Data Mining- Project Report

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 and do exploratory data analysis. Describe the data briefly.**

Head of the data

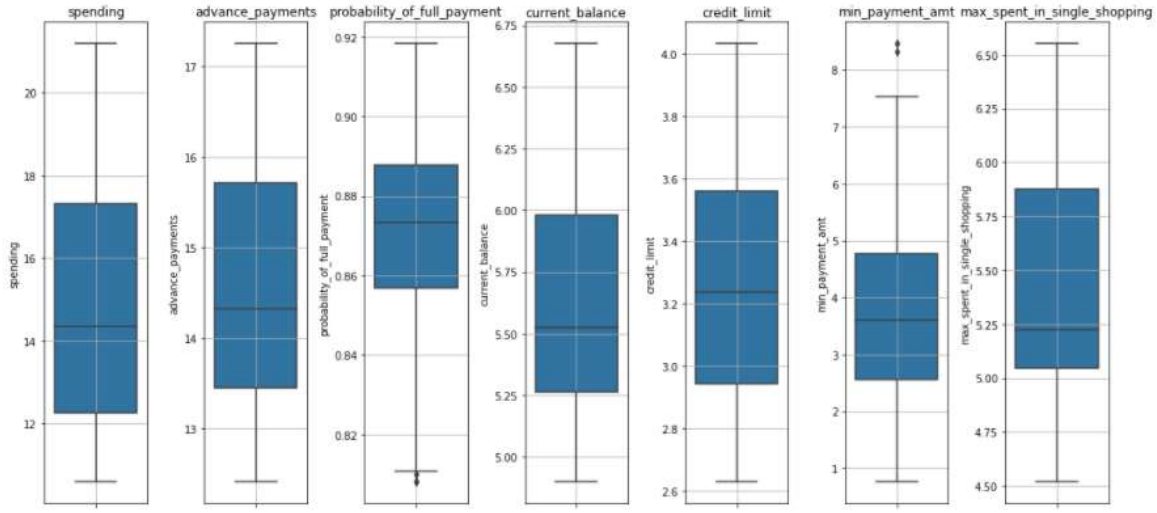
	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
0	19.94	16.92	0.8752	6.675	3.763	3.252	6.550
1	15.99	14.89	0.9064	5.363	3.582	3.336	5.144
2	18.95	16.42	0.8829	6.248	3.755	3.368	6.148
3	10.83	12.96	0.8099	5.278	2.641	5.182	5.185
4	17.99	15.86	0.8992	5.890	3.694	2.068	5.837

```
The shape of the dataset is (210, 7)
```

