Unsupervised learning project (88780)

Zohar Yakobi – 322691585 and Sharon Komissarov – 322880964.

April 3, 2020

Contents

The code to the project can be found in our git, in this link: `https://github.com/sharon200102/Unsupervised_project.git`

1 Abstract

1.1 The data

The data we chose is "Credit card Fraud data": 284,807 X 31 where each line represents a credit card transaction, the columns are a little more complicated: There are 3 categorical columns which are: "Class", "Amount", "Time".
There are another 28 columns that are the product of dimensionality reduction of PCA on the original data (which we probably did not receive due to privacy reasons). In this project we will use various models and methods in order to see how accurate they are in detecting whether a transaction is a normal payment or a fraud.

1.2 Our goals

- Understand the distribution of the data that was provided to us.
- Visualize the data and understand if it's separable.
- Determine the clustering algorithms we are going to use and decide which one gave the best performance.
- Decide whether a transaction can be represented as anomaly.
- Evaluate the results with tools that are suitable to imbalanced data as we have.

2 Description of the tasks

2.1 Statistics

We performed some basic statistics: we first calculated the correlation matrix between the columns of the data, but because the data we received went through a PCA process, there are no high correlations between the columns and therefore the correlation between the columns did not help us at all.

As I mentioned there is a column called "Class" that represents whether the transaction is fraud or not: 0 represents a legal transaction (not fraud), 1 represents fraud. According to an analysis we made, 0.172 percent of the data are frauds and 99.928 percent of the data are legal transactions:

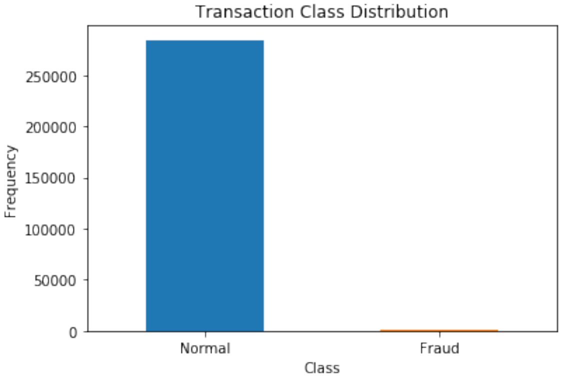


Figure 1

2.2 Visualization

We have made a number of basic visualizations: we compared between the average amount of the fraud and legal transactions and we found that the frauds bar is significantly larger:

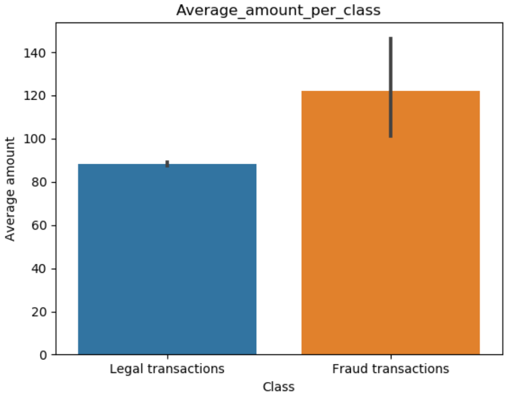


Figure 2

You can pay attention to another interesting visualization that shows that the frauds and legal deals are running roughly the same over time (meaning that time does not so much affect a transaction whether it is fraud or not):

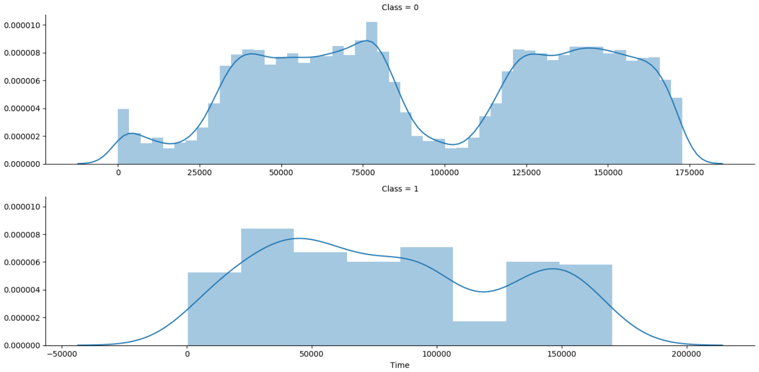


Figure 3

In addition, we plotted the distribution of each column and it seems standard:

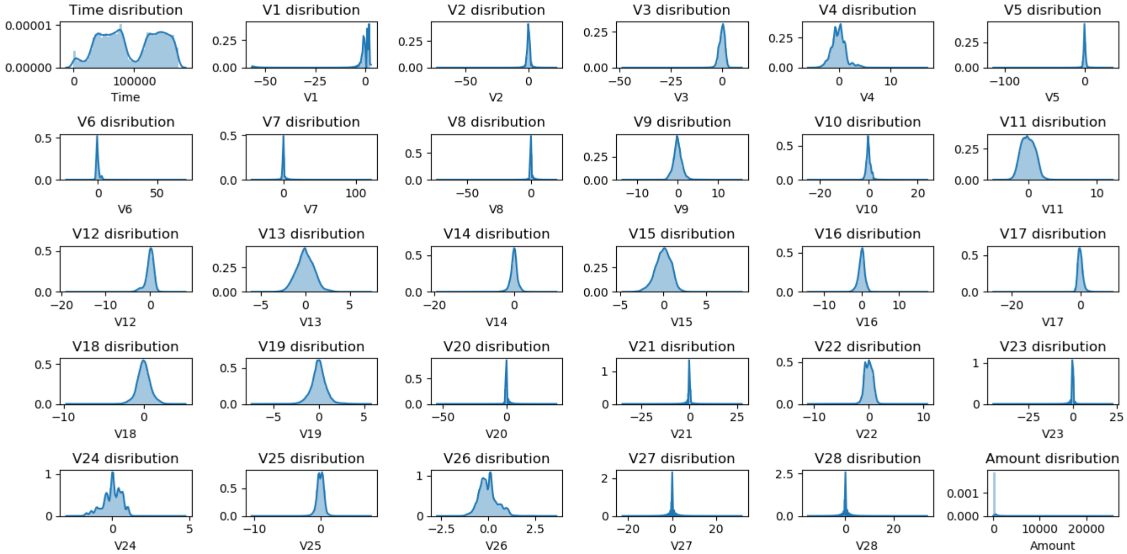


Figure 4

2.3 The significant actions in the project

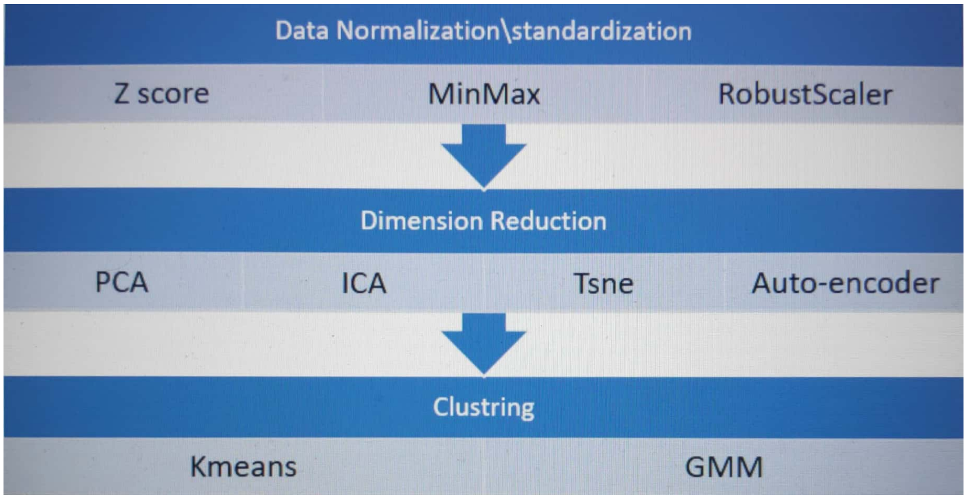
After seeing quite a few visualizations and statistics, we decided to reduce dimensions in many kinds of tools to make it easier to cluster and plot the data.

