Email Spam Detection in the Age of Cyber Threats: A Comprehensive Approach Integrating Cleaning, Analysis, and Modeling

Dr. Manju Arora[1], Chetan Jangid[2], Abhishek Rai[3]

[1]Department of Information Technology, Jagan Institute of Management Studies, Rohini, Delhi, India
[2,3]Department of Information Technology, Jagan Nath University, Bahadurgarh, Haryana, India

Abstract:
With the escalating sophistication of cyber threats, the imperative task of safeguarding email communication from spam has become increasingly challenging. This research presents a holistic approach to email spam detection, incorporating meticulous data cleaning, in-depth exploratory data analysis (EDA), and advanced modeling techniques. The study delves into the performance evaluation of diverse machine learning classifiers, including Gaussian Naive Bayes, Multinomial Naive Bayes, and Bernoulli Naive Bayes. Exploratory visualizations, such as word clouds and bar plots, contribute to a comprehensive understanding of the textual content in spam and ham messages.

Keywords: GNB, NB, BNS, SVM

1. Introduction

In the contemporary landscape of digital communication, the ubiquitous use of email has become an integral part of daily interactions. However, this convenience is not without its challenges, particularly with the ever-growing sophistication of cyber threats. Among these threats, email spam stands out as a pervasive and persistent issue that demands vigilant attention. As cybercriminals continually adapt their tactics, the imperative task of identifying and filtering spam has become increasingly complex.

This research endeavors to address the multifaceted challenge of email spam detection by presenting a comprehensive approach that integrates meticulous data cleaning, in-depth exploratory data analysis (EDA), and advanced machine learning modeling techniques. The escalating sophistication of cyber threats necessitates an approach that not only understands the intricacies of the data but also leverages state-of-the-art machine learning methodologies to counteract evolving adversarial tactics.

The initial phase of the study involves a meticulous data cleaning process aimed at fortifying the integrity of the dataset. Addressing missing values and eliminating duplicates lays the groundwork for subsequent analyses, ensuring that the conclusions drawn are based on a robust and reliable foundation. The subsequent exploratory data analysis offers insights into the inherent imbalances within the dataset, providing a nuanced understanding of the distribution of characters, words, and sentences in both spam and non-spam (ham) messages.

A strong text preparation pipeline is put in place since it is realized that the quality of the input data has a significant impact on how well machine learning models perform. Lowercasing, tokenization, removal of special characters, stop words, punctuation, and stemming collectively contribute to a refined and standardized textual dataset. The transformed text data undergoes vectorization using the Term Frequency-Inverse Document Frequency (TF-IDF) technique, enhancing the model's ability to discern relevant features amidst the complexity of email content.

The heart of the research lies in the model building phase, where a diverse set of machine learning classifiers is evaluated for their performance in spam detection. GNB, MN NB, BNS, SVM, KNN, RF, XGBoost, and ensemble learning techniques such as Voting Classifier and Stacking Classifier all come under scrutiny. Performance metrics, including accuracy and precision, serve as benchmarks for assessing the efficacy of each model, providing valuable insights into their suitability for practical applications.

The research culminates in the deployment of a well-performing model that achieves a delicate balance between accuracy and precision. Specifically, the model, grounded in the Multinomial Naive Bayes algorithm, is offered alongside the TF-IDF vectorizer, providing a tangible resource for those seeking to implement robust and efficient email spam detection systems. As cyber threats continue to evolve, the insights gleaned from this research contribute significantly to the ongoing discourse in the realms of cybersecurity and machine learning, offering a proactive approach to the persistent challenge of email spam in the age of cyber threats.

2. Literature Survey

The literature survey you provided offers valuable insights into various approaches for email spam detection, ranging from traditional methods to advanced machine learning and deep learning techniques.

Here's an optimistic summary of the key findings and contributions from the surveyed literature:

Identification of Spam Traits Across Email Providers :The authors proposed criteria based on email headers for effective spam identification. The study aimed at creating a universal spam message detection system applicable to major email providers, promoting a standardized approach.

Novel Method Based on Word Repetition Frequency :Introduced a novel method using word repetition frequency for categorizing emails.

