Contemporary Topics Assignment

Sriganesh Balamurugan – 11915001
Raghu Punnamraju – 11915010
Anmol More – 11915043

Table of Contents

Feature Engineering and Data Preparation

Provider Dataset :

i) **Not useful columns** – Claim_ID is just serial no. of claims and shouldn't be used. ClmProcedureCode_6 is completely empty column

ii) **Admission Date and Discharge Dates** – Missing values has been filled with Claim Start and Claim End Date, 517737 such rows were there.

iii) **Physician Related Columns** –

a. Attended Physician – There were only 1508 claims without an attending physician. To fill these missing values – Assumption is 'Operating physician' could be the attending physician, if not value has been filled with 'Other physician'

b. Operating physician and Other Physician – Not all admissions will be necessarily having an Operating and Other physician. Assuming – Same doctor might be acting in two or three roles. Based on these assumption missing values are filled.

   There were still 1483 claims, where we couldn't fill in any physician slots, and these were filled with 'NA' considering physicians is categorical variable, hence 'NA' physician can itself be a category.

iv) **Deductible Amount Paid** – There is a correlation of almost 1, between patient Type and Deductible Amount Paid. For Patient Type = 1, deductible amount is always 1068 in data. Accordingly all missing values are filled

v) **New Features Generated** –
   a. Days Stayed – No. of days of admission in each claim
   b. Days in Claim – No. of days taken from start to end of claim
   c. Days Admission to Claim – No. of days between claim start to admission. This can help to understand the pattern when the claim was filed.
   d. Diagnosis Count – Assuming multiple diagnosis might be involved in one admission, we just count the no. of diagnosis associated with each claim.
   e. Procedure Count – Again, assuming multiple procedures might be involved in one admission we count the no of procedures in each claim

Beneficiary Data :

Overall beneficiary Data is clean compared to provider data, with no missing values except for DOD (Date of Death) which is blank for known reason.

i) **Disease Count –** New feature column has been created, which is count of all diseases, so a person suffering from diabetes and stroke will be counted as 2, and so on.
ii) **Died** – Information of death is available only for 1421 patients. So, a new column 'Died' has been created to identify whether patient died or not. Since, DOD directly cannot be related to claim fraud. This can further be extended to calculate 'No. of days since discharge' person has died, but usefulness would be low as data is available for very small subset
iii) **Disease Indicators** – All disease indicators (ChronicCond_*) columns, have been coded 0 and 1, such that 1 represents the disease presence and 0 – Not present. instead of 1s and 2s
iv) **RenalDiseaseIndictor** – This is again coded as binary, such that 1 represents the presence of disease

Combined Data for Analysis :

Both provider and beneficiary data has been combined on 'Beneficiary_ID' and grouped by Provider ID (ie. Hospital ID) for analysis purpose. Combined data is highly skewed as there are only few big hospitals probably doing maximum no. of admissions.

Columns generated are aggregated over provider –

i) **Total Deductible Amount** – Amount deductible across all claims for a hospital, this could be a major indictor as insurance is directly related to money.
ii) **Average Deductible Amount** – Average of deductible amount.
iii) **Total Insurance Amount** – Amount of insurance across all claims for a hospital.
iv) **Average Insurance Amount** – Average of insurance amount
v) **Avg Days Stayed** – Average No. of days patient stayed
vi) **Avg Days in Claim** – Average No. of days for claim settlement
vii) **Avg Days Admission to Claim** – Average No. of days from admission to claim start
viii) **Avg Age** – Average age of patients in each hospital
ix) **Avg Diagnosis Count** – Avg No. of diagnosis counts
x) **Avg Procedure Count** – Avg No. of procedure involved
xi) **Avg Months PartA Cov** – Average No. of months before part A cover
xii) **Avg Months PartB Cov –** Average No. of months before part B cover
xiii) **Avg IP Reimbursement –** Average amount for IP admission of all beneficiaries of hospital
xiv) **Avg OP Reimbursement –** Average amount for OP of all beneficiaries of hospital
xv) **Avg IP Deductible Amt –** Average deductible amount of IP admission
xvi) **Avg OP Deductible Amt –** Average deductible amount of OP
xvii) **Avg Disease Count –** Average No. of diseases per hospital across all beneficiaries

Train and Test Data Preparation

Ref : https://towardsdatascience.com/sampling-techniques-for-extremely-imbalanced-data-part-ii-over-sampling-d61b43bc4879

Train and Test data is combined with combined data for analysis. Here is the size of data –
Train data – 3998 providers
Test data – 1412 providers

Fraud Vs No Fraud Count in train set

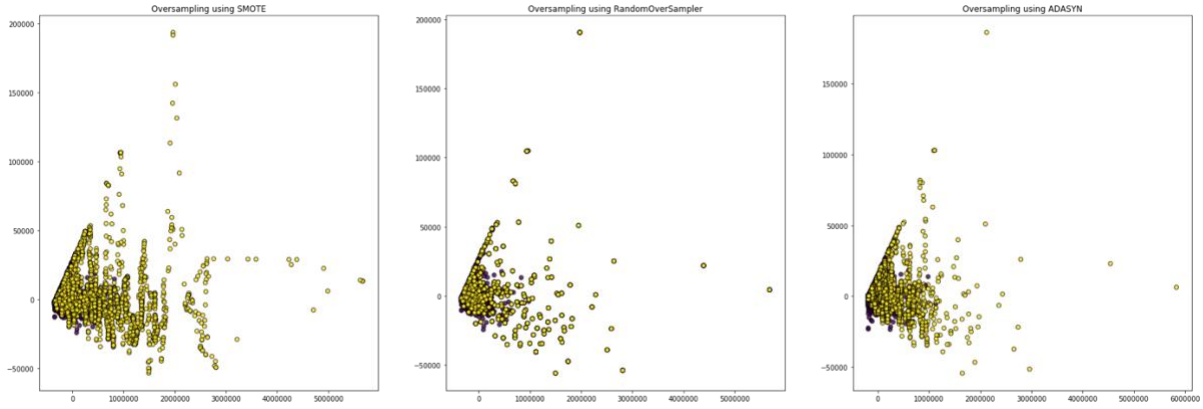
No. of fraudulent cases – 3615
No. of Non fraudulent cases – 383

