Online Credit Card Transactions Fraud Detection

A CASE STUDY REPORT

15CSE481 - MACHINE LEARNING AND DATA MINING

Submitted by

G Sai Shanthan Reddy (CB.EN.U4CSE18221)

A Sai Tharun (CB.EN.U4CSE18250)

BTECH CSE

Department of Computer Science and Engineering

Amrita School of Engineering

Coimbatore-112

November 2021

Table of contents

Note :- Clicking the page number will bring you to the Tables of Contents page

Abstract

Credit card fraud detection is presently the most frequently occurring problem in the present world. This is due to the rise in both online transactions and e-commerce platforms.

Credit card fraud generally happens when the card was stolen for any of the unauthorized purposes or even when the fraudster uses the credit card information for his use. In the present world, we are facing a lot of credit card problems.

To detect the fraudulent activities the credit card fraud detection system was introduced. This project aims to focus mainly on machine learning algorithms. The algorithms used are the random forest algorithm and the Naïve Bayes algorithm.

The Random Forest and the Naïve Bayes algorithms are compared and the algorithm that has the greatest accuracy and precision is considered as the best algorithm that is used to detect the fraud.

Introduction

As we are moving towards the digital world — cybersecurity is becoming a crucial part of our life. When we talk about security in digital life then the main challenge is to find the abnormal activity.

When we make any transaction while purchasing any product online — a good amount of people prefer credit cards. The credit limit in credit cards sometimes helps us make purchases even if we don't have the amount at that time. but, on the other hand, these features are misused by cyber attackers.

To tackle this problem we need a system that can abort the transaction if it finds fishy.

Here, comes the need for a system that can track the pattern of all the transactions and if any pattern is abnormal then the transaction should be aborted.

Today, we have many machine learning algorithms that can help us classify abnormal transactions. The only requirement is the past data and the suitable algorithm that can fit our data in a better form.

Related Work

Literature survey

https://thesai.org/Downloads/Volume11No12/Paper_65-Fraud_Detection_in_Credit_Cards.pdf

Due to the increasing number of customers as well as the increasing number of companies that use credit cards for ending financial transactions, the number of fraud cases has increased dramatically. Dealing with noisy and imbalanced data, as well as with outliers, has accentuated this problem. In this work, fraud detection using artificial intelligence is proposed. The proposed system uses logistic regression to build the classifier to prevent frauds in credit card transactions. To handle dirty data and to ensure a high degree of detection accuracy, a pre-processing step is used.

Credit Card Fraud Detection: Top ML Solutions in 2021 (spd.group)

Credit Card Fraud Detection with Machine Learning is a process of data investigation by a Data Science team and the development of a model that will provide the best results in revealing and preventing fraudulent transactions. This is achieved through bringing together all meaningful features of card users' transactions, such as Date, User Zone, Product Category, Amount, Provider, Client's Behavioral Patterns, etc. The information is then run through a subtly trained model that finds patterns and rules so that it can classify whether a transaction is fraudulent or is legitimate.

Credit Card Fraud Detection. Machine Learning Models and Deep Neural... | by Luke Sun | Towards Data Science

A deep neural network and two machine learning models are built to tackle the challenge and different model performances are compared along with data sampling techniques that can be implemented to improve the model.

Dataset Description

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 94682 transactions.

It contains only numeric input variables which are the result of Principal Component Analysis (PCA) transformation. Unfortunately, due to confidentiality issues, we cannot provide the original features and more background information about the data.

Column	Description
DOMAIN	The domain name of the customer's email address that was used for the transaction (Masked)
STATE	The state code of the customer's location.
ZIPCODE	The zip code of the customer's location.
TIME1	Hour feature #1 of the transaction.
TIME2 VIS1	Hour feature #2 of the transaction. Anonymized feature #1 for feature VIS.
VIS2	Anonymized feature #2 for feature VIS.
XRN1	Anonymized feature #1 for feature XRN.

XRN2	Anonymized feature #2 for feature XRN.
XRN3	Anonymized feature #3 for feature XRN.
XRN4	Anonymized feature #4 for feature XRN.
XRN5	Anonymized feature #5 for feature XRN.
VAR1	Anonymized feature #1 for feature VAR.
VAR2	Anonymized feature #2 for feature VAR.
VAR3	Anonymized feature #3 for feature VAR.
VAR4	Anonymized feature #4 for feature VAR.
VAR5	Anonymized feature #5 for feature VAR.
TRN_AMT	The transaction amount.
TOTAL_TRN_AMT	The total transaction amount.
TRN_TYPE	The type of transaction whether FRAUD or LEGIT.

