

Assignment 3**Lavinia Wang #1473704****Due Date:** Saturday, February 17th, 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 “sldata” 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:

Model Summary		
Specifications	Growing Method	CRT
	Dependent Variable	V12
	Independent Variables	V1, V2, V3, V4, V5, V6, V7, V8, V9, V10, V11
	Validation	Split Sample
	Maximum Tree Depth	20
	Minimum Cases in Parent Node	9
	Minimum Cases in Child Node	5
	Results	
	Independent Variables Included	V10, V1, V7, V11, V9, V6, V5, V3, V2, V8, V4
	Number of Nodes	11
	Number of Terminal Nodes	6
	Depth	3

Table 1: Final decision tree model summary for Lupus data

Training Sample

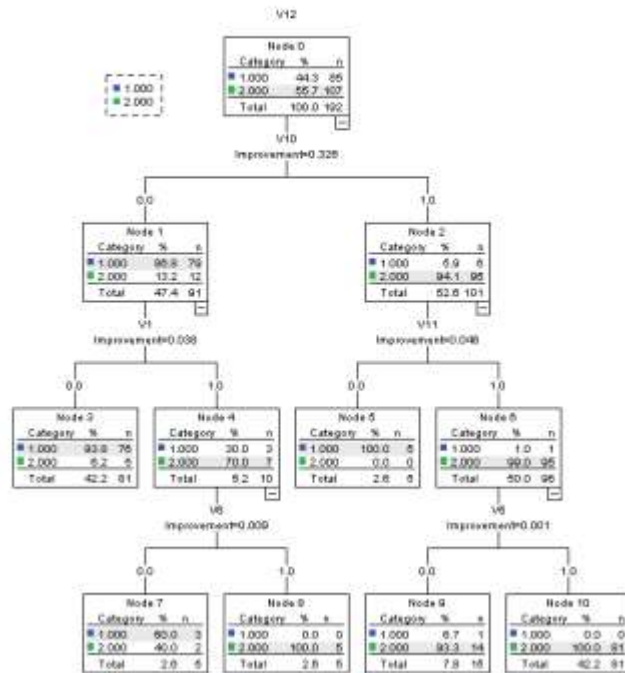


Figure 1: Final decision tree for Lupus data training sample

Test Sample

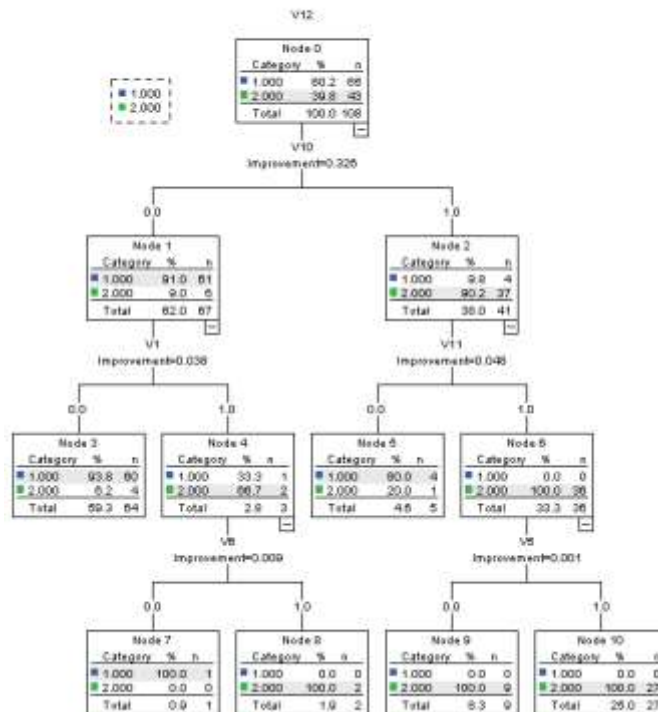


Figure 2: Final decision tree for Lupus data testing sample

Risk

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

Growing Method: CRT

Dependent Variable: V12

*Table 2: Final decision tree model for Lupus data risk table***Classification**

Sample Observed		Predicted		
		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: V12

*Table 3: Final decision tree model for Lupus data misclassification matrix***Independent Variable Importance**