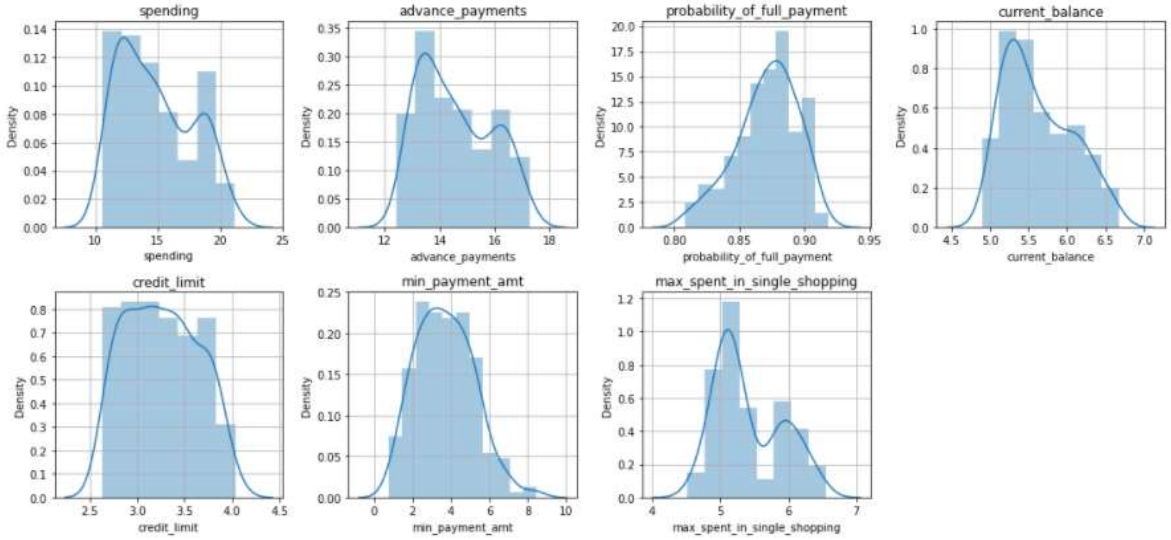
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 210 entries, 0 to 209
Data columns (total 7 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   spending                      210 non-null    float64
 1   advance_payments              210 non-null    float64
 2   probability_of_full_payment   210 non-null    float64
 3   current_balance               210 non-null    float64
 4   credit_limit                  210 non-null    float64
 5   min_payment_amt               210 non-null    float64
 6   max_spent_in_single_shopping  210 non-null    float64
dtypes: float64(7)
memory usage: 11.6 KB
```

EDA

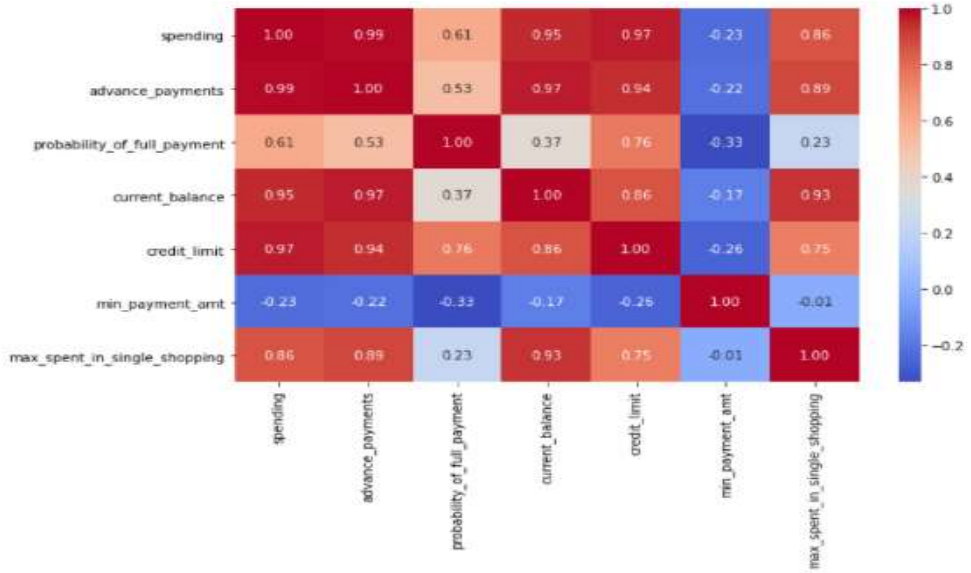
Finding the outliers in the Dataset



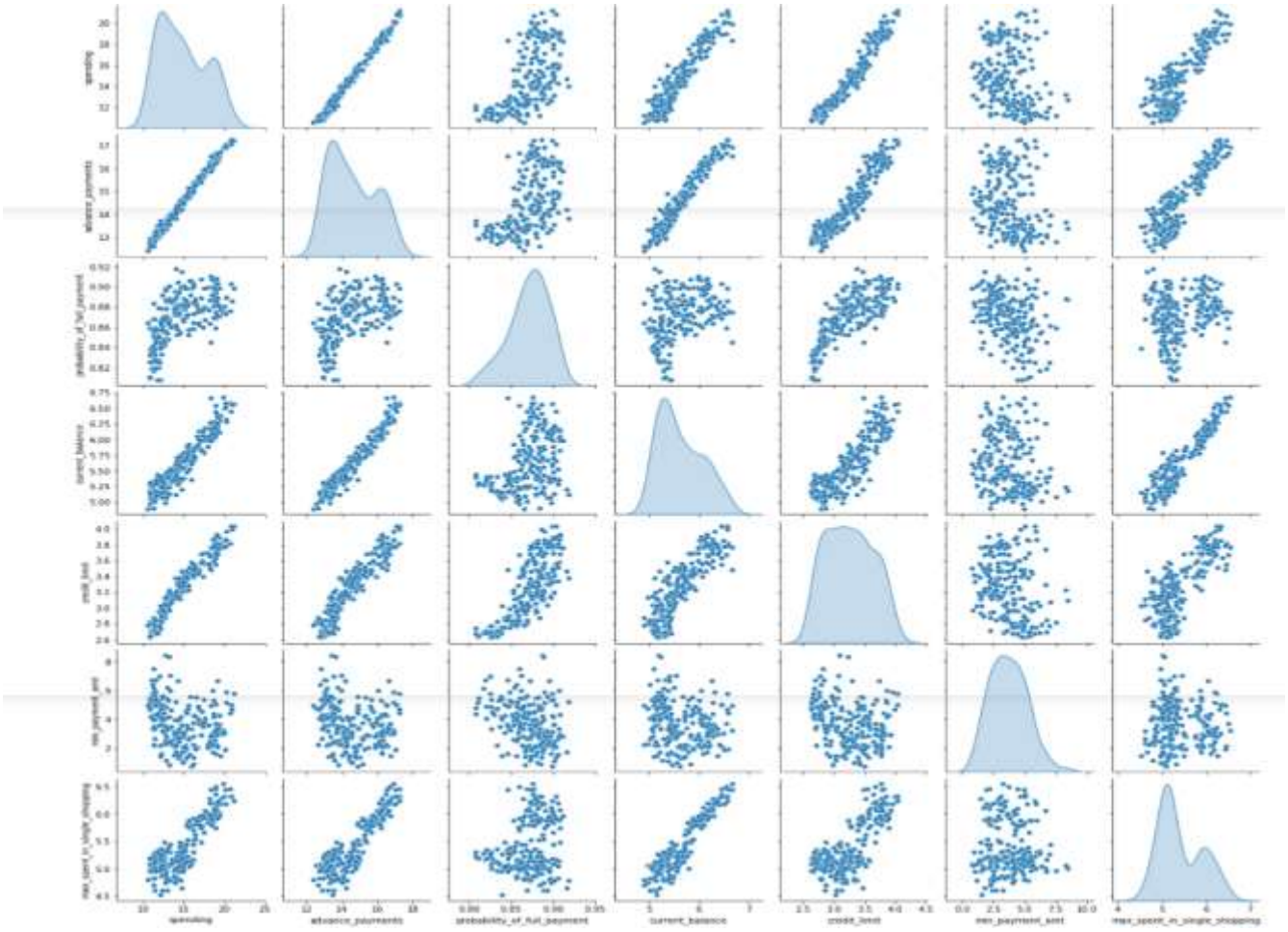
Distribution of all the variables in the dataset



Correlation matrix



Pair Plot of different variables



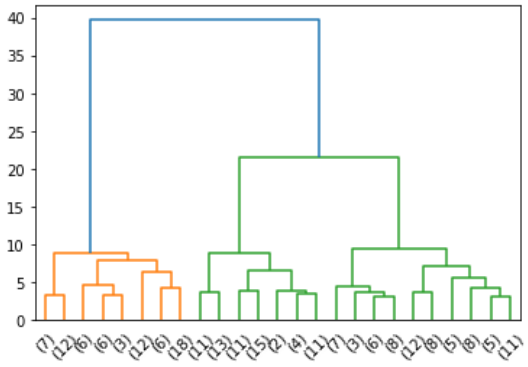
1.2 Do you think scaling is necessary for clustering in this case? Justify

Scaling needs to be done as the values of the variables are different and vary too much, for example the spending, advance_payments are in different values and this may get more weightage. For this reason, the data should be scaled.

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
0	1.754355	1.811968	0.178230	2.367533	1.338579	-0.298806	2.328998
