**FindDefault (Prediction of Credit Card fraud)**

**INTRODUCTION**

A credit card is one of the most used financial products to make online purchases and payments. Though the Credit cards can be a convenient way to manage your finances, they can also be risky. Credit card fraud is the unauthorized use of someone else's credit card or credit card information to make purchases or withdraw cash.

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) account for 0.172% of all transactions.

We have to build a classification model to predict whether a transaction is fraudulent or not.

The following are the steps we have implemented in order to create a classification model with the best accuracy and best prediction criterion.

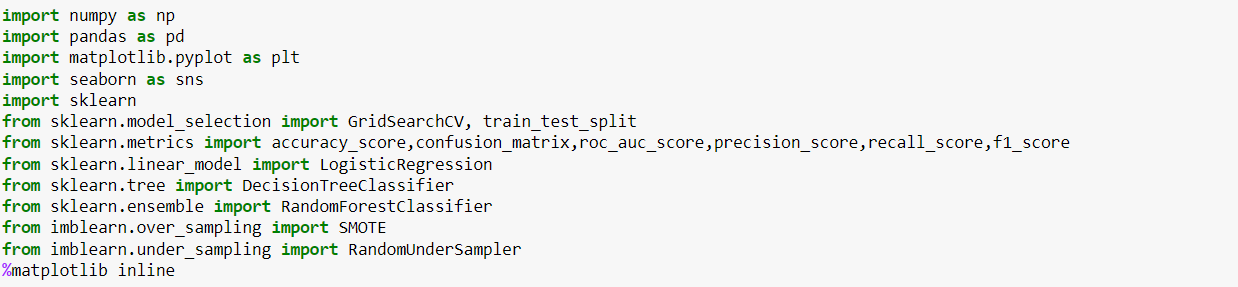
**IMPORTING THE NECESSARY LIBRARIES:**

In order to develop our model, we will have to import various library which we will be using throughout the model development phase.

The following are the libraries we will be primarily using throughout the process:

* Numpy
* Pandas
* Matplotlib
* Seaborn
* Scikit-learn (sklearn)
* Imbalanced-learn(imblearn)

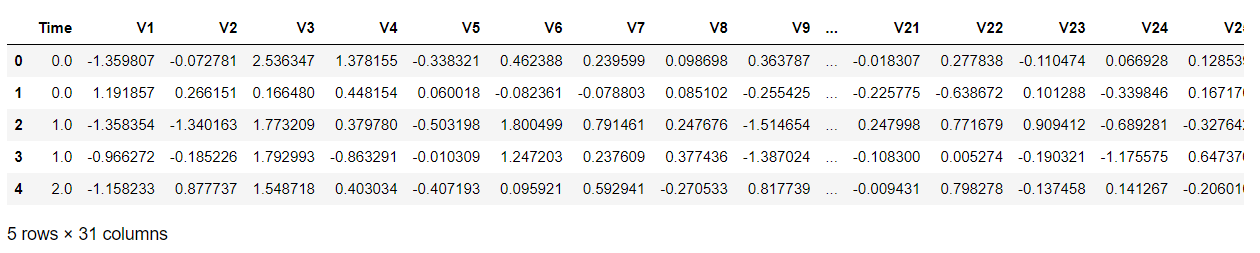
These libraries are used for various processes such as numerical calculation, data frame processing, visualization, developing Machine Learning models and so on.



Once we have imported the necessary libraries, we will proceed to the next step.

**READING THE DATA INTO A DATAFRAME:**

We will now read the .csv file using pandas and convert it into a data frame and assign it to a variable.



Looking at the data, we can see that there are 31 variables amidst which 30 are features and 1 is the label.

In this data, the variable **(‘Class’**) is our target variable. This variable has 2 categories:

* Non – fraudulent (0)
* Fraudulent (1)

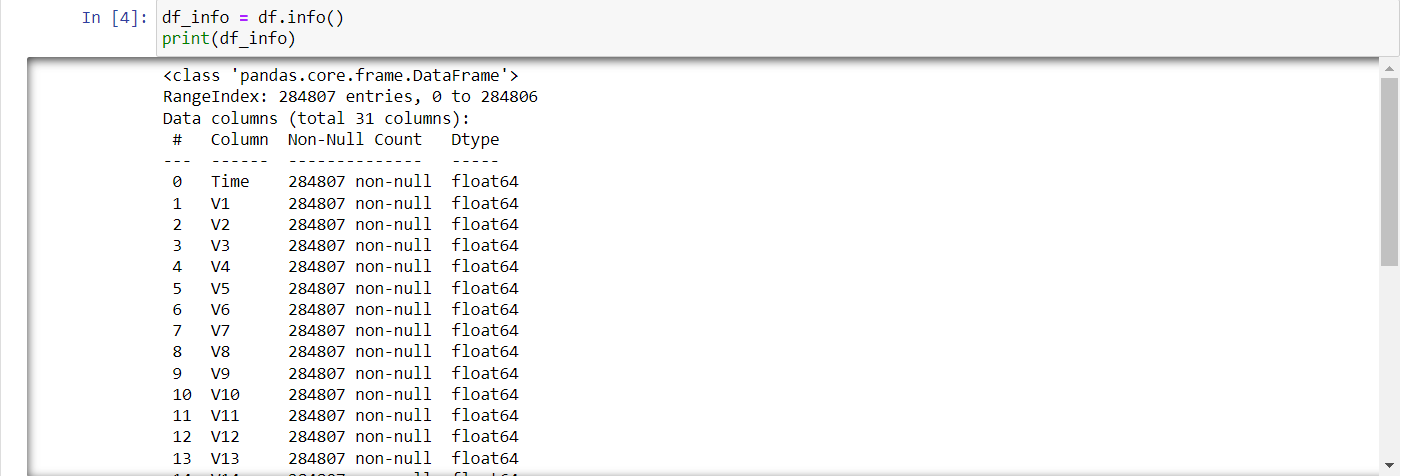
Since the data carries sensitive information about the clients, most of the features have been subjected to Principal Component Analysis (PCA) and the values have been intentionally hidden as a matter of security and privacy concerns.

