***Scope :***

Through this research article and project we can create a prototype intelligent credit card fraud system to tell the occurrence of fraud by credit card companies so that customers don't have to pay for the items they didn't buy. This system can extract hidden knowledge (patterns and relationships ) associated with credit card frauds from the given past 2 days' transactions via customers by credit cards database.

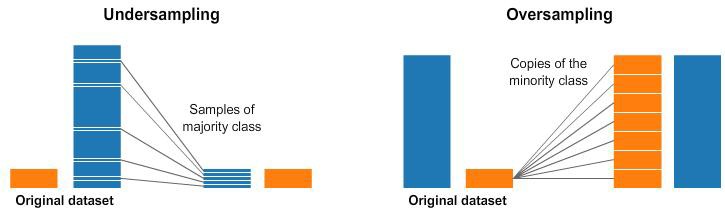
***Introduction:***

Credit card fraud has become common now a days. Credit card fraud is an inclusive term for fraud committed employing a payment card, like a credit card or debit card.The purpose is also to get goods or services or to create payment to a different account, which is controlled by a criminal. The Payment Card Industry Data Security Standard (PCI DSS) is that the data security standard created to assist financial institutions process card payments securely and reduce card fraud. Credit card fraud is authorised, where the real customer themselves processes payment to a different account which is controlled by a criminal, or unauthorised, where the account holder doesn't provide authorisation for the payment to proceed and also the transaction is administered by a third party. Credit card fraud can occur when unauthorized users gain access to a person's credit card information so as to form purchases, other transactions, or open new accounts. A few samples of credit card fraud include account takeover fraud, new account fraud, cloned cards, and cards-not-present schemes. This unauthorized access occurs through phishing, skimming, and data sharing by a user, oftentimes unknowingly. Nevertheless, we are able to still analyze some important aspects of the dataset. Regulators, card providers and banks take considerable time and energy to collaborate with investigators worldwide with the goal of ensuring fraudsters aren't successful. Cardholders' money is typically protected against scammers with regulations that make the card provider and bank accountable. The technology and security measures behind credit cards are continuously advancing, adding barriers for fraudsters attempting to steal money. In this kernel we are going to use various predictive models to work out how accurate they're in detecting whether a transaction is a normal payment or a fraud. As described within the dataset, the features are scaled and therefore the names of the features don't seem to be shown due to privacy reasons.

**Objective:**

# Resampling Technique

A widely adopted technique for dealing with highly unbalanced datasets is called resampling. It consists of removing samples from the majority class (under-sampling) and/or adding more examples from the minority class (over-sampling).



Despite the advantage of balancing classes, these techniques also have their weaknesses (there is no free lunch).

The simplest implementation of **over-sampling** is to duplicate random records from the minority class, which can cause overfishing.

In **under-sampling**, the simplest technique involves removing random records from the majority class, which can cause loss of information.

**These are some technique :**

## Random Under-Sampling

Under sampling can be defined as **removing some observations of the majority class**. This is done until the majority and minority class is balanced out.

Under sampling can be a good choice when you have a ton of data -think millions of rows. But a drawback to under sampling is that we are removing information that may be valuable.

## Random Over-Sampling

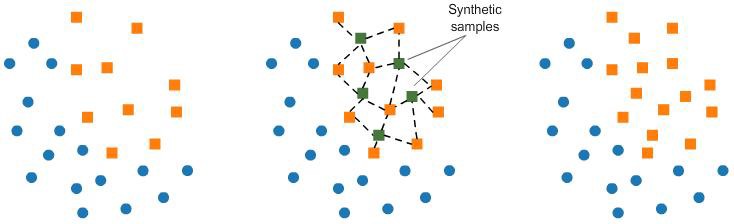
Oversampling can be defined as adding more copies to the minority class. Oversampling can be a good choice when you don’t have a ton of data to work with.

A con to consider when undersampling is that it can cause overfitting and poor generalization to your test set.

## Synthetic Minority Oversampling Technique (SMOTE)

This technique generates synthetic data for the minority class.

SMOTE (Synthetic Minority Oversampling Technique) works by randomly picking a point from the minority class and computing the k-nearest neighbors for this point. The **synthetic points are added** between the chosen point and its neighbors.



