Detection of Spam Mails using ML Algorithms

G. Nagavallika 1*, K. Krishna Surya [2], S. Triveni Chandrika [3], P.J.L. Valli Devi [4],
N. Sandeep Raj [5], M. Raja Sekhara Manikanta

Sri Vasavi Engineering College, Jawaharlal Nehru Technological University, Tadepalligudem

nagavallika.cse@srivasaviengg.ac.in ,002krishnasurya@gmail.com, sattitrivenichandrika@gmail.com,

vallidevi076@gmail.com ,sandeeprajnammi222@gmail.com, manikantameduri9@gmail.com

ABSTRACT: In today's digital landscape, email serves as a primary mode of communication. However, the proliferation of spam, unwanted and often irrelevant messages, presents challenges. To address this, various algorithms and filters has been created, however they struggle to keep pace with evolving spamming techniques. This paper proposes a method using binary and continuous probability distributions to build a spam filter, employing algorithms like Naive Bayes and Decision Trees. It also examines how overfitting affects Decision Tree accuracy, shedding light on the nuances of spam classification. By investigating the intricate balance between precision and generalization in spam detection, this research aims to enhance the efficacy of email filtering systems, ensuring that users receive only relevant and legitimate correspondence.

Keywords: Email, Machine learning, Algorithms, SVM, Naïve Bayes, Random Forest, Natural Language Tool Kit.

I.INTRODUCTION:

Email spam, also known as electronic mail spam, entails the sending of unsolicited or promotional mails to a set of groups of recipients without their prior consent. The prevalence of spam has surged over the past decade, presenting a significant challenge on the internet. It not only consumes storage space but also wastes time and slows down message delivery. While automatic email filtering has been touted as an effective spam detection method, spammers have devised ways to circumvent these filters effortlessly. In response, ML methods are currently being increasingly employed for spam detection, with text analysis and domain-based blacklists among the prominent approaches. However, reliance solely on content examination for filtering poses challenges, risking the inadvertent rejection of legitimate messages. Traditional approaches like blacklists, which prevent emails originating from particular domains or email addresses addresses, have become less effective as spammers employ newer domains. Alternatively, whitelisting involves prioritizing emails from trusted domains while relegating others to a lower priority queue, which are delivered only upon sender confirmation.

The distinction between "spam" and "ham," as defined by Wikipedia, lies in the solicited nature of messages. Spam refers to unsolicited bulk messages, including mass advertisements and malicious links, whereas "ham" denotes mails that are legitimate and are generally desired by recipients. ML methodologies offer enhanced efficiency by utilizing pre-classified datasets used for training, enabling the application of various algorithms such as Naïve Bayes, neural networks ,support vector machines and random forests for the purpose of email filtration.

Moreover, the advancement in ML algorithms have enabled the development of more sophisticated spam detection models, capable of adapting to evolving spamming techniques. These models leverage large datasets to improve accuracy and robustness in identifying spam emails while minimizing false positives. Additionally, collaborative filtering techniques, such as community-based spam detection, harness collective intelligence to enhance the effectiveness of spam filters. Despite these advancements, the cat-and-mouse game between spammers and filter developers continues, underscoring the need for ongoing innovation in spam detection technologies

Furthermore, the increasing reliance on email for both personal and professional communication underscores the importance of mitigating the impact of spam. Beyond the nuisance it poses to individual users, spam can have detrimental effects on businesses, including loss of productivity and damage to reputation. Therefore, the development of robust and adaptive spam filtering solutions remains a critical area of research and development in the field of cybersecurity. By leveraging machine learning and collaborative filtering techniques, alongside traditional methods, it is possible to create more effective defenses against the ever-evolving threat of email spam.

II. LITERATURE SURVEY:

Numerous studies have investigated the efficacy of various machine learning (ML) algorithms and ensemble classifiers for email spam detection. Suryawanshi et al. (2019) conducted an observational comparative analysis, evaluating different machine learning and combined classifiers to discern their effectiveness in spam detection. Similarly, Karim et al. (2019) provided an extensive review of smart spam email identification methods, shedding light on the evolving landscape of spam detection techniques. Agarwal and Kumar (2018) combined methodology utilizing Particle Swarm Optimization and Naïve Bayes for the detection of email spam offering insights into the potential synergy between different methodologies detection

Furthermore, Harisinghaney and colleagues (2014) investigated spam email classification of spam emails containing text and images algorithms such as KNN, Naïve Bayes, as well as Reverse DBSCAN, contributing to the understanding of diverse classification techniques. Additionally, Mohamad and Selamat (2015) evaluated the effectiveness of combined feature selection methods in unwanted mails classification, highlighting the importance of feature selection within enhancing classification accuracy. These studies collectively provide a rich body of literature that informs the development and refinement of spam detection systems, offering valuable insights into the strengths and limitations of different approaches.

III. PROBLEMS EXISTING IN CURRENT SYSTEM:

A) ***Ineffective Sifting Strategies***: Conventional strategies like boycotting and whitelisting are getting to be less successful as spammers discover ways to balk them utilizing modern spaces or by mirroring genuine senders. This comes about in a higher volume of spam emails coming to users' inboxes.

B) ***Limited Versatility:*** Existing channels frequently battle to adjust to advancing spamming methods. As spammers ceaselessly enhance, conventional channels depending exclusively on substance examination may miss modern sorts of spam, driving to an increment in untrue negatives.

