BUILDING A SMARTER AI-POWERED SPAM CLASSIFIER

Presented by:R Vijayalakshmi

Introduction

PROBLEM:
Unsolicited commercial email, commonly known as spam, is a pressing problem on the Internet.

It undermines the usability of the email system and also costs space thus delaying in system response.

GOAL :
The goal of the project is to design and develop a spam detection system for emails by using classifiers like NaiveBayes, NaiveBayes Multinomial, KNN, SVM.

Data Preparation

Publicly available email message data from lingspam data set is used in the project.

The data is downloaded from http://www.aueb.gr/users/ion/data/lingspam_public.tar.gz

Downloaded data which contains a series of folders with each having clean and spam message need to be restructured into *clean* and *spam* folders for training and validation purpose as a part of data preparation.

A special method of Pruning is designed and implemented for the purpose of restructuring the data to be used in the project

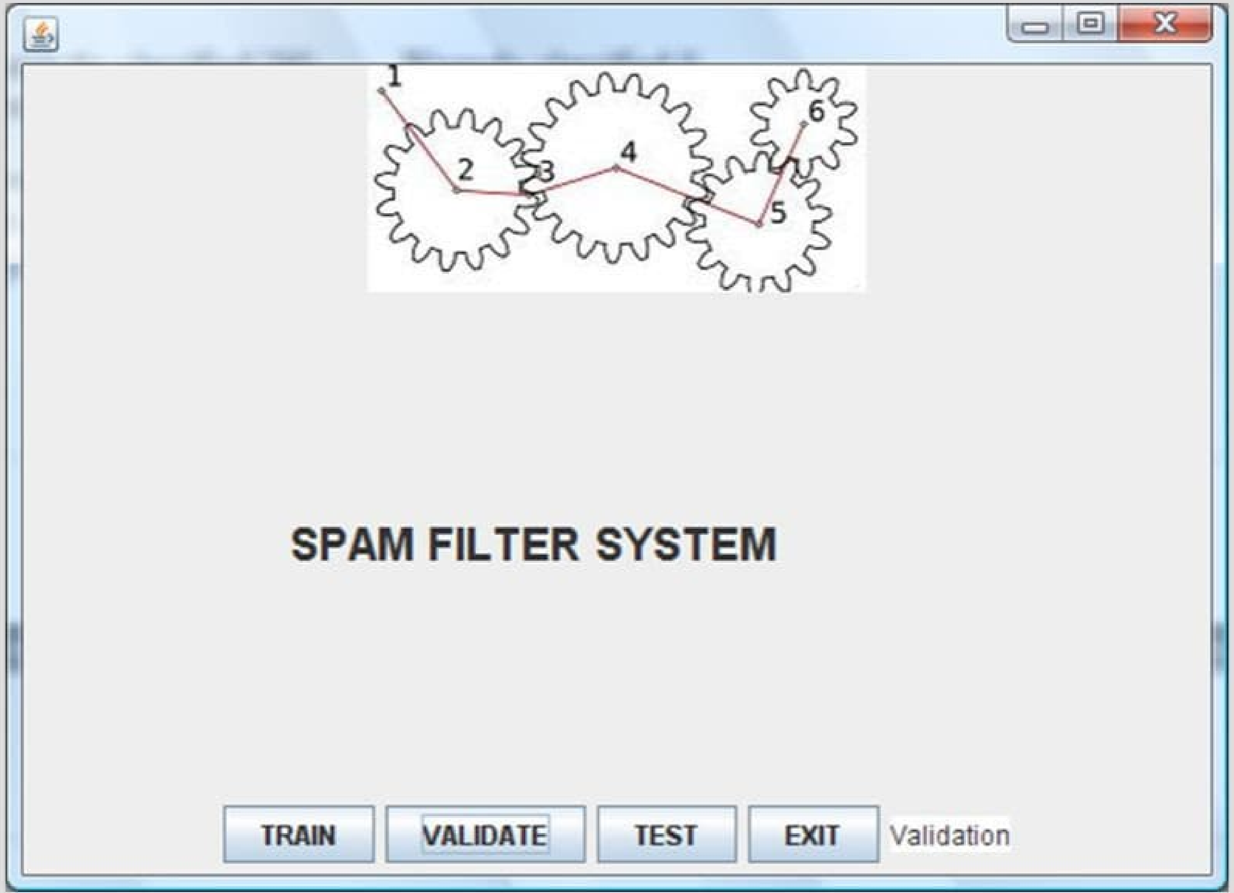