1	0.393582	0.253840	1.501773	-0.600744	0.858236	-0.242805	-0.538582
2	1.413300	1.428192	0.504874	1.401485	1.317348	-0.221471	1.509107
3	-1.384034	-1.227533	-2.591878	-0.793049	-1.639017	0.987884	-0.454961
4	1.082581	0.998364	1.196340	0.591544	1.155464	-1.088154	0.874813

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

Dendrogram



Scaled and clustered results of dendrogram with ward linkage

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	cluster
0	1.754355	1.811968	0.178230	2.367533	1.338579	-0.298806	2.328998	1
1	0.393582	0.253840	1.501773	-0.600744	0.858236	-0.242805	-0.538582	3
2	1.413300	1.428192	0.504874	1.401485	1.317348	-0.221471	1.509107	1
3	-1.384034	-1.227533	-2.591878	-0.793049	-1.639017	0.987884	-0.454961	2
4	1.082581	0.998364	1.196340	0.591544	1.155464	-1.088154	0.874813	1

From the above we can see that the model is clustered into almost three equal number of clusters. We can cluster the above into high, medium and low spending groups with 1 as high spending, 3 as medium and 2 as low spending groups.

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

KMeans clustering with n =2

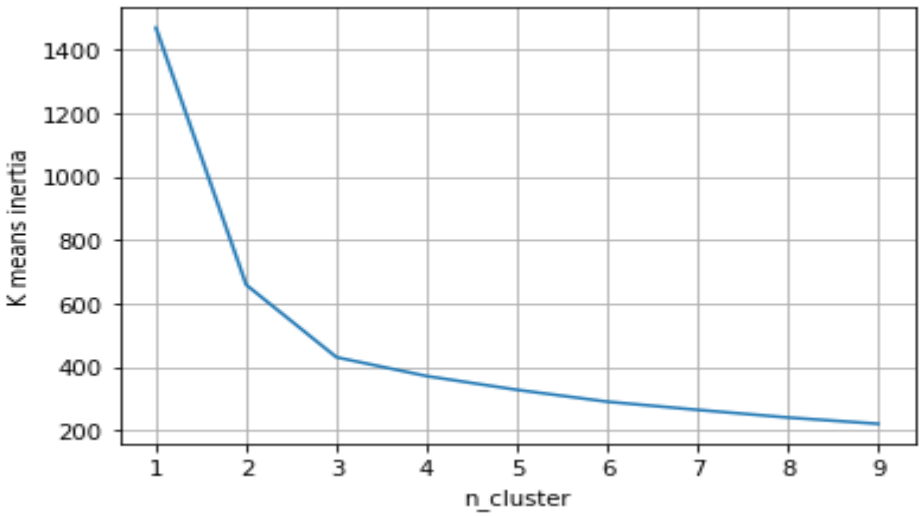
```
The kmeans clustering with n=2 has an inertia of 659.17
The kmeans clustering with n=2 has a silhouette score of 0.4658
The kmeans clustering with n=2 has a silhouette width of -0.0062
```

KMeans clustering with n =3

