Blockchain and Machine Learning for Fraud Detection: A Privacy-Preserving and Adaptive Incentive Based Approach

Increasing digital transaction making new ways for the fraudsters to perform fraud transaction and to detect such fraud transaction, financial organization heavily dependent on machine learning algorithms which required accurate dataset for correct prediction. In the existing system all organizations were using single centralized server to train ML algorithms whose database can be easily tamper by database administrator without getting detected. ML algorithms trained on tamper data will start predicting incorrect transaction and enable fraudsters to make successful transaction.

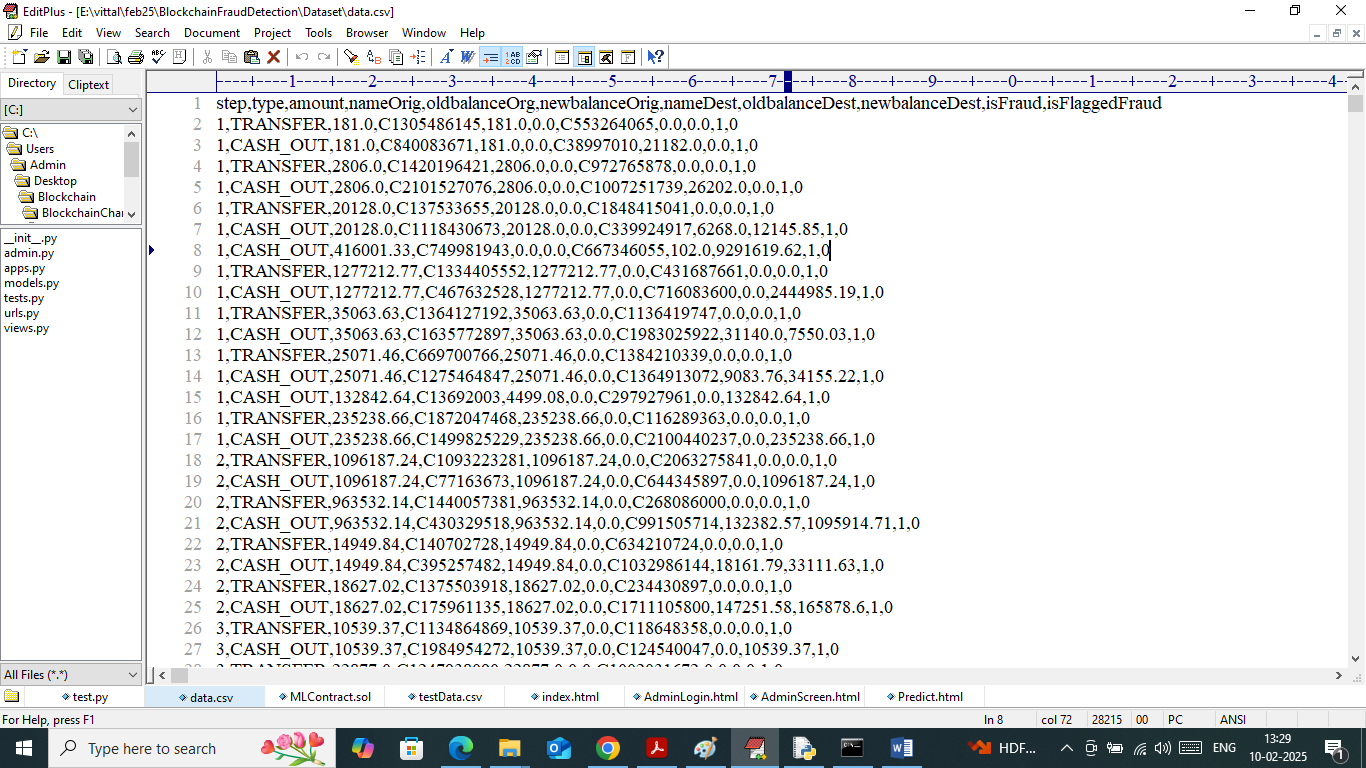
To avoid above issue author of this paper employing Blockchain technology to maintain ML models whose database cannot be tamper in any manner and Blockchain has inbuilt support for data privacy and security. Blockchain store each record as block/transaction and associate each block with unique hashcode and this hashcode get verified for subsequent block storage, if data tamper in any block then result to hashcode mismatch and get detected. This process of verification make Blockchain tamper proof.

In propose work author experimenting with various ML algorithms such as Passive Aggressive Classifier (PAC), Stochastic Gradient (SGD), Perceptron and Naïve Bayes and then training all this algorithm with incremental support to update model weights with new and old training data. Each algorithm performance is evaluated in terms of accuracy, precision, confusion matrix, recall and FSCORE. Model with best accuracy will be updated to Blockchain and all organization can train local model and update best model weights to Blockchain. Anytime we can obtained weights from Blockchain and then can perform prediction. Due to high amount of data so Blockchain will take more time for mining so author giving incentives based on Difficulty level. Difficulty level can be calculated by splitting data into multiple parts and each part will get trained incrementally. Parts with high number of records may consume more Blockchain mining time.

To train and test above algorithms performance author has used ‘Financial Transaction dataset’ which can be downloaded from below KAGGLE website

<https://www.kaggle.com/datasets/ealaxi/paysim1/data>

In below screen sowing dataset details

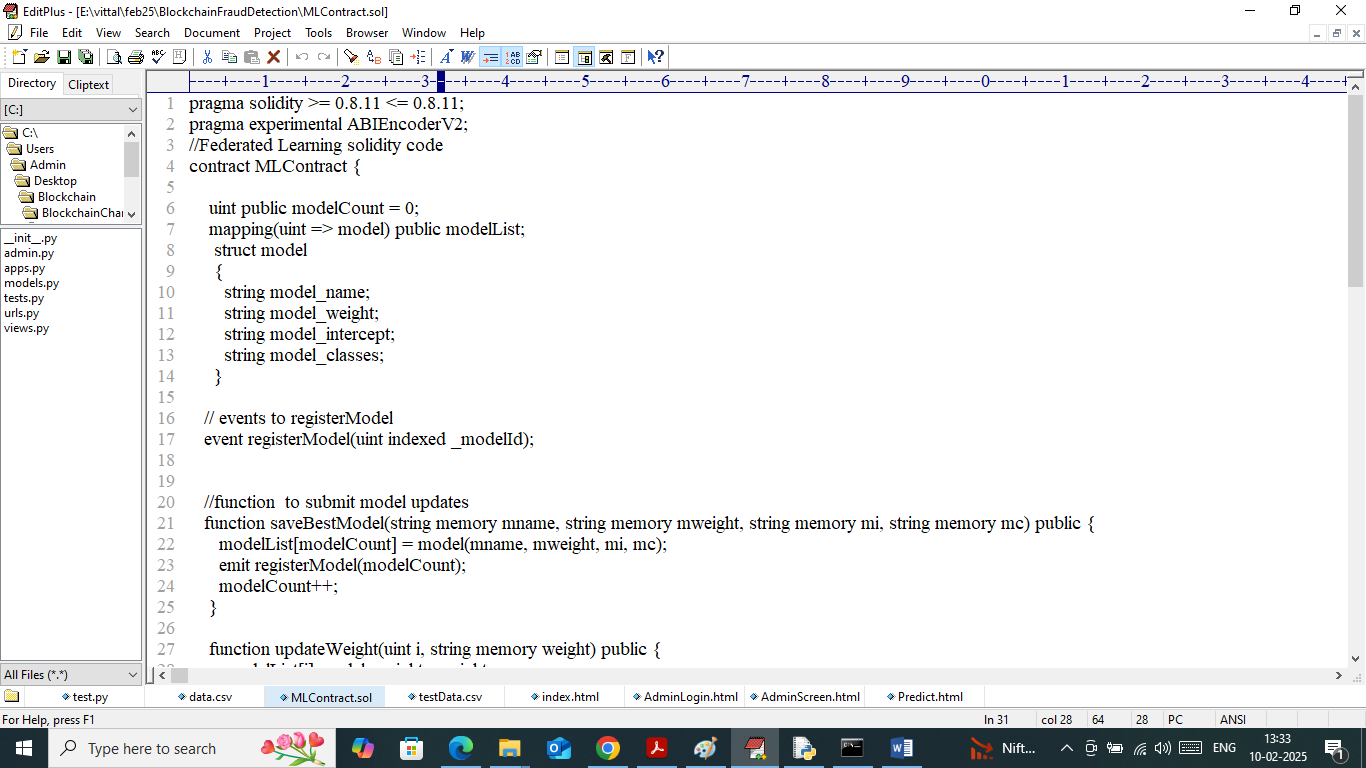


In above dataset screen first row contains dataset column names and remaining rows contains dataset values and in each record contains class label as 0 (normal) and 1 (fraud). So by using above dataset will train and test each algorithm performance.

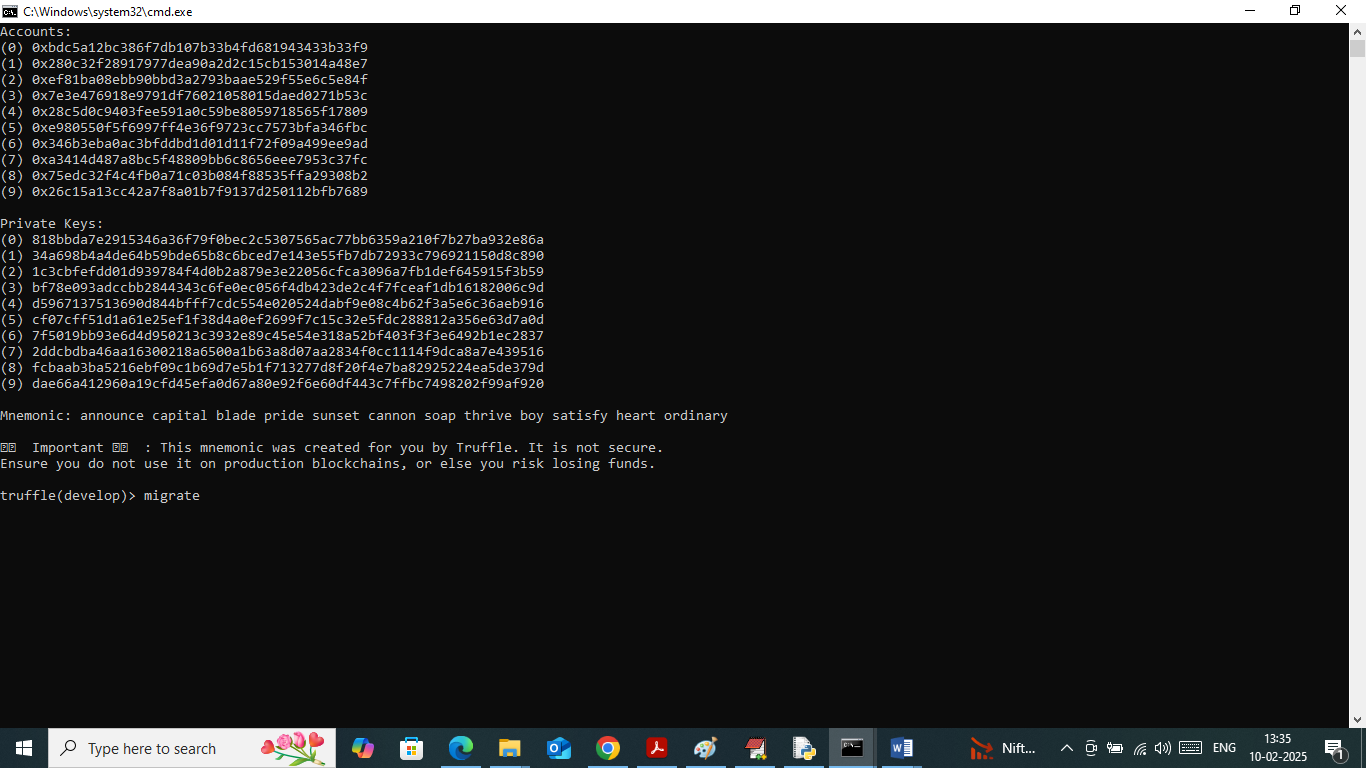
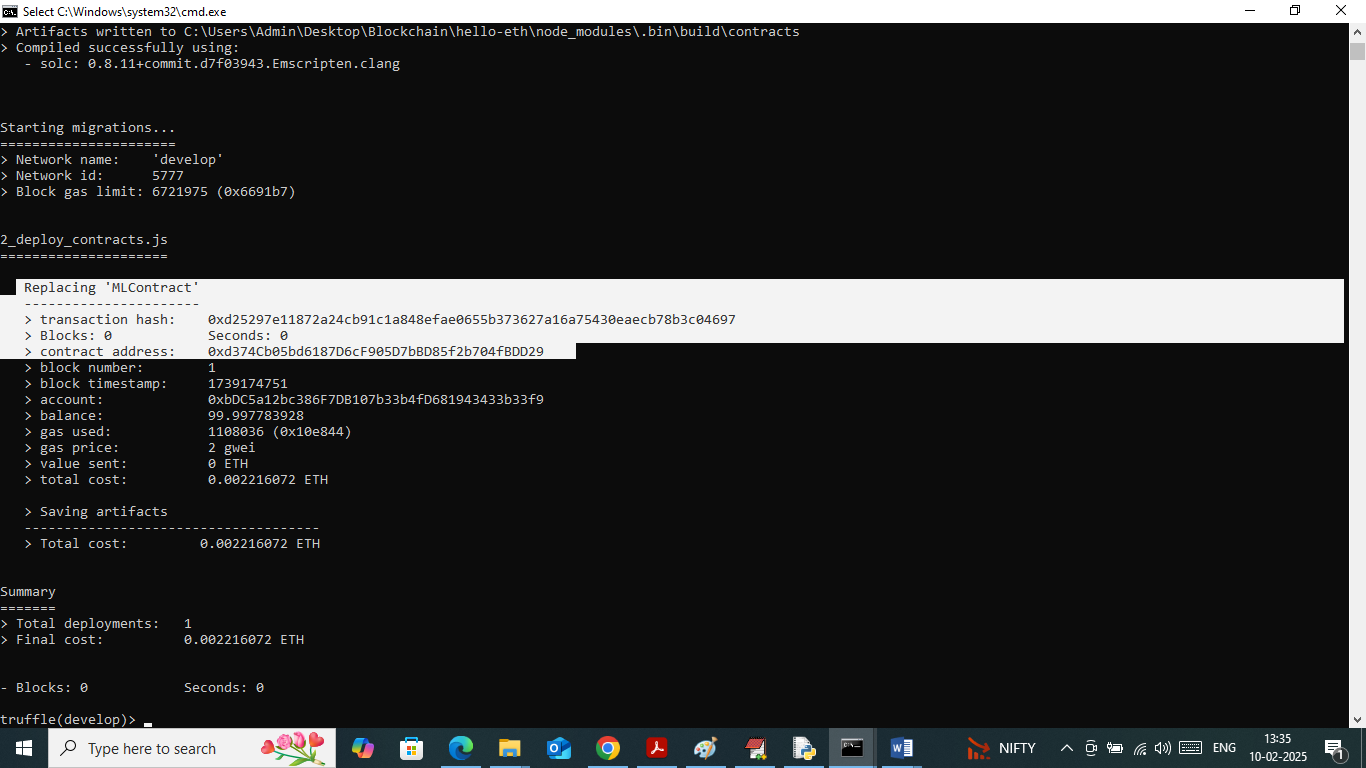
**Note:** to calculate difficulty level we have divided dataset into two parts where first part contains 10000 records as Difficulty Level1 and second part contains 30000 records as difficulty level2. Both levels of data get trained incrementally.

