iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** IV **Month of publication:** April 2024

DOI: https://doi.org/10.22214/ijraset.2024.60001

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue IV Apr 2024- Available at www.ijraset.com

Bayesian Classification for SMS Spam Detection in Mobile Devices

Rajith[1], Sandeep Shridhar Moger[2], R Sai Sudarshan[3], Balachandra Rao[4]
Department of Master of Computer Application, NMAM Institute of Technology

Abstract: ***The abundance of unwanted spam messages complicates the use of Short Message Service (SMS) for efficient communication in modern times. This study investigates developing and utilizing a Naive Bayes Theorem-based Ham/Spam detection system. Because of its ease of use and effectiveness in text classification tasks, the Naive Bayes classifier is used. A collection of SMS messages labeled as" spam" or" ham" (non-spam) makes up the dataset that was used for testing and training. Preprocessing methods, including tokenization, stop-word elimination, and stemming, are employed to extract pertinent features from the text messages. The Naive Bayes classifier learns how words relate to whether they're in a spam or non-spam message by looking at some examples from the dataset. Utilizing criteria such as accuracy, precision, and confusion matrix on a separate testing set, the classifier's performance is evaluated. Additionally, the impact of varying parameters such as smoothing techniques and feature selection methods on the classifier's performance is analyzed. The experimental results used to distinguishing between ham and spam messages in SMS communication***.

Keywords: SMS, Spam, Naive Bayes, Performance metrics, Validation technique, Building API, Android application

I. INTRODUCTION

A. Overview

SMS is one of the best methods for communication in daily life. Because of its extensive usage every month, the average user received 19.5 spam SMS, an increase of 15 percentage over the previous year. (2022) More than three out of five Americans (58 percentage) said they received more spam texts compared to previous year. Spam messages are useless messages that contain unwanted marketing promotions or serve as a social engineering tool for hackers. Spam messages refer to useless messages that contain unwanted marketing promotions or serve as a social engineering tool for hackers.

B. Naive Bayes

The supervised machine learning method Naive Bayes is derived from the well-known Bayes theorem. This approach is widely applied to high dimensional training datasets for text categorization. For email spam filtering, we will use the multinomial Naive Bayes and holdout strategy.

II. LITERATURE REVIEW

[7] The study proposes a hybrid bagging technique for spam email detection that combines the J48 (decision tree) and Naive Bayes algorithms. Through dataset division and result comparison, the hybrid system achieves a notable accuracy of 87.5 percentage. [2] Additionally, the paper offers a thorough review of recent advancements in machine learning-based spam filtering. It emphasizes the need to consider specific problem characteristics, such as concept drift, and highlights challenges in updating classifiers based on bag-of-words representations. While progress has been made, further exploration is needed for more realistic evaluation settings. [6]The paper investigates various forms of Naive Bayes for spam email filtering. By comparing them on realistic datasets, the study highlights the importance of acknowledging different Naive Bayes variants. The incremental training approach and ROC curves provide valuable insights into performance trade-offs. [10] The paper provides a novel explanation for the remarkable performance of Naive Bayes in classification tasks. It highlights the role of dependence distribution among attributes. Even when strong dependencies exist, Naive Bayes can be the best option if they disperse equally or cancel each other out. The study explores optimality conditions, especially under Gaussian distribution. A valuable contribution to understanding Naive Bayes behavior. [8] The paper highlights the challenges posed by email spam and the impact it has on users. It proposes a model using Bayes' theorem and Naive Bayes' Classifier to detect spam messages effectively. By considering IP addresses of senders, this approach aims to improve spam identification. [3]The paper sheds light on the challenges of spam email detection, emphasizing the dynamic environment and the presence of adversarial spammers.

