Credit Card Fraud Detection Using Random Forest & Cart Algorithm

In this project we are using python Random Forest inbuilt Cart algorithm to detect fraud transaction from credit card dataset, we downloaded this dataset from ‘kaggles’ web site from below URL

Dataset URL: <https://www.kaggle.com/mlg-ulb/creditcardfrau>

To provide privacy to users transaction data kaggles peoples have converted transaction data to numerical format using PCA Algorithm. Below are some example from dataset

"Time","V1","V2","V3","V4","V5","V6","V7","V8","V9","V10","V11","V12","V13","V14","V15","V16","V17","V18","V19","V20","V21","V22","V23","V24","V25","V26","V27","V28","Amount","Class"

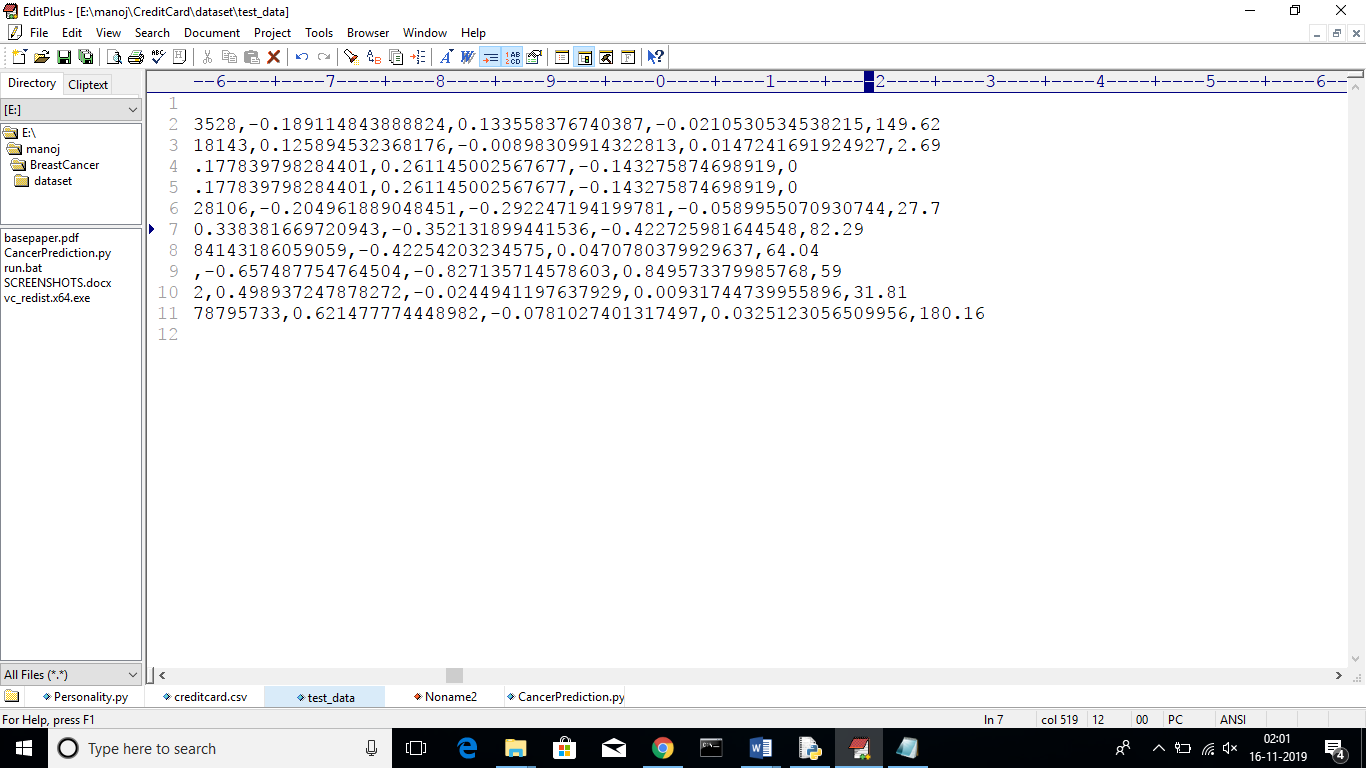
0,-1.3598071336738,-0.0727811733098497,2.53634673796914,1.37815522427443,-0.338320769942518,0.462387777762292,0.239598554061257,0.0986979012610507,0.363786969611213,0.0907941719789316,-0.551599533260813,-0.617800855762348,-0.991389847235408,-0.311169353699879,1.46817697209427,-0.470400525259478,0.207971241929242,0.0257905801985591,0.403992960255733,0.251412098239705,-0.018306777944153,0.277837575558899,-0.110473910188767,0.0669280749146731,0.128539358273528,-0.189114843888824,0.133558376740387,-0.0210530534538215,149.62,"0"

