EECS 440 Machine Learning Programming Problem 4 Writeup

Sixiao Zhang sxz603, Jianzhe Zhang jxz851, Junyi Zheng jxz990

For all problems, we use decision tree for the algorithm and voting for the dataset. Here is an example for our program output. It's the output of the ffeatsel when number of features is 3.

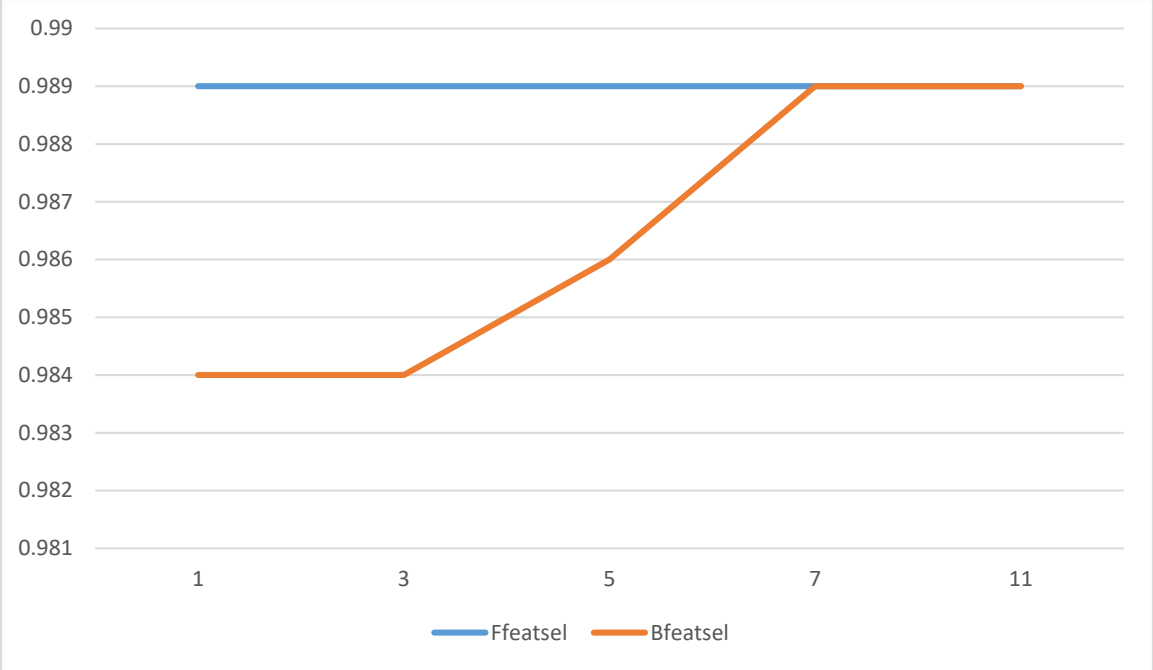
```
Algorithm: Decision Tree
Time consumed: 2.195446252822876 s
Size: 12 ,Maximum Depth: 3 ,First Feature: Repealing-the-Job-Killing-Health-Care-Law-Act
selected_features:  1 4 2
Classifier 2:
Accuracy: 0.989
Precision: 1.000
Recall: 0.974

Algorithm: Decision Tree
Time consumed: 2.3463218212127686 s
Size: 12 ,Maximum Depth: 3 ,First Feature: Repealing-the-Job-Killing-Health-Care-Law-Act
selected_features:  1 4 2
Classifier 3:
Accuracy: 1.000
Precision: 1.000
Recall: 1.000

Algorithm: Decision Tree
Time consumed: 2.067645311355591 s
Size: 10 ,Maximum Depth: 2 ,First Feature: Repealing-the-Job-Killing-Health-Care-Law-Act
selected_features:  1 3 2
Classifier 4:
Accuracy: 0.977
Precision: 1.000
Recall: 0.949

Algorithm: Decision Tree
Time consumed: 2.297356367111206 s
Size: 13 ,Maximum Depth: 3 ,First Feature: Repealing-the-Job-Killing-Health-Care-Law-Act
selected_features:  1 2 3
Classifier 5:
Accuracy: 0.978
Precision: 0.975
Recall: 0.975

Avg Time Consumed: 2.256 s
Accuracy: 0.989 0.010
Precision: 0.995 0.010
Recall: 0.980 0.019
Area under ROC: 0.986
```

We develop 5 classifiers according to the 5-fold cross validation, and for each classifier, we output the time consumed for the particular fold, and also output the size, depth, first feature, the features that we select, accuracy, precision, and recall of the final tree. The five lines at the very bottom are the average result of the 5-fold cross validation. We use these outputs to analysis the performance.

(a) Examine the effect of the number of features selected on the accuracy of the resulting classifier. Do this by plotting the accuracy of an algorithm using at least four different values of features retained on the x axis and the accuracy on the y axis. As the fifth point on the x axis, use all the features. Discuss the resulting graphs.

