**OPTIMIZING SPAM FILTERING WITH MACHINE LEARNING**

1. **INTRODUCTION**
   1. **OVERVIEW**

**(A BRIEF DESCRIPTION ABOUT THIS PROJECT).**

Spamming - as the abuse of electronic messaging systems - has become a real problem in recent years. Spam is in several ways not harmless. The least it does is taking away bandwidth from the internet users and time to process it. According to studies undertaken by M86Security, the ascending trend that European IT has lately followed (as compared to North America) involves not only positive aspects; with it came less pleasant phenomena like spam. If until recently America was the leading provider of spam, the European servers have now registered the largest source of spam. The increase in the amount of spam sent by the European servers is actually a natural consequence of the connection’s amplified speed. Email spam refers to sending irrelevant, inappropriate and unsolicited email messages to numerous people. This is possible due to the low entrance barrier and the low cost of sending emails, which makes it one of the most popular forms of spam. The purpose of email spam is advertising, promotion, and spreading backdoors or malicious programs. Currently, Phishing is also considered as one of the main goals of spammers when employing email spams.

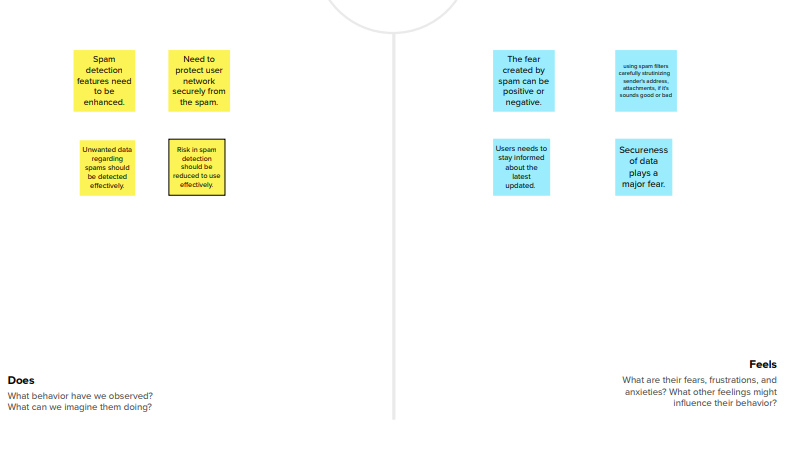
**1.2 PURPOSE**

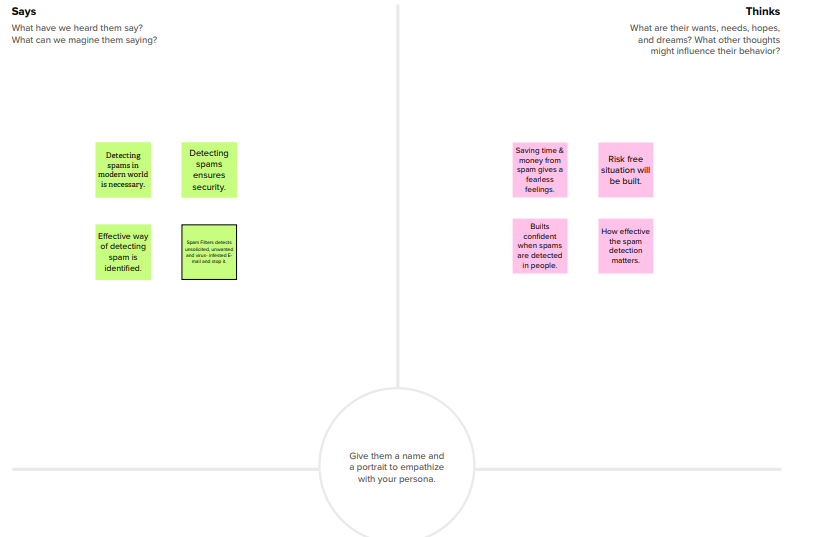
In recent years, anti-spam filters have become necessary tools for Internet service providers to tackle the continuously growing spam phenomenon. Current server side anti-spam filters are made up of several modules aimed at detecting different features of spam e-mails. We briefly present an overview of the most significant work done in this field.

With the benefits of spam filters, the security risk can be reduced since the user gets in hand the emails that have gone through various spam checks. Moreover, these email spam filters throw out malware, malicious, and virus-infested emails and protect user security.