**Blockchain Smart Contract Deployment**

Blockchain can store and retrieve data using Smart Contract which can be designed using SOLIDITY programming. Smart contract contains functions which can be called using any programming language to store and retrieve data. In propose work to manage ML weights we have designed following contract

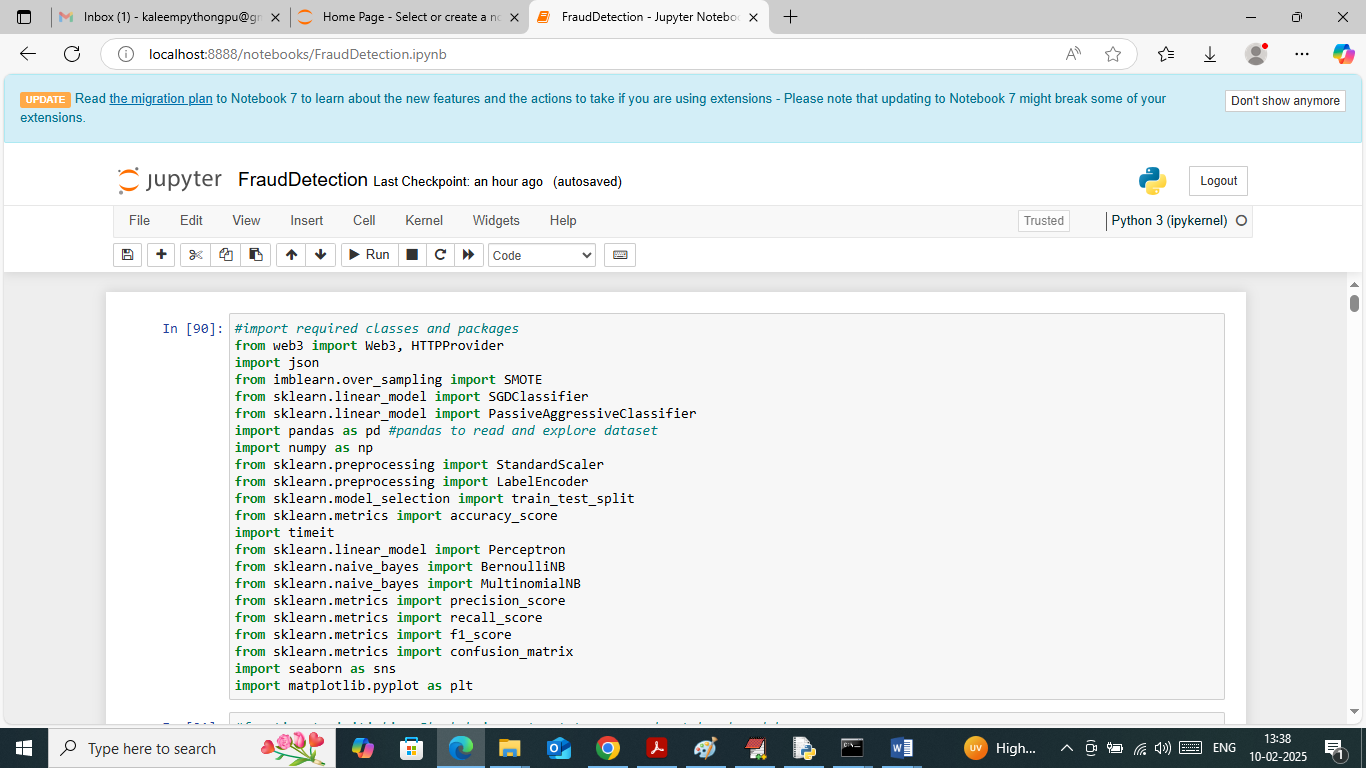


In above contract we have defined function to manage model weights and now we need to deploy above contract to Blockchain Ethereum using below steps

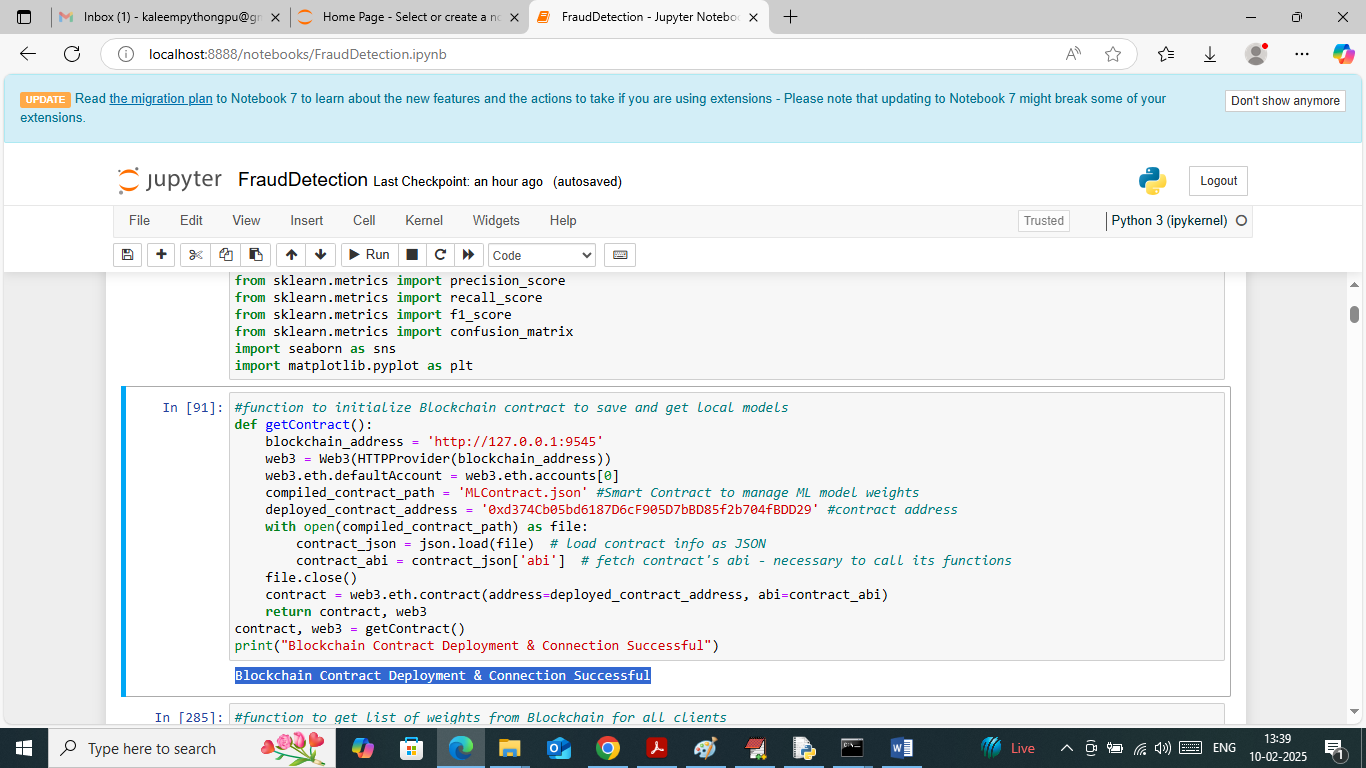
1. First go inside ‘hello-eth/node-modules/bin’ folder and then look and double click on ‘runBlockchain.bat’ file to get below page
2. 
3. In above screen Ethereum started with default accounts and private keys and then type command as ‘migrate’ and then press enter key to get below page
4. 
5. In above screen in white colour text can see ‘MLContract’ contract deployed and got contract address also and this address need to specify in python or any other programming language to call contract.
6. In above screens we have successfully deployed contract and running also and let this Ethereum running till project complete execution.

SCREEN SHOTS

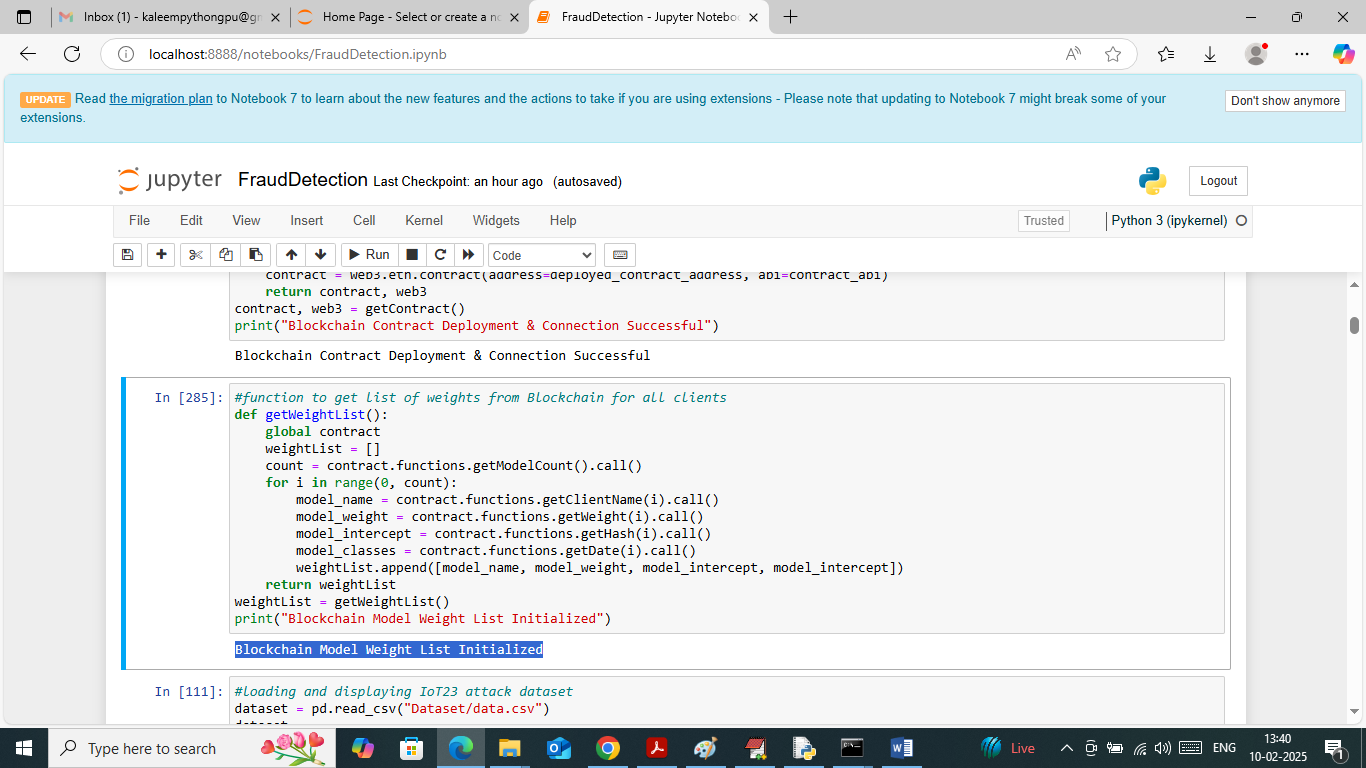
We have trained and test each ML model using JUPYTER notebook and then perform prediction on test data using flask framework. In below screen showing JUPYTER code and output with blue colour comments



