PRML Minor Project on Credit Card Fraud detection

Tarun Raj Singh (B21CS076) and Aryan Himmatlal Prajapati (B21EE012)

17 March 2023

1 Introduction

This Project Problem is associated with the Credit risk of a clienty failing to meet contractual obligations.The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions. In this project we have implemented a complete machine learning pipeline which at last predicts whether , given the details about a credit card , it is real or fake .

2 Highly Imbalanced data :

We find that the class 1 (fraud) is 577.876 times lesser than the negative class.

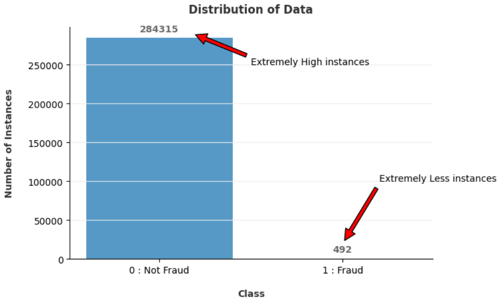


Figure 1: Bar graph showing the Highly imbalance nature of data set

Hence, any model trained on this imbalanced data is highly prone to be biased towards the negative class , hence we have to reduce the false negative predictions from the model using various imbalance handling methods.

3 If applying Ensemble methods on this imbalanced data :

We Tried to use AdaBoost , XGBoost and LGBM Models on this Imbalanced data and found that the Models are actually biased towards the negative class. This is clear from the fact that they are having high accuracy , precission for the negative class , but the overall f1 score for the positive class is very low .

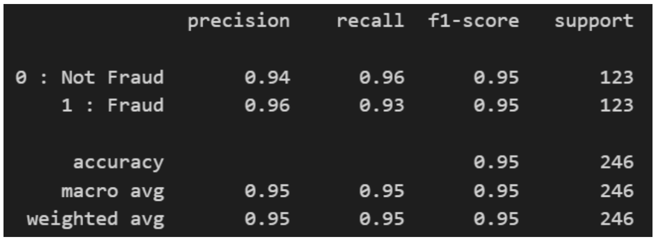
- For Adaboost : f1 = 0.73 , (Misclassify 39 (out of 123) Fraud Instances)
- For XGBoost : f1 = 0.85 , (Misclassify 29 (out of 123) Fraud Instances)
- For LGBM : f1 = 0.86 , (Misclassify 27 (out of 123) Fraud Instances)

But in all these model we are getting an accuracy close to 1 . This is due to the biased nature of the classifier towards the negative class and instances of negative class are far far more in number, hence the accuracy is more , but it is not a good model as it can not separate out fraud instances correctly . So we need to apply some imbalance handling methods and then train the classifiers .

4 Applying Imbalance handling methods using Adaboost :

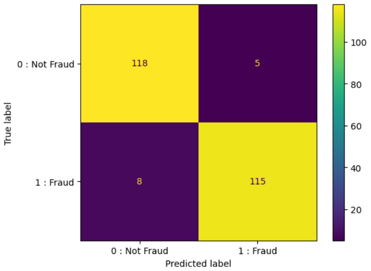
4.1 Using UnderSampling :

UnderSampling is a technique in Data Analysis where the size of the majority class is reduced to balance the class distribution. Henc, we randomly sampled 492 instances from the negative class and added them with the 492 instances of Positive class to create a balanced dataset and then applied the AdaBoost classifier.



```
                 precision    recall  f1-score   support

0 : Not Fraud         0.94      0.96      0.95       123
    1 : Fraud         0.96      0.93      0.95       123

     accuracy                             0.95       246
    macro avg         0.95      0.95      0.95       246
 weighted avg         0.95      0.95      0.95       246
```

(a) Classification report

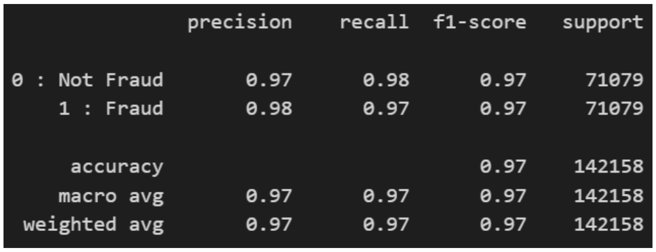


(b) Confusion Matrix

Figure 2: Results of AdaBoost Using Under Sampling

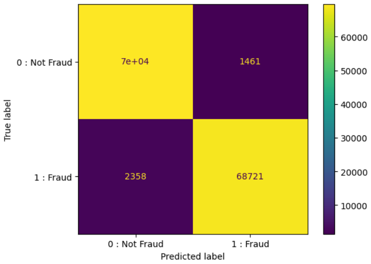
4.2 Using OverSampling :

