Credit Card Fraud Detection using Machine Learning

Kamal Sharma (Student No: 501072994)

Supervisor: Zekiye Erdem

Submission Date: 03rd Dec 2021

Table of Contents

[1. Introduction/ Brief context of the theme 3](#_Toc89465935)

[2. Summary of Research 3](#_Toc89465936)

[Research Question 4](#_Toc89465937)

[3. Dataset Description 4](#_Toc89465938)

[4. Methodology/Techniques 4](#_Toc89465939)

[5. Literature Review 6](#_Toc89465940)

[I. Champion-challenger analysis for credit card fraud detection: Hybrid ensemble and deep learning [5] 6](#_Toc89465941)

[II. A cost-sensitive decision tree approach for fraud detection [6] 7](#_Toc89465942)

[III. A Data Mining Based System For Transaction Fraud Detection [7] 7](#_Toc89465943)

[IV. Comparative Evaluation of Credit Card Fraud Detection Using Machine Learning Techniques [8] 8](#_Toc89465944)

[6. Results/conclusion 8](#_Toc89465945)

[7. Next Steps 10](#_Toc89465946)

[Chart 1 : Overall Methodology 11](#_Toc89465947)

[11](#_Toc89465948)

[Table 1: Statistical Analysis 12](#_Toc89465949)

[Fig 1. Feature Engineering using Random Forest (threshold = 0.024) 14](#_Toc89465950)

[Fig 2. Imbalance Dataset Treatment – Under sampling and Oversampling 15](#_Toc89465951)

[Table 2: Model Performance Chart 16](#_Toc89465952)

[Appendix 16](#_Toc89465953)

[References: 17](#_Toc89465954)

# 1. Introduction/ Brief context of the theme

We live in a digital world where most of our daily transactions happen online, whether its transferring money to an account or an online purchase with a credit card. With an increased online presence comes a greater risk fraudulent transaction. The biggest category in this area is fraud credit card transactions that cause worldwide financial losses. According to the 2019 Nilson Report, [[1]](#_References:)  card fraud losses worldwide have increased from 9.84 billion dollars in 2011 to 27.85 billion dollars in 2018, and are projected to reach more than 40 billion dollars in 2027. Credit card frauds can further be of two types.

1. Credit card present (CCP) frauds: Transaction that happen at a merchant store, where a physical card is needed. This contributes to around 19% of our fraud transactions.
2. Credit Card not present (CCNP) frauds: Payments performed on the internet, by phone or by email. They amount to 81% of the fraud payment card transactions.

For the purpose of this project, we are going to focus our attention on CCNP category.

# 2. Summary of Research

This project is aimed at identifying a fraud credit card transaction using Machine Learning and classification methods. What particularly makes this challenging is a high degree of skewness in the dataset as the non-fraudulent transactions are very high compared to fraudulent transaction and this highly resonates to any real-world situation. In our research we are going to use under sampling techniques, reducing the number of non-fraudulent transaction and oversampling techniques, that is increasing the number of fraudulent transactions to increase the accuracy of our prediction.

## Research Question

What machine learning algorithm provides sufficiently high degree of accuracy to determine/isolate a fraudulent credit card transaction from a non-fraudulent transaction?

# 3. Dataset Description

The dataset we are going to use for this research is “Credit Card Fraud Detection” [[2]](#_References:) hosted on Kaggle. ( https://www.kaggle.com/mlg-ulb/creditcardfraud)

The dataset 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. Features V1, V2, … V28 are the principal components obtained with PCA transformation of original variables, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

Please reference [Table 1](#_Table_1:_Dataset) below for a statistical analysis of the dataset.

# 4. Methodology/Techniques

This section provides an overview of the methodology/techniques used for this project to train and test machine learning model.

