Context

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.

From the moment the payment systems came to existence, there have always been people who will find new ways to access someone’s finances illegally. This has become a major problem in the modern era, as all transactions can easily be completed online by only entering your credit card information. Even in the 2010s, many American retail website users were the victims of online transaction fraud right before two-step verification was used for shopping online. Organizations, consumers, banks, and merchants are put at risk when a data breach leads to monetary theft and ultimately the loss of customers’ loyalty along with the company’s reputation.

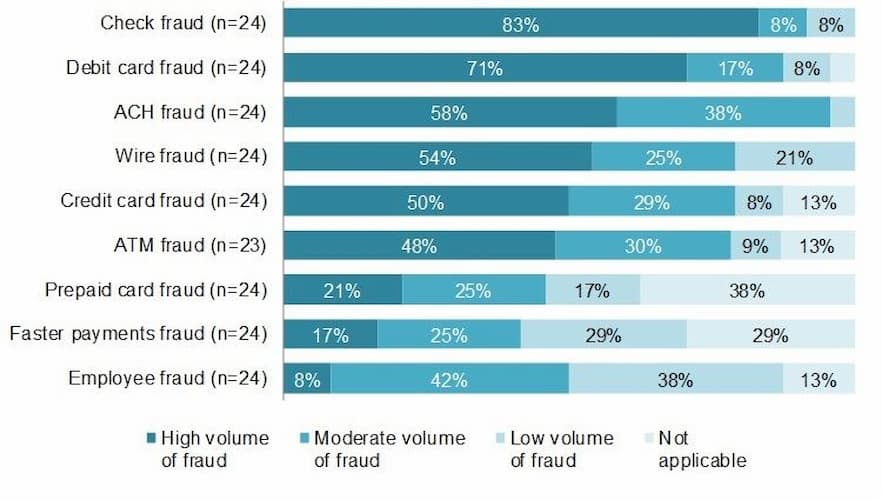
## What is the difference between ML Credit Card Fraud Detection and Conventional Fraud Detection?

Machine Learning-based Fraud Detection:

* Detecting fraud automatically
* Streaming and the ability to detect online fraud in real-time
* Less time needed for verification methods
* Identifying hidden correlations in data

Conventional Fraud Detection:

* The rules of making a decision on determining schemes should be set manually.
* Takes an enormous amount of time
* Multiple verification methods are needed; thus, inconvenient for the user
* Finds only obvious fraud activities



## What is Credit Card Fraud Detection and Prevention?

“Fraud detection is a set of activities that are taken to prevent money or property from being obtained through false pretenses.”

Fraud can be committed in different ways and in many industries. The majority of detection methods combine a variety of fraud detection datasets to form a connected overview of both valid and non-valid payment data to make a decision. This decision must consider IP address, geolocation, device identification, “BIN” data, global latitude/longitude, historic transaction patterns, and the actual transaction information. In practice, this means that merchants and issuers deploy analytically based responses that use internal and external data to apply a set of business rules or analytical algorithms to detect fraud.

**Credit Card Fraud Detection with Machine Learning** is a process of data investigation by a Data Science team and the development of a model that will provide the best results in revealing and preventing fraudulent transactions. This is achieved through bringing together all meaningful features of card users’ transactions, such as Date, User Zone, Product Category, Amount, Provider, Client’s Behavioral Patterns, etc. The information is then run through a subtly trained model that finds patterns and rules so that it can classify whether a transaction is fraudulent or is legitimate. Now you know what is fraud protection, let’s look at the most common types of threats.

## The Major Types of Credit Card Fraud

### **Clone transactions.**

Clone transactions are popular among the different types of credit card frauds. It simply means making transactions similar to an original one or duplicating a transaction. This can happen when an organization tries to get payment from a partner multiple times by sending the same invoice to different departments.

The conventional method of rule-based fraud detection algorithm does not work well to distinguish a fraudulent transaction from irregular or mistaken transactions. For instance, a user could click the submission button two times by accident or order the same product twice.  
The better option is if a system is capable of differentiating a fraudulent transaction from one made in error. Here, Machine Learning fraud detection methods would be more potent in differentiating clone transactions caused by human error and real fraud.

### **Account theft and suspicious transactions.**

When an individual’s personal information such as a Social Security number, a secret question answer, or date of birth is stolen by criminals, they can use this information to perform financial operations. A lot of fraud transactions are linked to identity theft, so financial fraud detection systems should pay the most attention to creating an analysis of a user’s behavior.

If there is a certain regularity in the way a client makes his payments, e. g. someone visits a certain bar once a week at the same time and always spends about $40 to $60. If the same account is used to make a payment at a bar located in another part of town and for a sum of more than $60, this behavior would be considered irregular. The next move would be to send a verification request to the card number owner in order to validate that he or she made the transaction.

