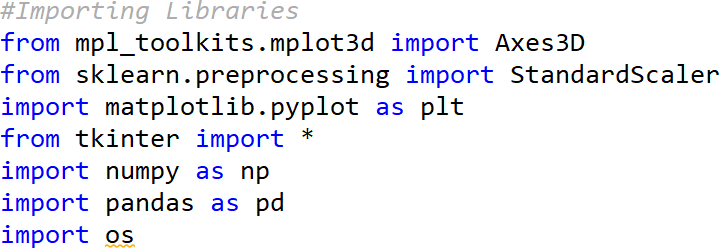
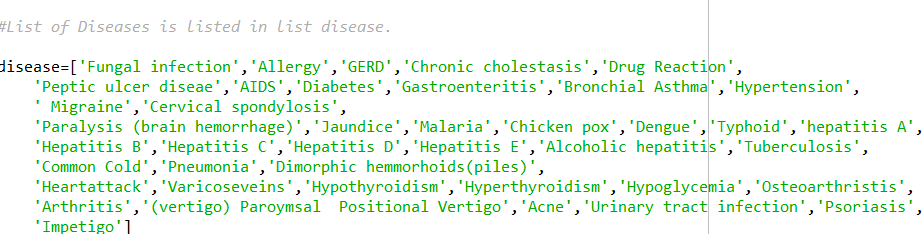
Compiled Report for Disease Prediction using Symptoms By- Moinuddin



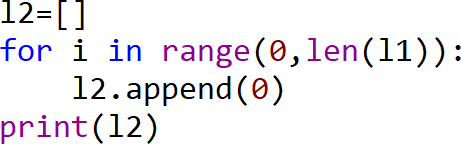
These are the imported libraries that are utilized to use various tools that are available in that specific library. Tkinter is used to build a Graphical User Interface in Python.



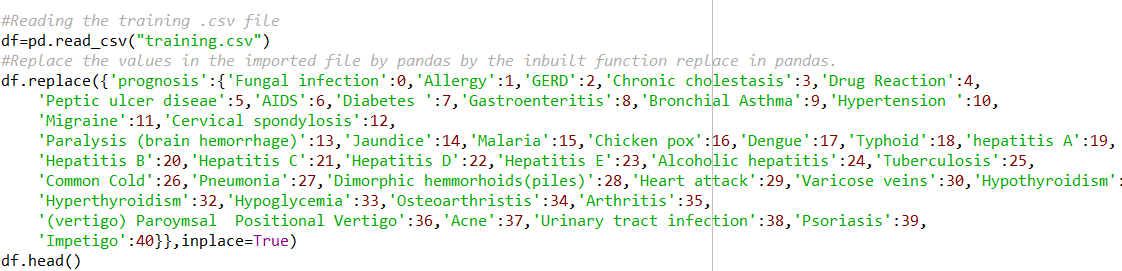
L1 is the list made for various Symptoms which are generally showed up in people for various Diseases.



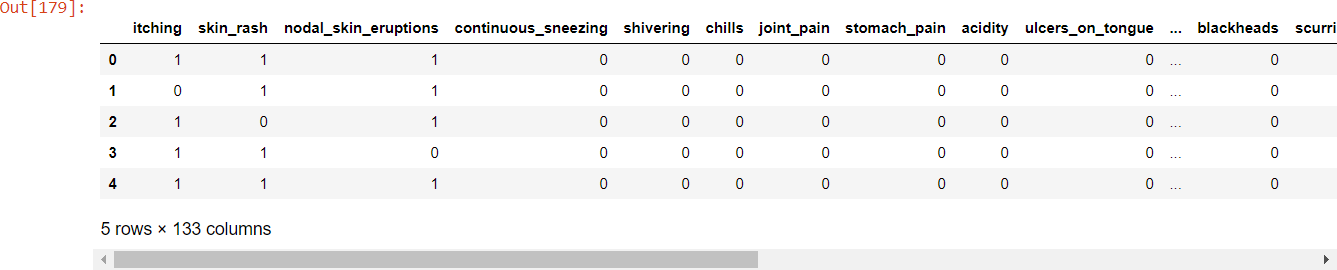
Disease is the list made for different Diseases which are for the most part appeared in different individuals.



