Assignment 3

Due Date: Monday, October 22nd, by midnight

Total number of points: 55 points plus 5 for extra credit

Problem 1 (20 points): This problem illustrates the classification approach by using decision trees and the Lupus data (you can download the data file "sledata" from D2L site, course documents for week 6). The data consists of 300 patient records. Each record contains 12 elements. The first 11 elements stand for different symptoms and the final element of each record indicates the diagnosis. Build a decision tree and report:

max depth	np	nc	Training Accuracy	Testing Accuracy	Complexity	#Nodes	#Terminal Nodes	SPSS Depth	Top 3 Important features
50	7	3	95.3	96.3	rules=7, depth=4	13	7	4	10,11,1
50	10	5	95.8	95.4	rules=7, depth=3	13	7	3	10,11,1
50	10	5	95.3	95.5	rules=7, depth=3	13	7	3	10,11,9
50	10	5	96.2	94.7	rules=7, depth=3	13	7	3	10,11,7
50	10	5	96	86.9	rules=6, depth=3	11	6	3	10,11,7
50	10	5	96.1	93.4	rules=5, depth=3	9	5	3	10,11,1
50	12	6	96.6	93.6	rules=6, depth=3	11	6	3	10,11,7
50	13	6	93.9	92.2	rules=4, depth=3	7	4	2	10,11,7
50	14	7	92.8	91.3	rules=5, depth=4	9	5	4	10,11,1
50	15	7	94.8	90.6	rules=5, depth=3	9	5	3	10,11,9
50	16	8	90.7	87.7	rules=3, depth=2	5	3	2	10,1,9
50	20	10	94.5	88.1	rules=4, depth=3	7	4	3	10,1,11
50	25	12	90.6	87.6	rules=3, depth=2	5	3	2	10,1,9
50	30	15	91.8	88.5	rules=3, depth=2	5	3	2	10,11,9
50	35	17	91.4	89.4	rules=3, depth=2	5	3	2	10,9,11
50	40	20	90.6	90.7	rules=2, depth=1	3	2	1	10,11,7
50	40	20	92.8	86.7	rules=2, depth=1	3	2	1	10,1,11
50	45	22	92	88.1	rules=2, depth=1	3	2	1	10,11,9
50	50	25	91.1	89.8	rules=2, depth=1	3	2	1	10,11,7
50	60	30	90.9	90.3	rules=2, depth=1	3	2	1	10,11,7

- np=7, nc=3 gets the highest train/test accuracies but the higher testing accuracy than training accuracy and complexity indicate that it is not the ideal model for this dataset
- np=10, nc=5 has the highest train/test accuracies on average. I include multiple examples of the results of running the decision tree classifier in SPSS. Though in one model, the train/test accuracy had a difference of about 10% which indicates overfitting.
- np=40, nc=20 ensures that the data is just split on symptom10 or the most important feature, which still produces relatively high accuracies for train/test. It also benefits from being simpler than the other models and no signs of serious overfitting.

1) **The decision tree and the criteria used for building the tree for deciding the best split and the stopping condition (such as which impurity measure, how many cases for parents and children per node, etc)**

   a. The splitting condition I used in SPSS was the Impurity Measure – Gini Index.

   b. I would also argue that an effective splitting criterion could simply be to split on symptom number 10. In this dataset, it is the most important feature and splitting on it alone yields comparable test accuracies to more complicated models on average. There are cases where train/test accuracy is higher for more specific decision trees that have low np and nc parameters (and therefore more terminal nodes and rules). These cases are also more complex, vary more, and the tradeoff is a few percent generally. Splitting on symptom 10 alone gets the diagnosis right about 9 times out of 10, which is fairly high for a small dataset and would require only keeping track of a single symptom. For the report though, I report on the splitting condition using Gini Index, as it may extrapolate better to incoming data down the line. Because this question also asks for the three most important features, I assume a depth of 3 is an acceptable level of complexity.

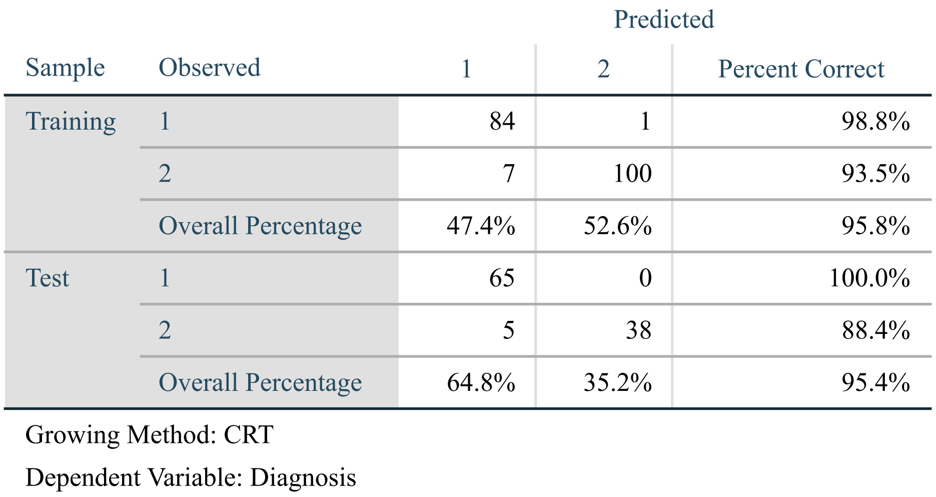
   c. For the greatest simplicity, splitting on symptom10 or the most important feature makes sense.

   d. For the best accuracy on average with a reasonable level of complexity, splitting using gini index and np=10, nc=5 makes sense. I report both below.

max depth	np	nc	Training Accuracy	Testing Accuracy	Complexity	#Nodes	#Terminal Nodes	SPSS Depth	Top 3 Important features
50	10	5	95.8	95.4	rules=7, depth=3	13	7	3	10,11,1
50	40	20	90.6	90.7	rules=2, depth=1	3	2	1	10,~~11,7~~

*I used the strikethrough on 11 and 7 for the 40/20 split because they don't actually get used by the model. With both these two cases, the models aren't underfitting the data or overfitting it. 10-5 np-nc is the best tree though.

Classification np=10, nc=5

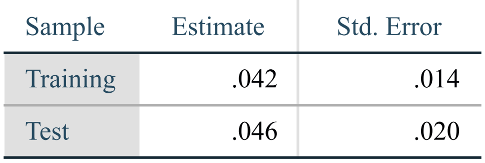


		Predicted		
Sample	Observed	1	2	Percent Correct
Training	1	84	1	98.8%
	2	7	100	93.5%
	Overall Percentage	47.4%	52.6%	95.8%
Test	1	65	0	100.0%
	2	5	38	88.4%
	Overall Percentage	64.8%	35.2%	95.4%

