



# BANKING CUSTOMER SEGMENTATION

# EDA on Banking Dataset

- ▶ We could see 7 variables in dataset and all are of float/numeric data type:

#	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

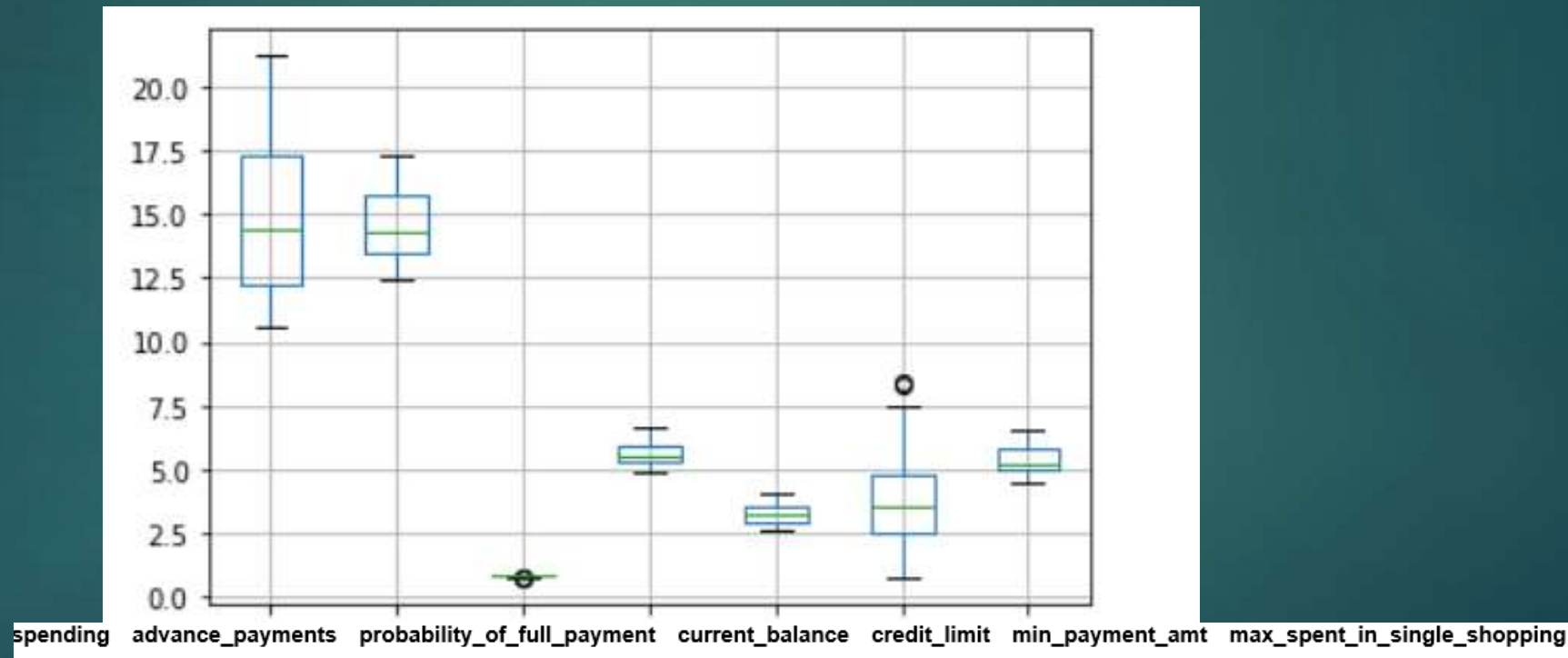
dtypes: float64(7)

- ▶ There are about 210 records of which there are no null value present among the variables

- ▶ From above Summary of variables we could note that 'spending' and 'advance\_payments' lie in one category of range while rest of the variables in another range. Since clustering techniques mainly involves calculating distance between points in a cluster it is better to scale the variables so that all are in a particular range otherwise distance calculated will lead to wrong clustering/categorization.

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
count	210.000000	210.000000	210.000000	210.000000	210.000000	210.000000	210.000000
mean	14.847524	14.559286	0.870999	5.628533	3.258605	3.700201	5.408071
std	2.909699	1.305959	0.023629	0.443063	0.377714	1.503557	0.491480
min	10.590000	12.410000	0.808100	4.899000	2.630000	0.765100	4.519000
25%	12.270000	13.450000	0.856900	5.262250	2.944000	2.561500	5.045000
50%	14.355000	14.320000	0.873450	5.523500	3.237000	3.599000	5.223000
75%	17.305000	15.715000	0.887775	5.979750	3.561750	4.768750	5.877000
max	21.180000	17.250000	0.918300	6.675000	4.033000	8.456000	6.550000

- Outlier detection: Statistically these plots depicts that there are outliers , but when looked at a perspective from this business approach we can conclude this is not a deviation from normal behavior(not an outlier).