Independent Variable	Importance	Normalized Importance
V10	.326	100.0%
V11	.233	71.5%
V1	.191	58.5%
V7	.149	45.6%
V9	.120	36.7%
V6	.109	33.4%
V3	.080	24.4%
V5	.072	22.0%
V2	.043	13.2%
V8	.039	12.1%
V4	.026	7.9%

Growing Method: CRT

Dependent Variable: V12

Table 4: Final decision tree model for Lupus data independent variable importance

K-fold Cross Validation(K=3)			Hold-out Partitioning (66%)			
	Accuracy	Complexity		Training	Testing	Complexity
N _P =25	92.30%	3	N _P =25	91.60%	88.70%	3
N _C =12			N _C =12			
N _P =20	95.00%	5	N _P =20	91.60%	93.60%	4
N _C =10			N _C =10			
N _P =15	95.00%	5	N _P =15	96.00%	89.70%	6
N _C =7			N _C =7			
N _P =10	95.70%	6	N _P =10	97.30%	92.30%	7
N _C =5			N _C =5			
N _P =9	95.70%	6	N _P =9	95.80%	95.40%	6
N _C =5			N _C =5			
N _P =8	95.70%	7	N _P =8	97.50%	92.20%	8
N _C =4			N _C =4			
N _P =7	95.70%	7	N _P =7	98.00%	91.00%	5
N _C =4			N _C =4			
N _P =6	95.70%	7	N _P =6	94.40%	98.10%	7
N _C =3			N _C =3			
N _P =5	95.70%	7	N _P =5	97.20%	96.50%	9
N _C =3			N _C =3			

Table 5: Decision tree model for Lupus data accuracy and complexity comparison

- 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)

My decision tree models were built using both K-fold cross validation with k=3 and hold-out partitioning validation. From k-fold cross validation models, we can see from case (N_P=10, N_C=5), the accuracy stays still. While in hold-out partitioning models, accuracy between training and testing is always changing. So my final model criteria is based on the least difference between training and testing with less complexity.

The final decision tree was built with a depth of 3; Minimum cases in parent node of 9; Minimum cases in Child node of 5 with a growing method of CRT (CRT growing method attempts to maximize within-node homogeneity) and the default impurity index of GINI.

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

The final tree has 11 (including the Root node) nodes with 6 of those being terminal nodes.

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

The most important features in building the lupus tree are V10, V11, & V1 as these are at the top nodes of the decision tree since a decision tree implicitly performs variable selection.

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

It is more obvious in the K-fold cross validation models that the complexity of the decision tree decreased (with less nodes) when the number of cases for each parent and child increased, because increasing the cases creates wider bins to fit more data and essentially partitioning the data less specifically.

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.

quality					
		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	

Table 6: Frequency table of variable quality of red wine dataset

Statistics		
quality		
N	Valid	1599
	Missing	0
Mean		5.64
Median		6.00
Mode		5
Std. Deviation		.808
Variance		.652
Skewness		.218
Std. Error of Skewness		.061
Kurtosis		.297
Std. Error of Kurtosis		.122
Minimum		3
Maximum		8

Table 7: Descriptive of quality distribution

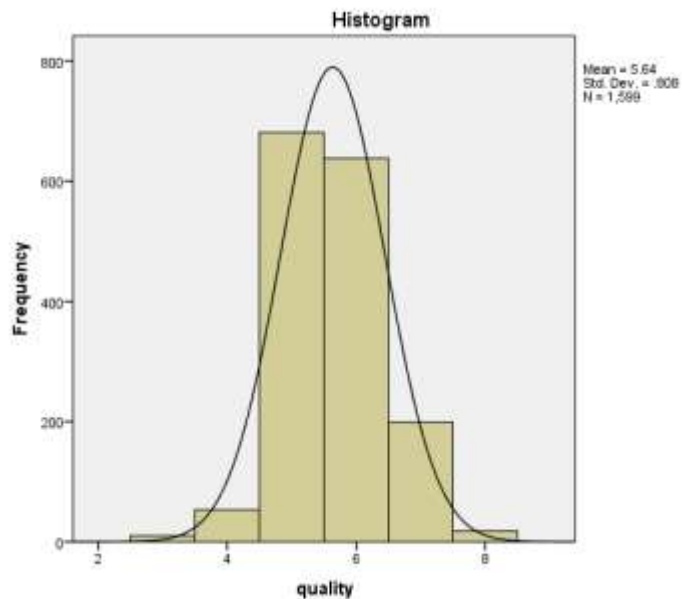



Figure 3: Distribution of quality

There are 6 classes within the quality variable. From figure 3, histogram of quality we can see a minimum of 3, maximum of 8 giving a range of 5. The mode is in class 5 and class 6 is slightly smaller. The distribution is symmetric with very thin tail meaning the majority of the data is distributed in the center.