C) ***Overreliance on Substance Examination:*** Numerous current spam channels depend intensely on substance examination, which postures challenges in precisely recognizing between spam and true-blue messages. This can lead to the inadvertent sifting out of vital emails, causing disappointment for users.

D) ***Productivity and Notoriety Affect:*** Spam not as it were squanders users' time but can too have negative impacts on businesses, counting misfortune of efficiency and harm to notoriety. Wasteful spam sifting arrangements worsen these issues.

Some of them are:

1. Ineffectual Sifting Strategies: Conventional strategies like boycotting and whitelisting are getting to be less viable as spammers discover ways to balk them utilizing unused spaces or by imitating true blue senders. This comes about in a higher volume of spam emails coming to users' inboxes.

2. Restricted Versatility: Existing channels frequently battle to adjust to advancing spamming methods. As spammers persistently improve, conventional channels depending exclusively on substance examination may miss modern sorts of spam, driving to an increment in untrue negatives.

3. Overreliance on Substance Examination: Numerous current spam channels depend intensely on substance examination, which postures challenges in precisely recognizing between spam and authentic messages. This can lead to the inadvertent sifting out of imperative emails, causing dissatisfaction for users.

4. Efficiency and Notoriety Affect: Spam not as it were squanders users' time but can moreover have hindering impacts on businesses, counting misfortune of efficiency and harm to notoriety. Wasteful spam sifting arrangements worsen these issues.

IV. PROPOSED STATEMENT:

To tackle the issue confinements of current spam sifting frameworks, This paper suggests a new method employing ML calculations, particularly Naïve Bayes and Choice Trees, nearby twofold and persistent likelihood conveyances. By leveraging these methods, the proposed framework points to improve the viability of spam location by:

1. Making strides Flexibility: The utilize of machine learning calculations empowers the framework to adjust to advancing spamming strategies by learning from huge datasets. This upgrades the system's capacity to precisely recognize unused sorts of spam, lessening untrue negatives.

2. Adjusting Exactness and Generalization: The framework looks at the complex adjust between exactness and generalization in spam location, guaranteeing that true blue messages are not incidentally sifted out whereas successfully distinguishing spam.

3. Decreasing Wrong Positives: By consolidating collaborative sifting methods and leveraging collective insights, the framework points to minimize untrue positives, hence progressing client involvement and productivity.

4. Upgrading In general Effectiveness: The proposed framework not as it were points to move forward spam location exactness but moreover points to improve generally productivity in mail communication by diminishing the volume of spam coming to users' inboxes.

Through the integration of progressed machine learning methods and collaborative sifting strategies, the proposed framework tries to give a strong and versatile arrangement to the ever-evolving challenge of mail spam, subsequently moderating its affect on clients and businesses alike.

V. METHODOLOGY:

A. Data Preprocessing:

When conducting data analysis, it's common to encounter extensive datasets comprising various formats and structures. Data inherently exists in diverse forms, including images, vedio files, audio recordings, and structured tables. Machines interpret data as binary code, consisting of 1s and 0s. Classic classifiers are employed for data classification, involving the process of analyzing data to identify significant classes.

These classifiers or models are developed to predict class labels, such as determining the risk level of a loan application. Data classification involves two primary steps:

B. Data Cleaning:

Data cleaning encompasses tasks such as handling missing values, noise reduction, outlier detection or removal, and addressing inconsistencies. It also involves data integration, which entails combining multiple databases, data records, or datasets. Additionally, data transformation is undertaken to standardize and normalize data to a particular scale. Data reduction aims to generate a concise overview of the given dataset while maintaining analytical accuracy.

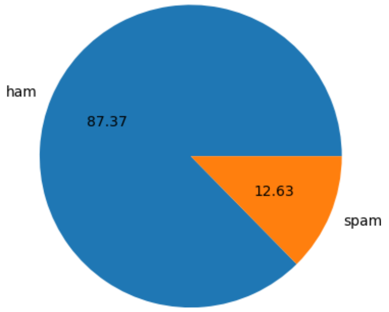
1. Stop Words:

Stop words refer to common English terms that contribute minimally to the overall meaning of a sentence. These words are able to reliably disregarded without altering the comprehension of the statement. For example, an inquiry such as "How to prepare a vegetarian cheese sandwich," search engines typically include words like "how," "to," "make," "a," "veg," "cheese," and "sandwich" in their search parameters. However, removing or disregarding these common words allows the search tool to focus for accessing webpages containing keywords like "veg," "cheese," and "sandwich," thereby yielding more relevant results.

2. Tokenization:

Tokenization involves dividing a continuous stream of text into coherent components, such as phrases, symbols, or words, known as tokens. These tokens are then employed for subsequent analysis, such as text parsing and mining. Here,Tokenization plays a valuable role in both semantics, aiding in text segmentation, and lexical analysis in computer science and engineering. Defining what constitutes a "word" can sometimes be challenging, as tokenization typically occurs at the word level. Tokenization enables the breakdown of text into logical units, facilitating various text processing tasks and enhancing language understanding in computational contexts.

3. Stemming:



Stemming in ML is a text processing technique used to reduce words to their root or base form. It aims to normalize words by removing affixes, such as prefixes and suffixes, to enhance text analysis and feature extraction.

By stemming words, variations of the same word are treated as identical, reducing vocabulary size and improving model performance. Stemming plays a vital role in inquiries such as text information and classification and retrieval.