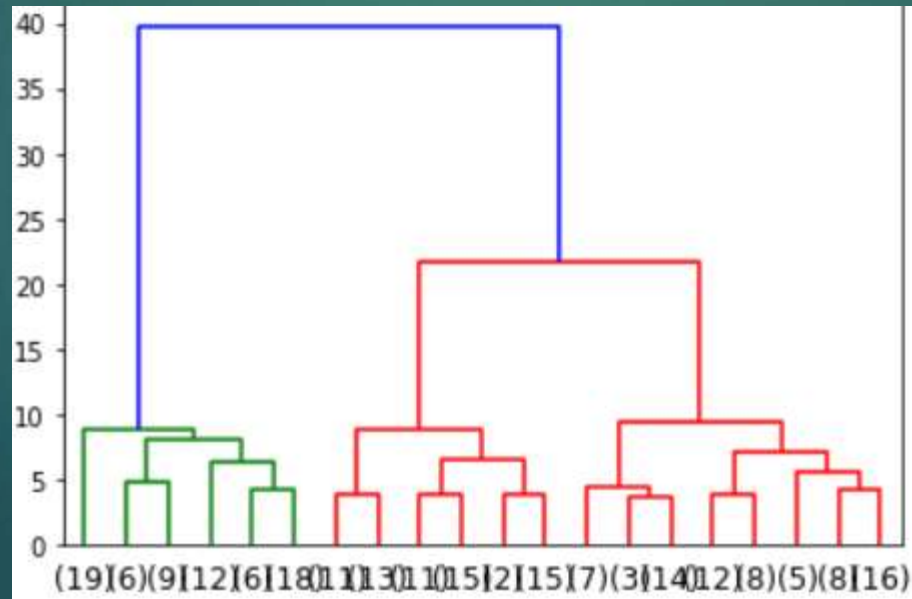
# Scaling

- ▶ From Data Dictionary we could see all are of unit currency except for the probability of full payment. Moreover variable having unit currency has variations, some are expressed in 100s and some other in 1000s. Therefore scaling becomes necessary, since clustering uses distance techniques to segregate clusters so if one feature variable is in different unit it would lead to wrong segmentation/clustering.

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)

# Hierarchical Clustering

- Based on Business approach one can choose either 2 clusters(Heavy user or Light user) or 3 clusters(Heavy,Medium or Light user).From below dendrogram choosing 2 clusters will be felt as optimum but its only property of the agglomerative clustering which will always result into 2 clusters given any dataset. So here 3 clusters is selected as optimum for segmentation.



- We could see sample segregation done by Hierarchical clustering
- 1-Heavy user,2-Light user,3-Medium user

spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	HCluste
19.94	16.92	0.8752	6.675	3.763	3.252	6.55	1
18.95	16.42	0.8829	6.248	3.755	3.368	6.148	1
17.99	15.86	0.8992	5.89	3.694	2.068	5.837	1
18.17	16.26	0.8637	6.271	3.512	2.853	6.273	1
18.55	16.22	0.8865	6.153	3.674	1.738	5.894	1
18.98	16.57	0.8687	6.449	3.552	2.144	6.453	1
17.98	15.85	0.8993	5.979	3.687	2.257	5.919	1
15.56	14.89	0.8823	5.776	3.408	4.972	5.847	1
19.51	16.71	0.878	6.366	3.801	2.962	6.185	1
18.72	16.34	0.881	6.219	3.684	2.188	6.097	1
16.87	15.65	0.8648	6.139	3.463	3.696	5.967	1
20.03	16.9	0.8811	6.493	3.857	3.063	6.32	1

spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	HCluste
10.83	12.96	0.8099	5.278	2.641	5.182	5.185	2
12.7	13.41	0.8874	5.183	3.091	8.456	5	2
12.02	13.33	0.8503	5.35	2.81	4.271	5.308	2
11.23	12.88	0.8511	5.14	2.795	4.325	5.003	2
12.15	13.45	0.8443	5.417	2.837	3.638	5.338	2
12.79	13.53	0.8786	5.224	3.054	5.483	4.958	2
10.8	12.57	0.859	4.981	2.821	4.773	5.063	2
12.7	13.71	0.8491	5.386	2.911	3.26	5.316	2
12.37	13.47	0.8567	5.204	2.96	3.919	5.001	2
13.07	13.92	0.848	5.472	2.994	5.304	5.395	2
12.62	13.67	0.8481	5.41	2.911	3.306	5.231	2
11.02	13	0.8189	5.325	2.701	6.735	5.163	2
11.35	13.12	0.8291	5.176	2.668	4.337	5.132	2

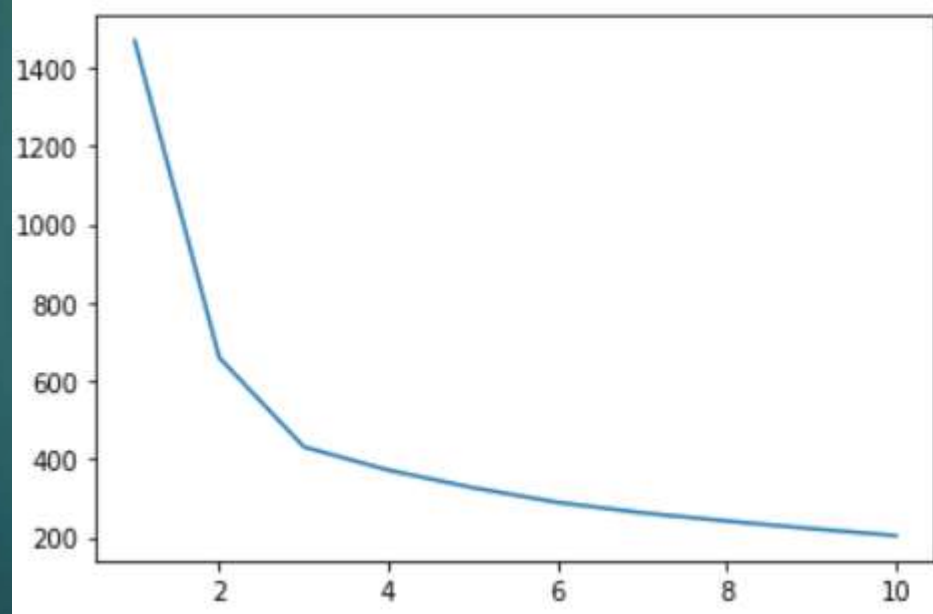
spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	HCluste
15.99	14.89	0.9064	5.363	3.582	3.336	5.144	3
13.74	14.05	0.8744	5.482	3.114	2.932	4.825	3
14.09	14.41	0.8529	5.717	3.186	3.92	5.299	3
12.1	13.15	0.8793	5.105	2.941	2.201	5.056	3
16.14	14.99	0.9034	5.658	3.562	1.355	5.175	3
13.22	13.84	0.868	5.395	3.07	4.157	5.088	3
15.11	14.54	0.8986	5.579	3.462	3.128	5.18	3
12.78	13.57	0.8716	5.262	3.026	1.176	4.782	3
13.78	14.06	0.8759	5.479	3.156	3.136	4.872	3
15.88	14.9	0.8988	5.618	3.507	0.7651	5.091	3
13.84	13.94	0.8955	5.324	3.379	2.259	4.805	3
13.16	13.82	0.8662	5.454	2.975	0.8551	5.056	3
14.79	14.52	0.8819	5.545	3.291	2.704	5.111	3
15.26	14.85	0.8696	5.714	3.242	4.543	5.314	3
14.34	14.37	0.8726	5.63	3.19	1.313	5.15	3



