Checking the dimensions of the training and test data**
  - `x_train (2100, 9)`
  - `x_test (900, 9)`
  - `y_train (2100,)`
  - `y_test (900,)`

- **Proportion of values in target's train & test sets**

  - `y_train`

```
0    0.697143
1    0.302857
Name: Claimed, dtype: float64
```

  - `y_test`

```
0    0.68
1    0.32
Name: Claimed, dtype: float64
```

Observations are almost equally distributed between the train and test sets w.r.t target classes.

2.1.9 Feature Scaling

The variables of the data set are of different scales i.e. one variable is in thousands and other in tens. For e.g. in our data set Duration is having values in thousands and age in just two digits. Since the data in these variables are of different scales, it is tough to compare these variables.

Feature scaling is the method used to standardize the range of features of data. Since, the range of values of data may vary widely, it becomes a necessary step in data pre-processing while using machine learning algorithms.

There are different ways of scaling, but here we are using Standard Scaler. In this method, we convert variables with different scales of measurements into a single scale.

Standard Scaler

- This standard scaler standardizes the features by removing the mean and scaling to unit variance.
- Scales the features such that the distribution is centred around 0, with a standard deviation of 1.
- The standard score of a sample x is calculated as:

$$z = (x - u) / s$$

where,

'u' is the mean of the training samples or zero if with_mean = False,

's' is the standard deviation of the training samples or one if with_std = False.

Inference:

Many machine learning algorithms perform better or converge faster when features are on a relatively similar scale and/or close to normally distributed.

2.1.10 Model the Data

2.1.11 Decision Tree Classifier:

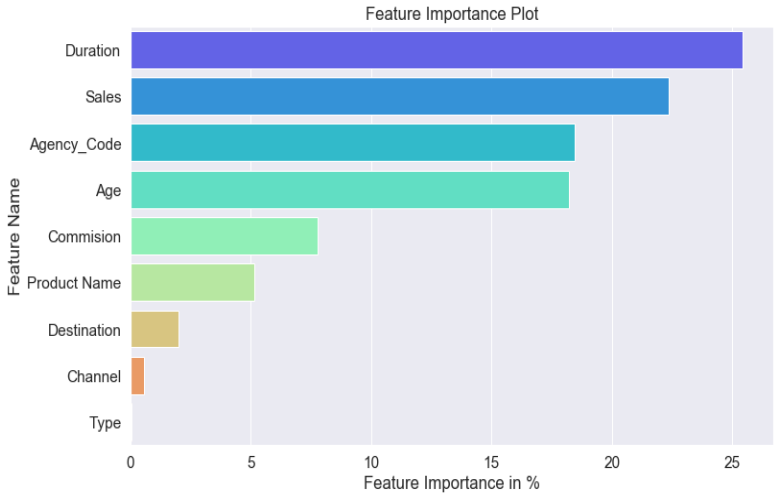
A decision tree is built on an entire dataset, using all the features/variables of interest.

2.1.11.1 CART Default Model:

Model parameters:

```
DecisionTreeClassifier(class_weight=None, criterion='gini',max_depth=None,
                       max_features=None, max_leaf_nodes=None, splitter='best',
                       random_state=0, min_weight_fraction_leaf=0.0,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2, presort=False,)
```

Important Features for DT model:



	Imp
Duration	0.254236
Sales	0.223650
Agency_Code	0.184451
Age	0.182367
Commision	0.077909
Product Name	0.051373
Destination	0.020003
Channel	0.005716
Type	0.000297

Inference:

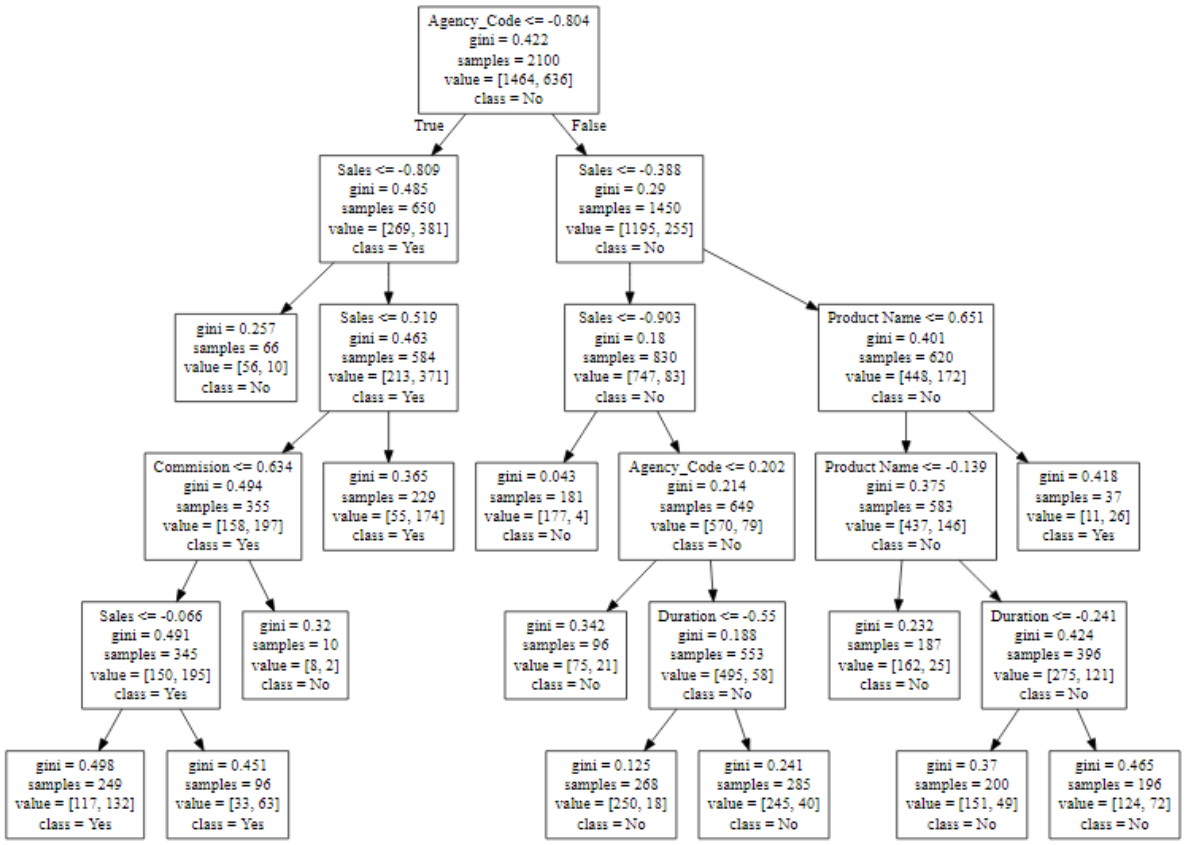
- Duration, Sales, Agency Code and Age are the most important features of Default CART Model

