# CREDIT CARD FRAUD DETECTION



# PROJECT DESCRIPTION:

* Data source
* Data preparation
* Data preprocessing
* Data modeling
* Evaluation and Deployment

DESCRIPTION:

Credit card fraud is an inclusive term for fraud committed using a payment card, such as a credit card or debit card. The purpose may be to obtain goods or services or to make payment to another account, which is controlled by a criminal. The Payment Card Industry Data Security Standard (PCI DSS) is the data security standard created to help financial institutions process card payments securely and reduce card fraud.

DESIGN THINIKING:

DATA SOURCE:

* The dataset was retrieved from an open-source website, Kaggle.com.
* It contains data of transactions that were made in 2013 by credit card users in Europe, in two days only.
* The dataset consists of 31 attributes, 284,808 rows.

DATA LINK:

<https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud>

DATA PREPROCESSING:

* As there are no NAs nor duplicated variables, the preparation of the dataset was simple the first alteration that was made to be able to open the dataset on Weka program is changing the type of the class attribute from Numeric to Class and identify the class as {1,0} using the program Sublime Text. Another alteration was made on the type as well on the R program to be able to create the model and the visualization.
* The 4.3 Data ModelingAfter making sure that the data is ready to get modeled the four models were created using both Weka and R. the model SVM was created using Weka only, as for KNN, Logistic Regression and NaïveBayes they were created using R and Weka.

IMPORTING THE REQUIRED LIBRARIES:

**Let’s start the development part of credit card fraud detection with machine learning by importing the necessary Python libraries.**

**import numpy as np**

**import pandas as pd**

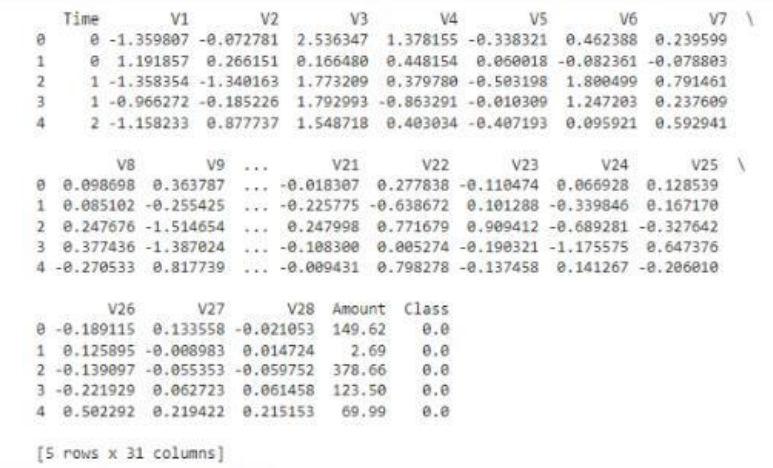
**import matplotlib.pyplot as plt**

**import seaborn as sns**

**from matplotlib import gridspec**

**IMPORTING DATASET:**

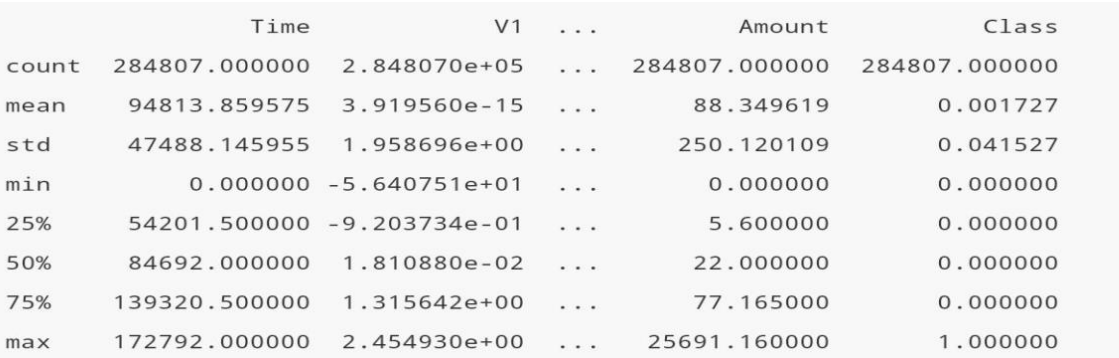
**df=pd.read\_csv('../input/creditcardfraud/creditcard.csv')**