# KMeans

- ▶ Considering WSS-plot and silhouette score we would be selecting the no:of clusters. List of within sum of squares variance K=1 to 10. There is a significant drop in WSS at k=2 and at k=3. Considering WSS drop (229 units) at K=3 is significant, for this case K=3 is considered optimum.

```
1470.0,  
659.1717544870407,  
430.65897315130053,  
371.301721277542,  
327.99669125815456,  
289.26681770892736,  
263.1068038591345,  
241.70893456432825,  
220.76230412935047,  
205.94704714782335
```



Sil-score K=2 0.46577247686580914

Sil-score K=3 0.4007270552751299



- ▶ Sample clusters formed by Kmeans algorithm
- 0-Medium User,1-Heavy user,2-Light user


spending	advance_payments	probability_of_full_paym	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	KClusters
15.99	14.89	0.9064	5.363	3.582	3.336	5.144	0
13.74	14.05	0.8744	5.482	3.114	2.932	4.825	0
14.09	14.41	0.8529	5.717	3.186	3.92	5.299	0
12.1	13.15	0.8793	5.105	2.941	2.201	5.056	0
16.14	14.99	0.9034	5.658	3.562	1.355	5.175	0
15.11	14.54	0.8986	5.579	3.462	3.128	5.18	0
12.78	13.57	0.8716	5.262	3.026	1.176	4.782	0
13.78	14.06	0.8759	5.479	3.156	3.136	4.872	0
15.88	14.9	0.8988	5.618	3.507	0.7651	5.091	0
13.84	13.94	0.8955	5.324	3.379	2.259	4.805	0
13.16	13.82	0.8662	5.454	2.975	0.8551	5.056	0
19.94	16.92	0.8752	6.675	3.763	3.252	6.55	1
18.95	16.42	0.8829	6.248	3.755	3.368	6.148	1
17.99	15.86	0.8992	5.89	3.694	2.068	5.837	1
18.17	16.26	0.8637	6.271	3.512	2.853	6.273	1
18.55	16.22	0.8865	6.153	3.674	1.738	5.894	1
18.98	16.57	0.8687	6.449	3.552	2.144	6.453	1
17.98	15.85	0.8993	5.979	3.687	2.257	5.919	1
15.56	14.89	0.8823	5.776	3.408	4.972	5.847	1
19.51	16.71	0.878	6.366	3.801	2.962	6.185	1
18.72	16.34	0.881	6.219	3.684	2.188	6.097	1
16.87	15.65	0.8648	6.139	3.463	3.696	5.967	1

spending	advance_payments	probability_of_full_paym	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	KClusters
10.83	12.96	0.8099	5.278	2.641	5.182	5.185	2
12.7	13.41	0.8874	5.183	3.091	8.456	5	2
12.02	13.33	0.8503	5.35	2.81	4.271	5.308	2
11.23	12.88	0.8511	5.14	2.795	4.325	5.003	2
12.15	13.45	0.8443	5.417	2.837	3.638	5.338	2
12.79	13.53	0.8786	5.224	3.054	5.483	4.958	2
10.8	12.57	0.859	4.981	2.821	4.773	5.063	2
13.22	13.84	0.868	5.395	3.07	4.157	5.088	2
12.7	13.71	0.8491	5.386	2.911	3.26	5.316	2
12.37	13.47	0.8567	5.204	2.96	3.919	5.001	2
13.07	13.92	0.848	5.472	2.994	5.304	5.395	2
12.62	13.67	0.8481	5.41	2.911	3.306	5.231	2
11.02	13	0.8189	5.325	2.701	6.735	5.163	2
11.35	13.12	0.8291	5.176	2.668	4.337	5.132	2

# Cluster Profiling/Recommendations



- ▶ Attached is clustered data set where 0-Medium spender,1-Heavy spender,2-Low Spender.(attached as a separate file)
- ▶ Bank can consider those in cluster 1 as special/gold members, they could term them as valuable customers and provide them with exciting cashback/reward points for each transactions.
- ▶ Min payment amount is low for cluster 0 compared to cluster 2 ,while amount spent is more for cluster 0 compared to 2.
- ▶ Customers belonging to cluster 0 might considered limiting in spending more, to boost spending they could be provided with free offers/cashback rewards at a purchase above specified price

- 
- ▶ Monthly Spending is low for Customers belonging to cluster 2, assumption is that these are people who does not shop often(given the min spent is comparable with those who are in cluster 1,they shop it at one go ,we can consider them as lazy shoppers).For those people we could offer them Discount/offer codes which could be redeemed/used on a particular retailer before a certain day. By doing we could encourage them spending more.
  - ▶ Credit limit for all the customer are on the same range, they can be increased by reducing the security deposit , interest rates.
  - ▶ Loans with lower interest can be made available to Customers having a high credit score.
  - ▶ Also customers can be allowed to make huge transaction depending credit score.

THANK YOU



# INSURANCE FIRM CASE STUDY

# EDA on Insurance Firm

- ▶ We could see from below information of variables there are about 6 categorical and 4 continuous variables .No null values present in the dataset and there are about 3000 observations

#	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

- Below is the summary of continuous variables we could see a negative value for the variable Duration ,since Duration variable depicts the duration of the tour we could consider this observation as invalid and can be removed from the dataset before creating the model.

