Rajesh Sakhamuru May 25, 2020

CS 6140 Assignment 1

1.1 a

Iris Binary Decision Tree

η_{min} Value	Average Accuracy	Std. Deviation
0.05	0.753	0.090
0.10	0.727	0.141
0.15	0.760	0.091
0.20	0.733	0.130

1.1 b

Spambase Binary Decision Tree

η_{min} Value	Average Accuracy	Std. Deviation
0.05	0.853	0.028
0.10	0.850	0.042
0.15	0.858	0.060
0.20	0.850	0.045
0.25	0.853	0.072

1.2 a

```
Confusion Matrix for nMin=0.15: (Columns are Actual, Rows are Predicted)
                 Iris-setosa  Iris-versicolor  Iris-virginica
Iris-setosa               50                0               0
Iris-versicolor            0               15               1
Iris-virginica             0               35              49
```

The Confusion Matrix has been populated using 10-fold cross validation at an nMin (threshold) of 0.15. We picked this threshold because it has the highest average accuracy over the 10 folds and the standard deviation is also on the lower side compared to the other thresholds.

From this matrix it is clear that while our decision tree predicts Iris-setosa and Iris-virginica with high accuracy, the majority of Iris-versicolor flowers are misclassified as Iris-virginica. This may be attributable to the method of classification that we used, to change all of the 4 features into binary features. Binary splits with just a few (4) features, when trying to categorize into 3 classes, is the reason for this high prediction error. Perhaps this type of thing could be improved by picking 2 splits, and converting each of the features into Tertiary splits, although that process would take exponentially longer test all possible combinations of 2 splits (so 3 classes per feature) for which splits give the best information gain.

1.2 b

```
Confusion Matrix for nMin=0.15: (Columns are Actual, Rows are Predicted)
      0     1
0  2335   199
1   453  1614
```

The Confusion Matrix has been populated using 10-fold cross validation at an nMin (threshold) of 0.15. We picked this threshold because it has the highest average accuracy over the 10 folds, although all of the accuracy values are within 1% of each other (85.8%) and its standard deviation is on the higher end compared to the other nMin values.

Around 11% of spam was misclassified as non-spam (false-negative) and around 16% of non-spam were misclassified as spam (false-positive). The overall accuracy of this model is markedly increased compared to the iris flowers decision tree dude to a big increase in the number of features and a huge increase in the amount of data being used, both of which increase overall information gain. Additionally, due to using binary splits, this model is not adversely affected like the iris-flower classifier, because there are only 2 target classes, so binary splits are more suited to this type of classification.

1.2 c

For the iris dataset and the spambase dataset, the nMin which gave the best result was 0.15. For each dataset, all accuracy averages across the nMin thresholds are within 1 standard deviation of each other and therefore the differences between them are not statistically significant.

However, ignoring standard deviation, both peak at nMin 0.15. This peak could indicate where there is a balance between overfitting and underfitting the training data in each fold. At a lower nMin, the leaf nodes of the decision tree are smaller, and therefore could be prone to overfitting, or essentially just memorizing the training data, rather than generalizing it for the model. At a higher nMin, with very large leaf nodes, the decision tree could be prone to underfitting, not clearly capturing the nuances between the classification categories.

Perhaps through increasing the quantity of data available, we can increase the number of folds as well, we could possibly get lower standard deviations at each threshold level to more precisely locate which threshold level best fits both the training and testing data.

2.1 a, b

Mushroom Multiway Decision Tree

η_{min} Value	Average Accuracy	Std. Deviation
0.05	0.996	0.002
0.10	0.997	0.001
0.15	0.997	0.002

Mushroom Binary Decision Tree

η_{min} Value	Average Accuracy	Std. Deviation
0.05	0.998	0.001
0.10	0.998	0.001
0.15	0.998	0.001

While the difference in accuracy between the multiway and binary decision trees is not statistically significant based on the standard deviations, we can see that there is likely some improvement in accuracy by switching to binary trees. This is because across the board the average accuracy is higher than with multiway decision trees and standard deviation is either the same or lower.

The reason the binary splits may have an advantage in accuracy compared to the trees with multiway splits, is because multiway splits can too quickly break the data into smaller subsets. This rapid splitting could result in a bias towards features with more classes, since they are more likely to give purer nodes resulting in overfitting. Binary trees are able to avoid this shortcoming.

2.2 a

```
MULTIWAY Confusion Matrix for nMin=0.10: (Columns are Actual, Rows are Predicted)
      e     p
e  4200    15
p     8  3901
```

This confusion matrix for a multiway decision tree for the mushroom data, was 99.7% accurate with a standard deviation of .1%. Edible mushrooms were incorrectly classified only .2% of the time, and poisonous mushrooms were incorrectly classified only .4% of the time. The matrix tells me that this is decision tree model can very accurately predict whether a mushroom is edible or poisonous based on the given features.

```
BINARY Confusion Matrix for nMin=0.10: (Columns are Actual, Rows are Predicted)
      e     p
e  4193     4
p    15  3912
```

This confusion matrix for a binary decision tree for the mushroom data, was 99.8% accurate with a standard deviation of .1%. Edible mushrooms were incorrectly classified only .4% of the time, and poisonous mushrooms were incorrectly classified only .1% of the time. The matrix tells me that this decision tree model can very accurately predict whether a mushroom is edible or poisonous based on the given features just like the multiway tree.

2.2 b

For both multiway and binary splits, there is little difference between the different values of nMin so it becomes difficult to pick an optimal nMin value between the ones tested. For the multiway tree, I choose nMin = 0.1 as optimal because it has both the highest accuracy and lowest standard deviation by very small margins. But, for the binary tree all 3 of the nMins tested had the same accuracies and standard deviations, so there really is no optimal nMin value.

This lack of change in accuracy between values of nMin could indicate that the classification of mushrooms into edible vs poisonous is reliant on a very few number of features for both binary and multiway trees, and therefore does not need very deep trees to achieve high accuracy. Much higher depth (nMin < 0.005) would probably result in significant overfitting because in order to generalize the data using a decision tree, it should not just be essentially memorizing the training data.

6 a

Housing Binary Regression Tree

η_{min} Value	Total Sum of Squared Errors	Average SSE Across Folds	Std. Deviation
0.05	3474.317	347.432	176.365
0.10	3493.626	349.363	198.813
0.15	3467.246	346.725	131.946
0.20	3481.598	348.160	111.050

6 b

The best nMin value for the regression tree is nMin=0.20. The total and average SSEs are all within 1% of each other, but standard deviation went down significantly at nMin=0.20. The low standard deviation indicates that the SSEs across the folds is more consistently near the average than in other nMin/threshold levels.

The probably causes of nMin being a significant impact on standard deviation, is that in regression trees the leaf nodes are averages of the leaf node data, so if we overfit with lower nMin values (like nMin=0.05 or 0.10), there are less data points in the leaf node and therefore more likely fluctuation in standard deviation of SSE across the folds.

As nMin increases, it will reach a certain point where the tree begins underfitting instead, but this point is not reached before nMin=0.20, so I cannot speculate what point that might be.