Sunilsakthivel Sakthi Velavan
Roni Khardon
CSCI-B 555
15 Sep.2023

Programming Project 1

Experiment 1:

For this experiment, I generated 3 different charts including learning curves for Naive Bayes to draw a contrast between m = 0 and m = 1 for the imdb, yelp and amazon datasets. For each dataset, we plot averages of the accuracy and standard deviations as a function of train set size.

Charts:

Figure 1

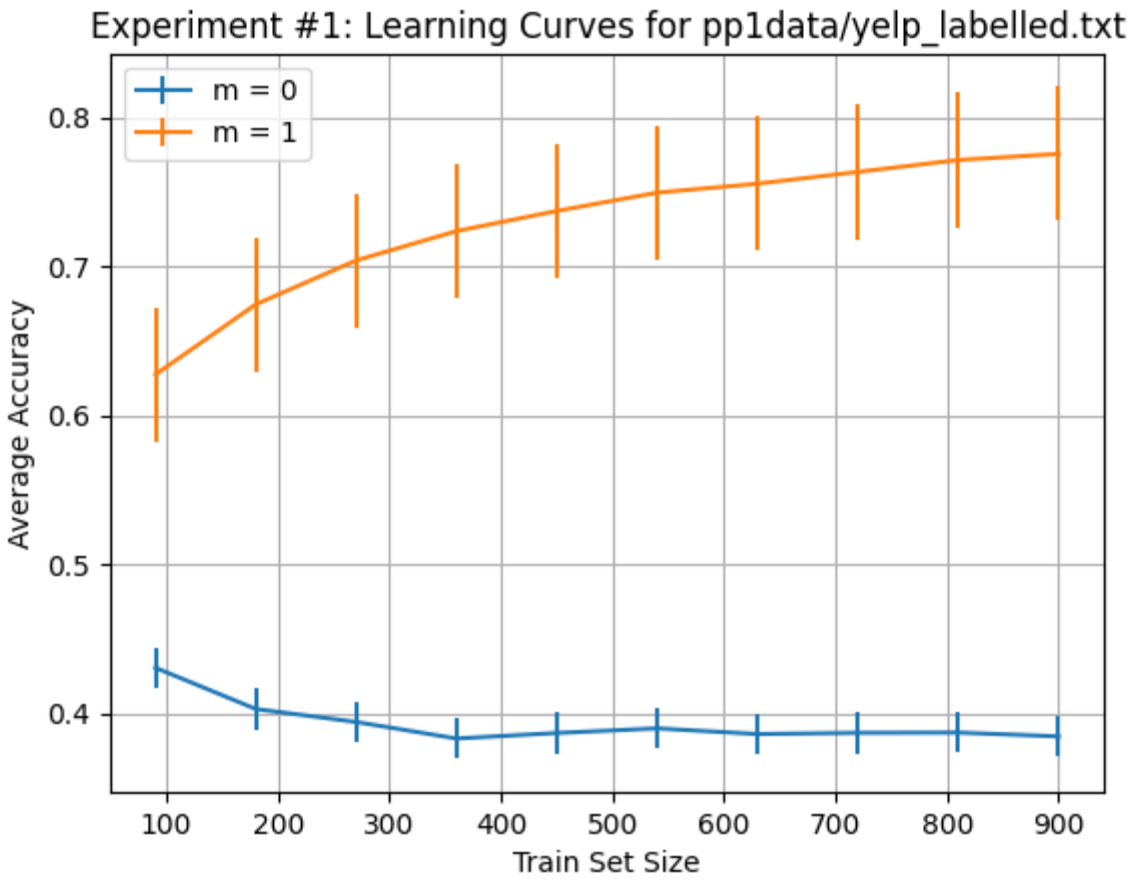


Figure 2

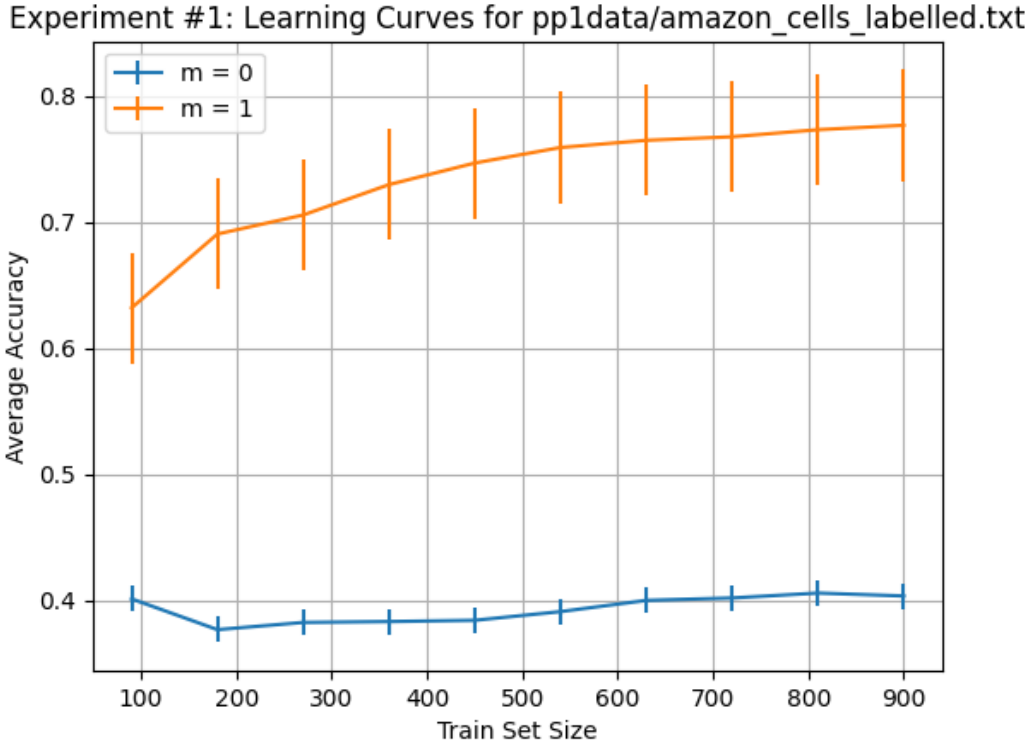
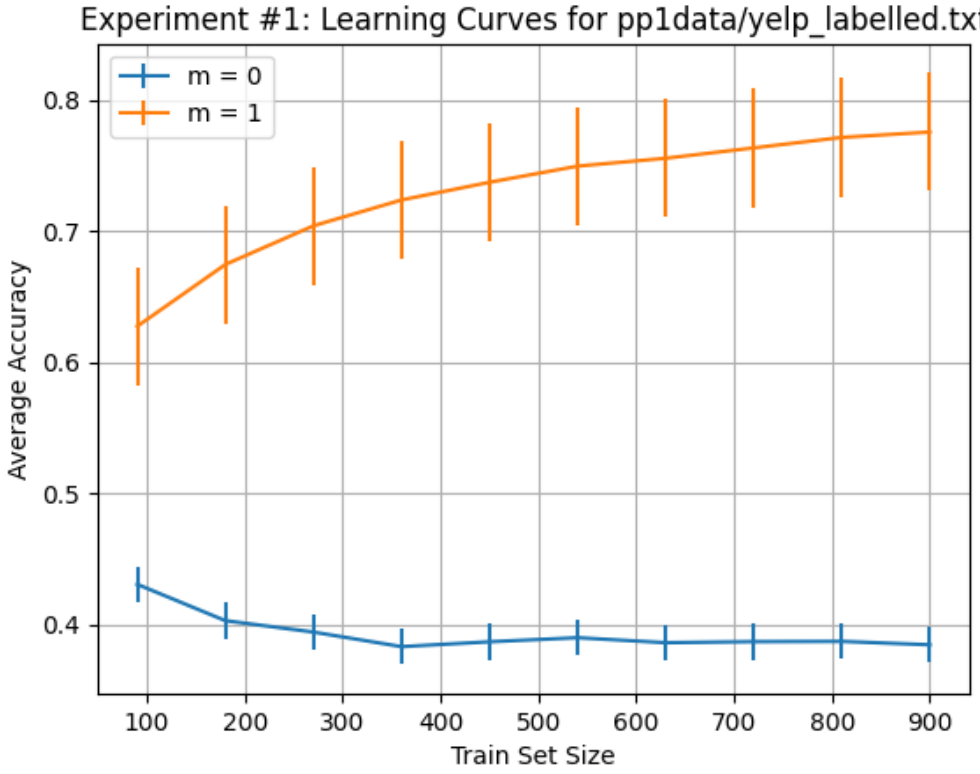


