Concordia University

Engineering and Computer Science

COMP 6721 Report

# Spam Detector using Naïve Bayes Approach

Date: April 26, 2020

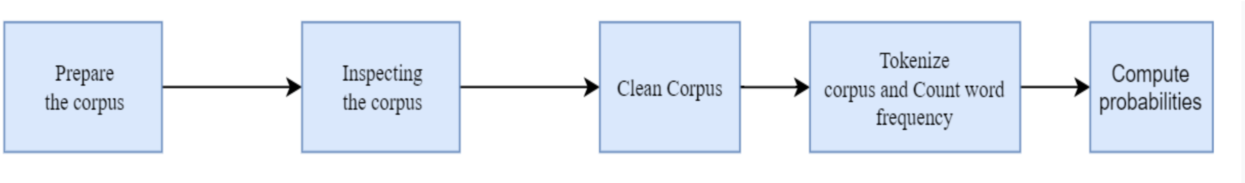
Submitted To: Dr. René Witte, P.Eng. (Associate Professor)

Team : HA-G05
Khyatibahen Chaudhary (40071098)
Jenny Mistry (40092281)
Himen Hitesh Sidhpura (40091993)

## Analysis

In order to calculate the accuracy, precision, recall and F1-measure for Spam & Ham Class as well as a confusion matrix, series of steps performed to generate classification result such as Common Aspect of Text Mining and Naïve Bayes Classifier Approach.

1. Common Aspect of Text Mining:

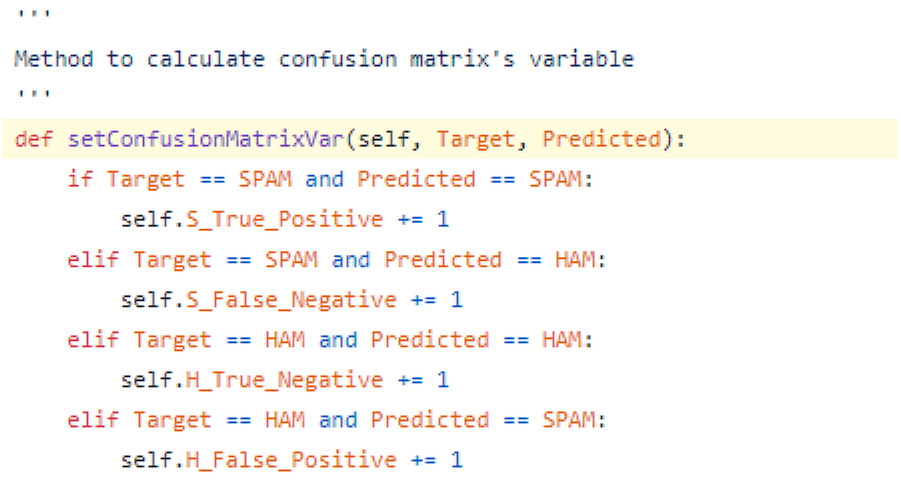


2. Naïve Bayes Classifier Approach:

Step 1: Build the Vocabulary of words by separating Spam and Ham from training Data.
Step 2: Store Vocabulary of words in a file.
Step 3: Train Classifier on Vocabulary.
Step 4: Evaluate Performance on Test data.
Step 5: Display Confusion and Evaluation Matrix

**Confusion Matrix:**

Once classifier is trained on Vocabulary, performance is evaluated on Test document. During evaluation, documents is predicted as SPAM or HAM class. To predict document as SPAM or HAM, predict method of Model file is executed. In predict method, HAM and SPAM score is calculated. Based on Score, document is predicted as SPAM or HAM. If HAM score is greater than SPAM score, document is classified as HAM otherwise, SPAM. Once document is predicted, Confusion matrix values are set. Below Figure shows various condition used to set confusion matrix values.



```
'''
Method to calculate confusion matrix's variable
'''
def setConfusionMatrixVar(self, Target, Predicted):
    if Target == SPAM and Predicted == SPAM:
        self.S_True_Positive += 1
    elif Target == SPAM and Predicted == HAM:
        self.S_False_Negative += 1
    elif Target == HAM and Predicted == HAM:
        self.H_True_Negative += 1
    elif Target == HAM and Predicted == SPAM:
        self.H_False_Positive += 1
```

We considered SPAM as a positive class and HAM as the negative class,

| | SPAM (Predicted) | HAM (Predicted) |
|---|---|---|
| SPAM (Actual) | 336 | 64 |
| HAM (Actual) | 6 | 394 |

**Evaluation Matrix:**

- **Accuracy**: $\frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$
- **Precision**: $\frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$
- **Recall**: $\frac{\mathrm{TP}}{\mathrm{TP+FN}}$
- **F1-Score**: $\frac{2 * \mathrm{Recall} * \mathrm{Precision}}{\mathrm{Recall} + \mathrm{Precision}}$

| | |
|---|---|
| **Accuracy** | 91.25 |
| **Precision** | 98.24 |
| **Recall** | 84 |
| **F1** | 90.566 |

In our model, there is high probability of getting mail in SPAM class due to very high precision. It was also found that in our model, around 16 % of a mail is predicted as HAM class instead of SPAM due to lower recall.

## References

[1] https://www3.nd.edu/~steve/computing_with_data/20_text_mining/text_mining_example.html#/
[2] https://towardsdatascience.com/spam-filtering-using-naive-bayes-98a341224038
[3] https://medium.com/coinmonks/spam-detector-using-naive-bayes-c22cc740e257
[4] https://en.wikipedia.org/wiki/Naive_Bayes_spam_filtering