Machine Learning Engineer Nanodegree

Capstone Project: Credit Card Fraud Detection Model

Akbarali Shaikh July 19, 2020

Table of Content

1. Project Overview

The banking industry works on credit and income from the interest which can be in a form of mortgage, other types of loan, or from Credit Card. As the cost of having credit cards reduced it got broadly accepted in society and many people started having a credit card. As the credit card reached the masses, new ways of fraud also came to existence. Today, CC fraud is a very big problem for the banking industry and it is plaguing financial industries for years, some countries it is less problematic than others

Solving this problem could bring about a reduced volume of fraudulent transactions in financial industries hereby saving them a huge volume of money lost and this problem can be avoided by using predictive analytics where machine learning algorithms are used to detect fraud patterns and determine future probabilities and trends.

2. Problem Statement

Each record has multiple columns and in our dataset, there are 31 columns including classification column which confirms if the record is genuine or a fraud. In real-time when the record is asked to be classified it is not realistically with a human eye to do classification.

The objective is to create an ML model that can classify the transaction in real-time with zero human intervention. In the real world, we have higher genuine transactions than fraudulent so is a case with our dataset.

3. Datasets and Inputs

The dataset used for this project is hosted on data.world. Dataset is normalized and all columns names are changed before it is shared (due to confidentiality). All features provided would be used in building the model.

The dataset contains 31 numerical features including record classification column means classifying the record as genuine or

fraudulent. The first 28 features are labeled V1, V2 to V28, and these are assumed derived from principal components obtained with Principal Components Analysis of the raw/original data.

4. Solution Statement

Dataset is unbalanced with more than 99% of the transaction been genuine vs fraudulent transaction. If we run the ML model on as is data it is the highly likely model will be biased.

To solve this problem, first, we will baseline the model on as is data then we will implement SMOTE function which will create more fraudulent data on train data set to train the model, post which we will implement min-max scaler to normalize the data and so the entire dataset is on the same scaler. Post which we will apply naive_predictor_accuracy assuming all entire data set is zero then what is a baseline accuracy of the model.

Once we have a baseline and cleaned data we will run a model on the model on different machine learning algorithms and it will be tested on validation set data. Post which we use hyper meter tuning on the best-chosen model.

These algorithms include Naïve Bayes Classifier, SVC, GaussianNB, LogisticRegression, KNeighborsClassifier, DecisionTreeClassifier, RandomForestClassifier, GradientBoostingClassifier, BaggingClassifier, ExtraTreesClassifier, XGBClassifier and AdaBoostClassifier.

5. Benchmark Model

Checked the details at scores but there are limited to know benchmark details but similar work is done as part of academics and competition and looks like, Accuracy, and F1 score are 2 important metrics frequently used in the classification of fraudulent transaction

6. Evaluation Metrics

In binary classification model, accuracy with minimal type 1 and 2 error is an important metric.

Type I error also referred to as False Positive: model predicts positive when it isn't.
Type II error also referred to as False Negative: model predicts negative when it isn't.

Accuracy is an important measure of a binary classification model, it shows how accurate the model predicts true and false. In our case, we have a highly imbalanced dataset, other evaluation metrics consider such class imbalance:

1. Precision
2. Recall

Recall measures the fraction of fraudulent transactions that are correctly predicted while measures that fraction of cases predicted to be fraudulent that are truly fraudulent.

Precision and Recall are defined as follows:

Recall: When the focus in catching all fraudulent transactions even in case if few genuine transactions are also categorized as fraudulent then this is a go-to metric. When the cost of failures is high, we want to recall the rate to be high.

Precision: How accurate the model performs e.g., correctly categorizing true as true. Ideally. When raising false alerts is high we want to have high precision

F beta score: It combines precision and recalls into one metric. The higher the score the better model. Value high than 1 means more emphasis on Recall as compared to Precision and if less than 1 it is vice versa.

F1 score (beta=1) It's the mean of precision and recall

F2 score (beta=2) It's a metric that combines precision and recall, putting 2x emphasis on recall - consider using it when recalling positive observations (fraudulent transactions) is more important than being precise.

7. Project Design

As we start with the project we will follow the following steps to choose the right ML model for our problem statement

1. We will load data from source (data.world)
2. Post loading in a data frame, we need to familiarize ourself with a dataset
3. Visualize the data
4. We will then start with ML
    1. We will start with importing relevant libraries for ML
    2. Splitting the data into train/validation/test sets
    3. Baselining the model with Naive Predictor - assuming if all predicting value is of one category and also on Logistic regression model
    4. Since we now have a baseline, we will try improving the model by
        - Balancing the dataset via SMOTE function and
        - Normalizing the dataset
        - **Note**: this will be applied only on the test dataset
    5. Now we will run the model on the different algorithm (we have a binary classification problem statement), hence we will try the following models
        - Logistic Regression
        - KNN Classifier
        - Decision Tree Classifier
        - Random Forest Classifier
        - Gradient Boosting Classifier
        - GaussianNB
        - SVC
        - BaggingClassifier
        - ExtraTreesClassifier
        - XGBClassifier
        - AdaBoostClassifier
    6. Apply Hyper-parameter tuning on top models

- Observation: Since we have an Imbalanced dataset we need to not only check for accuracy but also for Fi Score.