Gülce Erdoğan
21802781
CS464 – 001

CS464 Homework 1 Report

Question 1 - The CS 464 Case

Question 1.1: What is the probability that a student is motivated [$P(S_M)$]?

$$P(S_M) = P(H)P(S_M|H) + P(L)P(S_M|L) + P(F)P(S_M|F)$$

$$P(S_M) = \frac{64}{100}\frac{87}{100} + \frac{24}{100}\frac{21}{100} + \frac{12}{100}\frac{4}{100}$$

$$P(S_M) = \frac{5568}{10000} + \frac{504}{10000} + \frac{48}{10000} = \frac{6120}{10000} = 0.6120$$

Question 1.2: If a student is motivated, what is the probability that he/she will get a high grade [$P(H|S_M)$]?

$$P(H|S_M) = \frac{P(H)P(S_M|H)}{P(S_M)}$$

$$P(H|S_M) = \frac{\frac{64}{100}\frac{87}{100}}{\frac{6120}{10000}} = \frac{\frac{5568}{10000}}{\frac{6120}{10000}} = \frac{5568}{6120} = 0.9098$$

Question 1.3: If a student is unmotivated, what is the probability that he/she will get a high grade [$P(H|S_U)$]?

$$P(H|S_U) = \frac{P(H)P(S_U|H)}{P(S_U)} = \frac{P(H)P(S_U|H)}{1 - P(S_H)}$$

$$P(H|S_U) = \frac{\frac{64}{100}\frac{13}{100}}{1 - \frac{6120}{10000}} = \frac{\frac{832}{10000}}{\frac{3880}{10000}} = \frac{832}{3880} = 0.2144$$

Question 2 - Sports News Classification

Question 2.1: Class Imbalance Problem

1. The class distributions in the training set are not balanced since the ratio of the classes is not close to each other. When the training data is observed, it is seen that the number of instances for each class is as follows:
    - Athletics (0): 77
    - Cricket (1): 86
    - Football (2): 198
    - Rugby (3): 114
    - Tennis (4): 77

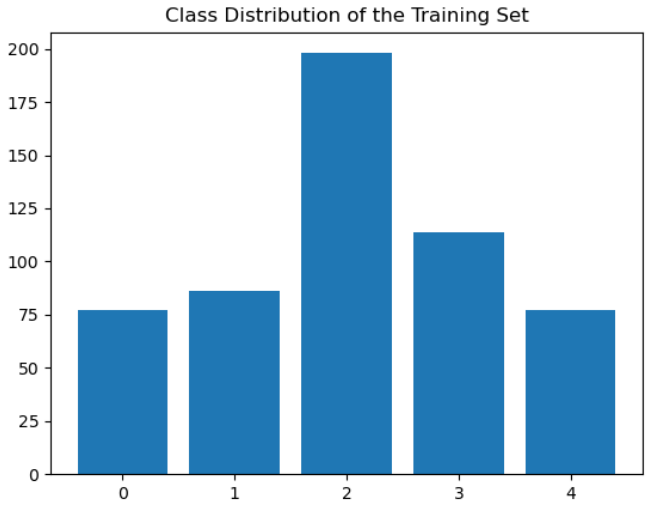

Figure 1: Class Distribution of the Training Set

2. As can be seen in Figure 1, the training set is skewed towards class 2 (football); therefore, the training dataset is not balanced. Having such an imbalanced training set affects the model being Multinomial Naive Bayes. The general formula for Multinomial Naive Bayes can be observed below:

$$\hat{y}_i = \arg\max_{y_k} P(Y = y_k|D_i) = \arg\max_{y_k} P(Y = y_k) \prod_{j=1}^{V} P\left(X_j|Y = y_k\right)^{t_{wj},i}$$

$$posterior\ probability\colon P(Y = y_k|D_i)$$

$$prior\ probability\colon P(Y = y_k)$$

$$likelihood\colon \prod_{j=1}^{V} P\left(X_j|Y = y_k\right)^{t_{wj},i}$$

As it can be understood from the equation, the likelihoods are similar even if for an unbalanced dataset. However, the prior distributions badly impact the posterior distribution since the class having more data has a higher prior probability which leads to a bias for that class [1]. This bias can result in overfitting to the class having more examples because of the non-uniformity [2].

