**Diabetes Prediction Using Extreme Learning Machine: Application of Health Systems**

This work describes the combination of Software Engineering and Machine Learning algorithms to improve disease prediction in health care systems and to minimize time taken to predict disease as we don’t have enough hospitals or bed to accommodate growing number of patients and we can solve this problem of predicting disease with less time by employing software and machine learning algorithms. This proposed implementation concept is known as SEMLHI (where SE refers to software and ML refers to machine learning and HI refers to health data). It consists of 4 components

1. Health Informatics Data: To predict any disease we need to build Machine Learning models by using datasets and this datasets often contains missing data, null and non-numeric data and this type of data could degrade ML prediction accuracy and to overcome from this problem author is applying PREPROCESSING on health care data to remove all missing and null values and then convert non-numeric data to numeric data by applying python SKLEARN PREPROCESSING classes. Often this dataset may contains unnecessary columns or attributes and to remove this attributes author applying dimensionality reduction algorithm called PCA. PCA (principal component analysis) remove unnecessary attributes from dataset and maintain only important attributes necessary to make correct prediction.
2. ML Algorithms: This module uses various machine learning algorithms such as Linear SVC, Multinomial Naïve Bayes, Random Forest, Logistic Regression, KNN and extreme learning machine (ELM). This algorithm train itself with available datasets and then generate a train model and then this train model will be applied on new test data to perform prediction. By using above algorithms we can make machine to learn and perform prediction without any human supports.
3. Machine Algorithm Model: Once after building above models then we can apply new test data on this model to predict whether patient lab reports are positive or negative.
4. Software: This module used by developers to check reliability of above modules by applying software quality check, UNIT TESTING and software verification.

In propose work by using various size of dataset author applying classification, clustering ore regression and to implement this concept author is using Palestine Hospital dataset and this dataset not available on internet and authors also not publish this dataset on internet so I am using INDIAN DIABETES dataset. I will used this dataset to train above ML algorithms and then perform UNITTESTING to check all ML algorithms are giving accurate accuracy values.

SCREEN SHOTS

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

Graphical user interface, application

Description automatically generated

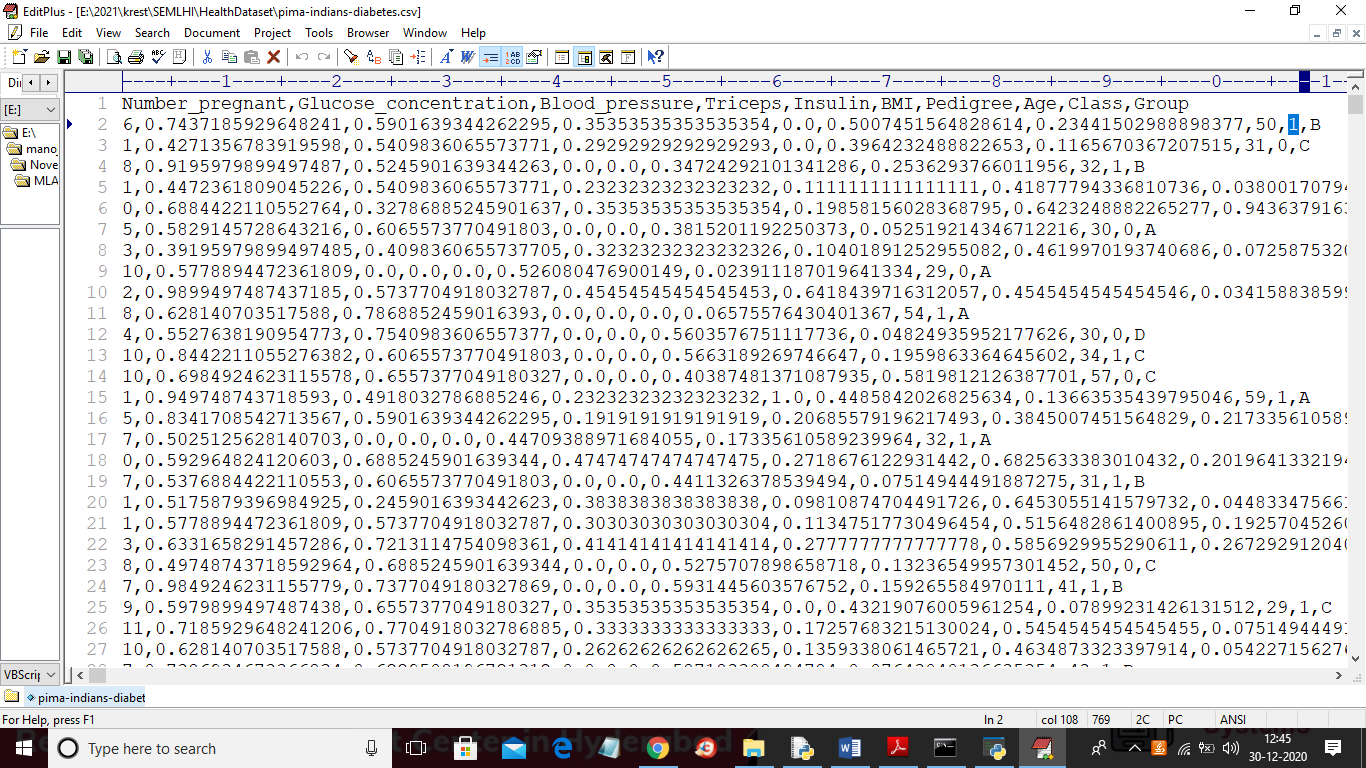
In above screen click on ‘Upload Healthcare Dataset’ button to upload dataset

