Spam Filter Using Natural Language Processing

1st Rinkal Ashishbhai Mehta
Masters Of Computer Science
Lakehead University
Thunder Bay Ontario
rpatel30@lakeheadu.ca

2nd Charmi Ashokbhai Patel
Masters Of Computer Science
Lakehead University
Thunder Bay Ontario
cpatel14@lakeheadu.ca

3rd Nishi Bhaveshkumar Patel
Masters Of Computer Science
Lakehead University
Thunder Bay Ontario
npatel38@lakeheadu.ca

***Abstract*—Email spam is probably the greatest risk to today's Internet. To manage this risk, numerous anti-spam channels have been created. In this paper, we report the performance of a spam filter using natural language processing.The filter has been tested with some datasets in the Spam Collection. Additionally, to the widely used bag-of-words feature, we further consider natural language feature extraction to discover new characteristics from email subjects. In online learning, we have used kneighbors classifier, decision tree classifier, random forest classifier, logistic regression classifier, SDG classifier, naive bayes classifier and SVM linear classifier. We have also used ensemble methods using the voting classifier and obtained the highest accuracy of 98.70% in the dataset we have used. On the basis of our results, we have created a classification report and a confusion matrix.**

***Index Terms*—Spam, classifier, feature-extraction, Accuracy**

I. Introduction

Spamming : With the popularity of the Internet, email is a part of our daily life. It is the most generally utilized medium for communication because of its easy accessibility, reliability, cost effectiveness and speed. Email is prone to spam emails on account of its wide use and the majority of its advantages as an authentic medium of communication. Most spam messages appear as publicizing or promotional materials like obligation reduction plans, getting rich snappy plans, betting opportunities, internet dating, health-related items and so on. The major disadvantages of spam messages are wastage of system assets , time, harm to the PCs and laptops due to viruses.Spammers generally have a structured customized layout messages to deliver their messages using a mass mailing software.

Natural Language Processing: Natural language processing belongs to the CS taxonomy as the child of Artificial Intelligence (AI).Natural Language Processing is a procedure for breaking down and speaking to normally happening writings at atleast one degrees of linguistic analysis to accomplish human-like language processing for a scope of tasks or applications.Naturally occurring texts can be of any language, mode, genre, etc. The texts can be oral or written and must be in a language utilized by people to communicate with each other. Significantly the text being analyzed ought not be explicitly built with the goal of the analysis, but instead it ought to be gathered from actual usage.In straightforward terms, natural language processing is the process written and spoken language for some helpful reason: to make an interpretation of dialects, to get data from the web on text data banks in order to answer questions, to carry on discussions with machines.Natural language processing approaches fall generally into four classifications: symbolic, statistical, connectionist and hybrid. In this paper,we have used statistical approach for the proposed solution. Statistical approaches utilize different mathematical strategies and regularly utilize huge text corpora to create approximate generalized models of semantic phenomena. This paper is consists of the proposed model for spam detection using NLP engine.

II. Proposed System

As shown in Fig 1, the first step in our proposed system is loading the dataset. Afetr loading the datasets we are going to pre-process the data. Pre-processing is important for mining the data or filtering the data. Then the spam control has been done. Spam control is the part of feature extraction.

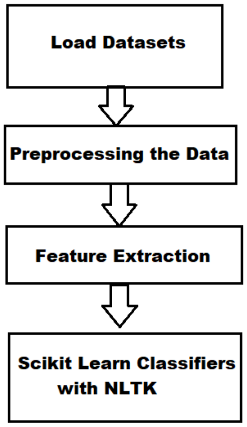


Fig. 1. Flow of the proposed system

Identify applicable funding agency here. If none, delete this.

A. *Loading the Dataset*

1) Load dataset of SMS messages (From UCI): We have taken a collection of SMS message data known as corpus. There are 5572 SMS messages in this data which is written in English language, and also serves as training example. The very first column of this data is the target variable containing the class label, which actually gives us the information about the data being ham or spam. The second column is the text od the SMS, which is stored in string.

But because the taget variable contains discrete values, this becomes a classification task for us.