2. Repeat **Problem 1** on the red wine data. 

Model Summary		
Specifications	Growing Method	CRT
	Dependent Variable	quality
	Independent Variables	fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol
	Validation	Split Sample
	Maximum Tree Depth	20
	Minimum Cases in Parent Node	20
	Minimum Cases in Child Node	10
Results	Independent Variables Included	alcohol, total sulfur dioxide, density, sulphates, chlorides, volatile acidity, pH, free sulfur dioxide, residual sugar, citric acid, fixed acidity
	Number of Nodes	57
	Number of Terminal Nodes	29
	Depth	7

Table 8: Final decision tree model summary for red wine data

Training Sample

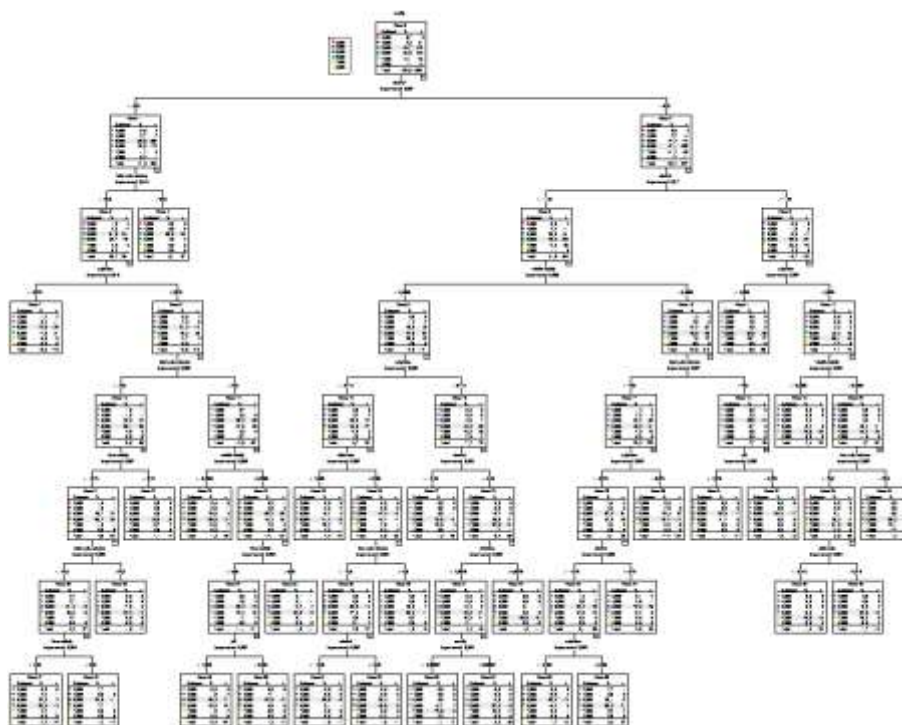


Figure 4: Final decision tree for red wine data training sample

Test Sample

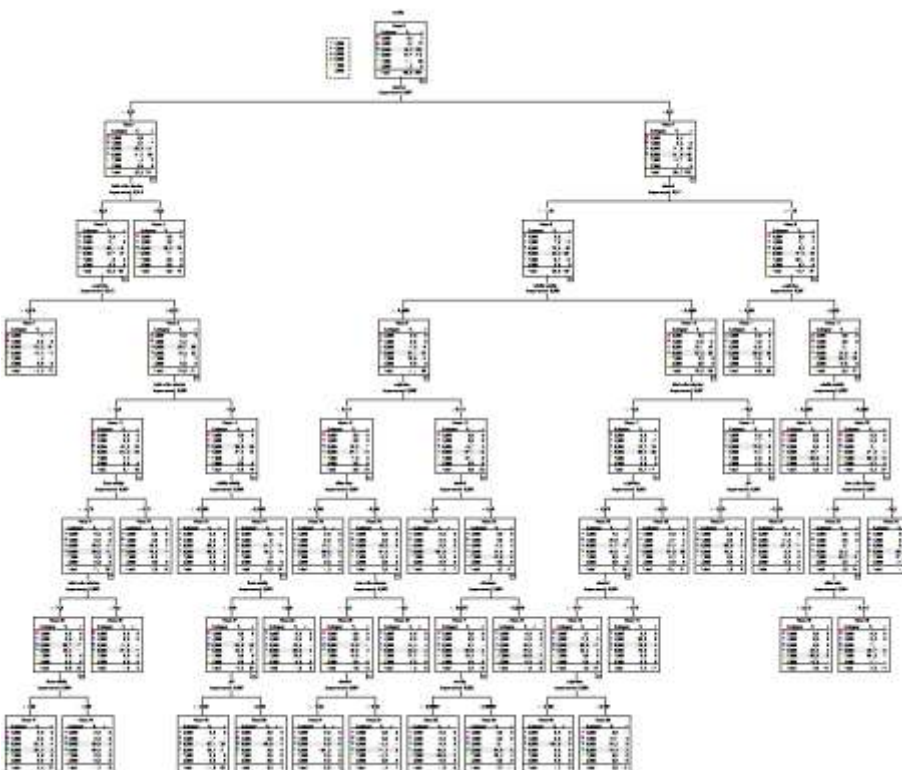


Figure 5: Final decision tree for red wine data testing sample

Risk		
Sample	Estimate	Std. Error
Training	.318	.014
Test	.375	.022

Growing Method: CRT

Dependent Variable: quality

Table 9: Final decision tree model for red wine data risk table

Classification								
Sample	Observed	Predicted						Percent Correct
		3	4	5	6	7	8	
Training	3	0	0	5	3	0	0	0.0%
	4	0	0	18	13	0	0	0.0%
	5	0	0	360	116	2	0	75.3%
	6	0	0	81	333	9	0	78.7%
	7	0	0	3	86	54	0	37.8%
	8	0	0	0	8	4	0	0.0%
	Overall Percentage	0.0%	0.0%	42.6%	51.1%	6.3%	0.0%	68.2%
Test	3	0	0	1	1	0	0	0.0%
	4	0	0	14	8	0	0	0.0%
	5	0	0	138	63	2	0	68.0%
	6	0	0	48	157	10	0	73.0%
	7	0	0	1	35	20	0	35.7%
	8	0	0	0	3	3	0	0.0%
	Overall Percentage	0.0%	0.0%	40.1%	53.0%	6.9%	0.0%	62.5%

Growing Method: CRT

Dependent Variable: quality

Table 10: Final decision tree model for red wine data misclassification matrix

Independent Variable Importance		
Independent Variable	Importance	Normalized Importance
