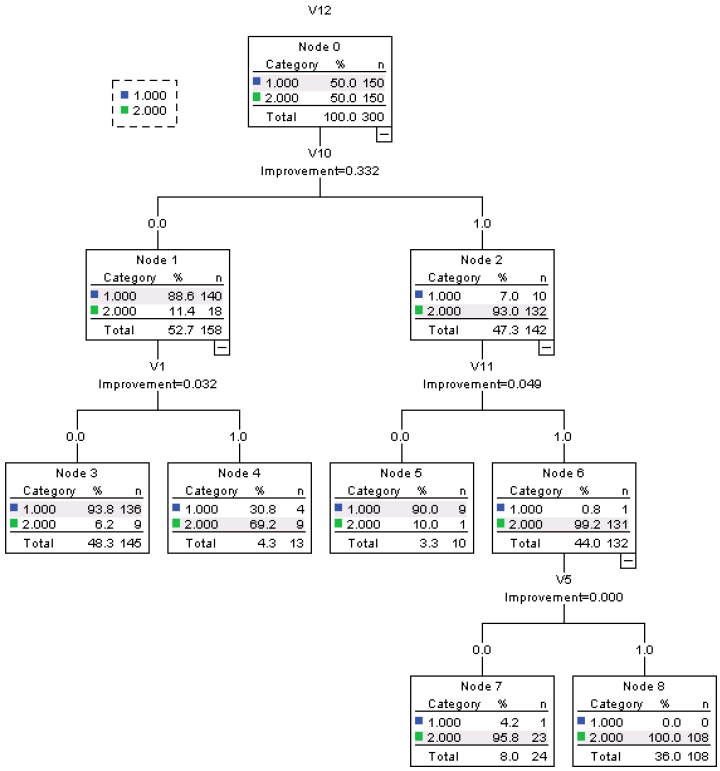
Assignment 3- IS467

Sarah Cummings

Problem 1- Decision Tree for lupus data

1. Decision tree and the criteria used for building the tree:



Classification

Observed	Predicted 1	Predicted 2	Percent Correct
1	145	5	96.7%
2	10	140	93.3%
Overall Percentage	51.7%	48.3%	95.0%

Growing Method: CRT
Dependent Variable: V12

Risk

Estimate	Std. Error
.050	.013

Growing Method: CRT
Dependent Variable: V12

For this decision tree, I used the CRT method with Gini as the impurity measure (splits are found that maximize the homogeneity of the child nodes with respect to the value of the target varaible. I used the automatic maximum tree depth, and 20/ 10 for minimum number of cases for parent/child nodes. I also used the automatic minimum change in improvement of 0.0001.

2. Number of nodes on final tree— how many are terminal:

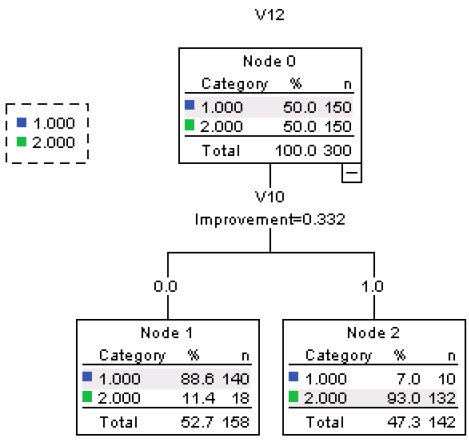
We had 9 nodes in the final tree, of which 5 are terminal.

3. Three most important features in building the tree:

The three most important features in building the three were variable 10, variable 1, and variable 11. There three features were considered the most important because using the Gini index, our algorithm chose them as the first three variables used for splitting.

4. Increase the number of cases for each parent and child. What do you notice with the complexity (number of nodes) of the tree?

I increased the parent/ child minimum to 50/25 and it decreased the number of nodes in the three. While my original tree had nine nodes, my new tree has three nodes. This is because further levels of the tree would have required child nodes with less than 25 cases. Since we do not have any more variables that fit our criteria, the tree must stop at the second level.



Problem 2- Wine quality classification and decision tree

1. Number of classes and distribution:

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	

Note there are six classes, quality values 3-8. These classes are very imbalanced. The majority of the wine fell into quality classes 5 and 6 (42.6% and 39.9% respectively). Only 0.6% of the wine fell into value 3, and 1.1% in value 8.