We can clearly see the issue of data imbalance and fraud cases are in minority, at just 10%

Oversampling Techniques Comparison

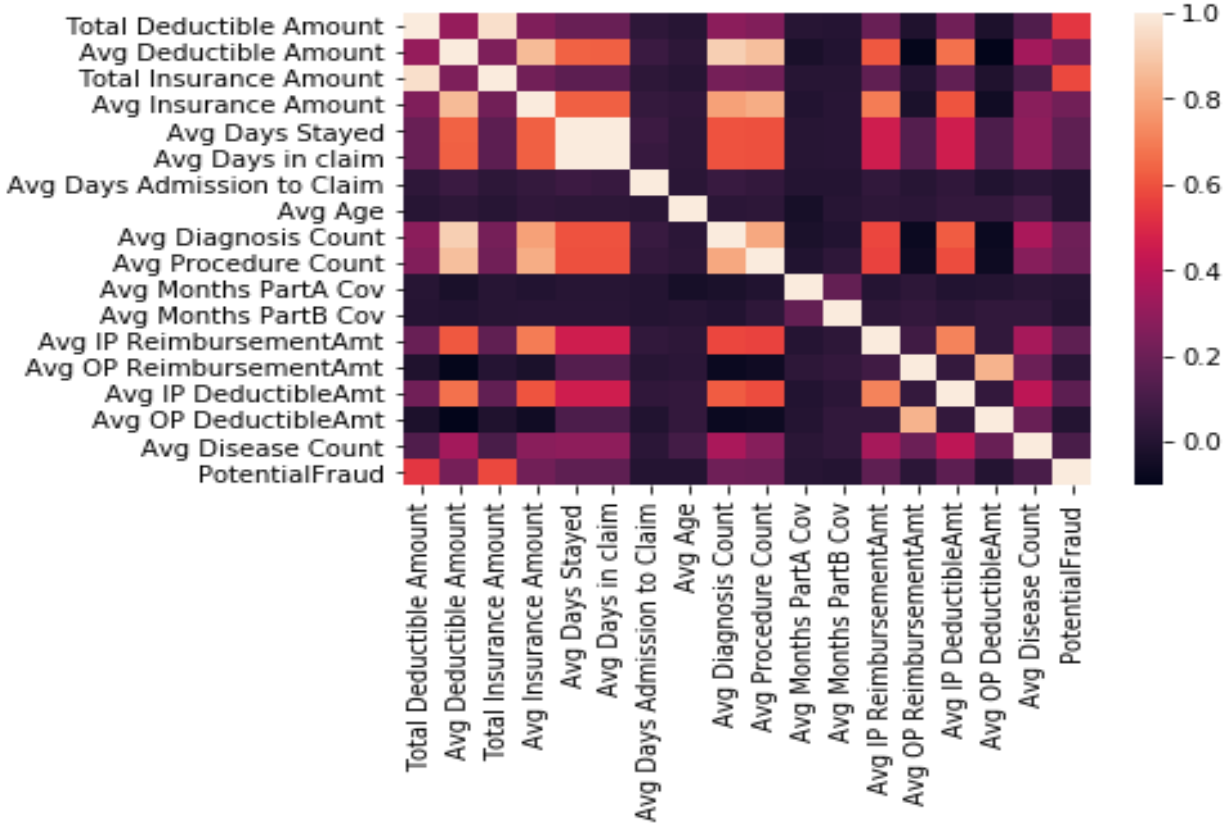
We used different oversampling techniques and found **Synthetic Minority Oversampling Technique** (SMOTE) best suited here. It creates the new under sampled classes by using K nearest neighbours and using interpolations.

We see that Random Oversampler and ADASYN is overfitting the data with not much dispersion between various classes generated.



Correlation plot

Looking at correlation plot we try to identify how features are affecting potential fraud providers