Problem Statement

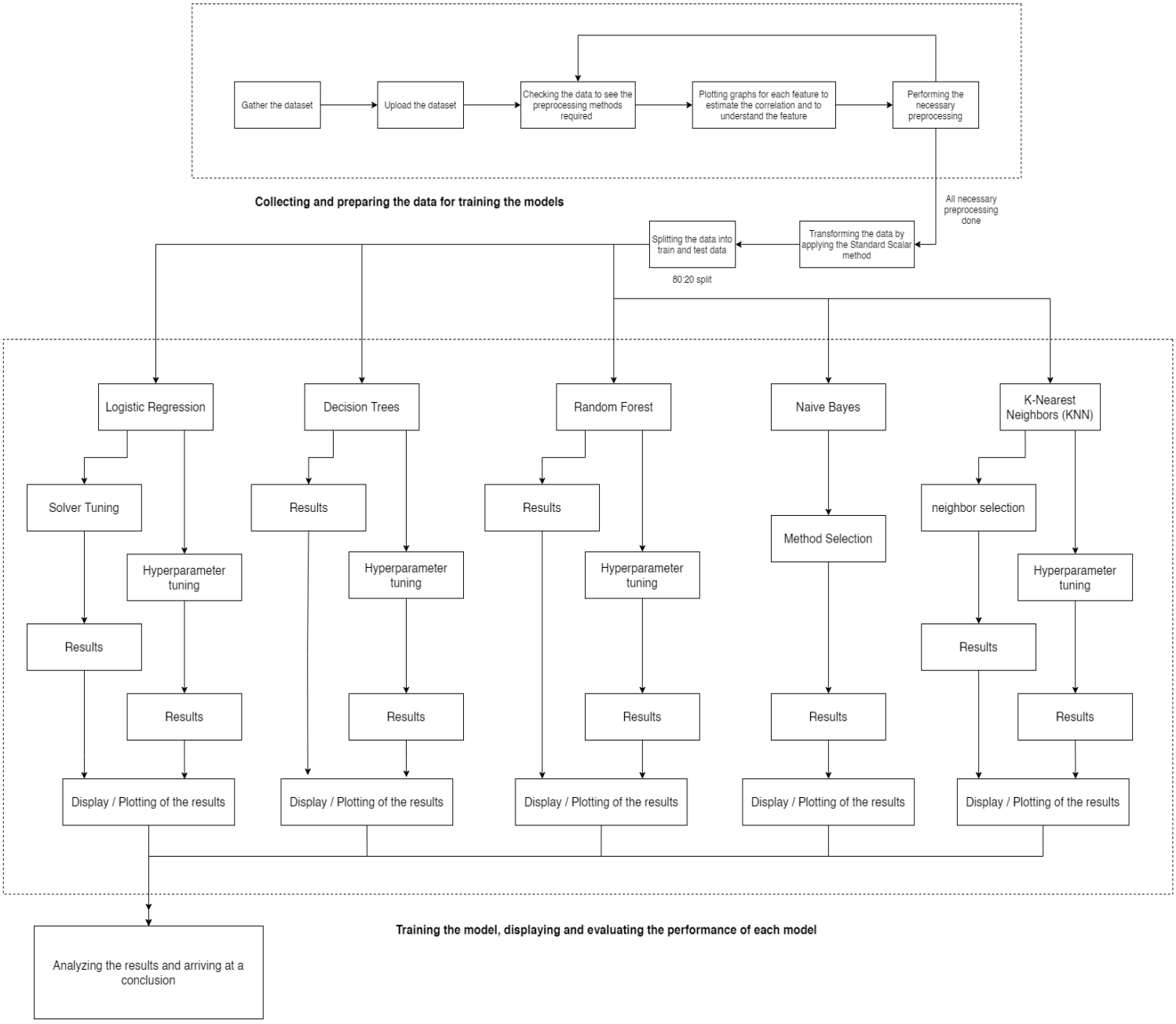
The Credit Card Fraud Detection Problem includes modeling past credit card transactions with the knowledge of the ones that turned out to be a fraud. This model is then used to identify whether a new transaction is fraudulent or not.

a. Motivation

Credit card frauds happen mainly due to customer negligence on which the credit card companies have no control over. But the credit card companies do have control over the transactions. With rising usage of online transactions and hacker attacks, acquiring knowledge and finding patterns of fraudulent transactions has become very important in order to stop such transactions from happening and in turn provide a sense of security to their customers.

