SHREYA BIRTHARE

34060222

1. Results:

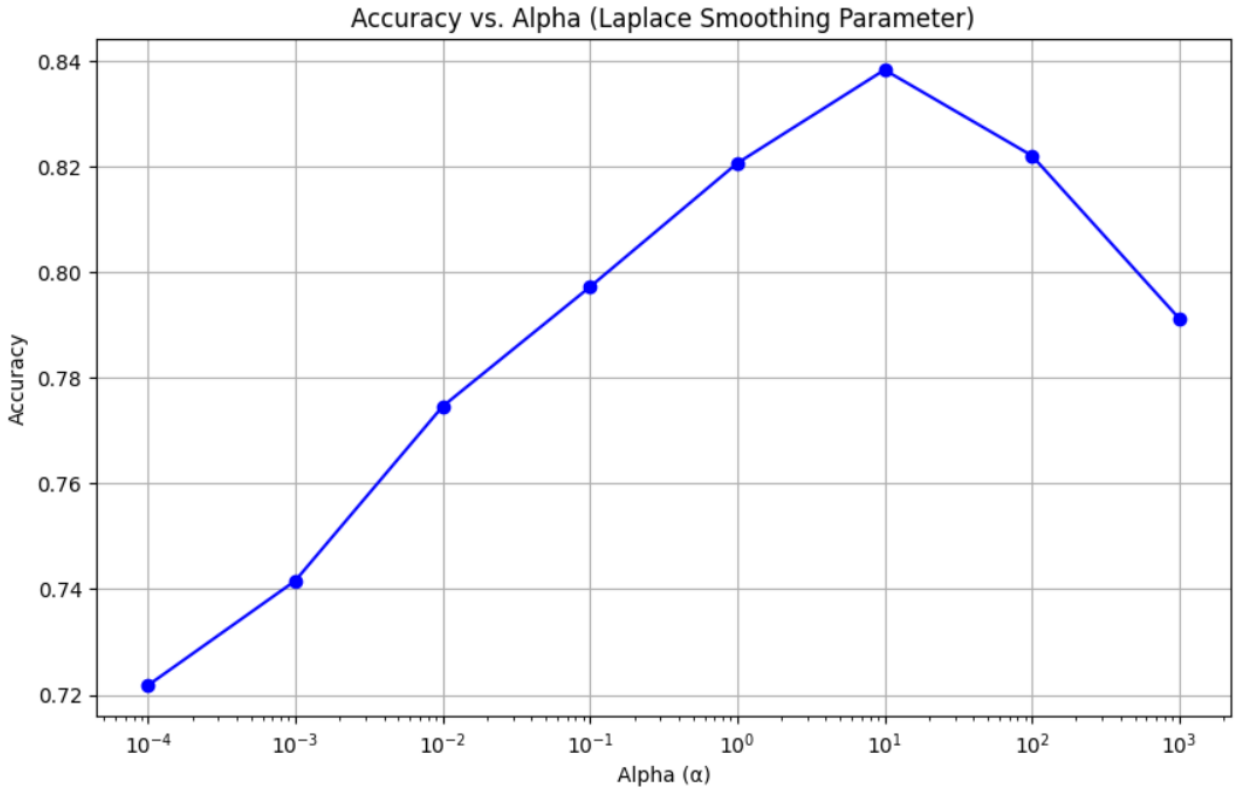
```
==========================================Q1==========================================
*****************RESULTS FOR posterior model*****************
Accuracy:  0.5294587577391652
Confusion Matrix [[tp, fp], [tn, fn]]:  [[1270, 1180], [1381, 1176]]
Precision:  0.5183673469387755
Recall:  0.5192150449713818
*****************RESULTS FOR log model*****************
Accuracy:  0.7467608951707891
Confusion Matrix [[tp, fp], [tn, fn]]:  [[1795, 538], [2009, 752]]
Precision:  0.7693956279468496
Recall:  0.7047506870828426
```