```
The kmeans clustering with n=3 has an inertia of 430.66
The kmeans clustering with n=3 has a silhouette score of 0.4007
The kmeans clustering with n=3 has a silhouette width of 0.0027
```

KMeans clustering with n= 4

```
The kmeans clustering with n=4 has an inertia of 371.30
The kmeans clustering with n=4 has a silhouette score of 0.3276
The kmeans clustering with n=4 has a silhouette width of -0.0538
```



From the above we can conclude that n-clusters of 3 is the best and suitable parameter from the silhouette score and samples,

- silhouette width ~ 1: the model is well separated
- silhouette width ~ 0: are separated but not well enough
- silhouette width ~ -1: then the model has done a blunder

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

Clusters from Dendrogram

cluster	1	2	3
spending	1.213983	-1.024932	-0.223402
advance_payments	1.217445	-0.999559	-0.250010
probability_of_full_payment	0.568505	-0.972589	0.347508
current_balance	1.198256	-0.881418	-0.340041
credit_limit	1.130594	-1.088249	-0.085328
min_payment_amt	-0.040697	0.832836	-0.725360
max_spent_in_single_shopping	1.242686	-0.583025	-0.656511
Freq	70.000000	67.000000	73.000000

Clusters from KMeans

clusterLabels	0	1	2
spending	-1.030253	1.256682	-0.141119
advance_payments	-1.006649	1.261966	-0.170043
probability_of_full_payment	-0.964905	0.560464	0.449606
current_balance	-0.897685	1.237883	-0.257814
credit_limit	-1.085583	1.164852	0.001647
min_payment_amt	0.694804	-0.045219	-0.661919
max_spent_in_single_shopping	-0.624809	1.292308	-0.585893
Freq	72.000000	67.000000	71.000000

Both the models have returned a set of three clusters predominantly

Group 1: High spending group

- The high spending group should be given incentives to spend more
- Max spent in single shopping is high, so memberships or discount can be given which
- Coupons and cashbacks can be given on every purchase
- More credit limit can be given

Group 2: Medium spending group

- This group has a moderate spending, the credit limit can be increased
- Since the probability of full payment is almost as equal as high spending group, cashbacks and can be made eligible for some early bird offers

Group 3: Low spending group

- Should be reminded to pay bills often
- Since credit score is low, should set a limit on their purchases
- Since min payment amount is high, products like daily needs can be targeted.

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?

Head of the data

	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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3000 entries, 0 to 2999
Data columns (total 10 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Age            3000 non-null   int64  
 1   Agency_Code    3000 non-null   object 
 2   Type           3000 non-null   object 
 3   Claimed        3000 non-null   object 
 4   Commision      3000 non-null   float64
 5   Channel        3000 non-null   object 
 6   Duration       3000 non-null   int64  
 7   Sales          3000 non-null   float64
 8   Product Name   3000 non-null   object 
 9   Destination    3000 non-null   object 
dtypes: float64(2), int64(2), object(6)
memory usage: 234.5+ KB
```