Our next step is the clustering itself, we wanted to cluster the data into two groups (legal and frauds) using clustering algorithms learned in class.

In addition to reducing dimensions and clustering, we have also turned to another aspect, which is anomaly detection to try and treat the frauds as anomalies.

3 The method and parmeters we used

3.1 Work process



Data Normalization\standardization

Z score MinMax RobustScaler

Dimension Reduction

PCA ICA Tsne Auto-encoder

Clustring

Kmeans GMM

Figure 5

3.2 Data Normalization/standardization

The methods chosen are widely used for data analysis purposes.
Each one has its advantages upon the other methods when dealing
with different aspects of the data (outliers, dispersion , etc).
therefore, we decided to examine a variety of techniques and to get
the best performance possible.
We used:

- Zscore
- MinMax
- Robust standardization

3.3 Dimension Reduction

3.3.1 Tsne

After the basic preprocessing and normalization, we run Tsne on a small amount of the data equally distributed between fraud and legal transactions.
Tsne can give us an indication if further clustering models will perform well in separating fraud cases from non-fraud cases.

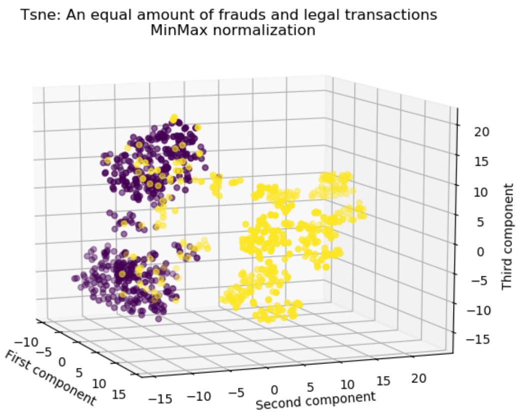


Figure 6

3.3.2 Linear methods

To avoid the curse of high dimensionality, for further clustering and to obtain visualization of the whole data, PCA had a major advantage due to the proportionally large two to three eigenvalues of the covariance matrix. The three largest eigenvalues described more than 80 percent of the variance.

We also implemented ICA because of the possibility that the data isn't normally distributed.

3.3.3 nonlinear methods

Since the previously introduced methods cannot describe more than linear connections of the data.
We tried an nonlinear technique which is the Auto-encoder.
In the implementation process of the Auto-encoder, we used a Neural network with two hidden layers, the architecture looked as follows.