OverSampling is a technique in Data Analysis where the size of the minority class is increased to balance the class distribution. Hence, we randomly sampled 284315 instances from the positive class with replacement and added them with the 284315 instances of negative class to create a balanced dataset and then applied the AdaBoost classifier.



```
                 precision    recall  f1-score   support

0 : Not Fraud         0.97      0.98      0.97     71079
    1 : Fraud         0.98      0.97      0.97     71079

     accuracy                             0.97    142158
    macro avg         0.97      0.97      0.97    142158
 weighted avg         0.97      0.97      0.97    142158
```

(a) Classification report

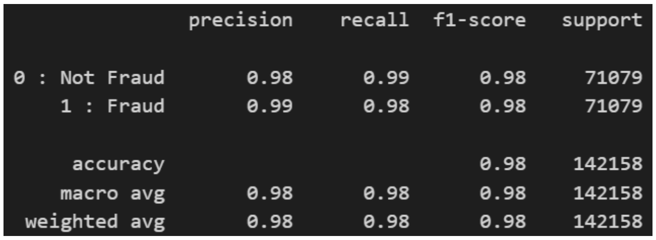


(b) Confusion Matrix

Figure 3: Results of AdaBoost Using Over Sampling

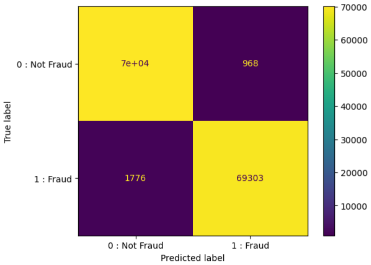
4.3 Using SMOTE (Synthetic Minority Over Sampling Technique) :

SMOTE (Synthetic Minority Over Sampling Technique) is a method for generating synthetic samples to balance class distribution in imbalanced datasets . Hence, we synthetically sampled 284315 instances from the positive class and added them with the 284315 instances of negative class to create a balanced dataset and then applied the AdaBoost classifier.



```
                 precision    recall  f1-score   support

0 : Not Fraud         0.98      0.99      0.98     71079
    1 : Fraud         0.99      0.98      0.98     71079

     accuracy                             0.98    142158
    macro avg         0.98      0.98      0.98    142158
 weighted avg         0.98      0.98      0.98    142158
```

(a) Classification report



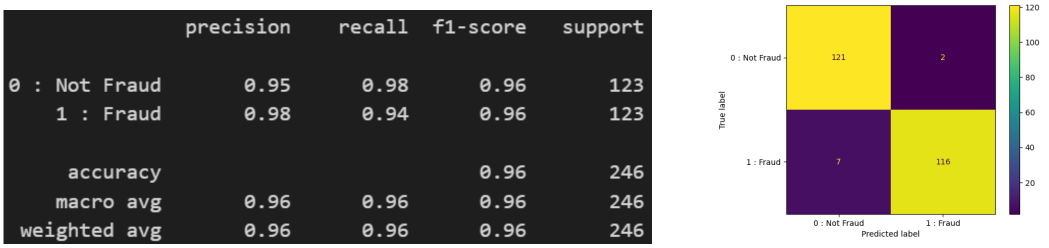
(b) Confusion Matrix

Figure 4: Results of AdaBoost Using SMOTE Sampling

5 Applying Imbalance handling methods using XGBoost :

5.1 Using UnderSampling :

Applied the XGBoost Classifier on the previouly generated undersampled balanced dataset.