©IJRASET: All Rights are Reserved

Unlike traditional reviews, it delves into real-world issues and strategies used by spammers. The study's empirical evaluation highlights the impact of dataset shift, revealing potential performance degradation [9]. The paper introduces the Naive Bayes classifier, a powerful probabilistic approach for classification tasks. It emphasizes its versatility across different domains and provides an implementation. By testing on a sample dataset, the study ensures the correctness of probabilistic computations [4]. The paper addresses the pressing issue of email spam, which poses risks such as phishing and fraud. By applying machine learning algorithms, it aims to identify fraudulent spam emails. The study evaluates various techniques and selects the best algorithm based on precision and accuracy. [5]The research addresses the pressing issue of spam emails by proposing an innovative approach utilizing email content exclusively to construct a keyword corpus, supplemented by text processing techniques to tackle obfuscation methods employed by spammers. The CSDMC2010 SPAM corpus dataset, which includes 4292 emails in the testing set and 4327 emails in the training set, produced encouraging results when the algorithm was tested. A high accuracy rate of 92.8 percent was attained. This research offers a meaningful contribution to the ongoing efforts in combating spam emails, showcasing its effectiveness in filtering potential spam content. [1]This research tackles the problem of SMS spam by utilizing various machine learning techniques such as logistic regression, Support Vector Machine (SVM), Naive Bayes algorithms and neural networks to effectively filter out unwanted text messages. By evaluating these method's accuracy, the study concludes that neural networks outperform other techniques, serving as the most effective classifier model for distinguishing between ham and spam messages. This research contributes valuable insights into combating SMS spam and highlights the superiority of neural networks in this context.

III. DATASET

The dataset used for this study is named spam.csv, sourced from kaggle.com. The SMS Spam Collection comprises a set of SMS-tagged messages gathered for research on SMS spam. It includes 5,574 messages in English, categorized as either ham (legitimate) or spam. Among these, 4,516 messages are ham, and 653 messages are spam. This collection of SMS identified messages was created specifically for researching SMS spam detection. The dataset have two columns v1 and v2.v1 indicates the message and v2 indicates the ham/spam.v2 column has two values ham and spam.

	v1	v2
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...
...	...	...

Fig. 1 Training dataset for naïve base algorithm

The pie chart illustrates the distribution of ham and spam messages within the dataset. The blue portion represents the ham messages, while the orange portion represents the spam messages. We can see that ham messages are more compared to spam messages in the dataset (fig 2).

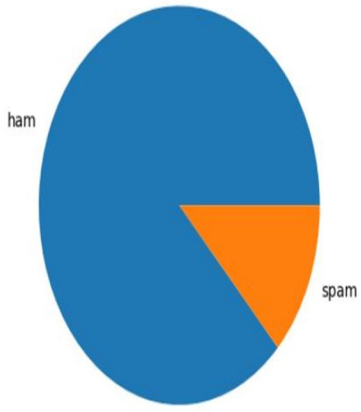


Fig. 2 Pie chart of ham and spam messages in dataset

IV. EXPERIMENTAL DESIGN

We will showcase all of our work on this project in this column.

A. Validation Technique

In this study, the holdout validation technique is employed. Holdout validation divides the dataset into, "train dataset" and "test dataset." The test dataset is used for evaluating the performance of the model on unseen data, and the training dataset is utilized to train the model. In our case, dataset is divided to 80 percentage training data and 20 percentage test data to partition the data for analysis.

B. Performance Metrics

Precision, accuracy, confusion matrix are the performance matrices used in this model.

True Positives (TP): The data point is predicted as positive in this model.

True Negatives (TN): The data point is predicted as negative in this model.

False Positives (FP): The data point is incorrectly predicted as positive in this model.

False Negatives (FN): The data point is incorrectly predicted as negative in this model.

1) *Accuracy:* It is the most widely used and rudimentary performance statistic in algorithms for classification. It is determined by dividing total number of accurate predictions by total number of forecasts made.

$$\text{Accuracy} = \frac{TP + FN}{TP + FP + TN + FN}$$

Fig. 3 Accuracy formula

2) *Precision:* Precision essentially refers to the model's accuracy in predicting future events. It's simple to compute using this formula:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Fig. 4 Precision formula

C. preprocessing

Removed some unnecessary columns in dataset such as v3,v4, etc

Ham and spam keyword in dataset encoded to 1 and 0 to fit the algorithm.

Converted the v1 and v2 column to target and text.

Removed unnecessary information from the dataset and looked for null values, which in our case turned out to be 0, indicating that there was no need for additional processing.

