TITLE PAGE

Email Classification to detect the spam emails using Novel Naive Bayes Algorithm And Comparing with Decision Tree Algorithm to improve Accuracy.

K.Mahendra[1], Mahesh[2]

K.Mahendra[1]
Research Scholar,
Department of Computer Science and Engineering,
Saveetha School of Engineering,
Saveetha Institute of Medical and Technical Sciences,
Saveetha University, Chennai, Tamil Nadu, India. Pincode: 602 105
kadurumahendra1424.sse@saveetha.com

C. Mahesh[2]
Research Guide, Corresponding Author
Department of Computer Science and Engineering,
Saveetha School of Engineering,
Saveetha Institute of Medical and Technical Sciences,
Saveetha University, Chennai, Tamil Nadu, India. Pincode: 602 105
maheshc.sse@saveetha.com

Keywords: Machine Learning, Supervised Learning, Spam detection, Ham, Novel Naive Bayes Classifier, Logistic Regression.

ABSTRACT

AIM: The proposed work focuses on Email Classification to detect the spam emails by using Novel Naive Bayes Algorithm on a dataset compared with the Decision Tree Algorithm.**MATERIALS AND METHODS :** Based on Accuracy, Email classification is performed by Novel Naive Bayes Algorithm (N = 21) of sample size and Decision tree Algorithm of sample size (N = 21) . **RESULTS:**Novel Naive Bayes Algorithm has the accuracy of (84.667%) which is relatively higher than Decision Tree Algorithm which has the accuracy (84.111%).**CONCLUSION:** Novel Naive Bayes Algorithm has finer accuracy of (84.667%) then Decision Tree Algorithm of Accuracy (84.11111%).

INTRODUCTION

The domain of "Email Classification to detect the spam emails using Novel Naive Bayes Algorithm" is situated within the field of machine learning, specifically in the area of email classification. This project employs common machine learning algorithms like Naive Bayes and decision trees to address the problem of identifying and filtering spam emails. Within the machine learning framework, the primary focus is on classifying emails as either spam or ham messages based on their content and attributes. The algorithms explored are Naive Bayes, which applies Bayesian statistical modeling to text classification, and decision tree, which build rule-based models to categorize emails. This project compares and contrasts the effectiveness of these algorithms for the spam detection task. Applications within this domain include building real-time spam filters to detect unwanted emails and developing personalized classifiers tailored to individual user preferences. The innovation comes from rigorously evaluating different machine learning approaches like Naive Bayes and decision trees on a common email dataset. By comparing multiple algorithms, the optimal email classification model can be determined to accurately separate spam from legitimate emails. This can lead to more robust spam filters and improved email experiences by reducing unwanted messages. Overall, this project provides unique insight into email classification within machine learning by contrasting established algorithms like Naive Bayes and decision trees.

(Desai 2016; Ke 2008) (Sanchez 2021) (Tailby 2007) (Desai 2016) (Hamisu 2021)

This Research paper was published by some of the major sites known as Google Scholar,Springer and IEEE.The study aims to unveil groundbreaking methodologies for detecting spam emails, providing critical insights for cybersecurity decision-making.It inspire innovative approaches for addressing email-related threats.This research serves as a foundation for robust email security strategies, aligning with broader initiatives encouraging users to transition from conventional private email practices to more secure public alternatives.

(Schiavo et al. 2024) (Sanchez 2021; Srinivas, Sucharitha, and Matta 2021) (Suthaharan 2015) (Salehinejad et al. 2023)

The naive Bayes model makes simplifying assumptions that each email attribute contributes independently to the spam classification. This may not fully capture the complex interdependencies in language and writing style that distinguish spam.he algorithm's inherent limitations become apparent when attempting to customize spam detection based on individual preferences or adapting to evolving email threats.The main goal of applying naive Bayes is to quickly classify emails to calculate accuracy.

(Varun Kumar and Ramamoorthy 2022) (Rayan 2022)

MATERIALS AND METHODS

This research initiative is supported by a meticulously curated email dataset designed to facilitate an in-depth analysis of spam email detection patterns, temporal influences on spam occurrences, and diverse content-related factors. The dataset comprises a wide array of emails originating from different sources and directed towards varied recipients, providing a nuanced exploration of spam conditions across various communication channels. Noteworthy is the inclusion of metadata, such as sender and recipient details, enabling effective categorization and analysis based on the email's origin and destination, thereby enhancing the dataset's applicability to diverse communication contexts.