Metrics such as standard deviation, averages, and high/low values are the most useful to spot irregular behavior. Separate payments are compared with personal benchmarks to identify transactions with a high standard deviation.

Then, the best choice is to validate the account holder if such a deviation occurs.

### **False application fraud.**

Credit card application fraud is often accompanied by account/identity/credit card theft. It means that someone applies for a new credit account or credit card in another person’s name. First, criminals steal the documents which will serve as supporting evidence for their fake application.

Anomaly detection helps to identify whether a transaction has any unusual patterns, such as date and time or the number of goods. If the algorithm spots such unusual behavior, the owner of the bank account will be protected by a few verification methods. This is a great method for credit fraud prevention.

### **Credit Card Skimming (electronic or manual).**

Credit card skimming or credit card forgery means making an illegal copy of a credit or bank card with a device that reads and duplicates information from the original card. Credit card scammers use machines named “skimmers” to extract card numbers and other credit card information, save it, and resell to criminals.

As in the case of identity theft, suspicious transactions made from a copy of an electronic or manual card will be revealed because of the information on the transaction. Classification techniques can define whether a transaction is fraudulent based on hardware, geolocation, and information about a client’s behavior patterns.

### **Account takeover.**

Last but not least, among types of credit card fraud is account takeover. Fraudsters can send deceptive emails to cardholders. The messages look pretty legitimate (e.g. very similar bank URLs and trustworthy logos), as if they were sent by the bank. In reality, such a message can be used to steal someone’s personal information, bank account numbers, and online passwords. If you click the wrong link or provide valuable information in response to a message from a fake bank website, within a couple of hours your bank account will be drained by the criminals into an account they hold.

To avoid this, AI-driven solutions rely on neural networks or pattern recognition. Neural networks can learn suspicious-looking patterns as well as to detect classes and clusters to use these patterns for fraud detection.

How Does Credit Card Fraud Happen?

Credit card fraud is usually caused either by card owner’s negligence with his data or by a breach in a website’s security. Here are some examples:

* A consumer reveals his credit card number to unfamiliar individuals.
* A card is lost or stolen and someone else uses it.
* Mail is stolen from the intended recipient and used by criminals.
* Business employees copy cards or card numbers of its owner.
* Making counterfeit credit cards.

When your card is lost or stolen, an unauthorized charge can happen; in other words, the person who finds it uses it for a purchase. Criminals can also forge your name and use the card or order some goods through a mobile phone or computer. Also, there is the problem of using a counterfeit credit card – a fake card that has the real account information that was stolen from holders. That is especially dangerous because the victims have their real cards, but do not know that someone has copied their card. Such fraudulent cards look quite legitimate and have the logos and encoded magnetic strips of the original one. Fraudulent credit cards are usually destroyed by the criminals after several successful payments, just before a victim realizes the problem and reports it.

**MODEL IMPLEMENATION:**

## Gather Sense of Our Data:

The first thing we must do is gather a **basic sense**of our data. Remember, except for the **transaction** and **amount** we dont know what the other columns are (due to privacy reasons). The only thing we know, is that those columns that are unknown have been scaled already.

### **Summary:**

* The transaction amount is relatively **small**. The mean of all the mounts made is approximately USD 88.
* There are no **"Null"** values, so we don't have to work on ways to replace values.
* Most of the transactions were **Non-Fraud** (99.83%) of the time, while **Fraud** transactions occurs (017%) of the time in the dataframe.

### **Feature Technicalities:**

* **PCA Transformation:**The description of the data says that all the features went through a PCA transformation (Dimensionality Reduction technique) (Except for time and amount).
* **Scaling:** Keep in mind that in order to implement a PCA transformation features need to be previously scaled. (In this case, all the V features have been scaled or at least that is what we are assuming the people that develop the dataset did.)

## Scaling and Distributing

In this phase of our kernel, we will first scale the columns comprise of **Time** and **Amount**. Time and amount should be scaled as the other columns. On the other hand, we need to also create a sub sample of the dataframe in order to have an equal amount of Fraud and Non-Fraud cases, helping our algorithms better understand patterns that determines whether a transaction is a fraud or not.

### **What is a sub-Sample?**

In this scenario, our subsample will be a dataframe with a 50/50 ratio of fraud and non-fraud transactions. Meaning our sub-sample will have the same amount of fraud and non fraud transactions.

### **Why do we create a sub-Sample?**

In the beginning of this notebook we saw that the original dataframe was heavily imbalanced! Using the original dataframe will cause the following issues:

* **Overfitting:**Our classification models will assume that in most cases there are no frauds! What we want for our model is to be certain when a fraud occurs.
* **Wrong Correlations:** Although we don't know what the "V" features stand for, it will be useful to understand how each of this features influence the result (Fraud or No Fraud) by having an imbalance dataframe we are not able to see the true correlations between the class and features.

