Credit Card Fraud Detection

INTRODUCTION: Outlier Detection and Analysis Methods

Outlier detection in credit card transactions is an important technique used by financial institutions and credit card companies to identify potentially fraudulent or suspicious activities. It involves analyzing transactional data to find unusual patterns or outliers that deviate from the normal behavior of cardholders. By detecting outliers, financial institutions can take timely action to prevent fraudulent transactions and protect their customers.

There are various approaches to performing outlier detection in credit card transactions. Here are some commonly used techniques:

1. Statistical Methods: Statistical methods involve analyzing transactional data using various statistical measures to identify outliers. This can include calculating measures such as mean, standard deviation, and Z-scores to determine if a transaction is significantly different from the average behavior.

2. Machine Learning Techniques: Machine learning algorithms can be trained on historical transaction data to learn patterns and identify outliers. Supervised learning algorithms, such as support vector machines (SVM) or random forests, can be trained on labeled data to classify transactions as normal or fraudulent. Unsupervised learning algorithms, such as clustering or density-based techniques, can also be used to detect anomalies based on deviations from the majority of transactions.

3. Time Series Analysis: Time series analysis techniques can be employed to identify outliers based on temporal patterns. Unusual spikes or dips in transaction volume or frequency at specific times can indicate fraudulent activities.

It's important to note that outlier detection is an ongoing process, and models need to be regularly updated to adapt to evolving fraud patterns. Additionally, combining multiple detection techniques and employing a layered approach can enhance the accuracy of detecting fraudulent transactions while minimizing false positives.

What is an outlier in machine learning?

An outlier is an individual point of data that is distant from other points in the dataset. It is an anomaly in the dataset that may be caused by a range of errors in capturing, processing or manipulating data. Outliers can skew overall data trends, so outlier detection methods are an important part of statistics. Outliers will be a consideration for any area that uses data to make decisions. If an organisation is gaining insight from data, outliers are a real risk.

Outlier detection is particularly important within machine learning. Models are trained on huge arrays of training data. The model understands the relationship between data points to help predict future events or categorise live data. Outliers in the training data may skew the model, lowering its accuracy and overall effectiveness. Outlier analysis and resolution can lengthen the training time too. Outliers can be present in any data or machine learning use case, whether that’s financial modelling or business performance analysis.

Types of Outliers

There are three main types of outliers relevant to machine learning models. Each type differs by how the anomalous data can be observed and what makes the data point stand apart from the rest of the data set. Types are an important consideration for outlier analysis as each has a different pattern to identify.

The three main types of outliers are:

1. Point outliers
2. Contextual outliers
3. Collective outliers

Point Outlier

A point outlier is an individual data point that sits outside of the range of the rest of the dataset. There may be a clear pattern, trend or grouping within the dataset, and an outlier as a data point will be significantly different to this. Point outliers can often be attributed to an error with the measurement or input of the data.

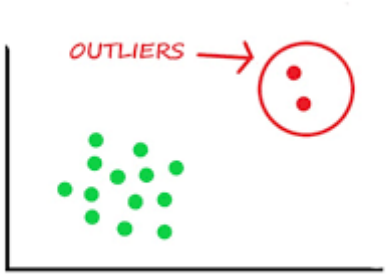

Fig. 1 Point Outlier

For example, an outlier may occur in the health sector data if a mistake is made when recording a unit of measurement within patient data. Missing a digit when recording the height of a patient would cause a noticeable point outlier within the dataset. This type of outlier can be relatively straightforward to identify through visual means. Point outliers are visible if the dataset is plotted across two or three dimensions, as the outlier as a data point would sit far apart from the rest of the dataset.

Contextual Outlier

A contextual outlier is when a data point is significantly different from the dataset, but only within a specific context. The context of a dataset may change seasonally or fluctuate with wider economic trends or behaviour. A contextual outlier will be noticeable when the context of the dataset changes. This could be seasonal weather changes, economic fluctuations, changes in customer behaviour for key holidays, or even the time of the day. For this reason, a contextual outlier may seem like a normal data point in other contexts.