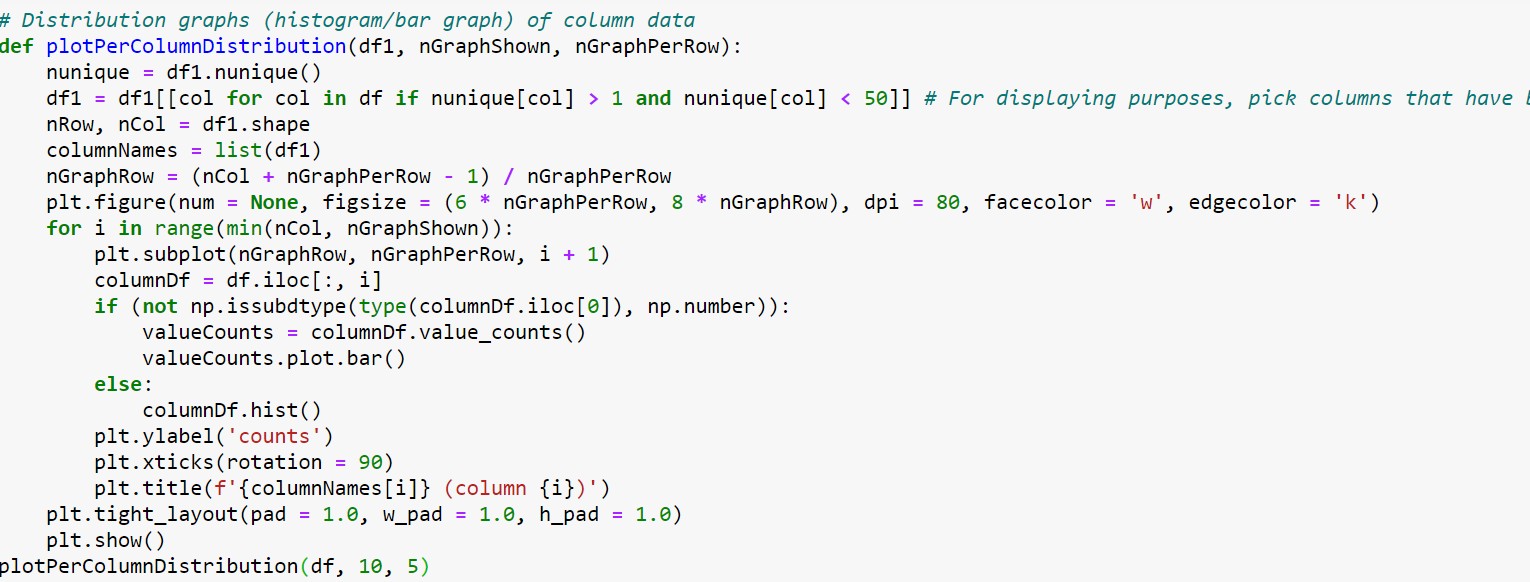
First L2 is the vacant list made. At that point, equivalent to a number of diseases in list L1, L2 is appended in a number of zeroes.



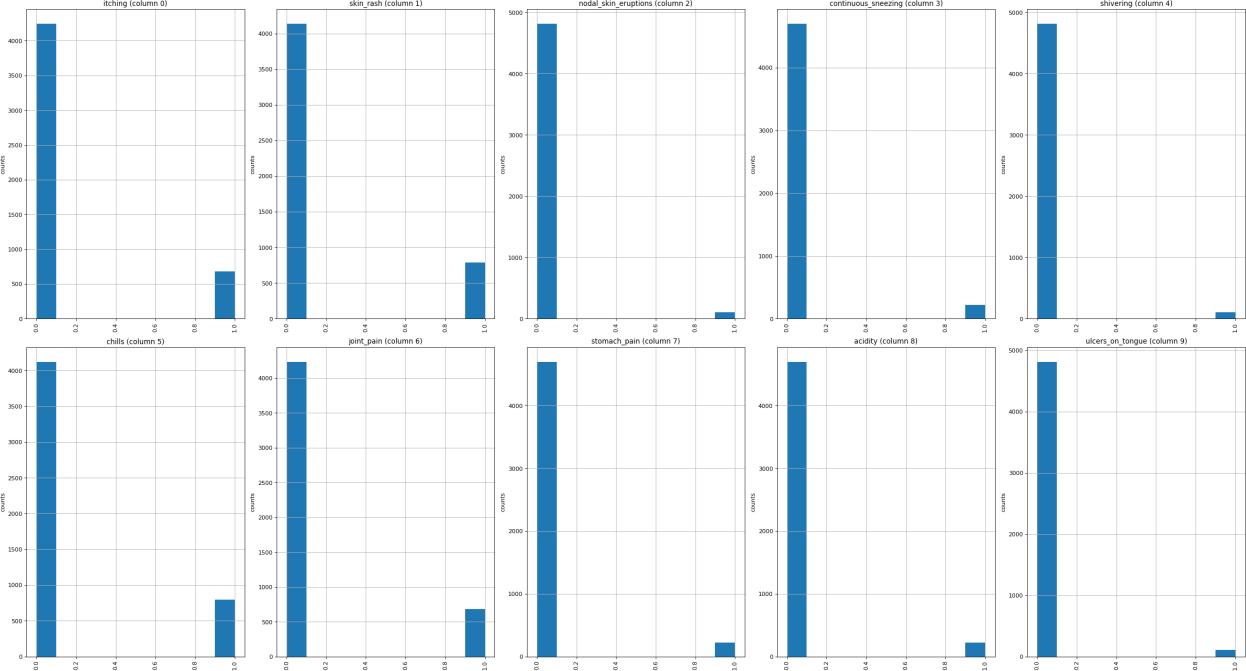
There is a CSV document containing diseases and symptoms, named training.csv, which is utilized to prepare the model. Read\_csv() function is utilized to store the information in the dataframe, named df. Utilizing replace() function, prognosis column that are the different diseases, it is replaced by the numbers from 0 to n-1, where n is the number of different diseases present in .csv record. Head() function is utilized to print the initial five rows of the preparation dataframe.



