Project Report for Data Mining

Submitted by

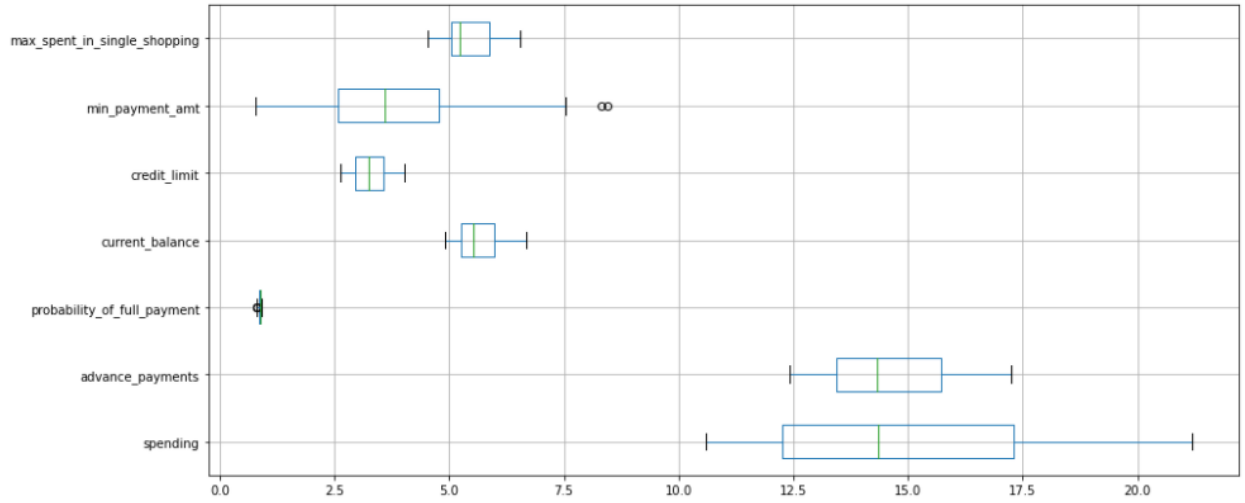
Priyadharshini K

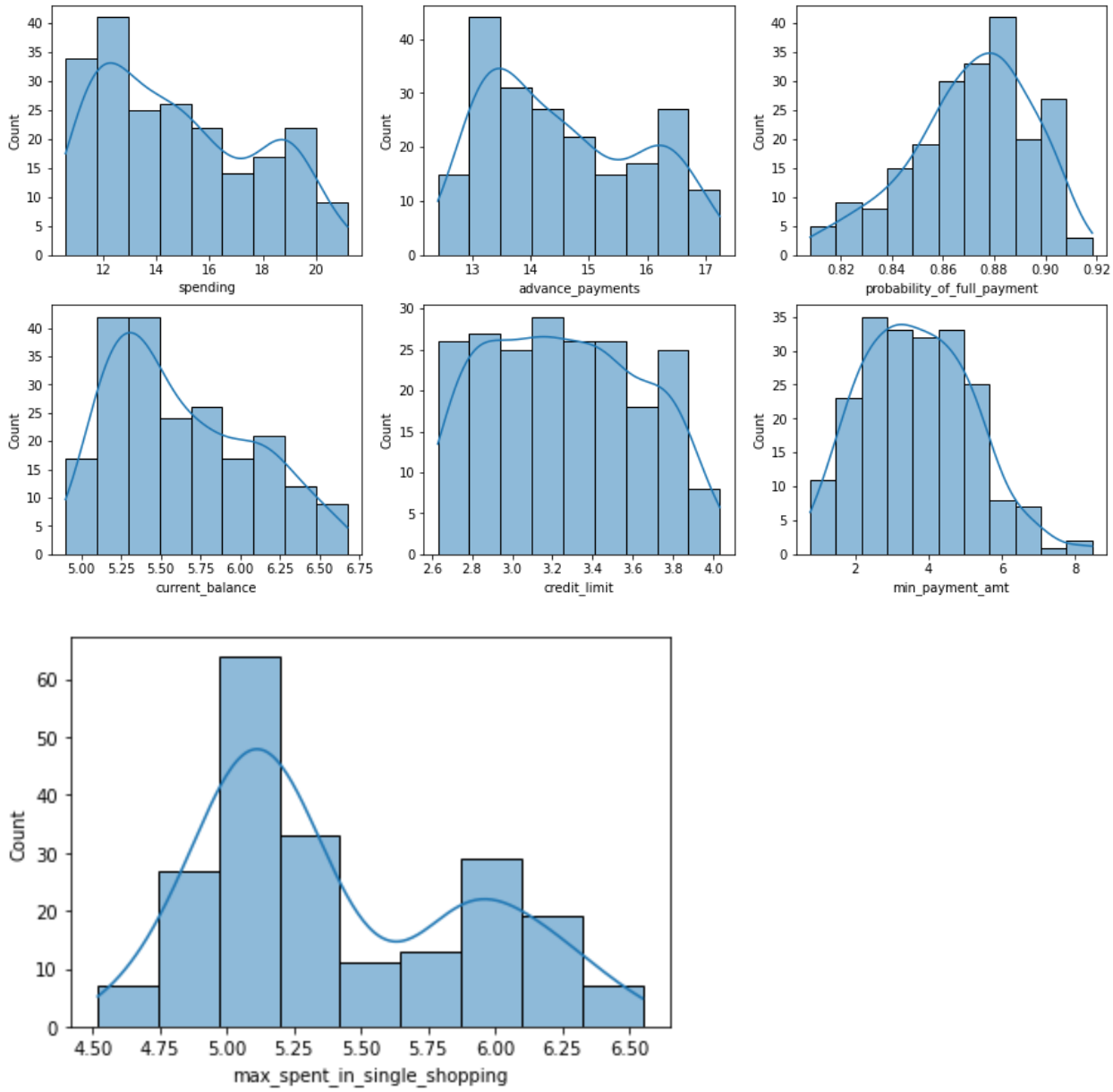
Problem 1: Clustering

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.

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

Univariate Analysis:

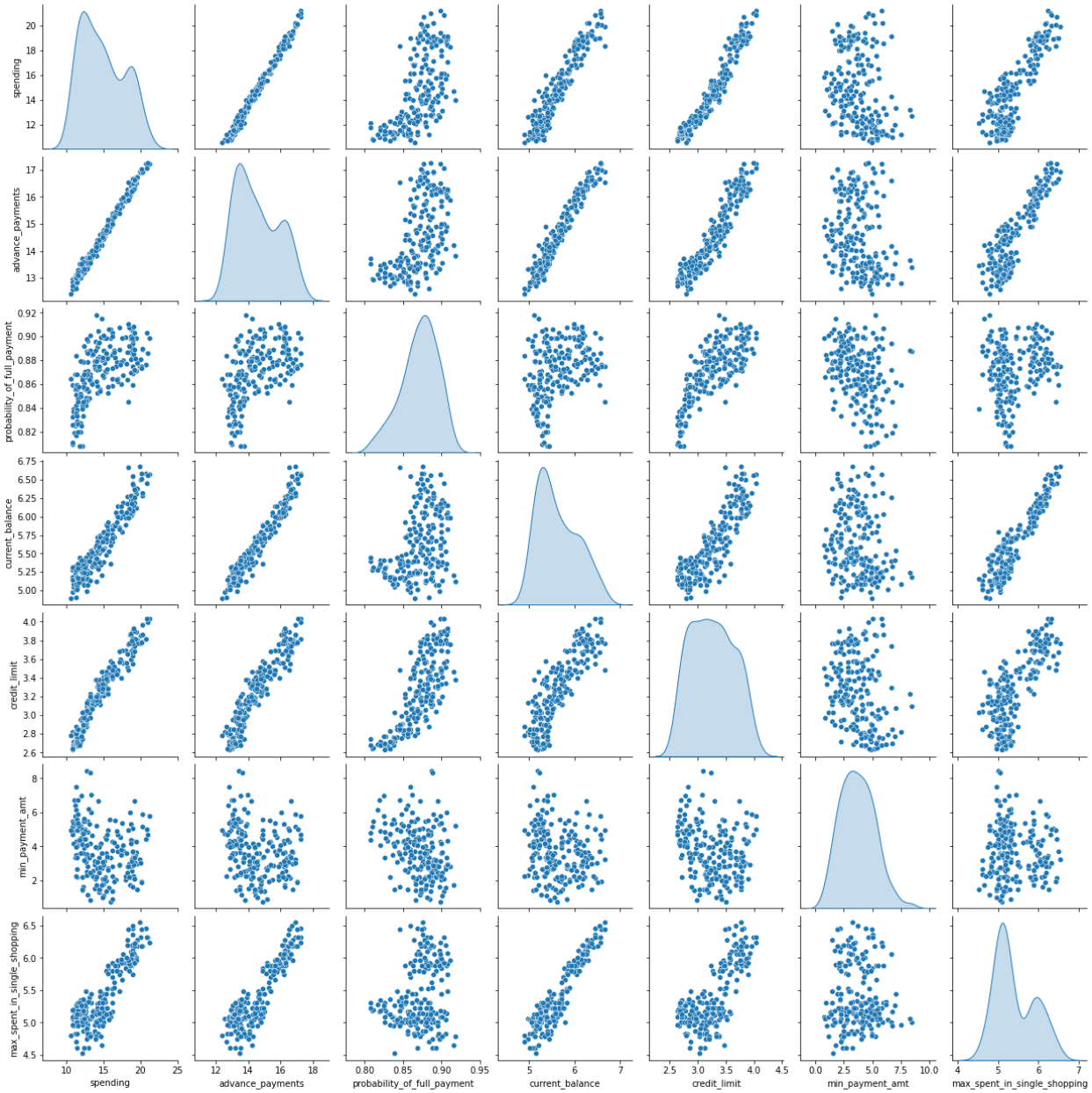


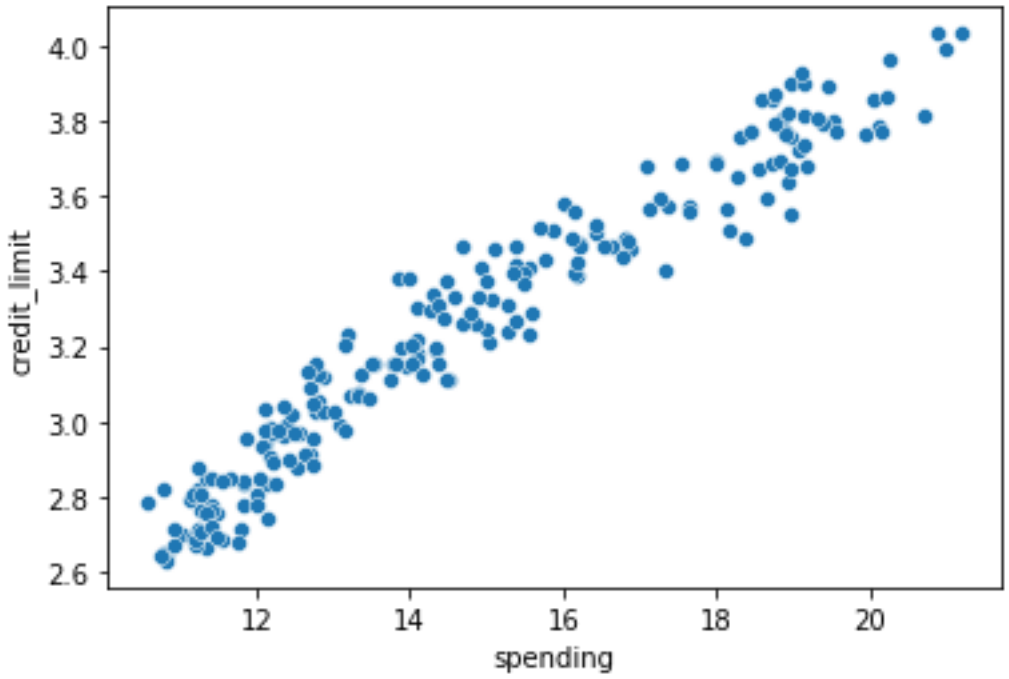


Observations from Univariate Analysis:

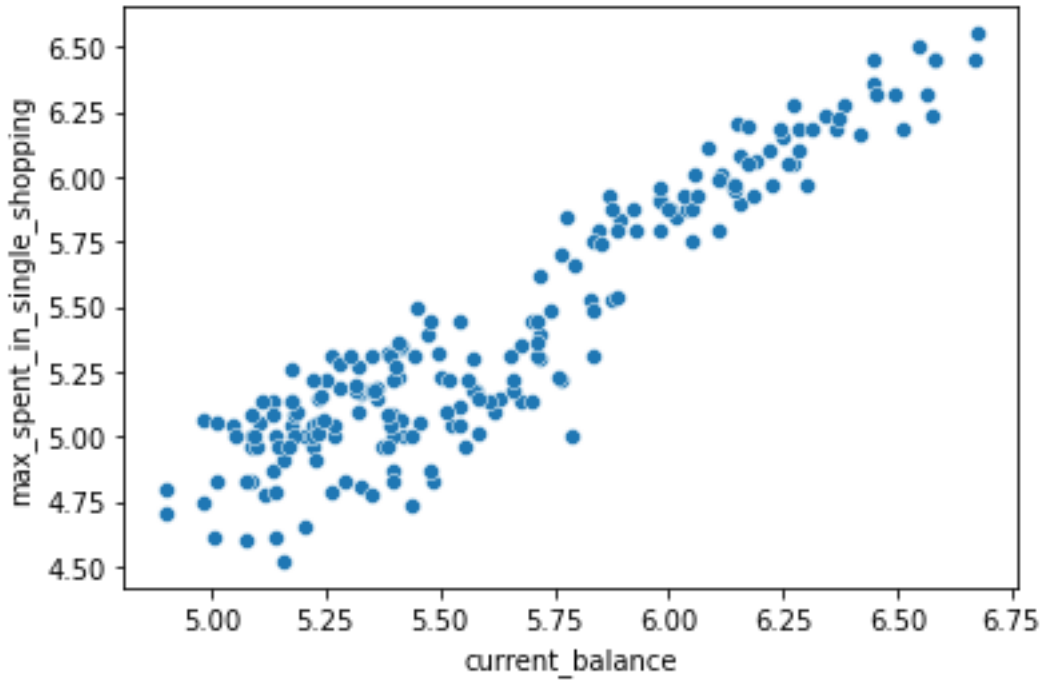
- All data variables are continuous integers and no categorical/ordinal vari able is present. Hence, regression model would be appropriate
- Outliers effect is very minimal as the dataset does not have outlier prese nce except for 'min_payment_amt' & 'probability_of_full_payment'
- Data is Normally distributed with slight skew for almost all features exce pt for credit_limit , max_spent_in_single_shopping

Bivariate Analysis:


spending
advance_payments
probability_of_full_payment
current_balance
credit_limit
min_payment_amt
max_spent_in_single_shopping

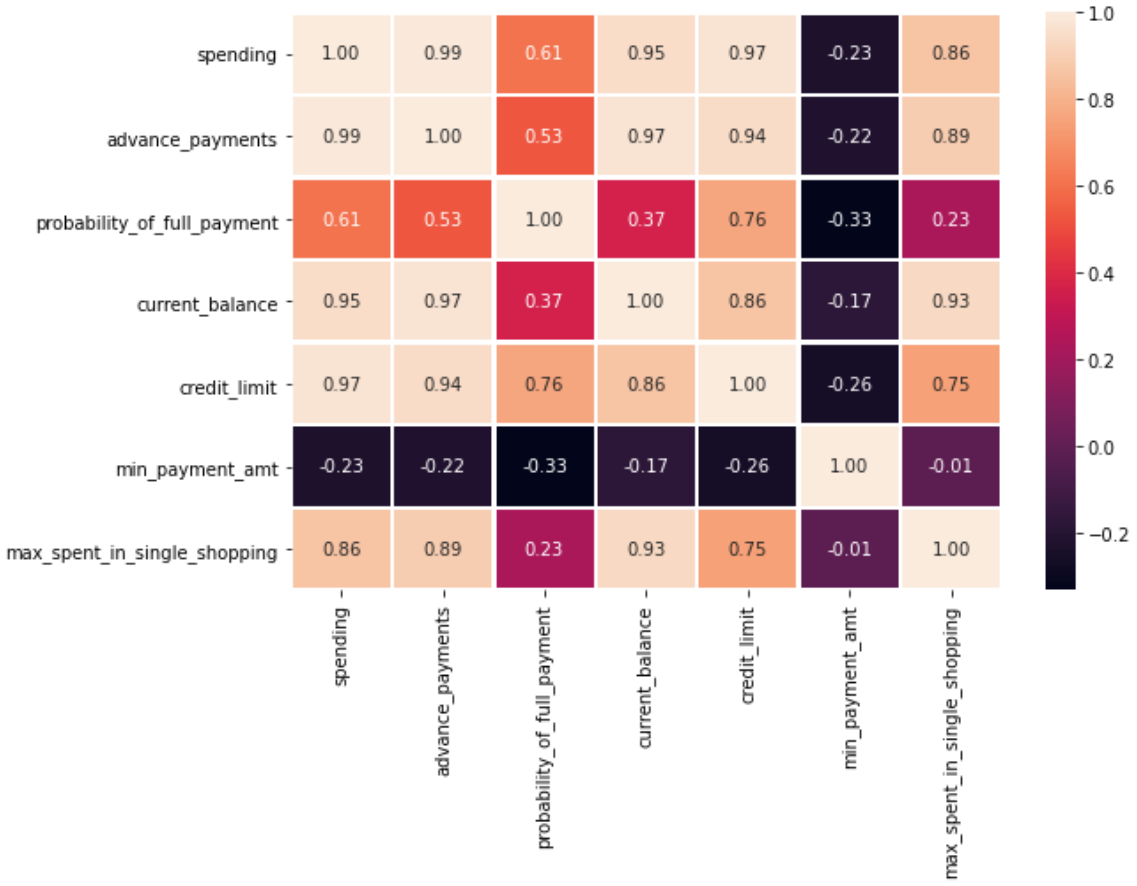


With higher credit limit, spending also increases



As the current balance increases, maximum amount spent in single shopping also is on the increase, advance payments and spending also increases

Multivariate Analysis



Good co-relation exists between spending, advance_payments, current_balance,credit_ limit

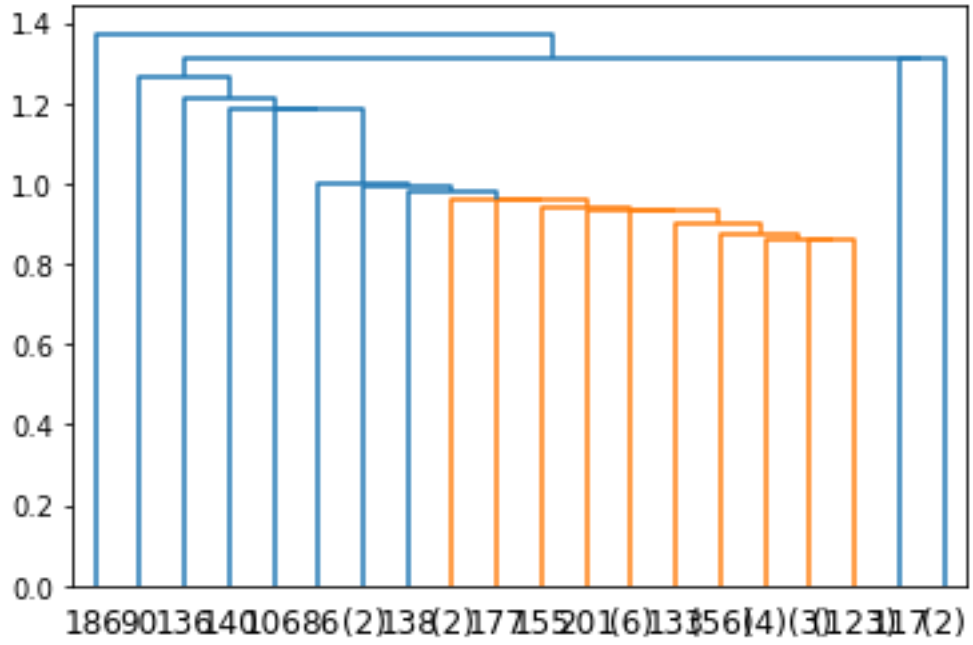
Poor/Bad co-relation exists between minimum payment amount, advance payments, probability of full payment, current balance and credit limit

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

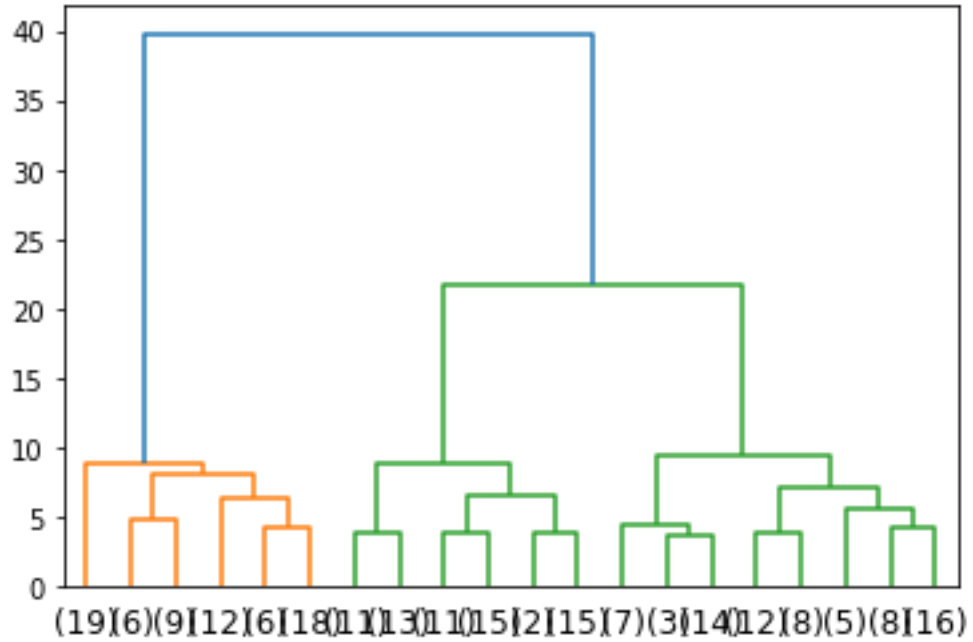
Yes, scaling is necessary for the data because all the variables of the data set are of different scales i.e. one variable (credit limit) is in 10000s and other variable (current balance, spending) is in 1000s. Since the data in these variables are of different scales, it is tough to compare these variables.
Feature scaling (also known as data normalization) is the method used to standardize the range of features of data. Since, the range of values of data is varying widely, it becomes a necessary step in data preprocessing while using machine learning algorithms.In this method, we convert variables with different scales of measurements into a single scale.
StandardScaler normalizes the data using the formula (x-mean)/standard deviation. This can also be done with the apply 'zscore'
To make analysis, it is important to bring all the variables on a same scale.
Hence Scaling the data is necessary for clustering.

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

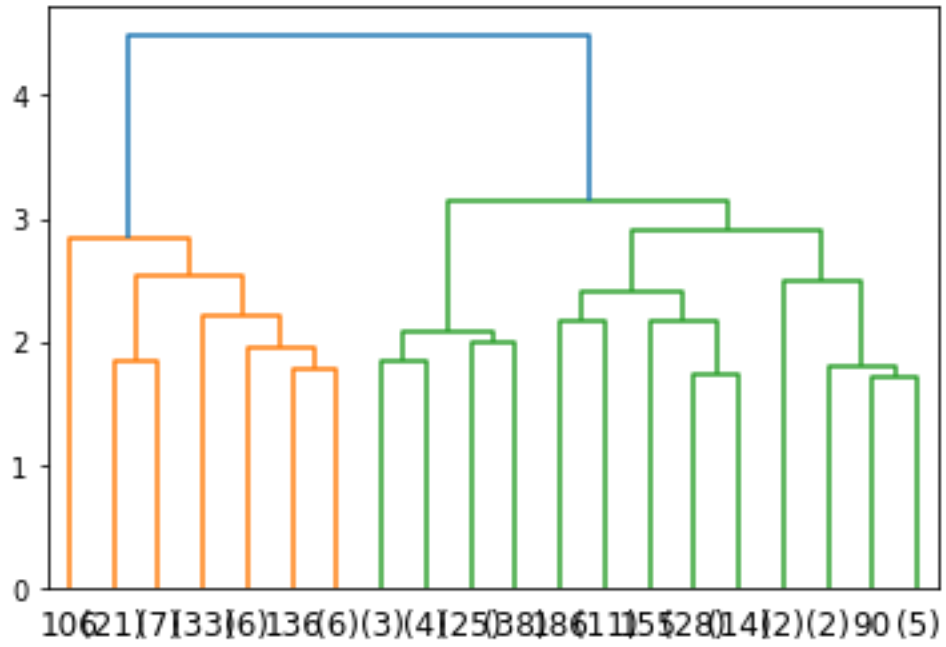
SINGLE Method:



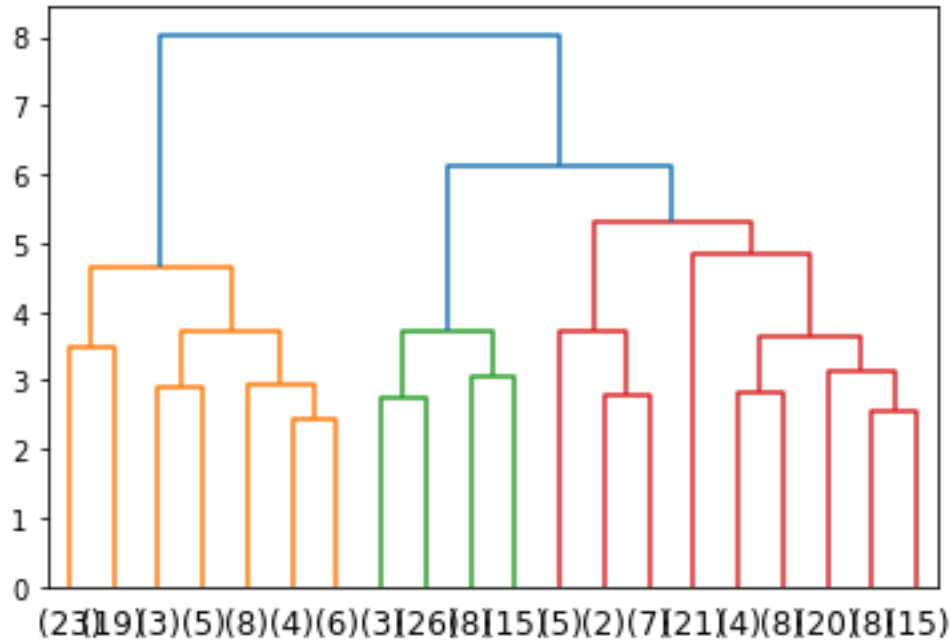
WARD Method:



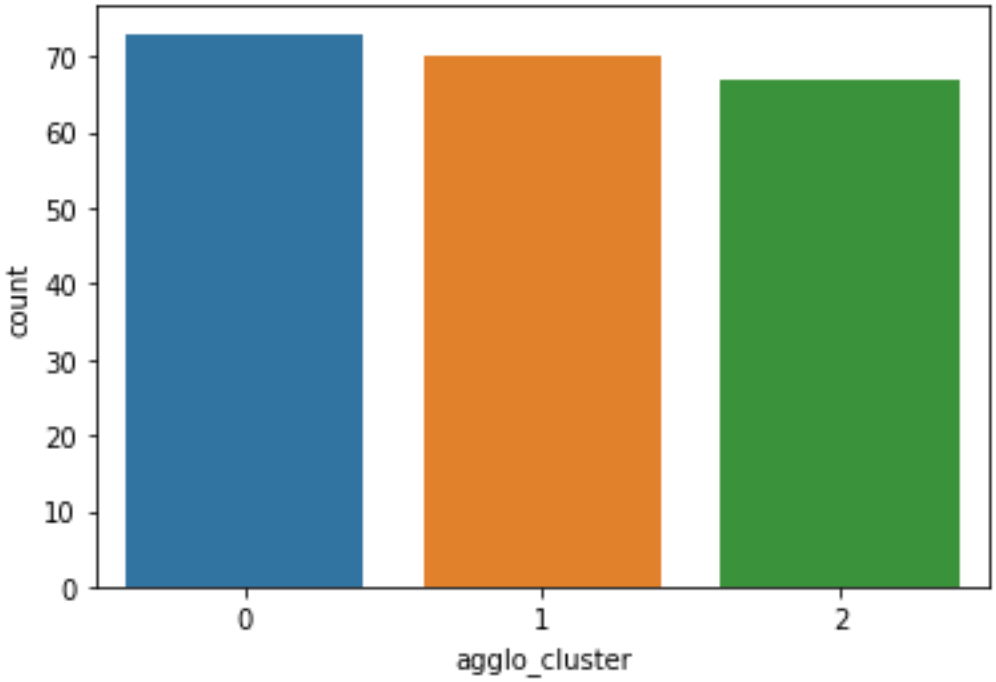
AVERAGE Method:



COMPLETE Method:



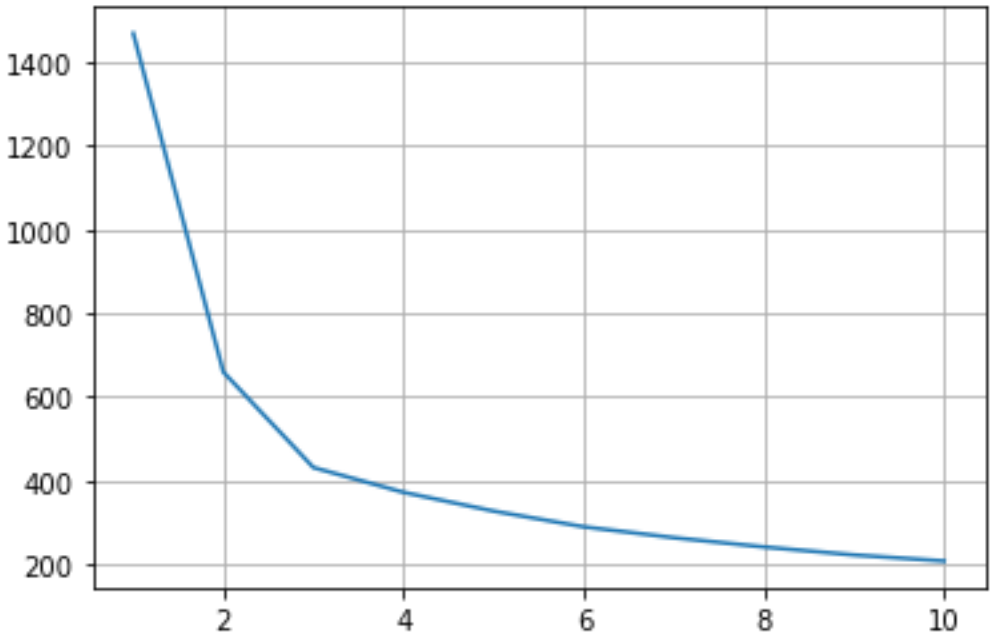
The optimum number of clusters using Dendrogram is 3. With 3 clusters the distance between the cluster units is optimum. With more than 3 clusters, the variance - distance between lines become insignificant. Hence 3 clusters are considered

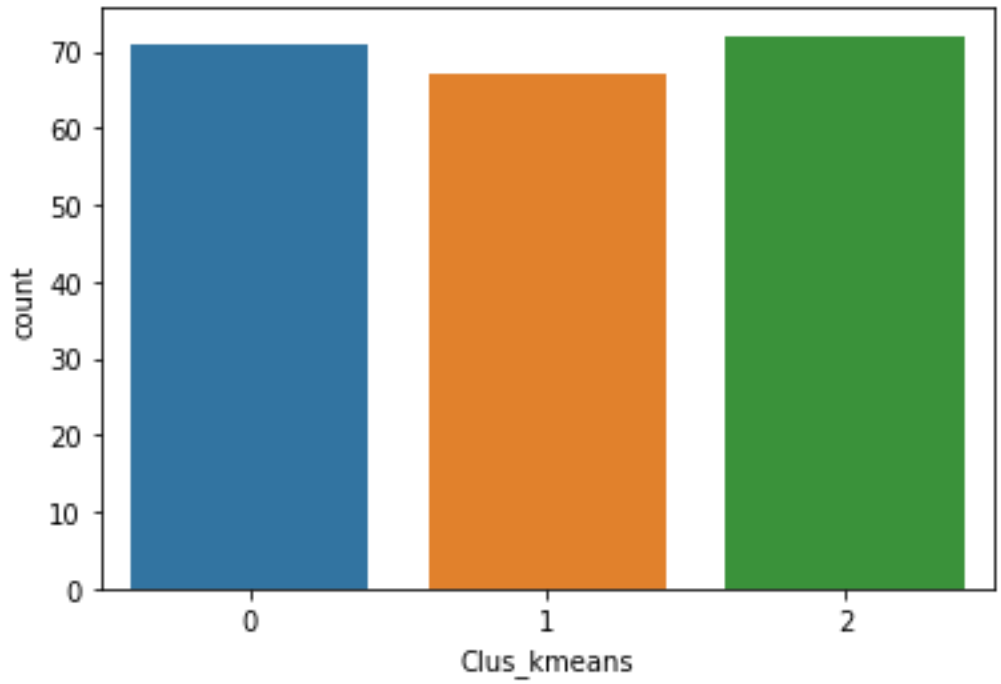


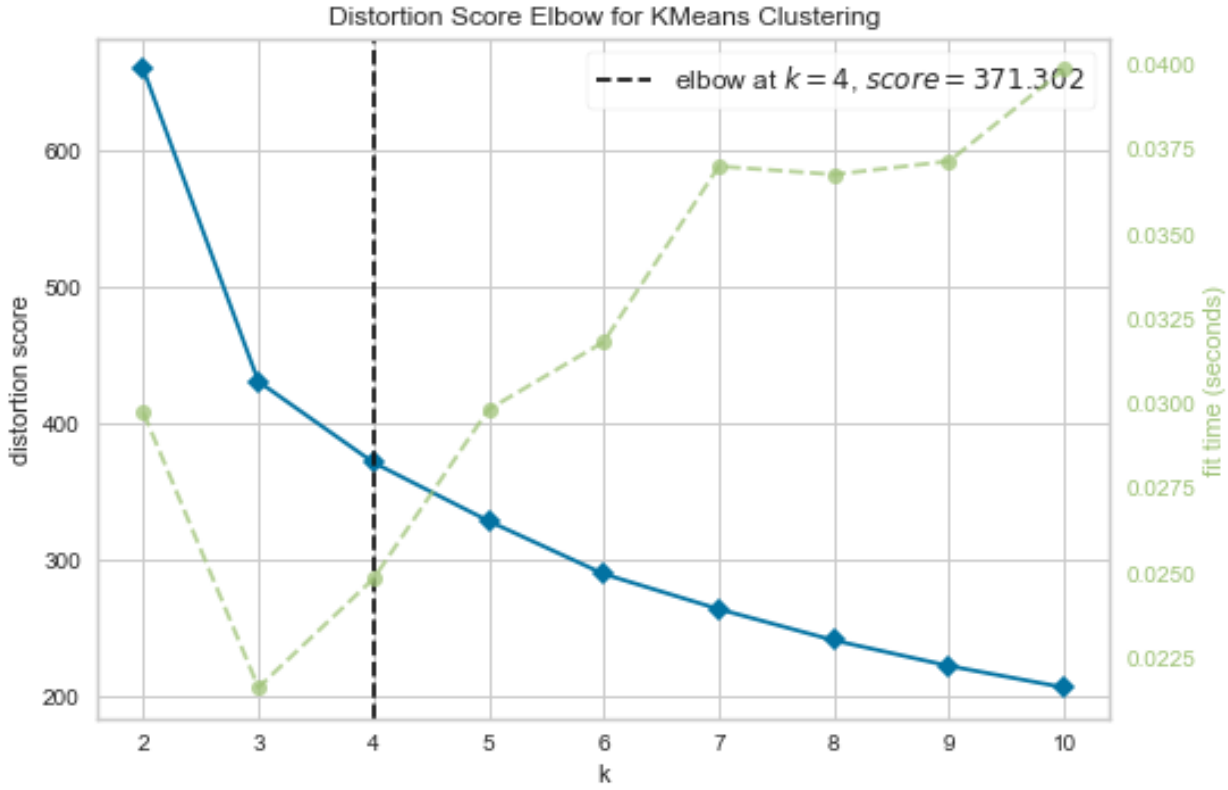
Custer segmentation is done with 3 groups

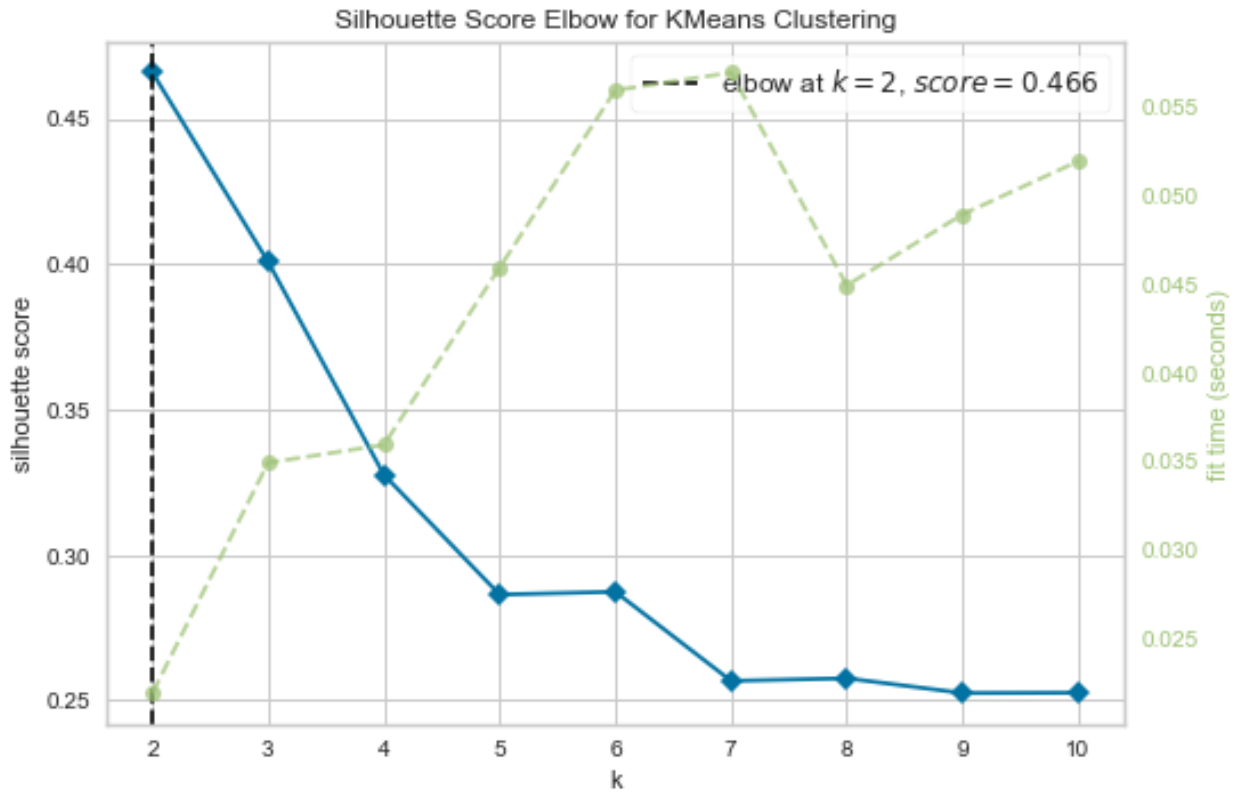
1.4 Apply K-Means clustering on scaled data and determine optimum clusters. Apply elbow curve and silhouette score. Explain the results properly. Interpret and write inferences on the finalized clusters.

