Credit Card Fraud Detection

Iheanyi Achareke

3/7/2020

Contents

1 Overview

1.1 Introduction

Credit card fraud is an inclusive term for fraud committed using a payment card, such as a credit card or debit card. According to the federal reserve, credit cards accounted for 21% of transactions in 2015 while debit cards accounted for 27% of transactions in the united states. About $190 billion is lost annually to credit card fraud in the United States. With the growth of card based online shopping and the prevalence of card payments at brick and mortar stores, it has become important that credit card companies are able to recognize and stop fraudulent card transactions.

This project seeks to apply data science techniques to this problem as part of the capstone project of the HarvardX Data Science profesional certificate.

1.2 Problem Definition

The problem to be addressed in this project is to predict fraudulent credit card transactions by using machine learning models. The models will be trained using supervised learning techniques.

1.3 Data Preparation and environment preparation

1.3.1 Environment Preparation

Our project environment is Microsoft R open R 3.5.3 with the following packages loaded from cran:

- tidyverse
- caret
- data.frame
- lubridate
- scales
- reshape2
- ROSE
- smotefamily
- rpart
- rpart.plot
- cowplot
- randomForest
- ROCR
- neuralnet

1.3.2 Data Preparation

This project is based on the Credit Card Fraud Detection Data set available at https://www.kaggle.com/mlg-ulb/creditcardfraud.

The data sets contains transactions made by credit cards in September 2013 by European cardholders. This data set presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The data set 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, we cannot provide the original features and more background information about the data. 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 data set. The feature 'Amount' is the transaction Amount, this feature can be used for example-dependent cost-sensitive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

In order to test our prediction models without over-fitting, the credit card data set will be split into a training set *cc_train* and a test set *cc_test*. We used a 90/10 split between the training and test set.

We shall use stratified sampling which means that the probability of seeing a fraudulent transaction will be approximately the same in both the training data and the test data. Stratified sampling also ensures that our model metrics are as close as possible to what we'd see in a whole population.

2 Method and Analysis

2.1 Data Exploration

To gain a preliminary understanding of the structure of the data set, we display the first few rows of the *credit_card* data set

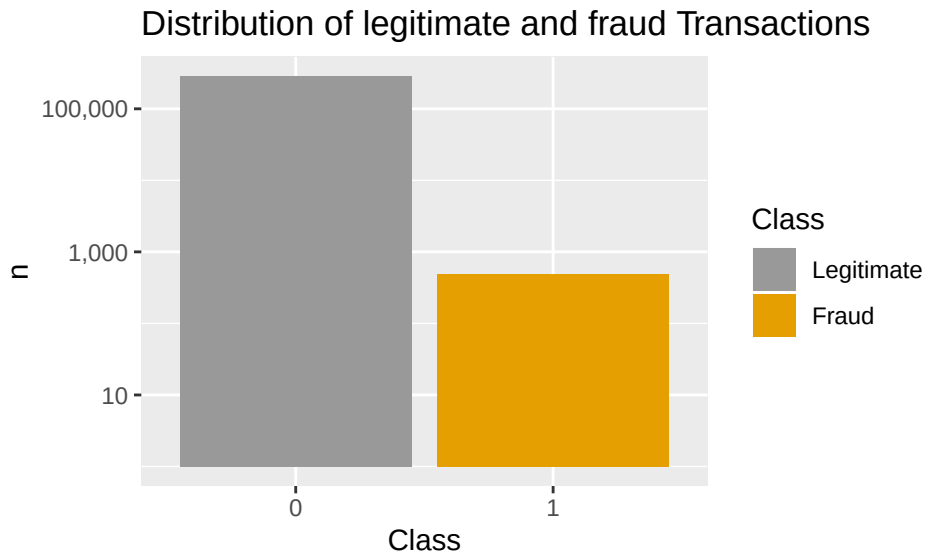
Time	V1	V2	V3	V4	V27	V28	Amount	Class
0	-1.3598071	-0.0727812	2.5363467	1.3781552	0.1335584	-0.0210531	149.62	0
0	1.1918571	0.2661507	0.1664801	0.4481541	-0.0089831	0.0147242	2.69	0
1	-1.3583541	-1.3401631	1.7732093	0.3797796	-0.0553528	-0.0597518	378.66	0
1	-0.9662717	-0.1852260	1.7929933	-0.8632913	0.0627228	0.0614576	123.50	0
2	-1.1582331	0.8777368	1.5487178	0.4030339	0.2194222	0.2151531	69.99	0
2	-0.4259659	0.9605230	1.1411093	-0.1682521	0.2538442	0.0810803	3.67	0
4	1.2296576	0.1410035	0.0453708	1.2026127	0.0345074	0.0051678	4.99	0
7	-0.6442694	1.4179635	1.0743804	-0.4921990	-1.2069211	-1.0853392	40.80	0
7	-0.8942861	0.2861572	-0.1131922	-0.2715261	0.0117474	0.1424043	93.20	0
9	-0.3382618	1.1195934	1.0443666	-0.2221873	0.2462193	0.0830756	3.68	0

The dimensions of the data set are:

Length	Columns
284807	31

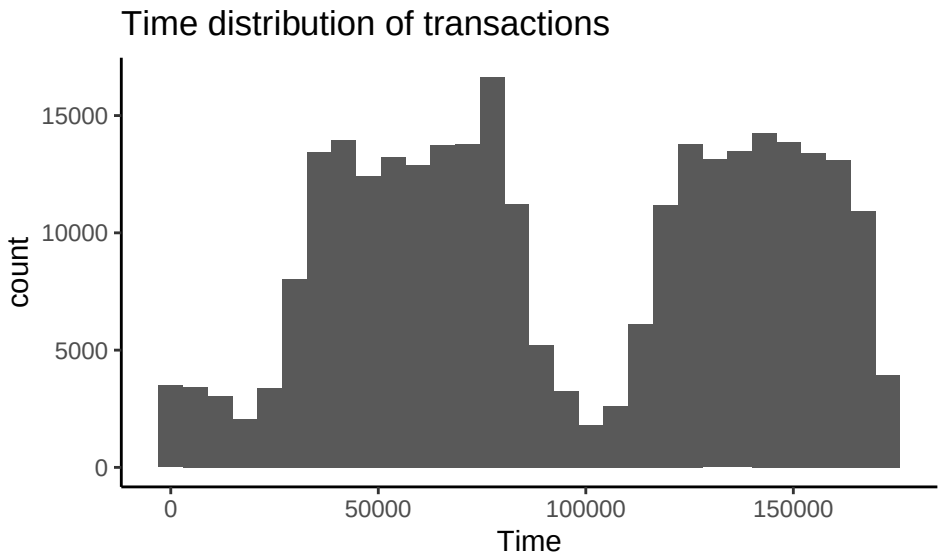
Next we check for any missing data in the data set:

Number of missing entries : 0

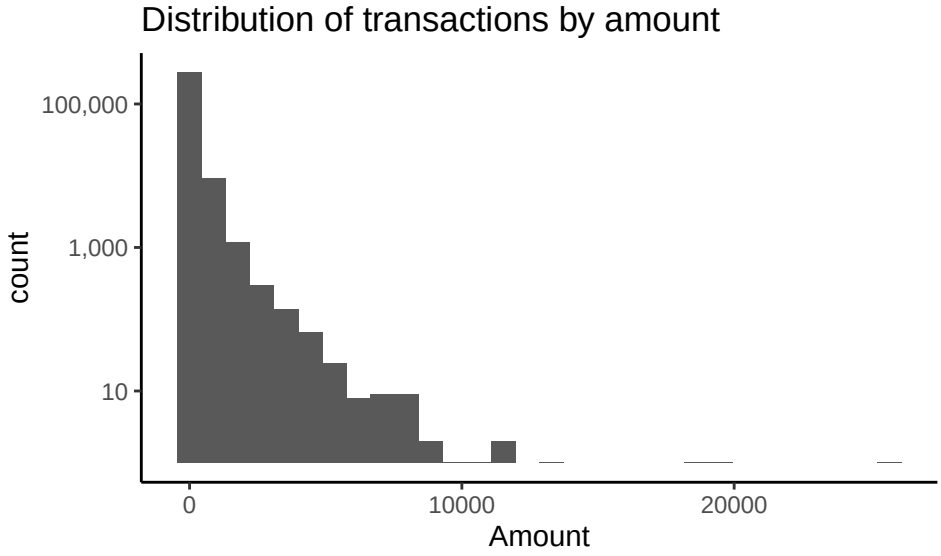


The distribution of fraud cases shows the data is unbalanced with fraud cases representing less than 1% of the data. This presents a challenge for training algorithms.

AS most of the data in the data set have been anonymized and scaled as V1 to v28, there is not mush to be learned from exploring them so i will focus on time and amount.

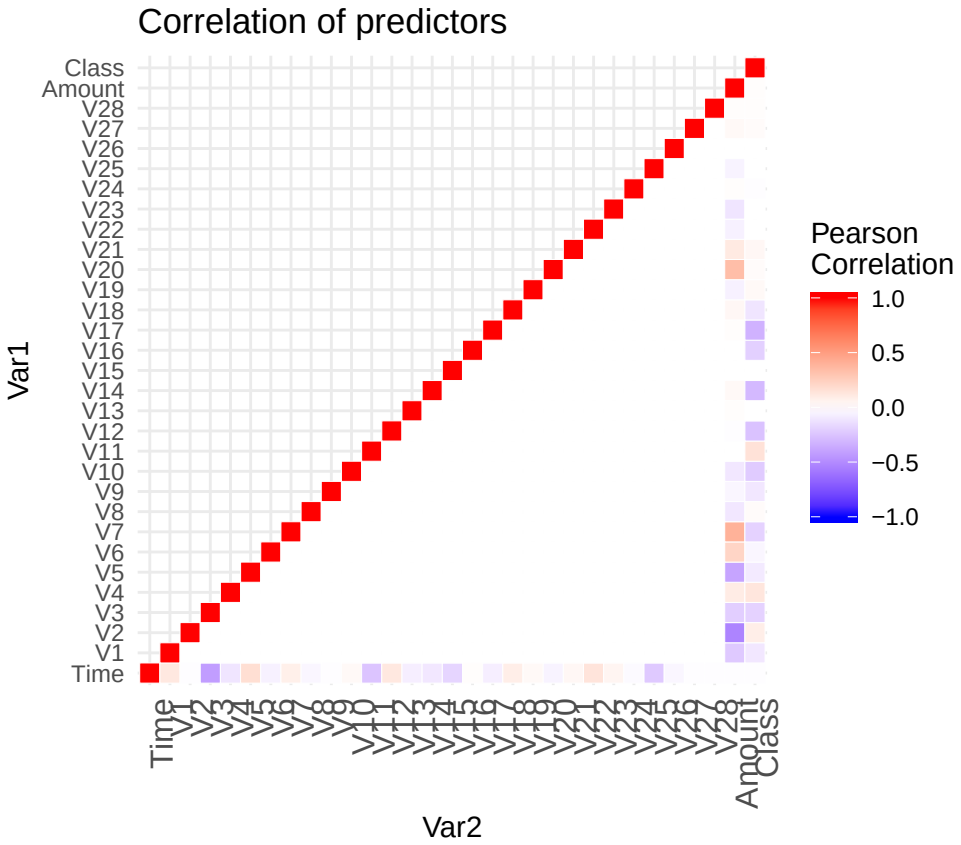


The data contains credit card transaction over a span of two days. The distribution of transactions over the time has two peaks with a period of low activity in between. This may be explained as a drop in transaction at night.



