CS-464 HW1

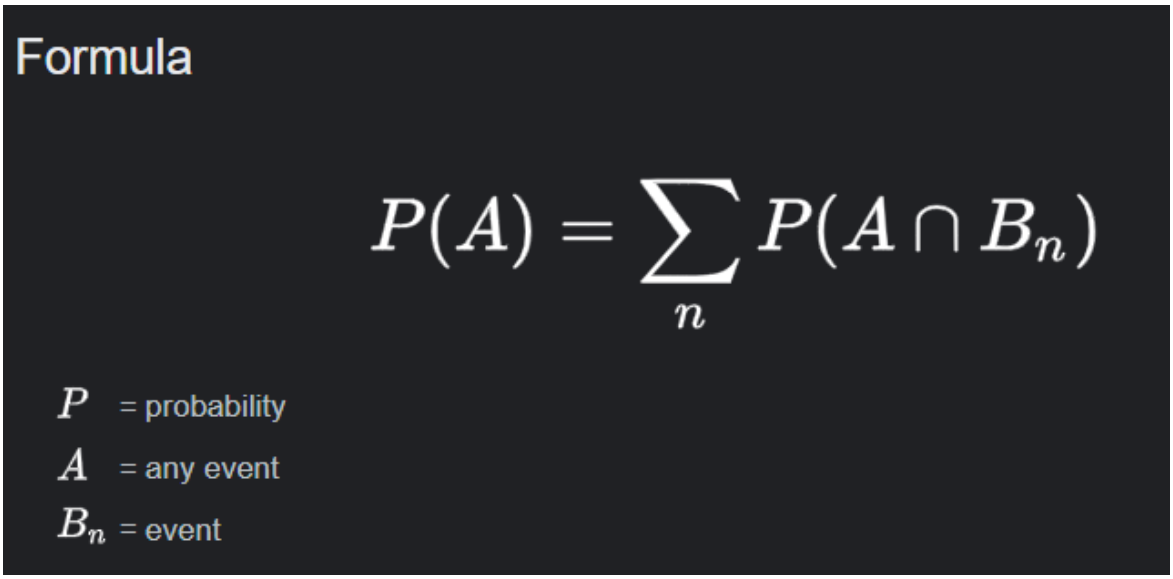
Muhammet Musa Gezer

21703973

EEE

1.1)

Using the law of total probability,

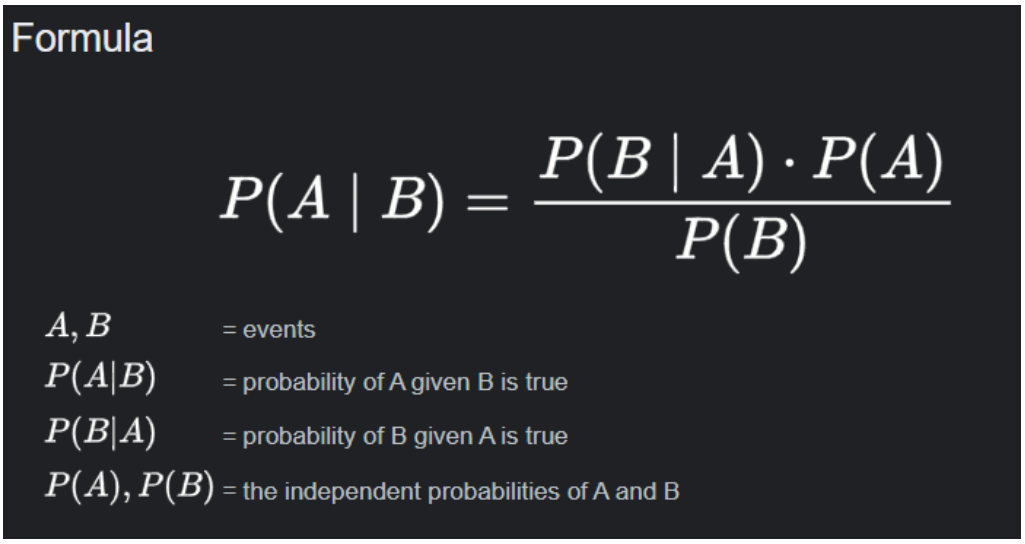


$$P(pos) = P(pop) * P(pos|pop) + P(mod) * P(pos|mod) + P(unp) * P(pos|unp)$$

$$P(pos) = 0.45 \times 0.95 + 0.30 \times 0.60 + 0.25 \times 0.1 = 0.6325$$

1.2)

Using the Bayes Rule,



$$P(pop|pos) = \frac{\left(P(pos|pop) \times P(pop)\right)}{P(pos)}$$

$$P(pop|pos) = \frac{0.95 \times 0.45}{0.6325} = 0.6759$$

1.3)

Using the Bayes Rule,

$$P(pop|notpos) = \frac{\left(P(notpos|pop) \times P(pop)\right)}{P(notpos)}$$

$$P(pop|notpos) = \frac{0.05 \times 0.45}{0.3675} = 0.0612$$

2.1.1)

The percentage of spam emails in the y_train.csv is 28.6,

```
number_of_spams = sum(y_train_dataframe["Prediction"])
total_number = len(y_train_dataframe["Prediction"])
ratio_of_spams = number_of_spams / total_number
print(f"The percentage of spam emails : {ratio_of_spams*100}%.")
```

```
The percentage of spam emails : 28.59560067681895%.
```

2.1.2

Yes, the training set is imbalanced, with 71% of the emails being ham and only 29% being spam. This imbalance can affect the model's performance in predicting spam emails since the model may be biased towards predicting ham emails. Because there are more ham mails than spam emails, the prior for ham mails is higher. If the test data or the distribution of emails in real world is different, this could lead to unwanted bias towards ham emails.

