Project: Credit Card Fraud Detection
CS725

Prepared by
Ashish Aggarwal (203050015)
Praneeth Reddy (203050019)
Debabrata Biswal (203050024)
Velugoti venkata Sai Baba Reddy (203050025)

Tuesday 8th December, 2020

Contents

1 Dataset Description

Dataset is available in Kaggle, click here to go to the download page.

The datasets contains transactions made by credit cards in September 2013 by European cardholders.

This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

It contains only numerical input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, original features and more background information about the data were not provided. Features V1, V2, ... V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount. Feature 'Class' is the response variable, and it takes value 1 in case of fraud and 0 otherwise.

2 Techniques Used

- Random forest classifier
- Logistic regression classifier
- Support vector machine classifier
- Fully connected neural network

3 Imbalance in Dataset

The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

If this highly imbalanced dataset is used as it is, the predictive models may overfit and will not classify fraud transactions accurately. The below figure clearly shows the imbalance in the class distribution.

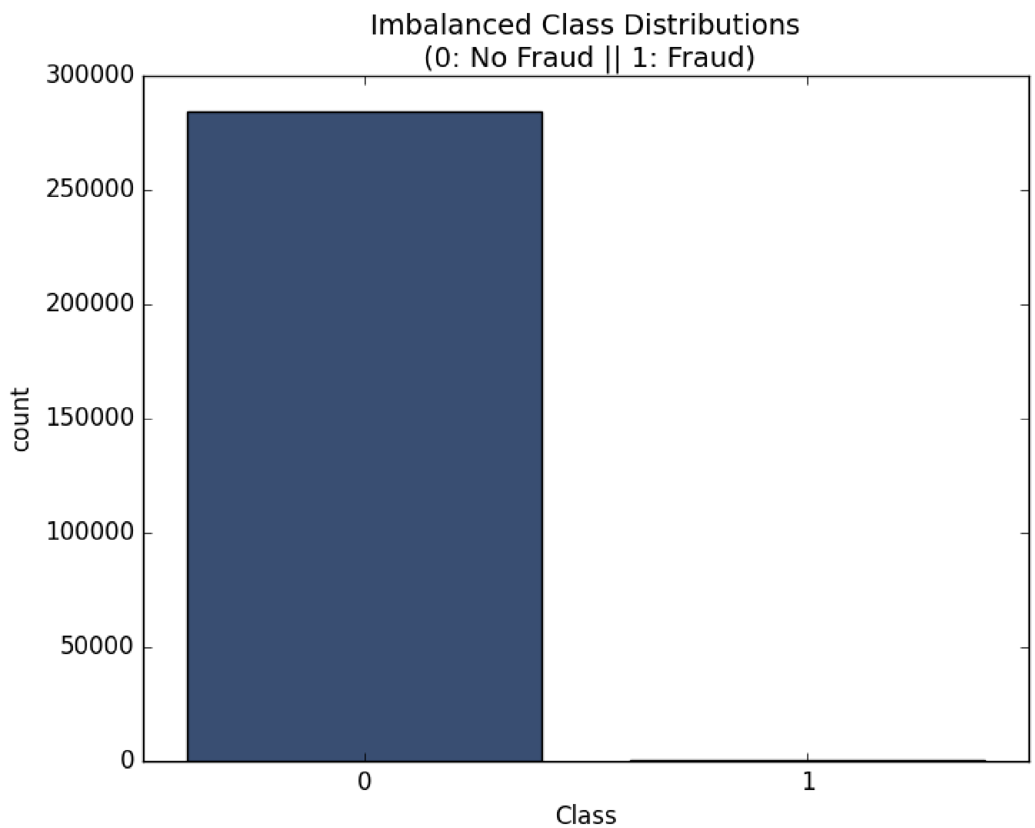

Figure 1: Imbalance in class distribution.

4 Tackling Imbalance in Dataset

4.1 Under Sampling

To overcome the issue of imbalance in the class feature, under sampling can be used. In this method, the 50/50 ratio of fraud and non-fraud transactions is achieved by randomly sampling fraud number of non-fraud transaction instances from the dataset.

Figure 2: Class distribution after under sampling.

4.2 Over Sampling with SMOTE

Similarly, over sampling can also be used. In this method, the 50/50 ratio of fraud and non-fraud transactions is achieved by synthesizing minority fraud transactions.To implement SMOTE we have used imblearn package.