The variables ‘Time’ and ‘Amount’ are the only ones that has not been transformed.

Once the data has been imported, we will proceed to conduct Exploratory Data Analysis (EDA) on the dataset to gather insights and see how to proceed further.

**EXPLORATORY DATA ANALYSIS (EDA):**

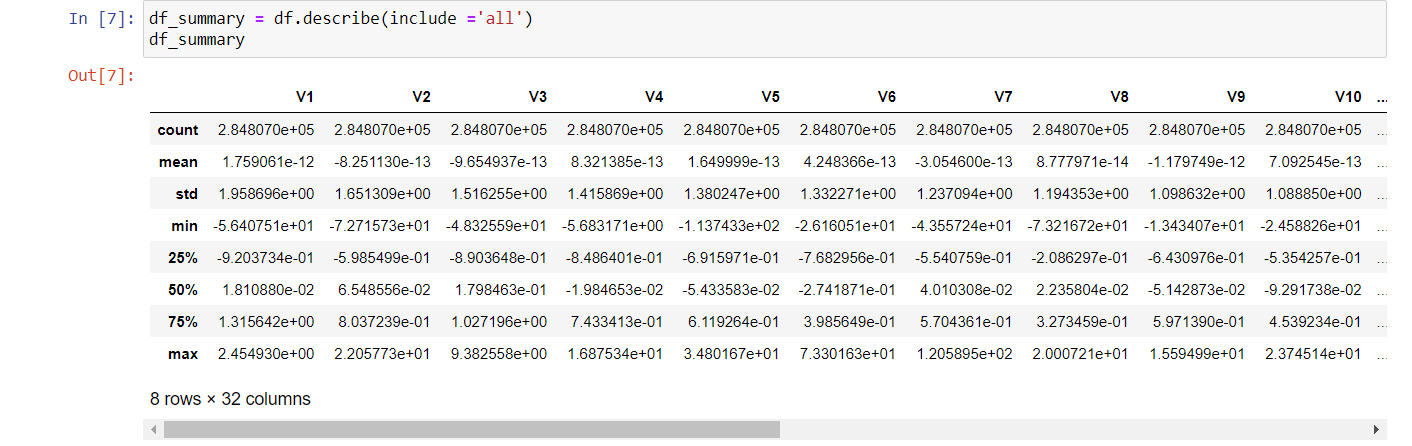
In this step, we will conduct a detailed analysis in our data to gather information on the type of data, preprocessing the data, visualization plots to gather insights.



The above image provided information on the datatypes of our variables.

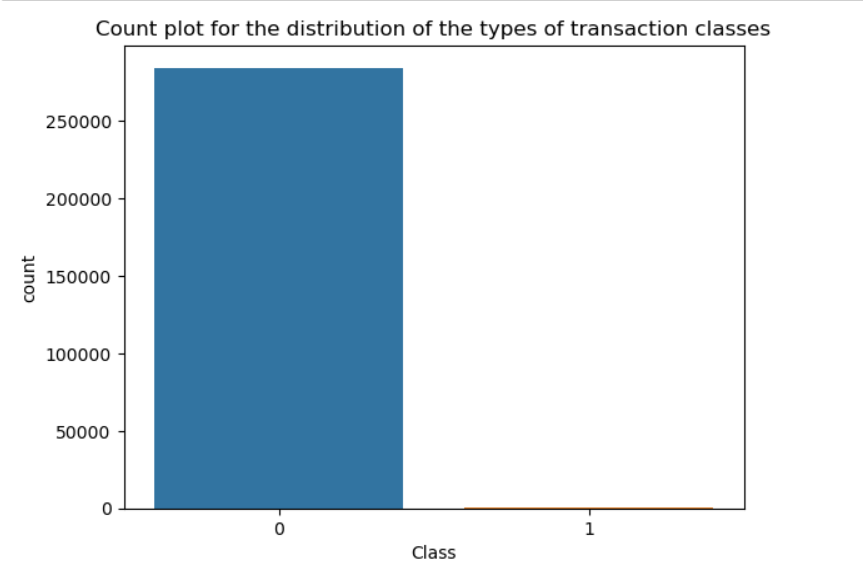
We were able to learn that the datasets comprise of only numerical data types. In addition to that, we can also see that there are no missing values in the dataset.

In the next step, we gather the statistical summary of the data in order to try and understand the distribution of the data.