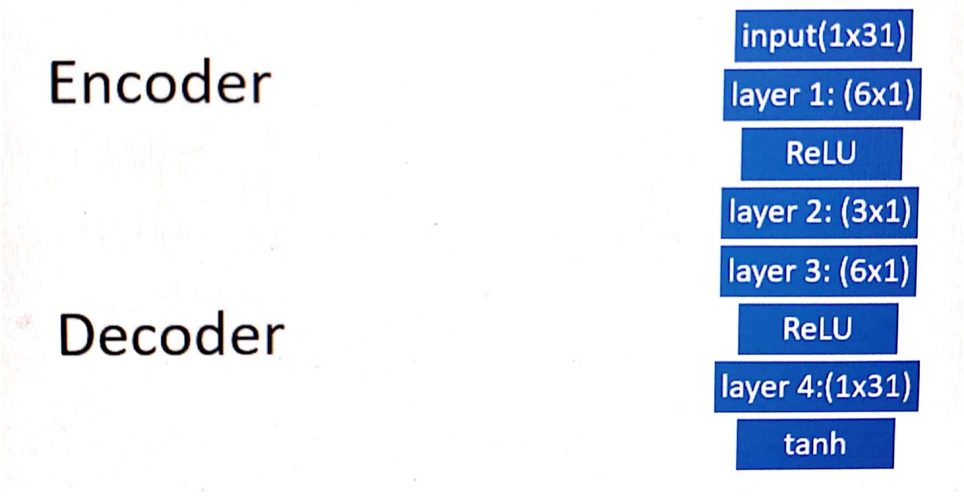


Figure 7

Also here we reduced the data into three dimensions for visualization purposes.

3.4 Clustering

We executed every possible combination (Brute force) of the normalization and dimension reduction methods mentioned above together with different clustering algorithms.
The clustering algorithms that we used are K-means and GMM.

The evaluation of clustering stability will be performed by silhouette analysis and the quality will be evaluated with confusion matrix.

3.5 Anomaly Detection

We used this technique on our dataset because we have solid evidences that the fraud transactions are behaving as anomalies.
As can be seen in figure 2 the average amount of a fraud transaction is significantly higher than the average of the legal ones.
This pattern can also be seen in the MSE loss function when using the autoencoder.
In addition, after performing a t-test it's clear that the mean of most columns is significantly different compering fraud and nonfraud transactions (the P-value is much lower than 0.05).
Therefore we used two methods to find these anomalies.

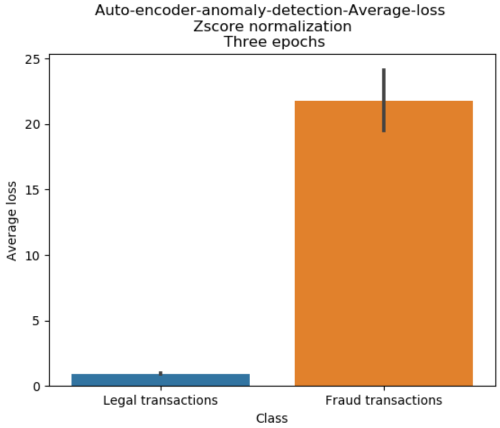


Figure 8

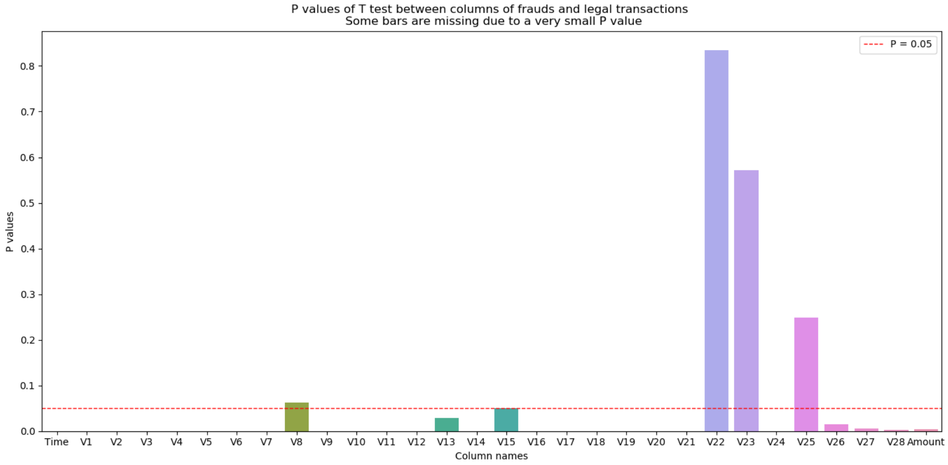


Figure 9

3.5.1 Autoencoder

The architecture here is identical to the Autoencoder for dimension reduction.
We trained the model using the whole data for only 3 epochs, due to the simple task given to the model (returning the input) adding more epochs didn't show any improvement.
The N transactions with the largest MSE on the model will be identified as anomalies (where N is a hyper-parameter).