B. CLASSIC CLASSIFIERS:

Classification in data analysis involves extracting models that describe significant data categories. A classifier or model is developed to predict class labels, such as determining the risk level of a loan application. This process consists of two main steps: the learning step, which constructs the classification model, and the classification step, where data is categorized based on the model's predictions.

1.NAÏVE BAYES:

The Naïve Bayes Classifier emerged as a pioneering tool for spam detection in 1998. This algorithm, falling under the domain of supervised learning, operates on the principles of Bayesian inference. By analyzing dependent events and leveraging probabilities derived from past events, it predicts future events. Naïve Bayes relies on the assumption of feature independence, enabling it to effectively classify spam emails based on word probabilities.

$$P(B) = \sum_{y} P(B|A)P(A)$$

2.SUPPORT VECTOR MACHINE:

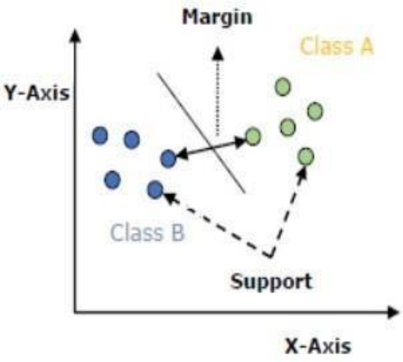


Fig.1. Plot depicting SVM.

The SVM is a prevalent algorithm in Supervised Learning, commonly utilized for solving classification problems in Machine Learning. This approach is rooted in the concept of decision points, with the primary objective of establishing a decisive line or boundary. Through SVM, a hyperplane is generated as the output, facilitating the classification of new samples.

In a two-dimensional setting, this hyperplane acts as a dividing line, separating divides the plane into two separate regions, each corresponding to a different class.

3.RANDOM FOREST:

The Random Forest classifier comprises a group of decision trees, each exhibiting distinctive traits regarding shape and size, forming an ensemble. This technique entails randomly choosing training data during tree construction and choosing subsets of input attributes randomly when splitting nodes.

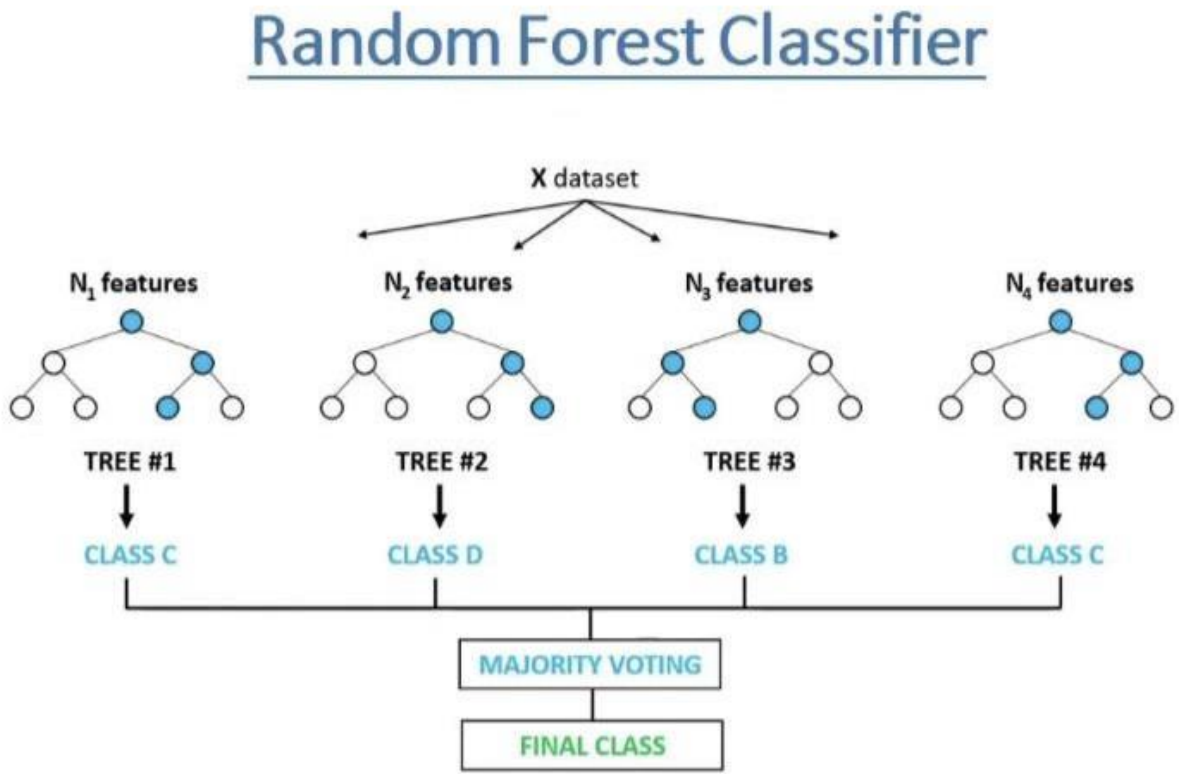


Fig.2.Chart of RFC Model

By introducing randomness, the aim is to decrease correlation among decision trees, thus improving the generalization error of the ensemble. This strategy ensures that features within the trees do not exhibit identical patterns, leading to enhanced model performance.

