Spam Detection Using Naive Bayes

Tejas Harish Borkar, 2K18/CO/373
tejasharishborkar_2k18co373@dtu.ac.in
Department of Computer Science and Engineering
Delhi Technological University Delhi, India

Tushar Ahuja, 2K18/CO/374
tusharahuja_2k18co374dtu.ac.in
Department of Computer Science and Engineering
Delhi Technological University Delhi, India

Abstract- Email spams are drastically rising year on year due to the increase in the number of internet users across the globe. The volume of information sent over the web is increasing at an exponential rate, and therefore the devices connected are considered vulnerable. Email spams are a drawback because emails are regarded as a trustful means of information transfer. Thus there is an increasing need to prevent email spams, and one way to do so is spam filtering. In this paper, multiple email spam datasets are taken from UCI Machine Learning repository and kaggle. Pre-processing techniques and Naive Bayes (NB) are applied to the dataset, and the performance of the algorithm is calculated.

Keywords- *spam detection, Pre-processing, spam, ham, Naive Bayes, Laplace smoothing, oversampling, supervised learning, accuracy, precision, recall, f-beta score.*

I. INTRODUCTION

One of the significant ways of communication among a large number of people is email. Ham messages are the one that's created by actual users, and spam messages are those created by promotional corporations and are usually sent in bulk . Thus there is a requirement for a method to filter spam messages for convenience of the actual users of email. The consequences of SMS spam are often frustrating and cause inconvenience to users. One crucial factor to keep in mind while designing spam filters is that there should not be any misclassification, i.e. ham messages being eliminated and blocked as spam. Despite the problems faced by actual users, an enormous amount of spam messages are sent daily. Therefore there is an even greater need for designing efficient and accurate spam filters.

II. LITERATURE REVIEW

There are various supervised machine learning algorithms & Statistical Spam Filtering Techniques for filtering spam and ham messages like naïve Bayes Algorithm, support vector machines algorithm, decision trees, artificial neural networks, and many more.

A. Naive Bayes

The Naive Bayes algorithm is based on Bayes Theorem. This algorithm classifies each object by looking at all of its features individually. Bayes Rule helps us to calculate the posterior probability for just one feature. The posterior probability of every item is calculated, and these probabilities are multiplied together to get the final probability. This procedure is repeated for the other class as well. The class with the greater probability determines what class the object should belong.

B. Support Vector Machine

SVMs have out-performed other learning algorithms with good generalization, the number of tuning parameters, and its firm theoretical background. The state of the art of Support Vector Machines evolved, mapping the learning data from input space into higher dimensional feature space where the classification performance is increased SVM's perform best when using binary features. SVM's provide good accuracy and speed. They also have significantly less training time as compared to other techniques.

C. Decision Trees

It is a commonly used tool for decision-making. A new instance is passed along the tree, and as per the values of the nodes, it is classified accordingly. C4.5, ID3, CART are some of the most popular decision tree algorithms. Spam detection using decision trees is done using recursive binary splitting.

D. Artificial Neural Networks (ANN)

As per the Neural Network theory, for static pattern classification, the best performance shows the layered feed-forward networks, called Multilayer Perceptrons, typically trained with static backpropagation. Their advantage is that they are reasonably easy to use, and they can approximate any input/output map. The main disadvantage is that they have considerable training time and require large amounts of training data. The method for spam detection employs attributes composed from characteristics of the elusive patterns that spammers use rather than the frequency or context of keywords in the messages. They find out which ANN configuration will have the best performance and least error to find the desired output.

III. RESEARCH METHODOLOGY

A. Datasets

- The datasets are a set of SMS tagged messages that have been collected for spam research.
- It contains a set of SMS messages in English tagged according to ham (legitimate) or spam.
- It is an example of **supervised learning**.