To solve the imbalance, there are some possible solutions. The first solution is to ignore the prior probabilities but this can cause further problems [1]. Another possible solution is to apply undersampling or oversampling being data analysis techniques to adjust class distribution against the imbalance problem. Some oversampling methods are random oversampling, synthetic minority oversampling, adaptive synthetic sampling, and data augmentation. In all these techniques, the main purpose is to increase the number of examples for classes having less data than others. Hence, they can prevent the data

imbalance by bringing the number of examples in classes close to each other. Some undersampling methods are random undersampling, cluster, tomek links, and undersampling with ensemble learning. In all these algorithms, the fundamental aim is to decrease the number of examples for classes having more data than others. Hence, they can prevent the data imbalance by removing data to bring the number of examples in each class close to each other [3]. One last possible solution is to use the Complement Naive Bayes method as the classifier being a version of Multinomial Naive Bayes. This algorithm is suitable for imbalanced datasets since it calculates the probabilities of not belonging and chose the minimum [2].

3. The validation and training sets have similar data distributions as can be seen in Figure 1 and Figure 2 which show the class distribution for training and validation sets, respectively. Regardless of the similarity of the training and validation distributions in this dataset, having distinct distributions can lead to inaccurate results. The model may be affected by such distinctions since it can be affected by bias. In Naive Bayes, the prior term may mislead the accuracy. Since the model is trained from a different data distribution, while trying the model with a validation set having a different distribution, due to huge differences in prior probabilities for each class, the results can be completely inaccurate.

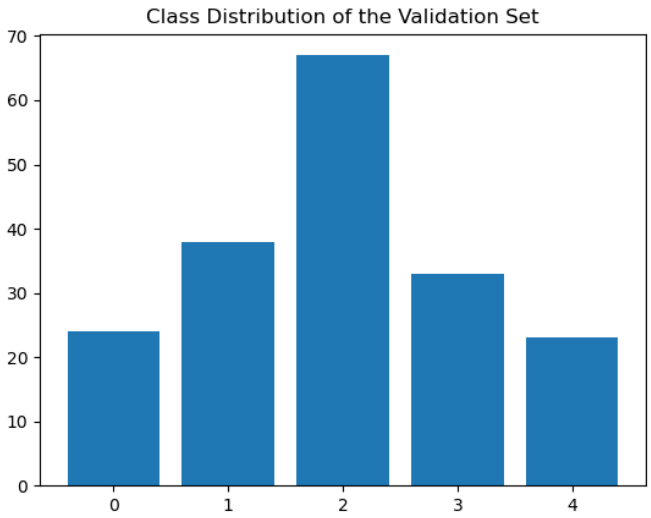

Figure 2: Class Distribution of the Validation Set

4. If the dataset has a skewed distribution, it can impact the accuracy. In such cases, although the accuracy is high, this does not mean that the model is the best if the data is unbalanced. Due to the bias of the class having more examples, the accuracy has also a bias towards that class; thus, even in some cases, it can ignore the classes having fewer examples. Therefore, the reported accuracy can mislead. Nevertheless, although imbalance distribution certainly affects accuracy, to what extent it misleads depends on

the degree of skewness. When the data is slightly skewed toward one class, using accuracy as a performance metric can still give proper results as in the BBCSportNews dataset. Therefore, in this dataset, accuracy doesn't mislead too much. However, when the imbalance is severe accuracy becomes an unreliable performance metric [4]. In such cases, using precision, recall, and F1 score is a better alternative to understanding the true performance of the model.

Question 2.2: Coding Naive Bayes with MLE Estimation

In this part of the assignment, I created the Multinomial Naive Bayes model with the MLE estimator.

$$\theta_{j\,|\,y=y_k} \equiv \frac{T_{j,y=y_k}}{\sum_{j=1}^{|V|} T_{j,y=y_k}}$$

$$\pi_{y=y_k} \equiv \mathbf{P}\left(Y = y_k\right) = \frac{N_{y_k}}{N}$$

Figure 3: MLE Estimator for Multinomial Naive Bayes

I created a Naive Bayes class that contains fit and predict functions. In the fit function, the likelihoods and priors for each class are calculated and stored in NumPy arrays. In the predict function, the predicted classes for new samples (validation dataset) are found by using these likelihoods and priors. In both functions, I tried to use NumPy arrays in order to prevent loops by utilizing NumPy functions.

