Prediction of Hypertension using Machine Learning

Unhealthy lifestyle and growing competition leading to stress which will directly put effect to blood pressure and this abnormal BP causes various diseases such as Heart attack, kidney damage and many more. So its mandatory to monitor BP level and based on abnormal reading necessary precautions must be taken to control BP.

In market many BP measuring devices are available but it will display only reading and will not mention various risk levels such as Low, high, normal etc. So in propose paper author utilizing Machine Learning Decision tree algorithm to predict one of various level based on current BP values. Showing current BP risk level can help person in easily understanding and can take precaution to control BP.

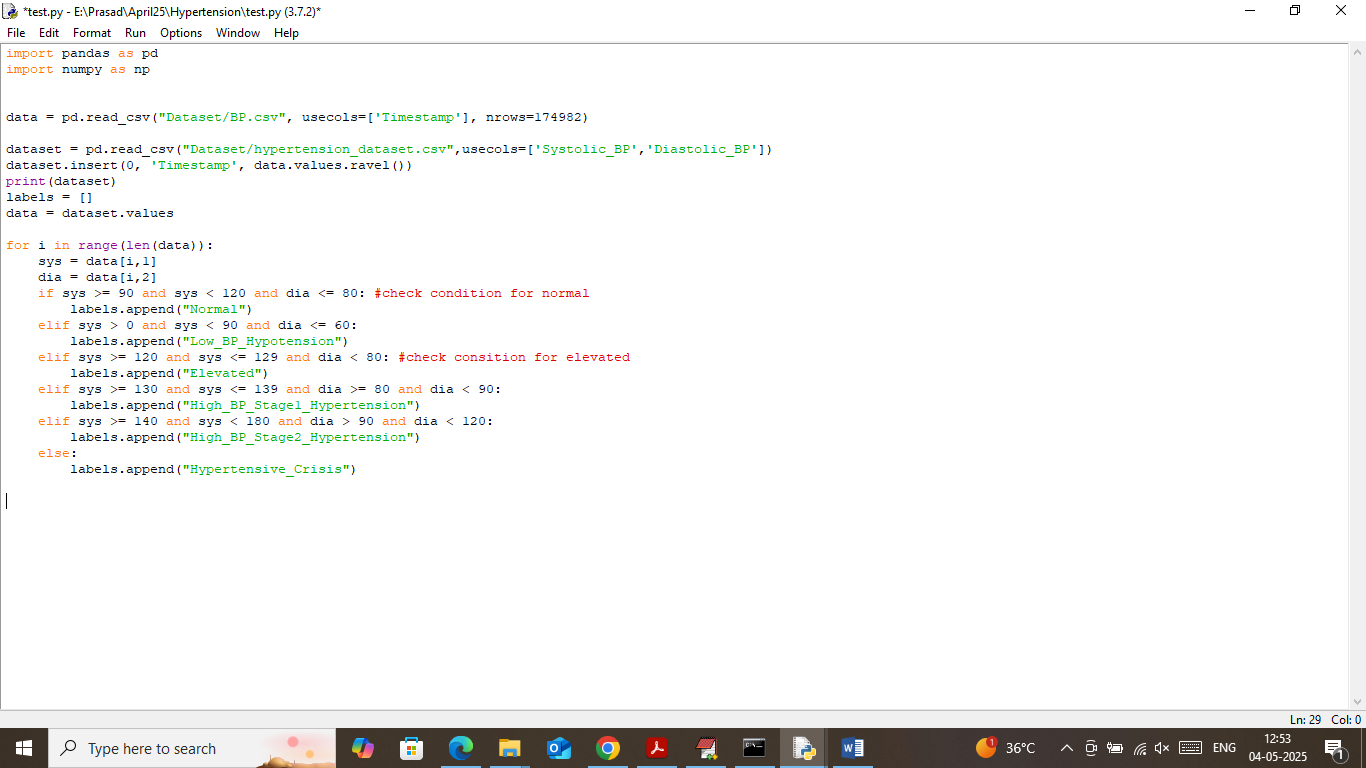
Propose algorithm can predict various risk levels such as Normal, Elevated, Low-BP-Hypotension, High-BP-Stage1-Hypertension, High-BP-Stage2-Hypertension and Hypertensive-Crisis.

This system can continuously read patient BP level and call ML module to predict risk level and then report predicted risk to medical professionals using Email messaging system.

To train ML algorithms we have used BP risk level dataset which can be download from below URL

<https://www.kaggle.com/datasets/nasirayub2/human-vital-sign-dataset>

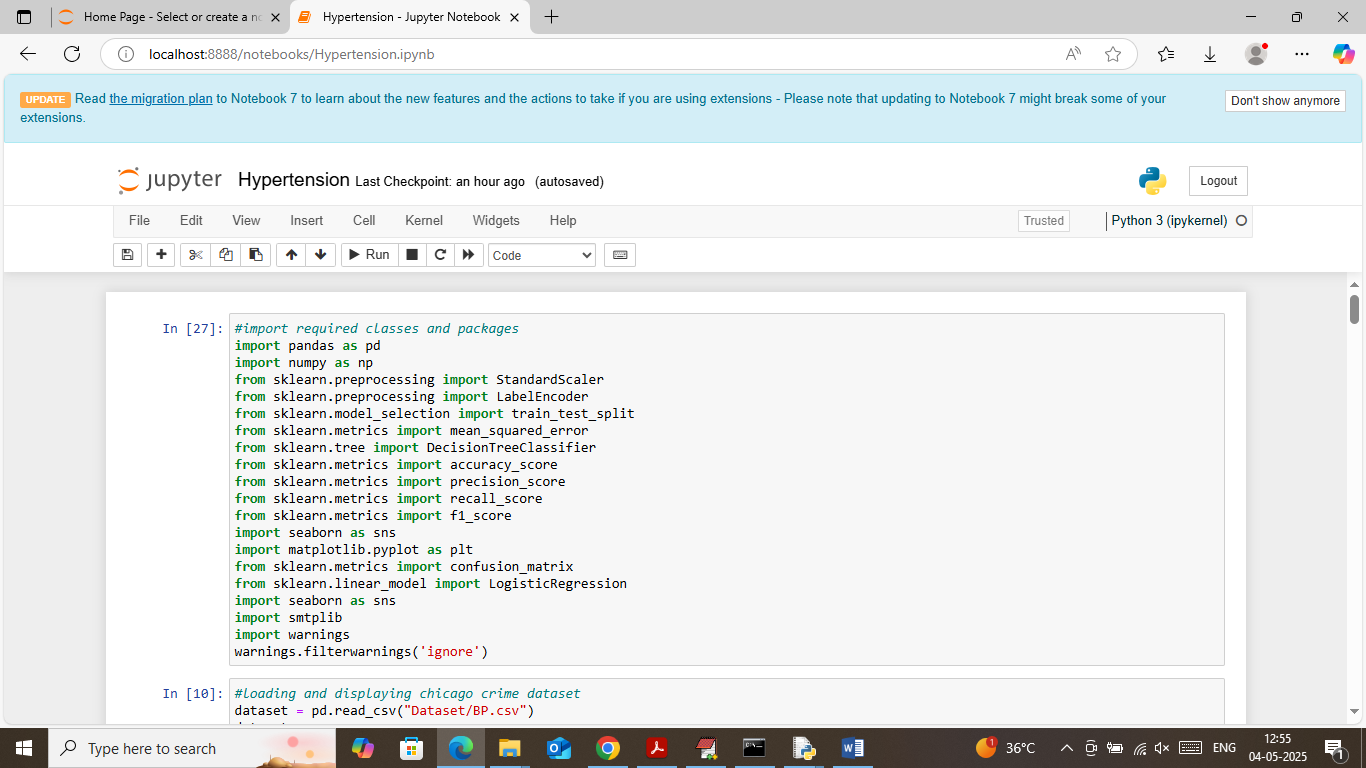
Above dataset reading is utilize to calculate various risk levels and this calculation showing in below screen



