Coursework	Fall 2021
Machine Learning (B555 - 17011)	Prof. Roni Khardon
Department of Computer Science, IU Bloomington	

Topic : Programming project 1 – Naïve Bayes for Text Classification Report
Submitted by : Pratap Roy Choudhury, MS in Data Science, Fall2021

1. Overview

In this assignment, the Naive Bayes algorithm with maximum likelihood and MAP solutions are implemented on a set of textual data, and it has been evaluated using cross validation on the task of sentiment analysis (as in identifying positive/negative product reviews).

2. Text Data for Sentiment Analysis

There are 3 text data files containing the text data and class labels.
Below are the data sources from where reviews were taken -

- imdb.com
- amazon.com
- yelp.com

Each dataset is given in a single file, where each example is in one line of that file. Each such example is given as a list of space separated words, followed by a tab character (\t), followed by the label, and then by a newline (\n).

Example : Crust is not good.\t0\nThe movie was awesome\t1…..etc

3. Implementation

3.1. Naive Bayes for Text Categorization

The "Naive Bayes for text categorization" algorithm is implemented as discussed in class. In our application every "document" is one sentence as explained above.
Algorithm steps :

For each training set:

i. Count the number of documents in each class (0/1)
ii. Get the tokens from all documents for each class (0/1)
iii. Find the probability of each class (0/1)
iv. Now, for a word given its class, get the probability ie, (#word in that class)/total #words in the vocabulary
v. Given the test set, parse each example in test set and for each word in the example, calculate the probability of that words to be in either of the class and predict it's class.

Important point for prediction: If a word in a test example did not appear in the training set at all (i.e. in any of the classes), then simply skip that word when calculating the score for this example. However, if the word did appear with some class but not the other then use the counts you have (zero for one class but non zero for the other).

Logic – Available words = (all words from test set) $\cap$ (all words from train set)

3.2. Maximum Likelihood and MAP Solutions

Here, we are using the feature of type "token in document is word w", so that each "token Feature" has as many values as words in the vocabulary (all words in training files). The maximum likelihood (and MAP) estimates of parameters are given by the solution for a Discrete distribution
(with a Dirichlet prior) for its parameters.

The maximum likelihood estimates of p(w|c) for word w and class c is $\#(w \cap c)/\#(c)$ where $\#(w \cap c)$ is the number of word tokens in examples of class c that are the word w and #(c) is the number of word tokens in examples of class c.

If we use a prior with parameter vector where all entries are equal to m+1, that is, (m+ 1)*1, the effect is that of adding a pseudo count of m to all entries. In this case, the MAP estimate of p(w|c) is $(\#(w^c)+m)/(\#(c)+mV)$ where V is the vocabulary size and other parameters are as above.

In our experiment, for multiple iteration of m =0,1 or m=0, 0.1, 0.2,…0.9 and 1, 2,…9, the MAP estimate of each token in training set has been calculated , where V is the vocabulary size of the training sets for each sub-samples.

3.3. Cross Validation

The program is implemented to read and parse a dataset for 10-fold stratified cross validation. In this experiment,

for each of the K fold which is 10:

i. for each 1 test portion, the remaining 9 portions of training sets are again split into 10 sub-samples of training set.
ii. The sub-samples iterate through K iterations and in each iteration, the training sub-sample gets increased 10% and the entire MLE and MAP estimations are done for the tokens present in examples of that training sub-sample.
iii. Once the model is built on the training sub-sample, the test data set for the 1st fold of K folds are classified and the accuracy, standard deviation of accuracy are stored, and the same runs for each of the K folds.

3.4. Learning Curves with Cross Validation

Learning curves evaluate how the predictions improve with increasing train set size. To measure this with cross validation we follow the below procedure.

For i = 1, 2….k, Repeat:

i. First generate the folds for cross validation, call these train-i, test-i
ii. Random shuffle the data set and then split into test and train set.
iii. Say train-i has N examples. Then use subsamples of train-i of sizes 0.1N, 0.2N,….0.9N, N as train sets and evaluate the prediction on test-i measuring the accuracy in each case.
iv. Calculate the average and standard deviation for each train set size.

This constitutes the learning curve.

4. Experiments

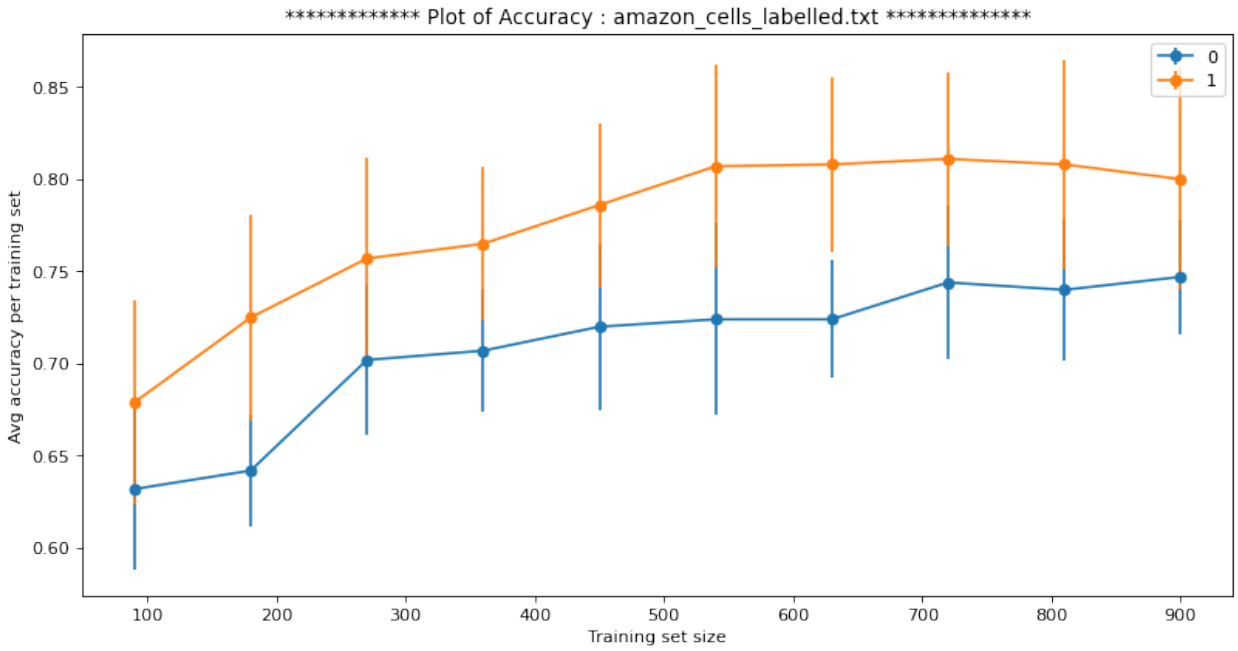
4.1. Exp 1: For each of the 3 datasets run stratified cross validation to generate learning curves for Naïve Bayes with m = 0 and with m = 1. For each dataset, plot averages of the accuracy and standard deviations (as error bars) as a function of train set size.