Figure 3



Observations

In all three datasets, as seen in Figure 1,2 and 3, the inclusion of m = 1 has a considerable impact on the values on the average accuracy when compared to the accuracy when m = 0. As a matter of fact, the m = 0 accuracy actually drops as the train set size grows and settles at an equilibrium point right below 40%. However, the m = 1 accuracies grow from 0.6 to near 0.8 as the train sizes grow, indicating a better overall performance with m = 1. Interestingly, the standard deviation of values tends to skew larger on the m=1 learning curve as opposed to the m = 0 learning curve as seen by the error ranges on the curves.

Experiment 2:

For this experiment, I generated 6 charts altogether with Figures 4-6 representing the cross validation accuracy and standard deviations as a function of the smoothing parameters m = 0.1-0.9, and Figures 7-9 representing the same as a function of the smoothing parameters m = 1-20

Figure 4

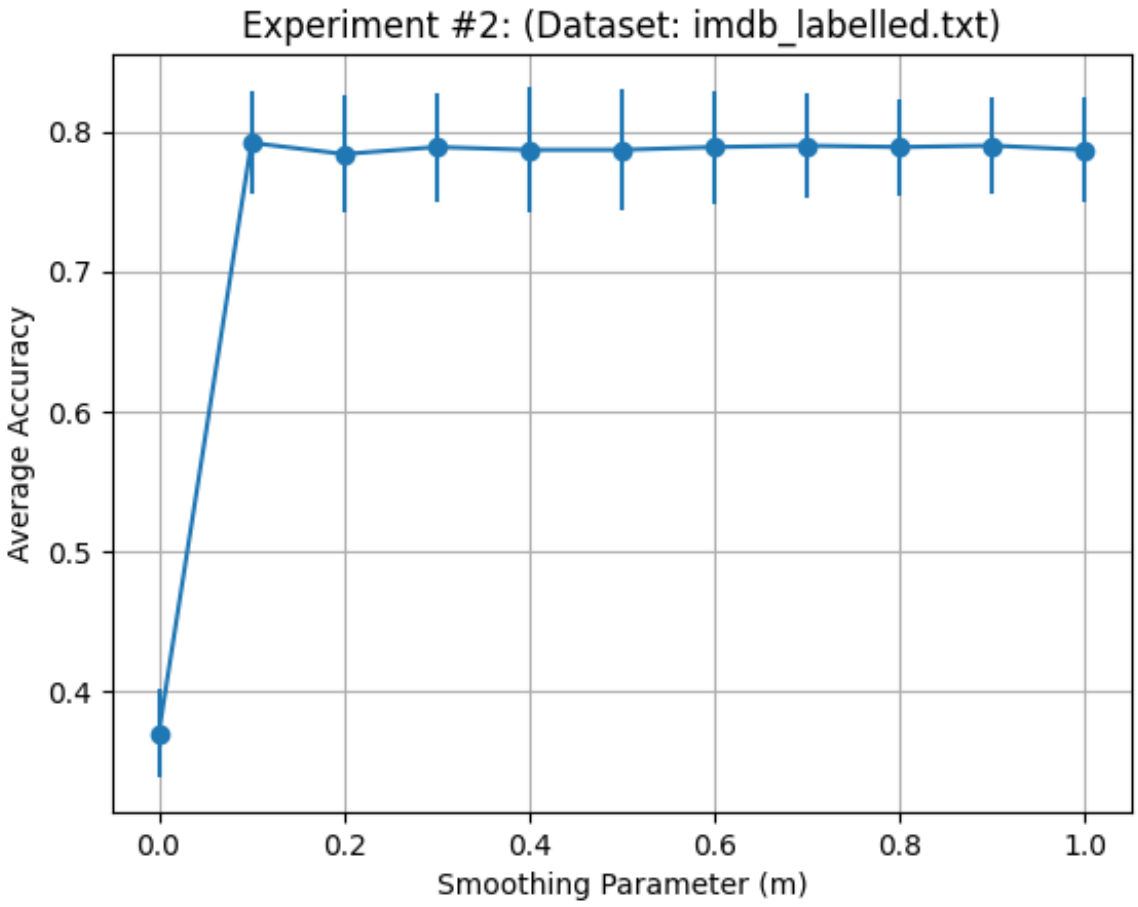


Figure 5

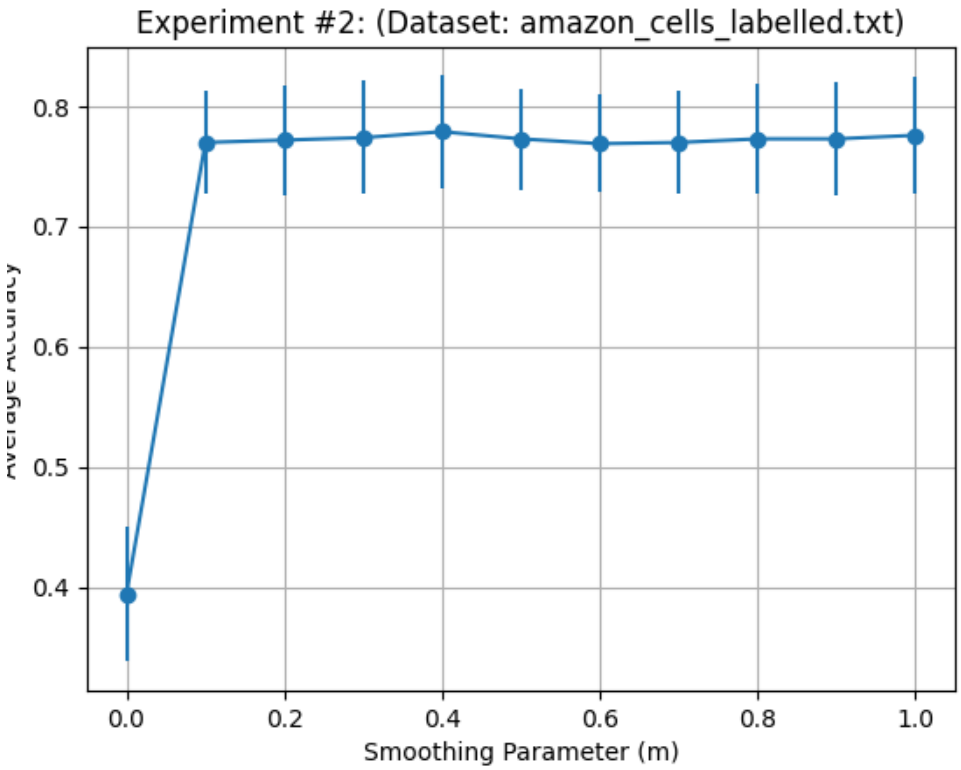


Figure 6

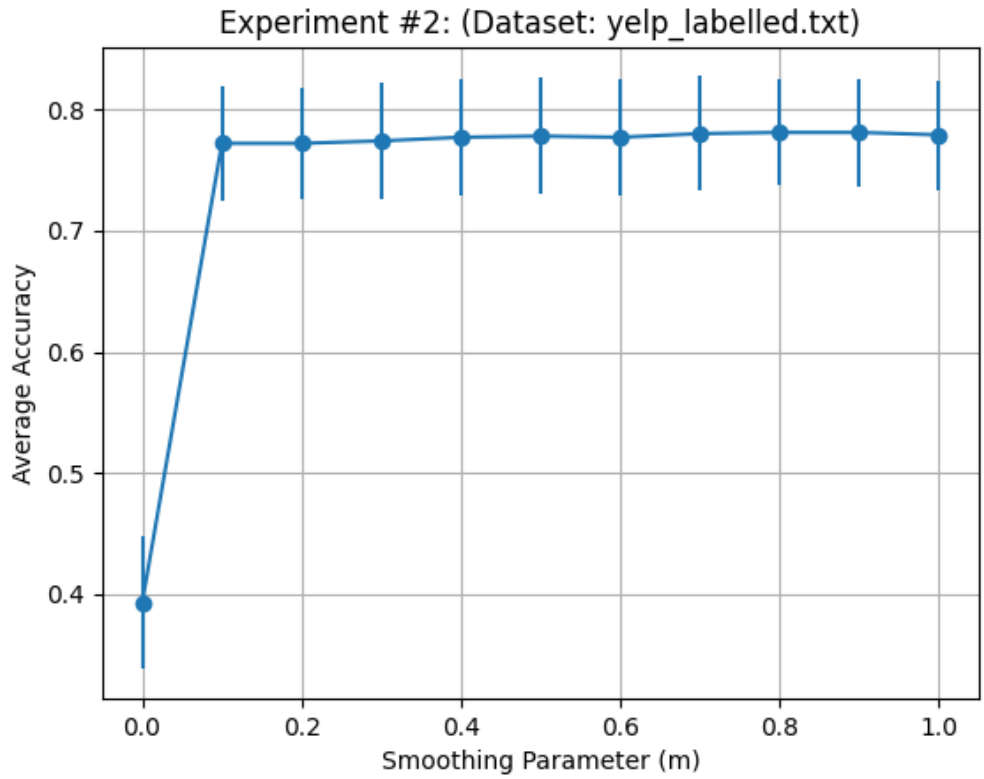


Figure 7

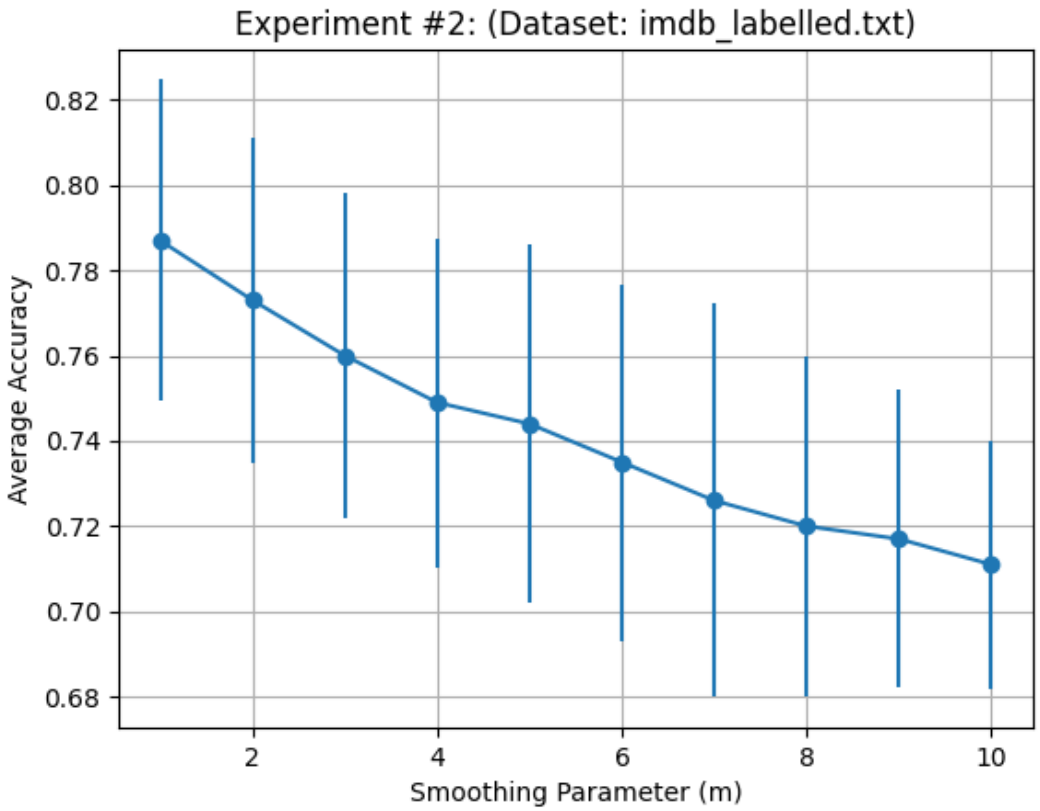


Figure 8

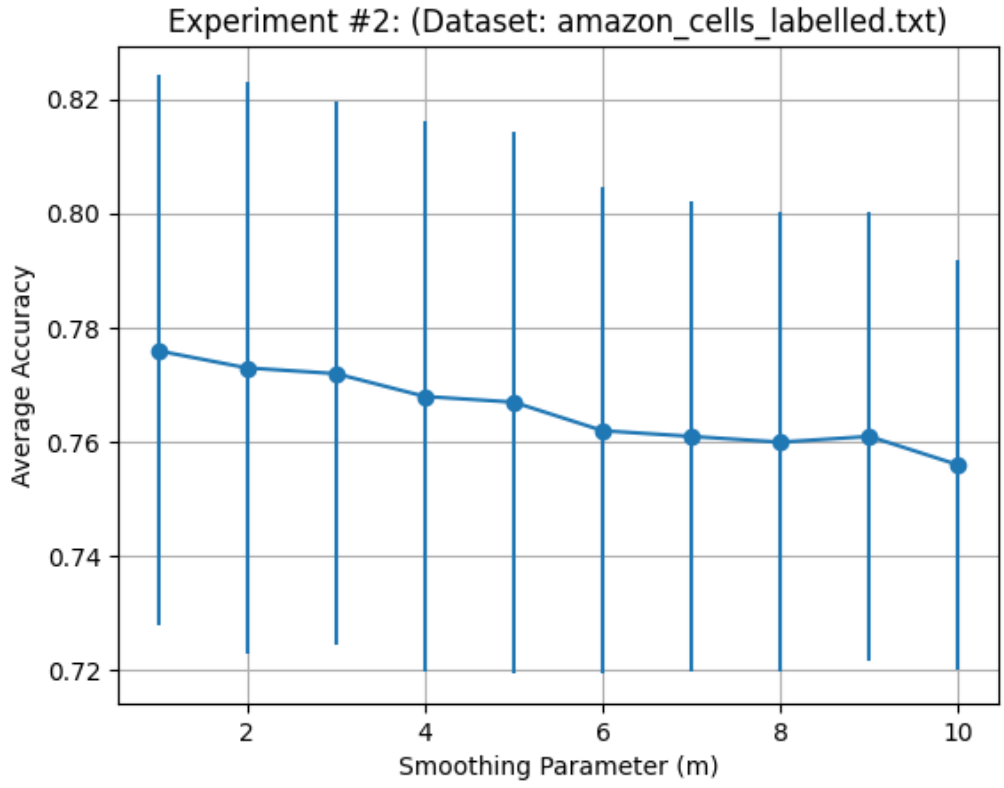
