Credit Card Fraud Detection using Isolation Forest, Local Outlier Factor and Robust Covariance

Soh Hong Yu  
Diploma in Applied Artificial Intelligence and Analytics  
School of Computing   
Singapore Polytechnic

***This technical paper aims to conduct an unsupervised learning approach to detect anomalies in credit card transactions. We aim to create a machine learning model that is able to identify the anomalies in transactions which are typically fraud. We will be using 3 techniques (Isolation Forest, Local Outlier Factor and Robust Covariance) and doing a comparison between them to see which technique is better at identifying fraud in credit card transactions.***

***Keywords—credit card, machine learning, python, fraud, comparison, unsupervised learning, isolation forest, local outlier factor, robust covariance***

# Introduction

With the rapid development of online payment, credit card fraud is a major concern. In 2020, about 75 bank customers in Singapore were affected by fraudulent credit card transactions. About $500,000 worth of money has been stolen by “Malicious actors” overseas. This transaction involves the diversion of SMS one-time passwords (OTPs) to overseas mobile network systems [1]. Many steps have been implemented to prevent any fraud from authorities reminding people to be vigilant. Banks also have bank investigators to look at the transaction data and likely indicate fraud. However, banks are using a manual process to identify fraud which takes a lot of time and manpower and might potentially lead to human error. Being able to create an unsupervised learning model to identify anomalies in the transactions. Isolation Forest, Local Outlier Factor and Robust Covariance are models which are used for anomaly detection. In this technical paper, we aim to compare the different anomaly detection models to see which model is better at identifying fraud in credit card transactions.

# Related Works

In the past, there have been similar works of identifying credit card frauds.

An earlier investigation of this problem by Vaishnavi Nath Dornadula, S Geetha,. [2]. They used different classification data to identify credit card frauds using decision tree classifiers, random forest classifiers and logistics regression. In the study, they were also using Synthetic Minority Over-Sampling Technique (SMOTE).

Another investigation was conducted to compare the difference between the anomaly detection methods [3]. The investigation involves many synthetic datasets and many real-world datasets like yeast\_molecular\_genetics dataset, Dstl Satellite Imagery Feature Detection dataset etc. They use many datasets to compare to look for the number of outliers using the Isolation Forest(iForest) and Local Outlier Factor(LOF). It is found that the precision of iForest has a better anomaly detection score compared to LOF.

# Methodology

## Data Set

The data set was collected by Université Libre de Bruxelles [4] for a research collaboration of the Worldline and Machine Learning Group for big data mining and fraud detection. However, due to confidentiality issues, the data providers were unable to give more information and background information about the dataset. 28 columns V1, V2, … V28 are principal components obtained with PCA. The only features that have not been transformed were the “Time”, “Amount” and “Class” features. “Time” refers to the seconds elapsed between each transaction. “Amount” refers to the amount of money involved in the transaction. “Class” is the response variable where the value 1 is fraud and 0 is not a fraud.

## Exploratory Data Analysis

Before we attempt to detect fraud with our data, we will first perform an exploratory data analysis (EDA) on our data. As the data is majority based on the principal components from PCA, it is relatively hard to use those columns. However, columns like “Amount” and “Time” can be used for EDA and for our case we will be separating the data by the “Class” columns. We note that the data have many variances as shown in figure 1.