3.5.2 Isolation forest

The algorithm is based on the fact that anomalies are data points that are few and different. As a result of these properties, anomalies are susceptible to a mechanism called isolation.

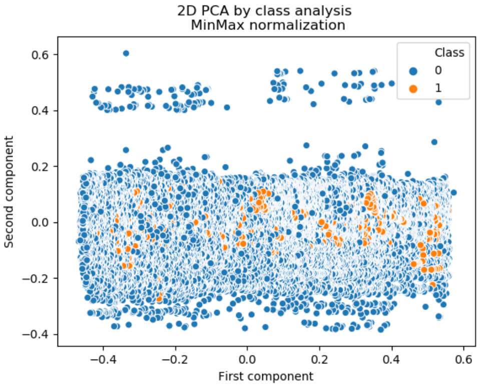
4 Results

4.1 Dimension reduction

Unfortunately, as can be seen the linear dimension reduction didn't separate the data well, regardless normalization.

4.1.1 Linear methods

PCA:



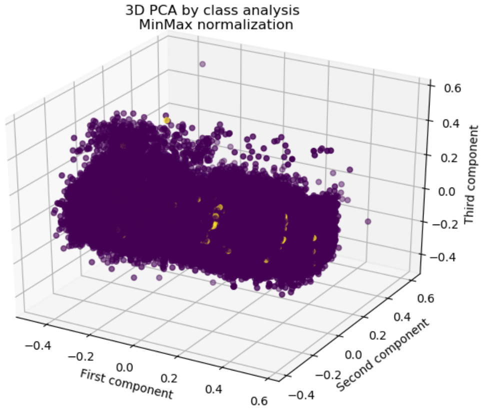
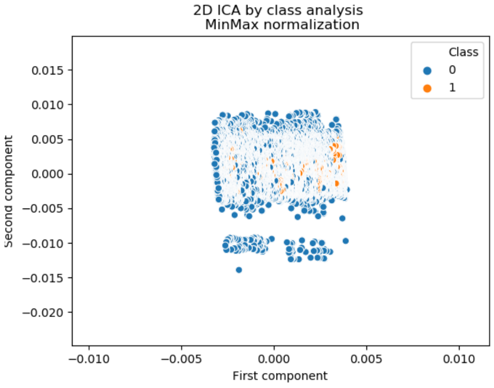


Figure 10, Figure 11

ICA:



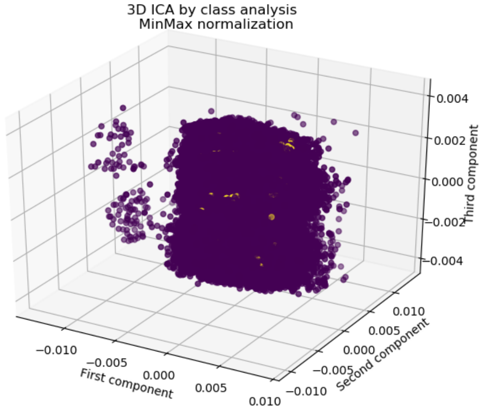


Figure 12, Figure 13

4.1.2 Nonlinear methods

Autoencoder:

In contrast to the methods above the auto encoder in combination with Z-score normalization, produced a fine separation of the fraud and legal transaction.

