Credit Card Fraud Detection using Machine Learning

Kamal Sharma (Student No: 501072994)

Supervisor: Zekiye Erdem

Submission Date: 18th Oct 2021

Table of Contents

[1. Introduction/ Brief context of the theme 3](#_Toc85492802)

[2. Summary of Research 3](#_Toc85492803)

[Research Question 4](#_Toc85492804)

[3. Dataset Description 4](#_Toc85492805)

[4. Methodology/Techniques 4](#_Toc85492806)

[5. Literature Review 5](#_Toc85492807)

[I. Champion-challenger analysis for credit card fraud detection: Hybrid ensemble and deep learning 5](#_Toc85492808)

[II. A cost-sensitive decision tree approach for fraud detection 7](#_Toc85492809)

[III. A Data Mining Based System For Transaction Fraud Detection 10](#_Toc85492810)

[IV. Comparative Evaluation of Credit Card Fraud Detection Using Machine Learning Techniques 11](#_Toc85492811)

[Chart 1 : Overall Methodology 14](#_Toc85492812)

[14](#_Toc85492813)

[Table 1: Statistical Analysis 15](#_Toc85492814)

[Appendix 16](#_Toc85492815)

[References: 16](#_Toc85492816)

# 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, 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” 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. 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
3. 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.
4. 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. 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.
   2. 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.
5. Evaluation: We will evaluate the performance of our models by computing accuracy, sensitivity (recall), specificity and precision.
6. Class Imbalance: We will apply undersampling technique (Random Undersampling) 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 [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

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

**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.

**Dataset**: The dataset has been collected by a leading card issuer company in South Korea and contains off-line transactions approved overseas. This dataset presents transactions that occurred over 14 months (from April 2015 to May 2016). The one-year training data was divided by a ratio of 80:20 for validation.

**Process:** Using the same training data, two models were developed individually and then compared through off-line evaluation data and post-launch evaluation data and finally selected the one that has better performance results.

The Champion Model – Hybrid Ensemble: This model is developed by domain experts and has been applied and operated in the real-world FDS for a long time. The model is an ensemble of various machine learning models such as decision trees, logistic regressions, and shallow neural networks, all at once.

The Challenger Model – Deep Learning: In the one-year training data, the distribution of the number of transactions per card is skewed: Thus, a feed-forward neural network structure was selected. Performance of the two models is measured using off-line test set and post-launch test data and using the evaluation metrices K-S statistics, AUROC, alert rate, precision, and recall

K–S statistics: It quantifies the difference between two distributions and is defined as the maximum value of the difference between two empirical cumulative distribution functions.

AUROC: It is a plot of true positive rate (sensitivity) against the false positive rate (1- specificity) at various cut off settings

Alert Rate: It takes into account the desired cut off for the fraud score of a transaction, considering affordable level for the investigators. Alerted Transactions/All transactions.

Precision: {fraudulent transactions caught by model} / {alerted transactions}

Recall: {fraudulent transactions caught by model} / {Fraudulent transactions}

**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

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

**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 while working with classification problem. 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

**Dataset:** The data used in this study is taken from a bank’s credit card data warehouses. For a time period of 12 months, the training set have about 22 million credit card transactions. The distribution of this data with respect to being normal or fraudulent is highly skewed with a ratio of 1:22,500. Stratified sampling technique is used to under sample the legitimate records to a meaningful number

Performance comparisons of the models over the test set are done over the newly defined cost-sensitive performance metric Saved Loss

Rate (SLR) which is the saved percentage of the potential financial loss that is the sum of the available usable limits of the cards from which fraudulent transactions are committed.

**Process/Approach:** Misclassification cost of a fraudulent record is defined as the available usable limit of the card used in the transaction instead of the amount of the transaction or a predefined fixed amount of cost. Thus, each false negative has a different misclassification cost and the performance of the model should be evaluated over a newly defined cost-sensitive metric SLR (Saved Loss Rate) which is percentage of the total amount of saved available usable limits instead of the metrics based on the number of frauds detected.

SLR = {Misclassification Cost Sum of Number of Frauds detected}

----------------------------------------------------------------------------------

{Misclassification Cost Sum of Total Number of Fraud}

The new cost-sensitive decision tree algorithm selects the splitting variable of a node, if a split is possible, based on the reduction of the total misclassification cost instead of reduction of impurity. At the beginning, all the transactions in the training set are assigned to the root node of the tree. First of all, the cost of the node is calculated. Next, both the total misclassification cost in the case of assigning the transactions of the node as fraudulent (CP) and the total misclassification cost in the case of assigning the transactions as normal (CN) are calculated. To calculate CP and CN, four different methods are used: CS – Direct Cost, , CS – Gini and CS – Information Gain.

CS- Direct Cost: Instead of using an impurity measure to find the splitting variable, this method chooses the variable which enables the biggest reduction in the total misclassification cost.