The datasets have been downloaded from three sources :

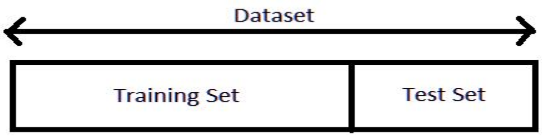
- https://archive.ics.uci.edu/ml/datasets/sms+spam+collection
- https://www.kaggle.com/ozlerhakan/spam-or-not-spam-dataset
- https://www.kaggle.com/venky73/spam-mails-dataset

Oversampling had to be done as the datasets were imbalanced.

- Oversampling is used to adjust the class distribution in a data set. In other words, it helps to adjust the ratio between the different classes/categories represented by the dataset.
- We performed over-sampling of the spam messages by adding spam entries to the dataset.
- The ratio of spam:ham before over-sampling: 14:86
- The ratio of spam:ham after over-sampling: 36:64

B. Validation Methods

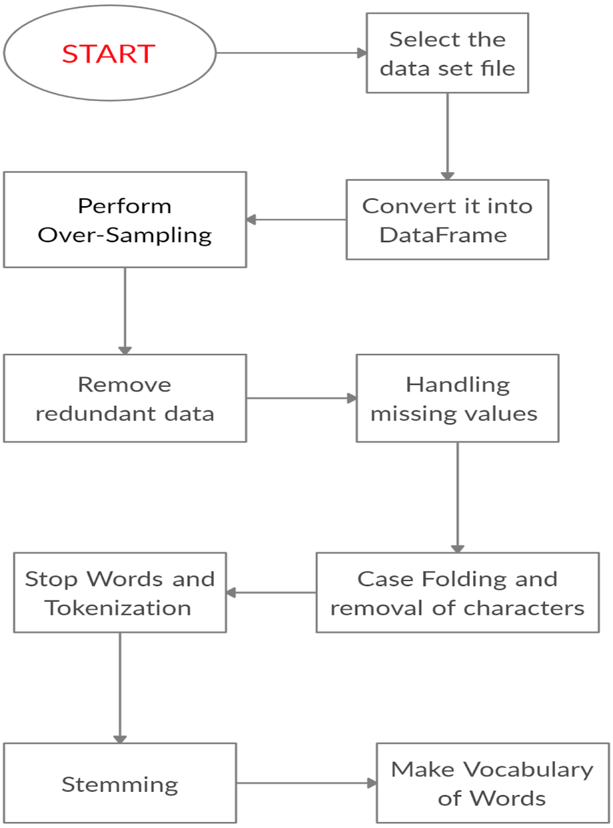
- **Hold-out Validation Technique**



  - The dataset is split into two sets: a training set and a testing set.
  - 80% of the data is used for training, and the remaining 20% for testing.
  - The dataset is randomized before the split is done to ensure that spam and ham messages are uniformly spread throughout the dataset.
  - The percentage of ham and spam messages in the training and testing sets are analyzed.
  - The percentages are expected to be close to what we have in the complete dataset.
- **Stratified K-Folds cross-validation**
  - In Stratified Sampling, let the population of a state be 55.7% male and 44.3% female. If we wish to choose 1000 people from that state and if 557 male (55.7% of 1000) and 443 female (44.3% for 1000), i.e. 557 male + 443 female (Total=1000 people) are picked for their opinion, then this chosen group of people represent the entire state.
  - Stratified k-fold cross-validation is the same as just k-fold cross-validation, but in Stratified k-fold cross-validation, stratified sampling is done instead of random sampling.

C. Preprocessing

The Preprocessing of the training data and testing is separately performed to simplify the process. The phases being taken in the preprocessing include :



- **Handling missing values:** We handle missing values by :
  - Deleting the row if Email is missing.
  - Replacing missing values with ham if the label is missing so that there is no risk of data loss.
- **Handling redundant values:** We delete the duplicate rows that is we keep only a single instance of every message.
- **Case Folding and removal of characters:** All the text is converted to lowercase and characters other than letters, numbers and punctuation are deleted.
- **Removing Stop Words and Tokenization:** In this process, we remove stopwords and tokenization is carried out to break the string into tokens or individual words.
- **Stemming**: Stemming is the process of converting different variants of a word to the root/base form of the word. For example, the words "playing", "plays" and "played" are converted to the root form, i.e. "play".Stemming is an essential part of the pipelining process in Natural language processing.