Table. 1 Accuracies of different number of features

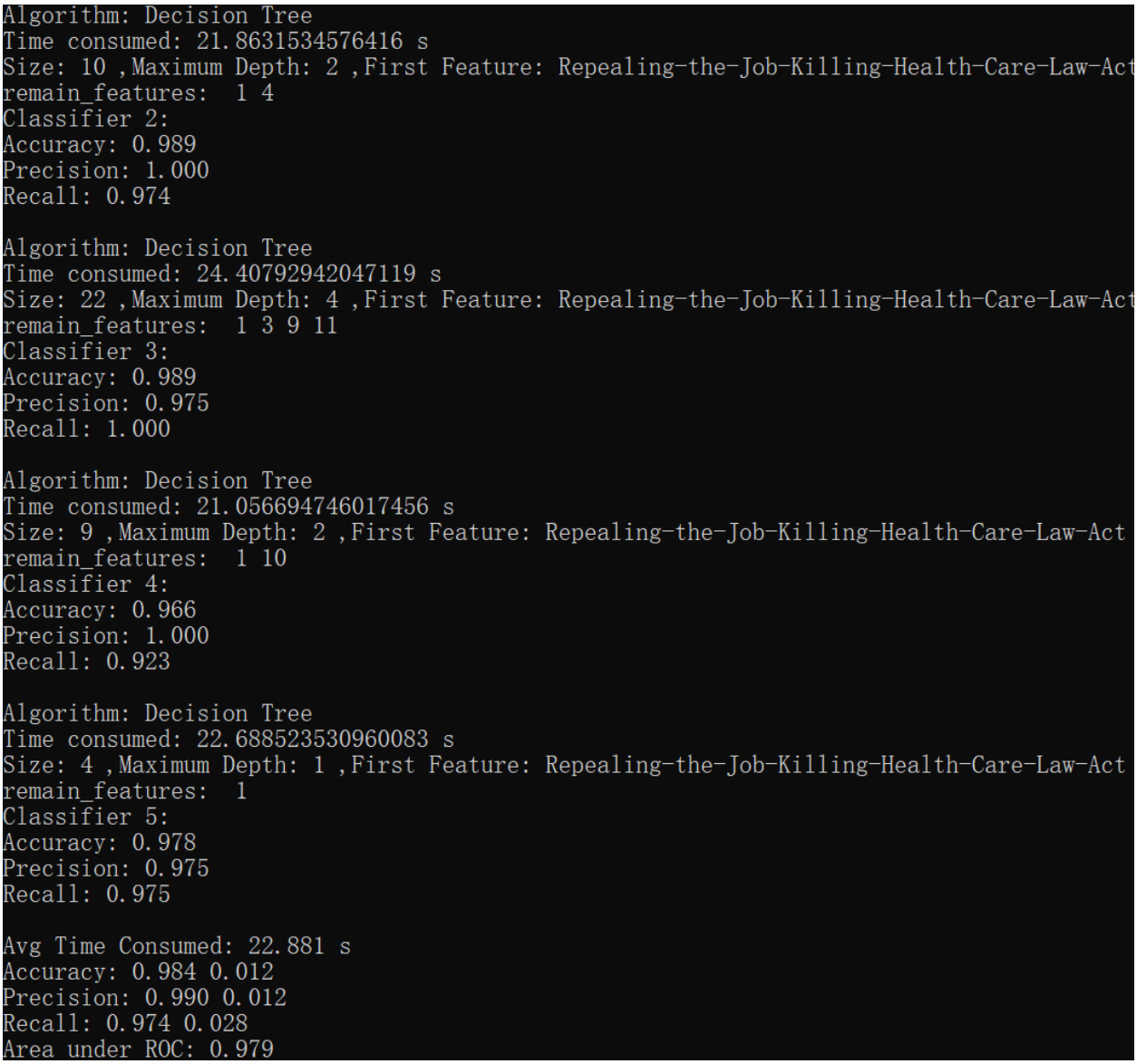
Features	1	3	5	7	11
Ffeatsel	0.989	0.989	0.989	0.989	0.989
Bfeatsel	0.984	0.984	0.986	0.989	0.989



From the plot, it can be concluded that the more features, the better accuracy. But because the voting dataset can be classified very well just by one particular feature called Repealing-the-Job-Killing-Health-Care-Law-Act, so this conclusion might not be very convincing from the above results, but we believe that this is true for most datasets.

(b) Compare the accuracy of the classifiers obtained with forward and backward feature selection using the graphs from (a). Do you notice a difference between the two?

From the plot, we can see that the accuracy of ffeatsel is always 0.989, and the accuracy of bfeatsel increase from 0.984 to 0.989. The reason for this might be like this. We see that, for bfeatsel when number of features is 1, the result is this.



In this image, we can see that the remaining features are sometimes more than 1. This is because that the algorithm finds that, if it drops one more feature, the accuracy would become worse, so it stops. But for ffeatsel, it would only have 1 feature. This might be the cause of the different accuracy.

(c) Compare the accuracy of forward and backward feature selection to boosted feature selection, which involves running boosting with decision stumps for *k* iterations, where *k* is the desired number of features.

Table. 2 Accuracies of different number of features including boost

Features	1	3	5	7	11
Ffeatsel	0.989	0.989	0.989	0.989	0.989
Bfeatsel	0.984	0.984	0.986	0.989	0.989
Boost	0.989	0.989	0.989	0.989	0.989

From Table 2 we can see that the accuracy of boost on voting is the same with ffeatsel, both very high accuracy.

(d) Discuss the differences in average runtime between the three feature selection approaches for a given number of features to be selected. In this runtime, do *not* include the time to train the final classifier. Which approach has the best combination of speed and accuracy in your opinion?

We test the runtime of each fold and average them across the 5 folds and get the results shown in Table 3.

Table. 3 Runtime of different number of features including boost

Features	1	3	5	7	11
Ffeatsel	0.387s	2.256s	5.173s	8.453s	12.346s
Bfeatsel	22.881s	23.002s	20.886s	18.098s	0.000s
Boost	0.388s	1.691s	3.417s	5.162s	7.493s

The 0.000s for bfeatsel when number of features is 11 is because that the program is written to do no feature selection when we select all the features.

We can see that the runtime of bfeatsel is the most, while the boost is the least. So consider the runtime and the accuracy together, the boost is the best approach. But because we implement the algorithms on the voting dataset, there is no significant difference on accuracy. Maybe the accuracy of forward feature selection is better that boost on most datasets, in this way forward feature selection might be better than boost, although the runtime is more than boost.