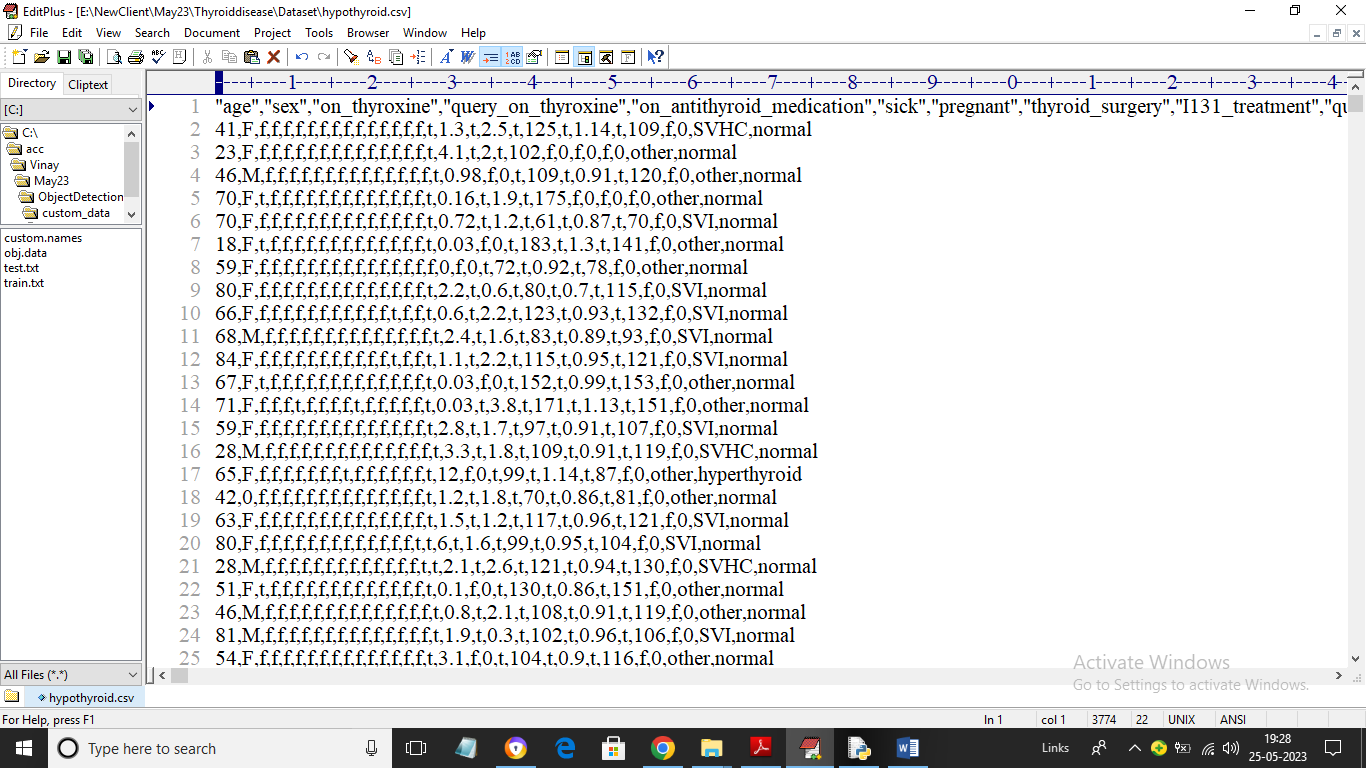
Thyroid Disease Classification Using Machine Learning Algorithms

In this paper author employing various machine learning algorithms such as SVM, Naïve Bayes, Decision Tree, Random Forest, KNN and MLP to predict thyroid disease. Each algorithm get trained on thyroid dataset which contains 3 different classes such Normal, Hyperthyroid and Hypothyroid. All algorithms performance is evaluated in terms of accuracy, precision, recall, FSCORE and confusion matrix and in all algorithms Random Forest and Decision Tree giving best accuracy.

In below screen we are showing dataset to train above algorithms



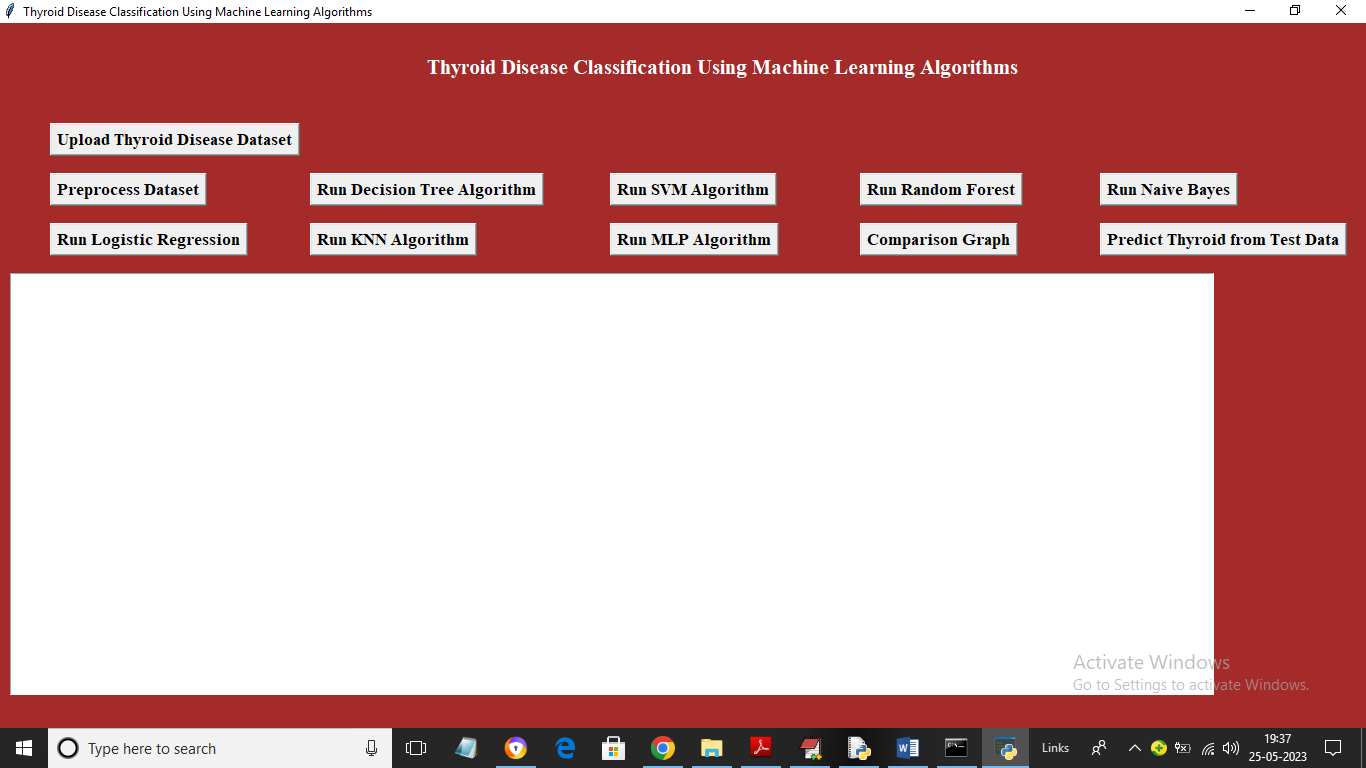
In above dataset screen first row contains dataset column names and remaining rows contains dataset values and in last column we have class labels called Normal, hyper and hypothyroid. Above dataset contains both numeric and non-numeric values but machine learning accept only numeric values so by employing Label Encoder class we can convert non-numeric data to numeric values. Label Encoder class assign integer ID to each unique non-numeric values.

