Udacity - Machine Learning Engineer Nanodegree - Project Capstone

Vinícius de Oliveira Silva

July 6, 2020.

Proposal

1. Domain Background:

In the financial context of credit card transactions, we are often faced with customers who are charged with paying for items that they have not actually purchased. We call this fraudulent credit card transactions. In order to avoid such a problem, it is important that companies are able to identify / recognize when a credit card transaction is a fraud or not.

The purpose of this project is to create a classifier model that is able to recognize fraudulent transactions using machine learning. The dataset used in this experiment has transactions made by credit cards in September 2013 by european cardholders.

It offers us 28 pre-generated features from the use of PCA (Principal Components Analysis), the time and amount involved in the transaction.

2. Problem Statement:

The dataset used in this study can be found at the link below.

Link Dataset link: https://www.kaggle.com/mlg-ulb/creditcardfraud

The question we want to answer is whether a credit card transaction is considered fraudulent or not. In other words, we have a dataset available with information on credit card transactions already labeled as fraudulent or not and, based on them, we want to train a machine learning model to identify whether new transactions are fraudulent.

3. Dataset and Inputs:

As mentioned earlier, the dataset used in this project can be easily downloaded from the Kaggle machine learning competition website below:

link: https://www.kaggle.com/mlg-ulb/creditcardfraud

There we will find a ***creditcard.csv*** file containing credit card transactions from European customers collected in the September 2013 period.

The dataset has transactions referring to 2 days of which we have 492 fraud examples out of 284807 transactions, being, therefore, an unbalanced dataset. Fraudulent transactions represent only 0.172% of total transactions.

The aforementioned .csv file presents only numeric variables that are the results of applying PCA (Principal Components Analysis) on the original features for reasons of confidentiality.

We have 28 features generated from the PCA that are called V1, V2, ..., V28. Only two of the original features have not been transformed by the PCA: Time and Amount.

“Time”: Time elapsed in seconds between each transaction and the first transaction in the dataset.

“Amount”: Corresponds to the Amount of the transaction.

“Class”: Represents the class to which each transaction belongs. It presents a value of 1 when it is a fraud and a value of 0 when it is not a fraud.

4. Solution Statement:

The solution will be a classification model to identify whether a credit card transaction is fraudulent or not. Before training the machine learning model, an analysis of the data will be done using the **Python numpy, pandas, matplotlib and seaborn** libraries. The number of samples of each class will be verified, as well as the correlation matrix between the numerical features will be calculated and a normalization of the data, so that they will have zero mean and unit variance. To train the model, we will analyze some classification techniques using the **scikit-learn library** and evaluate all of them using accuracy and also Precision-Recall curves. That done, we will choose the best model to fine-tune, using a grid search and, thus, finally present our classification model.

5. Benchmarck Model:

Initially we trained 6 different models with their default parameters and from the Precision-Recall curve for each training, we decided to refine the training of the Random Forest algorithm, in order to find the best hyperparameter configuration of it. As a result, our Benchmark model will be **SVM (Support Vector Machines)**, widely used to solve multidimensional data problems (as is the case) and which has already shown good results for this type of classification task.

6. Evaluation Metrics:

The metrics we will be to evaluate our models are the **accuracy of classification** and also **Precision-Recall curves**. Once we choose the best model, we will fine-tune it by using a **grid search technique** and also a **k-fold cross validation** method to decide which are its best hyperparameters to finally have our classifier.

7. Project Design:

a) Data visualization:

On this step, we visualize data and decide which features are pertinent to use in our future machine learning model.

We verify that our dataset has 30 numerical float64 variables and onde int64 which is our label. Then we check whether our dataset is unbalanced by using a pie chart.

Finally we calculate correlation matrix of our features to decide whether there are features strongly correlated to discard some of them.

b) Data pre-processing:

We normalize the input data of our test models, so that they have zero mean and unit variance.

c) Split dataset into train and test portions:

We split our original dataset from ***creditcard.csv*** into train and test parts in which we have 80% of them for training and 20% for testing.

d) Model selection:

Our model selection was made using the scikit-learn python library. We evaluate our data with Logistic Regression, Decision Tree, Random Forest, KneighborsClassifier, SVM with linear and rbf kernel types.

In order to decide which of them is the best, we used accuracy metric and Precision-Recall curves.

e) Fine-tuning the model selected:

After choosing our best model, we will adjust it using a grid search technique. We used the scikit-learn grid search to go through different values of hyperparameters of the chosen technique and decide which one is the best for our final classifier model.