Department of Computer Science and Engineering,

Indian Institute of Technology, Bombay

FML (CS725) Project Report On

Credit Card Fraud Detection

By

Manal Jain	**(213050008)**
Anuj Namdeo Fulari	**(213050049)**
Smit Harishbhai Gajjar	**(213050051)**
Arance Kurmi	**(213050056)**
Abhishek Kumar	**(213050076)**

Under the Guidance of

Prof. Preethi Jyothi

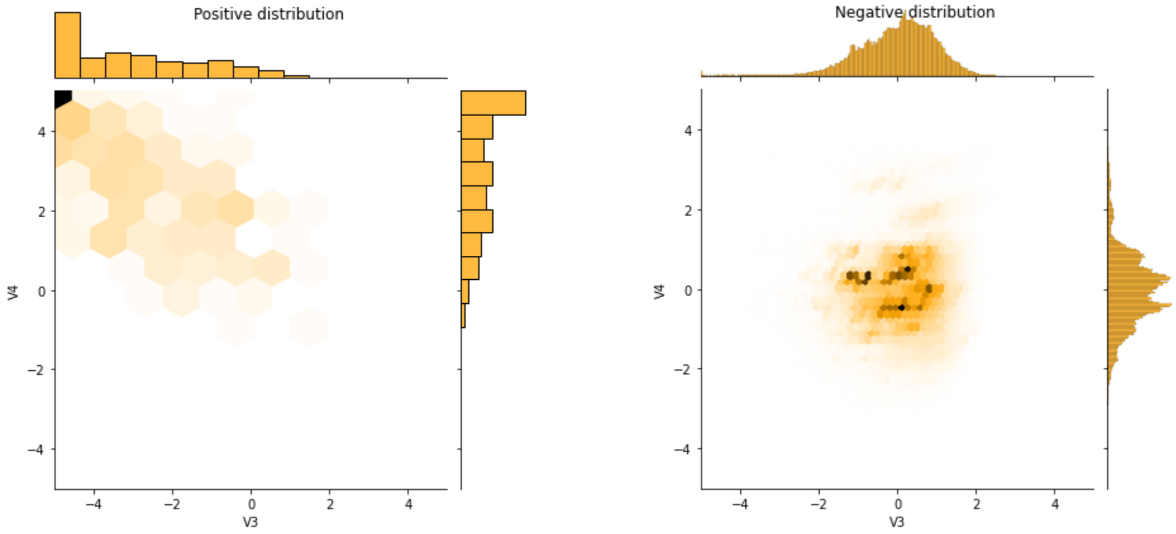
1. Problem Statement

A comparative analysis of various machine learning techniques for credit card fraud detection. But the main problem is data imbalance present in the dataset since the number of frauds is very less compared to all transactions.

2. Dataset

We have used a credit card fraud detection dataset from Kaggle. The dataset contains a total of 284,807 transactions out of which only 492 transactions (0.172%) are fraud.

3. Data Distribution



Note: These plots are plotted after data is normalized.

Above plots are joint positive and negative distributions of two features V3 and V4 from the dataset. As we can see the points corresponding to negative classes are clustered around the value 0 whereas most of the points corresponding to positive classes are taking extreme values. This is true for most of the pairs of features.

4. Result Analysis:

We have extensively tested different machine learning techniques for this problem and presented analysis on top-performing models along with the worst performing model, KMeans.

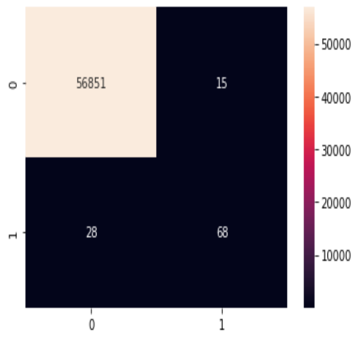
Note: All the precision, recall, and f1 score are of positive class (fraud cases).