To implement this project we have designed following modules

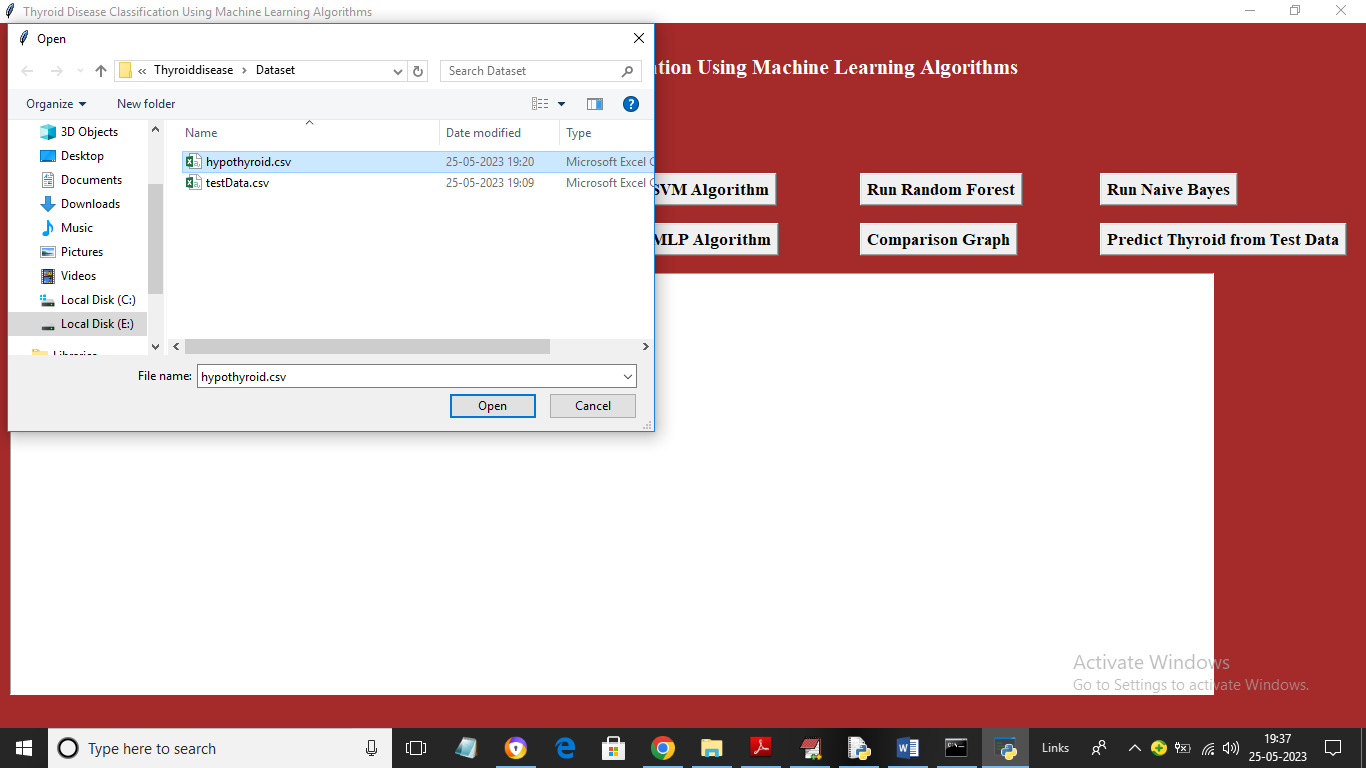
1. Upload Thyroid Disease Dataset: using this module we will upload dataset to application and then find and plot graph of different class labels or thyroid disease instances found in the dataset.
2. Preprocess Dataset: using this module we will Preprocess dataset such as converting non-numeric data to numeric values, replacing missing values, normalizing and shuffling dataset values and then split dataset into train and test where application using 80% dataset to train algorithm and use 20% test data to calculate algorithm prediction capability
3. Run Decision Tree Algorithm: 80% processed data will be input to decision tree algorithm to train a model and this model will be applied on 20% test data to calculate algorithm accuracy and other metrics
4. Run SVM Algorithm: 80% processed data will be input to SVM algorithm to train a model and this model will be applied on 20% test data to calculate algorithm accuracy and other metrics
5. Run Random Forest: 80% processed data will be input to Random Forest algorithm to train a model and this model will be applied on 20% test data to calculate algorithm accuracy and other metrics
6. Run Naive Bayes: 80% processed data will be input to Naïve Bayes algorithm to train a model and this model will be applied on 20% test data to calculate algorithm accuracy and other metrics
7. Run Logistic Regression: 80% processed data will be input to Logistic Regression algorithm to train a model and this model will be applied on 20% test data to calculate algorithm accuracy and other metrics
8. Run KNN Algorithm: 80% processed data will be input to KNN algorithm to train a model and this model will be applied on 20% test data to calculate algorithm accuracy and other metrics
9. Run MLP Algorithm: 80% processed data will be input to MLP algorithm to train a model and this model will be applied on 20% test data to calculate algorithm accuracy and other metrics
10. Comparison Graph: using this module we will plot comparison graph between all algorithms
11. Predict Thyroid from Test Data: using this module we will upload test data and then algorithm will predict weather test data is normal or contains thyroid

