**CASE: Fraud Detection - A classification problem with imbalanced data.**

The steps involved in writing the code are as follows;

1. **Importing the Packages**

For this project the primary packages to be used are;

|  |  |
| --- | --- |
|  | * Numpy - for working with arrays * Pandas |
|  | * Matplotlib.pyplot -for visualization |
|  | * Termcolor –for text customization * Scikit-learn- for data split, building and evaluating the classification models used |
|  |  |
|  |  |

# Importing Data

# The data used is the Kaggle Credit Card Fraud Detection dataset

# DATA PREPROCESSING

# This data is highly imbalanced as there are way more non-fraud cases than fraud cases.

# data preprocessing is a very important first step for anyone dealing with data sets. It involves cleaning and transforming the data so as to get valuable information from the data it gathers.

NORMALIZATION

The goal of normalization is to change the values of numeric columns in the dataset to a common scale, without altering differences in the ranges of values.

‘StandardScaler’ method is used to do this.

This improves data quality and subsequent performance of machine learning algorithms.

DATA SPLIT

 Using the defined variables, the data is split into a training set and testing set which is further used for modeling and evaluating. The data easily using the ‘train\_test\_split’ algorithm in python.

MODELLING

Since this is a classification problem.The following classification algorithms have been used;

Random forests

K-Nearest Neighbor (KNN)

They are both provided by scikit –learn library.

EVALUATION

This step helps come up with the best model to represent the data and show how well the model will perform in future.

The following are the evaluation metric used;

Accuracy Score

**Accuracy score = No.of correct predictions / Total no.of predictions**.

To do it in python, the ‘accuracy\_score’ method provided by the scikit-learn package is used.

F1 Score

**F1 score = 2 ( (precision \* recall) / (precision + recall) )**

It can be calculated using the ‘f1\_score’ method provided by the scikit-learn package.

Confusion Matrix

It represents results as a form of a matrix that shows true positives, false positives,true negatives and false negatives