University of New Haven
TAGLIATELA COLLEGE OF ENGINEERING
Electrical & Computer Engineering and Computer Science

Intro to Data Science (DSCI-6002-01)- Final Project

Email Spam Detection Using Machine Learning Algorithms

Proessor: Dr.Ardina Sula
Team-15
Vaishnavi Kukkala
Gnaneswari Vadepalli
Abbina Vamsi Krishna
Gude Venkata Naga Sandeep

TECHNICAL REPORT

CONTENTS

Executive Summary

The process of recognizing and eliminating unsolicited or undesired messages, which are usually sent via text or email, is known as spam detection. Spammers and other bad actors frequently send these messages with the aim of advertising a good or service, tricking the recipient into divulging personal information, or infecting their device with malware. Usually, machine learning methods are used for spam identification. These algorithms examine message content and look for patterns or traits that are frequently linked to spam. These algorithms can be trained on enormous datasets including labeled samples of both legitimate and spam messages, enabling them to become highly accurate at differentiating between the two.

Major Challenges:

- Evolving Tactics
- Sensitivity and Specificity Trade-off
- Imbalanced Datasets
- Context and Language Complexity
- Email Header Manipulation
- Dynamic Content
- Personalized Attacks (Spear Phishing)
- Encrypted Content
- Real-time Processing
- Legal and Ethical

Highlights of Project

Content Filtering:

- Keyword Analysis: Scanning email content for known spam keywords and phrases.
- Bayesian Filtering: Utilizing statistical algorithms to analyze the probability of an email being spam based on the occurrence of certain words or patterns.

Header Analysis:

- Sender Policy Framework (SPF): Confirming that the sender's IP address is permitted to send emails on behalf of the domain.
- DomainKeys Identified Mail (DKIM): This ensures that the email content was not altered in transit and that it is legitimately from the claimed sender.
- Domain-based Message Authentication, Reporting, and Conformance (DMARC): A protocol that adds authentication and reporting methods to SPF and DKIM.

Blacklists and Whitelists:

- Blacklists: Maintaining lists of known spammers, suspicious IP addresses, or domains to block or flag emails originating from these sources.
- Whitelists: Allowing emails from trusted senders or domains to pass through without extensive.

Collaborative Filtering:

- Community Feedback: Leveraging feedback from users to identify and block spam emails. This can include reporting mechanisms for users to mark emails as spam.
- Challenge-Response Systems:
- CAPTCHA Challenges: Requiring users to solve a challenge (e.g., CAPTCHA) to prove they are human and not automated spambots.
- Rule-Based Filtering:
- Customizable Rules: Allowing users or administrators to define specific rules for filtering emails based on criteria such as sender, subject, or content.

Continuous Updates:

- Security Updates: Regularly updating spam filters to adapt to new spamming techniques and evolving threats

Effective email spam detection often involves a combination of these techniques to provide a robust defense against the ever-changing landscape of spam and phishing attacks.