 It involves categorize incoming emails into spam and non-spam. Machine learning algorithms can be trained to filter out spam mails based on their content and metadata.

**2. PROBLEM DEFINITION & DESIGN THINKING**

 **2.1 EMPATHY MAP**



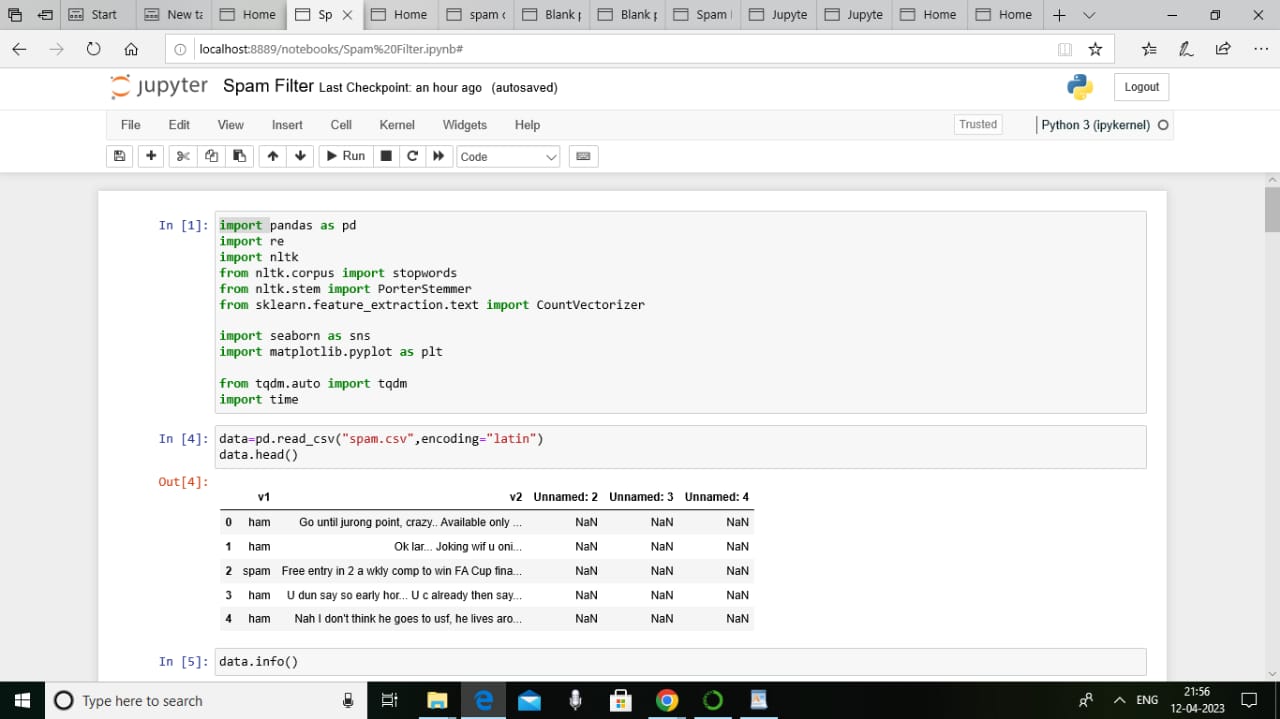
**2.2 IDEATION & BRAINSTORMING