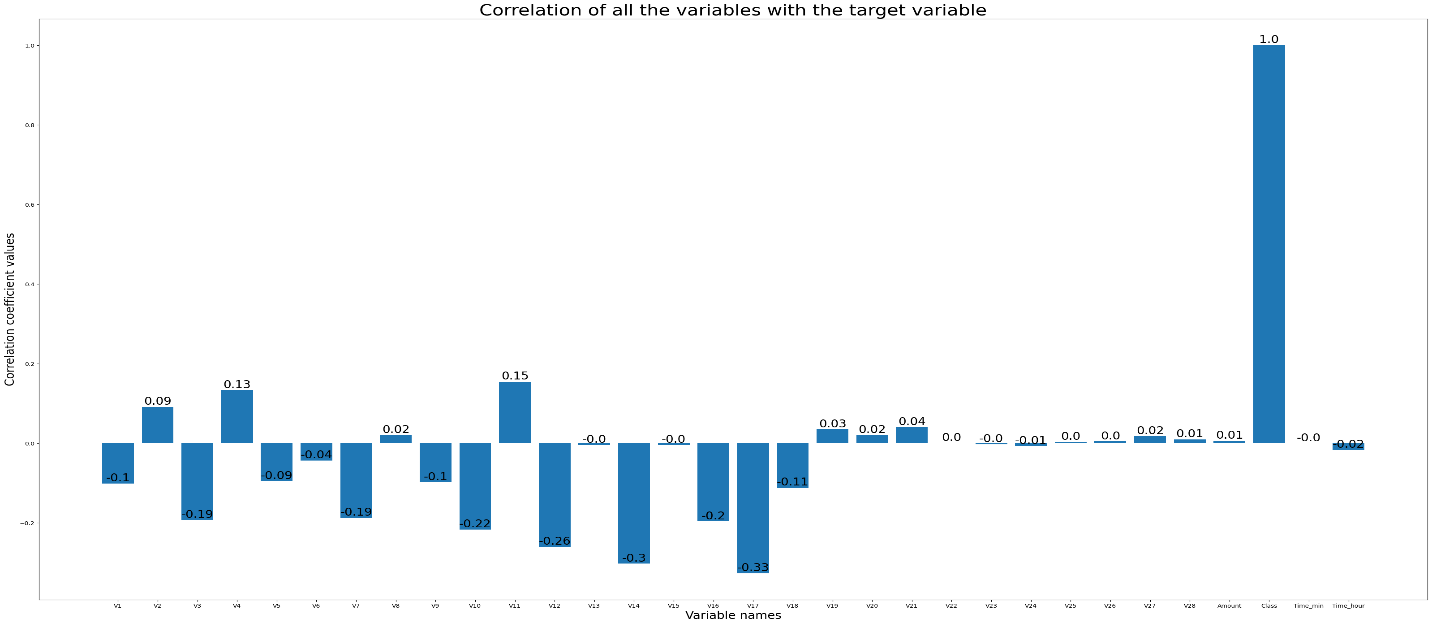
We can find essential information such as the mean, standard deviation, minimum value, maximum value and so on. Using this information, we can carry out essential processing steps to ensure our data is clean and for for modelling.

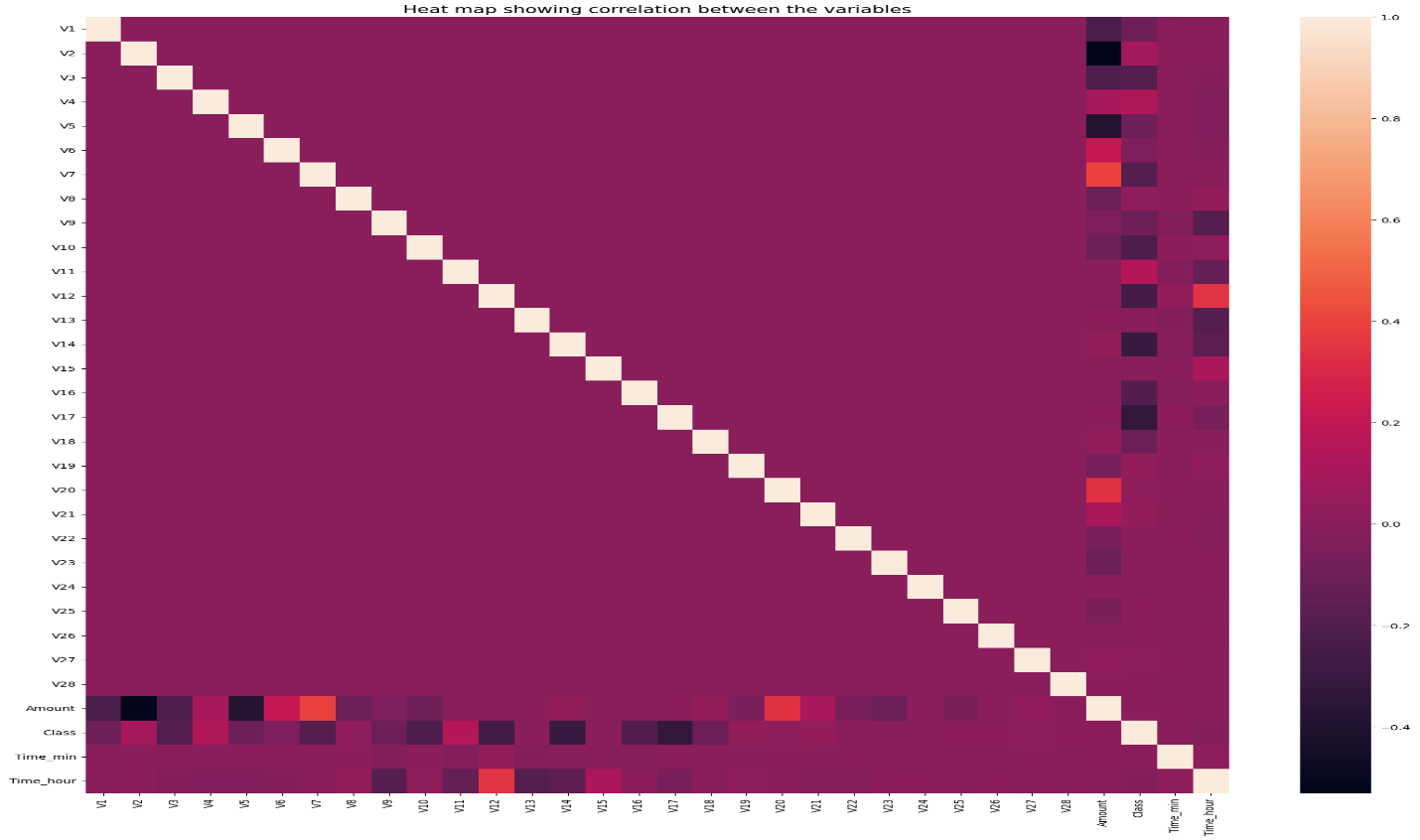
In the next step, we visualize the distribution of the target feature.



