Homework Four Report

Zhiyu Zhu (zzhu24)

Tree Structure:
Binary Tree!!!!

Coding Structure:

Tree Node:

Prediction of Class: decision
All set of attributes as an array: attribute
Left child tree: left_tree (all the sub trees with this certain attribute
Right child tree: right_tree (remained sub trees)

1.Decision Tree:

Algorithm:

```
=> Build the Tree
        => Check for base case:
                => No more attribute
                => No more data
                => All the data are in the same class
        => Calculate Gini Index and split on the attribute with smallest Gini Index
=> Make prediction and calculate the accuracy
```

2.Random Forest:

Using Threads

Algorithm:

```
=> Build the Tree
        => Bagging data according to the bagging proportion size
        => Check for base case:
                => No more attribute
                => No more data
                => All the data are in the same class
        => Stop spreading the tree when it reaches the maximum attribute size to reduce time
        => Calculate Gini Index and split on the attribute with smallest Gini Index
=> Forms N tree and use the majority vote for prediction
=> Calculate the accuracy
```

(Bagging and attribute proportion are chosen from [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9])

Performance and Accuracy:

Data: balance.scale

1.Decision Tree:

Training:

Accuracy= 1.0

Class	1	2	3
Specificity	1.0	1.0	1.0
Recall	1.0	1.0	1.0
Precision	1.0	1.0	1.0
F1 Score	1.0	1.0	1.0
F (Beta = 0.5) Score	1.0	1.0	1.0
F (Beta = 0.2) Score	1.0	1.0	1.0

Testing:

Accuracy = 0.711

Class	1	2	3
Specificity	0.8818	0.8293	0.8387
Recall	0.0	0.8529	0.7228
Precision	0.0	0.8056	0.7849
F1 Score	0.0	0.8256	0.7526
F (Beta = 0.5) Score	0.0	0.8146	0.7717
F (Beta = 0.2) Score	0.0	0.843	0.7344

The training accuracy is 1.0 and the tree has no need to be pruned.
The accuracy is only 0.711 in the testing case so we need to use ensemble methods.

2.Random Forest:

Training:

Accuracy= 0.8875

Class	1	2	3
Specificity	1.0	0.8645	0.9247
Recall	0.0	0.9731	0.9305
Precision	0.0	0.8619	0.9185
F1 Score	0.0	0.9141	0.9231
F (Beta = 0.5) Score	0.0	0.8823	0.9187
F (Beta = 0.2) Score	0.0	0.9486	0.275

Testing:

Accuracy= 0.8000

Class	1	2	3
Specificity	1.0	0.8699	0.7661
Recall	0.0	0.8431	0.9307
Precision	0.0	0.8431	0.7642
F1 Score	0.0	0.8431	0.8393
F (Beta = 0.5) Score	0.0	0.8431	0.7926
F (Beta = 0.2) Score	0.0	0.8431	0.8918

Ensemble method of Random forest has 0.8875 accuracy in training since we uses bagging and randomly selecting data.

In the testing case, however, Random forest method increases accuracy to 0.8000 because that the randomly selecting data.

The bagging size proportion is 0.5 for it gives most stable and relatively high accuracy.

Data: led

1.Decision Tree:

Training

Accuracy = 0.8596

Class	1	2
Specificity	0.8958	0.7774
Recall	0.7774	0.8958
Precision	0.7666	0.9014
F1 Score	0.772	0.8995
F (Beta = 0.5) Score	0.7688	0.9003
F (Beta = 0.2) Score	0.7752	0.8969

Testing

Accuracy = 0.8554

Class	1	2
Specificity	0.8889	0.7806
Recall	0.7806	0.8889
Precision	0.7590	0.9004
F1 Score	0.7697	0.8906
F (Beta = 0.5) Score	0.7632	0.8961
F (Beta = 0.2) Score	0.7762	0.8912

The training accuracy is 0.8596 and the tree has no need to be pruned. Since it has very small class number, its' training accuracy is not very closed to 1.0.

The accuracy is 0.8554 in the testing case and very closed to the accuracy of the training cases.

2.Random Forest:

Training

Accuracy = 0.8481

Class	1	2
Specificity	0.9518	0.7618
Recall	0.7618	0.9518
Precision	0.882	0.8942
F1 Score	0.8175	0.9221
F (Beta = 0.5) Score	0.855	0.9051
F (Beta = 0.2) Score	0.7831	0.9376

Testing

Accuracy = 0.8615

Class	1	2
Specificity	0.9093	0.755
Recall	0.755	0.9093
Precision	0.7887	0.8922
F1 Score	0.7715	0.9007
F (Beta = 0.5) Score	0.7817	0.8965
F (Beta = 0.2) Score	0.7615	0.9059

Ensemble method of Random forest has 0.8481 accuracy in training. It is not closed to 1.0 because the class number is relatively small and in the tree construction, we stop when they are all in the same class.

In the testing case, Random forest method increases accuracy to 0.8615 which is not very apparently improved compared to the pure decision tree method.

The bagging size proportion is 0.5 for it gives most stable and relatively high accuracy.

