THAKUR COLLEGE OF SCIENCE & COMMERCE
KANDIVALI (EAST), MUMBAI 400101

A PROJECT REPORT ON

SMS Spam Detection

For

Thakur College of Science & Commerce

By

Harshita Pandey

Submitted in Partial Fulfillment of Bachelor of Science (Computer Science)

Thakur College of Science and Commerce (Autonomous College)
[Permanently Affiliated to UNIVERSITY OF MUMBAI]

ACADEMIC YEAR 2023 - 2024

Department of Commuter Science

02/04/2021

(2023-2024)

Certificate of Approval

This is to certify that the project work entitled **"SMS Spam Detection"** is prepared by **Harshita Pandey** a student of **"Third Year Bachelor of Science (Computer Science)"** Program of Thakur College of Science & Commerce (University of Mumbai).

This is the original study work and important sources used have been duly acknowledged in the report. The report is submitted in partial fulfilment of B.Sc. (Computer Science) course as per rules of University of Mumbai.

Prof. Rahul Dhuru,
Project Guide

Prof. Ashish Trivedi,
Head, DCS

External Examiner

Index

Acknowledgement

Achievement is finding out what you would be doing rather than what you have to do. It is not until you undertake such a project that you realize how much effort and hard work it really is, what are your capabilities and how well you can present yourself or other things. It gives me immense pleasure topresent this report towards the fulfillment of my project.

It has been rightly said that we are built on the shoulder of others. For everything I have achieved, the credit goes to all those who had helped me to complete this project successfully.

I take this opportunity to express my profound gratitude to management of Thakur Degree College of Science & Commerce for giving me this opportunity to accomplish this project work.

I am very much Thankful to **Dr. (Mrs.) C. T. Chakraborty, Principal of Thakur College of Science & Commerce** for her kind Co-Operation in the completion of my project.

A Special Vote of Thanks to our HOD (Head of Department) **Mr. Ashish Trivedi** and to our Project Guide **Mr. Rahul Dhuru**. Thanks to all our teachers for helping and Guiding me Throughout the Project.

Finally, I would like to Thank all my Friends & entire Computer Science Department who directly or indirectly helped me in Completion of this Project and to my Family, without whose Support, Motivation and Encouragement this would not have been Possible.

Harshita Pandey

THAKUR COLLEGE OF SCIENCE & COMMERCE

Department of Computer Science

2023-2024

Students Name: Harshita Pandey

Project Name: SMS Spam Detection

College Name: Thakur College of Science and Commerce

PHASES	EXPECTED DATE OF COMPLETION	ACTUAL DATE OF COMPLETION
Preliminary Investigation	12/09/23	15/09/23
System Analysis	20/09/23	25/09/23
System Designing	30/09/23	1/10/23
System Coding	5/01/24	7/01/24
System Implementation	10/01/24	15/01/24
Report Submission	19/02/24	19/02/24

Organization Overview

The Thakur College of Science and Commerce (TCSC) is a college in Kandivali in Mumbai of Maharashtra, India running by Thakur Educational Trust.

Thakur College was started in 1992 to serve the needs of students passing SSC examination from the schools around Kandivali area and Thakur Vidhya Mandir which has already established itself as one of the schools in the area. It offers courses at primarily the higher secondary and under- graduate levels. The courses at the undergraduate and post graduate level are offered in affiliation with Mumbai University, Mumbai. An ISO 9001:2008 College with A grade as assessed by the National Assessment and Accreditation Council NAAC.

Name: Thakur College of Science and Commerce Founded: 1997
Address: Thakur College of Science and Commerce, Thakur Village Kandivali(E), Mumbai -400001
Motto: Journey towards Excellence Total Staff: 200
Number of Students: 12500 Email: Helpdesk@tcsc.org.in

Description

SMS spam detection is a critical area of research in cybersecurity, given the escalating sophistication of spamming techniques and the potential harm posed to individuals and organizations. Traditional rule-based spam filters have limitations in adaptability and accuracy, often failing to keep pace with evolving spam tactics. Machine learning, however, offers a promising alternative by enabling systems to automatically learn and adapt to new spam patterns. By training on labeled datasets containing examples of both spam and legitimate emails, machine learning algorithms can extract relevant features and patterns to classify incoming emails effectively. This project aims to harness the power of machine learning to create a dynamic and robust spam detection system capable of accurately identifying and filtering out spam emails/SMS while minimizing false positives. The significance ofeffective SMS/email spam detection extends beyond mere inconvenience, as spam emails/SMS often serve as vehicles for malicious activities such as phishing, fraud, and malware distribution. These threats can compromise sensitive information, damage reputations, and incur financial losses. Therefore, the development of reliable spam detection mechanisms is imperative for safeguarding individuals, businesses, and entire networks against these risks. Machine learning approaches offer a multifaceted solution by not only enhancing the accuracy of spamdetection but also enabling continuous learning and adaptation to emerging spam tactics. By leveraging advanced algorithms such as neural networks, support vector machines, and ensemble methods, this project aims to create a sophisticated SMS spam detection system capable of effectively mitigating various spam

1. Introduction

In the ever-evolving landscape of digital communication, SMS remains a cornerstone of modern correspondence, facilitating seamless communication across the globe. However, amidst the convenience of SMS, the persistent challenge of spam threatens to undermine its utility and security. Spam SMS, ranging from unsolicited advertisements to maliciousphishing attempts, not only clutter inboxes but also pose significant risks to users' privacy andsecurity. In response to this growing threat, the application of machine learning techniques presents a compelling avenue for bolstering email security through efficient spam detection mechanisms. This project seeks to explore and harness the capabilities of machine learning indeveloping a robust SMS spam detection system, thus fortifying the integrity of email communication channels and enhancing user confidence in digital correspondence. Facilitating seamless communication across the globe. However, amidst the convenience of email, the persistent challenge of spam threatens to undermine its utility and security. Spam emails, ranging from unsolicited advertisements to malicious phishing attempts, not only clutter inboxes but also pose significant risks to users' privacy and security. In response to this growing threat, the application of machine learning techniques presents a compelling avenue for bolstering email security through efficient spam detection mechanisms. This project seeksto explore and harness the capabilities of machine learning in developing a robust email spamdetection system, thus fortifying the integrity of email communication channels and enhancing user confidence in digital correspondence.