alcohol	.095	100.0%
sulphates	.072	75.9%
total sulfur dioxide	.063	66.2%
volatile acidity	.061	63.5%

fixed acidity	.028	29.5%
density	.026	26.8%
free sulfur dioxide	.024	25.2%
citric acid	.021	22.2%
residual sugar	.017	18.2%
chlorides	.017	17.3%
pH	.013	13.2%

Growing Method: CRT

Dependent Variable: quality

Table 11: Final decision tree model for red wine data independent variable importance

K-fold Cross Validation(K=10)			Hold-out Partitioning (66%)			
	Accuracy	Complexity		Training	Testing	Complexity
N _P =35	67.00%	28	N _P =35	63.40%	55.50%	17
N _C =18			N _C =18			
N _P =25	68.40%	35	N _P =25	67.70%	57.30%	21
N _C =12			N _C =12			
N _P =20	68.80%	39	N _P =20	68.20%	62.50%	29
N _C =10			N _C =10			
N _P =15	69.90%	45	N _P =15	68.80%	56.80%	33
N _C =7			N _C =7			
N _P =10	73.00%	62	N _P =10	74.20%	55.50%	51
N _C =5			N _C =5			
N _P =9	73.00%	62	N _P =9	75.90%	56.90%	52
N _C =5			N _C =5			
N _P =8	75.90%	79	N _P =8	78.20%	60.60%	70
N _C =4			N _C =4			
N _P =7	75.90%	79	N _P =7	74.50%	59.00%	60
N _C =4			N _C =4			
N _P =6	81.90%	120	N _P =6	81.10%	61.50%	85
N _C =3			N _C =3			
N _P =5	81.90%	120	N _P =5	81.70%	59.40%	90
N _C =3			N _C =3			


Table 12: Decision tree model for red wine data accuracy and complexity comparison

My decision tree models were built using both K-fold cross validation with k=10 and hold-out partitioning validation. In hold-out partitioning models, accuracy of training data is increasing when we decrease the number of cases for each parent and child, but accuracy of testing data fluctuates around 60%. We can conclude from case (N_P=15, N_C=7) overfitting is generated. So

my final model criteria is based on the least difference between training and testing with less complexity.