The data set contains 284,807 transactions. The mean value of all transactions is \$88.35 while the largest transaction recorded in this data set amounts to \$25,691.16. The distribution of the monetary value of all transactions is heavily right-skewed. The vast majority of transactions are relatively small and only a tiny fraction of transactions comes even close to the maximum.

Lastly, we investigate the correlation between the predictors and the Class.



The predictors do not have any correlation with each other and only a few are correlated to the Class. This may be due to the fact that the data was prepared using PCA, hence the predictors are principal components. The Class imbalance may also distort the importance of correlations to the class variable.

2.2 Data Preparation

The anonymized data has been scaled and centered around zero while the time and amount predictors have not. Without scaling them, they would unduly influence any logistic regression based algorithms or algorithms that rely on distance(KNN). To avoid this, we apply scaling to the fields using the following formula

$$z = \frac{x - \mu}{\rho}$$

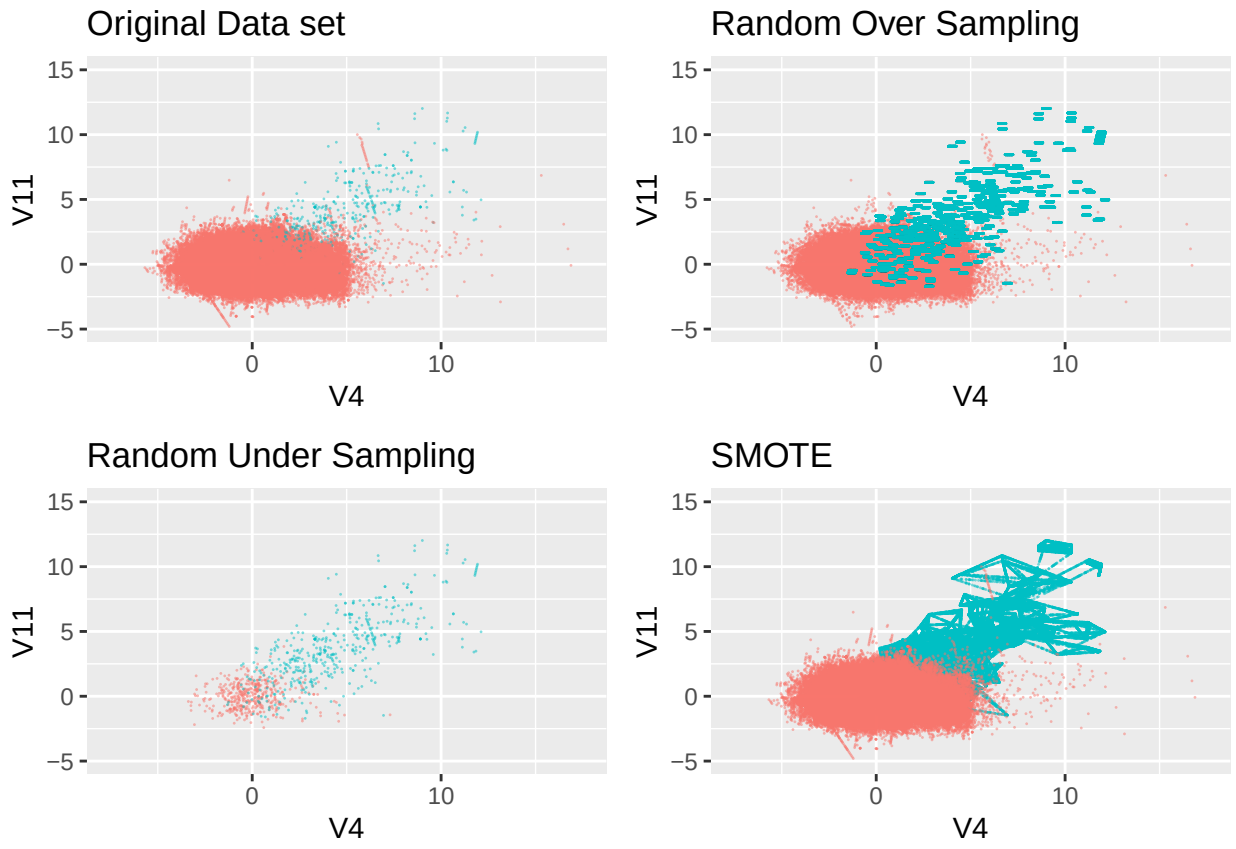
The next step in preparing the training data is to transform it such that it will allow algorithms to pick up specific characteristics that make a transaction more or less likely to be fraudulent. An algorithm will achieve >99% by predicting all transactions as non-fraudulent. This is not what we want. We want the algorithm to correctly identify fraudulent transactions.