1. Data Import : Download the dataset from Kaggle and import into the library.
2. Pre-processing: We will standardize the features using scale functions in Python, and also compensate for any discrepancies found during data exploration like treating missing values and outliers.
3. Data Exploration: At this stage we do an initial data analysis and visual explorations to understand what is in the dataset and other characteristics of the data. We also try to determine the quality of the dataset by looking at missing values and data types of the variables
4. Feature Engineering: The dataset involved has 29 features that are already converted to principal components. We will use **Random Forest Feature selection** method to isolate the most important features from the dataset to train the model for comparison. We run classification models and compare the performance of All feature models with the selected feature model and choose a threshold value that that does not drastically impact model performance.

Please reference [[Fig 1](#_Fig_1._Feature)] in appendix for details.

1. Class Imbalance: We will apply under sampling technique (Random Under sampling) and oversampling technique (SMOTE) to the dataset, and then evaluate if there is an increase in overall performance of the model as compared to the original datasets

Please reference [[Fig 2](#_Imbalance_Dataset_Treatment)] in appendix

1. Data Modeling: After standardization, we will split the dataset into training and test sets with an 80/20 ratio, and apply following ML models
   1. Random Forest Classifier: This is a classification algorithm that consists of many decision trees. It uses bagging and feature randomness when building each individual tree to create an uncorrelated forest of trees whose prediction is better than individual tree.
   2. Logistic Regression Model: A logistic regression will be used to model the outcome probability of a class such as pass/fail, win/loose and in our case fraud/ not fraud.
   3. K Nearest Neighbour Classifier : KNN is a versatile classification algorithm that relies on the idea that similar objects exist in close proximity. In this approach an algorithm is run several times with different values of K and choose the K that minimizes the number of errors encountered and maximizing the accuracy of predictions
2. Evaluation: We will evaluate the performance of our models by computing accuracy, sensitivity (recall), specificity and precision and F1 score.

Stability: To measure stability we use **k-fold cross validation** technique for each of the classification algorithms. Calculate the mean value of the score and compare with the model to determine if the model outputs are stable.

Efficiency: To measure efficiency we measure the time taken by each of the models for training and predicting the output values. We will use these values when comparing performance of different models.

1. Conclusion: In conclusion we will compare the results of the two models and against the original dataset as well as the re-sampled dataset to determine which model provides best results.

Please reference [Figure 1](#_Chart_1_:) below for a process flow of the overall methodology.

# 5. Literature Review

## I. Champion-challenger analysis for credit card fraud detection: Hybrid ensemble and deep learning [[5]](#_References:)

**Summary:** The research paper is aimed to prove if adopting a deep learning model for real world Fraud Detection System ( FDS) performs better then Ensemble model (which consists of many predictive models combined). It uses Artificial Neural networks as the challenger model, and compares its performance against Champion model (the Ensemble model) The paper also draws our attention to the overhead cost of fraud investigators, and the need to address two common types of misclassification problems i. False Alarms ii. Missed Frauds to offset these costs.

**Results**: Based on the results of the off-line and post-launch tests, it was confirmed that the challenger model based on deep learning performs better than the champion model based on the hybrid ensemble.

## II. A cost-sensitive decision tree approach for fraud detection [[6]](#_References:)

**Summary**: Most of the past studies work on constant misclassification costs, however each false negative (FN) has a unique misclassification cost attached to it. This study introduces a cost-sensitive decision tree approach that takes into account variable misclassification costs. A new cost-sensitive decision tree induction algorithm that minimizes the sum of misclassification costs while selecting the splitting attribute at each non-terminal node of the tree is developed and the classification performance is compared with those of the traditional classification methods

**Results**: Performance results of Cost sensitive decision trees with respect to Total Positive Rate (TPR) and Saved Loss Rate (SLR) are considerably higher than some of the traditional data mining methods such as decision trees, ANN and SVM.

## III. A Data Mining Based System For Transaction Fraud Detection [[7]](#_References:)

**Summary**: This paper aims to establish a fraud detection system based on the classification model of random forest and the data processing related to feature engineering. This paper proposes a semi-automatic fraud transaction detection. The automation part is a fraud transaction risk detection model based on random forest, and the core of the other half is an expert reviewer. If the output risk of the risk detection model is higher than the threshold value, it will be regarded as a high-risk transaction and transferred to the expert reviewer. The expert reviewer will combine the expertise and the information provided by the risk detection model to make further judgment.