### **Summary:**

* **Scaled amount**and **scaled time**are the columns with scaled values.
* There are **492 cases**of fraud in our dataset so we can randomly get 492 cases of non-fraud to create our new sub dataframe.
* We concat the 492 cases of fraud and non fraud, **creating a new sub-sample.**

### **Splitting the Data (Original DataFrame)**

Before proceeding with the **Random UnderSampling technique** we have to separate the orginal dataframe. **Why? for testing purposes, remember although we are splitting the data when implementing Random UnderSampling or OverSampling techniques, we want to test our models on the original testing set not on the testing set created by either of these techniques.** The main goal is to fit the model either with the dataframes that were undersample and oversample (in order for our models to detect the patterns), and test it on the original testing set.

## Random Under-Sampling:

In this phase of the project we will implement \*"Random Under Sampling"\* which basically consists of removing data in order to have a more **balanced dataset**and thus avoiding our models to overfitting.

#### **Steps:**

* The first thing we have to do is determine how **imbalanced** is our class (use "value\_counts()" on the class column to determine the amount for each label)
* Once we determine how many instances are considered **fraud transactions**(Fraud = "1") , we should bring the **non-fraud transactions** to the same amount as fraud transactions (assuming we want a 50/50 ratio), this will be equivalent to 492 cases of fraud and 492 cases of non-fraud transactions.
* After implementing this technique, we have a sub-sample of our dataframe with a 50/50 ratio with regards to our classes. Then the next step we will implement is to **shuffle the data** to see if our models can maintain a certain accuracy everytime we run this script.

**Note:** The main issue with "Random Under-Sampling" is that we run the risk that our classification models will not perform as accurate as we would like to since there is a great deal of **information loss** (bringing 492 non-fraud transaction from 284,315 non-fraud transaction)

## Anomaly Detection:

Our main aim in this section is to remove "extreme outliers" from features that have a high correlation with our classes. This will have a positive impact on the accuracy of our models.

### **Interquartile Range Method:**

* **Interquartile Range (IQR):**We calculate this by the difference between the 75th percentile and 25th percentile. Our aim is to create a threshold beyond the 75th and 25th percentile that in case some instance pass this threshold the instance will be deleted.
* **Boxplots:**Besides easily seeing the 25th and 75th percentiles (both end of the squares) it is also easy to see extreme outliers (points beyond the lower and higher extreme).

### **Outlier Removal Tradeoff:**

We have to be careful as to how far do we want the threshold for removing outliers. We determine the threshold by multiplying a number (ex: 1.5) by the (Interquartile Range). The higher this threshold is, the less outliers will detect (multiplying by a higher number ex: 3), and the lower this threshold is the more outliers it will detect.

\*The Tradeoff: \* The lower the threshold the more outliers it will remove however, we want to focus more on "extreme outliers" rather than just outliers. Why? because we might run the risk of information loss which will cause our models to have a lower accuracy. You can play with this threshold and see how it affects the accuracy of our classification models.

### **Summary:**

* **Visualize Distributions:**We first start by visualizing the distribution of the feature we are going to use to eliminate some of the outliers. V14 is the only feature that has a Gaussian distribution compared to features V12 and V10.
* **Determining the threshold:**After we decide which number we will use to multiply with the iqr (the lower more outliers removed), we will proceed in determining the upper and lower thresholds by substrating q25 - threshold (lower extreme threshold) and adding q75 + threshold (upper extreme threshold).
* **Conditional Dropping:**Lastly, we create a conditional dropping stating that if the "threshold" is exceeded in both extremes, the instances will be removed.
* **Boxplot Representation:**Visualize through the boxplot that the number of "extreme outliers" have been reduced to a considerable amount.

**Note:** After implementing outlier reduction our accuracy has been improved by over 3%! Some outliers can distort the accuracy of our models but remember, we have to avoid an extreme amount of information loss or else our model runs the risk of underfitting.

Classifiers (UnderSampling):

In this section we will train four types of classifiers and decide which classifier will be more effective in detecting **fraud transactions**. Before we have to split our data into training and testing sets and separate the features from the labels.

Summary:

* **Logistic Regression**classifier is more accurate than the other three classifiers in most cases. (We will further analyze Logistic Regression)
* **GridSearchCV**is used to determine the paremeters that gives the best predictive score for the classifiers.
* Logistic Regression has the best Receiving Operating Characteristic score (ROC), meaning that LogisticRegression pretty accurately separates **fraud**and **non-fraud**transactions.

Learning Curves:

* The **wider the gap** between the training score and the cross validation score, the more likely your model is **overfitting (high variance)**.
* If the score is low in both training and cross-validation sets this is an indication that our model is **underfitting (high bias)**
* **Logistic Regression Classifier** shows the best score in both training and cross-validating sets.

## A Deeper Look into LogisticRegression:

In this section we will ive a deeper look into the **logistic regression classifier**.