2.1.11.2 CART GridSearchCV model:

Model parameters:

```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=5,
                       max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=10, min_samples_split=300,
                       min_weight_fraction_leaf=0.0, presort=False,
                       random_state=None, splitter='best')
```

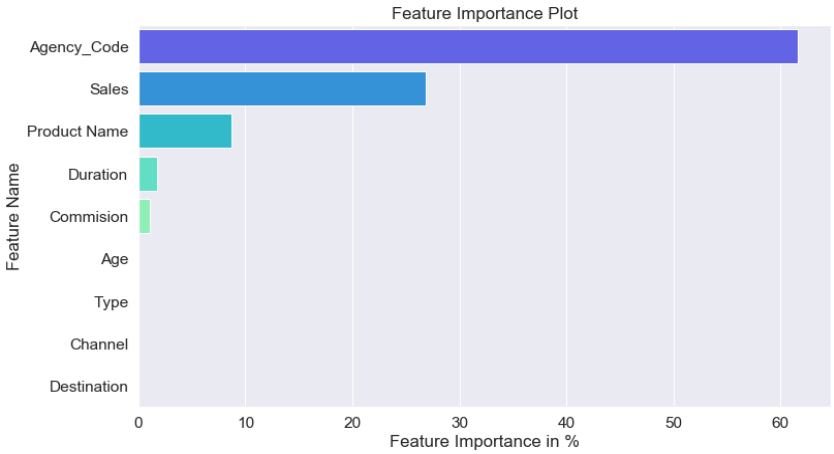
Model:



Inference:

- The 11.84% proportion of customers whose Sales is <= -0.066, and Agency code <= -0.804 shall claim the insurance.
- The 4.5% proportion of customers whose Sales is > -0.066, and Commision <= 0.634 shall claim the insurance.

Important Features for DT model:



	Imp
Agency_Code	0.615865
Sales	0.268488
Product Name	0.087369
Duration	0.017858
Commision	0.010420
Age	0.000000
Type	0.000000
Channel	0.000000
Destination	0.000000

Inference:

- Agency Code, Sales and Product Name are the most important features of CART Model
- Age, Type, Channel and Destination are of no use.

2.1.12 Random Forest Classifier:

Random forest randomly selects observations/rows and specific features/variables to build multiple decision trees from and then averages the results. It is an ensemble of Decision Trees whereby the final/leaf node will be either the majority class for classification problems or the average for regression problems. A random forest will grow many Classification trees and for each output from that tree, we say the tree 'votes' for that class. A tree is grown using the following steps:

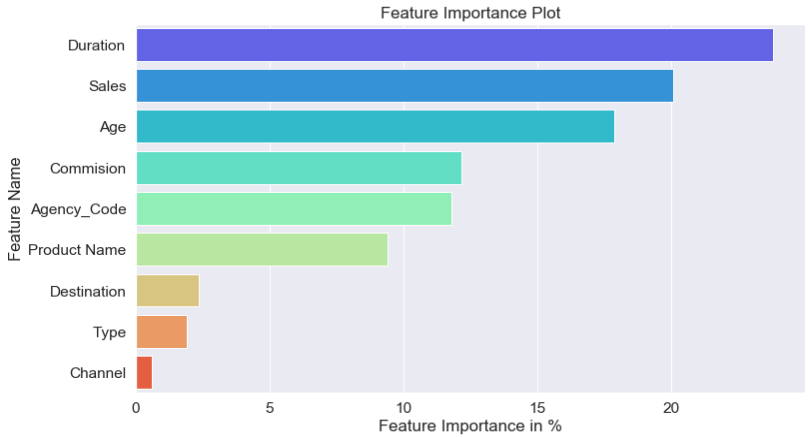
1. A random sample of rows from the training data will be taken for each tree.
2. From the sample taken, a subset of features will be taken to be used for splitting on each tree.
3. Each tree is grown to the largest extent specified by the parameters until it reaches a vote for the class.

2.1.12.1 RF Default Model:

Model parameters:

```
RandomForestClassifier(bootstrap=True, class_weight=None, n_jobs=None,
            max_depth=None, max_features='auto', max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=3, criterion='gini',
            min_weight_fraction_leaf=0.0, n_estimators=100,
            oob_score=False, random_state=0, verbose=0,warm_start=False)
```

Important Features for RF model:



	Imp
Duration	0.238014
Sales	0.200981
Age	0.178792
Commision	0.121732
Agency_Code	0.117890
Product Name	0.094095
Destination	0.023703
Type	0.018915
Channel	0.005879

Inference:

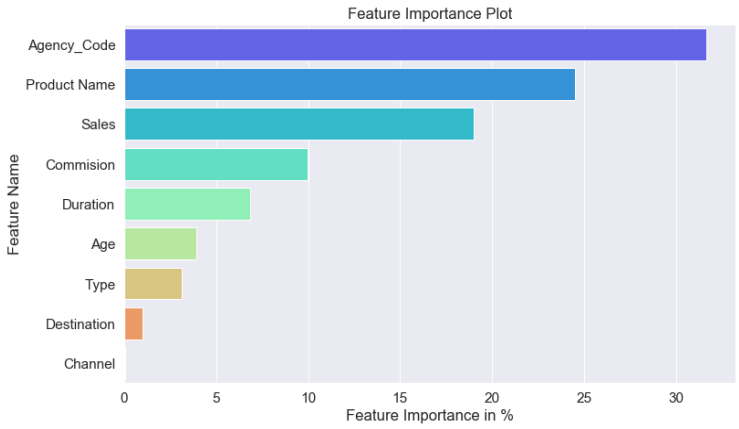
- Duration, Sales, Age and Commision are the most important features of Default RF Model

2.1.12.2 RF GridSearchCV model:

Model parameters:

```
RandomForestClassifier(bootstrap=True, class_weight=None,criterion='gini',
                       max_depth=30, max_features=5, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=20, min_samples_split=70,
                       min_weight_fraction_leaf=0.0, n_estimators=500,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```

Important Features for RF model:



	Imp
Agency_Code	0.316341
Product Name	0.244887
Sales	0.189827
Commision	0.099599
Duration	0.068322
Age	0.039380
Type	0.031093
Destination	0.010029
Channel	0.000521

Inference:

- Agency Code, Product Name and Sales are the most important features of RF Model.

2.1.13 Artificial Neural Network:

Artificial Neural Networks are organized in layers made up of interconnected nodes which contain an activation function that computes the output of the network. Each incoming data point receives a weight and is multiplied and added. A bias is added if the weighted sum equates to zero and then passed to the activation function.

2.1.13.1 NN Default Model:

Model parameters:

```
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=500, learning_rate='constant',
              learning_rate_init=0.001, max_iter=500, momentum=0.9,
              n_iter_no_change=10, nesterovs_momentum=True, power_t=0.5,
              random_state=0, shuffle=True, solver='sgd', tol=0.0001,
              validation_fraction=0.1, verbose=False, warm_start=False)
```

Inference:

- NN Model works like a black box and figures out how to perform their functions on their own. It determines their functions based on the sample input.
- The important features used cannot be seen.

2.1.13.2 NN GridSearchCV Model:

Model parameters:

```
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=400, learning_rate='constant',
              learning_rate_init=0.001, max_iter=1000, momentum=0.9,
              n_iter_no_change=10, nesterovs_momentum=True, power_t=0.5,
              random_state=None, shuffle=True, solver='adam', tol=0.01,
              validation_fraction=0.1, verbose=False, warm_start=False)
```

Inference:

- In NN GridSearchCV Model we set some parameters and check the performances.
- The important features used cannot be seen.

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.

2.1.14 Train and Test

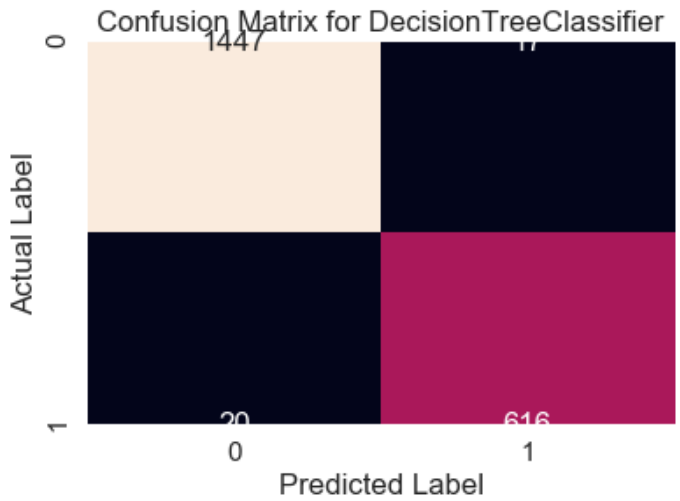
2.1.14.1 CART Default Model: (Train Dataset)

- **Getting the Predicted Classes and Probs**

	0	1
0	1.0	0.0
1	0.0	1.0
2	1.0	0.0
3	0.0	1.0
4	0.0	1.0

- Accuracy for CART default Train model is **98.23%**
- Classification report for Decision Tree default model is

	precision	recall	f1-score	support
0	0.99	0.99	0.99	1464
1	**0.97**	**0.97**	**0.97**	636
accuracy			0.98	2100
macro avg	0.98	0.98	0.98	2100
weighted avg	0.98	0.98	0.98	2100



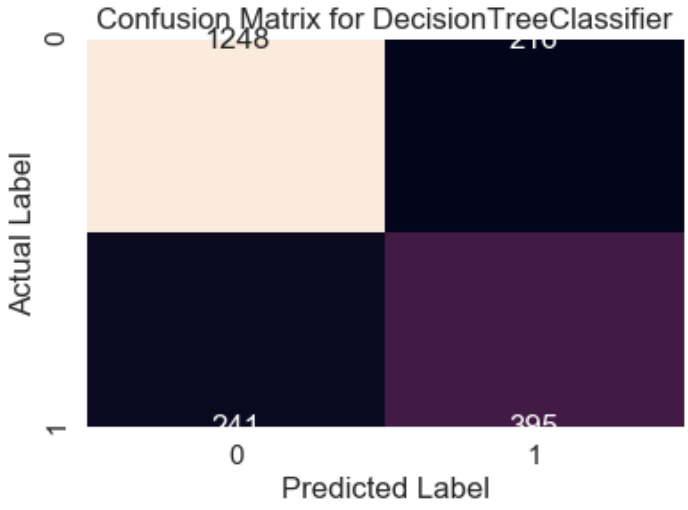
2.1.14.2 *CART GridSearchCV Model: (Train Dataset)*

- **Getting the Predicted Classes and Probs**

	0	1
0	0.859649	0.140351
1	0.240175	0.759825
2	0.932836	0.067164
3	0.469880	0.530120
4	0.755000	0.245000

- Accuracy for CART GridSearchCV Train model is **78.23%**
- Classification report for Decision Tree Classifier GridSearchCV model is

	precision	recall	f1-score	support
0	0.84	0.85	0.85	1464
1	**0.65**	**0.62**	**0.63**	636
accuracy			0.78	2100
macro avg	0.74	0.74	0.74	2100
weighted avg	0.78	0.78	0.78	2100



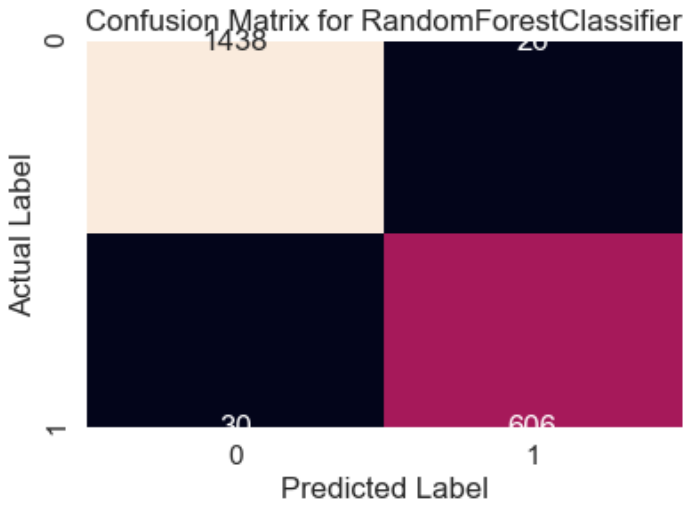
2.1.14.3 *RF Default Model: (Train Dataset)*

- **Getting the Predicted Classes and Probs**

	0	1
0	0.969500	0.030500
1	0.283333	0.716667
2	0.972667	0.027333
3	0.395667	0.604333
4	0.516905	0.483095

- Accuracy for RF default Train model is 9**7.33%**
- Classification report for Random Forest default model is

	precision	recall	f1-score	support
0	0.98	0.98	0.98	1464
1	0.96	0.95	0.96	636
accuracy			0.97	2100
macro avg	0.97	0.97	0.97	2100
weighted avg	0.97	0.97	0.97	2100



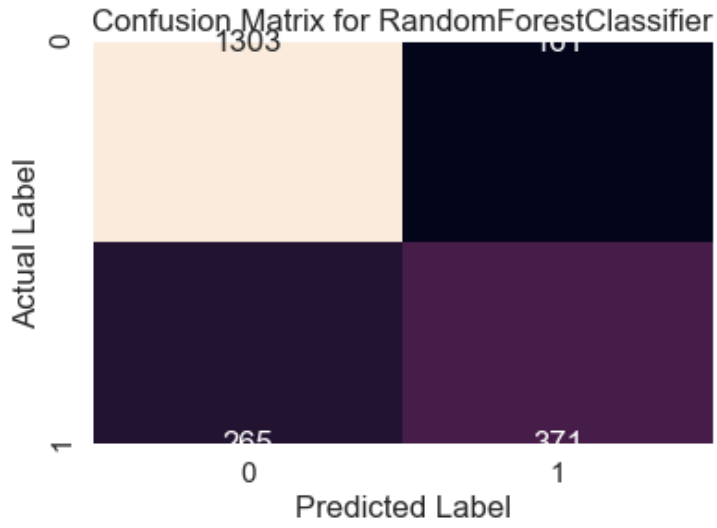
2.1.14.4 *RF GridSearchCV Model: (Train Dataset)*

- **Getting the Predicted Classes and Probs**

	0	1
0	0.919450	0.080550
1	0.275905	0.724095
2	0.946276	0.053724
3	0.496423	0.503577
4	0.770785	0.229215

- Accuracy for RF GridSearchCV Train model is **79.71%**
- Classification report for Random Forest GridSearchCV model is

	precision	recall	f1-score	support
0	0.83	0.89	0.86	1464
1	0.70	0.58	0.64	636
accuracy			0.80	2100
macro avg	0.76	0.74	0.75	2100
weighted avg	0.79	0.80	0.79	2100



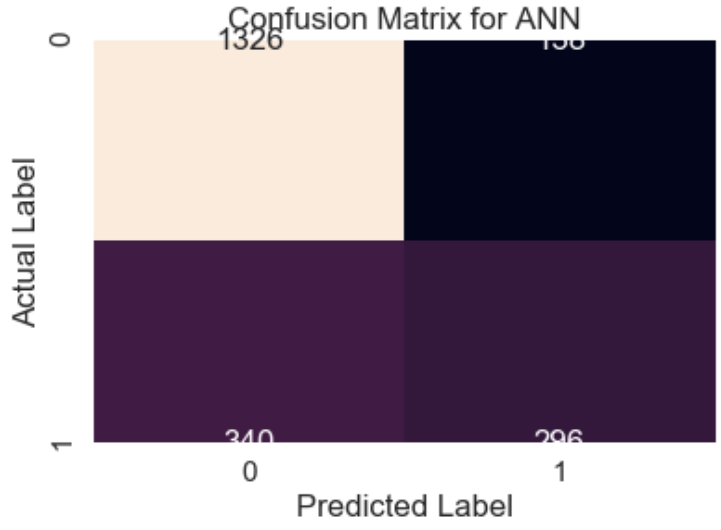
2.1.14.5 NN Default Model: (Train Dataset)

- **Getting the Predicted Classes and Probs**

	0	1
0	0.899593	0.100407
1	0.283410	0.716590
2	0.890750	0.109250
3	0.544896	0.455104
4	0.747059	0.252941

- Accuracy for NN default Train model is **77.23%**
- Classification report for NN default model is

	precision	recall	f1-score	support
0	0.80	0.91	0.85	1464
1	0.68	0.47	0.55	636
accuracy			0.77	2100
macro avg	0.74	0.69	0.70	2100
weighted avg	0.76	0.77	0.76	2100



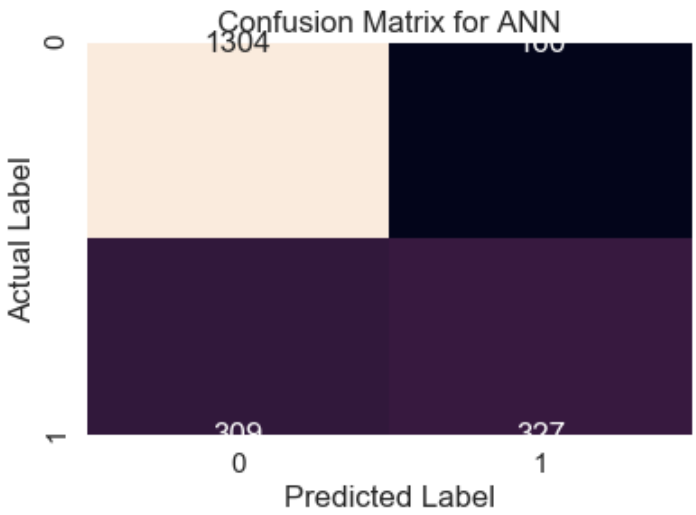
2.1.14.6 *NN GridSearchCV Model: (Train Dataset)*

- **Getting the Predicted Classes and Probs**

	0	1
0	0.907809	0.092191
1	0.234242	0.765758
2	0.915977	0.084023
3	0.510697	0.489303
4	0.777595	0.222405

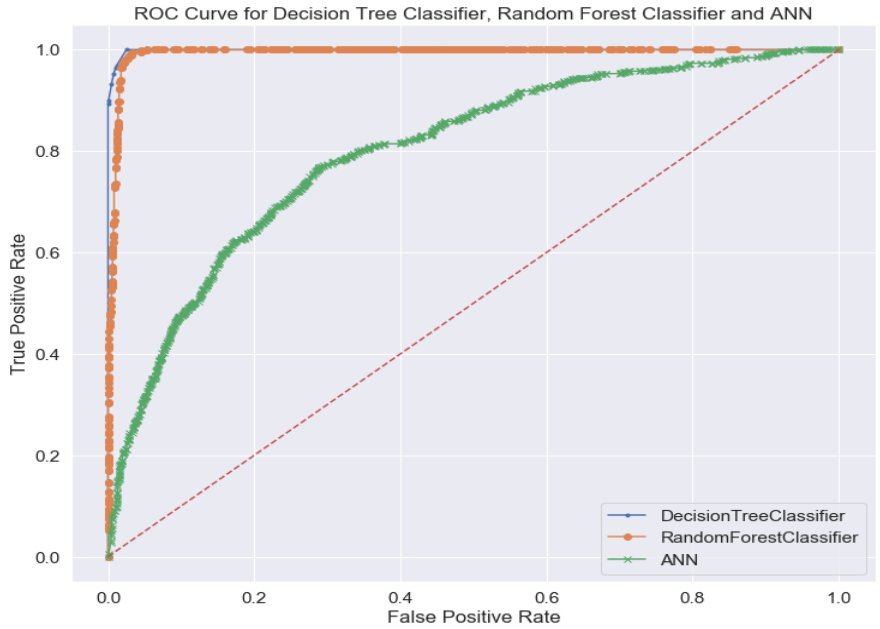
- Accuracy for NN GridSearchCV Train model is **77.66%**
- Classification report for NN GridSearchCV model is

	precision	recall	f1-score	support
0	0.81	0.89	0.85	1464
1	0.67	0.51	0.58	636
accuracy			0.78	2100
macro avg	0.74	0.70	0.71	2100
weighted avg	0.77	0.78	0.77	2100



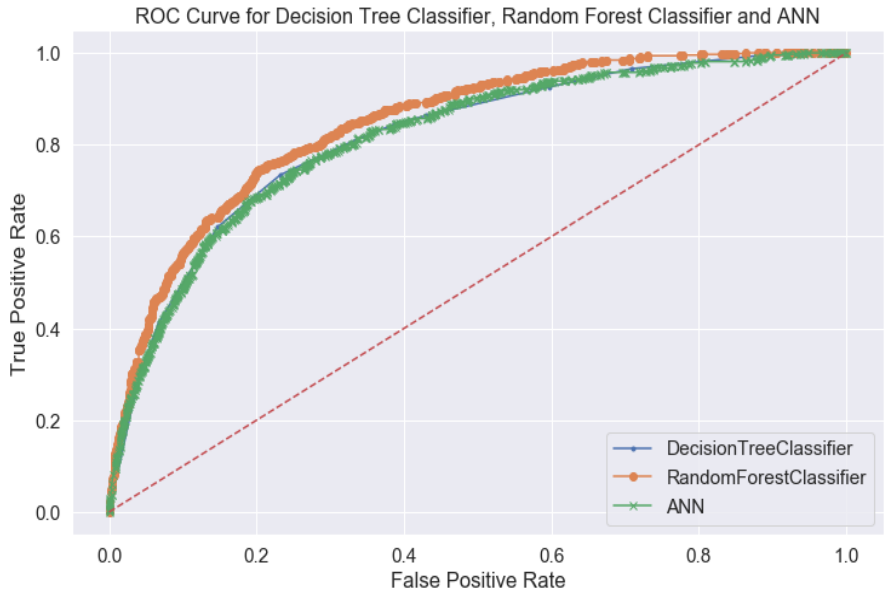
2.1.15 ROC curves and AUC for all Default models for the training data

- AUC for Decision Tree Classification Train Model is **99.90%**
- AUC for Random Forest Classification Train Model is **99.41%**
- AUC for Artificial Neural Network Train Model is **80.03%**



2.1.16 ROC curves and AUC for all GridSearchCV models for the training data

- AUC for Decision Tree Classification Train Model is **81.59%**
- AUC for Random Forest Classification Train Model is **84.59%**
- AUC for Artificial Neural Network Train Model is **81.53%**



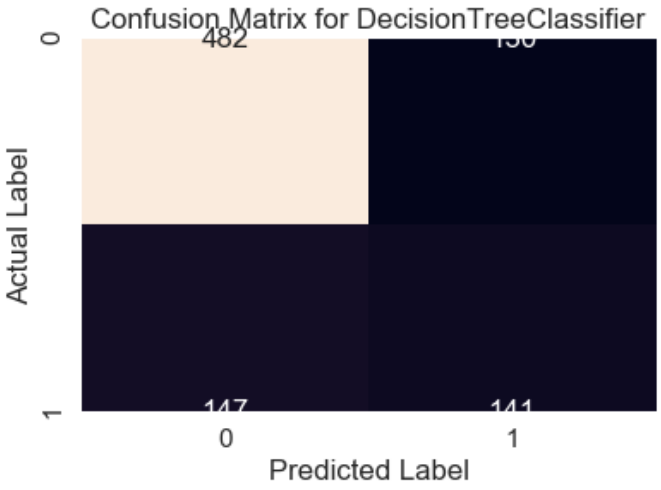
2.1.17 Verify and deploy: Model Evaluation

Comparing Default Models on the test set

2.1.17.1 CART Default Model: (Test Dataset)

- Accuracy for CART default Test model is **69.22%**
- Classification report for Decision Tree default model is

	precision	recall	f1-score	support
0	0.77	0.79	0.78	612
1	0.52	0.49	0.50	288
accuracy			0.69	900
macro avg	0.64	0.64	0.64	900
weighted avg	0.69	0.69	0.69	900

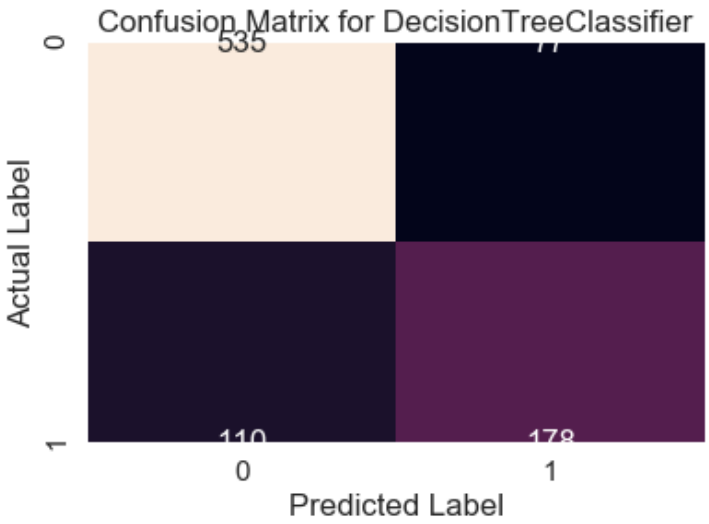