0,1.19185711131486,0.26615071205963,0.16648011335321,0.448154078460911,0.0600176492822243,-0.0823608088155687,-0.0788029833323113,0.0851016549148104,-0.255425128109186,-0.166974414004614,1.61272666105479,1.06523531137287,0.48909501589608,-0.143772296441519,0.635558093258208,0.463917041022171,-0.114804663102346,-0.183361270123994,-0.145783041325259,-0.0690831352230203,-0.225775248033138,-0.638671952771851,0.101288021253234,-0.339846475529127,0.167170404418143,0.125894532368176,-0.00898309914322813,0.0147241691924927,2.69,"0"

406,-2.3122265423263,1.95199201064158,-1.60985073229769,3.9979055875468,-0.522187864667764,-1.42654531920595,-2.53738730624579,1.39165724829804,-2.77008927719433,-2.77227214465915,3.20203320709635,-2.89990738849473,-0.595221881324605,-4.28925378244217,0.389724120274487,-1.14074717980657,-2.83005567450437,-0.0168224681808257,0.416955705037907,0.126910559061474,0.517232370861764,-0.0350493686052974,-0.465211076182388,0.320198198514526,0.0445191674731724,0.177839798284401,0.261145002567677,-0.143275874698919,0,"1"

Above bold names are the column names of this dataset and others decimal values are the content of dataset and in above 3 rows last column contains class label where 0 means transaction values are normal and 1 means contains fraud values.

Using above ‘CreditCardFraud.csv’ file we will train Random Forest algorithm and then we will upload test data file and this test data will be applied on Random Forest train model to predict whether test data contains normal or fraud transaction signatures. When we upload test data then it will contains only transaction data no class label will be there application will predict and give the result. See below test data file



In above screen in test data file there are no 0 or 1 values, application will predict from this test data using random forest and give the result.

