Email Spam Detection Using Naïve Bayes

Hrithik Vohra*

Dept. of Computer science & Engineering

Delhi Technological University

Delhi,India

Manoj Kumar**

Dept. of Computer science & Engineering

Delhi Technological University

Delhi,India

Abstract*—Today almost everyone around the globe is using emails with various purposes and hence an efficient growth in the no. of spam emails is witnessed with time which creates problem for the users to find their genuine/ham emails because of which their precious time is wasted and the system becomes less efficient. This is where E-mail spam/ham detection comes into play, playing a significant role in classifying the emails into spam or ham respectively and thus saving users a lot of time to fetch their E-mails. This research paper aims to apply the ML algorithm i.e. multinomial naïve bayes classifier to classify E-mails into spam or ham.We are also going to compare the 2 methods of vectorizing that are the Bag of Words (BOW) & Term frequency -Inverse document frequency (TF-IDF) models.

Keywords—email spam detection, ham or spam, bag of words, term frequency -inverse document frequency, multinomial naïve bayes

I. INTRODUCTION

A. *Overview*

Email is now one of the most important forms of communication. A study estimated that there are around 5 billion accounts and the number is much more higher for the number of emails flowing everyday. Due to this type of extensive use spam is one of the major threats posed to email users. A study found out that approx.. 70% of all email flown were spam in 2013. Now lets see why spam emails are a problem to us. Spam emails can be defined as the fake and useless emails send by the people itself either to promote their organisation/company or to make frauds and make money by cheating them .

B. *Motivation*

The E-mail users waste a lot of their precious time in sorting spam E-mails. Numerous copies of same messages are transmitted too many times which not only effects a company or organisation but also irritates the receiving user, causing stress, irritation and a tedious work performing this task to filter.

Therefore, an effective spam filtering system will be an important contribution to the society ,which can easily filter out the emails and taking the burden away from the email users. Here, we will be using Multinomial naïve bayes classifier for the classification of E-mails into spam/ham.

C. Naïve Bayes

Naïve Bayes algorithm is a supervised ML algorithm which is based on famous Bayes theorem.This algorithm is highly used in text classification including high dimensional training dataset.We will be using multinomial Naïve bayes and holdout technique for email spam filteration.

II. RELATED WORK

A great deal of research work has already been dedicated to this field and is still going on everyday to increase the efficiency of the system.We have read the following papers related with E-mail Spam/ham Detection. Researchers and scientists are constantly working to make the models more efficient to identify the spam mails. Works of some researchers have been discussed here below.

[1] A research in which an author proposed an E-mail filtering system using 2 different features selection methods to classify and at last the featuresg are sele ted using TF-IDF and rough set theory method.

[2] Another research includes implementation of K Nearest Neighbours algorithm & Naïve Bayes algorithms which shows precise results when applied on pre-processed data.

[3] A group of researchers made an email spam detection model by using the naïve bayes and the J48 classifier and then compared them in which they found out J48 performing well than the naïve bayes classifier.

[4] Authors proposed an ontology kind of based email filtering and classification method. The considered dataset they used is classified using J48 decision tree based algorithm.

III. DATASET

The dataset we used to train our model is email dataset. This is a very popular and a very common dataset among many data analysts/scientists who are working in this email spam detection field.

Our dataset name is emails.csv and it has been taken from Kaggle.com .It contains a total of 5695 unique emails out of which 4327 are ham emails and 1368 are spam emails ,hence in order to make the dataset equal we removed 2959 ham emails and finally our new dataset includes 1368 ham as well as spam emails. The emails.csv contain 2 columns text and spam in which the text columns have all the email messages and the spam column contains 2 values i.e. 1 denoting spam emails and 0 denoting ham emails. Below shows the dataset of first 5 emails:

TABLE 1. EMAILS CLASSIFIED AS SPAM/HAM

	text	spam
0	Subject: naturally irresistible your corporate...	1
1	Subject: the stock trading gunslinger fanny i...	1
2	Subject: unbelievable new homes made easy im ...	1
3	Subject: 4 color printing special request add...	1
4	Subject: do not have money , get software cds ...	1

Below we have a representation of length of emails in our dataset:

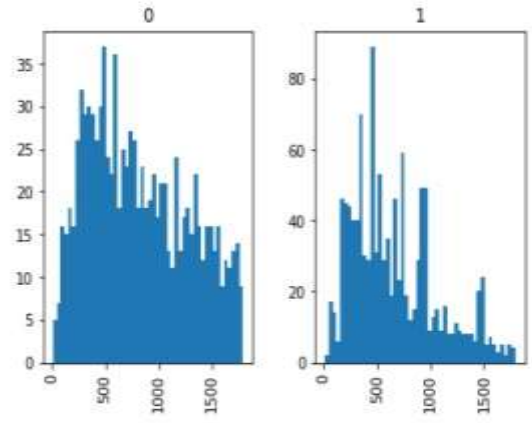

Fig 1. Graph length of email vs no. of Emails

Y-axis shows number of emails containing lengths on the X-axis. So here it shows there are all the types of emails available i.e. of various lengths.

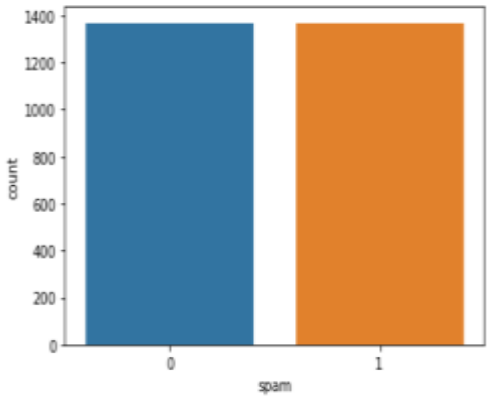

Fig 2. Equalising the no. of spam and ham emails for genuine results

IV. EXPERIMENTAL DESIGN

This column will highlight all the work we had done in this project.

A. Validation Technique Used

Here we are using the holdout validation technique. Hold-out is when you split up your dataset into a 'train dataset' and 'test dataset ' set. The training set is on what the model is trained on, and the testing dataset is used to see the performance of the model on the unseen test data. For splitting data we have split our dataset into 70% training and 30% test data.

B. Performance Metrics

The Performance metrics we have used in our model are *accuracy*, *precision* and *F1 score.*

True Positive (TP) – Situations where both the actual and predicted class are 1.

True Negative (TN) − Situations where both the actual and predicted class are 0.

False Positive (FP) – Situation where actual class of input data is 0 but the predicted class turns out 1.

False Negative (FN) − Situation where ,actual class of input data is 1 but the predicted class turns out 0.

1) Accuracy

It is the most common and the most primitive performance metric used in classification algorithms. It is defined as no. of correct/true predictions divided by the total no. of predictions.

$$\text{ACCURACY} = \frac{TP + FN}{TP + FP + TN + FN} \quad (1)$$

2) Precision

Precision basically means no. of correct prediction returned by the model we have built. We can easily calculate it by formula:

$$\text{PRECISION} = \frac{TP}{TP + FP} \quad (2)$$

3) F1-Score

This is considered to be one of the finest and better performance measure than the latter. It is defined as the harmonic mean of recall & precision. Mathematically, it is equal to the weighted average of recall & precision. Worst value it can take is 0 and the best is 1. It is calculated as :

$$\text{RECALL} = \frac{TP}{TP + FN} \quad (3)$$

$$F1 = 2 * (precision * recall) / (precision + recall) \quad (4)$$

C. Pre-processing

We have done a series of pre-processing on our data in the following order:

a) Removed duplicate data from the dataset and checked for null values which in our case was 0 so no further processing was required

b) Made the number of spam and ham emails equal in number. This is to make sure so that the predicted results are more real as both of them are equal and is not dominated by only 1 single class.

c) Removed the first word i.e. 'subject:' from all the text messages as it was unnecessary.

d) Removed all the punctuations, numerical values, stop words from each and every text, made sure every word is in lower case and at the last did stemming of all the words in the mail. *Stemming* is often defined as the procedure of reducing a word to its word stem known as *lemma*. It is done in order to remove the extra columns formed by the words that mean the same but are just in other form ex. visit, visited ,visits etc.

D. Algorithms Used

Naïve Bayes

Naïve Bayes algorithm is one of the machine learning supervised learning algorithm, which is based on the famous Bayes theorem .The algorithm is highly used in text classification problems including high-dimensional training dataset.

Naïve Bayes Classifier is one of the simple and most efficient Classification algorithms which helps us to build fast ML models as well as efficient training and testing to make correct and fast predictions. It is a probabilistic classifier, the whole basis of the algorithm depends on the probabilities calculated which means it predicts on the basis of the probability of an object.