****

**DESCRIBING THE DATA:**

**Print(data.shape)**

**Print(data.describe())**

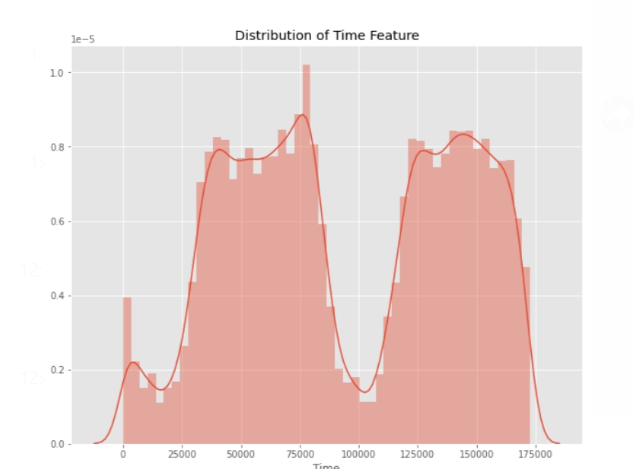
****

**VISUALIZATION:**

plt.figure(figsize=(10,8))

plt.title('Distribution of Time Feature')

sns.distplot(df.Time)

****

**FEATURE ENGINEERING:**

Feature engineering is the pre-processing step of machine learning, which extracts features from raw data. It helps to represent an underlying problem to predictive models in a better way, which as a result, improve the accuracy of the model for unseen data. The predictive model contains predictor variables and an outcome variable, and while the feature engineering process selects the most useful predictor variables for the model.

FEATURE ENGINEERING TECHNIQUES:

⦿Imputation

⦿Handling Outliers

⦿Log transform

⦿Binning

IMPUTATION:

* Feature engineering deals with inappropriate data, missing values, human interruption, general errors, insufficient data sources, etc.
* Missing values within the dataset highly affect the performance of the algorithm, and to deal with them “Imputation” technique is used. Imputation is responsible for handling irregularities within the dataset.

HANDLING OUTLIERS:

Standard deviation can be used to identify the outliers. For example, each value within a space has a definite to an average distance, but if a value is greater distant than a certain value, it can be considered as an outlier. Z-score can also be used to detect outliers.

LOG TRANSFORM:

Logarithm transformation or log transform is one of the commonly used mathematical techniques in machine learning. Log transform helps in handling the skewed data, and it makes the distribution more approximate to normal after transformation. It also reduces the effects of outliers on the data, as because of the normalization of magnitude differences, a model becomes much robust.

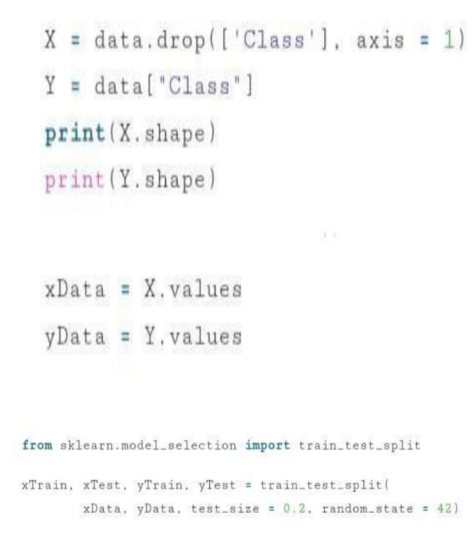
BINNING:

In machine learning, overfitting is one of the main issues that degrade the performance of the model and which occurs due to a greater number of parameters and noisy data. However, one of the popular techniques of feature engineering, “binning”, can be used to normalize the noisy data. This process involves segmenting different features into bins.

MODEL TRANING:

‣ Testing data: The testing set is a smaller portion of the data, usually around 20-30% of the dataset. It is kept separate and is not used during the model training phase. Instead, it is used to evaluate the model’s performance by making predictions or performing analyses and comparing them to the actual, known outcomes

‣ Training data:This subset contains a majority of the data, typically around 70-80% of the dataset. It is used to train machine learning models or perform data analysis tasks. The model learns patterns, relationships, and trends within the data from this set.



Evaluation:

‣ Correlation matrix

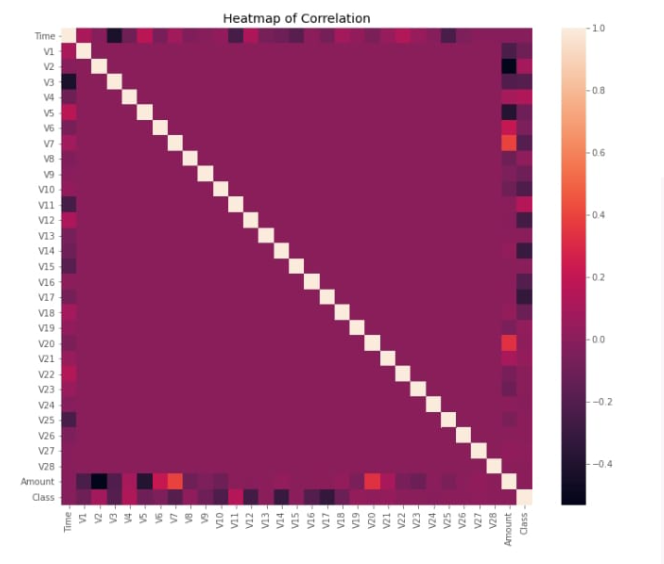
The correlation matrix graphically gives us an idea of how features correlate with each other and can help us predict what are the features that are most relevant for the prediction.

corr = df.corr()

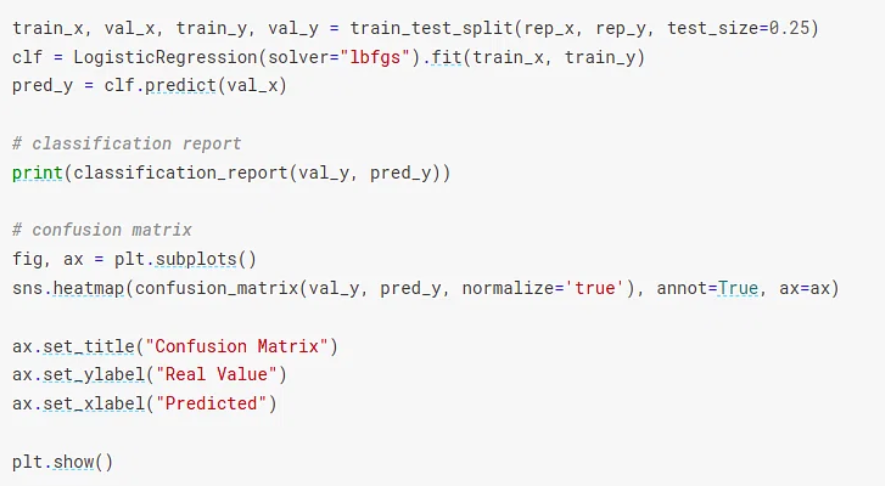
plt.figure(figsize=(12,10))

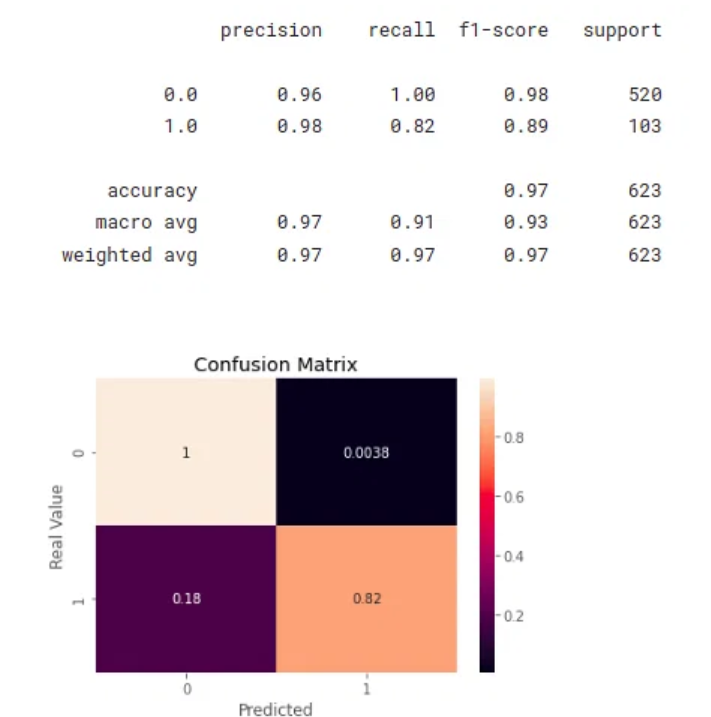
heat = sns.heatmap(data=corr)

plt.title('Heatmap of Correlation')



VISUALIZING THE CONFUSION MATRIX:





# Conclusion & Future Work:

Fraud detection is a complex issue that requires a substantial amount of planning before throwing machine learning algorithms at it. Nonetheless, it is also an application of data science and machine learning for the good, which makes sure that the customer’s money is safe and not easily tampered with.

Future work will include a comprehensive tuning of the Random Forest algorithm I talked about earlier. Having a data set with non-anonymized features would make this particularly interesting as outputting the feature importance would enable one to see what specific factors are most important for detecting fraudulent transaction.