SCREEN SHOTS

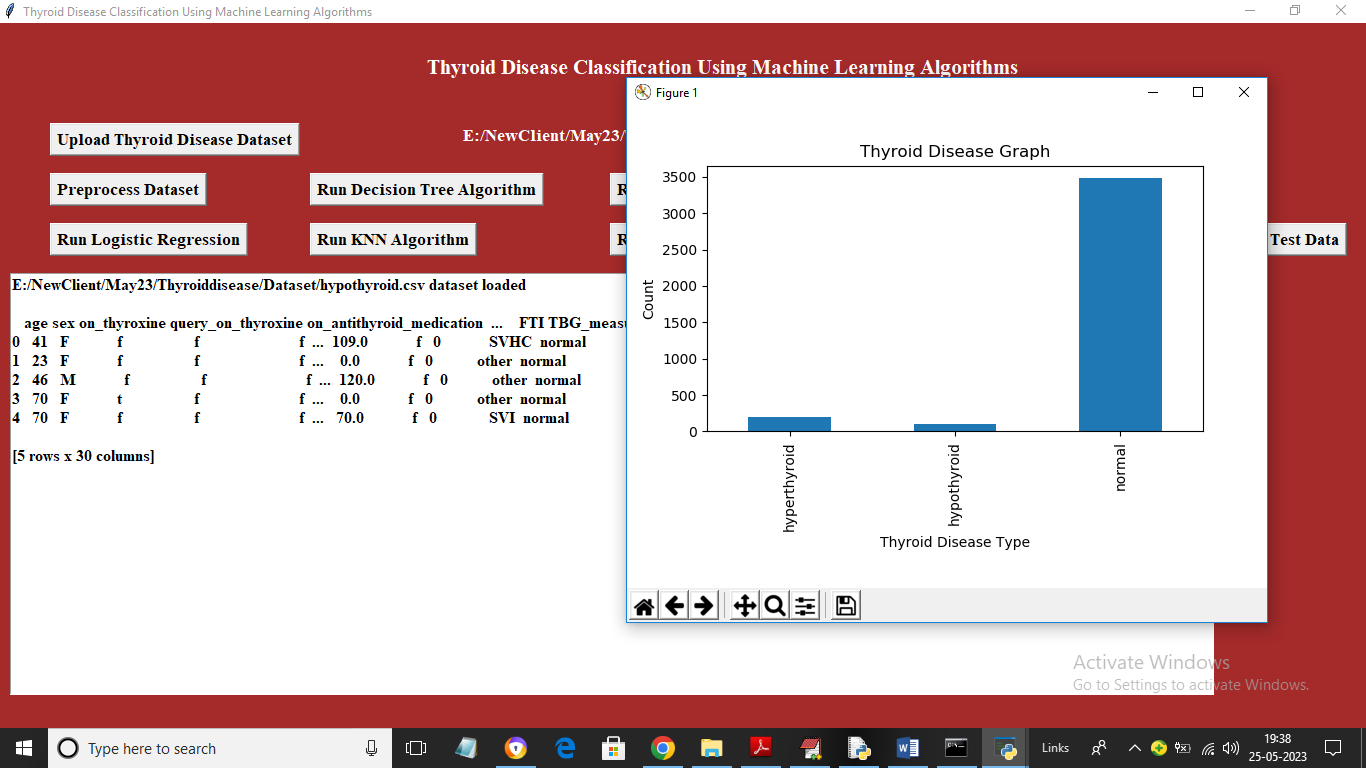
To run project double click on ‘run.bat’ file to get below screen



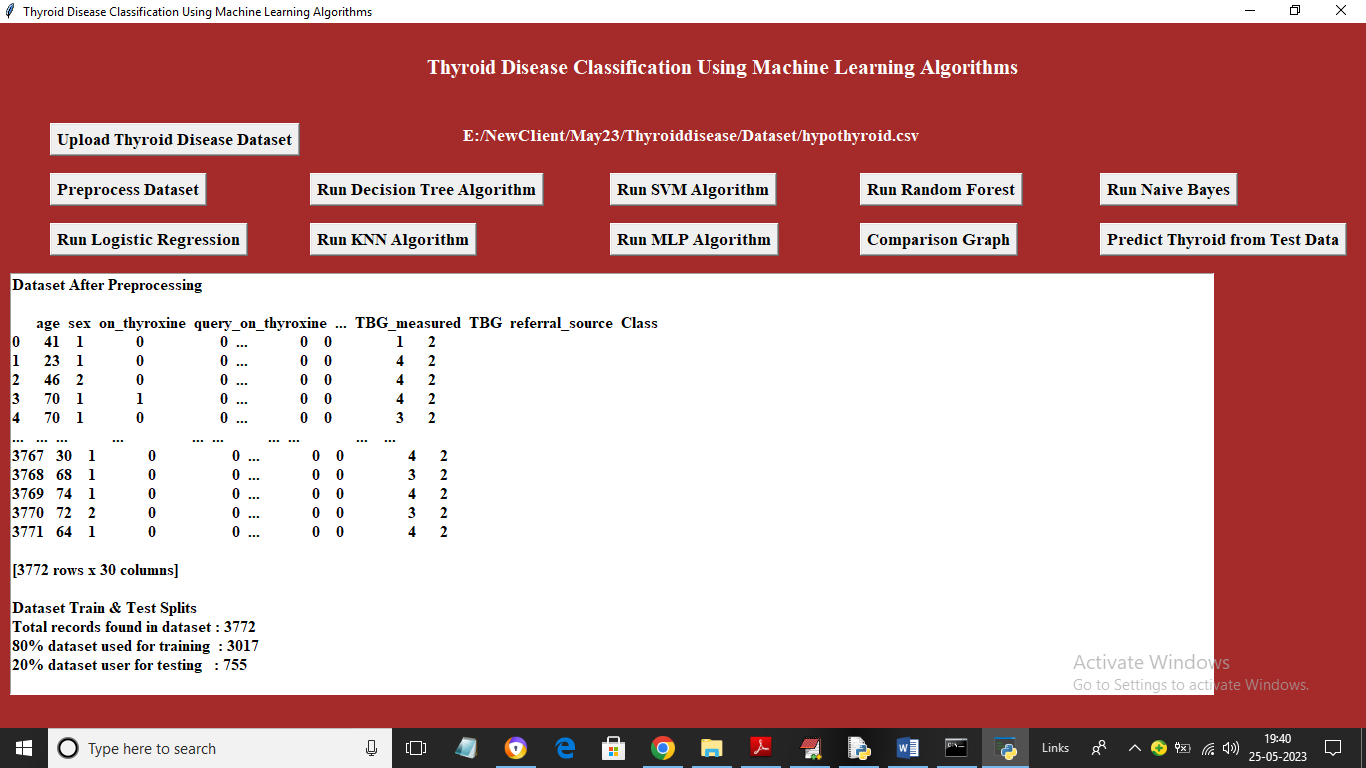
In above screen click on ‘Upload Thyroid Disease Dataset’ button to upload dataset and get below screen