VI. COMPUTING ALGORITHM:

1.1. Input the file or dataset for testing or training purposes.

1.2. Verify the dataset suitable for compatible encoding.

1.2.1. Whether the encoding is supported, proceed to the step 1.4 on next.

1.2.2. Whether the encoding is not supported, proceed to the step 1.3 on next.

1.3. Convert the texts encoding of provided file to a compatible encoding. After that attempt reading again.

1.4. Choose whether to "Train", "Test", or "Compare" the models utilizing the dataset.

1.4.1. Whether "Train" is selected, proceed to the step 1.5 on next.

1.4.2. Whether "Test" is selected, proceed to the step 1.6 on next.

1.4.3. Whether "Compare" is selected, proceed to the step 1.7 on next.

1.5. Whether "Train" is chosen:

1.5.1. Choose the classifier for training with the dataset provided.

1.5.2. Verify for the missing values (NANs) and duplicates.

1.5.3. Determine optimal values through Hyperparameter Optimization.

1.5.4. Prepare the Correct text for feature transformation.

1.5.5. model has to be trained here.

1.5.6. Save the Features and model and display the accurate results.

1.5.7. Choose the classifier to testing with the dataset provided.

1.5.8. Verify for duplicates and null values.

1.5.9. Load the stored model and saved features during the training phase.

1.5.10. Utilize the values that have been loaded to test the latest dataset.

1.5.11. Display the results.

1.6. If "Compare" is chosen :

1.6.1. Assess all classifiers by utilizing the provided dataset.

1.6.2. Display the outcomes of the classifiers.

VII.IMPLEMENTATION:

The implementation pertaining to the model carried out on the Anaconda Jupiter notebook platform, with a dataset sourced from the Kaggle website serving during the training phase dataset. The provided dataset undergoes initial checks for null values and duplicates to enhance machine performance. Subsequently, the dataset is split into two subsets, specifically the "train dataset" and the "test dataset," with a split ratio of 70:30. These subsets are then subjected to text-processing, wherein punctuation marks and stop words will be eliminated to yield clean words.

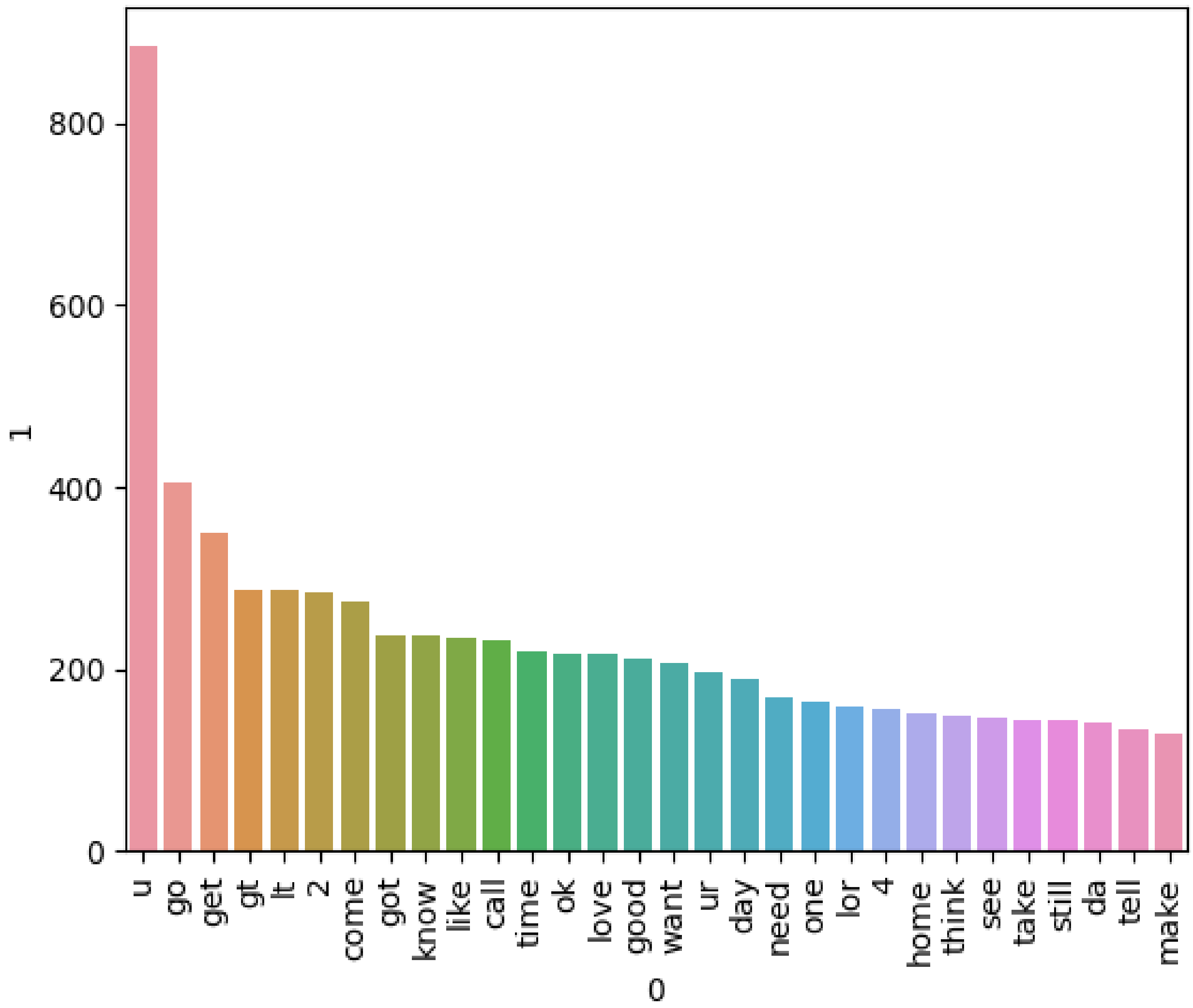


Fig.3.Plotting Spam Words for Classification

The resulting clean words undergo "Feature Transform," where they are used to construct a lexicon for the machines through the 'fit' and 'transform' processes. Additionally, the dataset undergoes "hyperparameter tuning" to determine optimal parameters for the all classifier based on the dataset.

