**Introduction:**

Working to spot the potential fraud, so that the consumers cannot bill for goods that they haven’t purchased. Therefore the aim is to create a classifier that shows whether the transaction is fraud.

This project named as “***Credit card fraud detection***” is basically with ***machine learning***. Code is detecting the credit card fraud transactions using different libraries of python. It is working with numpy, pandas, matplotlib and many other libraries.

We are simply going to build a classifier in machine learning which would show the fraud transactions. Data will be validated by using Decision-tree-classifier model. Graphs are used for visualizations of datasets.

The code is divided simply into 3 steps:

* Performing exploratory data analysis on the dataset
* Applying different machine learning algorithms to the dataset
* Training and evaluating our model on the dataset to validate predictions

**Credit Card Fraud Dataset:**

The dataset for this project is a csv file which consists of 31 Parameters, Due to some confidentiality issues 28 features are the result of PCA (Principles component analysis transformation) in which **an orthogonal linear transformation** takes place that transforms the data to a new coordinate system so that the greatest variance comes to the first coordinate, the second greatest to the second coordinate and so on. “Time” & “Amount” are not modified with PCA. There are over 284,807 transactions in this files, out of which 492 are fraud, so there are some misbalancing issues as well.

**Methodology:**

The project is based on machine learning algorithms, with some of the libraries in python.

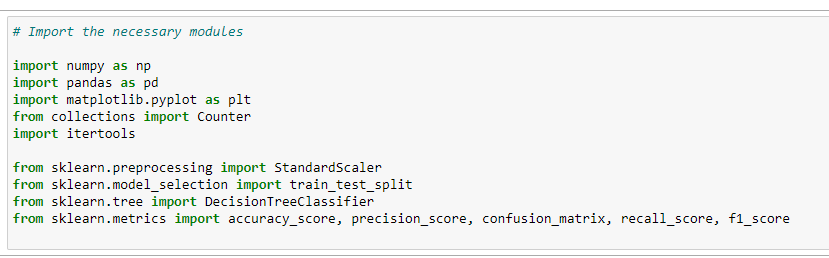
Tools and Libraries used in this project are:

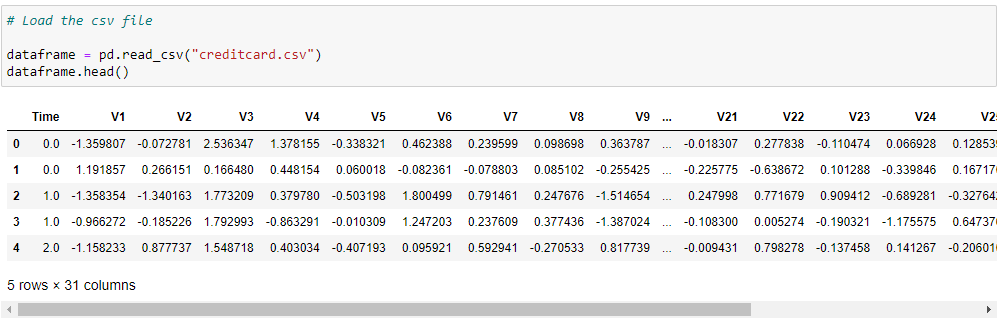
* Python
* Numpy
* Matplotlib
* Scikit-learn
* Collections ,Itertools
* Install and Import these libraries with various functionalities (preprocessing, standard scaling, normalizing, splitting, metrics) in them.
* After importing, use the data frames to load the csv dataset and work within the frames.
* Categorize the fraud and non-fraud transactions.
* Visualize the labels (columns) in the dataset with the help of graph plotting and visualization.
* Then perform the standard scaling over the data to reshape or normalize the data.
* Split the data using train\_test\_split function which trains and tests the dataset for old and new predicted values.
* Use the decision-tree-classifier model and plot the confusion matrix.
* Evaluate the decision-tree-classifier on the basis of different matrices(accuracy, precision, recall and F1 scores)