The program is executed for all the 3 datasets.

The sample outputs of average accuracy per training set size and the standard deviations of accuracy per training set size for both the smoothing parameters m = 0 and m = 1 are listed below. Also, the corresponding plots as a function of the training set size are displayed with the observations listed.

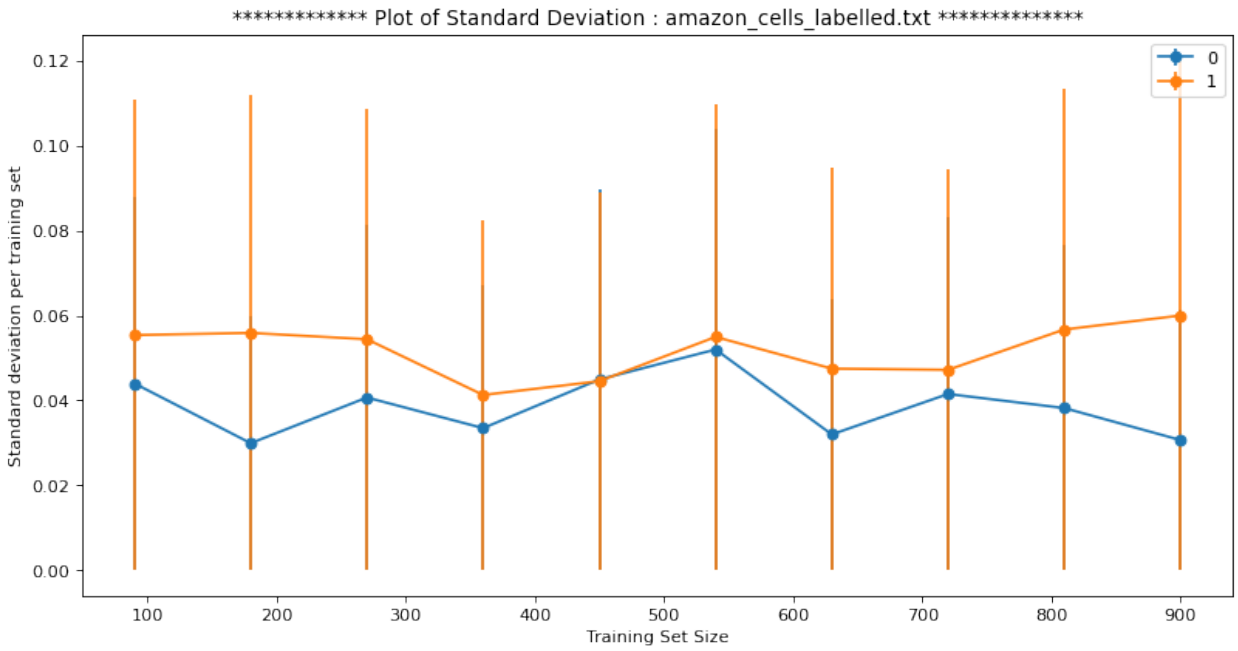
Data set **: *amazon_cells_labelled.txt***

```
********** Processing the data file : amazon_cells_labelled.txt **********

***************** RESULTS *****************

avg_accuracy for smoothing parameter m = 0: [0.6319999999999999, 0.6420000000000001, 0.702, 0.7070000000000001, 0.72,
0.724, 0.724, 0.744, 0.74, 0.747]

SD_ for smoothing parameter m = 0: [0.04400000000000002, 0.02993325909419153, 0.0406939798987516, 0.0334813380855664
1, 0.04494441010848846, 0.05199999999999999, 0.032, 0.04152107898405341, 0.03820994634908563, 0.030675723300355937]

avg_accuracy for smoothing parameter m = 1: [0.679, 0.725, 0.757, 0.7649999999999999, 0.786, 0.807, 0.808, 0.81099999
99999999, 0.808, 0.8]

SD_ for smoothing parameter m = 1: [0.05539855593785817, 0.05590169943749474, 0.054415071441651176, 0.041291645644125
17, 0.044542114902640l8, 0.05496362433464518, 0.047497368348151665, 0.047212286536451493, 0.05670978751503131, 0.0600
0000000000001]

 training data points : [90, 180, 270, 360, 450, 540, 630, 720, 810, 900]
```



We observe that, in Average accuracies Vs Training set size plots-

- Accuracy increases with the increasing training set size.
- The accuracy of the model is higher with the posterior knowledge (m=1) compared to that of the usual maximum likelihood with m=0.