Initially, I tried my code by ignoring the $log0$ issue and I found the accuracy to be 0.13, approximately. The number of wrong predictions, the accuracy, and the confusion matrix for this case can be seen in Figures 4 and 5.

```
Number of Wrong Prediction for Validation Set MLE :  161
Validation Set MLE Accuracy:  0.12972972972972974
Predicted   0  1  2  3  4
Actual
0          24  0  0  0  0
1          38  0  0  0  0
2          67  0  0  0  0
3          33  0  0  0  0
4          23  0  0  0  0
```

Figure 4: Results for MLE Estimator Without Small Number Assumption

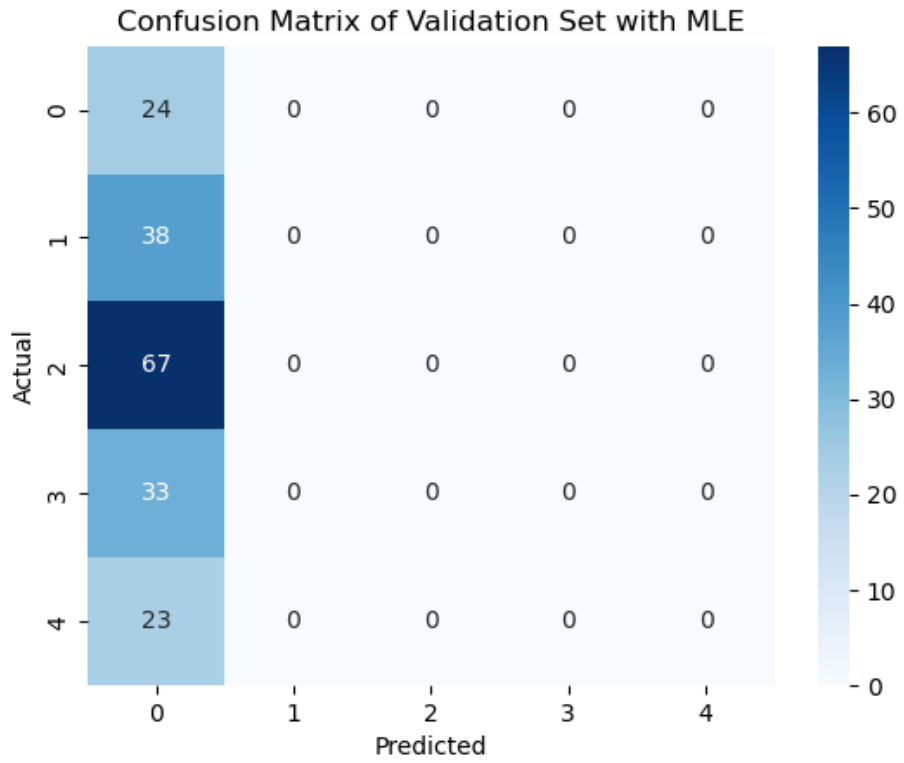

Figure 5: Confusion Matrix for MLE Estimator Without Small Number Assumption

Then, to fix this issue, I used $np.nan\_to\_num()$ function and I found the accuracy as 0.32, approximately. The number of wrong predictions, the accuracy, and the confusion matrix for this case can be seen in Figures 6 and 7.

```
Number of Wrong Prediction for Validation Set MLE :  126
Validation Set MLE Accuracy:  0.318918918918918 94
Predicted   0  1   2  3  4
Actual
0          24  0   0  0  0
1          33  5   0  0  0
2          44  0  23  0  0
3          30  0   0  3  0
4          19  0   0  0  4
```