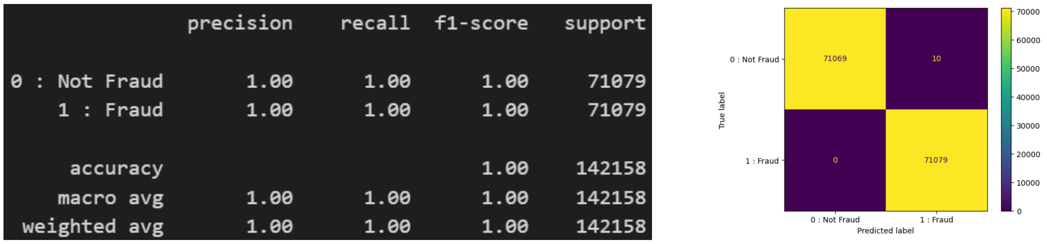
```
                precision    recall  f1-score   support

0 : Not Fraud        0.95      0.98      0.96       123
    1 : Fraud        0.98      0.94      0.96       123

     accuracy                           0.96       246
    macro avg        0.96      0.96      0.96       246
 weighted avg        0.96      0.96      0.96       246
```

(a) Classification report (b) Confusion Matrix

Figure 5: Results of XGBoost Using Under Sampling

5.2 Using OverSampling :

Applied the XGBoost Classifier on the previouly generated oversampled balanced dataset.



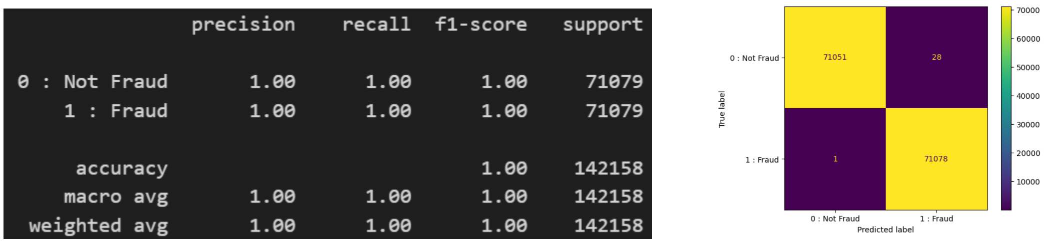
```
                precision    recall  f1-score   support

0 : Not Fraud        1.00      1.00      1.00     71079
    1 : Fraud        1.00      1.00      1.00     71079

     accuracy                           1.00    142158
    macro avg        1.00      1.00      1.00    142158
 weighted avg        1.00      1.00      1.00    142158
```

(a) Classification report (b) Confusion Matrix

Figure 6: Results of XGBoost Using Over Sampling

5.3 Using SMOTE (Synthetic Minority Over Sampling Technique) :

Applied the XGBoost Classifier on the previouly generated Synthetically sampled balanced dataset using SMOTE .



```
                precision    recall  f1-score   support

0 : Not Fraud        1.00      1.00      1.00     71079
    1 : Fraud        1.00      1.00      1.00     71079

     accuracy                           1.00    142158
    macro avg        1.00      1.00      1.00    142158
 weighted avg        1.00      1.00      1.00    142158
```

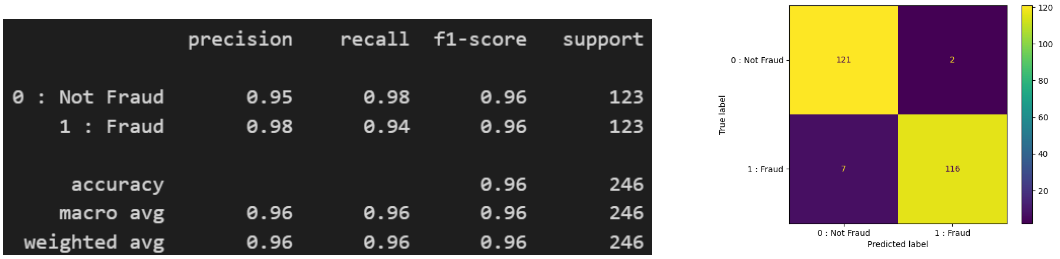
(a) Classification report (b) Confusion Matrix

Figure 7: Results of XGBoost Using SMOTE Sampling

6 Applying Imbalance handling methods using LGBM :

6.1 Using UnderSampling :

Applied the LGBM Classifier on the previouly generated undersampled balanced dataset.