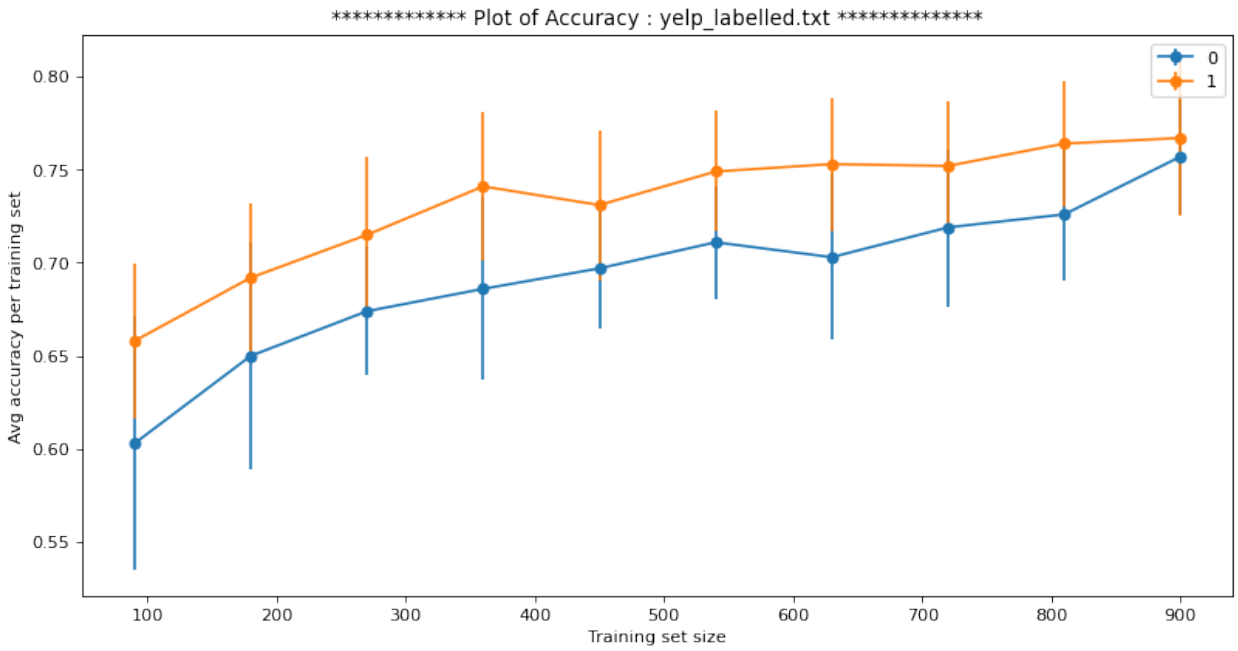
We observe that, in Standard Deviations of accuracies Vs Training set size plots-

- Standard deviations fluctuate between 0.03 to 0.05 with the increasing training set size.
- Standard Deviations of Model with m=1 is comparatively higher than that of with m=0 and they fluctuate between 0.04 to 0.06.
- With m=0, the model has lowest standard deviation with maximum training set size.

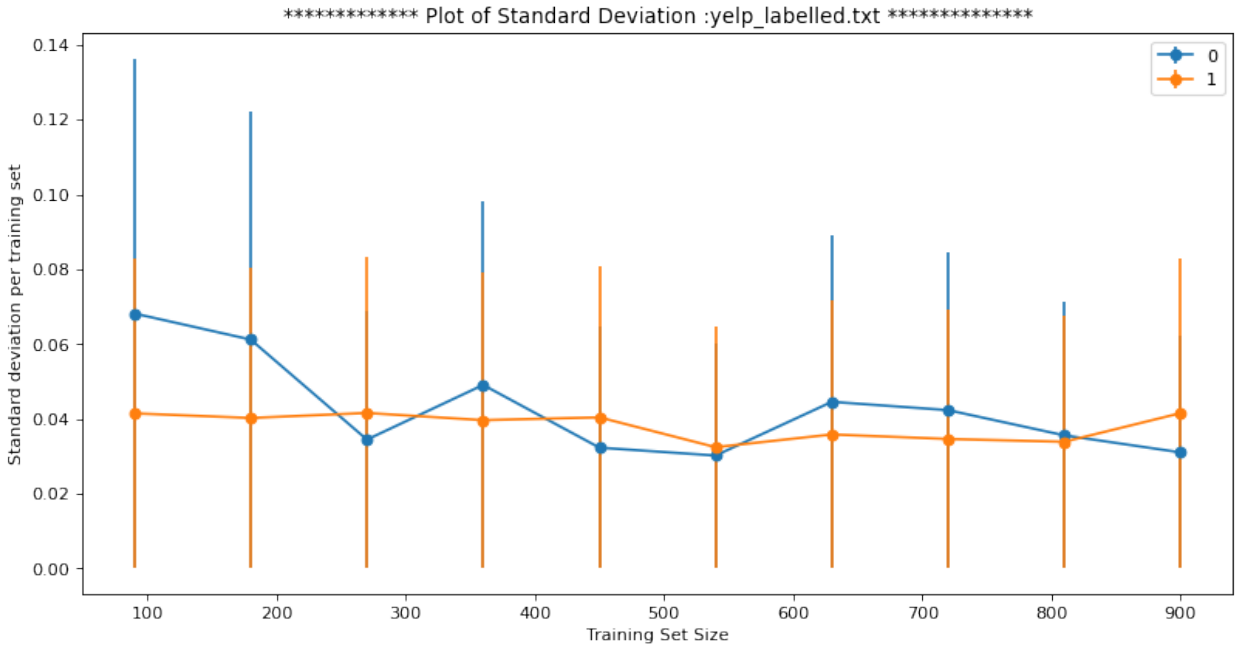
Similar Observations are followed in the other 2 datasets and the results and plots are given below.

