Credit Card Fraud Detection¶

The Credit Card Fraud Detection Problem includes modeling past credit card transactions with the knowledge of the ones that turned out to be fraud. This model is then used to identify whether a new transaction is fraudulent or not. My aim here is to detect 100% of the fraudulent transactions while minimizing the incorrect fraud classifications.

This is a potential area of a banking system who rely on clients' trust to provide 100% of security to their customers. All banks have a liability to give a worry free service and convince their users that they have the best high-tech technologies to deal with fraudulent transactions. It requires a touch of a data scientist's job and training a classification model based on given data.

I will try solve this problem from a dataset of my choice:

The dataset that is used for credit card fraud detection is derived from the following Kaggle URL

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

Data Wrangling

Overview

This section describes the various data cleaning and data wrangling methods 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.

Duplicates and Missing Values

For the analysis it was necessary to check for duplicates.
After analysis, no missing values and duplicates were found and no columns to drop.

Outliers and handling them

After making plots of Amount Distribution of an individual V, it is easy to see outliers. (unusual pattern of data that is really different or far from normal). What I did is I annotated a point which is far from normal visually. So nobody will miss it and will notice. But are those really outliers? To decide outliers or not, I can perform an IQR test.
This is my approach at this point.

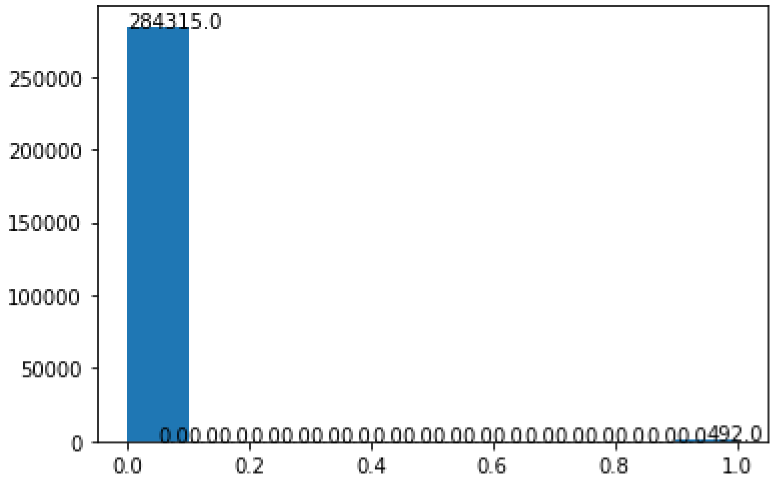
Statistical Data Analysis Report

Overview

This section describes investigating the data, finding trends and relationships between features, stating and testing a hypothesis applied to the
the “Credit Card Fraud Detection” dataset.

Exploratory Data Analysis

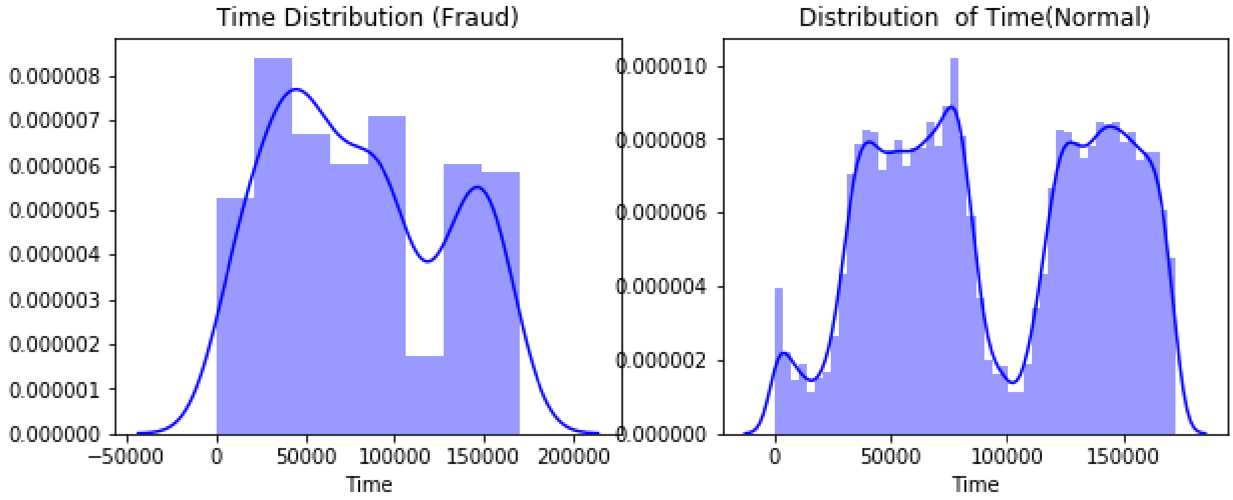
To be able to analyze data, I created labels for Class. Class is the target feature that is the final investigation prediction.
If Class is 1 , this is going to be a Fraud transaction.
If Class is 0, this is going to be a Normal transaction. This is even good already for a supervised learning algorithm to do.
Next, I computed and visualized the quantity of Normal vs Fraud transactions.



