Project 1 Report

By: Vaibhav Rao
Email: Vaibhavr@buffalo.edu
Person No.: 50375332

In this project we have created a model using Stochastic Gradient Descent classifier on the Wisconsin Diagnostic Breast Cancer dataset file as provide in the project (wdbc.dataset)

The project has been enclosed in the zip file as per specifications mentioned and include the file main.py.

TASK 1 (Plan Of Work 1-4)

For Reference:

1. Extract features values and Image Ids from the data: Process the original CSV data _les into a Numpy matrix or Pandas Dataframe. Apply feature scaling technique to make sure that all the features are in the same level of magnitude.
2. Data Partitioning: Partition your data into training and testing data. Randomly choose 80% of the data for training and the rest for testing.
3. Train using Scikit learn library: Use Scikitlearn library function SGDClassifier to train on the dataset.
4. Print results of Scikit learn library: Your code should print Accuracy, Precision, Recall and Confusion matrix resulting from scikitlearn SGDClassi_er model. Your report should describe the results.

These steps are carried out in the code Till Line 56.

We have extracted the values from the dataset file into a pandas dataframe. Note while extracting we did set the header=none since the dataset doesn't contain any column headers and so that our extraction does not miss any row in the data by discarding it as header as default.
Also we have segregated column 1 as the TARGET VARIABLE.
We have taken Column 2-31 as the FEATURES.
We have ignored column 0 which is the ID since it is not relevant for our model.

We also did apply feature scaling using Standard scaler on this data (Line 37-40)
For partitioning data randomly by choosing 80% data for train and 20% data for test we used train_test_split library function.

Then we just trained our classfier on the training data and did call predict on the remaining test data. (lines 47-49)

We did print the Confusion matrix and also the values like Accuracy score, Prediction Score and Recall Score. To validate the results we can easily do so from the confusion matrix obtained for this sample.

Confusion matrix from validation data
[[68 3]
 [1 42]]
From this we get TP = 42, TN=68, FP=3, FN=1.

Substituting in below formulas.

Accuracy = (42 + 68) / (42 + 68 + 3 + 1)
= 110/114
= **96.49 %**
This matches accuracy printed in Line 54 using accuracy_score.
Precision = 42 / 42 + 3
= **93.33 %**
Same is predicted in our code in line 55

Recall = 42 / 42 + 1
= **97.67 %**
Predicted in our code in line 56.

TASK 2 (Plan Of Work 5)

Using cross_val_score and cross_val_predict, We can see the accuracy score printed from the output of cross_val_score to be in the range 92% to 97%. There is not much fluctuation in 2 samples, however the third one is a bit off. This is still ok since this was a small dataset of just 569 values. Also we could try with other classifiers to see which one would be a best fit since this fluctuates a little when cross_val_Score uses different subset of the data.

Also we can see the accurace,recall,and precision scores of the cross_val_predict to be in the same range as of the classifier, Hence we can see our classifier is efficient and might not be much sensitive to different datasets.

TASK 3 (Plan Of Work 6)

We now plot the ROC curve for our classifier.
We take the positive class as Malignant (1) and Negative class as Benign(0)
We take the cross_val_predict on the training sample to get our predictions using decision function. We the use this value to get out FRP, TPR and Threshold (Line 87)

The Final ROC curve is to be referred to from the folder Plotted_curves in the .zip name "**roc_curve_plot_training.png**"

Additionally, I plotted the graph for Test data on the classifiers predicted values. However being very less deviant, the values are limited, hence plotting a more steep L curve.