```
[482 130]
[147 141]
```

2.1.17.2 CART GridSearchCV Model: (Test Dataset)

- Accuracy for CART GridSearchCV Test model is **79.22%**
- Classification report for Decision Tree Classifier GridSearchCV model is

	precision	recall	f1-score	support
0	0.83	0.87	0.85	612
1	0.70	0.62	0.66	288
accuracy			0.79	900
macro avg	0.76	0.75	0.75	900
weighted avg	0.79	0.79	0.79	900

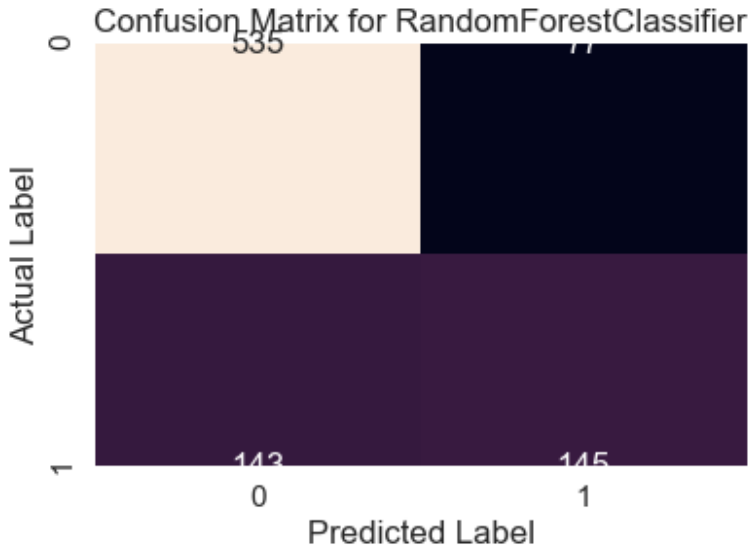