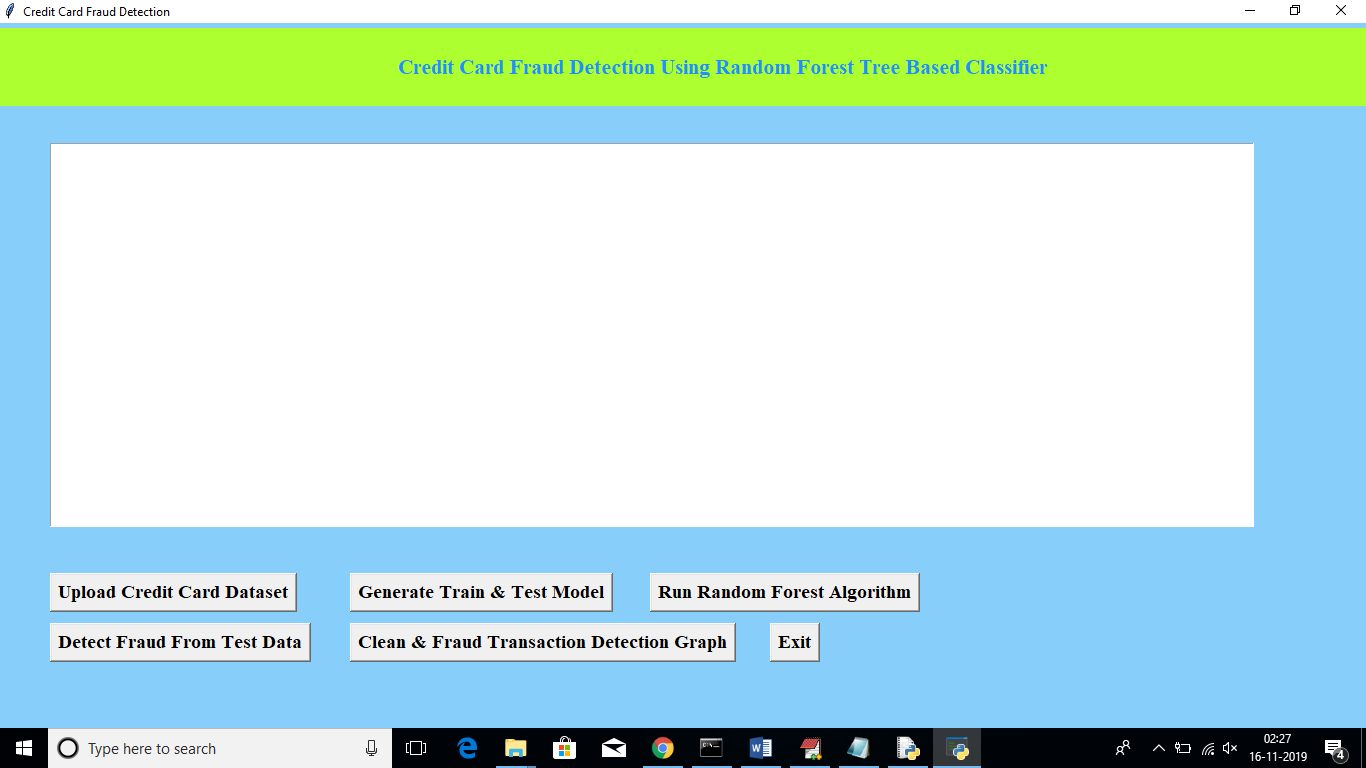
Random Forest Algorithm

Random forests is a supervised learning algorithm. It can be used both for classification and regression. It is also the most flexible and easy to use algorithm. A forest is comprised of trees. It is said that the more trees it has, the more robust a forest is. Random forests creates decision trees on randomly selected data samples, gets prediction from each tree and selects the best solution by means of voting. It also provides a pretty good indicator of the feature importance. Python SKLEARN inbuilt contains support for CART with all decision trees and random forest classifier.