2) Printing useful information about the dataset:

3) Check class distribution:

B. *Preprocessing the data*

The Fig 2 shows flow diagram of preprocessing the data.

- Convert class labels to binary values; 0 = ham, 1 = spam data
- Store the SMS messages data
- Using regular expressions to replace email addresses, URLs, phone numbers, other such numbers and symbols
- Remove punctuations and whitespaces
- Change all words to lower case
- Remove stopwords from text messages
- Word stemming using porter stemmer
- Creating Bag-Of-Words
- Printing total number of words and 15 most common words
- Use 1500 most common words as features
- Defining find feature function

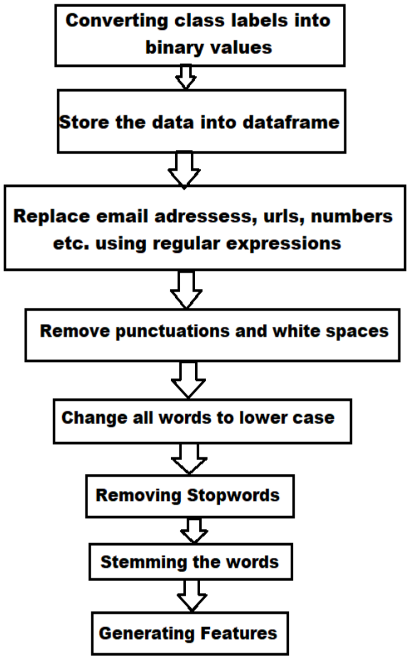

Converting class labels into binary values
Store the data into dataframe
Replace email adressess, urls, numbers etc. using regular expressions
Remove punctuations and white spaces
Change all words to lower case
Removing Stopwords
Stemming the words
Generating Features

Fig. 2. Flow of preprocessing the data

C. *Feature Extraction*

In the above mentioned pre-processing task, our corpus becomes enriched with meaningful terms. This makes it ready to construct features. The corpus can be split in a vocabulary of unique terms called tokenization. The following are the steps for tokenization:

- Creating Bag-Of-Words
- Printing total number of words and 15 most common words
- Use 1500 most common words as features
- Defining find feature function

D. *Scikit Learn Classifiers with NLTK*

- Train the model on training dataset and test it on testing dataset
- Define models to train Essemble methods - Voting Classifier
- Make class label predictions for testing set
- Print a confusion matrix and a classification report

III. Related Works

A. *Stemmimg using Porter Stemmer*

stemming is the way toward lessening curved (or once in a while inferred) words to their word stem, base or root structureby and large a composed word structure. The stem need not be indistinguishable from the morphological base of the word; it is normally adequate that related words guide to a similar stem, regardless of whether this stem isn't in itself a legitimate root.Fig 3 shows how Voting Classifier works. Usefulness:

1)Improving Effectiveness of IR and text mining.

- watching similar words
- mainly improve recall

2)reducing indexing size

- combining words with same roots may reduce indexing size as much as 40-50

fig 3 shows the flow diagram of stemming

B. *Bag-Of-Words*

Individual terms can be tokenized and generate into the model called Bag-Of-Words. But this model has its own glaring pitfall that it fails to capture the innate structure of natural human language. This following example of sentences display the same feature vector, even if theyconvey gramatically different meanings:

- Does Apple taste delicious?
- Apple does taste delicious.

Alternatively, every sequence of n-terms can be tokenized which is called n-grams. I.e. when we tokenize adjacent pairs of words, it yields bigrams. This n-gram model preserves the order of words and thus can capture more information than the bag of words model.