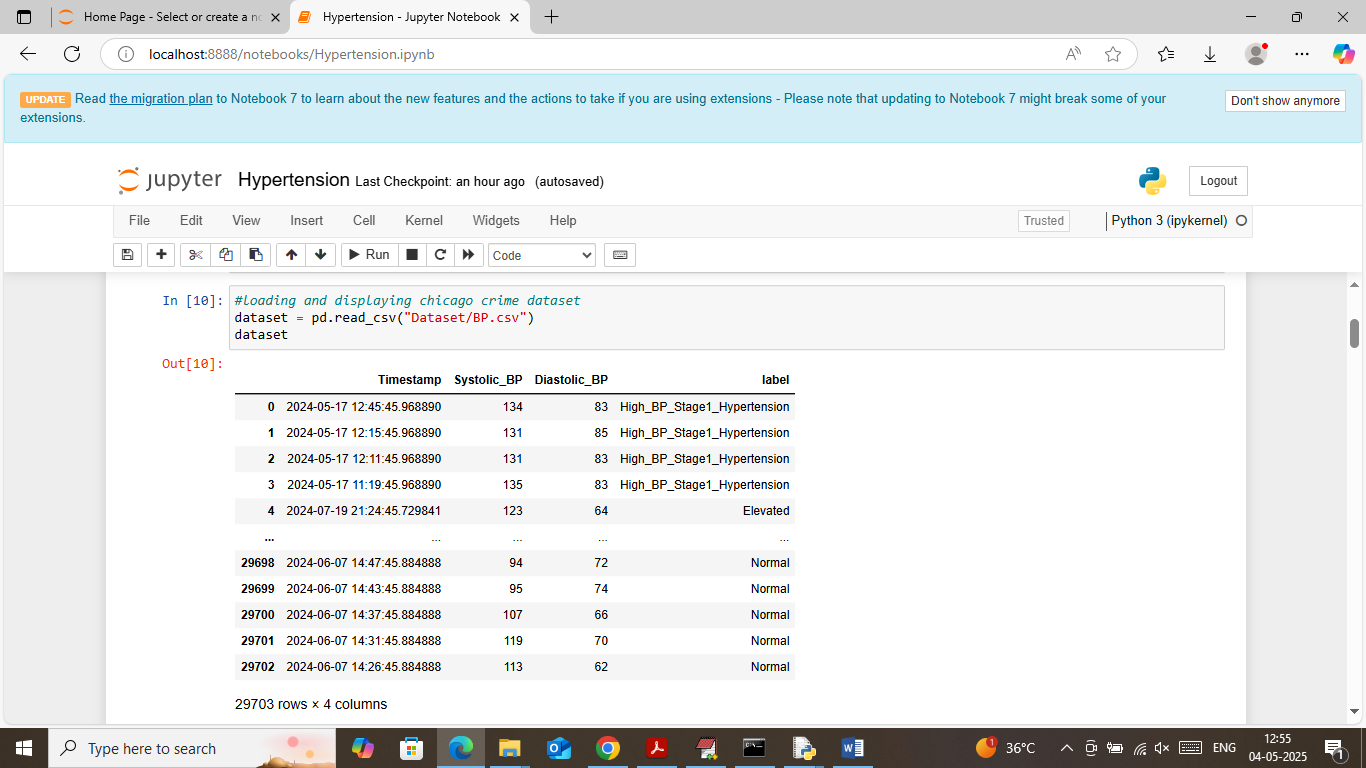
In above screen read red colour comments with condition to calculate risk levels and this levels will be input to Decision tree as prediction label.