Figure 6: Results for MLE Estimator With Small Number Assumption

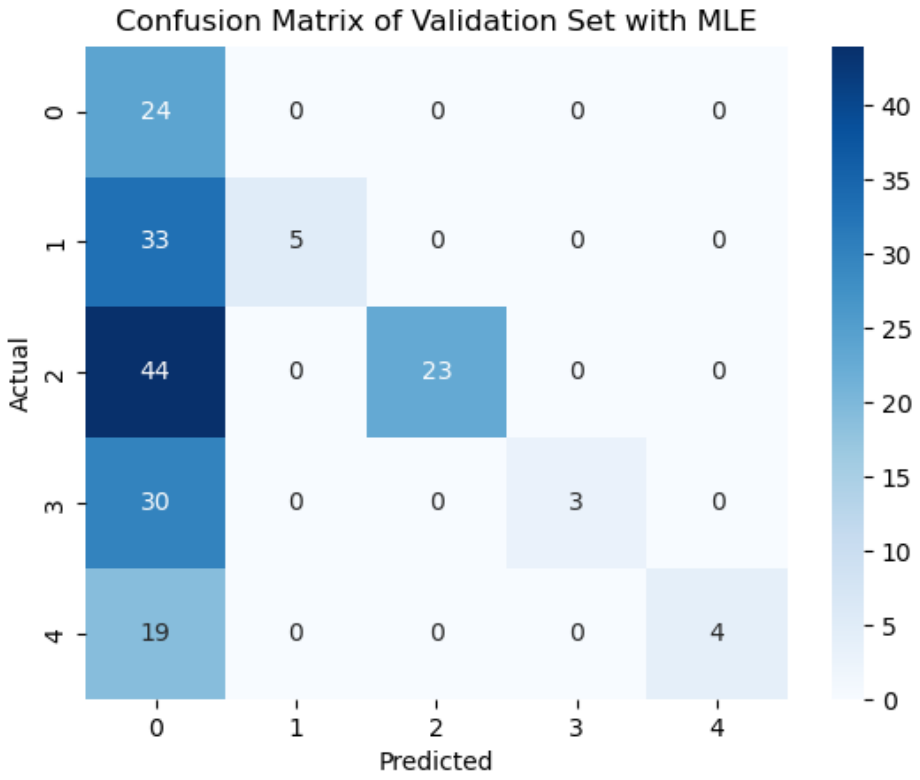

Figure 7: Confusion Matrix for MLE Estimator With Small Number Assumption

Question 2.3: Coding Naive Bayes with MAP Estimation

In this part of the assignment, I extended my Multinomial Naive Bayes model to the MAP estimator by using a fair Dirichlet prior.

$$\theta_{j\,|\,y=y_k} \equiv \frac{T_{j,y=y_k}+\alpha}{(\sum_{j=1}^{|V|} T_{j,y=y_k})+\alpha*|V|}$$

$$\pi_{y=y_k} \equiv \mathbf{P}\left(Y=y_k\right) = \frac{N_{y_k}}{N}$$

Figure 8: MAP Estimator for Multinomial Naive Bayes

I added Dirichlet prior (α) to my fit function and combine all my codes with $\alpha = 0$ for the MLE estimator and $\alpha = 1$ for the MAP estimator. The number of wrong predictions, the accuracy, and the confusion matrix for this case can be seen in Figures 9 and 10.

```
Number of Wrong Prediction for Validation Set MAP :  5
Validation Set MAP Accuracy:  0.972972972972973
Predicted   0   1   2   3   4
Actual
0          24   0   0   0   0
1           0  35   1   2   0
2           0   0  66   1   0
3           0   0   0  33   0
4           1   0   0   0  22
```