- **Vocabulary creation:**
  - The vocabulary is created, which means a list of all the unique words in our training set.
  - We will now use the vocabulary we created to make a new DataFrame.
  - Therefore, data access is simple, and this DataFrame is used to make the required data transformation.

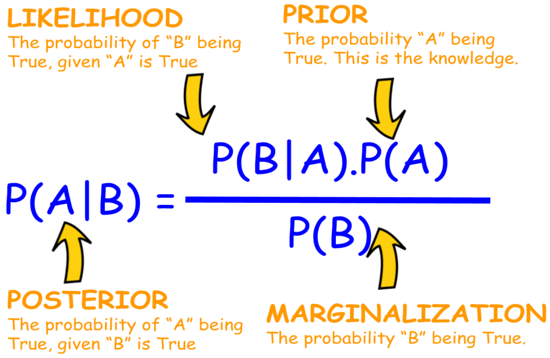
D. Data Analysis Techniques

We have used Naive Bayes classifier as our data analysis technique to build the spam detection model.

- The Naive Bayes algorithm is primarily based on Bayes' theorem with the assumption of independence between predictors.
- A Naive Bayesian model is relatively easy to build, as there is no complicated iterative parameter estimation. This makes it useful for massive datasets.
- Despite its simplicity, it does surprisingly well and is extensively used because it performs better than some sophisticated classification methods.

Algorithm:

The Bayes theorem enables us to calculate the posterior probability P(A|B) from P(A), P(B), and P(B|A). Naive Bayes classifiers are based on the assumption that the effect of the value of the predictor (B) on a given class (A) is independent of the values of other predictors. This is known as class conditional independence.


LIKELIHOOD
The probability of "B" being True, given "A" is True
PRIOR
The probability "A" being True. This is the knowledge.
P(A|B) = P(B|A).P(A) / P(B)
POSTERIOR
The probability of "A" being True, given "B" is True
MARGINALIZATION
The probability "B" being True.

- P(A|B): the posterior probability of class (target) given predictor (attribute).
- P(A): the prior probability of class.
- P(B|A): likelihood which is the probability of the given predictor class.
- P(B): the prior probability of the predictor.

Advantages of Naive Bayes

- When the assumption of independent predictors holds, a Naive Bayes classifier performs better as compared to other models.
- A small amount of training data is used to estimate the test data. So, the training period is less.
- It is easy to implement.

Disadvantages of Naive Bayes

- One of the main limitations is the assumption of independent predictors. An essential premise of the algorithm is conditional independence. But in many situations, it is impossible to get an entirely independent set of predictors.

- Suppose a categorical variable has a category in the testing data set, which was not previously observed in the training data set. In that case, the model assigns a 0 probability and will be unable to make a prediction which is often known as Zero Frequency. We can use smoothing technique.s to solve this issue. One of the most straightforward smoothing techniques is called Laplace estimation.

Calculating Constants and parameters(Naive Bayes Algorithm)

When a new message comes in, the model makes the classification based on the results it gets on the basis of these equations, where "w1" denotes the first word, and w1,w2, ..., wn is the entire message:

$$\mathrm{P(Spam|w_1, w_2, ..., w_n) \propto P(Spam) \cdot \prod_{i=1}^{n} P(w_i|Spam)}$$

$$\mathrm{P(Ham|w_1, w_2, ..., w_n) \propto P(Ham) \cdot \prod_{i=1}^{n} P(w_i|Ham)}$$

To calculate P(wi| spam) and P(wi|Ham), we need to use separate equations:

$$\mathrm{P(w_i|Spam)} = \frac{\mathrm{N_{w_i|Spam}} + \alpha}{\mathrm{N_{Spam}} + \alpha \cdot \mathrm{N_{Vocabulary}}}$$

$$\mathrm{P(w_i|Ham)} = \frac{\mathrm{N_{w_i|Ham}} + \alpha}{\mathrm{N_{Ham}} + \alpha \cdot \mathrm{N_{Vocabulary}}}$$