SCREEN SHOTS

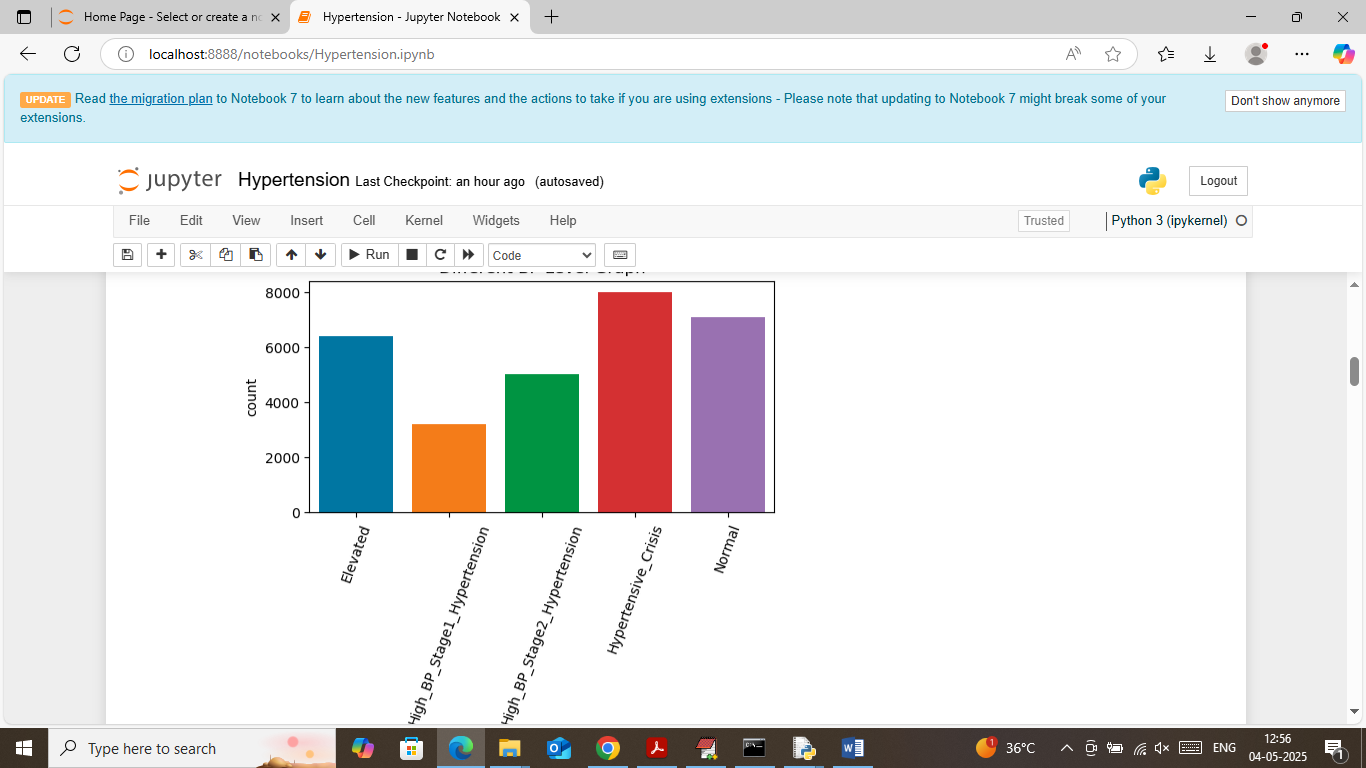
We have coded this project using JUPYTER notebook and below are the code and output screens with blue colour comments



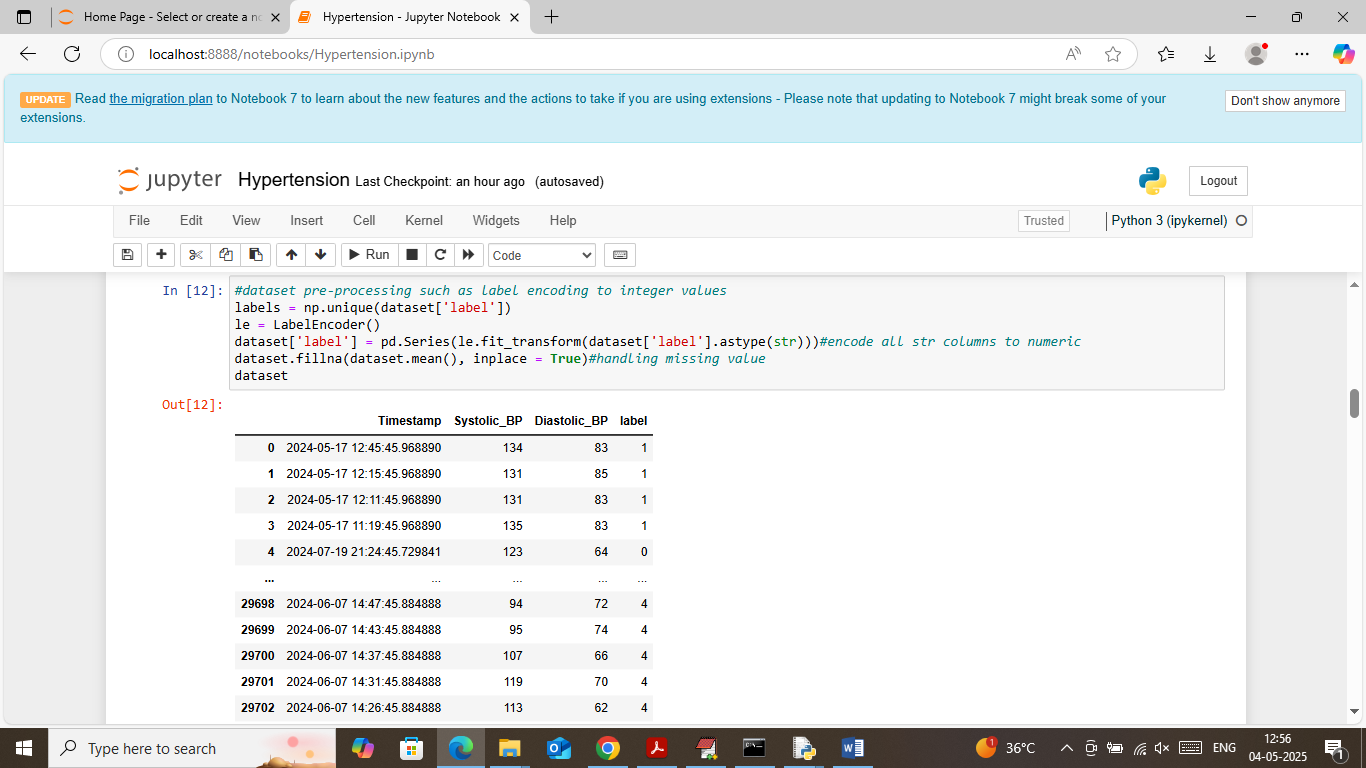
In above screen importing required python classes and packages