**SMOTE algorithm** works in 4 simple steps:

* 1. Choose a minority class as the input vector
  2. Find its k nearest neighbors (**k\_neighbors** is specified as an argument in the **SMOTE()** function)
  3. Choose one of these neighbors and place a synthetic point anywhere on the line joining the point under consideration and its chosen neighbor
  4. Repeat the steps until data is balanced

## Penalize Algorithms (Cost-Sensitive Training)

The next tactic is to use penalized learning algorithms that increase the cost of classification mistakes on the minority class.

A popular algorithm for this technique is Penalized-SVM.

During training, we can use the argument class\_weight=’balanced’ to penalize mistakes on the minority class by an amount proportional to how under-represented it is.

We also want to include the argument probability=True if we want to enable probability estimates for SVM algorithms.

## Change the algorithm

While in every machine learning problem, it’s a good rule of thumb to try a variety of algorithms, it can be especially beneficial with imbalanced datasets.

Decision trees frequently perform well on imbalanced data. In modern machine learning, tree ensembles (Random Forests, Gradient Boosted Trees, etc.) almost always outperform singular decision trees, so we’ll jump right into those:

Tree base algorithm work by learning a hierarchy of if/else questions. This can force both classes to be addressed.

## Advantage and disadvantages of Under-sampling Advantages

* It can help improve run time and storage problems by reducing the number of training data samples when the training data set is huge.

## Disadvantages

* It can discard potentially useful information which could be important for building rule classifiers.
* The sample chosen by random under-sampling may be a biased sample. And it will not be an accurate representation of the population. Thereby, resulting in inaccurate results with the actual test data set.

## Advantages and Disadvantage of over-sampling

**Advantages**

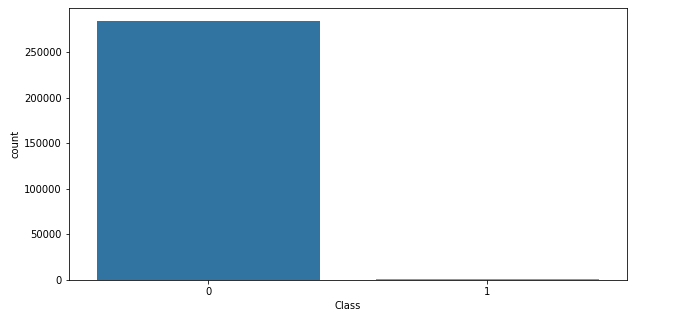
* Unlike under-sampling, this method leads to no information loss.
* Outperforms under sampling

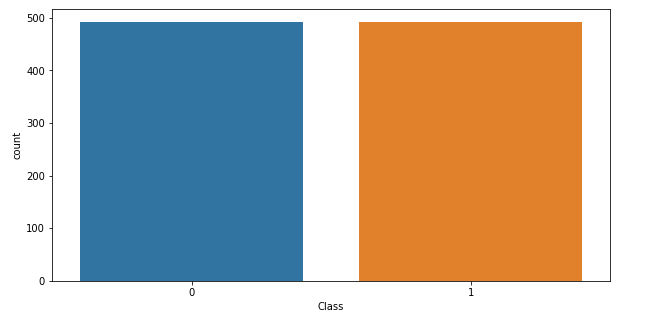
## Disadvantages

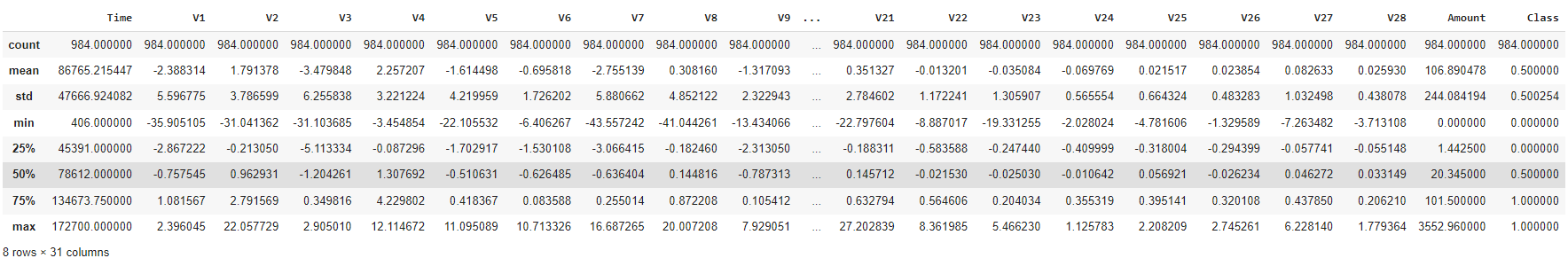
* It increases the likelihood of overfitting since it replicates the minority class events.

