Ömer Tuğrul
22002723

CS464: Introduction to Machine Learning - HW1

Question 1: Probability Review

1.1)

I can select one of the boxes with 0.5 probability. Then I made the calculations according to that:

$$\frac{1}{2}\left[\frac{2}{3}\cdot\frac{1}{2}\cdot\frac{1}{2}+\frac{1}{3}\cdot\frac{1}{4}\cdot\frac{1}{4}\right]+\frac{1}{2}\left[\frac{1}{2}\cdot\frac{1}{2}\cdot\frac{1}{2}+\frac{1}{2}\cdot\frac{1}{10}\cdot\frac{1}{10}\right]$$

$$=\frac{1}{2}\left(\frac{9}{48}+\frac{13}{100}\right)=\frac{27}{800}=0.15875$$

1.2)

Now we are given that we tossed coin two times, and we get two heads in a row.

Let say A: Two heads in a row, B: selected coin was fair, C: Box1 selected, D: Box2 selected.

$$P(B|A) = P(B,C|A) + P(B,D|A) = \frac{\frac{1}{12}}{\frac{127}{800}}+\frac{\frac{1}{16}}{\frac{127}{800}}=\frac{350}{381}=0.91864$$

1.3) E: The coin is red

$$P(E|A) = \frac{\frac{1}{400}}{\frac{127}{800}}=\frac{2}{127}=0.01575$$

Question 2: MLE and MAP

$$f(x;\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

2.1) We have n i.i.d. data points so likelihood function will be:

$$f(x_1,..,x_n;\mu,\sigma^2) = \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} = (\frac{1}{\sigma\sqrt{2\pi}})^n \cdot e^{\frac{-\sum_{i=1}^{n}(x_i-\mu)^2}{2\sigma^2}}$$

$$Loglikelihood = \log\big(f(x_1,..,x_n;\mu,\sigma^2)\big) = nlog\left(\frac{1}{\sigma\sqrt{2\pi}}\right) - \frac{\sum_{i=1}^{n}(x_i-\mu)^2}{2\sigma^2}$$

To find the maximum point w.r.t. μ take derivative of the loglikelihood and equate it to zero.

To find the maximum of the likelihood function, $\frac{\sum_{i=1}^{n}(x_i-\mu)}{\sigma^2} = 0$ so $\mu_{MLE} = \frac{\sum_{i=1}^{n} x_i}{n}$

2.2) Now, we know that the prior distribution of μ. Hence, we can find the MAP estimate of μ from posterior distribution.

From Bayes rule $f(\mu|x) = \frac{f(x|\mu)\cdot f(\mu)}{f(x)}$ denominator is not dependent on μ so only consider numerator.

Posterior $f(\mu|x) = \lambda(\frac{1}{\sigma\sqrt{2\pi}})^n \cdot e^{-\lambda\mu} \cdot e^{\frac{-\sum_{i=1}^{n}(x_i-\mu)^2}{2\sigma^2}}$ to find the maximum of the posterior distribution take logarithm and then derivative, equal it to zero.

$$-\lambda\mu - \frac{\sum_{i=1}^{n}(x_i-\mu)^2}{2\sigma^2} \; take\ derivative\ and\ equal\ it\ to\ zero \Rightarrow -\lambda + \frac{\sum_{i=1}^{n}(x_i-\mu)}{\sigma^2} = 0$$

Hence, $\mu_{MAP} = \frac{\sum_{i=1}^{n} x_{i-\lambda\sigma^2}}{n}$

2.3)

In continuous random variables we cannot find the probability of a single point, but we can only find interval of the probability. In this question it asks for a single point. For this reason, answer of $P(X_{n+1} = 1) = 0$