We can understand that the data is highly imbalanced from the above plot and we will need to rectify that before proceeding to the modelling phase.

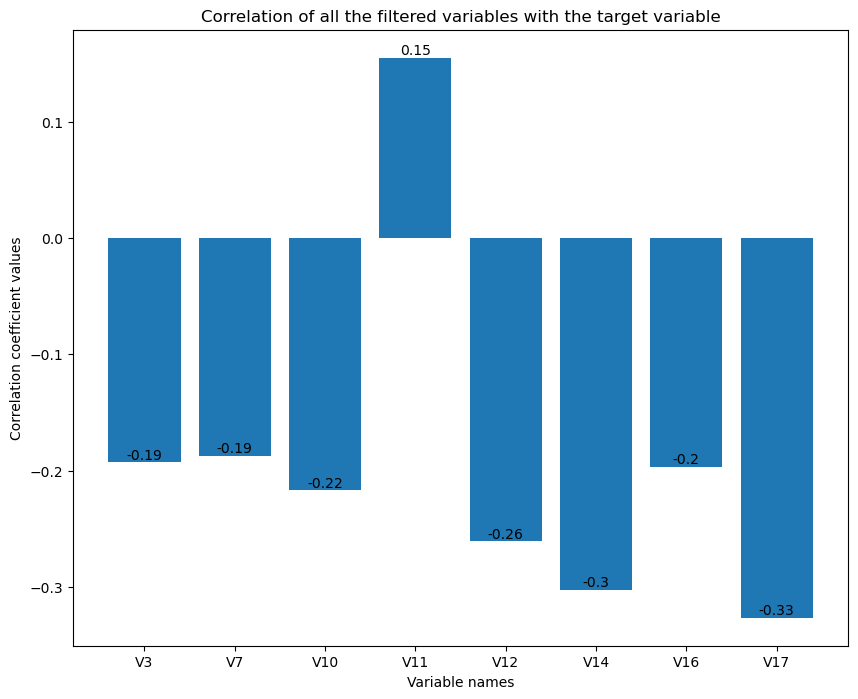
Followed by this, we will use correlation coefficient to understand the relationship between the features and the target variables. Based on the values, we will filter out the necessary variables that can be used for fitting into our model.





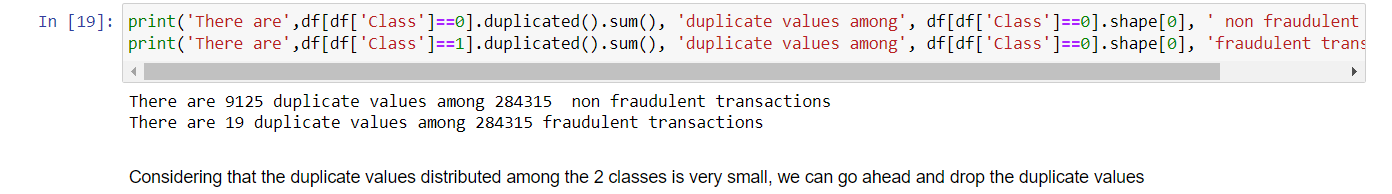
Based on the plots, we will filter out the necessary variables based on the requirement for our model.

We will then visualize the filtered-out variables to see the correlation coefficients.



**DATA CLEANING:**

In this step, since there are no null values in the dataset, we will proceed to check the presence of any duplicate rows in the data.



Since there are comparatively lesser duplicate rows distributed among the 2 classes, we will go ahead and delete the duplicate rows.

Once this is done, we will proceed with the next step, which is an important step in regards to this dataset.

**DEALING WITH UNBALANCED DATASET:**

In order to develop a Machine Learning model which gives the best results without the risk of any bias, it is essential to ensure that our dataset is balanced.

In our dataset, our data is highly imbalanced and biased toward the non-fraudulent transactions. Using the data as it is would result in an ML model which makes predictions which are highly biased.

In order to prevent that, we need to balance our dataset before proceeding further.

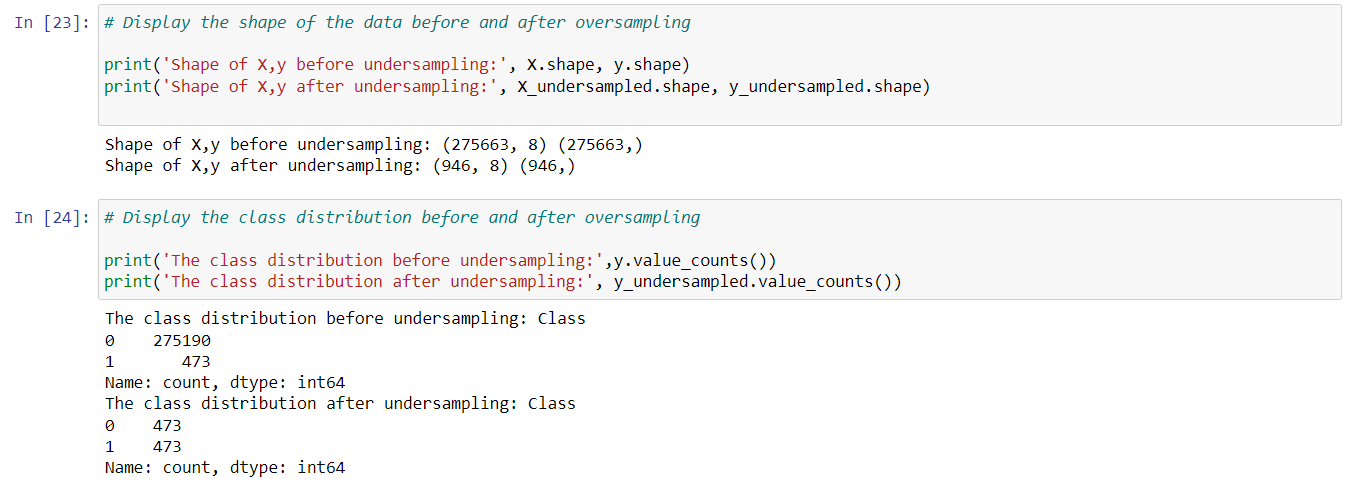
In order to balance our dataset, we can use two methods:

* Oversampling
* Under sampling

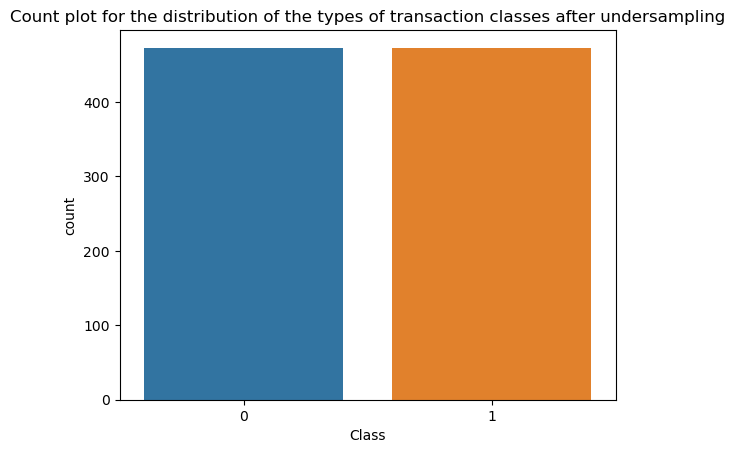
**Oversampling** is the method where the minority class in the target variable is artificially inflated by generating synthetic examples. One of the common methods employed in oversampling is SMOTE(Synthetic Minority Over - sampling Technique)

**Under sampling** is the method where all the data in the minority class is kept intact and the sample from the majority class is reduced in order to balance it with the minority class. One of the common and simplest method employed in under sampling is Random under sampling.

We will be using the under sampling technique here in our dataset to derive effective results



The above picture shows how the data has been balanced after it has been under sampled and there is a comparison between the data before and after under sampling.



This image shows a visual representation of how the data has been balanced so that there would be no bias during the prediction.

The next stage is the model selection stage wherein we will build and evaluate various models.

**MODEL SELECTION:**

Since this is a credit fraud detection problem, it involves creating a classification model.

We will try out various models and evaluate which model has the best performance.

The models that we will be trying out are the following:

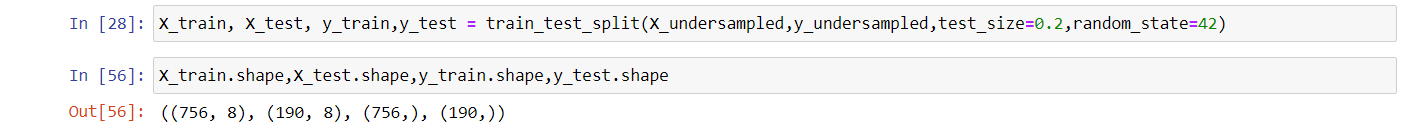
1. Logistic Regression
2. Decision Trees
3. Random Forest Classifier

**SPLITTING THE DATA INTO TRAIN AND TEST DATA:**

The first step involved in model selection is splitting our data into a train dataset and a test dataset.

**Train dataset:** This is the data we will be using to train the model. The machine will learn from this data so that it can predict on new unseen data.

Test dataset: This is a part of the dataset which we will be using to test the model performance. This will be the unseen data that we will feed into the model in order to test the prediction accuracy.



The above image shows how the data has been split into the train and test data.

Going further, we will try out our models one by one.

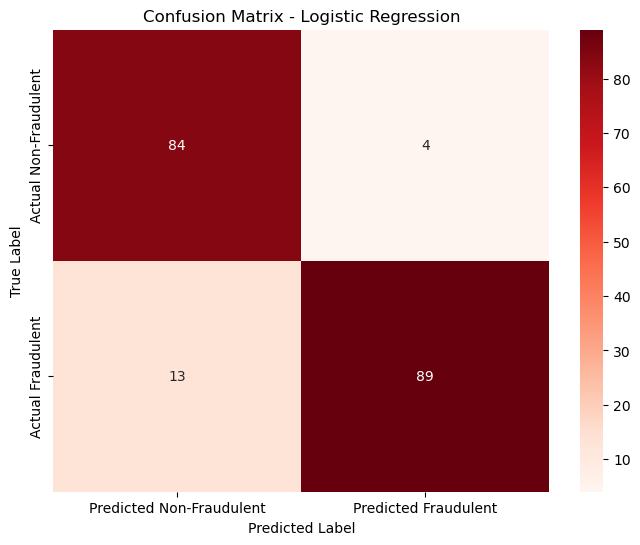
**LOGISTIC REGRESSION MODEL:**

First, we will try the logistic regression model and evaluate the metrics. We will initiate the model into a variable and fit our train data into the model.

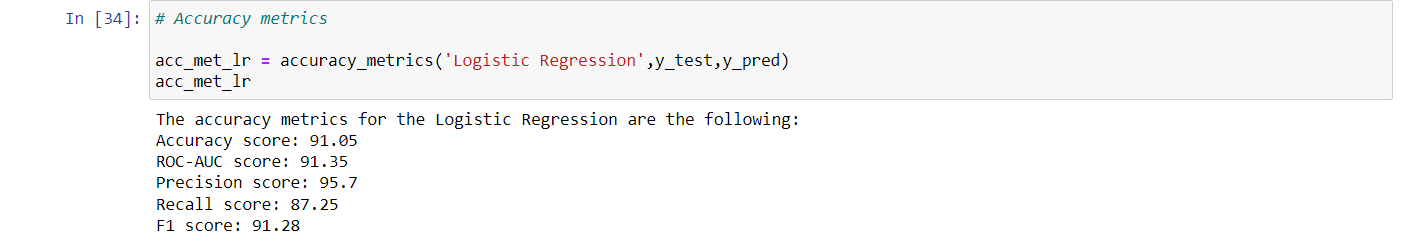
Once the model has been trained, we will predict the target variable on the test data and evaluate the metrics.