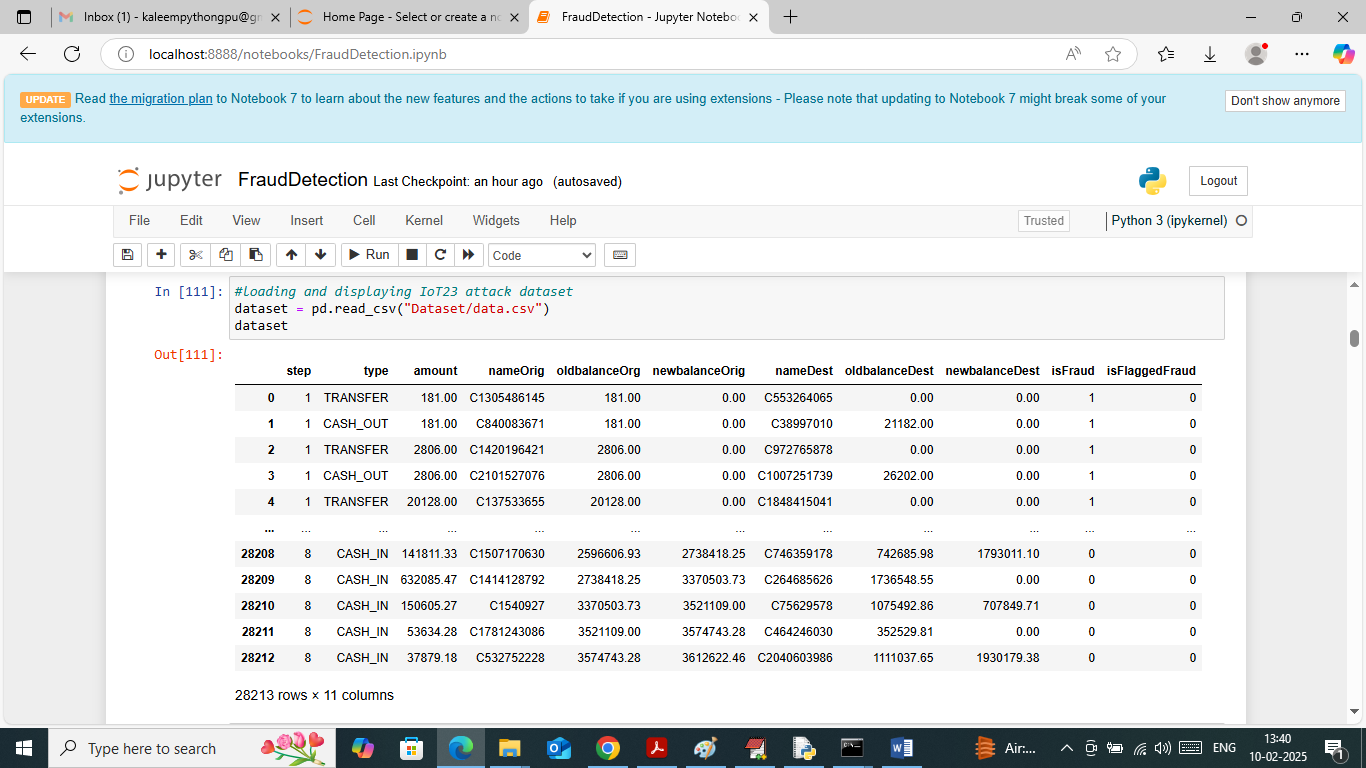
In above screen importing required python classes and packages



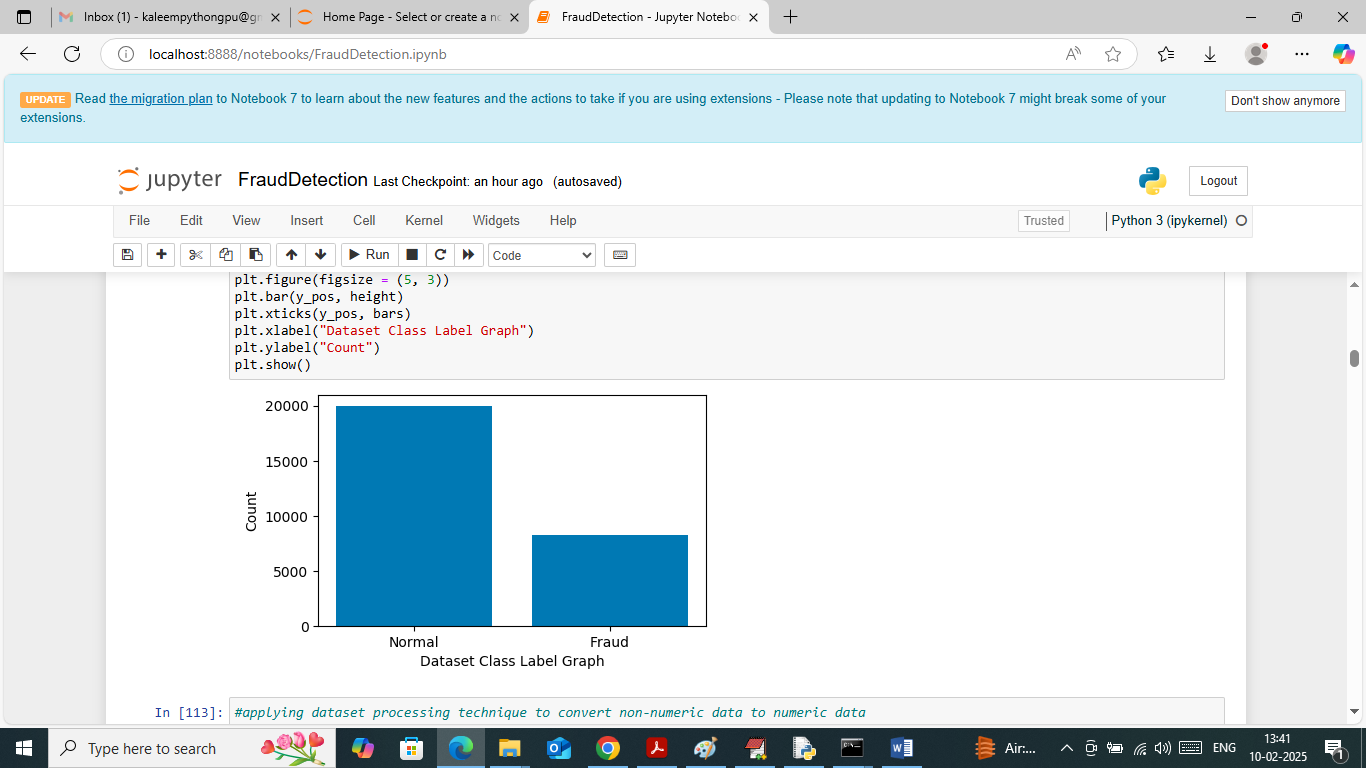
In above screen using contract address we are connecting to deployed smart contract running in Blockchain



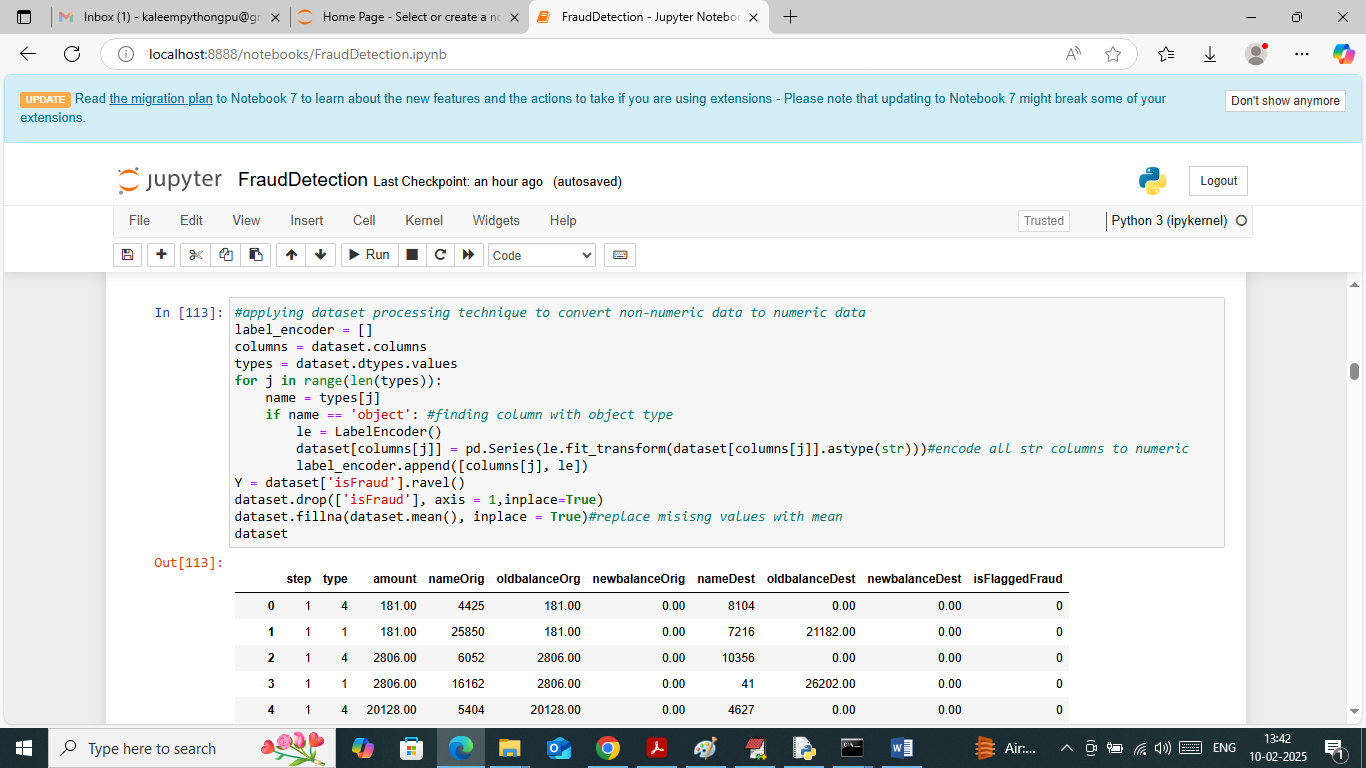
In above screen defining function to read existing available weights from Blockchain. In above screen connection to Blockchain successful



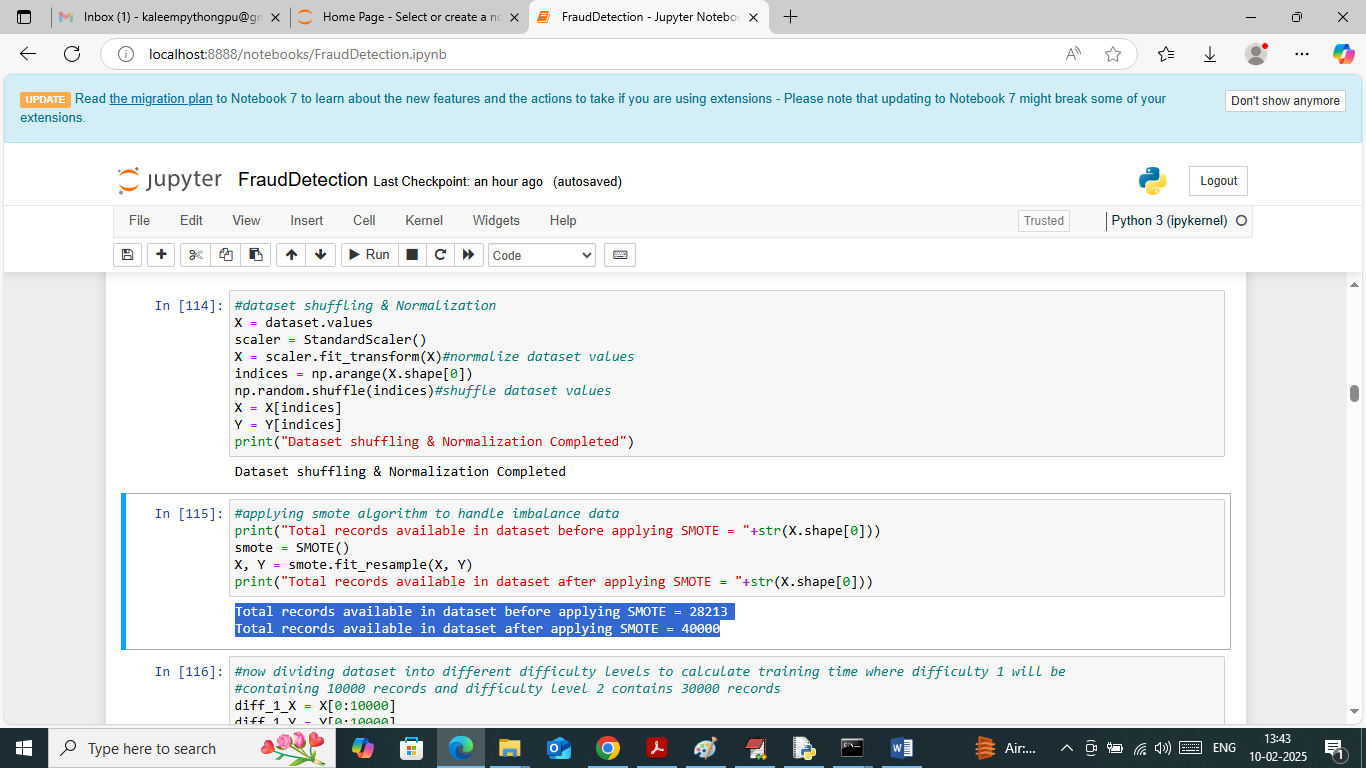
In above screen loading and displaying dataset values and above dataset contains some non-numeric values but ML take only numeric values so by applying Label Encoder class can convert non-numeric to numeric data