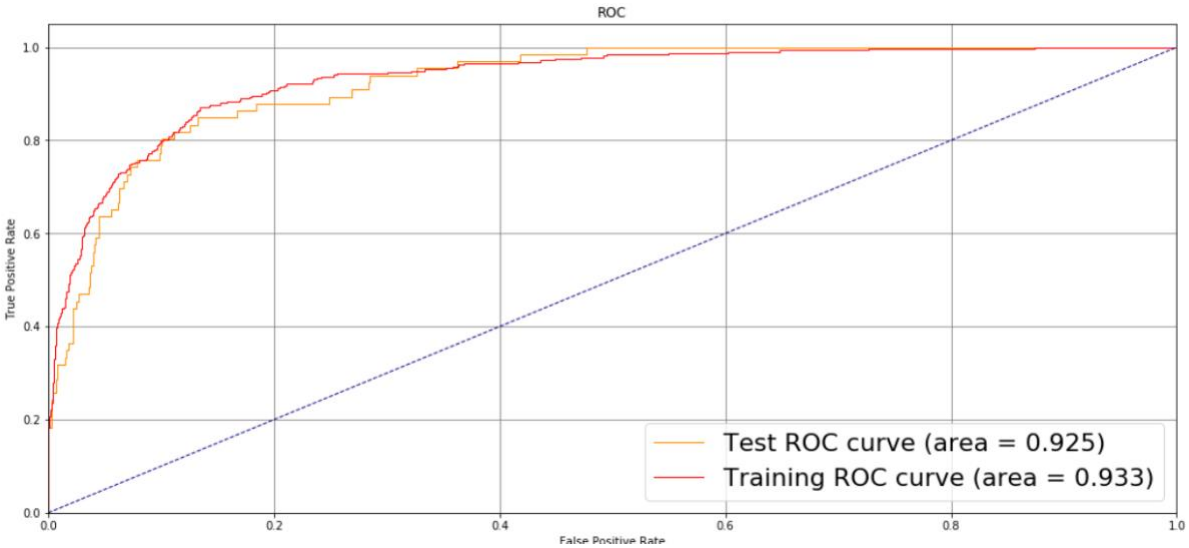


Analysis of Various Model to Predict Fraud

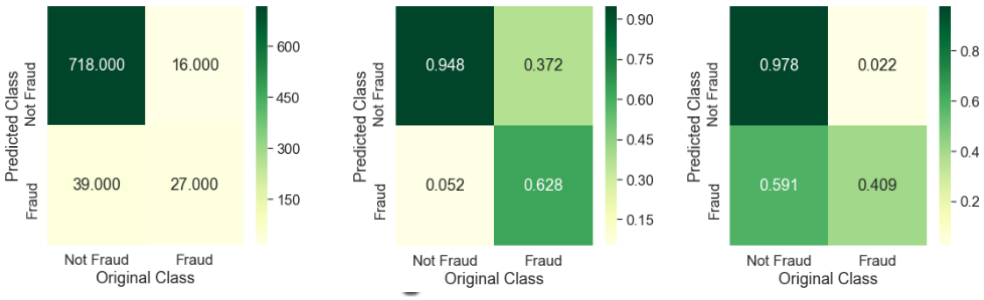
Post Feature Engineering and standardization of the data various binary classification techniques/models were applied. This was initially done without balancing the classes. The details of the same are shown below:

Logistic Regression (without Oversampling):

a. Train-Validation ROC – AUC value

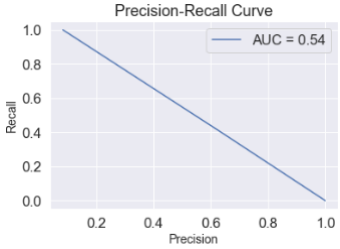


b. Confusion Matrix (Combined, Precision and Recall)



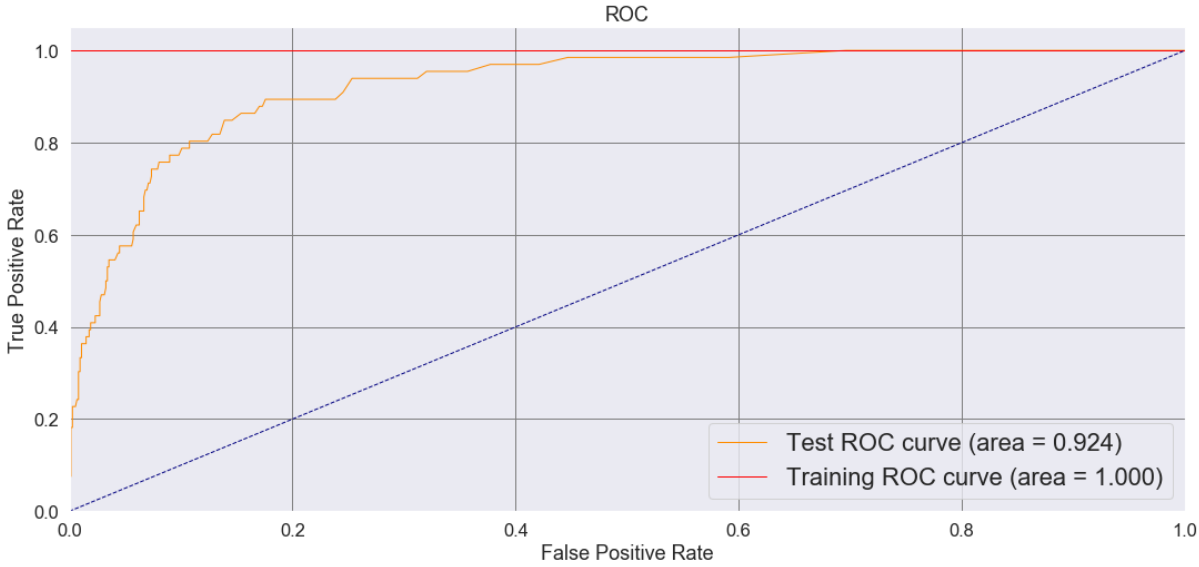
	Not Fraud	Fraud
Precision	0.948481	0.627907
Recall	0.978202	0.409091
F1-Score	0.963112	0.495413

c. Precision – Recall Curve

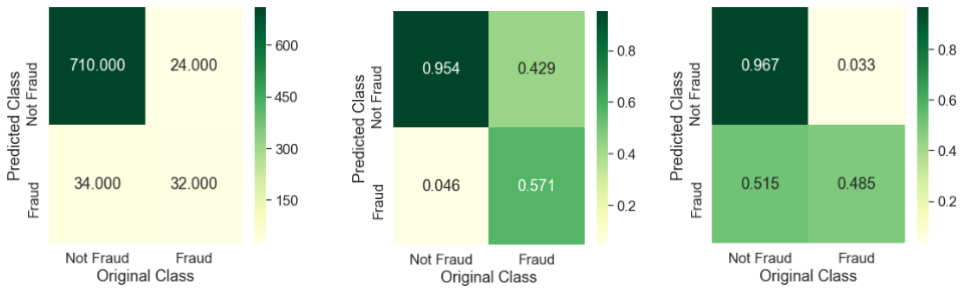


Random Forest (without Oversampling):

a. Train-Validation ROC – AUC value

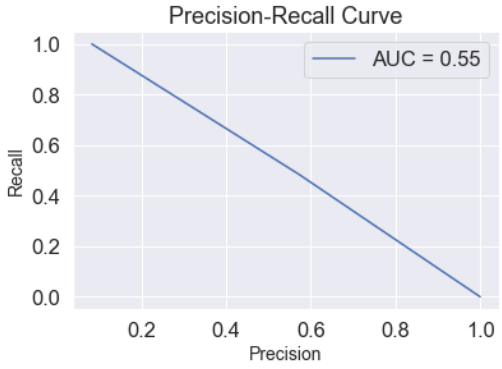


b. Confusion Matrix (Combined, Precision and Recall)



