Machine Learning Engineer Nanodegree

Capstone Proposal

Bharath Kumar Loganathan

August 10th, 2017

Credit Card Fraud Detection

Domain Background

Credit Card Fraud is defined as when an individual uses another individual's credit card for personal reasons. Credit card fraud has been divided into two types: Offline fraud and On-line fraud. Offline fraud is committed by using a stolen physical card at call center or any other place. On-line fraud is committed via internet, phone, shopping, web, or in absence of card holder.

In 1999, out of 12 billion transactions made annually, approximately 10 million—or one out of every 1200 transactions—turned out to be fraudulent. Also, 0.04% (4 out of every 10,000) of all monthly active accounts was fraudulent. As fraudsters are increasing day by day, it has become important for banks to solve this problem. I have personal interest in solving fraudulent problems in financial domain.

Problem Statement

Credit Card Fraud detection involves identifying scarce fraud activities among numerous legitimate transactions as quickly as possible. The number of fraudulent transactions is usually a very low fraction of the total transactions. Hence the task of detecting fraud transactions in an accurate and efficient manner is fairly difficult and challengeable. Therefore, development of efficient methods which can distinguish rare fraud activities from billions of legitimate transaction seems essential.

Datasets and Inputs

The datasets 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. 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. Unfortunately, due to confidentiality issues, original features and more background information about the data were not provided. This data set will be downloaded from Kaggle.

Solution Statement

The data set is labeled and to predict whether a transaction can be categorized into normal or fraud is what we want to know at the end. So, this is a supervised binary classification problem. The data set is imbalanced, so we have to develop a model which can accommodate this imbalanced dataset. I am going to try the following 3 methods as a solution. 1] Resampling the data which includes oversampling the minority class and undersampling the majority class. 2] Using Cost-sensitive algorithms 3] Use Tree Algorithms.

Benchmark Model

All the methods described above in "Solution Statement" will be tried and analyzed based upon the performance metrics such as fbeta_score, time, recall and precision scores. Then the best model will be chosen as the benchmark model and it will be optimized to further improve the performance and AUC-PR curve will be analyzed in the finally in this model.

Evaluation Metrics

Given the class imbalance ratio, we will measure the accuracy using the Area Under the Precision-Recall Curve (AUPRC). The precision-recall plot is a model-wide evaluation measure that is based on two basic evaluation measures – recall and precision.
The fbeta-score can be interpreted as a weighted average of the precision and recall, so fbeta-score will be a very useful evaluation of the model for this data set. fbeta-score reaches its best value at 1 and worst at 0. The beta parameter determines the weight of precision in the combined score. beta < 1 lends more weight to precision, while beta > 1 favors recall. I will be using beta value of 0.5 to give emphasis on precision as we don't want to misclassify the normal customer as fraud or vice versa. The formula of fbeta_score is

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}.$$

Project Design

Project work flow will be as of below steps:

1] Data Exploration: The data set is already pre-processed with PCA transformation. The only features that are not pre-processed are 'Time', 'Amount' and target variable. So, we will explore its relation to the data set.

2] Data Visualization: We will see how the data set is imbalanced, i.e., how the feature 'Class' which is the response variable is distributed.

3] Data Pre-processing: If required we will normalize the features 'Time' and 'Amount' after exploring its relationship with the dataset.

4] Resample: This is first way of handling the imbalanced data set. Data will be undersampled and oversampled .After that, models will trained separately for both the samples.

5] Tree Algorithms: The data will be trained using the DecisionTree Classifier algorithm.

6] Cost-Effective Algorithm: SVM Classifier with penalized weights will be learned using the data set.

7] Performance metrics of all the models described in the step 4, 5 & 6 above will evaluated and analyzed. Finally model with best metric will be chosen as the benchmark model.

8] This model will be optimized using hyperparameter tuning to yield the best performance.

9] The AUC- Precision-Recall curve will plotted for this model and analyzed.