CS 6140: Machine Learning Spring 2018

Assignment 1

Decision Tree

1. Classification Trees With Numerical Features

1.1 Growing Decision Trees

a. Iris Dataset

n_{min}	Average Accuracy	Standard Deviation
0.05	95.95238095	4.647433642
0.10	95.95238095	4.647433642
0.15	95.95238095	4.647433642
0.20	95.95238095	4.647433642

b. Spambase Dataset

n_{min}	Accuracy	Standard Deviation
0.05	90.59382423	1.414049983
0.10	89.0736342	1.735755431
0.15	86.27078385	2.407852902
0.20	86.152019	2.391984139
0.25	83.2304038	2.216038725

1.2 Interpreting the results

a. Iris Dataset

Best n_{min} = 0.05

Predicted	Iris-setosa	Iris-versicolor	Iris-virginica	__all__

Actual				
Iris-setosa	48	0	0	48
Iris-versicolor	0	49	1	50
Iris-virginica	0	5	44	49
__all__	48	54	45	147

Accuracy: 0.9591836734693877

Inference:

Classes	Iris-setosa	Iris-versicolor	Iris-virginica
P: Condition positive	48	50	49
N: Condition negative	99	97	98
TP: True Positive	48	49	44
TN: True Negative	99	92	97
FP: False Positive	0	5	1
FN: False Negative	0	1	5
TPR: (Sensitivity, hit rate, recall)	1	0.98	0.897959
TNR=SPC: (Specificity)	1	0.948454	0.989796
PPV: Pos Pred Value (Precision)	1	0.907407	0.977778
NPV: Neg Pred Value	1	0.989247	0.95098
FPR: False-out	0	0.0515464	0.0102041
FDR: False Discovery Rate	0	0.0925926	0.0222222
FNR: Miss Rate	0	0.02	0.102041
FOR: False omission rate	0	0.0107527	0.0490196
ACC: Accuracy	1	0.959184	0.959184
F1 score	1	0.942308	0.93617
MCC: Matthews correlation coefficient	1	0.912416	0.908025
Informedness	1	0.928454	0.887755
Markedness	1	0.896655	0.928758

b. Spambase Dataset

Best n_{min} = 0.05

Predicted Actual	Spam	Not Spam	__all__
Spam	1478	195	1673
Not Spam	201	2336	2437
__all__	1679	2531	4210

Accuracy: 90.593824228

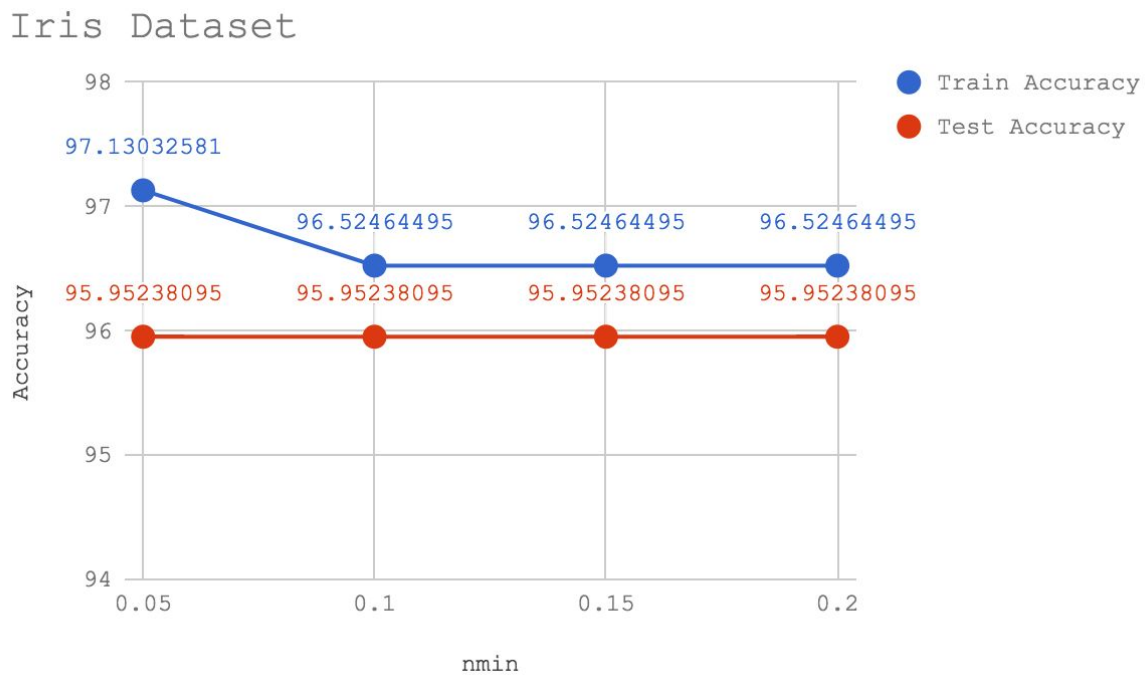
Inference:

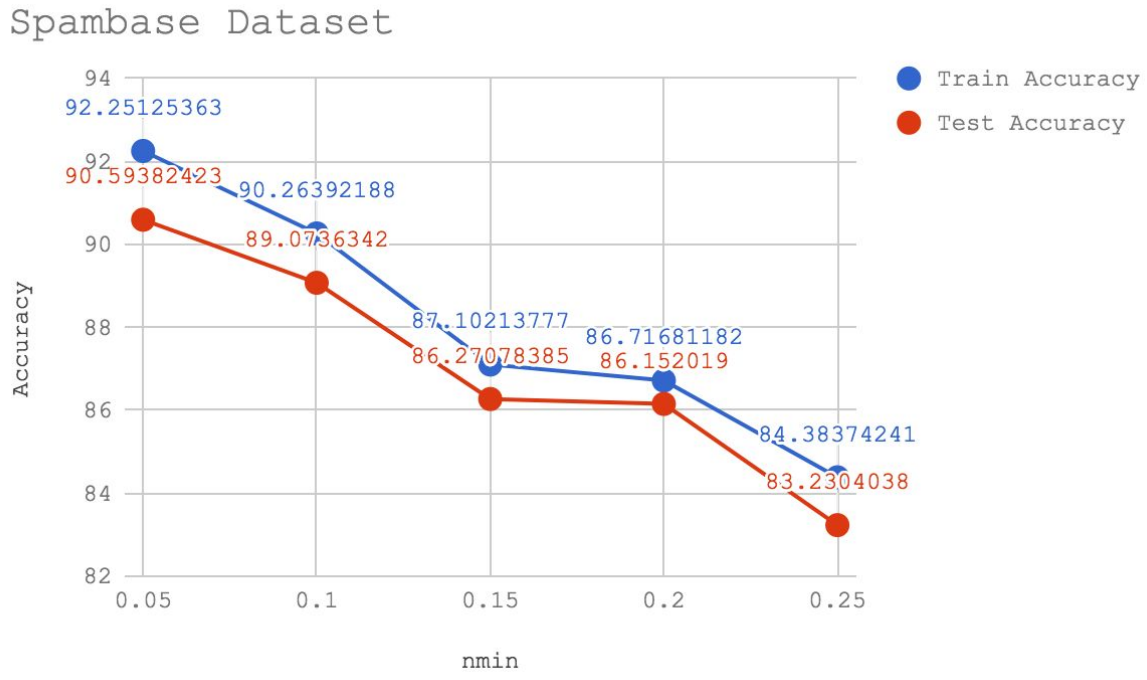
Classes	Values
P	1679
N	2531
TP	1478
TN	2336
FP	195
FN	201
TPR	0.8802858845
TNR	0.9229553536
PPV	0.8834429169
NPV	0.920772566
FPR	0.07704464638
FDR	0.1165570831
FNR	0.1197141155
ACC	0.9059382423
F1_score	0.8818615752
MCC	0.8037282129
informedness	0.8032412381
markedness	0.8042154829

c. N_{min}

As N_{min} value increases the accuracy of the classifier decreases. This claim can be supported by the tables in 1.1. As we increase the value of the N_{min} the classifier stops splitting the data and approximates the label. This can be clearly seen in the result table at 1.1.

d. Training and Testing Accuracy





In both the datasets there is clear evidence of overfitting. This is because the classifier is trained by maximizing its accuracy on the training dataset. The classifier memorizes the training data so it does better on training data but while it tries to predict on test data it tries to apply the same memorization technique which fails due to unknown data.

2. Classification Trees With Categorical Features

2.1 Multiway vs Binary Decision Trees

a. Multiway Decision Tree

n_{min}	Accuracy	Standard Deviation
0.05	99.7046749734	0.18406119385
0.10	99.4092895779	0.30027570125
0.15	99.4092895779	0.30027570125

b. Binary Decision Trees

n_{min}	Accuracy	Standard Deviation
0.05	51.7978484497	0.968720629586
0.10	51.7978484497	0.968720629586
0.15	51.7978484497	0.968720629586

Replacing categorical feature with binary features does not improve the accuracy of the classifier. It introduces a lot of attributes and thus resulting in minor subsets leading it to be not split further. Thus resulting in decrease in the accuracy.