Growing Method: CRT

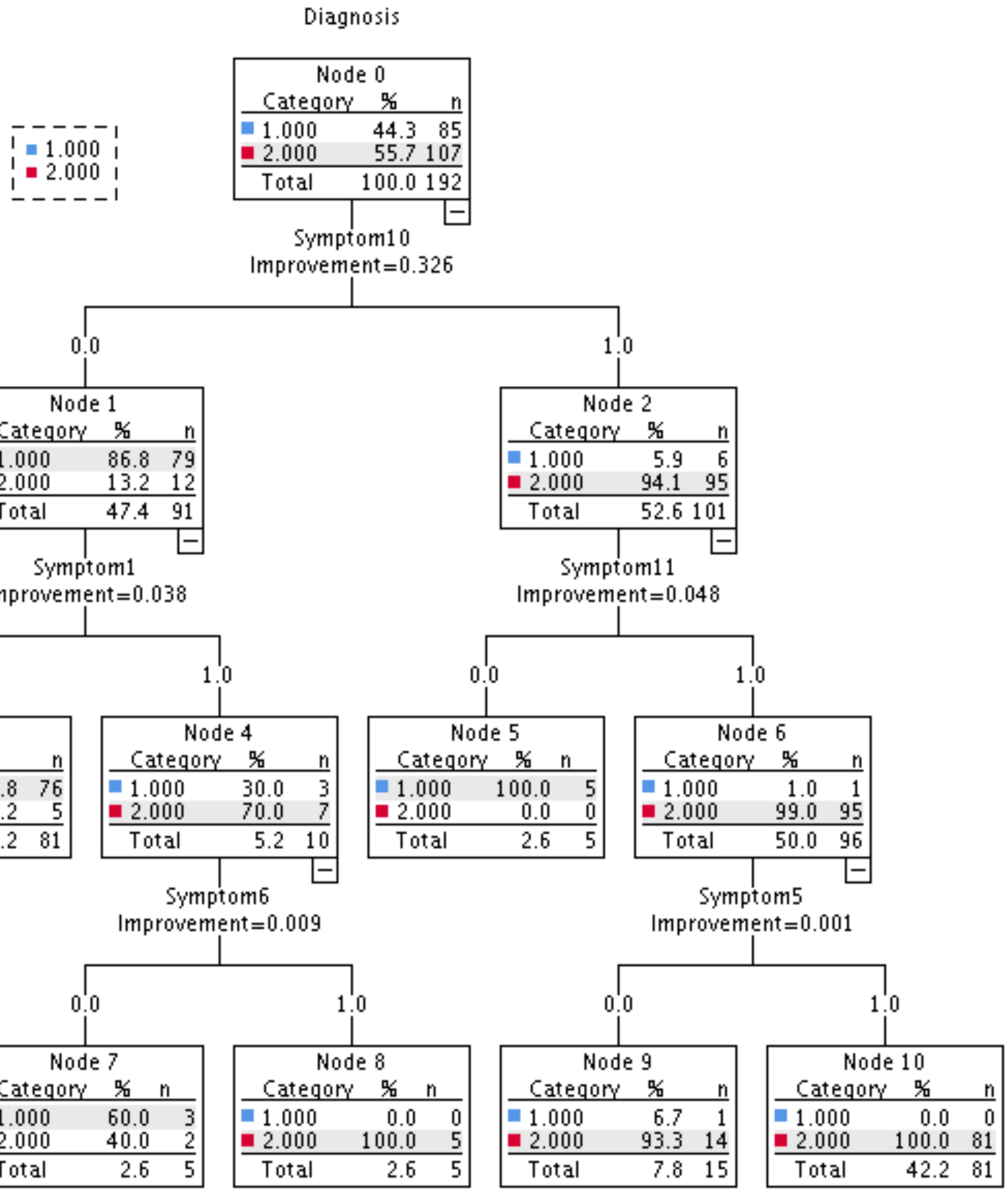
Dependent Variable: Diagnosis

Risk

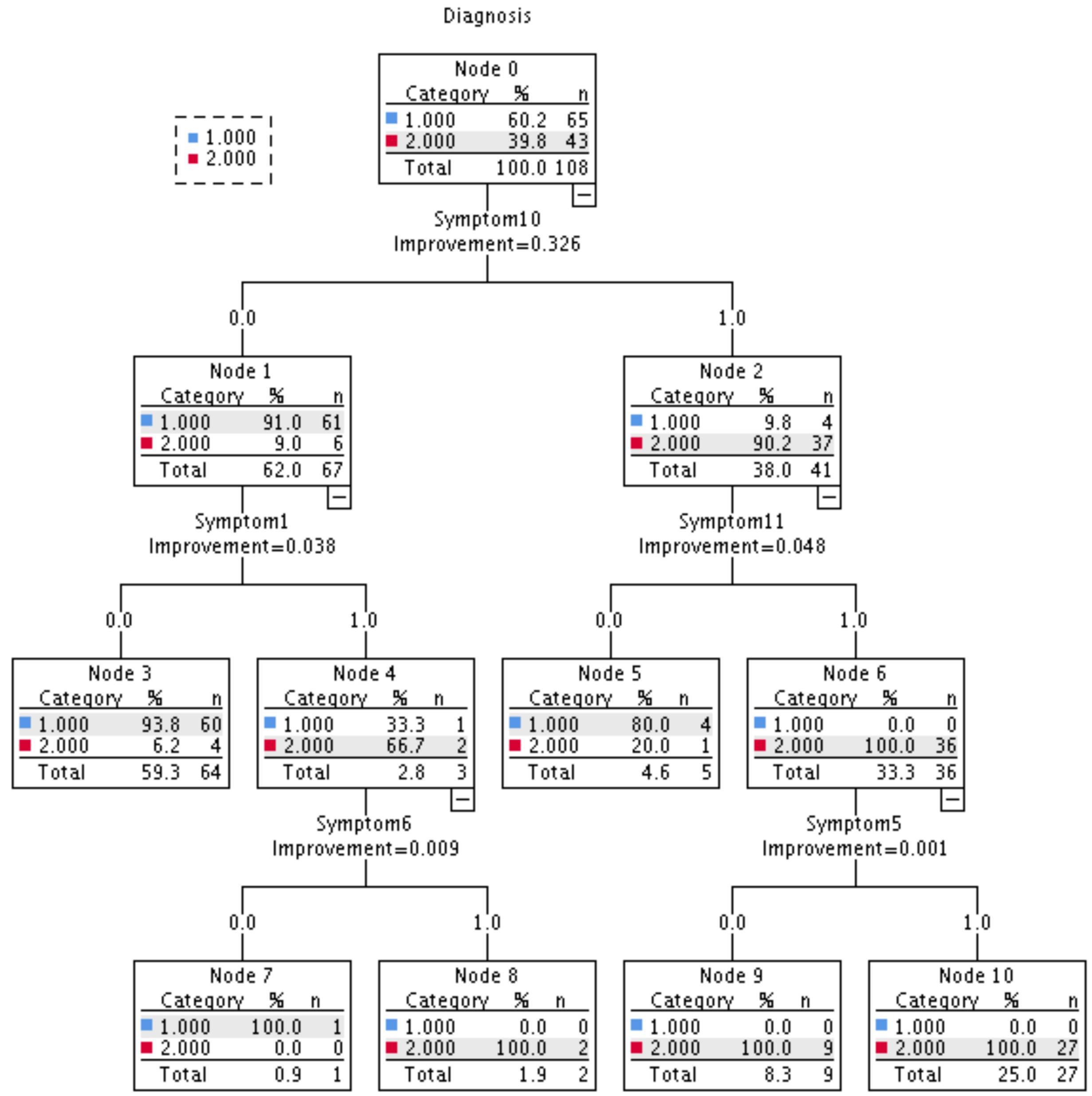


Sample	Estimate	Std. Error
Training	.042	.014
Test	.046	.020

Training Tree (np=10, nc=5, gini index 95.8% accuracy)



Testing Tree (np=10, nc=5, gini index, 95.4% accuracy)



2) How many nodes the final tree has and how many of them are terminal nodes;

a. The final tree has 13 nodes, 7 of which are terminal nodes. This means the tree has 7 rules.

3) What are the most important three Lupus data features in building the tree? Explain your answer.

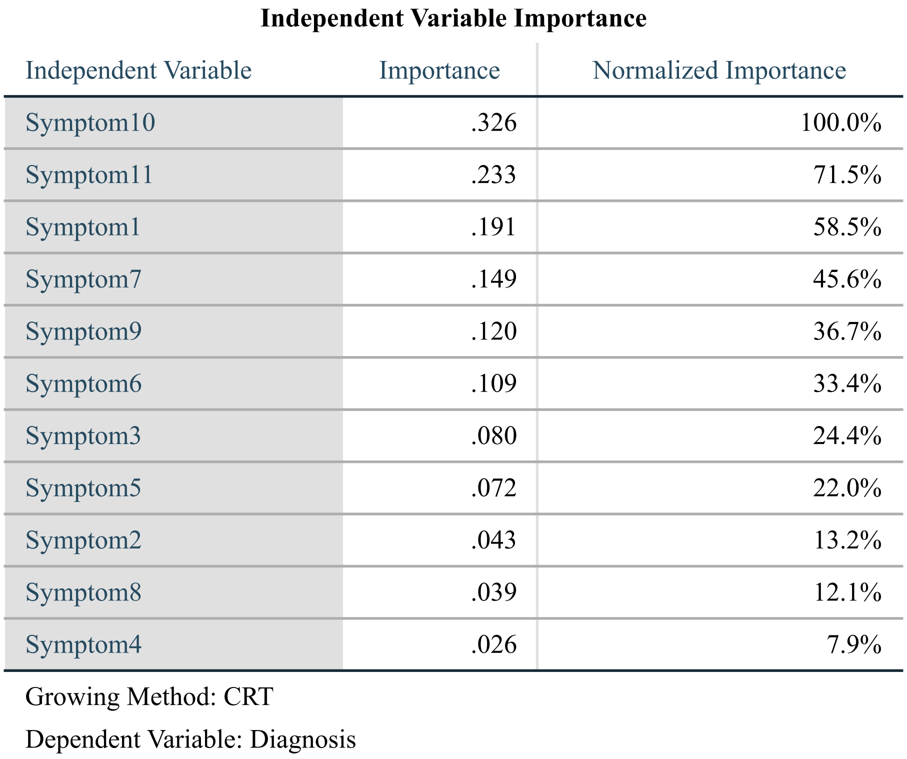
a. 10, 11, 1

b. The three most important features for this dataset depend on the random train/test split of the data.

c. Consistently, feature 10, or symptom10 is the most important feature in building the tree. When the minimum cases for parents and children are set high enough (np=40, nc=20), the decision tree uses only symptom10 to split the data, and achieves train/test accuracies around 90/90 %.

d. The second most important feature across all np/nc conditions is feature 11 or symptom 11.

e. The third most important feature changes between np/nc conditions and depends on the random split of the data into the train/test sets, but is generally feature 7 or symptom 7. In the best np=10, nc=5 run the tree had symptom 1 as it's 3rd most important feature.

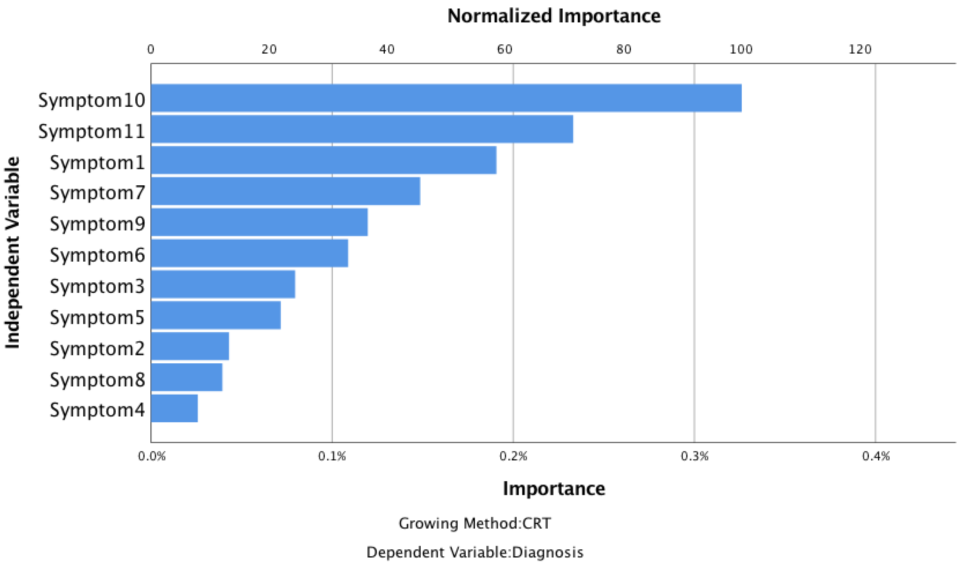


Independent Variable Importance

Independent Variable	Importance	Normalized Importance
Symptom10	.326	100.0%
Symptom11	.233	71.5%
Symptom1	.191	58.5%
Symptom7	.149	45.6%
Symptom9	.120	36.7%
Symptom6	.109	33.4%
Symptom3	.080	24.4%
Symptom5	.072	22.0%
Symptom2	.043	13.2%
Symptom8	.039	12.1%
Symptom4	.026	7.9%