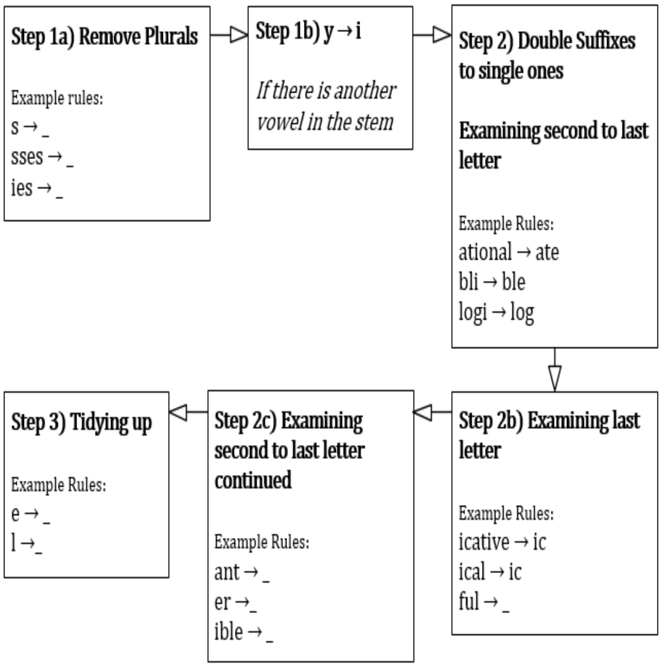


Fig. 3. Flow of Stemming

C. Classifiers

- K Nearest Neighbours
- Decision Tree
- Random Forest
- Logistic Regression
- SGD Classifier
- Naive Bayes
- SVM Linear

D. Ensembling using Voting Classifier

A voting classifier is an essemble learning method, and it is a sort of wrapper contains distinctive Machine learning classifiers to characterize the information with joined voting. There are 'hard' and 'soft' voting strategies to settle on a choice with respect to the objective class. Hard voting chooses as per vote number which is the majority wins. In soft voting, we can set weight an incentive to give more needs to specific classifiers as per their exhibition. In our project we have used Hard Voting classifier.And the figure 4 depicts the whole flow of voting classifier.

IV. Models to train

A. Classification Workflow

There are certain steps of classification that is performed at the backend whenever we perform classification algorithm. The inital step is to understand and recognize the potential features and labels. The charateristics or attributes which affects the label result can be considered as feature.

The classification has generally two phases:

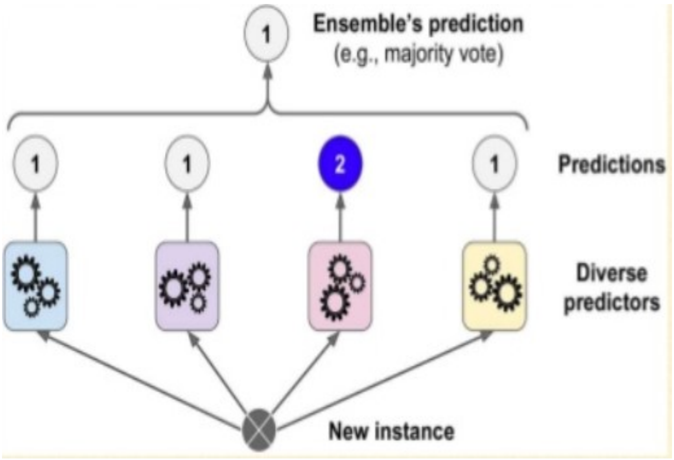


Fig. 4. Flow of Voting Classifier

- The Learning Phase : Classifiers train its model on a given dataset
- The Evaluation Phase : The performance of classifiers is tested. These performance can be evaluated on the basis of various parameters such as accuracy, error, precision, and recall.

B. K Nearest Neighbours

KNN or K-Nearest Neighbor is a algorithm of simple supervised classification which we assign a class to a new data point. Regression is been used in it. No assumptions on the data distribution is made in KNN. Therefore, it is said to be non-parametric. All the training data is been kept together for the future predictions by computing similarities between an input sample and each training instance.The deep flow diagram of KNN is shown in Fig 5.

K-Nearest Neighors can be summarized as below:

- The distance between new data pont with every training data is computed
- The mathematical measures like Euclidean distance, Hamming distance or Manhattan distance is been used for computing distance
- K entries in the database that are closest to the new data point is picked by the model
- Then the majority voting is done i.e. the common class/label from all the K-entries which would be the class of the new data point.