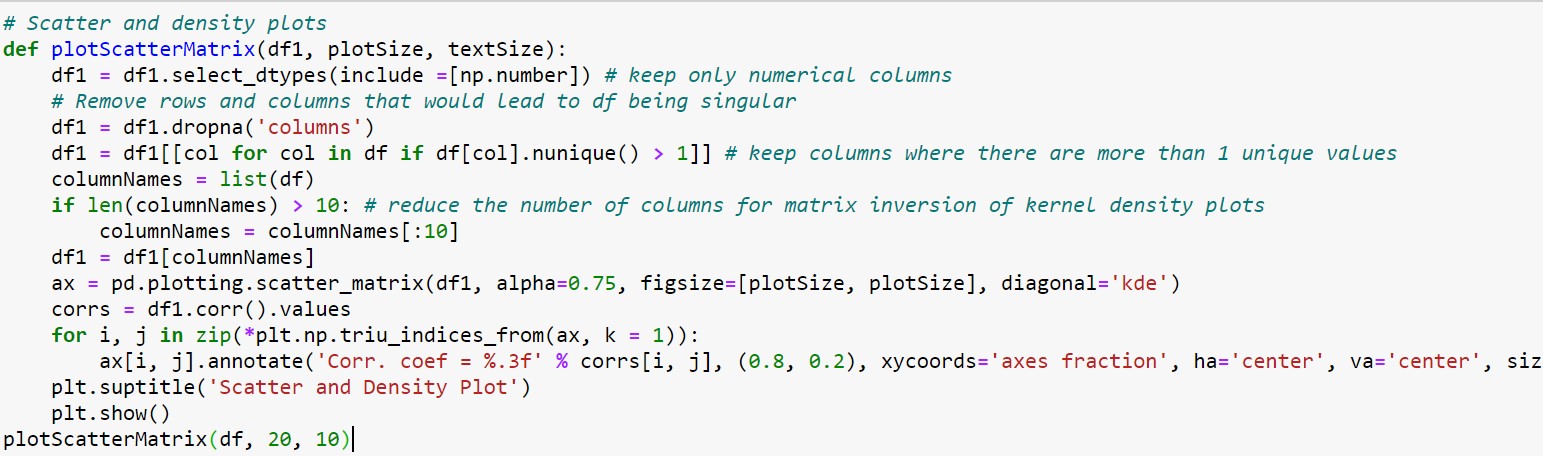
This is the output produced which contains the initial five rows of the dataframe df.



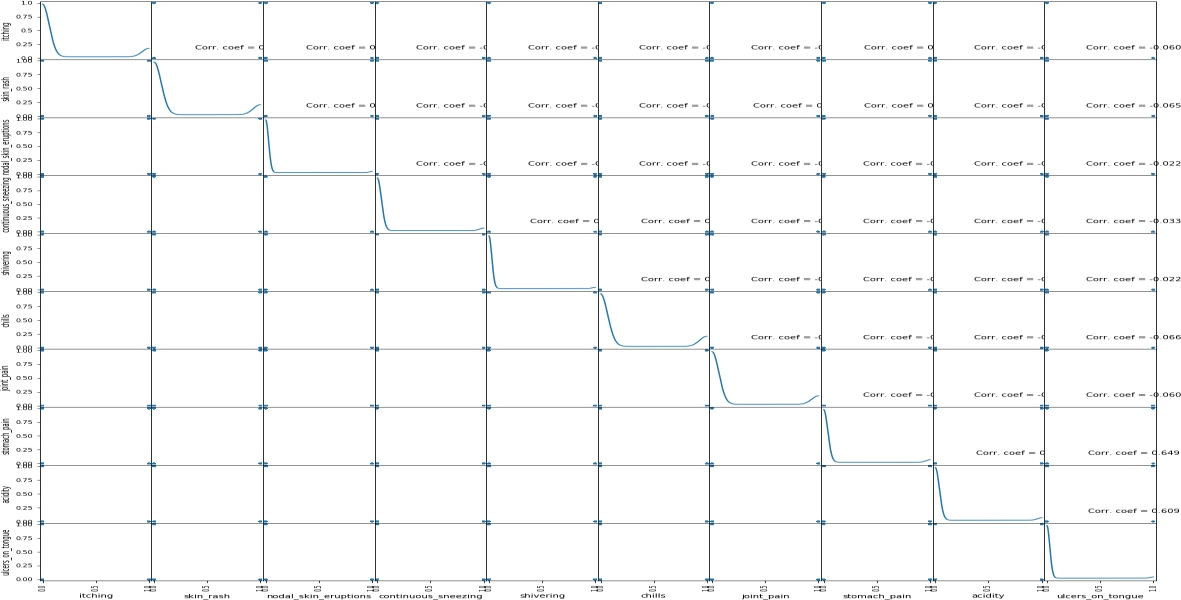
This is the code for the distribution graph of the columns of training.csv file.



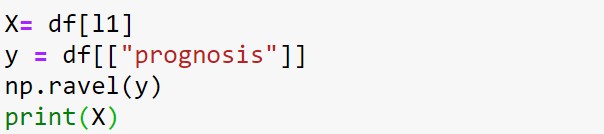
Output for the distribution graph of the columns of training.csv file.