Growing Method: CRT

Dependent Variable: Diagnosis



4) **Increase the number of cases for each parent and child. What do you notice with the complexity of the tree? Does it increase? Explain your answer.**

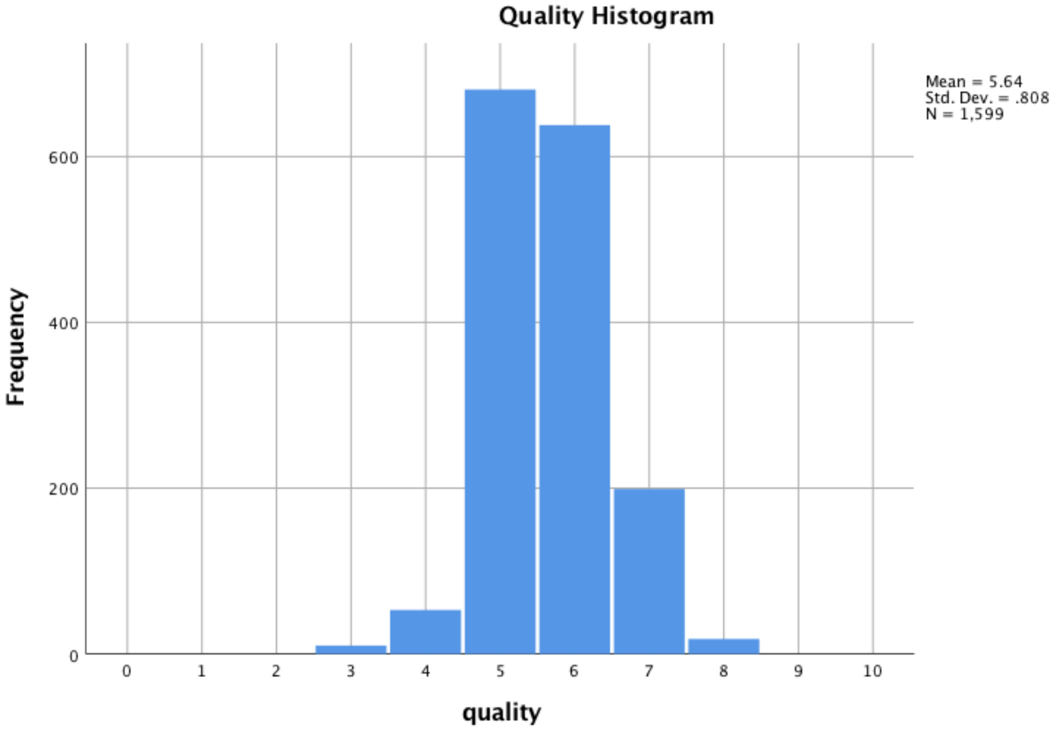
a. Increasing the minimum number of cases for each parent and child decreases the complexity of the tree. In the table below, this relationship is evidenced by the decreasing *#Terminal Nodes* and *SPSS Depth* as the np and nc parameters increase. At around np=40, the data is just split on symptom10, producing trees with a depth of 1 and 2 rules.

b. This decrease in complexity makes sense because more cases are needed in order to expand the tree, and in our small train/test sets there simply are not enough cases to meet high minimums.

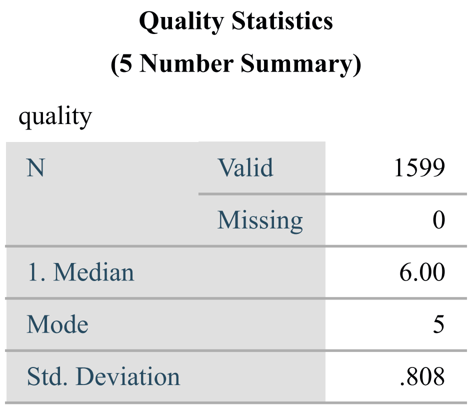
max depth	np	nc	Training Accuracy	Testing Accuracy	Complexity	#Nodes	#Terminal Nodes	SPSS Depth	Top 3 Important features
50	10	5	95.8	95.4	rules=7, depth=3	13	7	3	10,11,1
50	20	10	94.5	88.1	rules=4, depth=3	7	4	3	10,1,11
50	40	20	90.6	90.7	rules=2, depth=1	3	2	1	10,11,7
50	60	30	90.9	90.3	rules=2, depth=1	3	2	1	10,11,7

Problem 2 (30 points): This problem illustrates the effect of the class imbalance of the accuracy of the decision trees. Download the red wine quality data from the UCI machine learning repository at: **http://archive.ics.uci.edu/ml/datasets/Wine+Quality**

1. **Report how many classes (treat each quality level as a different class) are and what is the distribution of these classes for the red wine data is.**
    a. Classes: There are theoretically 11 classes, as quality is scored on an integer value between 0-10 with 0 being 'very bad' and 10 being 'very excellent'. In the red wine dataset however, there are 6 classes present. The quality measures recorded in the red wine dataset are 3, 4, 5, 6, 7, and 8.



Histogram 1: This histogram shows the frequency of each quality score recorded in the red win dataset.



Quality Statistics

(5 Number Summary)

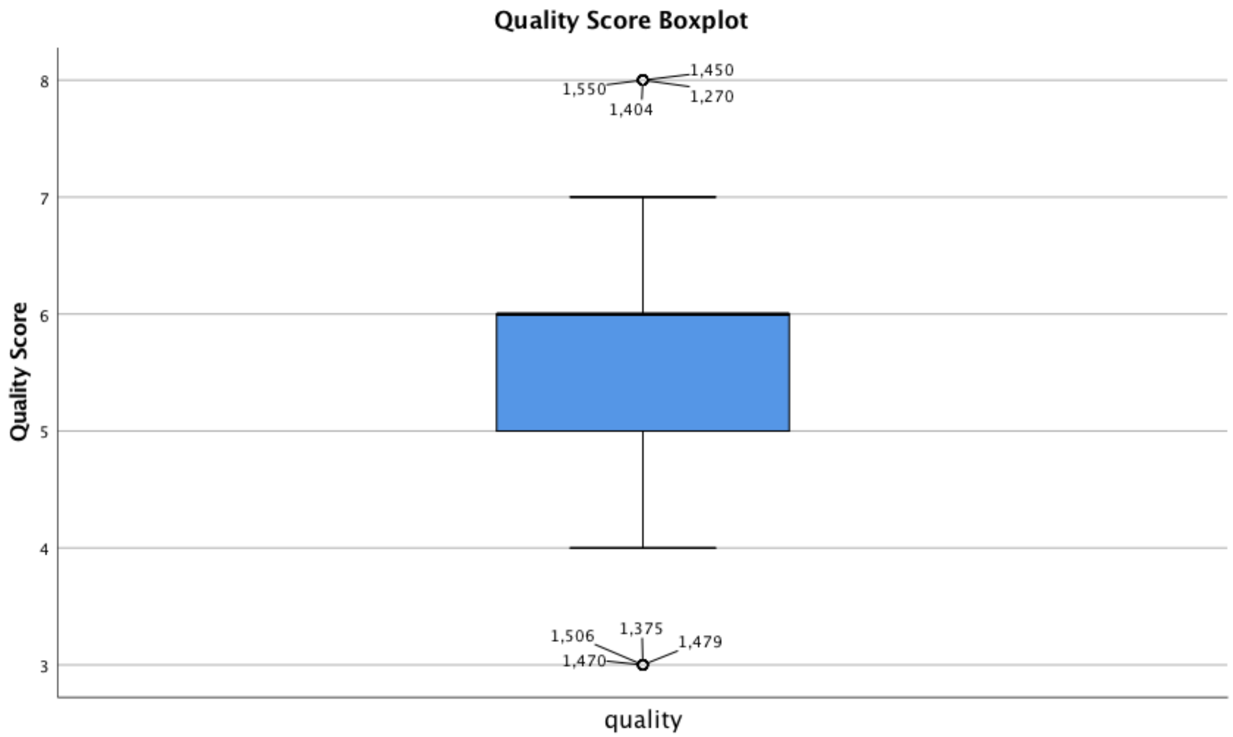
quality

N	Valid	1599
	Missing	0
1. Median		6.00
Mode		5
Std. Deviation		.808

2. Minimum		3
3. Maximum		8
Percentiles	4. 25	5.00
	50	6.00
	5. 75	6.00

Quality: Frequencies

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	3	10	.6	.6	.6
	4	53	3.3	3.3	3.9
	5	681	42.6	42.6	46.5
	6	638	39.9	39.9	86.4
	7	199	12.4	12.4	98.9
	8	18	1.1	1.1	100.0
	Total	1599	100.0	100.0	



Boxplot 1: This boxplot shows the distribution of data for the quality scores.

b. The histogram works better than the boxplot for describing the distribution of the quality score data, but it still shows that the majority of wines were scored as 5s or 6s.

c. To summarize, the data is imbalanced in terms of the number of cases of each quality score (or class). If the dataset was balanced, there would be a more equal number of cases for each quality score.

