Mike Araujo, Jeff Horne and Robert Francalangia

We decided on implementing a version of the OneR algorithm in C++. The algorithm works by classifying on the last value of the instance. On a training set of instances, the size of it which is decided by the user, the rule of the algorithm is developed. This classifying attribute must only have two values. For each attribute, the algorithm looks at each value of that attribute and makes a rule based on how often the class appears for that attribute. It counts it up and places the value into one of the two classifying categories. Based on what category is the majority, the algorithm makes a rule assigned to that attribute. If there is a tie then we determine the rule by selecting the first instance's class value. From there, we calculate the error rate by comparing the actual values to the rule. The rule that is picked to classify the test data set is the attribute rule with the lowest error rate. Error rate is determined by taking incorrectly classified instances in the attribute and dividing it by total instances in that attribute. The rule is then used on test set, which is all the instances that are not in the training set.

Compile command:

```
g++ -g -Wall oneR.cc Instance.cc linked_list.h -o  oneR
```

Run command:

```
./oneR
```

OneR Assumptions:

1. No numeric values, nominal values only
2. Population size is limited to 1000 instances
3. Attributes per instance is limited to 100 different attributes
4. Values per attribute limited to 100
5. Classification is binary, it can either be a 1 or a 0. This can signify yes or no, (this is the vernacular used in our code) play or don't play, recommend or don't recommend
6. The last value in an instance is what we classify on
7. We do not handle missing values, the data set must have a value for all attributes
8. When selecting a training set size, user must pick a percentage that is at least large enough that there will be enough instances to cover all attribute values. For example, if there is a data set with size 10 with an attribute with 3 values, the user must pick at least 30% because the algorithm needs to train on all three of those values before it can make a rule
9. User must spell .arff file correctly or the program will not work
10. No comments allowed in the .arff files

Our code began with creating an Instance class, the instance class contains 4 private variables and multiple public member functions, including a default constructor. On top of this class creation, our code can be broken up into multiple different stages. The stage progression is as follows: variable declaration, read in, split up, train the algorithm, test the algorithm, and print the results.

Class:

The Instance class was key to the correct implementation of this algorithm. Each instance has a two parallel arrays of strings the first array is the array attributes, paralleled by the array of values. The attribute array is essentially the labels of each attribute and is the same for all instances from each data set. The value array is unique per instance and holds the values to of the specific instance. The next private variable is an integer value called num_vals, this is the number of attributes, essentially this is the length of the attributes class. Finally there is the boolean variable called used, this is to signify whether the certain instance has been used in the training data set. The default constructor initializes all string values to “empty” and integer values to 0, and the booleans to false. There is an Init_Instance function that initializes the instance according to the values the read in function places into the holder arrays. There is a print_info function that prints the information about a single instance. There is a Set_Equal function that sets copies an instance and its values to another instance. There is a use function that sets the used boolean to true. There is a function that returns a boolean called Is_Used and then returns the value for used. There are Get_Value and Get_Att functions that take an integer parameter and returns the string from the values or attribute array at the position equal to the integer.

Variable declaration:

The program begins with variable declarations needed during the body of main.

Read In:

The read in portion is the code directly follows the variable declaration. It begins with prompting the user to input a file name. The file needs to be within the current folder the file is executing in. Furthermore, the user should be careful to make sure the data set being used obeys all of the assumptions made at the beginning of this report. Once the program has the file, it begins to parse the data. The program begins by creating an attribute holder array. This array will later be copied into each instance and be placed into the attribute array. The final step in the read in stage, the program collects the data values and begins populating the array of instances.

Split Up:

Once the program has placed the data values into the arrays, the program prompts the user to input the a percent of the population of data the user wishes to train the algorithm on. The user inputs a number between 10 and 90. The program then randomly assigns instances to the training set keeping track of which instances were moved, so as to sample without replacement, while these instances are moved the boolean variable used is changed to true. Then the program places the complements into the test array.

Train the Algorithm:

This is the true implementation of the algorithm. The program begins iterating through each instance, beginning with the first attribute. It checks the first attribute value of all the training data instances. Each value is put into a node, creating a linked list, that has a string

value, and two integer variables. Each integer value corresponds to a class, and are incremented respectively. The program then chooses the rule with the smallest error rate. It does this by saving the first error rate into a variable called error_rate, and after each attribute's error rate is tabulated the program checks whether or not it is less than the already saved error rate. To prepare to test the algorithm the program creates a best list that contains the values and classes for that attribute.