(Hershkop 2006) (Kaddoura et al. 2022)

In capturing temporal aspects, precise timestamps for each email are recorded, offering a detailed understanding of spam patterns across specific time intervals. This temporal granularity is pivotal for discerning patterns that may fluctuate based on factors like the time of day, week, or other temporal considerations. Ensuring data consistency, the dataset adheres to standardized methods for preprocessing and feature extraction, establishing a reliable foundation for analyzing the impact of different content-related factors on spam dynamics. This dataset stands as a valuable asset for researchers and practitioners seeking to unravel intricate relationships within email data, providing insights into effective spam detection strategies and bolstering email security measures.

(Cormack 2008)

Google Colab, a cloud-based platform, serves as an invaluable tool for dataset processing and analysis. It is accessed through a Google account, which allows users to create Jupyter Notebook-like environments for Python code execution, seamlessly integrating with Google Drive. It supports popular libraries like pandas and matplotlib, users can effortlessly load datasets, conduct exploratory data analysis, and implement data cleaning and preprocessing steps. Its interactive environment facilitates real-time collaboration and efficient sharing of insights.Its ability to harness GPU and TPU resources is particularly beneficial for machine learning tasks, enabling the training of models with enhanced efficiency. Furthermore, the platform's collaborative features and ease of sharing make it a robust solution for collaborative data science projects.

Algorithm 1

NAIVE BAYES CLASSIFIER

The Novel Naive Bayes Classifier(Conway and White 2011) is a supervised learning algorithm.It is mainly used in text classification that includes a high-dimensional training dataset. It uses the Bayes theorem to calculate the probability of an event. It follows the properties like strong independence, easily handles large datasets, and depends on probability distribution.

Pseudo Code

Step-1:Initially, all of us have to download and install all the packages and libraries.

Step-2:Import all packages that are downloaded.

Step-3:It needs to load the dataset and extract the ham and spam keywords.

Step-4:Clean the dataset which includes removing single letter words, truncating white spaces,tokenizing each and every message, deleting all punctuations, changing all the letters to lowercase, etc.

Step-5: Then split the dataset into test and train datasets.

Train: X_train, Y_train.

Test: X_test, Y_test

Step-6:Train the machine which is spam and ham when they triggered the spam and ham words.

Step-7: Load the Novel Naive Bayes classifier and train the model with the training dataset.

NB=NaiveBayesClassifier()

NB.fit(X_train,Y_train)

Step-8: Calculate the probability distribution $P(B|A)$ for every class using the Bayes theorem.

Step-9: Calculate the confusion matrix and find the Accuracy.

accuracy = sum(X_test.Label ==Y_ test.predicted)/len(X_test)

Algorithm 2

DECISION TREE ALGORITHM

A decision tree(Ismail et al. 2022) is a flowchart-like tree structure where an internal node represents feature(or attribute), the branch represents a decision rule, and each leaf node represents the outcome. The topmost node in a decision tree is known as the root node. It learns to partition on the basis of the attribute value. It partitions the tree in recursively manner call recursive partitioning. This flowchart-like structure helps you in decision making.

Pseudo Code

Step-1: Download and install the required libraries.

Step-2: Import the necessary libraries.

Step-3: Load the dataset and extract the features and target variable.

Step-4: Now clean the dataset

Step-5: Encode the target variable.

Step-6: Now divide the dataset into a test and train the dataset.

Step-7: Load the Decision Tree model and train the model with the training dataset.

Step-8: Predict the accuracy from the confusion matrix of the test dataset.

Statistical Analysis

The IBM SPSS Version 27 software is used to calculate the statistical variables like mean, standard deviation, standard error mean, mean difference, sig, and F value. The Independent Sample T-Test analysis is carried out in this research.The spam dataset with 4851ham words and 749 spam words is used to detect spam. The dependent variables are spam and ham. The independent variables are accuracy and count of words.

Results

The accuracy is taken as a measurement for comparing the Novel Naive Bayes Classifier and Decision Tree were run at different times in Google Colab with a sample size of 21.Table 1

represents the predicted accuracy and loss of Novel Naive Bayes Classifier .Table 2 represents the predicted accuracy and loss of Decision Tree .These 21 data samples are used for each algorithm along with their loss values to calculate statistical values that can be used for comparison.From the results, it is observed that the mean accuracy of Novel Naive Bayes Classifier was 84.66% and Decision Tree was 84.11%.Table 3 represents mean accuracy values for Naive Bayes Classifier and Decision Tree. Mean value of Naive Bayes Classifier better when compared with theDecision Tree with a standard deviation of 7.260 and 6.043 respectively.Table 4 shows the Independent sample T test data of Naive Bayes Classifier and with the significance value obtained is 0.585 (Two tailed, $p>0.450$). Figure 1 denotes the comparison of Naive Bayes Classifier and Decision Tree in terms of mean accuracy and loss.

