e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:10/October-2023 Impact Factor- 7.868 www.irjmets.com

E-MAIL SPAM DETECTION USING 5 DIFFERENT MACHINE LEARNING ALGORITHMS

Priyank Santosh Iyer[*1]

[*1]UG Student, Department Of Information Technology, B.K. Birla College(Autonomous),
Kalyan, Thane, Maharashtra, India.

ABSTRACT

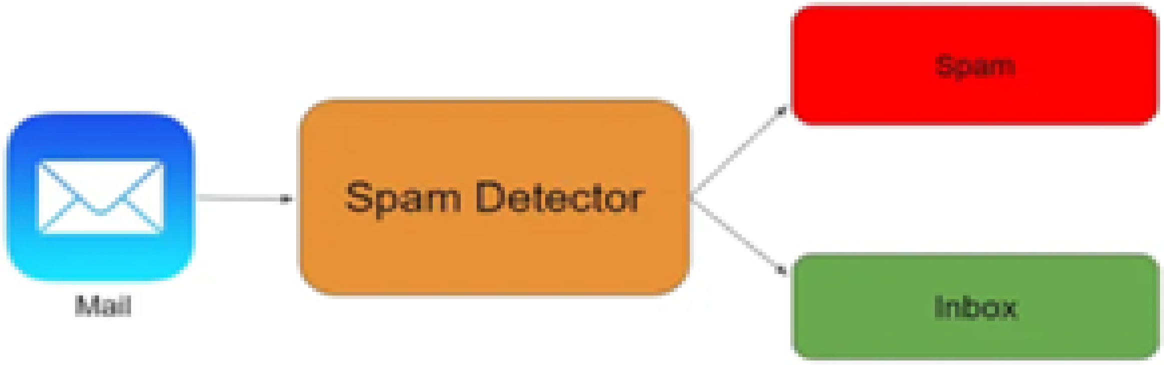
Email communication, an indispensable aspect of modern life, is frequently compromised by the relentless deluge of spam. These unsolicited and often malicious emails not only inundate inboxes but also pose significant security threats. In response to this challenge, machine learning algorithms have emerged as a promising solution for automating the detection of email spam. This research paper presents a comprehensive investigation into the application of machine learning techniques for email spam detection. In this paper, we evaluated the performance of four distinct models namely, K- Nearest Neighbor(KNN), Logistic Regression(LR), Naïve Bayes(NB), Random Forest(RF), and Decision Tree(DT), all trained on the same dataset. The study encompasses an extensive evaluation of these algorithms' performance, including metrics like accuracy, precision, recall, and F1 score. Additionally, the research considers the impact of feature selection and extraction methods on detection accuracy. The findings provide valuable insights for improving email spam detection systems, enhancing the precision of classification, and ultimately fortifying the security of electronic communication.

Keywords: Email Spam, Spam Detection, Machine Learning, Deep Learning.

I. INTRODUCTION

In an age characterized by unprecedented connectivity and the pervasive use of digital communication, email has emerged as one of the most integral tools for personal, professional, and organizational correspondence. However, the ubiquity of email has also paved the way for a growing menace: email spam. Spam emails, often unsolicited and irrelevant, not only clog our inboxes but can also harbor malicious content, phishing schemes, and malware, posing a significant threat to users, businesses, and the integrity of digital communication. Consequently, the need for robust email spam detection systems has never been more critical. This research paper delves into the multifaceted world of email spam detection, aiming to address the evolving challenges and innovative solutions in this realm. Email spam, also known as unsolicited bulk email, undermines the efficiency and security of electronic communication channels, making it a prime concern for individuals, organizations, and email service providers. To combat this pervasive problem, researchers and security experts have tirelessly worked to develop increasingly sophisticated and effective spam detection methods.

Over the years, email spammers have become adept at disguising their messages, employing advanced techniques such as obfuscation, social engineering, and cloaking to evade traditional detection algorithms. In response, the field of email spam detection has seen a paradigm shift, adopting a multidisciplinary approach that combines machine learning, natural language processing, and data mining techniques to differentiate legitimate messages from spam.



