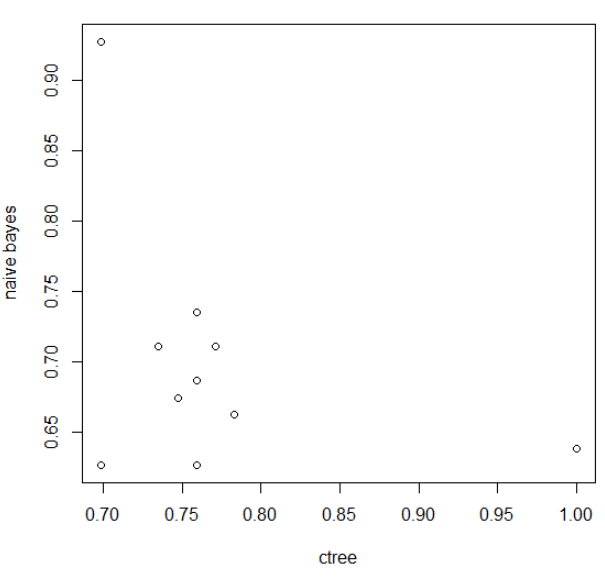
**Homework #5**

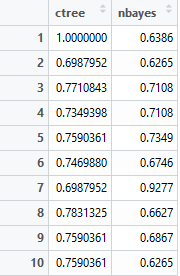
1. For this experiment you will investigate how different types of supervised learning algorithms perform on different data sets with different properties. That is, our goal is to compare Bayesian Learning (Naïve Bayes) with Decision Trees (C5.0). For your experiment in Part 1, pick 30 random seeds (include them in your report), then calculate the average accuracy of Naïve Bayes, C5.0 on each of the two data sets across the 10 runs (one run per seed). Afterwards, compare the average accuracies between the algorithms to evaluate their performances on the different data sets. In particular, evaluate: 1) How did each algorithm perform on “Monks”? Did one type of approach (Bayesian vs. Decision Tree) achieve significantly better performance on this data set? If so, why do you think this happened? If not, what does this say about the two approaches? (20 points)

(“Monks” dataset and information sheet attached)

Please submit R code in addition to answers to questions.

Accuracy results:

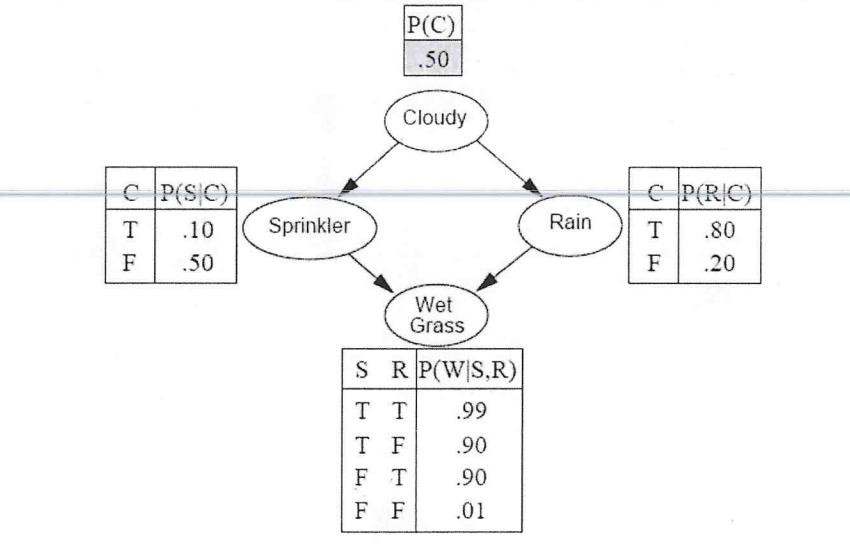


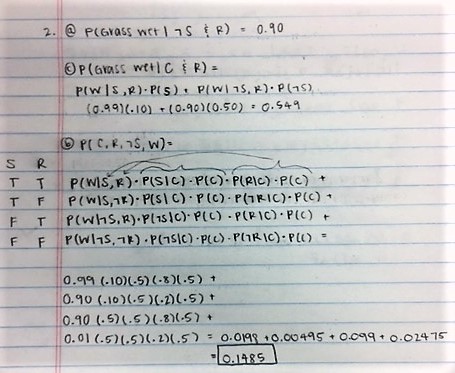


The c tree generally performed better than the naïve bayes algorithm on this dataset. This means that the attributes may be more dependent on each other than other algorithms, so the assumption that all attributes are independent that the naïve bayes algorithm uses may not be applicable for this data set. Both methods had relatively consistent performance with a low variance between accuracies.

1. Consider the rather famous Bayesian network shown below. Please answer the following questions using this network. You may calculate these by hand and submit a legible electronic copy (from a scan or photograph). Show all steps. (10 points)

* How likely is your lawn to be wet if you forget to turn on your sprinkler, but it rains at night?
* What is P(C,R,~S,W)? In other words, how likely is the grass to be wet and it is cloudy and rainy with the sprinkler off?
* Suppose you notice that it is cloudy and rainy. What is the probability that the grass is wet?





1. Take a look at the code for the Naïve Bayes function on page 136 of PDSwR. What does the variable “m” represent in the code in terms of what we learned in class and what you know about how NB works? What does “pPos” represent? Explain what is happening where the function calculates “scorePos” and/or “scoreNeg?” How does this relate to the calculations we made on the board when we studied the *PlayTennis* example and the class out-of-state example? (6 points)

pPos is the posterior probability of a positive outcome, which is equivalent to the calculations of out of state probabilities based on the various attributes summed together. pNeg represents the same calculation for in-state. ScorePos and ScoreNeg are the total likelihoods of the outcome being either positive or negative. This is calcuatled by taking the prior probability of a positive outcome and any given outcome and dividing by the probability of a positive outcome, which is equivalent to the conditional probability that an outcome is positive based on attribute a. This is summed for each attribute and the log is taken at each iteration so the score can be calculated using exponentials and give a product. We did this in the play tennis example when we inserted a new item and determined the probability of it being positive or negative and then chose the max as the final classifier, which is represented as m in the function.

1. Choose a simple data set from the UCI Repository. Use code to create a Bayesian network from the data. What are some basic inferences you are able to make (just two or three)? Please turn in annotated code and show your answer as an annotation. (15 points)

Attached