Data set **: *yelp_labelled.txt***

```
********** Processing the data file : yelp_labelled.txt **********

***************** RESULTS *****************

avg_accuracy for smoothing parameter m = 0: [0.603, 0.65, 0.674, 0.6859999999999998, 0.6969999999999998, 0.7110000000
000001, 0.703, 0.719, 0.726, 0.7569999999999999]

SD_ for smoothing parameter m = 0: [0.0681248853210044, 0.061155539405682635, 0.0344093010681705, 0.0490306026885250
4, 0.032264531609803346, 0.030149626863362672, 0.04450842616853577, 0.042296571965113196, 0.035552777669262355, 0.031
00000000000001]

avg_accuracy for smoothing parameter m = 1: [0.6580000000000001, 0.692, 0.715, 0.741, 0.7309999999999999, 0.749, 0.75
3, 0.752, 0.764, 0.7670000000000001]

SD_ for smoothing parameter m = 1: [0.04142463035441595, 0.040199502484483556, 0.041533119314590375, 0.03961060464067
67, 0.04036087214122115, 0.032388269481403324, 0.035791060336346596, 0.0345832329315812, 0.03382306905057554, 0.04148
4937025383084]

 training data points : [90, 180, 270, 360, 450, 540, 630, 720, 810, 900]
```



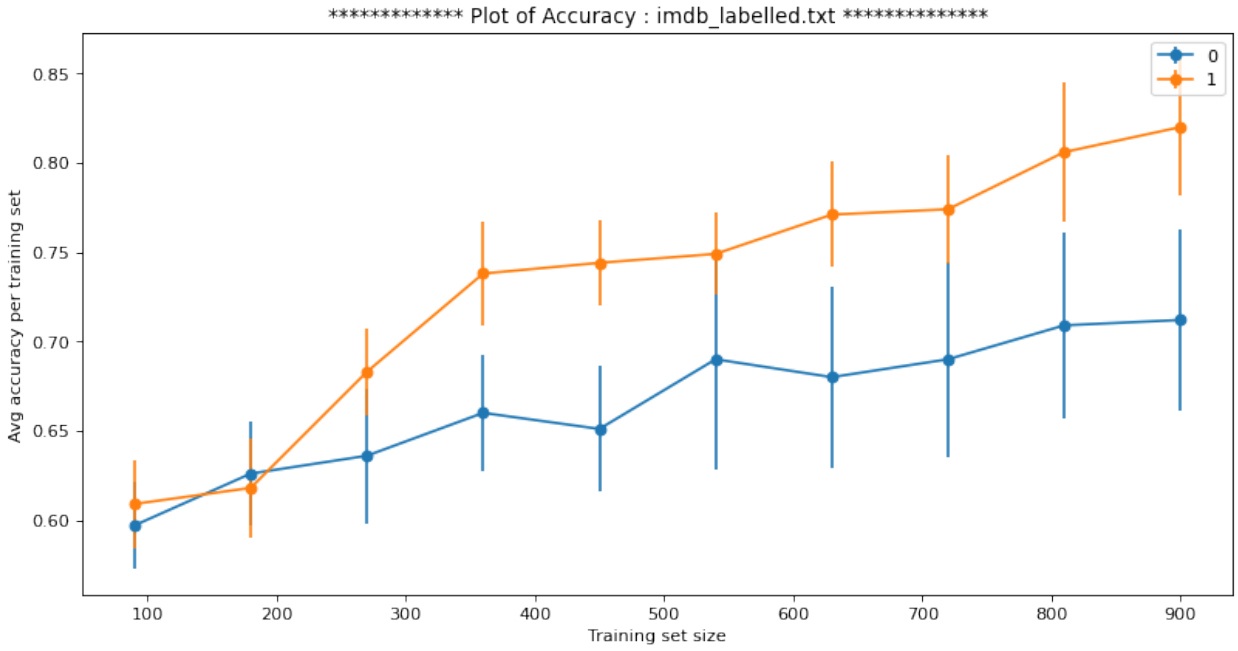
- Accuracy increases for bot m=0 and m=1 with the increasing training set size.



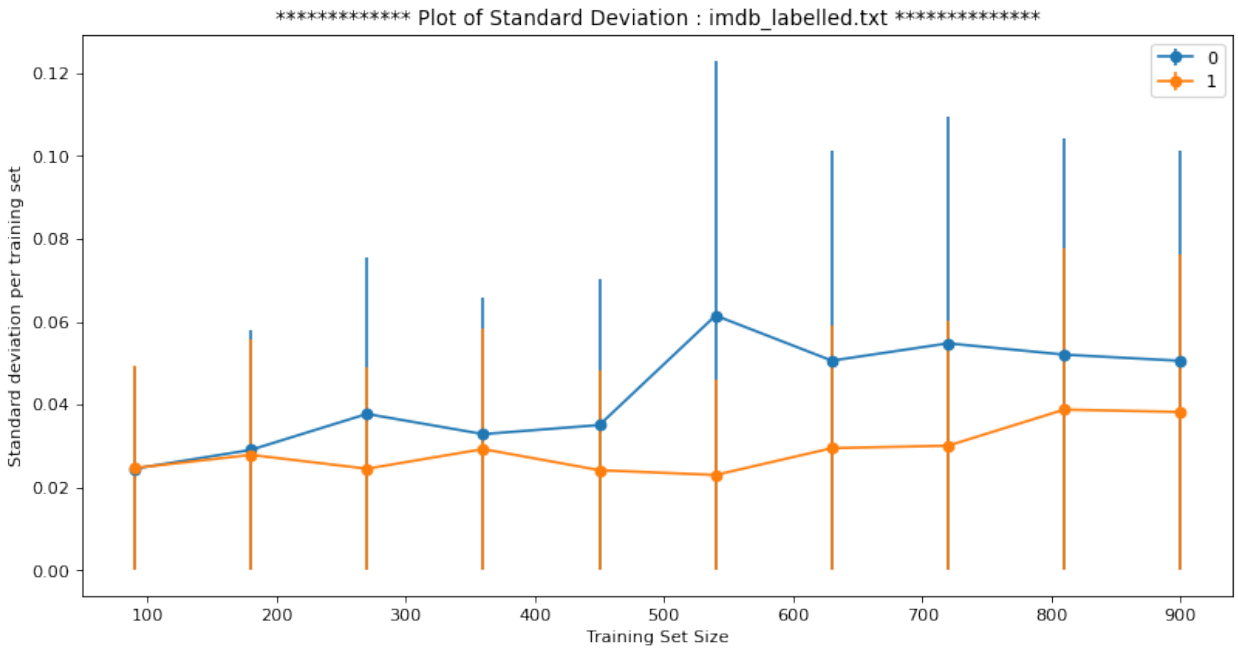
- The standard deviation with m=1 follows almost similar trend whereas with m=0, it decreases with the increasing training set size.