The final decision tree was built with depth of 7; Minimum cases in parent node of 20; Minimum cases in Child node of 10 with a growing method of CRT (CRT growing method attempts to maximize within-node homogeneity) and the default impurity index of GINI.

The final tree has 57 (including the Root node) nodes with 29 of those being terminal nodes.

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). 

	Original data	New data
New class 1	3 & 4	4
New class 2	5 & 6	5
New class 3	7 & 8	7

Table 13: Partitioning into bins with equal width with mode

The data is partitioned into bins using equal-width partitioning and the data were smoothed by mode.

Model Summary		
Specifications	Growing Method	CRT
	Dependent Variable	bin_quality
	Independent Variables	fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol
	Validation	Split Sample
	Maximum Tree Depth	20
	Minimum Cases in Parent Node	20
Results	Minimum Cases in Child Node	10
	Independent Variables Included	alcohol, density, fixed acidity, chlorides, pH, volatile acidity, residual sugar, total sulfur dioxide, sulphates, citric acid, free sulfur dioxide
	Number of Nodes	23
	Number of Terminal Nodes	12
	Depth	5

Table 14: Final decision tree model summary for binned red wine data

Training Sample

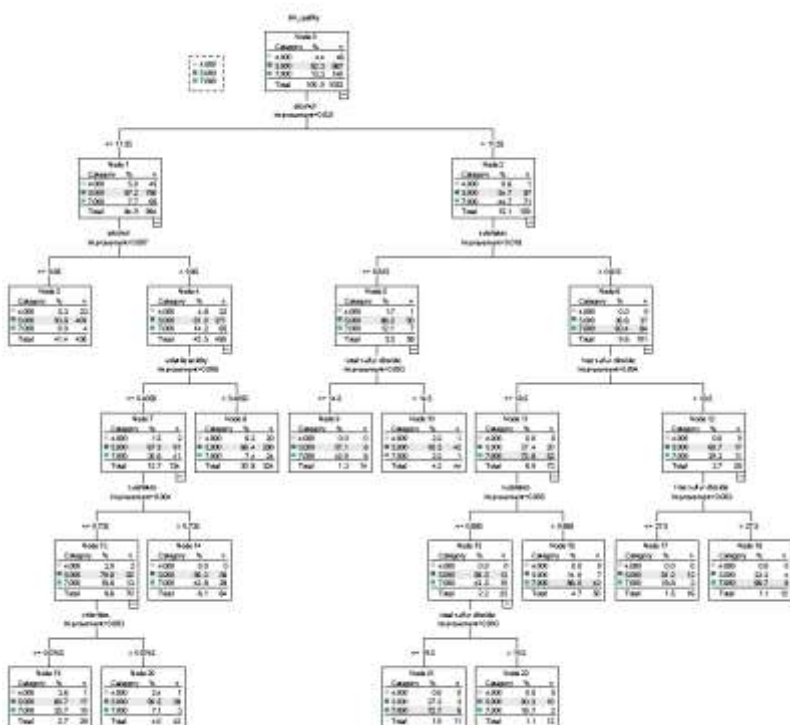


Figure 15: Final decision tree for binned red wine data training sample

Test Sample

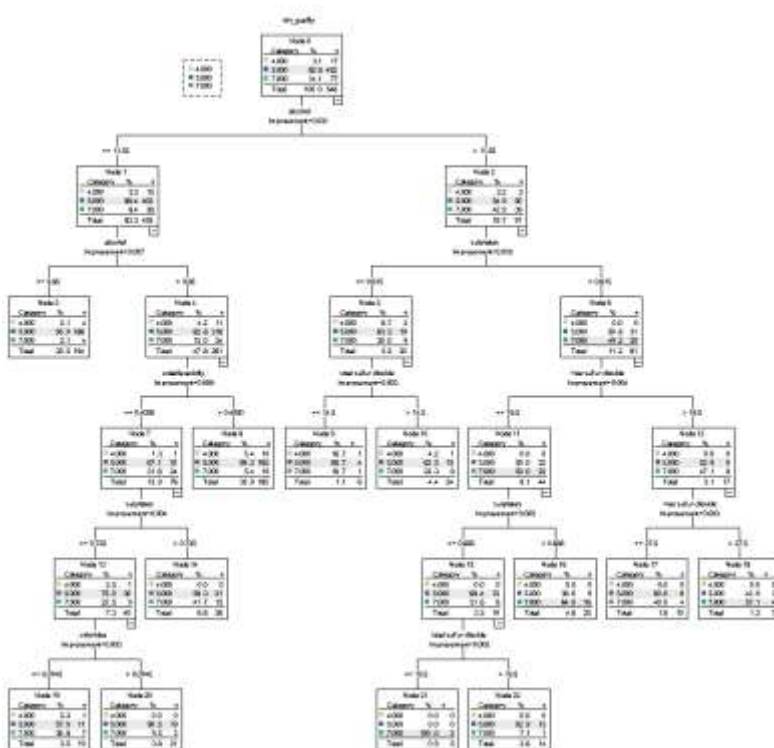


Figure 16: Final decision tree for binned red wine data testing sample

Risk		
Sample	Estimate	Std. Error
Training	.134	.010
Test	.148	.015

Growing Method: CRT

Dependent Variable: bin_quality

Table 15: Final decision tree model for binned red wine data risk table

Classification					
Sample	Observed	Predicted			
		4	5	7	Percent Correct
Training	4	0	46	0	0.0%
	5	0	853	14	98.4%
	7	0	81	59	42.1%
	Overall Percentage	0.0%	93.1%	6.9%	86.6%
Test	4	0	17	0	0.0%
	5	0	440	12	97.3%
	7	0	52	25	32.5%
	Overall Percentage	0.0%	93.2%	6.8%	85.2%

Growing Method: CRT