2.2 Interpreting the results

a. N_{min}

Multiway Decision Tree

Best n_{min} = 0.05

Predicted Actual	e	p	__all__
e	4200	16	4216

p	8	3900	3908
__all__	4208	3916	8124

Accuracy: 99.70457902511078

Inference:

Classes	Values
P	4208
N	3916
TP	4200
TN	3900
FP	16
FN	8
TPR	0.9980988593
TNR	0.9959141982
PPV	0.9962049336
NPV	0.9979529171
FPR	0.004085801839
FDR	0.003795066414
FNR	0.001901140684
ACC	0.9970457903
F1_score	0.9971509972
MCC	0.9940854514
informedness	0.9940130575
markedness	0.9941578507

Binary Decision Tree

Best n_{min} = 0.05

Predicted Actual	0	1	__all__
0	0	3916	3916
1	0	4208	4208
__all__	0	8124	8124

Accuracy: 51.7971442639094

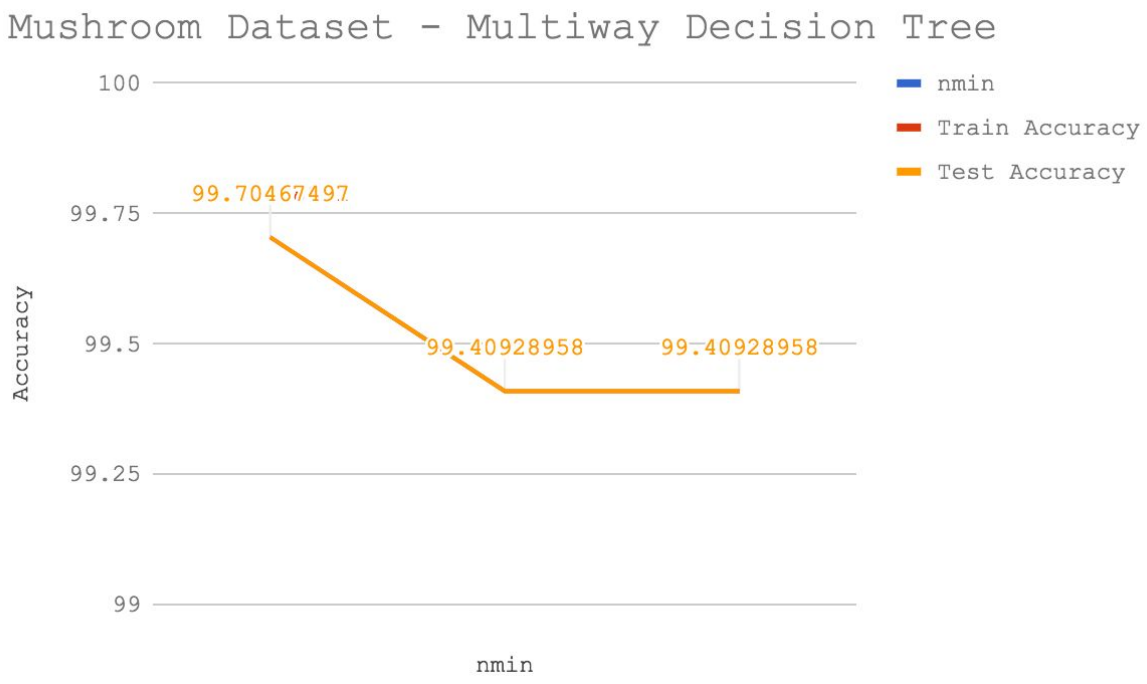
Inference:

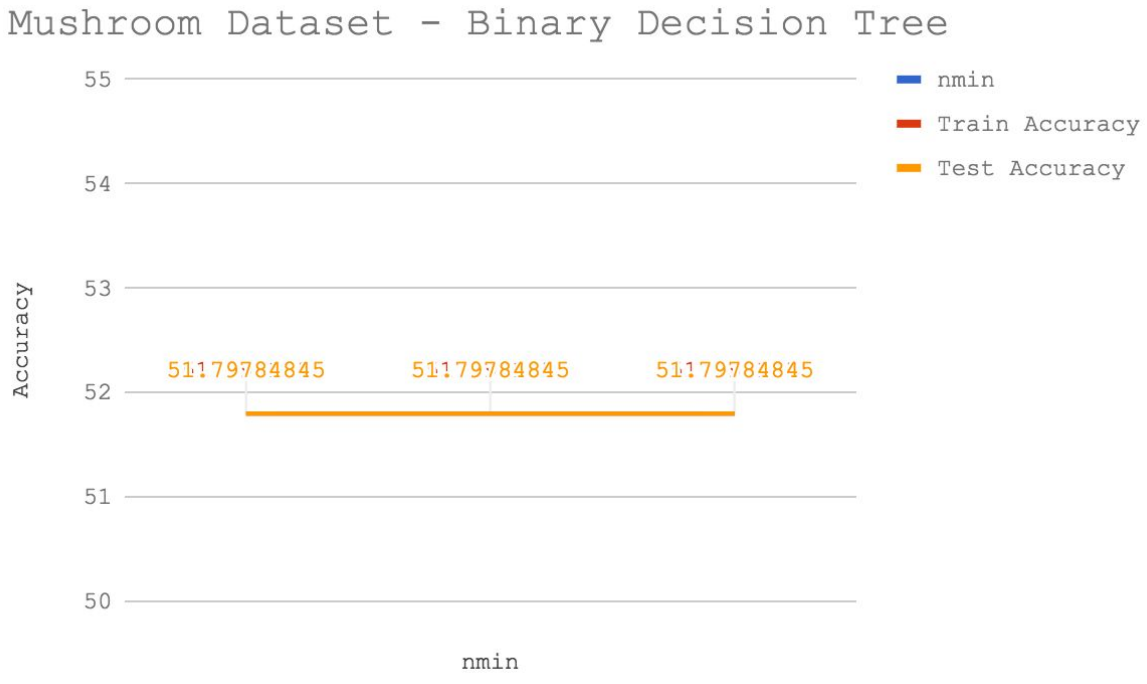
Classes	0	1
Population	8124	8124
P: Condition positive	3916	4208
N: Condition negative	4208	3916
TP: True Positive	0	4208
TN: True Negative	4208	0
FP: False Positive	0	3916
FN: False Negative	3916	0
TPR: (Sensitivity, hit rate, recall)	0	1
TNR=SPC: (Specificity)	1	0
PPV: Pos Pred Value (Precision)	NaN	0.517971
NPV: Neg Pred Value	0.517971	NaN
FPR: False-out	0	1
FDR: False Discovery Rate	NaN	0.482029
FNR: Miss Rate	1	0
ACC: Accuracy	0.517971	0.517971
F1 score	0	0.682452
MCC: Matthews correlation coefficient	NaN	NaN
Informedness	0	0
Markedness	NaN	NaN

b. n_{min}

The optimal nmin value is the same for both binary and multiway decision dress. From the results it looks like there isn't much difference in multi way and binary decision tree for similar data. It also shows that classifier needs to be set based on the dataset and we shouldn't force construct values to do one classifier type like binary in this case. Multi way is suited for discrete values and binary is suited for continuous values.

c. Training and Testing Accuracy





3. Entropy

a. Binary Feature

Entropy is given by the formula

$$H(S) = \sum_{x \in X} -p(x) \log_2 p(x)$$

Meaning the p(x) determines the value of H(S). Say if all the values in the node are positive the p(T) = 1 and P(F) = 0 thus making the H(S) as 0. The same holds if all the data are Negative. And the it reaches the maximum 1 only when the data is equally split between positive and negative nodes. For all other partitions the values of entropy is between 0 and 1.

b. Multi way branching

The formula for entropy remains the same and the same properties hold true. If the data represents only one partition the entropy is 0 and the entropy reaches a maximum when the data is split across all the partitions evenly gaining a value of 1. On all other partitions the entropy is between a 0 and 1.