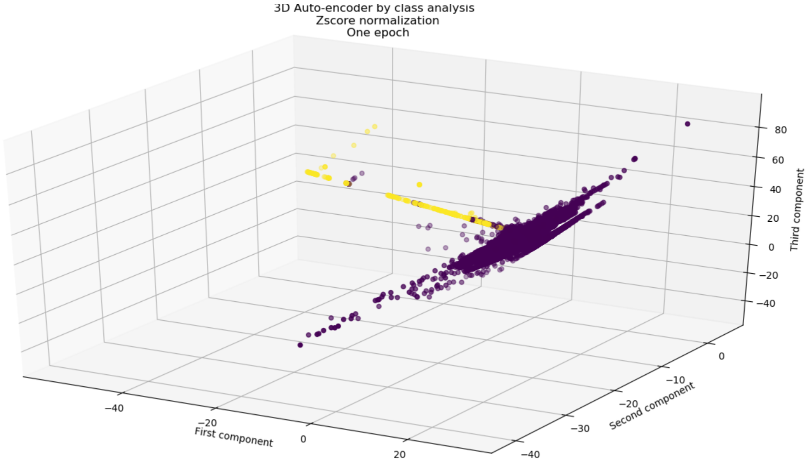


Figure 14

4.2 Clustering

Due to the incapability of the linear dimension reduction methods to show the separation of the data to clusters, the results of the clustering aren't satisfying.

K-means:

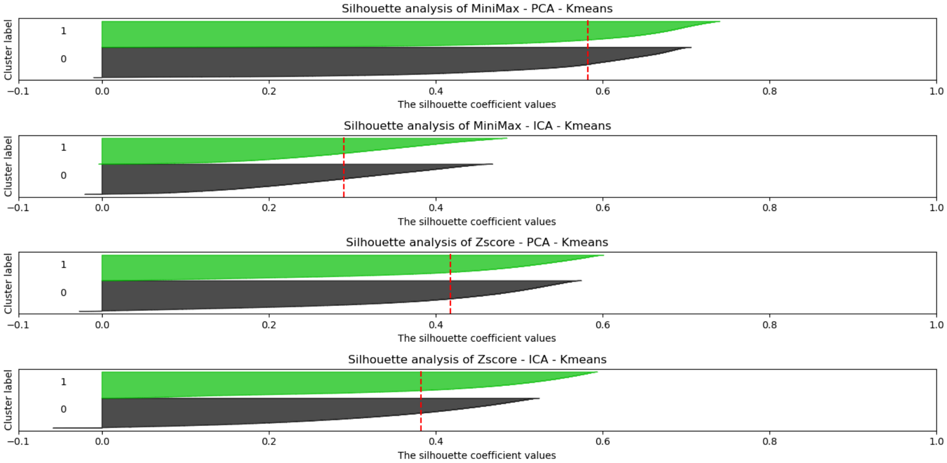


Figure 15

GMM:

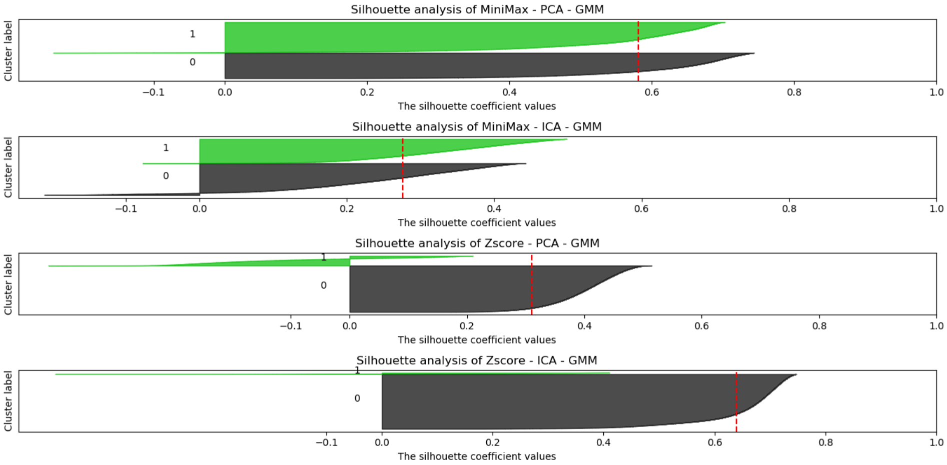


Figure 16

The only possible combinations size wise are Zscore - PCA - GMM and Zscore – ICA – GMM.

