HW1 CS464-1
Görkem Kadir Solun 22003214

1 Probability Review

Q1.1

$P(B)$ =The event that a blue coin is chosen
$P(R)$ = The event that the red coin is chosen
$P(Y)$ = The event that the yellow coin is chosen
$P(BOX1)$ =The probability that box 1 is chosen
$P(BOX2)$ = Event that box 2 is chosen

$$P(B) = P(B|BOX1) * P(BOX1) + P(B|BOX2) * P(BOX2) = 2/3 * 1/2 + 1/3 * 1/4 = 7/12$$

$$P(Y) = P(Y|BOX1) * P(BOX1) = 1/3 * 1/2 = 1/6$$

$$P(R) = P(R|BOX2) * P(BOX2) = 1/2 * 1/2 = 1/4$$

What we want
$P(A)$ = Event that tail comes twice

$P(T|B) = 1/2$ The event that blue coin landed tail.

$P(T|Y) = 3/4$ The event that yellow coin landed tail.

$P(T|R) = 9/10$ The event that red coin landed tail.

$$\begin{aligned} P(A) &= P(A|B) * P(B) + P(A|Y) * P(Y) + P(A|R) * P(R) \\ &= [P(T|B)]\text{^}2 * P(B) + [P(T|Y)]\text{^}2 * P(Y) + [P(T|R)]\text{^}2 * P(R) \\ &= 1/4 * 7/12 + 9/16 * 1/6 + 81/100 * 1/4 = 7/48 + 9/96 + 81/400 \\ &= 0.44208 \end{aligned}$$

Q1.2

Blue coins are fair, so let's only check blue coins.

$$P(B|A) = (P(A|B) * P(B))/P(A) = (1/2 * 1/2 * 7/12)/(7/48 + 9/96 + 81/400) = 0.32988$$

Q1.3

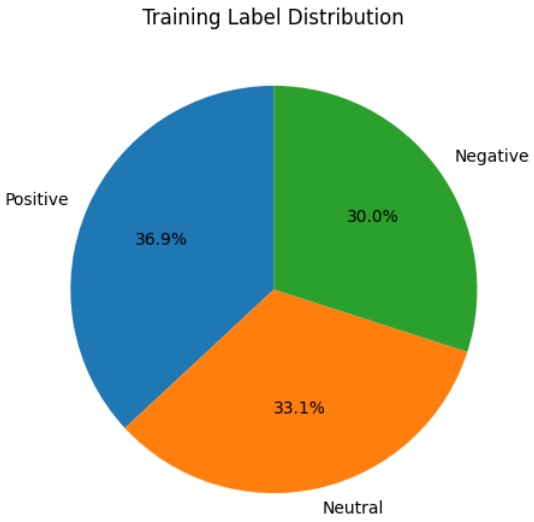
We check red this time.

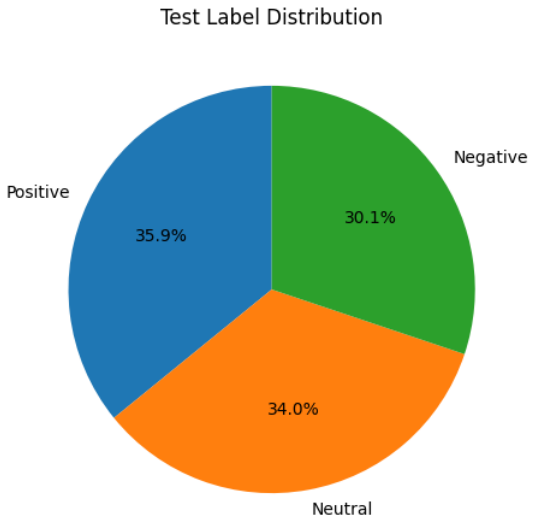
$$P(R|A) = (P(A|R) * P(R))/P(A) = (81/100 * 1/4)/(7/48 + 9/96 + 81/400) = 0.45806$$

2 Amazon Reviews Classification

Question 3.1

1. What are the percentages of each category in the y_train.csv y_test.csv? Draw a pie chart showing percentages.





Y can take 0,1,2, which are negative, neutral, and positive, respectively.