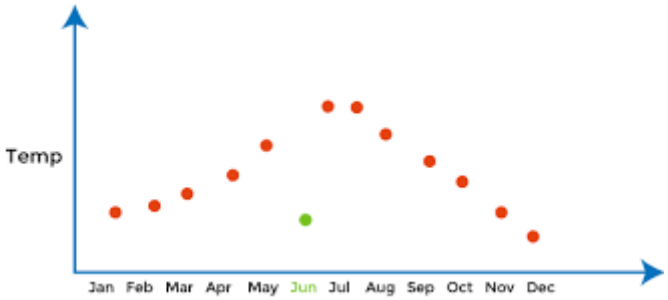

Fig. 2 *Contextual Outlier*

For example, in a dataset of UK temperatures over time, encompassing different years and seasons. A temperature reading below zero degrees at noon could be seen as normal during the winter. But this same reading would be deemed a contextual outlier if recorded at the height of summer during a heat wave. The data is contextualized within wider trends that are impacting the dataset.

Collective Outlier

A collective outlier is when a series of data points differ significantly from the trends in the rest of the dataset. The individual data points within a collective outlier may not seem like a point outlier or a contextual outlier. It's when the data points are considered as a collection that anomalous patterns are observed. For this reason collective outliers can be the hardest type of outlier to identify. Collective outliers are an integral part of monitoring for concept drift in machine learning. A sequence of data has shifted away from expected behaviour within the model.

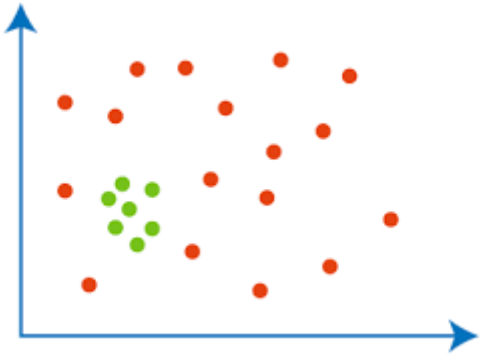

Fig. 3 *Collective Outlier*

For example, a time series plotting subscribers and unsubscribers to an email marketing list showing seasonal or daily fluctuations. A collective outlier could be flagged if the level of subscribed users stayed entirely static for many weeks with no fluctuation. Individual users unsubscribing and new users subscribing is a normal occurrence so a static count would be flagged as an outlier.

Taken in isolation, each data point is within the expected boundaries of the data so would not be flagged as a contextual or point outlier. But when taken as a series, the data's behaviour is flagged as anomalous. Once a collective outlier is identified, steps can be taken to investigate any systematic errors in the process.

Problem Statement

The objective of this project is to develop a credit card fraud detection system using the dataset from Kaggle. The dataset contains a large number of credit card transactions, with both fraudulent and non-fraudulent cases. The goal is to build a machine learning model that can accurately identify fraudulent transactions to help financial institutions mitigate the risk and minimize losses due to fraudulent activities.

Dataset

Kaggle

It is important that credit card companies are able to recognize fraudulent credit card transactions so that customers are not charged for items that they did not purchase.

The dataset contains transactions made by credit cards in September 2013 by European cardholders.

This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) accounts for 0.172% of all transactions.

It contains only numerical input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, we cannot provide the original features and more background information about the data. Features V1, V2, … V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for example-dependent cost-sensitive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