Mean, standard deviation and standard error mean for Naive Bayes Classifier are 85.29, 7.260 and 1.584 respectively. Similarly for Decision Tree, the mean, standard deviation and standard error mean are 83.71, 6.043 and 1.319 respectively. On the other hand, the loss values of Naive Bayes Classifier for mean, standard deviation and standard error mean are b11, b12 and b13 respectively. For the Decision Tree, the loss values of K-Means Clustering for mean, standard deviation and standard error mean are b21, b22 and b23 respectively.

The group statistics value along with mean, standard deviation and standard error mean for the two algorithms are also specified. The graphical representation of comparative analysis, means of loss between two algorithms of Naive Bayes Classifier and Decision Tree are classified. This indicates that Naive Bayes Classifier is significantly better with 84.667% accuracy when compared with Decision Tree classifier accuracy of 84.111%.

DISCUSSION

In the given study, the significance value obtained is 0.001 (Two tailed, $p<0.05$) which implies that Naive Bayes Classifier appears to be better than Decision Tree. Accuracy analysis of the Naive Bayes Classifier is analyzed as 84.667% whereas the accuracy ofdecisionTree is 84.111%.

In this research, a novel Naive Bayes algorithm is developed and evaluated for classifying emails as either spam or ham messages. The performance of this novel Naive Bayes approach is compared to a Decision Tree algorithm to determine if accuracy can be improved in detecting spam emails. Both the Naive Bayes and Decision Tree models are trained and tested on an email dataset that has been preprocessed. Quantitative performance measures such as accuracy, precision, recall, and F1-score are calculated using a confusion matrix for both models.

The Naive Bayes algorithm has certain advantages for email classification, such as fast model training and testing, the ability to work with categorical input features, and independence assumptions between features (Cichosz 2015). However, a limitation is the zero-frequency problem when dealing with sparse datasets (Trivedi et al. 2016). In contrast, Decision Tree models can handle interactions between features but may overfit noisy datasets. By comparing both models, this research aims to develop an optimal approach for spam detection that balances accuracy, training time, and generalization.

There are several limitations in developing an accurate spam detection system that need to be addressed in future research. One challenge is effectively extracting meaningful textual features and analyzing embedded links to differentiate ham from spam. Additionally, avoidance of misclassifying legitimate emails as spam is important to limit false positives. The models developed currently also require more comprehensive spam keyword dictionaries and methods to handle evolving trickery in spam emails over time

CONCLUSION

In conclusion, implementing the accuracy of the Novel Naive Bayes classifier is more when compared to the Decision Tree model in detecting spam messages.The accuracy value of the Naive Bayes Classifier is 84.667% whereas the accuracy value of Decision Tree is 84.111%. Based on the analysis, Naive Bayes Classifier (84.667%) performs better than Decision Tree (84.111%)

DECLARATIONS

Conflicts of Interests

No conflict of interest in this manuscript.

Authors Contribution

Author KM was involved in data collection, data analysis and manuscript writing. Author CM was involved in conceptualization, data validation and critical reviews of manuscripts.

Acknowledgements

The authors would like to express their gratitude towards Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences (Formerly known as Saveetha University) for providing the necessary infrastructure to carry out this work successfully.

Funding: We thank the following organizations for providing financial support that enabled us to complete the study.

1. Infosys solutions, Chennai
2. Saveetha University
3. Saveetha Institute of Medical and Technical Sciences
4. Saveetha School of Engineering

REFERENCES

1. Cichosz, Pawel. 2015. *Data Mining Algorithms: Explained Using R*. John Wiley & Sons.

2. Conway, Drew, and John Myles White. 2011. *Machine Learning for Email: Spam Filtering and Priority Inbox*. "O'Reilly Media, Inc."

3. Cormack, Gordon V. 2008. *Email Spam Filtering: A Systematic Review*. Now Publishers Inc.

4. Desai, Varun. 2016. *A Structure Based Technique for Spam Detection and Email Classification*.

5. Hamisu, Muhammad. 2021. *Classification of Emails for Internet Fraud Detection and Prevention Through the Application of Artificial Intelligence Techniques*.