To solve this problem, we will evaluate several methods of balancing the data.

Random over sampling: Random Oversampling involves supplementing the training data with multiple copies of some of the minority classes. Oversampling can be done more than once (2x, 3x, 5x, 10x, etc.) This is one of the earliest proposed methods, that is also proven to be robust. Instead of duplicating every sample in the minority class, some of them may be randomly chosen with replacement.

Random under Sampling: Randomly remove samples from the majority class, with or without replacement. This is one of the earliest techniques used to alleviate imbalance in the data set, however, it may increase the variance of the classifier and may potentially discard useful or important samples

Synthetic Minority Over-sampling Technique (SMOTE): There are a number of methods available to over-sample a data set used in a typical classification problem (using a classification algorithm to classify a set of images, given a labelled training set of images). The most common technique is known as SMOTE: Synthetic Minority Over-sampling Technique. To illustrate how this technique works consider some training data which has s samples, and f features in the feature space of the data. Note that these features, for simplicity, are continuous. As an example, consider a data set of birds for classification. The feature space for the minority class for which we want to over-sample could be beak length, wingspan, and weight (all continuous). To then over-sample, take a sample from the data set, and consider its k nearest neighbors (in feature space). To create a synthetic data point, take the vector between one of those k neighbors, and the current data point. Multiply this vector by a random number x which lies between 0, and 1. Add this to the current data point to create the new, synthetic data point.



The training set generated through random over sampling has balanced the data set by repeating existing points from the minority class. Although this method balances the data, algorithms that are trained on this will learn too much about the same few point and the would not generalize well. The random under sampling data set has balanced the data by sampling a few points from the majority class. The problem with this is that a lot of information on the majority class has been lost. The SMOTE data set overcomes both of these challenges, by creating synthetic points to boost the minority class, we maintain all the information we have about the majority class, and boost the proportion of the minority class without over fitting a few points.

2.3 Modeling Approach

We shall train a decision tree, random forest and artificial neural network on both the original data set and the balanced data set(SMOTE). Given the imbalance in the data set, accuracy will not be a good measure of model performance. Instead we will use receiver Receiver Operating Characteristics-Area Under the Curve or ROC-AUC performance measure. Essentially, the ROC-AUC outputs a value between zero and one, whereby one is a perfect score and zero the worst. If an algorithm has a ROC-AUC score of above 0.5, it is achieving

a higher performance than random guessing.

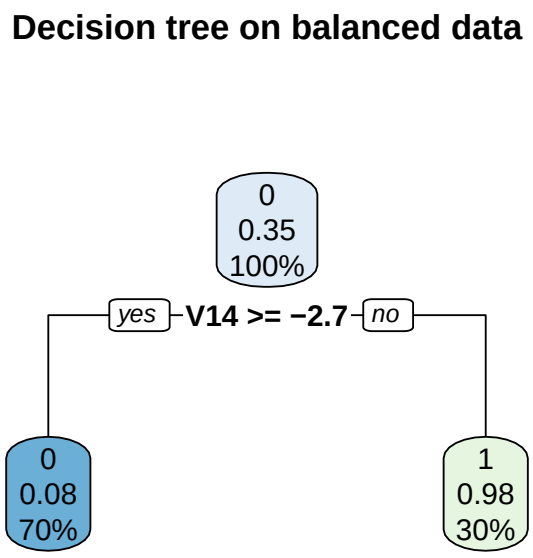
Confusion Matrix

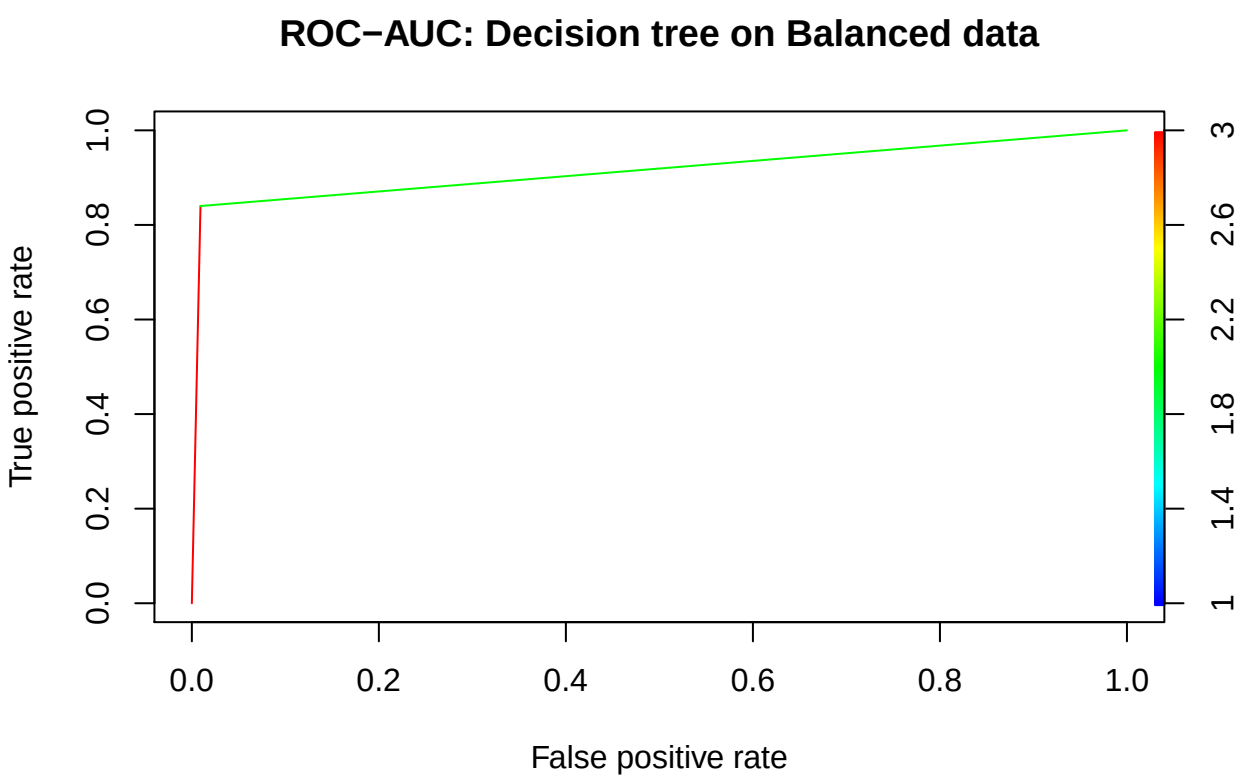
Class	Actual_Legit	Actual_Fraud
Predicted_Legit	True Positive	False Positive
Predicted_Fraud	False Negative	True Negative