[4]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 284807 entries, 0 to 284806
Data columns (total 31 columns):
 #   Column  Non-Null Count   Dtype  
---  ------  --------------   -----  
 0   Time    284807 non-null  float64
 1   V1      284807 non-null  float64
 2   V2      284807 non-null  float64
 ...
28  V28     284807 non-null  float64
29  Amount  284807 non-null  float64
30  Class   284807 non-null  int64  
dtypes: float64(30), int64(1)
memory usage: 67.4 MB
```

Given the class imbalance ratio, we recommend measuring the accuracy using the Area Under the Precision-Recall Curve (AUPRC). Confusion matrix accuracy is not meaningful for unbalanced classification.

Exploratory Data Analysis (EDA)

Analyzing the dataset to gain insights into the distribution of fraudulent and non-fraudulent transactions, identifying any class imbalance issues, and understanding the features' characteristics. Cleaning the dataset, handling missing values, scaling numerical features, and encoding categorical variables as necessary. Selecting relevant features and creating new informative features that can potentially improve the model's performance in detecting fraudulent transactions.

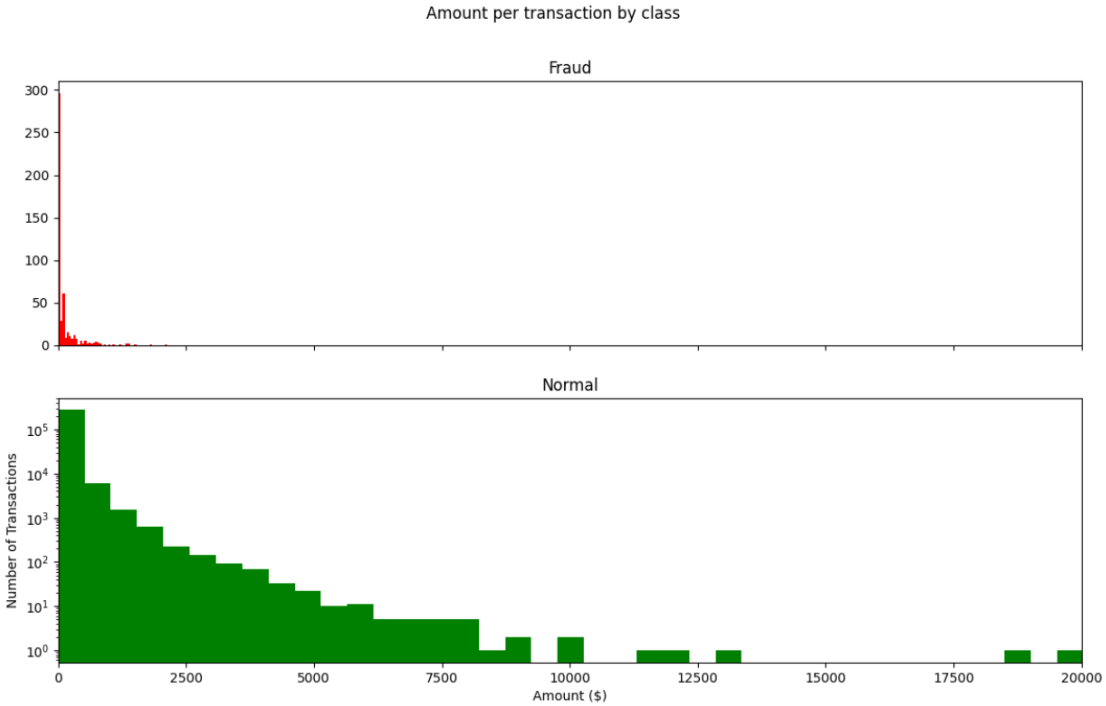

Fig. 5 *Amount per transaction by class*

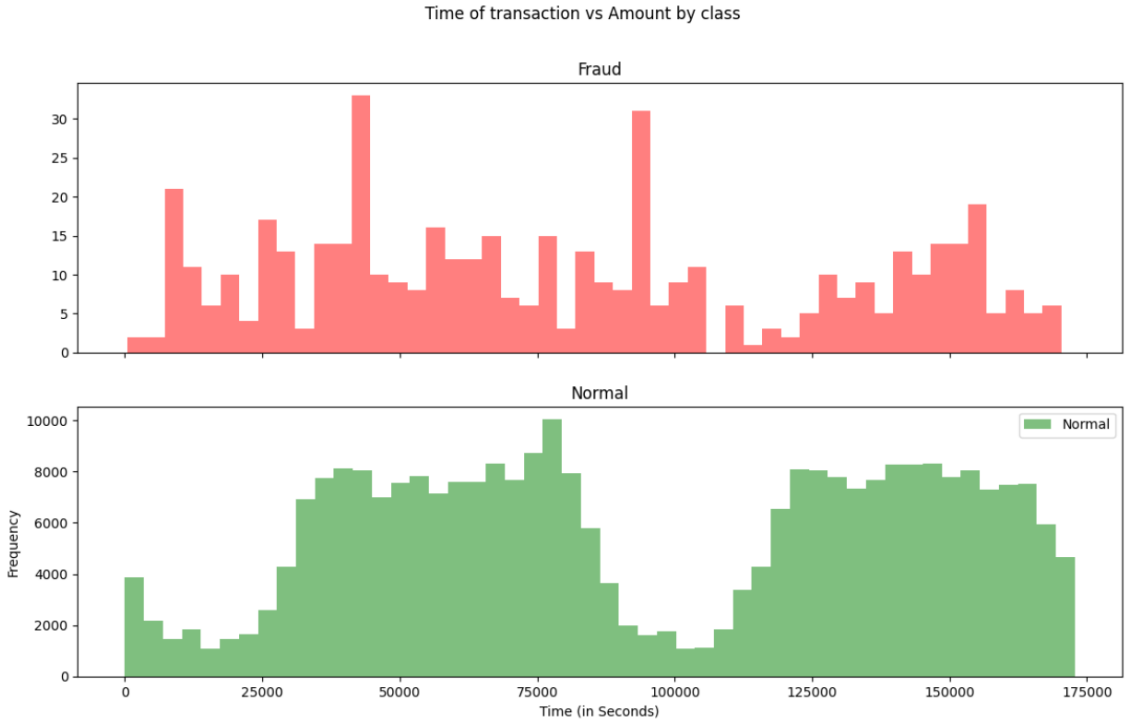

Fig. 6 *Time of transaction vs Amount by class*

Model & Analysis

Isolation Forest

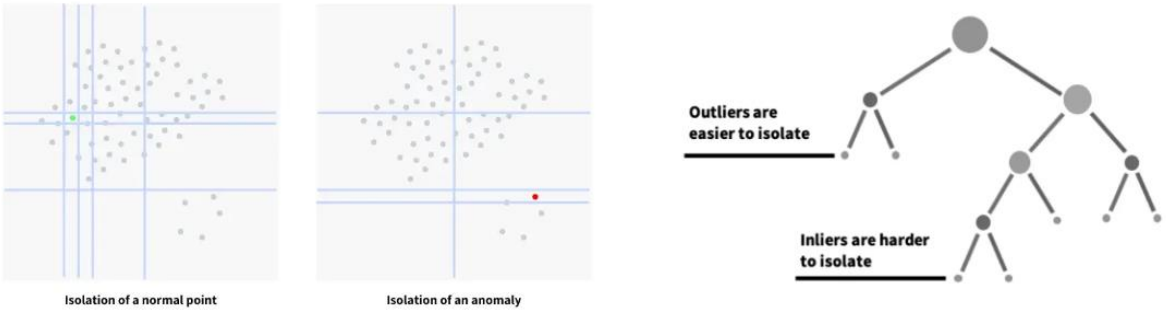

Fig. 7 *Isolation Forest*

Isolation Forest is an unsupervised machine-learning algorithm used for outlier detection. It leverages the concept of isolating anomalies to detect outliers in a dataset. It builds an ensemble of isolation trees, which are binary trees that randomly partition the data points. Anomalies are expected to have shorter average path lengths in the trees, making them easier to isolate. By assigning anomaly scores based on the average path lengths, the algorithm identifies outliers as data points with low scores. Isolation Forest is particularly effective in high-dimensional datasets and does not rely on assumptions about the data distribution.