Elbow curve




70
60
50
40
30
20
10
0
count
0
1
2
Clus_kmeans


Distortion Score Elbow for KMeans Clustering
elbow at $k = 4$, $score = 371.302$
600
500
400
300
200
distortion score
0.0400
0.0375
0.0350
0.0325
0.0300
0.0275
0.0250
0.0225
fit time (seconds)
2
3
4
5
6
7
8
9
10
k



Silhouette is a method to determine optimal number of clusters for given dataset. It defines as a coefficient of measure of how similar an observation to its own cluster compared to that of other clusters. The range of silhouette coefficient varies between -1 to 1. 1 value indicate that an observation is far from its neighbouring cluster and close to its own whereas -1 denotes that an observation is close to neighbouring cluster than its own cluster. The 0 value indicate the presence of observation on boundary of two clusters

Silhoutte score for 3 clusters is 0.4, which indicates that the observations in each clusters are separated from each other

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

Cluster 0: Low spending group

1. Though the credit limit of low spending group is comparatively lower than medium spending group they do spend high amount in a single shopping
2. Their probability of making full payment is less; also advance payments made by this group of customers is very less. Hence action needs to be taken to improve these factors
3. Minimum amount paid by the customer while making payments for purchases made monthly is higher and hence promotional offers an be offered to retain the stats

Cluster 1: High spending group

1. Advance amount paid by the customer in cash is the highest for this group and the probability of making full payment is also the highest. Promotional Offers to this group can focus on these aspects.
2. They maintain high current balance, interest rates on current account amount can be made benefical to this group.
3. Discount vouchers can be given to this group for spending maximum amount in a single shopping.

Cluster 2: Medium spending group

1. Appreciation in terms of cash back/reward points can be done for these people as their probability of making full payment is almost equal with respect to the high spend group.
2. They maintain good current balance and their credit limit is equally good
3. Minimum paid by the customer while making payments for purchases made monthly is the lowest for this group and hence appropriate action to improve this factor can be considered

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

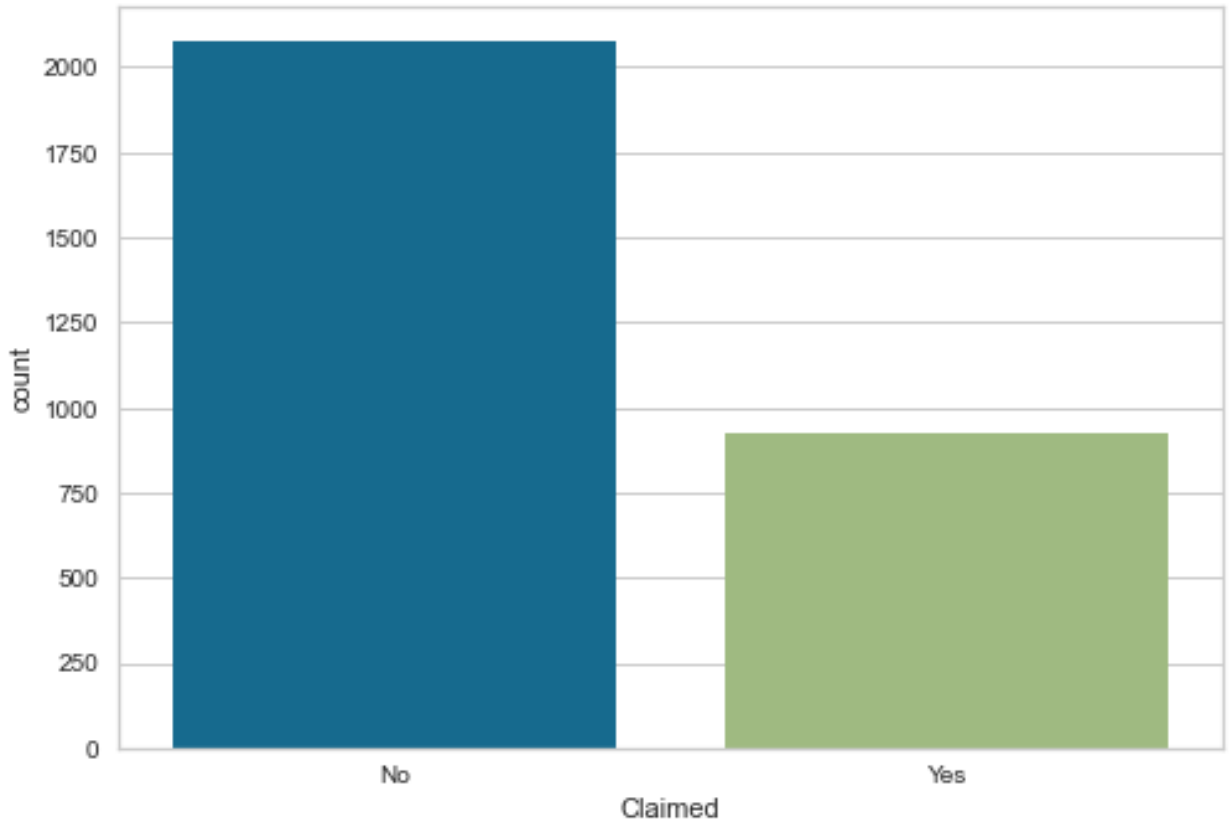
Claimed is the target variable while all others are the predictors.

Out of the 10 columns, 6 are object type, 2 are float type, while remaining 2 are int. Object - Agency_Code, Type,Claimed, Channel, Product Name and Destination Float - Commision, Sales Int - Age, Duration

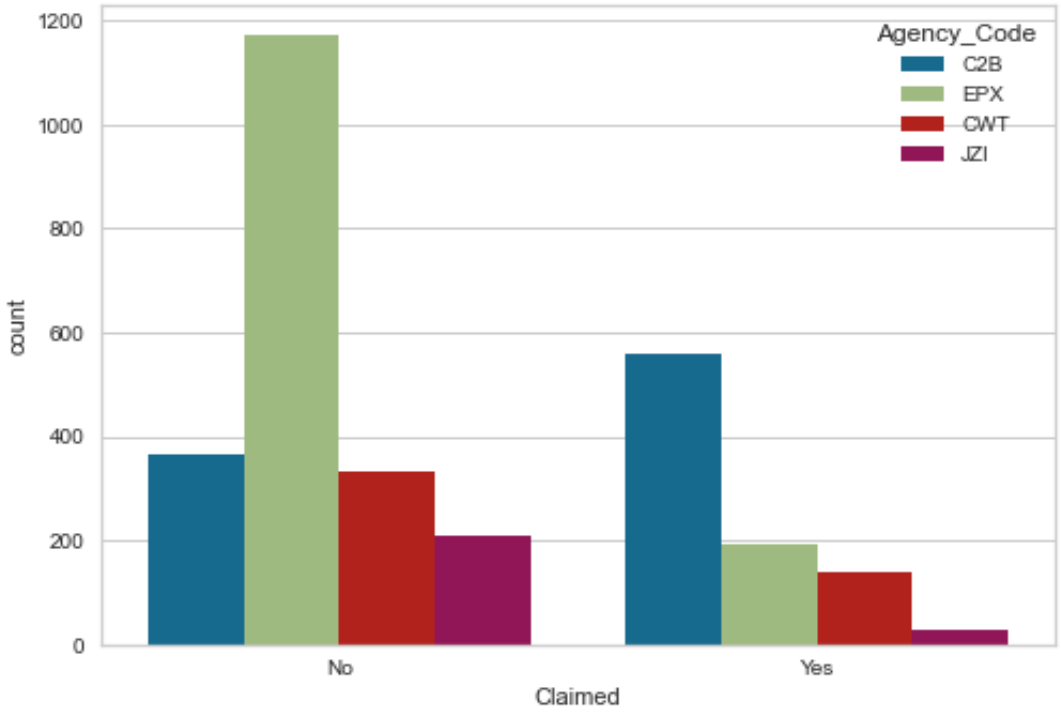
Many columns are of type object i.e. strings. These need to be converted to ordinal type

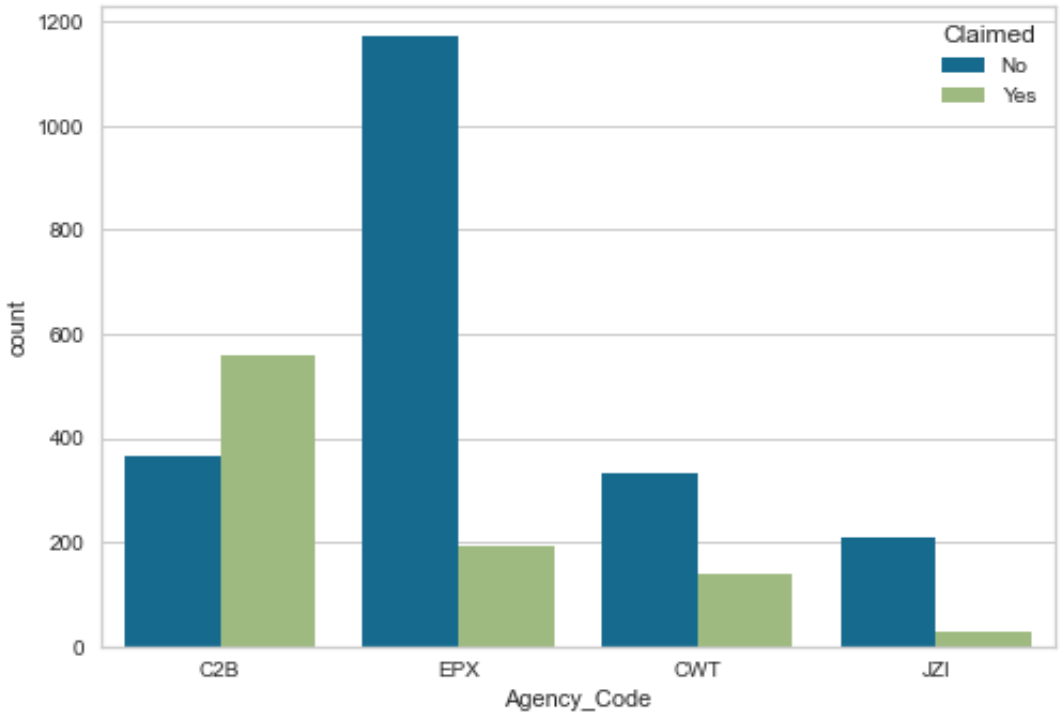
It appears there are also no missing values.

Univariate Analysis:



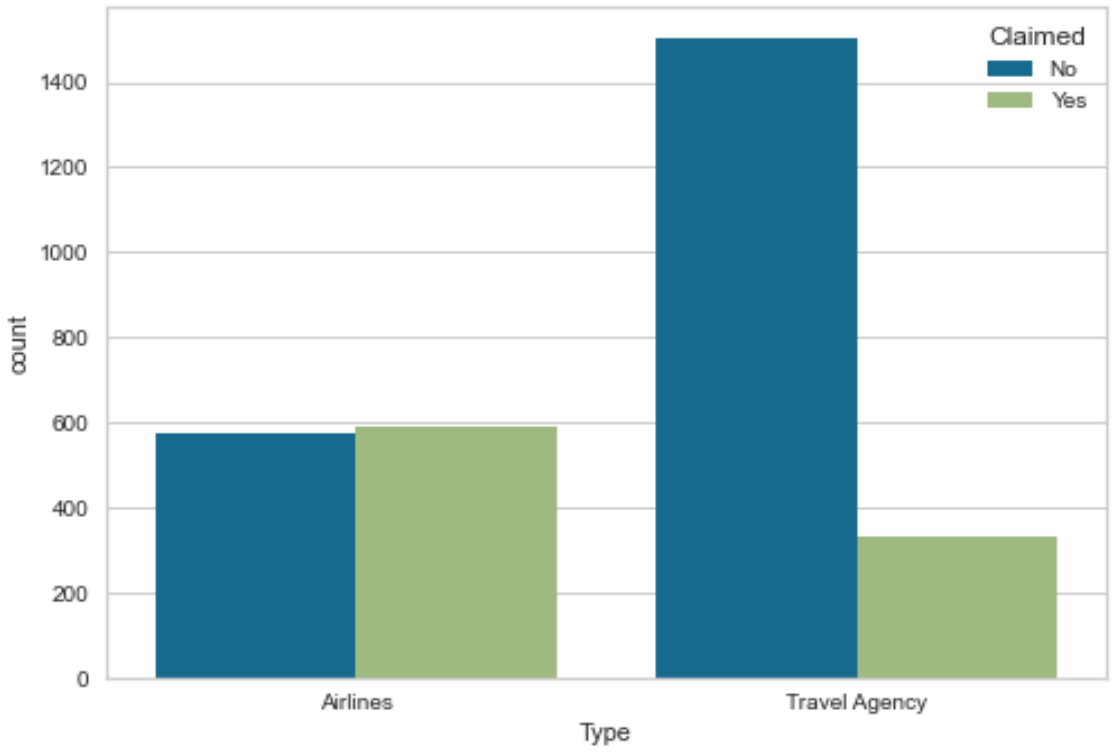
There are 30.8 % Claims

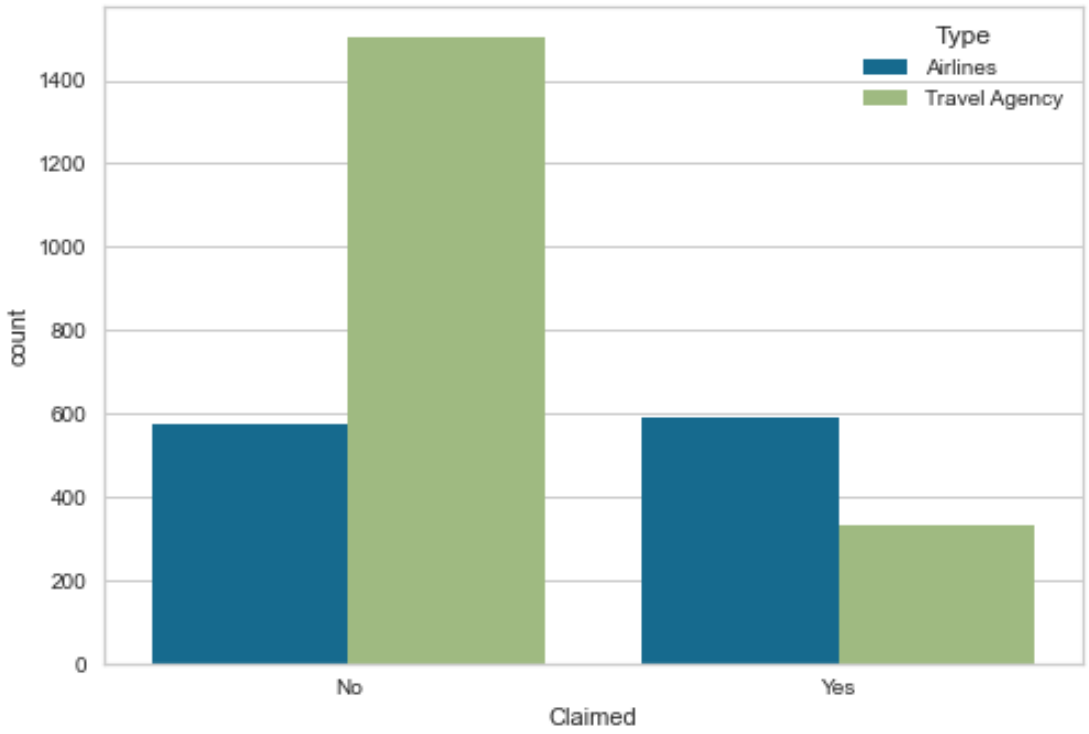




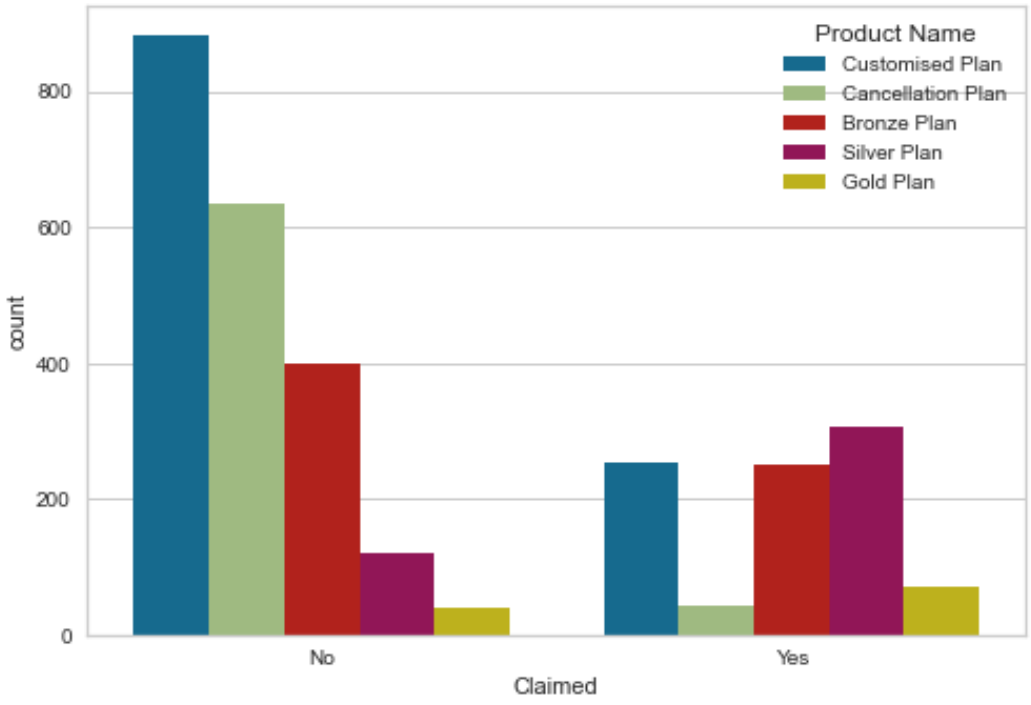
Tour firm 'C2B' has maximum number of claims though it has registered lesser number of insurance registrations. Outrightly, claims from 'C2B' has some issues and it needs to be investigated

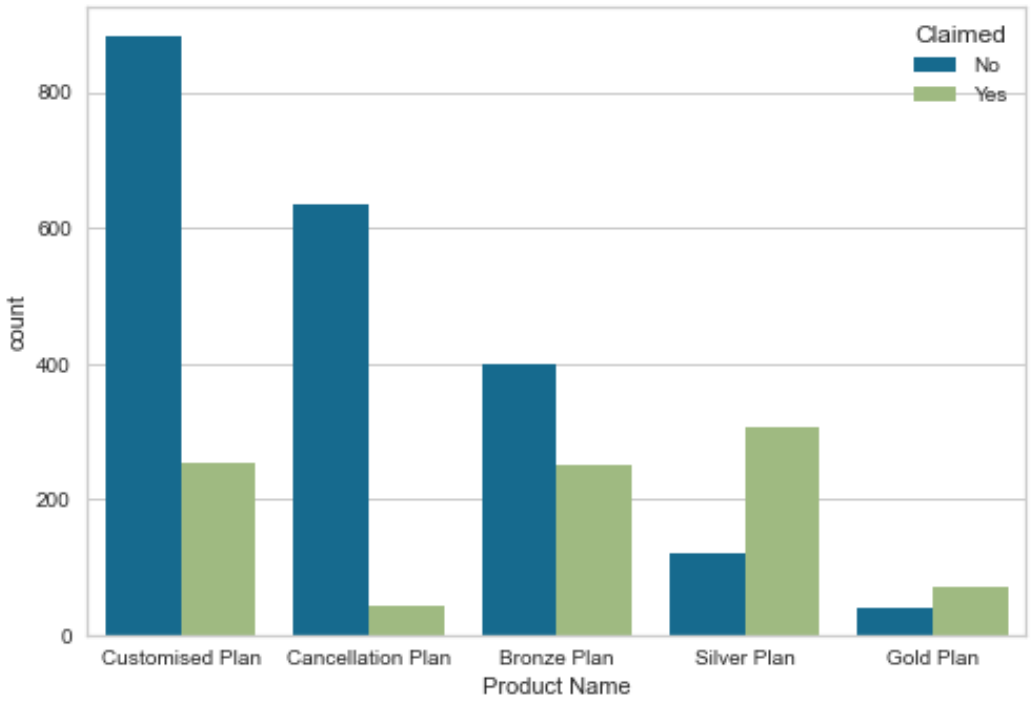
Tour firm 'EPX' has registered minimum number of claims



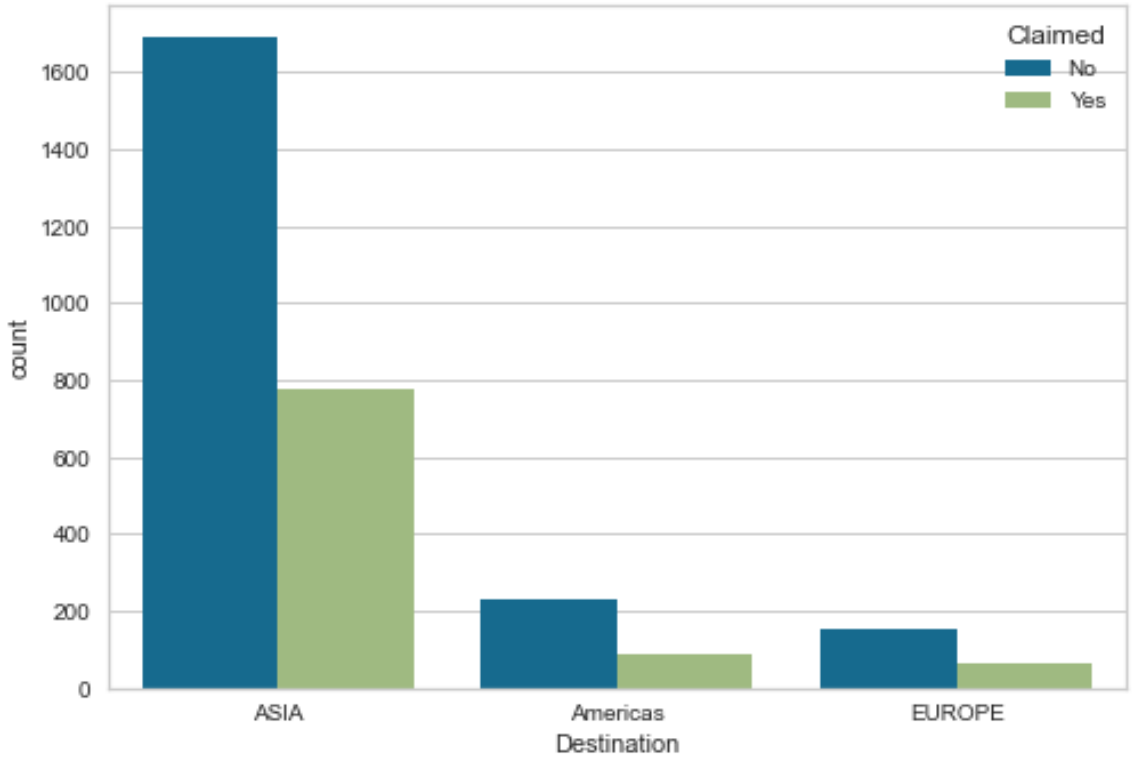


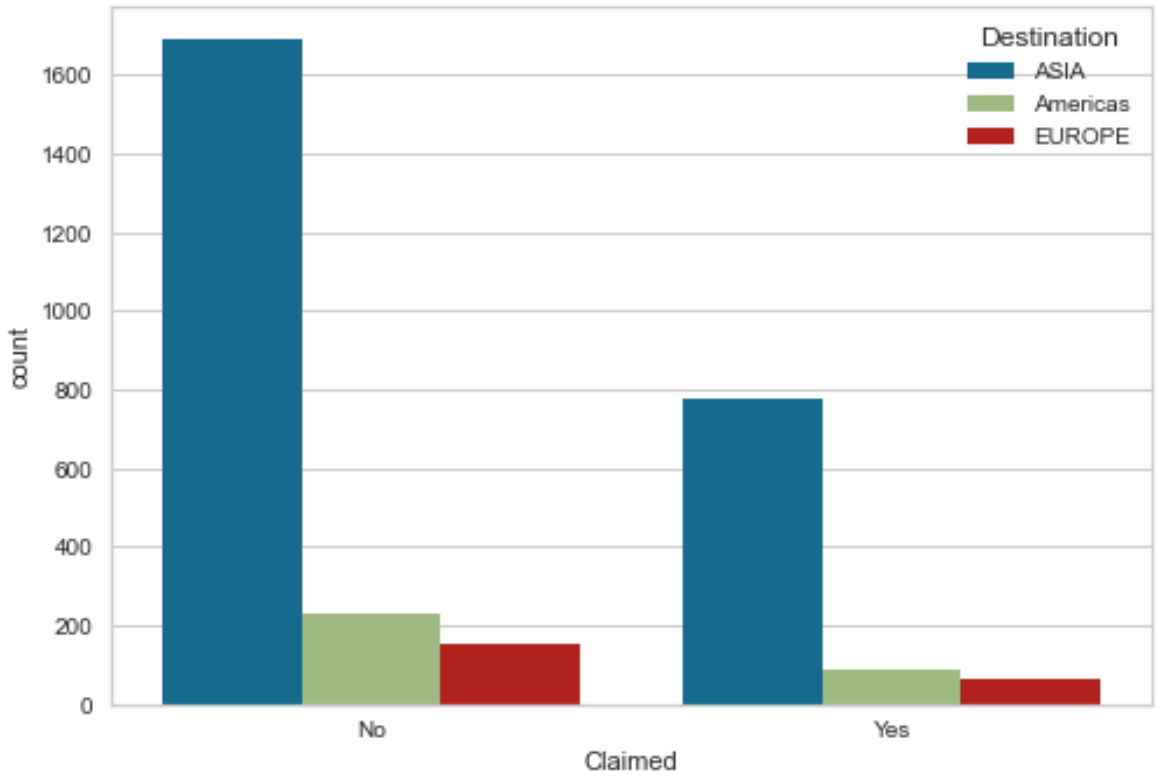
Airlines type have maximum % of claims than the travel agency.