Dependent Variable: bin_quality

Table 16: Final decision tree model for binned red wine data misclassification matrix

Independent Variable Importance		
Independent Variable	Importance	Normalized Importance
alcohol	.042	100.0%
sulphates	.033	79.7%
volatile acidity	.019	45.8%
total sulfur dioxide	.015	35.9%
fixed acidity	.014	33.3%
citric acid	.014	32.8%
chlorides	.013	30.8%
free sulfur dioxide	.013	30.7%
density	.009	22.5%
pH	.006	13.3%
residual sugar	.002	3.7%

Growing Method: CRT

Dependent Variable: bin_quality

Table 17: Final decision tree model for binned red wine data independent variable importance

K-fold Cross Validation(K=10)			Hold-out Partitioning (66%)			
	Accuracy	Complexity		Training	Testing	Complexity
N _P =35	86.30%	11	N _P =35	85.60%	83.50%	7
N _C =18			N _C =18			
N _P =25	87.20%	21	N _P =25	86.60%	84.20%	11
N _C =12			N _C =12			
N _P =23	87.20%	21	N _P =23	87.60%	83.80%	12
N _C =12			N _C =12			
N _P =20	87.50%	23	N _P =20	86.60%	85.20%	12
N _C =10			N _C =10			
N _P =15	88.20%	28	N _P =15	87.30%	84.30%	15
N _C =7			N _C =7			
N _P =10	89.40%	36	N _P =10	89.80%	82.10%	34
N _C =5			N _C =5			
N _P =9	89.40%	36	N _P =9	88.20%	82.20%	21
N _C =5			N _C =5			
N _P =8	90.20%	40	N _P =8	90.60%	84.30%	29
N _C =4			N _C =4			
N _P =7	90.20%	40	N _P =7	89.00%	82.70%	22
N _C =4			N _C =4			
N _P =6	92.00%	52	N _P =6	92.40%	81.90%	38
N _C =3			N _C =3			


Table 18: Decision tree model for binned red wine data accuracy and complexity comparison

My decision tree models were built using both K-fold cross validation with k=10 and hold-out partitioning validation. In hold-out partitioning models, accuracy of training data is increasing when we decrease the number of cases for each parent and child, accuracy of testing data is also improved. We can observe from case (NP=10, Nc=5) the difference between training and testing increased again. So my final model criteria is based on the least difference between training and testing with less complexity.