Here accuracy of the posterior model is 52.94% whereas accuracy of the log model is 74.67%. We observe that the accuracy of the log based model is better than the accuracy of the normal (posterior) model. The log model is more numerically stable here. Multiplying many small probabilities (posterior model) can lead to numerical underflow, where the product becomes too small for the computer to represent accurately, effectively becoming zero. This can distort the classification results. Logarithms mitigate this by transforming the product of probabilities into a sum of logarithms, greatly reducing the risk of underflow and maintaining good computational accuracy. We also see a significant improvement in precision and recall in the log based model when compared to the posterior model. This is because using log yields to numerically stable results, reducing the chances of overflows and zero probabilities thereby leading to effective/better classification of the results. We can see that the number of true positives and negatives increase and the number of false positive and false negative classification of documents decreases in the log based model

2. Results when alpha=1:

```
==========================================Q2==========================================
*****************RESULTS FOR log model*****************
Accuracy:  0.8205794205794206
Confusion Matrix [[tp, fp], [tn, fn]]:  [[1871, 279], [2236, 619]]
Precision:  0.8702325581395349
Recall:  0.7514056224899598
```

Alpha v/s Accuracy graph using the log based model:



From the graph we can see that as we vary alpha from 0.0001 to 10 the overall accuracy of the model increases. It is the highest ~84% when alpha=10. From alpha=100 to alpha=1000 we see that the overall accuracy decreases again. When alpha is too low i.e. from 0.0001 to 0.1 there is hardly or very less smoothening happening. The impact of having no occurrence of a word in the training data is almost the same as in the original formulation or very little without smoothing, leading to a high risk of zero probabilities, which can negatively affect the accuracy of the model. The model can become too sensitive to the training data and may not generalize well to unseen data, resulting in lower accuracy on the test set and potential overfitting on the training data. When alpha is moderate (alpha=1, 10) in our case, it introduces a moderate amount of smoothing, effectively mitigating the zero-frequency problem without heavily skewing the distribution of word probabilities. This helps the model to generalize better and handle unseen words effectively, leading to higher accuracy. When alpha is too high i.e. when the model is excessively smoothed, it leads to a distribution that does not represent the true data. It diminishes the influence of the actual word count and can bias the probability estimations towards the uniform distribution, which is not informative and can reduce the model's ability to make accurate predictions. The model might become too generalized and fail to capture the patterns in the data, which can reduce its effectiveness in distinguishing between classes, resulting in lower accuracy.

3. We used alpha=10 here as it resulted in the highest accuracy as seen in the second question, 100% of the training data and 100% of the testing data

```
====================================Q3====================================
*****************RESULTS FOR posterior model*****************
Accuracy:  0.66264
Confusion Matrix [[tp, fp], [tn, fn]]:  [[7878, 3812], [8688, 4622]]
Precision:  0.6739093242087254
Recall:  0.63024
*****************RESULTS FOR log model*****************
Accuracy:  0.83692
Confusion Matrix [[tp, fp], [tn, fn]]:  [[9924, 1501], [10999, 2576]]
Precision:  0.8686214442013129
Recall:  0.79392
```

4. Using alpha=10, 50% of the training data and 100% of the testing data

```
====================================Q4====================================
*****************RESULTS FOR posterior model*****************
Accuracy:  0.65864
Confusion Matrix [[tp, fp], [tn, fn]]:  [[7818, 3852], [8648, 4682]]
Precision:  0.6699228791773779
Recall:  0.62544
*****************RESULTS FOR log model*****************
Accuracy:  0.83512
Confusion Matrix [[tp, fp], [tn, fn]]:  [[9882, 1504], [10996, 2618]]
Precision:  0.8679079571403477
Recall:  0.79056
```

When comparing the stats for both the 3rd and the 4th question, we see that when we use 50% of the training data, the accuracies of both the posterior and the log model decrease, although by very slight margin. These changes are relatively small, indicating that both models are quite robust to the reduction in training data size, with the log models maintaining a notably higher accuracy overall. Precision drops slightly in both models when the training set size is reduced, which indicates that with less data, the model becomes marginally less precise in predicting the positive class. Recall also decreases for both models, suggesting that with less training data, the model's ability to identify all relevant instances of the positive class is somewhat diminished.

Analysis of Confusion Matrices:

- Posterior Model:
    - The number of true positives (TP) and false negatives (FN) changes slightly, indicating a small decrease in the model's ability to correctly classify positive instances.

    - The decrease in recall is consistent with the increase in false negatives when the training set is smaller.
    - The slight decrease in precision can be seen with the increase in false positives (FP).
- Log Model:
    - Similarly to the posterior model, there is a small decrease in the number of true positives and an increase in false negatives, which aligns with the slight drop in recall.
    - The precision of the log model is less affected by the reduction in training data compared to the posterior model.