	Age	Commision	Duration	Sales
count	3000.000000	3000.000000	3000.000000	3000.000000
mean	38.091000	14.529203	70.001333	60.249913
std	10.463518	25.481455	134.053313	70.733954
min	8.000000	0.000000	-1.000000	0.000000
25%	32.000000	0.000000	11.000000	20.000000
50%	36.000000	4.630000	26.500000	33.000000
75%	42.000000	17.235000	63.000000	69.000000
max	84.000000	210.210000	4580.000000	539.000000

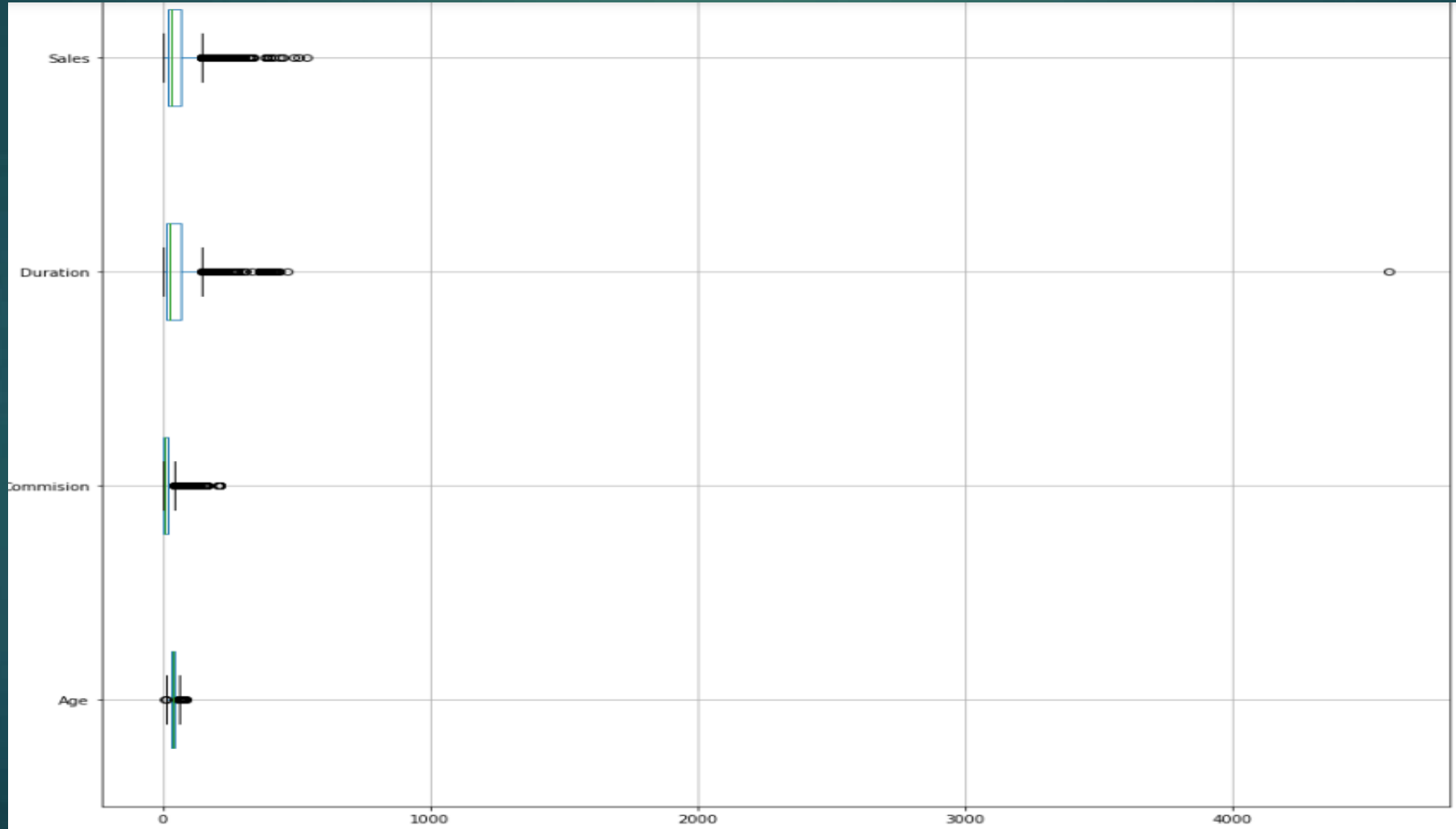
- There are about 139 duplicates present in the dataset which could be removed.

	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
63	30	C2B	Airlines	Yes	15.0	Online	27	60.0	Bronze Plan	ASIA
329	36	EPX	Travel Agency	No	0.0	Online	5	20.0	Customised Plan	ASIA
407	36	EPX	Travel Agency	No	0.0	Online	11	19.0	Cancellation Plan	ASIA
411	35	EPX	Travel Agency	No	0.0	Online	2	20.0	Customised Plan	ASIA
422	36	EPX	Travel Agency	No	0.0	Online	5	20.0	Customised Plan	ASIA
***	***	***	***	***	***	***	***	***	***	***
2940	36	EPX	Travel Agency	No	0.0	Online	8	10.0	Cancellation Plan	ASIA
2947	36	EPX	Travel Agency	No	0.0	Online	10	28.0	Customised Plan	ASIA
2952	36	EPX	Travel Agency	No	0.0	Online	2	10.0	Cancellation Plan	ASIA
2962	36	EPX	Travel Agency	No	0.0	Online	4	20.0	Customised Plan	ASIA
2984	36	EPX	Travel Agency	No	0.0	Online	1	20.0	Customised Plan	ASIA

139 rows x 10 columns



- We could see there are many outliers from the dataset. These outliers are not treated since the model used will have high tolerance to the outliers present in the dataset. Moreover practically these can be considered outliers only when approached statistically, these observations are regular and not an abnormal behaviour in practical situations



- ▶ All the feature variables are to be in numeric in nature to build a model so variables of data type object are converted to numeric type. After the conversion information of variables as follows

Data columns (total 10 columns):				
#	Column	Non-Null Count		Dtype
---	-----	-----		-----
0	Age	2860	non-null	int64
1	Agency_Code	2860	non-null	int8
2	Type	2860	non-null	int8
3	Claimed	2860	non-null	int8
4	Commision	2860	non-null	float64
5	Channel	2860	non-null	int8
6	Duration	2860	non-null	int64
7	Sales	2860	non-null	float64
8	Product Name	2860	non-null	int8
9	Destination	2860	non-null	int8

