CREDIT CARD FRAUD DETECTION

Abstract

This document describes Capstone proposal as part of Udacity Machine Learning Engineer Nandodegree course for building machine learning system to detect Credit Card Fraud

Contents

CREDIT CARD FRAUD DETECTION

DOMAIN BACKGROUND

Every year Trillions and Trillions of dollars of transactions are being done using credit cards thru online/regular purchase mode. As per **Nilson report[1]**, Global Card Fraud losses reached $21.84 Billion in year 2015 and will be close to $32 Billion in 2020. Hence it is extremely important that credit card fraud is detected on time which helps in stopping transaction and minimize financial loss. There has been extensive academic/industry research **(For e.g. Andrea Dal Pozzolo et al., 2015[2][3])** on this subject on using different Machine learning techniques to detect credit card fraud

PROBLEM STATEMENT

As described in domain background, we need to find if credit card transaction is fraudulent or non-fraudulent. As far as machine learning is concerned, it is supervised binary classification problem (as we have labelled dataset) where we need to flag if transaction is normal/ (Negative) / fraudulent (positive). Additional thing we need to consider is, it is unbalanced classification problem (Anomaly detection problem) where typically 99% of transactions are non-fraudulent one

DATASETS AND INPUTS

For this project – Dataset (Reference - https :// www.kaggle.com /dalpozz / creditcard_fraud)) collected and analysed during a research collaboration of Worldline and the Machine Learning Group (http://mlg.ulb.ac.be/ARTML) of ULB (Université Libre de Bruxelles) on big data mining and fraud detection will be considered(It is public dataset).

The datasets contain transactions made by credit cards in 1 month by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset 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. Due to confidentiality issues, original features are not provided. 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 dataset. The feature 'Amount' is the transaction Amount. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

SOLUTION STATEMENT

As this is unbalanced classification problem (Anomaly/fraud detection) we would require complex model to get correct predictions. Typically, complex models have high variance and a low bias while simple models have low variance and a high bias. In order to overcome variance/bias problem we can consider ensembles of models. Several works have found that ensembles of models very often outperform a single model (e.g., [4]), because averaging multiple models often reduces the variance of single models.

Well-known ensemble methods are bagging [5] and boosting [6]. So, Random forest/ Adaboost/ XGBOOST classification algorithms will be considered for this problem. The model giving best score (Recall/Precision/F1-Score) will finalized as solution.

Additionally, to counter unbalanced data, we will use SMOTE (Synthetic Minority Over -sampling Technique**[9]**). SMOTE is an oversampling method. It works by creating synthetic samples from the minor class instead of creating copies, this will help algorithm getting additional training data with positive examples

BENCHMARK MODEL

For this particular data set we see there are 492 frauds out of 284,807 transactions i.e 99.83% is non-fraudulent (Negative/0 class) and only 0.17% is fraudulent (positive /1 class). As data is heavily skewed towards negative class, any random classifier will tend to classify everything as negative and not predict any data as positive. So, we can consider below as benchmark –

1) Accuracy score – 99.83% ((284,807-492(Positive class))/ 284,807)
2) Precision Score – 0.00 (As random classifier will not predict anything as positive)
3) Recall score – 0.00 (As random classifier will not predict anything as positive)
4) F1-Score - – 0.00 (As random classifier will not predict anything as positive)
5) Area under precision recall curve – 0.5 (Typical Random score)

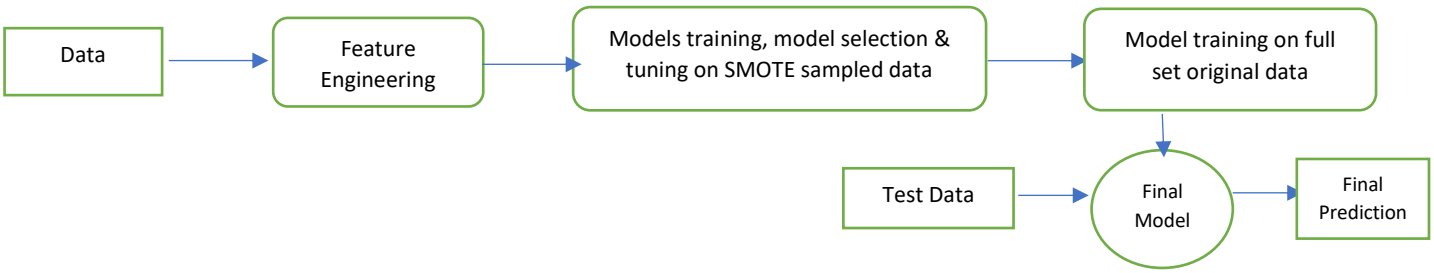
EVALUATION METRICS

Below metrics will be used to evaluate classification model (Considering unbalanced data)–

1. Accuracy: (True Positives + True Negatives)/ # of samples
2. Precision Score: (True Positives)/(True Positives + False Positives)
3. Recall score: (True Positives)/(True Positives + False Negatives)
4. F1-Score – (2 * True Positives)/(2 * True Positives+ False Positives+ False Negative) Area under precision recall curve – To use Sklearn.metrics.auc(Calculates using trapezoidal rule)**[8]**

PROJECT DESIGN

Project will follow typical machine learning pipeline –



1) **Feature Engineering –** Public data set for credit card as mentioned in **DATASET and INPUT section** will be used for training purpose. To begin with Feature engineering(normalization) will be done on amount column alone as rest of columns (v1 to v28) are already in normalized state and time column will be dropped as it more of sequence number

Data will be split into train and test sets with 70% to 30% ratio with proportionate classes present in training and test set

2) **Models Training & model selection on SMOTE sampled data**– As mentioned in solution section, Random forest/ Adaboost / XGBOOST classification algorithms will be trained on sampling data obtained using under SMOTE (Synthetic Minority Over-Sampling Technique) technique. Model giving best score will be selected.

3)**Model Tuning & Selection of hyper parameters on SMOTE sampled data -**Model selected in previous step will be tuned using sklearn. GridSearchCV**[7]** method and best hyper paratmeters values are identified (e.g. n_estimators, max_features, max_depth)

4)**Model Training & Prediction on original data set**– Finally above selected model will be trained on full data set using hyperparameters selected in above step and prediction will be done on test data and relevant scores (Refer Evaluation metrics) will be published

REFERENCE

1) https://www.nilsonreport.com/index.php
2) Andrea Dal Pozzolo, Giacomo Boracchi, Olivier Caelen, Cesare Alippi, and Gianluca Bontempi. Credit Card Fraud Detection with Alert-Feedback Interaction. Submitted to IEEE Transactions on Neural Networks and Learning Systems.
3) Andrea Dal Pozzolo, Olivier Caelen, Yann-Ael Le Borgne, Serge Waterschoot, and Gianluca Bontempi. Learned lessons in credit card fraud detection from a practitioner perspective. Expert Systems with Applications, 41(10):4915-4928, 2014.
4) Eric Bauer and Ron Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine learning, 36(1-2):105–139, 1999.
5) Leo Breiman. Bagging predictors. Machine learning, 24(2):123–140, 1996.
6) Robert E Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. Annals of statistics, pages 1651–1686, 1998.
7) http://scikit-learn.org /stable /modules /generated / sklearn.model_selection.GridSearchCV .html
8) http://scikit-learn.org/stable/modules/generated/sklearn.metrics.auc.html
9) http://www.jair.org/papers/paper953.html