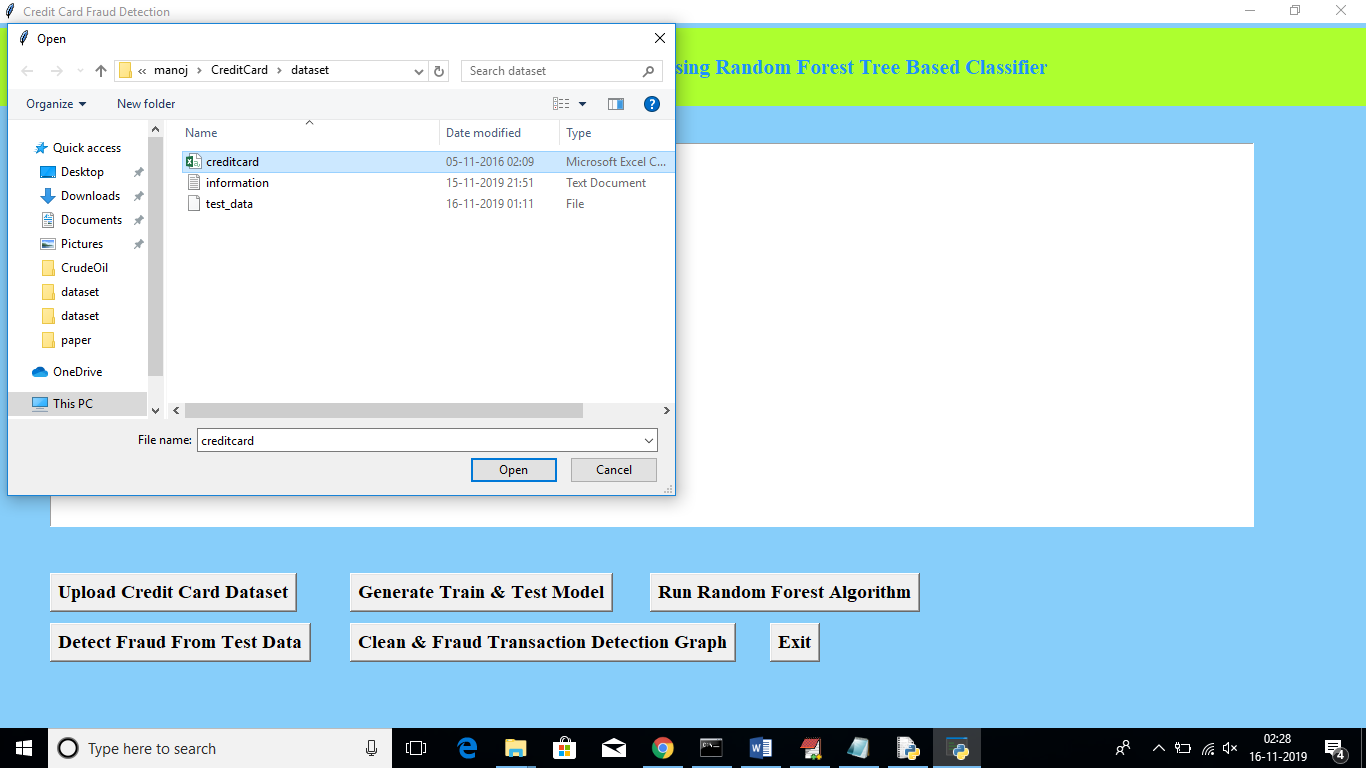
Random forests has a variety of applications, such as recommendation engines, image classification and feature selection. It can be used to classify loyal loan applicants, identify fraudulent activity and predict diseases. It lies at the base of the Boruta algorithm, which selects important features in a dataset.

Screen shots

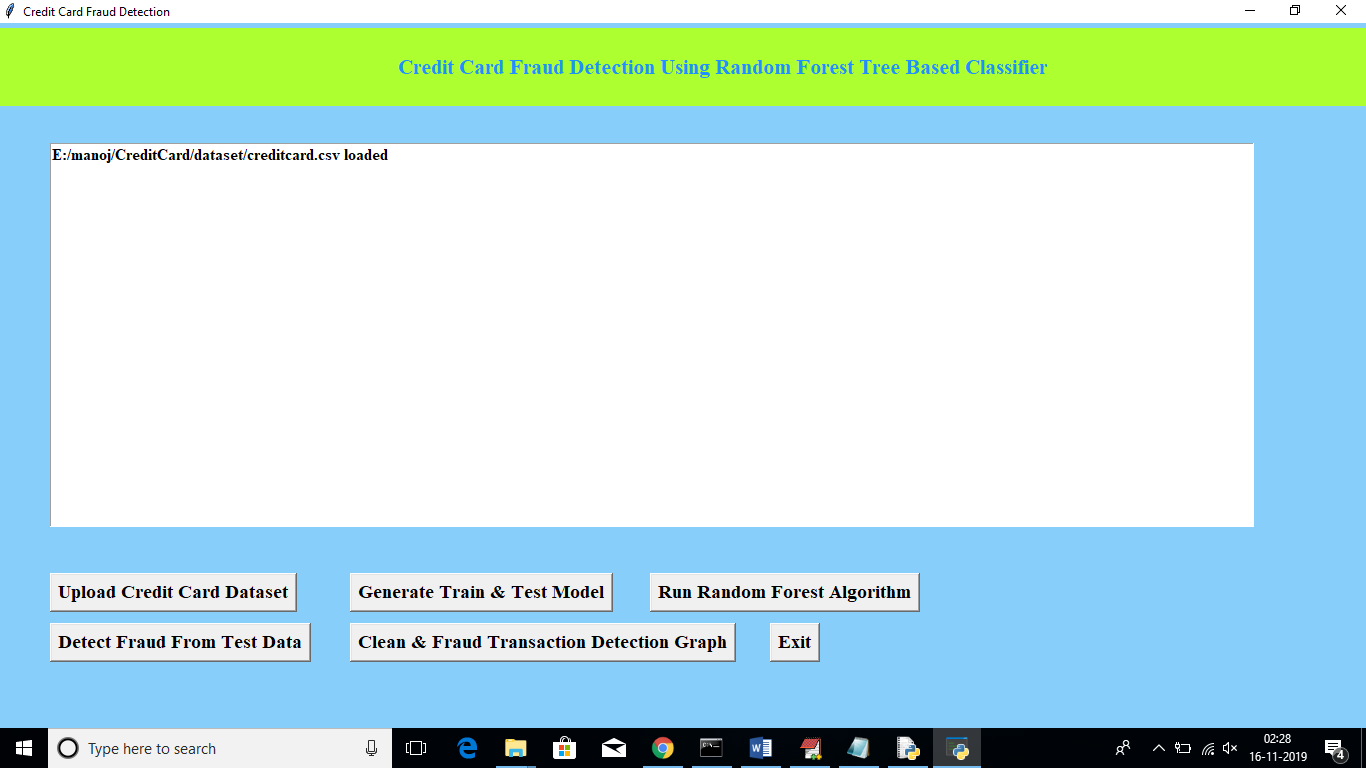
To run project double click on ‘run.bat’ file to get below screen



