Credit Card Fraud Detection using Machine Learning and Neural Networks

Abstract – ***With the rise of online payment credit cards have had a huge rule in our daily life and economy for the past two decades and it is important task for companies to identify fraud and non-fraud transactions. Multiple methods have been suggested for this problem and they each have their own pros and drawbacks. In this paper we will apply machine learning algorithms and artificial neural networks on the real-world dataset that is taken from Kaggle [1]. Our main goal is to detect all fraud cases. Moreover, we will compare the results of different linear, ensemble, voting and other methods from open-source libraries as well as with methods done in previous papers in this field.***

Keywords *Credit Card Fraud, Supervised Machine Learning, Artificial Neural Networks, Imbalanced classification.*

Introduction

In the modern world, credit cards are important part of our life as people receive their salary, do their shopping, pay their bills with the help of credit cards. Only in one day there are more than 1 billion credit card transactions are made according to The Nilson Report. For the fraudsters who are eager to steal it can be another opportunity. There is plethora of methods which scammers use. Only in 2018, without even presence of card more than 400 million dollars were stolen. Credit card frauds are most common type of identity theft, occurring 41% of all identity theft reports. Moreover, for the most part police cannot investigate on the credit card fraud due to its international nature.

Credit card fraud detection's goal is to decide if the given transaction is fraudulent or not according to the previous transaction data. Now the challenge in this type of dataset is that, when you want to train a model while measuring the accuracy the results will be higher than 90% even if the model labels all transactions as non-fraud and the reason for that is because these kinds of datasets are highly imbalanced. For example, in the data [1] that we will use only 492 transactions are fraud and 284315 transaction are not fraud. This means roughly 0.17 percent of all transactions.

In this paper I used multiple supervised learning algorithms, deep learning models and compared their ROC_AUC score, F1-Score, Precision and Accuracy on the real-world dataset.

Related works

Plethora of classical machine learning algorithms such as Decision Tree, Naïve Bayes, K-Nearest Neighbour, Support Vector Machine, Random Forest, XGBoost and other deep learning methods were applied on the process of the detection of credit card frauds. Tree based and ensemble algorithms were successful alongside with Artificial Neural Networks and Logistic Regression. In the past works done in this field, it was important to balance the data as there is a huge imbalance in the dataset between fraud and non-fraud transactions. The most common methods used for balancing were over sampling, under sampling.

In one study, the outlier mining was used to detect credit card frauds and it was more successful than anomaly detection with clustering.

In the study [1] which is done in 2001, Jun-ichi Takeuchi and Kenji Yamanishi constructed two stage algorithm which is based on unsupervised learning. In the first step of this technique, the algorithm trains Gaussian model for scoring unsupervised data. Later it imputes. Second part of the problem is where the labeled data is used for outlier detector.

Research that has been done in 2018 [2], used supervised machine learning methods such as Random Forest, Stacking Classifier, Logistic Regression and compared them with different metrics like Recall, Accuracy, Precision, etc. They eventually found out that Logistic Regression was the most accurate when it is picked as base estimator of Stacking classifier and it followed by Random Forest and XGB classifier.

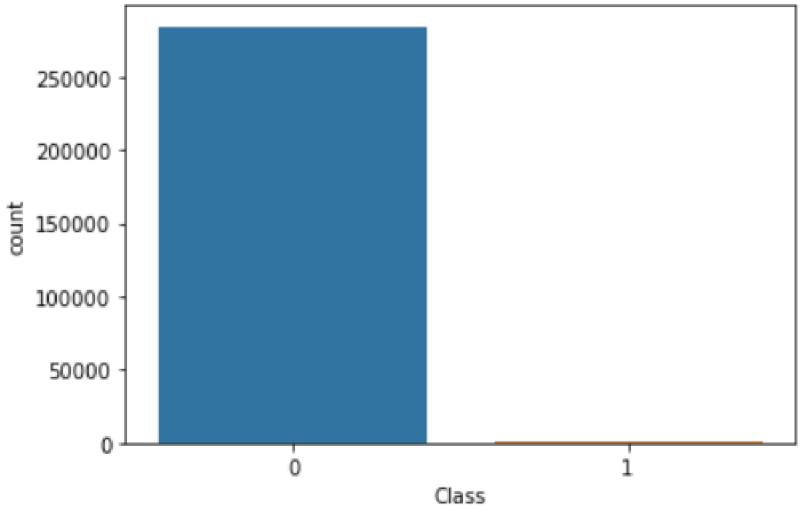
Other study [3], compared advantages and drawbacks of fraud detection methods. For instance, they have figured it out that although Hidden Markov Model is fast at detection, its accuracy is low, and it is not scalable for large data sets. On the other hand, Bayesian networks are good at accuracy while being expensive. Moreover, when it comes to artificial neural networks, they are portable, effective dealing with noisy data while being difficult to setup and having bad explanation capabilities. Another interesting point from this study was that they mention that there is no suitable metrics to evaluate the results of these prediction models as well as lack of adaptive credit card fraud detection systems.