The primary purpose of this project is to leverage machine learning algorithms to create an effective and reliable SMS spam detection system. By harnessing the power of data-driven approaches, the project aims to enhance the accuracy and efficiency of spam identification, thereby reducing the incidence of unwanted and potentially harmful emails infiltrating users'

inboxes. Additionally, the project seeks to contribute to the broader discourse on cybersecurity by offering insights into the application of machine learning in addressing contemporary threats to digital communication channels. Through empirical experimentation and analysis, the project endeavors to elucidate the effectiveness of different machine learning models and techniques in mitigating the scourge of SMS spam.

1.1 Problem Statement

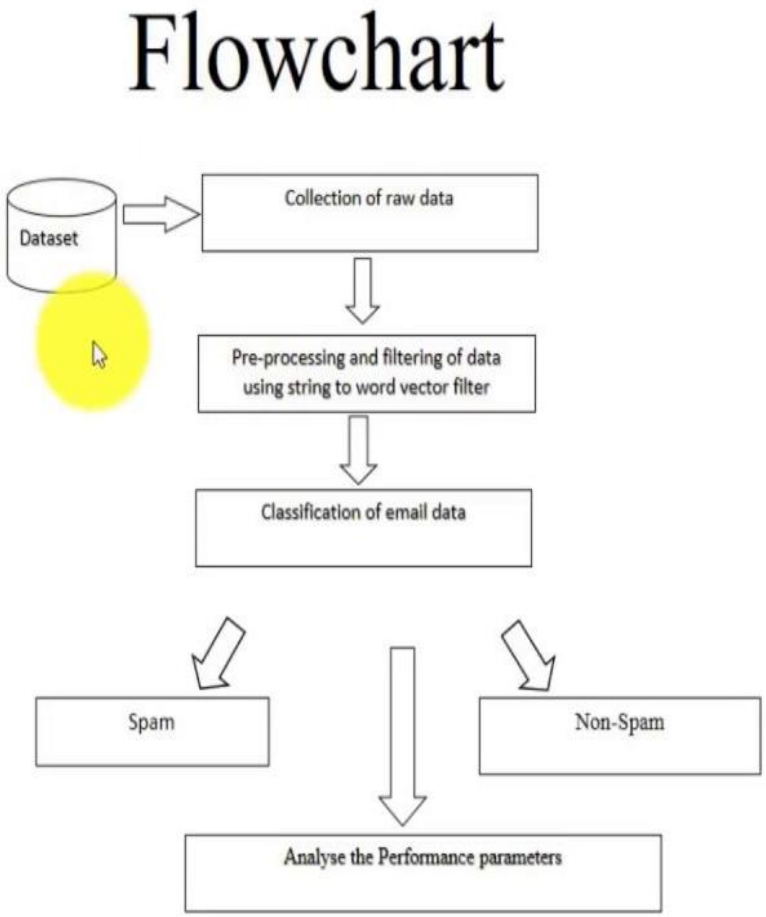
The proliferation of spam SMS poses a multifaceted challenge to users and organizations alike, encompassing issues of privacy infringement, information security breaches, and resource wastage. Traditional spam filtering techniques, often reliant on static rule-based systems, struggle to adapt to the dynamic and evolving nature of spam tactics. Consequently, there is a pressing need for more sophisticated and adaptive solutions capable of discerning between legitimate emails and spam with a high degree of accuracy. This project addresses this challenge by investigating the efficacy of machine learning algorithms in detecting and classifying spam emails. Through empirical evaluation and comparative analysis, the project aims to identify optimal approaches for spam detection, thereby contributing to the development of more resilient and responsive SMS security measures.

Limitation of Present System

Present SMS spam detection systems have made significant advancements in accurately identifying and filtering out spam SMS. However, they still face several limitations that can affect their effectiveness. One limitation is the reliance on predefined rules or heuristics to identify spam, which may not adapt well to new spamming techniques or subtle variations in spam content. Additionally, spammers continuously evolve their tactics, such as obfuscating text or using image-based spam, making it challenging for traditional detection methods to keep up. Moreover, legitimate emails with similar characteristics to spam, such as newslettersor marketing emails, may get falsely flagged as spam, leading to false positives and potentially causing inconvenience to users. Furthermore, some sophisticated spam campaigns employ social engineering techniques, making it difficult for automated systems to differentiate between genuine and malicious content. Another challenge is the emergence of highly targeted and personalized spam, which may evade detection by generic spam filters. Lastly, the sheer volume of emails processed daily poses scalability challenges for spam detection systems, necessitating efficient algorithms and infrastructure to handle the workload effectively. Overall, while present spam detection systems have come a long way, addressing these limitations will be crucial for further enhancing their accuracy and reliability in combating spam.

Proposed System

The proposed methodology for spam SMS and email both common filtering involves training a machine learning model using rough set theory. Incoming SMS are preprocessed and incorporated into a training corpus. Feature extraction is performed on the corpus to create training data, and rough set rules are generated. A machine learning model is then trained on this data to classify sms as spam or non-spam. Finally, the rough set rules are refined based on the model's classification re



2. System Requirements

1. Python
2. Any IDE to run Python
3. Dataset
4. Jupyter Book

3. System Design

Frontend:

Streamlit Web Application: The frontend is built using Streamlit, an open-source app framework for machine learning and data science. Streamlit provides a user-friendly interface for users to interact with the Fashion Recommender System.