Data set **: *imdb_labelled.txt***

```
********** Processing the data file : imdb_labelled.txt **********

***************** RESULTS *****************

avg_accuracy for smoothing parameter m = 0: [0.5970000000000001, 0.626, 0.6359999999999999, 0.6599999999999999, 0.65
1, 0.69, 0.6799999999999999, 0.6900000000000001, 0.7090000000000001, 0.7120000000000001]

SD_ for smoothing parameter m = 0: [0.024515301344262528, 0.029051678092667926, 0.03773592452822643, 0.03286335345030
996, 0.0350570962859162, 0.061481704595757594, 0.05059644256269404, 0.05477225575051661, 0.05204805471869242, 0.05055
6898639058136]

avg_accuracy for smoothing parameter m = 1: [0.609, 0.6180000000000001, 0.6829999999999999, 0.738, 0.7440000000000000
1, 0.749, 0.7709999999999999, 0.7739999999999998, 0.806, 0.82]

SD_ for smoothing parameter m = 1: [0.024677925358506155, 0.027856776554368263, 0.024515301344262504, 0.0292574776766
55614, 0.024166091947189168, 0.02300000000000002, 0.029478805945967353, 0.030066592756745846, 0.03878143885933062, 0.
0382099463490856]

 training data points : [90, 180, 270, 360, 450, 540, 630, 720, 810, 900]
```



- Accuracy increases significantly from nearly 0.6 to 0.8 with m=1.
- Though the accuracy trend is increasing with m=0 with the training set size increment, but that slower than that of m=1.



- The standard deviation with m=1 follows almost similar trend whereas with m=0, it shoot up to 0.065 from 0.03 when training set size is 540 but decreases thereafter.

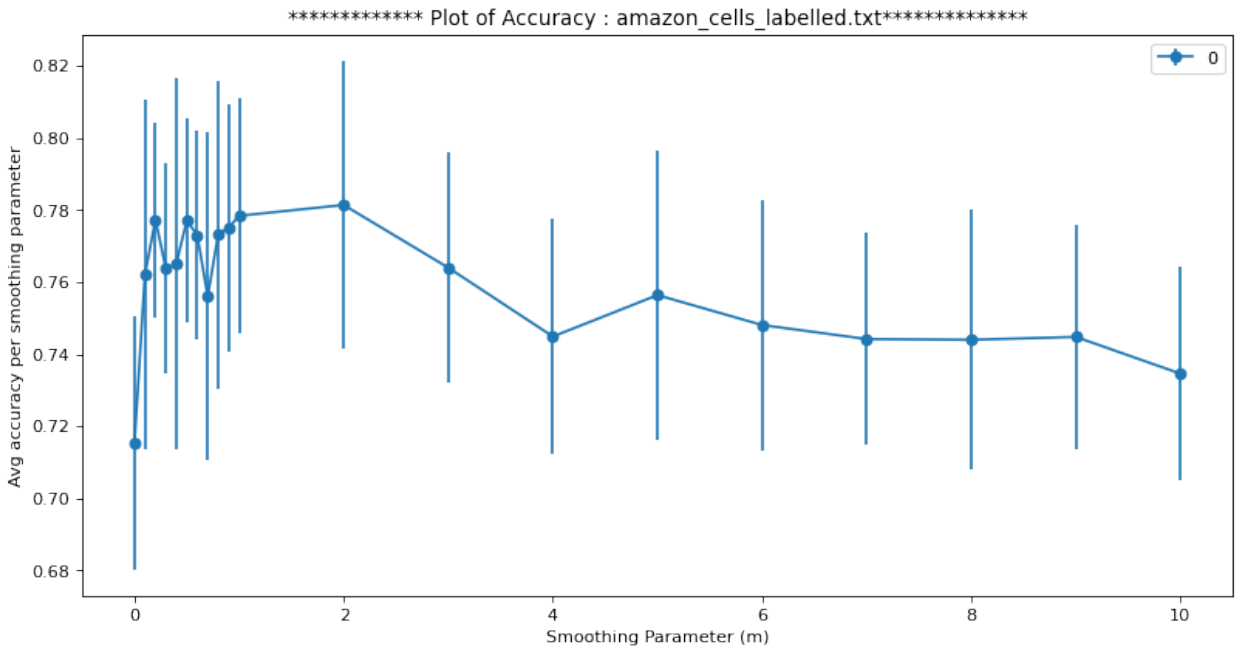
4.2. Exp 2: Run stratified cross validation for Naive Bayes with smoothing parameter m = 0, 0.1, 0.2,…0.9 and 1, 2, 3…10 (i.e., 20 values overall). Plot the cross-validation accuracy and standard deviations as a function of the smoothing parameter.

The program is executed for all the 3 datasets.

The sample outputs of average accuracy per smoothing parameters and the standard deviations of accuracy for all the smoothing parameters m = 0, 0.1, 0.2,..0.9, 1, 2…9 (20 values) are listed below. Also, the corresponding plots as a function of the training set size are displayed with the observations listed.