Graphical user interface, text, application

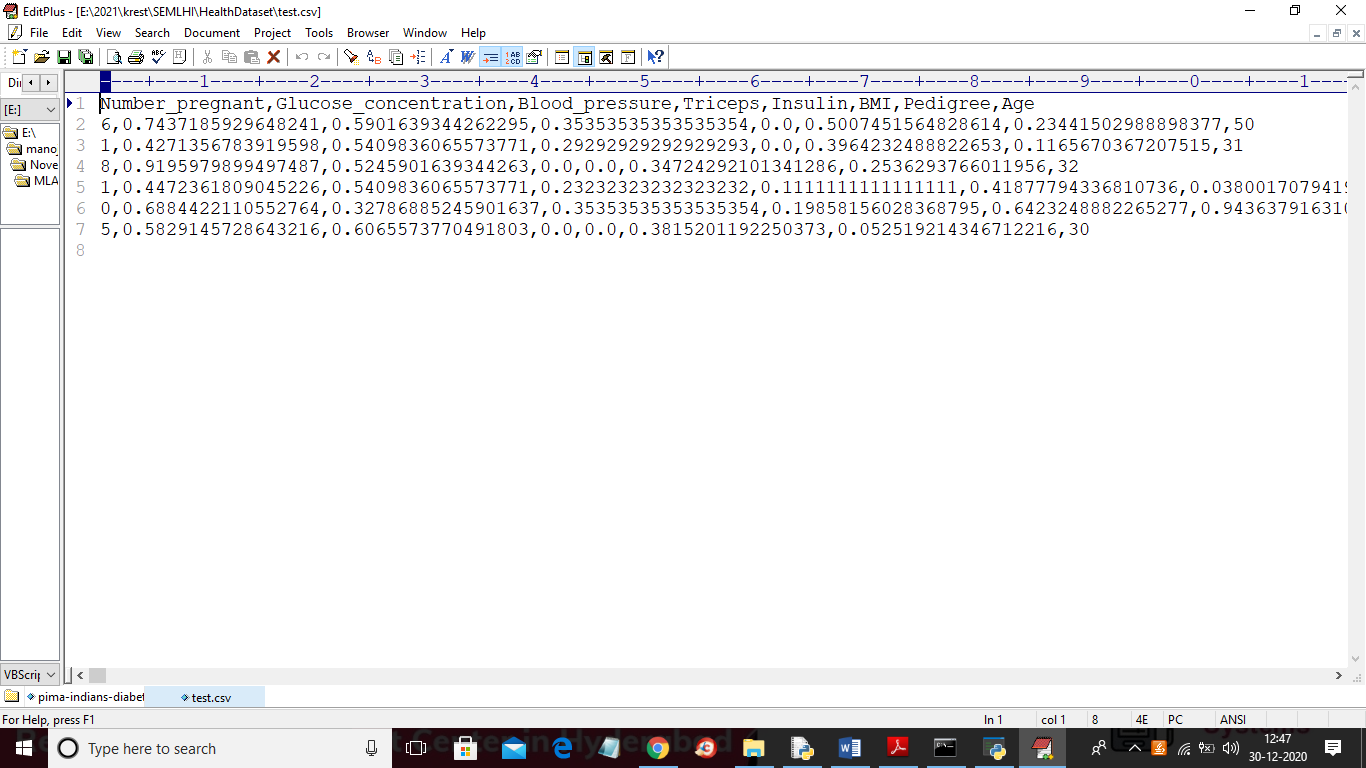
Description automatically generated

In above screen selecting and uploading diabetes dataset and then click on ‘Open’ button to load dataset.

In below screen of dataset we can see there is last label ‘Class’ which contains values as 0 and 1 where 0 means that lab values contains no disease and 1 means that lab values contains disease



In above dataset screen all values are the lab report values and ‘Class’ value contains 0 or 1 and ML algorithm will train with above lab report values and Class Value and then generate a model. Generated train model we will apply on below test data to predict class label. In below test dataset we can see there is no Class label column and ML will predict Class label by using alone lab values. See below test values



In above test lab report values there is no Class label. Now go back to output screen and then click on ‘Run Health Data Preprocessing’ button to remove missing values and then apply PCA dimensionality algorithm to get below graph

Graphical user interface, application

Description automatically generated

In above graph in top we can see names of columns and in boxes values with minus symbols are not important and only positive column values are important and ML algorithm will train only with positive values and now close above graph to get below screen

Graphical user interface, text

Description automatically generated

In above screen after applying pre-processing and PCA we got total records as 768 and application using 614 records to train ML algorithms and to generate model and then used 154 records to test that trained model and to calculate prediction accuracy. Now both train and test data are ready and now click on ‘Run Machine Learning Algorithms’ button to start training all ML algorithms on train and test data

Chart, scatter chart

Description automatically generated

In above graph green colour dots are the records which contains no disease and red colour dots are the records which contains disease and this graph generated for all 154 test records. Now close above graph to see all ML prediction accuracy

Graphical user interface, text, application

Description automatically generated

In above screen we can see prediction accuracy of each algorithm and from all algorithms ELM is giving good prediction accuracy and now all ML algorithms are ready with trained model and now click on ‘Predict Machine Algorithm Model’ button to upload new test records and then ML will predict whether new test records contain positive or negative disease

Graphical user interface, text, application

Description automatically generated

In above screen selecting and uploading ‘test.csv’ file and then click on ‘Open’ button to load test data and to get below prediction result.

Graphical user interface, text, application

Description automatically generated

In above screen for each test lab record ML predict whether disease is positive or negative. Now click on ‘Accuracy Comparison Graph’ button to get below graph

Chart, bar chart

Description automatically generated with medium confidence

In above graph x-axis represents ML algorithm names and y-axis represents accuracy of all those algorithms and from above graph we can conclude that extension ELM is giving better accuracy and now click on ‘Software Testing’ to check all ML are working properly and to get below screen

Graphical user interface, text

Description automatically generated

In above screen if all ML algorithms are working properly then we will get tests result as OK.