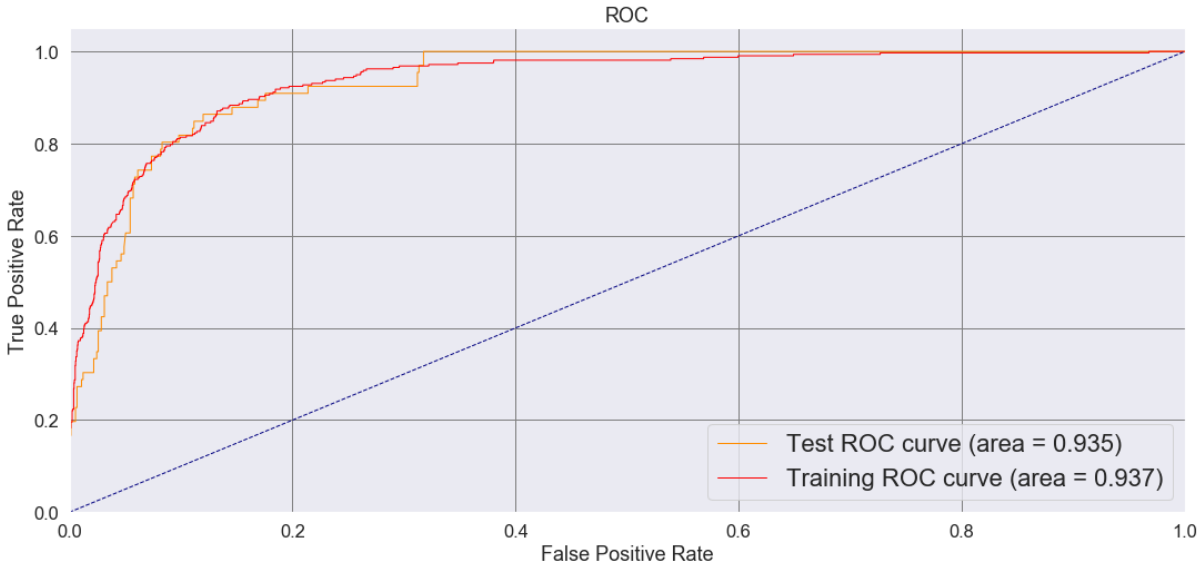
	Not Fraud	Fraud
Precision	0.954301	0.571429
Recall	0.967302	0.484848
F1-Score	0.960758	0.524590

c. Precision – Recall Curve

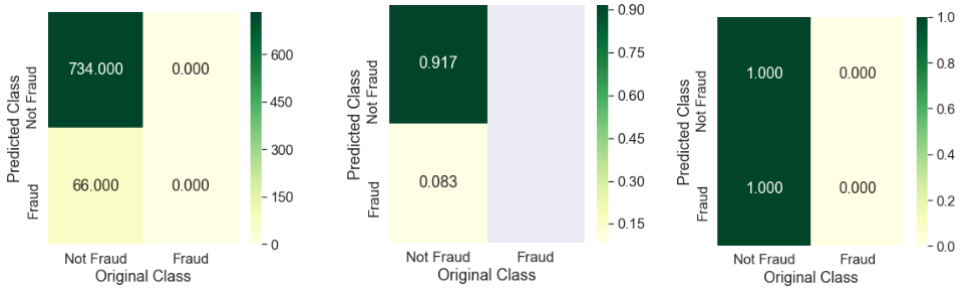


SVM (without Oversampling):

a. Train-Validation ROC – AUC value

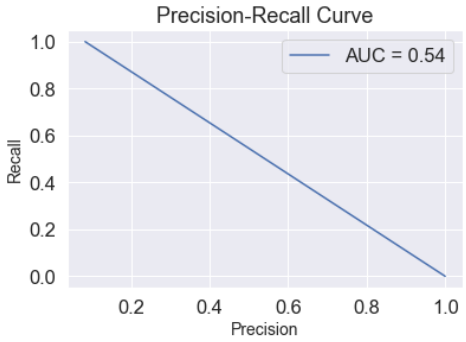


b. Confusion Matrix (Combined, Precision and Recall)



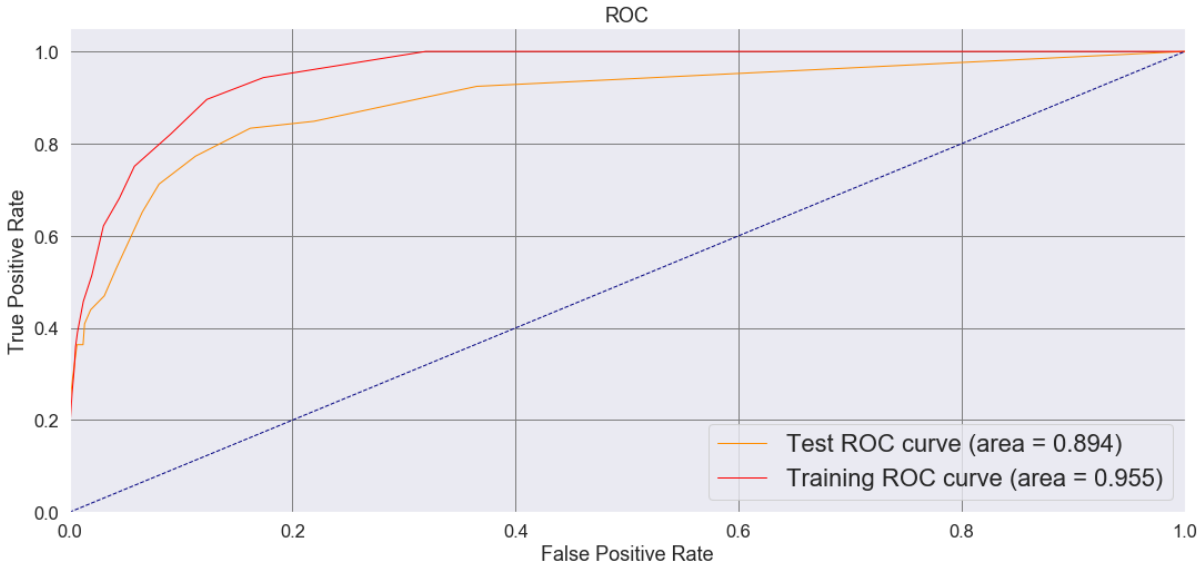
	Not Fraud	Fraud
Precision	0.917500	0.0
Recall	1.000000	0.0
F1-Score	0.956975	NaN

c. Precision – Recall Curve



KNN (without Oversampling):