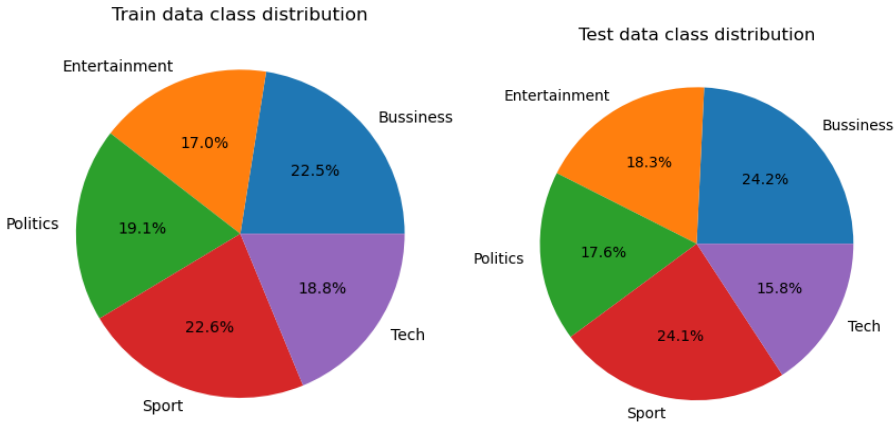
But we can find the likelihood of function given observations. It is stated that the distribution represents normal distribution with $\mu = 1\ and\ \sigma = 1$

Hence, $f(X_{n+1} = 1; \mu = 1, \sigma = 1) = \frac{1}{\sqrt{2\pi}} e^{\frac{-(x-1)^2}{2}} = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}}$

Question 3: BBC News Classification

3.1)

3.1.1) The distribution of each category is analyzed given in the pie charts below:



3.1.2) The prior distribution of each category is given below:

```
In [25]: prior_class_probs

Out[25]: {0: 0.22482014388489208,
          1: 0.17026378896882494,
          2: 0.19124700239808154,
          3: 0.2260191846522782,
          4: 0.18764988009592326}
```

3.1.3) As we see in the training data class distribution pie plot, the dataset is balanced between the classes. Since the maximum percentage is 22.6 and minimum percentage is 17, 5% difference can be considered as balanced dataset. But if we have an imbalanced dataset, it could affect the model performance. Since the prior of one class will be dominant the model will have more bias to that class.

3.1.4)

First, I found how many times the words “alien” and “thunder” occur in tech documents.

```
In [62]: X_train[y_train['class'] == 4]['alien'].sum()

Out[62]: 3

In [63]: X_train[y_train['class'] == 4]['thunder'].sum()

Out[63]: 0
```

The log ratio of them is found by the code below(result in the last line):

```
In [18]: alien_freq = 0
         for i in range(len(X_train[y_train['class'] == 4]['alien'].value_counts())):
             alien_freq += i*X_train[y_train['class'] == 4]['alien'].value_counts()[i]
         total_tech = X_train[y_train['class'] == 4].sum().sum()
         alien_given4 = alien_freq/total_tech
         alien_given4

Out[18]: 3.824920632896867e-05
```

```
In [20]: thunder_freq = 0
         total = 0
         for i in range(len(X_train[y_train['class'] == 4]['thunder'].value_counts())):
             alien_freq += i*X_train[y_train['class'] == 4]['thunder'].value_counts()[i]
             total += X_train[y_train['class'] == 4]['thunder'].value_counts()[i]
         thunder_given4 = thunder_freq/total_tech
         thunder_given4

Out[20]: 0.0
```