	Training (out of 100)	Test (out of 100)
0	689/2300 = 29.95652	211/700 = 30.14286
1	762/2300 = 33.12043	238/700 = 34.00000
2	849/2300 = 36.91304	251/700 = 35.85714

2. What is the prior probability of each class? Write your answer to the report.

	Training (out of 100)
0	689/2300 = 29.95652
1	762/2300 = 33.12043
2	849/2300 = 36.91304

3. Is the training set balanced or skewed towards one of the classes? Do you think having an imbalanced training set affects your model? If yes, please explain briefly how it can affect the model.

The overall dataset, comprising both training and validation data, shows a class distribution that closely aligns with that of the training set. This similarity indicates that no class is notably underrepresented as each comprises approximately one-third of the total dataset. However, a class imbalance can still lead the model to favor the majority class in predictions, potentially reducing accuracy for minority classes, as it has more examples to learn. Maintaining a balanced dataset is essential to minimize skew and enhance model performance. As discussed

in class, if the model is not trained sufficiently, it may be too simplistic or underfitting to capture the underlying feature patterns, resulting in lower prediction accuracy.

4. How many times do the words ”good” and ”bad” appear in the training documents with the label ”positive”, including multiple occurrences, and what are the log ratio of their occurrences within those documents, i.e, ln(P(good|Y = positive)) and ln(P(bad|Y = positive))?

The count of "good" occurrences when the output label is 2 (indicating a positive review) is 207. In contrast, the count of "bad" occurrences for the same output label is 12.

$P\ (good \mid Y\ =\ positive)\ =\ 207/849$ and $P\ (bad \mid Y\ =\ positive)\ =\ 12/849$

$$ln(207/849)/ln(12/849) = 0.33137$$

Question 3.2 Confusion Matrix for Multinomial Naïve Bayes (α = 0)

	Predicted			
Actual	Accuracy: 0.301	Negative	Neutral	Positive
	Negative	211	0	0
	Neutral	238	0	0
	Positive	251	0	0

Table 1

I did not set log to 1e-12; there is a commented-out block of code to set in the q2main.py file.

Question 3.3 Confusion Matrix for Multinomial Naïve Bayes (α=1)

	Predicted			
Actual	Accuracy: 0.649	Negative	Neutral	Positive
	Negative	151	45	15
	Neutral	73	86	79
	Positive	13	21	217

Table 2

Using additive smoothing with a Dirichlet prior (α=1) prevents zero probabilities for words that are absent in the texts. This approach enhances test set accuracy and generalization, particularly when training data is limited. My findings support this improvement, as evidenced by the higher accuracy in Table 2 compared to Table 1.

Question 3.4 Confusion Matrix for Bernoulli Naïve Bayes

	Predicted			
Actual	Accuracy: 0.641	Negative	Neutral	Positive
	Negative	113	90	8
	Neutral	29	180	29
	Positive	19	76	156

Table 3

The accuracy of both models is similar, with MNB being slightly more accurate. Representing information based on the presence or absence of words is like relying on word frequencies. MNB performs better in the negative class, with more correct predictions and fewer

misclassifications, especially in the positive category. BNB handles the neutral class significantly better, showing a higher correct classification rate and lower misclassification. MNB is considerably more effective in correctly predicting the positive class than BNB, with fewer samples misclassified. In summary, MNB is more suitable when the frequency of words contributes to the sentiment distinction, which explains its higher performance on negative and positive classes. BNB, however, captures neutral instances better, possibly due to its binary treatment of features. For tasks involving sentiment analysis where frequency matters, MNB might offer a slight advantage, while BNB might be beneficial in applications where binary presence/absence is critical. However, they are very similar and may change from task to task.