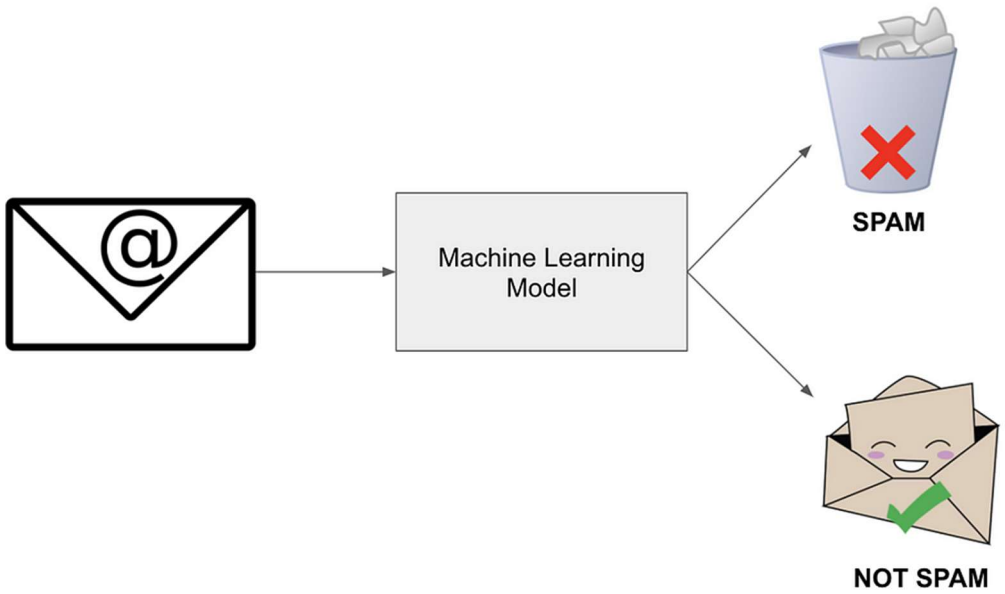
Abstract

The process of recognizing and eliminating unsolicited or undesired messages, which are usually sent via text or email, is known as spam detection. Spammers and other bad actors frequently send these messages with the aim of advertising a good or service, tricking the recipient into divulging personal information, or infecting their device with malware. Usually, machine learning methods are used for spam identification. These algorithms examine message content and look for patterns or traits that are frequently linked to spam. Efficient spam detection is a crucial responsibility for individuals and companies alike, as it can lessen the likelihood of phishing assaults, stop unsolicited communications from piling up in inboxes, and enhance cybersecurity in general.

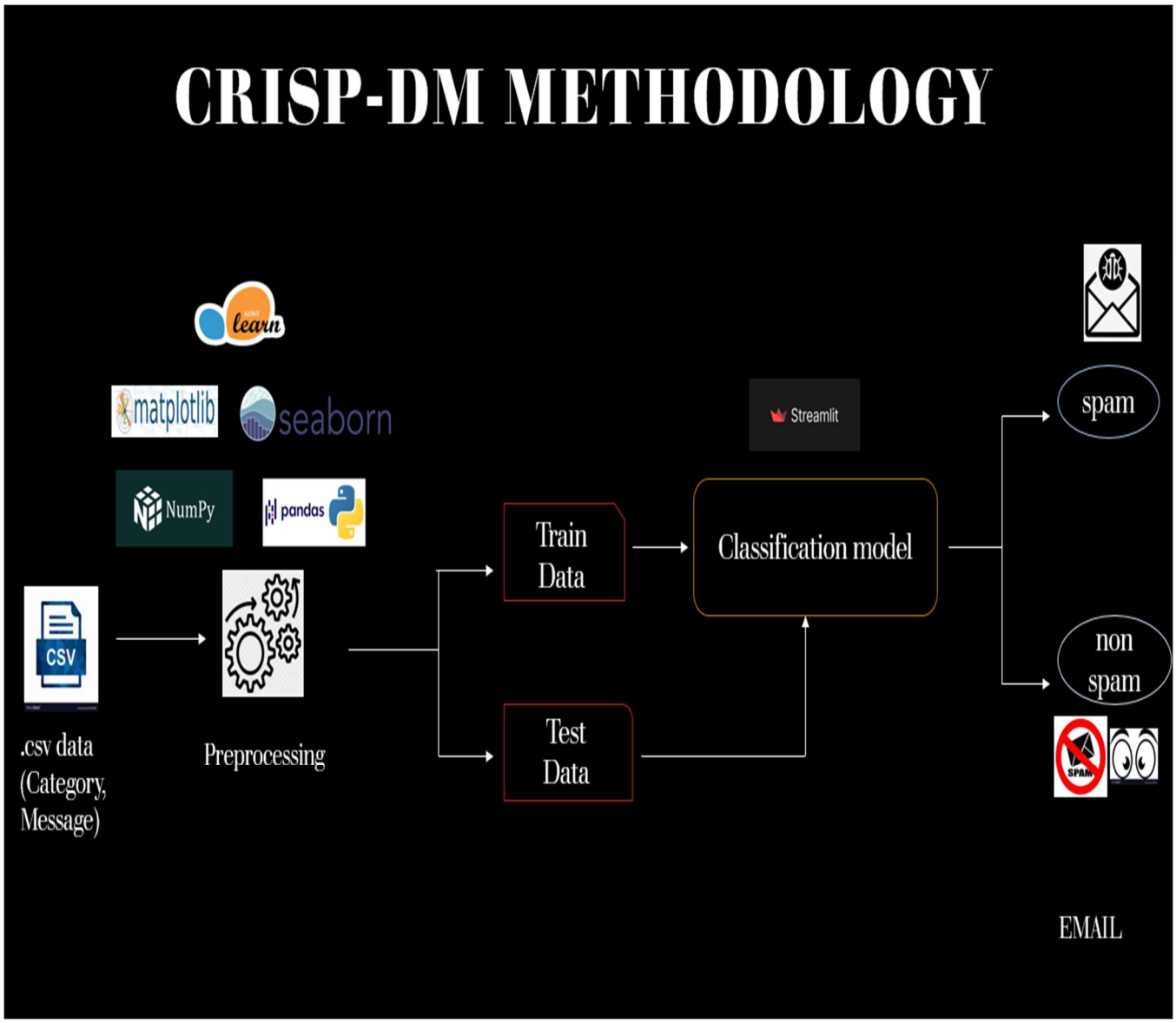
In this case, a variety of machine learning models are applied to separate spam from ham messages. A well-known dataset of spam emails is used to train the model, and several machine learning methods are used for classification.