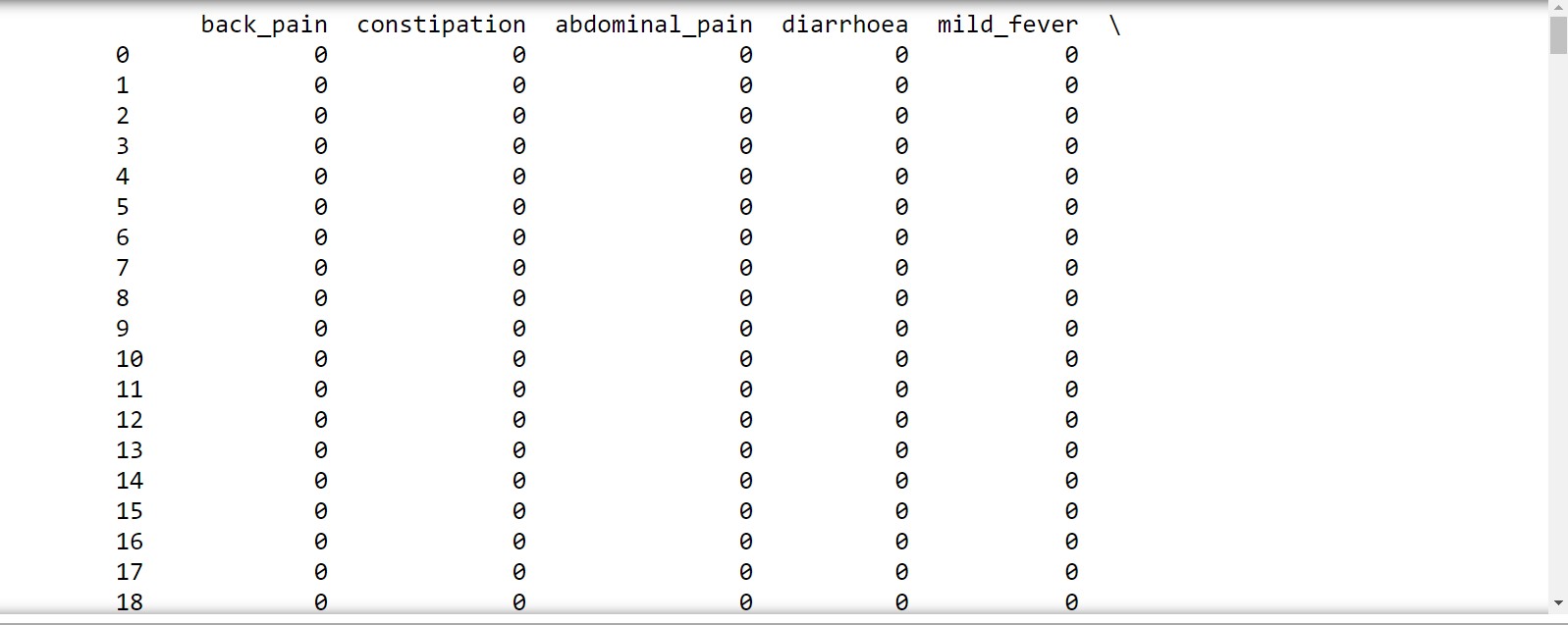
This is the code for the scatter and density plots of the columns of training.csv file.



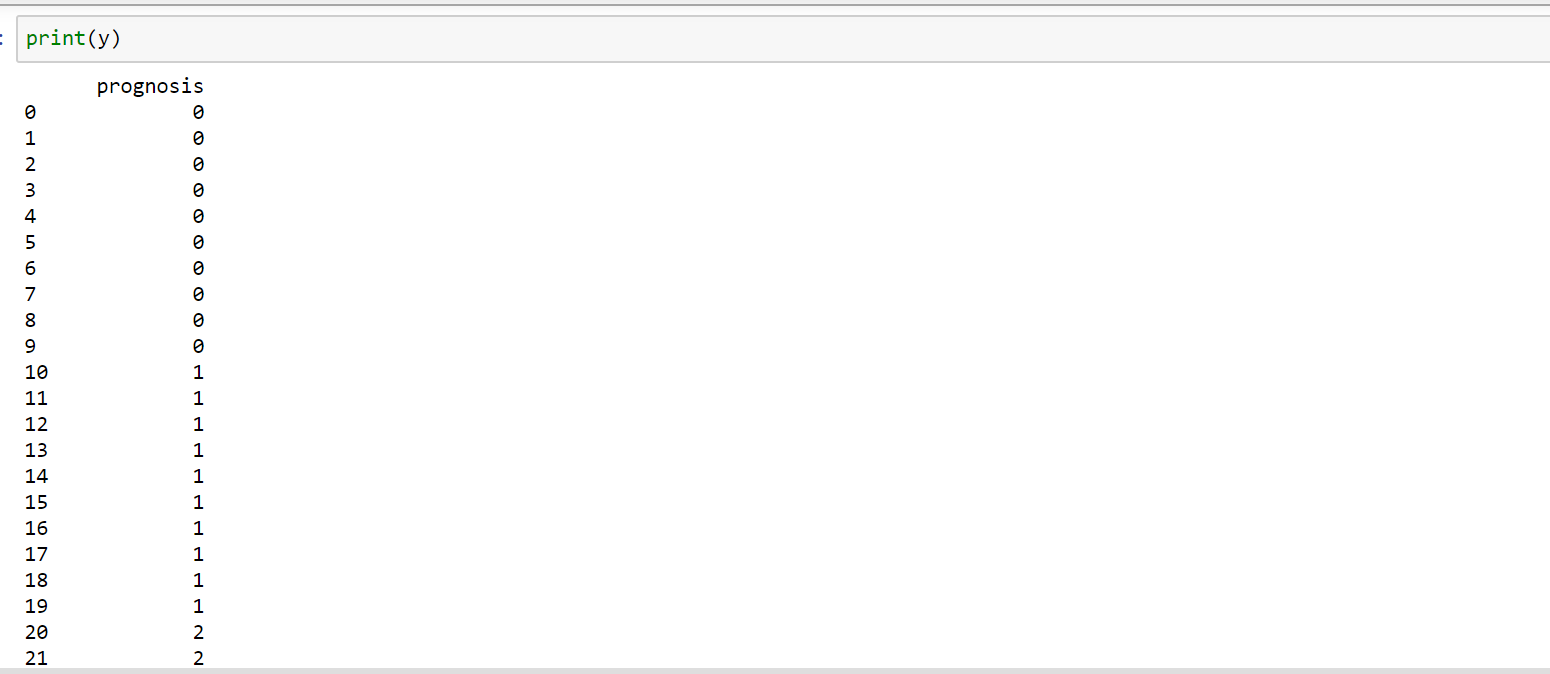
Output for the distribution graph of the columns of training.csv file.



