CS464
Introduction to Machine Learning

Homework 1

Mustafa Mert Türkmen

21902576 – IE

Section 2

Question 1: Probability Review

Question 1.1

We need to calculate the probability that the select a coin from one of the boxes and toss it two times and gets head for two times. P(Head,Head) should be found.

So, calculation can be done like this;

$$\frac{1}{2} \times \left(\frac{1}{3} * \frac{1}{4} * \frac{1}{4} + \frac{2}{3} * \frac{1}{2} * \frac{1}{2}\right) + \frac{1}{2} \times \left(\frac{1}{2} * \frac{1}{2} * \frac{1}{2} + \frac{1}{2} * \frac{1}{10} * \frac{1}{10}\right) = 0.15875$$

Question 1.2

We need to find the $P(Blue\ Coin|Head, Head)$

Using Bayes Theorem, we can say:

$$P(Blue\ Coin|Head, Head) = \frac{P(Head, Head|Blue\ Coin) \times P(Blue\ Coin)}{P(Head, Head)}$$

$$P(Head, Head|Blue\ Coin) = \frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$$

$$P(Blue\ Coin) = \frac{1}{2} \times \frac{1}{2} + \frac{1}{2} \times \frac{2}{3} = 0.58333$$

$$P(Head, Head) = 0.15875$$

So, it is equal to

$$P(Blue\ Coin|Head, Head) = \frac{0.25 \times 0.58333}{0.15875} = 0.91863$$

Question 1.3

$$P(Red\ Coin|Head, Head) = \frac{P(Head, Head|Red\ Coin) \times P(Red\ Coin)}{P(Head, Head)}$$

$$P(Head, Head|Red\ Coin) = \frac{1}{10} \times \frac{1}{10} = \frac{1}{100}$$

$$P(Blue\ Coin) = \frac{1}{2} \times \frac{1}{2} = 0.25$$

$$P(Head, Head) = 0.15875$$

$$P(Blue\ Coin|Head, Head) = \frac{0.01 \times 0.25}{0.15875} = 0.01575$$

Question 2: MLE and MAP

Question 2.1

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

Probability of samples D={x_1,….,x_n}

$$\mathrm{P}(\mathrm{D}|\mu,\sigma^2) = \prod_{i=1}^{n} f(x_i;\mu,\sigma^2)$$

$$\mathrm{P}(\mathrm{D}|\mu,\sigma^2) = \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^{\mathrm{n}} \prod_{i=1}^{n} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Taking the logarithm of the likelihood function (log-likelihood):

$$\ln \mathrm{P}(\mathrm{D}|\mu,\sigma^2) = \ln\left[\left(\frac{1}{\sigma\sqrt{2\pi}}\right)^{\mathrm{n}} \prod_{i=1}^{n} e^{-\frac{(x-\mu)^2}{2\sigma^2}}\right]$$

$$= -N\ln\sigma\sqrt{2\pi} - \sum_{i=1}^{n}\left[\frac{(x_i-\mu)^2}{2\sigma^2}\right]$$

The derivative of the log-likelihood with respect to (μ) is:

$$\frac{d}{d\mu}\ln \mathrm{P}(\mathrm{D}|\mu,\sigma^2) = \frac{d}{d\mu}(-N\ln\sigma\sqrt{2\pi}) - \frac{d}{d\mu}\left(\sum_{i=1}^{n}\left[\frac{(x_i-\mu)^2}{2\sigma^2}\right]\right)$$

Setting the derivative equal to zero:

$$= \sum_{i=1}^{n}\frac{(x_i-\mu)}{\sigma^2} = 0$$

$$= \sum_{i=1}^{n}(x_i) \text{ -n}\mu = 0$$

$$\mu_{\mathrm{MLE}} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

So, the Maximum Likelihood Estimation for (μ) is the sample mean:

$$\hat{\mu}_{\mathrm{MLE}} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Question 2.2

$$P(\mu|D) \propto P(D|\mu)P(\mu)$$

We know the $P(D|\mu)$ value from first question, to find $P(\mu)$ we can put μ to our function which is equal to:

$$\mathrm{P}(\mu) = \lambda e^{-\lambda\mu}$$

So;

$$P(\mu|D) \propto \frac{1}{n}\sum_{i=1}^{n} x_i \times \lambda e^{-\lambda\mu}$$