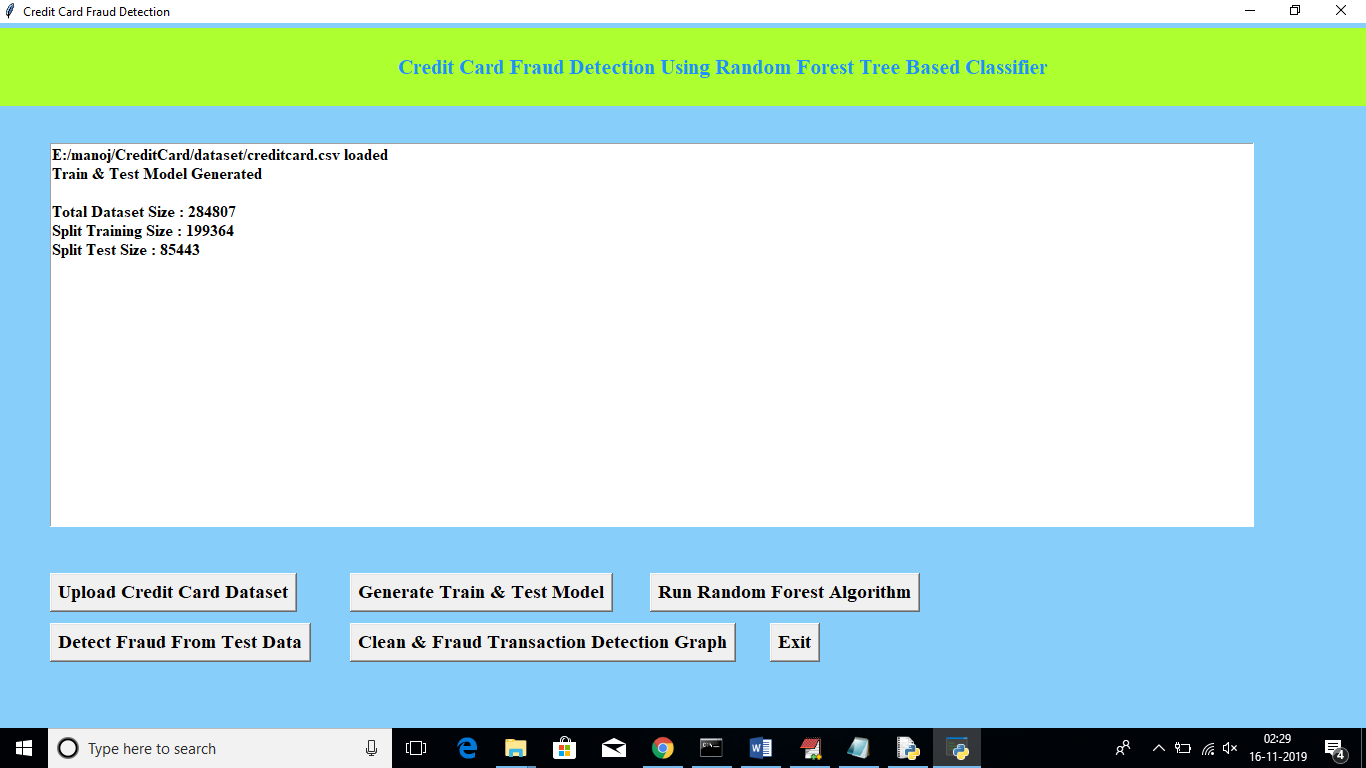
In above screen click on ‘Upload Credit Card Dataset’ button to upload dataset



