SPAM DETECTION USING NAÏVE BAYE'S ALGORITHM

Submitted by:

Shriya: 110168559

Mahathi: 110162023

- Algorithm

  Training Phase:

  - Step 1 : Read the training data file
  - Step 2 : Parse the file
  - Step 3: Look for domain specific features
  - Step 4: Note down the number of occurrences of these features in ham and spam.

  Testing Phase:

  - Step1: Read the test data file
  - Step2 : Look for domain specific features occurrence
  - Step 3: Compare the frequency of each feature against our test data frequency of spam and ham
  - Step4: If any of the frequency value is zero, Apply smoothing techniques
  - Step5: Calculate the spam probability (of a specific feature/word) using Naïve Baye's algorithm.

    $$\Pr(S|W) = \frac{\Pr(W|S)}{\Pr(W|S) + \Pr(W|H)}$$

  - Step 6: Similarly calculate the probability of the feature being ham(Pr(H/W)).
  - Step 7: Repeat Step 5 and Step 6 for all features of a test data
  - Step 8: Combine the individual probabilities. The formula for computing the combined probability is :

    $$p = \frac{1}{1 + e^{\eta}}$$

    Where :

    $$\eta = \sum_{i=1}^{N} \left[\ln(1 - p_i) - \ln p_i\right] .$$

  - Step 9: Mark the test data as spam if the probability of Pr(S/W) is higher than Pr(H/W)
  - Step 10 : End

Smoothing Techniques Used:

- **Laplace smoothing/ Additive Smoothing:**
  - If the frequency of a feature is zero, we add +1 to that feature and +1 to all the observed frequencies of that feature.

- **Weighted Smoothing**:
  - We assign some weight to the feature as follows:

$$\Pr'(S|W) = \frac{s \cdot \Pr(S) + n \cdot \Pr(S|W)}{s + n}$$

- where:
- $\Pr'(S|W)$ is the corrected probability for the message to be spam, knowing that it contains a given word ;
- s is the *strength* we give to background information about incoming spam ;
- $\Pr(S)$ is the probability of any incoming message to be spam ;
- n is the number of occurrences of this word during the learning phase ;
- $\Pr(S|W)$ is the spamicity of this word.

- **Domain specific features :**

  The Sahami's paper, talks about the following domain specific features that can improve the efficiency of categorizing a word as ham or spam.

  - Number of special characters occurring in spam is high
  - Junk email does not contain attachments
  - Time of spam email is mostly during the night
  - Domain such as .edu can never be a junk mail
  - More of capital letters and things related to "FREE" money is mostly spam

  In our program, we have taken the features of special characters occurring in spam.

- **Output :**
  - **No Smoothing**

    ```
    C:\Python34\python.exe NaiveBayesSpam.py 0 C:/ train C:/test
    Accurancy Percentage :90
    ```

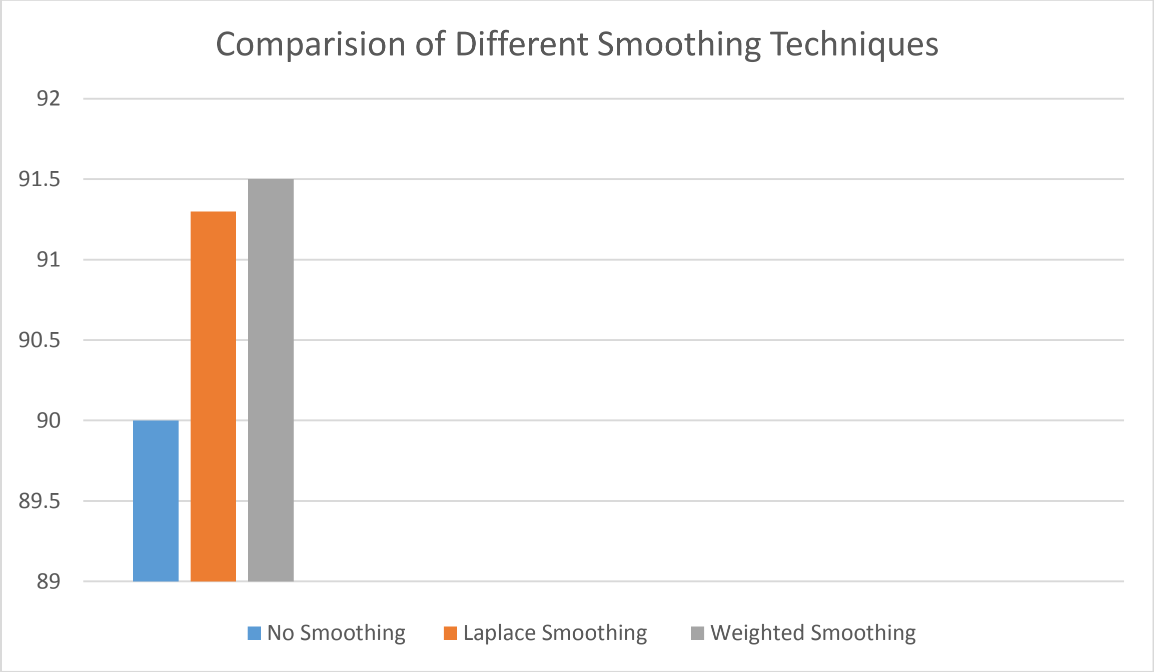
  - **Laplace Smoothing**

    ```
    C:\Python34\python.exe NaiveBayesSpam.py 1 C:/ train C:/test
    Accurancy Percentage :91.3
    ```

  - **Weighted Smoothing:**

    ```
     C:\Python34\python.exe NaiveBayesSpam.py 2 C:/ train C:/test
    Accurancy Percentage :91.5
    ```

- Comparision:


Comparision of Different Smoothing Techniques
No Smoothing
Laplace Smoothing
Weighted Smoothing