Question 2.3

To find the probability that a new data point which is equal to 1 in a normal distribution with mean 1 and standard deviation 1, we can use the probability density function of the normal distribution:

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

Substitute $\mu = 1, \sigma = 1$ and $\mathrm{x}_{n+1} = 1$ into the PDF:

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

So, the probability that the new data point,

$$= \frac{1}{\sqrt{2\pi}}$$

If we measured the new data point and obtained 2, the likelihood of this observation according to the PDF of the normal distribution is:

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

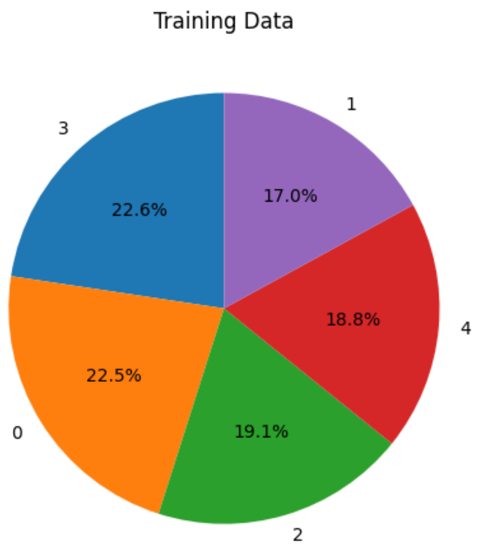
This gives the likelihood of the data point according to the normal distribution.

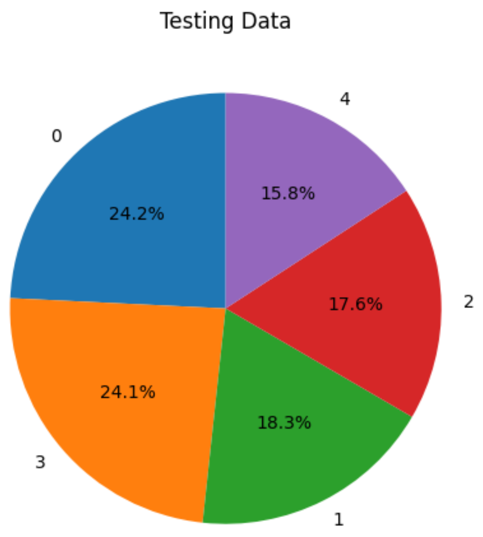
Question 3: BBC News Classification

Question 3.1

Question 3.1.1

I drew two pie charts of the distribution of classifications in y_train and y_test separately. We can see that out of 1668 news in the training data labeled 22.5% as Business, 17% as Entertainment, 19.1% as Politics, 22.6% as Sport and 18.8% as Tech. When we looked at the test data the %24.2 as Business, 18.3% as Entertainment, 17.5% as Politics, 24.1% as Sport and 15.8% as Tech.





Question 3.1.2

I write a function to find prior probabilities for each class. The prior probabilities can be sorted like this.

Class Business: Prior Probability = 0.225

Class Entertainment: Prior Probability = 0.170

Class Entertainment: Prior Probability = 0.191

Class Sport: Prior Probability = 0.226

Class Tech: Prior Probability = 0.188

Question 3.1.3

Our training data is not uniform, some classes have a higher percentage data than others. Business and Sport news have higher percentages (22.5% and 22.6%, respectively), while Entertainment has a lower percentage (17%). This imbalance may affect my model training, potentially leading to biased predictions and reduced accuracy in minority classes. Because their Prior Probabilities will be used in our model and some of the topic of news has high probability that may cause wrong label amount increasing. My model is trained to favor the business and sport news classes and may not have completely learnt some patterns that are specific to entertainment news.