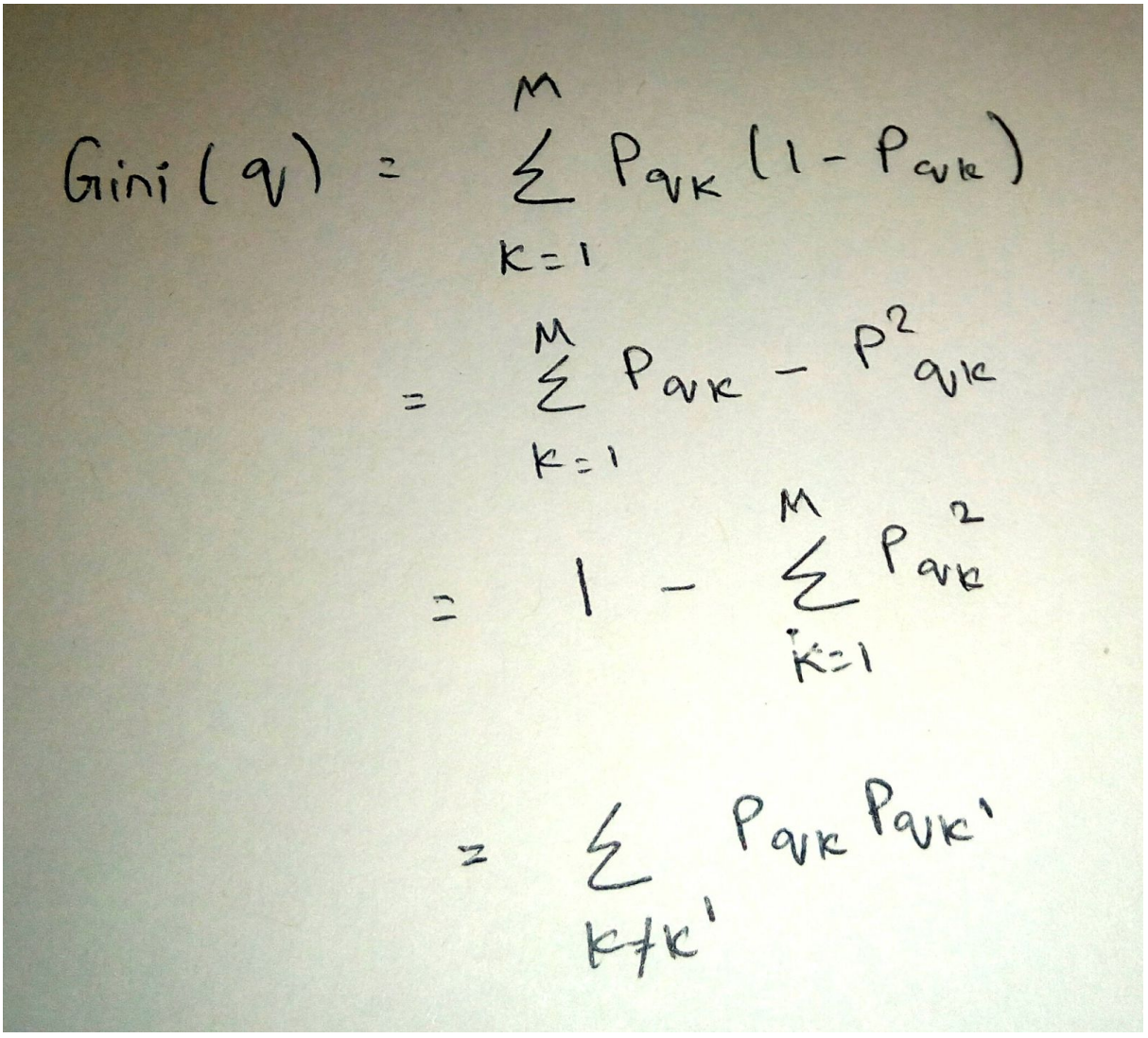
4. Gain and Impurity Measures

Information gain is given by the formula

$$IG(A, S) = H(S) - \sum_{t \in T} p(t) H(t)$$

Where H(S) is the entropy of the set S and T are the subsets created from S on splitting it with attribute A. p(t) is the proportion of the elements in the subset t and H(t) is the entropy of subset t. Here H(S), p(t) are both constant, since the entropy of set S is based on the data in the set and can't be changed and p(t) is the proportion of elements in subset that has already been derived. If the information gain is to be maximized over the attribute A all the subsets t would have to have minimum value for the entropy of the subset, meaning the subsets have to be homogenous for maximum information gain. Thus proving the hypothesis that in order to maximise the information gain the entropy of the subsets have to reduced ,i.e. The subsets should have lesser impurity.

5. Gini Index



6. Regression Trees

a. Average and Standard Deviation

n_{min}	SSE	Standard Deviation
0.05	9.124402167	1.407351182
0.10	9.06023714	1.499889548
0.15	9.143563291	0.8855054003
0.20	9.123485783	1.081914901

b. nmin

Yes nmin impacts the results. The RMSE varies based on the nmin value like the result on the table above. The nmin stopping criteria restricts the data from getting split when there set is fewer than the nmin. Thus during the predict phase on the test data there are a few cases that are the most common value when nmin is higher than when nmin is minimum.

c. training and test SSE