Another study [4] from 2019 tried interesting approach. This team used clustering technique to divide the data into three different groups according to the transaction amount. They used range partitioning for it. In the next step they used Sliding-Window method by aggregating transactions into groups and then extracting patterns in cardholder’s behavior. Minimum, maximum and average of transaction amounts made by cardholders were calculated. And every time when there is a new transaction made by the new transactions is fed to the window while the old one is removed from it. Later they use different classifiers on each cluster using patterns and extract fraud case signs. To overcame imbalance on dataset, they use SMOTE because oversampling does not provide better results.

The most recent study [5] that was done in 2019 used new machine learning algorithms to detect outliers. That team used Local Outlier Factor and Isolation Forest algorithms which at the moment are considered most popular outlier detection methods in the industry. Their accuracy 99.6% while they had lower precision at 33%. The reason for low precision in the data is huge imbalance.

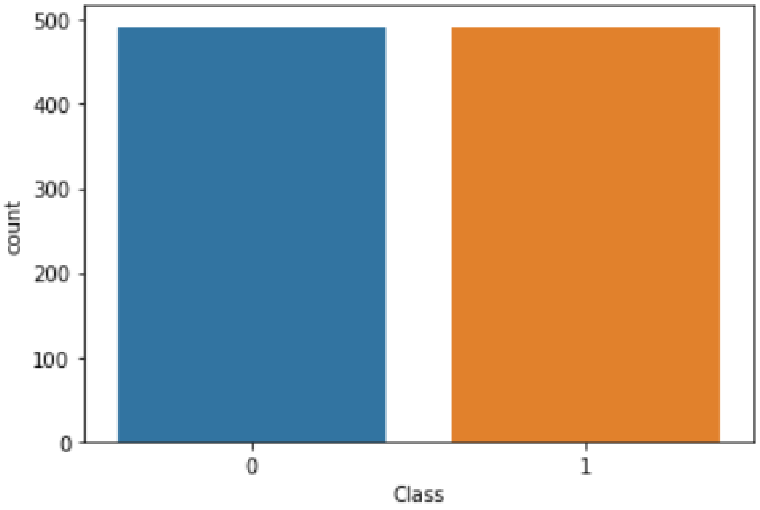
Materials and methods

The dataset that we have picked is one of the most famous datasets in Kaggle and it contains transactions made by credit cards in September 2013 by European cardholders. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions and the figure below shows it visually:

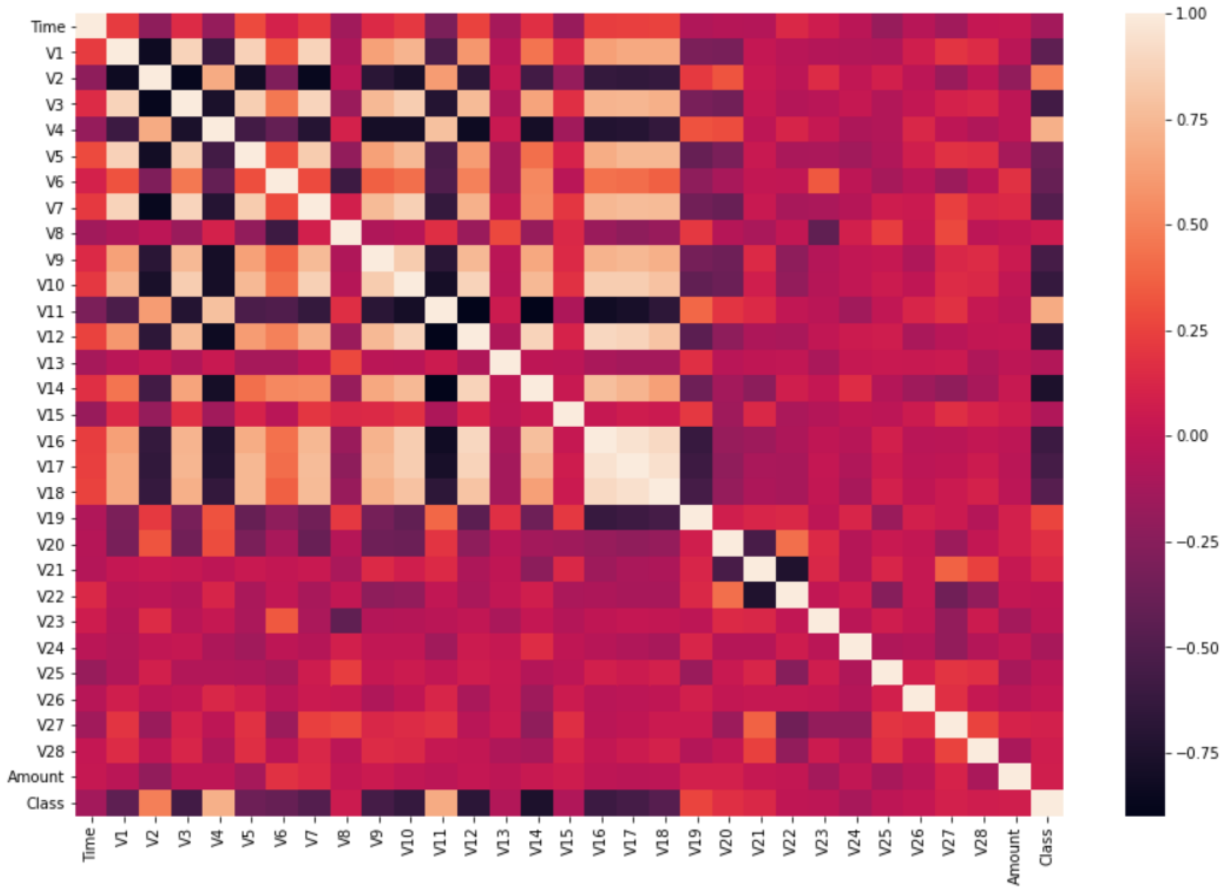


If we use **accuracy = (TP+TN)/ (TP+TN+FP+FN***)* to calculate how well our model works this method would not be efficient. For instance, in our case it is enough to label all rows as non-frauds and our accuracy will be more than 99%.

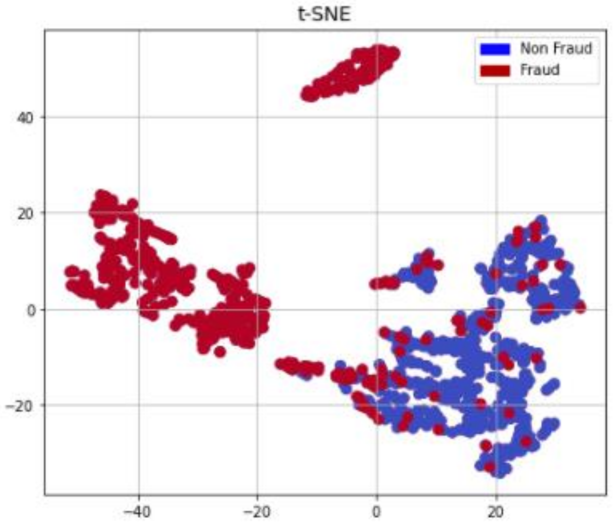
There are few things we could do: Over sampling, under sampling, Generating Synthetic Samples, Using Tree algorithms, using penalized models. Experimenting on these methods we found out that using tree algorithms on under sampled data gave us the best results. Here is how our under sampled data looks like.



After under sampling we have left with 492 fraud ad 492 non fraud rows. In the dataset we have total 31 columns. 28 of them is labelled from V1-V28 and there is Time, Amount and Class (fraud or non-fraud) which target column. The correlation between them is like this.



The interesting pattern we need here is in the last row (Class) and we can see that columns V10, V14, V12 have negative correlation with class variable. By using only those three columns and using dimensionality reduction we were able to find some interesting patterns in the data.