The reduction in recall for both models indicates that the positive class is slightly more affected by the change in training set size. The model becomes a bit less capable of identifying all positive instances, but this effect is relatively small.

5. In the use case of classifying movie reviews we can say that movie reviews would be read and used by user to determine if or not they should watch a movie (a recommender system). Out of high accuracy, precision and recall, we can say that high precision is slightly more important than the other two. A false positive (a bad movie incorrectly labeled as good) directly affects the user's experience, leading to dissatisfaction and possibly loss of trust in the recommendation system. Users may have limited time and may base their viewing choices on these classifications. A high precision ensures that when a movie is classified as having a positive review, it is likely to be a good movie. While high recall is also important to avoid missing good movies, a false negative (a good movie labeled as bad) is less impactful because users still have many other correctly identified positive movies to choose from. A good movie incorrectly labeled as bad might be missed, but it doesn't degrade the user experience directly since the user is not likely to watch it based on the negative classification. High precision ensures that when a movie is recommended based on positive classification, it is truly well-received and likely to be enjoyed. Trust in the platform's recommendations is built on consistently accurate positive predictions. This precision protects against the negative user experience associated with poor recommendations and supports business goals related to user engagement and satisfaction. While accuracy and recall are also important, in a context where user experience is directly tied to the recommendations they receive, the precision of those recommendations takes on a heightened importance.

6. Using alpha=10, 10% of positive training instances, 50% of negative training instances, and 100% of the testing data

```
====================================Q6====================================
*****************RESULTS FOR posterior model*****************
Accuracy:  0.49892
Confusion Matrix [[tp, fp], [tn, fn]]:  [[3231, 3258], [9242, 9269]]
Precision:  0.49791955617198336
Recall:  0.25848
*****************RESULTS FOR log model*****************
Accuracy:  0.5002
Confusion Matrix [[tp, fp], [tn, fn]]:  [[6, 1], [12499, 12494]]
Precision:  0.8571428571428571
Recall:  0.00048
```

For Balanced dataset (Q4):

```
====================================Q4====================================
*****************RESULTS FOR posterior model*****************
Accuracy:  0.65864
Confusion Matrix [[tp, fp], [tn, fn]]:  [[7818, 3852], [8648, 4682]]
Precision:  0.6699228791773779
Recall:  0.62544
*****************RESULTS FOR log model*****************
Accuracy:  0.83512
Confusion Matrix [[tp, fp], [tn, fn]]:  [[9882, 1504], [10996, 2618]]
Precision:  0.8679079571403477
Recall:  0.79056
```

Impact of unbalanced dataset:

For both models, accuracy decreases notably when trained on the unbalanced dataset. This is expected as the models are now biased towards the class with more training data (negative reviews). For the posterior model, precision drops, indicating that it becomes less reliable in its positive predictions. The log model's precision is quite high. However, this is misleading because the log model is hardly predicting any positives at all, as evidenced by the extremely low recall. The few times it does predict positive, it happens to be correct, hence the inflated precision score. There is a significant decrease in recall for both models, especially the log model, which now almost never correctly identifies positive instances. This is a direct consequence of the unbalanced training data, where the model has seen so few positive examples that it fails to learn the characteristics of a positive review effectively. For the posterior model, the number of true positives and true negatives both drop, while false positives and false negatives increase, which is consistent with the decrease in accuracy and recall. For the log model, the confusion matrix shows an extreme case where almost all instances are predicted as negative, which is why recall is near zero. The model is heavily biased towards the negative class due to the

disproportionate amount of negative training data. In terms of the impact on each class, the positive class (the minority class in the unbalanced dataset) is more affected. The models trained on the unbalanced data are not capturing the characteristics of the positive reviews adequately, leading to a large number of false negatives. Training with an unbalanced dataset has significantly compromised the models' ability to accurately classify the minority class (positive reviews). The models have become biased towards the majority class (negative reviews), leading to poor recall, and in the case of the log model, precision is misleadingly high. This demonstrates the importance of having a balanced dataset for training, or at least applying techniques to account for the imbalance, to ensure that the model can generalize well to all classes.