The data has 6 objects , 2 integers and 2 floats

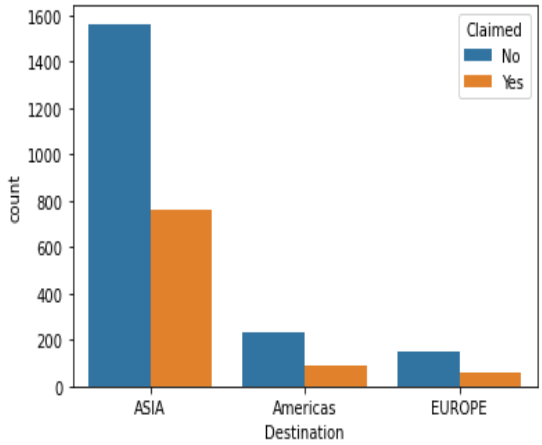
```
object     6
int64      2
float64    2
```

Dropping all the duplicate values

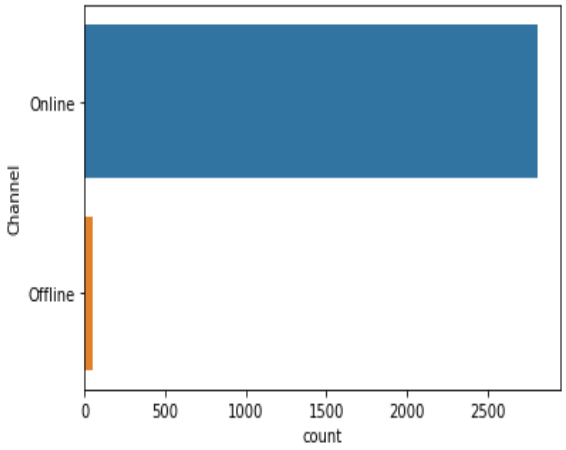
```
Number of duplicate rows = 139
```

EDA

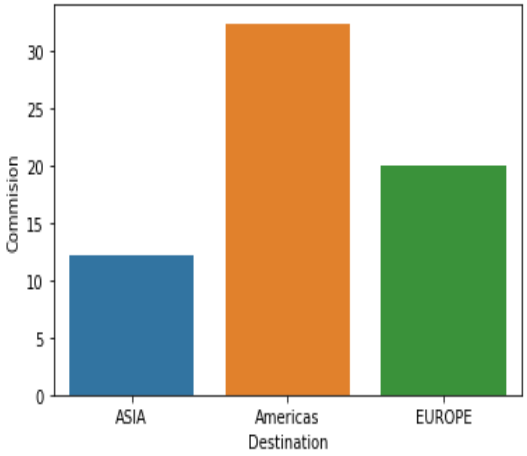
Destination count

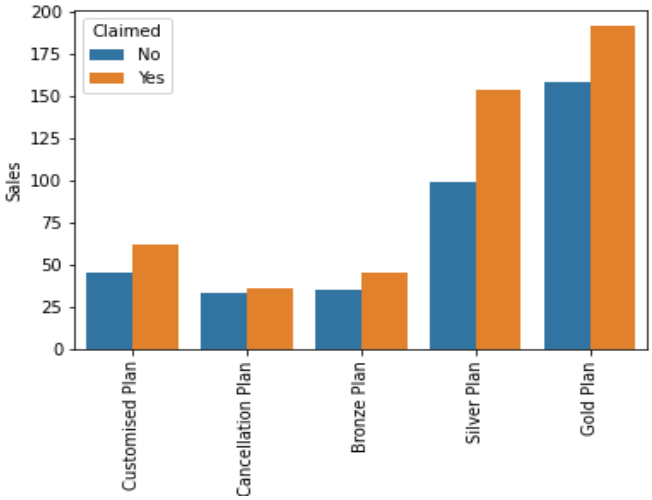


Channel Count

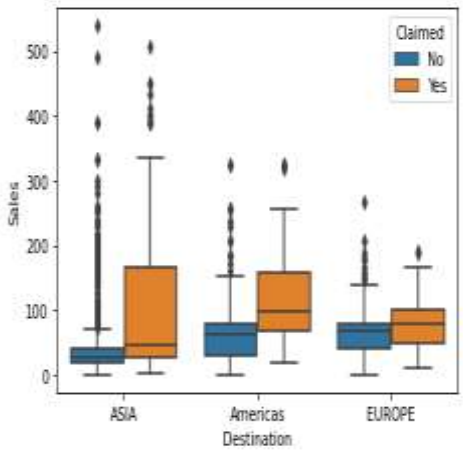


Destination vs Commission



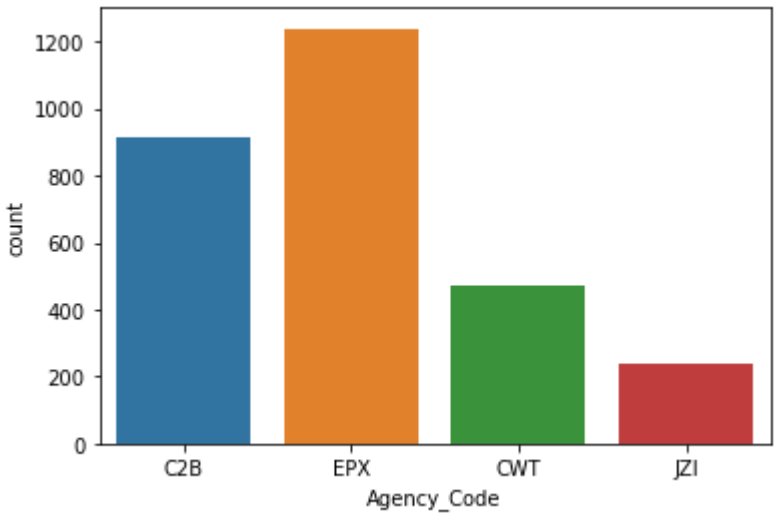


Decision vs Sales vs Claimed

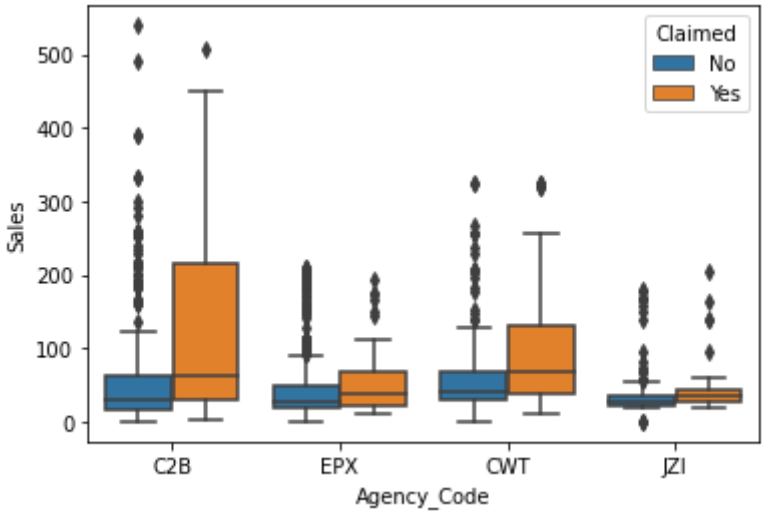


Sales Vs Product Name vs Claimed

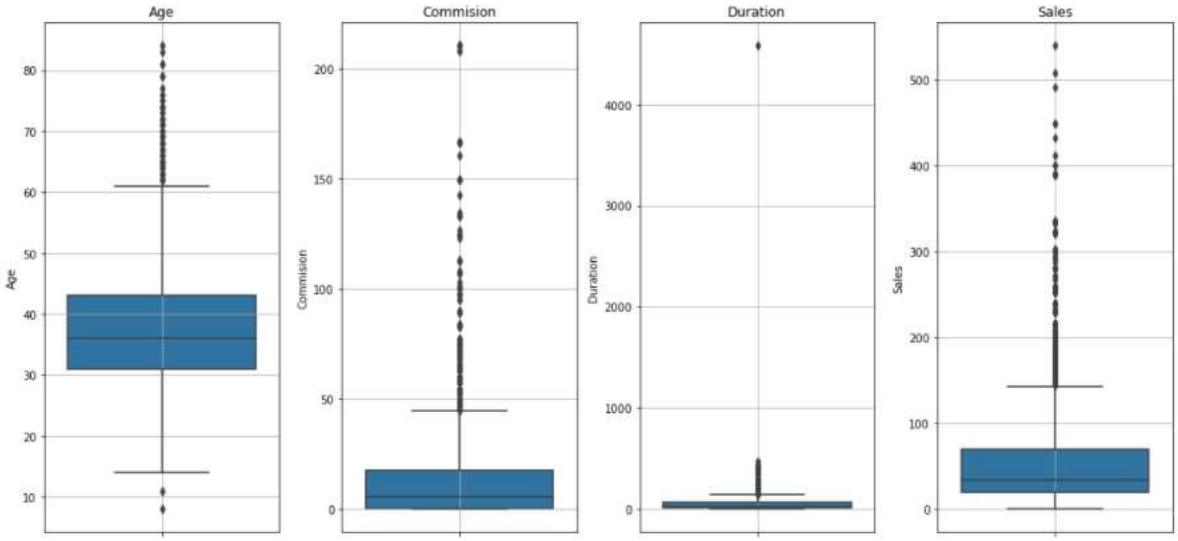
Agency code Count



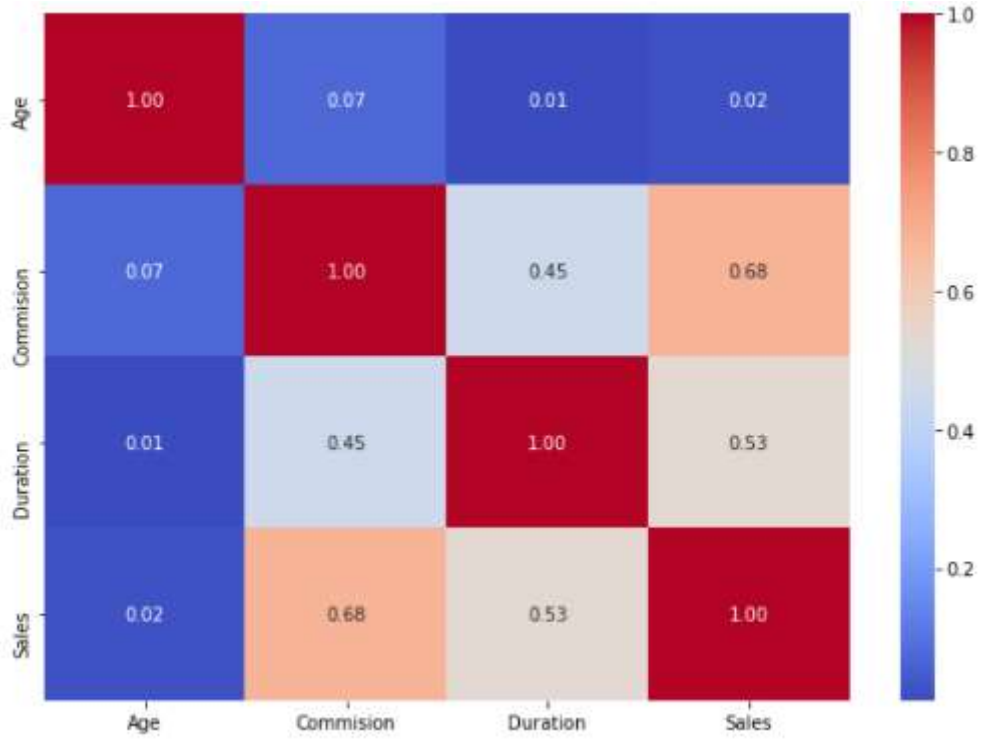
Agency code vs Sales vs Claimed



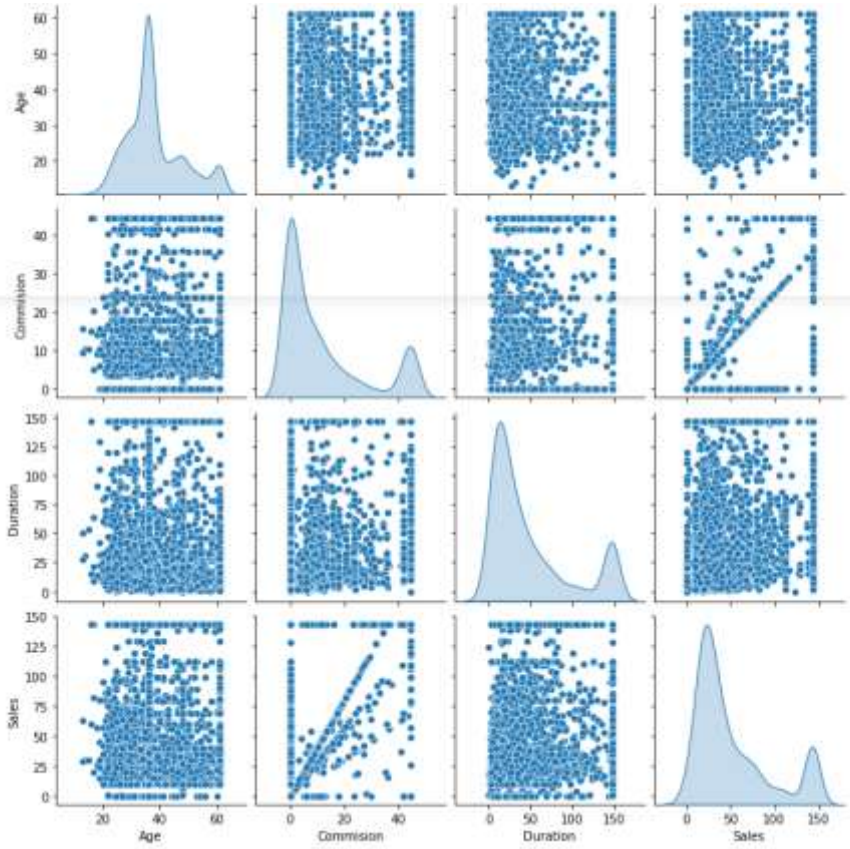
Box plot of continuous variables



Correlation heat map



Pair plot of variables



Different Parameters and their split in the given data with respect to the groups

```
AGENCY_CODE :  4
C2B     913
CWT     471
EPX    1238
JZI     239
Name: Agency_Code, dtype: int64
--------
--------
TYPE :  2
Airlines         1152
Travel Agency    1709
Name: Type, dtype: int64
--------
--------
CLAIMED :  2
No     1947
Yes     914
Name: Claimed, dtype: int64
--------
--------