```
[535  77]
[110 178]
```

2.1.17.3 RF Default Model: (Test Dataset)

- Accuracy for RF default Test model is **75.55%**
- Classification report for Random Forest default model is

	precision	recall	f1-score	support
0	0.79	0.87	0.83	612
1	0.65	0.50	0.57	288
accuracy			0.76	900
macro avg	0.72	0.69	0.70	900
weighted avg	0.75	0.76	0.75	900

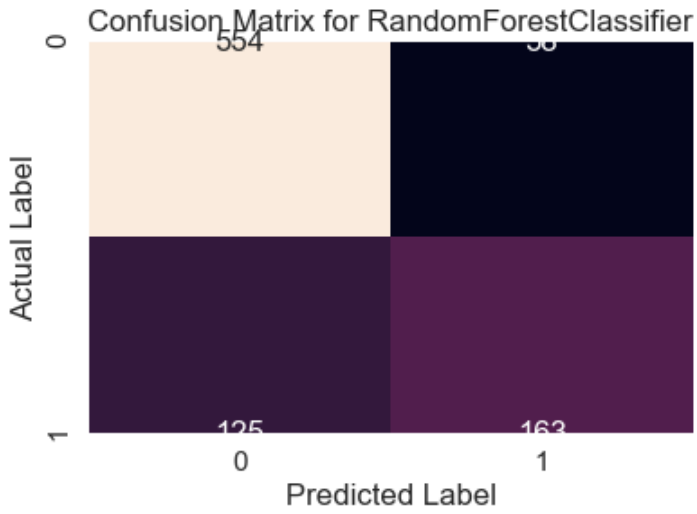