User Interface (UI): The Streamlit app allows users to upload an image of a fashion item they are interested in. It displays the uploaded image, along with recommendations and extracted features.

Interactive Elements: The UI includes interactive elements such as file upload buttons, image displays, and recommendation grids. These elements enable users to easily interact with the system and view recommendations.

Backend:

Feature Extraction: The backend includes functionality for feature extraction from uploaded images. This task is performed using a pre-trained ResNet50 CNN model, which is capable of extracting high- level features from images.

Recommendation Engine: The backend contains a recommendation engine powered by a Nearest Neighbors algorithm. It takes the extracted features from the uploaded image and finds similar fashion items in the dataset based on feature similarity.

Data Loading: The backend loads precomputed feature embeddings and corresponding filenames from files (embeddings.pkl and filenames.pkl, respectively) to use in the recommendation process.

Middleware:

Communication Layer: The middleware facilitates communication between the frontend and backend components. It handles requests from the frontend, such as image uploads, and forwards them to the appropriate backend components for processing.

Data Processing: The middleware coordinates data processing tasks such as feature extraction and recommendation generation. It interacts with the backend to execute these tasks and prepares the results to be displayed in the frontend UI.

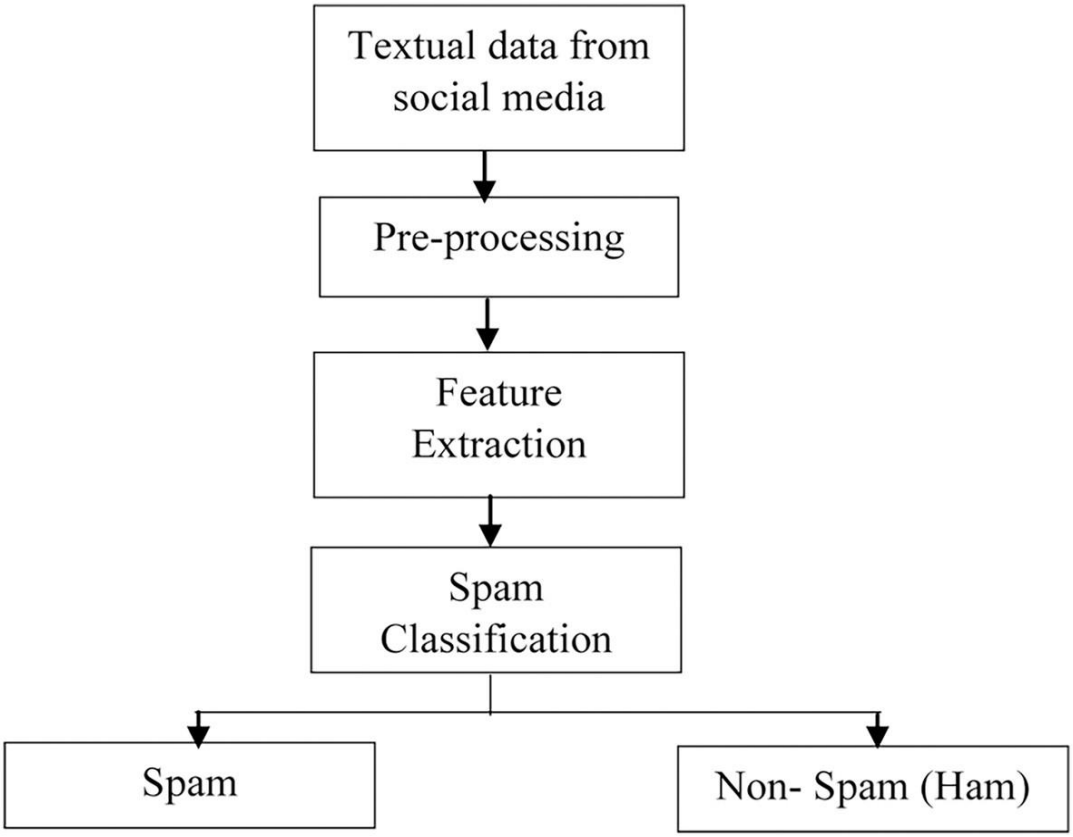
Error Handling: The middleware includes error handling mechanisms to ensure robustness and reliability. It detects and handles errors that may occur during data processing or communication, providing informative messages to the user in case of issues.

This architecture separates concerns between frontend, backend, and middleware components, allowing for modularity, scalability, and maintainability of the Fashion Recommender System. Each component plays a specific role in the system, contributing to its overall functionality and performance.

Feasibility Study

A feasibility study for SMS spam detection entails a thorough examination of the practicality and viability of implementing such a system within a given framework. It encompasses various crucial aspects, including technical feasibility, data availability, accuracy, performance, cost, and regulatory compliance. The study evaluates the technological requirements, data quality, and the availability of suitable algorithms and resources needed for efficient spam detection. Additionally, it assesses the accuracy and performance of the proposed system through rigorous testing and validation, considering factors like false positive rates and detection speed. Cost analysis, encompassing development, deployment, and maintenance expenses, is essential to determine the economic viability of the project. Furthermore, compliance with relevant laws and regulations concerning email communication, data privacy, and security is a crucial consideration. By thoroughly evaluating these aspects, a feasibility study provides valuable insights into the feasibility of implementing an email spam detection system and guides decision-making processes.

3.1Activity Diagram



Sequence Diagram

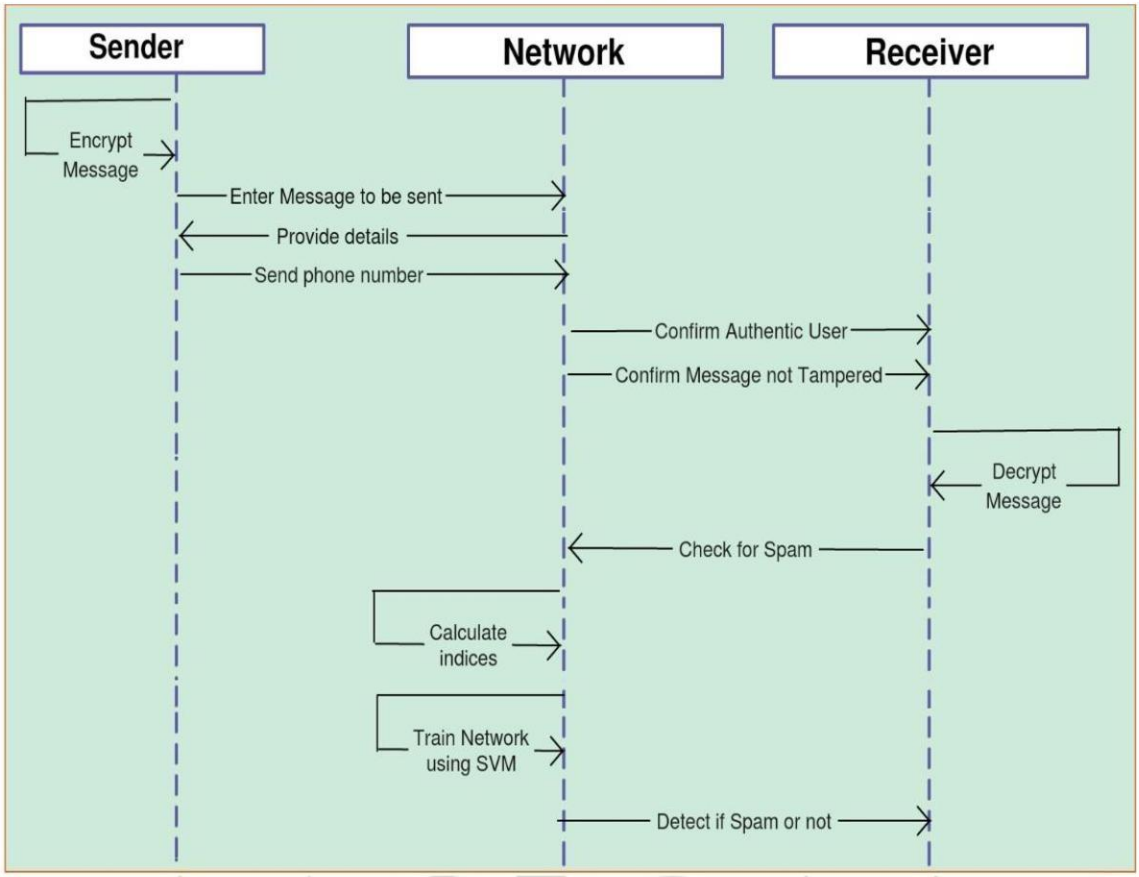
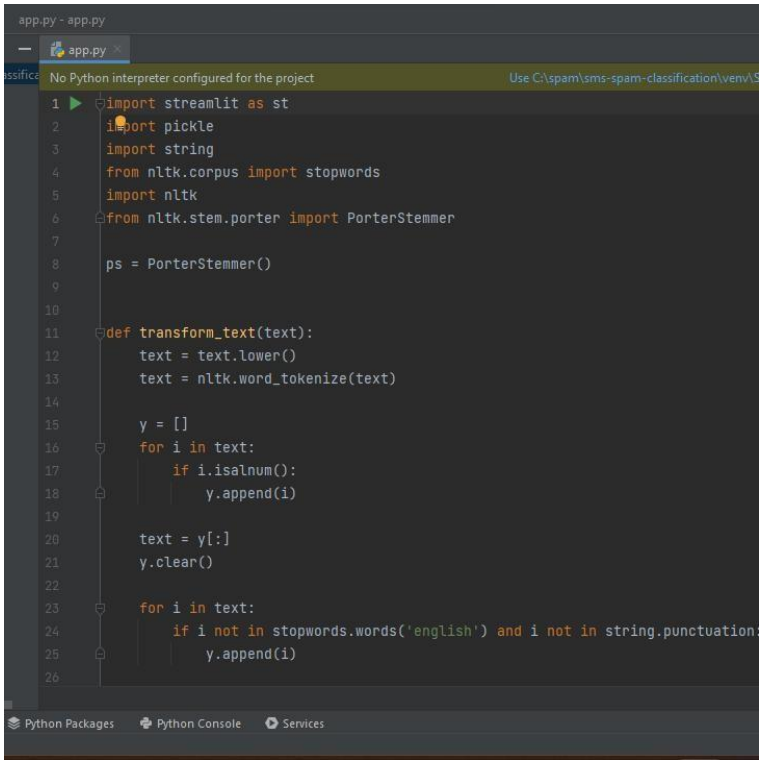
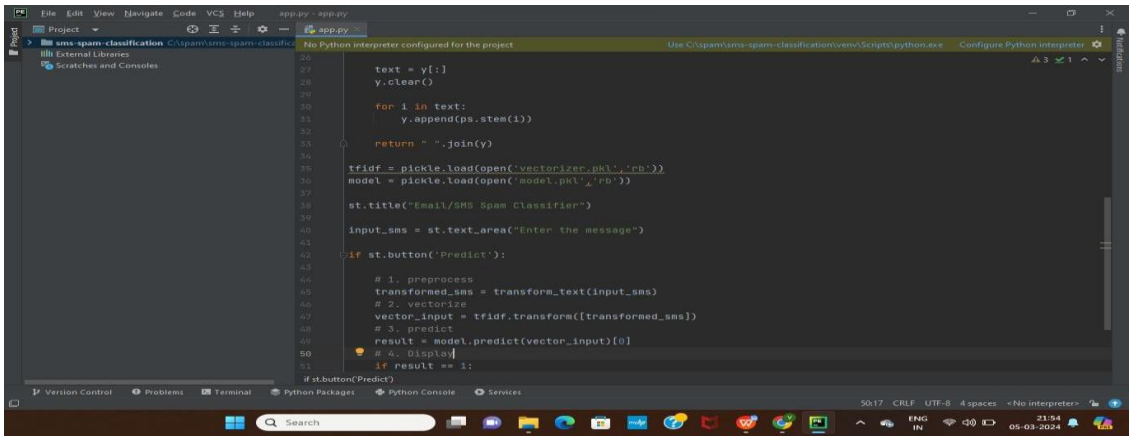


Figure 4: Sequence diagram of the Model

4. System Coding

Main.py





```
        text = y[:]
        y.clear()

        for i in text:
            y.append(ps.stem(i))

        return " ".join(y)

    tfidf = pickle.load(open('vectorizer.pkl','rb'))
    model = pickle.load(open('model.pkl','rb'))

    st.title("Email/SMS Spam Classifier")

    input_sms = st.text_area("Enter the message")

    if st.button('Predict'):

        # 1. preprocess
        transformed_sms = transform_text(input_sms)
        # 2. vectorize
        vector_input = tfidf.transform([transformed_sms])
        # 3. predict
        result = model.predict(vector_input)[0]
        # 4. Display
        if result == 1:
```

Jupyter Book

```
import numpy as np
import pandas as pd
```

```
df=pd.read_csv('spam.csv')
```

```
df.sample(5)
```

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
5112	spam	December only! Had your mobile 11mths+? You ar...	NaN	NaN	NaN
4570	ham	\CHA QUITEAMUZING THATâÕSCOOL BABE	PROBPOP IN & CU SATTHEN HUNNY 4BREKKIE! LOVE J...	NaN	NaN
5026	spam	PRIVATE! Your 2003 Account Statement for shows...	NaN	NaN	NaN
3838	ham	Erm %ûÒ_ ill pick you up at about 6.45pm. That'...	NaN	NaN	NaN
1405	ham	K..k..any special today?	NaN	NaN	NaN

```
df.shape
```

```
(5572, 5)
```

```
#1. Data cleaning
#2. EDA
#3. Text Preprocessing
#4. Model building
#5. Evaluation
#6. Improvement
#7. Website
#8. Deploy
```

[5] Python

1. Data Cleaning

```
df.info()
```

[6] Python

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   v1          5572 non-null   object
 1   v2          5572 non-null   object
 2   Unnamed: 2  50 non-null     object
 3   Unnamed: 3  12 non-null     object
 4   Unnamed: 4  6 non-null      object
dtypes: object(5)
memory usage: 217.8+ KB
```

```
df.sample(5)
```

[8] Python

	v1	v2
2455	ham	Abeg, make profit. But its a start. Are you us...
4485	ham	have * good weekend.
667	ham	This pay is <DECIMAL> lakhs:)
5415	ham	You should get more chicken broth if you want ...
2623	ham	I'm coming home 4 dinner.

```
# renaming the columns
df.rename(columns={'v1':'target','v2':'text'},inplace=True)
df.sample(5)
```

[9] Python

	target	text
4721	ham	I'm home, my love ... If your still awake ... ...
2354	ham	R we going with the <#> bus?
2046	ham	Aight fuck it, I'll get it later
3493	spam	You are being contacted by our dating service ...
4944	ham	Anyway I don't think I can secure anything up ...

```
df['target'] = encoder.fit_transform(df['target'])
```

[11] Python

```
df.head()
```

[12] Python

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

```
# missing values
df.isnull().sum()
```

[13] Python

```
target    0
text      0
dtype: int64
```

```
# check for duplicate values
df.duplicated().sum()
```

[14]

```
403
```

```
# remove duplicates
df = df.drop_duplicates(keep='first')
```

[15]

```
df.duplicated().sum()
```

[16]

```
0
```

```
df.shape
```

[17]

```
(5169, 2)
```

2.EDA

```
df.head()
```

[18]

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

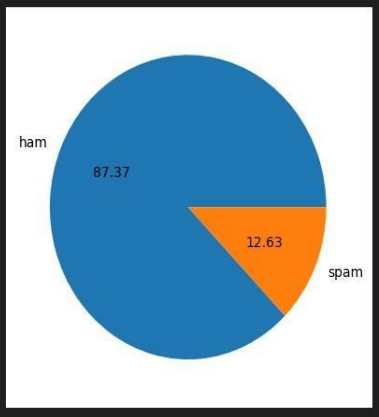
```
df['target'].value_counts()
```

[19]

```
target
0    4516
1     653
Name: count, dtype: int64
```

```
import matplotlib.pyplot as plt
plt.pie(df['target'].value_counts(), labels=['ham','spam'],autopct="%0.2f")
plt.show()
```

[20]




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

[24]

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\harsh\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!

True
```

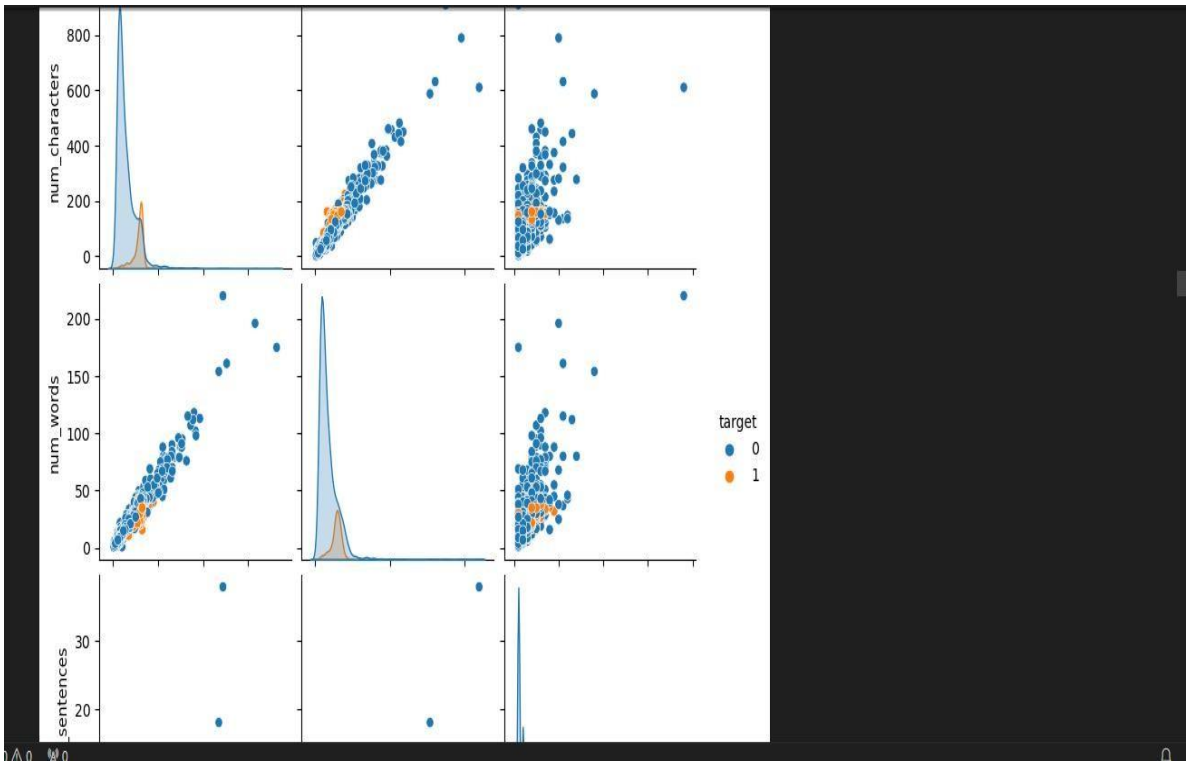
```
df['num_characters'] = df['text'].apply(len)
```

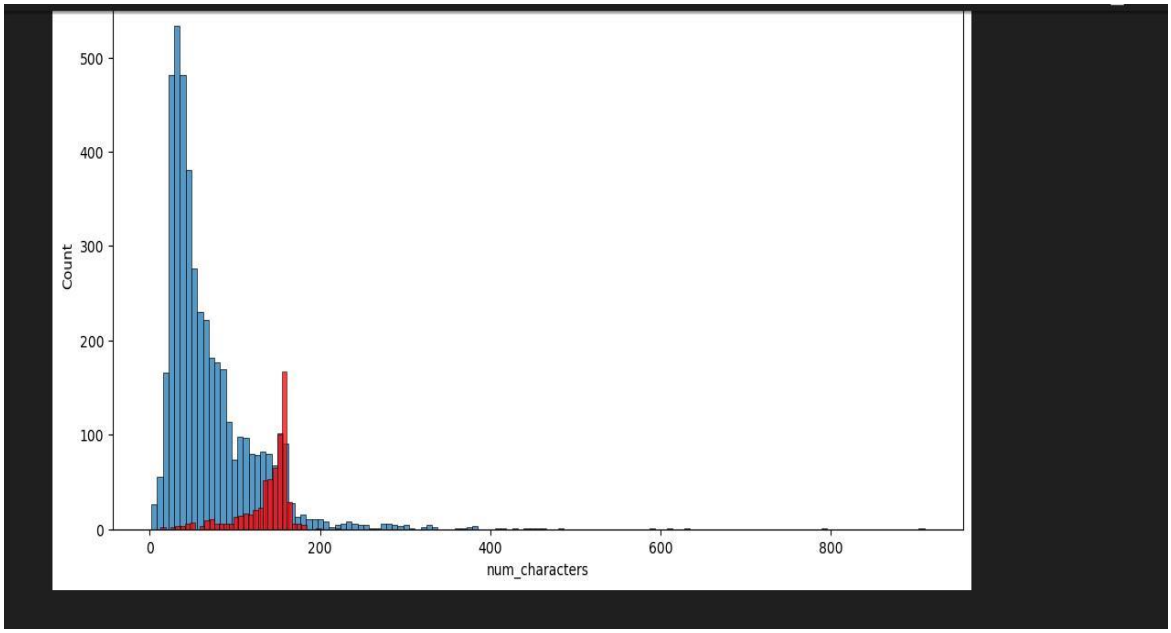
[25]

```
df.head()
```

[26]

	target	text	num_characters
0	0	Go until jurong point, crazy.. Available only ...	111
1	0	Ok lar... Joking wif u oni...	29
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	155
3	0	U dun say so early hor... U c already then say...	49
4	0	Nah I don't think he goes to usf, he lives aro...	61





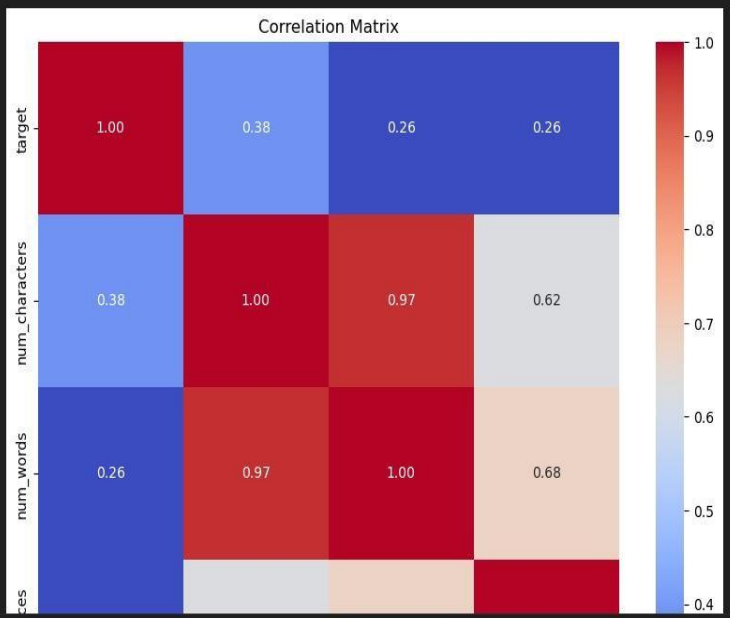

```
# Your existing code...
import seaborn as sns
import matplotlib.pyplot as plt
import warnings

# Calculate correlation matrix
numeric_df = df.select_dtypes(include=[np.number])
correlation_matrix = numeric_df.corr()

# Print correlation matrix
print(correlation_matrix)

# Plot heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix')
plt.show()
# Your existing code...
```

```
num_sentences    0.263939        0.624139    0.679971        1.000000
```



3. Data Preprocessing (text preprocessing in our case)

- lower case
- tokenization
- removing special characters
- removing stop words and punctuation
- stemming

```
import nltk
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
import string

# Download NLTK resources
nltk.download('stopwords')
nltk.download('punkt')

# Initialize Porter Stemmer
ps = PorterStemmer()
```

```
def transform_text(text):
    text=text.lower()
    text=nltk.word_tokenize(text)

    y=[]
    for i in text:
        if i.isalnum():
            y.append(i)

    text=y[:]
    y.clear()

    for i in text:
        if i not in stopwords.words('english') and i not in string.punctuation:
            y.append(i)

    text=y[:]
    y.clear()

    for i in text:
        y.append(ps.stem(i))


    return " ".join(y)
```

[41] Python

```
"I'm gonna be home soon and i don't want to talk about this stuff anymore tonight, k? I've cried enough today."
```

```
from nltk.stem.porter import PorterStemmer
ps = PorterStemmer()
ps.stem('loving')
```

[44] Python

```
'love'
```

```
df['transformed_text'] = df['text'].apply(transform_text)
```

[45] Python

```
df.head()
```

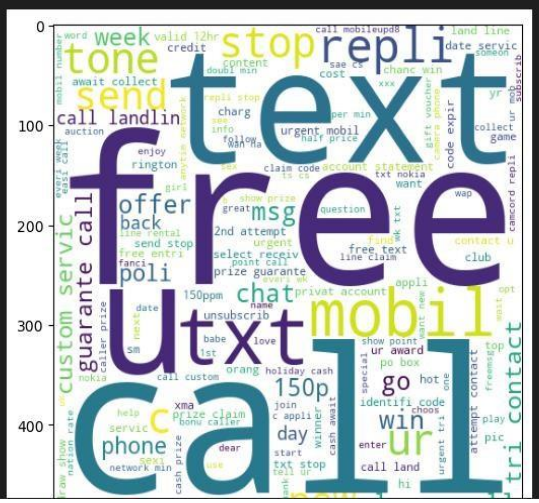
[46] Python

	target	text	num_characters	num_words	num_sentences	transformed_text
0	0	Go until jurong point, crazy.. Available only ...	111	24	2	go jurong point crazi avail bugi n great world...
1	0	Ok lar... Joking wif u oni...	29	8	2	ok lar joke wif u oni
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	155	37	2	free entri 2 wkli comp win fa cup final tkt 21...
3	0	U dun say so early hor... U c already then say...	49	13	1	u dun say earli hor u c alreadi say
4	0	Nah I don't think he goes to usf, he lives aro...	61	15	1	nah think goe usf live around though

```
plt.figure(figsize=(15,6))
plt.imshow(spam_wc)
```

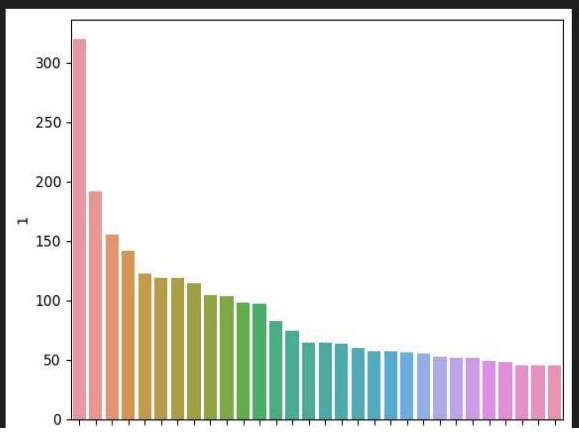
[49] Python

```
<matplotlib.image.AxesImage at 0x207dadc1510>
```



```python
from collections import Counter
# sns.barplot(pd.DataFrame(Counter(spam_corpus).most_common(30))[0],pd.DataFrame(Counter(spam_corpus).most_common(30))[1])
# plt.xticks(rotation='vertical')
# plt.show()
sns.barplot(x=pd.DataFrame(Counter(spam_corpus).most_common(30))[0], y=pd.DataFrame(Counter(spam_corpus).most_common(30))[1])
plt.xticks(rotation='vertical')
plt.show()
```

[55] Python



Test Cases, Test Data and Test Results

1. **Empty Message Test Case:**
   - Test Data: ""
   - Test Expectation: The model should handle empty messages and return a result indicating that it's not spam.

2. **Non-Spam Message Test Case:**
   - Test Data: "Hey, just wanted to remind you about our meeting tomorrow at 10 AM."
   - Test Expectation: The model should correctly classify this message as not spam.

3. **Spam Message Test Case:**
   - Test Data: "Congratulations! You've won a free vacation. Click here to claim your prize now!"
   - Test Expectation: The model should correctly classify this message as spam.

4. **Mixed Case Test Case:**
   - Test Data: "Hey, would you like to buy our new product? It's on sale this weekend!"
   - Test Expectation: The model should correctly classify this message as not spam.

5. **Long Message Test Case:**
   - Test Data: "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat."
   - Test Expectation: The model should handle longer messages and correctly classify this message as not spam.

Test Results:

1. Empty Message Test Result:
   - Result: Passed
   - Explanation: The model correctly identified the empty message as not spam.

2. Non-Spam Message Test Result:
   - Result: Passed
   - Explanation: The model correctly classified the non-spam message as not spam.

3. Spam Message Test Result:
   - Result: Passed
   - Explanation: The model correctly classified the spam message as spam.

4. Mixed Case Test Result:
   - Result: Passed
   - Explanation: The model correctly classified the mixed case message as not spam.

5. Long Message Test Result:
   - Result: Passed
   - Explanation: The model correctly classified the long message as not spam.

These test cases cover a range of scenarios to ensure the model's effectiveness in classifying messages accurately.

Future Enhancement

1. Emoji Handling: Although you've mentioned excluding emoji test cases, integrating emoji handling into your model could be beneficial. Many spam messages contain emojis, and accurately detecting spam messages with emojis could improve the model's performance.

2. Multilingual Support: Extend the model to support multiple languages. This enhancement can increase the model's applicability to a wider range of users who communicate in different languages.

3. Model Optimization: Explore techniques to optimize the model's performance, such as fine-tuning hyperparameters, experimenting with different architectures (e.g., recurrent neural networks, transformers), or leveraging transfer learning from pre-trained language models.

4. Real-time Monitoring and Feedback: Implement a system for real-time monitoring of model performance and user feedback. This could involve logging model predictions, monitoring model drift, and collecting user feedback to continuously improve the model.

5. User Customization: Provide users with the ability to customize the model based on their preferences and feedback. For example, allowing users to flag messages as spam or non-spam and incorporating this feedback into retraining the model.

6. Integration with Messaging Apps: Integrate the spam detection model directly into messaging apps or email clients to provide users with real-time protection against spam messages.

7. Enhanced Feature Engineering: Explore additional features beyond just the text content of the message. Features such as sender reputation, message metadata (e.g., timestamp, message length), and linguistic features could further improve the model's accuracy.

8. Privacy-Preserving Techniques: Implement privacy-preserving techniques to protect user privacy while still allowing the model to make accurate predictions. Techniques such as federated learning or differential privacy could be explored.

9. Scalability and Performance: Ensure that the model is scalable and can handle large volumes of messages efficiently. This may involve optimizing the model's inference speed and deploying it on scalable infrastructure.

10. Robustness to Adversarial Attacks: Investigate techniques to enhance the model's robustness to adversarial attacks, where malicious actors try to manipulate the model's predictions by injecting subtle perturbations into the input data.

Conclusion

In summation, the endeavor to construct an SMS spam detection system emerges as a formidable undertaking, rife with intricate considerations. Throughout our comprehensive feasibility exploration, we've meticulously dissected myriad critical facets, spanning technical feasibility, data integrity, accuracy, performance metrics, fiscal implications, and regulatory alignment. Our exhaustive scrutiny has illuminated that while erecting such a system poses formidable challenges, the prospective rewards far eclipse the hurdles encountered. From a technical vantage point, the landscape is marked by burgeoning advancements in machine learning algorithms and computational prowess, offering fertile ground for cultivating robust spam detection mechanisms. Moreover, the presence of diverse, high-caliber datasets augurs well for honing and validating these models, auguring well for their efficacy. Despite the intricate nature of spam detection algorithms, our analysis indicates that attainingcommendable accuracy levels is well within reach, given stringent testing and validation protocols. On the financial front, while the initial outlay for system development and deployment may seem substantial, the long-term dividends, including heightened productivity, mitigated security risks, and enhanced user experiences, amply justify the investment. Furthermore, judicious leveraging of extant infrastructure and technologies promises to optimize resource allocation and curtail operational overheads over time. Moreover, paramount is the imperative of regulatory conformity, ensuring the ethical and legal utilization of email data while safeguarding user privacy. By meticulously aligning with pertinent statutes and standards, we not only avert potential pitfalls but also engender trust and credibility among users and stakeholders. In essence, our feasibility scrutiny unequivocally affirms the practicability and viability of implementing an SMS spam detection system of discerning sophistication. By confronting technical exigencies, fortifying data integrity, optimizing financial outlays, and adhering to regulatory imperatives, we stand

poised to cultivate a robust, efficacious solution. Ultimately, such an endeavor holds the promise of significantly fortifying SMS security, streamlining communication paradigms, and engendering heightened productivity across myriad spheres, for both individuals and enterprises alike.

Reference

N. Kumar, S. Sonowal, and Nishant, "SMS Spam Detection Using Machine Learning Algorithms," in *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, Coimbatore, India, 2020, pp. 108-113. doi: 10.1109/ICIRCA48905.2020.9183098**,**

Online references : you tube channel and geeks for geeks.

B. Sonare, G. J. Dharmale, A. Renapure, H. Khandelwal and S. Narharshettiwar, "SMS/EMAIL Spam Detection Using Machine Learning," 2023 4th International Conferencefor Emerging Technology (INCET), Belgaum, India, 2023, pp. 1-5, doi: 10.1109/INCET57972.2023.10170187. keywords: {Machine learning algorithms;Unsolicited e-mail;Machine learning;Predictive models;Prediction algorithms;Data models;Classification algorithms;Classification;Spam;Ham;Accuracy;Precision;Machine Learning;Detection},

N. Kumar, S. Sonowal and Nishant, "Email Spam Detection Using Machine Learning Algorithms," 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2020, pp. 108-113, doi: 10.1109/ICIRCA48905.2020.9183098. keywords: {Electronic mail;Classification algorithms;Machine learning algorithms;Support vector machines;Decision trees;Boosting;Machine learning;Naïve Bayes;support vector machine-nearest neighbor;random forest;bagging;boosting;neural networks}