Putting the Symptoms in X and prognosis/diseases in y for training the model.



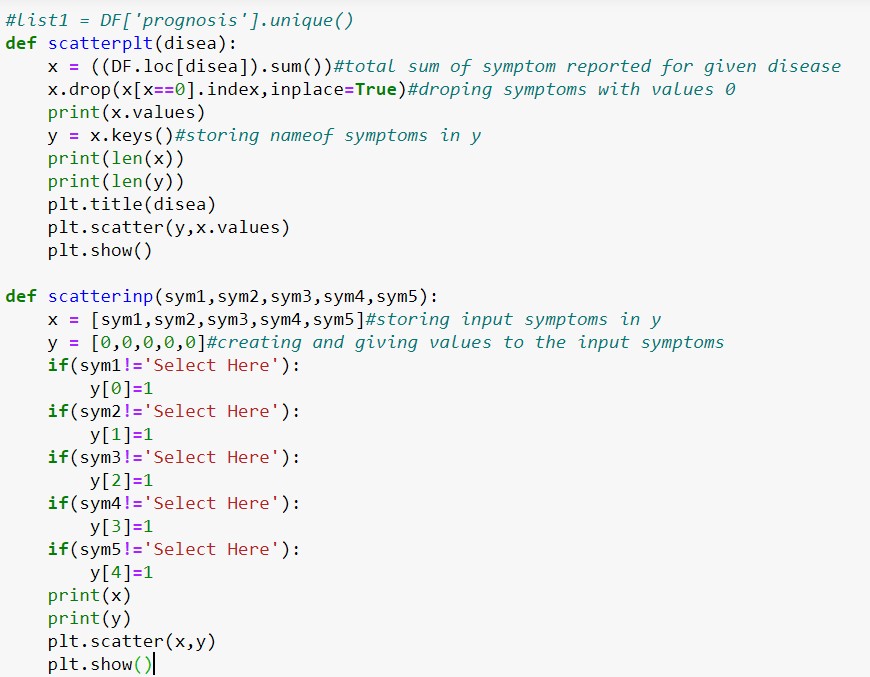
Output for the print(X) in which different symtoms has the values ‘0’ or ‘1’ according to their presence in the particular diseases



Output for the print(y) in whci different disease has values according to their symptoms.

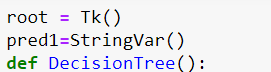
To build the precision of the model, we utilized four distinctive algorithms which are as per the following

* Decision Tree algorithm
* Random Forest algorithm
* KNearestNeighbour algorithm
* Naive Bayes algorithm

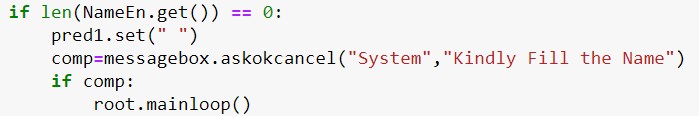


These are the function to plot the scatterplot of the predictions of the diseases and the symptoms entered by the user.

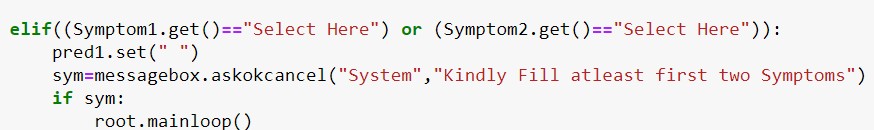
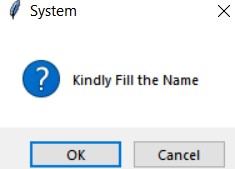
Decision Tree Function:



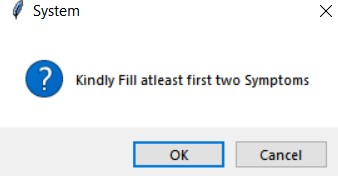
Root=Tk() is used to for start working with the tkinter to build the gui. Definition of DecisionTree() function. “pred1” is used to store the predicted disease using decision tree algorithm.



If user tries to run the gui without entering the name, then System will prompt the following message.

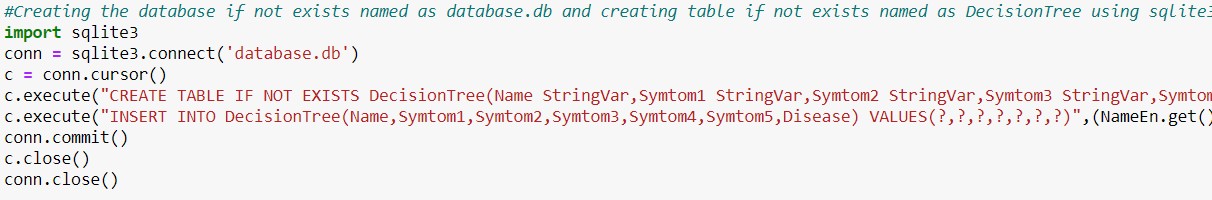


After filling the name, user have to fill five symptoms and out of which first two are compulsory. If user will not select atleast two symptoms, then following message will be prompt from the system