# CART, Random Forest and ANN

- ▶ Best optimized parameter for Decision Tree, Random Forest and ANN respectively is as follows:

```
{'criterion': 'gini', 'max_depth': 10, 'min_samples_leaf': 25, 'min_samples_split': 45}
```

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',  
                       max_depth=10, max_features=None, max_leaf_nodes=None,  
                       min_impurity_decrease=0.0, min_impurity_split=None,  
                       min_samples_leaf=25, min_samples_split=45,  
                       min_weight_fraction_leaf=0.0, presort='deprecated',  
                       random_state=None, splitter='best')
```

```
{'max_depth': 11, 'max_features': 6, 'min_samples_leaf': 12, 'min_samples_split': 35, 'n_estimators': 300}
```

```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,  
                       criterion='gini', max_depth=11, max_features=6,  
                       max_leaf_nodes=None, max_samples=None,  
                       min_impurity_decrease=0.0, min_impurity_split=None,  
                       min_samples_leaf=12, min_samples_split=35,  
                       min_weight_fraction_leaf=0.0, n_estimators=300,  
                       n_jobs=None, oob_score=False, random_state=None,  
                       verbose=0, warm_start=False)
```

```
{'activation': 'relu', 'hidden_layer_sizes': (200, 200, 200), 'max_iter': 10000, 'solver': 'adam', 'tol': 0.1}
```

```
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=(200, 200, 200), learning_rate='constant',  
              learning_rate_init=0.001, max_fun=15000, max_iter=10000,  
              momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True,  
              power_t=0.5, random_state=None, shuffle=True, solver='adam',  
              tol=0.1, validation_fraction=0.1, verbose=False,  
              warm_start=False)
```

# Performance Metrics

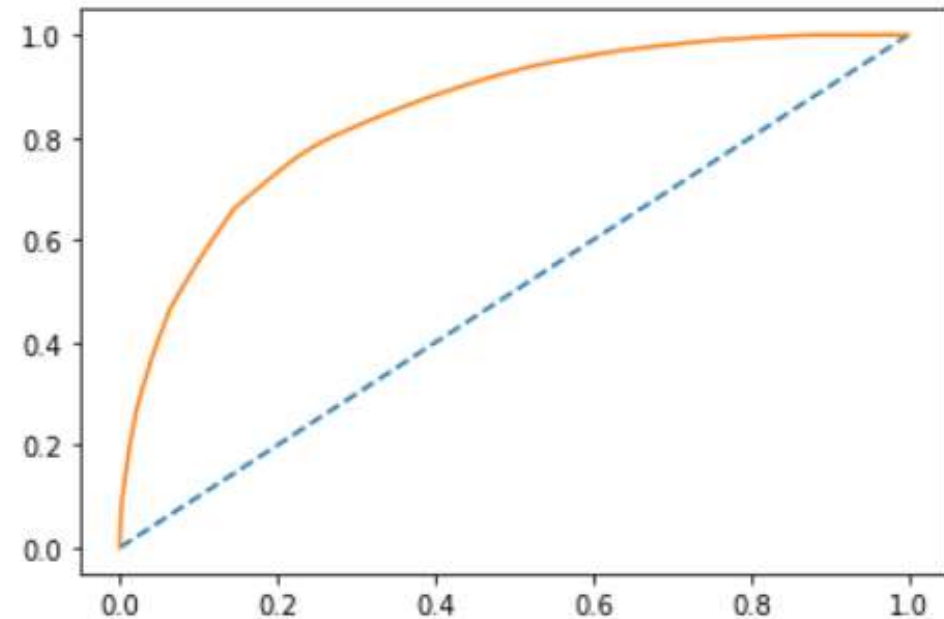
## Cart-Model:

- ▶ Train: Below is precision, recall, F1 score and accuracy of the dataset. AUC and FPR vs TPR graph is also shown.

```
cart_train_precision 0.69  
cart_train_recall    0.66  
cart_train_f1        0.68  
cart_accuracy        0.7912087912087912
```

AUC: 0.849

[<matplotlib.lines.Line2D at 0x19a8f4054e0>]

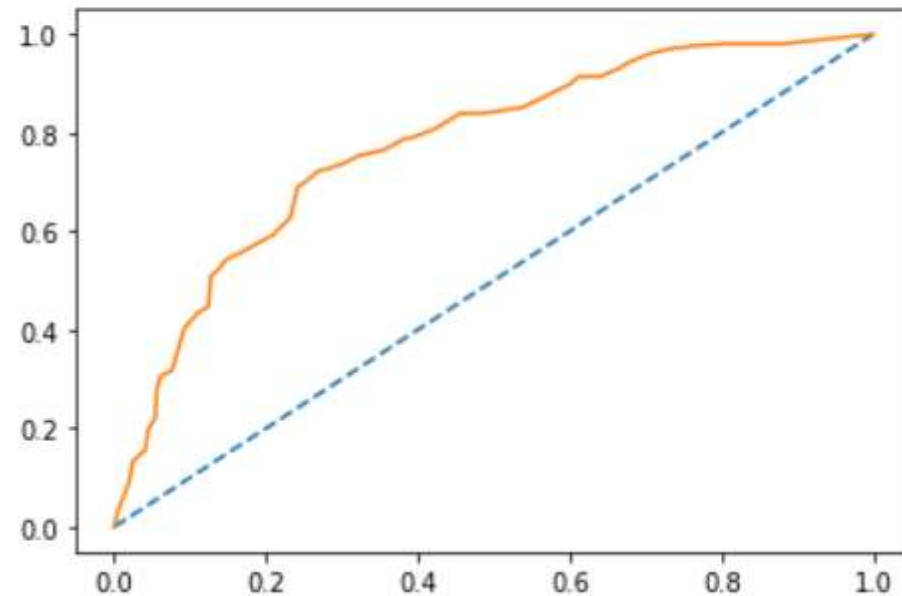


- Test: We could CART model for test data has average accuracy but below average recall and F1-score. AUC for FPR vs TPR is also pretty decent.