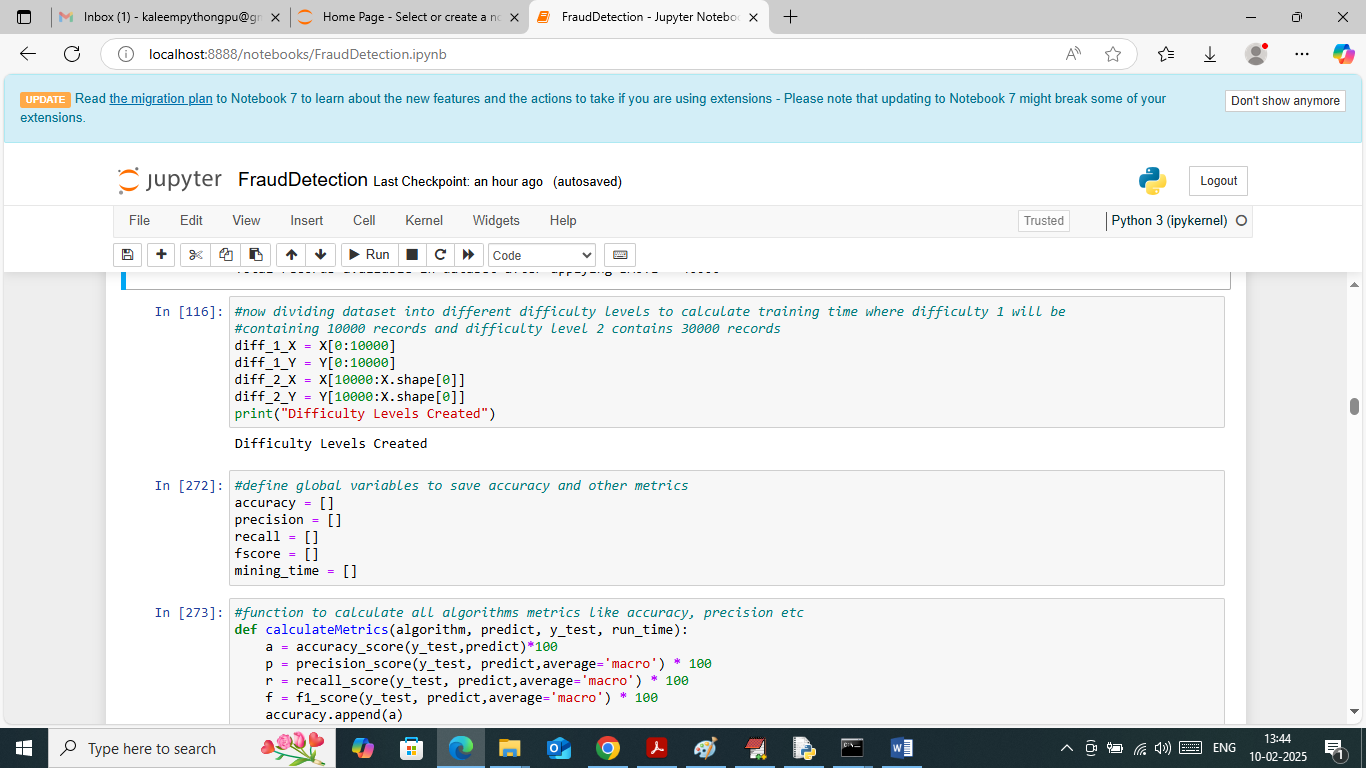
In above screen visualizing graph of normal and fraud transaction exists in dataset



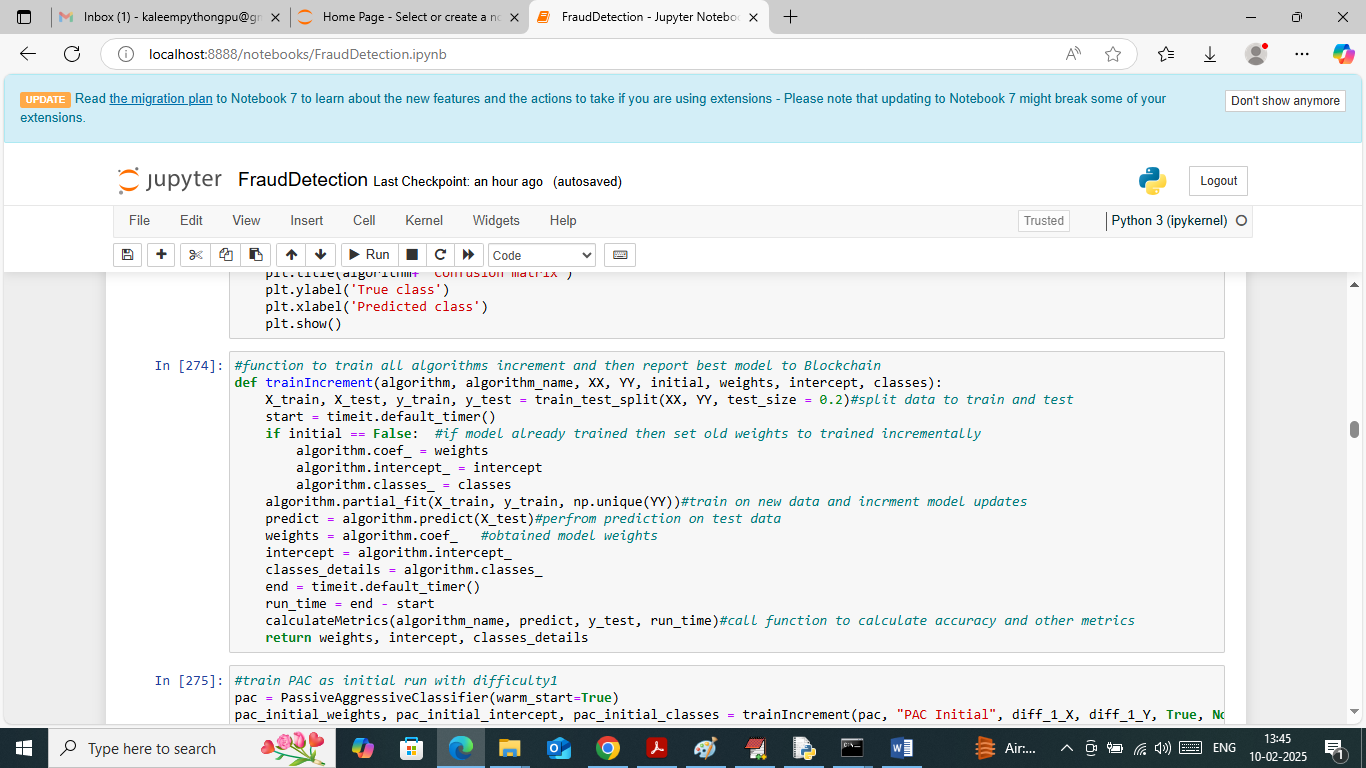
In above screen applying label encoder class to convert non-numeric data to numeric data and then replacing missing values with MEAN



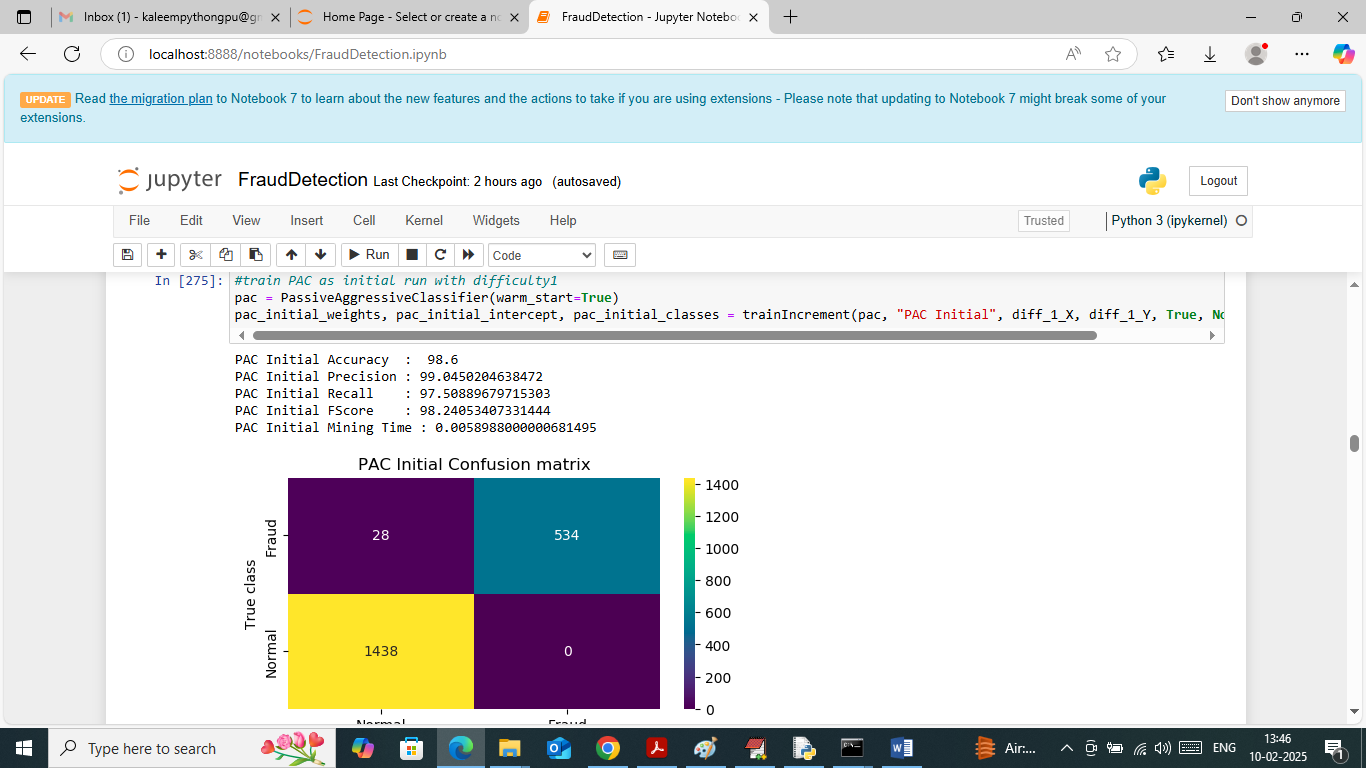
In above screen applying processing technique in first block to shuffle and normalized dataset values. In second block applying SMOTE technique to handle imbalance issues. In above screen before applying smote dataset were having 28000 records and after applying smote dataset size increased to 40000