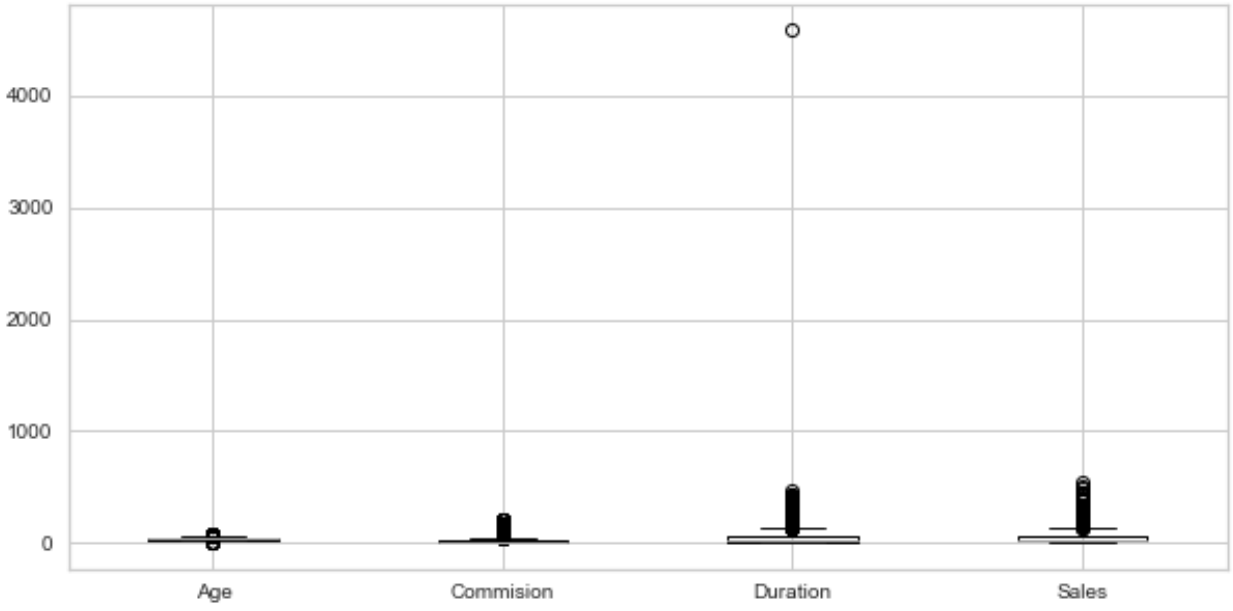


- Silver Plan customers have more number of claims and Cancellation Plan customers have the minimum number of claims
- There are more number of customers in Customised Plan and less number of customers in Gold Plan

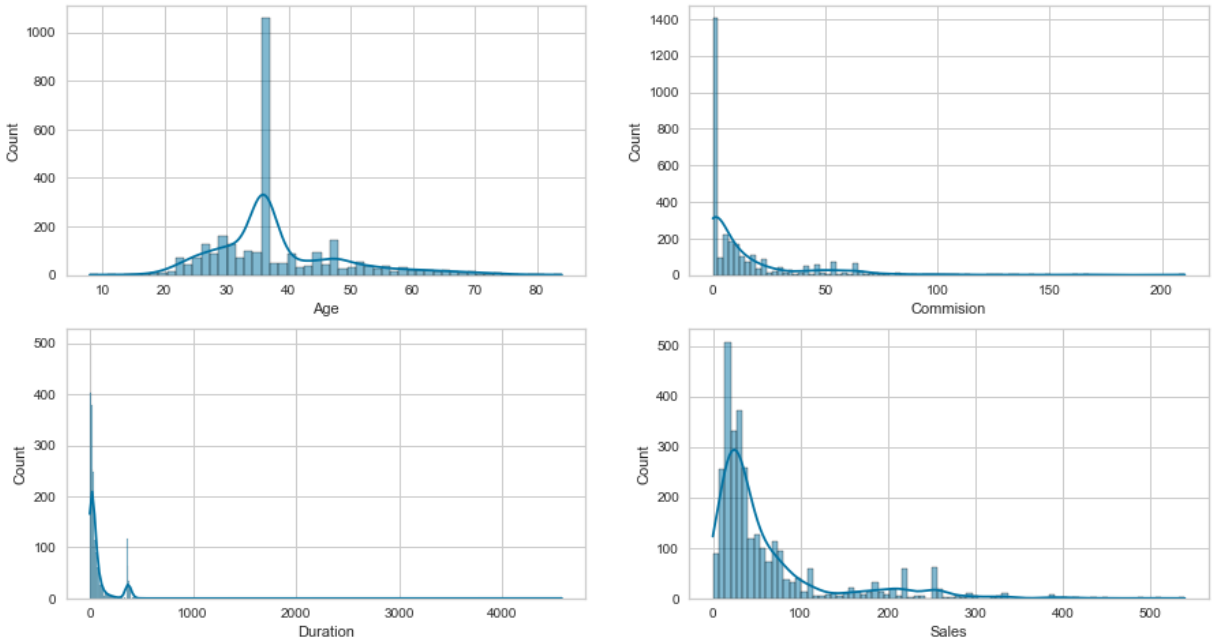




Tour Destination 'Asia' has the maximum number of Claims and tour destination 'Europe' has minimum number of claims

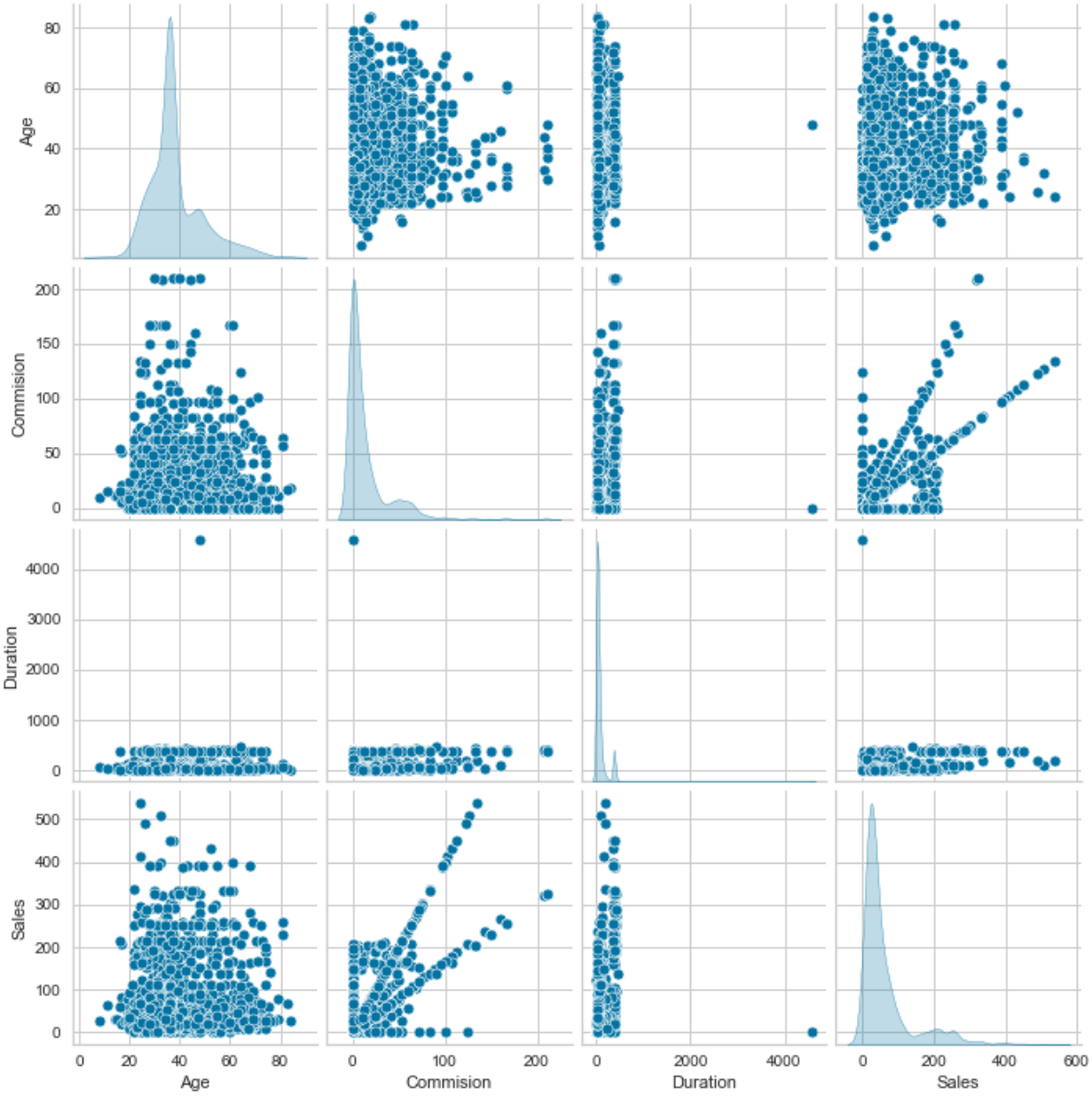


All the variables have outliers present



Age is normally distributed Commission, Duration and Sales are left skewed

Bivariate Analysis



There is no relation between all of the variables, and all are independent There is a slight correlation between sales and commission.

Multivariate Analysis

There is correlation between commision and sales variables. Over all the magnitude of correlations between the variables are very less.

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

Data is split as 70% for Training and 30% for Test with random state as 1

Building a Decision Tree Classifier

```
param_grid = {
    'criterion': ['gini'],
    'max_depth': [10,20,30,50],
    'min_samples_leaf': [50,100,150],
    'min_samples_split': [150,300,450],
}

dtcl = DecisionTreeClassifier(random_state=1)

grid_search = GridSearchCV(estimator = dtcl, param_grid = param_grid, cv = 10)
```

```
grid_search.fit(X_train, train_labels)
print(grid_search.best_params_)
best_grid = grid_search.best_estimator_
best_grid
```

```
{'criterion': 'gini', 'max_depth': 10, 'min_samples_leaf': 50, 'min_samples_split': 150}

DecisionTreeClassifier(max_depth=10, min_samples_leaf=50, min_samples_split=150,
                       random_state=1)
```

Building a Random Forest Classifier

```
param_grid = {
    'max_depth': [10],## 20,30,40
    'max_features': [6],## 7,8,9
    'min_samples_leaf': [10],## 50,100
    'min_samples_split': [50], ## 60,70
    'n_estimators': [300] ## 100,200
}

rfcl = RandomForestClassifier(random_state=1)

grid_search = GridSearchCV(estimator = rfcl, param_grid = param_grid, cv = 5)
```

```
grid_search.fit(X_train, train_labels)
```

```
GridSearchCV(cv=5, estimator=RandomForestClassifier(random_state=1),
             param_grid={'max_depth': [10], 'max_features': [6],
                         'min_samples_leaf': [10], 'min_samples_split': [50],
                         'n_estimators': [300]})
```

```
grid_search.best_params_
```

```
{'max_depth': 10,
 'max_features': 6,
 'min_samples_leaf': 10,
 'min_samples_split': 50,
 'n_estimators': 300}
```

```
best_grid = grid_search.best_estimator_
```

```
best_grid
```

```
RandomForestClassifier(max_depth=10, max_features=6, min_samples_leaf=10,
                       min_samples_split=50, n_estimators=300, random_state=1)
```

Building a Neural Network Classifier

```
param_grid = {
    'hidden_layer_sizes': [100], # 50, 200
    'max_iter': [2500], #5000,2500
    'solver': ['adam'], #sgd
    'tol': [0.01],
}

nncl = MLPClassifier(random_state=1)

grid_search = GridSearchCV(estimator = nncl, param_grid = param_grid, cv = 10)
```

```
grid_search.fit(X_train, train_labels)
grid_search.best_params_
#{'hidden_layer_sizes': 100, 'max_iter': 2500, 'solver': 'adam', 'tol': 0.01}
```