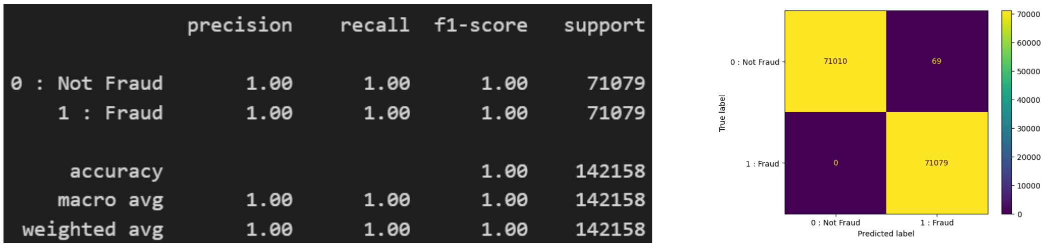
```
                precision    recall  f1-score   support

0 : Not Fraud        0.95      0.98      0.96       123
    1 : Fraud        0.98      0.94      0.96       123

     accuracy                           0.96       246
    macro avg        0.96      0.96      0.96       246
 weighted avg        0.96      0.96      0.96       246
```

(a) Classification report (b) Confusion Matrix

Figure 8: Results of LGBM Using Under Sampling

6.2 Using OverSampling :

Applied the LGBM Classifier on the previouly generated oversampled balanced dataset.



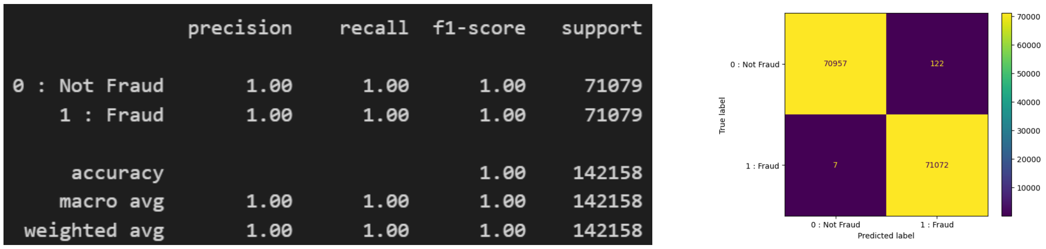
```
                 precision    recall  f1-score   support

0 : Not Fraud         1.00      1.00      1.00     71079
    1 : Fraud         1.00      1.00      1.00     71079

     accuracy                           1.00    142158
    macro avg         1.00      1.00      1.00    142158
 weighted avg         1.00      1.00      1.00    142158
```

(a) Classification report (b) Confusion Matrix

Figure 9: Results of LGBM Using Over Sampling

6.3 Using SMOTE (Synthetic Minority Over Sampling Technique) :

Applied the LGBM Classifier on the previouly generated Synthetically sampled balanced dataset using SMOTE .

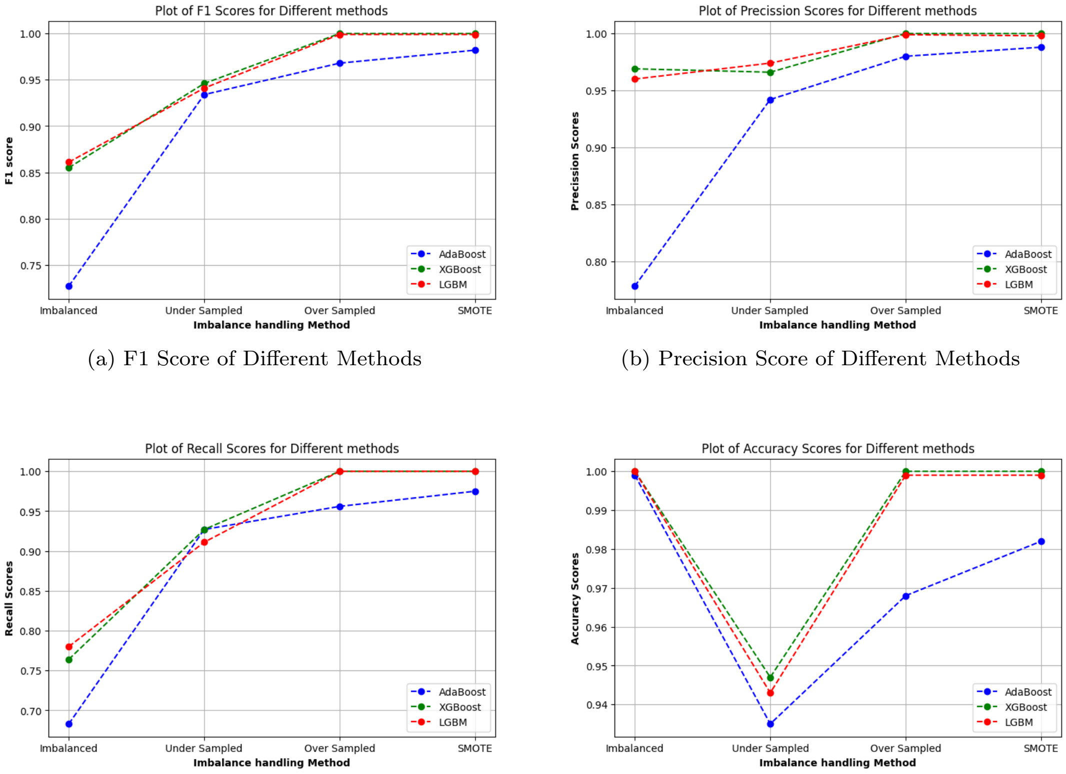


```
                 precision    recall  f1-score   support

0 : Not Fraud         1.00      1.00      1.00     71079
    1 : Fraud         1.00      1.00      1.00     71079

     accuracy                           1.00    142158
    macro avg         1.00      1.00      1.00    142158
 weighted avg         1.00      1.00      1.00    142158
```