As was reported, in this dataset I have 284315.0 of Normal transactions and 492 of Fraud transactions. This is very imbalanced. I am going to have a hard time to train a good model just having this little information about Fraud.

Now I know Class labels so I can figure out the mean of each V by Class.

After taking a look at the mean I could tell that Fraud vs Normal transactions are negatively covariate. This is meaningful information. I can tell a pattern and what is different between them. Also, interestingly, Amount of Fraud means is greater than Amount of Normal means.



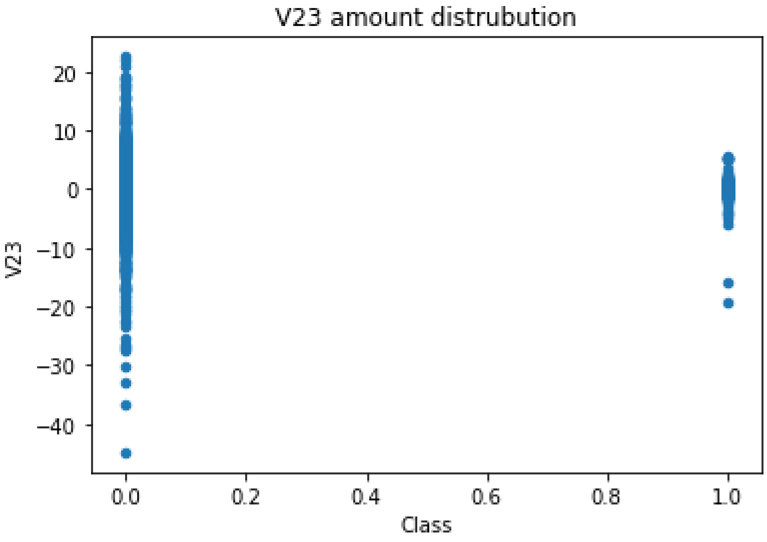
Then I just visualized Time distribution of fraudulent transactions vs Normal.

These two distributions do not look different a lot but plots tell about how different distributions over the time only. So I do not have meaningful information here. By looking at the distributions we can have an idea how skewed are these features. Normal Distribution does not look normal. Distribution rises and declines, up and down. Fraud distribution has less turbulence as it is frequency is much less.

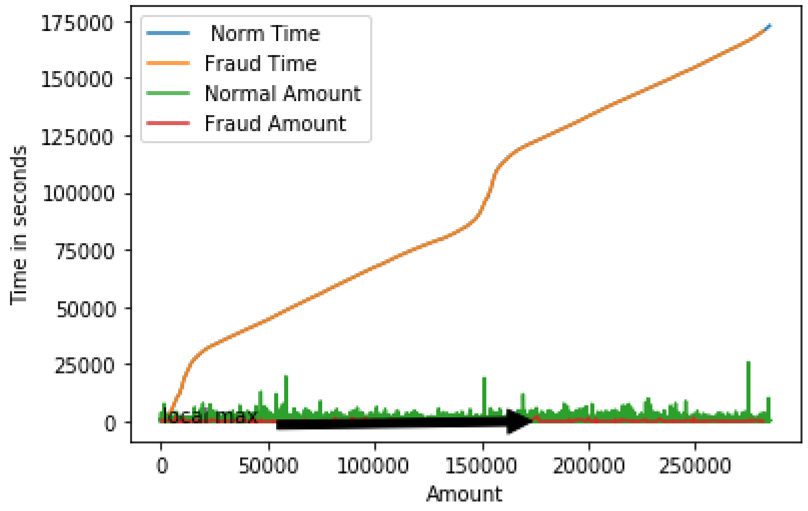
But at what time did the fraud transaction happen?