The confusion matrix also provides a good measure of performance. The True negatives represent fraudulent transactions that have been correctly classified. False positives are fraudulent transactions that have been classified as legitimate. These must be minimized or eliminated as each transaction in this category represents lost cash. The false negatives are legitimate transactions that will be declined as fraudulent transactions. These create a bad user experience and if they are sufficiently high, will result in lost revenue from aggrieved customers.

2.3.1 Decision Tree

For a start, we train a decision tree model on the original data set and the SMOTE balanced set.





This decision tree achieves a ROC-AUC of 0.9153749. This is quite impressive considering the simplicity of the model

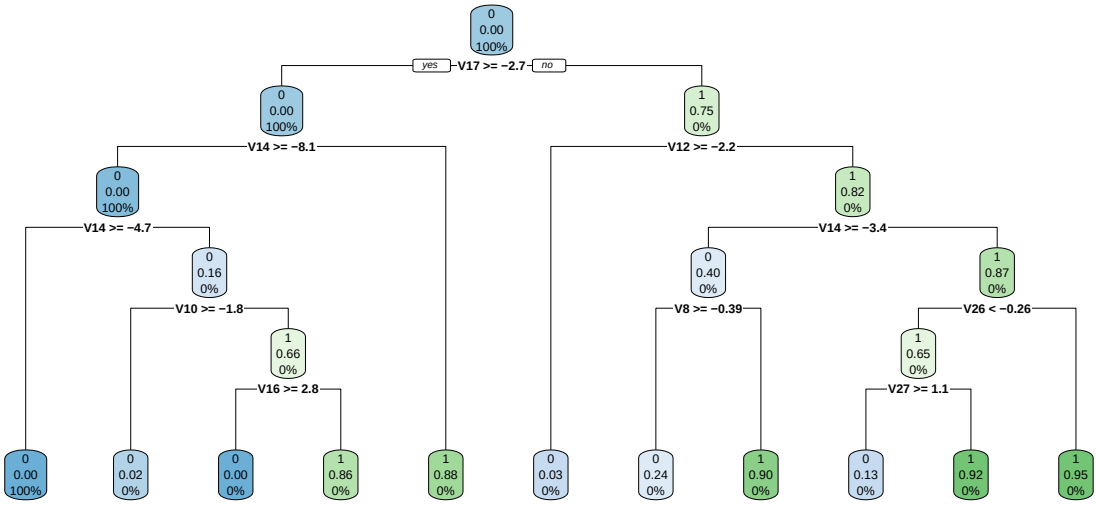
Confusion Matrix: Decision tree on SMOTE data

	0	1
0	28169	8
1	263	42

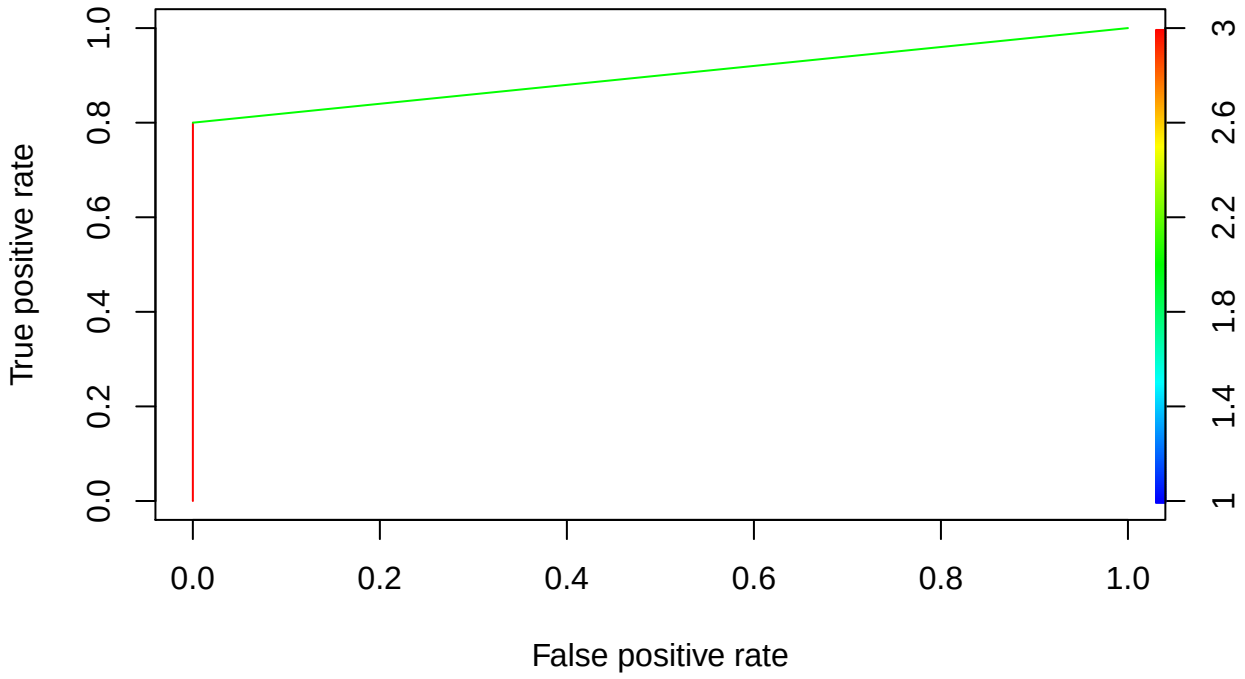
From the confusion matrix, the model correctly identified 84% of the fraudulent cases