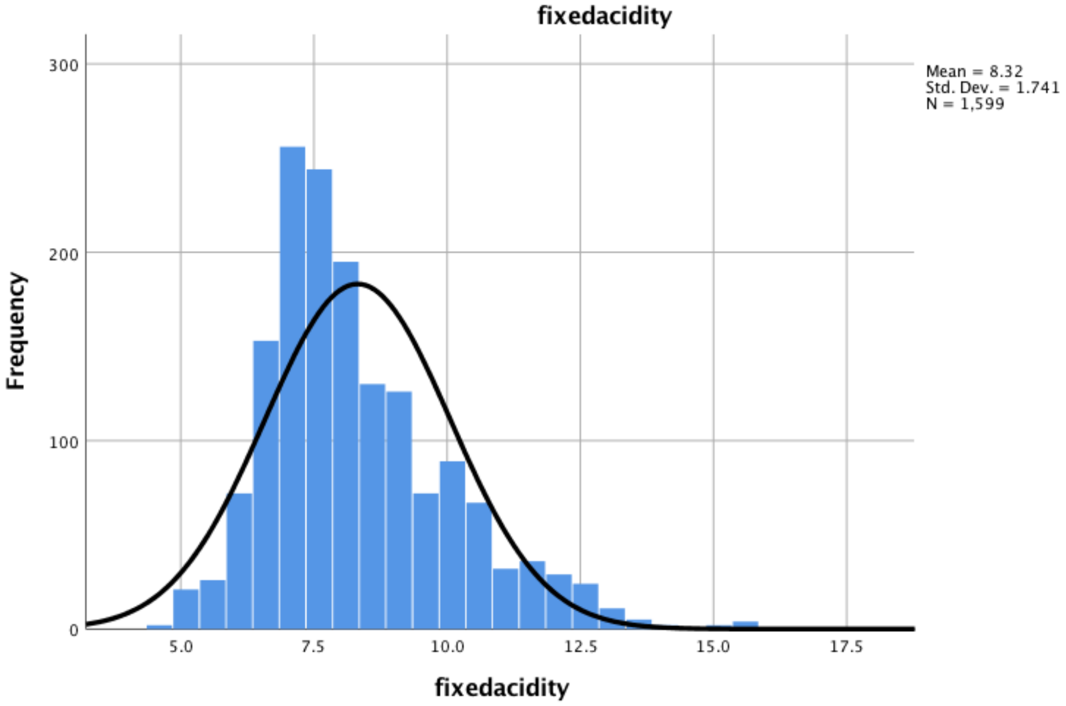
2. Repeat Problem 1 on the red wine data.

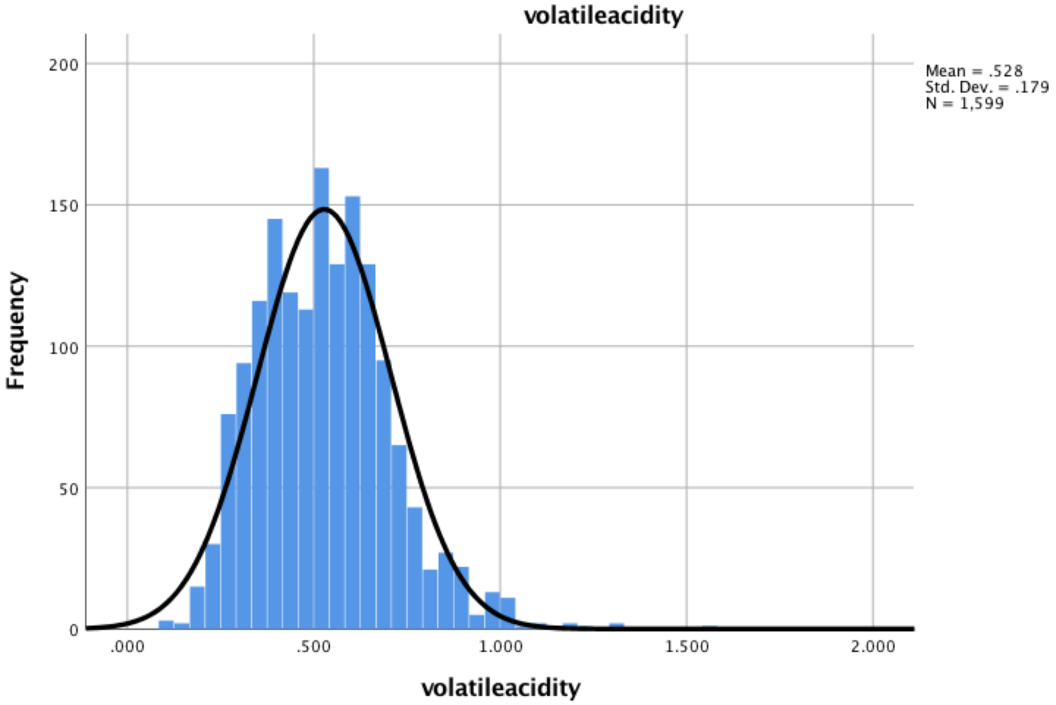
a. For this problem, the 5 Number Summary and histograms are provided for each of the features. pH, density, and volatile acidity are notable normal distributions. All of the features are fairly normally distributed though, with the exception of citric acid content. Most have some degree of skew to them as well.

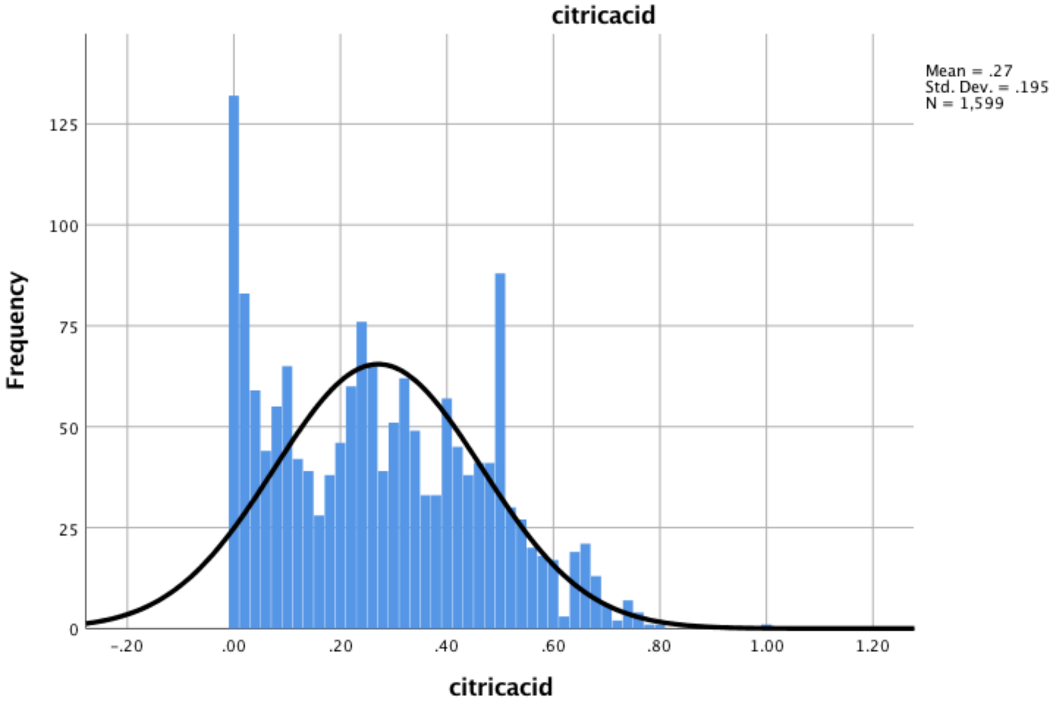
Red Wine Data Statistics (5 Number Summary)

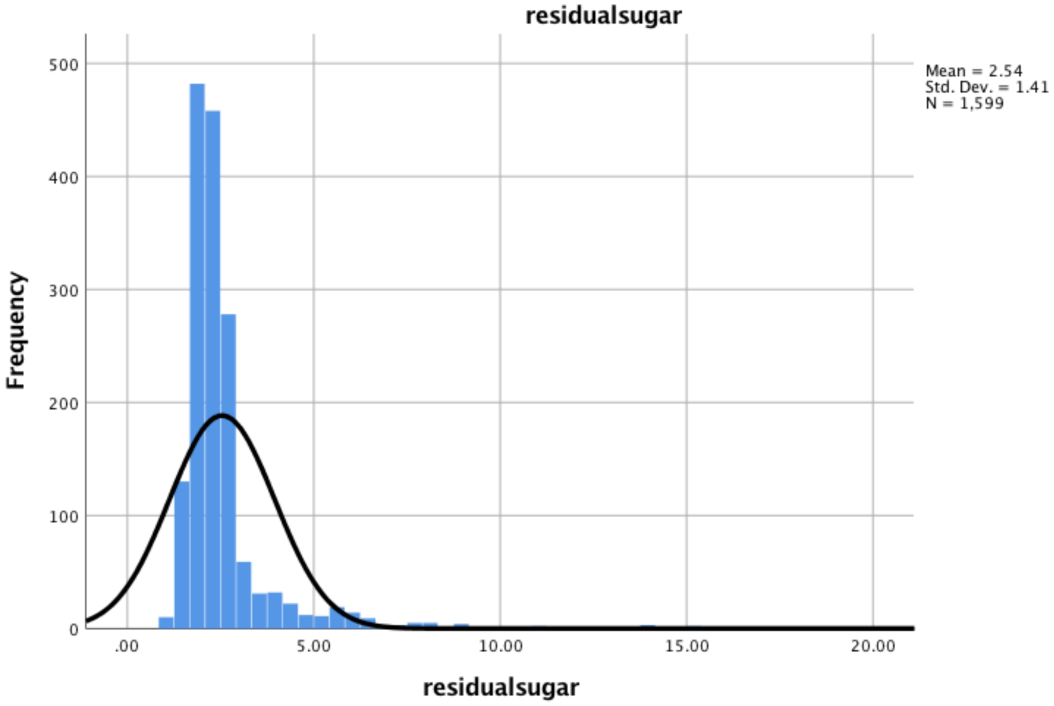
		Fixed acidity	Volatile acidity	Citric acid	Residual sugar	chlorides	Free sulfur dioxide	Total sulfur dioxide	density	pH	sulphates	alcohol
N	Valid	1599	1599	1599	1599	1599	1599	1599	1599	1599	1599	1599
	Missing	0	0	0	0	0	0	0	0	0	0	0
Median		7.900	.52000	.2600	2.2000	.07900	14.000	38.000	.9967500	3.3100	.6200	10.20
Mode		7.2	.600	.00	2.00	.080	6.0	28.0	.99720	3.30	.60	9.50
Std.Dev		1.7411	.179060	.1948	1.40993	.047065	10.4602	32.8953	.00188733	.15439	.16951	1.06
Minimum		4.6	.120	.00	.90	.012	1.0	6.0	.99007	2.74	.33	8.40
Maximum		15.9	1.580	1.00	15.50	.611	72.0	289.0	1.00369	4.01	2.00	14.90
	Q1	7.100	.39000	.0900	1.9000	.07000	7.000	22.000	.9956000	3.2100	.5500	9.50
	50	7.900	.52000	.2600	2.2000	.07900	14.000	38.000	.9967500	3.3100	.6200	10.20