**Results:** Performance of different models is evaluated and it is observed that the random forest model is superior to the other two models (Logistic Reg and SVM) in terms of AUC ROC score and accuracy.

## IV. Comparative Evaluation of Credit Card Fraud Detection Using Machine Learning Techniques [[8]](#_References:)

**Summary:** This research is focused on identifying external card frauds which account for the majority of credit card frauds. This paper aims to conduct comparative analysis of identification of fraudulent activity on credit card utilizing support vector machine, k-nearest neighbour technique, naïve bayes and logistic regression techniques on Credit card dataset.

**Results:** Four classification models were assessed using evaluation metrices, and logistic regression showed the highest accuracy results (99.074 %) in detecting credit card fraud. Based on the results from this exploration credit card companies should consider using Logistic Regression algorithm for fraud detection.

# 6. Results/conclusion

Data imbalance poses a great challenge in building an effective classification model. In this project we have used some different techniques to treat data imbalance we will do a performance comparison of original dataset and the under sampled and oversampled dataset using Random Forest, Regression and KNN classification algorithms.

For this imbalanced dataset accuracy will not be a good validation metric because of the high class imbalance it will predict the majority class (non-fraud transaction). Therefore, we will use Precision, Recall and F1 score for comparing the performance of these classification models.

Precision is a measure of True positive class that belong to the positive class.

**Precision** = True Positive / (True Positive + False Positive)

Recall is a measure of how well the positive class was predicted.

**Recall** = True Positive / (True Positive + False Negative)

F-score combines Precision and Recall into one single score to balance out the two parameters.

**F-Measure** = (2 \* Precision \* Recall) / (Precision + Recall)

From comparing the results obtained from different models we observe Random Forest classification with the original dataset has the highest F1 score of (0.84). However, while comparing different models we also need to keep in mind the time complexity or the execution time for each of the classification models. Comparing both these metrices KNN algorithm has comparatively equivalent F1 score (0.83) but with the execution time reduced to one fourth of the the Random Forest. So with these observations we can conclude that among the different models that are included in scope of this project KNN algorithm provides best results in terms of predicting the Fraud transactions correctly and with a minimum execution time.

Please reference [[Table 2](#_Table_2:_Model)] for model performance

# 7. Next Steps

In this research we have tried to address the data imbalance problem to be able to design a model that can predict minority class (Fraud Transactions) with a greater accuracy then the original datasets. The performance of our model is fairly high, but there are improvement opportunities in this research. To take it to the next level the process can further be improved by combining the under sampling and oversampling techniques and use a hybrid model that under samples the majority class and oversamples the minority class at the same time.

## Chart 1 : Overall Methodology

* Import dataset
* Import relevant libraries for data handling and visualization.

1. Import Dataset

* + Standardize numerical variables using scale functions.
  + Impute missing values (if any) and Treat outliers.
  + Remove variable dependencies.

2. Pre processing

## 

3. Exploratory Data Analysis

* + Analyze and visualize dataset variables using statistical techniques
  + Find missing values.
  + Find dependencies and apply RF feature selection to select imp features.

## 

* + Apply Random under sampling and SMOTE oversampling to treat class imbalance.
  + Re-train the models and compare results to original

4. Treat Class Imbalance

* + Split dataset into train-test sets.
  + Train Random Forest, Logistic Reg and KNN Classification models using training set and predict data for test
  + User k-Fold cross validate to test stability of each model.

5. Data Modeling

* + Use model to predict values for test set.
  + Evaluate model performance based on Precision, Recall and F1score and the time complexity.

