**Tumor Cancer Prediction**

**TA: Alaa Tarek**

|  |  |
| --- | --- |
| ID | Name |
| 20201700348 | **سلمي رامي محمود ابراهيم** |
| 20201700295 | **ريناد هيثم محمد اسعد** |
| 20201700294 | **ريمون مجدي متي لوقا** |
| 20201700390 | **سيف الدين وائل السيد نصر حجاب** |
| 20201700303 | **زياد حاتم عزت السيد** |
| 20201701070 | **حسان جمال حسان حسني** |
| 20201700391 | **سيف ايمن احمد عبد الخالق** |

**Introduction**

The following report will contain the methods we used for this project step by step and why we used these methods.

* First Dividing the project into many classes to achieve the reusability, we divide it into six classes:
  1. [source] which is the main class.
  2. [PRE\_PROCESS] which contain the preprocessing of the dataset to make the module easy to train.
  3. [ALGORITHMS] which contain all the four algorithms used in the project (logistic regression, decision tree, SVM, random forest classifier).
  4. [PREDICTION] which print all the prediction of the four algorithms.
  5. [VOTING] which contain the vote function to see if the person will get tumor or not.
  6. [SCALING] which contain function to appear the normalization of the dataset.
* We make a constructor for each class which all contain all variable we need it.
* [PRE\_PROCESS] class is used to preprocess the dataset to make the module easy to train and it contain:
  1. Dropping all null values by using [dropna ()] function.
  2. Removing all duplicate values by using [drop\_duplicates ()] function.
  3. Changing (diagnosis) column to (0 and 1) instead of (B and M) to be able to normalize the dataset after that by using [LabelEncoder] library.
  4. splitting the dataset into independent dataset which contain all columns from (F1 to F30) columns and dependent dataset which contain (diagnosis) column.
  5. splitting our dataset in 75% training our model and 25% to test our model by using [train\_test\_split] library.
  6. standarazible the data by using [StandardScalar] library for the [X\_train] and[X\_test].

Graphical user interface, text, application, email

Description automatically generated

* [ALGORITHMS] class used four algorithms (logistic regression, decision tree, SVM, random forest classifier) and import these libraries from [sklearn] library:
  1. training our model by using these four algorithms and print the accuracy score by using our training dataset.
  2. Then import the metrices from [sklearn] library to get the confusion matrix from it and get some measurement as accuracy which is calculated by [(tp+tn)/(tp+tn+fp+fn)] and precision which is calculated by [tp/(tp+fp)] and Recall which is calculated by [tp/(tp+fn)] from our confusion matrix by using our test dataset.
  3. In SVM algorithm there is three kernels used to train our model (linear, polynomial, radial basis function(rbf)) but linear kernel is the most kernel who print the high accuracy so choose it.
  4. In logistic regression we use [solver='liblinear’] to be able to train the model in the small data set

**Logistic Regression algorithm**

**Text

Description automatically generated**

**Text

Description automatically generated**

**Decision Tree algorithm**

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

**SVM algorithm**

**A picture containing text

Description automatically generated**

**Text

Description automatically generated**

**Random forest algorithm**

**Text

Description automatically generated with medium confidence**

**Text, letter

Description automatically generated**

* [PREDICTION] class used to predict by four algorithms the model to see who will get tumor in every algorithm.

Prediction class

A screenshot of a computer

Description automatically generated with medium confidence

Text

Description automatically generated

A picture containing text, indoor

Description automatically generated

* [VOTING] class used to vote in our prediction between the three algorithms (logistic regression, decision tree, svm) to be sure who will get tumor by see every prediction in each algorithm what it predicts if he gets tumor or not and the more prediction, we take it and print it.

Voting class

Graphical user interface, text, application, email

Description automatically generated

Text, table

Description automatically generated

* [SCALING] class used for data scaling to print the normalization of the model which equal [Z = (X - Mean)/Variance] by import [Normalizer] from [sklearn. preprocessing] then fit the data then using distribution plot by using [seaborn] library and we print the standardization also with normalization by make [KDE = True] and print the plot by [matplotlib] library.

Chart, histogram

Description automatically generated

* [SOURCE] class which is the main class for our project:
  1. Reading the dataset by using (read\_csv) function in [pandas] library.
  2. Calling [PRE\_PROCESS] class to the dataset to preprocess it.
  3. Calling [ALGORITHMS] file and make four objects from it each object is identified by one class from the four algorithms (logistic regression, decision tree, svm, random forest classifier).
  4. Calling [PREDICTION] class to print all the prediction of four algorithms.
  5. Calling [VOTING] class to vote in our prediction.
  6. Calling [SCALING] class to show the normalization graph.
  7. Then use [joblib] library to save and load model for the four algorithms by making variable and get to it the name of the (.sav) file and used the (dump) function to put in (.sav) file the algorithm we need to be can load and use it after that.
  8. Then make a new dataset and read it and preprocess it and print their accuracy score by load files we do that to make it easier to use the three algorithms to predict without train the model again.
  9. Calling [PREDICTION] class to print all the prediction of four algorithms.
  10. Calling [VOTING] class to vote in our prediction.
  11. Calling [SCALING] class to show the normalization graph.