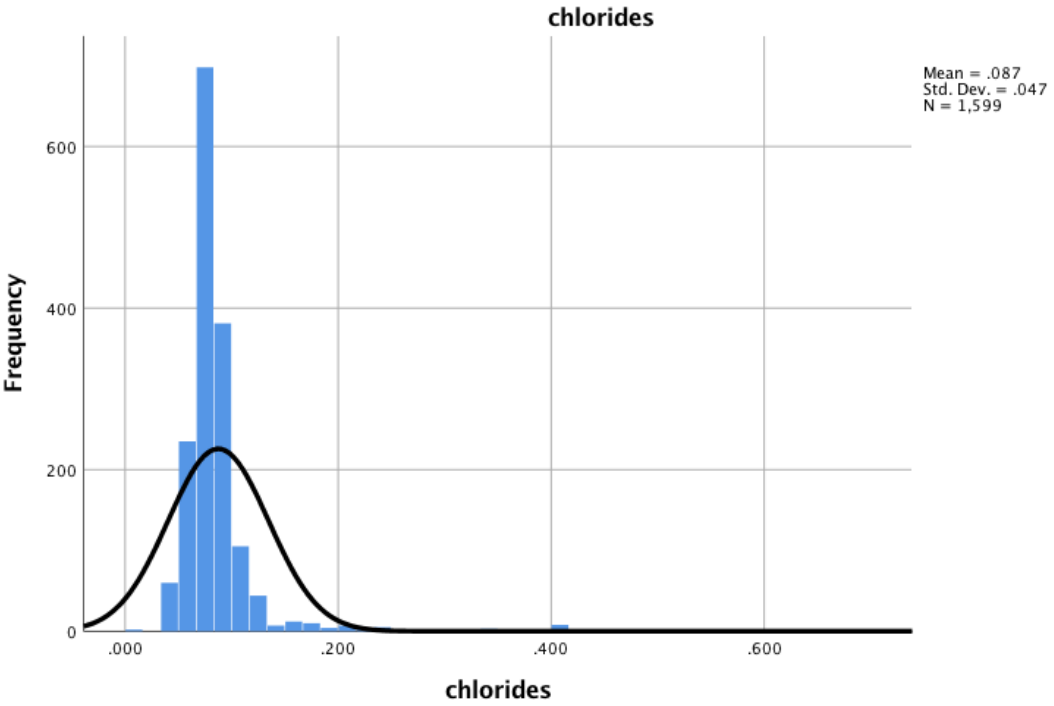
Q3	9.200	.64000	.4200	2.6000	.09000	21.000	62.000	.9978400	3.4000	.7300	11.10

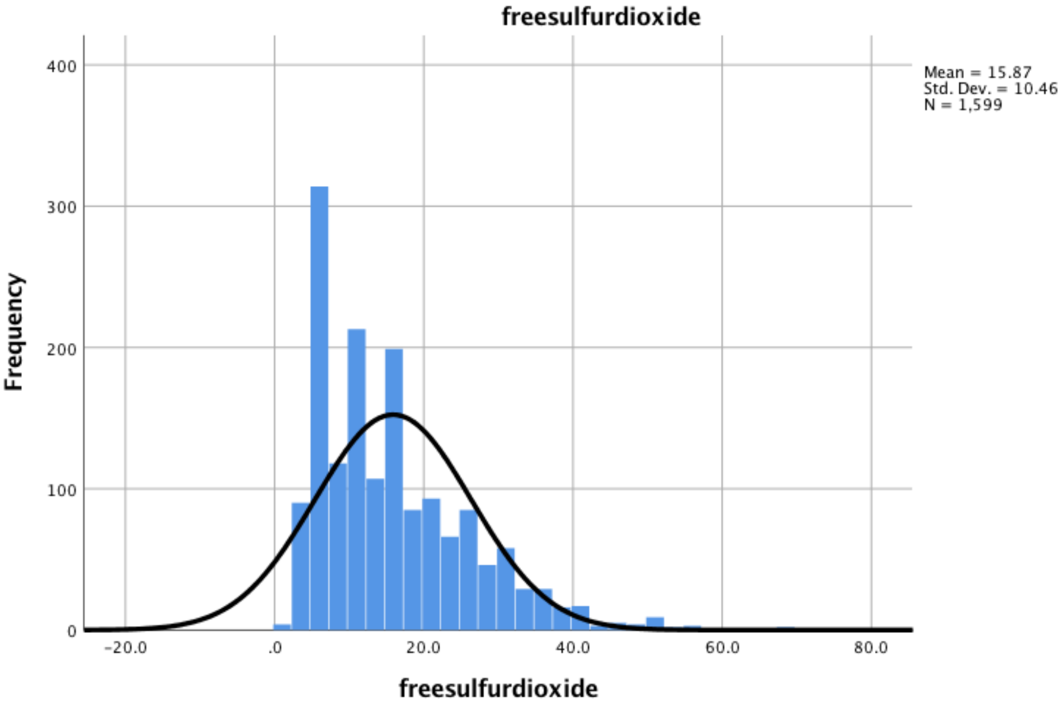


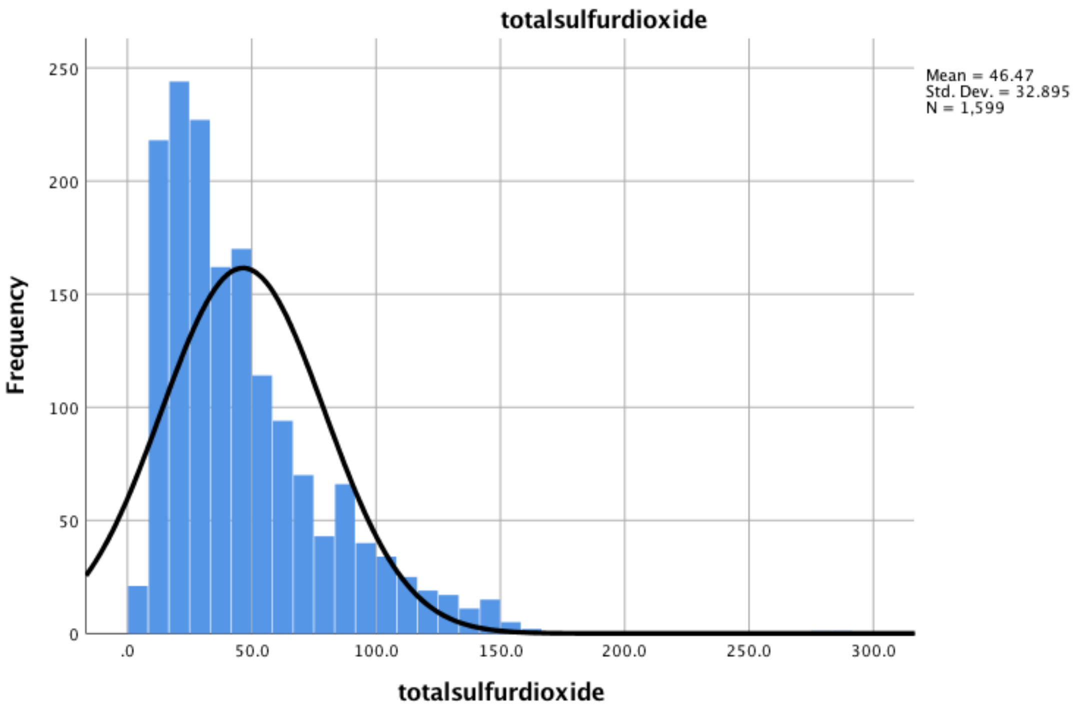


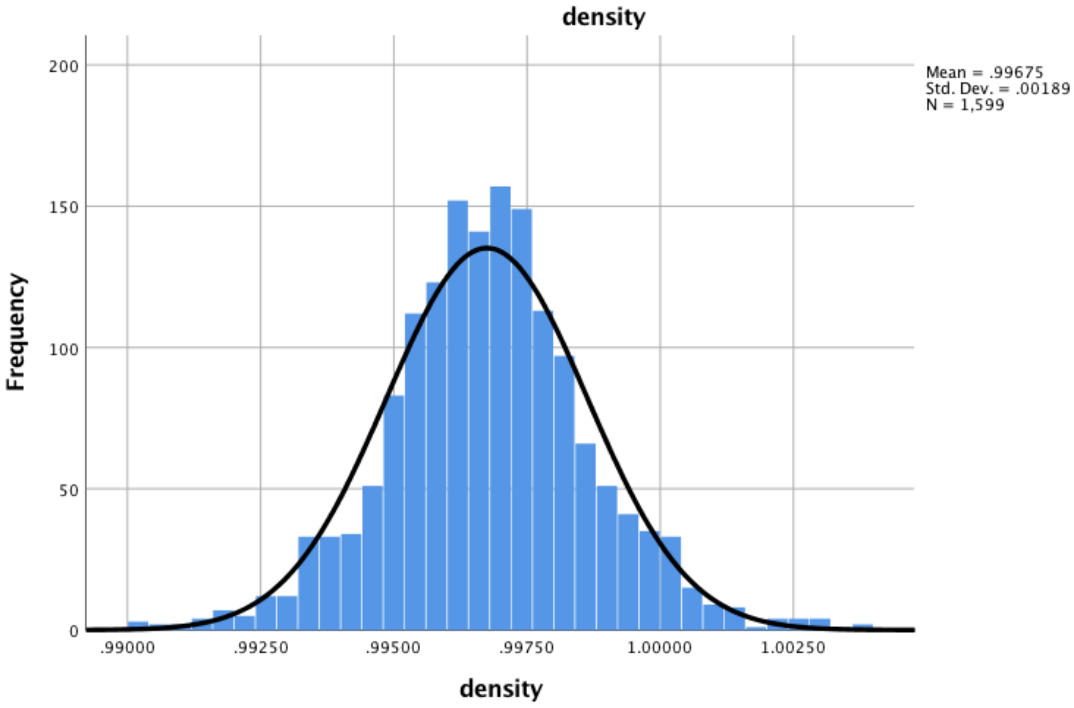

citricacid
Frequency
125
100
75
50
25
0
-.20
.00
.20
.40
.60
.80
1.00
1.20
citricacid
Mean = .27
Std. Dev. = .195
N = 1,599

