Compsci 361 Assignment 2

Hasnain Cheena
190411106
hche737

Part A

Three runs of the experiment were conducted. Within each run the order the algorithms is changed. The results of each run are shown below:

Run 1

```
Dataset                  (1) trees.RandomFo | (2) meta.AdaBoo (3) meta.Baggin
------------------------------------------------------------------------------
iris                    (100)   94.67(5.01) |   95.40(5.74)     94.47(5.69)
car                     (100)   94.67(1.74) |   70.02(0.16) *   91.66(2.13) *
balance-scale           (100)   81.48(3.62) |   71.77(4.24) *   83.68(3.65)
------------------------------------------------------------------------------
                                    (v/ /*) |         (0/1/2)         (0/2/1)
```

Run 2

```
Dataset                  (1) meta.AdaBoostM | (2) meta.Baggin (3) trees.Rando
------------------------------------------------------------------------------
iris                    (100)   95.40(5.74) |   94.47(5.69)     94.67(5.01)
car                     (100)   70.02(0.16) |   91.66(2.13) v   94.67(1.74) v
balance-scale           (100)   71.77(4.24) |   83.68(3.65) v   81.48(3.62) v
------------------------------------------------------------------------------
                                    (v/ /*) |         (2/1/0)         (2/1/0)
```

Run 3

```
Dataset                  (1) meta.Bagging ' | (2) trees.Rando (3) meta.AdaBoo
------------------------------------------------------------------------------
iris                    (100)   94.47(5.69) |   94.67(5.01)     95.40(5.74)
car                     (100)   91.66(2.13) |   94.67(1.74) v   70.02(0.16) *
balance-scale           (100)   83.68(3.65) |   81.48(3.62)     71.77(4.24) *
------------------------------------------------------------------------------
                                    (v/ /*) |         (1/2/0)         (0/1/2)
```

Part B

Iris dataset

We have no statistical evidence (at a 5% significance level) of a difference in performance between the AdaBoost, RandomForest and BaggingTree algorithms on the iris dataset. All three algorithms performed similarly well. This indicates a strong correlation between the attributes and labels within the underlying data.

Moreover, the iris dataset must have a low amount of mislabelled data and outliers/label noise. This is because AdaBoost was able to weight instances appropriately, correcting for true errors.

Further, the attributes in the iris dataset are of reasonably similar predictive strength. This is because if there was a large discrepancy in the predictive strength of the attributes the BaggingTree algorithm would have performed poorly.

Car dataset

We have statistical evidence (at a 5% significance level) as shown by the paired t-test that RandomForest performed the best, followed by BaggingTree and then AdaBoost.

The AdaBoost algorithm did not perform well hinting there must be significant label noise and/or data labelling errors in the underlying data. This is because if positive instances are labelled negative (due to error or label noise) each iteration of AdaBoost will continue to incorrectly increase the weights on these instances. Therefore, these error/outlier datapoints end up having strong influence over the final model. This causes the incorrect predictions leading to a lower accuracy compared to RandomForest and BaggingTree.

RandomForest performed the best among all three algorithms. This is because of RandomForest's feature bagging characteristic. This method randomly selects a subset of the predictors to train each tree in the ensemble. Thus, each tree in the ensemble created by the RandomForest was more different than in the BaggingTree algorithm. This means the trees within the RandomForest ensemble were able to correctly classify a relatively bigger pool of cases leading to higher accuracy.

In comparison, the BaggingTree algorithm was affected by the underlying discrepancy between the strength of predictors in the dataset. The car dataset must contain a few attributes with very strong predictive power relative to the other attributes. The BaggingTree algorithm considers all the attributes available at each split in the tree, thus the same strong predictors will be used at the top of each tree. Therefore, each tree is similar to one another meaning the ensemble is able to correctly classify a smaller pool of cases than the RandomForest ensemble, leading to a lower accuracy.

Balance-Scale dataset

We have statistical evidence (at a 5% significance level) as shown by the paired t-test that AdaBoost performed worse than both the RandomForest and BaggingTree algorithms. We have no statistical evidence to differentiate performance between RandomForest and BaggingTree on the Balance-Scale dataset.

The AdaBoost algorithm performed worse than the other two algorithms. This indicates there must be significant label noise and/or mislabelling of data in the dataset. Due to the label noise/errors, AdaBoost will incorrectly weight instances adversely affecting the learning process. These instances have influenced the final model learned, shown by the significant difference in accuracy between AdaBoost and the other two algorithms.

The RandomForest and BaggingTree performed similarly well indicating trees created by both algorithms were different enough from one another to cover a wide variety of instances. Therefore, different combinations of attributes can be used to predict the target labels. This indicates the underlying data must have attributes of similar predictive strength.

Part C

The paired t-test results (shown in the tables in Part A) differ depending on which order the algorithms are run. Each table shows a different view of the results because, in each run, a different algorithm has been used as the baseline for the paired t-test. The accuracy of the baseline algorithm is compared to the accuracies of the other two algorithms to determine whether there is a significant difference in performance and subsequently to conclude which algorithm is better. Thus, because each run gives you pairwise rankings, three runs are needed to evaluate the ranking of all three algorithms.

Part D

I believe RandomForest is more reliable than AdaBoost and the BaggingTree algorithm.

Firstly, RandomForest is more reliable than AdaBoost because of AdaBoost's sensitivity to label noise and/or labelling errors. All data sources will contain errors and noise, so this is a huge disadvantage when using AdaBoost.

Secondly, RandomForest is more reliable than the BaggingTree algorithm. Both BaggingTree and RandomForest perform bagging on the instances but RandomForest additionally performs feature bagging. Thus, the tree's created by RandomForest are typically more different from one another than the tree's created by the BaggingTree algorithm. This means that generally, the ensemble created by the RandomForest will be more diverse, covering a wider variety of cases compared to the ensemble created by the BaggingTree algorithm.