Leveraged important sentences containing keywords and applied the K-Mean algorithm for efficient categorization.

Addressed the issue of cyberattacks and phishing, highlighting the importance of email categorization in preventing personal credential acquisition.

Utilization of Bayesian Classifiers : Employed Bayesian classifiers for identifying spam emails and adjusting to new spam types.

Emphasized the acquisition of personal credentials and the role of BCc in the classification process.

Contributed to the ongoing efforts to combat malicious actors using email systems.

Pattern Recognition and Machine Learning :Proposed a solution to identify recurring patterns flagged as spam using machine learning techniques.

Considered various factors such as header, domain, and Cc/Bcc in the classification process. Offered an alternate architecture for spam filtering by incorporating features and a pre-trained machine learning model.

Compared the effectiveness of six popular algorithms and found the Bi-gram algorithm to outperform others.

Contributed to the understanding of algorithmic approaches for detecting spam in different datasets.

Advancements in Deep Learning :Srinivasan et al. showcased the impact of word embedding in deep learning for email spam detection, outperforming traditional representation techniques.

Soni introduced a profound learning model, THEMIS, achieving a high accuracy of 99.84% for identifying phishing emails.

Hassanpur et al. utilized word2vec for email representation and achieved an accuracy rate of over 96%, surpassing industry-standard techniques.

NLP Approaches and Ensemble Learning :Egozi et al. applied NLP approaches to identify phishing emails, achieving accuracy with an ensemble learning model. Seth et al. emphasized the importance of comparing accuracy and datasets in spam filtering algorithms, contributing to the evaluation of current studies.

3. Proposed system

The proposed system introduces a robust spam categorization system designed to effectively distinguish between spam and ham messages. The primary challenge of manually identifying spam messages, especially when spammers send them repeatedly, is addressed through automated techniques.

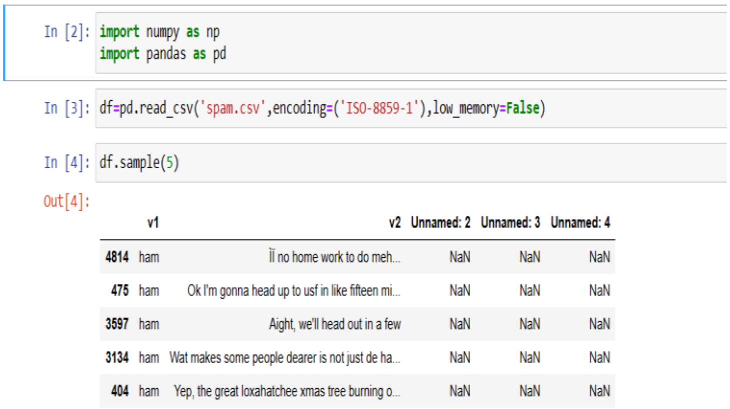
Spam Categorization Module:

1. Login: The user will enter his encrypted name and password to access the main page. The approved page will appear once the user logs in successfully; if not, error messages will be presented.
2. Compose Input: the sender will compose the new email message based on the category, Message automatically delivered to the spam\ham section.
3. Send Email message The user's whole mailbox will be kept on this page. Every email that is received will be listed and arranged according to spam and ham.
   Spam: Any inbound spam emails from an individual will be accepted in this area.
   Ham: Any incoming emails from an individual will be accepted in this area.
4. Predict this section will predict all the spam incoming emails sent by an individual, is spam or ham.
5. View spam email This section store all the spam messages sent by the individual.
6. View ham email This section store all the ham messages sent by the individual.

A message that arrives in our inbox will be exported to a dataset. The NBC will determine whether or not it believes this message is spam. The model must be trained, as described in the section below, before it can determine if an email is spam or not.

Email Spam detention using Machine learning.