Although the average silhouette score of the other combinations is high their cluster size doesn't match the ratio between the legal transactions and the frauds.

Therefore, we have made a confusion matrix to both combinations:

Zscore - PCA - GMM:

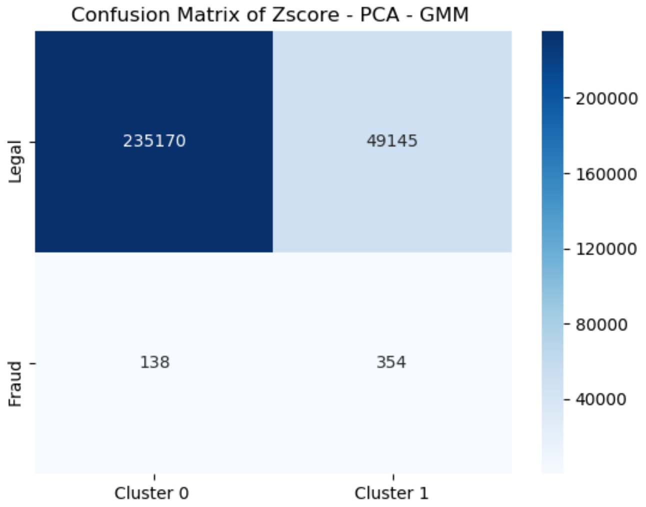


Figure 17

Zscore - ICA - GMM:

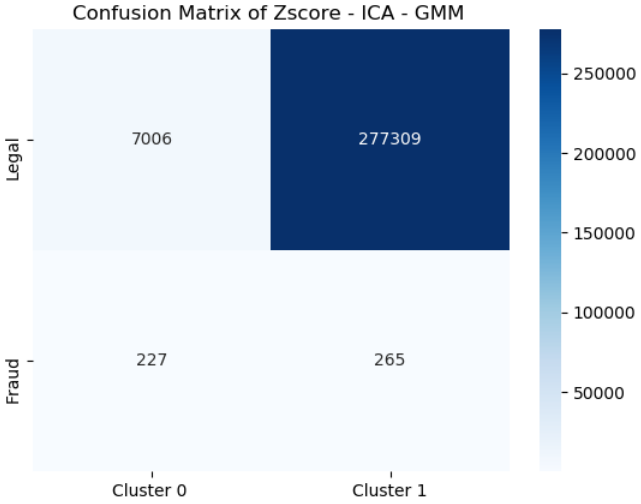


Figure 18

The results are inconclusive, and it can't be said that the frauds are divided into one cluster.
Both clusters don't have a significant difference in the number of frauds transactions in them.
The number of legal transactions in both clusters is enormous relatively to the frauds.

We came up with the idea that it's possible that the legal transactions consist of several groups, i.e. different distributions.
For this reason, we will cluster the data after the autoencoder transformation using more than two clusters.
The best combination is Zscore – Autoencoder - GMM with six clusters which gave extraordinary results:

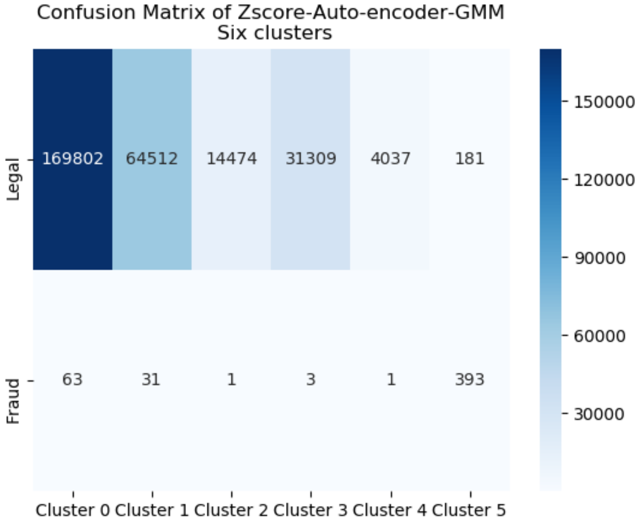


Figure 19

It appears that "cluster 5" represent the frauds cluster, and the rest represent the legal transactions.
The fraud recall is almost 80 percent and the precision is almost 70 percent.
It's likely that more clusters could have led to better results, but overfitting seems Inevitable.