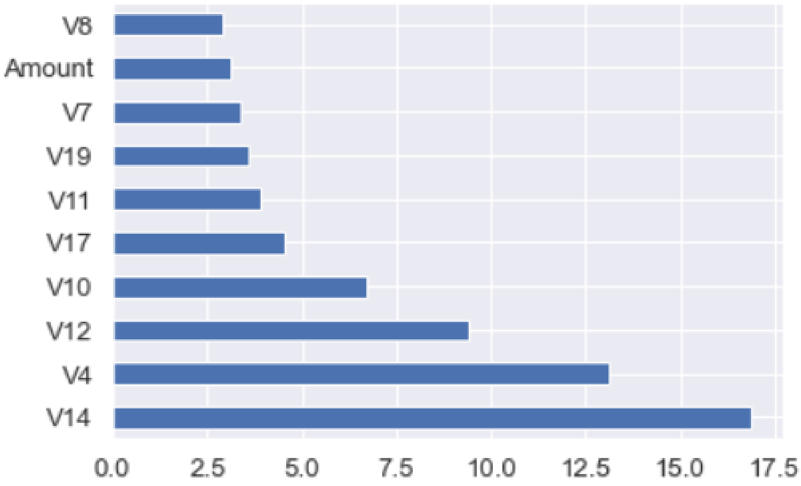
Results

Based on the experiments we decided to divide data to 80-20 train test split ratio. Below in the table you can witness Recall, ROC AUC, F1 scores of different machine learning algorithms combined with stratified cross validations gave.

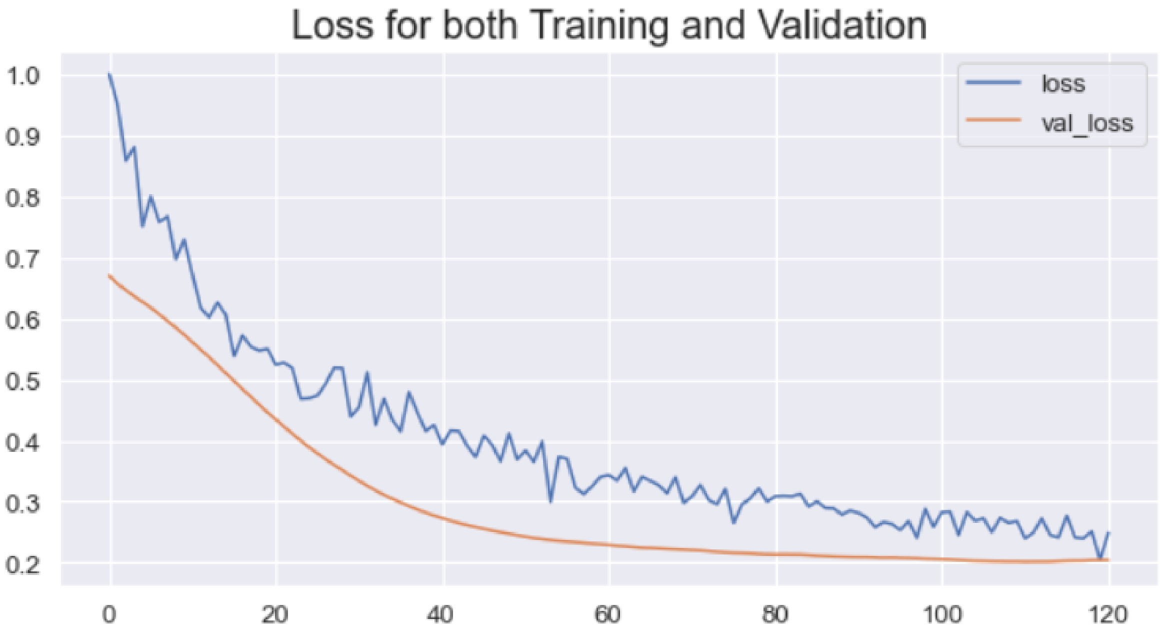
Model	ROC AUC	Recall Score	F1 score
Gradient Boosting	0.986	0.960	0.93
Random Forest	0.991	0.948	0.97
Logistic Regression	0.976	0.975	0.68
Logistic Regression with Bagging	0.979	0.983	0.69
LGBMClassifier	0.992	0.962	0.962
CatBoostClassifier	0.993	0.958	0.969
TensorFlow model	0.977	0.897	0.921

From the results above in the table we can conclude Tree Based Algorithms performed better.

CatBoostClassifier has the highest ROC score followed Light GBM and Random Forest. I did not have good results with XGBClassifier and LGB is 7 times faster that it. Logistic Regression had the highest recall but a very low f1 score. Like in ROC Cat boost and Light GBM had highest f1 scores.



When comes to feature selection columns V14, V4, and V12 were three most useful columns. While training neural networks on TensorFlow there was not much difference on the loss and accuracy of model after around 90 epochs.



