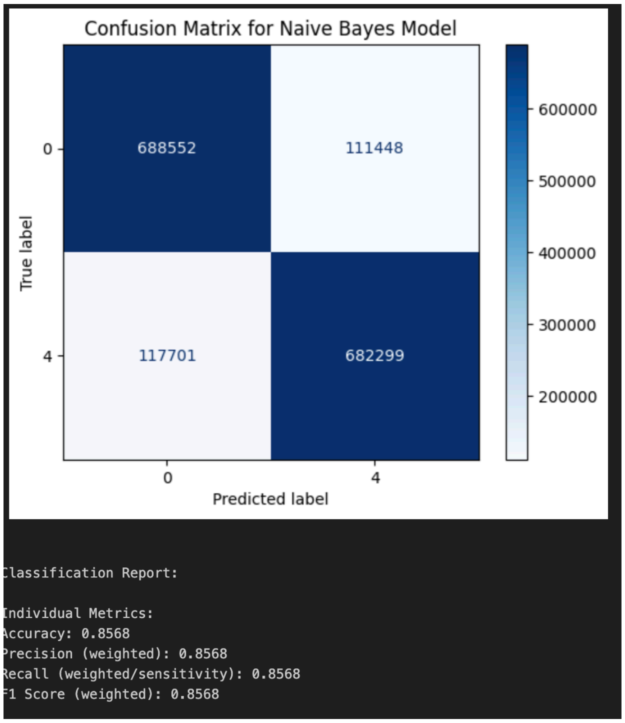
# Homework 1 Write-up

Confusion Matrix and Metrics:



```
Classification Report:

Individual Metrics:
Accuracy: 0.8568
Precision (weighted): 0.8568
Recall (weighted/sensitivity): 0.8568
F1 Score (weighted): 0.8568
```

a=1^e-16

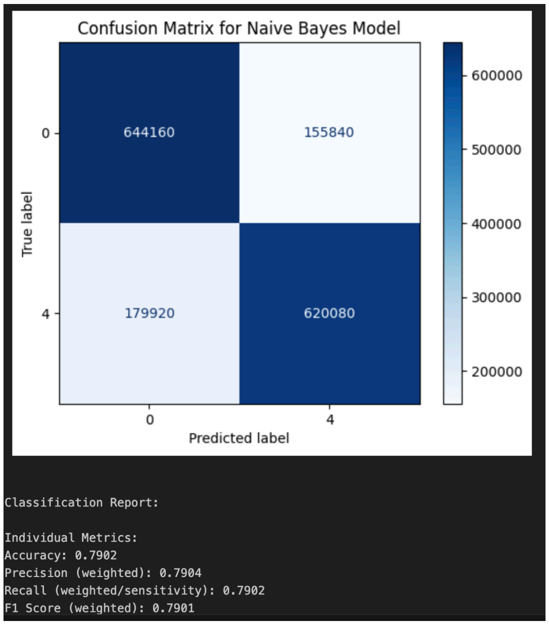
1a. For the metrics of my model, I have as seen in the photo: Accuracy as 85.68%, Precision as 85.68%, Recall as 85.68% and F1 as 85.68%. This means that my data is well balanced. It correctly identified 688,552 instances of Class 0 and 682299 instances of Class 4. The slight discrepancy between False Positives (117701) and False Negatives (111448) suggests the model is almost equally likely to misclassify the information between the 2 classes, and the margin is small enough that the model remains highly effective.

1b. Yes this model is effective. It would be able to be used to see the “buzz” or “vibes” of a tweet, An 86% accuracy rate makes the model much more production-ready than when I ran it with the a value in laplace being larger than current. At this level, it could reliably be used to auto-sort user reviews with high confidence. If we continue to shrink a closer to 0 than I think the accuracy will increase but only by so much, if we added more classes I think it could distinguish between them a lot easier.

1c. I feel like the Naive Bayes is decently reliable in this case where I used the laplace of a=1xe^-16. For the tweets and sentiments, all it’s doing is trying to scale the presence of strongly used words and Naive Bayes does a good job of that. For my methodology, I started with the a value = 1, then 1xe^-6, and 1xe^-16 which as I went the accuracy and other metrics drastically improved. The only drawbacks that Naive Bayes has is that it assumes independence between the words, I feel like if it was implemented where there was dependencies, then it would be way more reliable.

1d. Honestly the things that I found most challenging was that the assignment itself seemed vague for some of the parts and I had to dive into the lectures to even try to remotely figure out what it was asking, also the switching around the words sentiment and label completely threw me off. Because we weren't taught how to code this in Python, I had to look up alot of different basic functions with sklearn

a=1xe^-6:



| True label \ Predicted label | 0 | 4 |
|---|---|---|
| 0 | 644160 | 155840 |
| 4 | 179920 | 620080 |

```
Classification Report:

Individual Metrics:
Accuracy: 0.7902
Precision (weighted): 0.7904
Recall (weighted/sensitivity): 0.7902
F1 Score (weighted): 0.7901
```