1. The algorithm is trained using the Kaggle dataset, which is displayed in below:-

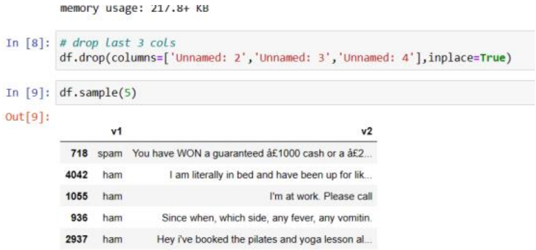


```
In [2]: import numpy as np
        import pandas as pd

In [3]: df=pd.read_csv('spam.csv',encoding=('ISO-8859-1'),low_memory=False)

In [4]: df.sample(5)

Out[4]:
```

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
4814	ham	Ì_ no home work to do meh...	NaN	NaN	NaN
475	ham	Ok I'm gonna head up to usf in like fifteen mi...	NaN	NaN	NaN
3597	ham	Aight, we'll head out in a few	NaN	NaN	NaN
3134	ham	Wat makes some people dearer is not just de ha...	NaN	NaN	NaN
404	ham	Yep, the great loxahatchee xmas tree burning o...	NaN	NaN	NaN

Figure no 1 :- Dataset

2. There are several fields in the dataset, some of which are optional. Thus, eliminate any columns that are not necessary. The column names need to be changed.

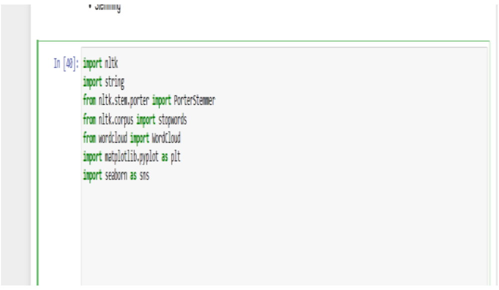


```
memory usage: 217.8+ KB

In [8]: # drop last 3 cols
        df.drop(columns=['Unnamed: 2','Unnamed: 3','Unnamed: 4'],inplace=True)

In [9]: df.sample(5)
Out[9]:
```

	v1	v2
718	spam	You have WON a guaranteed â£1000 cash or a â£2...
4042	ham	I am literally in bed and have been up for lik...
1055	ham	I'm at work. Please call
936	ham	Since when, which side, any fever, any vomitin.
2937	ham	Hey i've booked the pilates and yoga lesson al...

Figure no 2:- Classification dataset

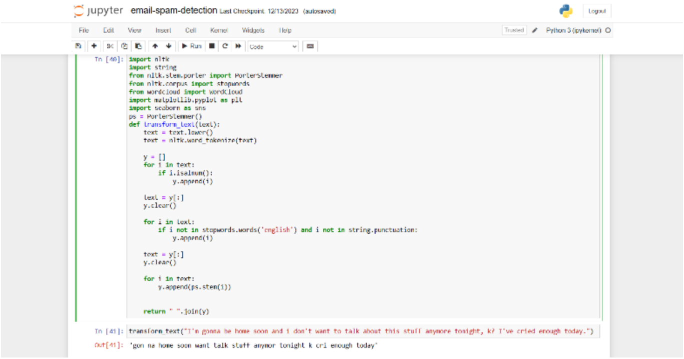
3. With the help of NLTK (Natural Language Tool Kit) for the text processing, Using Matplotlib you can plot graphs , histogram and bar plot and all those things ,Word Cloud is used to present text data and pandas for data manipulation and analysis .



```
In [40]: import nltk
         import string
         from nltk.stem.porter import PorterStemmer
         from nltk.corpus import stopwords
         from wordcloud import WordCloud
         import matplotlib.pyplot as plt
         import seaborn as sns
```

Fig no 3:- Import Libraries

4. Created a function called as transform_text is used for the removing Stopwords ,Punctuation and Stemming.

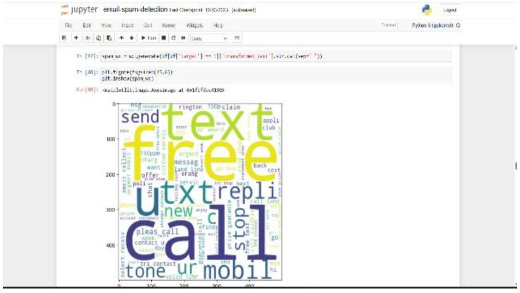


jupyter email-spam-detection Last Checkpoint: 12/13/2023 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

```
In [40]: import nltk
         import string
         from nltk.stem.porter import PorterStemmer
         from nltk.corpus import stopwords
         from wordcloud import WordCloud
         import matplotlib.pyplot as plt
         import seaborn as sns
         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)

In [41]: transform_text("I'm gonna be home soon and i don't want to talk about this stuff anymore tonight, k? I've cried enough today.")
Out[41]: 'gon na home soon want talk stuff anymor tonight k cri enough today'
```