2. Decision tree:

i. Tree and criteria used:

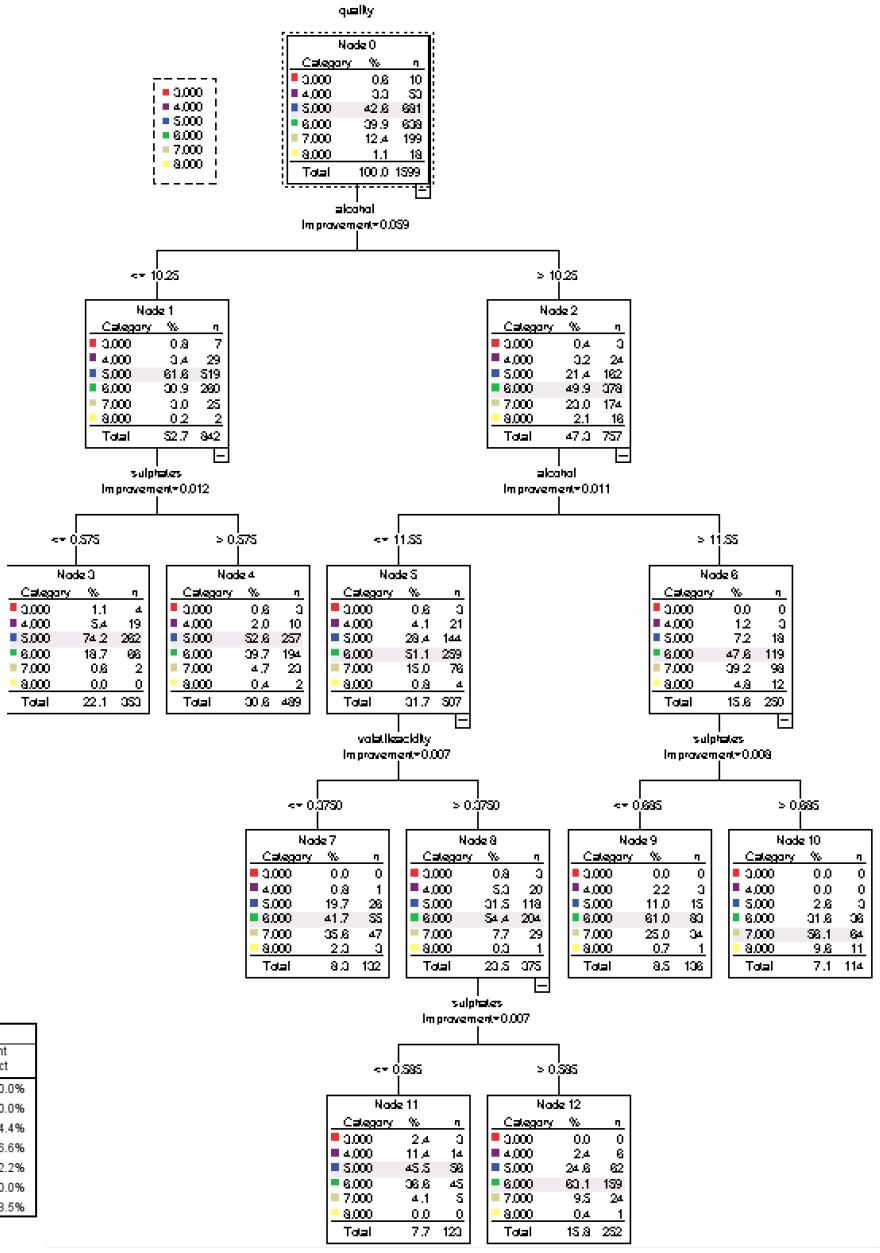
I used minimum number of parent cases of 200 and minimum number of child cases 100. Again I used CRT, with Gini as the impurity measure, the automatic maximum tree depth, and minimum change in improvement of 0.0001. Tree at right:

ii. Number of nodes on final tree: 13, of which 7 are terminal.

iii. Three most important features: alcohol, sulfates, and volatile acidity. The alcohol value was used to split the data in the first level and part of the second level, meaning that by our criteria it was the most relevant attribute. Then, sulphates and volatile acidity were used next to split the data.

iv. Change in complexity after increasing cases:

For this data, the change in cases had a dramatic affect of the number of nodes and complexity of our tree. I initially made a tree with 50/25 as the minimum number of parent/ child cases and the tree was huge.