MAP**

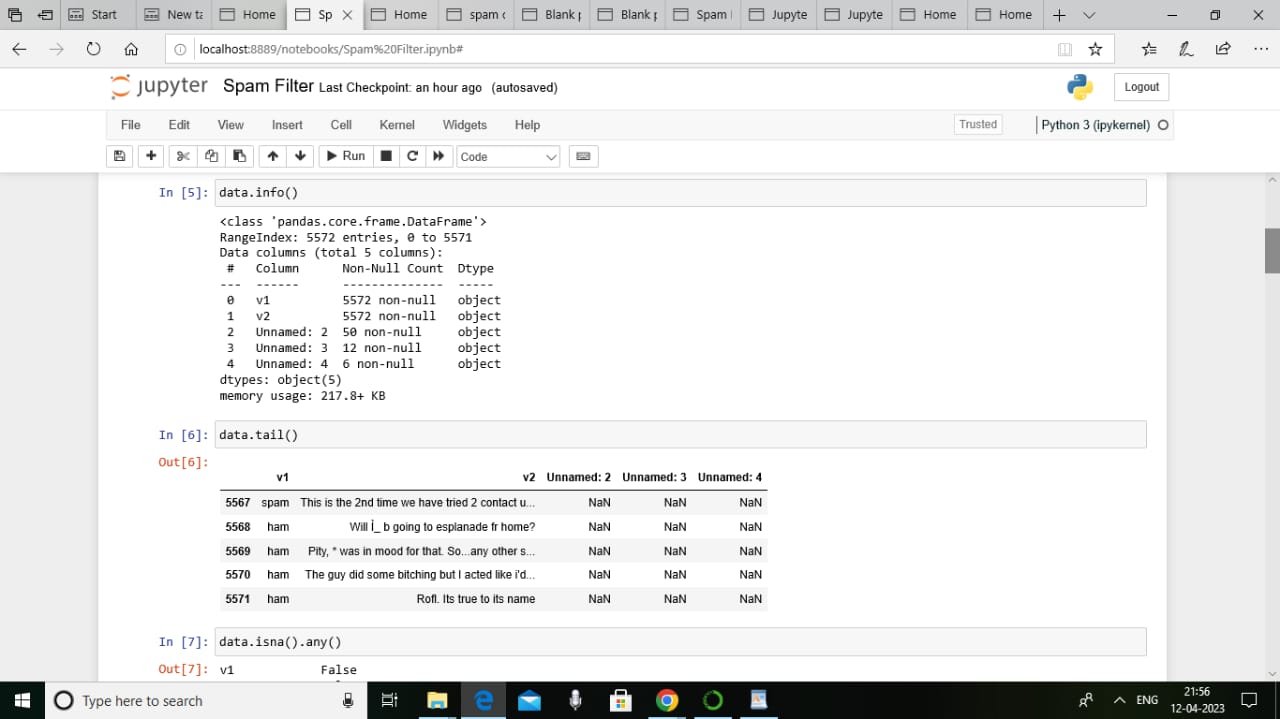
**BRAINSTORMING MAP**

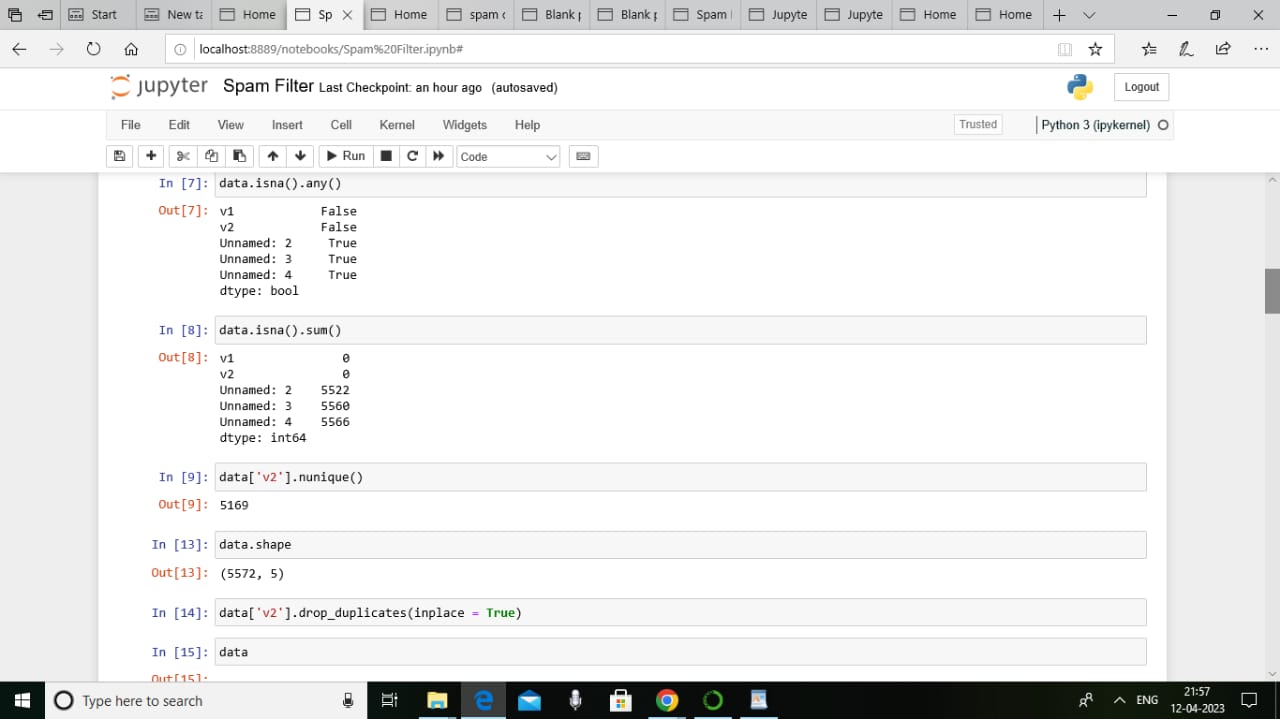
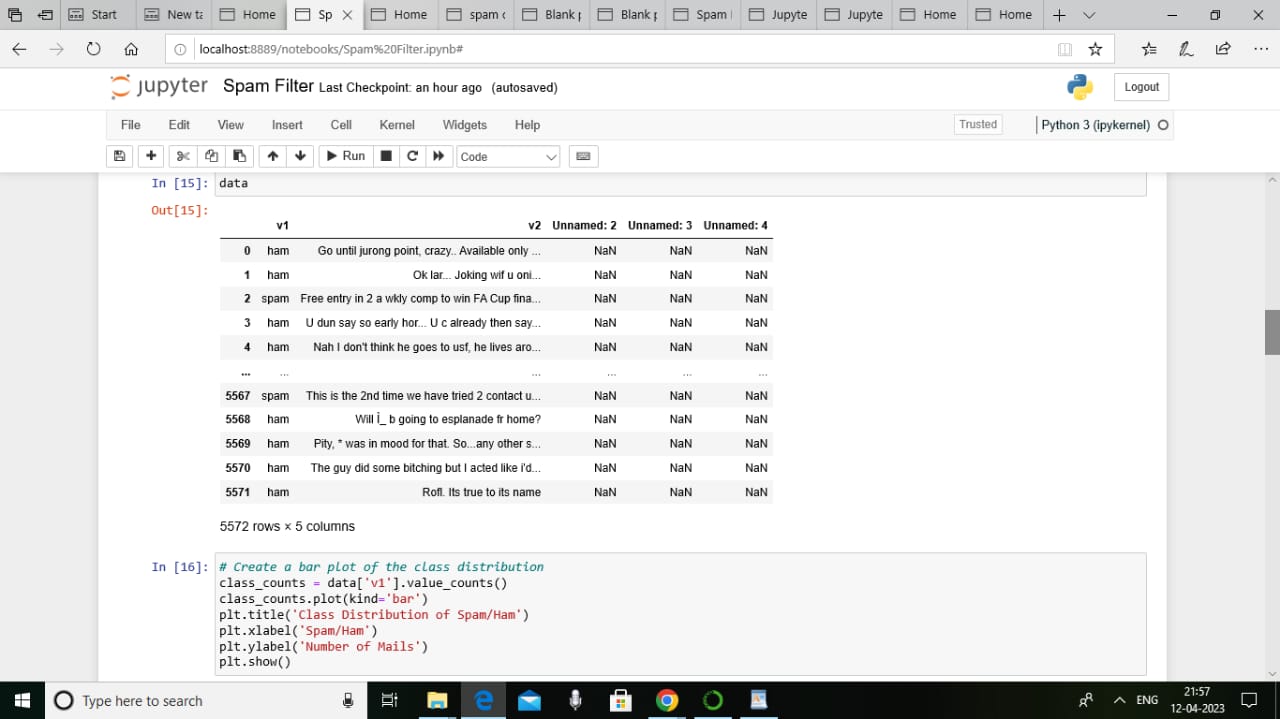