Next we train a decision tree on the Original data set without any balancing

Decision tree on unbalanced data



ROC−AUC: Decision tree on Unbalanced data



This model obtains a ROC-AUC score of 0.8999648

Confusion Matrix: Decision tree on Unbalanced data set

	0	1
0	28430	10
1	2	40

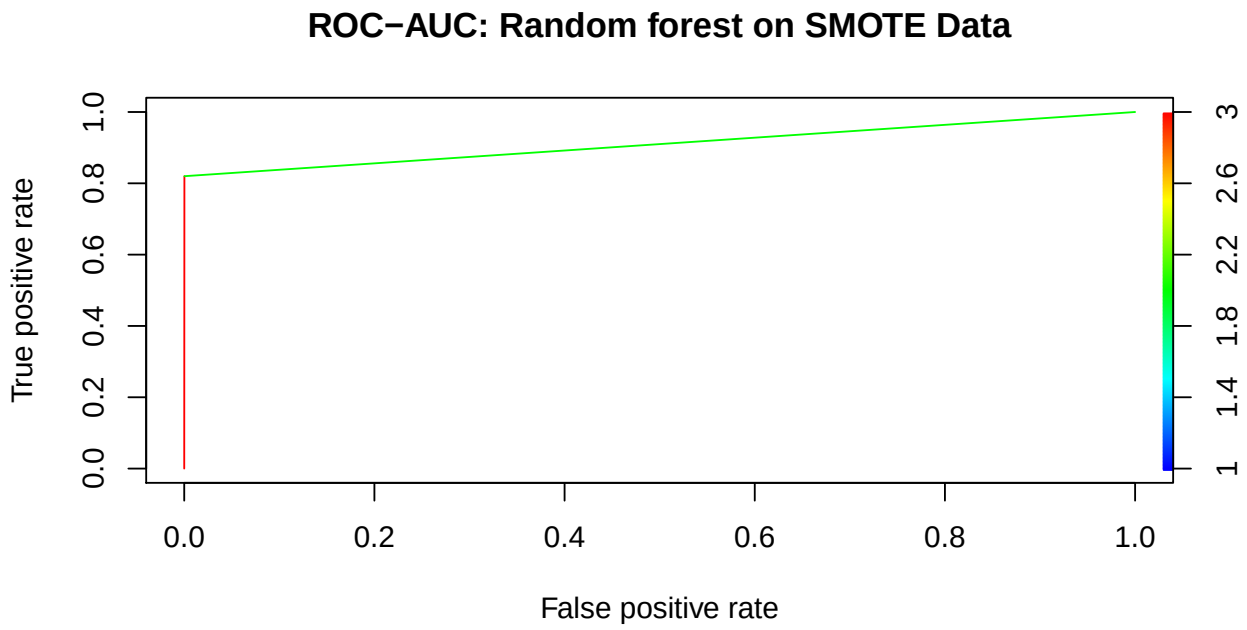
Only 80% of the fraud cases were identified. The model trained with the SMOTE data set did a better job at identifying fraud cases

method	SMOTE	original
Decision tree	0.9153749	0.8999648

The decision tree on the original set achieved a ROCAUC of 0.8999648 while the SMOTE data set achieved 0.9153749. The model trained with SMOTE balanced data set recorded an improvement of 1.6834743%

2.3.2 Random Forest

Next we train and compare random forest models on the SMOTE and original data set. The performance of the model is presented below:



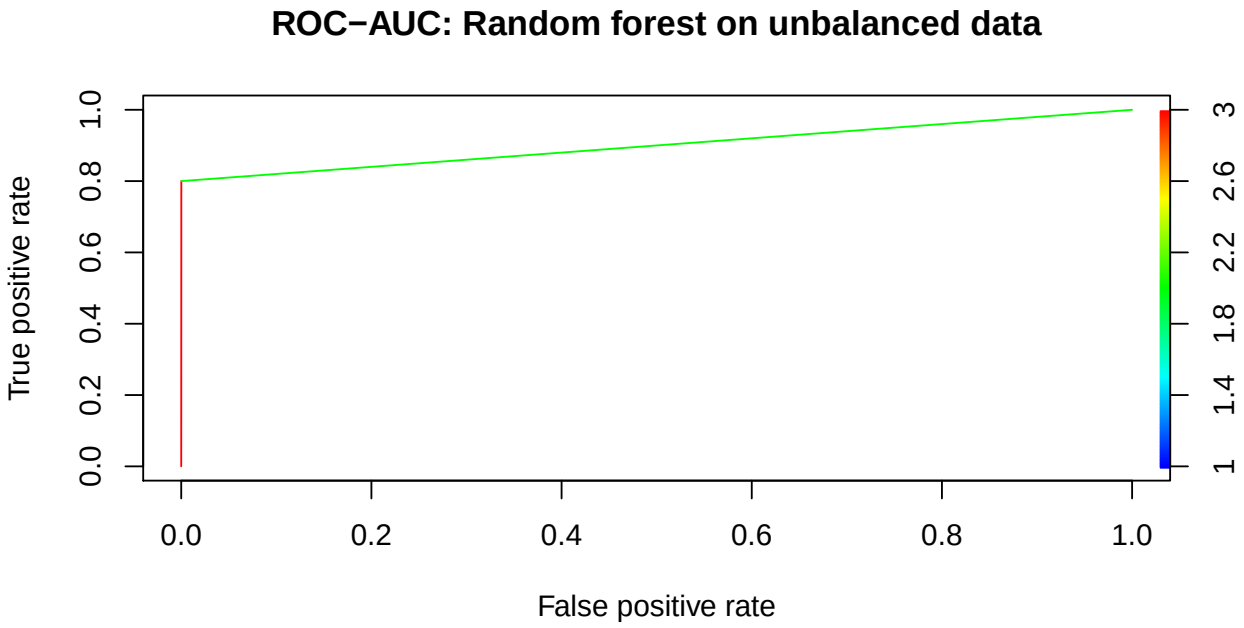
The Random Forest achieved and ROC-AUC of 0.9099121

Confusion Matrix: Random Forest on SMOTE data

	0	1
0	28427	9
1	5	41

The random forest identified 82% of fraud cases.

Next we train a random forest on the unbalanced data. The performance is shown below:



Confusion matrix: Random Forest on unbalanced data

	0	1
0	28429	10
1	3	40

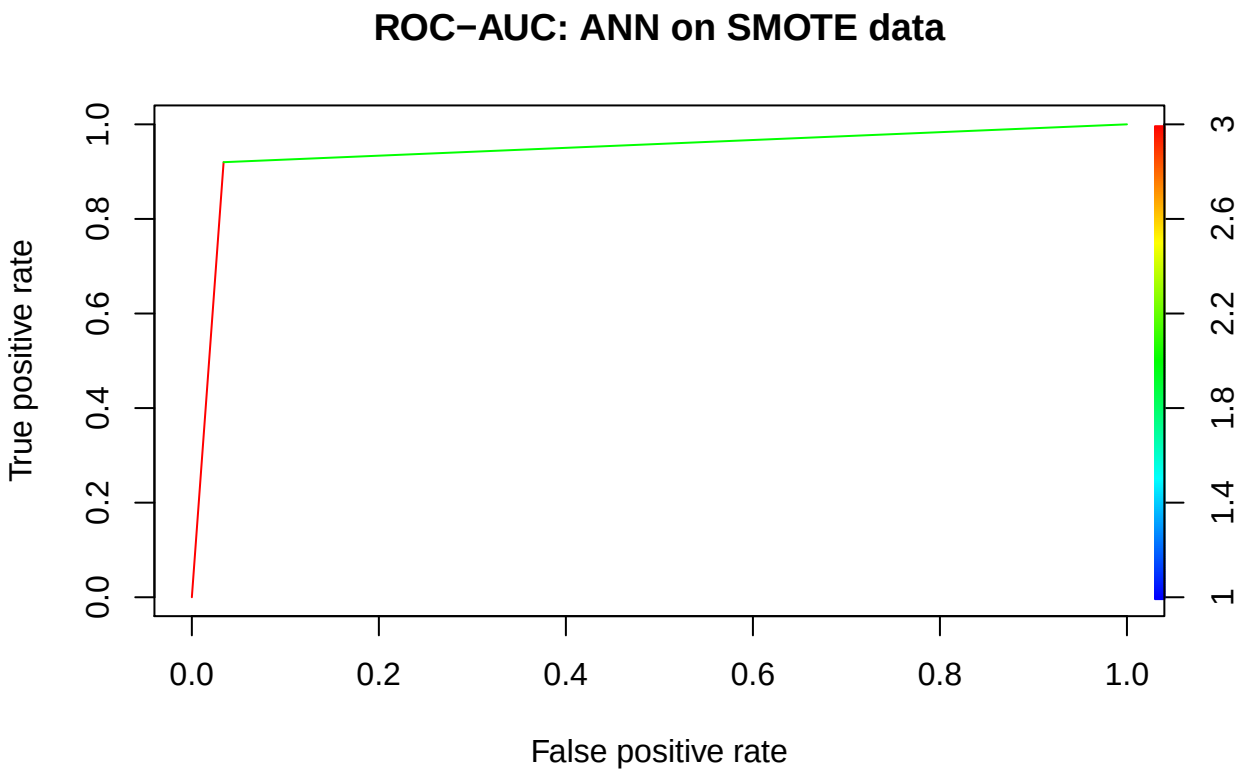
method	SMOTE	original
Decision tree	0.9153749	0.8999648
Random forest	0.9099121	0.8999472

The model trained on the SMOTE balanced data set shows an improvement over the performance of the original data set. This can be attributed to the fact that balancing the data set allows the algorithm place equal importance on both classes.