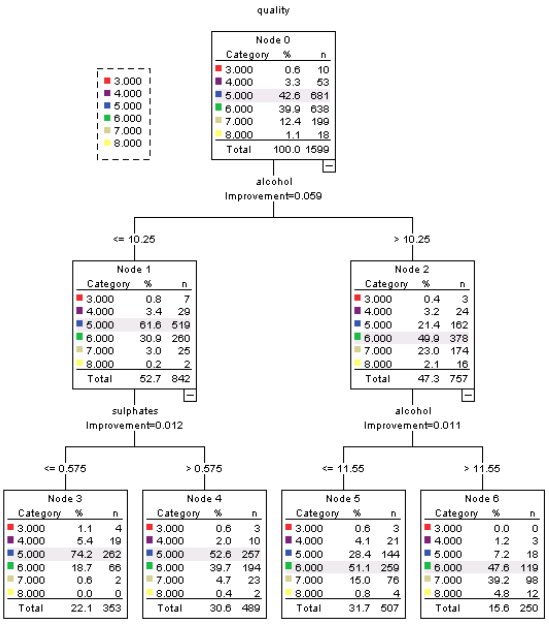


Classification

Observed	Predicted 3	4	5	6	7	8	Percent Correct
3	0	0	10	0	0	0	0.0%
4	0	0	43	10	0	0	0.0%
5	0	0	575	103	3	0	84.4%
6	0	0	305	297	36	0	46.6%
7	0	0	30	105	64	0	32.2%
8	0	0	2	5	11	0	0.0%
Overall Percentage	0.0%	0.0%	60.4%	32.5%	7.1%	0.0%	58.5%

Growing Method: CRT
Dependent Variable: quality

The tree below was creating using minimum parent cases of 500 and child cases 250. As you can see, the tree shrinks in size as it did for our other data set.



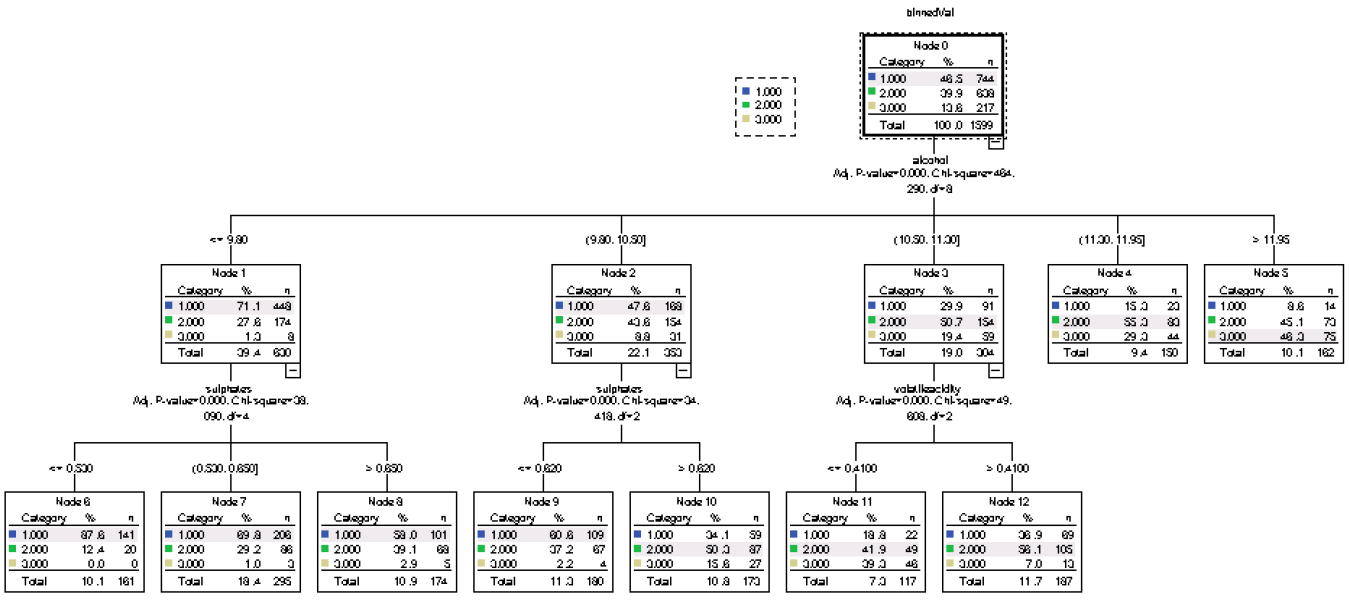
Classification

Observed	Predicted						
	3	4	5	6	7	8	Percent Correct
3	0	0	7	3	0	0	0.0%
4	0	0	29	24	0	0	0.0%
5	0	0	519	162	0	0	76.2%
6	0	0	260	378	0	0	59.2%
7	0	0	25	174	0	0	0.0%
8	0	0	2	16	0	0	0.0%
Overall Percentage	0.0%	0.0%	52.7%	47.3%	0.0%	0.0%	56.1%