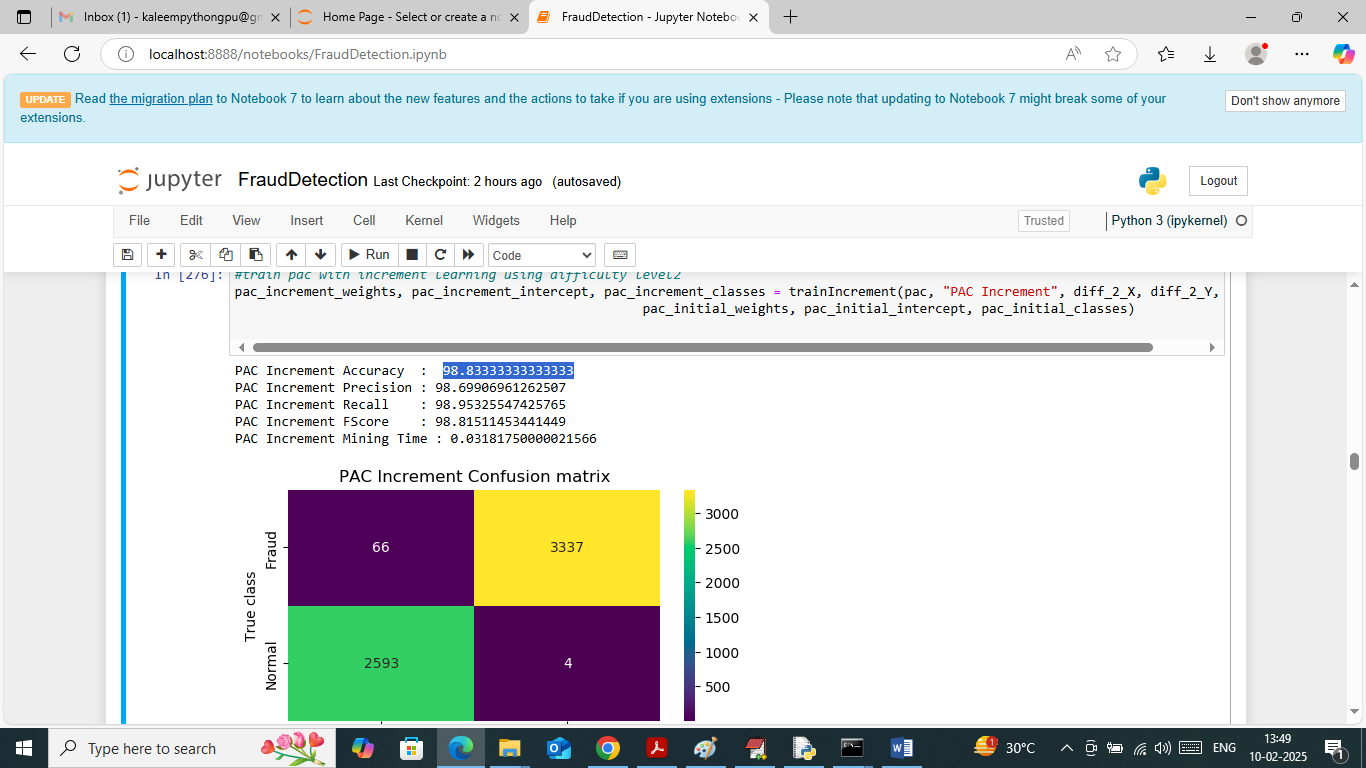
In above screen dividing dataset into two difficulty levels were difficulty 1 having 10000 records and difficulty 2 having 30000 records. In next blocks defining function to calculate accuracy and other metrics



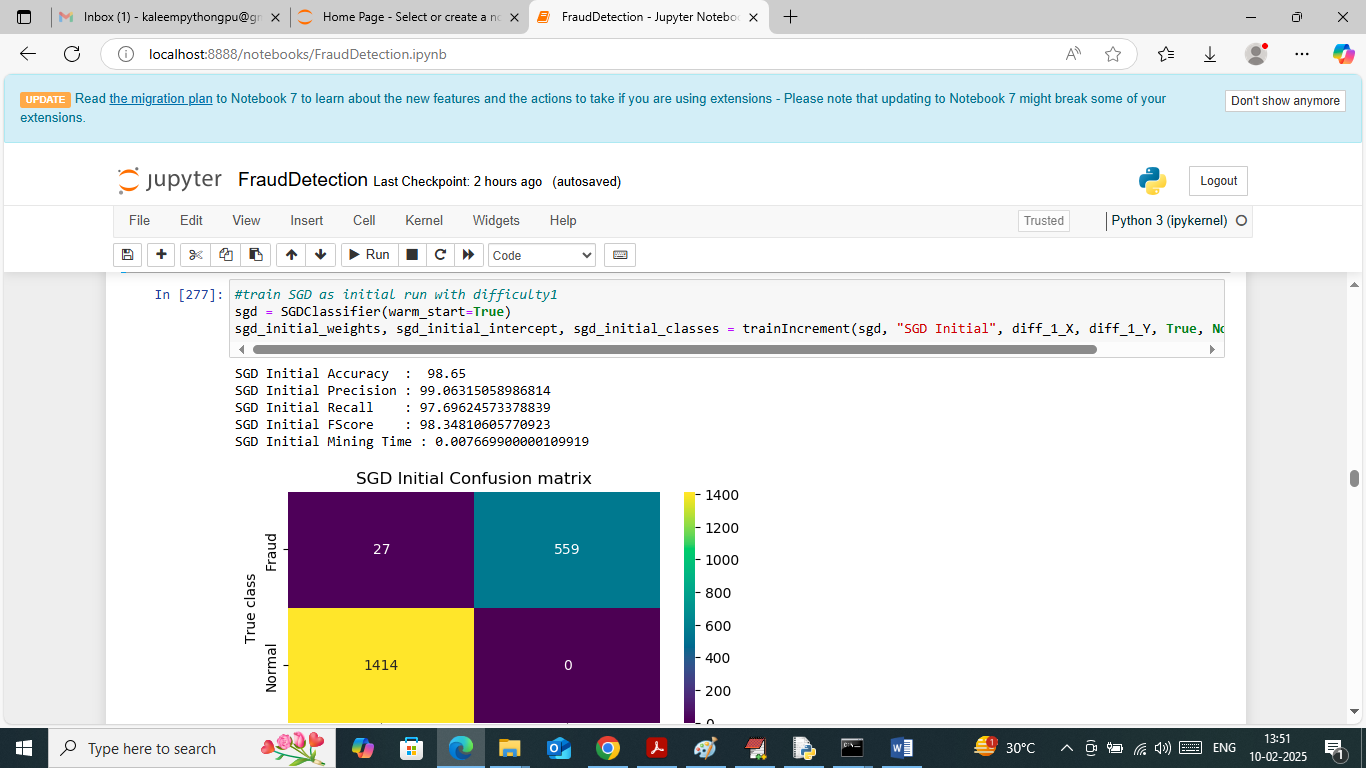
In above screen defining function to train algorithm incrementally and then calculate accuracy and other metrics and then getting weights to update to Blockchain



In above screen training PAC algorithm initially with difficulty 1 (10000 records) and then PAC got 98.6% accuracy and can see other metrics like precision, recall and FSCORE. In above confusion matrix graph x-axis represents Predicted Labels and y-axis represents True Labels (normal or fraud) and then yellow and light green boxes in diagonal represents correct prediction count and remaining blue boxes represents incorrect prediction count which are very few. In above screen can see training time also



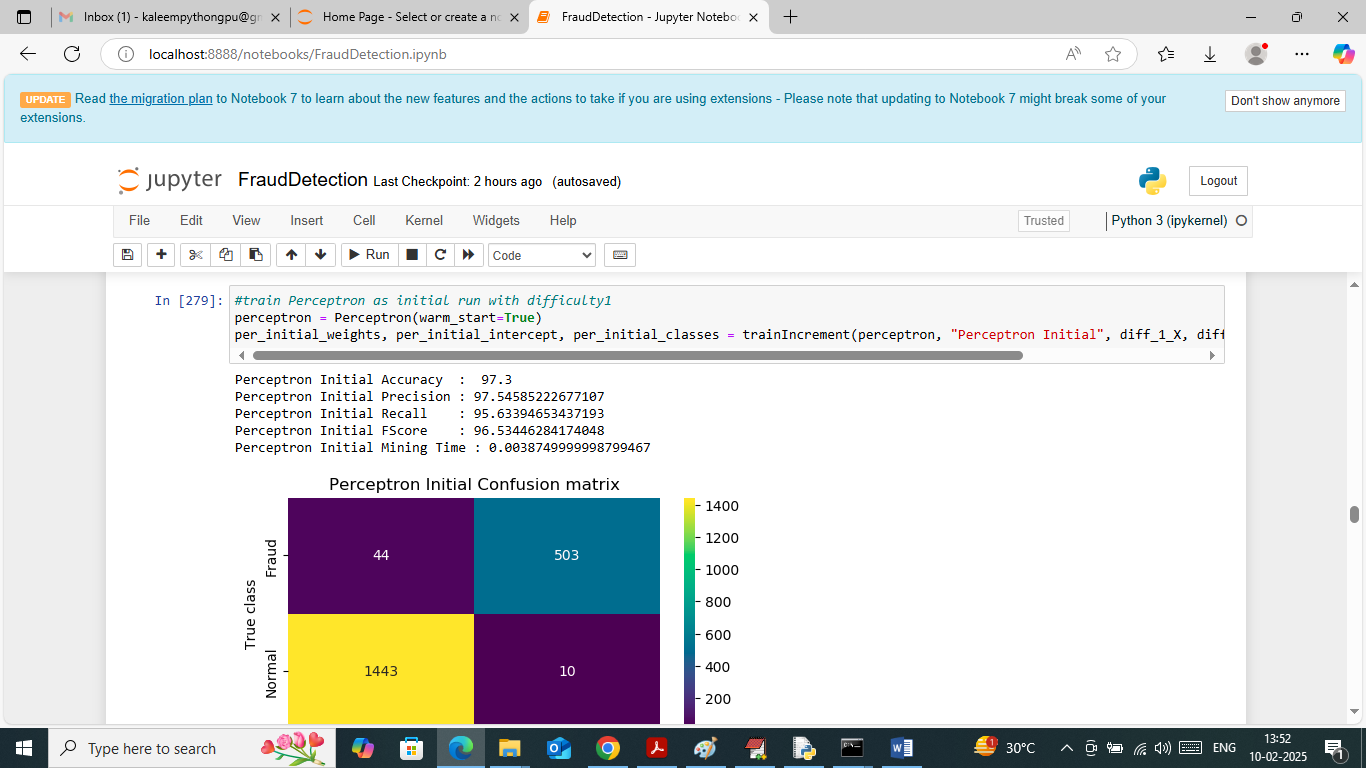
In above screen PAC trained incrementally on difficulty level 2 (30000 records) and then got 98.83% accuracy and can see other metrics output also



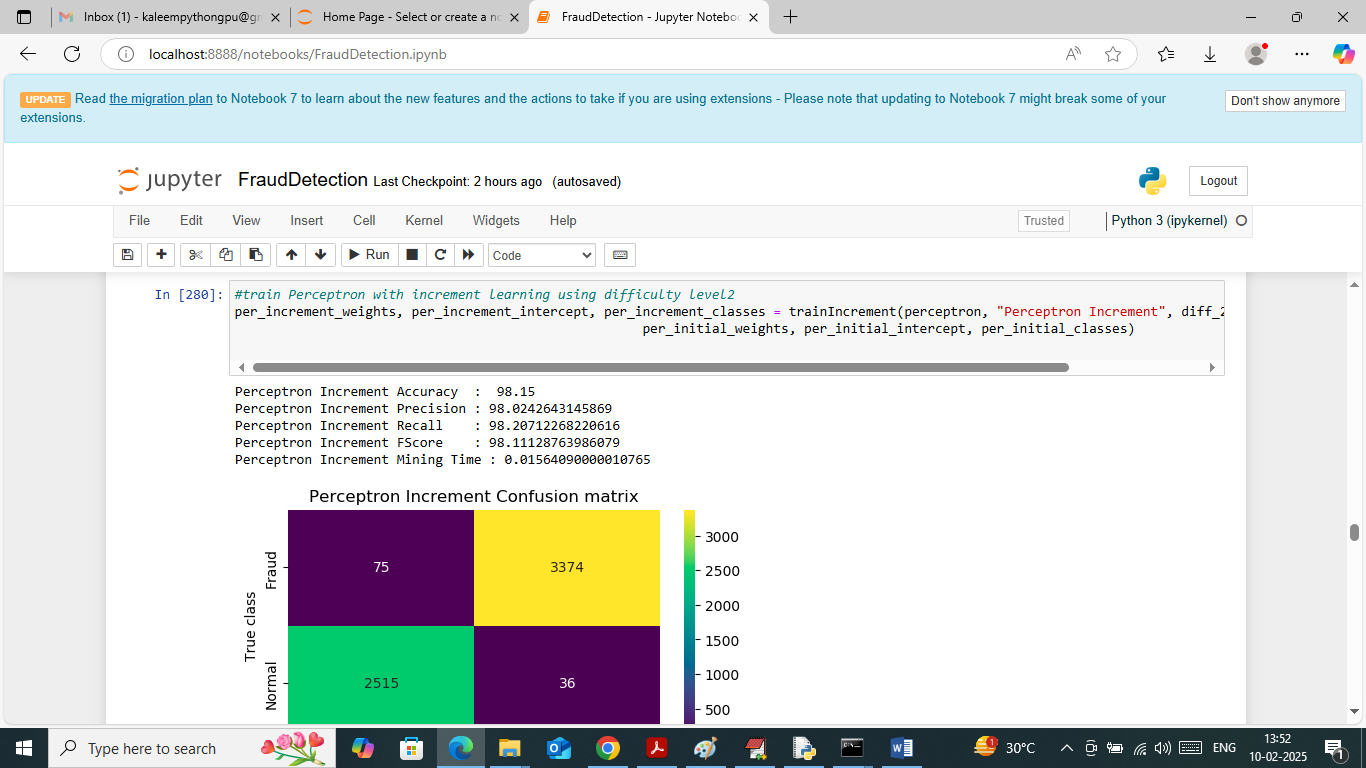
In above screen SGD on initial data got 98.65% accuracy