Fig no 4: Transform_text

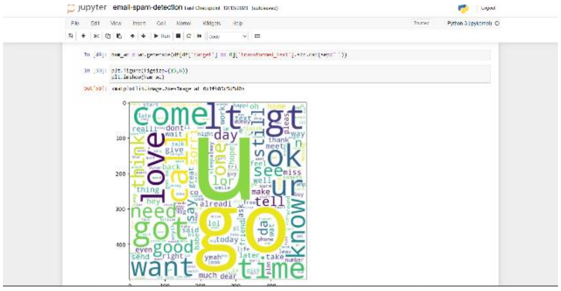
5. We need to find most repeated word in the spam detection to be used word cloud library.



```
In [47]: spam_wc = wc.generate(df[df['target'] == 1]['transformed_text'].str.cat(sep=" "))

In [48]: plt.figure(figsize=(15,6))
         plt.imshow(spam_wc)
Out[48]: <matplotlib.image.AxesImage at 0x1f5fd...4190>
```

Fig no 5:-Spam Word Cloud

6. We need to find most repeated word in the ham detection to be used word cloud library.

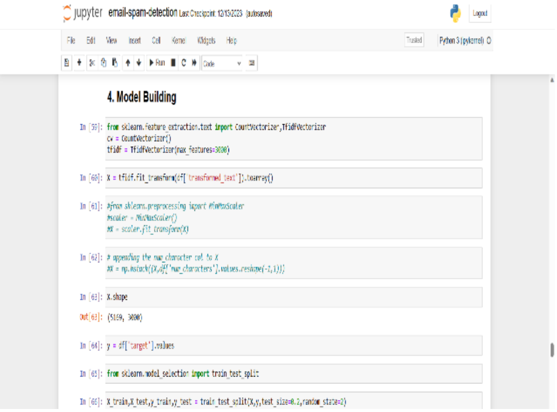


jupyter email-spam-detection Last Checkpoint: 12/13/2023 (autosaved)

```
In [49]: ham_wc = wc.generate(df[df['target'] == 0]['transformed_text'].str.cat(sep=" "))

In [50]: plt.figure(figsize=(15,6))
         plt.imshow(ham_wc)

Out[50]: <matplotlib.image.AxesImage at 0x1f6b1c5d5d0>
```

Fig no 6: - ham Word Cloud

7. Model Building

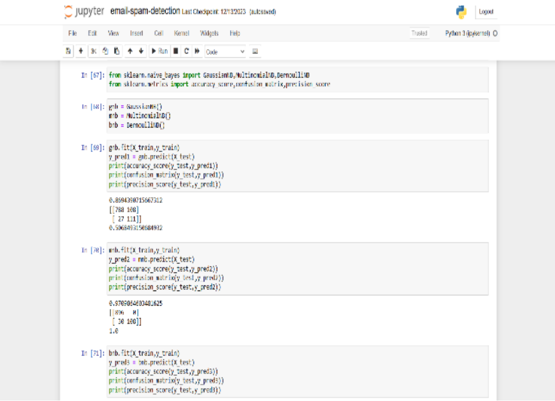


jupyter email-spam-detection Last Checkpoint: 12/13/2023 (autosaved)