C. Decision Tree

The flow of the decision tree algorithm is shown in fig 6. and the steps are as follows:

- The best attributes is been selected using Attribute Selection Measures (ASM) which is used to split the records
- Attribute is converted into a decision node and breaks the dataset into smaller subsets
- Tree building begins by repeating the process recursively for each child node till one of the condition matches:

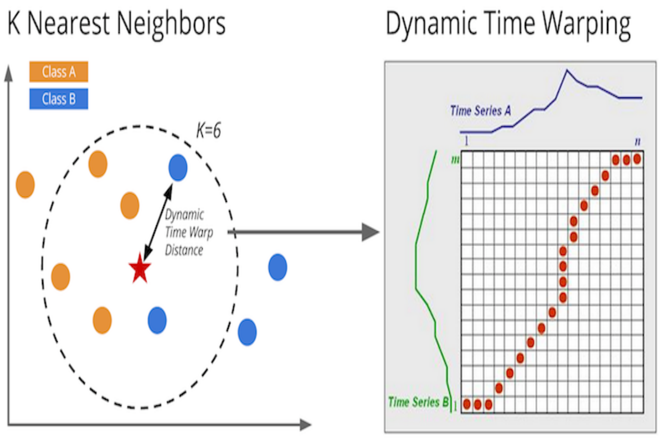


Fig. 5. K Nearest Neighbours

* All the tuples belong to the same attribute value.
* There are no more remaining attributes.
* There are no more instances.

D. Random Forest

This algorithm is used for classification and regression. It is basically an ensemble learning method, made of multiple decision trees. These impacts are been averaged , which improvises the prediction of random forests.The fig 7 depicts the flow of random forest.

Following are the four steps for Random Forest Classifier algorithm:

- Random samples are selected from a given dataset
- Decision tree is constructed for each sample and we get the prediction result from each decision tree.
- The prediction result with the most votes is selected for the final prediction

E. Logistic Regression

Many classification problems like spam detection, diabetes prediction, etc can be solved by Logistic Regression. It is said to be one of the most simple and commonly used Machine Learning algorithm for two-class classification. This algorithm is very easy to implement and be used as baseline for any binary classification problems. Logistic Regression describes and estimates the relationship between one dependent binary variable and independent variable.

In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.). In other words, the logistic regression model predicts P(Y=1) as a function of X.

F. SGD Classifier

What is SGD Classifier?

SGD Classifier actualizes regularized direct models with Stochastic Gradient Descent. All in all, what is stochastic gradient Descent? Stochastic gradient descent considers just 1 irregular point while changing loads not at all like slope plummet which thinks about the entire preparing information. All things considered stochastic angle drop is a lot quicker than inclination plunge when managing enormous informational indexes. Here is a pleasant answer on Quora which clarifies in detail the contrast between slope plunge and stochastic angle plummet.

G. Naive Bayes

Naive Bayes is a factual characterization system dependent on Bayes Theorem. It is one of the least difficult administered learning calculations. Naive Bayes classifier is the quick, precise and dependable calculation. Naive Bayes classifiers have high exactness and speed on enormous datasets.

Naive Bayes classifier expect that the impact of a specific component in a class is free of different highlights. For instance, a credit candidate is attractive or not relying upon his/her salary, past advance and exchange history, age, and area. Regardless of whether these highlights are associated, these highlights are as yet considered freely. This suspicion streamlines calculation, and that is the reason it is considered as guileless. This supposition that is called class restrictive freedom.

Naive Bayes classifier calculates the probability of an event in the following steps:

Step 1: Calculate the prior probability for given class labels
Step 2: Find Likelihood probability with each attribute for each class
Step 3: Put these value in Bayes Formula and calculate posterior probability.
Step 4: See which class has a higher probability, given the input belongs to the higher probability class.

H. SVM Linear

SVM offers high exactness contrasted with different classifiers, for example, calculated relapse, and decision trees. It is known for its kernel trick to deal with nonlinear input spaces. It is utilized in an assortment of uses, for example, face dtection, classification of emails, arrangement of messages, news articles and web pages, classification of genes, and handwriting recognition.