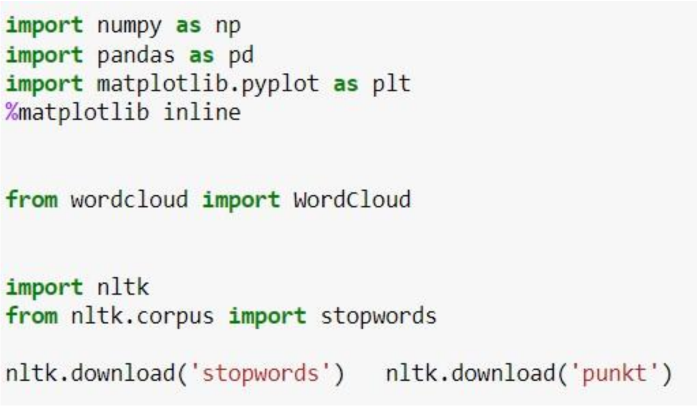
II. LITERATURE REVIEW

The evolution of spam emails has constrained the development of efficient detection methods to safeguard users and maintain the integrity of email communication. In this study, we focus on five prominent machine learning algorithms—that are good at figuring out spam from real messages. Lots of research has been done on these methods, showing what they're good at and where they could be better. Comparing these methods helps us see which one works best. A. Karim, S. Azam, B. Shanmugam, K. Kannoorpatti and M. Alazab.[2]They describe a focused literature survey of Artificial Intelligence Revised (AI) and Machine learning methods for email spam detection. K. Agarwal [3] and T. Kumar. Harisinghaney et al. (2014)and Mohamad & Selamat (2015) have used the "image and textual dataset for the e-mail spam detection with the use of various methods. However, there are still challenges, like dealing with tricky data and changing spam tactics. This review looks at all this research to see what we've learned so far and how our study can add to it, making it easier to stop spam. By building on these studies, we aim to improve the way we detect spam, making email safer and more reliable for everyone, so emails can be trusted and used without worry.

III. METHODOLOGY

Step 1:

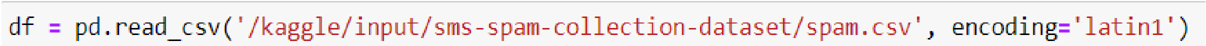
In the first step import necessary python libraries namely numpy, pandas, and matplotlib. The numpy library as np is imported to perform numerical operations. The pandas as pd is imported for data manipulation and analysis. The matplotlib library as plt is imported for data visualization. Then import Word Cloud from wordcloud for text visualization. Further import NLTK and from NLTK import stop words. NLTK is imported for natural language processing.



```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline


from wordcloud import WordCloud


import nltk
from nltk.corpus import stopwords

nltk.download('stopwords')   nltk.download('punkt')
```

Step 2:

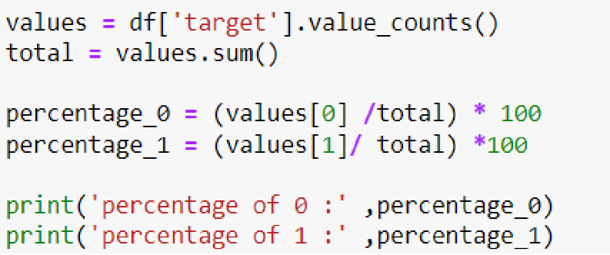
In the second step the data set is loaded after running the below line of code the data from the 'spam.csv' file will be loaded into the df DataFrame, allowing you to perform various data analysis and manipulation tasks using pandas. This can access and work with the data in the DataFrame using various pandas functions and methods.[1]



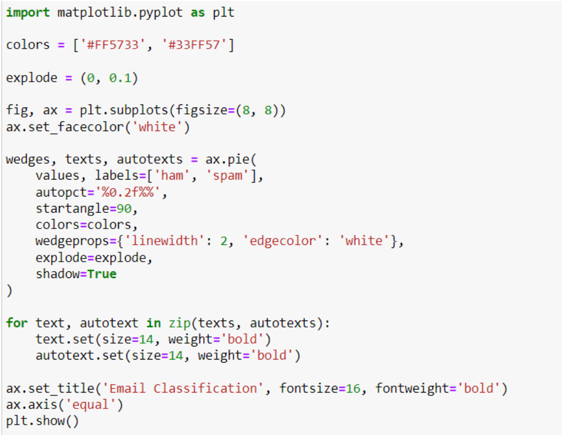
```
df = pd.read_csv('/kaggle/input/sms-spam-collection-dataset/spam.csv', encoding='latin1')
```

Step 3:

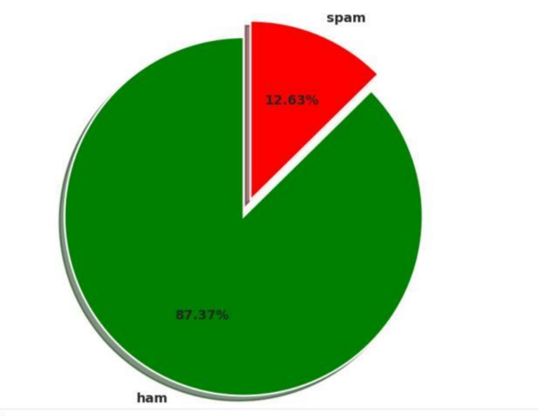
In this step the EDA is performed. EDA i.e. Exploratory Data Analysis refers to the method of studying and exploring record sets to apprehend their predominant traits, discover patterns, locate outliers, and identify relationships between variables. Here we check percentage of Ham and Spam.



```
values = df['target'].value_counts()
total = values.sum()

percentage_0 = (values[0] /total) * 100
percentage_1 = (values[1]/ total) *100

print('percentage of 0 :' ,percentage_0)
print('percentage of 1 :' ,percentage_1)
```



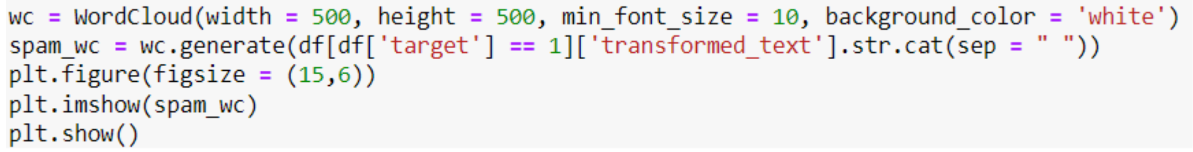
```
import matplotlib.pyplot as plt

colors = ['#FF5733', '#33FF57']

explode = (0, 0.1)

fig, ax = plt.subplots(figsize=(8, 8))
ax.set_facecolor('white')

wedges, texts, autotexts = ax.pie(
    values, labels=['ham', 'spam'],
    autopct='%0.2f%%',
    startangle=90,
    colors=colors,
    wedgeprops={'linewidth': 2, 'edgecolor': 'white'},
    explode=explode,
    shadow=True
)

for text, autotext in zip(texts, autotexts):
    text.set(size=14, weight='bold')
    autotext.set(size=14, weight='bold')

ax.set_title('Email Classification', fontsize=16, fontweight='bold')
ax.axis('equal')
plt.show()
```



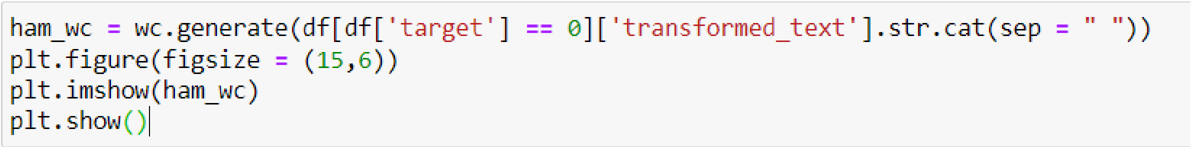
The percentage of ham is too high (87.37%) as compare to spam messages percentage. so the data is imbalance.

Step 4:

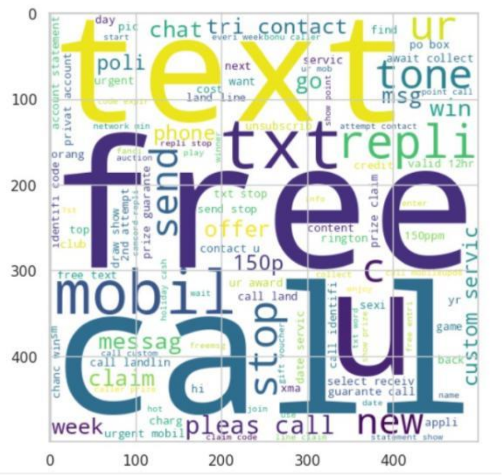
Creating a word cloud of spam messages and not spam messages. Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance.