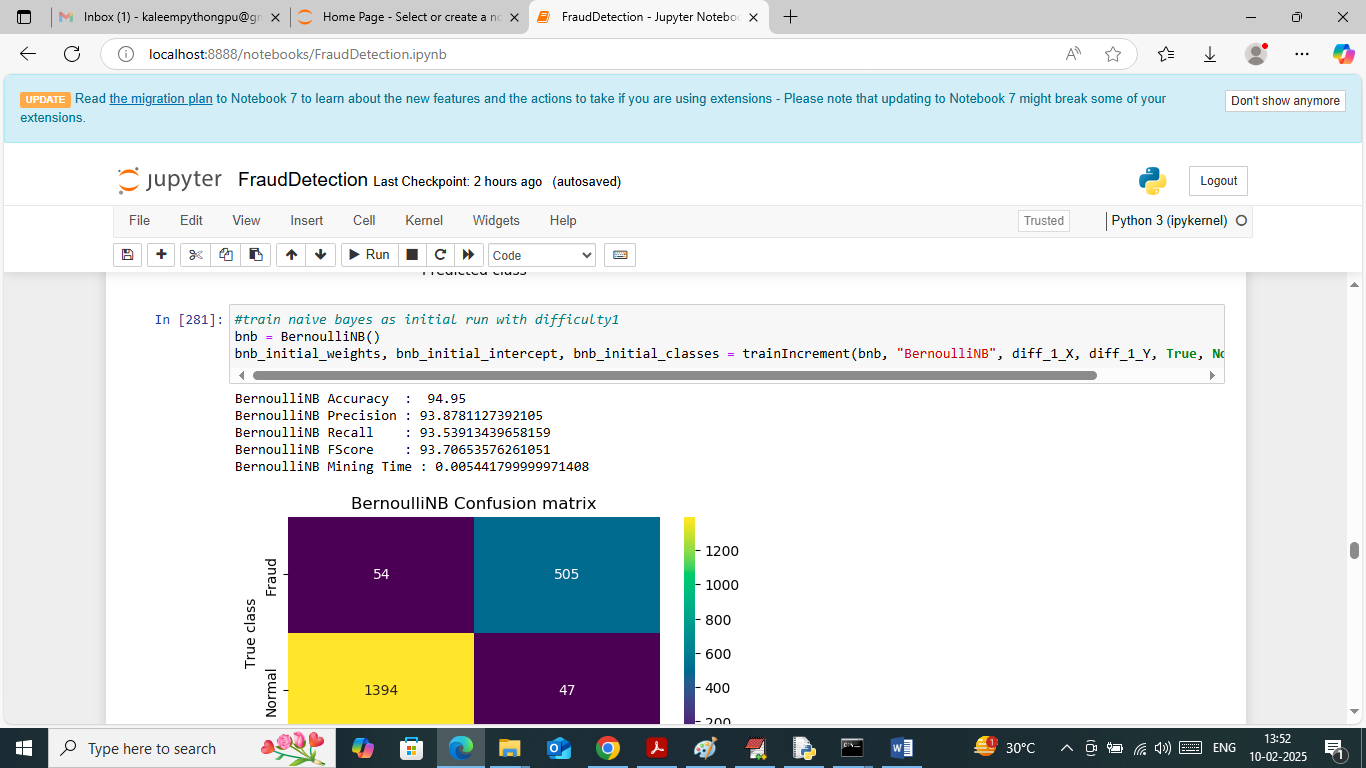
In above screen SGD on increment data got 99.83% accuracy



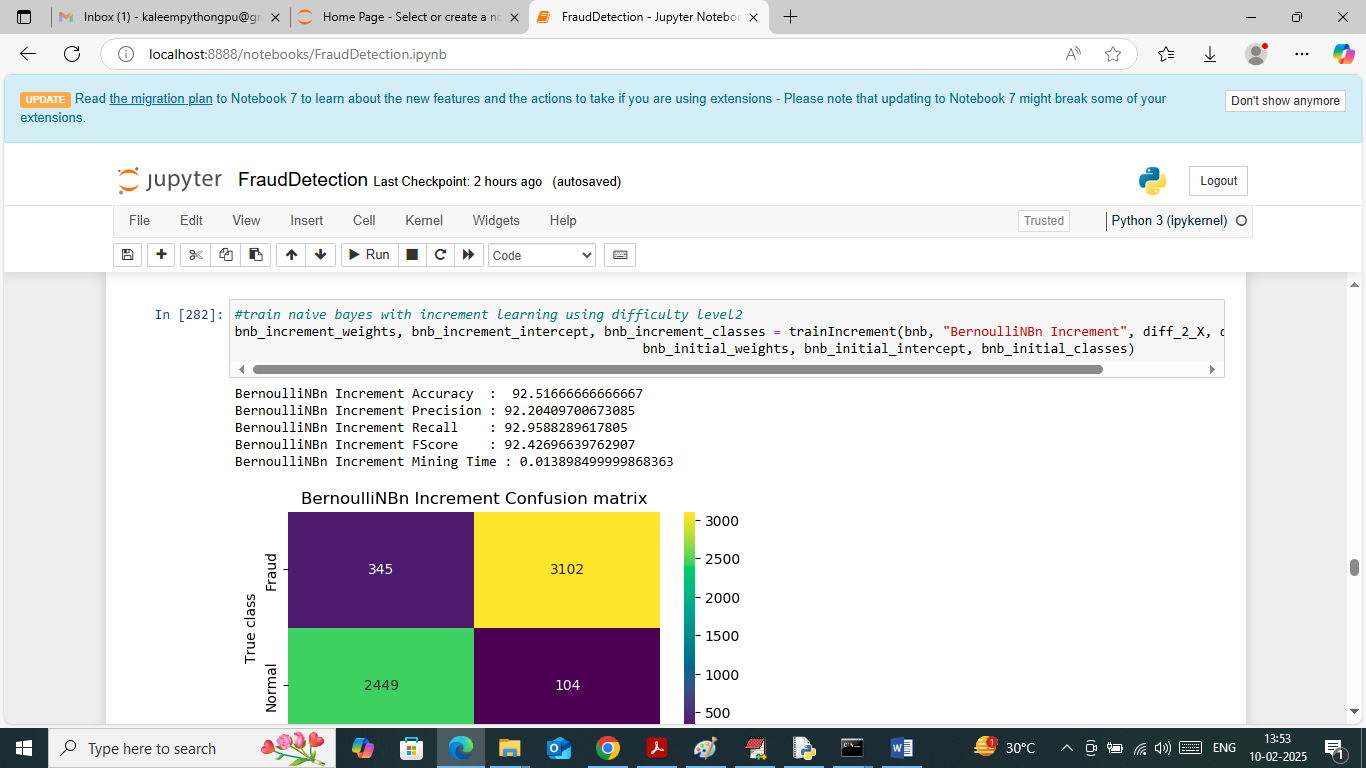
In above screen Perceptron got 97% accuracy on initial data



In above screen perceptron got 98% accuracy on increment training



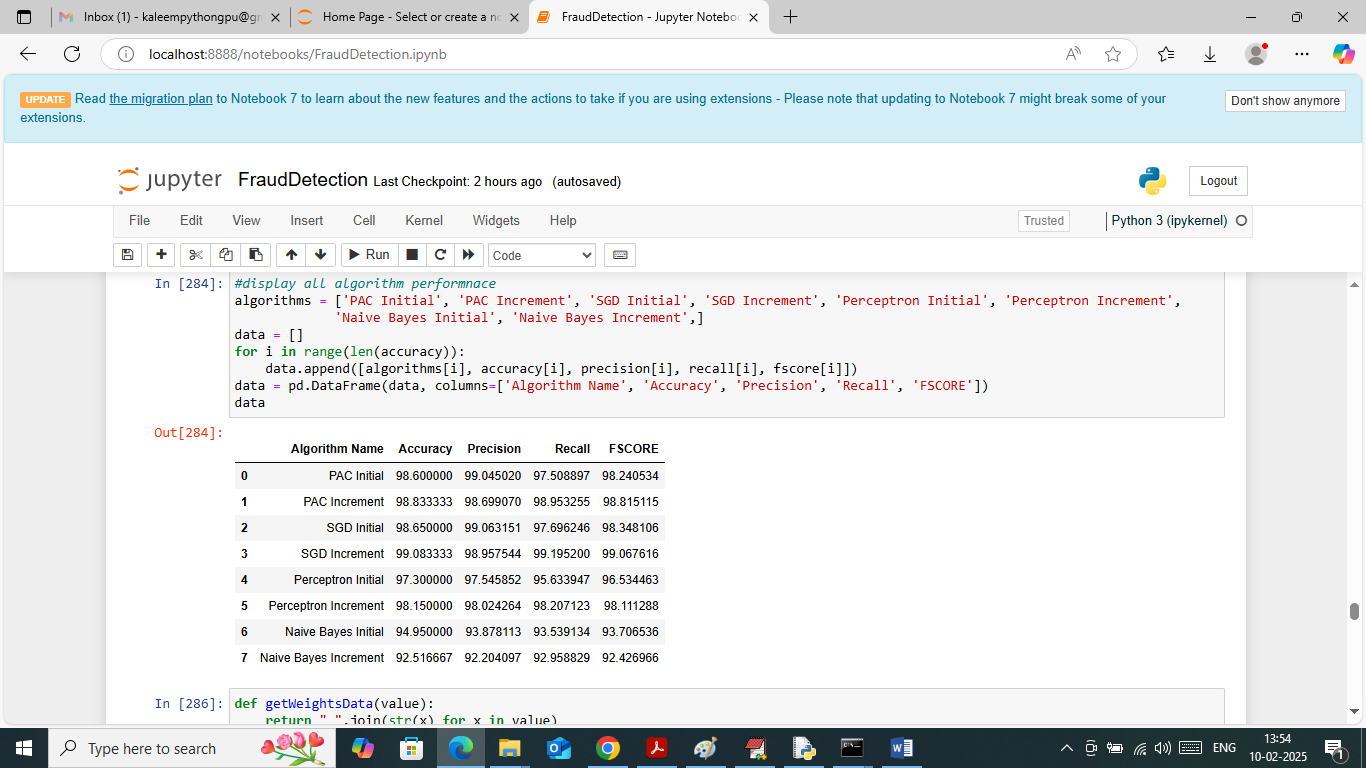
In above screen Naïve Bayes got 94% accuracy on initial data



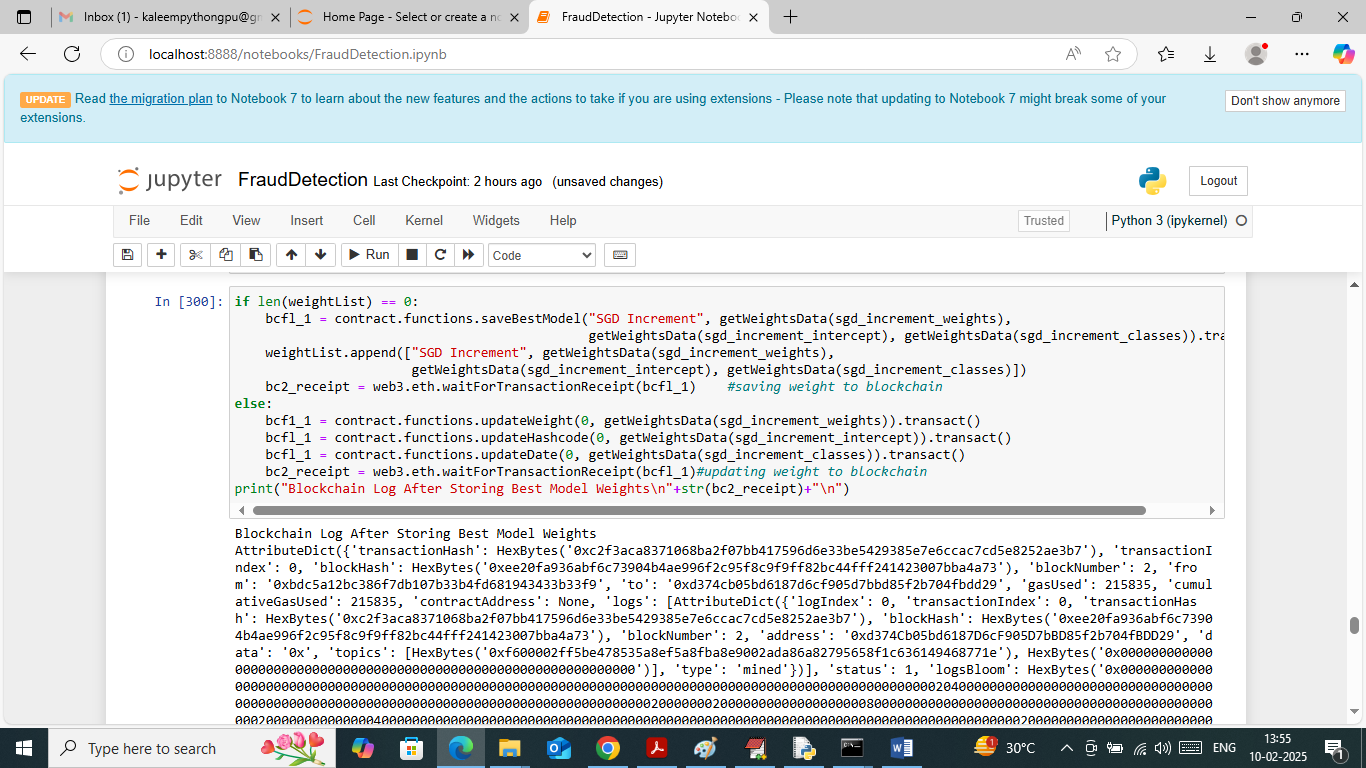
In above screen naïve bayes got 92% accuracy on increment training



In above screen visualizing all algorithm performance where x-axis represents algorithm names and y-axis represents accuracy and other metrics in different colour bars.