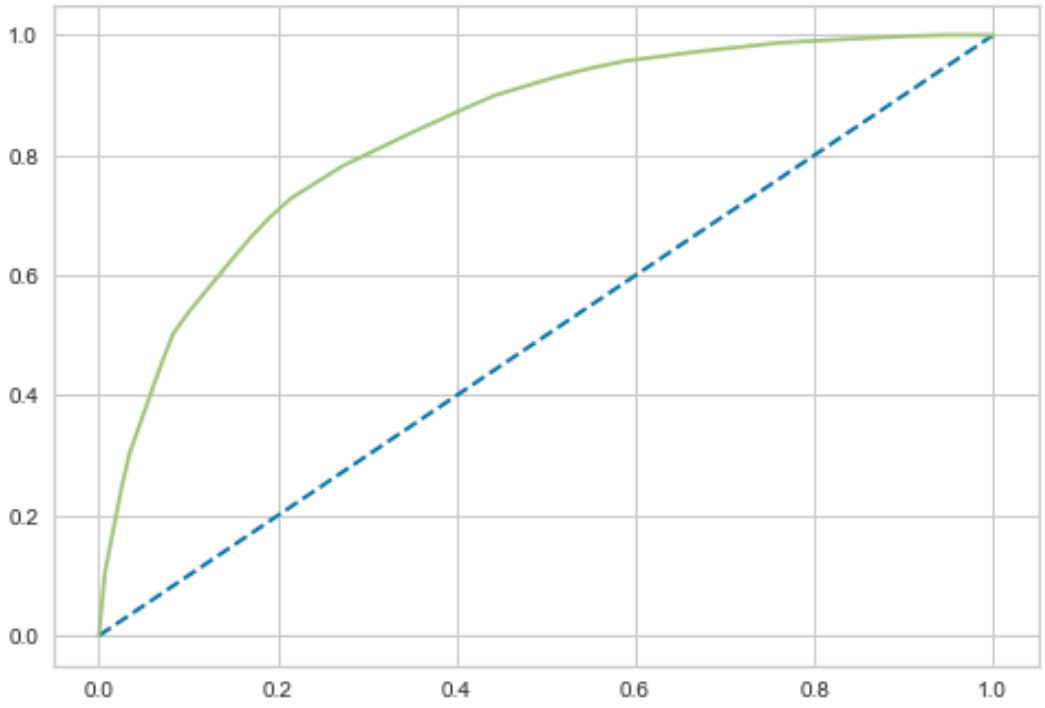
```
{'hidden_layer_sizes': 100, 'max_iter': 2500, 'solver': 'adam', 'tol': 0.01}
```

```
best_grid = grid_search.best_estimator_
best_grid
```

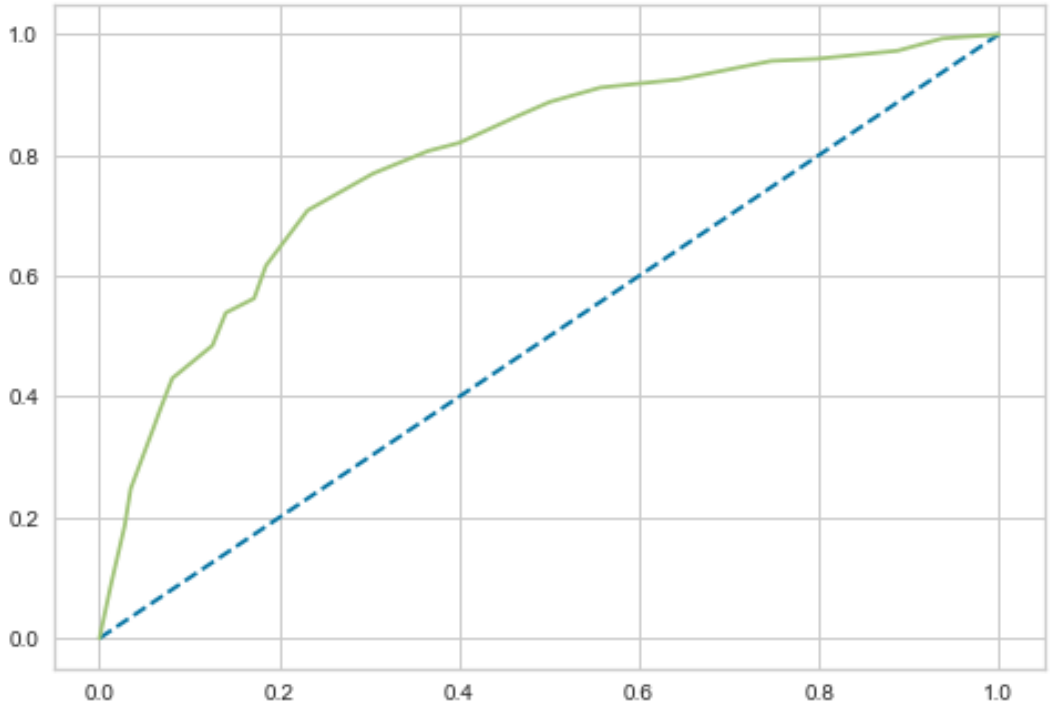
```
MLPClassifier(hidden_layer_sizes=100, max_iter=2500, random_state=1, tol=0.01)
```

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

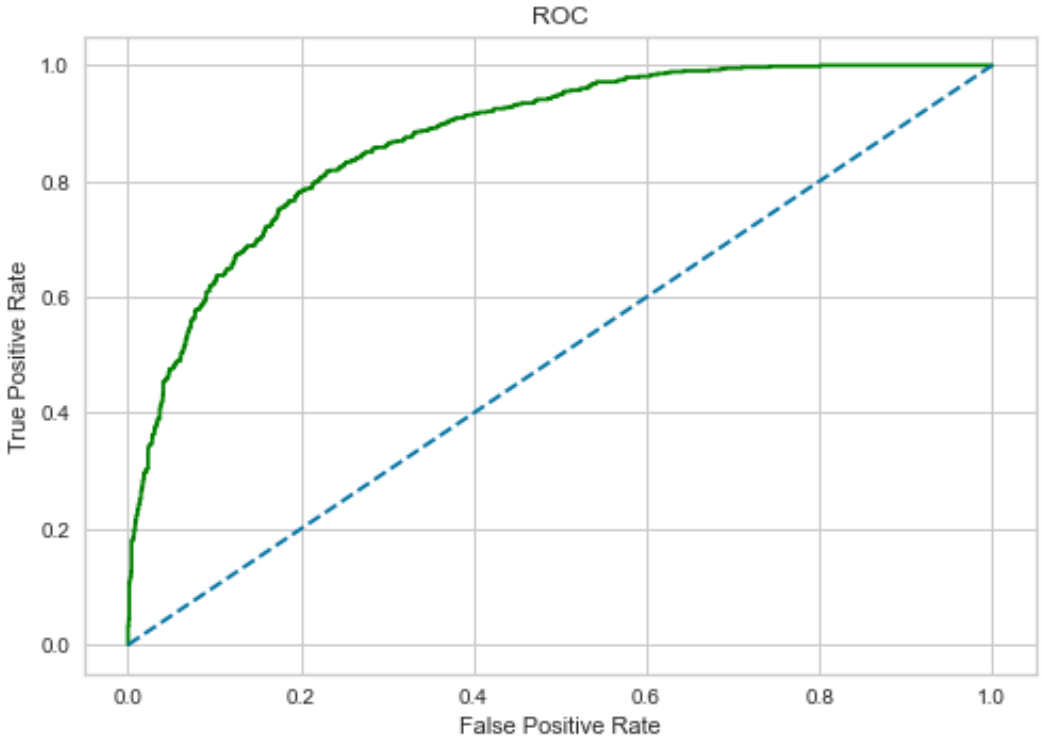
AUC and ROC for the training data – CART Model



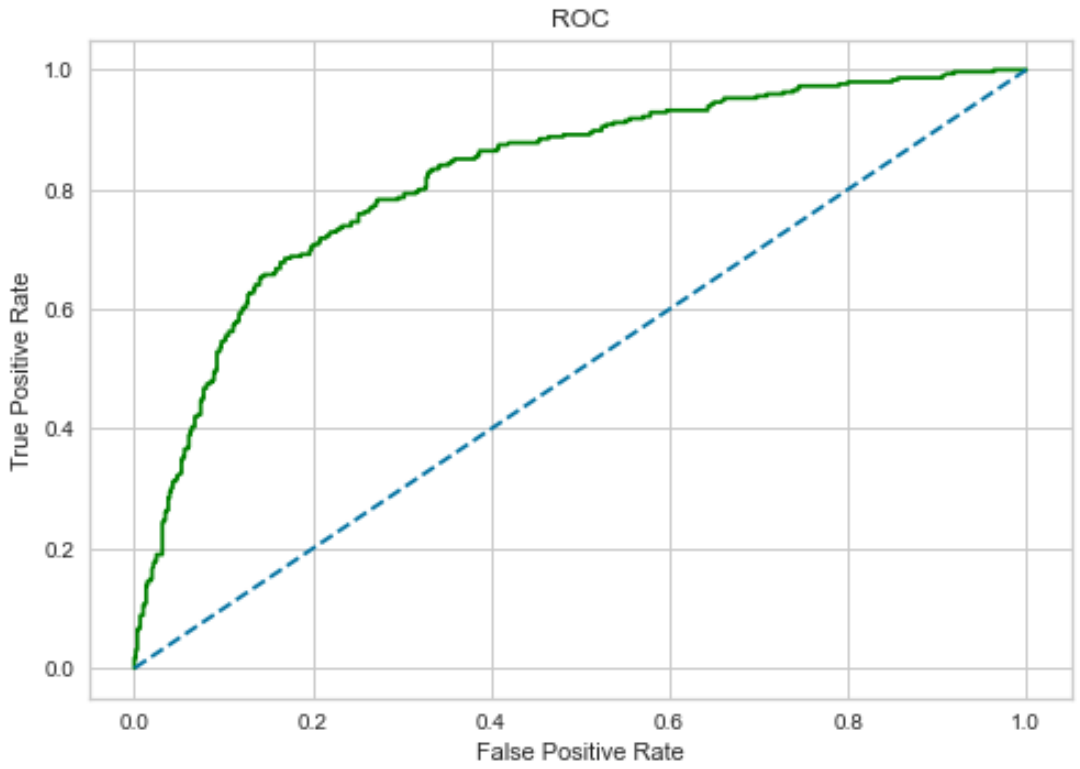
AUC and ROC for the test data – CART model



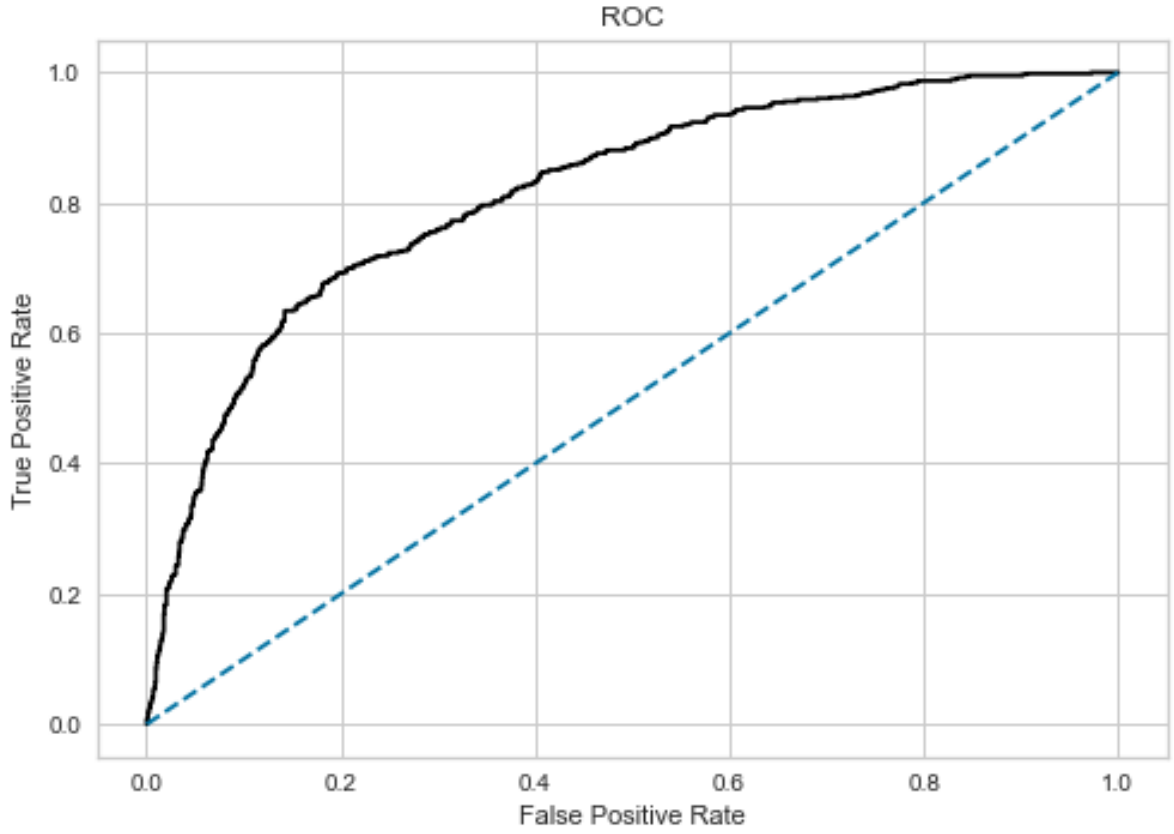
RF Model Performance Evaluation on Training data – Random Forest