Local Outlier Factor (LOF)

LOF (Local Outlier Factor) is an unsupervised anomaly detection algorithm that assesses the local density deviation of a data point compared to its neighbors. It quantifies the degree of abnormality of a data point based on its relative density.

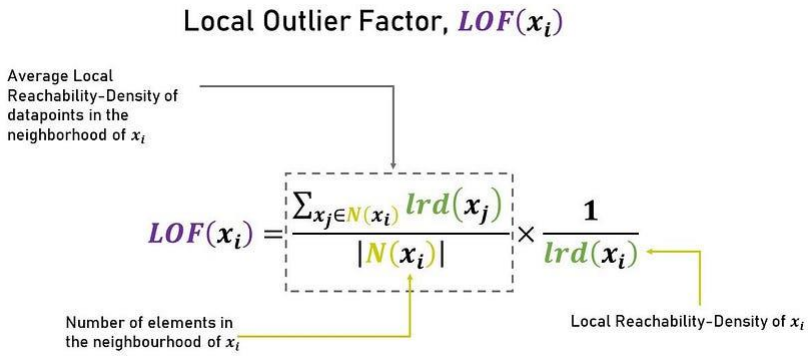

Fig. 8 *Local Outlier Factor*

The LOF algorithm works by calculating a local reachability density for each data point, which represents how isolated or tightly grouped the point is compared to its neighbors. Anomaly scores are assigned based on the degree to which a point's density deviates from the density of its neighbors. Points with significantly lower density compared to their neighbors are considered outliers with higher LOF scores.

LOF is effective in identifying anomalies in datasets with varying densities or clusters of different sizes. It can handle data with complex structures and does not rely on strict assumptions about the data distribution. LOF provides a local perspective on anomalies, allowing for more fine-grained anomaly detection in the dataset.

One Class Support Vector Machine (SVM)

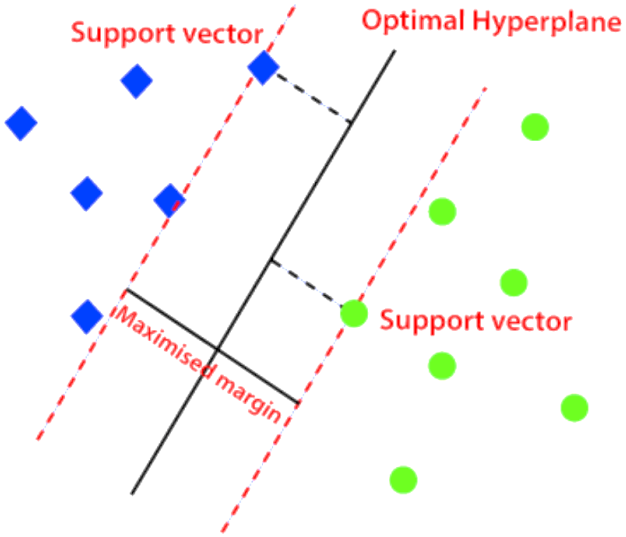

Fig. 9 *One Class SVM*

SVM (Support Vector Machine) is a supervised machine learning algorithm used for classification and regression tasks. It finds an optimal hyperplane that separates data points into different classes or predicts continuous values. SVM aims to maximize the margin between classes, making it robust to outliers. It can handle linearly separable data and can also utilize kernels to handle non-linearly separable data by mapping it to a higher-dimensional feature space. SVM is effective in high-dimensional spaces, but it can be computationally expensive for large datasets due to its quadratic time complexity.

Accuracy