The final decision tree was built with depth of 5; Minimum cases in parent node of 20; Minimum cases in Child node of 10 with a growing method of CRT (CRT growing method attempts to maximize within-node homogeneity) and the default impurity index of GINI.

4. How the performance of the best classification model on the original class variable compares with the accuracy of the best classification model on the binned classification variable?

There is a huge improvement in accuracy after binning the quality. The accuracy on training increased from 68.2% to 86.8%, which is 18.6% growth and that on testing increased from 62.5% to 85.2%, which is 22.7% growth.

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.

I changed the dependent variable quality class into binary variable, which has only two level. 0 represents “not good” quality with scores range from 0 to 6 and 1 represent “good” quality with scores higher or equal to 7.

The final decision tree was built with depth of 4; Minimum cases in parent node of 35; Minimum cases in Child node of 18 with a growing method of CRT (CRT growing method attempts to maximize within-node homogeneity) and the default impurity index of GINI. Complexity was decreased to 9. Testing accuracy increased by 3.6%.

Model Summary		
Specifications	Growing Method	CRT
	Dependent Variable	binary_quality
	Independent Variables	fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol
	Validation	Split Sample
	Maximum Tree Depth	20
Results	Minimum Cases in Parent Node	35
	Minimum Cases in Child Node	18
	Independent Variables Included	alcohol, density, chlorides, fixed acidity, pH, volatile acidity, total sulfur dioxide, citric acid, residual sugar, free sulfur dioxide, sulphates
	Number of Nodes	17
	Number of Terminal Nodes	9
	Depth	4