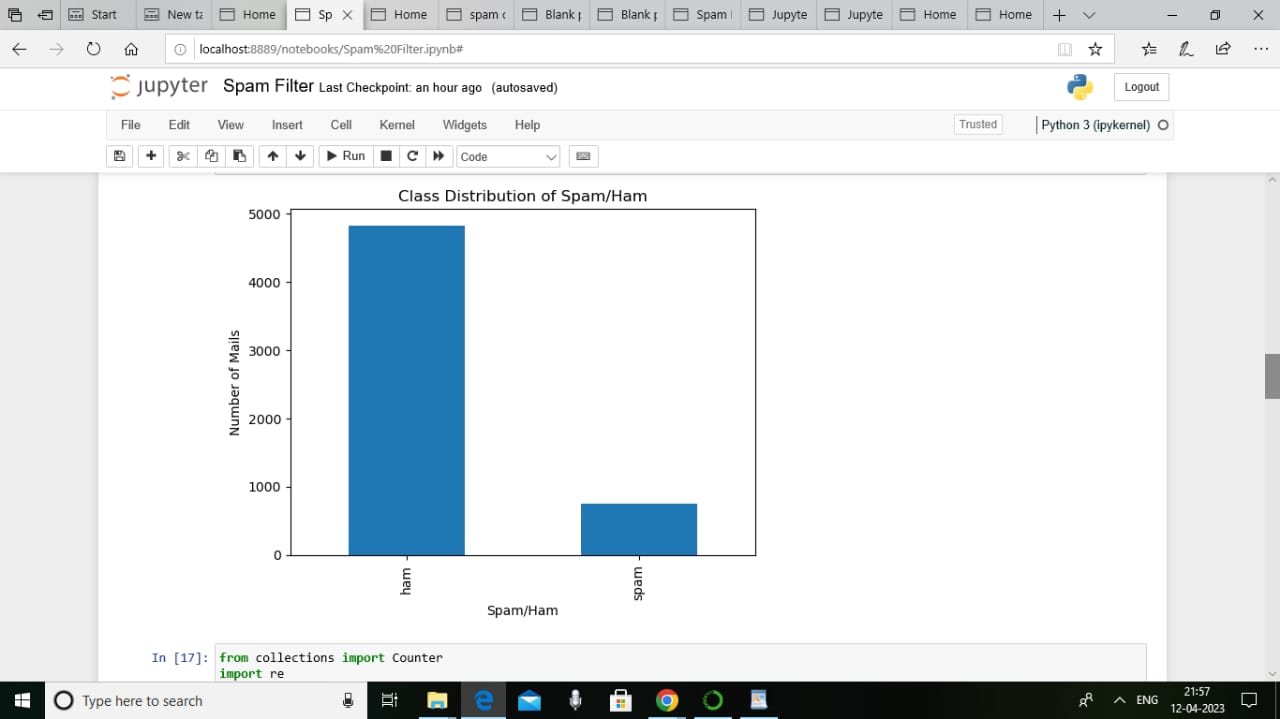
**3. RESULT**

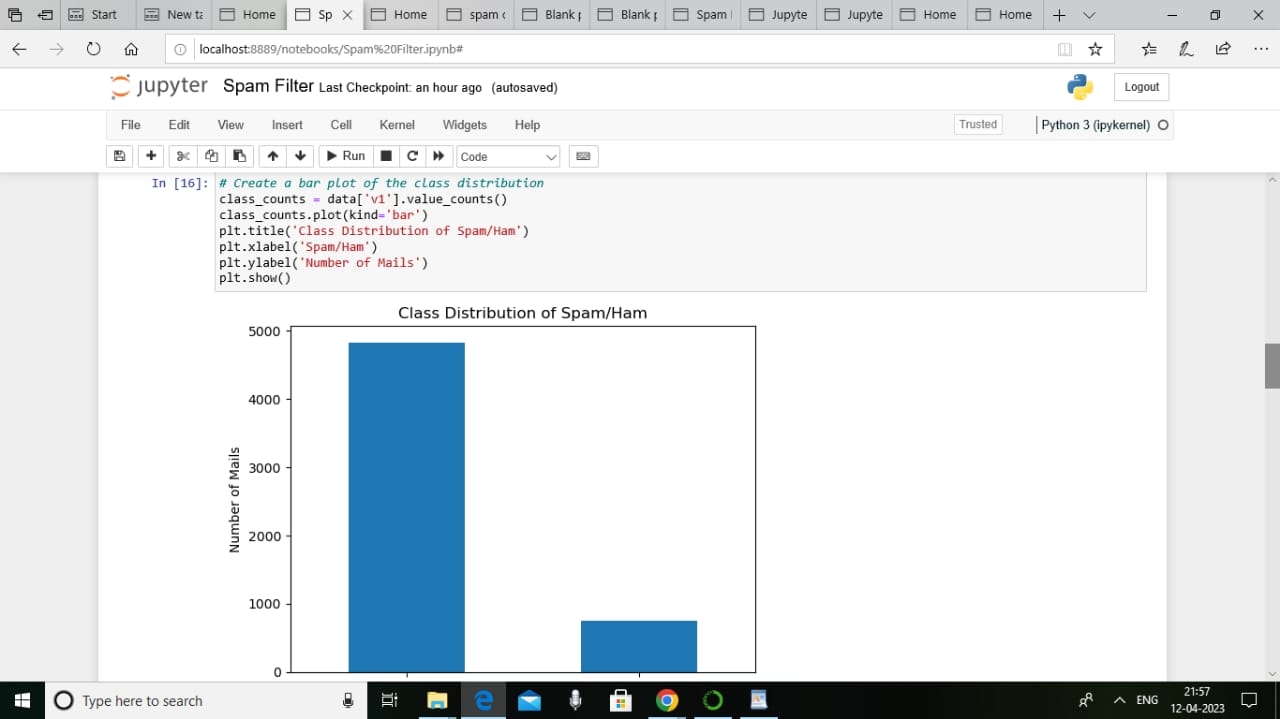