$\mathrm{N_{w_i|Spam}}$ = the number of times the word $\mathrm{w_i}$ occurs in spam messages

$\mathrm{N_{w_i|Ham}}$ = the number of times the word $\mathrm{w_i}$ occurs in ham messages

$\mathrm{N_{Spam}}$ = total number of words in spam messages

$\mathrm{N_{Ham}}$ = total number of words in ham messages

$\mathrm{N_{Vocabulary}}$ = total number of words in the vocabulary

$$\alpha = 1 \quad (\alpha \text{ is a smoothing parameter})$$

Now the constants are calculated

- P(Spam) and P(Ham)
- N_{Spam}, N_{Ham}, $N_{Vocabulary}$

We also use Laplace smoothing and set $\alpha = 1$.

Need of Laplace Smoothing

$$\mathbf{P(Spam|w_1, w_2, ..., w_n) \propto P(Spam) \cdot \prod_{i=1}^{n} P(w_i|Spam)}$$

- It is performed to overcome the disadvantage of Naive Bayes.
- If any term in the Right-Hand side is 0, that is if a given word doesn't occur in spam messages, then the entire probability becomes zero, that is not a desired trait.
- Therefore Laplace smoothing is needed so that the probability remains non zero.
- The same is true in the case of ham messages as well.

CLASSIFYING A NEW MESSAGE

Take as input a new message (w1, w2, ..., wn).

The values of P(Spam|w1, w2, ..., wn)=x and P(Ham|w1, w2, ..., wn)=y are calculated and compared.

- If y > x, then the message is classified as ham.
- If x < y, then the message is classified as spam.
- If x = y, then the algorithm may request human help.

Some messages may contain words that are not part of the vocabulary. Such words are ignored while calculating the probabilities.

Performance Measures

- Accuracy: The ratio of the number of correct predictions to the total predictions made.

Accuracy = (TP+TN)/total

- False Positive (FP): The number of misclassified non-spam emails.
- False Negative (FN): The number of misclassified spam emails
- True Positive (TP): The number of spam messages correctly classified as spam.
- True Negative (TN): The number of non-spam emails correctly classified as non-spam.
- Recall: Percentage of spam messages managed to block.

r = TP/(TP+FN)

- Precision: Percentage of correct message for spam emails.

p = TP/(TP+FP)

- F Beta-measure: Weighted average of precision and recall.

F-Beta-measure = (1+beta2)p*r / (beta2)(p+r)

Choosing appropriate beta value for f beta score

- In case if FP and FN both are equally important, then we use beta = 1
- In case if FP is more critical than FN, then we reduce the value of beta (between 0 to 1)
- In case if FN is more critical than FP than we increase beta value (greater than 1)

In our case, if a ham message arrives and if the model reports this message as spam, then this is big trouble for the user/client, i.e. the number of misclassified non-spam emails is significant (False Positive).

So we will use beta = 0.5 (also known as F0.5 score)

F0.5 score = 5*p*r/(p+r)

IV. EXPERIMENTAL DESIGN

Research Variable:

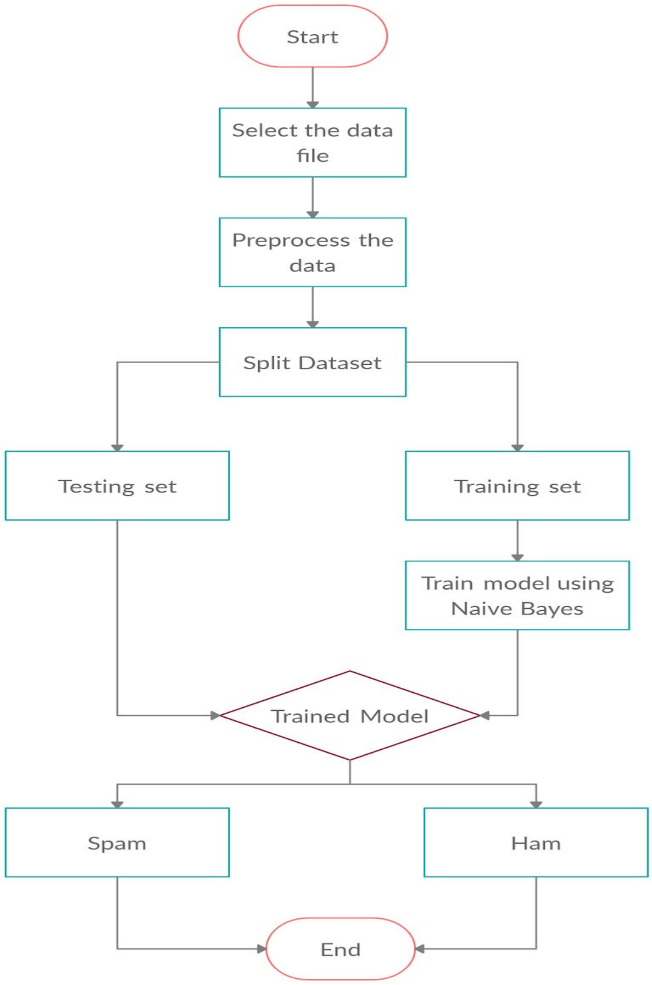
Dependent variable

- The dependent variable is the label that is whether the message is classified as spam/ham.
- It is a nominal variable.

Independent variable

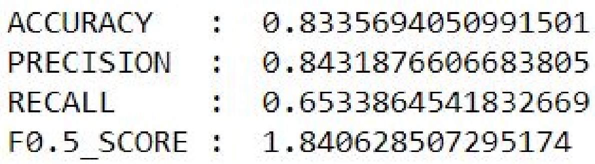
- These are the indices of the words stored in vocabulary (0 to total no. of words in vocabulary-1).
- It is a metric ratio variable as real 0 has a meaning that is the absence of a word.

Flow chart of the entire process



V. RESULTS AND DISCUSSION

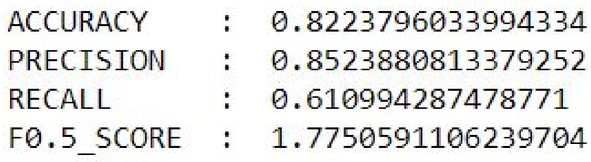
1. Using Hold Out Validation



```
ACCURACY    :  0.8335694050991501
PRECISION   :  0.8431876606683805
RECALL      :  0.6533864541832669
F0.5_SCORE  :  1.840628507295174
```

2. Stratified K fold Cross Validation

Mean values for k=10



```
ACCURACY     :  0.8223796033994334
PRECISION    :  0.8523880813379252
RECALL       :  0.610994287478771
F0.5_SCORE   :  1.7750591106239704
```

As it can be observed from the performance measures, hold out validation has a slightly better accuracy whereas as stratified k folds cross-validation has slightly better precision. If we consider recall and F0.5_Score then hold out validation outperforms stratified k folds cross-validation.

VI. VALIDITY THREADS

- Conditional independence is a vital assumption of Naive Bayes. This is often not valid in many real-world scenarios. Thus it sometimes oversimplifies the problem and gives a sub-par performance.
- Naive Bayes often gives incorrect results if Laplace smoothing is not used.
- In the case of imbalanced data, accuracy cannot be considered as a good performance measure.

VII. CONCLUSION

Email spam filtering is a crucial issue in the context of network security. Naive Bayes classifier has a vital role in the process of email spam filtering. The performance of Naive Bayes classifier is based on datasets used. It has been observed that datasets with fewer instances of emails and attributes give better performance. They also offer high precision, i.e. the percentage of spam messages managed to block.

VIII. REFERENCES

- Arifin, D. D., & Bijaksana, M. A. (2016, September). Enhancing spam detection on mobile phone Short Message Service (SMS) performance using FP-growth and Naive Bayes Classifier. Google.com. In 2016 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob) (pp.80-84). IEEE.
- https://app.creately.com/ for creating flow charts.
- https://www.kaggle.com/ for datasets.