So according to this description, Fraud transactions happened between 406sec and 170348sec. I applied a filter to display data only for this timestamp and looked if I can find any trends.

Trends are different now. I was wrong about Class covariance. V24 now is a negative number for both Fraud and Normal. So this pattern that I assumed was not true. To find correlation and trends I made a plot of a random feature and class to look into if there is a relation between them. If V23 has both negative numbers for Fraud and Normal, is this a negative or positive correlation or no correlation at all.

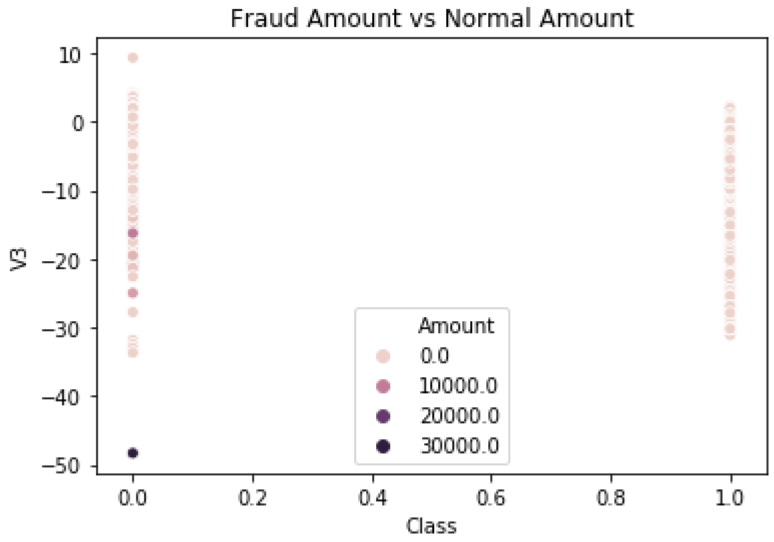


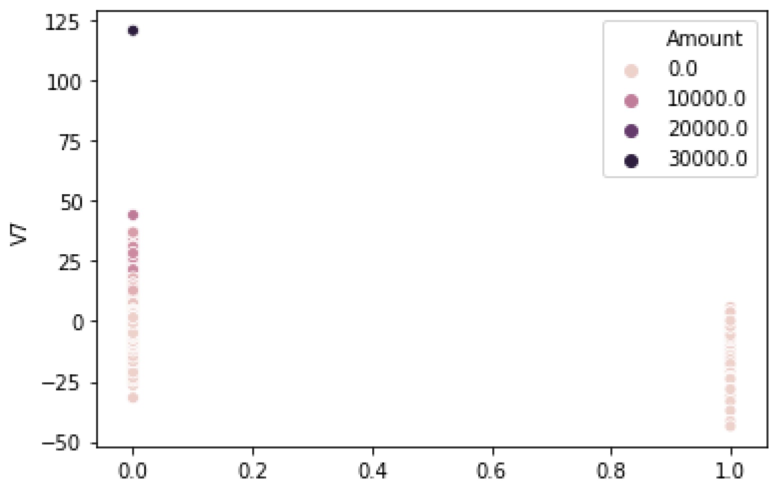
So, according to the plot there is no correlation at all. So my assumptions are wrong. There is no positive or negative trend. And the same idea will apply for all V's.
I took time to look at what the plot will tell about transactions amount of Fraud vs Normal.