4. Model Building

```
In [59]: from sklearn.feature_extraction.text import CountVectorizer,TfidfVectorizer
         cv = CountVectorizer()
         tfidf = TfidfVectorizer(max_features=3000)

In [60]: X = tfidf.fit_transform(df['transformed_text']).toarray()

In [61]: #from sklearn.preprocessing import MinMaxScaler
         #scaler = MinMaxScaler()
         #X = scaler.fit_transform(X)

In [62]: # appending the num_character col to X
         #X = np.hstack((X,df['num_characters'].values.reshape(-1,1)))

In [63]: X.shape

Out[63]: (5169, 3000)

In [64]: y = df['target'].values

In [65]: from sklearn.model_selection import train_test_split

In [66]: X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=2)
```

Fig 7: - Training



jupyter email-spam-detection Last Checkpoint: 12/13/2023 (autosaved)

```
In [67]: from sklearn.naive_bayes import GaussianNB,MultinomialNB,BernoulliNB
         from sklearn.metrics import accuracy_score,confusion_matrix,precision_score

In [68]: gnb = GaussianNB()
         mnb = MultinomialNB()
         bnb = BernoulliNB()

In [69]: gnb.fit(X_train,y_train)
         y_pred1 = gnb.predict(X_test)
         print(accuracy_score(y_test,y_pred1))
         print(confusion_matrix(y_test,y_pred1))
         print(precision_score(y_test,y_pred1))

         0.8694390715667312
         [[788 108]
          [ 27 111]]
         0.5068493150684932

In [70]: mnb.fit(X_train,y_train)
         y_pred2 = mnb.predict(X_test)
         print(accuracy_score(y_test,y_pred2))
         print(confusion_matrix(y_test,y_pred2))
         print(precision_score(y_test,y_pred2))

         0.9709864603481625
         [[896   0]
          [ 30 108]]
         1.0

In [71]: bnb.fit(X_train,y_train)
         y_pred3 = bnb.predict(X_test)
         print(accuracy_score(y_test,y_pred3))
         print(confusion_matrix(y_test,y_pred3))
         print(precision_score(y_test,y_pred3))
```

Fig no 8: - Testing

6. Methodology

The methodology employed in this research aims to rigorously evaluate and compare the performance of multiple machine learning algorithms for email spam detection. The process involves several key steps, including data preparation, feature extraction, model training, and evaluation.

- **Data Preprocessing:**

Raw email data undergoes preprocessing to enhance its suitability for machine learning. This includes:
Text Cleaning: Removal of HTML tags, special characters, and irrelevant formatting.
Tokenization: Breaking down text into individual words and sentence tokens.

Stop word Removal: Elimination of common words that do not contribute significantly to classification.
Stemming: Reducing words to their root form to capture core meanings.

- **Feature Extraction:**

The transformed text data is converted into integer features suitable for machine learning models. We employ the Term FI and DF (TF-IDF) vectorization technique to represent the textual content in a form that can be processed by the classifiers.

- **Selection of Machine Learning Classifiers:**

We consider a diverse set of machine learning classifiers to legitimate messages. The dataset is selected to represent real-world encompass various algorithmic paradigms:

1. Logistic Regression
2. Support Vector Machines
3. Naive Bayes
4. Decision Trees
5. k-Nearest Neighbors
6. Random Forest
7. AdaBoost
7. Bagging
8. Extra Trees
9. Gradient Boosting
10. XGBoost

- **Evaluation Metrics:**

To assess the performance of each model, we employ a set of standard evaluation metrics:
Precision: Proportion of correct positive predictions among all positive predictions.
Recall: Proportion of true positive predictions among true positive cases.
F1-score: Harmonic mean of precision and recall, providing a balanced metric.
Accuracy: Proportion of correctly classified instances among the total instances.

- **Model Training:**

Each selected classifier is trained on the pre-processed and vectorized indicative of spam or non-spam emails.

- **Cross-Validation:**

To mitigate the risk of overfitting, we perform k-fold cross-validation, where the dataset is partitioned into k subsets, and the model is trained and evaluated k times, using a different subset for evaluation in each iteration.

- **Hyperparameter Tuning:**

Hyperparameter tuning is conducted to optimize the performance of each classifier. Grid search or randomized search techniques are employed to find the best combination of hyperparameter values.

- **Comparison and Analysis:**

The performance metrics obtained from each classifier are thoroughly compared and analyzed. Insights into the strengths and weaknesses of each algorithm are derived from this comparative analysis.