Upon obtaining these values, machine is trained using a random state, and the condition of the features and trained model is preserved for subsequent use in testing unobserved data.The results are displayed using Streamlit.

Given Below is the flowchart for the method used in implementation of the project identification of spam mails.

FLOWCHART:

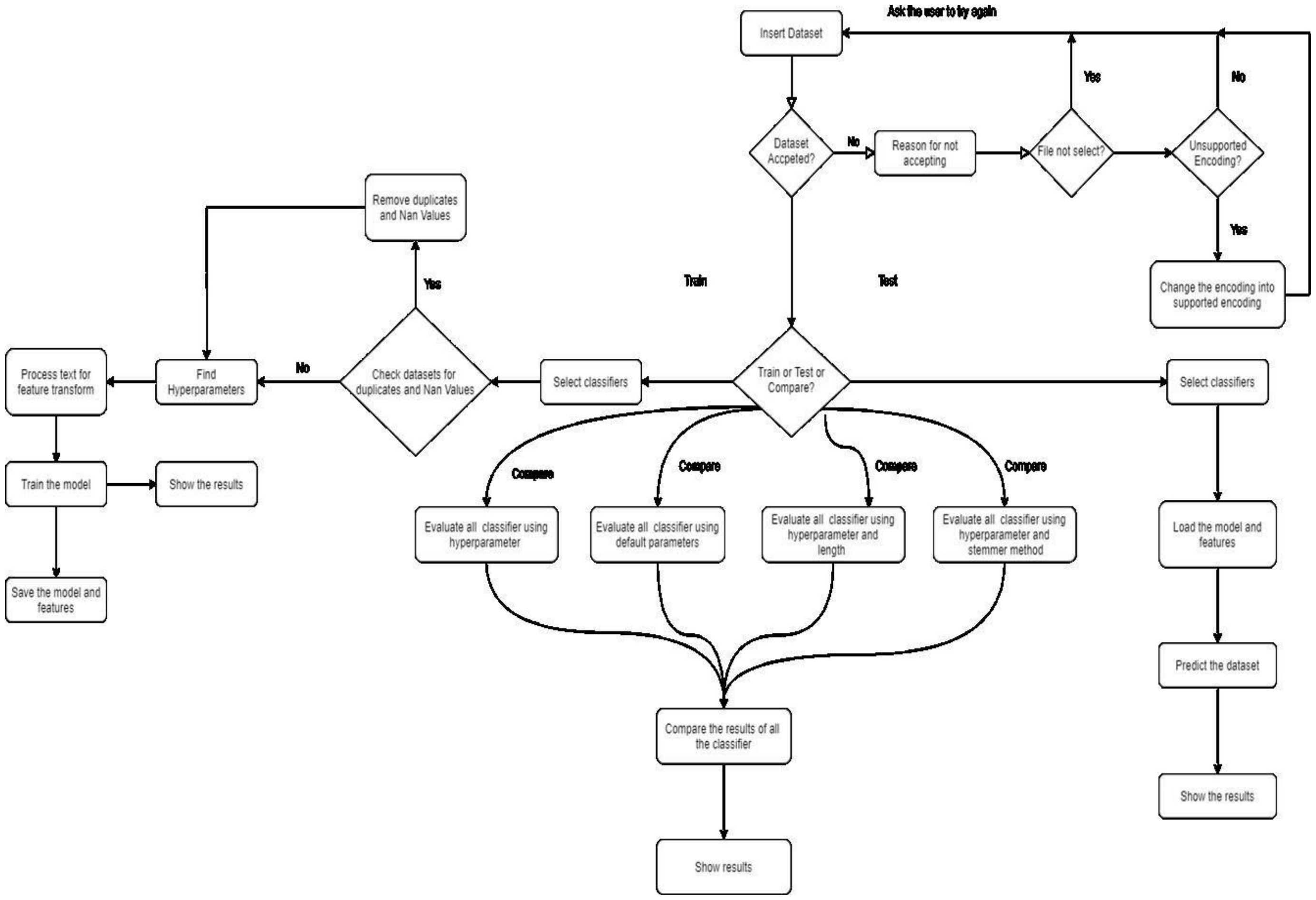

Ask the user to try again
Insert Dataset
Yes
No
Dataset Accpeted?
No
Reason for not accepting
File not select?
Unsupported Encoding?
Yes
Change the encoding into supported encoding
Remove duplicates and Nan Values
Yes
Train
Test
Process text for feature transform
Find Hyperparameters
No
Check datasets for duplicates and Nan Values
Select classifiers
Train or Test or Compare?
Select classifiers
Compare
Compare
Compare
Compare
Train the model
Show the results
Evaluate all classifier using hyperparameter
Evaluate all classifier using default parameters
Evaluate all classifier using hyperparameter and length
Evaluate all classifier using hyperparameter and stemmer method
Load the model and features
Save the model and features
Predict the dataset
Compare the results of all the classifier
Show the results
Show results

Fig.4.Flow Chart of System Architecture

VIII. RESULT ANALYSIS:

The model we've built has undergone training employing various classifiers to assess and compare results for enhanced accuracy. Every classifier provides its evaluated outcomes to the user, enabling comparison to determine if the data has categorized as "spam" or "ham." The outcomes from each and every classifier are presented in tables and graphs for improved comprehension. The dataset of training is sourced from the Kaggle website, specifically named "spam_mail.csv." For testing the machine, which was trained has a separate CSV formatted file containing unseen data, termed "mail_data.csv," has been created. Once text editing is complete, The document is set for the sake of template. Create a duplicate of the required template file using the special Save As function and adhere to the naming format designated under the conference of the title of your paper. import your formatted text file on this newly generated document. You are currently prepared to format your paper using the scroll- down usage of window where it is on left of Micro soft word formatting toolbar.

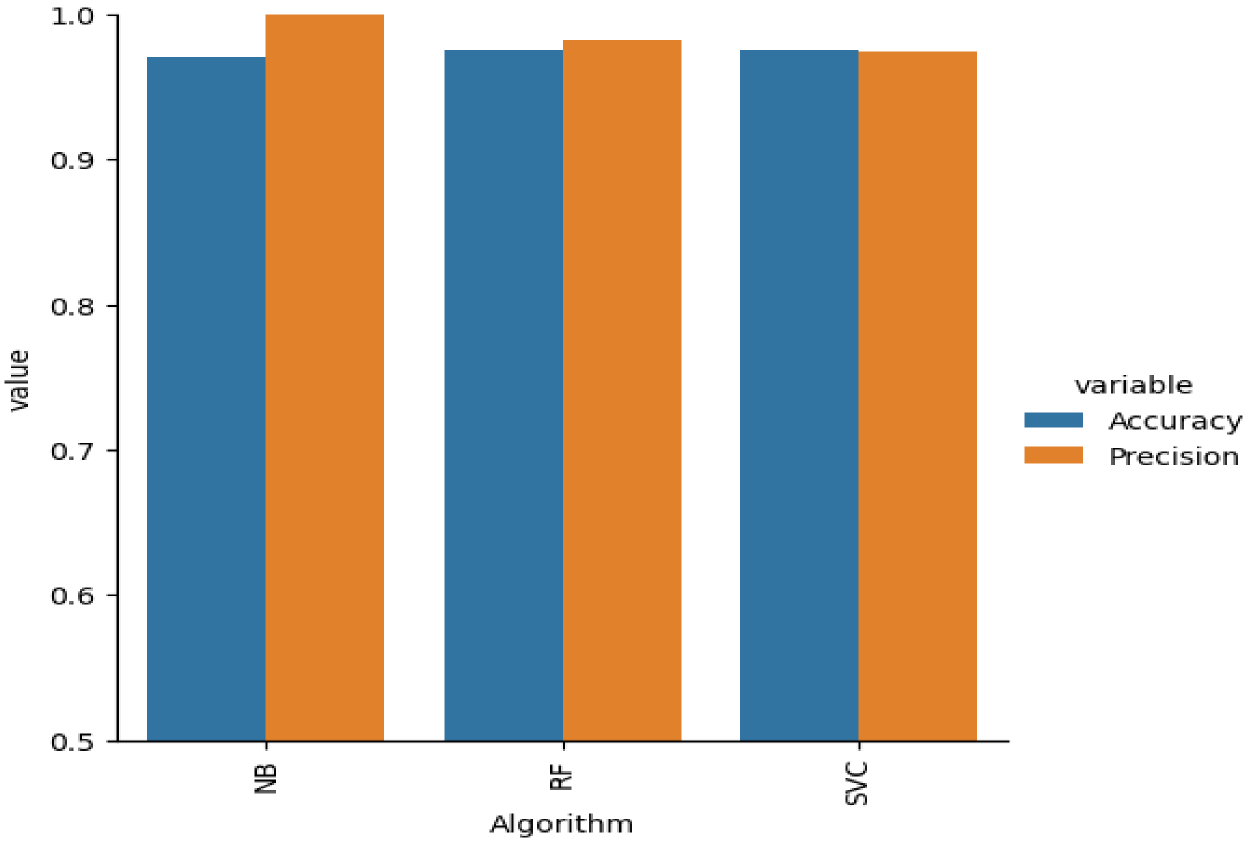


Fig.5.Plot b/w algorithm accuracy and value precision

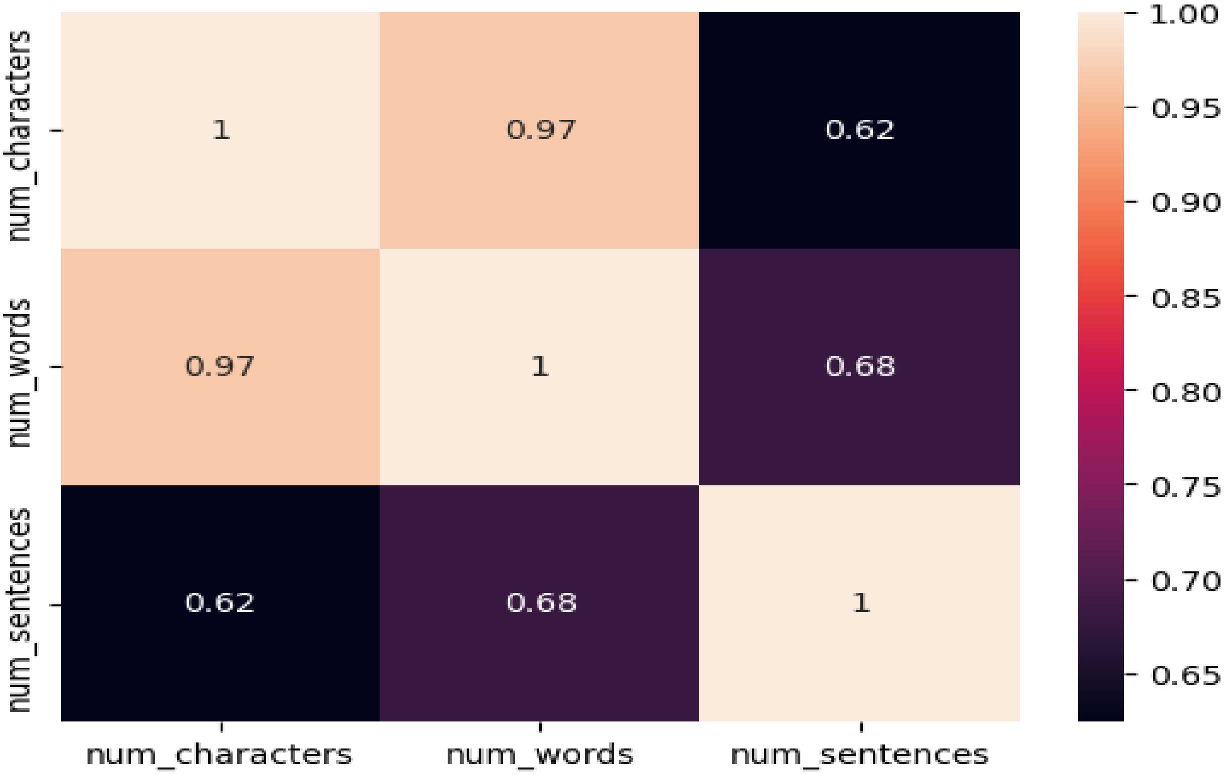


Fig.6. Correlation matrix between no. of characters, no. of words and no. of sentences.