Test the Algorithm:

With the remaining portion of the data, the program applies the rule it generated and checks the actual class and generates a success rate. When handling attributes that are split, the algorithm selects the first classifying value in the data set. This may change depending on the randomization of the instances.

One of our greatest strategies in determining whether our code was working, was we trained on 100% of the data. This is not an effective way to generate real rules but using a data set with a small number of instances gave us a great idea of where our algorithm stood with respect to completeness. As you can see from the data below, when we trained the algorithm on 100% of the data the algorithm matched Weka each time.

We used supervised learning, so we already knew the results and could check to make sure that what we were doing was correct.

Comparison of our learning algorithm to other small datasets:

To begin, our original assumption we made decided that our algorithm could only classify data based on a binary scale. The class could only be a zero or one. So, a blind guess would average 50% correct. The two data sets we found in Weka that satisfied the assumptions were the non numeric weather data and the voter data. We had to tweak the voter data a little because some of the values were missing, we combed through the data and deleted all instances that had a case of missing values. The resulting data set had 116 instances. We also modified the contact lense data to make it binary, on the basis of should the individual need contacts or not.

We ran the Weka algorithms many times and realized their percent correct does not change. This must mean that the selection of the training data set in Weka is algorithmic and consistently picks the same rule. Our algorithm assigns the training set completely random and consequently, our training set may be unbiased because it was assigned completely randomly. Conversely the training set could mean it is not consistent. Weka probably has an algorithm that creates a training set that is both an unbiased and consistent representation of the data set as a whole. Our algorithm does not have this processing power and consequently whilst running simulations the rules and rate correct varied widely. Once, when using a 30% training set, our algorithm picked four instances out of the weather set the and developed a rule based on windy. The rule was if windy = false then play = no, and if windy = true then play = yes. It randomly picked the four instances that would in most cases be the outlier. To confidently get the best rule within our algorithm you should run it many times and record the rules and success rates.

Weather Data Correctly Classified N = 14

Percent Trained	Weka	Our Algorithm
30%	4/10 = 40%	5/10 = 50%
60%	2/6 = 33.33%	3/6 = 50%
90%	0/1 = 0.0%	1 / 2 = 50%
100%	10/14 = 71.43%	10/14 = 71.43%

Contact Lenses Data Correctly Classified N = 24

Percent Trained	Weka	Our Algorithm
30%	7/17 = 41.17%	14/17 = 82.35%
60%	9/10 = 90%	8/10 = 80%
90%	2/2 = 100%	3/3 = 100%
100%	21/24 = 87.5%	21/24 = 87.5%

Voter Data Correctly Classified N = 116

Percent Trained	Weka	Our Algorithm
10%	89/104 = 85.57%	103/105 = 98.0952%
30%	80/81 = 98.76%	80/82 = 97.561%
60%	45/46 = 97.82%	46/47 = 97.8723%
90%	12/12 = 100%	12/12 = 100%
100%	114/116 = 98.27%	114/116 = 98.27%

Our results match very closely what WEKA produces. This is not surprising as we believe our algorithm to closely match WEKA's. One major difference we see is how the algorithms treat rounding. Ours places the extra values in the testing set when there is a fraction in the split values, while WEKA places the rounded values in the training set. Algorithms that have less assumptions may be better if the data is missing values. Data that has attributes with many

values may not be suitable for OneR. We see in the above tables a lower coverage in the data sets with more values, but great coverage in the Voter Data data set which only has two values for each attributes. If it is the case that there are many values for the attributes it may be more helpful to use a decision tree or naive bayes than a rule based algorithm.

Division of Labor:

Mike began the project by writing the read in stage. Jeff and Rob took up the baton and tried to debug this stage, though they were unsuccessful. Rob wrote the original split up code. Jeff wrote the code to read in a file from the user input. Jeff and Rob attempted to understand pointers enough so that they could begin implementing the actual algorithm while Mike added these two pieces of code to the main file. Mike then took over and implemented the training, testing, and all the other portions of the code. Meanwhile, Jeff and Rob wrote the report.