- **Visualization:**

Visualizations, including confusion matrices and ROC curves, are generated to provide a more intuitive understanding of the models' performance.

- **Implementation in Streamlit Application:**

The best-performing model, determined through the comparative analysis, is implemented in a Streamlit application. The application facilitates real-time spam detection and provides an interactive interface for users to classify emails.dataset. The training process involves optimizing model parameters to enhance predictive accuracy. The training set is used to teach the model to discern patterns.

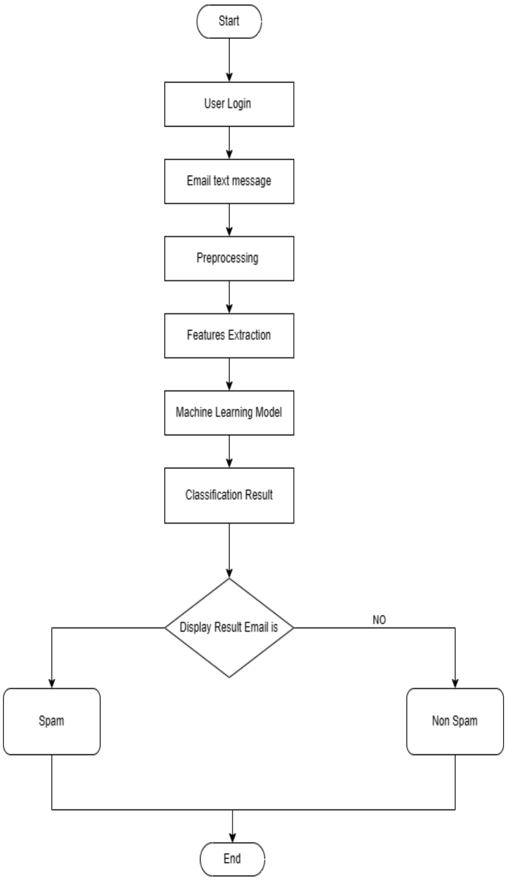


Fig:- 9 Methodology

7. ALGORITHM

The implementation of multiple ML algorithms in your research paper allows for a comprehensive exploration of their effectiveness in email spam detection. Below is a detailed breakdown of the algorithms you've included in your code:

- **Logistic Regression:**

LR that works well for binary classification applications is logistic regression. When the connection between characteristics and the log-odds of being spam is considered to be linear, it is well-suited for spam detection as it represents the chance that an instance belongs to a certain class.

- **SVM**

SVM is a powerful classification algorithm that works well for both linear and non-linear data. It attempts to find the hyperplane that best separates the data into different classes. SVM can effectively capture complex relationships in high-dimensional spaces.

- **Naive Bayes:**

NB is a personality classifier based on Bayes' theorem. Despite its simplicity and the assumption of features independence (naivety), it often performs well in practice. It is particularly efficient for text classification tasks like spam detection.

- **Decision Trees:**

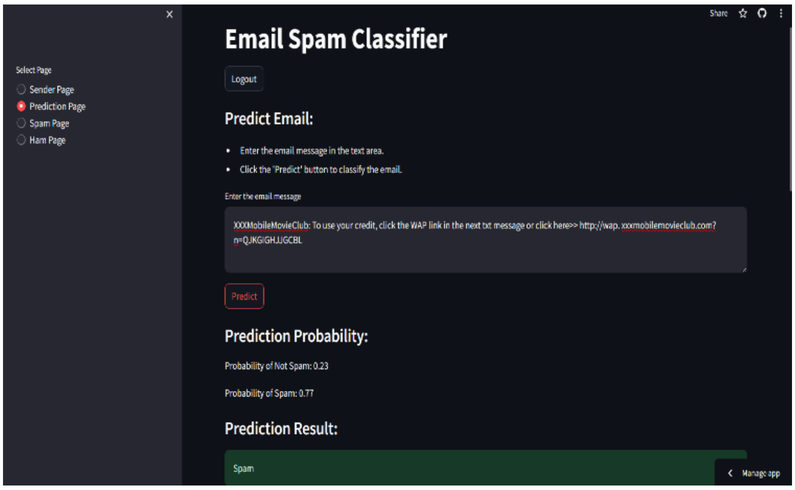
Decision Trees are tree-like models that make decisions based on features' values. They are easy to interpret and understand. Decision Trees can capture complex decision boundaries and are prone to overfitting, which ensemble methods like Random Forest aim to mitigate.