Removed all stop words, number values, punctuation, and ensured that all words were written in lower case. Finally, all words in the mail were stemmed. The process of cutting words down to their root word is called stemming..eg : Imagine you have the word "running". The main part of the word is "run". Root word is used for computation purpose.

D. Algorithm Used

1) *Naive Bayes:* The popular Bayes theorem serves as the foundation for the Naive Bayes algorithm. Text classification problems frequently employ this technique, especially when working with high-dimensional training datasets. The Naive Bayes algorithm is straightforward to implement and comprehend, making it accessible for beginners. It can handle a large amount of data efficiently without any problems. One of the main advantage of utilizing Naive Bayes is its capacity to manage missing data, a prevalent challenge in real-world datasets. Additionally, it necessitates less data in comparison to other machine learning algorithms. Naive Bayes operates by computing the probability of an object.

2) *Bayes Theorem:* Based on prior knowledge, Bayes' theorem is used to calculate the probability of a hypothesis, relying on conditional probability. The Bayes theorem formula is

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

Fig. 4 Bayes Theorem formula

- Let's denote events as A and B.
- The probability that event A will occur given that event B has occurred is represented by the symbol P(A|B).

• The probability that event B will occur given that event A has occurred is represented by the symbol P(B|A).
• The probabilities of event A and event B, respectively, are shown by P(A) and P(B), presuming that they are independent.
Dataset with multiple features/attributes set x
X = (x1, x2, x3, . . . , xn)
When we substitute X, we get

$$P(y|x_1,\ldots,x_n) = \frac{P(x_1|y)P(x_2|y)\ldots P(x_n|y)P(y)}{P(x_1)P(x_2)\ldots P(x_n)}$$

We aim to identify the class y with the highest probability.

$$y = \arg\max_{y} P(y)\prod_{i=1}^{n} P(x_i|y)$$

The above given equation helps us find the class that best explains the observed features based on prior knowledge and conditional probabilities. It's commonly used in machine learning for classification tasks.
Types of naive Bayes model:

- Gaussian: The Gaussian model is only applicable to characteristics with an abnormal distribution. For example, if predictors accept continuous values rather than discrete ones, this suggests that a Gaussian distribution is used to derive the values.
- Multinomial: When the data has a multinomial distribution, the Multinomial Naive Bayes classifier is applied. In essence, it functions by considering the frequency of each word within every text or document. Its main application lies in document classification tasks, where it categorizes a given document into specific categories like Politics, Sport, Education, etc. This classifier leverages word frequencies as predictive features. For this project, we implemented the multinomial algorithm.
- Bernoulli: Like the Multinomial classifier, the Bernoulli classifier utilizes independent Boolean values (0 or 1) as predictor variables. It might seem a bit complex, but this is how it works: it determines whether a word is contained in a text or not. This approach is preferred in scenarios where the frequency of the word is insignificant, and the only relevant factor is whether the term appears or not. This paradigm is often used in document classification.

E. Building API

To connect the Android application with the machine learning model, we have to build an API. It will help to send messages to the model and receive the result.

1) *Create serialized object of model:* Create a serialized object of a machine-learning model using a pickle module in Python. This object is used in API to detect spam messages. To build a serialized object, we have used the pickle module in Python to convert the ML object to serialized data.
2) *Creating API:* In this project, we used the flask library of Python to create API. This API has an entry point to receive messages and send the result generated by the machine learning model. It follows the REST architecture to implement the communication between the Android application and API. (fig.5)

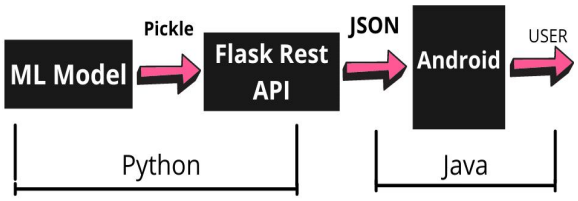


Fig. 5 android application with API

3) *Creating android application:* An Android application was developed with the capability to send requests and receive responses from an API. This functionality was achieved by utilizing the OkHttp library within the Android Studio environment. The application's robust design allows for efficient communication with the API, ensuring seamless data exchange and enhancing the overall user experience. (fig.6)