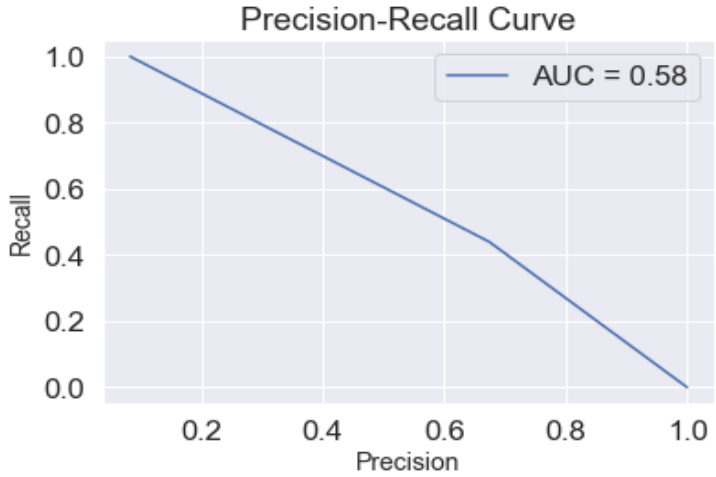
a. Train-Validation ROC – AUC value



b. Confusion Matrix (Combined, Precision and Recall)

	Not Fraud	Fraud
Precision	0.951123	0.674419
Recall	0.980926	0.439394
F1-Score	0.965795	0.532110

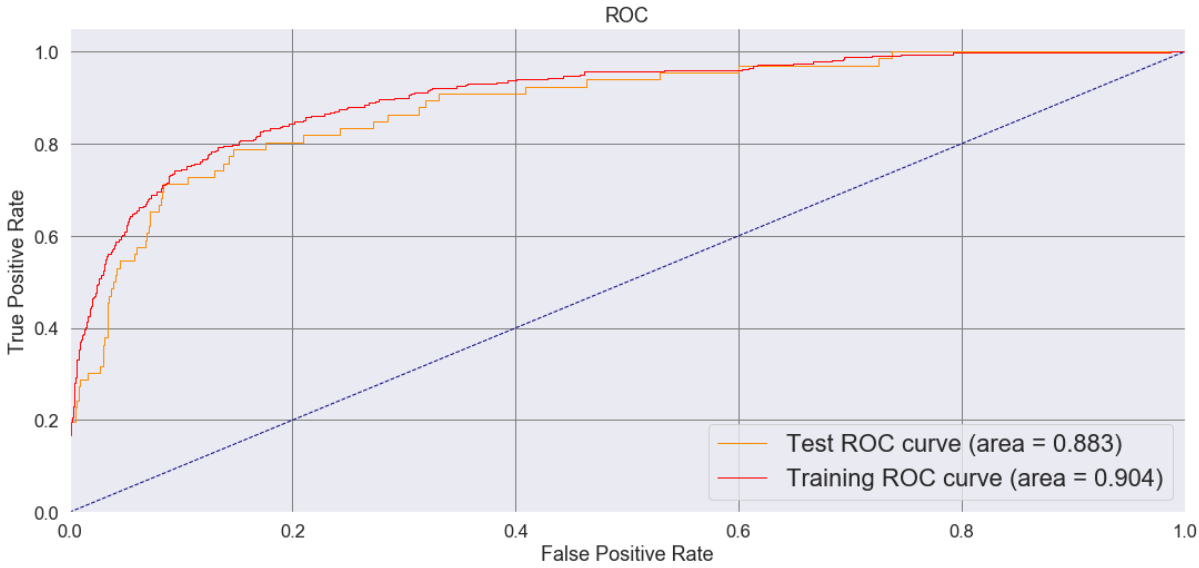
c. Precision – Recall Curve



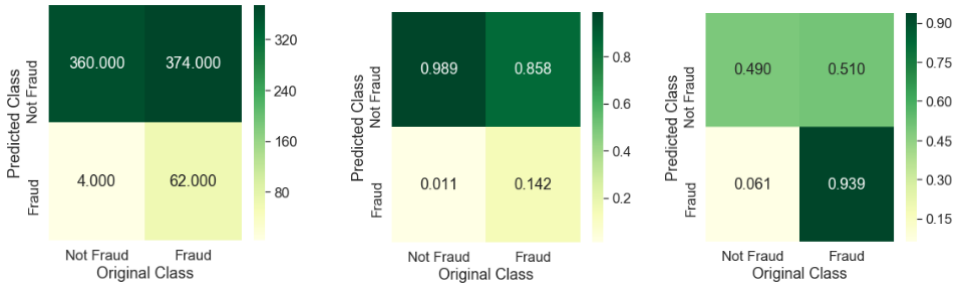
Next, SMOTE technique is applied to oversample the data and the classes are now perfectly balanced (50:50) as a result of the same.

Neural Network (without Oversampling):

a. Train-Validation ROC – AUC value

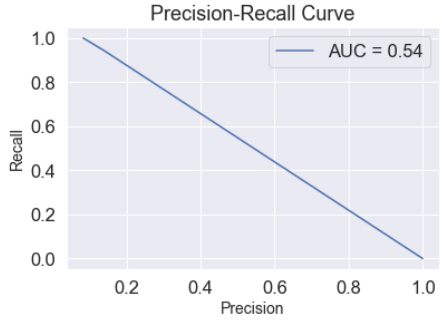


b. Confusion Matrix (Combined, Precision and Recall)