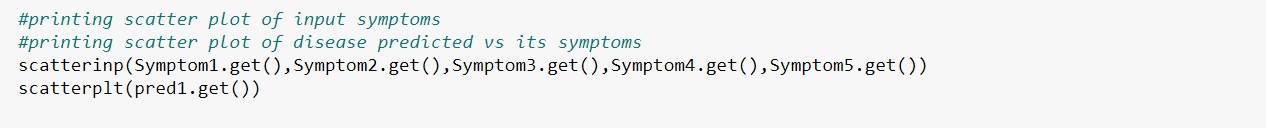


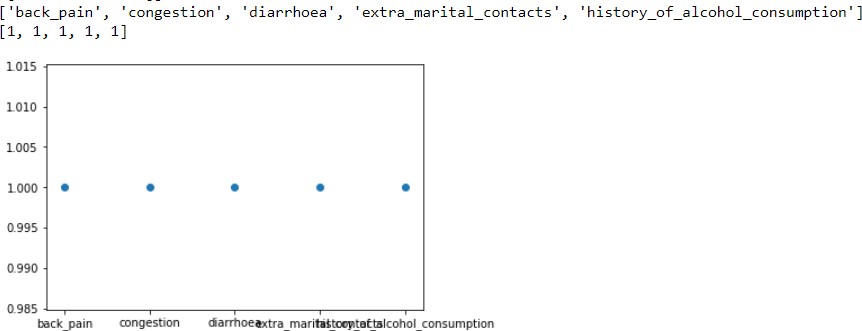


DecisionTreeClassifier() is used to train the model and predict the disease on testing dataset according to symptoms entered by the user. Final disease for decision tree is stored in a variable named “pred1”. Accuracy of predicting the disease is printed using accuracy\_score and confusion matrix is created using confusion\_matrix which are imported from sklearn.metrices.

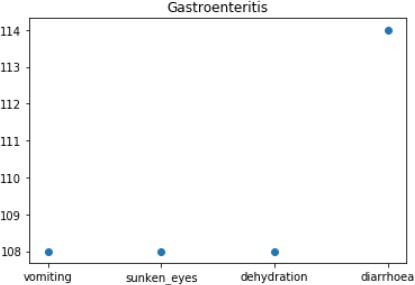


Creating a database named “database.db”, if not exists, using “sqlite3” database. For storing data for decision tree algorithm "DecisionTree" table is created, if not exists, in database.db using “CREATE” function in sqlite. Values are inserted in DecisionTree table using “INSERT” function in sqlite.





The scatterplot for the symtoms which are given by the user as input



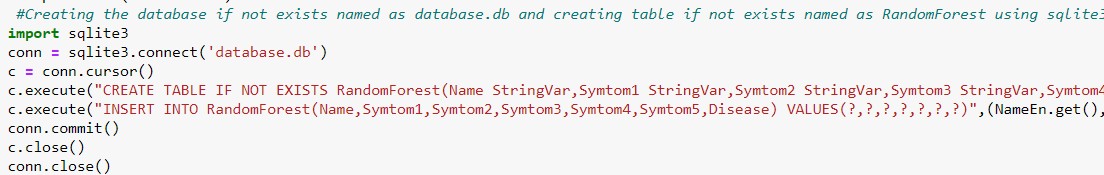
The scatter plot for the disease on the basis of symptoms which given by the user as input Random Forest function:



Definition of randomforest() function. “pred2” is used to store the predicted disease using random forest algorithm.

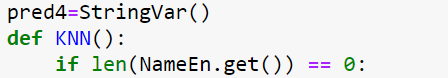


RandomForestClassifier() is used to train the model and predict the disease on testing dataset according to symptoms entered by the user. Final disease for random forest is stored in a variable named “pred2”. Accuracy of predicting the disease is calculated using accuracy\_score and confusion matrix is created using confusion\_matrix which are imported from sklearn.metrices.



Same databse is used i.e.,databse.db that is used in decision tree algorithm with different table for random forest algorithm which is name as “RandomForest”.

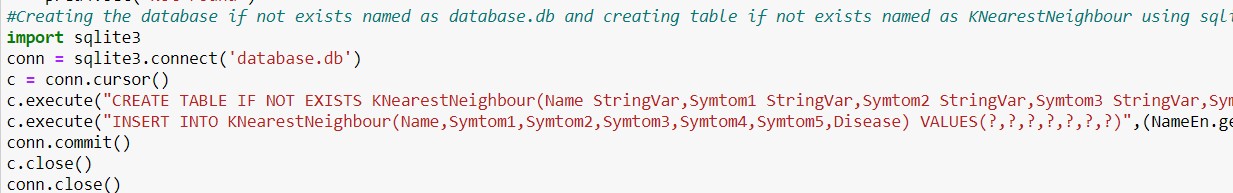
kNearest Neighbour function:



Definition of KNN() function. “pred4” is used to store the predicted disease using kNearestNeighbour algorithm.

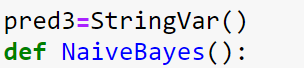


KNeighboursClassifier() is used to train the model and predict the disease on testing dataset according to symptoms entered by the user. Final disease for kNearestNeighbour is stored in a variable named “pred4”. Accuracy of predicting the disease is calculated using accuracy\_score and confusion matrix is created using confusion\_matrix which are imported from sklearn.metrices.



Same databse is used i.e.,databse.db that is used in all previous algorithms with different table for kNearest Neighbour algorithm which is name as “KNearestNeighbour”.