8. RESULTS AND DISCUSSIONS

When we receive message in the ham and spam mails, that message will be as shown below. This message will be detected as spam or not.

Using Bayes' theorem and Naive Bayes' classifier, the exported message will be classified as spam or not once all the previously mentioned procedures are completed. The likelihood of words in spam and ham messages will also be determined in order to make this determination. The messages that were identified as spam and ham are depicted in the figures below. In the event that the message is exported from the inbox to the dataset, it can be determined that it is spam using the trained dataset, the Bayes theorem, and the Naive Bayes classifier. The results are displayed below.

1. Predict this section



Email Spam Classifier

Predict Email:

- Enter the email message in the text area.
- Click the 'Predict' button to classify the email.

Enter the email message

XXXMobileMovieClub: To use your credit, click the WAP link in the next txt message or click here>> http://wap. xxxmobilemovieclub.com?n=QJKGIGHJJGCBL

Prediction Probability:

Probability of Not Spam: 0.23

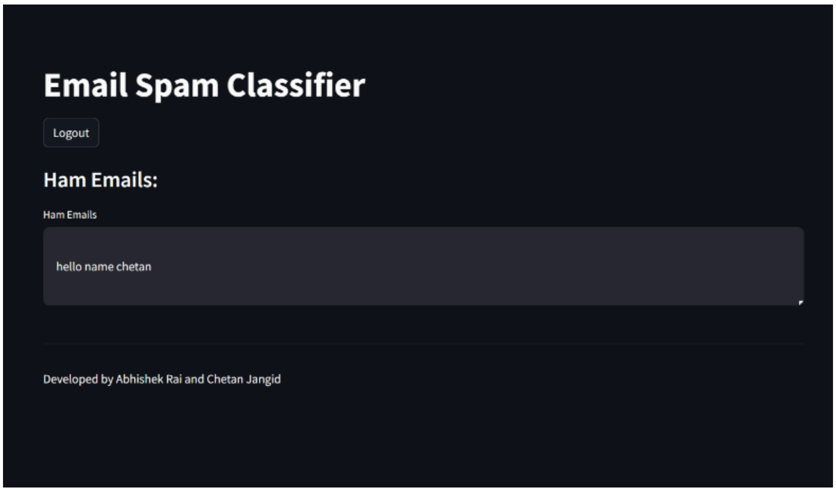
Probability of Spam: 0.77

Prediction Result:

Spam

Fig no:- 10 Predict email

2. View ham email



Email Spam Classifier

Ham Emails:

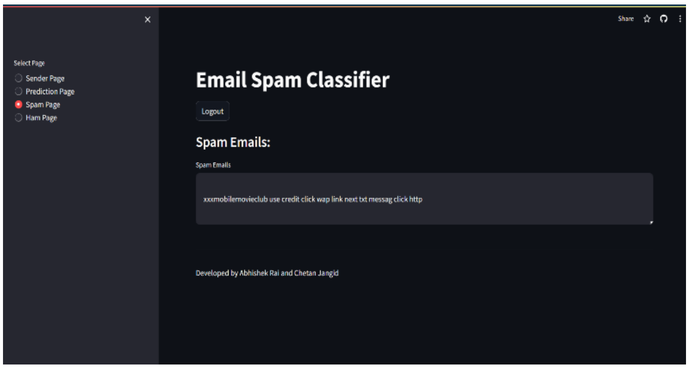
Ham Emails

hello name chetan

Developed by Abhishek Rai and Chetan Jangid

Fig No:- 11 Ham Emails

3. **View spam email**



Select Page
○ Sender Page
○ Prediction Page
● Spam Page
○ Ham Page

Email Spam Classifier

Spam Emails:

Spam Emails

xxxmobilemovieclub use credit click wap link next txt messag click http

Developed by Abhishek Rai and Chetan Jangid

Fig No :-12 Spam Emails

9. Conclusion:

This research introduces a holistic approach to email spam detection amidst escalating cyber threats. Meticulous data cleaning, exploratory analysis, and advanced modeling techniques were employed to fortify defenses. Diverse machine learning classifiers, including Gaussian Naive Bayes and ensemble methods, were rigorously evaluated, with the Multinomial Naive Bayes-based model emerging as a resilient choice.