CHANNEL :  2
Offline      46
Online     2815
Name: Channel, dtype: int64
--------
--------
PRODUCT NAME :  5
Bronze Plan           645
Cancellation Plan     615
Customised Plan      1071
Gold Plan             109
Silver Plan           421
Name: Product Name, dtype: int64
--------
--------
DESTINATION :  3
ASIA        2327
Americas     319
EUROPE       215
Name: Destination, dtype: int64
```

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

Decision Tree model

```
paramdt = {
    'max_depth': [6,10],
    'min_samples_leaf': [20,40],
    'min_samples_split': [100, 200]
}
dtmodel = DecisionTreeClassifier(criterion='gini',random_state=1)
gridsearchdt = GridSearchCV(estimator=dtmodel , param_grid=paramdt,cv=3)
gridsearchdt.fit(X_train,y_train)
```

```
GridSearchCV(cv=3, estimator=DecisionTreeClassifier(random_state=1),
             param_grid={'max_depth': [6, 10], 'min_samples_leaf': [20, 40],
                         'min_samples_split': [100, 200]})
```

Best parameters :

```
{'max_depth': 6, 'min_samples_leaf': 20, 'min_samples_split': 200}
```

Important Features:

	Imp
Age	0.012793
Agency_Code	0.559433
Type	0.000000
Commision	0.039784
Channel	0.000000
Duration	0.053915
Sales	0.289953
Product Name	0.044122
Destination	0.000000