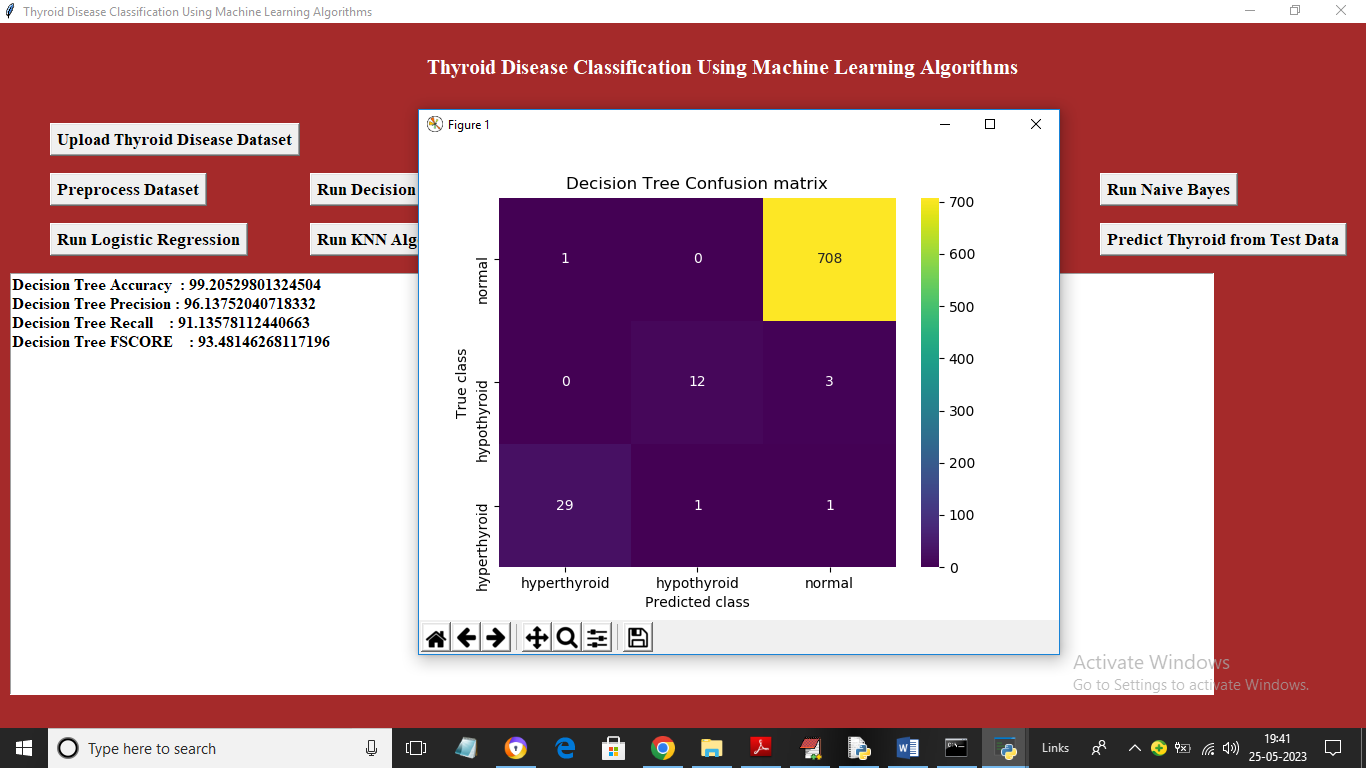
In above screen selecting and uploading dataset file and then click on ‘Open’ button to load dataset and get below output



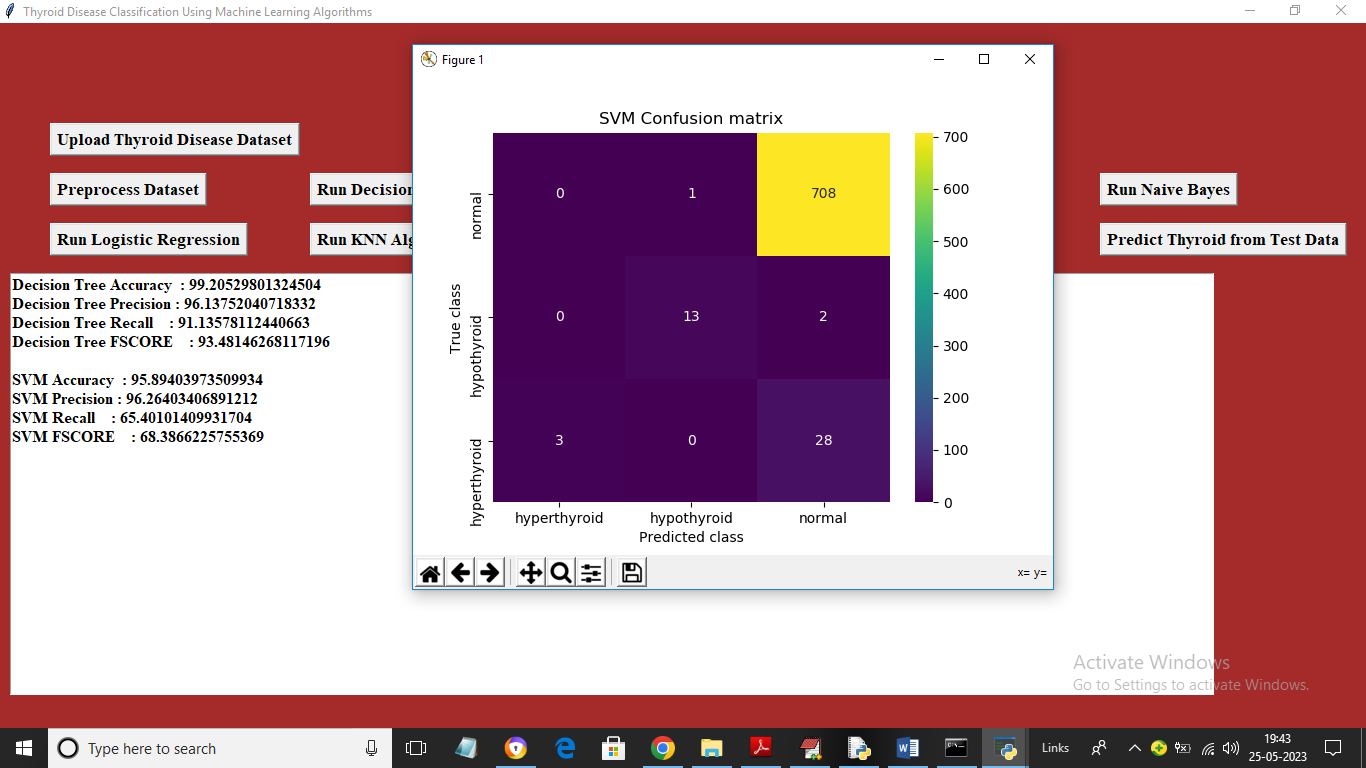
In above screen dataset loaded and we can see dataset contains some numeric and some non-numeric values so by applying Processing technique we can clean above data. In above graph x-axis represents thyroid disease type and y-axis represents count of each disease. Now close above graph and then click on ‘Preprocess Dataset’ button to process dataset and get below output