6. Hershkop, Shlomo. 2006. *Behavior-Based Email Analysis with Application to Spam Detection*.

7. Ismail, Safaa S. I., Romany F. Mansour, Rasha M. Abd El-Aziz, and Ahmed I. Taloba. 2022. "Efficient E-Mail Spam Detection Strategy Using Genetic Decision Tree Processing with NLP Features." *Computational Intelligence and Neuroscience* 2022 (March): 7710005.

8. Kaddoura, Sanaa, Ganesh Chandrasekaran, Daniela Elena Popescu, and Jude Hemanth Duraisamy. 2022. "A Systematic Literature Review on Spam Content Detection and Classification." *PeerJ. Computer Science* 8 (January): e830.

9. Ke, Shih-Wen. 2008. *Automatic Email Classification*.

10. Rayan, Alanazi. 2022. "Analysis of E-Mail Spam Detection Using a Novel Machine Learning-Based Hybrid Bagging Technique." *Computational Intelligence and Neuroscience* 2022 (August): 2500772.

11. Salehinejad, Hojjat, Anne M. Meehan, Parvez A. Rahman, Marcia A. Core, Bijan J. Borah, and Pedro J. Caraballo. 2023. "Novel Machine Learning Model to Improve Performance of an Early Warning System in Hospitalized Patients: A Retrospective Multisite Cross-Validation Study." *EClinicalMedicine* 66 (December): 102312.

12. Sanchez, Jorge. 2021. *Advance-Fee Scam Email Classification Using Machine Learning*.

13. Schiavo, Giuseppina, Francesca Bertolini, Samuele Bovo, Giuliano Galimberti, María

14. Muñoz, Riccardo Bozzi, Marjeta Čandek-Potokar, Cristina Óvilo, and Luca Fontanesi. 2024. "Identification of Population-Informative Markers from High-Density Genotyping Data through Combined Feature Selection and Machine Learning Algorithms: Application to European Autochthonous and Cosmopolitan Pig Breeds." *Animal Genetics*, January. https://doi.org/10.1111/age.13396.

15. Srinivas, Mettu, G. Sucharitha, and Anjanna Matta. 2021. *Machine Learning Algorithms*

and Applications. John Wiley & Sons.

16. Suthaharan, Shan. 2015. *Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning*. Springer.

17. Tailby, Ross. 2007. *Email Classification in a Corporate Environment*.

18. Trivedi, Premal S., Paul J. Rochon, Janette D. Durham, and Robert K. Ryu. 2016. "National Trends and Outcomes of Transjugular Intrahepatic Portosystemic Shunt Creation Using the Nationwide Inpatient Sample." *Journal of Vascular and Interventional Radiology: JVIR* 27 (6): 838–45.

19. Varun Kumar, K., and M. Ramamoorthy. 2022. "Naive Bayes Classifier Algorithm for Spam Detection of Email to Improve Accuracy and in Comparison with Decision Tree Algorithm." *Journal of Pharmaceutical Negative Results*, September, 49–55.

TABLES AND FIGURES

Table1. Accuracy and Loss Analysis of Naive Bayes Classifier

Iterations	Accuracy(%)
1	**79.05**
2	**70.23**
3	**77.12**
4	**85.18**
5	**87.16**
6	**87.41**
7	**89.67**
8	**90**
9	**91.23**
10	**95.41**
11	**79.07**

12	**70.13**
13	**77.24**
14	**85.13**
15	**87.16**
16	**87.19**
17	**89.20**
18	**90.56**
19	**91.72**
20	**91.14**
21	**95.82**

Table2. Accuracy and Loss Analysis of Decision Tree

Iterations	Accuracy(%)
1	**77.14**
2	**72.78**
3	**78.14**
4	**81.19**
5	**84.67**
6	**87.76**
7	**86.54**
8	**89.23**
9	**90.43**
10	**90.32**
11	**77.65**

12	**72.78**
13	**78.10**
14	**81.02**
15	**84.09**
16	**87.78**
17	**86.16**
18	**89.54**
19	**90**
20	**90.18**
21	**90.79**

Table 3. Group Statistical Analysis of Naive Bayes Classifier and Decision Tree. Mean, Standard Deviation and Standard Error Mean are obtained for 21 samples. Naive Bayes Classifier has higher mean accuracy and lower mean loss when compared to Decision Tree

