**SPAMFILTER**

**Group Members**

51. Ragini Patil

57. Shubham Sahu

58. Neelabh Sarkar

**Problem Statement**

To design a spam filter which would be sufficiently able to distinguish whether any given message is spam or ham(not spam) using a Machine Learning classification tecnique called Naive Bayes Classifier which is based on Bayes’ Theorem in Probability. It follows supervised learning approach without any dimensional reduction and hence has a comparatively higher time complexity for classification of data, however any such method can be easily incorporated pre-classification.

**Naive Bayes Classification**

Classifier works by correlating the token(words) with classification sets(spam and ham) and then uses Bayes’ Theorem to calculate probability that an email is spam or not.

**Terminologies**:-

P(Spam) = Probability that given message is Spam(0.5)

P(Ham) = Probability that given message is Ham(0.5)

P(Word|Spam) = Probability that “Word” appears in Spam messages

P(Word|Ham) = Probability that “Word” appears in Ham messages

P(Spam|Word) = Probability that message is spam knowing that the word”Word” occurs in it.

True Positive = Message is Spam and filter detects it as Spam

True Negative = Message is Ham and filter detects it as Ham

False Postive = Message is Ham and filter detects it as Spam

False Negative = Message is Spam and filter detects it as Ham

The formula used by the Spam Filter to classify is derived from Bayes’ Theorem

P(Spam|Word) = \_\_\_\_\_\_\_\_\_P(Word|Spam) \* P(Spam)\_\_\_\_\_\_\_\_\_\_\_\_

P(Word|Spam) \* P(Spam) + P(Word|Ham) \*P(Ham)

P(Spam) and P(Ham) in real life scenario is around 0.8 and 0.2 respectively, however our classification has no priori reason for making this assumption and hence sets values as 0.5 for each.

The other advantage of this assumption is that any Spam message can be tolerated as Ham message, but one must not declare an important Ham message for user as Spam.

P(Word|Spam) and P(Word|Ham) can be obtained from the training data as it denotes the ratio of (Given Word Count)/(Total Word Count) in ham as well as spam.

The above formula gives us the Spamicity and Hamicity of only that word but there are multiple words which contribute to the overall classification of message as spam or ham. We can either filter out words with high Spamicity or Hamicity for quick performance or consider all words(as in our case) and compensate time performance.

Overall spamicity of message can be determined by the formula

p = \_\_\_\_\_\_p1\*p1\*p3\*..........\*pn\_\_\_\_\_\_\_

p1\*p2\*...\*pn+(1-p1)\*(1-p2)\*...(1-p3)

For better calculation by avoiding multiplication of very small floating numbers, we use the transformation

p = \_\_1\_\_

1+e^k

where k = 1-Summation-n( log(1-pn) -log(pn))

and pn is spamicity/ hamicity for that given word

For conditions where a word has some probability exactly equal 0 or 1, the modification in probability of word is made as

new P(Spam|Word) = s\*P(Spam) + n\*P(Spam|Word)

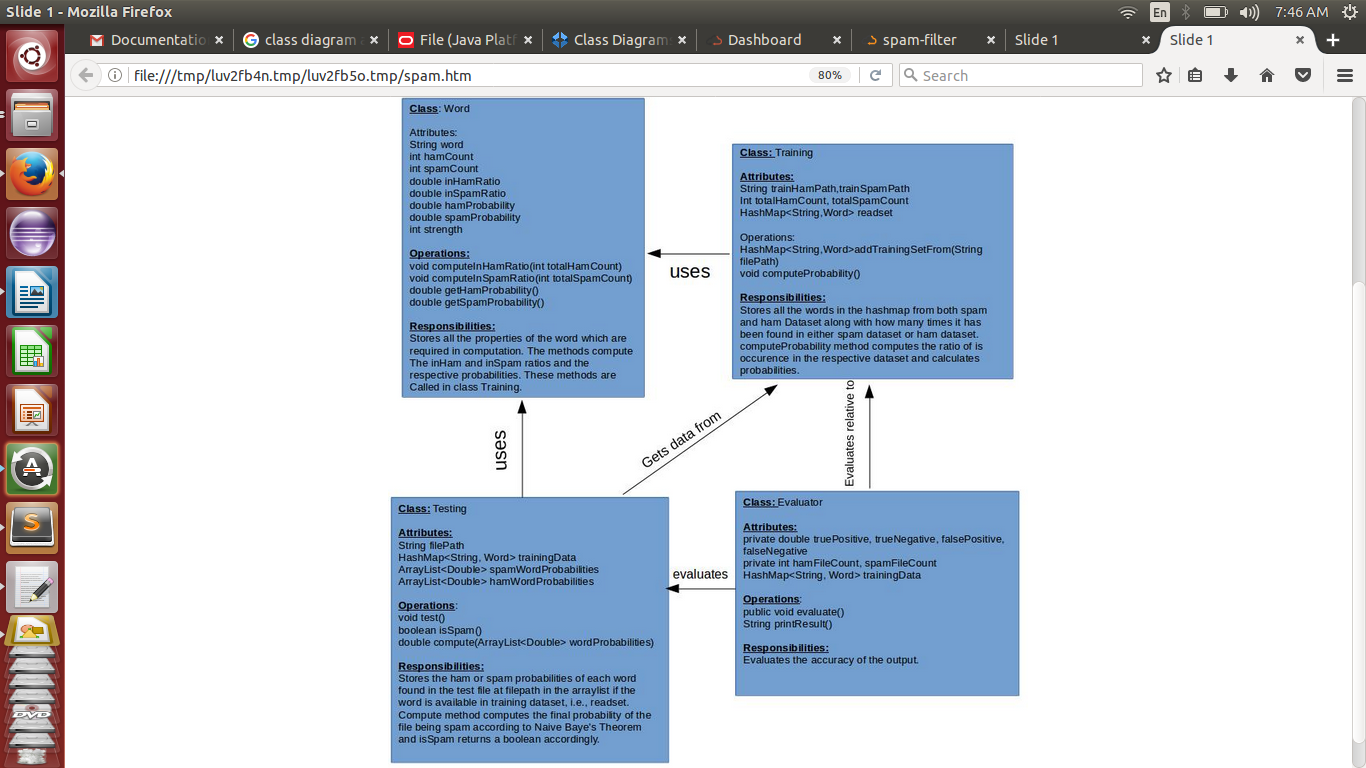
s+n

where s is strength factor, considering 3 which would mean that if word count for that message being spam is more than 3 then confidence of it being spam is more

n = word count in spam

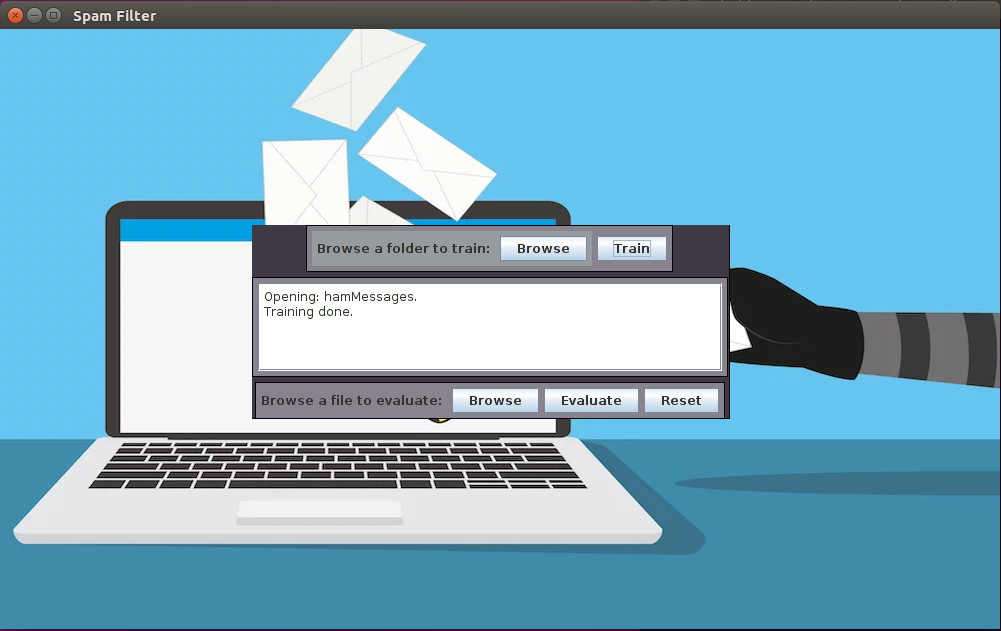
And finaly if P(Spam) > P(Ham), then message is classified as a Spam or Ham otherwise.