Conclusion

Without knowing the columns real names, it was difficult to perform feature engineering. Selecting less columns was not good for the overall results. So eventually we feed all columns to the model. First, we explored the dataset which was crucial step otherwise we would not be aware that dataset was imbalanced. Second, we did some more exploratory analysis which we were able to find which variables had interesting correlation with each other. Applying dimensionality reduction methods to those selected columns helped us to separate frauds and non-frauds.

Using metrics methods such as recall, f1 score and receiver operating characteristics area under the curve we have witnessed that tree-based models outperform Neural Networks and other classic machine learning methods.

References

[1] https://www.kaggle.com/mlg-ulb/creditcardfraud

[2] Sahil Dhankhad, Emad A. Mohammed, Behrouz Far “Supervised Machine Learning Algorithms for Credit Card Fraudulent Transaction Detection: A Comparative Study”

[3] Samaneh Sorournejad, Zahra Zojaji, Reza Ebrahimi Atani, Amir Hassan Monadjemi, “A Survey of Credit Card Fraud Detection Techniques: Data and Technique Oriented Perspective.”

[4] Vaishnavi Nath Dornadula, Geeth S, “Credit Card Fraud Detection using Machine Learning Algorithms”.

[5] S P Maniraj, Aditya Saini, Swarna Deep Sarkar, Shadab Ahmed, “Credit Card Fraud Detection using Machine Learning and Data Science”.

[6] Andrea Dal Pozzolo, Olivier Caelen, Reid A. Johnson and Gianluca Bontempi. Calibrating Probability with Undersampling for Unbalanced Classification. In Symposium on Computational Intelligence and Data Mining (CIDM), IEEE, 2015

[7] Dal Pozzolo, Andrea; Caelen, Olivier; Le Borgne, Yann-Ael; Waterschoot, Serge; Bontempi, Gianluca. Learned lessons in credit card fraud detection from a practitioner perspective, Expert systems with applications,41,10,4915-4928,2014, Pergamon.

[8] https://catboost.ai/

[9] https://lightgbm.readthedocs.io/en/latest/

[10] Fabrizio Carcillo, Yann-Aël Le Borgne, Olivier Caelen, Frederic Oblé, Gianluca Bontempi Combining Unsupervised and Supervised Learning in Credit Card Fraud Detection Information Sciences, 2019

[11] https://scikit-learn.org/stable/getting_started.html

[12] https://www.kaggle.com/janiobachmann/credit-fraud-dealing-with-imbalanced-datasets

[13] https://www.kaggle.com/joparga3/in-depth-skewed-data-classif-93-recall-acc-now

[14] https://www.kaggle.com/currie32/predicting-fraud-with-tensorflow

[15] https://www.kaggle.com/nareshbhat/outlier-the-silent-killer

[16] https://www.kaggle.com/nschneider/gbm-vs-xgboost-vs-lightgbm

[17] Dal Pozzolo, Andrea Adaptive Machine learning for credit card fraud detection ULB MLG PhD thesis (supervised by G. Bontempi)

[18] Bertrand Lebichot, Yann-Aël Le Borgne, Liyun He, Frederic Oblé, Gianluca Bontempi Deep-Learning Domain Adaptation Techniques for Credit Cards Fraud Detection, INNSBDDL 2019: Recent Advances in Big Data and Deep Learning, pp 78-88, 2019

[19] Carcillo, Fabrizio; Dal Pozzolo, Andrea; Le Borgne, Yann-Aël; Caelen, Olivier; Mazzer, Yannis; Bontempi, Gianluca. Scarff: a scalable framework for streaming credit card fraud detection with Spark, Information fusion,41, 182-194,2018, Elsevier.

[20] Xiaohan Yu, Xianwei Li, Yiyang Dong, Ruizhe Zheng. “A Deep Neural Network Algorithm for Detecting Credit Card Fraud.”

[21] N. Mahmoudi, E. Duman, “Detecting credit card fraud by Modified Fisher.

Discriminant Analysis”

[22] “GitHub (2019). Feature selector.” Available on GitHub application.

[23] https://www.tensorflow.org/tutorials/structured_data/feature_columns

[24] https://optuna.org/#code_examples

[25] Fatima Zohra El Hlouli, Jamal Riffi, Mohamed Adnane Mahraz, Ali El Yahyaouy, Hamid Tairi “Credit Card Fraud Detection Based on Multilayer Perceptron and Extreme Learning Machine Architectures”