```
Isolation Forest: 73
Accuracy Score :
0.9974368877497279
Classification Report :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     28432
           1       0.26      0.27      0.26        49

    accuracy                           1.00     28481
   macro avg       0.63      0.63      0.63     28481
weighted avg       1.00      1.00      1.00     28481


Local Outlier Factor: 97
Accuracy Score :
0.9965942207085425
Classification Report :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     28432
           1       0.02      0.02      0.02        49

    accuracy                           1.00     28481
   macro avg       0.51      0.51      0.51     28481
weighted avg       1.00      1.00      1.00     28481
```

```
Support Vector Machine: 8516

Accuracy Score :

0.7009936448860644

Classification Report :

              precision    recall  f1-score   support

           0       1.00      0.70      0.82     28432
           1       0.00      0.37      0.00        49

    accuracy                           0.70     28481
   macro avg       0.50      0.53      0.41     28481
weighted avg       1.00      0.70      0.82     28481
```

Concluding Discussion

- **Isolation Forest Outperforms Others:** Isolation Forest detected fewer errors (73) compared to Local Outlier Factor (97) and Support Vector Machine (8516), showcasing its superiority in anomaly detection.

- **High Accuracy of Isolation Forest:** Isolation Forest achieved an accuracy of 99.74%, outperforming both Local Outlier Factor (99.65%) and Support Vector Machine (70.09%).

- **Better Precision and Recall with Isolation Forest:** Isolation Forest exhibited superior precision and recall for fraud cases (27%) compared to Local Outlier Factor (2%) and Support Vector Machine (0%), indicating its effectiveness in identifying fraud instances.

- **Overall Performance of Isolation Forest:** The Isolation Forest method performed exceptionally well in detecting fraud cases, achieving an overall detection rate of around 30%.

- **Potential Improvements:** To enhance classifier performance, consider increasing the sample size or exploring deep learning algorithms. Additionally, complex anomaly detection models may be employed for improved accuracy, acknowledging the associated computational costs.

References

1. https://github.com/krishnaik06/Credit-Card-Fraudlent
2. https://towardsdatascience.com/5-anomaly-detection-algorithms-every-data-scientist-should-know-b36c3605ea16