We will initially use a confusion matrix to assign the actual values to the prediction values to see how well the model is predicting.

Once the matric has been visualized, we will evaluate the metrics for the model in order to assess the accuracy.



The above figure shows the confusion matrix for the logistic regression with the count of actual and predicted transactions.

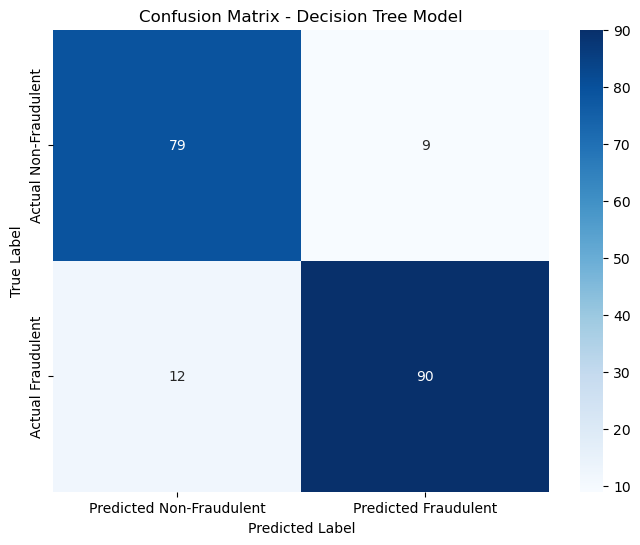


This image shows the accuracy metrics for this model with an accuracy score of **91.05%.**

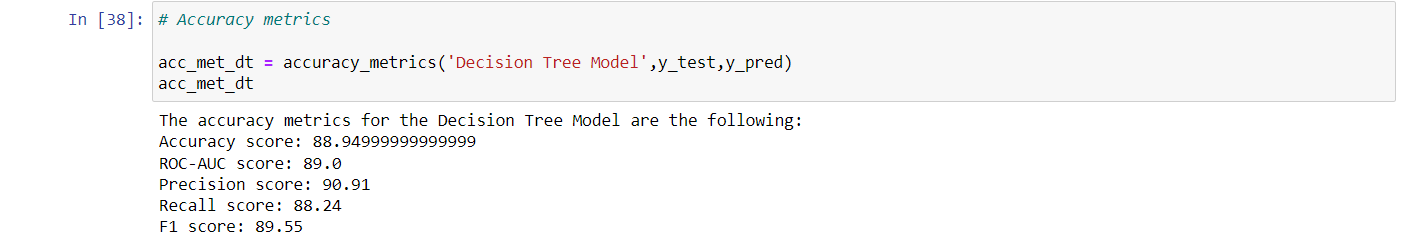
Next, we will try out the decision tree model to see the performance and evaluate it.

**DECISION TREES:**

We will be following the same procedure for this model as well. Once the model is fitted and used for prediction, we will visualize the confusion matrix and then evaluate the performance metrics.



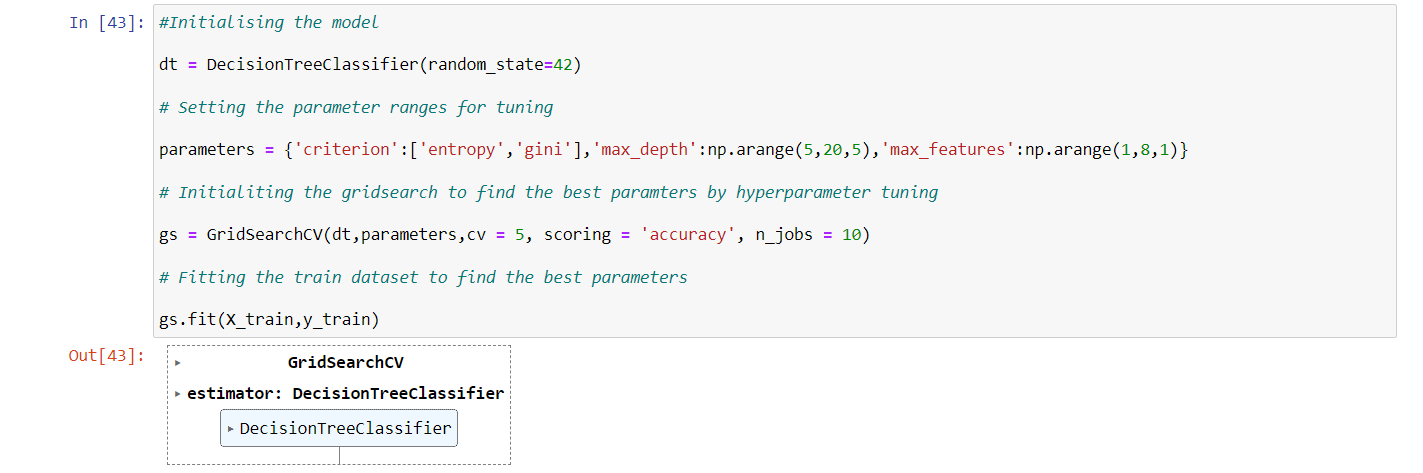
The above shows the confusion matrix for the decision tree model along with the distribution of the true and predicted values.