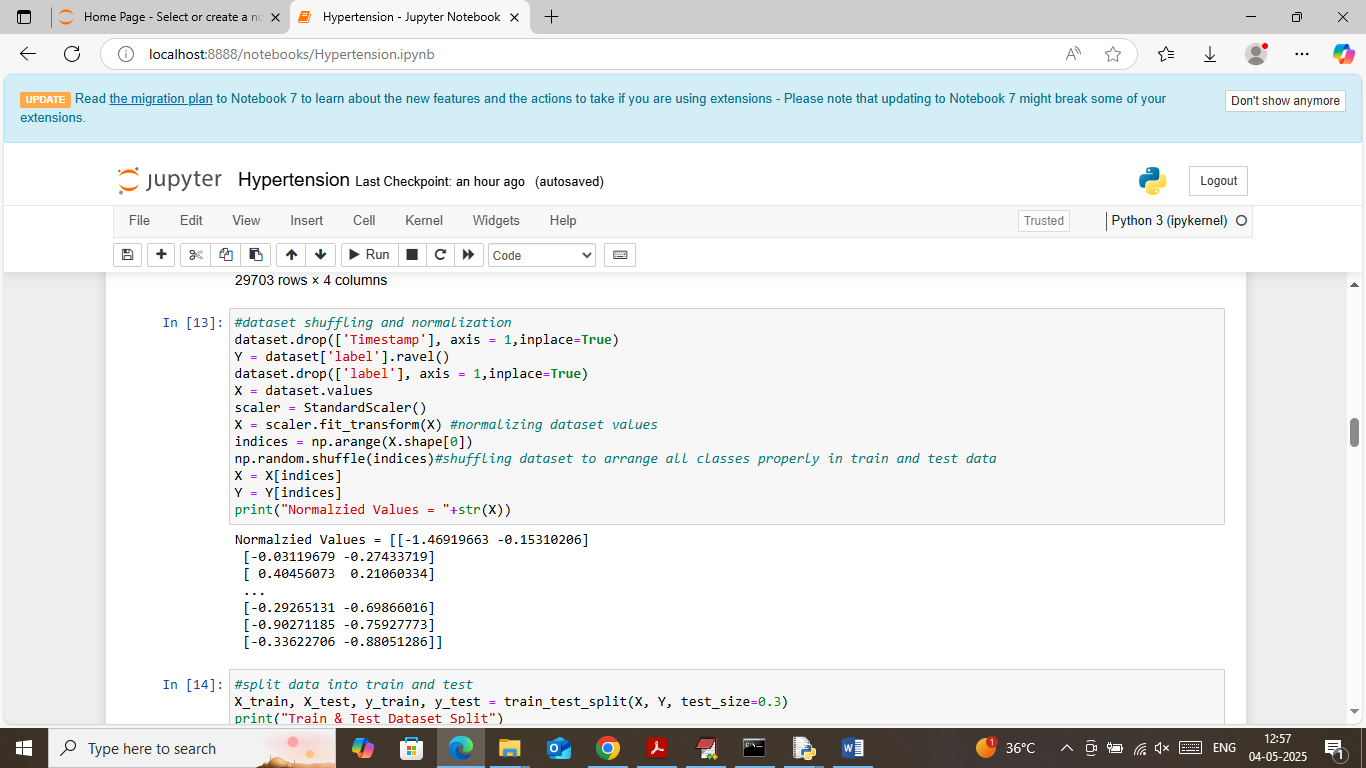
In above screen loading and displaying BP values dataset



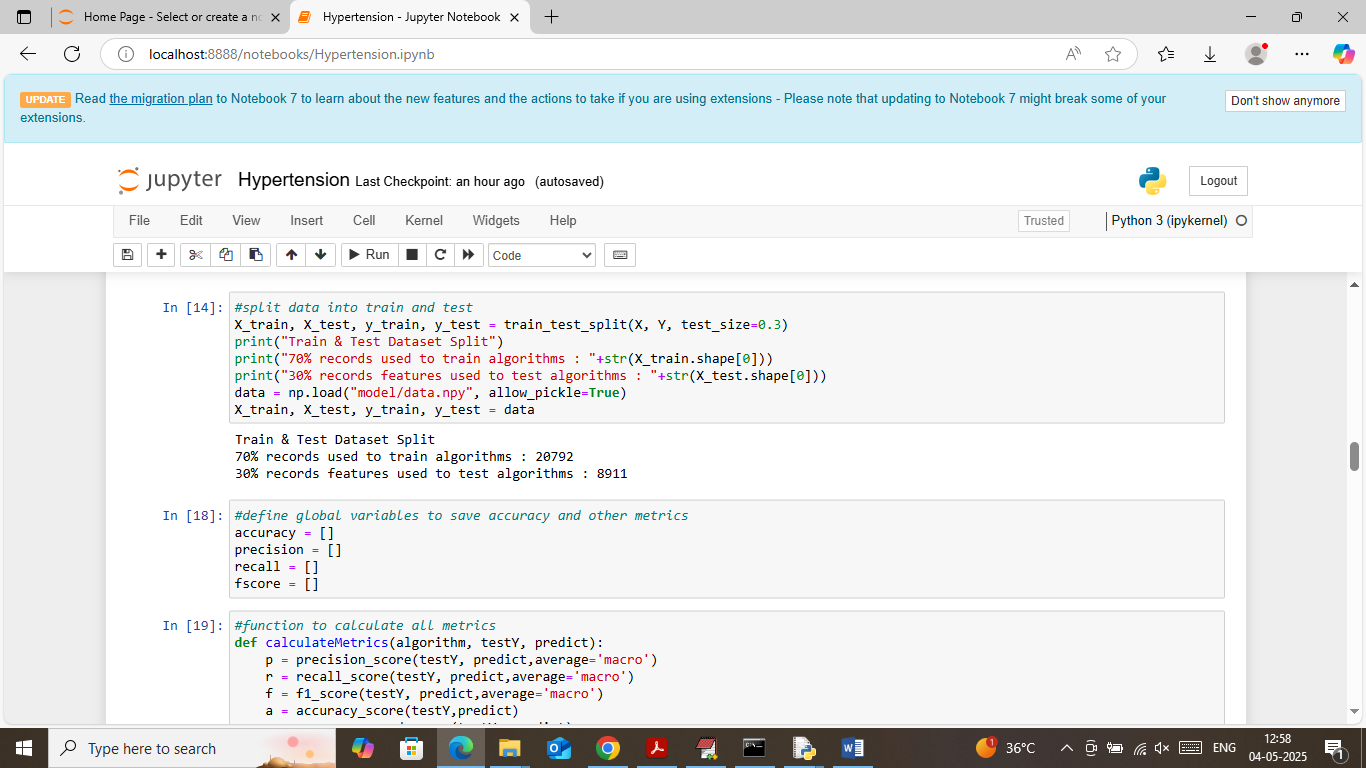
In above screen visualizing graph of various risk level available in dataset where x-axis represents risk type and y-axis represents count