Table 19: Improved decision tree model summary for binned red wine data

Training Sample

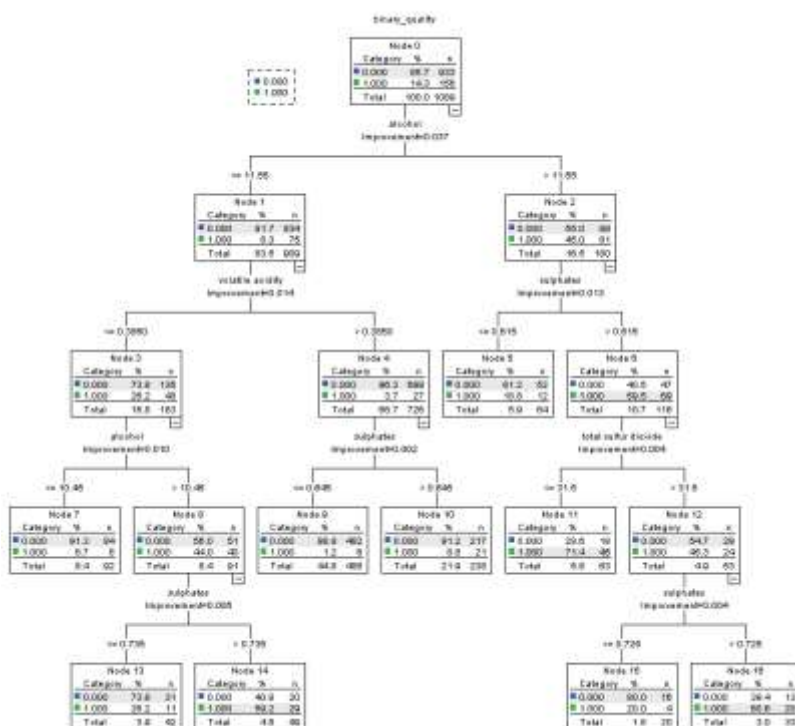


Figure 17: Improved decision tree for binned red wine data training sample

Test Sample

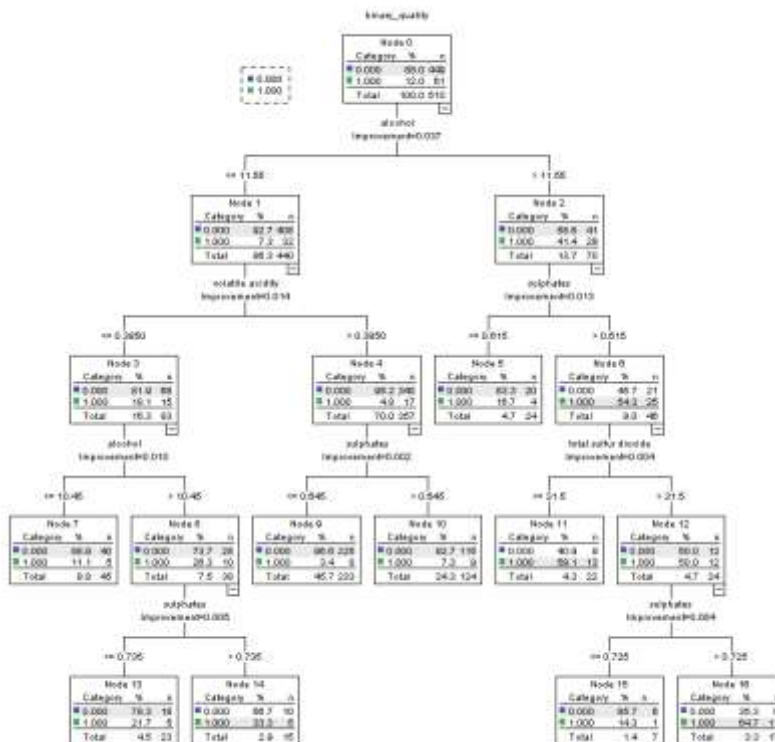


Figure 18: Improved decision tree for binned red wine data testing sample

Risk		
Sample	Estimate	Std. Error
Training	.104	.009
Test	.112	.014

Growing Method: CRT

Dependent Variable: binary_quality

Table 20: Improved decision tree model for binned red wine data risk table

Classification				
Sample	Observed	Predicted		
		0	1	Percent Correct
Training	0	882	51	94.5%
	1	62	94	60.3%
	Overall Percentage	86.7%	13.3%	89.6%
Test	0	424	25	94.4%
	1	32	29	47.5%
	Overall Percentage	89.4%	10.6%	88.8%

Growing Method: CRT

Dependent Variable: binary_quality

Table 21: Improved decision tree model for binned red wine data misclassification matrix

Independent Variable Importance		
Independent Variable	Importance	Normalized Importance
alcohol	.049	100.0%
sulphates	.028	58.5%
volatile acidity	.022	44.8%
citric acid	.013	27.2%
chlorides	.011	23.4%
total sulfur dioxide	.011	21.7%
fixed acidity	.010	20.5%
density	.007	14.3%
residual sugar	.007	13.9%
pH	.006	13.2%
free sulfur dioxide	.005	9.4%

Growing Method: CRT

Dependent Variable: binary_quality

Table 22: Improved decision tree model for binned red wine data independent variable importance

Problem 3 (5 points): Differentiate between the following terms:

a. clustering and classification

Classification method belongs to the supervised learning category. In general, in classification you have a set of predefined labels and want to know which label a new object belongs to. Algorithms applied in classification problems include Decision Tree, K-nearest neighbor, Neural Networks, Supportive Vector Machines, and Random Forest etc. Clustering, aka unsupervised learning, tries to group a set of objects and find whether there is some relationship between the objects. There is no predefined label in clustering. Algorithms applied include K-means clustering and hierarchical clustering.

b. training and testing

Training (data) is a partition of the data to which a model is built on and refined and the testing (data) is the data that run on completed model for verification/validation of accuracy.

c. parametric reduction techniques and non-parametric reduction techniques

Parametric reduction techniques include: N(mean, standard deviation) and regression models which assume a certain model form to represent original data and store only model parameters to achieve reduction. Non-parametric reduction techniques include: histogram, clustering and sampling which does not assume any model initially and make fewer assumptions about the data, and thus can be applicable in more scenarios.

d. principal component analysis and forward selection

Principal component analysis is a method used in feature extraction, meaning “combining” the essence of attributes by creating an alternative, smaller set of variables. Forward selection is a method used in feature selection when we run regression model to find statistically significant parameters.

e. covariance matrix and correlation matrix

Correlation is the normalized covariance. A covariance matrix is a more generalized form of a simple correlation matrix. Correlation matrix has a fixed range $[-1, 1]$ where -1 means the two variables are inversely correlated and 1 means the two variables are positively correlated with 0 meaning no correlation. In covariance matrix, the minimum is 0 and no maximum bound.