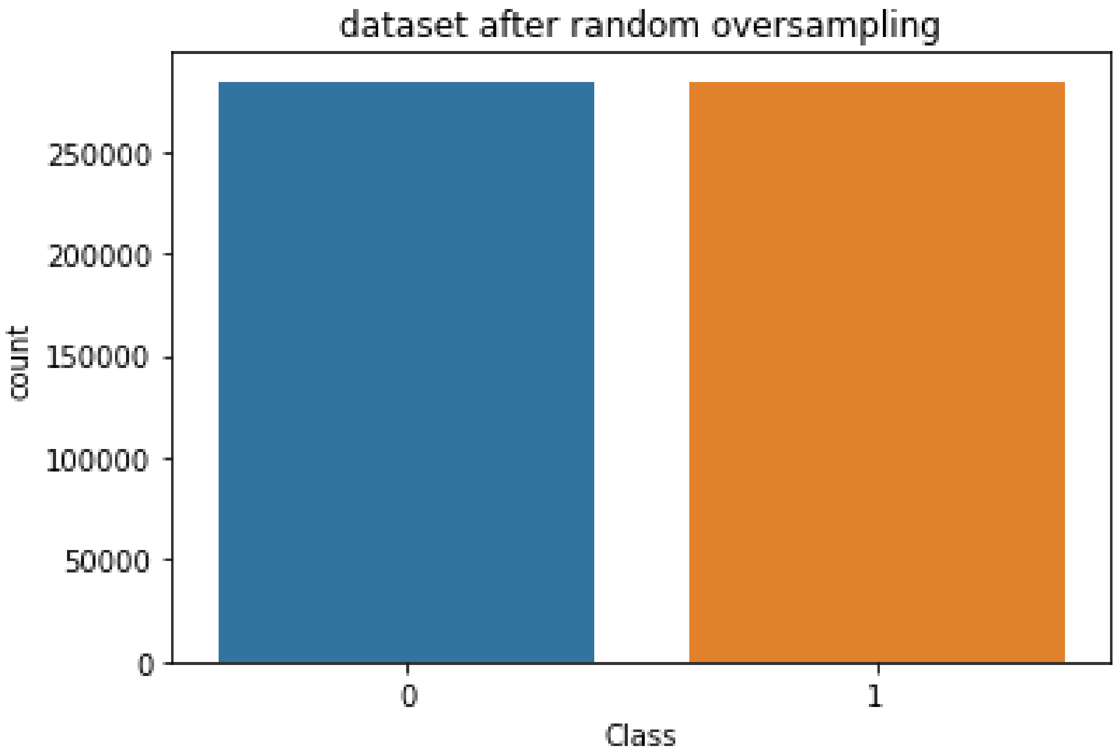

Figure 3: Class distribution after over sampling.

5 Results

Table 1: Models performance without balancing

Classifier	Performance without balancing			
	Accuracy	**Precision**	**Recall**	**F1 Score**
Logistic Regression	0.99	0.73	0.54	0.62
NeuralNetwork	0.99	0.41	0.78	0.54
SVM	0.99	0	0	0
Random Forest	0.99	0.92	0.77	0.84

Table 2: Models performance after under sampling

Classifier	After Under Sampling			
	Accuracy	**Precision**	**Recall**	**F1 Score**
Logistic Regression	0.97	0.06	0.91	0.11
NeuralNetwork	0.003	0.002	1	0.003
SVM	0.82	0	0.23	00
Random Forest	0.98	0.06	0.93	0.12

Table 3: Models performance after over sampling

Classifier	After Over Sampling			
	Accuracy	**Precision**	**Recall**	**F1 Score**
Logistic Regression	0.98	0.09	0.88	0.16
NeuralNetwork	0.96	0.04	0.93	0.07
SVM	-	-	-	-
Random Forest	0.99	0.82	0.84	0.83

6 Confusion Matrices

6.1 Without balancing

Table 4: Logistic Regression

Actual	**Predicted**	
	Non-Fraud	**Fraud**
Non-Fraud	71067	22
Fraud	51	62

Table 5: NeuralNetwork

Actual	**Predicted**	
	Non-Fraud	**Fraud**
Non-Fraud	71037	52
Fraud	35	78

Table 6: SVM

Actual	**Predicted**	
	Non-Fraud	**Fraud**
Non-Fraud	71089	0
Fraud	113	0

Table 7: Random Forest

Actual	**Predicted**	
	Non-Fraud	**Fraud**
Non-Fraud	71082	7
Fraud	26	87

6.2 After Under Sampling

Table 8: Logistic Regression

Actual	Predicted	
	Non-Fraud	Fraud
Non-Fraud	69882	1207
Fraud	12	101

Table 9: NeuralNetwork

Actual	Predicted	
	Non-Fraud	Fraud
Non-Fraud	82	71007
Fraud	0	113

Table 10: SVM

Actual	Predicted	
	Non-Fraud	Fraud
Non-Fraud	55197	15892
Fraud	87	26

Table 11: Random Forest

Actual	Predicted	
	Non-Fraud	Fraud
Non-Fraud	69491	1598
Fraud	8	105

6.3 After Over Sampling

Table 12: Logistic Regression

Actual	Predicted	
	Non-Fraud	Fraud
Non-Fraud	70052	1037
Fraud	13	100

Table 13: NeuralNetwork

Actual	Predicted	
	Non-Fraud	Fraud
Non-Fraud	70563	526
Fraud	13	100

Table 14: Random Forest

Actual	Predicted	
	Non-Fraud	Fraud
Non-Fraud	71068	21
Fraud	18	95

7 Conclusion

As we can see the Random Forest seems to be performing better in every aspect(Both recall and Precision).Also the accuracy is not a really a good measure here.(As a model which always output class 0 will give 99.8 % accuracy.)So we can choose appropriate model according to our need.(i.e - high precision or high recall).But anyways Random forest clearly comes out as the winner here.

8 References

https://towardsdatascience.com/methods-for-dealing-with-imbalanced-data-5b761be45a18
https://www.kaggle.com/janiobachmann/credit-fraud-dealing-with-imbalanced-datasets

9 Link to the codes

https://git.cse.iitb.ac.in/ashishaggarwal/CreditCardFraudDetection/tree/master/project