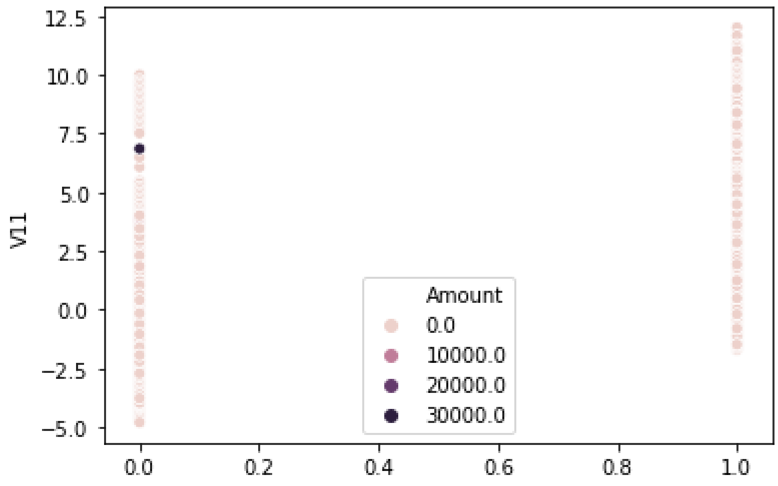


Perfect. Time of Normal and Fraud distributed equally just Normal Time tail is longer. Annotation will show the local max of a distribution where there was a spike. In this case, I show a local max of the fraud transaction amount. This plot is very meaningful. It tells at what time transactions happened by grouping it as Amount. Fraud local max, for example, happened at about 125000sec. And in the same manner it tells about any transactions.

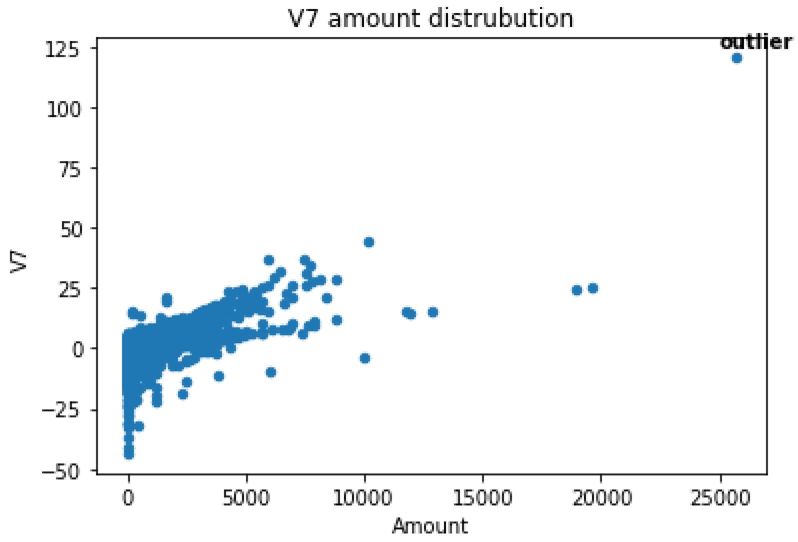
Next what I did is finding out Amount of Fraudulent transactions vs Normal for an individual V.







Unfortunately, these detailed plots did not detect any amount of fraudulent transactions for these individual V's. But we can go further and check them all. Doing it will take a lot of time that's why for this case I just took random V's and did not detect it according to visualizations.
But what catched my attention is V7 Normal distribution. In V7 plot distribution of Normal transactions has a suspicious number that is far away from all numbers. This outlier is highly likely to be fraud.



This plot proves the correlation matrix. The correlation is positive as the correlation matrix identified. It also shows a point which is far from norm and I annotated this point as outlier but this is possibly a fraud here because this point is very different from all the normal points. This is called "Anomaly Detection".

Anomaly detection is the process of identifying unexpected items or events in data sets, which differ from the norm. And anomaly detection is often applied on unlabeled data which is known as unsupervised anomaly detection. Anomaly detection has two basic assumptions:

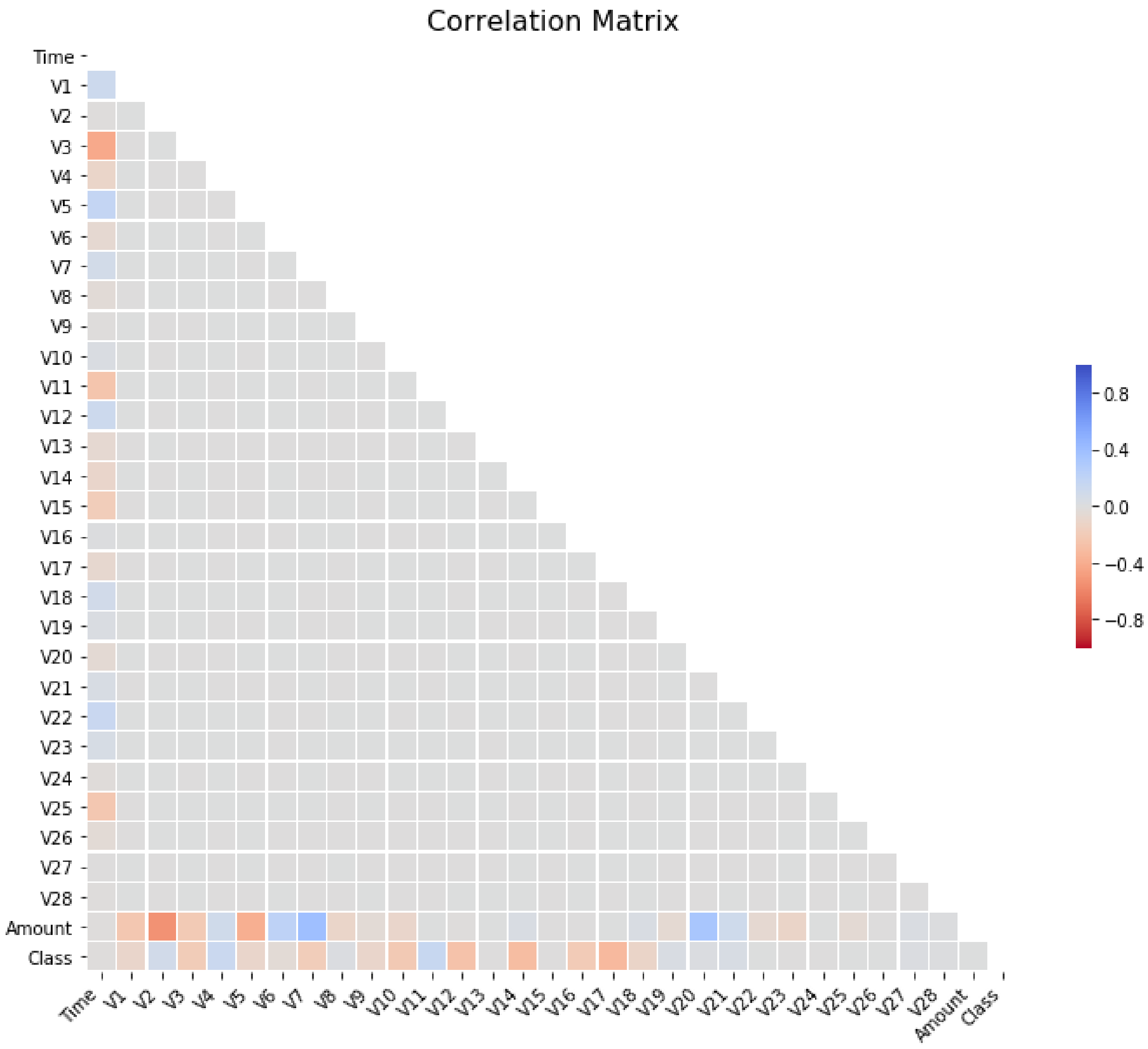
Anomalies only occur very rarely in the data.
Their features differ from the normal instances significantly.

I annotated a point of interest to pay attention for further analysis.
Statistical Analysis
There are two types of statistical analysis: Descriptive and Inferential. Descriptive statistics are used to describe a situation. Inferential statistics are used to explain the chances of occurrence of an event.