Figure 9: Results for MAP Estimator

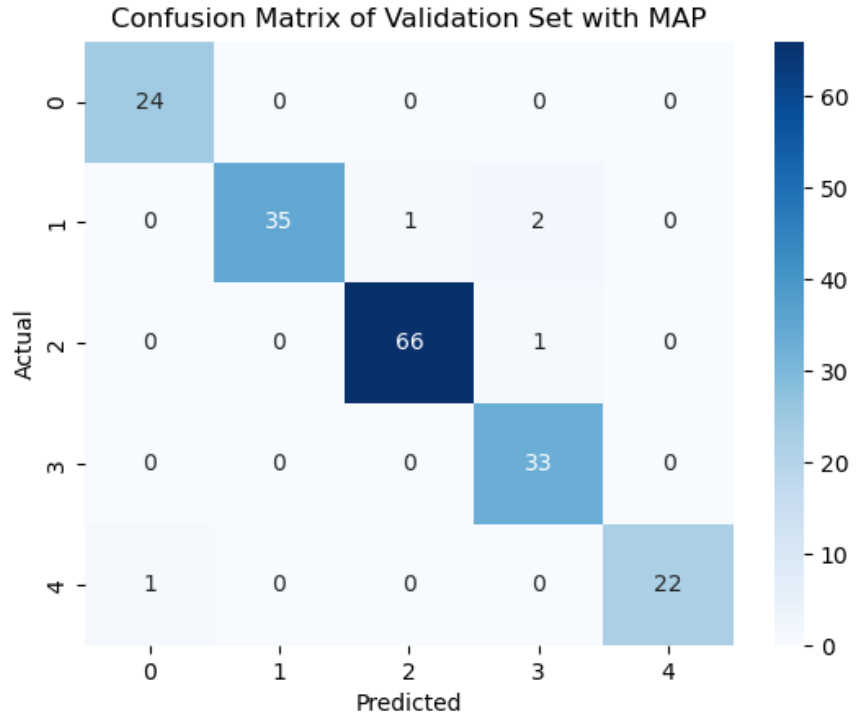

Figure 10: Confusion Matrix for MAP Estimator

Question 2.4: Comparison and Results of Models

In the first estimator (MLE), the accuracy is relatively low. The underlying reason for this problem is the zero probability problem. In our dataset, features can have zero likelihood properties if they a class does not contain a feature in all its samples. This situation is very likely in such a dataset where stop words are removed from the features. Therefore, the features (words)

become more class-specific which causes the zero probability problem. In that case, using a Dirichlet prior (α) (MAP) helps us to prevent this and solve this problem with Laplace smoothing. One imaginary occurrence is added to all words in all samples. Therefore, it is impossible that a likelihood is equal to zero which increases the accuracy by smoothing the algorithm. Hence, Dirichlet prior's effect on the model is to increase the accuracy by preventing the ignorance of zero likelihoods (assign them the value 1). Additionally, as I said, the dataset's structure is prone to zero probabilities since it does not contain stop words. Removing common words (features) causes the class-specific distribution of the word occurrences. Therefore, without Dirichlet prior, the algorithm is prone to zero probabilities which leads to low accuracy. However, with Dirichlet prior, removing stop words lead to a better model and a higher accuracy [5].

References

[1] "How I was using naive Bayes (incorrectly) till now — part-2." [Online]. Available: https://towardsdatascience.com/how-i-was-using-naive-bayes-incorrectly-till-now-part-2-d31feff72483. [Accessed: Oct. 28, 2022].

[2] "Complement naive Bayes (CNB) algorithm," *GeeksforGeeks*, Sep. 05, 2020. [Online]. Available: https://www.geeksforgeeks.org/complement-naive-bayes-cnb-algorithm/. [Accessed: Oct. 28, 2022].

[3] "Oversampling and undersampling in data analysis," *Wikipedia*, Oct. 20, 2022. [Online]. Available: https://en.wikipedia.org/wiki/Oversampling_and_undersampling_in_data_analysis#Oversampling_techniques_for_classification_problems. [Accessed: Oct. 28, 2022].

[4] J. Brownlee, "Failure of classification accuracy for imbalanced class distributions," *Machine Learning Mastery*, Jan. 21, 2021. [Online]. Available: https://machinelearningmastery.com/failure-of-accuracy-for-imbalanced-class-distributions/. [Accessed: Oct. 28, 2022].

[5] "Laplace smoothing in naïve Bayes algorithm | by Vaibhav Jayaswal ..." [Online]. Available: https://towardsdatascience.com/laplace-smoothing-in-na%C3%AFve-bayes-algorithm-9c237a8bdece. [Accessed: Oct. 28, 2022].