B365 Hw 4

Zac Monroe

October 2018

1. Done.

2. (a) Run prob2.r to see a printout of this 3D table.

   (b) Source prob2.r in R and then call the function bayes_age_ed() to see the functionality of my Bayes' classifier. The inputs are the age decade of the person and the education level of the person. The classifier outputs N.

   (c) Source prob2.r in R and then call the function bayes() to see the functionality of my more generalized Bayes' classifier. The inputs are the age decade, education level, gender, and region of the person. The classifier outputs N.

   (d) My degree of confidence in this classification is extremely low. Exactly 5 female, post-secondary-educated people from the SA region in their 50's came out to vote and provided a definitive Yes/No vote.

   (e) The prior vote distribution (Y or N) came out to be 887 N and 863 Y. This is roughly 50.7% N and 49.3% Y.

   (f) I made a function for each trait. These functions are seen in prob2.r.

   (g) The naive Bayes' classifier would classify such a person as a No-voter because the probability that each of those traits were reality for a person given that they were a No-voter was higher than them being a Yes-voter. The naive Bayes' classifier said to multiply each of $P(\text{trait}|\text{Cls})$ for each considered trait and take the Cls that produced the maximum value, which is No in our case. Run prob2.r to see this printed out.

3. Run prob3.r to see how my naive Bayes' classifier performs. It is astounding to see how well it performs based on such poor-quality tests. Comments in prob3.r help to explain my logic.

4. (a) This is true. Run prob4.r to see a printout for proof of this.

   (b) Run prob4.r to see an error rate. It is often $\sim$70%.

   (c) The naive Bayes' classifier chooses the most likely class based on the data. The data is that it is almost always more likely that the tests will give a positive result just in general.