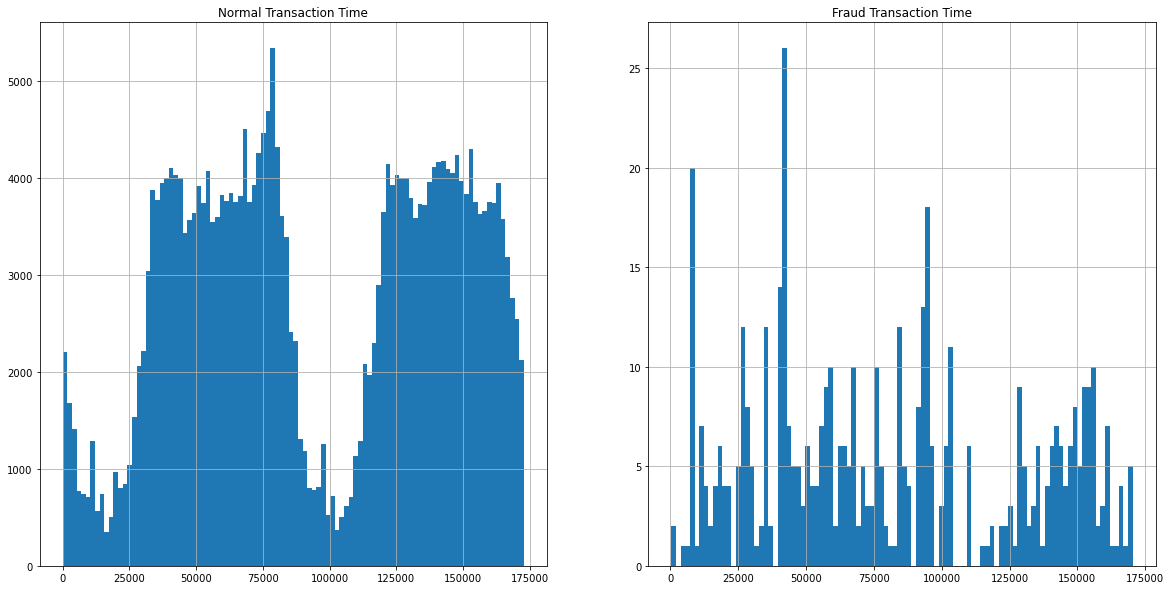


Figure 1, Exploratory Data Analysis

## Data preparation

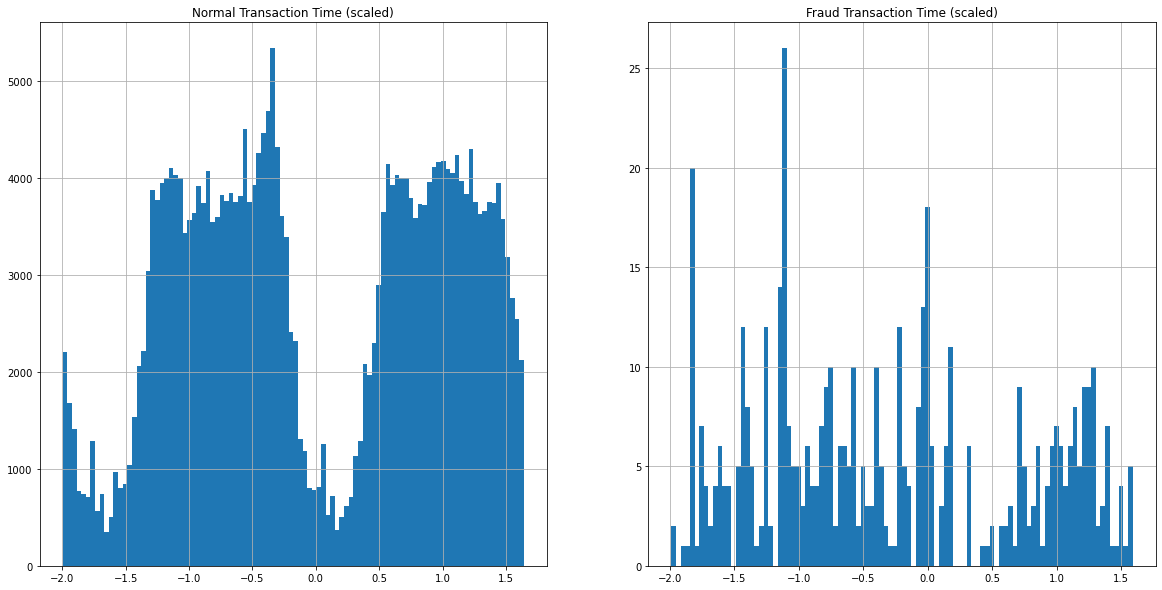
With most clustering/anomaly detection algorithms, the values and the distance between the points play a big role in the cluster, especially if one uses models like Density-based spatial clustering (DBSCAN), which takes the distance between the points as a clustering method. To reduce the distance between points without changing the value’s scale, we will use the StandardScaler to scale the data. StandardScaler will scale the value so that it has a mean of 0 and a standard deviation of 1. Figure 2, shows the effect of standardisation.