Works Cited

CPlusPlus.com; http://www.cplusplus.com/reference/istream/istream/. Accessed 2017 April 3

CPlusPlus.com; http://www.cplusplus.com/reference/ios/boolalpha/. Accessed 2017 April 3

CPlusPlus.com; http://www.cplusplus.com/reference/cstring/. Accessed 2017 April 3

Labs and Assignments from CSCI-131 and CSCI-132, taught by Laurie Smith-King in the Fall of 2015, and Saba Kadady in the Spring of 2016 respectively.

Stock, James H, and Mark W. Watson. *Introduction to Econometrics* 3rd ed. (Boston:Addison-Wesley, 2011). 107-144

Witten, Ian H, Eibe Frank, and Mark A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. (Amsterdam: Elsevier, 2011), Chapter 4

SAMPLE OUTPUT:
Note: Training instances and testing instances are printed but will not fit in the terminal for vote.arff (too many instances)

Vote.arff output:

10%

The one rule is 'physician-fee-freeze'
'n': 'democrat'
'y': 'republican'

Number Correctly Classfied: 103
Number Incorrecly Classified: 2
Rate Correct: 98.0952%

30%

The one rule is 'physician-fee-freeze'
'n': 'democrat'
'y': 'republican'

Number Correctly Classfied: 80
Number Incorrecly Classified: 2
Rate Correct: 97.561%

60%

The one rule is 'physician-fee-freeze'
'y': 'republican'

'n': 'democrat'

Number Correctly Classfied: 46
Number Incorrecly Classified: 1
Rate Correct: 97.8723%

90%

The one rule is 'physician-fee-freeze'
'y': 'republican'
'n': 'democrat'

Number Correctly Classfied: 12
Number Incorrecly Classified: 0
Rate Correct: 100%

100%

The one rule is 'physician-fee-freeze'
'n': 'democrat'
'y': 'republican'

Number Correctly Classfied: 114
Number Incorrecly Classified: 2
Rate Correct: 98.2759%

Contact Lenses:
[rcfran17@radius finalHw3]$ ./oneR
Which File do you want to use? contact-lenses.arff
enter a percentage of the list to train on (integers only):
30
Printing Training Set

age
young
spectacle-prescrip
hypermetrope
astigmatism
yes
tear-prod-rate
reduced
contact-lenses
none

age
presbyopic
spectacle-prescrip
hypermetrope
astigmatism
no
tear-prod-rate
reduced
contact-lenses
none

age
young
spectacle-prescrip
myope
astigmatism
no
tear-prod-rate
normal
contact-lenses
contacts

age

presbyopic
spectacle-prescrip
hypermetrope
astigmatism
yes
tear-prod-rate
reduced
contact-lenses
none

age
young
spectacle-prescrip
hypermetrope
astigmatism
yes
tear-prod-rate
normal
contact-lenses
contacts

age
pre-presbyopic
spectacle-prescrip
hypermetrope
astigmatism
yes
tear-prod-rate
reduced
contact-lenses
none

age
young
spectacle-prescrip

myope
astigmatism
yes
tear-prod-rate
normal
contact-lenses
contacts

Printing Testing Set

age
young
spectacle-prescrip
myope
astigmatism
no
tear-prod-rate
reduced
contact-lenses
none

age
pre-presbyopic
spectacle-prescrip
myope
astigmatism
no
tear-prod-rate
normal
contact-lenses
contacts

age
presbyopic

spectacle-prescrip
myope
astigmatism
no
tear-prod-rate
reduced
contact-lenses
none

age
pre-presbyopic
spectacle-prescrip
myope
astigmatism
yes
tear-prod-rate
reduced
contact-lenses
none

age
pre-presbyopic
spectacle-prescrip
hypermetrope
astigmatism
no
tear-prod-rate
normal
contact-lenses
contacts

age
pre-presbyopic
spectacle-prescrip
hypermetrope

astigmatism
no
tear-prod-rate
reduced
contact-lenses
none

age
pre-presbyopic
spectacle-prescrip
hypermetrope
astigmatism
yes
tear-prod-rate
normal
contact-lenses
none

age
young
spectacle-prescrip
myope
astigmatism
yes
tear-prod-rate
reduced
contact-lenses
none