Bayes' Theorem

Bayes theorem also known as Bayes' Rule or Bayes' law, is used in determining the probability of a hypothesis with prior knowledge. It futher also depends on the conditional probability.

Bayes Theorem formulates the following:

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

A, B = events
$P(A|B)$ = probability of A given B is true
$P(B|A)$ = probability of B given A is true
$P(A), P(B)$ = the independent probabilities of A and B

For more than one features given that they are strictly independent of each other (x1,x2,…xd) the formula now becomes:

$$P(Y|X_1, X_2, \ldots, X_d) = \frac{P(X_1, X_2, \ldots, Xd|Y)\,P(Y)}{P(X_1, X_2, \ldots, X_d)} \quad (5)$$

$$P(X_1, X_2, \ldots, X_d|Y_j) = P(X_1|Y_j)\,P(X_2|Y_j)\ldots P(X_d|Y_j) \quad (6)$$

Types of Naïve Bayes Model:

The 3 types of Naive bayes model are given below:

Gaussian: The Gaussian model works for only features that follow a normal distribution i.e. if predictors take continuous values instead of discrete, means the values are a result of gaussian distribution.

Bernoulli: The Bernoulli classifier works similar to the Multinomial, but the predictor variables here are independent Booleans variables(0 or 1) it's a bit complicated but yes it works this way .Such that if a certain word is present in a document or not. The model is preferred where we are not concerned about the frequency of the word and the only thing that matters is whether the word is present or not as simple as that. This model is widely used in document classification.

Multinomial: Multinomial Naïve Bayes classifier is used when the data is multinomial distributed basically depending upon the frequency of each word in every text or document. It's primarily used in document classification problems like to classify a particular document into categories such as Sport, Politics, Education, etc.This classifier uses word frequency for predictors.

In the project we are using the Multinomial Naïve Bayes Model.

MultinomialNB is basically the implementation of naïve Bayes algorithm for multinomially distributed data.

It is one of the 3 classical naive Bayes variants as shown above that is widely used in text classifications (where data is represented as word vector counts or Bag Of words (Bow), and also TF-IDF vector also get the privilege to run on this model). The distribution is essentially perfomed parametrized by vectors θy=(θy1,…,θyn) for each class y,(here we have just 2 classes i.e. 0-ham & 1-spam)where n implies the no. of features (i.e. it is the size of vocabulary) & θyi is P(xi|y) of feature i and it simply means that n is equal to the number of unique words in the incoming text also the power of each P(xi|y) is equal to the number of times the particular word is present in the incoming text .

The parameters θy is calculated by the formula stated below it uses a special parameter whose use is stated below:

$$\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n} \tag{7}$$

where Nyi=∑x∈Txi denotes the Frequency or the no. of times the word i is appearing in a particular class, and Ny=∑i=1nNyi denotes the total number of words present in a particular class y.

The α is an important parameter here as it prevents a particular probability of becoming 0 , this is important because there might be case where the whole probability is becoming 0 due to absence of some word. Initially multinomial uses α=1 and is also called Laplace smoothing .

1. Bag of Words Model (BoW)

This is one of the most used and efficient model for vectorization that is converting words/strings to a unique number that can be recognised by the compiler .The Bag of words (BOW) model is the simple form of text representation in numbers..

2. Term Frequency-Inverse Document Frequency Model (TF-IDF)

It is the upgraded version of bag of words model vectorizer where it is related to the importance of a particular word in a particular text compared with the whole document .Below is demonstrated how importance for every word in every text is calculated.

Term Frequency (TF)

It is the Measure of how frequent a term 't', appears in a document/text . Here n is the no. of times the term 't' comes in each text/doc. Thus, each and every text term of each and every text would have its own Term Frequency value .

$$tf_{t,d} = \frac{n_{t,d}}{Number\ of\ terms\ in\ the\ document} \tag{8}$$

Inverse Document Frequency (IDF)

IDF is the Measure of how important or significant a term in a docunment is. This when combined with the above calculated term frequency gives the overall importance of each word in each text.

$$idf_t = \log \frac{number\ of\ documents}{number\ of\ documents\ with\ term\ 't'} \tag{9}$$

$$(tf\_idf)_{t,d} = tf_{t,d} * idf_t \tag{10}$$

Bag of words just creates a set of string to number convert vectors containing count of word occurrences in all text messages , while the TF-IDF model being the higher version of the latter brings focus on information on the more important words as well as the less important words.