### **Terms:**

* **True Positives:** Correctly Classified Fraud Transactions
* **False Positives:** Incorrectly Classified Fraud Transactions
* **True Negative:** Correctly Classified Non-Fraud Transactions
* **False Negative:** Incorrectly Classified Non-Fraud Transactions
* **Precision:**True Positives/(True Positives + False Positives)
* **Recall:**True Positives/(True Positives + False Negatives)
* Precision as the name says, says how precise (how sure) is our model in detecting fraud transactions while recall is the amount of fraud cases our model is able to detect.
* **Precision/Recall Tradeoff:**The more precise (selective) our model is, the less cases it will detect. Example: Assuming that our model has a precision of 95%, Let's say there are only 5 fraud cases in which the model is 95% precise or more that these are fraud cases. Then let's say there are 5 more cases that our model considers 90% to be a fraud case, if we lower the precision there are more cases that our model will be able to detect.

### **Summary:**

* **Precision starts to descend** between 0.90 and 0.92 nevertheless, our precision score is still pretty high and still we have a descent recall score.

# **Test Data with Logistic Regression:**

## Confusion Matrix:

**Positive/Negative:** Type of Class (label) ["No", "Yes"] **True/False:** Correctly or Incorrectly classified by the model.

**True Negatives (Top-Left Square):** This is the number of **correctly** classifications of the "No" (No Fraud Detected) class.

**False Negatives (Top-Right Square):** This is the number of **incorrectly** classifications of the "No"(No Fraud Detected) class.

**False Positives (Bottom-Left Square):** This is the number of **incorrectly** classifications of the "Yes" (Fraud Detected) class

**True Positives (Bottom-Right Square):** This is the number of **correctly** classifications of the "Yes" (Fraud Detected) class.

### **Summary:**

* **Random UnderSampling:** We will evaluate the final performance of the classification models in the random undersampling subset. **Keep in mind that this is not the data from the original dataframe.**
* **Classification Models:**The models that performed the best were **logistic regression**and **support vector classifier (SVM)**

## Neural Networks Testing Random UnderSampling Data vs OverSampling (SMOTE):

In this section we will implement a simple Neural Network (with one hidden layer) in order to see which of the two logistic regressions models we implemented in the (undersample or oversample(SMOTE)) has a better accuracy for detecting fraud and non-fraud transactions.

### **Our Main Goal:**

Our main goal is to explore how our simple neural network behaves in both the random undersample and oversample dataframes and see whether they can predict accuractely both non-fraud and fraud cases. Why not only focus on fraud? Imagine you were a cardholder and after you purchased an item your card gets blocked because the bank's algorithm thought your purchase was a fraud. That's why we shouldn't emphasize only in detecting fraud cases but we should also emphasize correctly categorizing non-fraud transactions.

### **The Confusion Matrix:**

Here is again, how the confusion matrix works:

* **Upper Left Square:**The amount of **correctly** classified by our model of no fraud transactions.
* **Upper Right Square:** The amount of **incorrectly**classified transactions as fraud cases, but the actual label is **no fraud**.
* **Lower Left Square:** The amount of **incorrectly**classified transactions as no fraud cases, but the actual label is **fraud**.
* **Lower Right Square:** The amount of **correctly** classified by our model of fraud transactions.

### **Summary (Keras || Random UnderSampling):**

* **Dataset:**In this final phase of testing we will fit this model in both the **random undersampled subset** and **oversampled dataset (SMOTE)**in order to predict the final result using the **original dataframe testing data.**
* **Neural Network Structure:**As stated previously, this will be a simple model composed of one input layer (where the number of nodes equals the number of features) plus bias node, one hidden layer with 32 nodes and one output node composed of two possible results 0 or 1 (No fraud or fraud).
* **Other characteristics:** The learning rate will be 0.001, the optimizer we will use is the AdamOptimizer, the activation function that is used in this scenario is "Relu" and for the final outputs we will use sparse categorical cross entropy, which gives the probability whether an instance case is no fraud or fraud (The prediction will pick the highest probability between the two.)

Content

The datasets 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) account 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-dependant cost-senstive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

Inspiration

Identify fraudulent credit card transactions.

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.

Acknowledgements

The dataset has been collected and analysed during a research collaboration of Worldline and the Machine Learning Group ([http://mlg.ulb.ac.be](http://mlg.ulb.ac.be/)) of ULB (Université Libre de Bruxelles) on big data mining and fraud detection.  
More details on current and past projects on related topics are available on <https://www.researchgate.net/project/Fraud-detection-5> and the page of the [DefeatFraud](https://mlg.ulb.ac.be/wordpress/portfolio_page/defeatfraud-assessment-and-validation-of-deep-feature-engineering-and-learning-solutions-for-fraud-detection/" \t "_blank) project