age
presbyopic
spectacle-prescrip
myope
astigmatism
no

tear-prod-rate
normal
contact-lenses
none

age
pre-presbyopic
spectacle-prescrip
myope
astigmatism
no
tear-prod-rate
reduced
contact-lenses
none

age
pre-presbyopic
spectacle-prescrip
myope
astigmatism
yes
tear-prod-rate
normal
contact-lenses
contacts

age
presbyopic
spectacle-prescrip
myope
astigmatism
yes
tear-prod-rate
normal

contact-lenses
contacts

age
presbyopic
spectacle-prescrip
myope
astigmatism
yes
tear-prod-rate
reduced
contact-lenses
none

age
presbyopic
spectacle-prescrip
hypermetrope
astigmatism
no
tear-prod-rate
normal
contact-lenses
contacts

age
young
spectacle-prescrip
hypermetrope
astigmatism
no
tear-prod-rate
reduced
contact-lenses
none

age
presbyopic
spectacle-prescrip
hypermetrope
astigmatism
yes
tear-prod-rate
normal
contact-lenses
none

age
young
spectacle-prescrip
hypermetrope
astigmatism
no
tear-prod-rate
normal
contact-lenses
contacts

The one rule is tear-prod-rate
reduced: none
normal: contacts

Number Correctly Classified: 14
Number Incorrectly Classified: 3
Rate Correct: 82.3529%
[rcfran17@radius finalHw3]$

[rcfran17@radius finalHw3]$ ./oneR
Which File do you want to use? contact-lenses.arff

enter a percentage of the list to train on (integers only):
60
Printing Training Set

age
pre-presbyopic
spectacle-prescrip
hypermetrope
astigmatism
yes
tear-prod-rate
normal
contact-lenses
none

age
presbyopic
spectacle-prescrip
myope
astigmatism
yes
tear-prod-rate
reduced
contact-lenses
none

age
presbyopic
spectacle-prescrip
hypermetrope
astigmatism
no
tear-prod-rate
reduced
contact-lenses

none

age
pre-presbyopic
spectacle-prescrip
myope
astigmatism
yes
tear-prod-rate
normal
contact-lenses
contacts

age
presbyopic
spectacle-prescrip
myope
astigmatism
no
tear-prod-rate
reduced
contact-lenses
none

age
young
spectacle-prescrip
myope
astigmatism
no
tear-prod-rate
reduced
contact-lenses
none

age
young
spectacle-prescrip
myope
astigmatism
yes
tear-prod-rate
normal
contact-lenses
contacts

age
presbyopic
spectacle-prescrip
hypermetrope
astigmatism
yes
tear-prod-rate
reduced
contact-lenses
none

age
pre-presbyopic
spectacle-prescrip
myope
astigmatism
no
tear-prod-rate
normal
contact-lenses
contacts

age
pre-presbyopic

spectacle-prescrip
hypermetrope
astigmatism
no
tear-prod-rate
reduced
contact-lenses
none

age
presbyopic
spectacle-prescrip
myope
astigmatism
yes
tear-prod-rate
normal
contact-lenses
contacts

age
pre-presbyopic
spectacle-prescrip
myope
astigmatism
no
tear-prod-rate
reduced
contact-lenses
none

age
pre-presbyopic
spectacle-prescrip
hypermetrope

astigmatism
no
tear-prod-rate
normal
contact-lenses
contacts

age
young
spectacle-prescrip
myope
astigmatism
yes
tear-prod-rate
reduced
contact-lenses
none

Printing Testing Set

age
presbyopic
spectacle-prescrip
myope
astigmatism
no
tear-prod-rate
normal
contact-lenses
none

age
young
spectacle-prescrip

myope
astigmatism
no
tear-prod-rate
normal
contact-lenses
contacts

age
young
spectacle-prescrip
hypermetrope
astigmatism
no
tear-prod-rate
normal
contact-lenses
contacts

age
presbyopic
spectacle-prescrip
hypermetrope
astigmatism
no
tear-prod-rate
normal
contact-lenses
contacts

age
presbyopic
spectacle-prescrip
hypermetrope
astigmatism

yes
tear-prod-rate
normal
contact-lenses
none

age
pre-presbyopic
spectacle-prescrip
myope
astigmatism
yes
tear-prod-rate
reduced
contact-lenses
none

age
young
spectacle-prescrip
hypermetrope
astigmatism
yes
tear-prod-rate
normal
contact-lenses
contacts

age
pre-presbyopic
spectacle-prescrip
hypermetrope
astigmatism
yes
tear-prod-rate

reduced
contact-lenses
none

age
young
spectacle-prescrip
hypermetrope
astigmatism
yes
tear-prod-rate
reduced
contact-lenses
none

age
young
spectacle-prescrip
hypermetrope
astigmatism
no
tear-prod-rate
reduced
contact-lenses
none