Random Forest

```
paramrf = {
    'max_depth': [10, 20],
    'max_features': [3, 5],
    'min_samples_leaf': [8,10],
    'min_samples_split': [45, 50],
    'n_estimators': [300, 350]
}
rfmodel = RandomForestClassifier(random_state=1)
gridsearchrf = GridSearchCV(estimator=rfmodel,param_grid=paramrf,cv=3)
gridsearchrf.fit(X_train ,y_train)
```

```
GridSearchCV(cv=3, estimator=RandomForestClassifier(random_state=1),
             param_grid={'max_depth': [10, 20], 'max_features': [3, 5],
                         'min_samples_leaf': [8, 10],
                         'min_samples_split': [45, 50],
                         'n_estimators': [300, 350]})
```

Best Parameters

```
{'max_depth': 10, 'max_features': 5, 'min_samples_leaf': 8,
'min_samples_split': 50, 'n_estimators': 350}
```

Important features

```
                   Imp
Age           0.074849
Agency_Code   0.306426
Type          0.021782
Commision     0.095508
Channel       0.002516
Duration      0.095484
Sales         0.196204
Product Name  0.192793
Destination   0.014439
```

```
The accuracy of the RF model on training data is 80.17
The accuracy of the RF model on testing data is 78.93
```

ANN

```
paramann = {
    'hidden_layer_sizes': [100,200],
    'activation': ['logistic', 'relu'],
    'solver': ['sgd','adam'],
    'tol': [0.01,0.001],
    'max_iter' : [10000]
}
annmodel = MLPClassifier()
gridsearchann = GridSearchCV(estimator=annmodel , param_grid=paramann , cv = 3)
gridsearchann.fit(X_train , y_train)
```

```
GridSearchCV(cv=3, estimator=MLPClassifier(),
             param_grid={'activation': ['logistic', 'relu'],
                         'hidden_layer_sizes': [100, 200], 'max_iter': [10000],
                         'solver': ['sgd', 'adam'], 'tol': [0.01, 0.001]})
```

Best Parameters

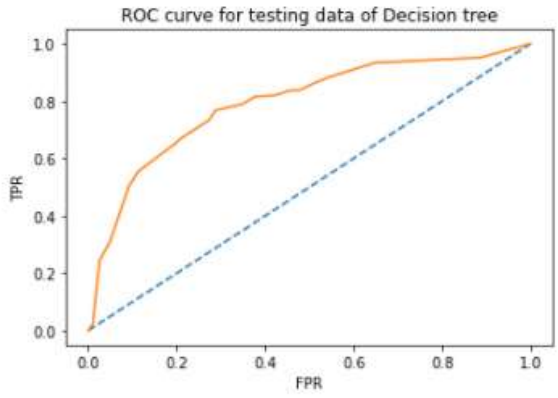
```
{'activation': 'relu', 'hidden_layer_sizes': 200,
 'max_iter': 10000,
 'solver': 'adam',
 'tol': 0.001}
```

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**

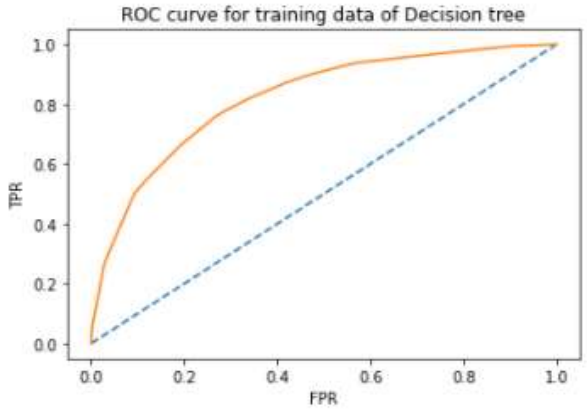
Decision Tree

```
The accuracy of the DT model on training data is 77.62
The accuracy of the DT model on testing data is  78.0
```

AUC: 0.792



AUC: 0.820




```
The area under the curve score for training set is 0.82
The area under the curve score for testing set is 0.79
```

Classification report for training data

	precision	recall	f1-score	support
0	0.79	0.90	0.85	1359
1	0.71	0.51	0.59	643
accuracy			0.78	2002
macro avg	0.75	0.71	0.72	2002
weighted avg	0.77	0.78	0.76	2002

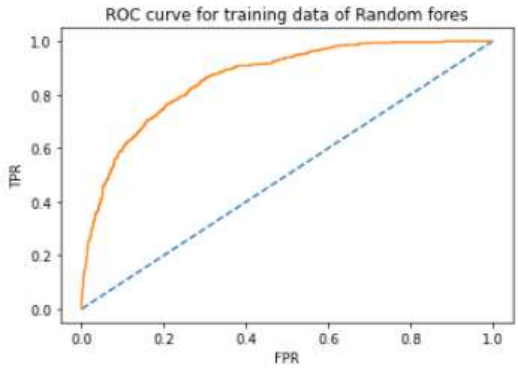
Classification report for testing data

	precision	recall	f1-score	support
0	0.80	0.91	0.85	588
1	0.72	0.50	0.59	271
accuracy			0.78	859
macro avg	0.76	0.70	0.72	859
weighted avg	0.77	0.78	0.77	859

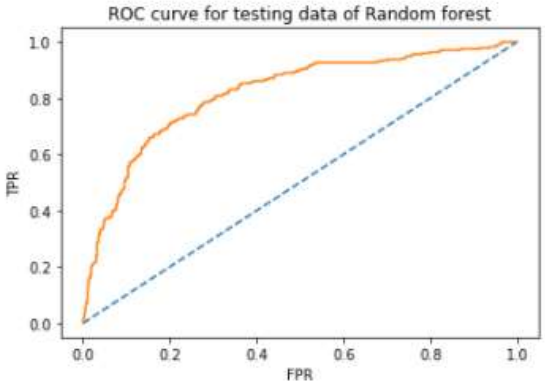
Random Forest

```
The accuracy of the RF model on training data is 80.17
The accuracy of the RF model on testing data is 78.93
```