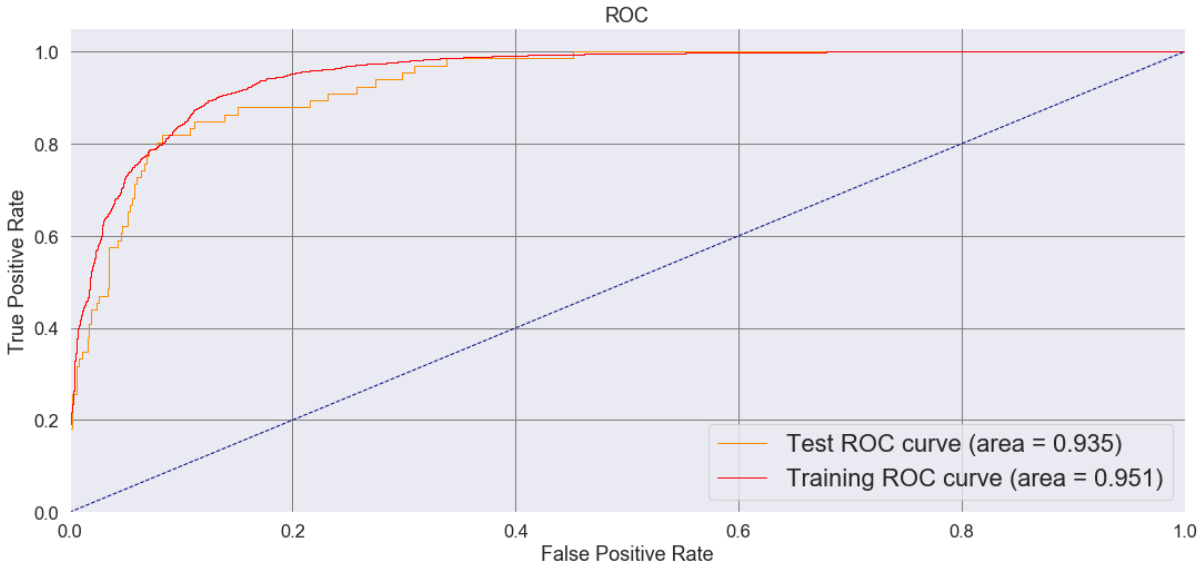
	Not Fraud	Fraud
Precision	0.989011	0.142202
Recall	0.490463	0.939394
F1-Score	0.655738	0.247012

c. Precision – Recall Curve



Logistic Regression (WITH Oversampling):

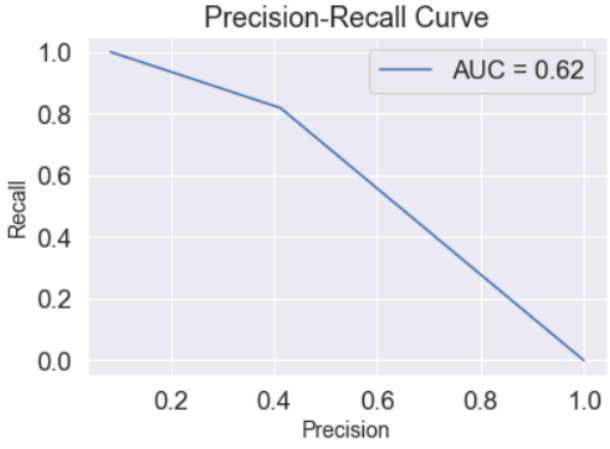
a. Train-Validation ROC – AUC value



b. Confusion Matrix (Combined, Precision and Recall)

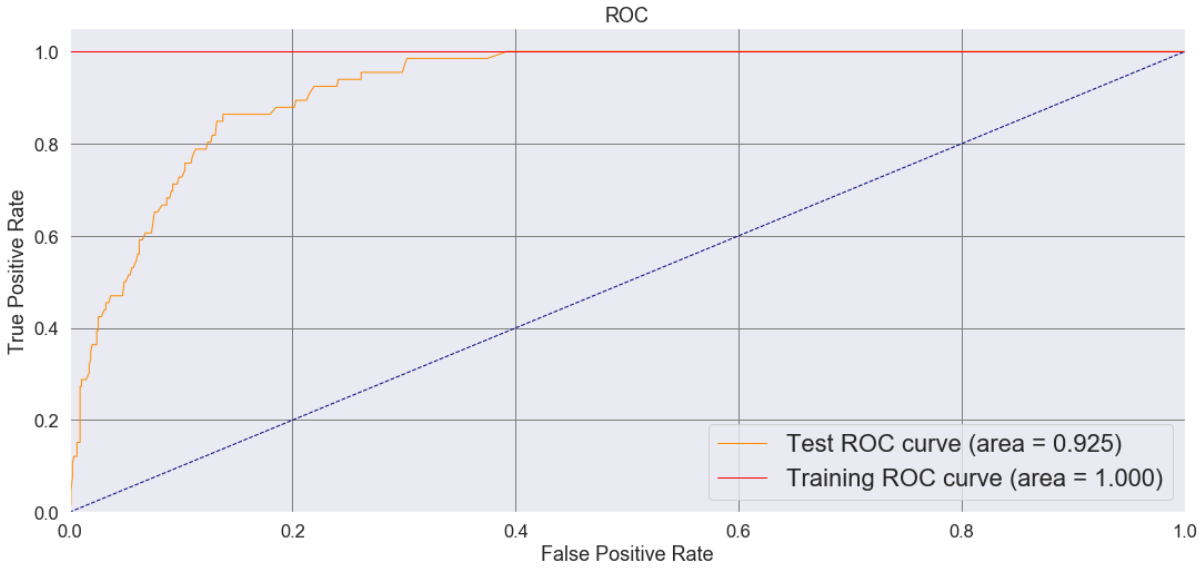
	Not Fraud	Fraud
Precision	0.982063	0.412214
Recall	0.895095	0.818182
F1-Score	0.936565	0.548223

c. Precision – Recall Curve



Random Forest (WITH Oversampling):

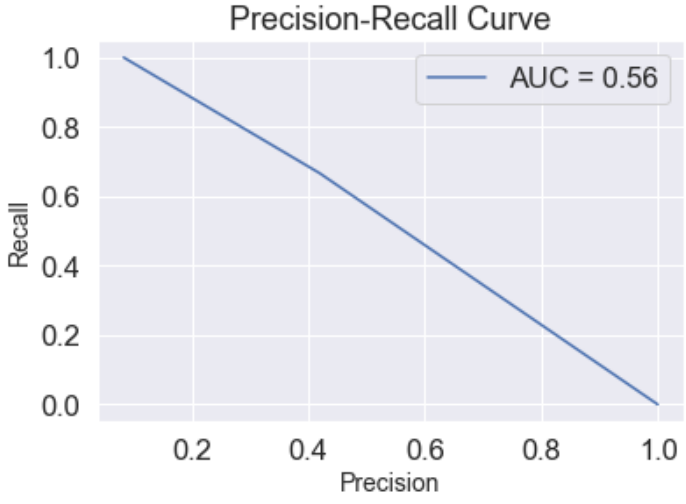
a. Train-Validation ROC – AUC value



b. Confusion Matrix (Combined, Precision and Recall)