Part1: Descriptive statistics

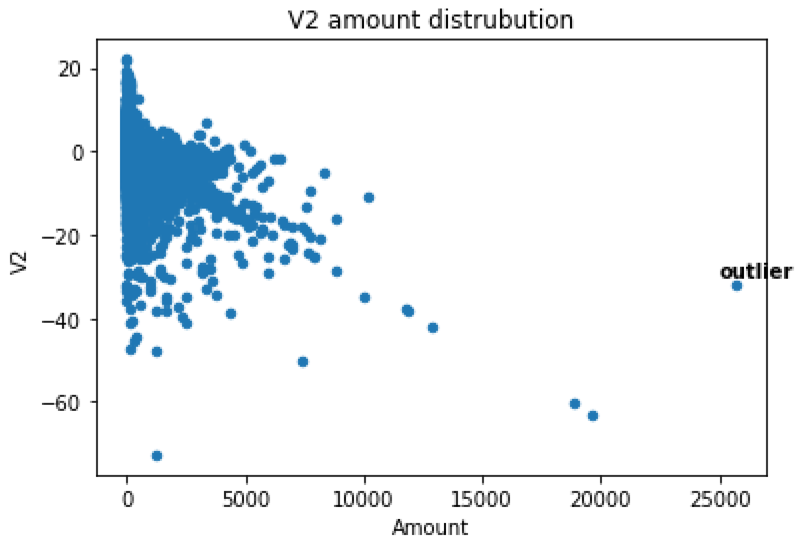
This is a summary of a statistical description for each feature in the dataset: their mean, their standard deviation and quartiles. I can even find the median and mode of any feature for this type of analysis.
Next, it is good to make a Correlation Matrix. Correlation Matrix will tell if any relation between features are present.

The correlation coefficient (also known as the Pearson correlation coefficient) measures how well two variables are related in a linear (straight line) fashion, and is always called r. r lies between -1 and +1. A value of r = -1 means that the two variables are exactly negatively correlated, i.e., as one variable goes up, the other goes down. A value of r = +1 means that the two variables are exactly positively correlated, i.e., as one variable goes up, the other goes up. A value of r = 0, means that the two variables are not linearly related.
This is going to be meaningful information. But I will not detect a proper correlation because I have very imbalanced data.



Negative Correlations: V2, Amount, and V5, Amount are negatively correlated.
Positive Correlations: V7, Amount, and V20, Amount are positively correlated.

I have some more correlations found but not so strong.
And again, data is so imbalanced that can produce wrong correlations.



As you can see, this plot proves the correlation matrix is correct on this relation. There is a negative correlation between these two variables. Outlier point is possibly fraud activity that will be classified as ANOMALY.

Part2 Inferential Statistics

I had a hypothesis about V2 Fraud and V2 Normal means are not statistically significantly different. I am testing two groups' populations. In this case, I will perform the so-called Ttest.
Ttest
Ttest_indResult(statistic=18.761176823315797, pvalue=1.2885797612095705e-59)

A standard alpha level is .05, and pvalue= 1.2885797612095705e-59 is smaller than .05, so I am going to reject the null hypothesis which asserts there is no difference between our sample mean and the population mean. For this two-tail test, we reject the Null and we conclude that there is a statistically significant difference.

Yes , I have a % confidence that my Null hypothesis is wrong. But what error size? I can find it by finding Cohen's d. Cohen's d is the difference measure between two groups. This is called effect size. I computed the effect size of how my two groups differed.

Cohen's d
1.1178246710860031

Small effect = 0.2
Medium Effect = 0.5
Large Effect = 0.8

So according to this scale, my effect size is even larger than 1. This tells me that difference between my two tested groups is really big

Conclusion

I explored data and found trends, negative and positive correlation, outliers. Of course, I can find more and discover more trends and correlations, and outliers if I take a closer look and take more time but at this point is good for now. I compared two groups in my Null Hypothesis and have a probability value in favor of Alternative Hypothesis. I found how big the difference is between the two groups and it is big. I compared two groups but if I have more than two groups(Fraud and Normal) to compare I can use ANOVA.

Machine Learning Application Report

Overview

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

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	[284305, 10] [81,411 Good job did on both. Detected almost all normal transactions correctly. Detected almost all fraudulent transactions correctly.

Model Algorithm	Metrics: Confusion Matrix
KNeighborsClassifier	[284315,0] [98,394] Good job did on both. Detected almost all normal transactions correctly. Detected almost all fraudulent transactions correctly.

Model Algorithm	Metrics:

	Confusion Matrix
Autoencoder	**[284038,277]** **[352, 140]** **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.