Machine Learning Engineer Nanodegree

Capstone Proposal

Carina Thobe

30.04.2017

Domain Background

Fraud Detection is a very important research area for businesses and banks since it has a direct impact on their profits. There are many industries that have an interest in detecting fraudulent behaviour as early as possible: Banks need to identify criminal transactions that are drawn illegally from their customers' accounts, telecommunication companies need to know if the customer is trustworthy for a device payable by instalment and insurance companies desire to reveal insurance fraud.

The European Central Bank estimated that in 2012 the total value of credit card fraud was about to be 1.33 € billion in 2012, which represents an increase of almost 15% compared to the previous year (European Central Bank, 2014). This example illustrates that fraud is a very important issue that companies and banks are highly interested in fixing. Therefore different data mining and machine learning methods are in use, such as classification, regression, clustering or prediction techniques in order to detect fraudulent cases (Dal Palozzo, 2015).

Problem Statement

A bank institute is interested in identifying credit card fraud as early as possible in order to avert damage from its customers and from themselves. The aim is to predict for a transaction whether it is fraud or not. Therefore they can apply machine learning and data mining methods that use past data which is already classified in fraud and non-fraud cases. This data should have one dependent variable (1=fraud/0=no fraud) that is going to be predicted and several independent variables such as amount of transaction, timestamp of transaction, target destination/country, country of transaction origin, frequency of transactions of card holder, ratio of foreign transations of card holder and so on. The quality of the model depends also on the quality of the predicting variables.

With using supervised learning methods such as regression, decision trees, neural networks, SVM or Bayes Learning it is possible to train a model that will predict whether a case is fraud or non-fraud when given new, previously unseen data. After examination whether the prediction was right or not, those cases can be used to update the algorithm and to make it better (this is the machine learning part).

Datasets and Inputs

The data for the analysis are credit card transactions generated by European credit cardholders. The dataset has 284,807 transactions with 492 cases of fraud, which is only 0.172% of all transactions. The variables of the dataset are all transformed by a PCA transformation and unfortunately there is no meta data that explains the meaning of the variables (due to confidentiality). There are 2 variables that are not transformed: “time” and “amount”, whereas the latter is the amount of the transaction and the first is the time between each transaction and the first transaction in seconds. The variable "Class" is the response variable (1=fraud, 0=no fraud). The dataset was released by a research collaboration of Worldline and the Machine Learning Group of the University of Bruxelles (ULB) (Caelen, Dal Pozzolo, Johnson, & Bontempi, 2015).

I am going to download the dataset from kaggle: https://www.kaggle.com/dalpozz/creditcardfraud Then I will split the dataset into 2 parts, training and testing set. Whereas I will start with 80% training and 20% as testing set.

Solution Statement

The aim is to classify whether a transaction is fraud or not. Therefore I am going to apply different supervised learning methods and compare the results to each other in order to find out which one performs best:

- Logistic Regression (this will be the benchmark model)
- Decision Tree
- Neural Network
- SVM
- (maybe Bayesian Learner, if any of those performs really bad)

First I will do feature engineering and find out whether I should remove some of the variables or not. For example if there are some variables that have zero variance, which means they have in all data points the same value, then I will eliminate that feature. Since all features, except the amount and time, are already transformed by a PCA I won't need to do a normalize them. But I will examine the variable "amount" and (if necessary) normalize it.

Since the data is quite imbalanced I will figure out whether there is a need to resample (e.g. undersampling). Also I will divide the data into training and testing data subset. I will train the model on the subset of training data and then test it on the previously unseen testing data. After implementing the models I will compare them based on the evaluation metrics and conclude which one is best for the data given.

Benchmark Model

A very simply heuristic model could be predicting "non-fraud" in 100% of test cases and that would lead to an accuracy of 99.83% which is already quite high. This is due to fact that the dataset has only 492 cases of fraud and is thereby imbalanced. With regard to the business background this is not meaningful since just one fraud case can already cause a lot of damage to the business. Therefore a bank has a very strong interest in detecting every single fraud case.