Figure 2, Scaled values of Time and Fraud Time

Next, we will be splitting our data into 2 different subsets, the train and test datasets. All fraud data points will go into the test dataset, while the majority of normal data points will be contained inside the training data while some will be in the test dataset. As we are training the model to look for patterns for fraud data, it should study and trained on data that is not fraudulent. The model will be able to pick up patterns from the non-fraudulent data and when an outlier (fraudulent data) is detected, the model will be able to identify it. We will be splitting the data to a ratio of about 75:25 of training data to test data respectively.

# Isolation Forest

First, before we train the models we need to understand how the models work, let’s start off with what is Isolation Forest. Isolation Forest(IF) is very similar to Random Forests. It will randomly sub-sample data and run the data through a tree-based model. Randomly IF will select a random feature (similar to Random Forest) and continuously branch until a certain random threshold is met this will form an Isolation Tree. If the value of the datapoint is less than the selected threshold then it will go to the left branch else it will go to the right branch. The process is similar to cutting an apple with rotten spots until the apple is cleaned. It will repeat til each data point has been isolated or reached its max depth. The above steps will repeat themselves to create the binary trees. With more Isolation Trees that are formed, it will become an Isolation Forest [5] This process is shown in Figure 3.

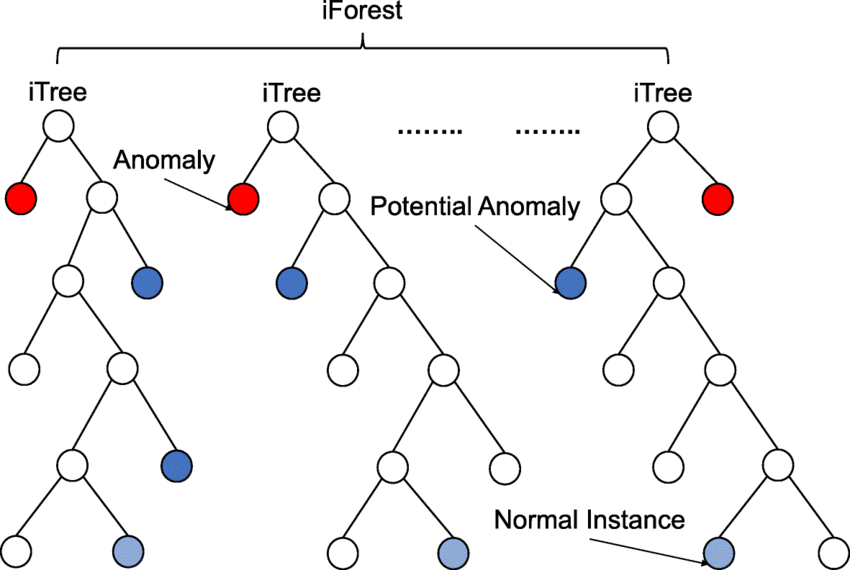


Figure. 3. How Isolation Forest is created [6]

After creating the Isolation Forest, an anomaly score is assigned to each data point based on the depth of the tree. This score is an aggregation of the depth obtained from each of the iTrees. An anomaly score of -1 is assigned to anomalies and 1 to normal points based on the contamination(percentage of anomalies present in the data) parameter provided. [5] Figure 4 shows how the points are split in the tree through a scatter plot.