```
cart_test_precision 0.65  
cart_test_recall    0.48  
cart_test_f1       0.55  
cart_test_accuracy 0.7447552447552448
```

AUC: 0.774

[<matplotlib.lines.Line2D at 0x19a90e72fd0>]



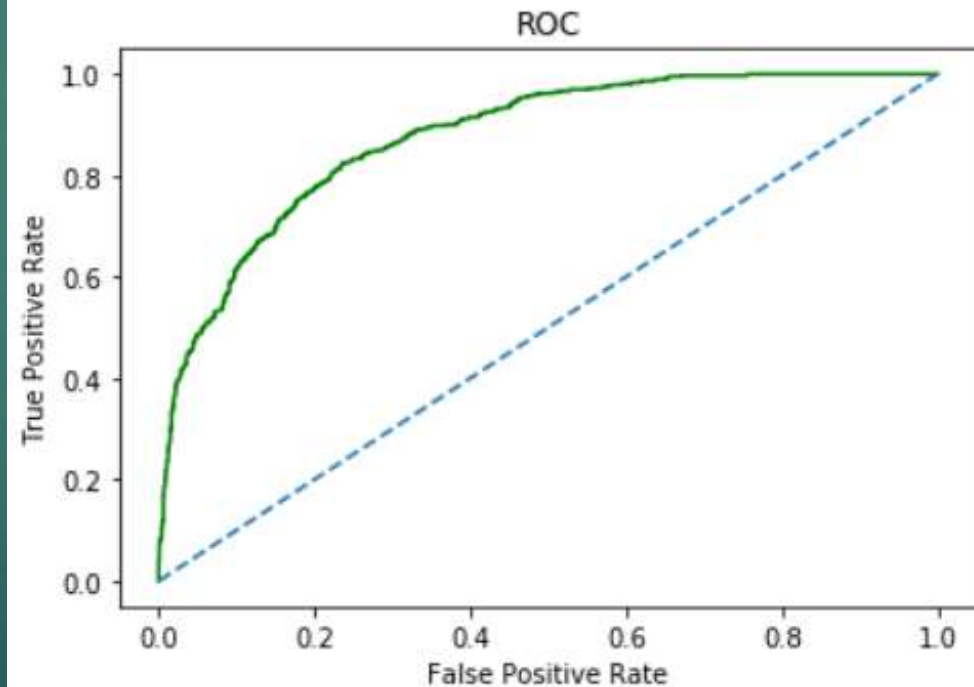
## Random Forest:

- ▶ Train: Model has good F1 score and AUC for FPR vs TPR is also good. Model performs well for training dataset

```
rf_train_precision 0.63  
rf_train_recall    0.68  
rf_train_f1       0.75  
rf_train_acc      0.8071928071928072
```

Area under Curve is 0.8758204459248595

Text(0.5, 1.0, 'ROC')

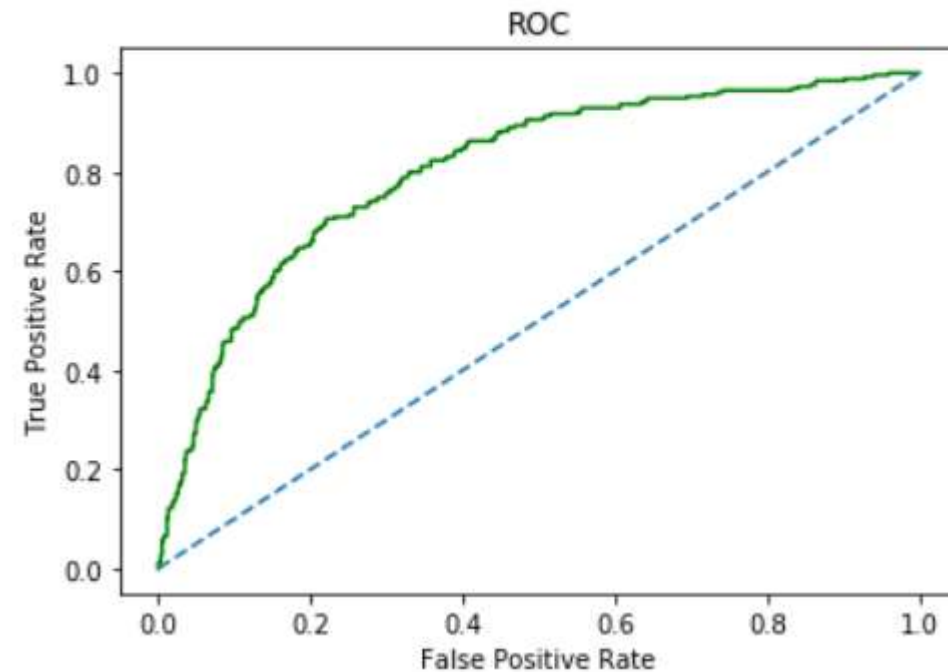


Test:RF model has poor recall on test data overall it has average accuracy and has a good AUC score