In this exercise, you will utilize scikit-learn in Python. In the event that you might want to study this Python bundle, I prescribe you investigate our Supervised Learning with scikit-adapt course.

SVM is an energizing calculation and the ideas are generally basic. The classifier isolates information focuses utilizing a hyperplane with the biggest measure of edge. That is the reason a SVM classifier is otherwise called a discriminative classifier. SVM finds an ideal hyperplane which aides in grouping new information focuses.

For the most part, Support Vector Machines is viewed as an arrangement approach, it yet can be utilized in the two kinds of grouping and relapse issues. It can without much of a stretch handle different ceaseless and all out factors. SVM builds a hyperplane in multidimensional space to isolate

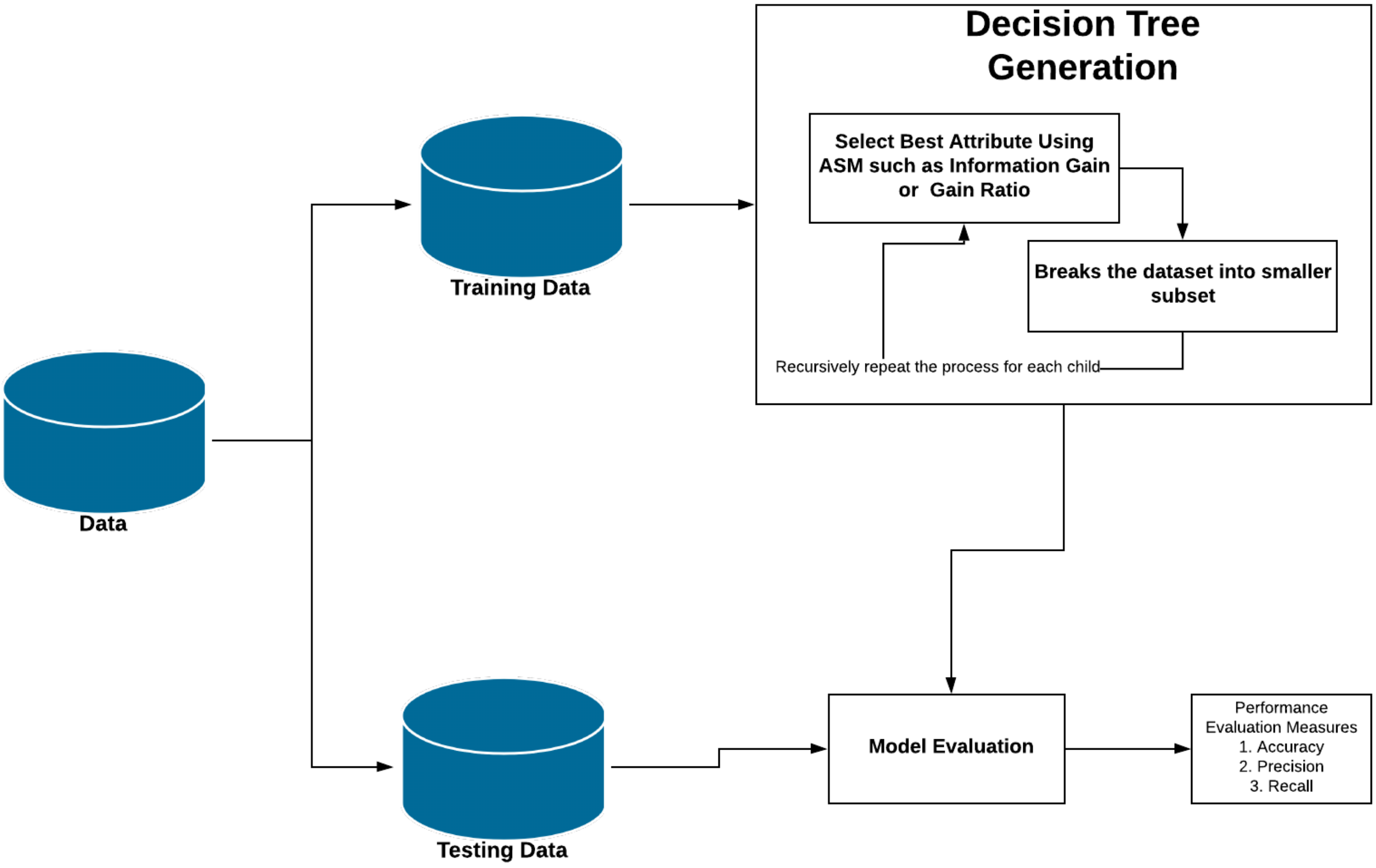

Decision Tree Generation
Select Best Attribute Using ASM such as Information Gain or Gain Ratio
Breaks the dataset into smaller subset
Recursively repeat the process for each child
Training Data
Data
Testing Data
Model Evaluation
Performance Evaluation Measures
1. Accuracy
2. Precision
3. Recall

Fig. 6. Flow of Decision Tree

various classes. SVM creates ideal hyperplane in an iterative way, which is utilized to limit a blunder. The center thought of SVM is to locate a most extreme negligible hyperplane(MMH) that best partitions the dataset into classes.

How does SVM functions?