2.1.3

The percentage of spam emails in the test set is 30 percent, which is highly close to the ratio in the train set. The reported accuracy may not be as affected by class imbalance as it would be in a more severely imbalanced scenario. However, even in this case, the reported accuracy could be misleading as the model is biased towards the majority class. The model may have high accuracy but low performance on spam emails. Even if the model classifies all emails as spam, it can reach accuracy value of 70 percent which is misleading. Therefore, it is important to evaluate the model's performance using alternative metrics.

2.2)

The accuracy of the multinomial naïve bayes model is 94.40 %.

The confusion matrix :

```
Confusion Matrix:
-----------------
   |  0  |  1  |
---|-----|-----|
 0 | 705 | 13  |
 1 | 45  | 272 |
---|-----|-----|
array([[705.,  13.],
       [ 45., 272.]])
```

The 1s that are estimated as 0s are more than zeros that are estimated 1, this could be because of the fact that the dataset is skewed towards 0s and equality situation considered as 0s.

58 of the predictions were false as it could be seen in the confusion matrix.

2.3)

The accuracy using fair Dirichlet prior is 94.78 %.

The confusion matrix:

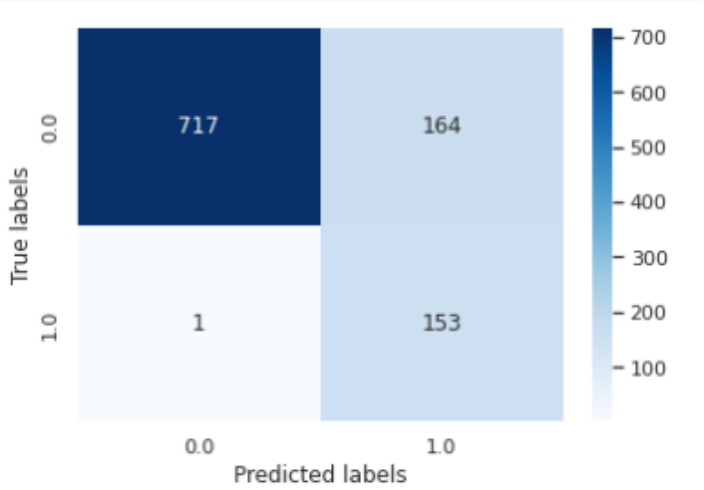
```
Confusion Matrix:
-----------------
   |  0  |  1  |
---|-----|-----|
 0 | 681 | 37  |
 1 | 17  | 300 |
---|-----|-----|
The accuracy is 94.78%.
```

Wrongly estimated 1s decreased and 0s increased. In total, there are 54 wrongly predicted values. This could be because of the fact that if there is -inf in both sides, it was evaluated as 0 before. Now, we get rid of -infinities with the Dirichlet method which provide us with a healthier solution. The number of wrong predictions is now more proportional to the amount of data in each class. By incorporating the Dirichlet prior, we are essentially smoothing the probability estimates by adding a small amount of probability mass to each possible value, which can help to reduce the impact of sparsity and noise in the data.

2.4)

The accuracy of the Bernoulli naïve bayes algorithm is 84.05%.

The confusion matrix:



Number of zeros predicted as 1s are higher in this case. Number of wrongly predicted values is 165. The accuracy of the model is smaller. As it is seen in this example, in cases where the features are counts, Multinomial Naive Bayes is usually the better choice. The accuracy is smaller because these models do not use the frequency of the words in the model.

2.5)

In conclusion, based on the results obtained from the three models - Multinomial Naïve Bayes, Multinomial Naïve Bayes with fair Dirichlet prior, and Bernoulli Naïve Bayes - we can draw the following comparisons:

Accuracy: The Multinomial Naïve Bayes with fair Dirichlet prior had the highest accuracy (94.78%), followed by the Multinomial Naïve Bayes (94.40%) and the Bernoulli Naïve Bayes (84.05%). This indicates that incorporating the Dirichlet prior provided some improvement in classification performance.

Confusion Matrix: The confusion matrices for each model revealed different patterns of misclassification. The Multinomial Naïve Bayes model had a higher number of false negatives (spam emails classified as ham), likely due to the class imbalance. The fair Dirichlet prior helped balance the misclassifications, resulting in a more proportional distribution of errors. In the Bernoulli Naïve Bayes model, there were more false positives (ham emails classified as spam), indicating a different bias in classification.

Applicability: For datasets with count-based features, such as the frequency of words in emails, the Multinomial Naïve Bayes models (with or without Dirichlet prior) generally outperformed the Bernoulli Naïve Bayes model. The Bernoulli model, which only considers the presence or absence of features, might not capture the nuances of the data as effectively as the Multinomial models.

By incorporating the Dirichlet prior, the model managed to better address the class imbalance issue and provide a more reliable classification performance. The addition of the fair Dirichlet prior allowed for a smoother probability estimation and mitigated the impact of sparsity and noise in the data, leading to an improved overall accuracy and a more balanced distribution of errors across both classes.

In summary, when working with imbalanced datasets or count-based features, it is essential to consider different modeling approaches and performance metrics to ensure a reliable and accurate representation of the model's performance. In this case, the Multinomial Naïve Bayes model with a fair Dirichlet prior proved to be the most effective approach among the three models tested, demonstrating the importance of addressing class imbalance and utilizing appropriate priors in classification tasks.