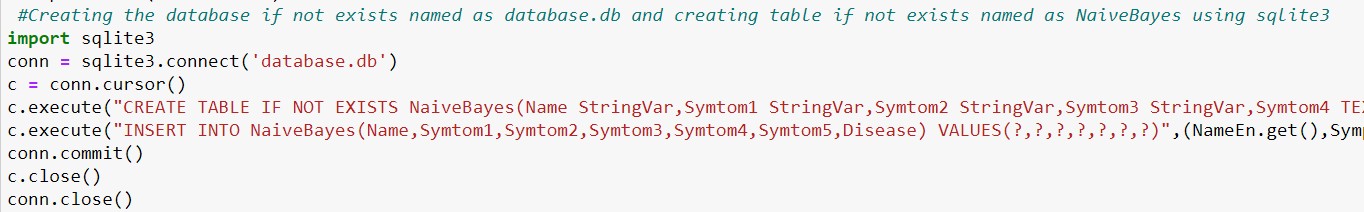
Naïve Bayes function:



Definition of NaiveBayes() function. “pred3” is used to store the predicted disease using Naïve Bayes algorithm.

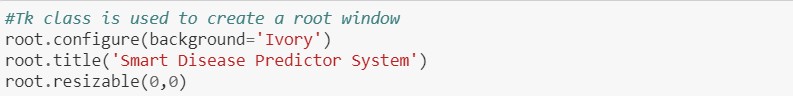


GaussianNB() is used to train the model and predict the disease on testing dataset according to symptoms entered by the user. Final disease for Naïve Bayes is stored in a variable named “pred3”. Accuracy of predicting the disease is calculated using accuracy\_score and confusion matrix is created using confusion\_matrix which are imported from sklearn.metrices.

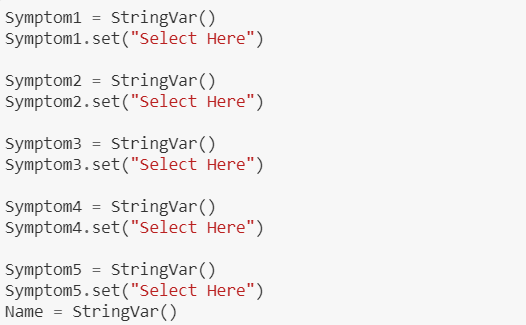


Same databse is used i.e.,databse.db that is used in all previous algorithms with different table for Naïve Bayes algorithm which is name as “NaiveBayes”.

Building the Graphical User Interface:



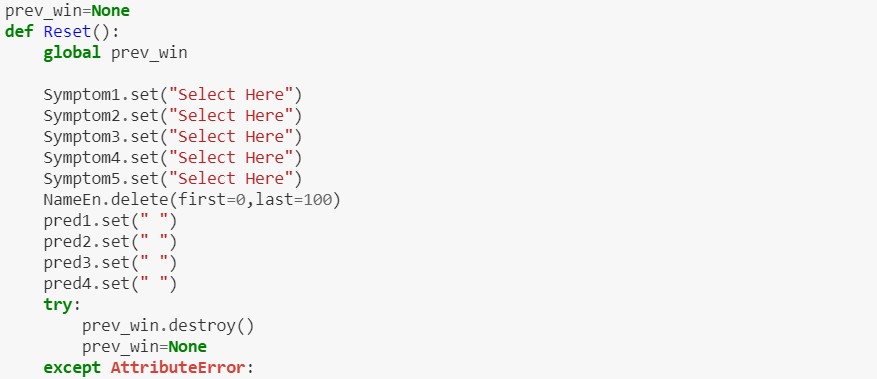
Graphical User Interface is build using tkinter library in Python. Root is used to start the GUI. It is configured with the background that is set to “Ivory”. GUI titlt is given as “Smart Disease Predictor System” using title() function in tkinter library. Resizable function is used to fix the size GUI.



Here, variables are defined like Name, Symptom1, Symptom2, etc and they initialised to “Select Here” using set() function in tkinter library.



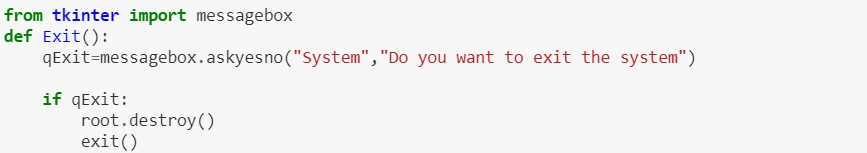
This is how the above variables are looking like in GUI.



It is the definition of the function “Reset()” which is used to reset the GUI inputs which are given by the user. It is called when user click on the button “Reset Inputs” from the GUI.



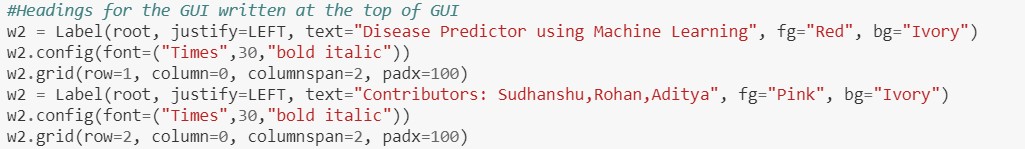
“Reset Inputs” button in GUI



It is the definition of the function “Exit()” which is used to come out from the GUI. It is called when user click on the button “Exit System” from the GUI.



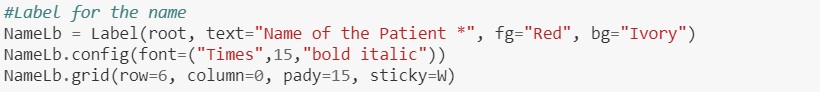
“Exit System” button in GUI



“W2” is the label created for showing the headings in the GUI using label() function from tkinter library. Two text are written under label w2 in row1 and row2 with font features as “Times”,”30”,”bold italic”.