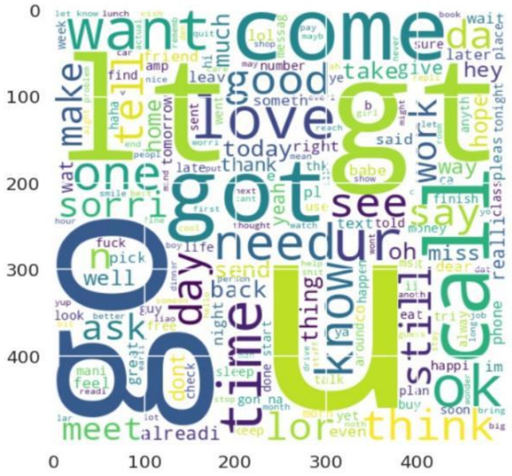


```
wc = WordCloud(width = 500, height = 500, min_font_size = 10, background_color = 'white')
spam_wc = wc.generate(df[df['target'] == 1]['transformed_text'].str.cat(sep = " "))
plt.figure(figsize = (15,6))
plt.imshow(spam_wc)
plt.show()
```



```
ham_wc = wc.generate(df[df['target'] == 0]['transformed_text'].str.cat(sep = " "))
plt.figure(figsize = (15,6))
plt.imshow(ham_wc)
plt.show()
```

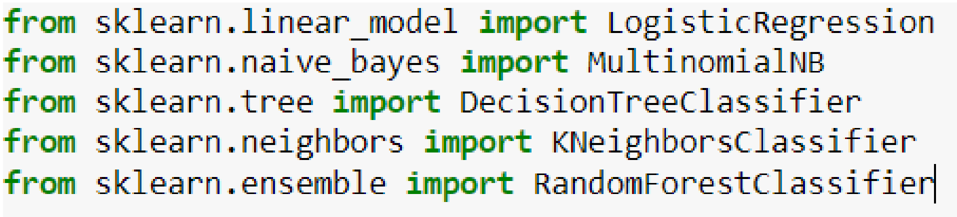




Word Cloud for Spam Messages **Word Cloud for Not spam Messages**

Step 5:

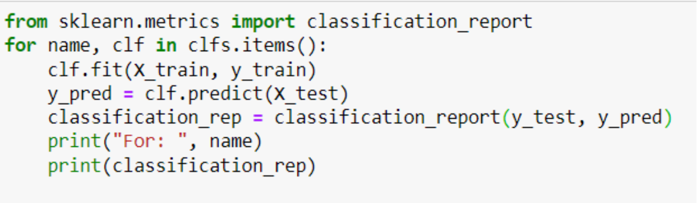
In this step we will import the 5 machine learning models for spam detection namely Logistic regression, naive_bayes, DecisionTree , KNeighboursClassifier and RandomForest and later initialize them.



```
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
```

Step 6:

In this step the models are trained and saved successfully.



```
from sklearn.metrics import classification_report
for name, clf in clfs.items():
    clf.fit(X_train, y_train)
    y_pred = clf.predict(X_test)
    classification_rep = classification_report(y_test, y_pred)
    print("For: ", name)
    print(classification_rep)
```

IV. RESULTS

The below table describes the comparative analysis of the five machine learning models used in this research for detecting spam emails.

Table 1: Comparative Analysis of ML models for Email Spam Detection

Comparative Analysis of ML models for Email Spam Detection					
Models	**LR**	**NB**	**DT**	**KNN**	**RF**
Accuracy	0.95	0.97	0.93	0.90	0.97
Precision	0.95	1.0	0.85	1.0	0.97
F1 score	0.82	0.89	0.71	0.45	0.89
Recall	0.72	0.80	0.61	0.29	0.81

V. CONCLUSION

Naive Bayes (NB) and Random Forest (RF) appear to be the top-performing models, achieving high accuracy, precision, and F1 scores.

Logistic Regression (LR) also performed well but with slightly lower recall.

Decision Tree (DT) and K-Nearest Neighbor (KNN) showed lower overall performance in terms of accuracy and F1 score, with KNN particularly having low recall.

The choice of the best model depends on the specific requirements of the email spam detection system. If minimizing false positives is crucial, NB might be preferred. If a balanced approach between precision and recall is desired, RF could be a good choice. However, the final decision should consider the specific trade-offs and real-world implications of the model's performance. Additionally, further fine-tuning and validation may be needed for a comprehensive assessment.

VI. REFERENCES

[1] Dataset.from.kaggle:https://www.bing.com/search?q=kaggle&pc=0SLN&ptag=C999N9998D071422A00ED787AAB& form=0A0909&conlogo=CT3210127

[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.