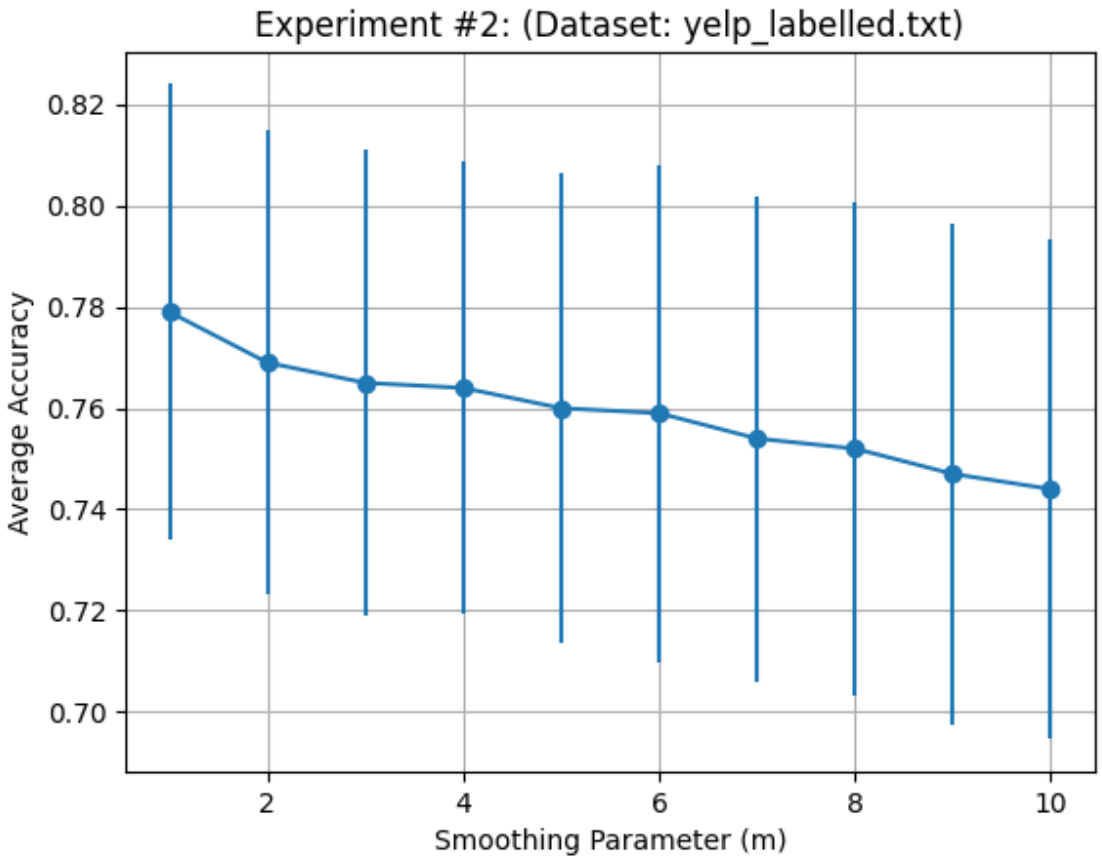


Figure 9



Observations

In Figures 4-6, where the m falls in the range of 0.1-1.0, one can notice the trend of the avg accuracy score taking a quick spike up from 40% to close to 80% between an m value of 0.0 and 0.1 after which accuracy values settle around the 80% mark until m = 1.0. This suggests that a small amount of smoothing (m = 0.1) helps the classifier generalize better. In direct contrast, when looked at from a larger scale, i.e Figures 7-9, the larger range of values of m being between 0-10 actually starts to drop the average accuracy score on a negative trend. The negative trend in accuracy suggests that with larger values of m, the classifier is over-smoothing the data, causing it to perform less effectively.