Machine Learning Engineer Nanodegree

Capstone Proposal

Robert Heyse
September 1st, 2019

## Proposal

### Domain Background

Fraud is causing companies, and private persons, to lose billions of dollars, with the amount of companies exposed to fraud increasing continuously (https://en.wikipedia.org/wiki/Data_analysis_techniques_for_fraud_detection). Fraud prevention is one important measure to combat this problem, however this might fail since fraudsters can adapt to different prevention systems. Therefore, methods to detect fraud once prevention has failed are necessary. Machine learning is an effective technology to detect fraudulent activities such as money laundering, credit card fraud, computer intrusion, or online payment fraud (http://metalab.uniten.edu.my/~abdrahim/ntl/Statistical%20Fraud%20Detection%20A%20Review.pdf). In this project, I will focus on detecting the latter.

Statistical fraud detection methods can be distinguished between supervised and unsupervised techniques. In supervised methods, labelled data of fraudulent and non-fraudulent transactions are used to build and train a model that predicts whether a given transaction is fraudulent. Unsupervised methods, on the other hand, seek accounts, customers or transactions that are dissimilar from the norm and flag them for closer inspection. Some techniques that have been historically applied to detect fraud range from linear discriminant analysis, to tree-based algorithms, to neural networks (http://metalab.uniten.edu.my/~abdrahim/ntl/Statistical%20Fraud%20Detection%20A%20Review.pdf).

### Problem Statement

Fraud detection in online payment transactions is a key problem for payment service providers. The ambition of payment service providers when implementing a fraud detection system is to identify and flag all fraudulent transactions, while keeping the false positive rate low. By flagging all fraudulent transactions, businesses will minimize their fraud loss and increase their revenue. A high false positive rate would have a negative effect on customer experience, as transactions might be blocked or an additional effort to verify a transaction might be required (https://www.kaggle.com/c/ieee-fraud-detection/overview/description).

### Datasets and Inputs

The dataset for this project is obtained from the IEEE-CIS Fraud Detection Competition on Kaggle (https://www.kaggle.com/c/ieee-fraud-detection/data). The data for the competition has been provided by Vesta, one of the leading e-commerce payment solution providers, guaranteeing more than $18B in transactions annually. It comes from real-world e-commerce transactions and contains a wide range of features from device type to product features.

The data is broken into two files, `identity` and `transaction`, which are joined by `TransactionID`. Not all transactions have corresponding identity information.

The following features are part of the `transaction` table (https://www.kaggle.com/c/ieee-fraud-detection/discussion/101203):

- TransactionDT: timedelta from a given reference datetime (not an actual timestamp)
- TransactionAMT: transaction payment amount in USD
- ProductCD: product code, the product for each transaction
- card1 - card6: payment card information, such as card type, card category, issue bank, country, etc.
- addr: address
- dist: distance
- P_ and (R__) emaildomain: purchaser and recipient email domain
- C1-C14: counting, such as how many addresses are found to be associated with the payment card, etc. The actual meaning is masked.
- D1-D15: timedelta, such as days between previous transaction, etc.
- M1-M9: match, such as names on card and address, etc.
- Vxxx: Vesta engineered rich features, including ranking, counting, and other entity relations.

Categorical Features:

- ProductCD
- card1 - card6
- addr1, addr2
- Pemaildomain Remaildomain
- M1 - M9

In the `identity` table, network connection information, and digital signature associated with transactions are found. They are collected by Vesta's fraud detection system and digital security partners. The following categorical features are part of the `identity` table (https://www.kaggle.com/c/ieee-fraud-detection/discussion/101203):

- DeviceType
- DeviceInfo
- id12 - id38

The transaction data is labelled with the variable `isFraud`, indicating whether a transaction has been reported as a fraud or not. I will use the dataset to build a supervised learning model that predicts the probability whether a given online transaction is fraudulent.

### Solution Statement

To solve the problem described above, I will apply supervised learning algorithms on the dataset at hand. The labeled data in the training set will serve to train the algorithm. The input of the algorithm will be a combination of the features described above. It might be necessary to modify certain features, to combine or to split them. The output of the algorithm will be the probability that a given online transaction is fraudulent.

I will use an evaluation metric to measure the performance of different algorithms against each other.

### Benchmark Model

There is a wide range of supervised learning algorithms that can be applied for fraud detection purposes (https://arxiv.org/pdf/1611.06439.pdf). One popular method that is being applied in the field are decision trees. Decision trees are easy to interpret, they are able to handle categorical features, do not require feature scaling, and are able to capture non-linearities and feature interactions . Tree ensemble methods, such as random forests, are often among the top performing algorithms for classification tasks (https://pages.databricks.com/rs/094-YMS-629/images/financial-fraud-detection-decision-tree.html)). Decision trees and their results are clearly measurable. Their output can be categorical or probabilistic. Their performance can be measured with the evaluation metric described below.

### Evaluation Metrics

It is important to implement a machine learning model that correctly identifies fraudulent transactions, while keeping the false positive rate low. The evaluation metric will therefore be the area under the ROC curve (AUROC), which will take both the true positive rate and the false positive rate into account.
(https://en.wikipedia.org/wiki/Receiver_operating_characteristic).

When plotting the ROC curve, the true positive rate is plotted on the y axis against the false positive rate on the x axis. The true positive rate in this problem is the percentage of fraudulent transactions that have been correctly identified as such. The false positive rate in this problem is the percentage of non-fraudulent transactions that have been identified as fraudulent. A diagonal ROC curve represents a random process, and would be equivalent to an AUROC of 0.5. The maximum achievable AUROC would be 1. I will keep this in mind when interpreting the evaluation results.

### Project Design

The workflow structure I will employ when solving this problem is as follows:

1. Data loading and overview: I will get a first glance at the data, check how many columns there are, check for missing values, ...
2. Data exploration: I will analyze different features and their distributions
3. Feature engineering: I might chose to engineer new features or change the existing ones
4. Data preparation: I need to bring the data to the right input format for modelling. I might chose to drop features.
5. Model training: I will train different models on the data
6. Prediction: I will use the trained models to do predictions on the data
7. Model evaluation: I will evaluate the models based on the AUROC metric.

In the first part, I will use functions such as .head() or .isnull() to get a good overview of the data.
In the second part, I will use different types of graphs such as histograms to visualize the data. This will help me to learn more about the different features.
In the third part, I might use tools such as one-hot encoding to engineer features. I might also choose to aggregate features.
In the fourth part, I might drop features and define the final training and test datasets.
In the fifth part, I will train different supervised learning models on the data. I will use different high-performing tree-based models such as Random Forests, LightGBM or XGBoost.
In the sixth part, I will predict the fraud probabilities for the transactions in the test data set for each trained algorithm.
Finally, I will compute the AUROC metric for each mode in order to evaluate them, using scikit-learn’s sklearn.metrics.roc_auc_score.