This made me decide that a heuristic model is not adequate. The benchmark model for this problem will be a logistic regression which can in most cases deliver robust predictions. It will be measured by the same evaluation metrics as the challenging models.

Evaluation Metrics

There are classic accuracy measurements such as

- Accuracy = (TP+TN)/Total
- Precision = TP/(TP+FP)
- Recall = TP/(TP+FN)

True Class ↓Predicted Class→	Fraud (1)	Non Fraud (0)
Fraud (1)	TP = true positive	FN = False negative
Non Fraud (0)	FP = false positive	TN = True negative

In our case it is important to maximize the number of true positive labelled cases and to minimize the false negative labelled cases. The false positive cases are not as bad for the business as the false negative cases, because the latter will cause tremendous harm to the business. The false positive cases will probably result in extra work for manually checking whether it is really fraud or not. This makes *Recall* an important metric to examine. Accuracy is not the measurement of choice because the data is highly imbalanced and precision does not take the worst case of false negatives in account (Descoins, 2013).

The authors of the dataset already point out the difficulty due to the imbalance of fraud and non-fraud cases and therefore recommend using the *Area Under the Precision-Recall Curve* (Caelen, Dal Pozzolo, Johnson, & Bontempi, 2015; Dal Palozzo, 2015). The Precision-Recall Curve is calculated as plotting a set of pairs of precision and recall for a number of thresholds and connections those dots with a line. Then the area under the curve can be calculated as definite integral (Richardson & Domingos, 2006).

I am going to consider both recall and area under the precision recall curve.

Project Design

First of all I am going to do *exploratory data analysis* to get to know the data and develop a feeling for the variables. I will:

- Find out if all input variables are filled, are there missing values and what to do with them
- Find out if there are outliers, if there are I will decide whether I should eliminate them or should keep them because they are relevant (E.g. in case of "amount" this could be an indicator for fraud
- Do feature selection (sklearn.feature_selection) and find out if there are some variables that have no explanatory value and that can be deleted
- Find out whether I should normalize "amount" and "time"

Second I am going to divide the dataset into a *training and test* set. I will start with 80% training and 20% testing data. For the training data I will try to do resample the set, which means I will take all fraud cases of the training set und just a portion of the non-fraud cases ("undersampling"). If that seems to be a bad idea I will just use the stratify parameter in train_test_split and assure that I do not worsen the imbalances.

Third I am going to implement the chosen models:

- Logistic Regression (this will be the benchmark model)
- Decision Tree
- Neural Network
- SVM
- If any of the above will perform really bad I will try out a Bayesian Learning Algorithm

Then I will compare the evaluation metrics defined above of the models to the benchmark model and draw a conclusion.

Literature

Caelen, O., Dal Pozzolo, A., Johnson, R. A., & Bontempi, G. (2015). Calibrating Probability with Undersampling for Unbalanced Classification. *Symposium on Computational Intelligence and Data Mining (CIDM), IEEE*.

Dal Palozzo, A. (2015, 12). *Adaptive Machine Learning for.* Retrieved 04 30, 2017, from http://www.ulb.ac.be/di/map/adalpozz/pdf/Dalpozzolo2015PhD.pdf

Descoins, A. (2013, 03 25). *Tryolabs*. Retrieved 04 30, 2017, from https://tryolabs.com/blog/2013/03/25/why-accuracy-alone-bad-measure-classification-tasks-and-what-we-can-do-about-it/

European Central Bank. (2014, 02 25). *European Central Bank*. Retrieved 04 30, 2017, from https://www.ecb.europa.eu/press/pr/date/2014/html/pr140225.en.html

Richardson, M., & Domingos, P. (2006, 01 26). *Markov Logic Networks: Online Appendix*. Retrieved 04 30, 2017, from http://aiweb.cs.washington.edu/ai/mln/auc.html