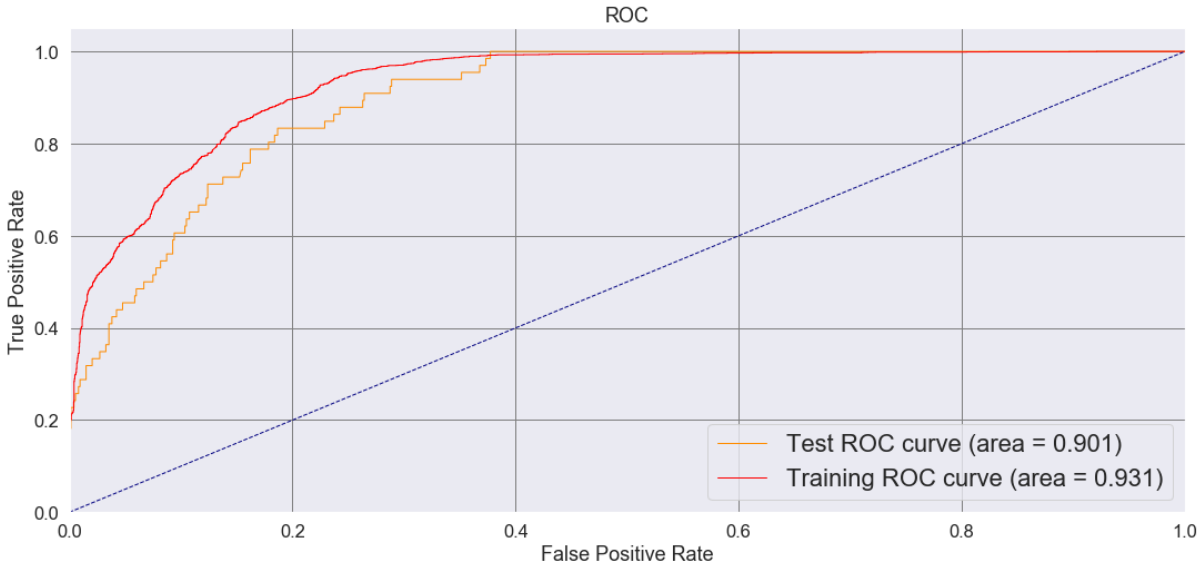
	Not Fraud	Fraud
Precision	0.968345	0.419048
Recall	0.916894	0.666667
F1-Score	0.941917	0.514620

c. Precision – Recall Curve



SVM (WITH Oversampling):

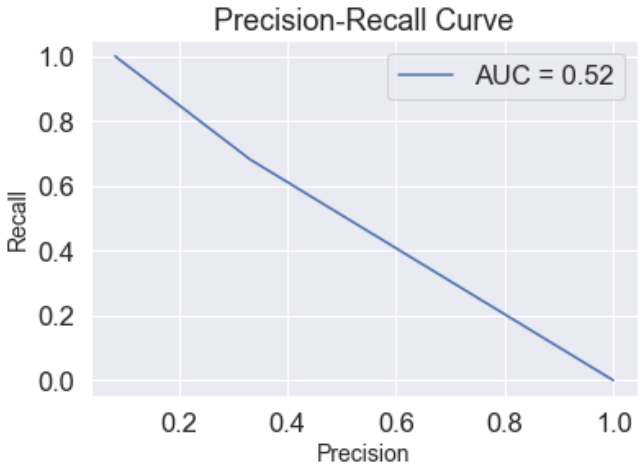
a. Train-Validation ROC – AUC value



b. Confusion Matrix (Combined, Precision and Recall)

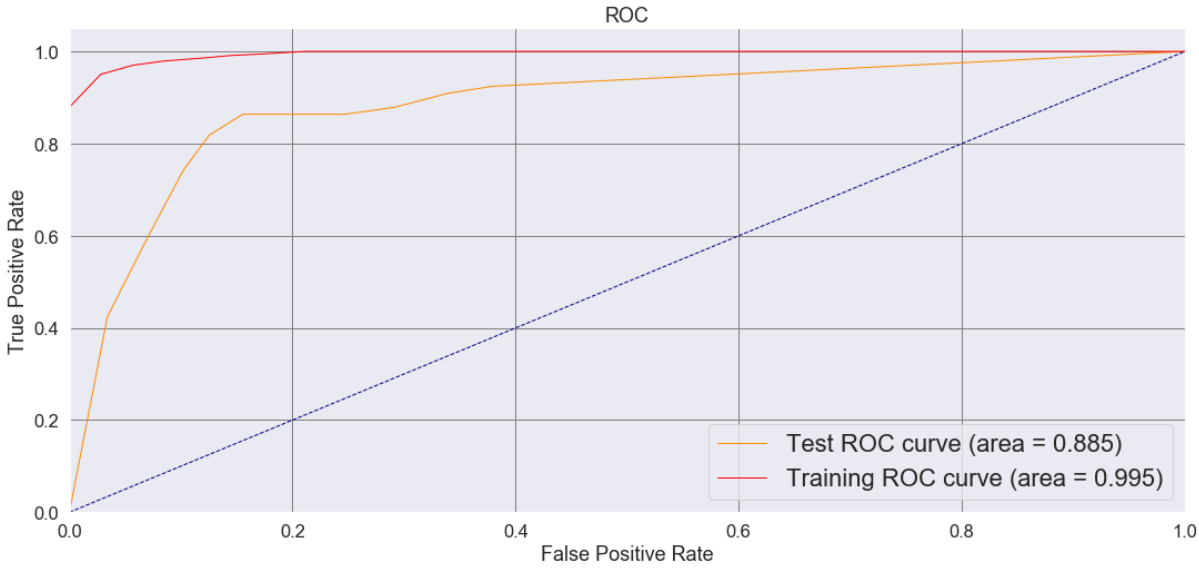
	Not Fraud	Fraud
Precision	0.968373	0.330882
Recall	0.876022	0.681818
F1-Score	0.919886	0.445545