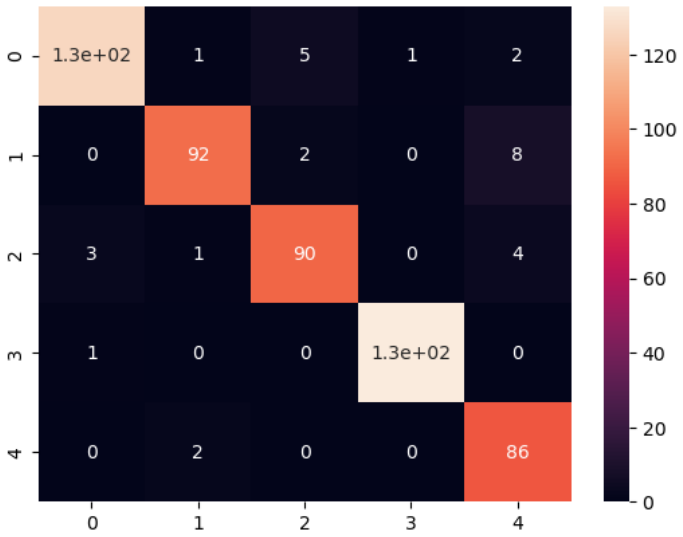
```
In [21]: #LOG ratio
         log_alien_tech = np.log(alien_given4)
         log_thunder_tech = np.log(thunder_given4)
         log_alien_tech,log_thunder_tech

         C:\Users\user\AppData\Local\Temp\ipykernel_51852\850315015.py:3: RuntimeWarning: divide by zero encountered in log
           log_thunder_tech = np.log(thunder_given4)

Out[21]: (-10.171387747476452, -inf)
```

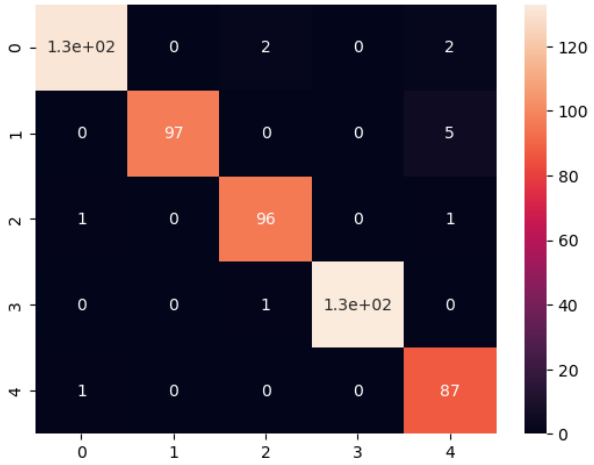
3.2) Multinomial Naïve Bayes Classification

I train the multinomial naïve bayes classifier on test set and the accuracy result found as 0.946. To simulate the behavior of the -inf, I used -10^12.



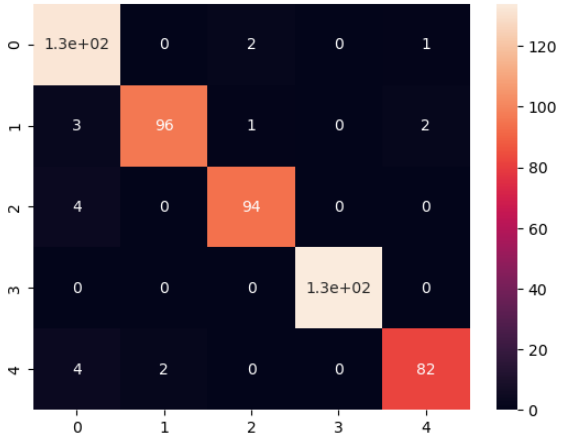
3.3) Multinomial Naïve Bayes classification with Laplace smoothing

In this part I again used my code from previous part, but I add dirichlet prior $\alpha = 1$. Dirichlet prior (Laplace smoothing $\alpha = 1$) is important to handle the zero counts in the data. By ensuring, model is not assigning a zero probability to a category only because it is not appeared in training set. This prevents overfitting to the training data. The model's performance is improved in the test set as expected. The accuracy result was found as 0.977.



3.4) Bernoulli Naïve Bayes Classification

In this part, I trained Bernoulli naïve bayes classifier and accuracy found as 0.966. I used -10^12 instead of -inf. Again, I used Laplace smoothing.



This result is better than multinomial naïve bayes classifier without Dirichlet prior. But it is worse than MNB classifier with Laplace smoothing.

Both Bernoulli and Multinomial naïve bayes models are used for text classification. The Bernoulli model looks for features absence and multinomial model looks for features frequency. So, if in one dataset absence of feature is meaningful, then the frequency of feature than Bernoulli model works better. In the opposite situation, multinomial model works better. Hence, our choice will depend on the nature of the dataset. In our dataset, multinomial model worked better since frequencies are involved.