```
[535  77]
[143 145]
```

2.1.17.4 *RF GridSearchCV Model: (Test Dataset)*

- Accuracy for RF GridSearchCV Test model is **80.22%**
- Classification report for Random Forest GridSearchCV model is

	precision	recall	f1-score	support
0	0.82	0.91	0.86	612
1	0.74	0.57	0.64	288
accuracy			0.80	900
macro avg	0.78	0.74	0.75	900
weighted avg	0.79	0.80	0.79	900

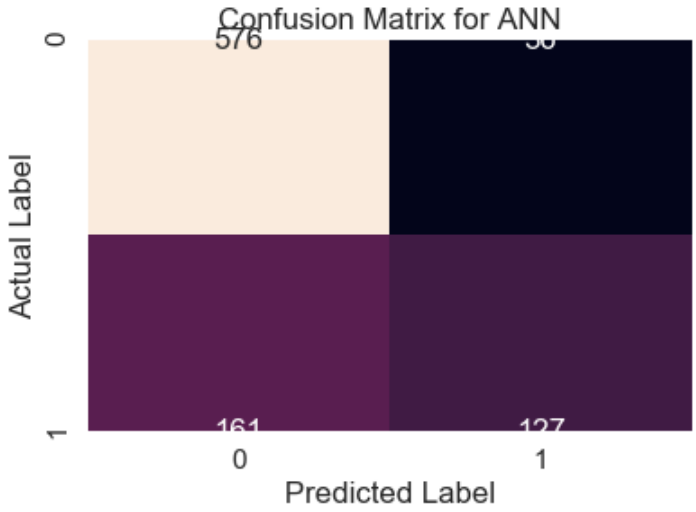



```
[554  58]
[125 163]
```

2.1.17.5 *NN Default Model: (Test Dataset)*

- Accuracy for NN default Test model is **78.11%**
- Classification report for NN default model is

	precision	recall	f1-score	support
0	0.78	0.94	0.85	612
1	0.78	0.44	0.56	288
accuracy			0.78	900
macro avg	0.78	0.69	0.71	900
weighted avg	0.78	0.78	0.76	900

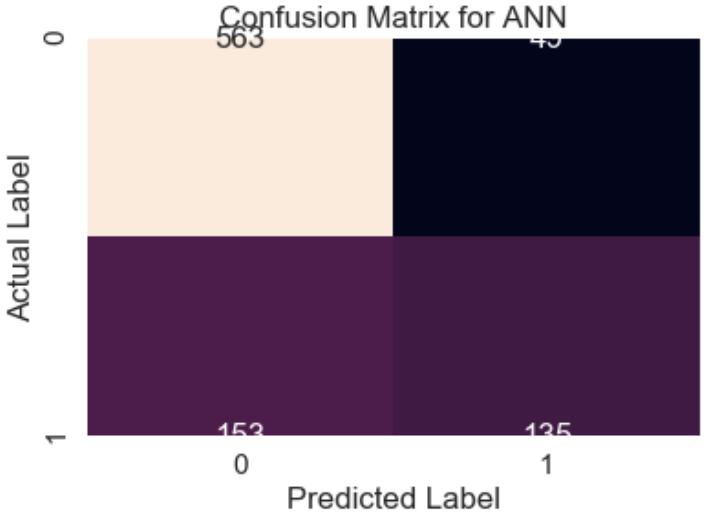



```
[576  36]
[161 127]
```

2.1.17.6 NN GridSearchCV Model: (Test Dataset)

- Accuracy for NN GridSearchCV Test model is **77.44%**
- Classification report for NN GridSearchCV model is

	precision	recall	f1-score	support
0	0.79	0.92	0.85	612
1	0.73	0.47	0.57	288
accuracy			0.78	900
macro avg	0.76	0.69	0.71	900
weighted avg	0.77	0.78	0.76	900



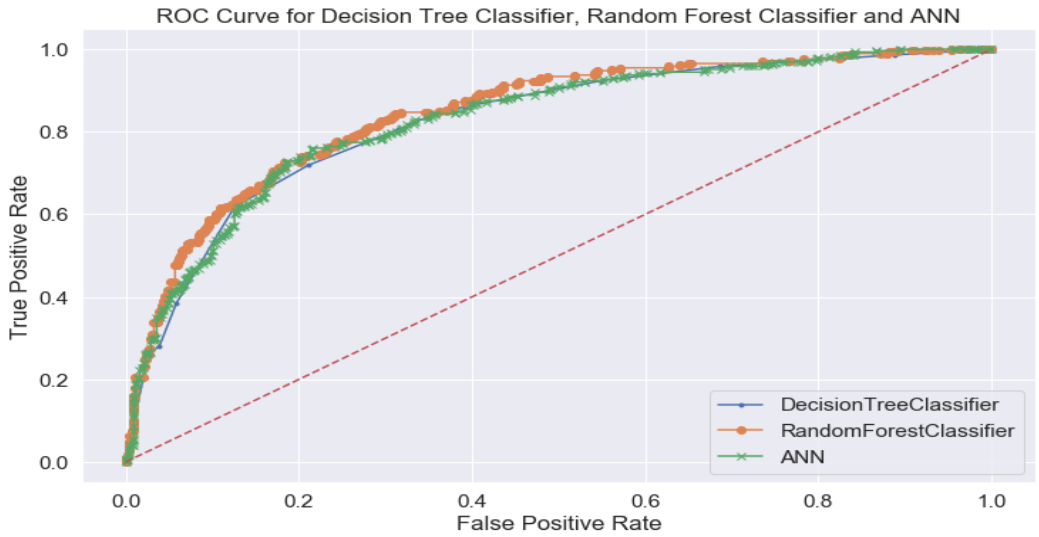