V. RESULTS AND ANALYSIS

Here we will show the results that we got for our 2 models using the classification report.

Bag of Words (Bow) Model

```
Accuracy->  0.9866017052375152

Precision->  0.9874686716791979

f1-Score->  0.986232790988736
```

The ROC curve and AUC score:

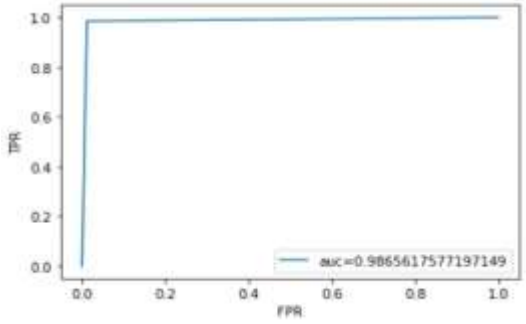

Fig 4. ROC curve-BOW model

We can see that we got pretty good results with an accuracy of 98.66 % hence the model is pretty good for email spam detection.

Term Frequency-Inverse Document Frequency (TF-IDF) Model

```
Accuracy->  0.9841656516443362

Precision->  0.9874055415617129

f1-Score->  0.9836888331242158
```

The ROC curve and AUC score:

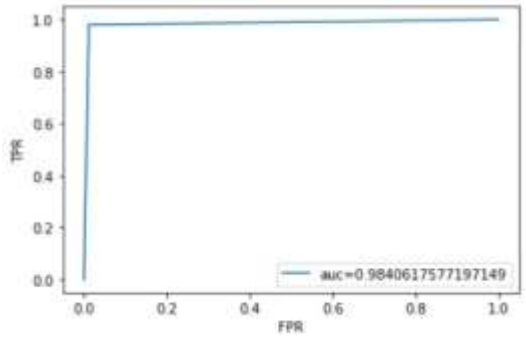

Fig 5. ROC curve-TF-IDF model

This model also showed great results also with an accuracy of 98.41 although the predicted correct results were a slightly less than the bag of words model it still is considered more better model due to functionality of giving higher value to the more important words in the dataset.

TABLE 2: FINAL RESULTS FOR EACH MODEL

PERFORMANCE METRIC / MODEL USED	Accuracy	Precision	F1-Score
BoW	0.986601	0.987468	0.986232
TF-IDF	0.984165	0.987405	0.983688

VI. CONCLUSION

With the help of different libraries, functions and algorithms we were able to perform email spam detection i.e. to classify an incoming mail as spam or ham along with this we got very good results on our testing dataset.

We used performance measures like accuracy, precision, F1-score and AUC score to test our model and got an accuracy around 98% ,we got good results but we can work further on bigger dataset and try to improve the model further. Some of the things we can do is make pre-processing like we can work upon the alpha value that is been used in the multinomial naïve bayes classifier all of this can be worked upon to make the model more efficient on both small and big datasets that can be implemented in real world. Further we can also use the boosting technique in machine learning to remove weak classifier learning functions and replace them with strong classifier one's which is basically to reduce bias in the model.

ACKNOWLEDGMENT

We are extremely grateful to Dr Manoj Kumar for providing us with his supervision and guidance throughout the project. He has been our guiding light and motivated us to complete our project. We would also like to extend sincere gratitude to all our department faculty members for their assistance and unwavering support in making this project successful. We would also like to thank our parents to motivate us during the tough times when we encountered any problem and providing us moral support.

REFERENCES

[1] Priti Sharma,Uma Bhardwaj"Machine Learning based Spam E-Mail Detection"Department of Computer Science & Applications, Maharshi Dayanand University. International Journal of Intelligent Engineering and Systems, Vol.11, No.3, 2018

[2] T. S. Guzella, and W. M. Caminhas, "A review of machine learning approaches to spam filtering", Expert Systems with Applications, Vol.36, No.7, pp.10206-10222, 2009.