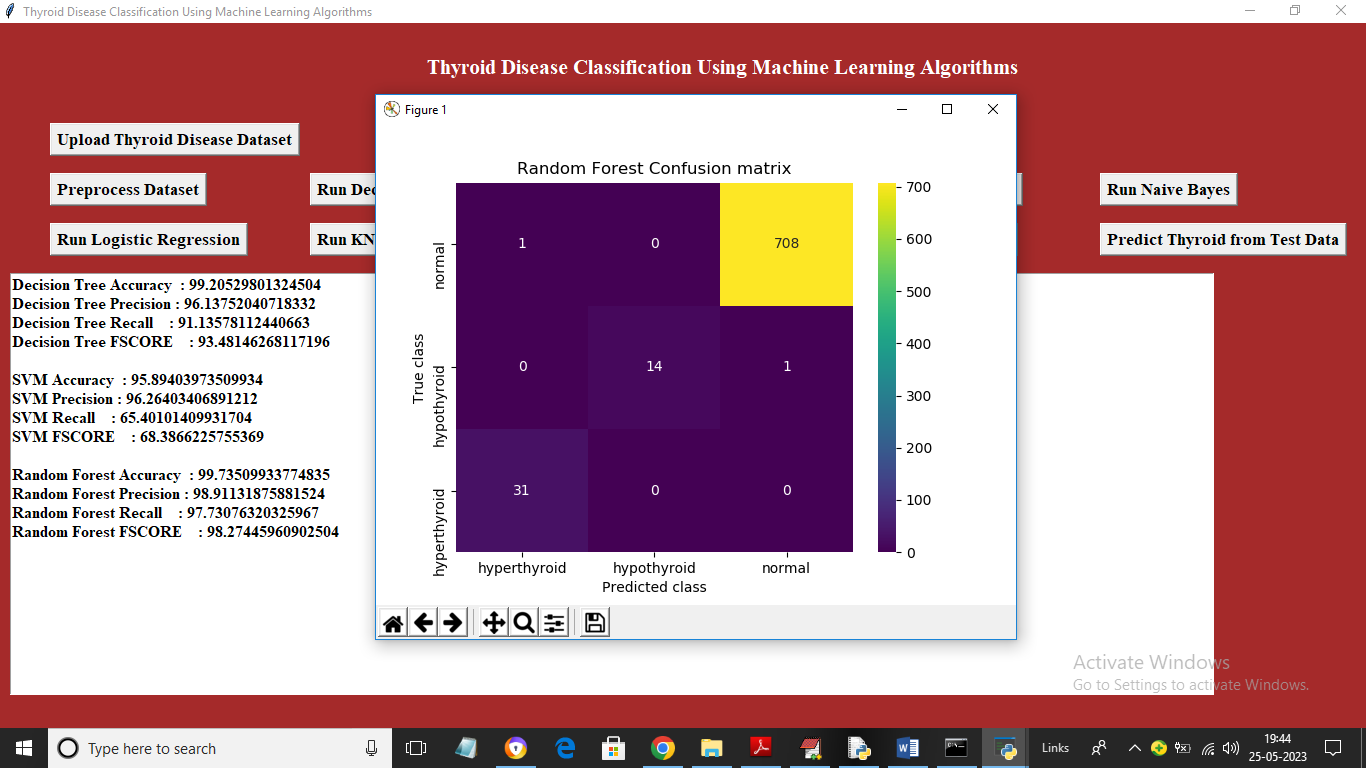
In above screen we can see entire dataset is now converted to numeric values and in last line we can see size of dataset and train and test split details. Now click on ‘Run Decision Tree Algorithm’ button to train decision Tree and get below output



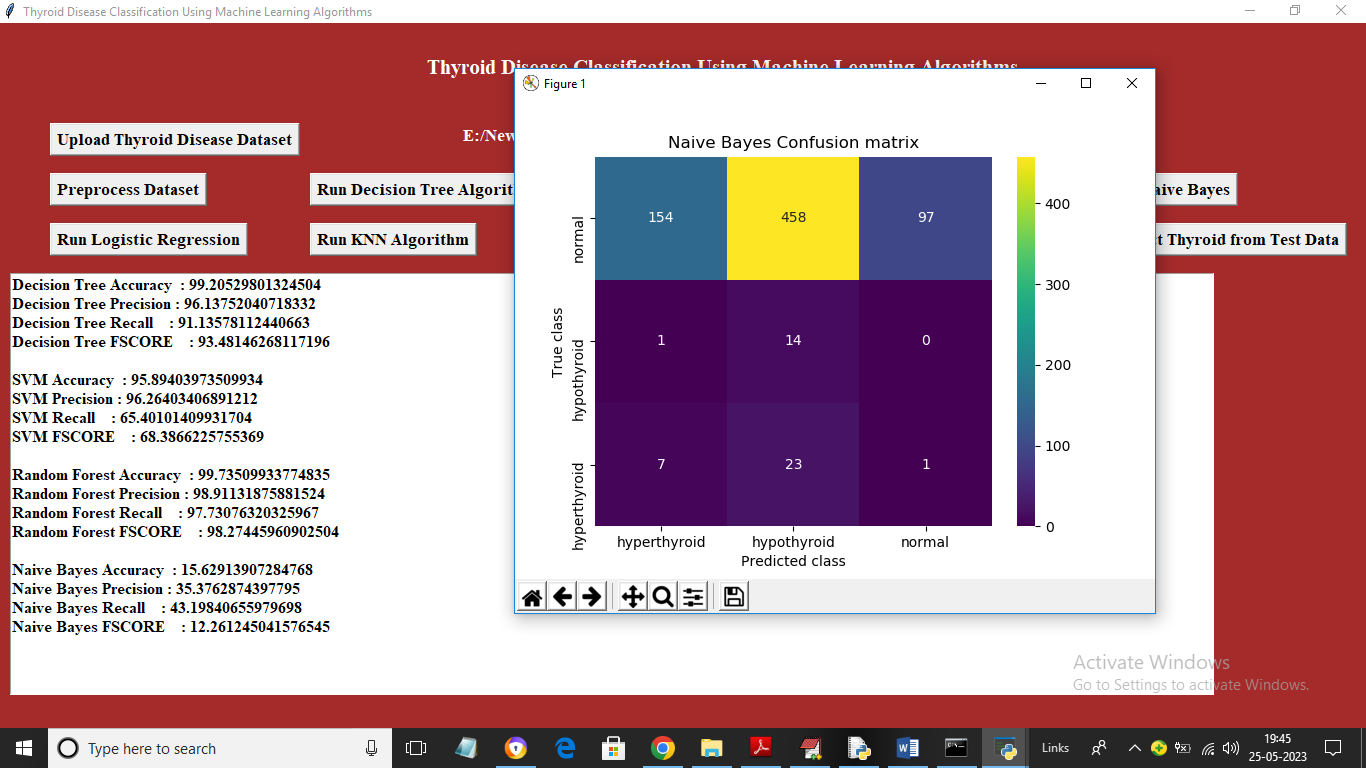
In above screen with decision tree we got accuracy as 99% and we can see other metrics also and in confusion matrix graph x-axis represents Predicted Labels and y-axis represents True Labels and all boxes in yellow colour box diagnol contains correct prediction count and remaining boxes contains incorrect prediction count which are very few. Now close above graph and then click on ‘Run SVM Algorithm’ button to get below output