Work Done

a. Flow Diagram



Flow Diagram showing 5 different ML models

b. Preprocessing and data preparation

- Understanding DataSet
- Categorical Features of Dataset
- Checking for null values
- Duplication Rate & Completeness Ratio
- Checking duplicate rate
- Checking no of unique values per feature
- Dropping excess column
- Transforming the data by performing a combination of oversampling and undersampling

c. Models

- Logistic Regression
  - Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. Some of the examples of classification problems are Email spam or not spam, Online transactions Fraud or not Fraud, Tumor Malignant or Benign. Logistic regression transforms its output using the logistic sigmoid function to return a probability value.

- Decision Trees
  - The Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving **regression and classification problems** too.
  - The goal of using a Decision Tree is to create a training model that can be used to predict the class or value of the target variable by **learning simple decision rules** inferred from prior data(training data).
  - In Decision Trees, for predicting a class label for a record we start from the **root** of the tree. We compare the values of the root attribute with the record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

- Random Forest Classification
  - Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of **ensemble learning,** which is a process of *combining multiple classifiers to solve a complex problem and to improve the performance of the model.* ***"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."*** Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.
  - **The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.**

- Naïve Bayes Classification
  - It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.
  - For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as 'Naive'.
  - The Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

- K-Nearest Neighbor (KNN)
  - K-NN is a non-parametric algorithm that assumes the similarity between the new case/data and available cases and puts the new case into the category that is most similar to the available categories.
  - KNN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suited category by using K- NN algorithm.
  - It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.

Analysis and Inference

1. We can infer that since the dataset in question is huge (104161 records) and the testing data is 20%, i.e 2083 records and the dataset is heavily imbalanced towards ‘LEGIT’ class, the metric accuracy is not going to be a correct metric to measure the performance of the model.
2. Instead we use the metrics F1 score (a measure of a model's accuracy on a dataset), Area under Receiver Operating Curve (AUROC), Area under Precision-Recall curve (AUPRC) to measure the performance of the models.
3. For each model we are first analysing the performance metrics with the default parameters and then are using GridSearchCV to tune the hyperparameters to ensure that we find the best parameters for each model.
4. We are then analysing the performance of the models built using the best parameters obtained from hyperparameter tuning.
5. From the results we arrive at some interesting conclusions like which is the best model, proving the hypothesis that tree based models tend to overfit on highly imbalanced datasets, KNN with n_neighbours=1 is the best algorithm to train our model.

Conclusion

The problem for which we are trying to build a Machine Learning model is a classification problem with the aim being to classify if a credit card transaction is ‘Legit’ or ‘Fraud’. Here we have done a deep analysis on the various features given in the dataset to remove/clean and use the proper ones directly for training the models. We also used various machine learning algorithms to train the model, while also tuning the hyperparameters using GridSearchCV to arrive at the best possible parameters for each of the models and have also done a deep analysis on which algorithm is the best to train the model. The best algorithm was K-Nearest Neighbor (KNN) Classifier while the tree algorithms were overfitting, which is attributed to the heavy imbalance in the dataset towards the ‘Legit’ class.

Thus, we were able to train 5 models, tune their hyperparameters and then arrive at the best algorithm to train the model, which was the K-Nearest Neighbor (KNN) Classifier.

References

https://thesai.org/Downloads/Volume11No12/Paper_65-Fraud_Detection_in_Credit_Cards.pdf

https://scikit-learn.org/stable/modules/naive_bayes.html

https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

https://datascience.stackexchange.com/questions/35713/i-got-100-accuracy-on-my-test-set-is-there-something-wrong

https://analyticsindiamag.com/guide-to-hyperparameters-tuning-using-gridsearchcv-and-randomizedsearchcv/

https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

https://medium.com/analytics-vidhya/credit-card-fraud-detection-logistic-regression-121d2dd35e2d

https://towardsdatascience.com/intuition-behind-log-loss-score-4e0c9979680a