Growing Method: CRT
Dependent Variable: quality

3. Bin and repeat:

The binning process was tricky for me. After binning the classes, the best percentage correctness I could get for my tree was 85%, but that tree had 0% correctness for my first bin. I decided to stick with a lower overall accuracy that did a better job at predicting that first bin.
I binned 3-4 together to represent bad wine (labeled as bin 1), 5-6 to represent average wine(bin 2), and 7-8 to represent good wine (bin 3). The following tree was obtained, using CHAID method, 200 minimum parent cases and 100 minimum child cases:



Classification

Observed	Predicted			
	1.00	2.00	3.00	Percent Correct
1.00	557	173	14	74.9%
2.00	241	324	73	50.8%
3.00	12	130	75	34.6%
Overall Percentage	50.7%	39.2%	10.1%	59.8%

Growing Method: CHAID
Dependent Variable: binnedVal

Model Summary

Specifications	Growing Method	CHAID
	Dependent Variable	binnedVal
	Independent Variables	fixedacidity, volatileacidity, citricacid, residualsugar, chlorides, freesulfurdioxide, totalsulfurdioxide, density, pH, sulphates, alcohol
	Validation	None
	Maximum Tree Depth	3
	Minimum Cases in Parent Node	200
	Minimum Cases in Child Node	100
Results	Independent Variables Included	alcohol, sulphates, volatileacidity
	Number of Nodes	13
	Number of Terminal Nodes	9
	Depth	2

4. The binning tree is a bit better than the non-binned tree, but I am still not completely satisfied with the results. The initial tree had a hard time predicting the first bin class, and I had to increase the minimum case required for parent and child nodes quite a bit before it was accurately predicting for the first bin at all.

5. I would consider binning in such a way that the bins were more uniform in frequency. Though it made most sense to me to divide into three bins the way I did, perhaps 5-6 quality scored cases should be separated so that the bins are even closer to being equal.

Problem 3- Differentiate between:

1. feature selection and feature extraction

Both feature selection and feature extraction are methods to reduce dimensionality. Feature selection methods reduce dimensionality by selecting and keeping only the most important features of the data. This method does no transformation of the current features, but rather keeps just a subset of those current features. Feature extraction, however, reduces dimensionality by transforming the existing features into a lower dimension space. Principal component analysis is a method of feature extraction.

2. training and testing

In data analysis, the training set is used to create a model and the testing set is used to test that model. Generally, the training set is a random subset of an initial set of data.

3. parametric reduction techniques and non-parametric reduction techniques

There are two types of statistical methods for outlier detection, parametric reduction and non-parametric reduction methods. While all methods for outlier detection involve learning a general model and finding values that have low-probability of occurrence, the methods for learning these general models vary. Parametric methods assume that the normal data objects are generated by a parametric distribution with a given parameter. Then, a probability density function formed from the parameter gives the probability that an object is generated by this distribution. The smaller this probability, the greater chance that the object is an outlier. A non-parametric method tries to determine the model from the output data. These methods also often assume the number and nature of the parameters are flexible.

4. uniform binning and non-uniform binning

The two most common ways to bin noisy data are equal depth binning and equal width binning. In equal width binning, the bins all have a uniform size/range. The biggest problem with this type of binning is that outliers may dominate the bin, and skewed data is not represented very well. In equal depth binning, we bin such that each has the same number of observations. These bins are hence uniform in their frequency counts. This binning is much better for skewed data and has great scalability, but often has issues when it comes to categorical attributes. Both methods are useful, and the dat scientist should consider the data before picking a binning method.

5. covariance matrix and correlation matrix

Correlation and covariance are both importance in the data integration process as they help us detect redundancy in our data. Generally, we use a correlation matrix when variables are on the same scale, and we use a covariance matrix when the data is normalized.
Given x1 and x2, covariance(x1, x2)= (x11- x1bar)(x21-x2bar)+… /n-1.
The covariance matrix is the same as the correlation matrix if the variables are all normalized.
The correlation matrix gives the Pearson correlation between each variable, r, where $-1 \leq r \leq 1$.