(a) Classification report (b) Confusion Matrix

Figure 10: Results of LGBM Using SMOTE Sampling

7 Plots for Performance on different model selections and Different Balancing Methods :



(a) F1 Score of Different Methods (b) Precision Score of Different Methods

(a) Recall Scores of Different Methods (b) Accuracy Scores of Different Methods

Figure 11: Performances of all models and different Balancing methods.

8 Picking out the best model :

We used all these boosting algorithms in order to reduce bias and used sampling methods to reduce the imbalance in the dataset and finally tried different configurations and observed the confusion matrix. There were 3 models where the accuracy score for both classes was close to 1 , but we checked all the metrics and chose that model which had maximum scores for most of the metrics :

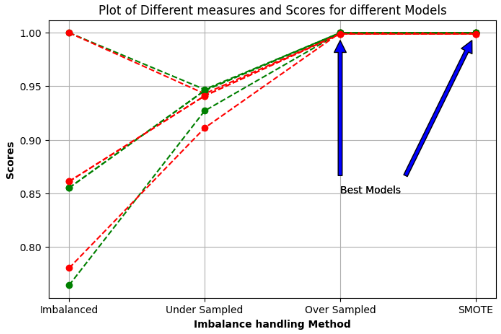
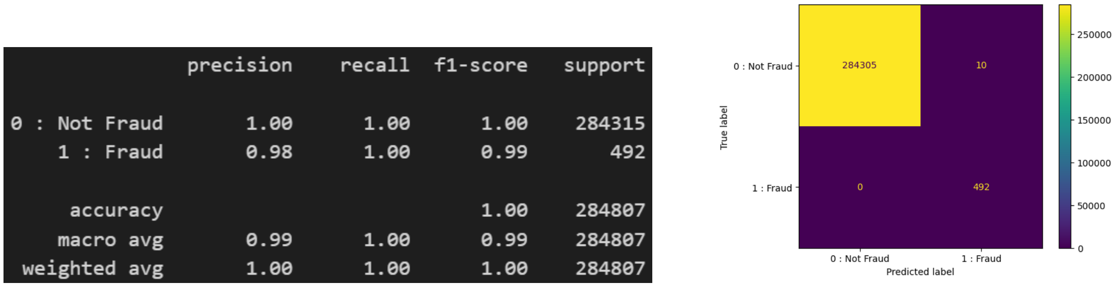


Figure 12: Plot for all metrics and all models in order to observe the best model

Chose the best model as Oversampling model using XGBoost and finally tested the complete dataset and got the results :



```
               precision    recall  f1-score   support

0 : Not Fraud       1.00      1.00      1.00    284315
    1 : Fraud       0.98      1.00      0.99       492

     accuracy                           1.00    284807
    macro avg       0.99      1.00      0.99    284807
 weighted avg       1.00      1.00      1.00    284807
```

(a) Classification report

(b) Confusion Matrix

Figure 13: Results for the best model on the complete dataset original

9 Conclusion :

We saw that our best model classifies 492 Fraud instances as Fraud correctly (492/492) , accuracy = 1. On the same hand it is able to classify 284305 Non fraud Instances correctly out of 284315 Non Fraud Instances.

Hence , None of the Fraud instances is misclassified and only 10 out of 284315 Non fraud instances are classified as fraud.
Hence our model works well for both classes and is not biased towards any class , Hence using the above mentioned Balancing techniques and Classifiers we handled the bias and imbalance in the data set and got a classifier trained which can handle both classes efficiently and without missing fraud instances .

Thank You