Question 3.1.4

I define a function which is *calculate_likelihood* for calculating the log ratio of the word's occurrences in these news, it says include the multiple occurrences so I count the total number of words in this specific class of news. I find:

ln(P(alien | Y = Tech)) = -4.647590901872044

ln(P(thunder | Y = Tech)) = -1000000000000.0

Thunder word has no occurrence in the tech news so its total number of word amount equals zero which will result in ln(0). I assume that simulate the behavior of the number '-inf', I assign an arbitrarily small number to this value as -10^{12} to handle overflow this issue.

Question 3.2: Multinomial Naive Bayes

My *calculate_accuracy* and *confusion_matrix* functions gives the following results:

- Accuracy ~ 0.946
- Confusion Matrix:

		Actual Values				
Predicted Values		Business	Entertainment	Politics	Sport	Tech
	Business	126	0	3	1	0
	Entertainment	1	92	1	0	2
	Politics	5	2	90	0	0
	Sport	1	0	0	133	0
	Tech	2	8	4	0	86

Question 3.3: Multinomial Naive Bayes with Smoothing

My *calculate_accuracy* and *confusion_matrix* functions gives the following results:

- Accuracy ~ 0.977
- Confusion Matrix:

		Actual Values				
Predicted Values		Business	Entertainment	Politics	Sport	Tech
	Business	131	0	1	0	1
	Entertainment	0	97	0	0	0
	Politics	2	0	96	1	0
	Sport	0	0	0	133	0
	Tech	2	5	1	0	87

We can see from the above result, using smoothing slightly increased our accuracy. The prediction rate is increased for the majority classes which can count as Business and Sport news. We achieve this success by decreasing the effects of words that appear rarely by adding a prior value. This is an important metric, as it is the main purpose of this model, identifying news topics. Therefore, using smoothing was a great choice for words because lots of words do not appear on classes so they took negative infinity likelihood.

Question 3.4: Bernoulli Naive Bayes Model

My *calculate_accuracy* and *confusion_matrix* functions gives the following results:

- Accuracy ~ 0.966
- Confusion Matrix:

		Actual Values				
		Business	Entertainment	Politics	Sport	Tech
Predicted Values	Business	132	3	4	0	4
	Entertainment	0	96	0	0	2
	Politics	2	1	94	0	0
	Sport	0	0	0	134	0
	Tech	1	2	0	0	82

The initial observation may be that Multinomial models with smoothing may be preferable over the Bernoulli model, given the comparatively lower accuracy. The Multinomial model without smoothing has poorer performance, emphasized by its higher false classified values across multiple classes, indicating its limitations in correctly identifying instances from each category. In contrast, the normal Multinomial and Smoothed Multinomial models exhibit competitive accuracy, but the Smoothed Multinomial model performs well. However, the Smoothed Multinomial model proves more potential in a real world context due to its lower false classified values across various classes, signifying its enhanced ability to identify instances from each category correctly.

The results also indicate that the Bernoulli Naive Bayes model performs better than its Multinomial Naive Bayes model in our classification task. Many words in the news have no occurrence, so adding their likelihood to a prior value will increase our model's accuracy.

While accuracy remains a key metric, considering other metrics becomes even more crucial in a multi-class scenario. Metrics such as precision, recall, and F1-score for each class can provide a more comprehensive understanding of the models' performance across different categories. This detailed analysis would help make a more informed decision about the best-performing model for each class in this multi-class classification task.

So overall, Multinomial Naive Bayes with Smoothing and Bernoulli Naive Bayes models are preferable to developing classification models with words.

Use of external libraries:

I primarily utilized the pandas library for reading CSV files and creating a structured representation through a pandas DataFrame. This choice proved advantageous as pandas' DataFrame offers convenient functions for tasks like adding columns and copying data frames, which were instrumental in my implementation. However, my primary reliance for efficient numerical computations lay with the NumPy library. I frequently converted the DataFrame to a NumPy array to leverage NumPy's array operations, which significantly improved computational efficiency compared to nested for loops. Key NumPy functions such as np.where(), np.count_nonzero(), np.log10() and np.sum() played a pivotal role in streamlining my calculations. I also use the Math library to find logarithmic value of prior probabilities. Additionally, I incorporated the matplotlib library to generate plots for visualization purposes, enhancing the quality of the graphical representations included in my report. I also use -10^{12} for simulation purposes of -inf.