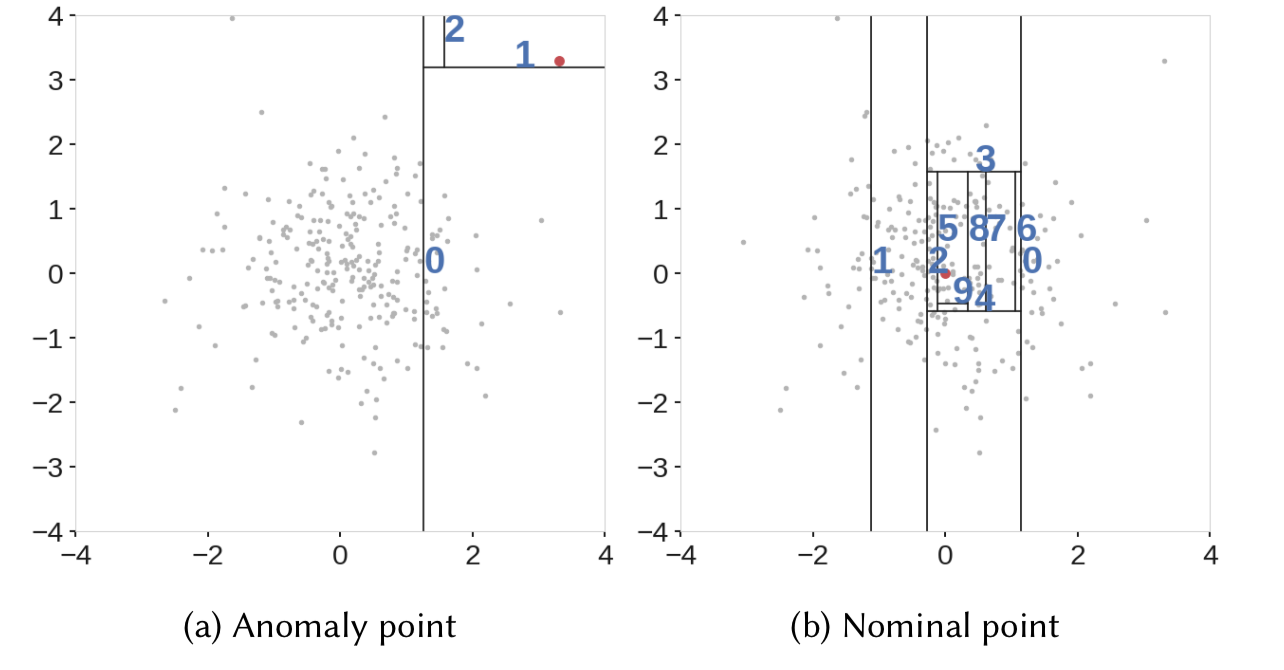


Figure. 4. How data is split based on the point being an anomaly and a normal point [5]

# Local Outlier Factor

Local Outlier Factor(LOF) is an unsupervised algorithm which computes the local density deviation of a given data point with respect to its neighbors. It considers outliers as the samples that have a substantially lower density than their neighbors. [7] The distance between each individual points is calculated thru the process of pythagoras’ theorem. Figure 5 shows how the distance between the points are calculated.

