Machine Learning Application Report

Overview

This section describes the various Machine Learning algorithms used and applied to the "Credit Card Fraud Detection" dataset.

Summary Files

I took a creditcards.csv file where Class is the target variable in there already prelabeled as 1 and 0 to easier classify as Fraud or Non-Fraud transaction and do not have to find them first. Dataset is ready for training a model but very imbalanced to make accurate classifications.

The dataset that is used for credit card fraud detection is derived from the Kaggle https://www.kaggle.com/mlg-ulb/creditcardfraud The dataset has been collected and analysed during a research collaboration of Worldline and the Machine Learning Group (http://mlg.ulb.ac.be) of ULB (Université Libre de Bruxelles) on big data mining and fraud detection.

Feature Engineering

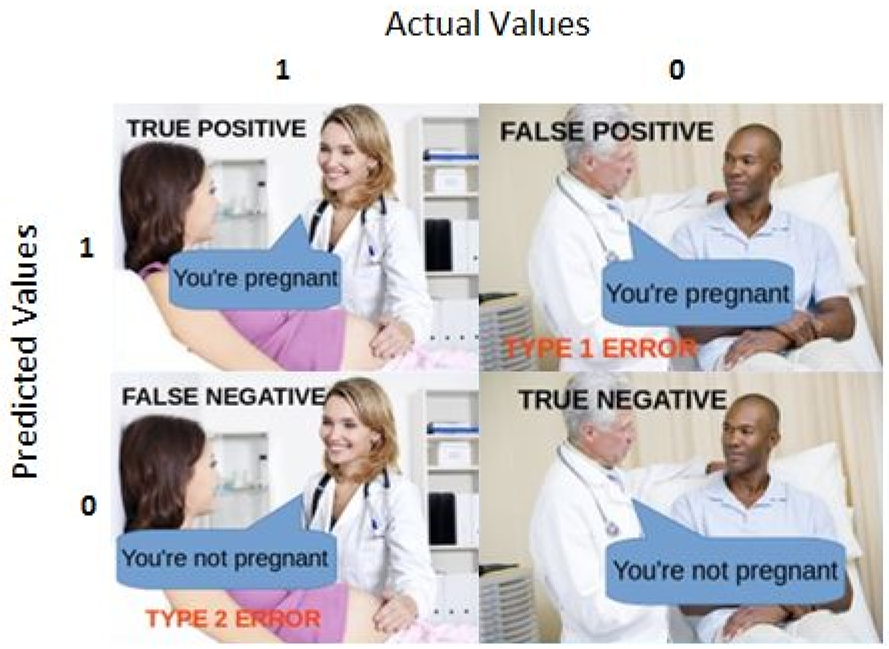
In this section I had to prepare all the data to make it readable and standardized for computations. I had data to be scaled and labeled.

Training a model

There are many algorithms invented to train a classification and anomaly detection model. This will take a lot of time to test them all. That's why I tried only four, the most common:IsolationForest,Support Vector Machine,KNeighborsClassifier, Autoencoder.

Findings and conclusions

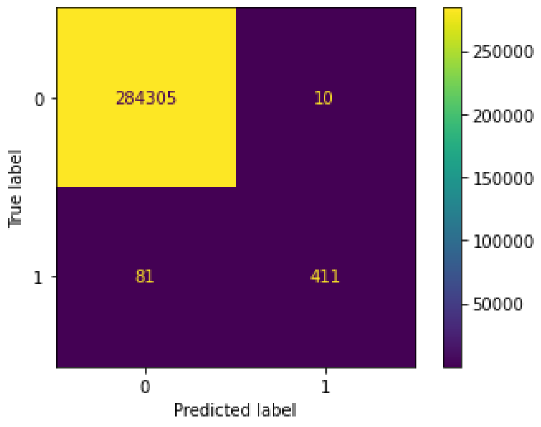
To evaluate models's performances, I need to use special metrics. There are a variety of methods but I do not need to use them all and use only a proper one which in my case is a confusion matrix.
Let me explain the confusion matrix. Confusion Matrix tells me how many a Classifier classified correct values and incorrect out of all counts in details.



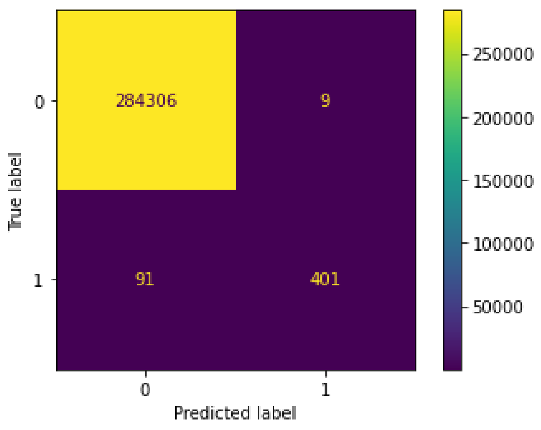
In other words, I am going to have numbers of my correct classified or detected values and number of my incorrect classified or detected values.
I had a binary classification problem so I am going to have a number of correctly detected normal transactions and a number of correct fraudulent transactions as well.
To say it in detail, the upper left corner is correct normal transactions and the lower right is correct fraudulent transactions.

Let’s see what it will tell.

Model Algorithm	Metrics: Confusion Matrix
IsolationForest:	[401, 283914] [175, 317] Too bad. Only 401 of normal transactions detected correctly. But good news, 317 fraudulent transactions detected correctly. Good percent from total.

Model Algorithm	Metrics: Confusion Matrix
Support Vector Machine	 Good job did on both. Detected almost all normal transactions correctly.

	Detected almost all fraudulent transactions correctly.

Model Algorithm	**Metrics: Confusion Matrix**
KNeighborsClassifier	 **Good job did on both. Detected almost all normal transactions correctly. Detected almost all fraudulent transactions correctly.**

Model Algorithm	**Metrics: Confusion Matrix**
Autoencoder	**Model1** **[283155 1160]** **[336 156]**

	Model2 **[[284205 110]** **[282 210]]** **Autoencoder probably need better training and do a better job detecting fraudulent transactions**

Model Algorithm	**Metrics: Confusion Matrix**
Support Vector Machine	**[284305** out of 284315 correctly **, 10** incorrectly **]** **[81** incorrectly**, 411** out of 492 correctly**]**

I justify this using this metric in my problem because my problem has very imbalanced data to detect and secondly, my problem is to detect correct fraud values, not to classify in general. Accuracy metric will not be trustworthy here. According to the confusion matrix, I have the best results in Support Vector Machine. I never know which one is the best for all cases. In one case , Isolation Forest will win, in another case, Autoencoder will win. There is no one Algorithm that will work perfectly for all cases. It depends on a dataset features it is made of and its properties. Obviously, the Support Vector Machine performed the best. Because it detected the highest number of all fraud transactions that I have in my dataset in total. If I have 492 fraud transactions in my dataset, SVM detects almost all of them 411. Other algorithms did a good job too but SVM did the best. Even if KNeighborsClassifier classified all Normal transactions correctly, it did worse on Fraud. My goal is to detect all Fraudulent transactions instead**.** Of course , I want to use the Support Vector Machine algorithm in my case and problem. Support Vector Machine is designed for classification and regression problems.SVM is based on the idea of finding a hyperplane that best separates the features into different domains.So in my problem it works properly and the best of all. There are techniques that will fix imbalanced data problems:Undersampling and Oversampling. I can balance data artificially just for training models properly and achieve better results in classification accuracy. This is how I can improve my project.