5. Link to Github :

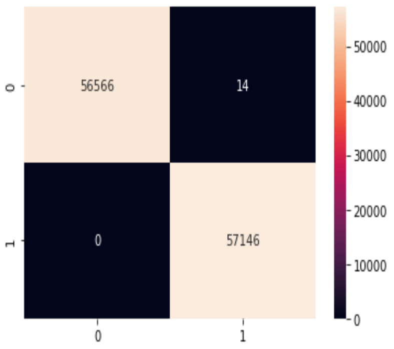
https://github.com/AnujF1005/Credit-Card-Fraud-Detection-Multiple-Techniques

A) Random Forest

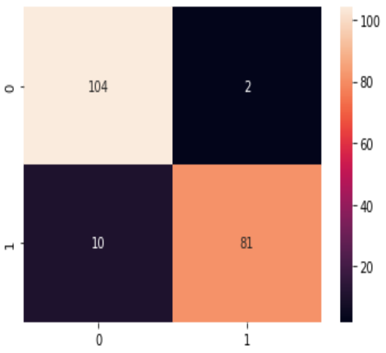
	Without Sampling	With OverSampling	With UnderSampling
Precision	0.82	1.00	0.98
Recall	0.72	1.00	0.88
F1 score	0.77	1.00	0.92



Without Sampling



With Oversampling



With Undersampling

Analysis: Random Forest gave an accuracy of 99.98% without any sampling method. Now to remove the imbalance we tried two methods: OverSampling and UnderSampling

UnderSampling: When we had done the undersampling by reducing the number of samples quite significantly, we saw a decrease in the accuracy and also in the recall value. Here Under Sampling is not a good strategy as it might have ignored some important factors.

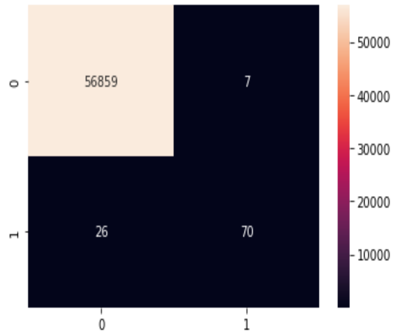
OverSampling: With OverSampling there was not much effect on accuracy, but if we see the confusion matrix, the number of false negatives was 0 and the number of false positives also decreased.

Random Forest is built on decision trees where each tree is sensitive to class imbalance. Since in oversampling we reduced the imbalance significantly that's why the accuracy got increased whereas in undersampling there was somewhat imbalance that caused the accuracy to drop.

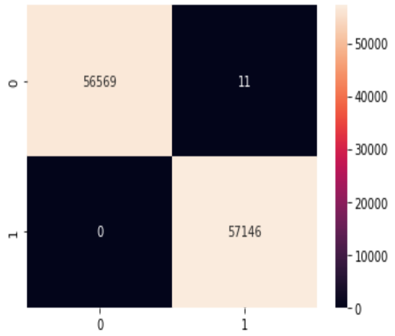
B) XGBoost

(Reference Paper: https://www.e3s-conferences.org/articles/e3sconf/pdf/2020/74/e3sconf_ebldm2020_02042.pdf)

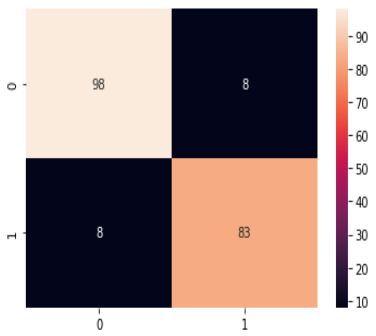
	Without Sampling	With OverSampling	With UnderSampling
Precision	0.91	1.00	0.91
Recall	0.73	1.00	0.91
F1 score	0.81	1.00	0.91



Without Sampling



With Oversampling

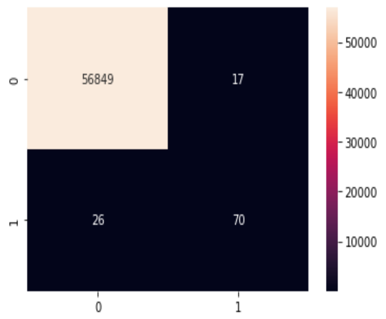


With Undersampling

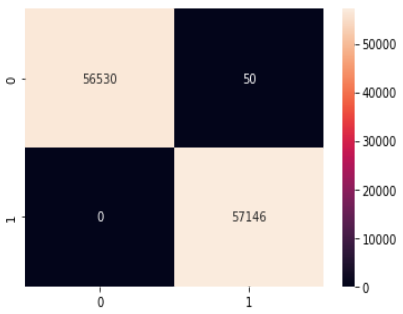
Analysis: Xgboost is a hybrid technique which is based on gradient boosting with Newton Raphson method and is a type of ensemble learning technique, which uses boosted trees. This technique performed one of the best out of all the techniques since it works independent of the distribution of data and its dimensionality. Moreover, after selecting a subset of data points, the tree is traversed depth first for pruning so that only important features are explored prior to the other features. We tried to set the parameters used in the paper and got best results among all other values of parameters experimented.