Data: nursery

1.Decision Tree:

Training
Accuracy = 0.9920

Class	1	2	3	4	5
Specificity	1.0	1.0	1.0	1.0	1.0
Recall	1.0	1.0	1.0	1.0	1.0
Precision	1.0	1.0	1.0	1.0	1.0
F1 Score	1.0	1.0	1.0	1.0	1.0
F (Beta = 0.5) Score	1.0	1.0	1.0	1.0	1.0
F (Beta = 0.2) Score	1.0	1.0	1.0	1.0	1.0

Testing
Accuracy = 0.9914

Class	1	2	3	4	5
Specificity	0.9933	1.0	0.9946	1.0	0.9996
Recall	0.9887	0.9538	0.9883	1.0	0.0
Precision	0.9862	1.0	0.9883	1.0	0.0
F1 Score	0.9875	0.9764	0.9883	1.0	0.0
F (Beta = 0.5) Score	0.9867	0.9904	0.9883	1.0	0.0
F (Beta = 0.2) Score	0.9882	0.9627	0.9883	1.0	0.0

The training accuracy is 0.9920 and the tree has no need to be pruned. This data set has comparably large number of classes.
Therefore the testing accuracy is relatively high as 0.9914. There is no need to prune the tree in this case.

2.Random Forest:

Training

Accuracy = 0.9707

Class	1	2	3	4	5
Specificity	0.9622	1.0	0.9943	1.0	1.0
Recall	0.988	0.5556	0.9537	1.0	0.5
Precision	0.9279	1.0	0.9868	1.0	1.0
F1 Score	0.957	0.7143	0.97	1.0	0.6667
F (Beta = 0.5) Score	0.9393	0.8621	0.98	1.0	0.8333
F (Beta = 0.2) Score	0.9754	0.6098	0.9602	1.0	0.5556

Testing

Accuracy = 0.946

Class	1	2	3	4	5
Specificity	0.931	0.9994	0.9904	1.0	0.9996
Recall	0.9781	0.5909	0.8867	1.0	0.0
Precision	0.8735	0.963	0.977	1.0	0.0
F1 Score	0.9228	0.7324	0.9297	1.0	0.0
F (Beta = 0.5) Score	0.8926	0.8553	0.9575	1.0	0.0
F (Beta = 0.2) Score	0.9552	0.6404	0.9034	1.0	0.0

Ensemble method of Random forest has 0.9707 accuracy in training. It is very closed to 1.0 and the accuracy is very stable.
In the testing case, Random forest method increases accuracy to 0.946 which is not very apparently improved compared to the pure decision tree method.
The bagging size proportion is 0.5 for it gives most stable and relatively high accuracy.

Data: synthetic.social

1.Decision Tree:

Training

Accuracy = 1.0

Class	1	2	3	4
Specificity	1.0	1.0	1.0	1.0
Recall	1.0	1.0	1.0	1.0
Precision	1.0	1.0	1.0	1.0
F1 Score	1.0	1.0	1.0	1.0
F (Beta = 0.5) Score	1.0	1.0	1.0	1.0
F (Beta = 0.2) Score	1.0	1.0	1.0	1.0

Testing

Accuracy = 0.481

Class	1	2	3	4
Specificity	0.8238	0.8146	0.832	0.8376
Recall	0.4851	0.4204	0.5172	0.502
Precision	0.5019	0.4239	0.4819	0.5141
F1 Score	0.4934	0.4221	0.499	0.5079
F (Beta = 0.5) Score	0.4985	0.4232	0.4886	0.5116
F (Beta = 0.2) Score	0.4884	0.4211	0.5098	0.5043

In the training case, the accuracy is 1.0 but the testing accuracy is only 0.481.
The number of attribution is large and the tree is very large. The decision tree method is time consuming and also poorly accurate.

2.Random Forest:

Training

Accuracy = 0.917

Class	1	2	3	4
Specificity	0.9757	0.9715	0.9718	0.9619
Recall	0.8975	0.8781	0.9193	0.9477
Precision	0.9228	0.912	0.9181	0.8914
F1 Score	0.91	0.8947	0.9187	0.9187
F (Beta = 0.5) Score	0.9176	0.905	0.9183	0.9021
F (Beta = 0.2) Score	0.9025	0.8847	0.919	0.9358

Testing

Accuracy = 0.659

Class	1	2	3	4
Specificity	0.9085	0.8954	0.8711	0.8711
Recall	0.6082	0.5918	0.7284	0.7137
Precision	0.7087	0.6473	0.6306	0.6547
F1 Score	0.6546	0.6183	0.676	0.6829
F (Beta = 0.5) Score	0.686	0.6354	0.648	0.6657
F (Beta = 0.2) Score	0.626	0.6022	0.7065	0.7011

According to the decision tree method, we know that the tree need to be pruned. In the Random Forest Method, the bagging proportion is 0.1 for it gives most stable and relatively high accuracy. The attribute proportion is 0.2. We randomly chose 0.2 of all attributes and stop the tree when it reaches the limitation so that it will not take very long time. The accuracy is increased to 0.659 in the testing case.