```
rf_test_precision 0.64  
rf_test_recall 0.56  
rf_test_f1 0.6  
rf_test_acc 0.7738927738927739
```

Area under Curve is 0.8048938314961143

Text(0.5, 1.0, 'ROC')





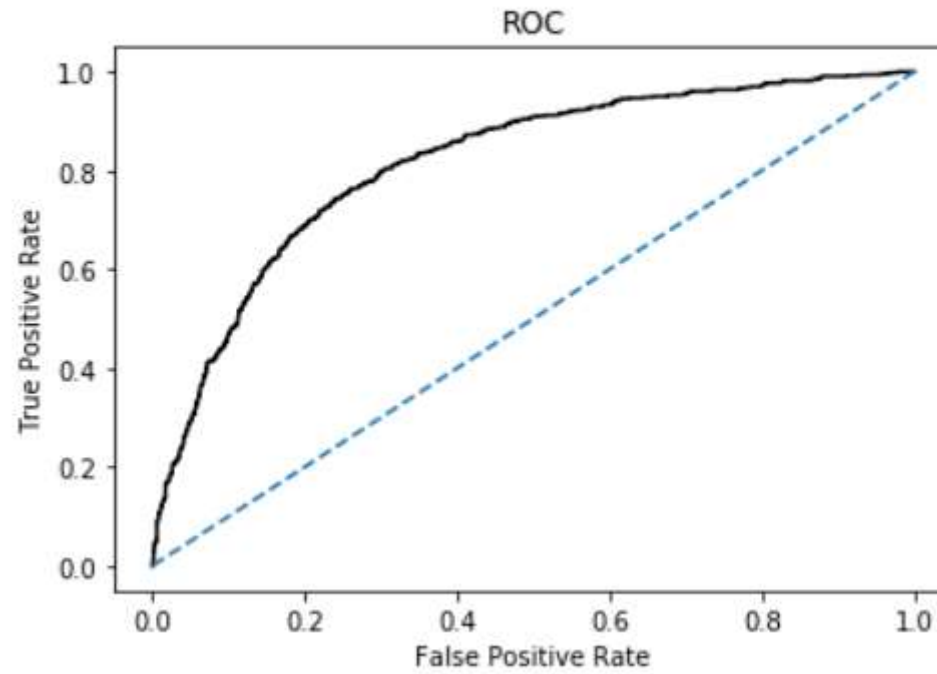
- ▶ ANN model

- ▶ Train : ANN model for training dataset has overall decent scores for precision, accuracy, recall and AUC .

```
nn_train_precision 0.53  
nn_train_recall    0.6  
nn_train_f1       0.69  
nn_train_acc      0.7657342657342657
```

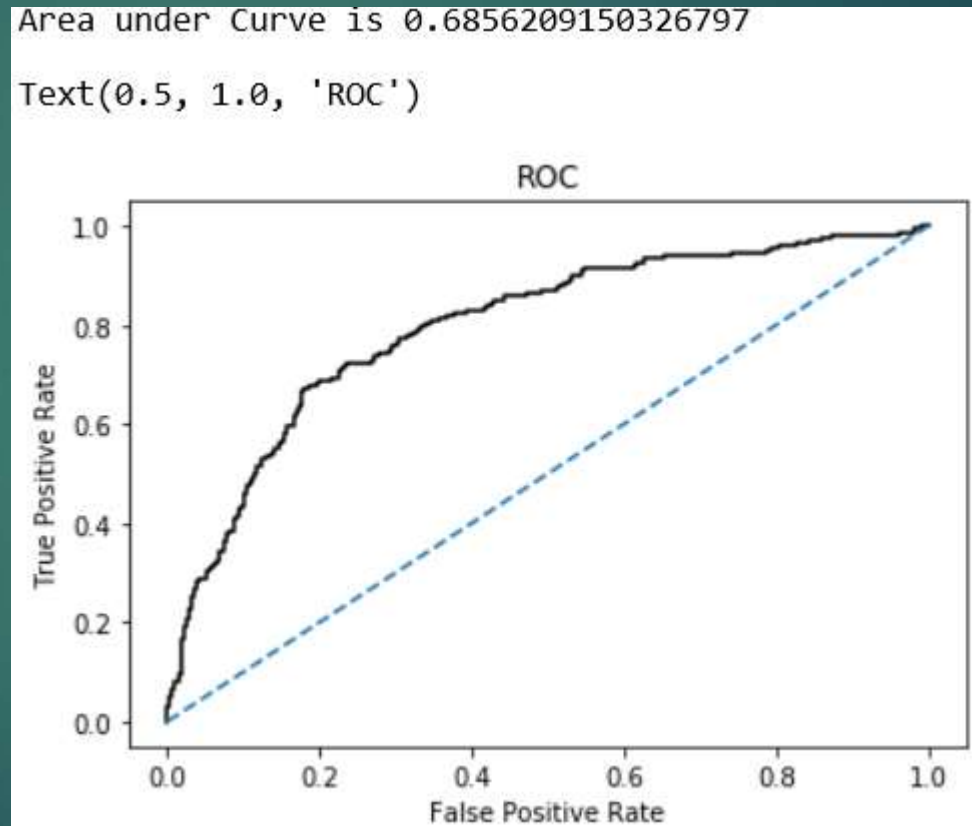
Area under Curve is 0.7059857384493529

Text(0.5, 1.0, 'ROC')



- ▶ Test: ANN model on test data has very poor recall, F1-score and precision. Accuracy is pretty average and AUC for ROC is also below par

