Assignment 1

Logistic Regression and AdaBoost for Classification

Najrin Sultana
Student ID : 1605042

How to Run :

```
python 1605042.py <path_to_dataset_1>
<path_to_train_dataset_2> <path_to_test_dataset_2>
<path_to_dataset_3>
```

Inside the **main** function code for running experiments on three datasets are located in three different sections. Hence to run the experiment on a specific dataset, rest of the sections should be commented out.

The **train** function runs logistic regression on the given dataset and returns the hypothesis parameters. Given the features and hypothesis parameters, the **predict** function returns the predictions and given the original label and predictions, the **compute_metric** function computes the necessary metrics. For the sake of comfortable visualization, all the outputs are written to a file called “out.txt” in the same folder the script is being run.

Dataset 1:

Logistic Regression :

Performance measure	Training	Test
Accuracy	0.7965921	0.80908445
Sensitivity	0.5266272	0.56034482
Specificity	0.8964259	0.89066918
Precision	0.6528117	0.62700964
False discovery rate	0.3471882	0.37299035
F1 score	0.5829694	0.59180576

Adaboost :

Number of boosting rounds	Training	Test
5	0.7955271	0.81263307
10	0.7946396	0.81050390
15	0.7951721	0.81192334
20	0.7951721	0.81192334

Dataset 2:

Logistic Regression :

Performance measure	Training	Test
Accuracy	0.8245754	0.82666912
Sensitivity	0.5466139	0.54524180
Specificity	0.9127427	0.91371129
Precision	0.6652180	0.66151419
False discovery rate	0.3347819	0.33848580
F1 score	0.6001120	0.59777651

Adaboost :

Number of boosting rounds	Training	Test
5	0.8449986	0.8443584
10	0.8453364	0.8452797
15	0.8453364	0.8452797
20	0.8453364	0.8452797

Dataset 3:

Logistic Regression :

Performance measure	Training	Test
Accuracy	0.99581527	0.996413628
Sensitivity	0.84061696	0.854368932
Specificity	0.99888234	0.999389747
Precision	0.93696275	0.967032967
False discovery rate	0.06303724	0.032967032
F1 score	0.88617886	0.907216494

Adaboost :

Number of boosting rounds	Training	Test
5	0.99586509	0.996612871
10	0.99591491	0.996612871
15	0.99591491	0.996612871
20	0.99591491	0.996612871

Observation :

As the third dataset is highly unbalanced and the positive samples were very rare, as suggested I took all the positive samples and around **25k** negative samples **(50 times the positive samples)** and then shuffled and splitted into train test sets. For this skewness in the dataset, the trained model will more likely predict a sample to be negative. Since the test dataset also has this skewness and has a very high proportion of negative samples, the accuracy is very high. If the negative samples are taken around **2500 (5 times the positive samples)**, the accuracy slightly decreases. All the metrics are given below:

Train Set :
Accuracy : 0.9699279966116052
Sensitivity : 0.8221649484536082
Specificity : 0.9989863152559554
Precision : 0.9937694704049844
False discovery rate : 0.006230529595015576
F1 : 0.8998589562764457

Test Set :
Accuracy : 0.9763113367174281
Sensitivity : 0.8653846153846154
Specificity : 1.0
Precision : 1.0
False discovery rate : 0.0
F1 : 0.9278350515463918