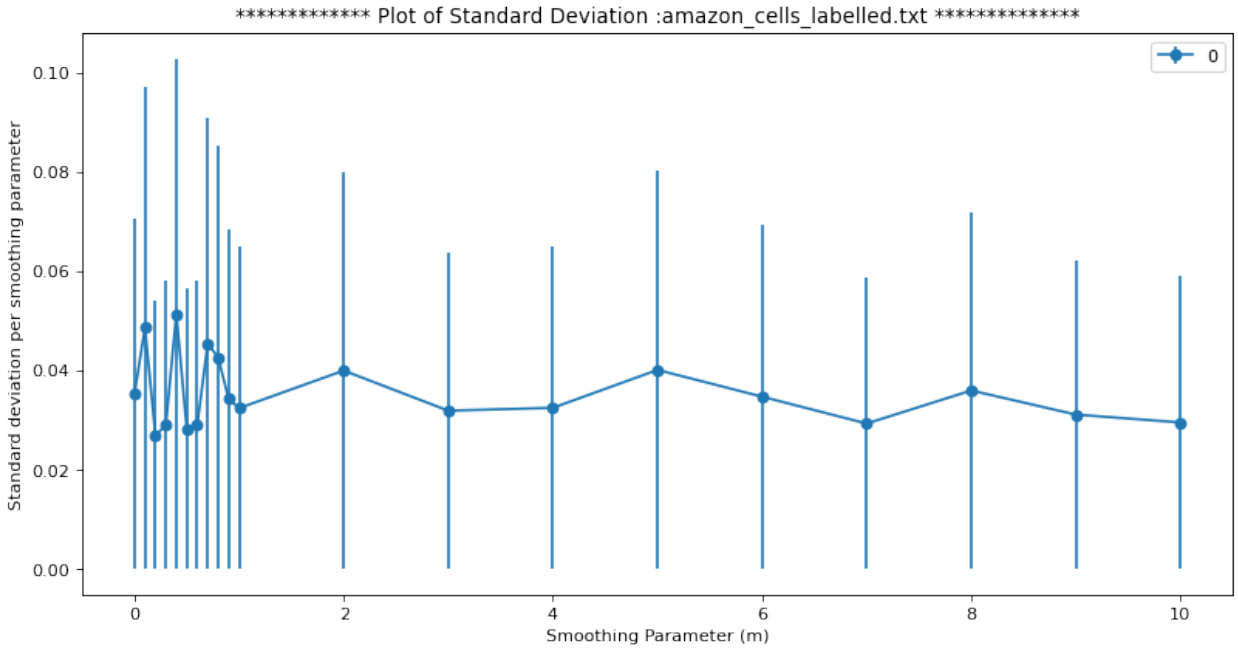
Data set **: *amazon_cells_labelled.txt***

```
********** Processing the data file : amazon_cells_labelled.txt **********

********** RESULTS **********

avg_accuracy for 20 smoothing parameter : [0.7154, 0.7621, 0.7773000000000001, 0.7638000000000001, 0.7651, 0.7771, 0.
773, 0.756, 0.7732, 0.7751, 0.7784000000000001, 0.7814, 0.764, 0.7449000000000001, 0.7564000000000001, 0.748099999999
9999, 0.7442, 0.744, 0.7447999999999999, 0.7346000000000001]

avg_SD for 20 smoothing parameter : [0.03534202151470315, 0.04865443323512538, 0.02697524926125251, 0.029047893112156
813, 0.0514089877800018, 0.028216770953965502, 0.029086867934635774, 0.045418090755168256, 0.042664750988103704, 0.03
428622320993777, 0.0324432207816562, 0.04000944060599361, 0.03192964535186488, 0.032508815446363894, 0.04013888522739