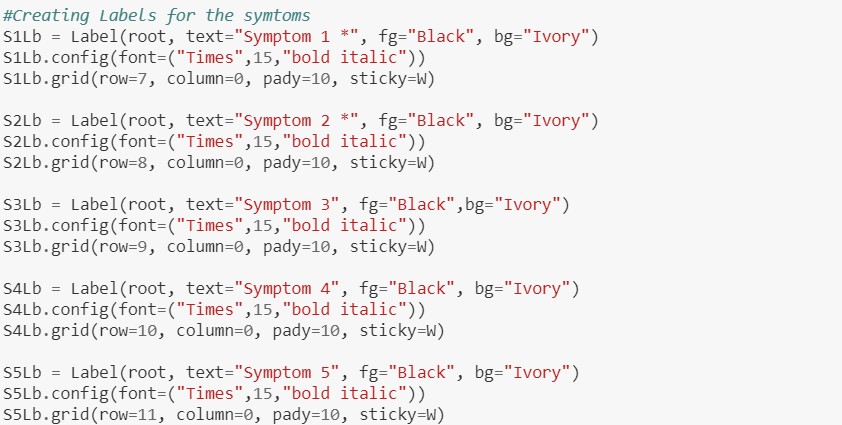
This is how w2 label is available in the GUI



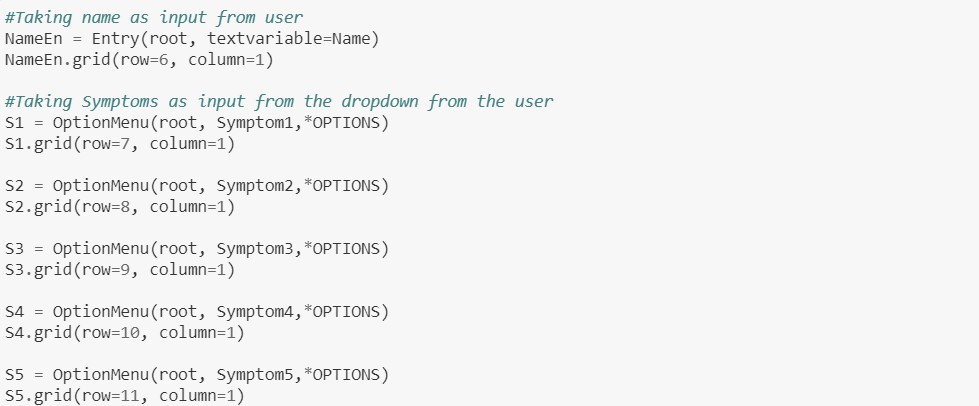
“NameLb” is the label created for showing the “Name of the Patient \*” using label() function in tkinter library. It is configured using config() function and the grid of the label is set using grid() function.



NameLb label in GUI. ‘ \* ’ shows that it is compulsory for the user to give his/her name.



These are the labels for showing the Symptoms of the disease. It is created using label() function from tkinter library. Its features are configured using config() function and their grid is set by using grid() function from tkinter library.



NameEn is the entry box created for getting the name of the patient using Entry() function in tknirter library. S1, S2, S3, S4, S5 are the option menu used to get symtoms from the user which is created using Optionmenu in tkinter library. \*OPTIONS is the list of unique symtoms.



List of symptoms available for user





These are the labels created for showing the texts of different algorithms



Buttons created for predicting the disease using different algorithms.

* Press Prediction 1 for Decision tree algorithm
* Press Prediction 2 for Random forest algorithm
* Press Prediction 3 for Naive bayes algorithm
* Press Prediction 4 for K-Nearest neighbour
* Press Reset Inputs for resetting the inputs
* Press Exit System for Exiting from the System





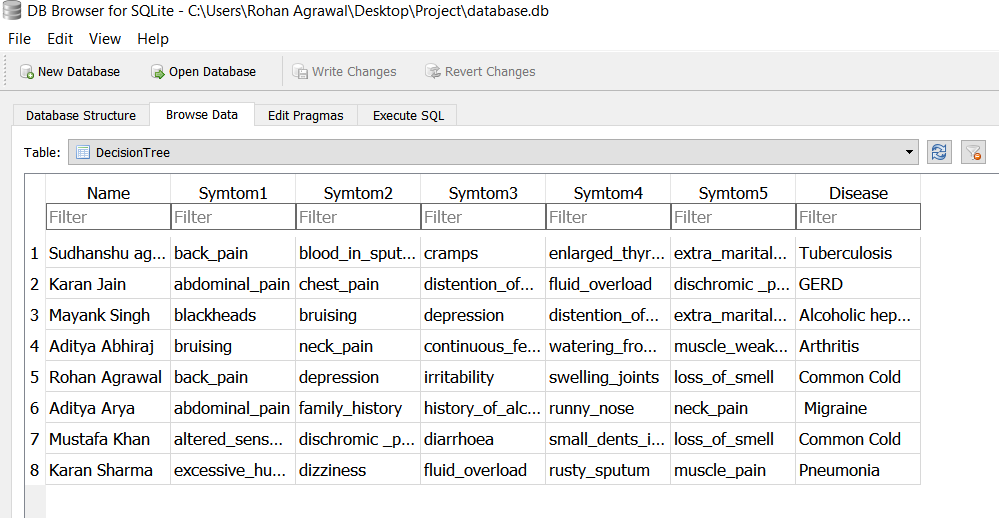


These are label created for showing the predicted disease using different algorithm.





This is the calling of the GUI.



The database Created using Sqlite3