C) K-Nearest Neighbours

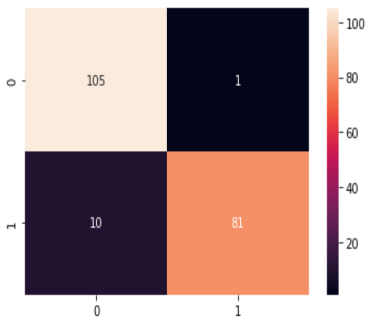
	Without Sampling	With OverSampling	With UnderSampling
Precision	0.80	1.00	0.99
Recall	0.73	1.00	0.89
F1 score	0.77	1.00	0.94



Without Sampling



With Oversampling

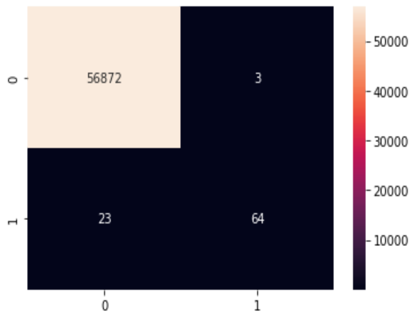


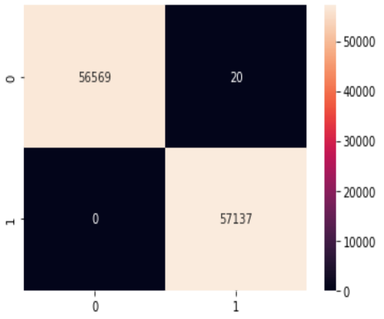
With Undersampling

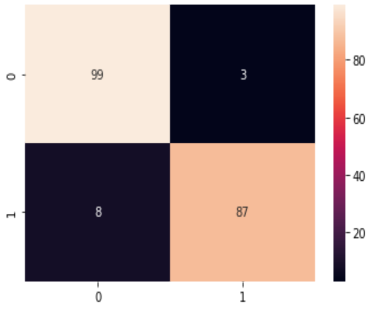
Analysis: KNN checks k nearest neighbours to determine in which class the given point will belong to. Here, since the data distribution is skewed towards 0 class, it will find majority voting of the classes of k nearest neighbours and will most likely predict 0 more than 1. So, it didn't perform well in the data without sampling. After oversampling and undersampling, k nearest neighbours for each data point will be somehow evenly distributed and it will pick the correct majority class. So, it performed much better in those cases.

D) Adaboost

	Without Sampling	With OverSampling	With UnderSampling
Precision	0.96	1.00	0.97
Recall	0.74	1.00	0.92
F1 score	0.83	1.00	0.94







Without Sampling

With Oversampling

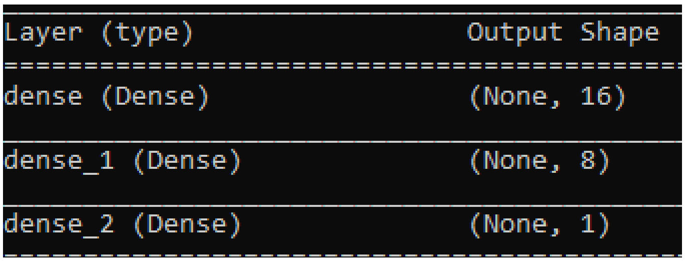
With Undersampling

Note: The base estimator used for Adaboost is a Decision Tree Classifier with a max-depth of 2. Initially, Adaboost is trained using Decision Tree Classifier of max-depth of 1 but it fails to give good results since it doesn't incur much non-linearity.

Analysis: Adaboost is a boosting algorithm that is sensitive to outliers. And as we see in the data distribution, most of the fraud classes are taking extreme values (outliers). Due to this AdaBoost gave a good result mainly after doing oversampling.

E) Neural Network

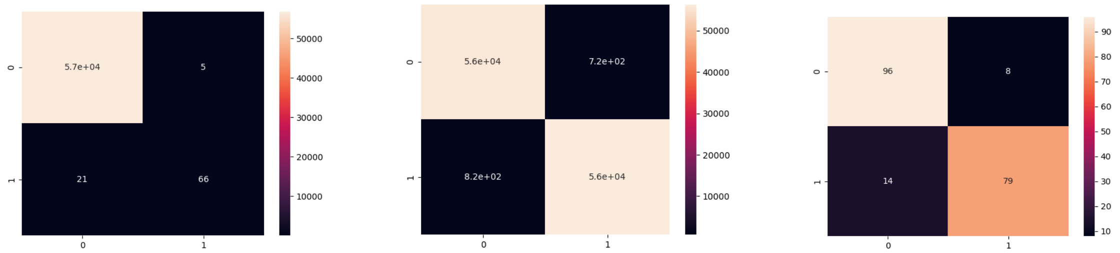
Architecture:



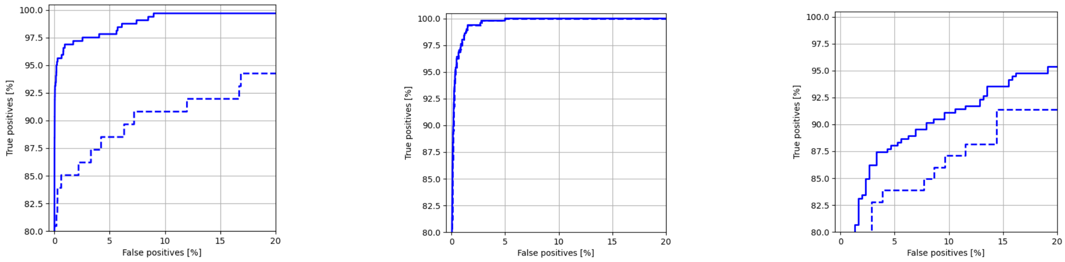
```
Layer (type)                 Output Shape
=========================================
dense (Dense)                (None, 16)

dense_1 (Dense)              (None, 8)

dense_2 (Dense)              (None, 1)
```

Results:

	Without Sampling	With OverSampling	With UnderSampling
Precision	0.93	0.99	0.91
Recall	0.76	0.99	0.85
F1 score	0.84	0.99	0.88



ROC Curve



Dashed line: ROC for validation data
Continuous line: ROC for train data

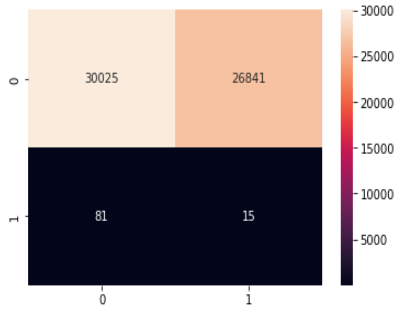
Without Sampling **With Oversampling** **With Undersampling**

Note: Neural Network is trained using early stopping monitoring validation data precision

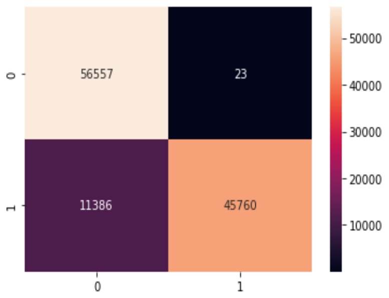
Analysis: Without any sampling, recall is less compared to precision since data is imbalanced and not enough data corresponding to positive class in the dataset. As seen in the ROC curve, it is evident that the model seems to be overfitting training data since the curve for train data varies from the curve for validation data. With oversampling, a number of records belonging to positive and negative classes become equal and Neural Network gave better precision and recall for the positive class. Also, ROC shows well overlap for validation and train data, which means the model is neither overfitted nor underfitted. With undersampling, performance dropped because the amount of data decreased and from ROC, the model seems to overfit the training data.

F) **KMeans:** (Bad performance compared to other models)

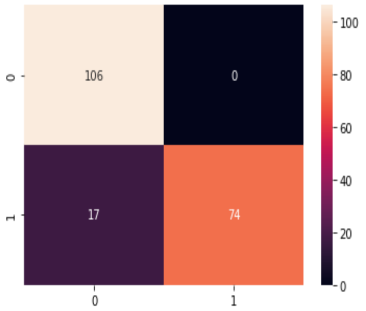
	Without Sampling	With OverSampling	With UnderSampling
Precision	0.00	1.00	1.00
Recall	0.16	0.80	0.81
F1 score	0.00	0.89	0.90



Without Sampling



With Oversampling



With Undersampling

Analysis: KMeans performed worst in all cases as expected. KMeans does not care about the high dimensionality of data and simply forms clusters around the entire data distribution.

5. Conclusion

We presented an evaluation of different Machine Learning algorithms on highly imbalanced data of Credit card frauds. Oversampling helps in training the Machine Learning models better on such imbalanced data. From evaluation, we find out that ensemble learning methods are giving good performance on such data. Also, neural networks are performing good with the hidden layers more than two. KMeans failed to show good results because KMeans does not take into consideration the relationship between features and classes or the dimensionality of features and simply tries to form clusters around the closest data points.