	Group	**N**	**Mean**	**Std. Deviation**	**Std. Error Mean**
Accuracy	Naive Bayes	21	84.667	7.2600	1.584
	Decision Tree	21	84.111	6.043	1.319

INDEPENDENT SAMPLES TEST

		Leve ne's Test for Equality of Variances		T-test for Equality of means						
		f	sig	t	dif	sig. (2 tailed)	Mean difference	Std. error difference	95% confidence interval of difference	
									Lower	Upper
Efficiency	Equal variances assumed	0.303	0.001	0.762	40	0.450	1.571	2.061	2.595	5.737
	Equal variances not assumed			0.762	38.724	0.450	1.571	2.061	2.599	5.742

Table 4. Independent Sample T-test: Naive Bayes Classifier is insignificantly better than Decision tree with p value <0.585 (Two tailed, p<0.450)

Table 5. Comparison of the Naive Bayes Classifier and Decision tree with their accuracy

CLASSIFIER	ACCURACY(%)
Naive Bayes Classifier	84.667
Decision Tree	84.111

G GRAPH

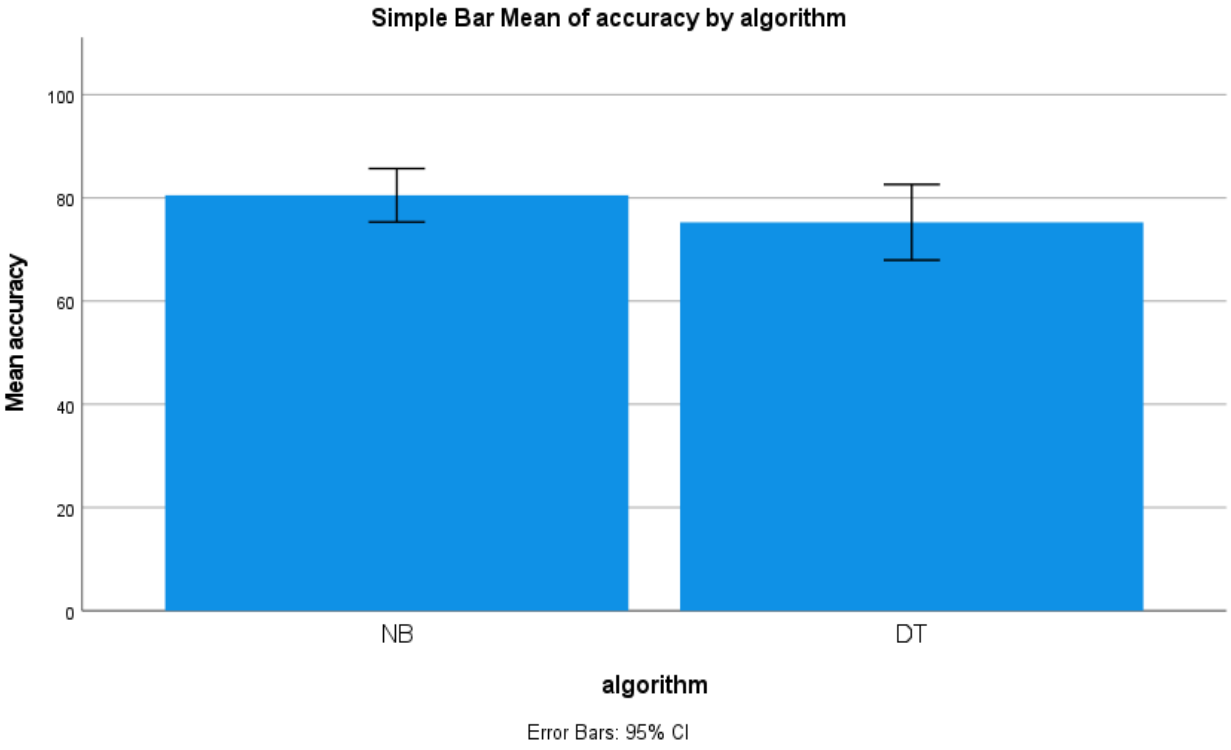

Fig 1. Comparison of Naive Bayes Classifier and Decision Tree. Classifier in terms of mean accuracy and loss. The mean accuracy of Naive Bayes Classifier is better than Decision Tree.Classifier; Standard deviation of Naive Bayes Classifier is slightly better than Decision Tree. X Axis: Naive Bayes Classifier Vs DecisionTree Classifier and Y Axis: Mean accuracy of detection with //mean value//