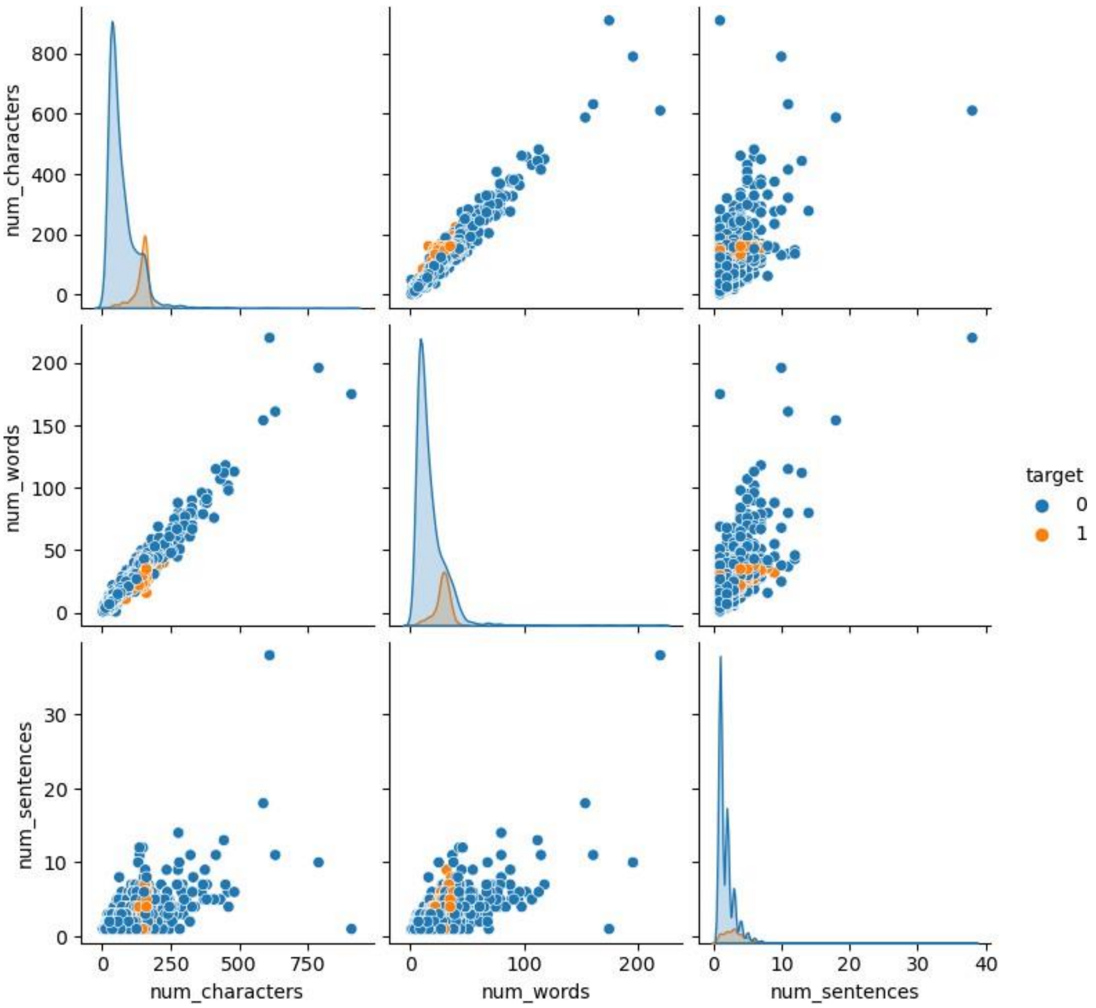


Fig.7.Scatter Plot b/w targeted words and sentences

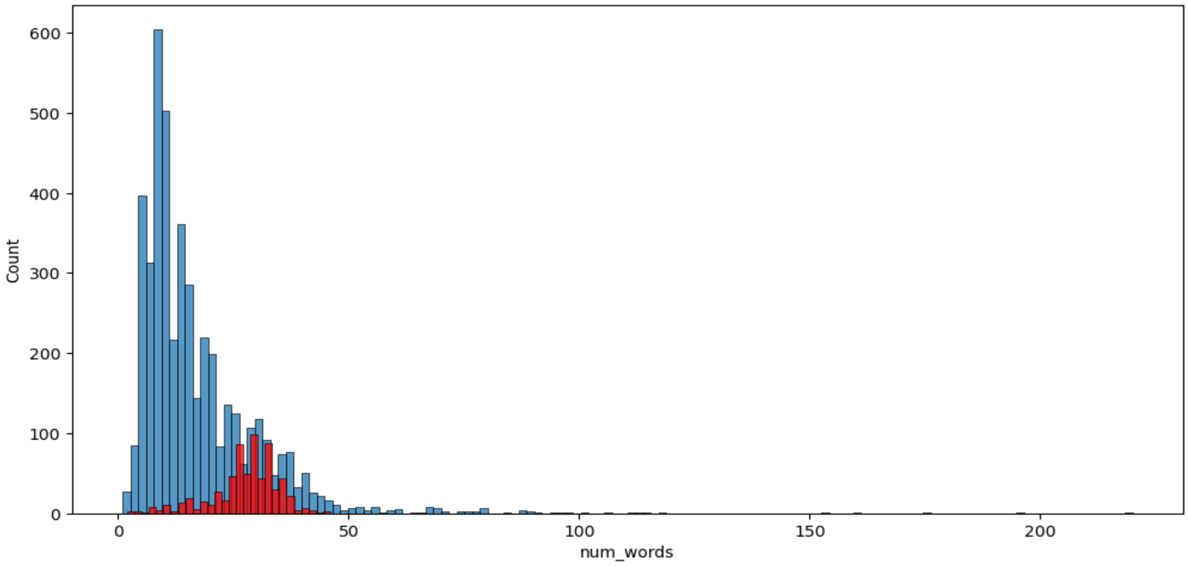


Fig.8.Plot b/w no. of words repeated and its count

IX. CONCLUSION:

Based on these findings, it can be induced that Multinomial Naïve Bayes yields the most favorable comes about, in spite of fact that it is obliged by class-conditional autonomy, driving to intermittent misclassification of tuples. Gathering strategies, be that as it may, have illustrated adequacy by leveraging different classifiers for class forecast. Given the predominance of e-mail communication, our project's testing capabilities are constrained by the estimate of the corpus. As a result, our spam discovery system relies exclusively on e- mail substance or maybe than space names or other criteria, hence displaying a restricted see of the mail body. There is plentiful room for enhancement in our venture. Potential upgrades include:

"Effective spam mail classification is basic for precisely categorizing emails and recognizing between spam and true-blue messages."

This strategy can be received by huge organizations to channel approaching emails and prioritize the conveyance of desired messages.