CS – Class Probability: The relative frequency of the classes (class probabilities) are integrated in the cost calculation functions to add the effect of the class distributions to the node costs.

CS – Gini: the square of class probabilities are integrated in the cost calculation functions to add the effect of the class distributions to the node costs in a different way inspired from the Gini index impurity measure used in C&RT

CS – Information Gain: Negative of relative class frequencies are integrated in the cost calculation functions to add the effect of the class distributions to the node costs in a different way inspired from the information gain impurity measure used in ID3

**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.

The performance CS – Direct Cost method is not optimum and indicates that we cannot use misclassification cost without incorporating the class distribution or an impurity measure in cost calculations. However, using other cost-sensitive approaches which incorporate such an information in the cost calculations show a significant improvement in the classification performance.

## III. A Data Mining Based System For Transaction Fraud Detection

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

**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.

**Dataset**: The data of training the model is IEEE-CIS data set. The data set contains more than 1 million samples, and each sample contains more than 400 characteristic variables, including financial characteristics and nonfinancial characteristics. The dataset is first cleaned to eliminate some outliers and missing data, transformed and the statistical data such as maximum, mean and standard deviation are extracted. Then, Recursive feature elimination with cross validation (RFECV) is used to eliminate some unimportant features. Final step is to implement a classifier based on random forest to detect transaction fraud risk.

**Process/Approach**: Random forest method is used for building a classifier where each tree is generated by a random vector of independent sampling, and each tree votes to find the most popular category to classify the input. It can process a large number of inputs and determine the most important characteristics. Therefore, further feature mining is carried out on the data extracted by RFECV.

Performance of the model is compared with other classical machine learning models like support vector machine and logistic regression by using accuracy and AUC ROC score as the evaluation metric.  AUC ROC score is the area under the receiver operating characteristic curve, which is created by drawing the relationship between true positive rate (TPR) and false positive rate (FPR) under different threshold settings.

TPR=TP/ TP+FN

FPR=FP/ FP+TN

**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

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

**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.

**Dataset:** Dataset used in this research comes from Kaggle. This dataset presents 3075 transactions with 12 features of transactions in CSV file. The features contain the average amount of transaction per day, transaction amount, if declined or not, foreign transaction or not, if it’s of high risk, and six-month average balance in the dataset. The class label ‘is fraudulent’ is Y or N for fraud and legal transactions respectively.

**Process/Approach:** In the learning phase a classifier system is created and supplied with the extracted information. In Data preprocessing categorical values are converted to integers, class feature is converted from Y/N to binary values 0s and 1s. 80 % of the data is used for training set and 20% for test set. Because of high imbalance in the class variable a range is assigned to fraudulent and non fraudulent class in training and test data. These four machine learning techniques are used to train the model.

Logistic Regression: Both logistic regression function and sigmoid function are used to carry out binary classification with a threshold of 0.5.

K- nearest neighbour (KNN) : K-nearest neighbor algorithm is used to predict the attributes of an informational point to other points based on its relative position using Euclidean distance measure.

Naïve Bayes: Classification is based on probability estimates from known values and known probabilities. It is a supervised machine learning algorithm which is represented by

P (A|B) = {P(B|A). P(A)} / P(B)

It calculates the posterior likelihood P (A|B), the likelihood of outcome (A) provided certain conditions (B).

Support Vector Machine: The algorithm creates a line or a hyperplane which separates the data into classes Fraudulent and non-Fraudulent.

Performance of the models is evaluated and compared based on accuracy, sensitivity (recall), specificity and precision.

**Results/Conclusion:**

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.

## Chart 1 : Overall Methodology

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

1. Import Dataset

* + Analyze and visualize dataset variables using statistical techniques
  + Find missing values.
  + Find dependencies.

2. Data Exploration

## 

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

3. Data Manipulation/ Pre processing

## 

* + Split dataset into train-test sets.
  + Build Regression model using training set
  + Build Random forest classifier model using the same training set.

4. Data Modeling

* + Use model to predict values for test set.
  + Calculate confusion matrix and evaluate model performance based on Accuracy, sensitivity (recall), specificity and precision.

5. Evaluation

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

6. Treat Class Imbalance

* + Compare results for the two models under original and modified dataset.
  + Identify best performance based on evaluation metrices.
  + Evaluate if Under/Oversampling techniques affect the performance of the model

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 |

# Appendix

## References:

Credit Card Frud Detection dataset | url:

<https://www.kaggle.com/mlg-ulb/creditcardfraud>

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

Machine learning for credit card fraud detection

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

Fraud Detection using Machine Learning:

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

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>

A cost-sensitive decision tree approach for fraud detection

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

A Data Mining Based System For Transaction Fraud Detection

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

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>

#### Link to Github repository:

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