The deployment of this model, accompanied by the TF-IDF vectorizer, offers a practical resource for robust email spam detection systems. The insights gained contribute not only to current cybersecurity understanding but also provide guidance for future developments in countering evolving cyber threats within email communication.

1. Connect this project with API: Connecting a spam and ham email detection with an API (Application Programming Interface) can provide several benefits and enhance the overall functionality and usability of the system.
2. Dynamic Mail reading: Make the project Dynamic in nature which involves more user’s interactions, currently the project is in the static form.
3. Deploy this project with the help of AWS: In future connect the project with AWS for better User Interface.
4. Letting the User Customize the Website: Giving you more control to set how strict or lenient you want your spam filter to be.
5. Looking at More Than Just Text: Checking not just written words but also images, videos, voice and other things in emails to catch tricky spam.

10. References:

1. Thashina Sultana, K A Sapnaz, Fathima Sana, Mrs. Jamedar Najath.,” Email based Spam Detection”, International Journal of Engineering Research & Technology (IJERT)ISSN: 2278-0181,2020.
2. H. Faris, A. M. Al-Zoubi, A. A. et al., “An intelligent system for spam detection and identification of the most relevant features based on evolutionary random weight networks,” *Information Fusion*, vol. 48, pp. 67–83, 2019.
3. Cihan Varol, Hezha M.Tareq Abdulhadi “Comparison of String Matching Algorithms on Spam Email Detection”, International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism Dec, 2018.

4. Duan, Lixin, Dong Xu, and Ivor Wai-Hung Tsang. "Domain adaptation from multiple sources: A domaindependent regularization approach." IEEE Transactions on Neural Networks and Learning Systems 23.3 (2012).
5. K. S. Adewole, N. B. Anuar, A. Kamsin, K. D. Varathan, and S. A. Razak, “Malicious accounts: dark of the social networks,” *Journal of Network and Computer Applications*, vol. 79, pp. 41–67, 2017.
6. N. Kumar and S. Sonowal, “Email spam detection using machine learning algorithms,” in *Proceedings of the 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, pp. 108–113, Coimbatore, India, 2020.
7. HarjotKaur,Er.PrinceVerma:International Journal of Engineering Sciences and Research Technology Survey on “Email Spam Detection using Supervised Approach with Feature Selection” : April,2017
8. ”Analysis of Naïve Bayes Algorithm for Email Spam Filtering across MultipleDatasets”: Nurul Fitriah Rusland, Norfaradilla Wahid, Shahreen Kasim and Hanayanti, Volume 226, International Research and Innovation Summit (IRIS2017) 6–7 May 2017, Melaka,Malaysia
9. Harisinghaney, Anirudh, Aman Dixit, Saurabh Gupta, and Anuja Arora. "Text and image-based spam email classification using KNN, Naïve Bayes and Reverse DBSCAN algorithm." In Optimization, Reliabilty, and Information Technology (ICROIT), 2014 International Conference on, pp.153-155. IEEE, 2014
10. Mohamad, Masurah, and Ali Selamat. "An evaluation on the efficiency of hybrid feature selection in spam email classification."
11. In Computer, Communications, and Control Technology (I4CT), 2015 International Conference on, pp. 227-231. IEEE, 2015
12. Shradhanjali, Prof. Toran Verma “E-Mail Spam Detection and Classification Using SVM and Feature Extraction”in International
13. 7. W.A, Awad & S.M, ELseuofi. (2011). Machine Learning Methods for Spam E-Mail Classification. International Journal of Computer Science & Information Technology. 3. 10.5121/ijcsit.2011.3112.