4.3 Anomaly detection

4.3.1 Autoencoder

By testing the autoencoder with all the different normalizations, we found it **extra sensitive** to the normalization performed in the preprocessing stage.

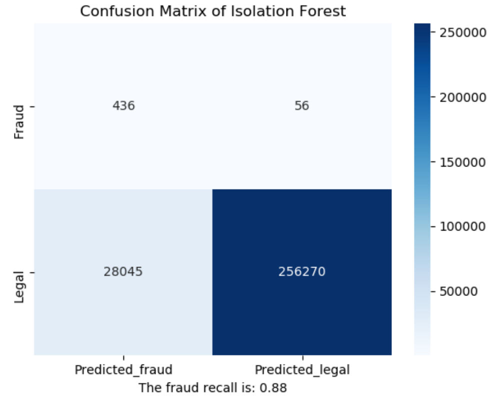
The method that gave the best results was the Zscore.
As it can be seen (figure 21), the fraud recall is almost 90 percent which means that the majority of the frauds is recognized by the model as fraud transactions.

These results are satisfying the need of identifying frauds.

Although the fraud precision is quite low, its acceptable because of the significant Imbalance in the data.

4.3.2 Isolation forest

By testing the isolation forest with all the different normalizations, we found that **it's not sensitive** to the normalization performed in the preprocessing stage.
The results are almost similar to the autoencoder as well as the conclusions (figure 20).



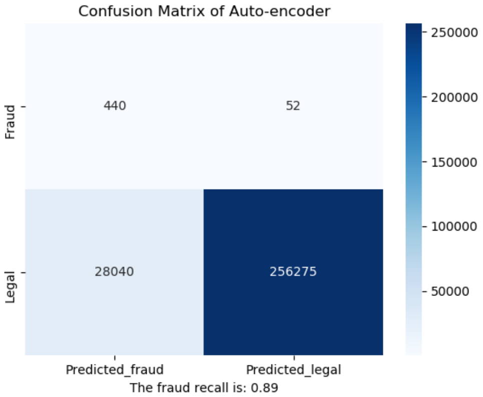


Figure 20, Figure 21

5 Conclusion

In this project we tried to reflect everything we learned during the course.

We were fascinated to explore which algorithms will performe well and which won't.

The fact that the data wasn't easily separable made us try different methods and aspects that we didn't see directly in the lectures.

Each normalization gave different results for each method, MinMax and Zsocre showed the best results when reducing dimensions, while RobustScaler and Zscore gave the best results in anomaly detection.

The tsne visualization was very beneficial and gave us motivation to continue the analysis.

Although the variance explained by the principal components was relatively high, all the linear dimensionality reduction algorithms couldn't visibly separate the data.

The autoencoder transformation resulted with fair separation of the data.

In addition, we couldn't implement KPCA and tsne on the whole data because of running time reasons.

As expected, it wasn't possible to accomplish desirable results from the clustering after using linear methods.

Increasing the number of clusters and using the autoencoder transformation led to much better results.

Alos, we decided to review other directions such as anomaly detection.

A wrong approach could have been using the accuracy evaluation. But we chose the precision – recall and confusion matrix evaluation, which in them our models performed well.

From a supervised point of view, it could have been possible to use the given labels in order to subsample the data to a group that has an equal number of frauds and legal transactions. This will dramatically reduce the imbalance of the data and may make the supervised and unsupervised models better.

Finally, we came to a main conclusion that fraud and legal transactions distribute similarly which made the tusk of fraud detection difficult.