c. Precision – Recall Curve

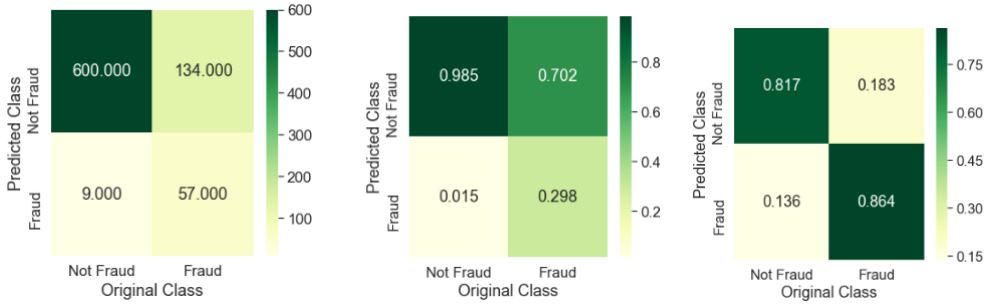


KNN (WITH Oversampling):

a. Train-Validation ROC – AUC value



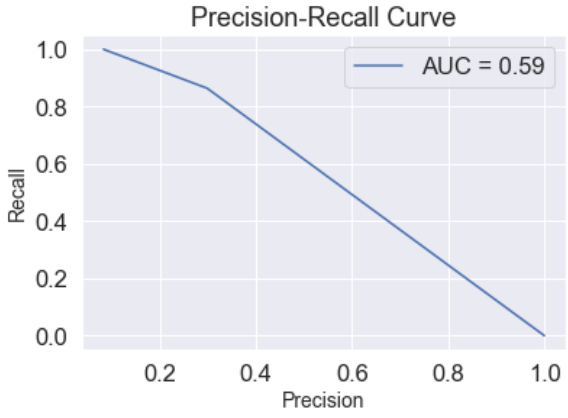
b. Confusion Matrix (Combined, Precision and Recall)



Original Class

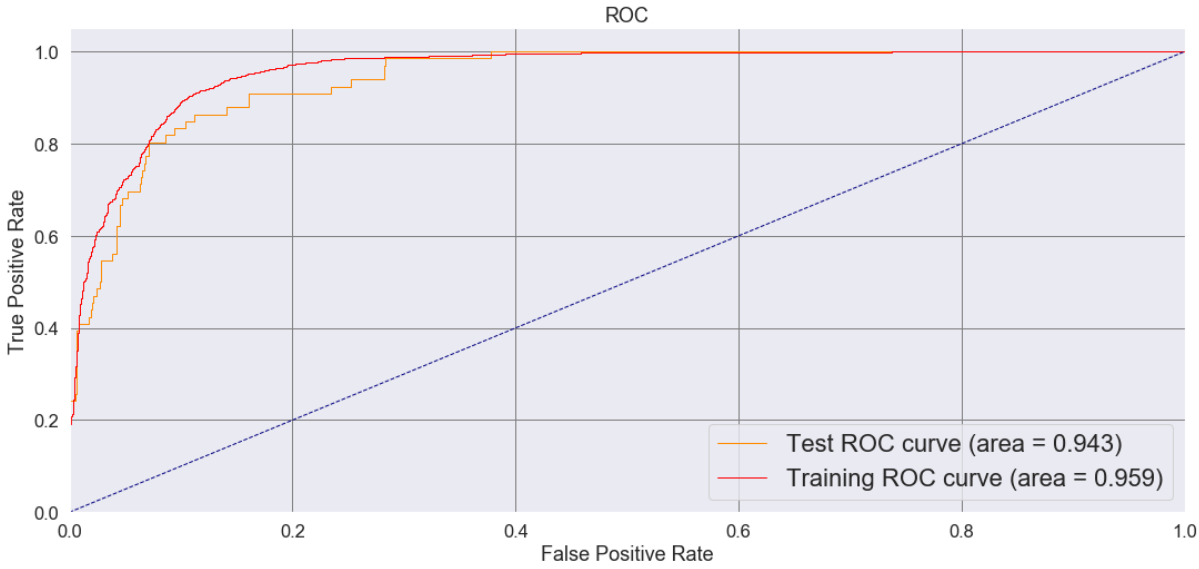
	Not Fraud	Fraud
Precision	0.985222	0.298429
Recall	0.817439	0.863636
F1-Score	0.893522	0.443580

c. Precision – Recall Curve



Neural Network (WITH Oversampling):

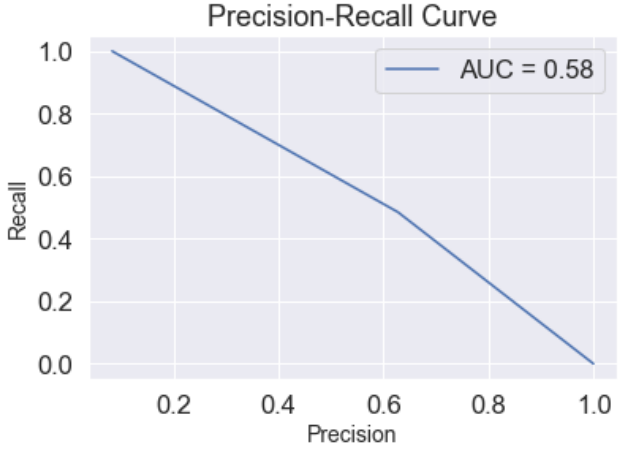
a. Train-Validation ROC – AUC value



b. Confusion Matrix (Combined, Precision and Recall)

	Not Fraud	Fraud
Precision	0.954606	0.627451
Recall	0.974114	0.484848
F1-Score	0.964262	0.547009

c. Precision – Recall Curve



Final Summary

Oversampling	Model	Test-AUC	Precision-Recall AUC
No	Logistic Regression	0.925	0.54
No	Random Forest	0.924	0.54
No	SVM	0.935	0.54
No	KNN	0.894	0.58
No	Neural Network	0.883	0.54
Yes	Logistic Regression	0.935	0.62
Yes	Random Forest	0.925	0.56
Yes	SVM	0.901	0.52
Yes	KNN	0.885	0.59
Yes	Neural Network	0.943	0.58

Conclusion →

Based on above analysis we can see that **Neural Networks (with Oversampling)** is having the best Validation set performance in terms of AUC and Precision – Recall and same will be used to predict the label (Fraud/Not Fraud) on the unlabeled Test Set. The result of the same is attached in the Excel (Test_Data.xls).