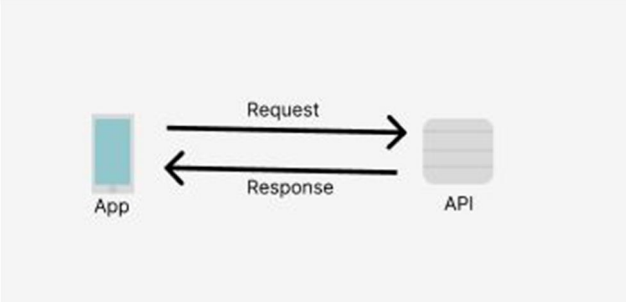


Fig. 5 Android app with API

4) *Communication:* Communication between the Android application and the REST API is performed using the JSON data format. Due to its lightweight nature and readability, JSON is efficient for data interchange. Its ease of writing also contributes to its effectiveness in this context.

V. RESULTS AND ANALYSIS

Below, we'll present the outcomes obtained from our three models utilizing the classification report.

1) Gaussian: We can see Gaussian have low precision and low accuracy. Due to it's low accuracy and precision can impact its performance, especially in specific scenarios like spam detection. (fig. 6)

```
Accuracy 0.8762088974854932
precision 0.5231481481481481
```

Fig. 5 Accuracy and precision value of Gaussian

2) Multinomial: We can see a good result in multinomial naive Bayes. We consider this to algorithm. In spam detection, precision is more important because we don't want to wrongly label important emails (ham) as spam. This is because missing an important email can be a big problem. So, we focus more on precision to avoid this. So, we selected this algorithm for classifying ham/spam in sms. (fig. 7)

```
Accuracy 0.9738878143133463
precision 1.0
```

Fig. 7 Accuracy and precision value of Multinomial

3) Bernoulli: Its performance is better but not that much compared to multinomial naive Bayes. It has lower precision than multinomial. As mentioned above in spam detection we consider precision more to select proper model. (fig. 8)

```
Accuracy 0.9835589941972921
precision 0.984
```