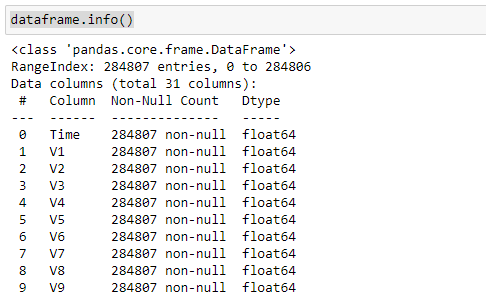
**Experiments & Results:**

Following is the complete step by step code with results to conclude the fraud detection system:

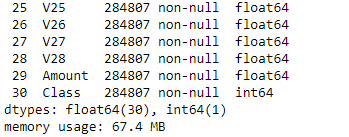
1. Install and import all the libraries and functionalities to be used.



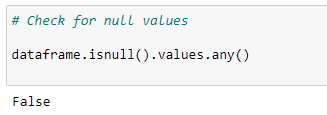
1. Load the csv file containing the dataset by using data frames, and display n rows.
2. Show the type of the data values in the dataset by using .info( ).



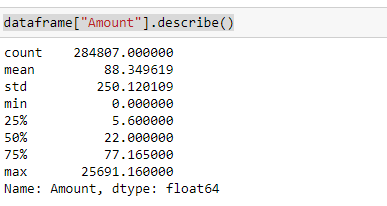
-



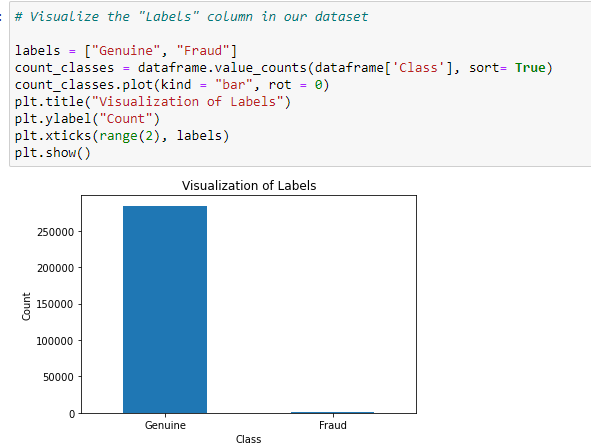
1. Check if any entry is null or not null



1. From the column, “Amount” describe all the possible operational outputs.



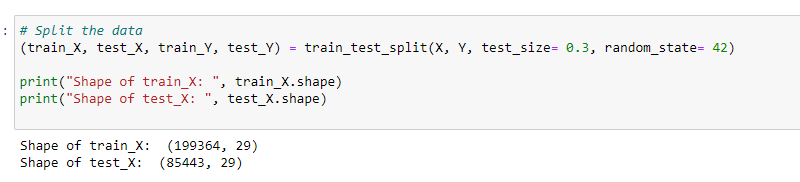
1. Plot the graph using the data frame with the class column in descending order.