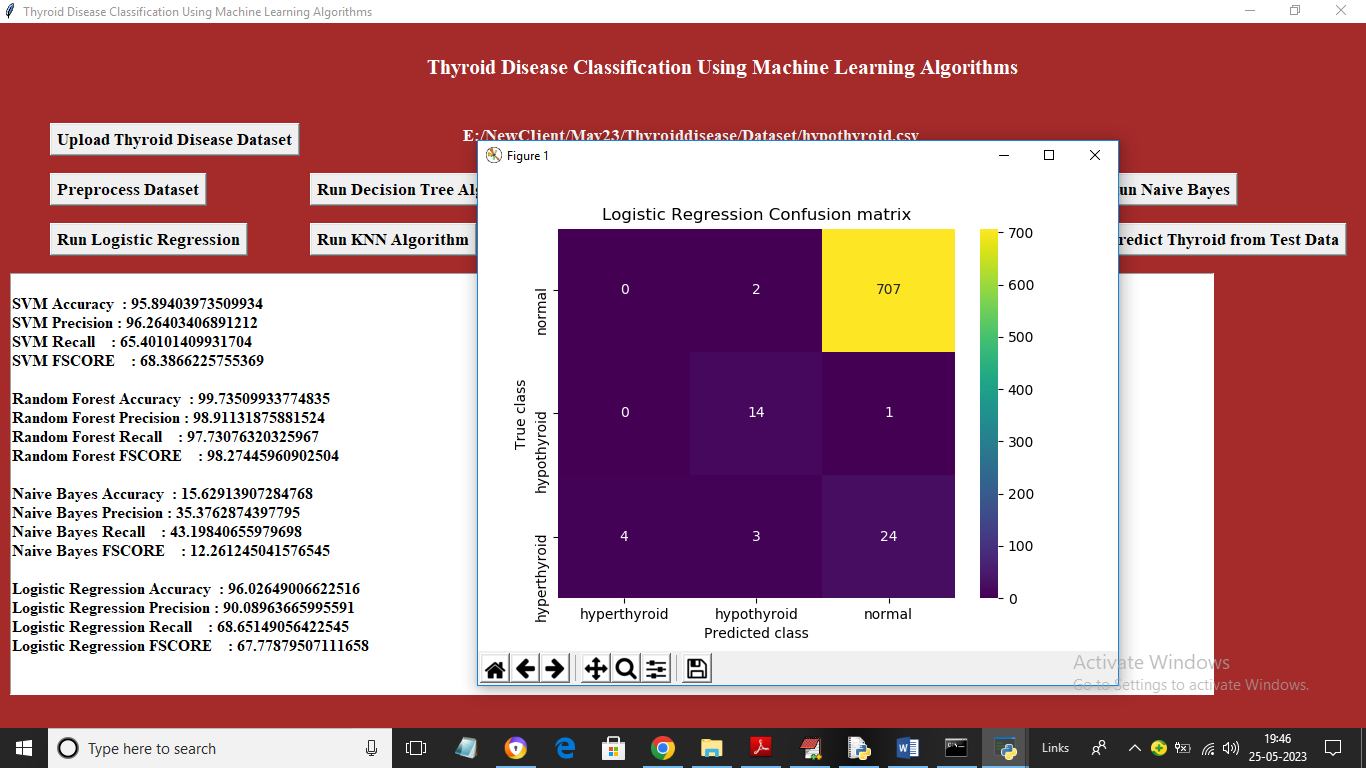
In above screen with SVM we got 95% accuracy and we can see other metrics and confusion matrix graph. Now click on ‘Run Random Forest Algorithm’ button to train Random Forest and get below output



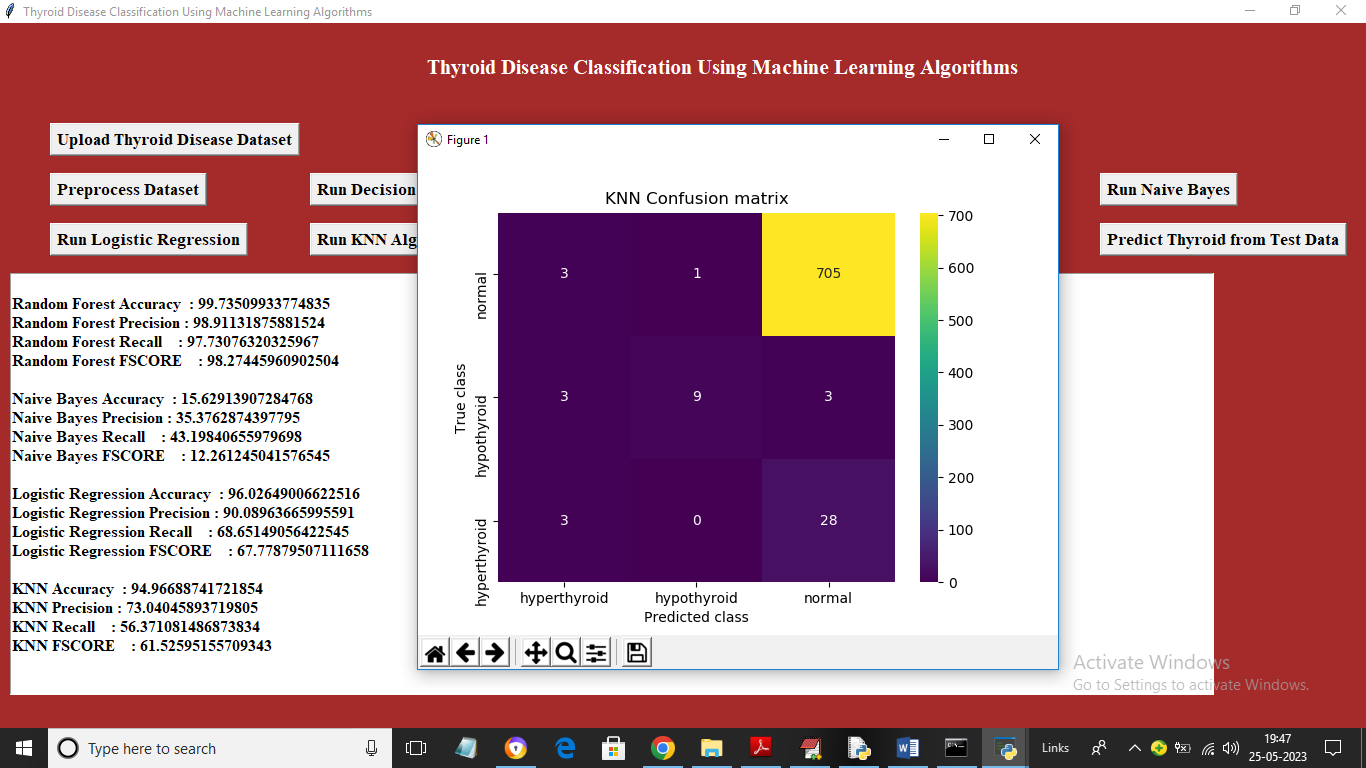
In above screen with Random Forest we got 99% accuracy and now click on ‘Run Naïve Bayes Algorithm’ button to get below output