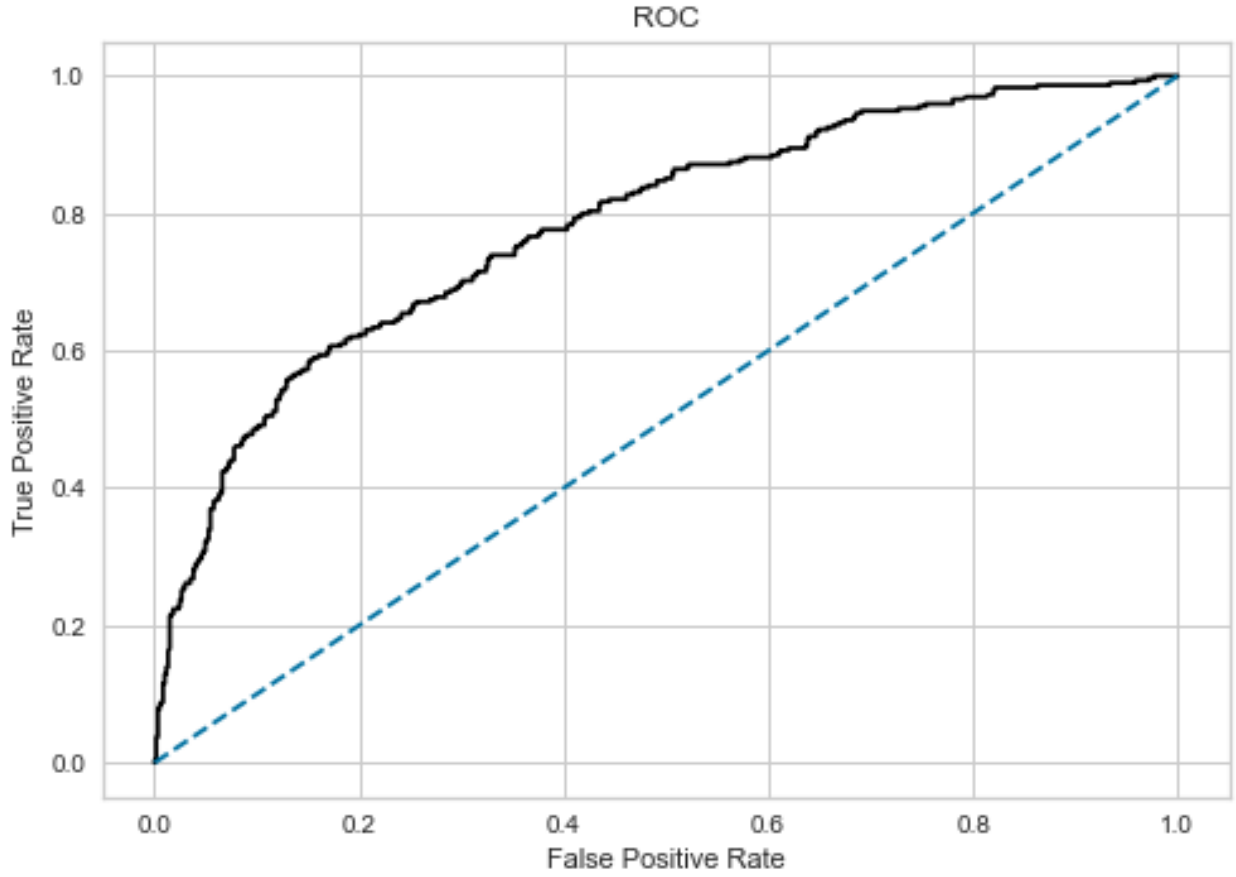
RF Model Performance Evaluation on Test data – Random Forest



NN Model Performance Evaluation on Train data



NN Model Performance Evaluation on Test data



CART Conclusion:

Train Data:

AUC: 83.6%

Accuracy: 79%

Precision: 72%

Recall:50%

f1-Score: 59%

Test Data:

AUC: 79.4%

Accuracy: 75%

Precision: 73%

Recall:38%

f1-Score: 50%

RANDOM FOREST Conclusion:

Train Data:

AUC: 87.4%

Accuracy: 82%

Precision: 74%

Recall:61%

f1-Score: 67%

Test Data:

AUC: 82.2%

Accuracy: 77%

Precision: 73%

Recall:47%

f1-Score: 58%

NEURAL NETWORK Conclusion:

Train Data:
AUC: 87.4%
Accuracy: 79%
Precision: 67%
Recall:58%
f1-Score: 62%

Test Data:
AUC: 78.27%
Accuracy: 76%
Precision: 71%
Recall:48%
f1-Score: 57%

CART MODEL:

Confusion Matrix for the training data

```
confusion_matrix(train_labels, ytrain_predict)
```

```
array([[1349,  122],
       [ 313,  316]], dtype=int64)
```

```
#Train Data Accuracy
cart_train_acc=best_grid.score(X_train,train_labels)
cart_train_acc
```

```
0.7928571428571428
```

```
print(classification_report(train_labels, ytrain_predict))
```

```
              precision    recall  f1-score   support

           0       0.81      0.92      0.86      1471
           1       0.72      0.50      0.59       629

    accuracy                           0.79      2100
   macro avg       0.77      0.71      0.73      2100
weighted avg       0.78      0.79      0.78      2100
```

Confusion Matrix for test data

```
confusion_matrix(test_labels, ytest_predict)
```

```
array([[564,  41],
       [183, 112]], dtype=int64)
```

```
#Test Data Accuracy
cart_test_acc=best_grid.score(X_test,test_labels)
cart_test_acc
```

```
0.7511111111111111
```

```
print(classification_report(test_labels, ytest_predict))
```

```
              precision    recall  f1-score   support

           0       0.76      0.93      0.83       605
           1       0.73      0.38      0.50       295

    accuracy                           0.75       900
   macro avg       0.74      0.66      0.67       900
weighted avg       0.75      0.75      0.72       900
```

RANDOM FOREST MODEL:

RF Model Performance Evaluation on Training data

```
confusion_matrix(train_labels,ytrain_predict)
```

```
array([[1335,  136],
       [ 246,  383]], dtype=int64)
```

```
rf_train_acc=best_grid.score(X_train,train_labels)
rf_train_acc
```

```
0.8180952380952381
```

```
print(classification_report(train_labels,ytrain_predict))
```

```
              precision    recall  f1-score   support

           0       0.84      0.91      0.87      1471
           1       0.74      0.61      0.67       629

    accuracy                           0.82      2100
   macro avg       0.79      0.76      0.77      2100
weighted avg       0.81      0.82      0.81      2100
```

RF Model Performance Evaluation on Test data

```
confusion_matrix(test_labels,ytest_predict)
```

```
array([[554,  51],
       [155, 140]], dtype=int64)
```

```
rf_test_acc=best_grid.score(X_test,test_labels)
rf_test_acc
```

```
0.7711111111111111
```

```
print(classification_report(test_labels,ytest_predict))
```

```
              precision    recall  f1-score   support

           0       0.78      0.92      0.84       605
           1       0.73      0.47      0.58       295

    accuracy                           0.77       900
   macro avg       0.76      0.70      0.71       900
weighted avg       0.77      0.77      0.76       900
```

NEURAL NETWORK MODEL:

NN Model Performance Evaluation on Training data

```
confusion_matrix(train_labels,ytrain_predict)
```

```
array([[1289,  182],
       [ 262,  367]], dtype=int64)
```

```
nn_train_acc=best_grid.score(X_train,train_labels)
nn_train_acc
```

```
0.7885714285714286
```

```
print(classification_report(train_labels,ytrain_predict))
```

```
              precision    recall  f1-score   support

           0       0.83      0.88      0.85      1471
           1       0.67      0.58      0.62       629

    accuracy                           0.79      2100
   macro avg       0.75      0.73      0.74      2100
weighted avg       0.78      0.79      0.78      2100
```

NN Model Performance Evaluation on Test data

```
confusion_matrix(test_labels,ytest_predict)
```

```
array([[547,  58],
       [154, 141]], dtype=int64)
```

```
nn_test_acc=best_grid.score(X_test,test_labels)
nn_test_acc
```

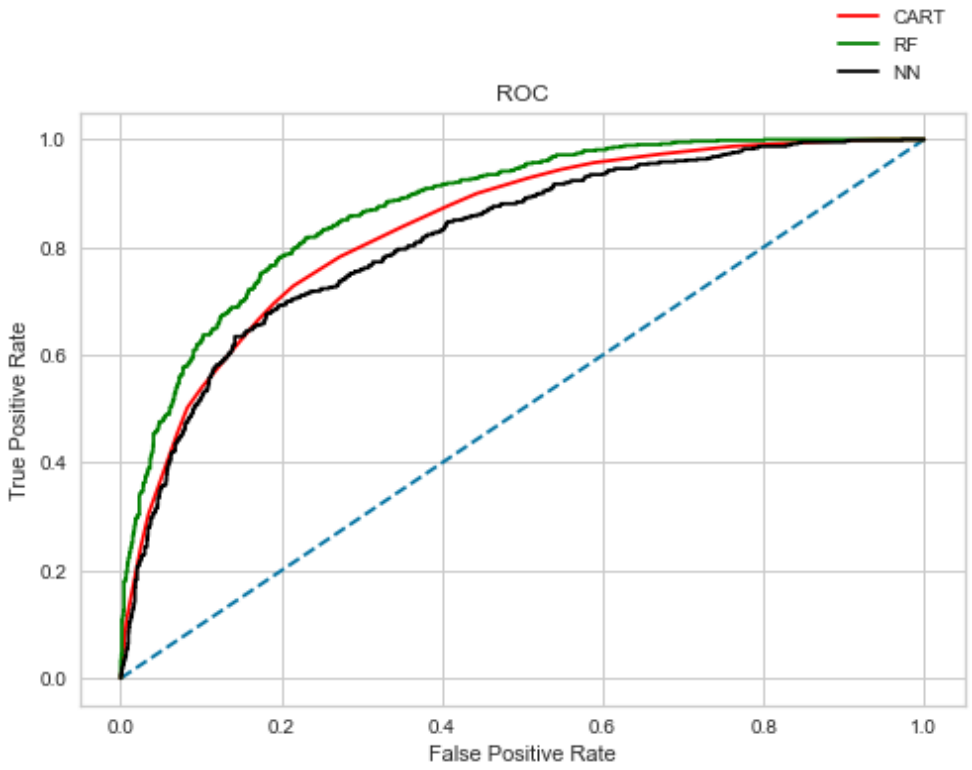
```
0.7644444444444445
```

```
print(classification_report(test_labels,ytest_predict))
```

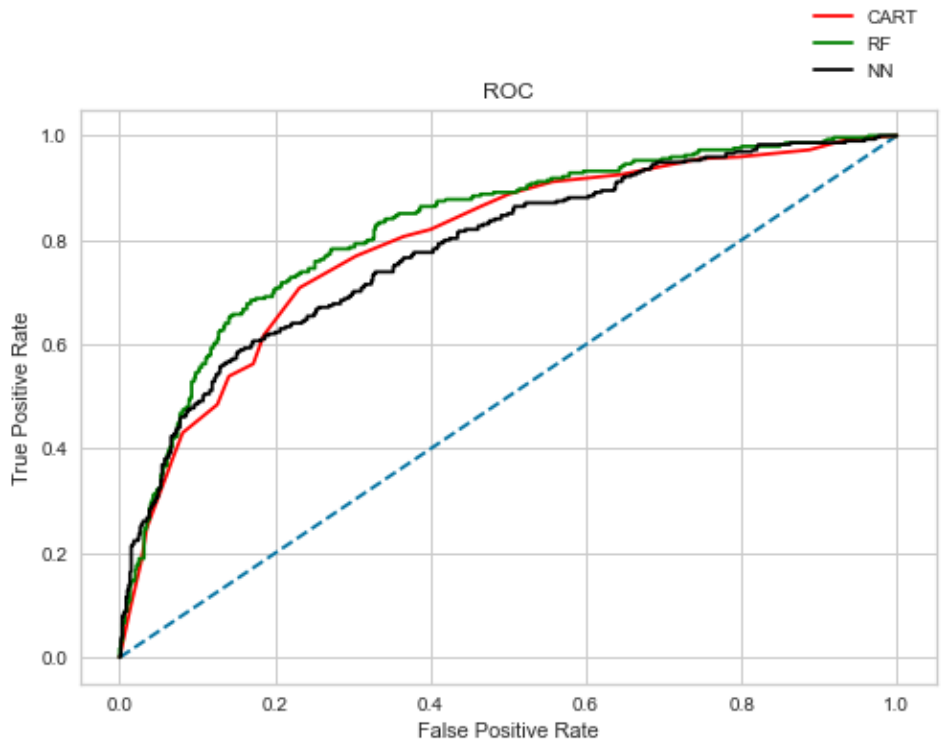
```
              precision    recall  f1-score   support

           0       0.78      0.90      0.84       605
           1       0.71      0.48      0.57       295

    accuracy                           0.76       900
   macro avg       0.74      0.69      0.70       900
weighted avg       0.76      0.76      0.75       900
```

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

ROC Curve for the 3 models on the Training data



ROC Curve for the 3 models on the TEST data



]

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

1. Out of the 3 models, Random Forest has slightly better performance than the Cart and Neural network model

2. best_grid model has improved performance and saves processing time especially in such large datasets.

3. Accuracy, AUC, Precision and Recall for test data is almost inline with training data. This proves no overfitting or underfitting has happened, and overall the model is a good model for classification

4. Agency code, Sales and Product Name (in same order of preference) are the most important variables in determining if a Claim Status is Yes or No

5. The Overall model performance is moderate enough to start predicting if any new customer claim will be Yes or No