Fig. 8 Accuracy and precision value of Bernoulli

VI. CONCLUSION

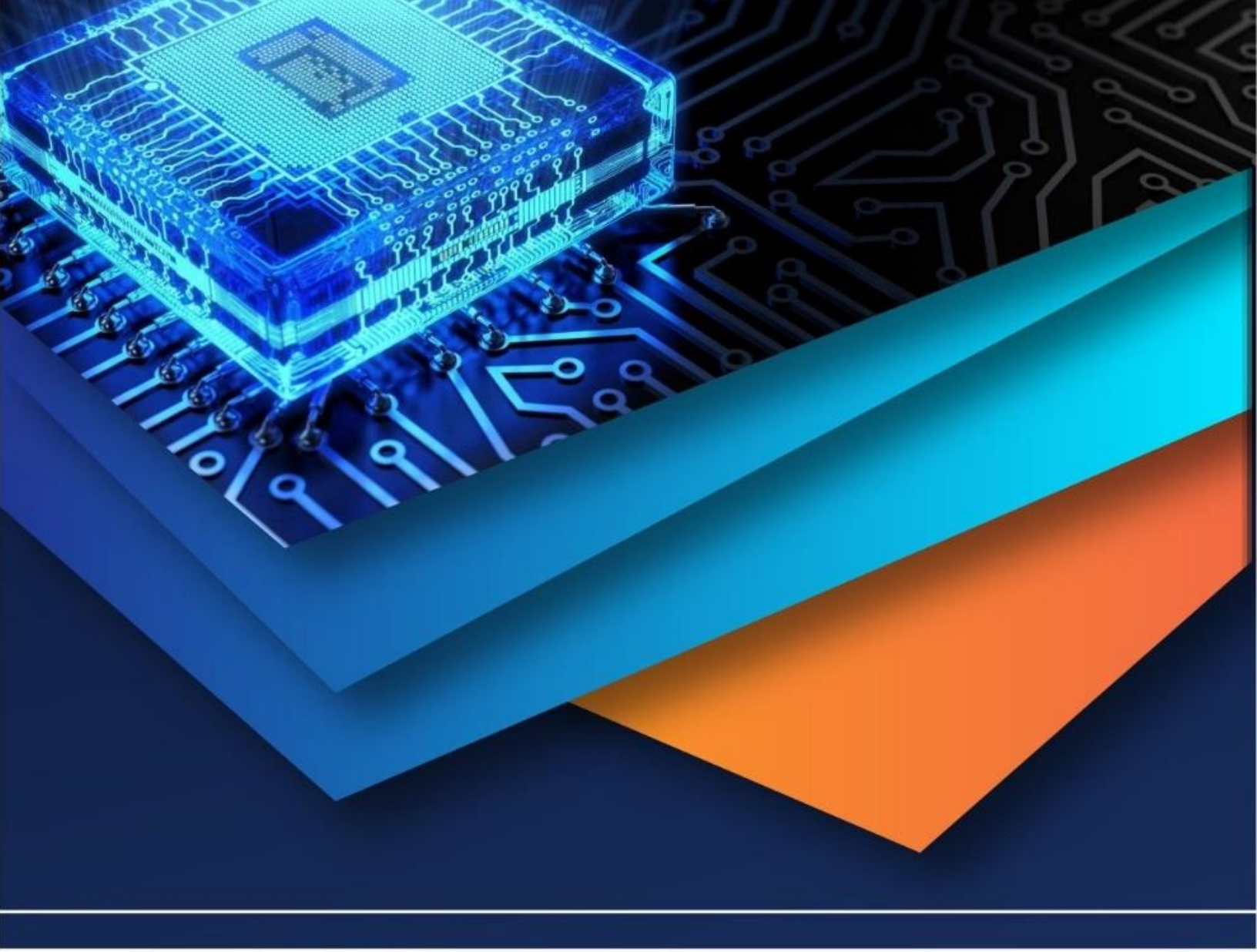
In this study, using the SMS Spam Collection dataset as our main emphasis, we created and put into use a Naive Bayes classifier for SMS spam identification. By using preprocessing methods and parameter analysis, we were able to differentiate spam from ham transmissions with a notable degree of accuracy. Our findings demonstrate how well the Naive Bayes algorithm performs in this situation. To enhance the model's performance further, future efforts could involve exploring alternative machine learning strategies and refining feature selection methods.

REFERENCES

[1] Amani Alzahrani and Danda B Rawat. Comparative study of machine learning algorithms for sms spam detection. In 2019 SoutheastCon, pages 1–6. IEEE, 2019.

[2] Thiago S Guzella and Walmir M Caminhas. A review of machine learning approaches to spam filtering. Expert Systems with Applications, 36(7):10206–10222, 2009.

[3] Francisco Ja´nez-Martino, Roc ˜ ´ıo Alaiz-Rodr´ıguez, V´ıctor Gonzalez- ´ Castro, Eduardo Fidalgo, and Enrique Alegre. A review of spam email detection: analysis of spammer strategies and the dataset shift problem. Artificial Intelligence Review, 56(2):1145–1173, 2023.
[4] Nikhil Kumar, Sanket Sonowal, et al. Email spam detection using machine learning algorithms. In 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), pages 108– 113. IEEE, 2020.
[5] Pingchuan Liu and Teng-Sheng Moh. Content based spam e-mail filtering. In 2016 International Conference on Collaboration Technologies and Systems (CTS), pages 218–224. IEEE, 2016.
[6] Vangelis Metsis, Ion Androutsopoulos, and Georgios Paliouras. Spam filtering with naive bayes-which naive bayes? In CEAS, volume 17, pages 28–69. Mountain View, CA, 2006.
[7] Priti Sharma and Uma Bhardwaj. Machine learning based spam e-mail detection. International Journal of Intelligent Engineering & Systems, 11(3), 2018.
[8] Thashina Sultana, KA Sapnaz, Fathima Sana, and Jamedar Najath. Email based spam detection. International Journal of Engineering Research & Technology (IJERT), 2020.
[9] Feng-Jen Yang. An implementation of naive bayes classifier. In 2018 International conference on computational science and computational intelligence (CSCI), pages 301–306. IEEE, 2018.
[10] Harry Zhang. The optimality of naive bayes. Aa, 1(2):3, 2004



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)