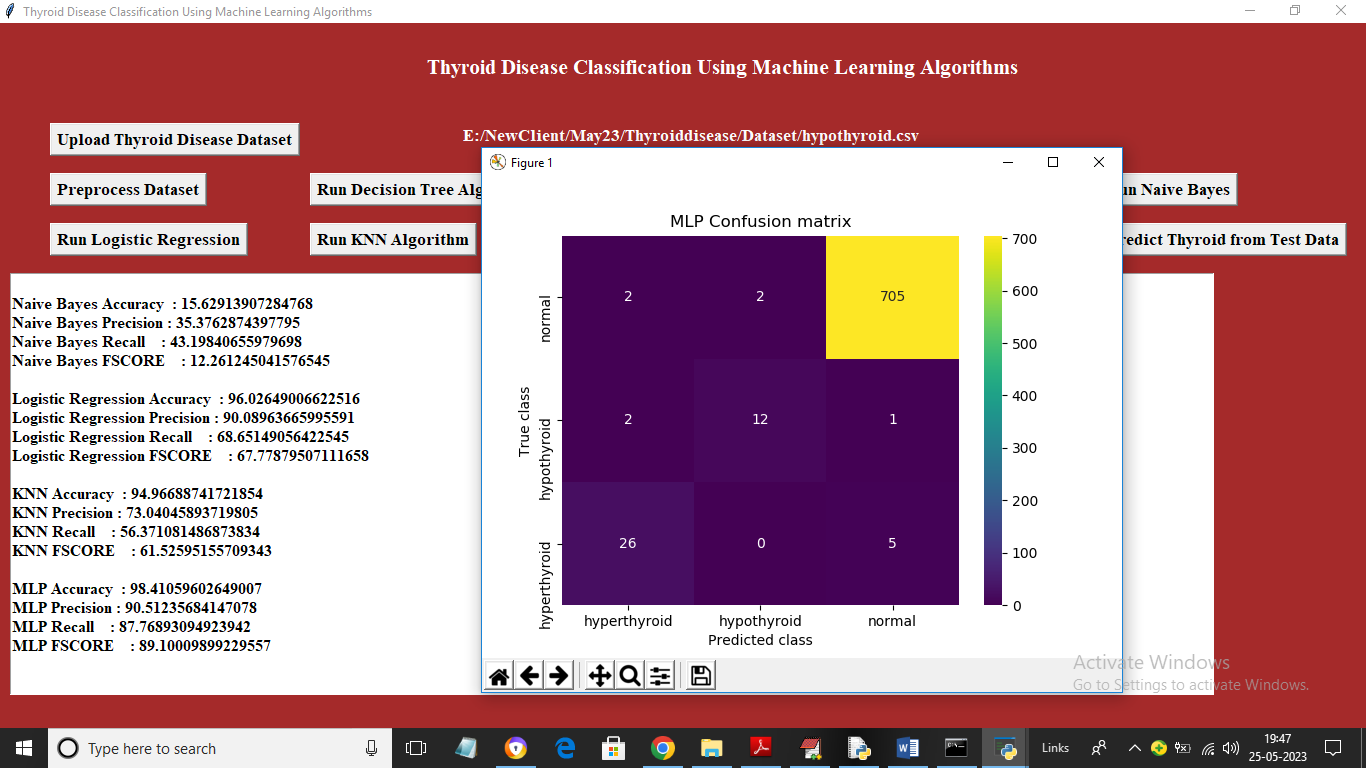
In above screen Random Forest got 15% accuracy and now click on ‘Run Logistic Regression’ button to get below output



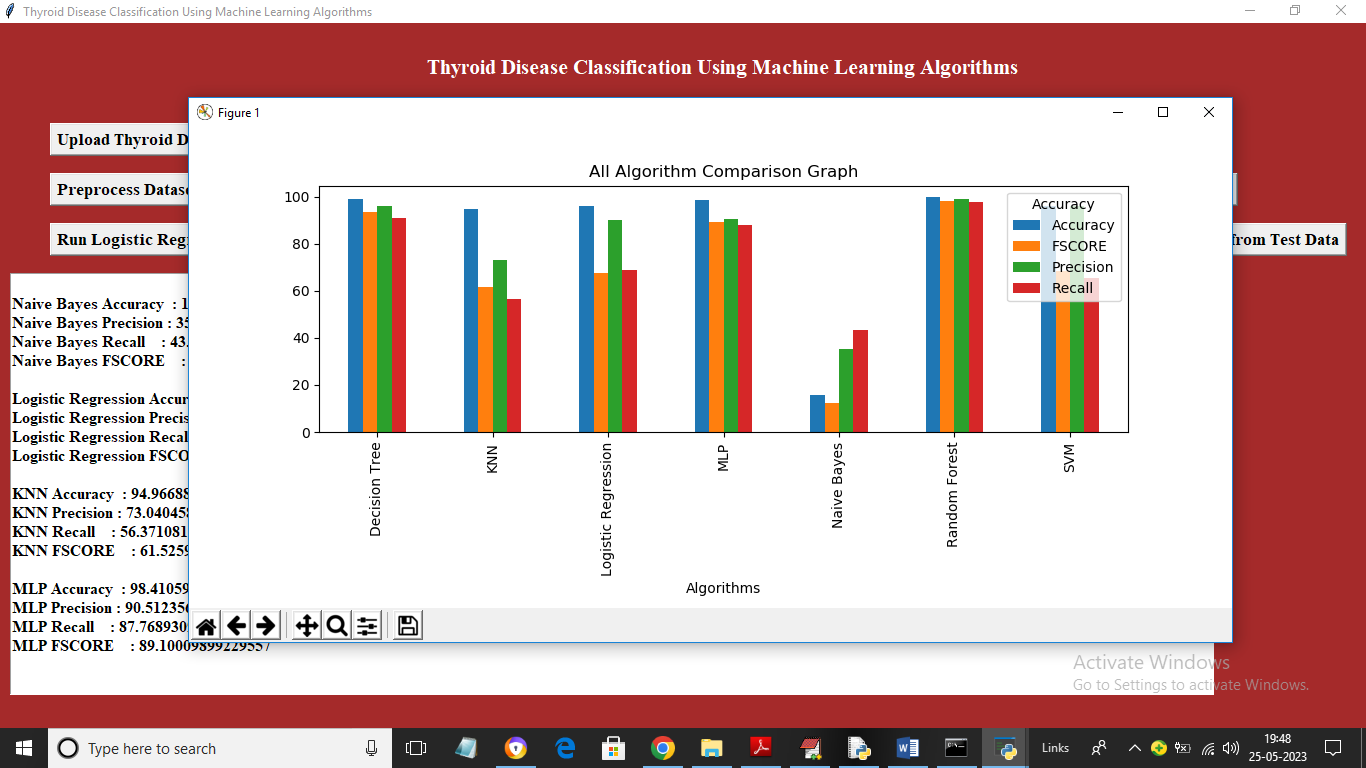
In above screen Logistic Regression got 96% accuracy



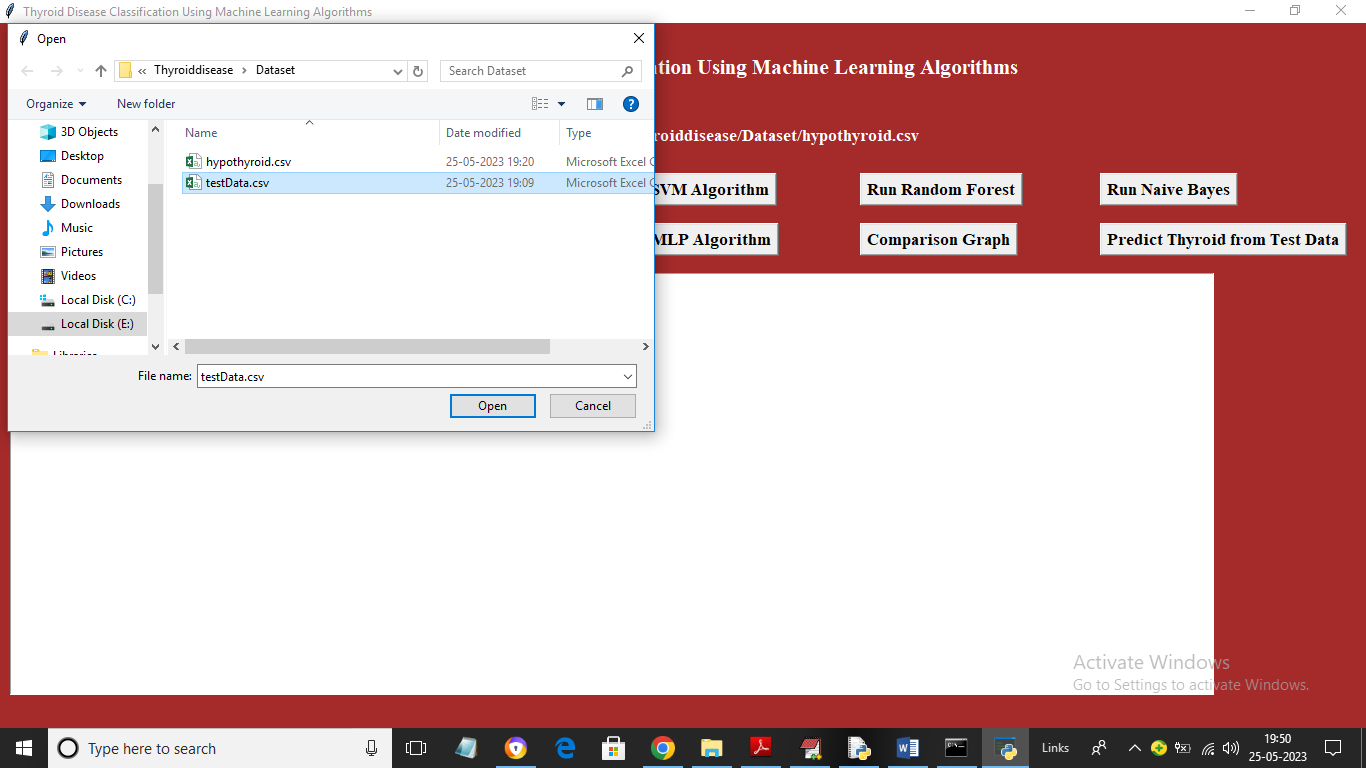
In above screen KNN got 94% accuracy



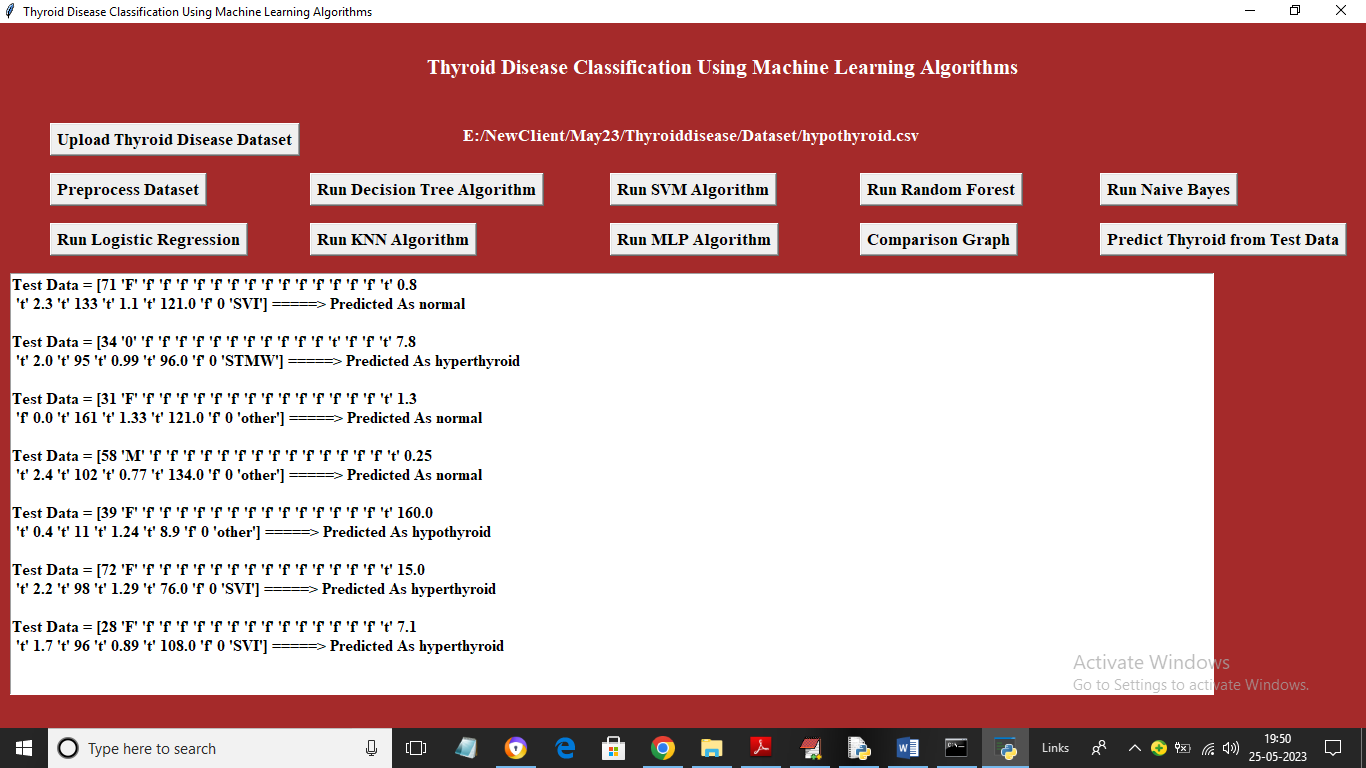
In above screen MLP got 98% accuracy and now click on ‘Comparison Graph’ button to get below graph



In above graph x-axis represents algorithm names and y-axis represents accuracy and other metrics in different colour bars and in all algorithms Random Forest and Decision Tree got highest accuracy and now close above graph and then click on ‘Predict Thyroid from Test Data’ button to upload test data and get prediction output



In above screen selecting and uploading ‘Test Data’ file and then click on ‘Open’ button to get below output



In above screen in square bracket we can see TEST data values and after =🡺 arrow symbol we can see predicted thyroid disease as Normal or Hypo or Hyper