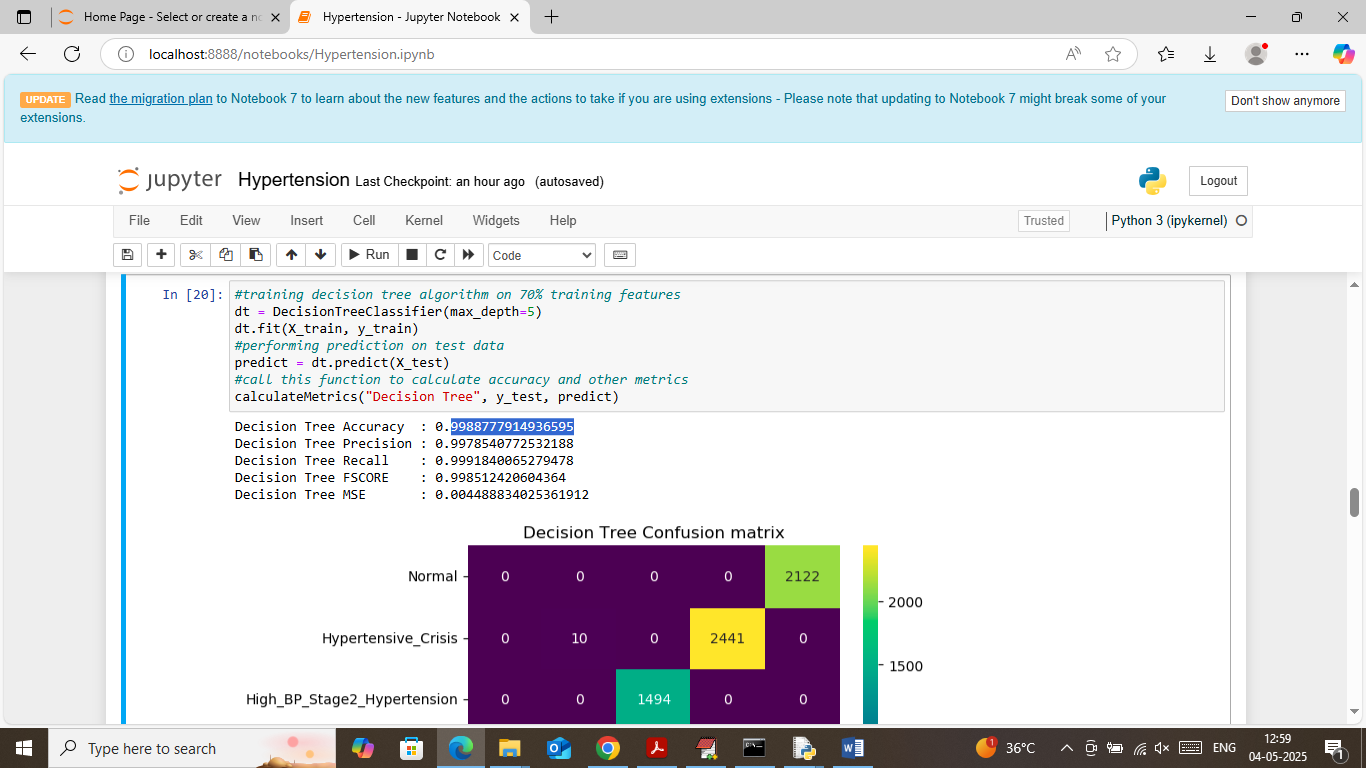
In above screen applying label encoding class to convert all non-numeric risk levels to numeric format and then displaying numeric converted class labels



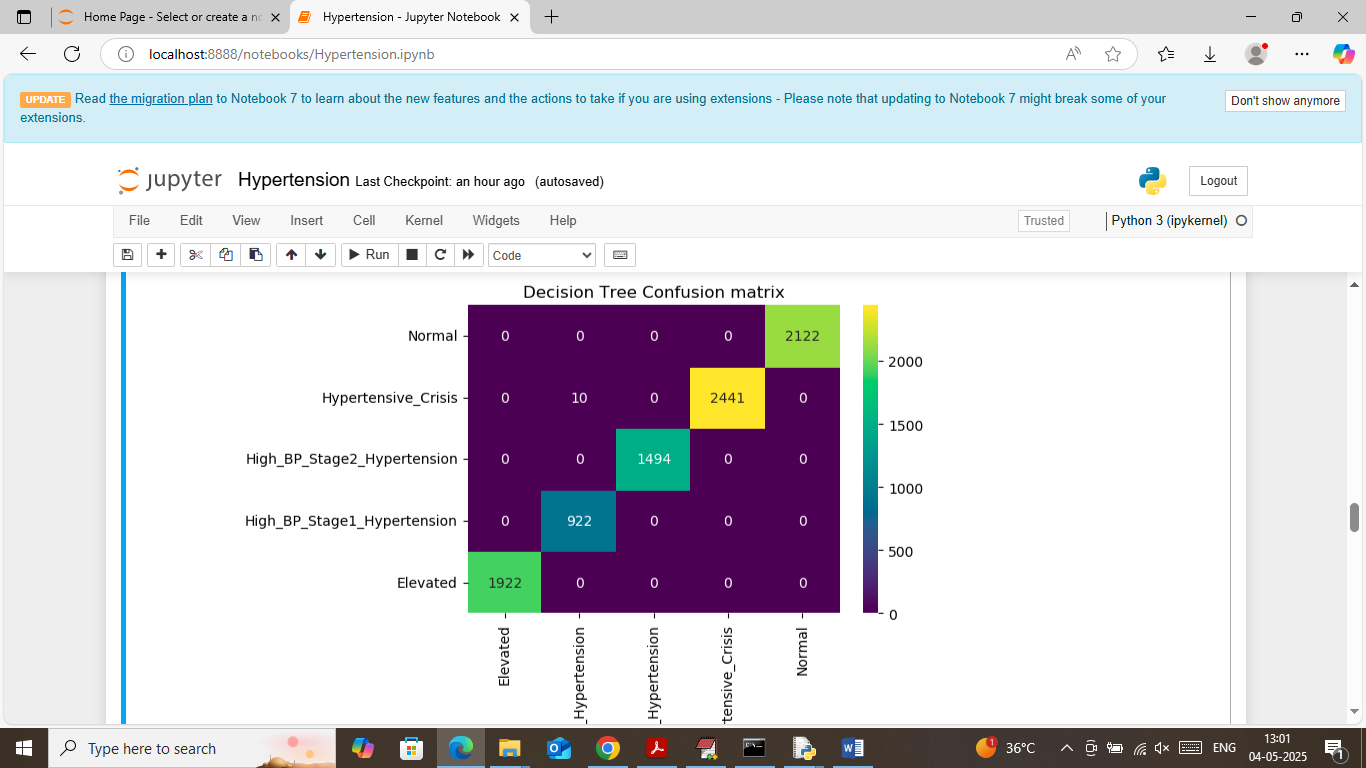
In above screen applying dataset pre-processing techniques such as shuffling and normalization and then displaying normalized values. Shuffling help in arranging all class labels properly in train and test data



