Machine Learning Engineer Nanodegree
Capstone Proposal

Anshul Sharma
June 8th 2019
Credit Card Fraud Detection

Domain Background

Credit card fraud is a wide-ranging term for theft and fraud committed using or involving a payment card, such as a credit card or debit card, as a fraudulent source of funds in a transaction. The purpose may be to obtain goods without paying or to obtain unauthorized funds from an account. Credit card fraud is also an adjunct to identity theft.
Although incidences of credit card fraud are limited to about 0.1% of all card transactions, they have resulted in huge financial losses as the fraudulent transactions have been large value transactions. It is important that credit card companies are able to recognize fraudulent credit card transactions so that customers are not charged for items that they did not purchase. What we need is an algorithm, which could classify a transaction as fraudulent or non-fraudulent. Doing so will benefit both the credit card companies and the customers who have to go through the ordeal.

Problem Statement

This is a binary classification problem. Inputs are the 30 features of the dataset and the goal is to predict whether the transaction is fraudulent. I will be tackling this as a classification problem. Then I plan to use a deep neural network to train the dataset. The features are pretty well drawn out, will have to see if any preprocessing is required . The target here is either “fradulaent” or “non-fraudulent” transaction.

Datasets and Inputs

The dataset is provided by Kaggle and 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, the original features and more background information about the data could not be 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, this feature can be used for example-dependant cost-senstive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

Solution Statement

The solution will be predictions of either fradulent or non-fraudulent transaction. First I will do some preprocessing to clean the data and then do some visualization of the data to get some understanding. Then I will study the realtions between the features and see if any of the features could be excluded. For training models, I will compare Deep Neural Network with various classifier algorithms since this is a classification problem. Finally, I will select the best model for this problem and fine tune parameters to get the best accuracy.

Benchmark Model

For this problem, the benchmark model will be a Deep Neural Network. I will try to beat its performance with other classifiers .

Evaluation Metrics

Given the class imbalance ratio, I would be using metrics such as Precision, Recall, Area Under the Precision-Recall Curve (AUPRC), ROC curve, e.t.c. to measure the accuracy. Confusion matrix accuracy is not meaningful for unbalanced classification because if out of millions of transactions I'm not able to detect 1000 fraudulent transactions and there are only these many fraudulent transactions then also the accuracy of the model would be great although the model is doing absolutely nothing. In order to evaluate the model, I'll be using a deep neural network as my benchmark model and compare it with other classifiers based on the metrics mentioned above. My main is to determine each and every fraudulent transaction with the least error.

Project Design

Before even start training models, I will first take a glimpse of the data to see what the shape and is and how it is formatted. Then I'll perform preprocessing on the data to clean it by removing the null values and make sure all the features are scaled properly so that it doesn't hamper the training process. Since in this case, the features have already gone using PCA, I don't think PCA 4 feature selection is required. I may perform some graph visualization for better understanding of the data distribution. This depends on whether I can find such an existing implementation/library or whether I have enough time to do it from scratch. To train models, I plan to choose 3-4 different models to compare. Because this is a classification problem, a few approaches in my head would be a deep neural network, decision trees, SVM, KNN, and random forest. Using cross-validation I can find which model performs best, and then use that one to tweak relative parameters. I expect to spend 40% of the time on data cleaning part and 60% of the time on training models and tweaking parameters. The final accuracy will be calculated against the test data set provided by Kaggle.

Reference

1. https://www.kaggle.com/mlg-ulb/creditcardfraud
2. https://en.wikipedia.org/wiki/Credit_card_fraud
3. https://keras.io/
4. https://scikit-learn.org/stable/