6. Evaluation

* + Compare performance of different models with original , under sampled and over sampled data
  + Identify best performance based on evaluation metrices.
  + Evaluate if Under/Oversampling techniques affect the performance of the model

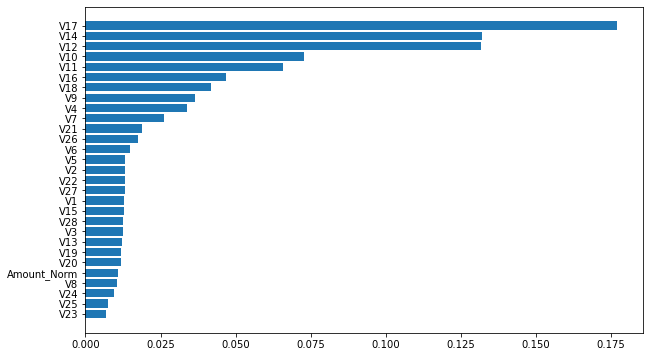
7. Summary

## Table 1: Statistical Analysis

No of observations = 284807

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Variable | Mean | Std Dev | Min | 25% | 50% | 75% | Max |
| Time | 9.48E+04 | 47488.14596 | 0 | 54201.5 | 84692 | 139320.5 | 172792 |
| V1 | 3.92E-15 | 1.958696 | -56.40751 | -0.920373 | 0.018109 | 1.315642 | 2.45493 |
| V2 | 5.69E-16 | 1.651309 | -72.715728 | -0.59855 | 0.065486 | 0.803724 | 22.05773 |
| V3 | -8.77E-15 | 1.516255 | -48.325589 | -0.890365 | 0.179846 | 1.027196 | 9.382558 |
| V4 | 2.78E-15 | 1.415869 | -5.683171 | -0.84864 | -0.019847 | 0.743341 | 16.87534 |
| V5 | -1.55E-15 | 1.380247 | -113.743307 | -0.691597 | -0.054336 | 0.611926 | 34.80167 |
| V6 | 2.01E-15 | 1.332271 | -26.160506 | -0.768296 | -0.274187 | 0.398565 | 73.30163 |
| V7 | -1.69E-15 | 1.237094 | -43.557242 | -0.554076 | 0.040103 | 0.570436 | 120.5895 |
| V8 | -1.93E-16 | 1.194353 | -73.216718 | -0.20863 | 0.022358 | 0.327346 | 20.00721 |
| V9 | -3.14E-15 | 1.098632 | -13.434066 | -0.643098 | -0.051429 | 0.597139 | 15.595 |
| V10 | 1.77E-15 | 1.08885 | -24.588262 | -0.535426 | -0.092917 | 0.453923 | 23.74514 |
| V11 | 9.17E-16 | 1.020713 | -4.797473 | -0.762494 | -0.032757 | 0.739593 | 12.01891 |
| V12 | -1.81E-15 | 0.999201 | -18.683715 | -0.405571 | 0.140033 | 0.618238 | 7.848392 |
| V13 | 1.69E-15 | 0.995274 | -5.791881 | -0.648539 | -0.013568 | 0.662505 | 7.126883 |
| V14 | 1.48E-15 | 0.958596 | -19.214325 | -0.425574 | 0.050601 | 0.49315 | 10.52677 |
| V15 | 3.48E-15 | 0.915316 | -4.498945 | -0.582884 | 0.048072 | 0.648821 | 8.877742 |
| V16 | 1.39E-15 | 0.876253 | -14.129855 | -0.468037 | 0.066413 | 0.523296 | 17.31511 |
| V17 | -7.53E-16 | 0.849337 | -25.162799 | -0.483748 | -0.065676 | 0.399675 | 9.253526 |
| V18 | 4.33E-16 | 0.838176 | -9.498746 | -0.49885 | -0.003636 | 0.500807 | 5.041069 |
| V19 | 9.05E-16 | 0.814041 | -7.213527 | -0.456299 | 0.003735 | 0.458949 | 5.591971 |
| V20 | 5.09E-16 | 0.770925 | -54.49772 | -0.211721 | -0.062481 | 0.133041 | 39.4209 |
| V21 | 1.54E-16 | 0.734524 | -34.830382 | -0.228395 | -0.02945 | 0.186377 | 27.20284 |
| V22 | 7.96E-16 | 0.725702 | -10.933144 | -0.54235 | 0.006782 | 0.528554 | 10.50309 |
| V23 | 5.37E-16 | 0.62446 | -44.807735 | -0.161846 | -0.011193 | 0.147642 | 22.52841 |
| V24 | 4.46E-15 | 0.605647 | -2.836627 | -0.354586 | 0.040976 | 0.439527 | 4.584549 |
| V25 | 1.45E-15 | 0.521278 | -10.295397 | -0.317145 | 0.016594 | 0.350716 | 7.519589 |
| V26 | 1.70E-15 | 0.482227 | -2.604551 | -0.326984 | -0.052139 | 0.240952 | 3.517346 |
| V27 | -3.66E-16 | 0.403632 | -22.565679 | -0.07084 | 0.001342 | 0.091045 | 31.6122 |
| V28 | -1.21E-16 | 0.330083 | -15.430084 | -0.05296 | 0.011244 | 0.07828 | 33.84781 |
| Amount | 8.83E+01 | 250.120109 | 0 | 5.6 | 22 | 77.165 | 25691.16 |
| Class | 1.73E-03 | 0.041527 | 0 | 0 | 0 | 0 | 1 |