The principle goal is to isolate the given dataset in the most ideal manner. The separation between the either closest focuses is known as the edge. The goal is to choose a hyperplane with the greatest conceivable edge between help vectors in the given dataset. SVM looks for the greatest peripheral hyperplane in the accompanying advances:

Produce hyperplanes which isolates the classes in the most ideal manner. Left-hand side figure indicating three hyperplanes dark, blue and orange. Here, the blue and orange have higher arrangement blunder, yet the dark is isolating the two classes effectively.

Select the privilege hyperplane with the most extreme isolation from the either closest information focuses as appeared in the right-hand side figure.

V. Results

The results are shown in Table 1 and Figure 8.

TABLE I
CONFUSION MATRIX

	precision	*Recall*	*f1-score*	*support*
0	0.99	1	0.99	1195 [a]
1	0.99	0.91	0.95	198 [a]
micro avg	0.99	0.99	0.99	1393 [a]
macro avg	0.99	0.96	0.97	1393 [a]
weighted avg	0.99	0.99	0.99	1393 [a]

[a]

VI. Conclusion

In this examination, we looked into machine Learning draws near and their application to the field of spam separating. A survey of the best in class calculations been connected for arrangement of messages as either spam or ham is given. The endeavors made by various scientists to taking care of the issue of spam using AI classifiers was examined. The development of spam messages throughout the years to sidestep channels