Steps involved in Spam Filtering

1) **Training:** Trains the system with two class data

2)**Validation:** In this step, we are going to validate the classifier by providing the known structured data and check how the system is classifying the data to measure its performance.

3)**Testing:** In this step, we are going to test the classifier to classify the raw email messages into two classes as 'clean' (copied into Inbox) 'spam' (copied into spam folder).

Main Screen

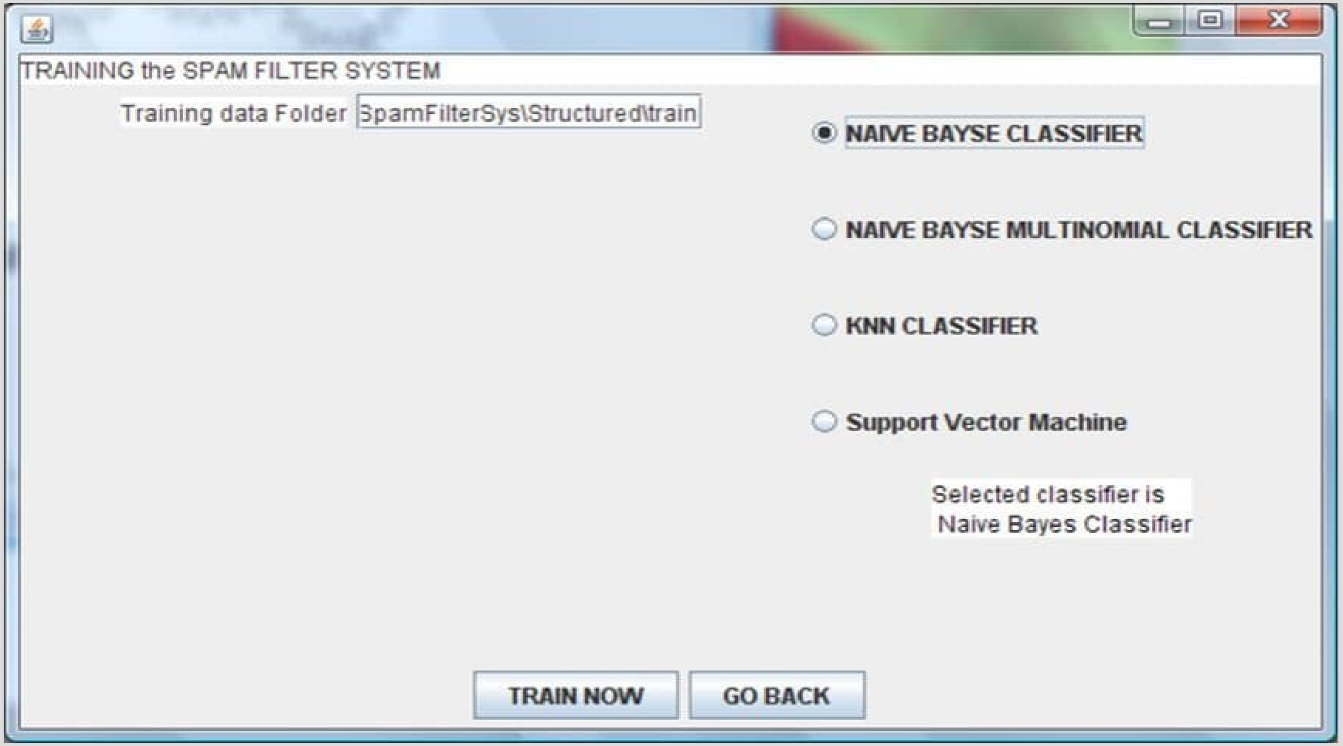


1
2
3
4
5
6
SPAM FILTER SYSTEM
TRAIN
VALIDATE
TEST
EXIT
Validation

Training

- In this phase, we are going to train the system with the data of known classes.
- I am using Part 1 to part 7 as a training data to the system.
- Input to the training step is the path of structured data which is already classified into two classes.
- Then the system builds the model using any one of the following algorithms:
    - Naïve Bayes
    - Naïve Bayes Multinomial
    - K-Nearest Neighbour
    - Support Vector Machine