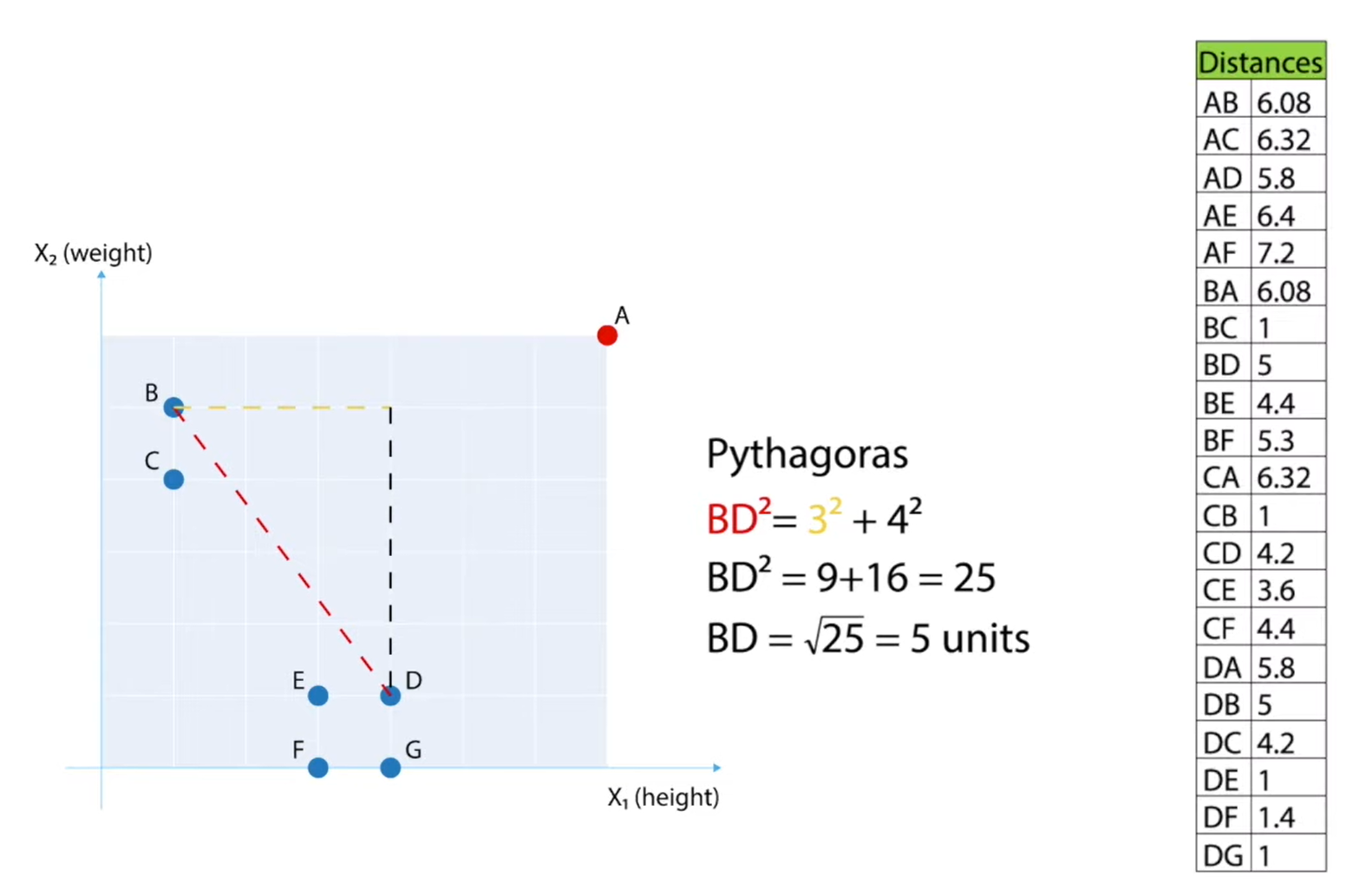


Figure. 5. Distance between points are calculated [8]

Afterwards, a mini KNearestNeighbour algorithm is used to find the number of k declared in the parameters. A Local Reachability Density (LRD) calculation will be done. LRD calculation works by estimating the distance at which points can be found by its neighbors. The LRD is the count of the items in the K Nearest Neighbor set which was calculated above. [9]

Equations used to compute LOF

1. Robust Covariance

Robust covariance, sometimes referred to elliptic envelope, methods are based on the fact that outliers lead to an increase of the values (entries) in Σ, making the spread of the data apparently larger. Consequently, |Σ| (the determinant) will also be larger, which would theoretically decrease by removing extreme events. [10] In statistics, deviation can be accessed by the Z-score. The generalisation of the Z-score for a point xi in the case of a p-dimensional multi-variate probability distribution with some mean μ and covariance matrix Σ is known as Mahalonobis distance di.

Equation used to calculate Mahalonobis distance

This metric access how many standard deviations σ away xᵢ is from μ. However, in the presence of outliers, the mean and covariance matrix computed will be distorted which makes the Mahalonobis distance useless. Hence the robust method was created. The minimum covariance determinant estimator (MCD) is one of such robust methods. it is used to estimate the covariance matrix of highly contaminated datasets.



Equation used to calculate MCD

Robust covariance will use the covariance data to find out the center of the data and will draw an ellipse around the data points as shown in Figure 6.

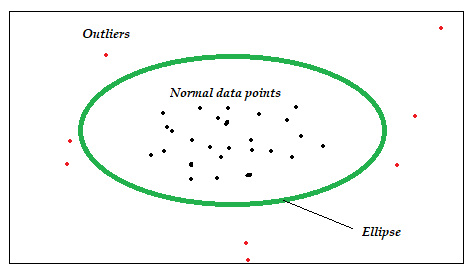


Figure 6, How robust covariance works

1. Difference between Models

As well-known outlier detection algorithms, Isolation Forest(iForest) and Local Outlier Factor(LOF) have been widely used. However, iForest is only sensitive to global outliers, and is weak in dealing with local outliers. Although LOF performs well in local outlier detection, it has high time complexity. [2] Robust covariance is able to work for Gaussian Distributed data as it will draw ellipse around the data and counts the others as outliers.