Introduction

Time complexity is a crucial factor in every facet of this large globe. These days, we communicate more quickly by sending and receiving emails, or electronic mail. Every day, 4 million people use this technology. It must therefore be quite safe to use. Through the Internet, it developed become a vital service for billions of people. because of its mass message delivery and ease of use. Over time, electronic mail emerged as a trend.



Methodology



Discussion

Email spam detection using machine learning algorithms is a sophisticated and effective approach to identify and filter out unwanted or malicious emails. Machine learning leverages computational models to automatically learn patterns and make predictions based on data. Machine learning algorithms are capable of being trained on enormous data set, which includes both spam and non-spam (ham) messages to gain insight into the distinguishing aspects of the two mails in relation to spam messages in the emails detection. Here's a discussion on key aspects of email spam detection using machine learning:

Feedback Loops: User feedback on misclassified emails can be used to update the model and enhance its performance over time.

Imbalanced Data: The imbalance between the number of spam and non-spam emails in a dataset can pose challenges. Techniques like oversampling or undersampling may be applied.

Adversarial Attacks: Spammers may intentionally craft emails to deceive machine learning models. Robust model architectures and ongoing monitoring are necessary to address this challenge.

In conclusion, email spam detection using machine learning offers a powerful and adaptive solution. By continuously learning from data patterns, these algorithms can effectively distinguish between legitimate and unwanted emails, contributing to a more secure and efficient email communication environment.

Data Collection:

In the initial step, we are visiting download the dataset from Kaggle (https://www.kaggle.com/datasets/shantanudhakadd/email-spam-detection-datasetclassification) which contains 5572 records of two columns i.e., Message and Category. we'd wish to import the required python libraries to perform operations such as NumPy, Pandas, and sklearn then the downloaded dataset has got to be uploaded by the method “read_ csv” in pandas’ library. Data Pre-processing must be done by checking null values within the data.

Data Preprocessing:

LABEL ENCODING: Later, label encoding is required which is described as the act of translating labels into numerical values that can be read by machines. Spam is tagged "0" and Ham Mail is designated "1".

Feature Extraction: Machine learning models require relevant features to make predictions. Features in the context of emails could include sender information, email content, subject line, and other metadata.

Labeling Data: A dataset needs to be labeled with examples of spam and non-spam emails to train a supervised machine learning model.

Feature Engineering:

NLP Techniques: Natural Language Processing (NLP) methods can be employed to analyze the content of emails. This includes techniques such as tokenization, stemming, and sentiment analysis.

Behavioral Features: Analyzing the behavioral patterns of emails, such as the frequency of certain words or the structure of the email body.

Model Training:

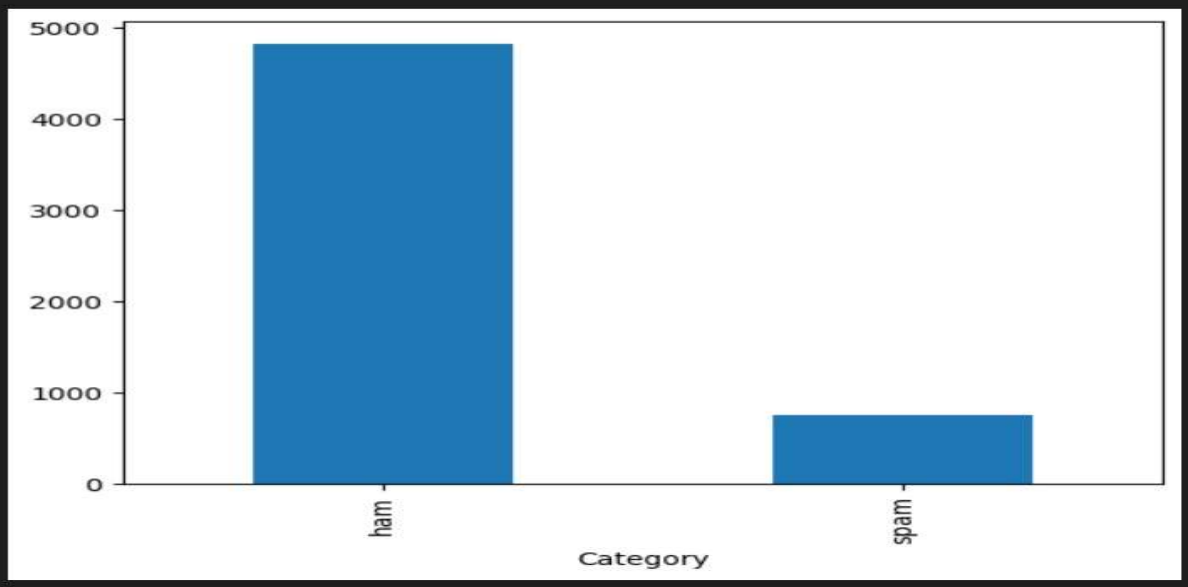
Training Set: The machine learning model is trained on a labeled dataset, learning to distinguish between spam and non-spam based on the provided features.

Cross-Validation: To ensure the model's generalizability, cross-validation techniques may be employed to assess its performance on different subsets of the data.

EDA:

```
df['Category'].value_counts().plot(kind='bar')
```

```
<Axes: xlabel='Category'>
```



```
sns.scatterplot(x=df['Category'],y=df['Message'])
plt.xticks(rotation='vertical')
```

Python

```
([0, 1], [Text(0, 0, 'ham'), Text(1, 0, 'spam')])
```

Model Evaluation:

Metrics: Measures of performance like precision, recall, accuracy, and F1 score are utilized for evaluating how successfully the model finds spam and non-spam emails.

Confusion Matrix: Provides a detailed description of each algorithm's estimates, differentiating between true positives, true negatives, false positives, and false negatives.

Adaptability:

Continuous Learning: Machine learning models can be designed to adapt to evolving spam patterns by incorporating continuous learning mechanisms.

Feedback Loops: User feedback on misclassified emails can be used to update the model and enhance its performance over time.

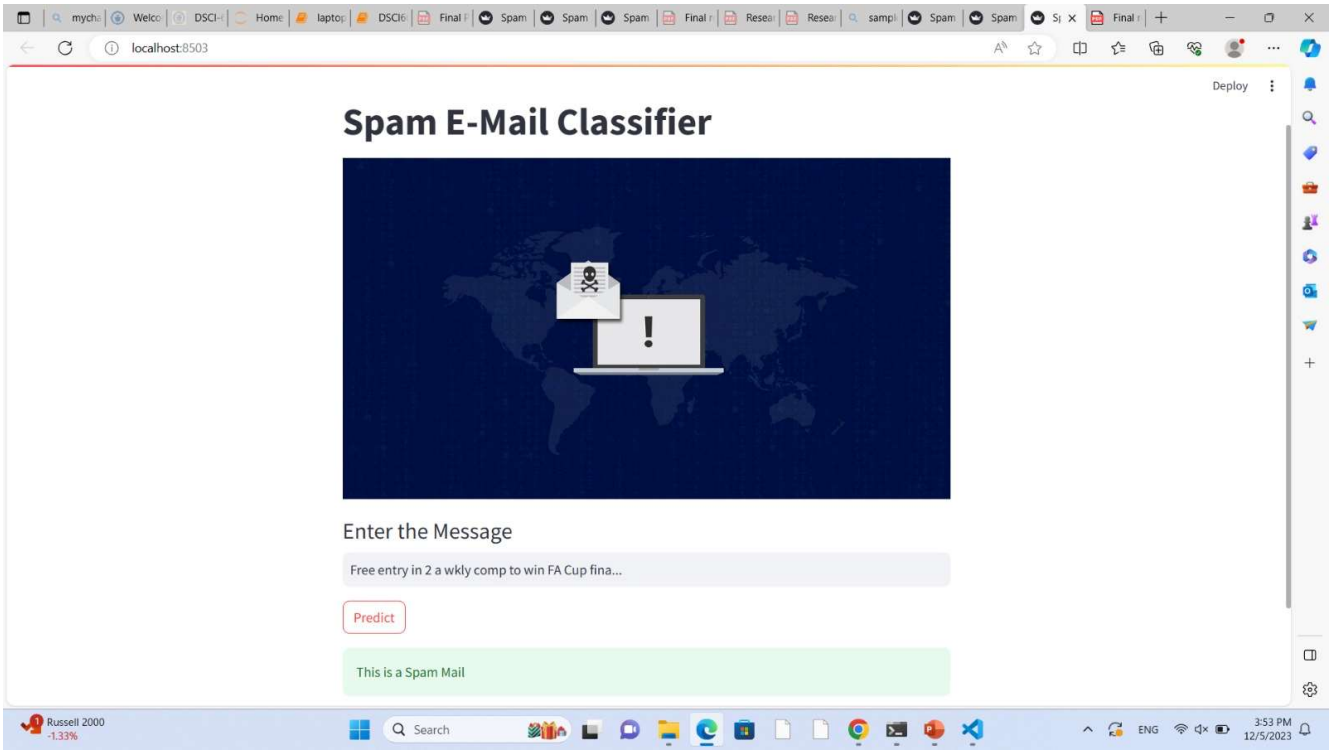
Challenges: Imbalanced Data: The imbalance between the number of spam and non-spam emails in a dataset can pose challenges. Techniques like oversampling or undersampling may be applied.