**Class Diagram:**

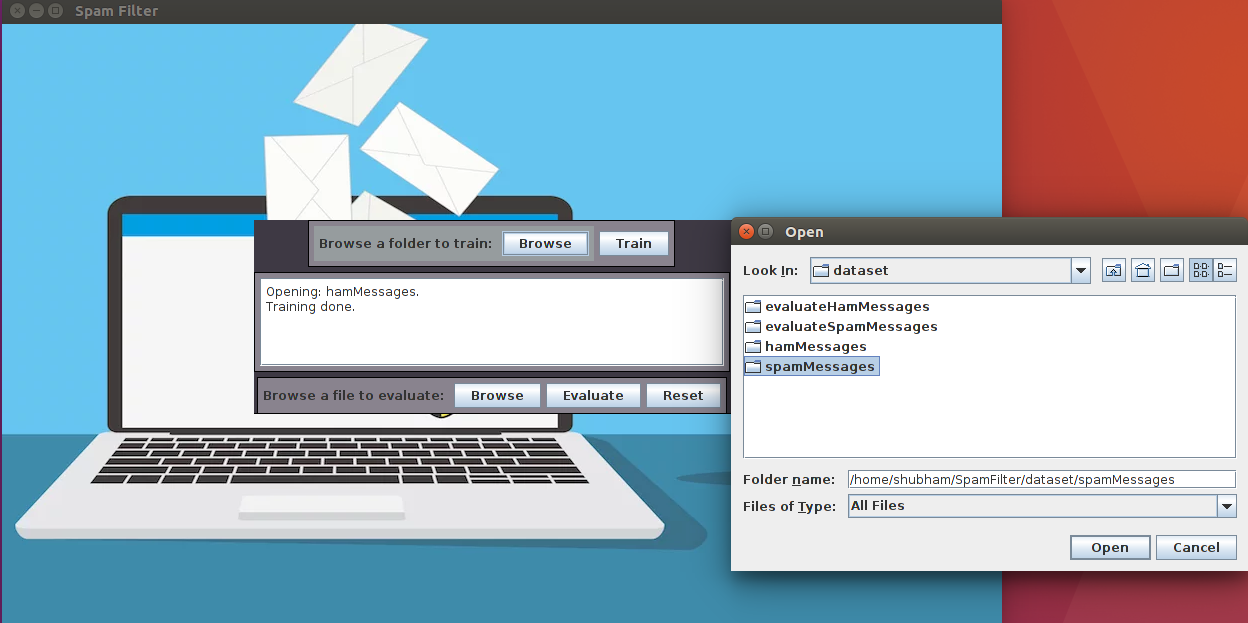
****

**Screenshots:**

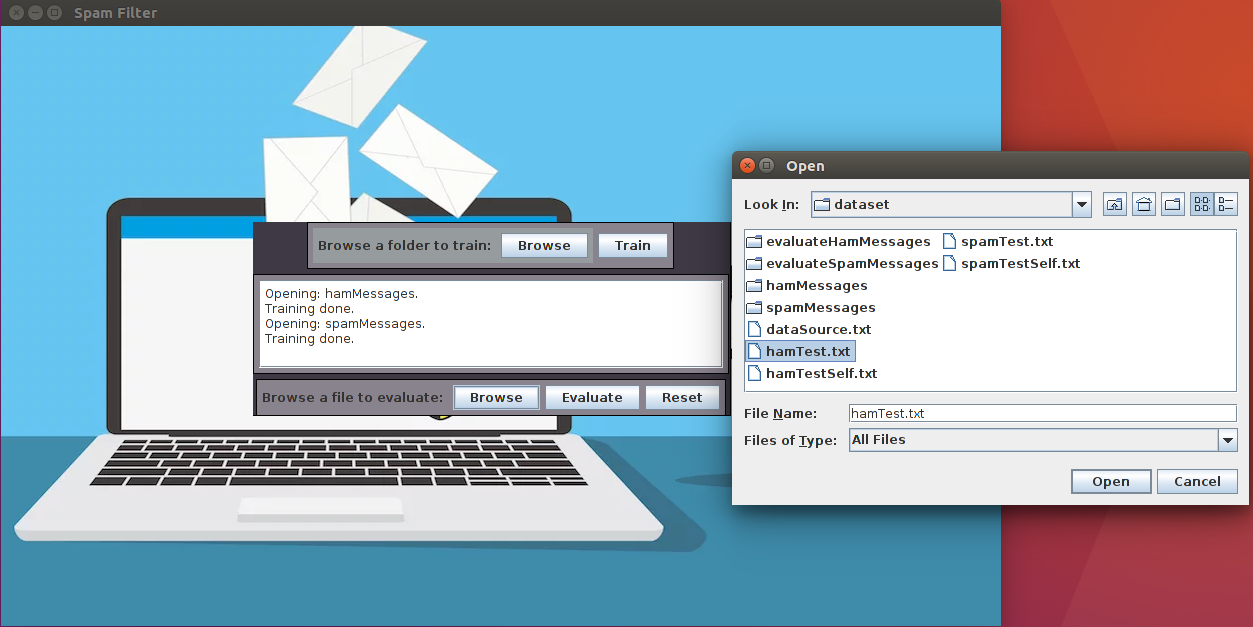
1.Training Data



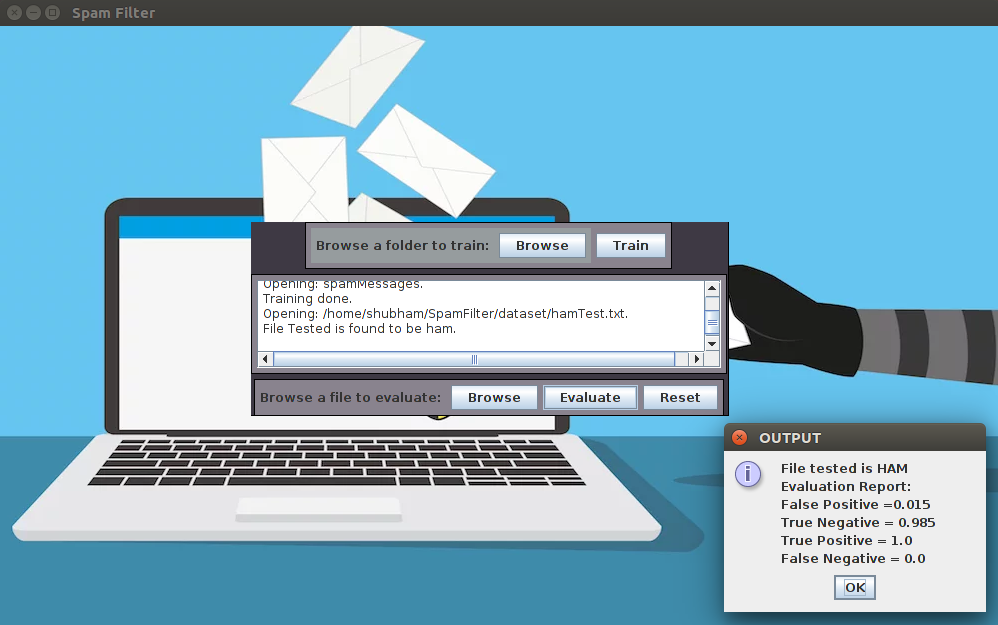
2.Selecting Training Data

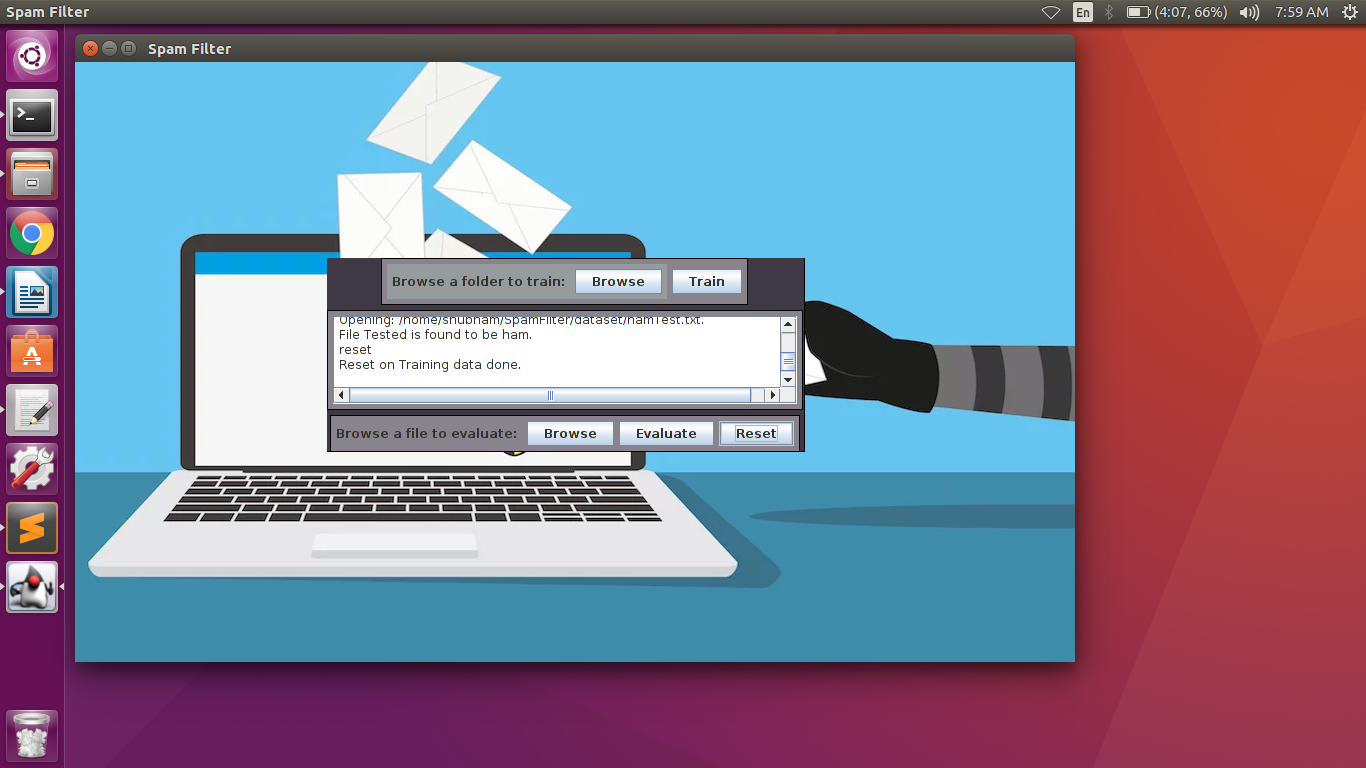


3.Selecting Test File



4.Evaluation and Result



5.Reset Training Data

**Java Features:**

1. File Chooser:

File choosers provide a GUI for navigating the file system, and then either choosing a file or directory from a list, or entering the name of a file or directory. To display a file chooser, you usually use the JFileChooserAPI to show a modal dialog containing the file chooser. Another way to present a file chooser is to add an instance ofJFileChooser to a container.

2. Option Pane

A Dialog window is an independent subwindow meant to carry temporary notice apart from the main Swing Application Window. Most Dialogs present an error message or warning to a user, but Dialogs can present images, directory trees, or just about anything compatible with the main Swing Application that manages them.

For convenience, several Swing component classes can directly instantiate and display dialogs. To create simple, standard dialogs, you use the JOptionPane class.

Using JOptionPane, you can quickly create and customize several different kinds of dialogs. JOptionPane provides support for laying out standard dialogs, providing icons, specifying the dialog title and text, and customizing the button text

showMessageDialog: Displays a modal dialog with one button, which is labeled "OK" (or the localized equivalent). You can easily specify the message, icon, and title that the dialog displays

Then we also have used the general features like JFrame,JButton, JTextArea etc.

**Future Scope:**

1. **Eliminating Neutral** words such as “a”, “the”, “is”,

2. **Word Pattern** such as “give us your password” instead of “give”, “us”, your”and “password” gives more sensitivity to context

3. **Mixed Methods** by merging Naive Bayes with some other feature selection and dimesion reduction

4. **OCR** for e-mail as words can be in the form of non-text images which remain undetected