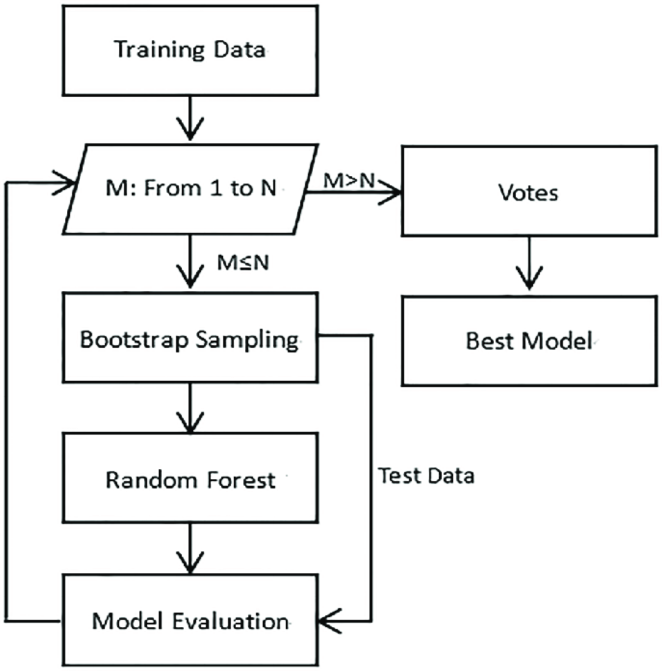
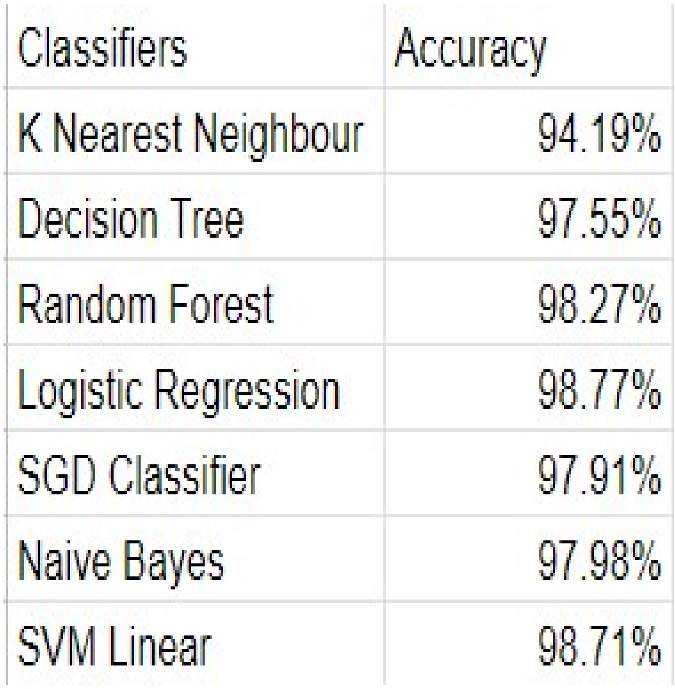


Fig. 7. Flow of Random Forest



Classifiers	Accuracy
K Nearest Neighbour	94.19%
Decision Tree	97.55%
Random Forest	98.27%
Logistic Regression	98.77%
SGD Classifier	97.91%
Naive Bayes	97.98%
SVM Linear	98.71%

Fig. 8. Comparison of accuracy of all classifiers

was inspected. The fundamental engineering of email spam channel and the procedures associated with separating spam messages were investigated. The paper reviewed a portion of the freely accessible data sets and execution measurements that can be utilized to quantify the adequacy of any spam channel. The difficulties of the Machine Learning calculations in productively dealing with the threat of spam was brought up and similar investigations of the Machine Learning methods accessible in writing was finished. We additionally uncovered some open research issues related with spam channels. As a rule, the figure and volume of writing we evaluated demonstrates that critical advancement have been made will at present be made in this field. Having examined the open issues in spam separating, further research to improve the adequacy of spam channels should be finished. This will make the advancement of spam channels to keep on being a functioning exploration field for academician and industry professionals looking into AI systems for viable spam sifting. Our expectation is that exploration understudies will utilize this paper as a spring board for doing subjective research in spam separating utilizing AI, profound inclining and profound antagonistic learning calculations.

VII. References

Awad M., Foqaha M. Email spam classification using hybrid approach of RBF neural network and particle swarm optimization. Int. J. Netw. Secur. Appl. 2016;8(4) [Google Scholar]Awad M, Foqaha M (2016) Email spam classification using hybrid approach of RBF neural network and particle swarm optimization. Int. J. Netw. Secur. Appl., 8(4).

2. Fonseca D.M., Fazzion O.H., Cunha E., Las-Casas I., Guedes P.D., Meira W., Chaves M. Measuring characterizing, and avoiding spam traffic costs. IEEE Int. Comp. 2016;99 [Google Scholar]Fonseca DM, Fazzion OH, Cunha E, Las-Casas I, Guedes PD, Meira W, Chaves M (2016) Measuring characterizing, and avoiding spam traffic costs. IEEE Int. Comp., 99.

3.https://www.researchgate.net/figure/Ensemble-Random-Forest-algorithm$_{fig}1_{3}$19052112

4. https://www.researchgate.net/figure/Ensemble-Random-Forest-algorithm$_{fig}1_{3}$19052112

5. https://www.datacamp.com/community/tutorials/decision-tree-classification-python

6. https://www.slideshare.net/CloudxLab/ensemble-learning-and-random-forests