X. REFERENCES:

1. Suryawanshi, Shubhangi & Goswami, Anurag & Patil, Pramod.(2019). Email Spam Detection: An Empirical Comparative Study of Different ML and Ensemble Classifiers. 69-74. 10.1109/IACC48062.2019.8971582.
2. Karim, A., Azam, S., Shanmugam, B., Krishnan, K., & Alazab, M. (2019). A Comprehensive Survey for Intelligent Spam Email Detection. *IEEE Access*, *7*, 168261-168295. [08907831]. https://doi.org/10.1109/ACCESS.2019.2954791
3. K. Agarwal and T. Kumar, "Email Spam Detection Using Integrated Approach of Naïve Bayes and Particle Swarm Optimization," 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2018, pp. 685-690.
4. Harisinghaney, Anirudh, Aman Dixit, Saurabh Gupta, and Anuja Arora. "Text and image-based spam email classification using KNN, Naïve Bayes and Reverse DBSCAN algorithm." In Optimization, Reliabilty, and Information Technology (ICROIT), 2014 International Conference on, pp 153 -155. IEEE, 2014
5. Mohamad, Masurah, and Ali Selamat. "An evaluation on t he efficiency of hybrid feature selection in spam email classification." In Computer, Communications, and Control Technology (I4CT), 2015 International Conference on, pp 227 -231. IEEE, 2015