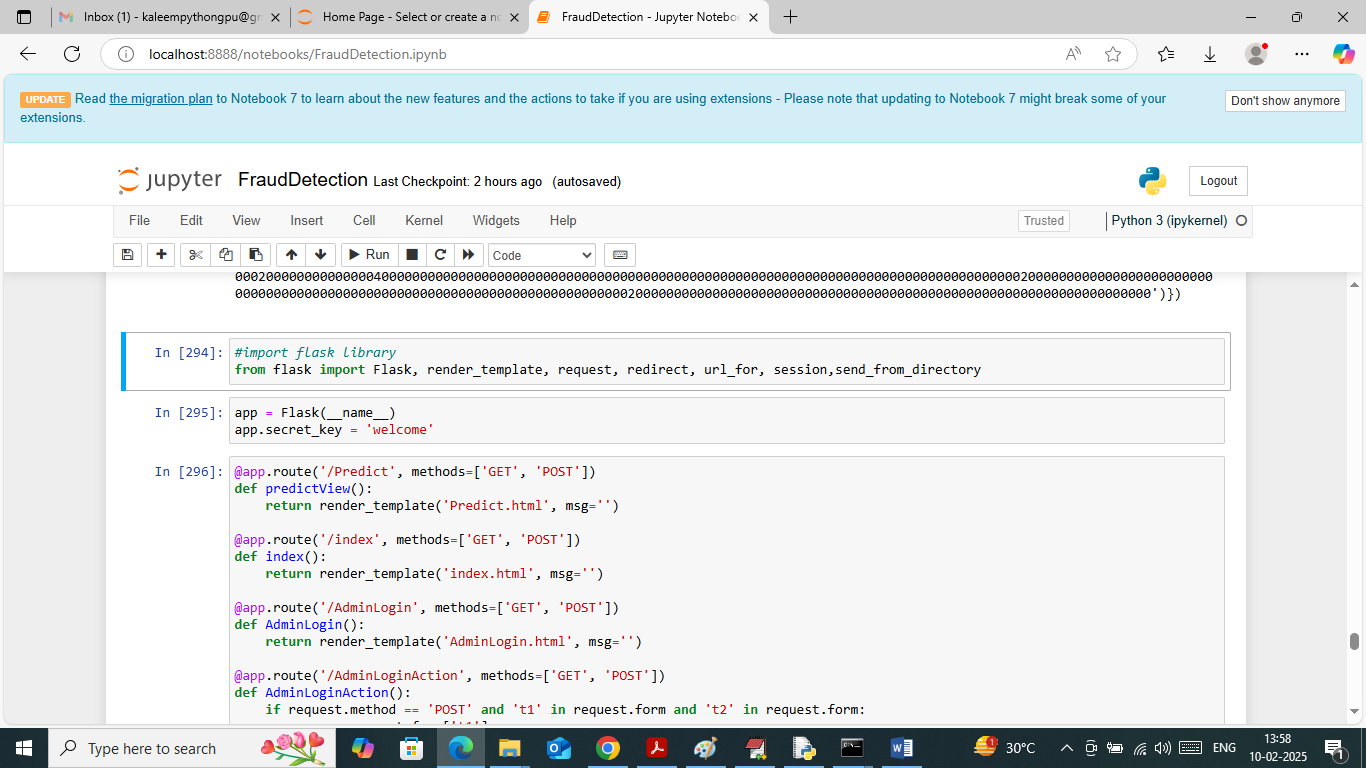


In above screen showing all algorithms performance in tabular format and in all algorithms SGD with increment training got high accuracy

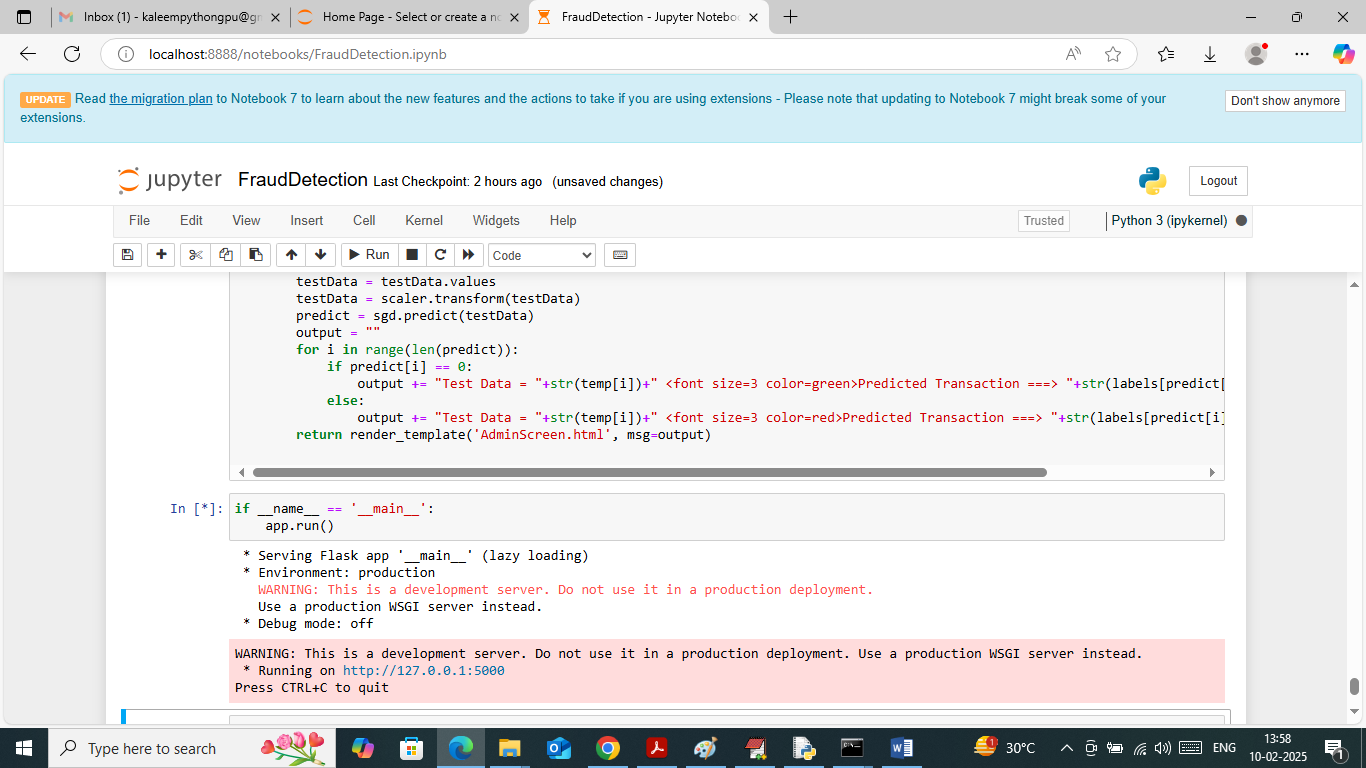


In above screen saving SGD weights to Blockchain as it got high accuracy and consider as best model. In above screen after saving weights to Blockchain we got all log details which contains details like Block No, transaction no, hash code and many other details.

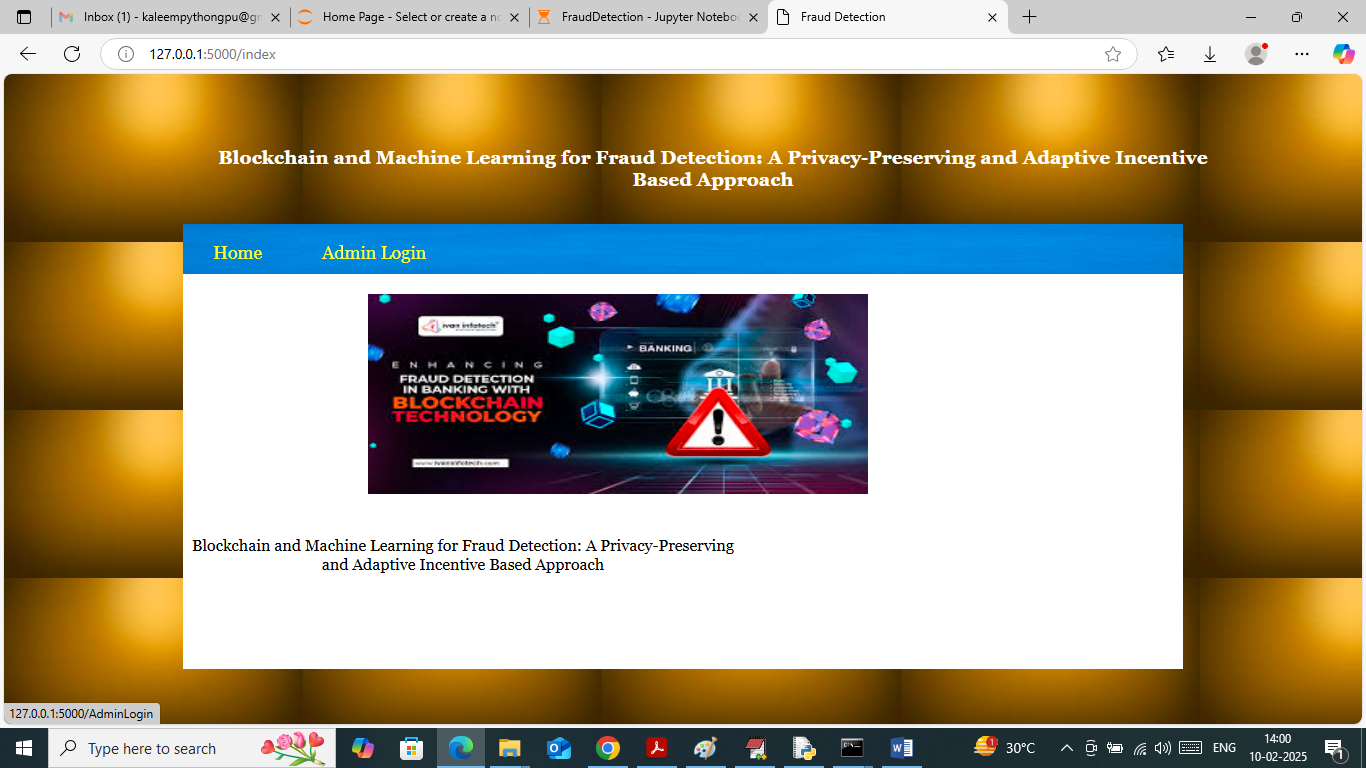
IN ABOVE SCREENS WE TRAINED ML MODELS INCREMENTALLY AND THEN SAVE BEST WEIGHTS IN BLOCKCHAIN AND NOW WILL DETECT FRAUD TRANSACTION USING BELOW WEB OUTPUT



