

Project Report

Logic Used:

1. Divide the available dataset into Train and Test Data set
2. Taking the train dataset, divide/classify each document into positive or negative class based on the label provided.
3. After classifying, cleanse the data for any special symbols like [!,@,#\$%^&*()] etc. and remove the labels from the end of each document.
4. Now split the sentences into individual words and store in an array.
5. Now create a unique vocabulary set from the documents from both the classes.
6. Now count the occurrences of each word in the vocabulary set and store it in an array.
7. Now calculate the probability of each word given the class (positive or negative)
8. After calculating the probabilities of each word according to its class, store the word and its probability in a dictionary based on class.
9. Now take the test data and for each document in test data, find the probability of each word of the document from the dictionary created above and take the logarithmic sum of every word's probability from each class' dictionary.
10. Compare the sum of the probability from the positive and negative class. If the sum of the positive class is greater than that of the negative class, the document will be classified as positive, else negative.
11. After classification of documents, compare the predicted label with the actual given labels and calculate the accuracy of the prediction.

A) Experiment 1:

For each of the 3 datasets run stratified cross validation to generate learning curves for Naive 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. It is insightful to put both $m = 0$ and $m = 1$ together in the same plot. What observations can you make about the results?

Dataset = **Amazon**

1. Value of m used = 1
Prediction accuracy = 66%

[illegible]

2. Value of m used = 0
Prediction accuracy = 57.49%

```
Run: main
"C:\Users\Abhishek Sharma\PycharmProjects\firstproject\venv\Scripts\python.exe" "C:/Users/Abhishek Sharma/PycharmProjects/firstproject/main.py"
Value of "m" used: 0
Value of "m" used: 0
Predicted Labels : ['1', '1', '1', '1', '1', '1', '0', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '0', '1', '0', '0', '0', '0']
Actual Labels : ['1', '1', '1', '1', '0', '1', '0', '0', '1', '1', '1', '0', '1', '1', '1', '0', '0', '1', '1', '1', '1', '1', '1', '1', '0', '1', '0', '1', '0', '1', '0', '0', '0', '0']
Accuracy is: 57.49999999999999%

Process finished with exit code 0
```

Dataset = YELP

1. Value of m used = 1
Prediction accuracy = 32%

```
"C:\Users\Abhishek Sharma\PycharmProjects\firstproject\venv\Scripts\python.exe" "C:/Users/Abhishek Sharma/PycharmProjects/firstproject/main.py"
```

```
Value of "m" used: 1  
Value of "m" used: 1  
Predicted Labels : ['1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '0', '0', '1', '1', '0', '1', '1', '1', '1', '1', '1', '1',  
Actual Labels : ['0', '1', '0', '1', '1', '1', '0', '1', '0', '1', '0', '0', '0', '0', '0', '0', '1', '1', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0']  
Accuracy is: 32.0%
```

```
Process finished with exit code 0
```

2. Value of m used = 0
Prediction accuracy = 33.5%

[illegible]

Dataset = IMDB

1. Value of m used = 1
Prediction accuracy = 55.5%

[illegible]

2. Value of m used = 0
Prediction accuracy = 50%

The screenshot shows the PyCharm Run window with the following output:

```
Run: main x
"C:\Users\Abhishek Sharma\PycharmProjects\firstproject\venv\Scripts\python.exe" "C:/Users/Abhishek Sharma/PycharmProjects/firstproject/main.py"
Value of "m" used: 0
Value of "m" used: 0
Predicted Labels : ['0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '1', '0', '0', '0', '0', '0', '0', '0', '1', '0', '0', '0', '0', '0', '0', '0', '0', '0', '1', '0', '0', '0']
Actual Labels : ['0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1']
Accuracy is: 98.0%

Process finished with exit code 0
```

Observations and Conclusions:


1. In Amazon and IMDB datasets, the prediction accuracy increased with increasing value of m . This was however not true for the YELP dataset, wherein the accuracy increased slightly when m was decreased.
2. The effect of the value of smoothing parameter ' m ' could *perhaps* be dependent on the size of the dataset where in large datasets (like Amazon, IMDB), increase in m results in higher accuracy whereas in smaller datasets (like Yelp), increasing m results in lower accuracy.
3. Also it was observed that the overall accuracy depends on the size of the datasets. For example, the accuracy in case of Amazon and IMDB datasets (which are of larger size) was above 50% (even 66% in case of Amazon) however, the accuracy in YELP, being the smaller dataset of the three was only about 33%.

Experiment 2:

Run stratified cross validation for Naive Bayes with smoothing parameter $m = 0, 0.1, 0.2, \dots, 0.9$ and $1, 2, 3, \dots, 10$ (i.e., 20 values overall). Plot the cross validation accuracy and standard deviations as a function of the smoothing parameter. What observations can you make about the results?