859, 0.03475552016054868, 0.029338870138215668, 0.03600029027985806, 0.03114164319852688, 0.029563082326589996]

 all smothing parameters : [0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```



We observe that, in Average accuracies Vs Smoothing parameters plot-

- Model accuracies are significantly higher with $0 < m \ll 1$.
- With m=0, the model accuracy is the lowest as it has no posterior.
- Accuracy decreases with increment in smoothing parameter ie m values.



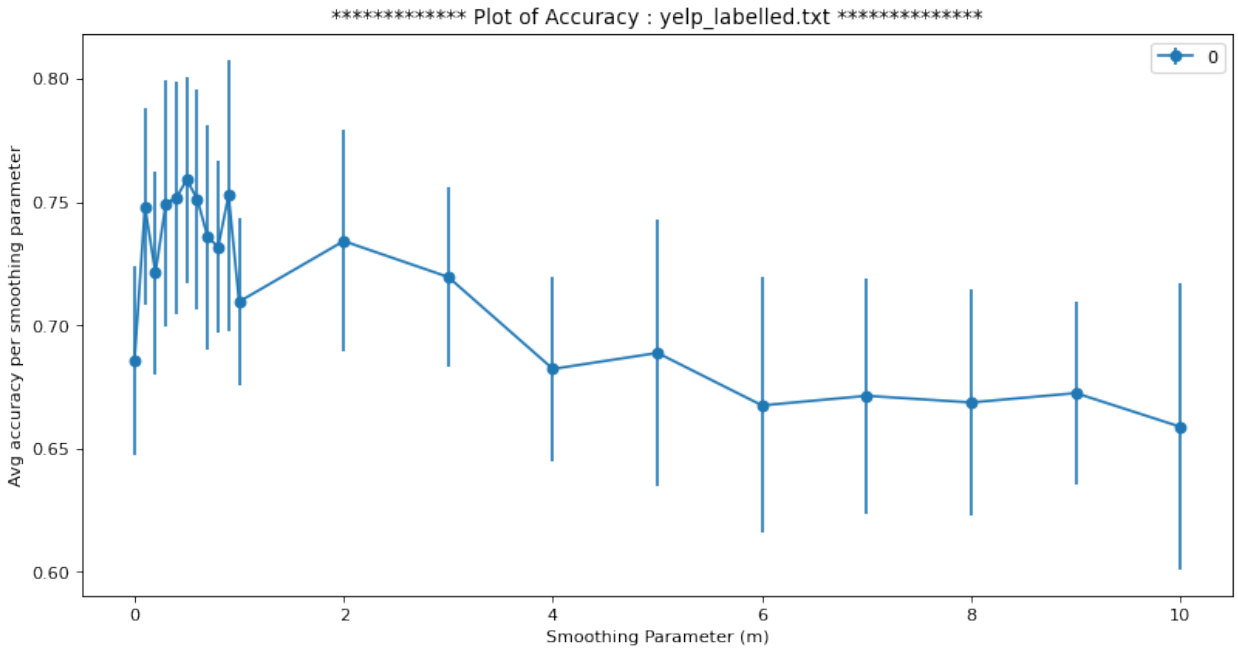
We observe that, in Standard Deviations of accuracies Vs Smoothing parameter plot-

- Standard deviation of accuracies is lowest for m=0.2
- Comparatively lower standard deviations of accuracies are seen in models with parameters between 0 and 1

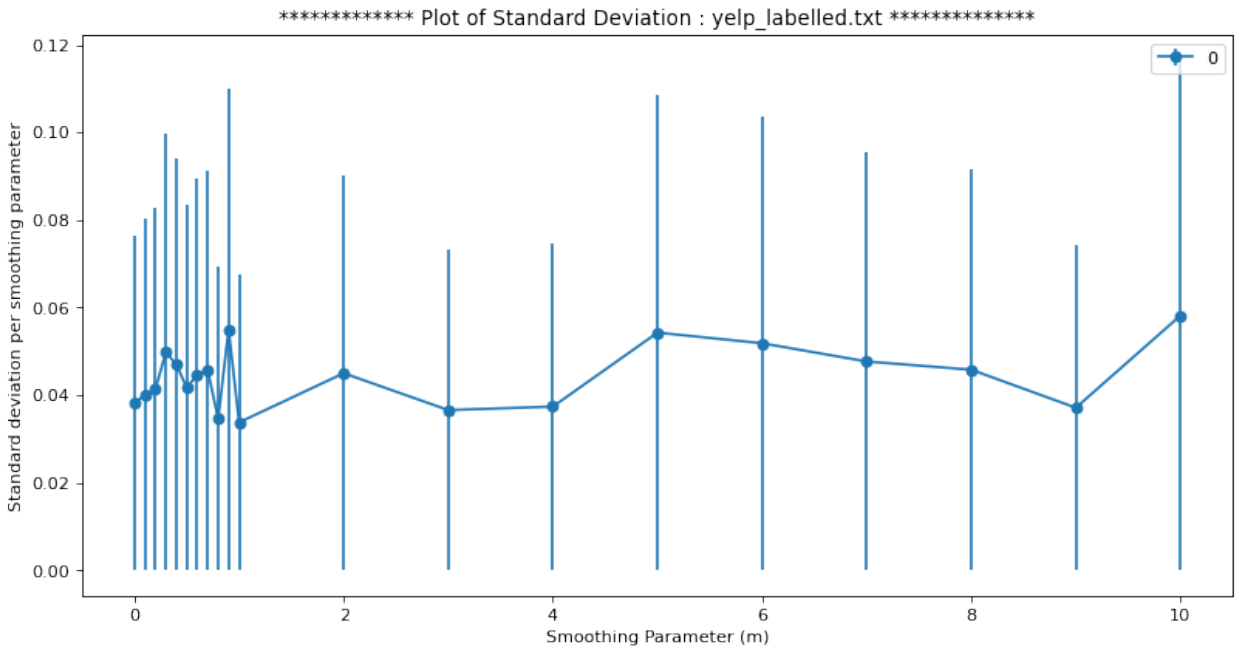
Similar Observations are followed in the other 2 datasets and the results and plots are given below.

Data set **: *yelp_labelled.txt***