The one rule is tear-prod-rate
normal: contacts
reduced: none

Number Correctly Classfied: 8
Number Incorrecly Classified: 2
Rate Correct: 80%
[rcfran17@radius finalHw3]$

[rcfran17@radius finalHw3]$ ./oneR
Which File do you want to use? contact-lenses.arff
enter a percentage of the list to train on (integers only):
90
Printing Training Set

age
young
spectacle-prescrip
myope
astigmatism
no
tear-prod-rate
reduced
contact-lenses
none

age
presbyopic
spectacle-prescrip
myope
astigmatism
yes
tear-prod-rate
normal
contact-lenses
contacts

age
pre-presbyopic
spectacle-prescrip
hypermetrope
astigmatism

no
tear-prod-rate
reduced
contact-lenses
none

age
young
spectacle-prescrip
myope
astigmatism
yes
tear-prod-rate
reduced
contact-lenses
none

age
presbyopic
spectacle-prescrip
hypermetrope
astigmatism
no
tear-prod-rate
reduced
contact-lenses
none

age
young
spectacle-prescrip
hypermetrope
astigmatism
yes
tear-prod-rate

reduced
contact-lenses
none

age
young
spectacle-prescrip
hypermetrope
astigmatism
yes
tear-prod-rate
normal
contact-lenses
contacts

age
pre-presbyopic
spectacle-prescrip
myope
astigmatism
no
tear-prod-rate
normal
contact-lenses
contacts

age
pre-presbyopic
spectacle-prescrip
hypermetrope
astigmatism
yes
tear-prod-rate
reduced
contact-lenses

none

age
young
spectacle-prescrip
hypermetrope
astigmatism
no
tear-prod-rate
reduced
contact-lenses
none

age
pre-presbyopic
spectacle-prescrip
myope
astigmatism
yes
tear-prod-rate
reduced
contact-lenses
none

age
pre-presbyopic
spectacle-prescrip
myope
astigmatism
yes
tear-prod-rate
normal
contact-lenses
contacts

age
presbyopic
spectacle-prescrip
hypermetrope
astigmatism
yes
tear-prod-rate
normal
contact-lenses
none

age
young
spectacle-prescrip
myope
astigmatism
no
tear-prod-rate
normal
contact-lenses
contacts

age
presbyopic
spectacle-prescrip
myope
astigmatism
yes
tear-prod-rate
reduced
contact-lenses
none

age
presbyopic

spectacle-prescrip
myope
astigmatism
no
tear-prod-rate
normal
contact-lenses
none

age
pre-presbyopic
spectacle-prescrip
hypermetrope
astigmatism
yes
tear-prod-rate
normal
contact-lenses
none

age
presbyopic
spectacle-prescrip
myope
astigmatism
no
tear-prod-rate
reduced
contact-lenses
none

age
young
spectacle-prescrip
myope

astigmatism
yes
tear-prod-rate
normal
contact-lenses
contacts

age
young
spectacle-prescrip
hypermetrope
astigmatism
no
tear-prod-rate
normal
contact-lenses
contacts

age
presbyopic
spectacle-prescrip
hypermetrope
astigmatism
no
tear-prod-rate
normal
contact-lenses
contacts

Printing Testing Set

age
pre-presbyopic
spectacle-prescrip

hypermetrope
astigmatism
no
tear-prod-rate
normal
contact-lenses
contacts

age
presbyopic
spectacle-prescrip
hypermetrope
astigmatism
yes
tear-prod-rate
reduced
contact-lenses
none

age
pre-presbyopic
spectacle-prescrip
myope
astigmatism
no
tear-prod-rate
reduced
contact-lenses
none

