CS 383 Assignment 4

Kevin Tayah (kst46@drexel.edu)

March 13, 2021

1. Theory Questions

1. a) The entrophy $H(Y)$ of the training set can be calculated with the following algorithm:

$$H(P(v_1), \ldots, P(v_n)) = \sum_{i=1}^{n} -P(v_i) \log_2 P(v_i)$$
$$H = 2(-\tfrac{3}{21} \log_2 \tfrac{3}{21}) + 2(-\tfrac{4}{21} \log_2 \tfrac{4}{21}) + 2(-\tfrac{1}{21} \log_2 \tfrac{1}{21}) + (-\tfrac{5}{21} \log_2 \tfrac{5}{21})$$
$$H = 2.62499047619$$

b) The weighted average entropies of the class labels of the subsets created by variables x_1 and x_2 are the following: First we must calculate the entropy of each feature when it's is T or F:

$$H_{(x_1,T)} = -\frac{3}{21} \log_2 \frac{3}{21} - \frac{4}{21} \log_2 \frac{4}{21} - \frac{1}{21} \log_2 \frac{1}{21} = 1.06587619048$$

$$H_{(x_1,F)} = -\frac{4}{21} \log_2 \frac{4}{21} - \frac{1}{21} \log_2 \frac{1}{21} - \frac{3}{21} \log_2 \frac{3}{21} - \frac{5}{21} \log_2 \frac{5}{21} = 1.55882857143$$

$$H_{(x_2,T)} = -\frac{3}{21} \log_2 \frac{3}{21} - \frac{4}{21} \log_2 \frac{4}{21} - \frac{3}{21} \log_2 \frac{3}{21} = 1.25779047619$$

$$H_{(x_2,F)} = -\frac{4}{21} \log_2 \frac{4}{21} - 2(\frac{1}{21} \log_2 \frac{1}{21}) - \frac{5}{21} \log_2 \frac{5}{21} = 1.36691428571$$

Given these individual entropies, we can calculate the weighted average entropies:

$$E_{x_1} = \sum_{i=1}^{K} \frac{|C_i|}{N} H(P(v_1, 1), \ldots, P(v_i, K))$$
$$E_{x_1} = (\frac{8}{21} H_{(x_1,T)}) + (\frac{13}{21} H_{(x_1,F)})$$
$$E_{x_1} = 1.37103718821$$
$$E_{x_2} = \sum_{i=1}^{K} \frac{|C_i|}{N} H(P(v_1, 1), \ldots, P(v_i, K))$$
$$E_{x_2} = (\frac{10}{21} H_{(x_2,T)}) + (\frac{11}{21} H_{(x_2,F)})$$
$$E_{x_2} = 1.31495056689$$

We have $E_{x_1} = 1.37103718821$ and $E_{x_2} = 1.31495056689$

2. a) The class priors are $P(A = Yes) = 3/5$ and $P(A = No) = 2/5$. They are the probabilities that a sample's class is an A or not without using any of the features.

b) After we standardize our data given, we will result in the following feature values:

$$\begin{pmatrix} 0.76024952 & -0.69351937 \\ -0.25584038 & -0.69974033 \\ 1.35469667 & -0.7168134 \\ -0.31804996 & -0.71093805 \\ 1.9837047 & -0.70374939 \end{pmatrix}$$

I used my standardized function used in my naive bayes and logistic regression algorithms to standardize this data. However, I made a modification to standardize the whole data and not each feature. Here is a sample of this code:

```
def standardize(data):
    return (data - np.mean(data)) / np.std(data, ddof=1)
```

After, I standardized my data, I was able to extract the means and variance's of each feature within the set of classes. For the class where $A = No$, we had the means of $\{1.66920068, -0.7102814\}$ and variances of $\{0.197825551, 8.53342168 * 10^{-5}\}$. For the class where $A = Yes$, we had the means of $\{0.06211973, -0.70139925\}$ and variances of $\{0.366506412, 7.79166545 * 10^{-5}\}$.

c) Given the calculated data above, we can predict the class given the features we are given. In this case we are given a feature 1 of '242 characters' and a feature 2 of '4.56 average word length'. First we will standardize this using the mean and standard deviation from our training data. From this we result in the following x values: $\{0.94758484, -0.74271607\}$. Using this, we can calculate our class probabilities as such:

$$P(y = No|x) = P(y = No)P(x = x_1|y = No)P(x = x_2|y = No)$$
$$P(y = Yes|x) = P(y = Yes)P(x = x_2|y = Yes)P(x = x_2|y = No)$$

Where we generate our Gaussian probability as such: $p(x) = \frac{e^{-\frac{(x-\mu)^2}{2*\sigma^2}}}{\sigma * \sqrt{2 * \pi}}$ where σ and μ are corresponding to the class we are calculating our probability for. Using all of this, we can calculate our posterior for the data given to be:

$$P(Y = No) = 0.4 * 2.60188172 * 10^{-3} * 1.19209290 * 10^{-7} = 4.205631935332507 * 10^{-9}$$
$$P(Y = Yes) = 0.6 * 5.87989962 * 10^{-2} * 1.19209290 * 10^{-7} = 4.205631935332507 * 10^{-9}$$

Since the posterior for Y=Yes is greater, our predicted class label is Yes.

2. Naive Bayes

These are the classifications statistics on the testing data results created from my implementation of a Naive Bayes algorithm:

$$Precision : 8.0$$
$$Recall : 303.0$$
$$f - measure : 15.588424437299036$$
$$Accuracy : 0.7984344422700587$$

3. Logistic Regression

These are the classifications statistics on the testing data results created from my implementation of a Logistic Regression algorithm:

$$Precision : 66.0$$
$$Recall : 52.0$$
$$f - measure : 58.16949152542373$$
$$Accuracy : 0.9243313763861709$$