The above images show the evaluation metrics for the decision tree model. It has an accuracy score of **88.94%**

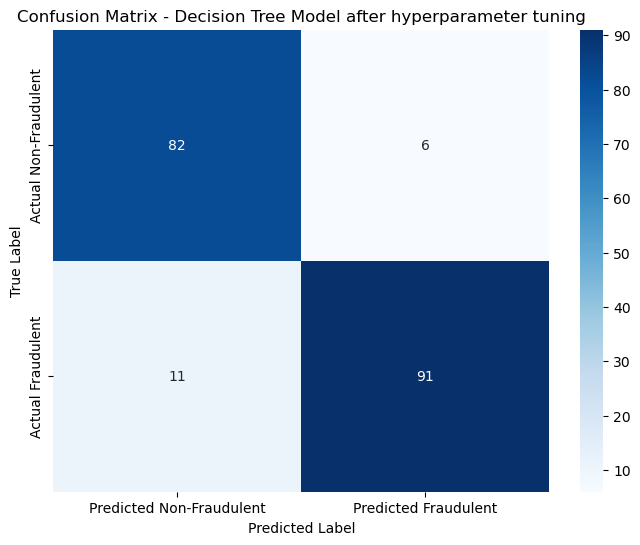
In order to see if we can improve the performance of this model, we will further perform hyperparameter tuning in this model to find the best parameters in order to get the best performance.

We will use Grid Search CV in order to iterate through the ranges we provide for the parameters. This will help in getting the best parameters and the best model.

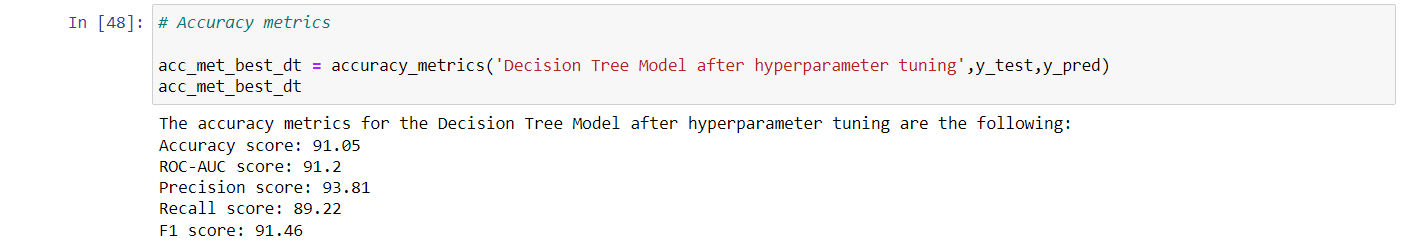


The above image shows the process as to how hyperparameter tuning is done.

Once we get the best mode, we will follow the same process of fitting the model and predicating the target variable. A confusion matrix is then generated and finally the evaluation metrics are calculated.



The above image shows a slight improvement in the prediction values after hyperparameter tuning.

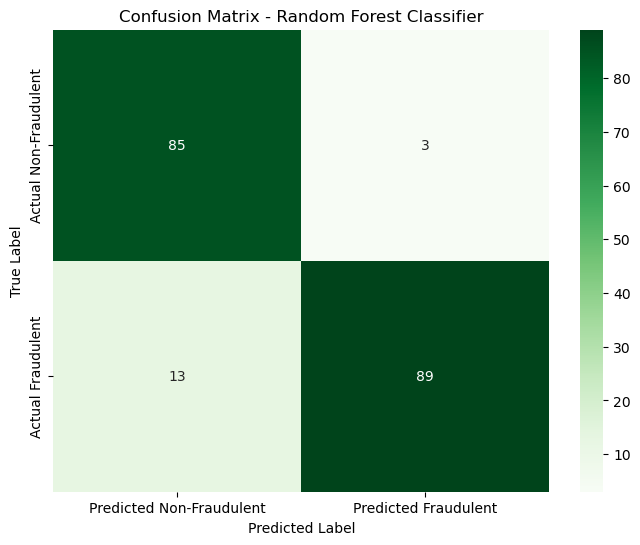


There also seems to be an improvement in the evaluation metrics. This model has an accuracy score of **91.05%**, which is better than the model performance without hyperparameter tuning.

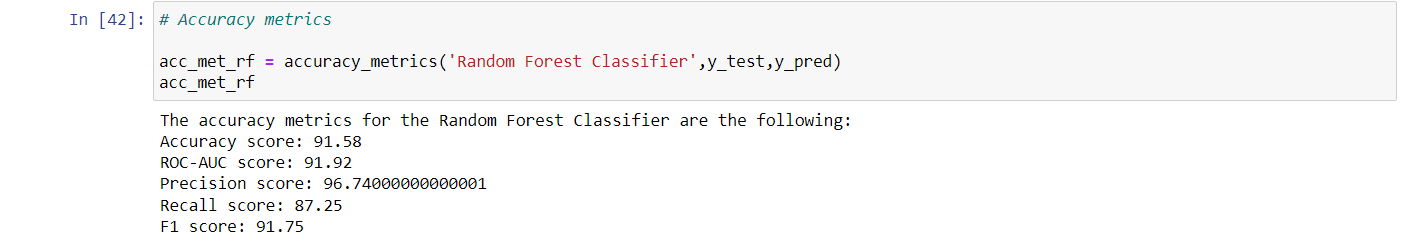
Lastly, we will try out the Random Forest Classifier and see how it performs in comparison with the other models we tested.

**RANDOM FOREST CLASSIFIER**

We will be following the exact same steps for this model as well. Post the fitting and prediction, the confusion matrix is generated.