The one rule is tear-prod-rate
reduced: none
normal: contacts

Number Correctly Classfied: 3
Number Incorrecly Classified: 0
Rate Correct: 100%
[rcfran17@radius finalHw3]$

Which File do you want to use? contact-lenses.arff
enter a percentage of the list to train on (integers only):
100
Printing Training Set

age
presbyopic
spectacle-prescrip
hypermetrope
astigmatism
no
tear-prod-rate
normal
contact-lenses
contact

age
presbyopic
spectacle-prescrip
myope
astigmatism
yes
tear-prod-rate
reduced
contact-lenses
none

age
young
spectacle-prescrip

myope
astigmatism
yes
tear-prod-rate
reduced
contact-lenses
none

age
young
spectacle-prescrip
hypermetrope
astigmatism
no
tear-prod-rate
reduced
contact-lenses
none

age
young
spectacle-prescrip
hypermetrope
astigmatism
no
tear-prod-rate
normal
contact-lenses
contact

age
pre-presbyopic
spectacle-prescrip
hypermetrope
astigmatism

yes
tear-prod-rate
reduced
contact-lenses
none

age

young
spectacle-prescrip
myope
astigmatism
no
tear-prod-rate
reduced
contact-lenses
none

age
presbyopic
spectacle-prescrip
myope
astigmatism
no
tear-prod-rate
reduced
contact-lenses
none

age
pre-presbyopic
spectacle-prescrip
hypermetrope
astigmatism
no

tear-prod-rate
reduced
contact-lenses
none

age
pre-presbyopic
spectacle-prescrip
myope
astigmatism
yes
tear-prod-rate
normal
contact-lenses
contact

age
pre-presbyopic
spectacle-prescrip
myope
astigmatism
no
tear-prod-rate
normal
contact-lenses
contact

age
young
spectacle-prescrip
hypermetrope
astigmatism
yes
tear-prod-rate
reduced

contact-lenses
none

age
pre-presbyopic
spectacle-prescrip
myope
astigmatism
yes
tear-prod-rate
reduced
contact-lenses
none

age
pre-presbyopic
spectacle-prescrip
hypermetrope
astigmatism
no
tear-prod-rate
normal
contact-lenses
contact

age
young
spectacle-prescrip
hypermetrope
astigmatism
yes
tear-prod-rate
normal
contact-lenses
contact

age
pre-presbyopic
spectacle-prescrip
hypermetrope
astigmatism
yes
tear-prod-rate
normal
contact-lenses
none

age
presbyopic
spectacle-prescrip
myope
astigmatism
yes
tear-prod-rate
normal
contact-lenses
contact

age
young
spectacle-prescrip
myope
astigmatism
no
tear-prod-rate
normal
contact-lenses
contact

age

presbyopic
spectacle-prescrip
hypermetrope
astigmatism
yes
tear-prod-rate
normal
contact-lenses
none

age
presbyopic
spectacle-prescrip
hypermetrope
astigmatism
yes
tear-prod-rate
reduced
contact-lenses
none

age
presbyopic
spectacle-prescrip
hypermetrope
astigmatism
no
tear-prod-rate
reduced
contact-lenses
none

age
pre-presbyopic
spectacle-prescrip

myope
astigmatism
no
tear-prod-rate
reduced
contact-lenses
none

age
young
spectacle-prescrip
myope
astigmatism
yes
tear-prod-rate
normal
contact-lenses
contact

age
presbyopic
spectacle-prescrip
myope
astigmatism
no
tear-prod-rate
normal
contact-lenses
none

Printing Testing Set

Testing being performed on training

The one rule is tear-prod-rate
normal: contact
reduced: none

Number Correctly Classfied: 21
Number Incorrecly Classified: 3
Rate Correct: 87.5%