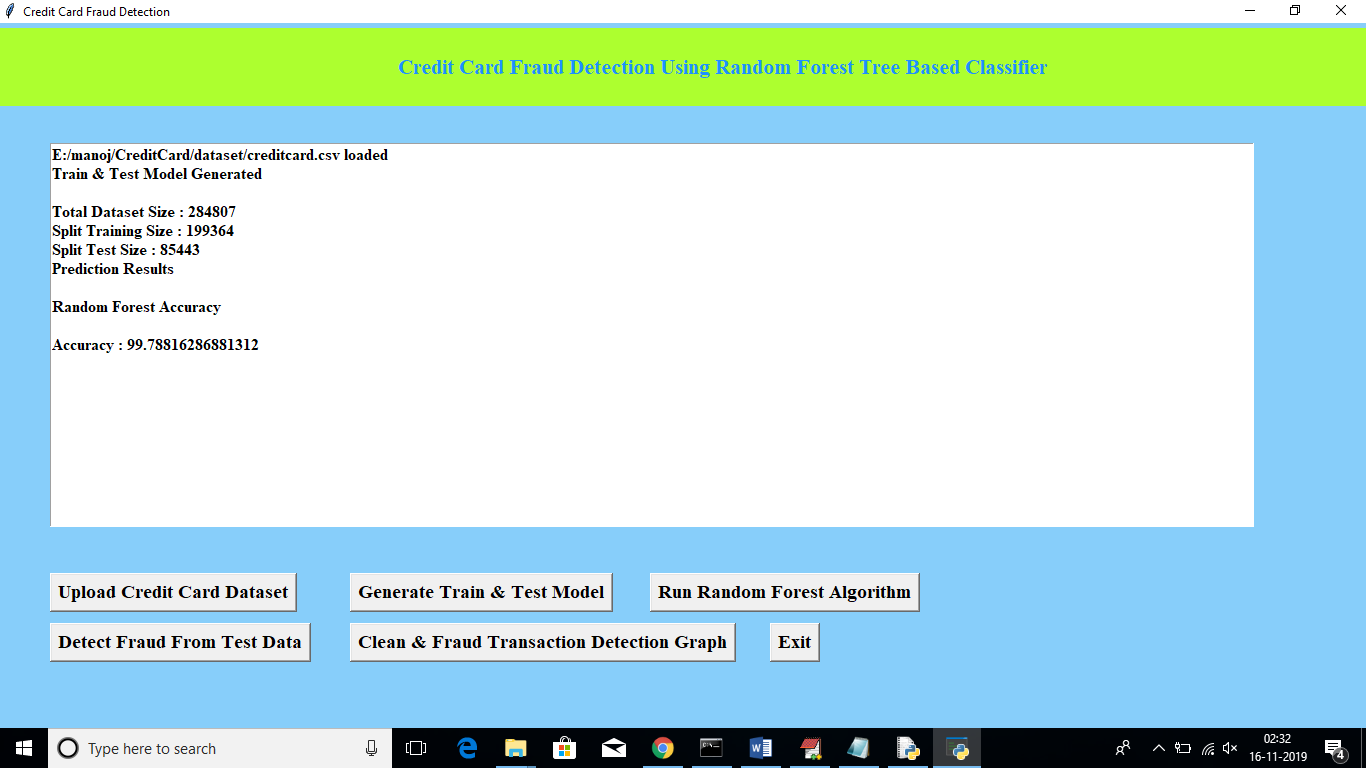
After uploading dataset will get below screen