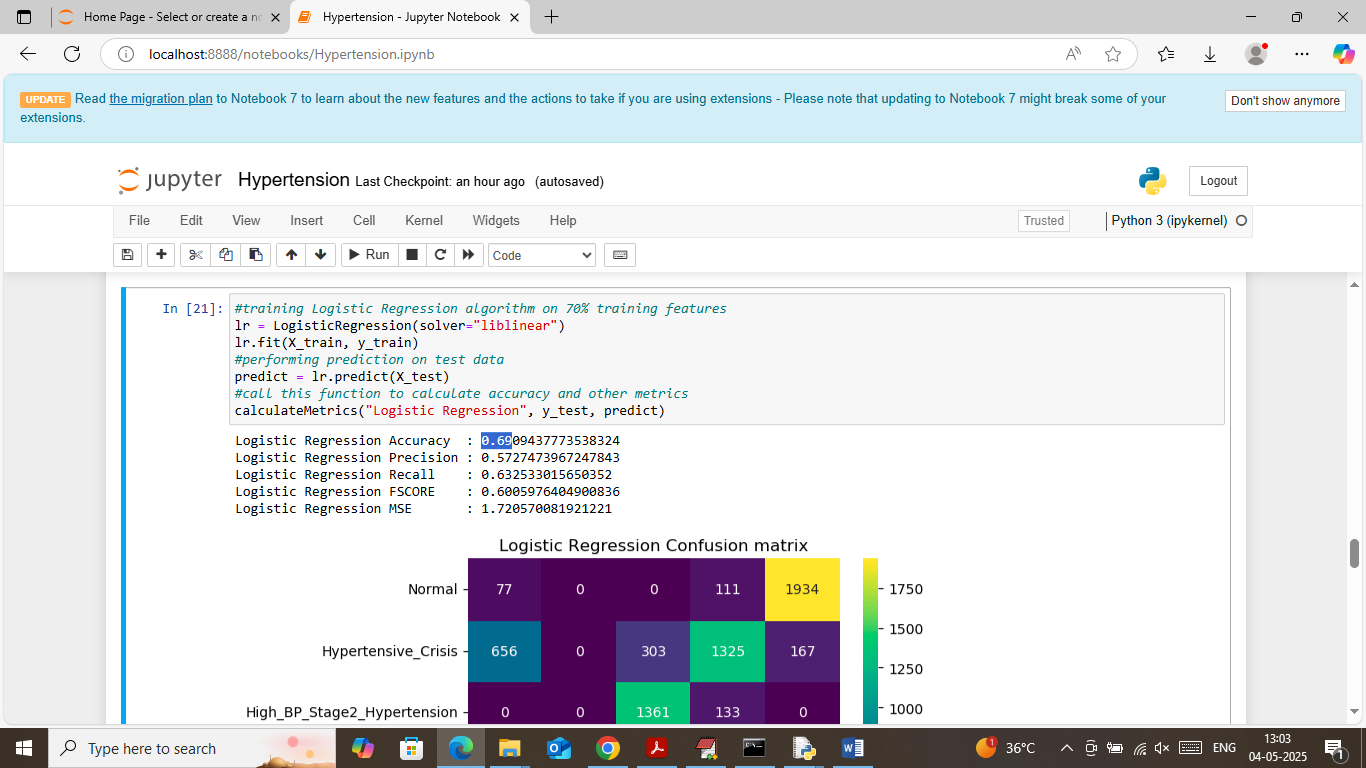
In above screen splitting dataset into train and test where application using 70% data for training and 30% for testing and then displaying train and test size records. In next block defining function to calculate accuracy and other metrics



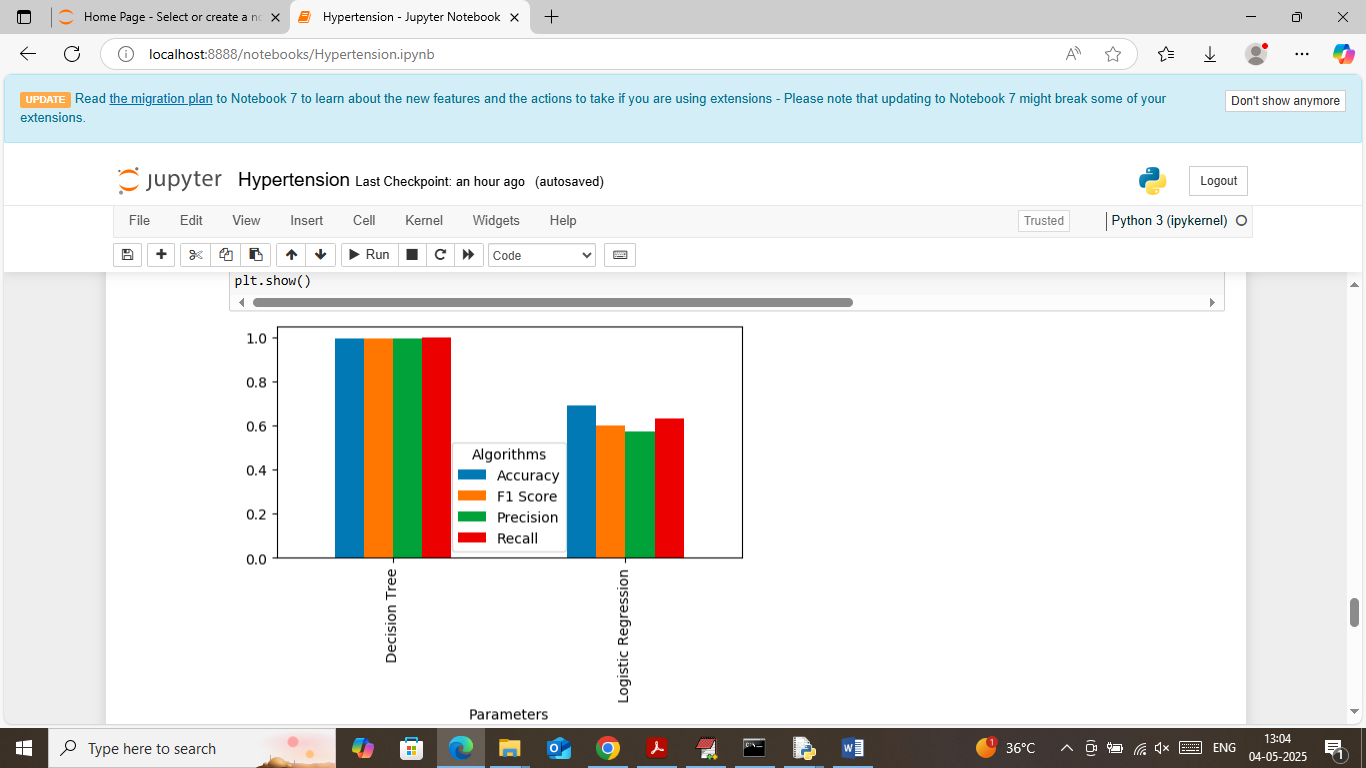
In above screen training decision tree algorithm on train features and then performing prediction on test data and then decision tree got 99% accuracy and MSE error as 0.0044% only and can see other metrics like precision, recall and FSCORE. In below screen showing confusion matrix graph