1. Implementing Models

The purpose of the three models is to detect anomalies in the data. The following figures show how we will implementing the model.

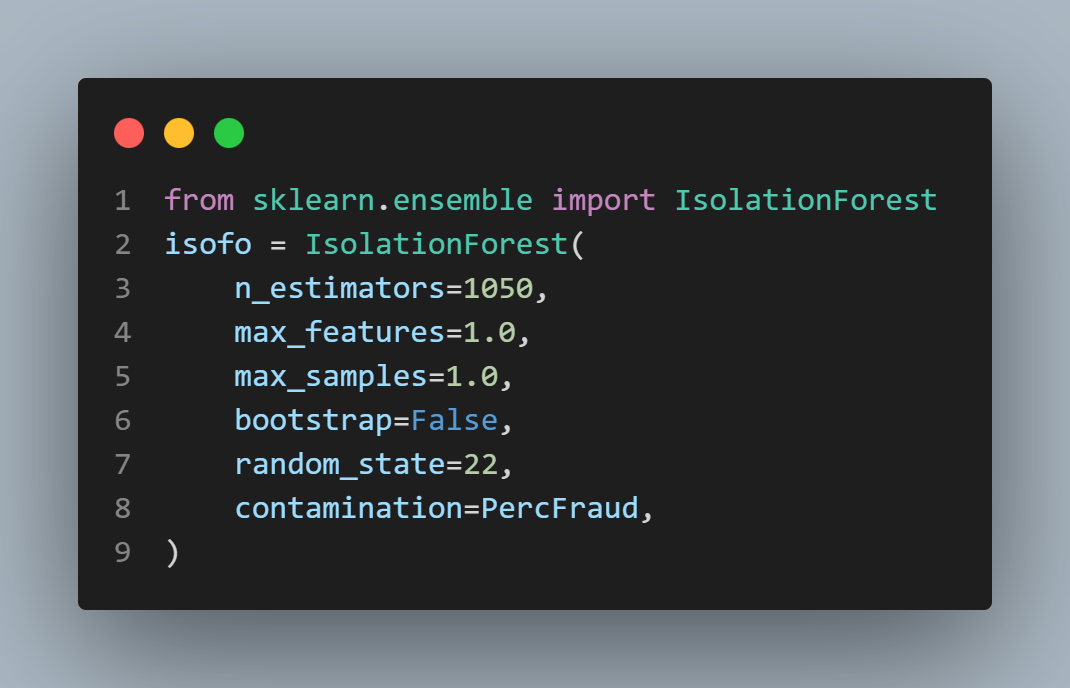


Figure. 7. Code implementation for training model using IsolationForest



Figure. 8. Code implementation for training model using LocalOutlierFactor



Figure. 9. Code implementation for training model using RobustCovariance

1. Results and Discussion

After training the model, we will be using the X data to predict and check if the y\_pred values is correct.

|  |  |  |
| --- | --- | --- |

Figure. 10. Code implementation for model prediction

As we are using IF, LOF and Robust Covariance, all the outliers will be recorded as -1 and we need to convert the values to match the values in the datatset where fraud = 1 then it should be the outlier. Hence we will need to convert.

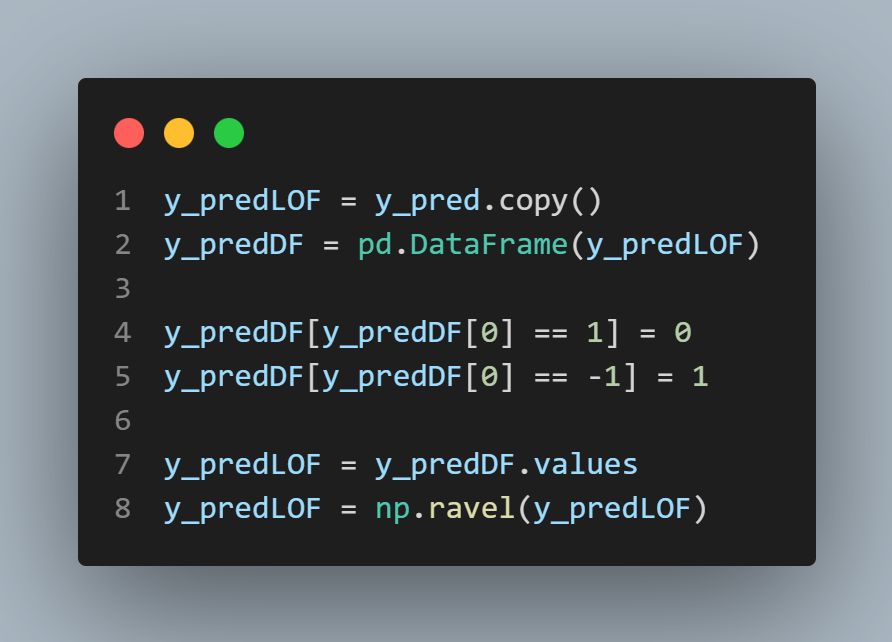


Figure 11, Conversion of outliers and inliers predicted by the models

We use sklearn’s metrics precision\_recall\_fscore\_support to get the values.

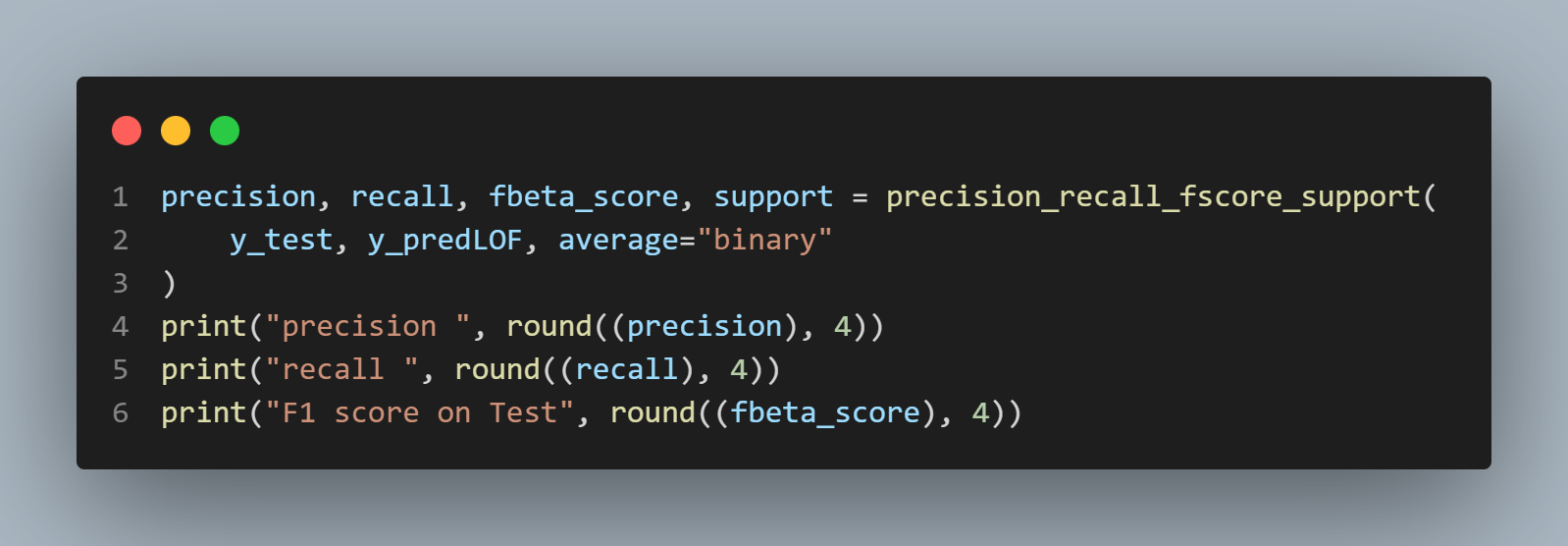


Figure 12, Scoring metrics used to compare the models

We will be using this metrics to compare the models proficiency in detecting outliers. The following shows the data collected.