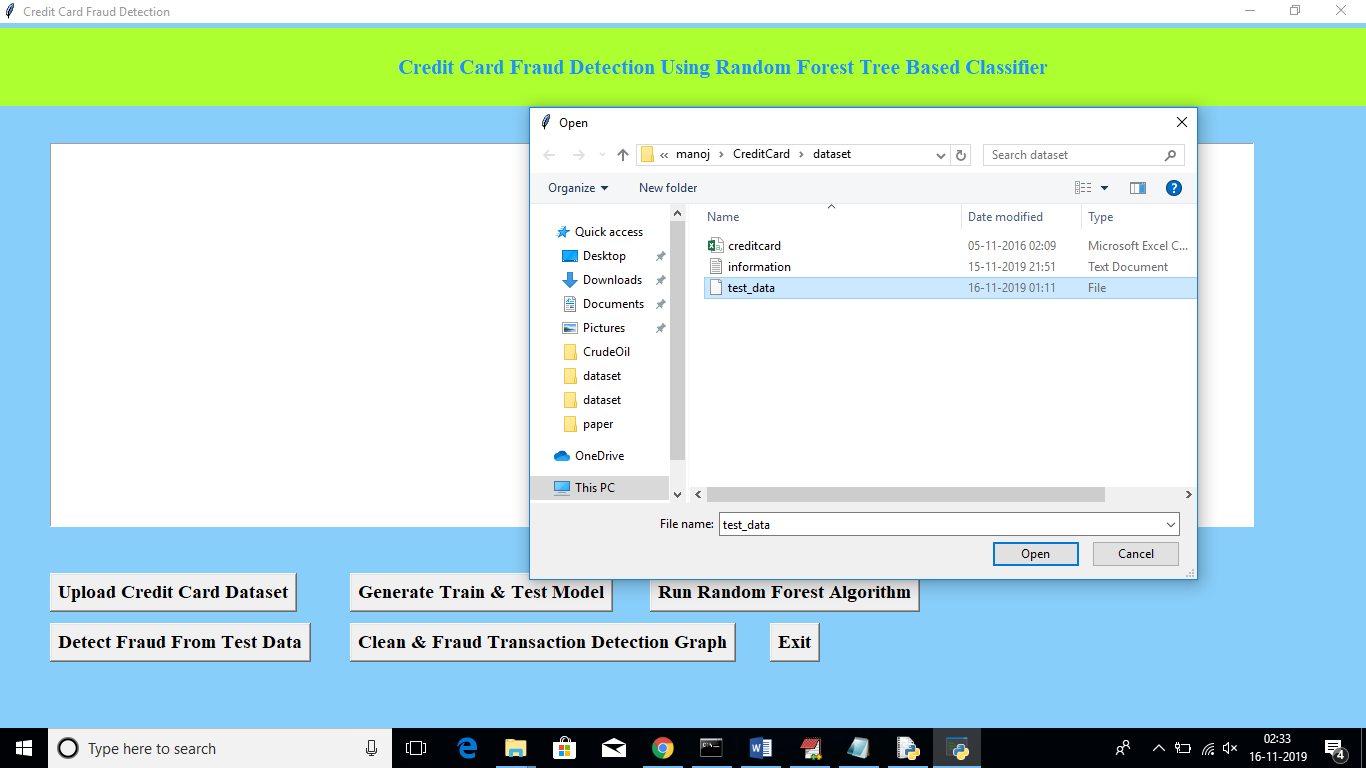
Now click on ‘Generate Train & Test Model’ to generate training model for Random Forest Classifier



In above screen after generating model we can see total records available in dataset and then application using how many records for training and how many for testing. Now click on “Run Random Forest Algorithm’ button to generate Random Forest model on train and test data