Weather Data
[rcfran17@radius finalHw3]$ ./oneR
Which File do you want to use? weather.nominal.arff
enter a percentage of the list to train on (integers only):
30
Printing Training Set

outlook
overcast
temperature
cool
humidity
normal
windy
TRUE
play
yes

outlook
overcast
temperature
hot
humidity
high
windy

FALSE
play
yes

outlook
rainy
temperature
cool
humidity
normal
windy
FALSE
play
yes

outlook
sunny
temperature
cool
humidity
normal
windy
FALSE
play
yes

Printing Testing Set

outlook
sunny
temperature
mild
humidity
high

windy
FALSE
play
no

outlook
overcast
temperature
hot
humidity
normal
windy
FALSE
play
yes

outlook
sunny
temperature
hot
humidity
high
windy
TRUE
play
no

outlook
sunny
temperature
mild
humidity
normal
windy
TRUE

play
yes

outlook
sunny
temperature
hot
humidity
high
windy
FALSE
play
no

outlook
rainy
temperature
mild
humidity
normal
windy
FALSE
play
yes

outlook
rainy
temperature
mild
humidity
high
windy
TRUE
play
no

outlook
rainy
temperature
cool
humidity
normal
windy
TRUE
play
no

outlook
rainy
temperature
mild
humidity
high
windy
FALSE
play
yes

outlook
overcast
temperature
mild
humidity
high
windy
TRUE
play
yes

The one rule is outlook
overcast: yes
rainy: yes
sunny: yes

Number Correctly Classfied: 5
Number Incorrecly Classified: 5
Rate Correct: 50%
[rcfran17@radius finalHw3]$

[rcfran17@radius finalHw3]$ ./oneR
Which File do you want to use? weather.nominal.arff
enter a percentage of the list to train on (integers only):
60
Printing Training Set

outlook
overcast
temperature
hot
humidity
high
windy
FALSE
play
yes

outlook
rainy
temperature
mild
humidity
normal
windy

FALSE
play
yes

outlook
sunny
temperature
cool
humidity
normal
windy
FALSE
play
yes

outlook
rainy
temperature
mild
humidity
high
windy
TRUE
play
no

outlook
sunny
temperature
mild
humidity
high
windy
FALSE
play

no

outlook
overcast
temperature
mild
humidity
high
windy
TRUE
play
yes

outlook
sunny
temperature
mild
humidity
normal
windy
TRUE
play
yes

outlook
rainy
temperature
mild
humidity
high
windy
FALSE
play
yes

Printing Testing Set

outlook
rainy
temperature
cool
humidity
normal
windy
TRUE
play
no

outlook
sunny
temperature
hot
humidity
high
windy
TRUE
play
no

outlook
overcast
temperature
hot
humidity
normal
windy
FALSE
play
yes

outlook
sunny
temperature
hot
humidity
high
windy
FALSE
play
no

outlook
rainy
temperature
cool
humidity
normal
windy
FALSE
play
yes

outlook
overcast
temperature
cool
humidity
normal
windy
TRUE
play
yes

The one rule is outlook
overcast: yes
rainy: yes
sunny: yes

Number Correctly Classfied: 3
Number Incorrecly Classified: 3
Rate Correct: 50%
[rcfran17@radius finalHw3]$

Which File do you want to use? weather.nominal.arff
enter a percentage of the list to train on (integers only):
100
Printing Training Set

outlook
sunny
temperature
mild
humidity
normal
windy
TRUE
play
yes

outlook
overcast
temperature
hot
humidity
normal
windy
FALSE
play

yes

outlook
sunny
temperature
cool
humidity
normal
windy
FALSE
play
yes

outlook
sunny
temperature
mild
humidity
high
windy
FALSE
play
no

outlook
rainy
temperature
cool
humidity
normal
windy
TRUE
play
no

outlook
rainy
temperature
mild
humidity
high
windy
FALSE
play
yes

outlook
overcast
temperature
cool
humidity
normal
windy
TRUE
play
yes

outlook
rainy
temperature
mild
humidity
normal
windy
FALSE
play
yes

outlook
sunny

temperature
hot
humidity
high
windy
FALSE
play
no

outlook
overcast
temperature
hot
humidity
high
windy
FALSE
play
yes

outlook
rainy
temperature
mild
humidity
high
windy
TRUE
play
no

outlook
overcast
temperature
mild

humidity
high
windy
TRUE
play
yes

outlook
rainy
temperature
cool
humidity
normal
windy
FALSE
play
yes

outlook
sunny
temperature
hot
humidity
high
windy
TRUE
play
no

Printing Testing Set

Testing being performed on training

The one rule is outlook

sunny: no
overcast: yes
rainy: yes

Number Correctly Classfied: 10
Number Incorrecly Classified: 4
Rate Correct: 71.4286%
[rcfran17@radius finalHw3]$