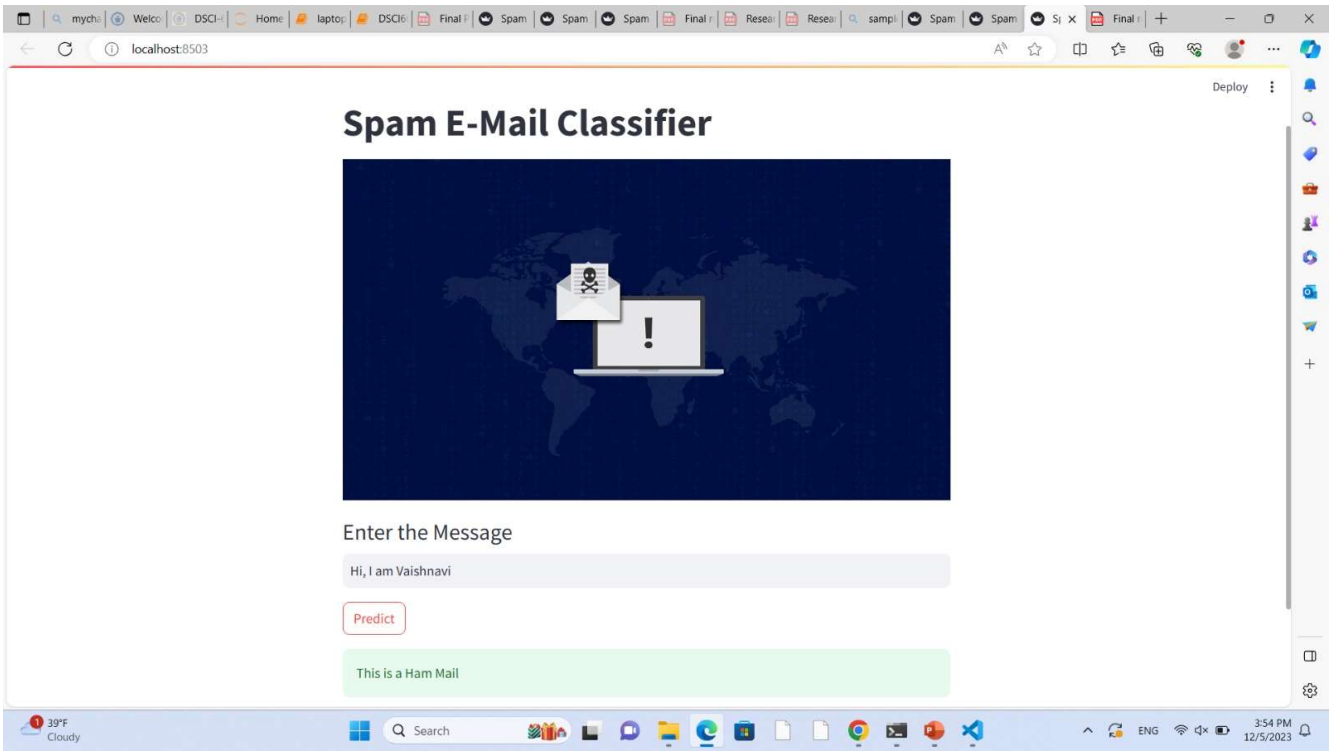
Adversarial Attacks: Spammers may intentionally craft emails to deceive machine learning models. Robust model architectures and ongoing monitoring are necessary to address this challenge.

In conclusion, email spam detection using machine learning offers a powerful and adaptive solution. By continuously learning from data patterns, these algorithms can effectively distinguish between legitimate and unwanted emails, contributing to a more secure and efficient email communication environment.

DEPLOYMENT:

We used streamlit for deployment.





RESULTS:

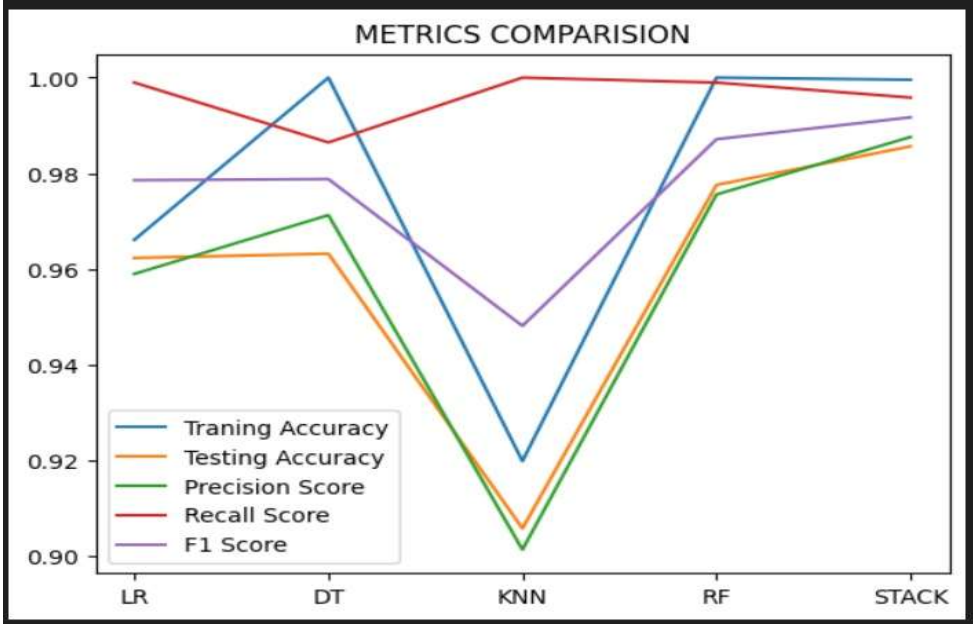
1) Testing Accuracy: Random Forest (RF) and the stacked model (STACK) have relatively high testing accuracies (97.76% and 98.57%, respectively). High accuracy indicates that these models perform well on the testing dataset.