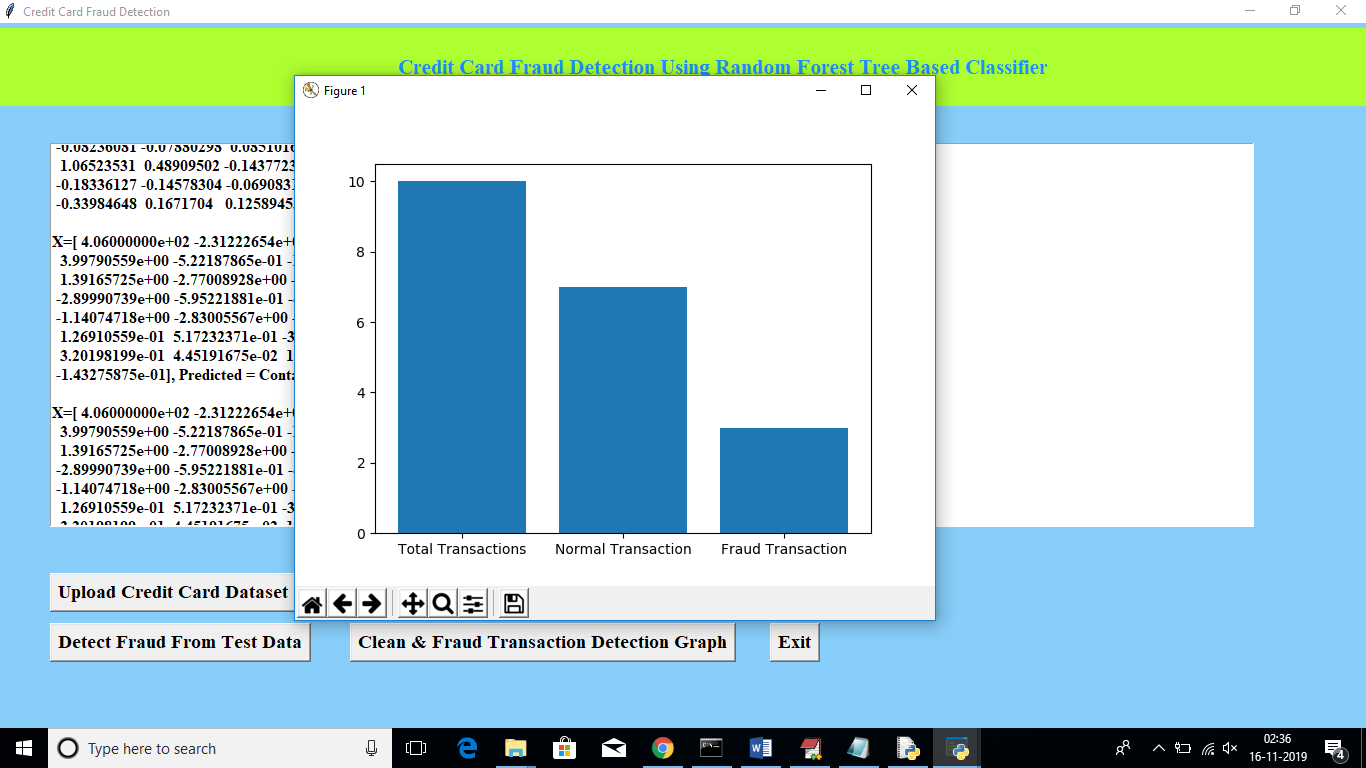
In above screen we can see Random Forest generate 99.78% percent accuracy while building model on train and test data. Now click on ‘Detect Fraud From Test Data’ button to upload test data and to predict whether test data contains normal or fraud transaction



In above screen I am uploading test dataset and after uploading test data will get below prediction details



In above screen beside each test data application will display output as whether transaction contains cleaned or fraud signatures. Now click on ‘Clean & Fraud Transaction Detection Graph’ button to see total test transaction with clean and fraud signature in graphical format. See below screen



In above graph we can see total test data and number of normal and fraud transaction detected. In above graph x-axis represents type and y-axis represents count of clean and fraud transaction