```
nn_test_precision 0.65  
nn_test_recall    0.48  
nn_test_f1       0.55  
nn_acc_test      0.7680652680652681
```



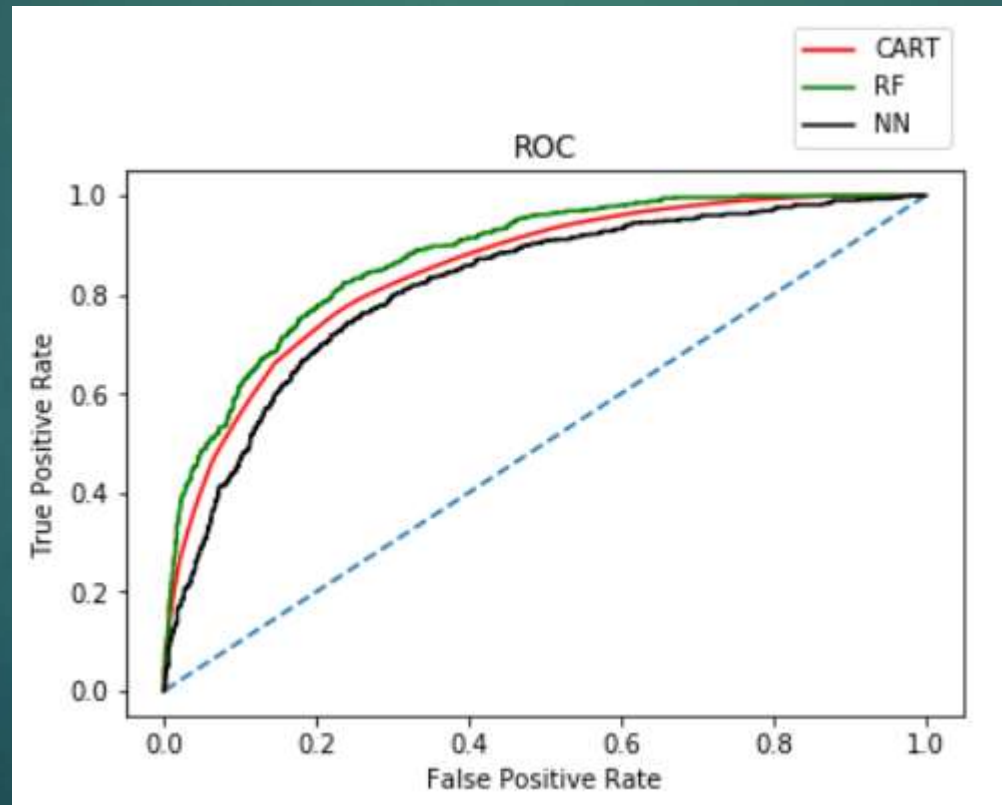
# Comparison of Models

- ▶ From below summarization of models we could see Random Forest outperforms CaRT and ANN model . We could see accuracy of the ANN model for both training dataset and test dataset performs equally well. Ability of ANN model to identify True data points as true is very poor. Recall for CaRT and RF are comparable but RF edges past the CaRT model.

	CART Train	CART Test	Random Forest Train	Random Forest Test	Neural Network Train	Neural Network Test
<b>Accuracy</b>	0.79	0.74	0.81	0.77	0.77	0.77
<b>AUC</b>	0.85	0.77	0.88	0.80	0.71	0.69
<b>Recall</b>	0.66	0.57	0.68	0.56	0.60	0.48
<b>Precision</b>	0.69	0.57	0.63	0.64	0.53	0.65
<b>F1 Score</b>	0.68	0.57	0.75	0.60	0.69	0.55

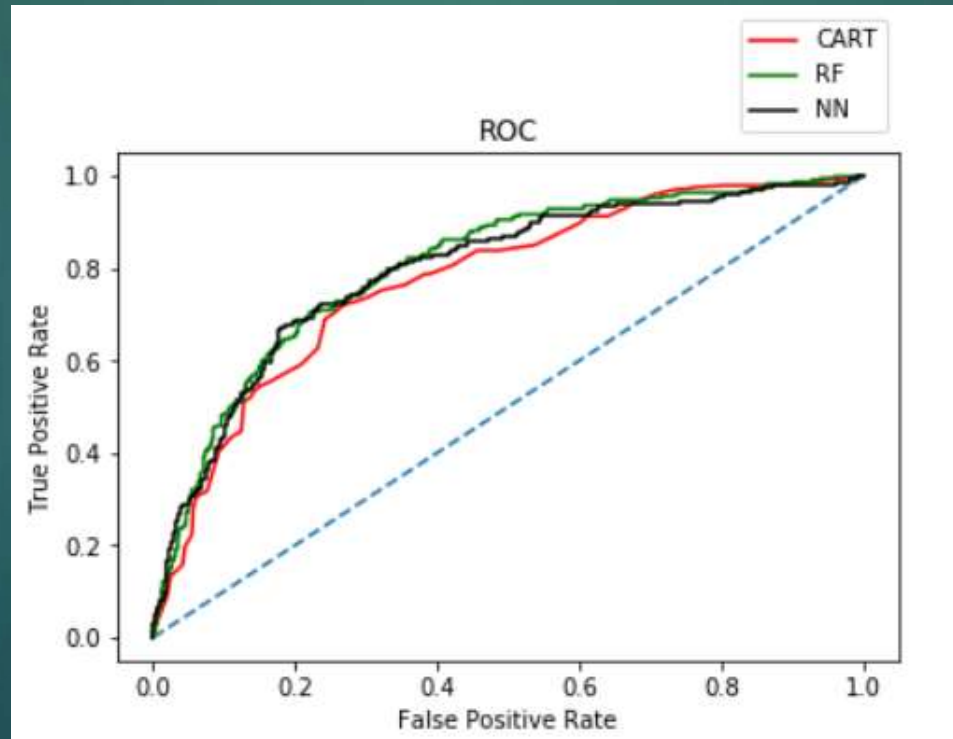
# Models vs Training Dataset

- ▶ From below ROC curve we could see RF has good AUC score as when compared to other 2 models and Ann model has the lowest AUC score



# Models vs Test Dataset

- ▶ We could see AUC score for CaRT model is poor for test data while ANN and RF ROC curve overlaps at times. ANN shows inconsistency for test data while ROC curve for RF model is a clean curve. CaRT overgrows RF and ANN farther away from Y axis (shows inconsistency)




# Inferences

- ▶ We could see Agency code , Sales and Product Name together significantly constitutes whether Insurance is claimed or not

	Feature_imp
Agency_Code	0.317959
Sales	0.208435
Product Name	0.170352
Duration	0.105533
Commision	0.089796
Age	0.069579
Type	0.019531
Destination	0.016911
Channel	0.001906

- ▶ The probability of Claiming insurance is more when the Agency code is C2B, but when combined product plan Bronze , Silver or Customized the chance of claiming insurance increases.

- 
- ▶ Since Agency Code significantly contributes to whether a insurance is claimed or not. We could have more standardized terms of policy for all the product plans.
  - ▶ Revision of Insurance Product plans is required. Terms defining the plan could be revised. More stricter agreements could be made to reduce the unneeded claims
  - ▶ Analysing the tour firms and learning about the company's history would somehow reduce the unneeded insurance claims , since mostly claim rate depends on Agency code.
  - ▶ Revisiting the terms and agreements between the insurance firm and tour firm would give an idea to proceed with the next step to find why higher claim rate.