Training Screen


TRAINING the SPAM FILTER SYSTEM

Training data Folder: SpamFilterSys\Structured\train

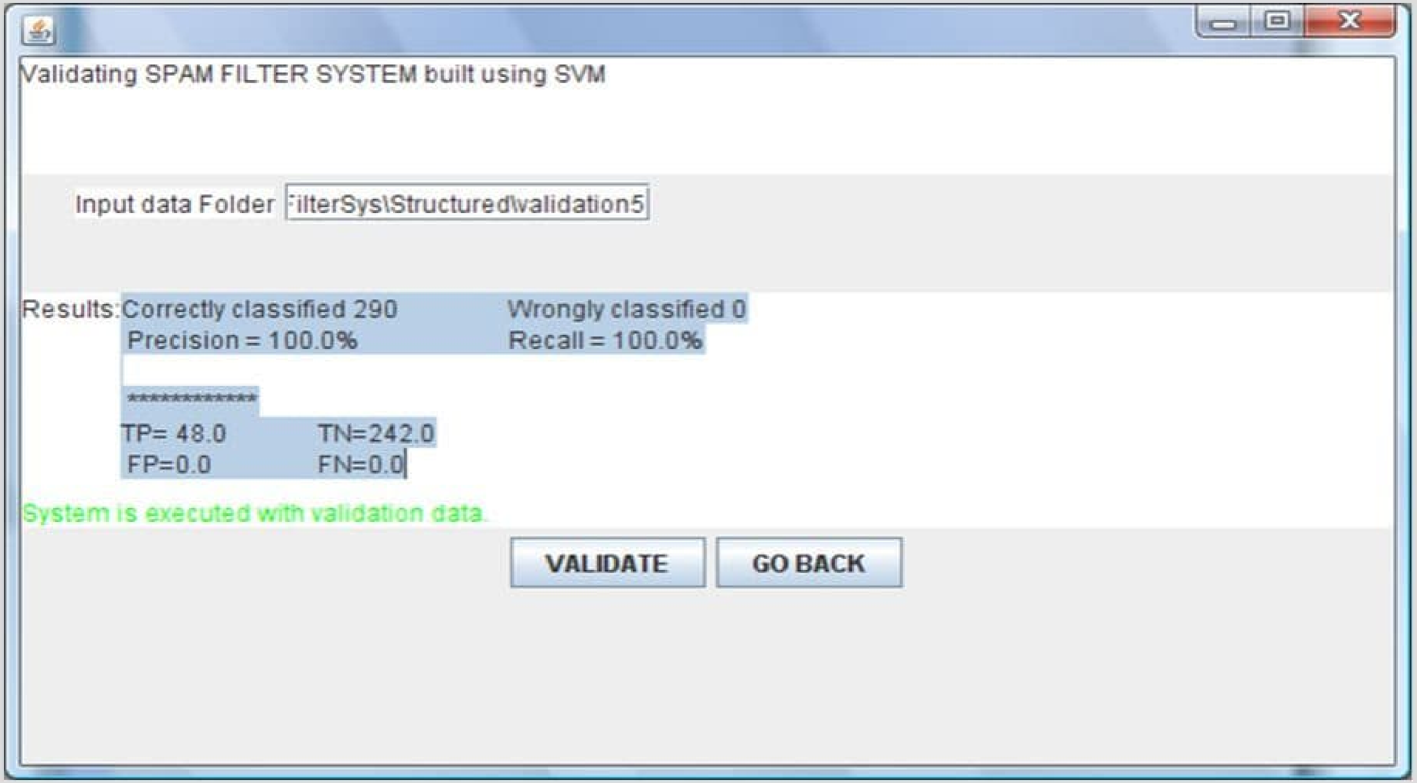
- [✓ NAIVE BAYSE CLASSIFIER]
- NAIVE BAYSE MULTINOMIAL CLASSIFIER
- KNN CLASSIFIER
- Support Vector Machine