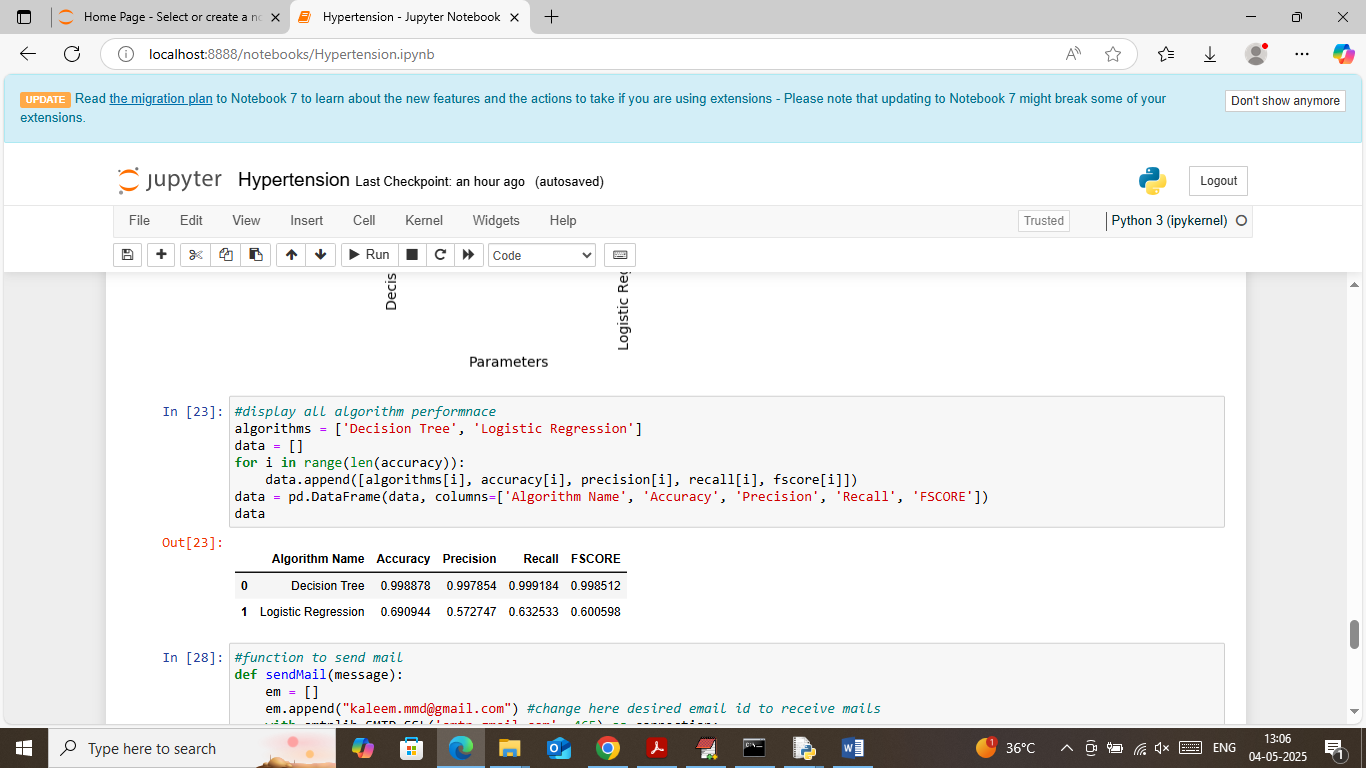
In above decision tree confusion matrix graph x-axis represents predicted Risk level and y-axis represents True Labels and then all different colour boxes in diagonal represents correct prediction count and remaining blue boxes represents incorrect prediction count which are very few