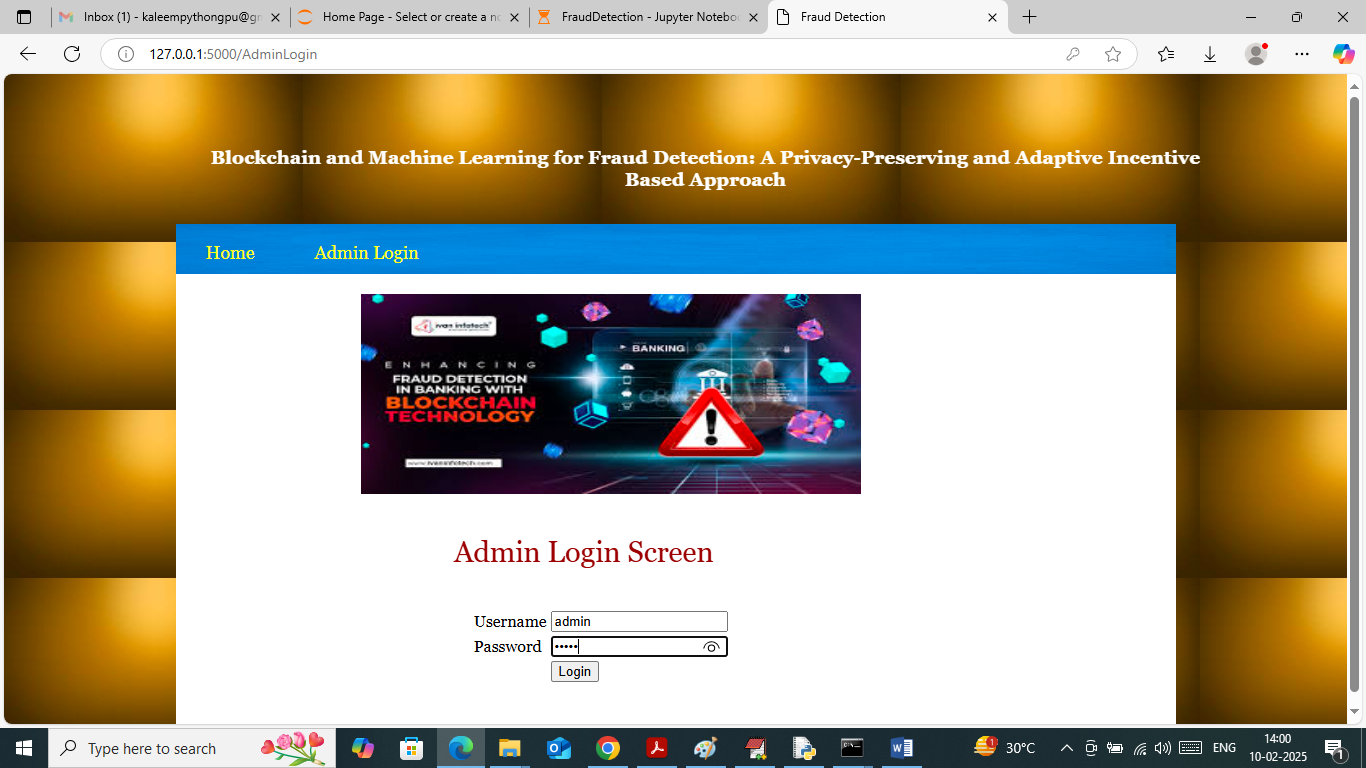
In above screen run all flask blocks to start FLASK web server and get below page



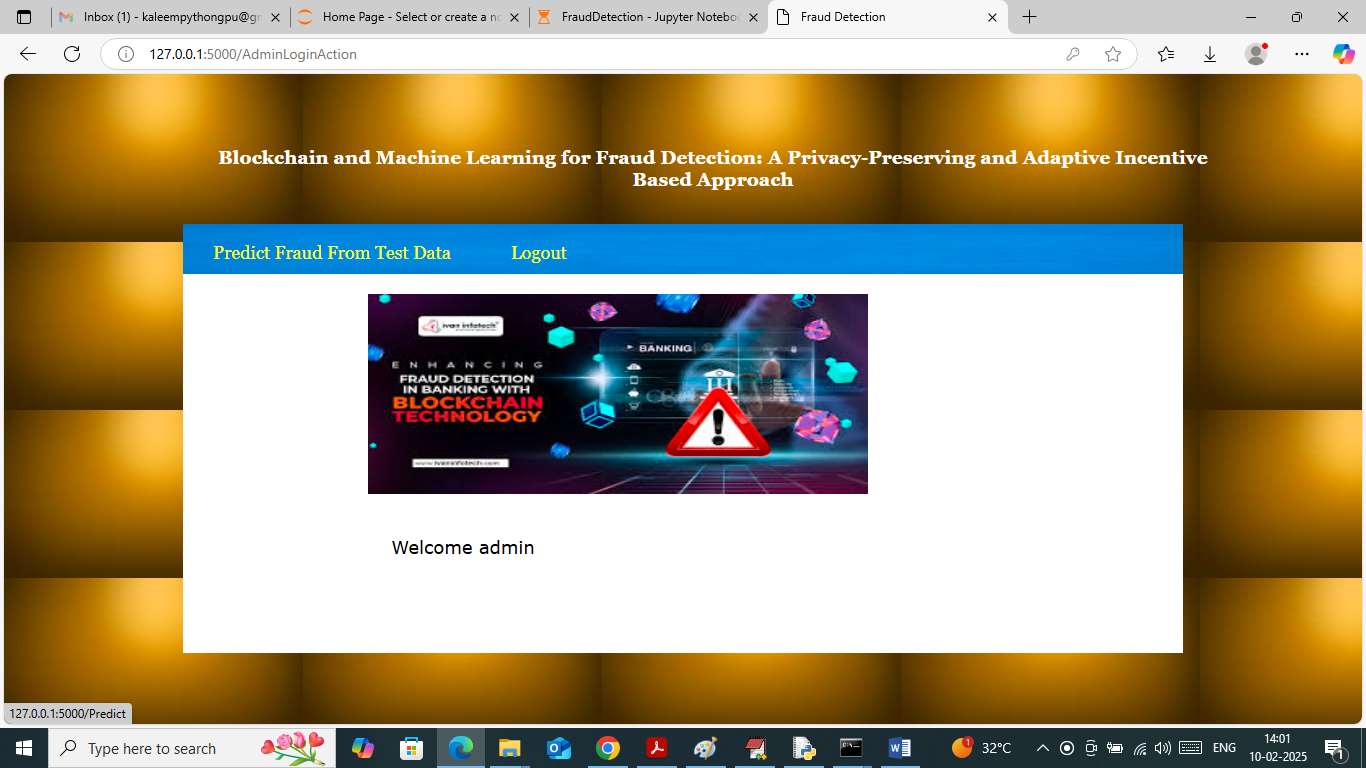
In above screen flask server started and now open browser and enter URL as <http://127.0.0.1:5000/index> and then press enter key to get below page



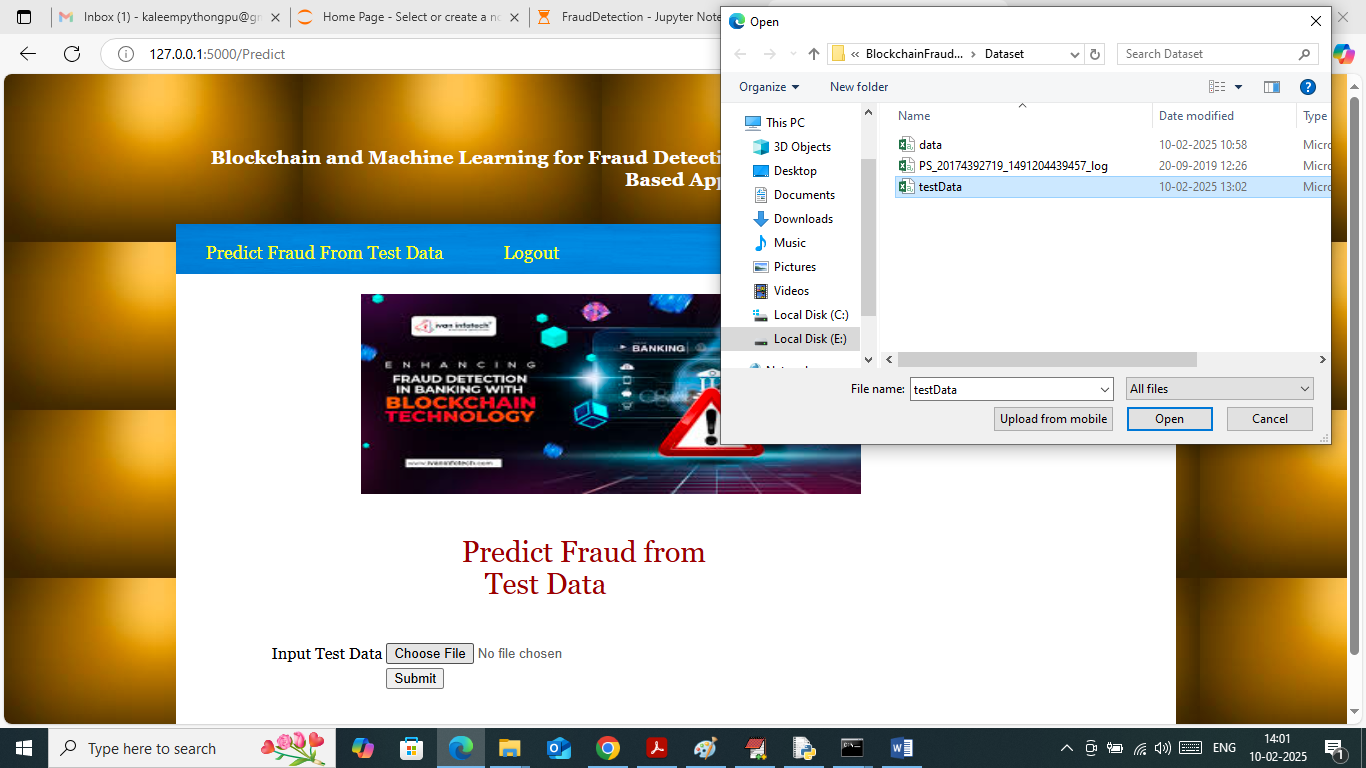
In above screen click on ‘Admin Login’ link to get below page



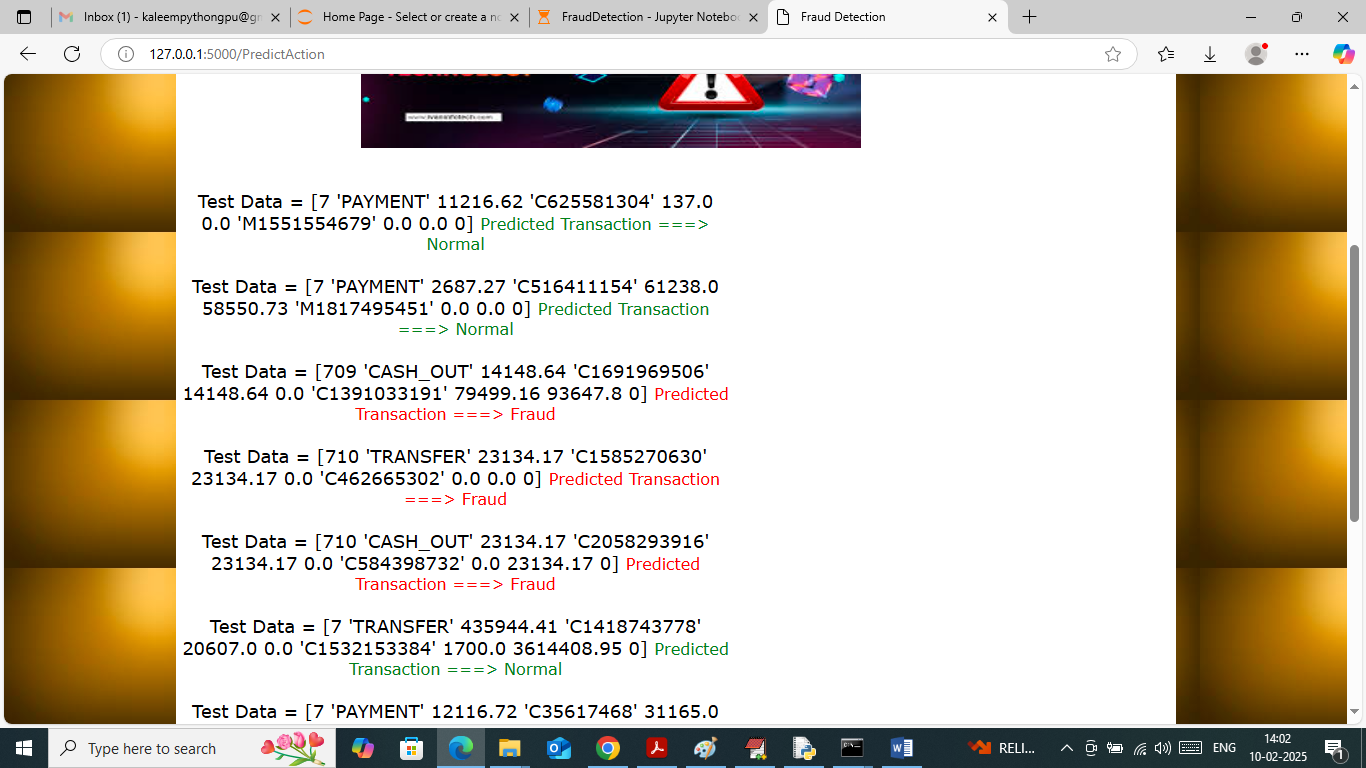
In above screen admin is login by using username and password as ‘admin and admin’ and after login will get below page



In above screen click on ‘Predict Fraud from Test Data’ link to get below page



In above screen admin is selecting and uploading test transaction data and then click on ‘Open and submit’ button to get below page



In above screen in square brackets admin can see Test data values and after =🡺 arrow symbol can see predicted values as Normal or Fraud for given test data.