```
[563  49]
[153 135]
```

2.1.18 ROC curves and AUC for all Default models for the test data

- AUC for Decision Tree Classification Test Model is **64.38%**
- AUC for Random Forest Classification Test Model is **81.32%**
- AUC for Artificial Neural Network Test Model is **82.17%**

2.1.19 ROC curves and AUC for all GridSearchCV models for the test data

- AUC for Decision Tree Classification Test Model is **82.43%**
- AUC for Random Forest Classification Test Model is **84.48%**
- AUC for Artificial Neural Network Test Model is **82.90%**



Inference:

Random Forest model is better than the Decision Tree and ANN Since it has greater AUC on test dataset. It has higher Positive Rates even in lower threshold values.

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

	D_CART Train	D_CART Test	CART Train	CART Test	D_RF Train	D_RF Test	RF Train	RF Test	D_NN Train	D_NN Test	NN Train	NN Test
Accuracy	0.98	0.69	0.78	0.79	0.97	0.76	0.80	0.80	0.77	0.78	0.78	0.77
AUC	1.00	0.64	0.82	0.82	0.99	0.81	0.85	0.84	0.80	0.82	0.82	0.83
Recall	0.97	0.49	0.62	0.62	0.95	0.50	0.58	0.57	0.47	0.44	0.51	0.49
Precision	0.97	0.52	0.65	0.70	0.96	0.65	0.70	0.75	0.68	0.78	0.67	0.71
F1 Score	0.97	0.50	0.63	0.66	0.96	0.57	0.64	0.65	0.55	0.56	0.58	0.58

Inference:

Default Models: (D_Cart,D_RF and D_NN models)

- Area under the curve of RF on the training data is 99%, which indicates very high performance that all classes have been correctly classified. Whereas on the test data model performance is with AUC 84%, which is almost 15% less than the performance of the training data.
- Since we are building a model to predict if a person will claim the insurance or not, for practical purposes, we will be more interested in correctly classifying 1 (Claimed) than 0 (Unclaimed).
- If a person unclaimed, is incorrectly predicted to be claimed, in this situation, the cost is more severe, than when we incorrectly predict a person, who actually claims, as unclaimed.
- From the Random Forest model, looking at the Accuracy, Sensitivity/Recall, Specificity, Precision and AUC, we have almost 100% results on the training data, whereas on the Test data, performance is lesser, especially in predicting Class 1. This is because overfitting has happened on the training data, and therefore the model is weak in generalizing and predicting any new data.
- From the CART model, Area under the curve of CART on the training data is 100%, looking at the Accuracy, Sensitivity/Recall, Specificity, Precision and AUC, we have almost 80% results on the training data, whereas on the Test data, performance is lesser, especially in predicting Class 1. This is also because overfitting has happened on the training data, and therefore the model is weak in generalizing and predicting any new data.
- From the NN model, Area under the curve of NN on the training data is 80%, looking at the Accuracy, Sensitivity/Recall, Specificity, Precision and AUC, we have almost 80% results on the training data as well as the test data. Therefore, the NN model is better as compared to the other default models.

GridSearchCV: (Cart, RF and NN models)

- In these models, we have hard-coded the hyper parameter values (e.g. n_estimators for Random Forest and increasing the layers in the ANN) to make better predictions. We can optimize/fine-tune the random forest model, by trying different values for the hyper parameters to see if the model performance is improving.
- Random forest will reduce variance part of error rather than bias part, so on a given training data set decision tree may be more accurate than a random forest or ANN. But on an unexpected validation data set, Random forest and ANN always wins in terms of accuracy.
- Accuracy for Decision Tree Classifier model is 79.22%
- Accuracy for Random Forest Classifier model is 79.55%
- Accuracy for ANN model is 77.66%

2.5) Inference: Basis on these predictions, what are the business insights and recommendations

Inference:

- Out of the 3 models, Random Forest has slightly better performance than the Cart and Neural network model with area under the curve (AUC) of 84.25% (82.43% and 82.90%) respectively.
- Overall, all the 3 models are reasonably stable enough to be used for making any future pr edictions. From CART and Random Forest Model, the variable "Agency_Code" is found to be the most useful feature amongst all other features for predicting if a person would cl aim or not.
- Accuracy, AUC, Precision and Recall for test data is almost in line with training data for NN and RF model and CART. This proves no overfitting or underfitting has happened, and overall, the model is a good model for classification
- **Agency Code, Sales and Product Name** (in same order of preference) are the most important variables in determining if a person will claim the insurance or not using RF model

CART - Decision Trees

- Over-fitting can occur with a flexible model like decision trees where the model which memorize the training data and learn any noise in the data as well. This will make it unable to predict the test data.

Random Forests

- RF performs internal cross-validation (i.e. using out-of-bag samples) and only has a few tuning parameters.The fundamental reason to use a random forest instead of a decision tree is to combine the predictions of many decision trees into a single model. The logic is that a single even made up of many mediocre models will still be better than one good model. Random forests are less prone to overfitting because of this.
- Random Forest is less computationally expensive and does not require a GPU to finish training. A random forest can give you a different interpretation of a decision tree but with better performance.

Artificial Neural Network:

- They keep learning until it comes out with the best set of features to obtain a satisfying predictive performance.
- If all we cared about was the prediction, a neural network would be the the best algorithm used all the time. But in an industry setting, we need a model that can give meaning to a feature/variable to stakeholders. And these stakeholders will likely be anyone other than someone with a knowledge of deep learning or machine learning.