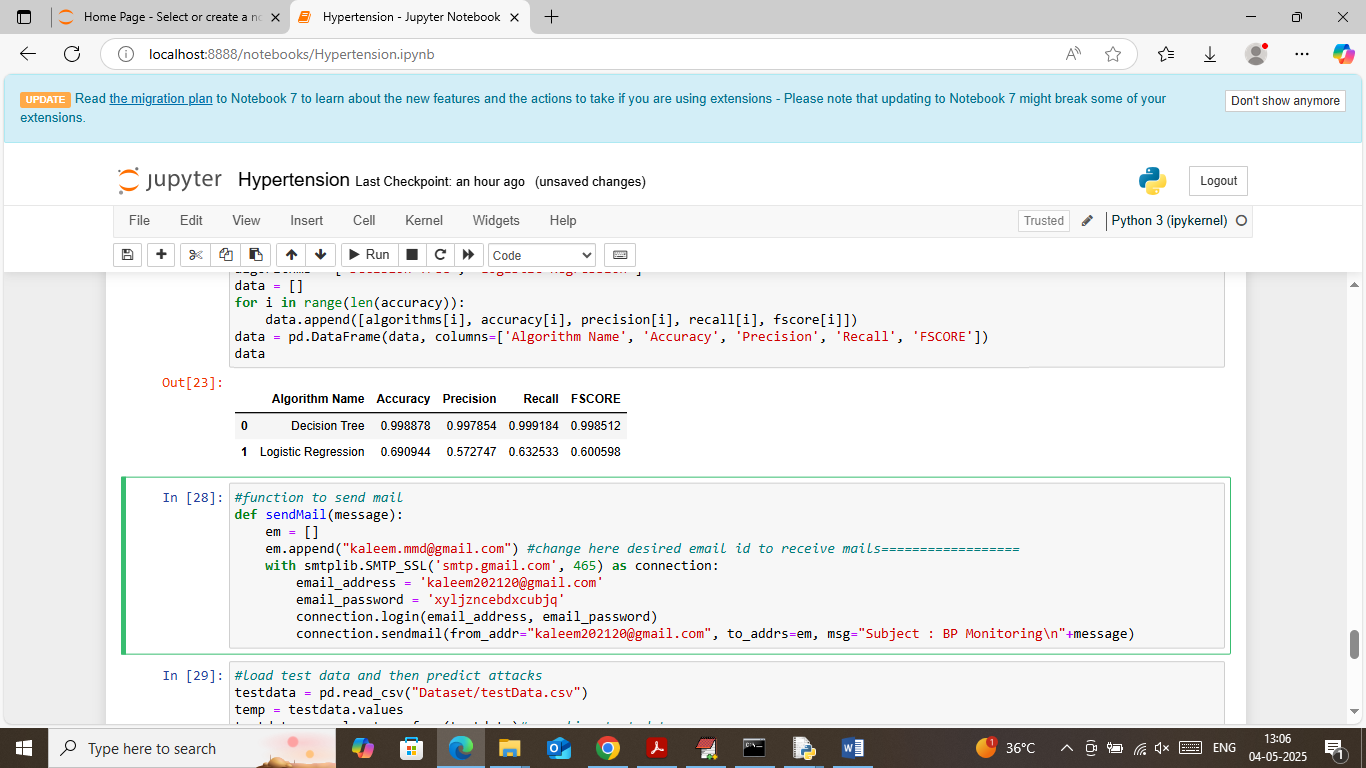
In above screen training existing Logistic Regression algorithm which got 69% accuracy and can see other metrics also



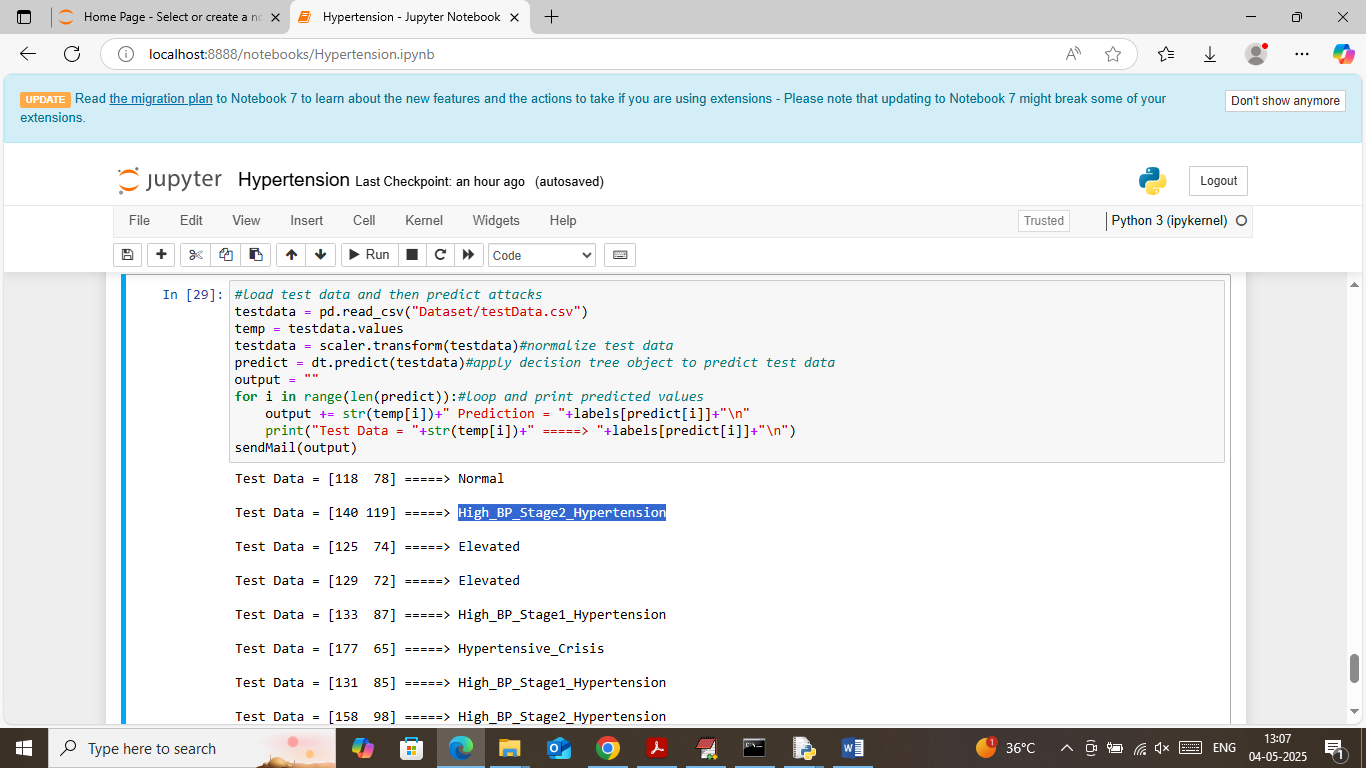
In above screen showing comparison graph between both algorithms where x-axis represents algorithm names and y-axis represents accuracy and other metrics and in both algorithms Decision tree got high performance



In above screen showing comparison table between both algorithms



In above screen defining function to send email alert and you can change email to your email id where I am showing “====================” comments



In above screen reading test data values and then applying decision tree object to predict risk level. In above output in square bracket can see BP test data values like systolic and diastolic and then after =🡺 arrow symbol can see predicted risk level.

After executing above test data will get below output in email