2.3.3 Artificial Neural Network

Finally we will train an artificial neural network on the SMOTE data and original data. The performance of the model is as follows:

Due to the limited computational resources available for training, we will train the model with only 10% of the training data.

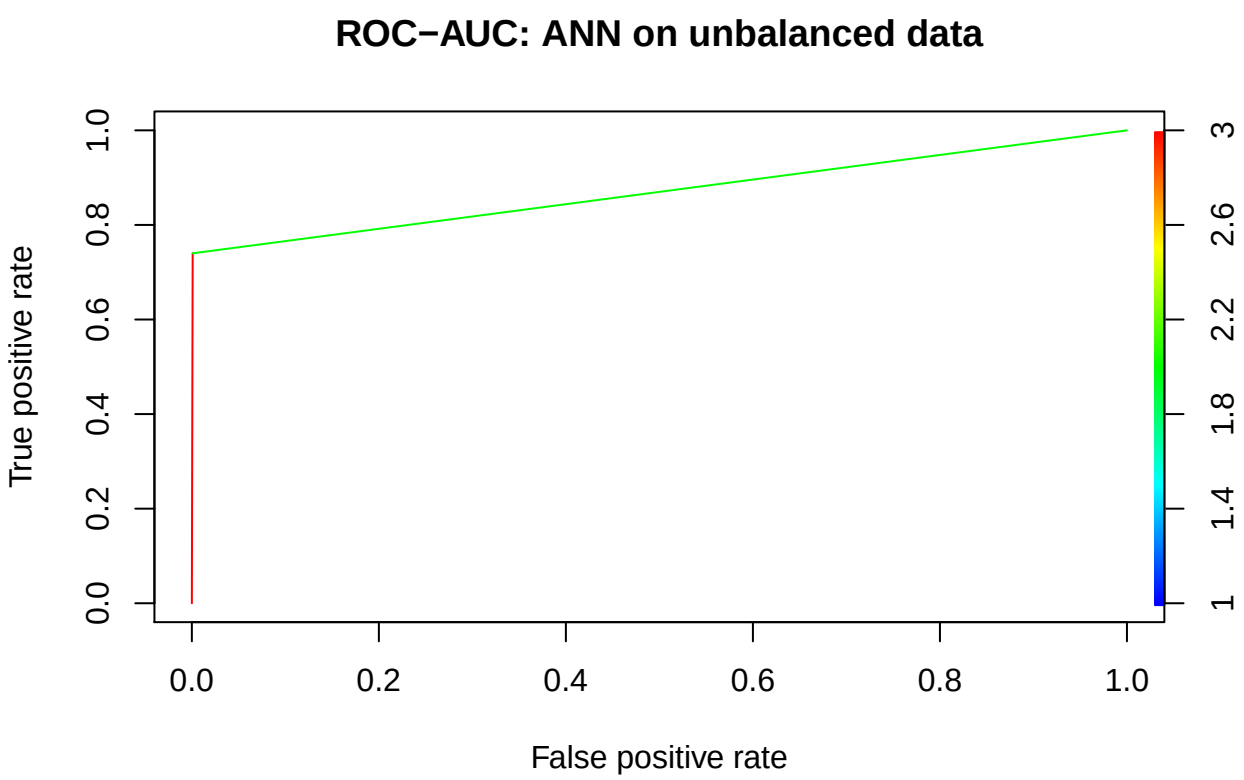


Confusion Matrix: ANN on SMOTE date

	0	1
0	27466	4
1	966	46

The artificial neural network identified 92% of the fraud cases. But this comes at a cost. The number of false negatives increased to 966. While this model performs well in detecting fraud cases, it also causes an increased amount of user complaints due to legitimate transactions being declined.

For comparison, we train the same model on the original data set.



method	SMOTE	original
Decision tree	0.9153749	0.8999648
Random forest	0.9099121	0.8999472
Artificial Neural Network	0.9430121	0.8695604

Confusion Matrix: ANN on unbalanced data

	0	1
0	28407	13
1	25	37

3 Results

method	SMOTE	original
Decision tree	0.9153749	0.8999648
Random forest	0.9099121	0.8999472
Artificial Neural Network	0.9430121	0.8695604

The summary of the results show that all the models performed better when they are trained on the balanced data set. The artificial neural network performed best out of the models trained and achieved a ROC-AUC of 0.9430121

Confusion Matrix Summary

Method	TP	FP	FN	TN
decision tree on SMOTE	28169	8	263	42
decision tree on Original	28430	10	2	40
Random Forest on SMOTE	28427	9	5	41
Random Forest on Original	28429	10	3	40
ANN on SMOTE	27466	4	966	46
ANN on Original	28407	13	25	37

From the confusion matrix, The artificial neural network is clearly able to properly classify more fraud cases than the other models.

4 Conclusion

The artificial neural network proved to be superior to both the random forest and decision tree. The results can be improved by tuning the parameters of the models and trying ensemble methods such as XGBoost and lightGBM. The Artificial neural network can also be improved by training it on the complete training set.

5 Acknowledgements

The data set has been collected and analysed during a research collaboration of Worldline and the Machine Learning Group (http://mlg.ulb.ac.be) of ULB (Université Libre de Bruxelles) on big data mining and fraud detection. More details on current and past projects on related topics are available on https://www.researchgate.net/project/Fraud-detection-5 and the page of the DefeatFraud project