***ANALYSIS:***

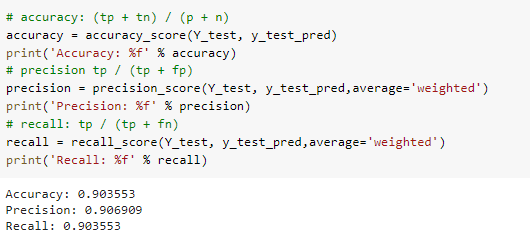
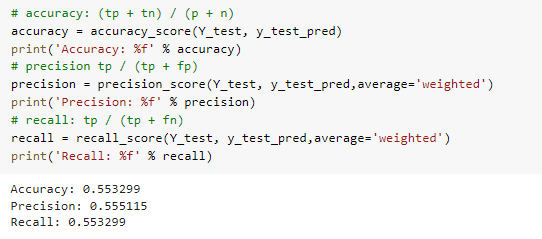
Old Dataset: New Dataset:



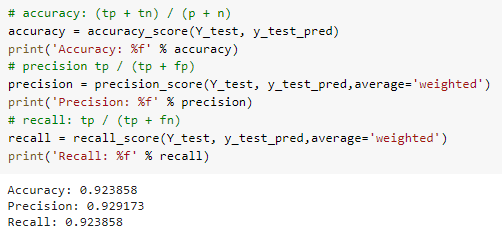
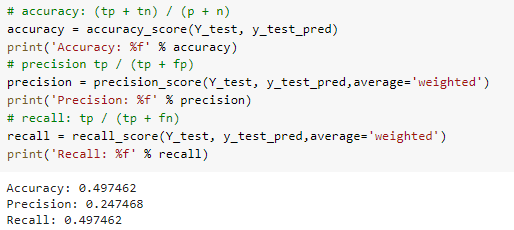


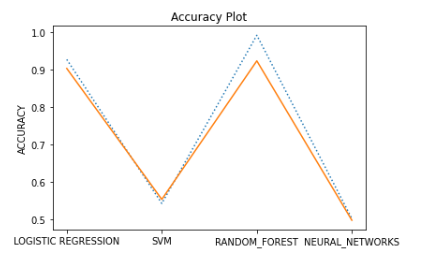




***LIBRARIES IMPORTED :***

1. **Pandas ->**

* Pandas Is a software library written for python programming, for data manipulation and analysis.
* It is widely used for data structures and operation for manipulating numerical tables.
* Reshaping and pivoting of data set, fancy indexing, label slicing column insertion and deletion etc.

1. **Matplotlib ->**

* Plotting library for python programming, provides and object-oriented application programming interface for embedding plots.
* Provides various forms for plotting a graph for numerical tables like line plot, histogram,

scatter plot contour plot etc.

1. **NumPy ->**

* NumPy adds support for large multidimensional array and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.
* Core function of NumPy is “ndarray” for n-dimension array, data structure.

1. **Sklearn** ->

* Simple and efficient tools for predictive data analysis
* Accessible to everybody, and reusable in various contexts
* Built on NumPy, SciPy, and matplotlib
* Open source, commercially usable - BSD license

***Conclusion***

The presence of a class imbalance in the data can be a major challenge while training a robust model. The above-discussed techniques can be used to handle the class imbalance prior to train a machine learning model.

One can also employ cost-sensitive learning or penalize the algorithms that increase the cost of classification of majority classes. Also, Decision Trees and Random Forest should be preferred over other machine learning algorithms as they tend to perform well on imbalanced data.

To summarize, in this article, we have seen various techniques to handle the class imbalance in a dataset. There are actually many methods to try when dealing with imbalanced data. Hope this article was useful.

# *References*

<https://elitedatascience.com/imbalanced-classes>

<https://towardsdatascience.com/methods-for-dealing-with-imbalanced-data-5b761be45a18>

<https://scikit-learn.org/stable/index.html>

<https://www.kaggle.com/code/janiobachmann/credit-fraud-dealing-with-imbalanced-datasets/data>