Selected classifier is
Naive Bayes Classifier

Validation

- In this step, we are going to validate the model which is built during training phase.
- I am using Parts 8 and 9 of lingspam data set for validation.
- The input to the validation is the dataset of known classes.
- Then the system classifies the data into classes using the model.
- This phase is used to check how the system is classifying the data to measure(like *Precision* and *Recall*) its performance and thus compare the algorithms to select the optimal method.

Validation Screen



Validating SPAM FILTER SYSTEM built using SVM

Input data Folder `FilterSys\Structured\validation5`

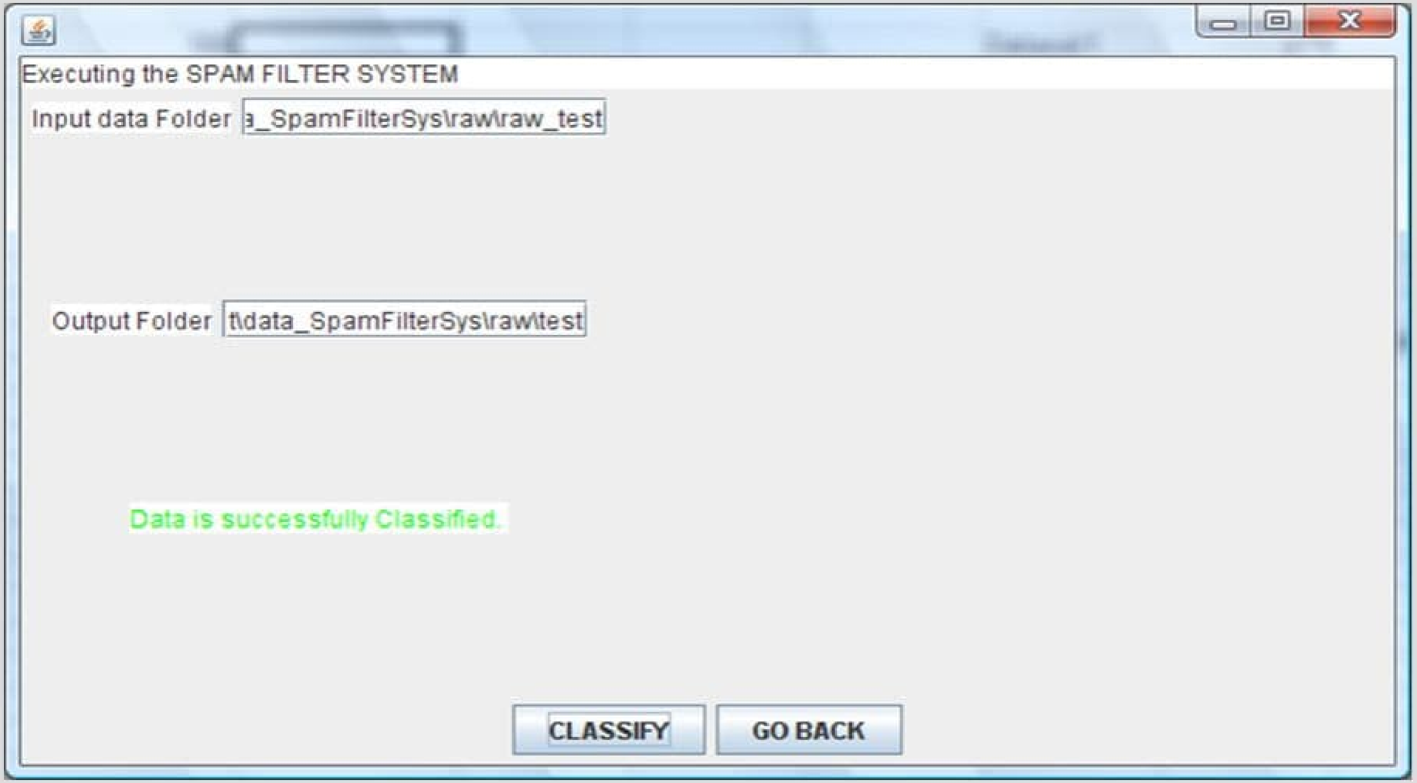
```
Results:Correctly classified 290          Wrongly classified 0
         Precision = 100.0%               Recall = 100.0%

        ************
        TP= 48.0        TN=242.0
         FP=0.0         FN=0.0
```

System is executed with validation data.

Testing/Classification

- In this step, we are going to use the classifier
- I am using Part 10 of the lingspam dataset as an input to test the model.
- Then the system classifies the given dataset into two classes 'spam' and 'clean'.
- Classified file is then copied into *Inbox* folder if it is classified as clean or into *Spam* folder otherwise.

Test Screen



Executing the SPAM FILTER SYSTEM

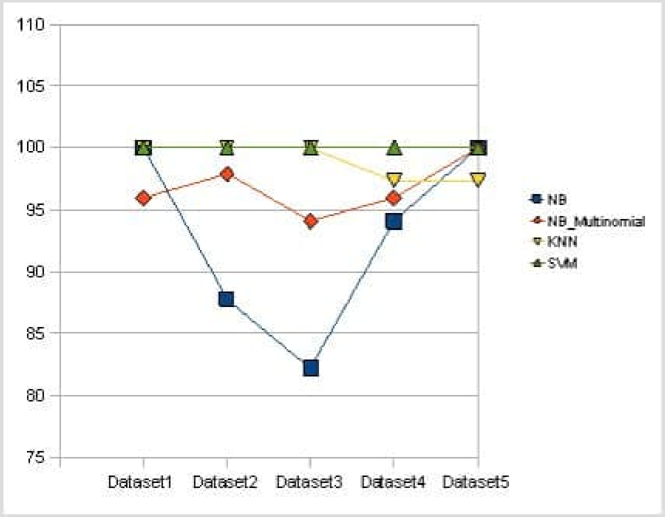
Input data Folder a_SpamFilterSys\raw\raw_test

Output Folder t\data_SpamFilterSys\raw\test

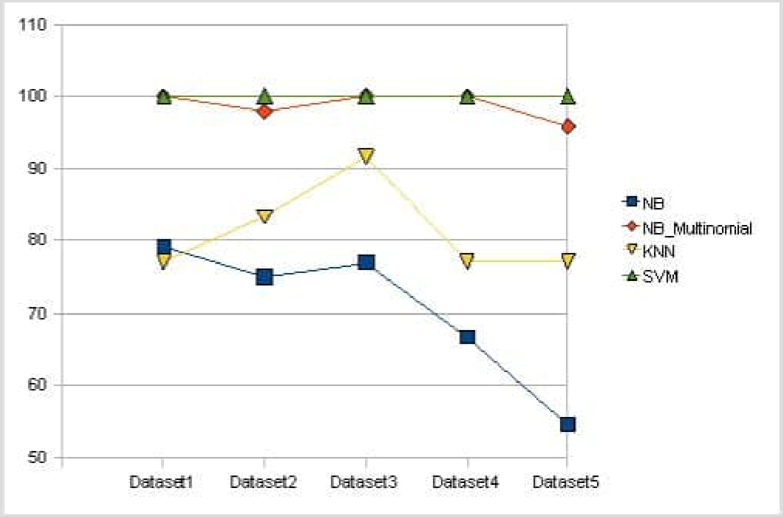
Data is successfully Classified.

Results

Precision



Recall



Conclusion

Spam filter system is implemented using WEKA java API and implementing 4 algorithms.

Based on results Support Vector Machine and Naïve Bayes Multinomial classifier seem to perform better in classification.

This system can be further enhanced by including the functionality of reading Web Pages so that web browsing can be made a spam-free experience by avoiding advertisements and unwanted malicious websites.