Dataset being used: AMAZON

1. Value of $m = 0.1$
Accuracy = 57.9%



Run main

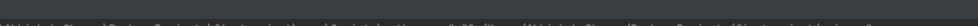
```

"C:\Users\Abhishek Sharma\PycharmProjects\firstproject\venv\Scripts\python.exe" "C:\Users\Abhishek Sharma\PycharmProjects\firstproject\main.py"
Value of "m" used: 0.1
Value of "m" used: 0.1
Predicted Labels : ['1', '1', '1', '1', '1', '1', '0', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '1', '0', '1', '1', '1', '1', '1', '1', '1', '0', '1', '0', '1', '0', '0']
Actual Labels : ['1', '1', '1', '1', '0', '1', '0', '0', '1', '1', '1', '1', '0', '1', '1', '1', '1', '0', '1', '1', '1', '1', '1', '1', '1', '1', '1', '0', '1', '0', '1', '0', '1', '0', '0']
Accuracy is: 57.99999999999999%

Process finished with exit code 0

```

2. Value of $m = 0.2$
Accuracy = 60%



```

Run: main
"C:\Users\Abhishek Sharma\PycharmProjects\firstproject\venv\Scripts\python.exe" "C:/Users/Abhishek Sharma/PycharmProjects/firstproject/main.py"
Value of "m" used: 0.2
Value of "m" used: 0.2
Predicted Labels : ['1', '1', '1', '1', '1', '0', '1', '1', '1', '1', '0', '1', '1', '1', '1', '1', '1', '0', '1', '1', '1', '1', '1', '1', '0', '1', '0', '1', '0', '0', '0']
Actual Labels : ['1', '1', '1', '0', '1', '0', '0', '1', '1', '1', '0', '1', '1', '1', '0', '0', '1', '1', '0', '1', '1', '1', '1', '1', '0', '1', '0', '1', '0', '0', '0']
Accuracy is: 60.0%

Process finished with exit code 0

```

3. Value of $m = 0.3$
Accuracy = 62.5%

```

Run: main x
"C:\Users\Abhishek Sharma\PycharmProjects\firstproject\venv\Scripts\python.exe" "C:/Users/Abhishek Sharma/PycharmProjects/firstproject/main.py"
Value of "m" used: 0.3
Value of "m" used: 0.3
Predicted Labels : ['1', '1', '1', '1', '1', '0', '1', '1', '1', '1', '0', '1', '1', '1', '1', '1', '1', '0', '1', '1', '1', '1', '1', '1', '0', '1', '1', '0', '1', '0', '1', '0', '0']
Actual Labels : ['1', '1', '1', '1', '0', '1', '0', '0', '1', '1', '1', '1', '0', '1', '1', '1', '1', '0', '0', '1', '1', '1', '1', '1', '1', '1', '1', '1', '0', '1', '1', '0', '1', '0', '0', '0', '0']
Accuracy is: 62.5%

Process finished with exit code 0

```

4. Value of $m = 0.4$
Accuracy = 63.5%

[illegible]

5. Value of $m = 0.5$
Accuracy = 64%

[illegible]

6. Value of $m = 0.6$
Accuracy = 64.5%

[illegible]

7. Value of $m = 0.7$
Accuracy = 64.5%

The screenshot shows the PyCharm interface with the Run console open. The terminal output displays the following information:

```
"C:\Users\Abhishek Sharma\PycharmProjects\firstproject\venv\Scripts\python.exe" "C:/Users/Abhishek Sharma/PycharmProjects/firstproject/main.py"  
Value of "a" used: 0.7  
Value of "a" used: 0.7  
Predicted Labels : [['1', '1', '1', '1', '1', '0', '1', '1', '1', '1', '0', '1', '1', '1', '1', '1', '1', '0', '0', '1', '1', '1', '1', '1', '1', '0', '1', '0', '1', '0', '0',  
Actual Labels : ['1', '1', '1', '1', '0', '1', '0', '0', '1', '1', '1', '0', '1', '1', '1', '0', '0', '1', '1', '0', '1', '1', '1', '1', '1', '0', '1', '0', '1', '0', '1', '0', '0',  
Accuracy is: 64.5%  
  
Process finished with exit code 0
```

8. Value of $m = 0.8$
Accuracy = 65.5%

The screenshot shows the PyCharm Run console output for a script named main.py. The output displays the value of 'm' as 0.8, predicted labels, actual labels, and an accuracy of 65.5%. The process finished with exit code 0.

```
Run: main
"C:\Users\Abhishek Sharma\PchamProjects\firstproject\venv\Scripts\python.exe" "C:/Users/Abhishek Sharma/PchamProjects/firstproject/main.py"
Value of "m" used: 0.8
Value of "n" used: 0.8
Predicted Labels : ['1', '1', '1', '1', '1', '0', '1', '1', '1', '1', '0', '1', '1', '1', '1', '1', '1', '0', '0', '1', '1', '1', '1', '1', '1', '1', '0', '1', '0', '1', '0', '0']
Actual Labels : ['1', '1', '1', '0', '1', '0', '0', '1', '1', '1', '0', '1', '1', '1', '0', '0', '1', '1', '1', '1', '1', '1', '1', '1', '0', '1', '0', '1', '0', '1', '0', '0']
Accuracy is: 65.5%

Process finished with exit code 0
```

9. Value of $m = 0.9$
Accuracy = 65.5%

Also, it is observed that the higher the size of Dataset, the higher is the prediction accuracy.

Plot of accuracy Average vs Std Deviation:

