Nicholas Fang

A6

1. Explain how to turn P(E|H) to P(H|E), with E=Evidence and H=Hypothesis in layman's terms.
2. Your explanation should be no more than two paragraphs.
3. Use an example from real life to ground the explanation.

The Bayes theorem equation turns to out a simple math equation, more easily visualized when you can plug in some numbers and understand how the equation is interchangeable. This equation is an equation of probabilities and has useful application in classification. P(E|H) can turn into P(H|E) because one conditional probability can be calculated using the other conditional probability through a few simple steps. Just like in multiplication, in order to isolate the variable, you can multiply and divide the equation to get to the desired result.

$$P(E|H) = P(H|E) * P(A) / P(B) \rightarrow P(E|H) * P(B) = P(H|E) * P(A) \rightarrow P(E|H) * P(B) / P(A) = P(H|E)$$

One example from such use cases is in spam filtering for email. In order to find the probability of spam (H) given words (E), we can take this equation given a dataset and compute the right-hand side of the equation. Or given that if one only knows the probability of spam given words P(spam|words), we can multiply the left-hand side P(spam|words) by the P(words) and divide it by the P(spam) in order to isolate that variable and solve for P(words|spam). A simple math equation visualized below:

$$\Pr(\text{spam}|\text{words}) = \frac{\Pr(\text{words}|\text{spam})\Pr(\text{spam})}{\Pr(\text{words})}$$

1. For this exercise use any four of these five datasets to build a spam filter with the Naïve Bayes approach.
2. Use that filter to check the accuracy of the remaining dataset.
3. Make sure to report the details of your training and the model.

A bit more complicated than expected. The training set used 4/5 of the datasets included in the zip file while the testing data was conducted with the Shakira data. Taking the 2 columns from the datasets, CONTEXT and CLASS, I was able to keep track of key words in spam or real messages when classifying the training data. Afterwards, I was able to evaluate probability values to the words given spam, probability of spam, and probability of words in real message. The resulting model produced about an **83.2% accuracy** when used to predict the testing set.

The trickiest part of this spam filtering problem was to separate the words in the CONTENT columns from a single string into separate words. Afterwards, following the NB theorem, I was able to evaluate what the probabilities of different variables from the Bayes equation. Interesting to note that spam messages included a much higher variation of words versus that of words used in real messages. The probability values compared determined whether the message would be classified as spam or as a real message, with a resulting 1 or 0 returned. The average accuracy rates my model returned were fairly accurate. However, with a larger dataset to build a training model off of, the predictions could become more accurate.