## Fig 1. Feature Engineering using Random Forest (threshold = 0.024)



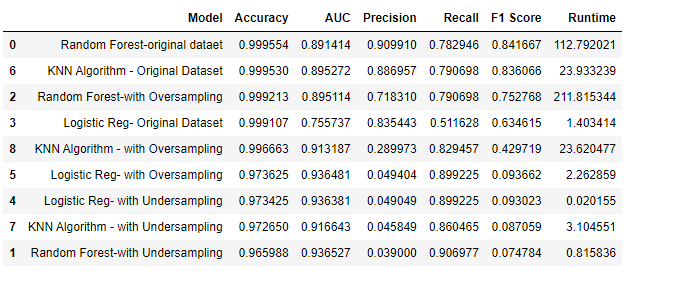
## Fig 2. Imbalance Dataset Treatment – Under sampling and Oversampling







## Table 2: Model Performance Chart

Appendix

## References:

[1] Nilson Report Issue 1164 | Nov 2019 Url : <https://nilsonreport.com/upload/content_promo/The_Nilson_Report_Issue_1164.pdf>

[2] Credit Card Frud Detection dataset | url: <https://www.kaggle.com/mlg-ulb/creditcardfraud>

[3] Machine learning for credit card fraud detection

<https://fraud-detection-handbook.github.io/fraud-detection-handbook/Chapter_2_Background/MachineLearningForFraudDetection.html>

[4] Fraud Detection using Machine Learning:

<http://cs229.stanford.edu/proj2018/report/261.pdf>

[5] Champion-challenger analysis for credit card fraud detection: Hybrid ensemble and deep learning.

<https://www-sciencedirect-com.ezproxy.lib.ryerson.ca/science/article/pii/S0957417419302167#bib0020>

[6] A cost-sensitive decision tree approach for fraud detection

<https://www-sciencedirect-com.ezproxy.lib.ryerson.ca/science/article/pii/S0957417413003072>

[7] A Data Mining Based System For Transaction Fraud Detection

<https://ieeexplore-ieee-org.ezproxy.lib.ryerson.ca/document/9342376>

[8] Comparative Evaluation of Credit Card Fraud Detection Using Machine Learning Techniques

<https://ieeexplore-ieee-org.ezproxy.lib.ryerson.ca/stamp/stamp.jsp?tp=&arnumber=8978372>

[9] <https://medium.com/analytics-vidhya/credit-card-fraud-detection-part-i-1061d4c0bd68>

[10] <https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761>

[11] <https://machinelearningmastery.com/tour-of-evaluation-metrics-for-imbalanced-classification/>

#### Link to Github repository:

<https://github.com/kamal027/CapstoneProject.git>