[3] V. Metsis, I. Androutsopoulos and G. Paliouras (2006). Spam filtering with Naive Bayes – Which Naive Bayes? 3rd Conf. on Email and Anti-Spam (CEAS)

[4] H.Zhang (2004).The optimality of Naive Bayes. Proc. FLAIR

[5] Thashina Sultana , K A Sapnaz , Fathima Sana , Jamedar Najath, 2020, Email based Spam Detection, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 09, Issue 06 (June 2020),

[6] Jáñez-Martino, F., Alaiz-Rodríguez, R., González-Castro, V. et al. A review of spam email detection: analysis of spammer strategies and the dataset shift problem. Artif Intell Rev (2022).

[7] F. -J. Yang, "An Implementation of Naive Bayes Classifier," 2018 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 2018, pp. 301-306, doi: 10.1109/CSCI46756.2018.00065.

[8] N. Kumar, S. Sonowal and Nishant, "Email Spam Detection Using Machine Learning Algorithms," 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2020, pp. 108-113, doi: 10.1109/ICIRCA48905.2020.9183098.

[9] P. Liu and T. -S. Moh, "Content Based Spam E-mail Filtering," 2016 International Conference on Collaboration Technologies and Systems (CTS), Orlando, FL, USA, 2016, pp. 218-224, doi: 10.1109/CTS.2016.0052.

[10] S. Suryawanshi, A. Goswami and P. Patil, "Email Spam Detection : An Empirical Comparative Study of Different ML and Ensemble Classifiers," 2019 IEEE 9th International Conference on Advanced Computing (IACC), Tiruchirappalli, India, 2019, pp. 69-74, doi: 10.1109/IACC48062.2019.8971582.

[11] A. Karim, S. Azam, B. Shanmugam, K. Kannoorpatti and M. Alazab, "A Comprehensive Survey for Intelligent Spam Email Detection," in IEEE Access, vol. 7, pp. 168261-168295, 2019, doi: 10.1109/ACCESS.2019.2954791.

[12] G. Singh, B. Kumar, L. Gaur and A. Tyagi, "Comparison between Multinomial and Bernoulli Naïve Bayes for Text Classification," 2019 International Conference on Automation, Computational and Technology Management (ICACTM), London, UK, 2019, pp. 593-596, doi: 10.1109/ICACTM.2019.8776800.

[13] Y. Huang and L. Li, "Naive Bayes classification algorithm based on small sample set," 2011 IEEE International Conference on Cloud Computing and Intelligence Systems, Beijing, China, 2011, pp. 34-39, doi: 10.1109/CCIS.2011.6045027.

[14] W. A. Qader, M. M. Ameen and B. I. Ahmed, "An Overview of Bag of Words;Importance, Implementation, Applications, and Challenges," 2019 International Engineering Conference (IEC), Erbil, Iraq, 2019, pp. 200-204, doi: 10.1109/IEC47844.2019.8950616.

[15] D. Rani, R. Kumar and N. Chauhan, "Study and Comparision of Vectorization Techniques Used in Text Classification," 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2022, pp. 1-6, doi: 10.1109/ICCCNT54827.2022.9984608.

[16] C. -z. Liu, Y. -x. Sheng, Z. -q. Wei and Y. -Q. Yang, "Research of Text Classification Based on Improved TF-IDF Algorithm," 2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE), Lanzhou, China, 2018, pp. 218-222, doi: 10.1109/IRCE.2018.8492945.

[17] P. V. Raja, K. Sangeetha, G. SuganthaKumar, R. V. Madesh and N. K. K. Vimal Prakash, "Email Spam Classification Using Machine Learning Algorithms," 2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS), Coimbatore, India, 2022, pp. 343-348, doi: 10.1109/ICAIS53314.2022.9743033.

[18] N. Govil, K. Agarwal, A. Bansal and A. Varshney, "A Machine Learning based Spam Detection Mechanism," 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2020, pp. 954-957, doi: 10.1109/ICCMC48092.2020.ICCMC-000177.

[19] V. S. Vinitha and D. K. Renuka, "Performance Analysis of E-Mail Spam Classification using different Machine Learning Techniques," 2019 International Conference on Advances in Computing and Communication Engineering (ICACCE), Sathyamangalam, India, 2019, pp. 1-5, doi: 10.1109/ICACCE46606.2019.9080000.

[20] A. Junnarkar, S. Adhikari, J. Fagania, P. Chimurkar and D. Karia, "E-Mail Spam Classification via Machine Learning and Natural Language Processing," 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, 2021, pp. 693-699, doi: 10.1109/ICICV50876.2021.9388530.