Healthy Heart Predictor

Project Link:
https://colab.research.google.com/drive/1Q2E4ZOWiSmNk6BlEbNPnyypLKnlUUaY7?usp=sharing

Introduction: An acquittance of ours had done a project based on the anomaly in heart beats. From the above idea and various other parameters of the human body, we decided to use Machine Learning to predict whether a heart is healthy or not.

Machine learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so.

LOGISTIC REGRESSION:

1) Collection of data with various parameters
    - The medical information was obtained in the form of 14 parameters (age, sex, blood pressure, max heart rate, etc.) based on multiple ECG'S from website named Kaggle.
    - The data set also a target value for each instance:
      0→ Person has healthy heart.
      1→ Person has an unhealthy heart.
    - References: https://www.kaggle.com/ronitf/heart-disease-uci

2) Process the data and make it fit to feed into our model
    - At first, we didn't know what libraries need to be imported to execute the logistic regression model but upon research we learnt about the following dependencies.
      NumPy, Pandas and sklearn.
    - Dependencies are libraries and functions that are required to execute the project.

      ----> NumPy: NumPy is used in making Numpy array and is similar to lists in python.

----> Pandas: It is used for creating data frames which are structured tables and is in readable format.
----> Sklearn: The sklearn library contains a lot of efficient tools for machine learning and statistical modelling including classification, regression, etc, via consistent interface in Python.

- We loaded the downloaded data set to the pandas data frame which was called as data_heart.
- Checked the missing values in the data set which may occur due to data corruption or failure to record data. This should be done to avoid biased estimate which may lead to an invalid conclusion.
- Now we count the value/distribution of 0's and 1's of which we need an almost equal amount. (138-165).

References:

1)https://www.w3schools.com/python/numpy/numpy_creating_arrays.asp
2) https://www.tutorialspoint.com/python_pandas/index.htm
3) https://www.tutorialspoint.com/numpy/index.htm
4)https://www.tutorialspoint.com/scikit_learn/scikit_learn_logistic_regression.htm
5)https://www.bing.com/videos/search?q=sklearn+logistic+regression&docid=608011917220973970&mid=1E034FB866585BC3A18A1E034FB866585BC3A18A&view=detail&FORM=VIRE

3) Splitting the data set into training data and test data
    - First the data is split into A and B which contains the values of features apart target and the target respectively which is further split into training data and test data.
    - We used 80% of the data for training and the rest for testing.
    - Stratified the set(B_test, B_train) to make sure that 0's and 1's are evenly distributed between the training set and testing set.

References:

1) https://towardsdatascience.com/train-test-split-c3eed34f763b
2) https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-

algorithms/#:~:text=The%20reason%20is%20that%20when%20the%20dataset%20is,test%20set%20to%20effectively%20evaluate%20the%20model%20performance.

3) https://www.geeksforgeeks.org/splitting-data-for-machine-learning-models/

4) https://www.r-bloggers.com/2020/06/why-balancing-your-data-set-is-important/#:~:text=One%20of%20the%20rules%20in%20machine%20learning%20is%2C,90%20observations%20and%20class%20B%20with%2010%20observations.

4) Logistic regression model

- It is a method that is used in Machine learning for binary classification problems, i.e problems with two class values to make probability prediction.

References:

1) https://www.analyticsvidhya.com/blog/2021/08/conceptual-understanding-of-logistic-regression-for-data-science-beginners/

2) https://www.upgrad.com/blog/logistic-regression-for-machine-learning/

3) https://writersbyte.com/understanding-and-implementing-logistic-regression-algorithm-part-1-python-machine-learning/

4) https://www.geeksforgeeks.org/ml-logistic-regression-using-python/?ref=lbp

5) https://www.geeksforgeeks.org/understanding-logistic-regression/

5) To the trained model, feeding the data and procuring the prediction.

- Logistic regression is used to train the model. The function fit() is used to find the relation or pattern in the target values and features in the training data set.
- Model evaluation: The model predicts and compares the target vales with the original target values. The amount of target values correctly estimated gives the accuracy score. (required to be greater than 75%)

- The accuracy of our training data set came up to almost 85% and test data to 82%
- Now, we build a predictive system which takes in features as input values.
- We convert the tuple into numpy array to facilitate the process of reshaping. We reshape it to tell the model to predict the target value for only one instance and not for the whole data set.
- Now we print the predicted target value and check it against the original dataset
- We got the desired result.

References:

1) https://www.freecodecamp.org/news/how-to-build-and-train-linear-and-logistic-regression-ml-models-in-python/
2) https://vitalflux.com/python-train-model-logistic-regression/

BT20CSE004- Abhirami Settypalli

BT20CSE127-Shreya Datta

BT20CSE128-Bheemarapu Srijani

NOTE: If the program doesnt run in the system ,please do upload the data set into the Google colab.

The link for the data set: https://www.kaggle.com/ronitf/heart-disease-uci