AUC: 0.863



AUC: 0.817




```
The accuracy of the RF model on training data is 80.17
The accuracy of the RF model on testing data is 78.93
```

Classification report for training data

	precision	recall	f1-score	support
0	0.82	0.90	0.86	1359
1	0.74	0.59	0.66	643
accuracy			0.80	2002
macro avg	0.78	0.75	0.76	2002
weighted avg	0.80	0.80	0.80	2002

Classification report for testing data

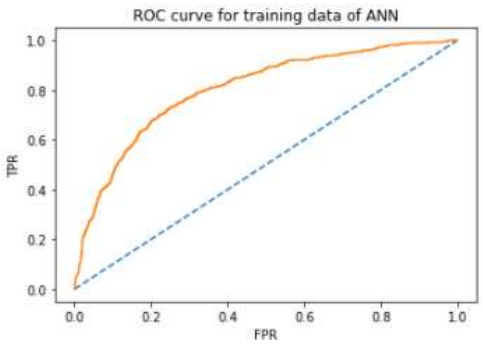
	precision	recall	f1-score	support
0	0.82	0.89	0.85	588
1	0.70	0.57	0.63	271
accuracy			0.79	859
macro avg	0.76	0.73	0.74	859
weighted avg	0.78	0.79	0.78	859

<AxesSubplot:>

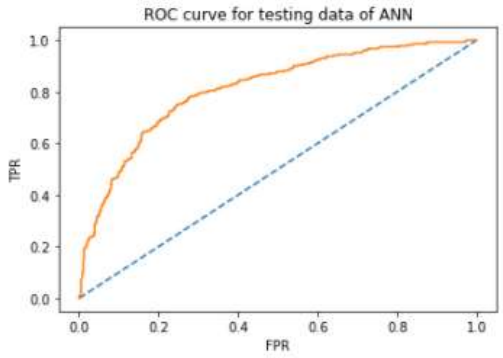
Artificial Neural Network

```
The accuracy of the ANN model on training data is 76.42
The accuracy of the ANN model on testing data is 76.95
```

AUC: 0.802



AUC: 0.810




```
The area under the curve score for training set is 0.802
The area under the curve score for testing set is 0.81
```

Classification report for training data

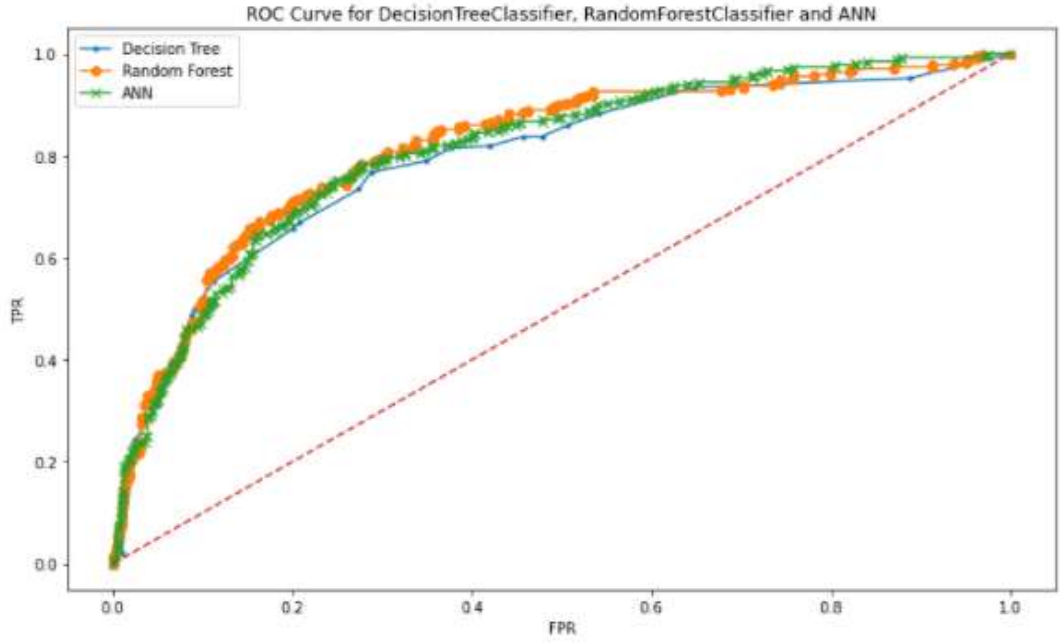
	precision	recall	f1-score	support
0	0.79	0.89	0.84	1359
1	0.68	0.49	0.57	643
accuracy			0.76	2002
macro avg	0.74	0.69	0.71	2002
weighted avg	0.75	0.76	0.75	2002

Classification report for testing data

	precision	recall	f1-score	support
0	0.79	0.90	0.84	588
1	0.69	0.49	0.57	271
accuracy			0.77	859
macro avg	0.74	0.69	0.71	859
weighted avg	0.76	0.77	0.76	859

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

```
Area under the curve for Decision Tree Classification Model is 0.7922942239626478
Area under the curve for Random Forest Classification Model is 0.8165367623063986
Area under the curve for Artificial Neural Network Model is 0.8101262645279514
```



From above we can infer that the RF model is the best model and has better accuracy, precision, recall and f1-score.

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

1. Even though the claim is mostly from Asia when compared to America and Europe the commision is a major blackhole in America and Europe. This problem needs to be addressed.
2. JZI seems to be spending the least and can be increased with more sales concentration in the same.
3. Most of the insurance is done online, so the offline customers can be pushed to go online so to increase the profits.
4. The airlines seem to make more claims so a premium can be charged accordingly.
5. People from gold plan claim a lot , so the gold plan can be further split into platinum plan and maybe more sales can be obtained.