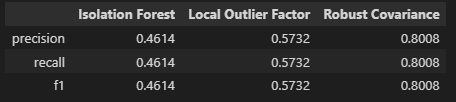


Fig. 11. Scores for each of the different models

Based on the score, this shows that the Robust Covariance is the best model out of the other models. This is likely due to the complexity of the models. Isolation Forest is use to cut data down, Local Outlier checks each individual point and does a check for the surrounding density. Robust Covariance calculate the covariance and draws a ellipse that separates the data. The credit card fraud data after it has been gone through Principal Component Analysis has become more gaussian like which is why the robust covariance matrix is so good at identifying the outliers.

1. Conclusion

In this technical paper, we have investigated the best anomaly detection model for credit card fraud detection which is Robust Covariance. However, more improvements can be made to the models like Hyperparameters Tuning.

##### Acknowledgement

I thank Dr Wilson and Dr Peter, who implemented this technical paper section into the assignment.

##### References

1. Tham, D., 2022. About S$500,000 stolen in fraudulent card payments involving diversion of SMS one-time passwords. [online] CNA. Available at: <https://www.channelnewsasia.com/singapore/credit-card-fraud-banks-divert-sms-otp-overseas-imda-mas-spf-2179541> [Accessed 20 July 2022].
2. Vaishnavi Nath Dornadula, S Geetha, 2019,Credit Card Fraud Detection using Machine Learning Algorithms,Procedia Computer Science, Volume 165, 2019, Pages 631-641, ISSN 1877-0509, Available  
   <https://www.sciencedirect.com/science/article/pii/S187705092030065X>. [Accessed 20 July 2022].
3. Cheng, Zhangyu & Zou, Chengming & Dong, Jianwei. (2019). Outlier detection using isolation forest and local outlier factor. RACS '19: Proceedings of the Conference on Research in Adaptive and Convergent Systems. 161-168. 10.1145/3338840.3355641.
4. Université Libre de Bruxelles, 2021. Credit Card Fraud Detection. [online] Kaggle.com. Available at: <https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud> [Accessed 20 July 2022].
5. Akshara\_416, 2022. Isolation Forest | Anomaly Detection with Isolation Forest. [online] Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2021/07/anomaly-detection-using-isolation-forest-a-complete-guide/> [Accessed 11 August 2022].
6. Regaya, Yousra & Fadli, Fodil & Amira, Abbes. (2021). Point-Denoise: Unsupervised outlier detection for 3D point clouds enhancement. Multimedia Tools and Applications. 80. 1-17. 10.1007/s11042-021-10924-x.
7. scikit-learn. 2022. Outlier detection with Local Outlier Factor (LOF). [online] Available at: <http://scikit-learn.org/stable/auto\_examples/neighbors/plot\_lof\_outlier\_detection.html> [Accessed 12 August 2022].
8. Favaits, M., 2022. #145 - Tutorial | Anomaly Detection | Local Outlier Factor | LOF Algorithm. [online] Youtu.be. Available at: <https://youtu.be/CePgbdVdLvg> [Accessed 12 August 2022].
9. doedotdev, 2022. Local Outlier Factor | Simple Example By Hand. [online] Medium. Available at: <https://doedotdev.medium.com/local-outlier-factor-example-by-hand-b57cedb10bd1> [Accessed 12 August 2022].
10. Tancev, G., 2022. Robust Covariance for Anomaly Detection. [online] Medium. Available at: <https://towardsdatascience.com/robust-covariance-for-anomaly-detection-9c68b1ec4c4b> [Accessed 12 August 2022].
11. scikit-learn. 2022. Robust covariance estimation and Mahalanobis distances relevance. [online] Available at: <https://scikit-learn.org/stable/auto\_examples/covariance/plot\_mahalanobis\_distances.html> [Accessed 12 August 2022].
12. Hubert, M, Debruyne, M, Rousseeuw, PJ. Minimum covariance determinant and extensions. WIREs Comput Stat. 2018;10:e1421. https://doi.org/10.1002/wics.1421