Machine Learning Assignment Report

Sk. Sabit Bin Mosaddek (1805106)

How to run:

To run, we can follow the following format,

Dataset1:

```
python3 ./1805106.py --dataset_no 1 \
                     --input_file 'dataset1/WA_Fn-UseC_-Telco-Customer-Churn.csv' \
                     --epochs 1000 \
                     --number_of_learners 5 \
                     --feature_count 20 \
                     --mini_batch_size 100 \
                     --missing_value 'mean' \
                     --learning_rate 100 \
                     --k_fold 1 \
                     --dataset_size 10000
```

Dataset2 (here the train and test files path are concatenated with '$'):

```
python3 ./1805106.py --dataset_no 2 \
                     --input_file 'dataset2/adult.data$dataset2/adult.test' \
                     --epochs 100 \
                     --number_of_learners 5 \
                     --feature_count 45 \
                     --mini_batch_size 1000 \
                     --missing_value 'mean' \
                     --learning_rate 100 \
                     --k_fold 1 \
                     --dataset_size 40000
```

Dataset 3:

```
python3 ./1805106.py --dataset_no 3 \
                     --input_file 'dataset3/creditcard.csv' \
                     --epochs 100 \
                     --number_of_learners 5 \
                     --feature_count 13 \
                     --mini_batch_size 10000 \
                     --missing_value 'mean' \
                     --learning_rate 100 \
                     --k_fold 1 \
                     --dataset_size 20000 \
                     --all_positive True
```

Result Analysis

Dataset 1:

Performance measure of Logistic Regression implementation:

Performance measure	Training	Test
Accuracy	0.797835	0.821859
True positive rate (sensitivity, recall, hit rate)	0.521509	0.527933
True negative rate (specificity)	0.899103	0.921979
Positive predictive value (precision)	0.654485	0.697417
False discovery rate	0.345515	0.302583
F1 score	0.580479	0.600954

The **accuracy** of Adaboost implementation with Logistic Regression:

Number of boosting rounds	Training	Test
5	0.791090	0.816182
10	0.792865	0.819021
15	0.791622	0.816182
20	0.792155	0.816891

Observation: Logistic Regression may have fit the model well enough and when we introduced more learners using adaboost, it may happen that some bad data has diverted the result.

Dataset 2:

performance measure of Logistic Regression implementation:

Performance measure	Training	Test
Accuracy	0.829059	0.833241
True positive rate (sensitivity, recall, hit rate)	0.484377	0.489860
True negative rate (specificity)	0.938390	0.939445
Positive predictive value (precision)	0.713776	0.714448
False discovery rate	0.438666	0.285552
F1 score	0.577116	0.581212

The **accuracy** of Adaboost implementation with Logistic Regression:

Number of boosting rounds	Training	Test
5	0.829827	0.832811
10	0.832192	0.836251
15	0.833789	0.836988
20	0.833543	0.835268

Dataset 3:

performance measure of Logistic Regression implementation:

Performance measure	Training	Test
Accuracy	0.994650	0.995401
True positive rate (sensitivity, recall, hit rate)	0.760814	0.777778
True negative rate (specificity)	0.999337	0.999796
Positive predictive value (precision)	0.958333	0.987179
False discovery rate	0.041667	0.012821
F1 score	0.848227	0.870056

The **accuracy** of Adaboost implementation with Logistic Regression:

Number of boosting rounds	Training	Test
5	0.994650	0.995401
10	0.994650	0.995401
15	0.994650	0.995401
20	0.994650	0.995401

Observation

After completing the assignment, I have the following observations:

- Increasing epoch and minibatch size will never hurt the model
- Increasing feature_count will improve at first and after a certain point it will start to hurt the model
- The better the model, the effect of adaboost starts to decrease. We can see that in dataset 1 where adaboost seems to affect negatively on the performance. Dataset 3 is saturated with the model for which adaboost couldn't provide any improvement. Only dataset 2 is improved with adaboost.
- Changing learning rate affects the performance