```
*********** Processing the data file : yelp_labelled.txt ***********

***************** RESULTS *****************

avg_accuracy for 20 smoothing parameter : [0.6855999999999999, 0.7482, 0.7213, 0.7495, 0.7518, 0.7590999999999999, 0.
7512, 0.7358, 0.7319, 0.7528, 0.7096, 0.7342, 0.7196, 0.6823, 0.6888, 0.6675000000000001, 0.6714, 0.6687000000000001,
0.6725, 0.6588]

avg_SD for 20 smoothing parameter : [0.03821117962218975, 0.040094325460275734, 0.04139814489083863, 0.04979811324095
066, 0.046977334724472616, 0.04167846107326066, 0.0446874164538093, 0.045700628911115995, 0.0346012897990867, 0.05496
3156176592566, 0.03376765509099079, 0.0450030229649126, 0.036583801354119504, 0.0373803757228278, 0.0542715912305714
2, 0.051846110359770624, 0.047682171817024915, 0.04580169533908714, 0.0371190279992422, 0.05817663315003725]

 all smothing parameters : [0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```



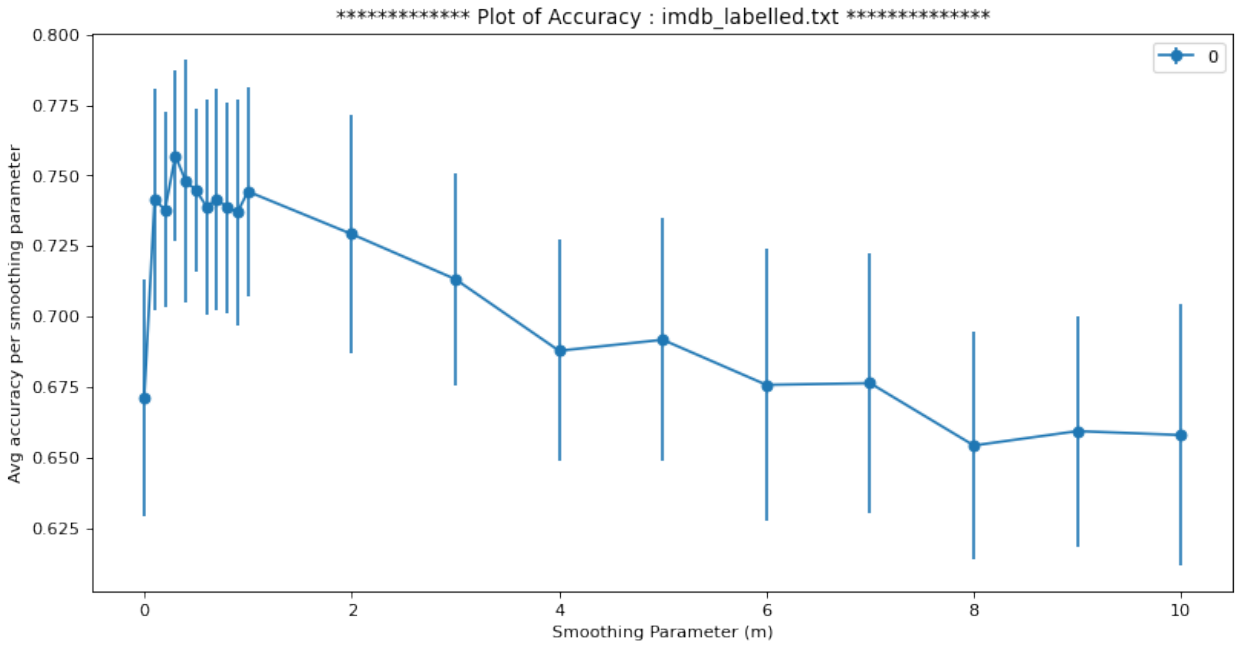
- Accuracies decreases as smoothing parameters increases from 1. It follows the highest accuracies with m between 0 and 1.



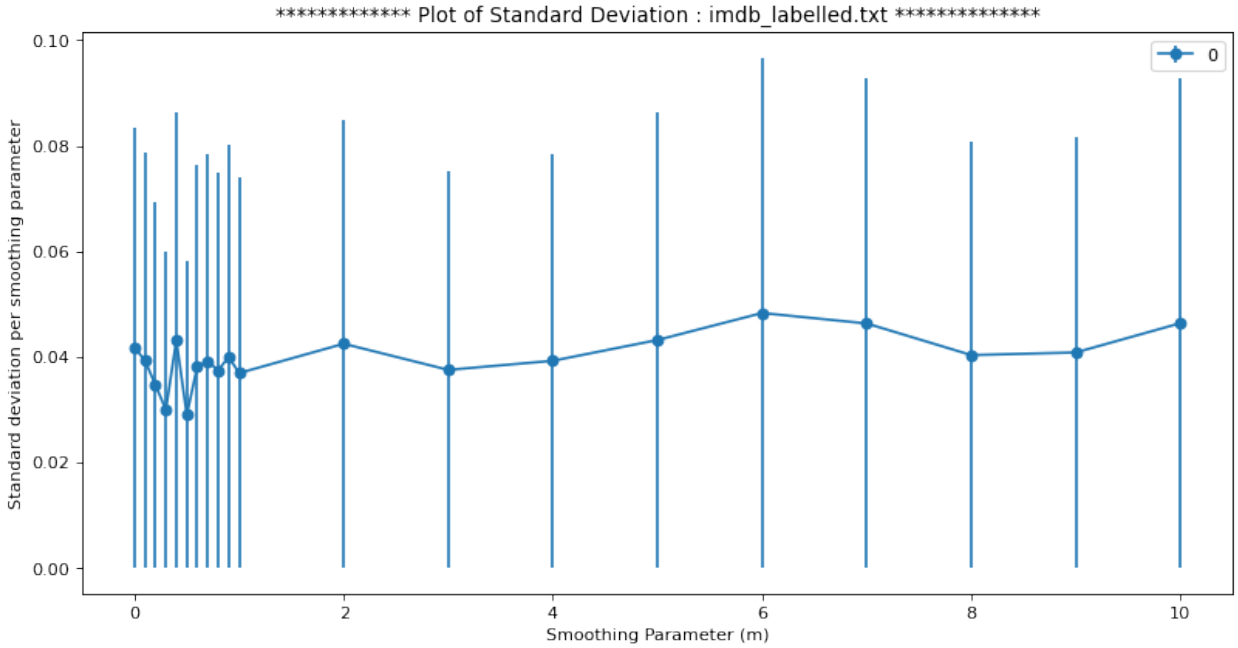
- Standard Deviations are higher for higher m values and lowest between 0<<m<<1.

Data set **: *imdb_labelled.txt***

```
********** Processing the data file : imdb_labelled.txt **********

***************** RESULTS *****************

avg_accuracy for 20 smoothing parameter : [0.6711, 0.7414000000000001, 0.7378000000000001, 0.757, 0.748, 0.7447, 0.73
90000000000001, 0.7414999999999999, 0.7386000000000001, 0.737, 0.7443000000000001, 0.7292, 0.7132, 0.6878, 0.6917, 0.
6757000000000001, 0.6763, 0.6542000000000001, 0.6593, 0.6578999999999999]

avg_SD for 20 smoothing parameter : [0.04176680621712815, 0.03933093078177694, 0.03470475476879882, 0.030013365371145
222, 0.043181385771552755, 0.029028219390188188, 0.038173931899825556, 0.03924472098490827, 0.037430643017761715, 0.0
400217198489294, 0.03694506117845514, 0.042498992370236556, 0.03755252864877298, 0.039277895146115176, 0.043212505941
612266, 0.048324900566029116, 0.04634605795962006, 0.040352475054334945, 0.04085381301808634, 0.046404883107772764]

 all smothing parameters : [0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```



- Accuracies decreases as smoothing parameters increases from 1. It follows the highest accuracies with m between 0 and 1. But at m=0, the model has its lowest accuracy.



- Models with parameters between 0 and 1 have lesser standard deviation compared to other models with parameters greater than 1.