BITS F464 Machine Learning

Naive Bayes Classifier

1. Introduction:

- In order to categorize objects based on certain features, we use certain classifiers. Naïve Bayes is one such classifier which is a probabilistic classifier basing upon the Bayes Theorem. Usually naïve bayes classifiers are used for sentimental analysis. In this report we will be using the naïve bayes classifier to categorize the data.

2. Data Preprocessing:

- "We first shuffle the input data"
- First the non-important characters such as 'tab', 'new line' etc. are dealt with. For example, the tab is **replaced with a space** and a new line is **replaced with an empty string**.
- Then the entire sentence or one line in the dataset is **converted to lowercase**. This is because we should not differentiate between the words such as 'Or' and 'or'.
- After that we split the sentence by spaces. After splitting **we replace any special characters or numbers** with an empty string. We take care not to replace the last character which is meant to signify the sentiment of the sentence by 0 or 1.

```
['painful', 'on', 'the', 'ear', '0']
['lasted', 'one', 'day', 'and', 'then', 'blew', 'up', '0']
```

3. Training and Testing:

- While training the data, we keep two things in the loop. One, the number of times a word has come in a sentiment-0 sentence and the number of times it has come in a sentiment-1 sentence. The other being the total number of words in sentiment-0 and sentiment-1 separately. The latter is used in Laplace smoothing.
- **Laplace smoothing**: This is useful where a sentence contains a word that has not ever occurred in a positive sentence but other words greatly signify that it is a positive sentence. In other words, if we do not use Laplace smoothing the classifier will detect this type of sentence as negative because the probability of positive will be 0 (since it contains a word whose probability of occurring in a positive sentence is 0). This is where Laplace smoothing helps by **shifting the probability** of each word by some units. **Laplace smoothing improved the average score by around 10 %.**

```
'out': {0: 12, 1: 3},
'outgoing': {1: 1},
'outlet': {0: 2, 1: 1},
'outperform': {1: 1},
'outside': {1: 1},
'over': {0: 5, 1: 3},
```

- Finally we classify a test sentence as a positive if the probability of it being positive is greater than it being negative and vice versa.

4. Results:

```
PS C:\Users\kaila\Documents\ML-Assignments\NB> python .\nb.py
0.76
0.83
0.785
0.815
0.795
```

So, from above figure, it is clear that the accuracies are: 76.0, 83.0, 78.5, 81.5, 79.5
The mean of the accuracies is: 79.7
The SD of the accuracies is: 2.42
Therefore the **Accuracy** can be reported as **79.7 ± 2.42**
and **F-score** is **0.81 ± 0.017**