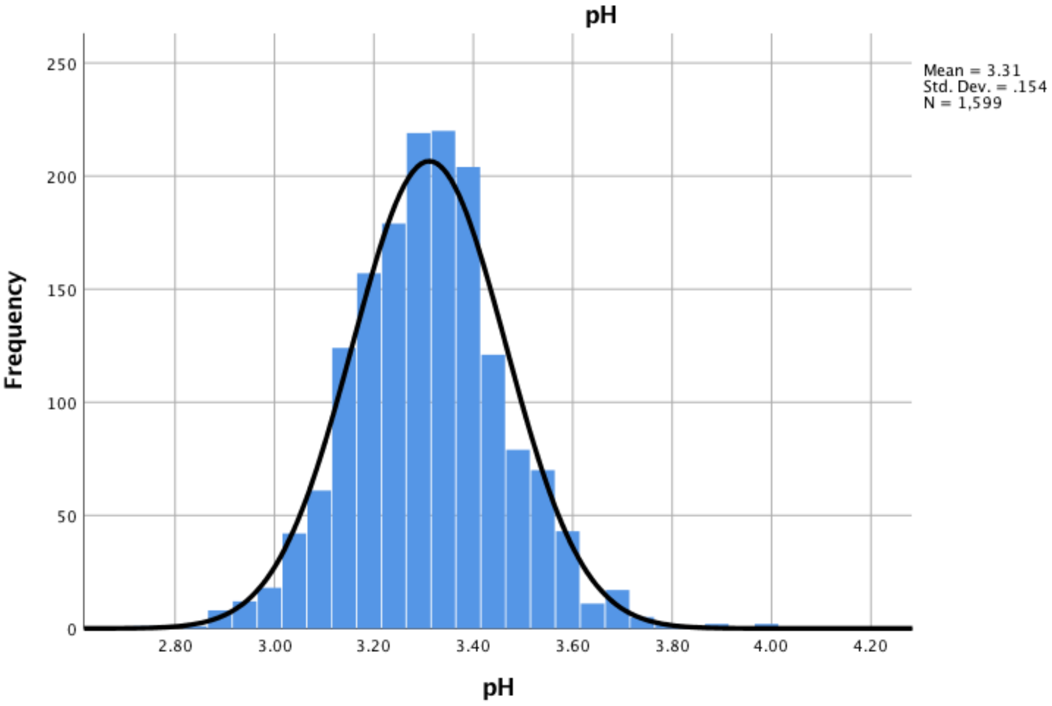

residualsugar
Frequency
500
400
300
200
100
0
.00
5.00
10.00
15.00
20.00
residualsugar
Mean = 2.54
Std. Dev. = 1.41
N = 1,599

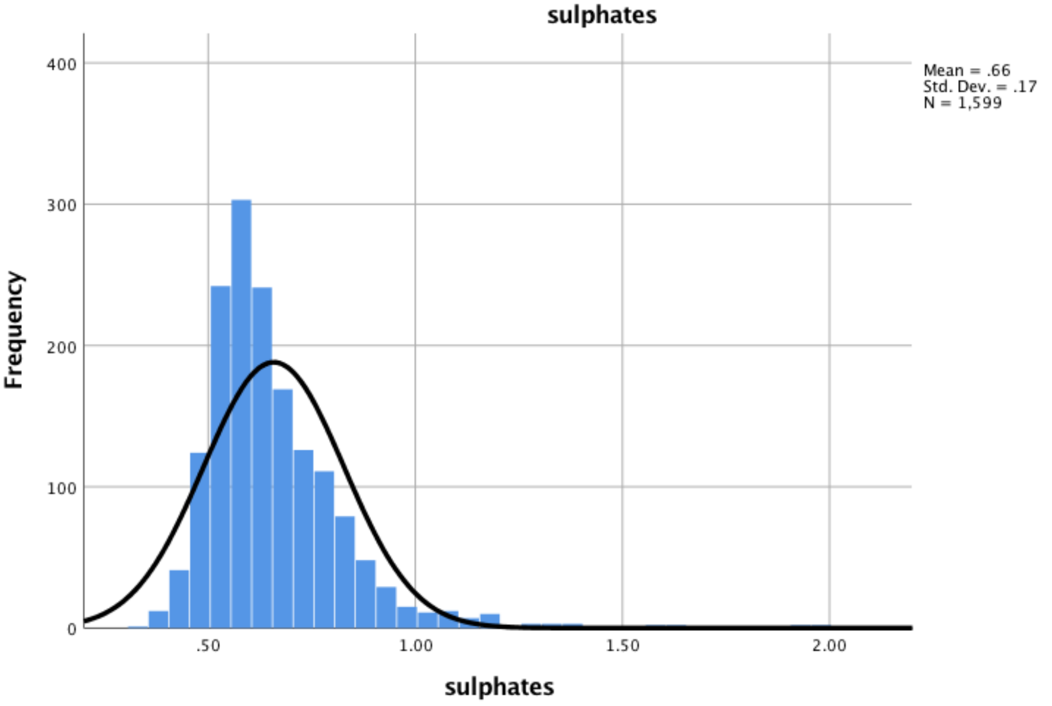

chlorides
Mean = .087
Std. Dev. = .047
N = 1,599
Frequency
600
400
200
0
.000
.200
.400
.600
chlorides

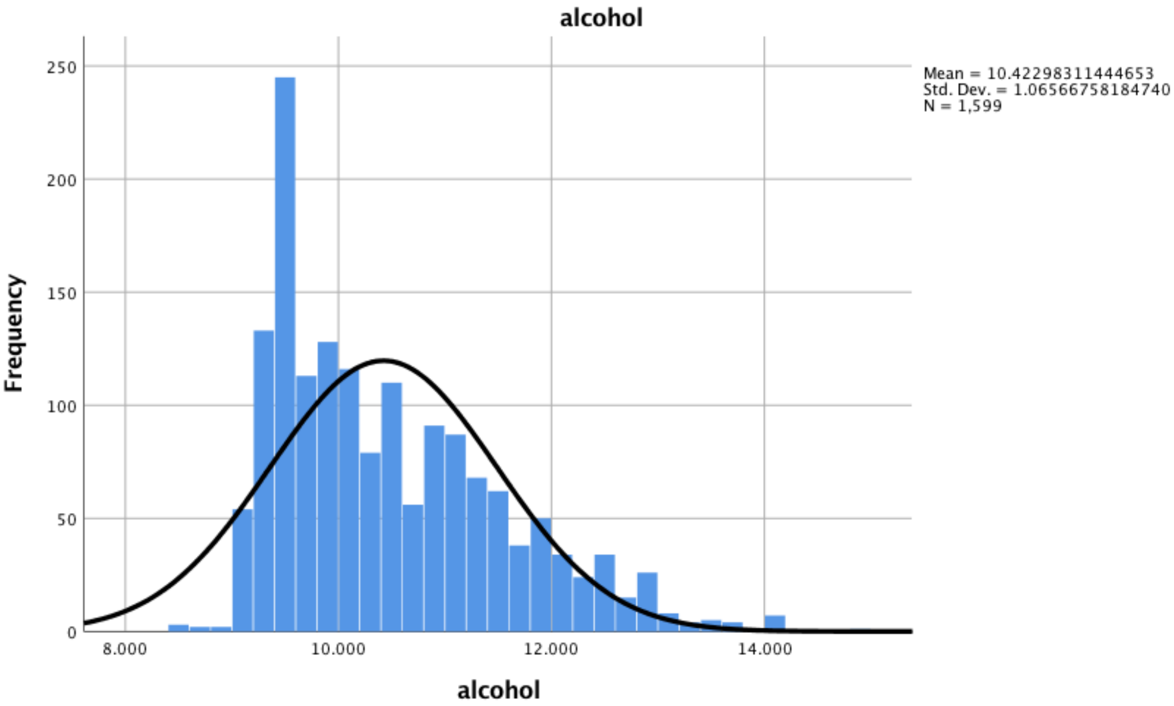

freesulfurdioxide
Mean = 15.87
Std. Dev. = 10.46
N = 1,599
Frequency
400
300
200
100
0
-20.0
.0
20.0
40.0
60.0
80.0
freesulfurdioxide


totalsulfurdioxide
Mean = 46.47
Std. Dev. = 32.895
N = 1,599
Frequency
0
50
100
150
200
250
.0
50.0
100.0
150.0
200.0
250.0
300.0
totalsulfurdioxide

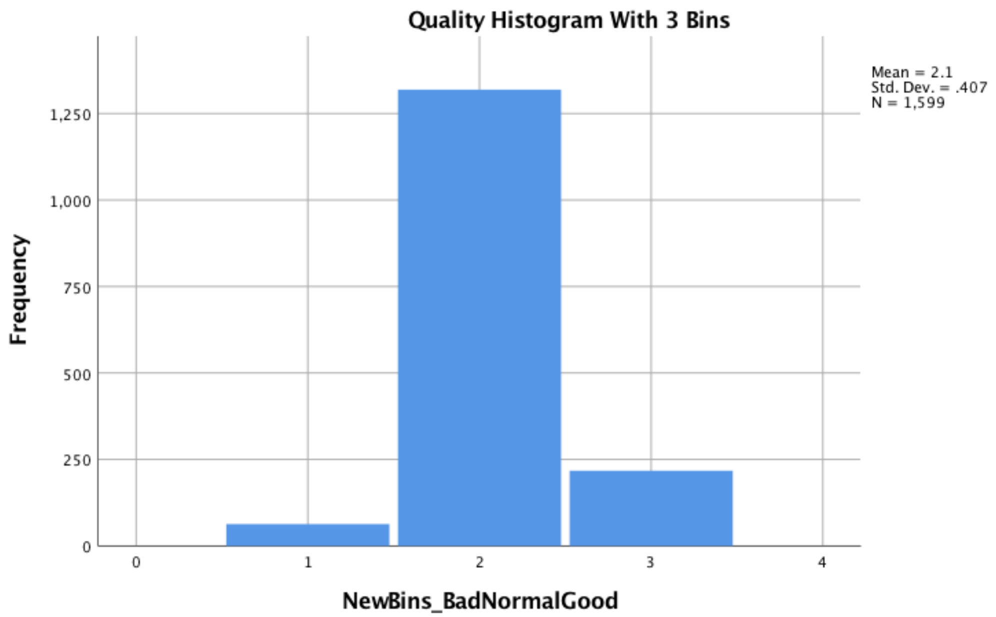

density
Mean = .99675
Std. Dev. = .00189
N = 1,599
Frequency
0
50
100
150
200
.99000
.99250
.99500
.99750
1.00000
1.00250
density


pH
Mean = 3.31
Std. Dev. = .154
N = 1,599
Frequency
0
50
100
150
200
250
2.80
3.00
3.20
3.40
3.60
3.80
4.00
4.20
pH


sulphates
Mean = .66
Std. Dev. = .17
N = 1,599
Frequency
0
100
200
300
400
.50
1.00
1.50
2.00
sulphates



3. **Now bin the class variable in such a way that data is not so imbalanced with respect to the class variable. Repeat Problem 1 but on the wine data with less number of classes (the binned class variable).**
    a. I report below two different binning schemes.
    b. The first splits the data into a bad bin (scores = 0,1,2,3, or 4), normal bin (score = 5 or 6), and good bin (scores = 7,8,9, or 10). This doesn't do too much to balance out the data, but retains some of the integrity of the test. It should also be noted that the bins really work out to Bin1=3,4 ; Bin2=5,6 ; Bin3=7,8 because those are the values that actually show up in the dataset.



Bin Histogram1: This shows the frequency of each quality score as it appears in the new bins. This histogram has 3 bins. Bin=1 for scores 0,1,2,3, and 4. Bin=2 for scores 5 and 6. Bin=3 for scores 7,8,9, and 10.

3 Bins Statistics (5 Number Summary)

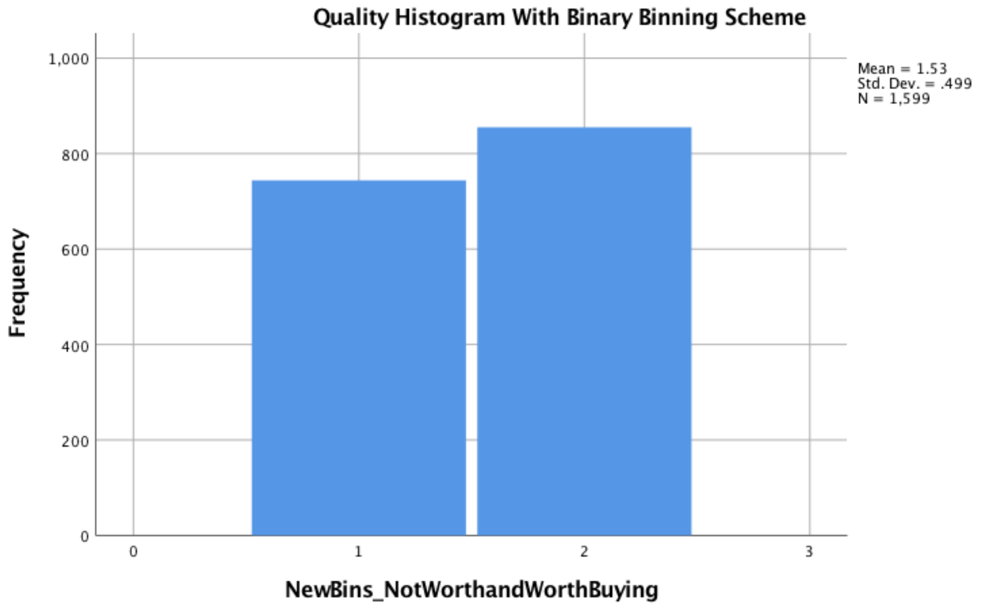
NewBins_BadNormalGood

N	Valid	1599
	Missing	0
Median		2.00
Mode		2
Std. Deviation		.407
Minimum		1
Maximum		3
Percentiles	Q1	2.00
	Q3	2.00

Frequencies NewBins_BadNormalGood

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1 = bad	63	3.9	3.9	3.9
	2 = normal	1319	82.5	82.5	86.4
	3 = good	217	13.6	13.6	100.0
	Total	1599	100.0	100.0	

c. The second binning operation I performed was to split the data up into two bins. The first bin was for wines Not Worth Buying (score = 0,1,2,3,4,5) and Worth Buying (score = 6,7,8,9,10). This produces closer to a 50/50 split in the data, so it is very balanced.



Bin Histogram2: This shows the frequency of quality scores falling into the new bins. Not Worth Buying for score<=5 (1) and Worth Buying for score<5 (2).

2Bins Statistics (5 Number Summary)

NewBins_NotWorthandWorthBuying

N	Valid	1599
	Missing	0
Median		2.00
Mode		2
Std. Deviation		.499
Minimum		1
Maximum		2

Percentiles	Q1	1.00
	Q3	2.00

NewBins_NotWorthandWorthBuying

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1 = Not Worth Buying	744	46.5	46.5	46.5
	2 = Worth Buying	855	53.5	53.5	100.0
	Total	1599	100.0	100.0	

4. **How does the performance of the best classification model on the original class variable compare with the accuracy of the best classification model on the binned classification variable?**
   - **a.** The accuracy on the binned classification model using a decision tree is better than it was on the original.
   - **b.** Original = 66.4% accuracy, **3Bins Model = 86.3% accuracy**, 2Bins Model = 77% accuracy
   - **c.** Interestingly, the 3Bins decision tree had the best performance with an accuracy of 86.3%.
   - **d.** The best model I used was a decision tree, max depth = 50, np=40, nc=20, cross-validation with 5 folds, split on impurity measure = gini index. This was also notably better than simply using the 60/40 training testing split which only achieve 85% accuracy.

Original: Classification np=40, nc=20,

	Predicted						
Observed	3	4	5	6	7	8	Percent Correct
3	0	0	9	1	0	0	0.0%
4	0	0	30	22	1	0	0.0%
5	0	0	555	116	10	0	81.5%
6	0	0	186	407	45	0	63.8%
7	0	0	31	68	100	0	50.3%
8	0	0	1	6	11	0	0.0%
Overall Percentage	0.0%	0.0%	50.8%	38.8%	10.4%	0.0%	66.4%

Growing Method: CRT

Dependent Variable: quality

Original: Risk

Method	Estimate	Std. Error
Resubstitution	.336	.012
Cross-Validation	.408	.012

Growing Method: CRT

Dependent Variable: quality

3Bins: Classification

	Predicted			
				Percent
Observed	1 = bad	2 = normal	3 = good	Correct
1 = bad	0	62	1	0.0%
2 = normal	0	1270	49	96.3%
3 = good	0	107	110	50.7%
Overall Percentage	0.0%	90.0%	10.0%	86.3%

Growing Method: CRT

Dependent Variable: NewBins_BadNormalGood

3Bins: Risk

Method	Estimate	Std. Error
Resubstitution	.137	.009
Cross-Validation	.163	.009

Growing Method: CRT

Dependent Variable: NewBins_BadNormalGood

2Bins: Classification

	Predicted		
	1 = Not Worth	2 = Worth	Percent
Observed	Buying	Buying	Correct

1 = Not Worth Buying	570	174	76.6%
2 = Worth Buying	194	661	77.3%
Overall Percentage	47.8%	52.2%	77.0%

Growing Method: CRT

Dependent Variable: NewBins_NotWorthandWorthBuying

Risk

Method	Estimate	Std. Error
Resubstitution	.230	.011
Cross-Validation	.278	.011

Growing Method: CRT

Dependent Variable: NewBins_NotWorthandWorthBuying

5. **Do you have any other ideas on how you can improve the results further? Showing that your idea will actually work will be graded with five extra credit points.**
   a. I thought linear discriminant analysis might improve the classification accuracy but was incorrect. The 83% accuracy is better than on the original, but worse than the decision tree used.

Classification Results[a,c]

			Predicted Group Membership			
		NewBins_BadNormalGood	1 = bad	2 = normal	3 = good	Total
Original	Count	1 = bad	8	53	2	63
		2 = normal	13	1242	64	1319
		3 = good	0	132	85	217
	%	1 = bad	12.7	84.1	3.2	100.0
		2 = normal	1.0	94.2	4.9	100.0
		3 = good	.0	60.8	39.2	100.0
Cross-validated[b]	Coun	1 = bad	6	55	2	63

	t	2 = normal	15	1239	65	1319
		3 = good	0	134	83	217
	%	1 = bad	9.5	87.3	3.2	100.0
		2 = normal	1.1	93.9	4.9	100.0
		3 = good	.0	61.8	38.2	100.0

a. 83.5% of original grouped cases correctly classified.

b. Cross validation is done only for those cases in the analysis. In cross validation, each case is classified by the functions derived from all cases other than that case.

c. 83.1% of cross-validated grouped cases correctly classified.

b. The original researchers who explored this dataset found SVM to be the best at modeling the data but I have been unable to test this out in SPSS.

c. Random Forests might also offer an improvement, but again I have been unable to figure out how to use this in SPSS.

Problem 3 (5 points): Given the decision tree in Figure 1, show how the new examples in Table 1 would be classified by filling in the last column in the table. If an example cannot be classified, enter UNKNOWN in
the last column. For each example, explain your answer by writing down the path from the root to the leaf that corresponds to that specific example.

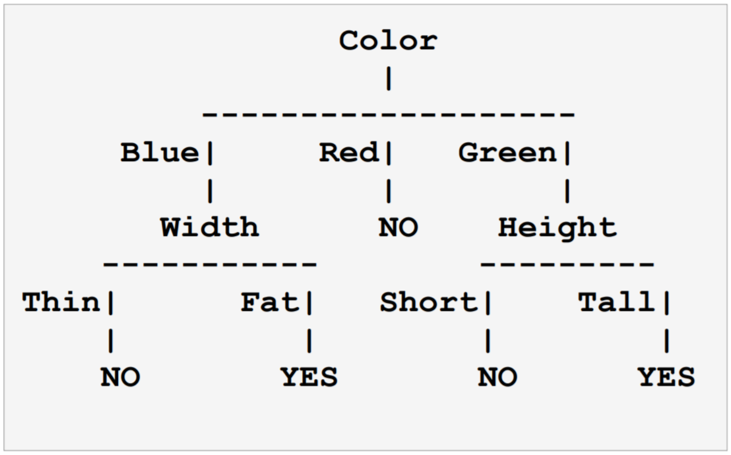


Figure 1: Decision tree

Table 1: Data for Problem #3

Example	Color	Height	Width	Class	Path (root→leaf)
A	Red	Short	Thin	**NO**	(Color=Red) → (Class=NO)
B	Blue	Tall	Fat	**YES**	(Color=Blue) → (Width=Fat) → (Class=YES)
C	Green	Short	Fat	**NO**	(Color=Green) → (Height=Short) → (Class=NO)
D	Green	Tall	Thin	**YES**	(Color=Green) → (Height=Tall) → (Class=YES)
E	Blue	Short	Thin	**NO**	(Color=Blue) → (Width=Thin) → (Class=NO)

Problem 1 Appendix:

Classification np=7, nc=3

		Predicted		
Sample	Observed	1	2	Percent Correct

Training	1	93	1	98.9%
	2	8	91	91.9%
	Overall Percentage	52.3%	47.7%	95.3%
Test	1	56	0	100.0%
	2	4	47	92.2%
	Overall Percentage	56.1%	43.9%	96.3%

Growing Method: CRT

Dependent Variable: Diagnosis

Classification t1 np=10, nc=5

		Predicted		
Sample	Observed	1	2	Percent Correct
Training	1	90	5	94.7%
	2	4	91	95.8%
	Overall Percentage	49.5%	50.5%	95.3%
Test	1	51	4	92.7%
	2	1	54	98.2%
	Overall Percentage	47.3%	52.7%	95.5%

Growing Method: CRT

Dependent Variable: Diagnosis

Classification t2 np=10, nc=5

		Predicted		
Sample	Observed	1	2	Percent Correct
Training	1	91	1	98.9%
	2	6	88	93.6%
	Overall Percentage	52.2%	47.8%	96.2%
Test	1	58	0	100.0%
	2	6	50	89.3%
	Overall Percentage	56.1%	43.9%	94.7%

Growing Method: CRT

Dependent Variable: Diagnosis

Risk np=10, nc=5

Sample	Estimate	Std. Error
Training	.038	.014
Test	.053	.021

Growing Method: CRT

Dependent Variable: Diagnosis

Classification t3 np=10, nc=5

		Predicted		
Sample	Observed	1	2	Percent Correct
Training	1	86	2	97.7%
	2	5	86	94.5%
	Overall Percentage	50.8%	49.2%	96.1%
Test	1	59	3	95.2%
	2	5	54	91.5%
	Overall Percentage	52.9%	47.1%	93.4%

Growing Method: CRT

Dependent Variable: Diagnosis

Classification np=15, nc=7

		Predicted		
Sample	Observed	1	2	Percent Correct
Training	1	102	1	99.0%
	2	9	82	90.1%
	Overall Percentage	57.2%	42.8%	94.8%
Test	1	47	0	100.0%
	2	10	49	83.1%
	Overall Percentage	53.8%	46.2%	90.6%

Growing Method: CRT

Dependent Variable: Diagnosis

Classification np = 20, nc=10

Sample	Observed	Predicted		
		1	2	Percent Correct
Training	1	98	7	93.3%
	2	4	90	95.7%
	Overall Percentage	51.3%	48.7%	94.5%
Test	1	38	7	84.4%
	2	5	51	91.1%
	Overall Percentage	42.6%	57.4%	88.1%

Growing Method: CRT

Dependent Variable: Diagnosis

Classification np=25, nc=12

Sample	Observed	Predicted		
		1	2	Percent Correct
Training	1	83	18	82.2%
	2	1	101	99.0%
	Overall Percentage	41.4%	58.6%	90.6%
Test	1	40	9	81.6%
	2	3	45	93.8%
	Overall Percentage	44.3%	55.7%	87.6%

Growing Method: CRT

Dependent Variable: Diagnosis

Classification t1 np=30, nc=15

Predicted

Sample	Observed	1	2	Percent Correct
Training	1	99	7	93.4%
	2	12	83	87.4%
	Overall Percentage	55.2%	44.8%	90.5%
Test	1	41	3	93.2%
	2	6	49	89.1%
	Overall Percentage	47.5%	52.5%	90.9%

Growing Method: CRT

Dependent Variable: Diagnosis

Classification t2 np=30, nc=15

		Predicted		
Sample	Observed	1	2	Percent Correct
Training	1	87	7	92.6%
	2	9	93	91.2%
	Overall Percentage	49.0%	51.0%	91.8%
Test	1	53	3	94.6%
	2	9	39	81.3%
	Overall Percentage	59.6%	40.4%	88.5%

Growing Method: CRT

Dependent Variable: Diagnosis

Classification np=35, nc=17

		Predicted		
Sample	Observed	1	2	Percent Correct
Training	1	94	5	94.9%
	2	11	77	87.5%
	Overall Percentage	56.1%	43.9%	91.4%
Test	1	46	5	90.2%

	2	7	55	88.7%
	Overall Percentage	46.9%	53.1%	89.4%

Growing Method: CRT

Dependent Variable: Diagnosis

Classification t1 np=40,nc=20

		Predicted		
Sample	Observed	1	2	Percent Correct
Training	1	88	7	92.6%
	2	11	86	88.7%
	Overall Percentage	51.6%	48.4%	90.6%
Test	1	52	3	94.5%
	2	7	46	86.8%
	Overall Percentage	54.6%	45.4%	90.7%

Growing Method: CRT

Dependent Variable: Diagnosis

Classification t2 np=40, nc=20

		Predicted		
Sample	Observed	1	2	Percent Correct
Training	1	100	6	94.3%
	2	8	81	91.0%
	Overall Percentage	55.4%	44.6%	92.8%
Test	1	40	4	90.9%
	2	10	51	83.6%
	Overall Percentage	47.6%	52.4%	86.7%

Growing Method: CRT

Dependent Variable: Diagnosis

Classification np=45, nc=22

		Predicted		
Sample	Observed	1	2	Percent Correct
Training	1	95	5	95.0%
	2	11	88	88.9%
	Overall Percentage	53.3%	46.7%	92.0%
Test	1	45	5	90.0%
	2	7	44	86.3%
	Overall Percentage	51.5%	48.5%	88.1%

Growing Method: CRT

Dependent Variable: Diagnosis

Classification np=50, nc=25

		Predicted		
Sample	Observed	1	2	Percent Correct
Training	1	93	7	93.0%
	2	11	91	89.2%
	Overall Percentage	51.5%	48.5%	91.1%
Test	1	47	3	94.0%
	2	7	41	85.4%
	Overall Percentage	55.1%	44.9%	89.8%

Growing Method: CRT

Dependent Variable: Diagnosis

Classification np=55, nc=22

		Predicted		
Sample	Observed	1	2	Percent Correct
Training	1	88	7	92.6%
	2	11	86	88.7%
	Overall Percentage	51.6%	48.4%	90.6%

Test	1	52	3	94.5%
	2	7	46	86.8%
	Overall Percentage	54.6%	45.4%	90.7%

Growing Method: CRT

Dependent Variable: Diagnosis

Classification np=60, nc=30

		Predicted		
Sample	Observed	1	2	Percent Correct
Training	1	87	7	92.6%
	2	10	83	89.2%
	Overall Percentage	51.9%	48.1%	90.9%
Test	1	53	3	94.6%
	2	8	49	86.0%
	Overall Percentage	54.0%	46.0%	90.3%

Growing Method: CRT

Dependent Variable: Diagnosis

Classification np=16, nc=8

		Predicted		
Sample	Observed	1	2	Percent Correct
Training	1	85	16	84.2%
	2	2	91	97.8%
	Overall Percentage	44.8%	55.2%	90.7%
Test	1	38	11	77.6%
	2	2	55	96.5%
	Overall Percentage	37.7%	62.3%	87.7%

Growing Method: CRT

Dependent Variable: Diagnosis

Classification np=14, nc=7

		Predicted		
Sample	Observed	1	2	Percent Correct
Training	1	92	11	89.3%
	2	4	101	96.2%
	Overall Percentage	46.2%	53.8%	92.8%
Test	1	44	3	93.6%
	2	5	40	88.9%
	Overall Percentage	53.3%	46.7%	91.3%

Growing Method: CRT

Dependent Variable: Diagnosis

Classification np=13, nc=6

		Predicted		
Sample	Observed	1	2	Percent Correct
Training	1	101	0	100.0%
	2	12	85	87.6%
	Overall Percentage	57.1%	42.9%	93.9%
Test	1	48	1	98.0%
	2	7	46	86.8%
	Overall Percentage	53.9%	46.1%	92.2%

Growing Method: CRT

Dependent Variable: Diagnosis

Classification np=12, nc=6

		Predicted		
Sample	Observed	1	2	Percent Correct
Training	1	107	1	99.1%
	2	6	92	93.9%

	Overall Percentage	54.9%	45.1%	96.6%
Test	1	42	0	100.0%
	2	6	46	88.5%
	Overall Percentage	51.1%	48.9%	93.6%

Growing Method: CRT

Dependent Variable: Diagnosis

Risk np=12, nc=6

Sample	Estimate	Std. Error
Training	.034	.013
Test	.064	.025

Growing Method: CRT

Dependent Variable: Diagnosis