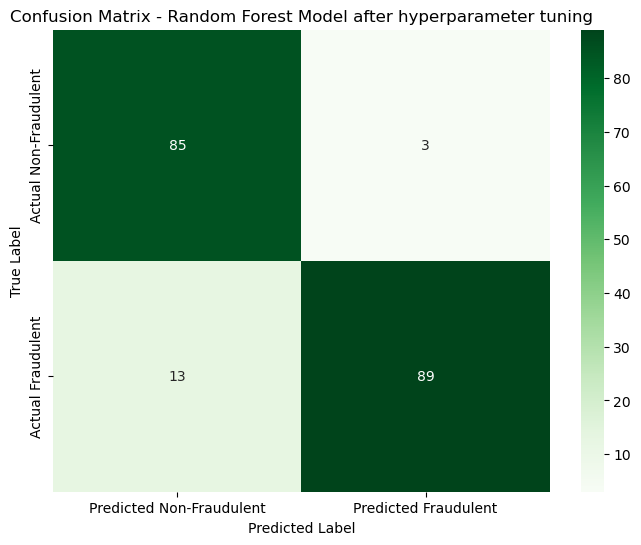
The above image shows the distribution for the Random Forest Classifier.



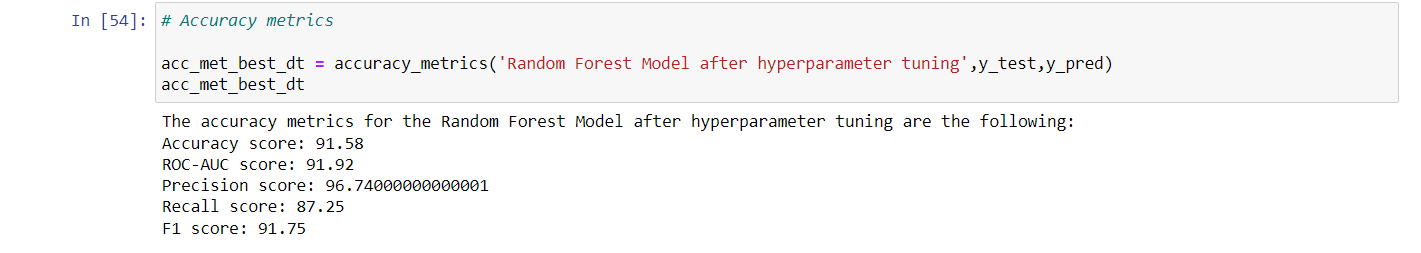
Based, on the accuracy metrics, we can see that the Random Forest Classifier has an accuracy score of **91.58%**.

In order to try and improve the performance, we will do hyperparameter tuning on this model as well.

After obtaining the best model and following the standard procedure, the results we obtained are shown below.



The confusion matrix does not show any different even after hyperparameter tuning.



Even after tuning, the accuracy score remains the same at 91.58%

**FINDINGS FROM THE MODEL:**

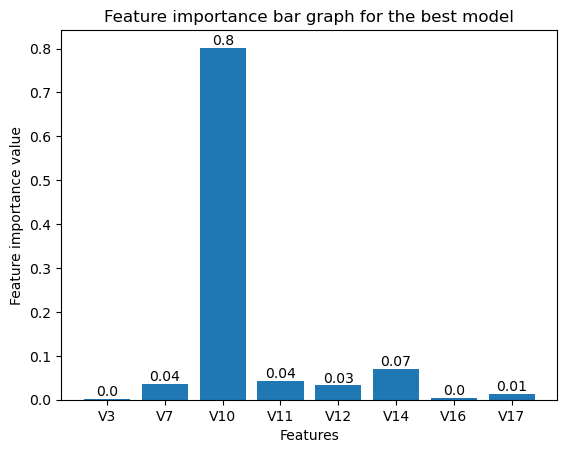
After hyperparameter tuning, we could see a minor improvement in the accuracy metrics of the Decision Tree Model (91.05%) and close to no improvement in the Random Forest Model (91.58%). Also, we can see that the accuracy metric for the logistic regression (91.05%) was the same as that of the Decision Tree Model.

Since there are no major differences between the accuracy metrics of Random Forest Model in comparison with the other 2 models, we will drop this model since this is computationally heavy and time consuming to build.

Taking the other 2 models into consideration, since this is a fraud detection model, it is essential that that machine predicts the fraudulent charges better over the non-fraudulent charges. In this case, based on the confusion matrix values, we can see that the fraudulent predictions for the Decision Tree have a slight edge over that of the Logistic Regression model. Henceforth, we will be going forward with the **Decision Tree Model** as our **golden model**.

**FEATURE IMPORTANCE:**

Lastly, we will plot the feature importance from our golden model in order to see which feature has the highest influence and which has the lowest. All the values have been rounded off to the 2 values. We finally plot the values in a bar graph in order to have a visual understanding of the same.

****

The above image shows the feature importance distribution of the variables for our best model.

**CONCLUSION:**

**MODEL DEVELOPMENT**: Throughout this entire project, we followed a step wise well-established methodology followed by other optimization and improvement methods in order to build the best model (**Decision tree model with an accuracy of 91.05%**)

**DATA HANDLING**: Considering that the data provided was highly imbalanced, we had identified the importance of balancing the data. Failing to do so would have resulted in prediction that would be highly biased. We used an under-sampling method in order to handle the imbalance issue.

**OPTIMIZATION METHODS**: After building our models and getting a decent accuracy, we implementing hyperparameter tuning in order to try and tweak up the accuracy. We used Grid Search Cross Validation in order to identify the best parameters for the model.

**EVALUATION METRICS**: Even through the Random Forest Model had the highest accuracy metrics, we were able to observe that there were no major differences in comparison with the other 2 models. Since the Random Forest Classifier is heavy on computation and expensive to perform, we went ahead with the decision tree model which had a better prediction of the fraudulent transactions in comparison with the logistic model.

**REAL-WORLD APPLICATION**: Since there are billions of transactions happening on a daily basis, it would be close to impossible to monitor each and every transaction manually. In order to overcome this issue, this specific model would greatly help financial sectors in identifying fraudulent transactions automatically so that necessary actions can be taken immediately.

**FUTURE SCOPE**: For future improvement of the process, we can try implementing various other models to improve its performance and also monitors the performance degradation overtime so that certain improvements can be made to the model to ensure high performance accuracy.