2) Precision Score: The stacked model (STACK) has the highest precision score (98.76%). This means that, among the instances predicted as spam, the stacked model has the highest percentage of true spam instances.
3) Recall Score: K-Nearest Neighbors (KNN) has a perfect recall score (100%). This indicates that KNN correctly identifies all actual spam instances.
4) F1 Score: The stacked model (STACK) has the highest F1 score (99.17%). The F1 score is a balanced metric that considers both precision and recall.

	Traning Accuracy	Testing Accuracy	Precision Score	Recall Score	F1 Score
LR	0.966121	0.962332	0.959000	0.998958	0.978571
DT	1.000000	0.963229	0.971282	0.986458	0.978811
KNN	0.919901	0.905830	0.901408	1.000000	0.948148
RF	1.000000	0.977578	0.975585	0.998958	0.987133
STACK	0.999551	0.985650	0.987603	0.995833	0.991701

VISUALIZATION

```
alg = ['LR','DT','KNN','RF','STACK']
plt.plot(alg,a1)
plt.plot(alg,a2)
plt.plot(alg,a3)
plt.plot(alg,a4)
plt.plot(alg,a5)
legend = ['Traning Accuracy', 'Testing Accuracy', 'Precision Score', 'Recall Score', 'F1 Score']
plt.title("METRICS COMPARISION")
plt.legend(legend)
plt.show()
```



Conclusion

- The choice of the best model depends on the specific goals and requirements of the email spam detection problem.
- The spam email categorization is extremely important in classifying email messages as well as identifying different emails which are spam or non-spam.
- Based on these parameters, the stacking approach (STACK) seems to be a high performer in terms of accuracy, precision, recall, and Formula 1 score. However, the best model may be determined by the unique trade-offs.
- Filtering of emails will be performed on the foundation of these trusted and confirmed domain names.
- This approach is capable of being utilized by a large organization to distinguish between respectable emails which contain only those emails people wish to obtain.

References

[1] Sjarif, Nila, & Amir, N. (2019). SMS Spam Message Detection using Term Frequency-Inverse Document Frequency and Random Forest Algorithm. Procedia Computer Science , 509-515.

[2] Narayan, Vipul, et al. "To Implement a Web Page using Thread in Java." (2017).

[3] Srivastava, Swapnita, and P. K. Singh. "HCIP: Hybrid Short Long History Table-based Cache Instruction Prefetcher." International Journal of Next-Generation Computing 13.3 (2022).

[4] Srivastava, Swapnita, and P. K. Singh. "Proof of Optimality based on Greedy Algorithm for Offline Cache Replacement Algorithm." International Journal of Next-Generation Computing 13.3 (2022).

[5] Smiti, Puja, Swapnita Srivastava, and Nitin Rakesh. "Video and audio streaming issues in multimedia application." 2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence). IEEE, 2018.

[6] Qaiser, Shahzad, & Ali, R. (2018). Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents. International Journal of Computer Applications , 25-29.

[7] Awasthi, Shashank, et al. "A Comparative Study of Various CAPTCHA Methods for Securing Web Pages." 2019 International Conference on Automation, Computational and Technology Management (ICACTM). IEEE, 2019.