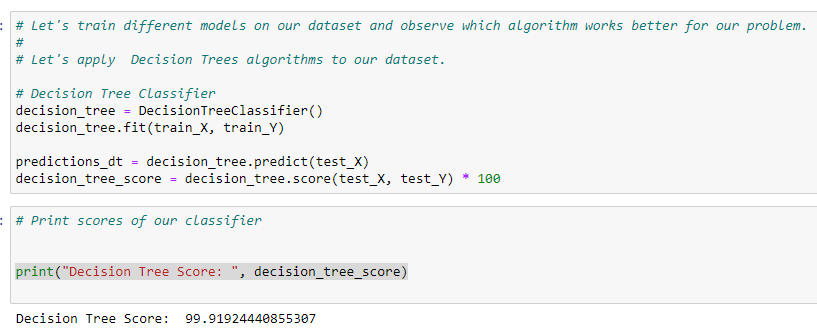


1. Use the standard scaler to reshape the values to a new column and then drop the time

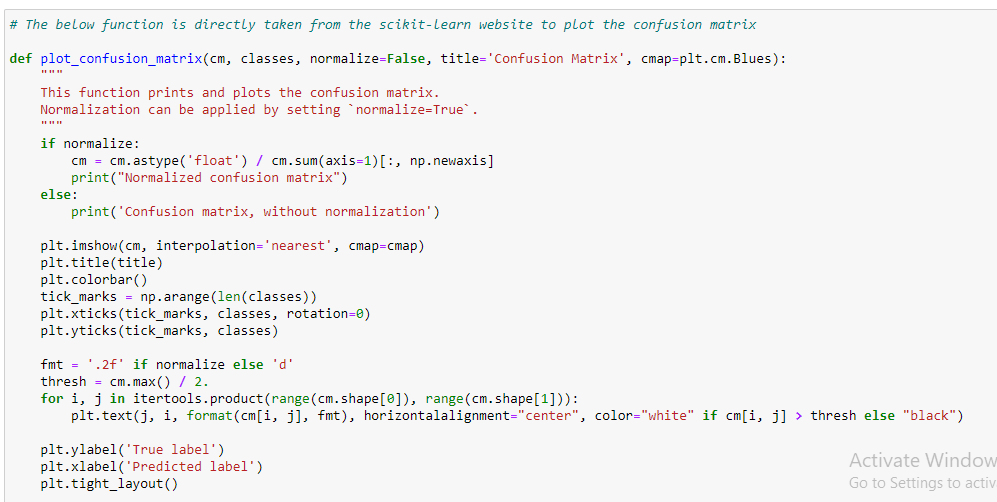
and amount columns.



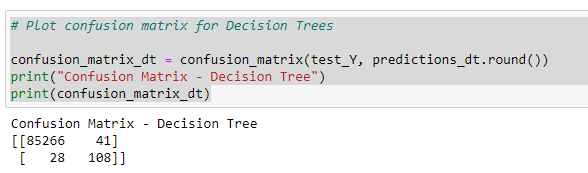
1. Split, train and test the data by using train\_test\_split( ), returns you shape of X & Y.
2. Use the decision tree classifier to fit in the train\_test\_split and then calculate the score after predicting the values into a new frame.



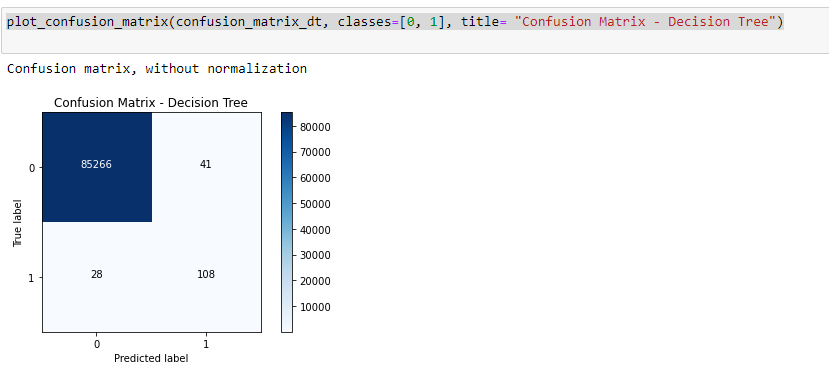
1. Now plot the confusion matrix for decision tree, build a graph giving the title, x and y y ticks and colorbars to it. After looping the structure it will give you a plotted graph.

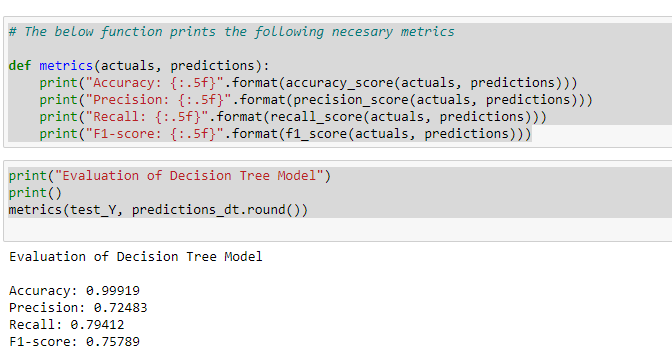


1. Consider a new frame and make a decision confusion matrix from it.



1. Call the functions and use the plotting parameters to visualize the graph.



1. At the end, print the matrices which show the evaluation of decision tree on the basis of accuracy, precision, recall and F1 scores.