**(FINAL FINDINGS (OUTPUT) OF THE PROJECT ALONG WITH SCREENSHOT).**

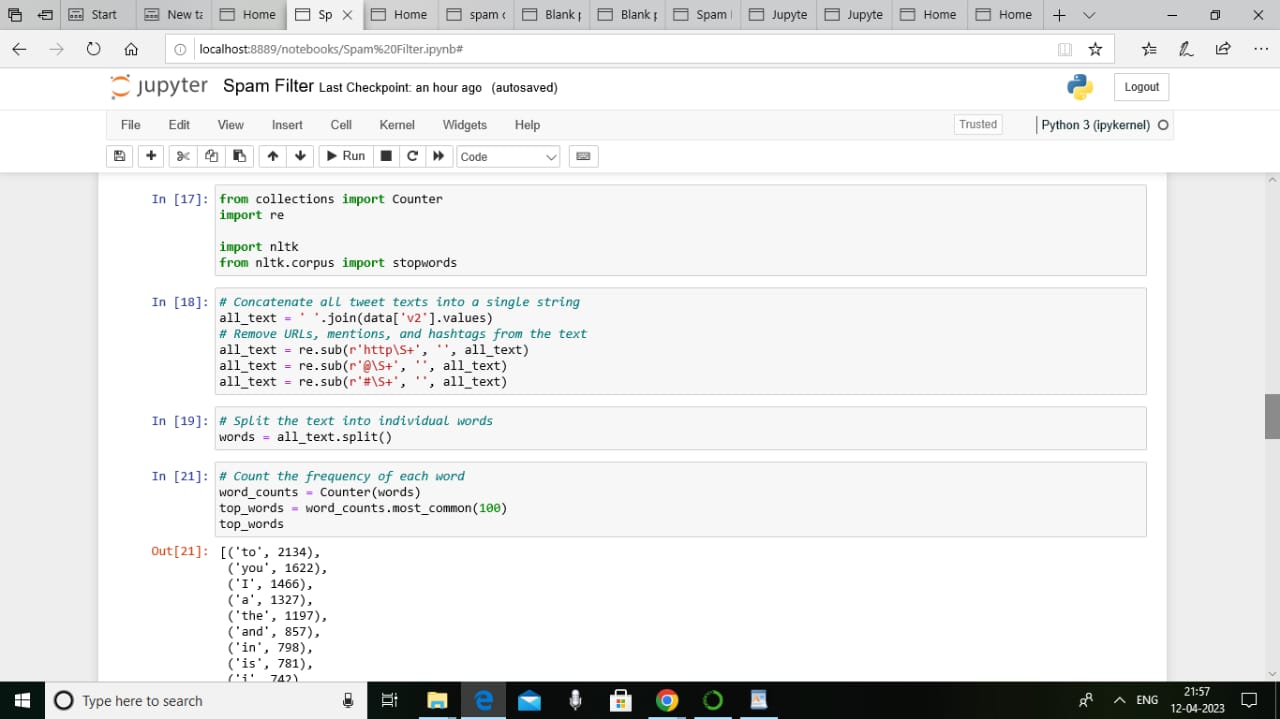
****

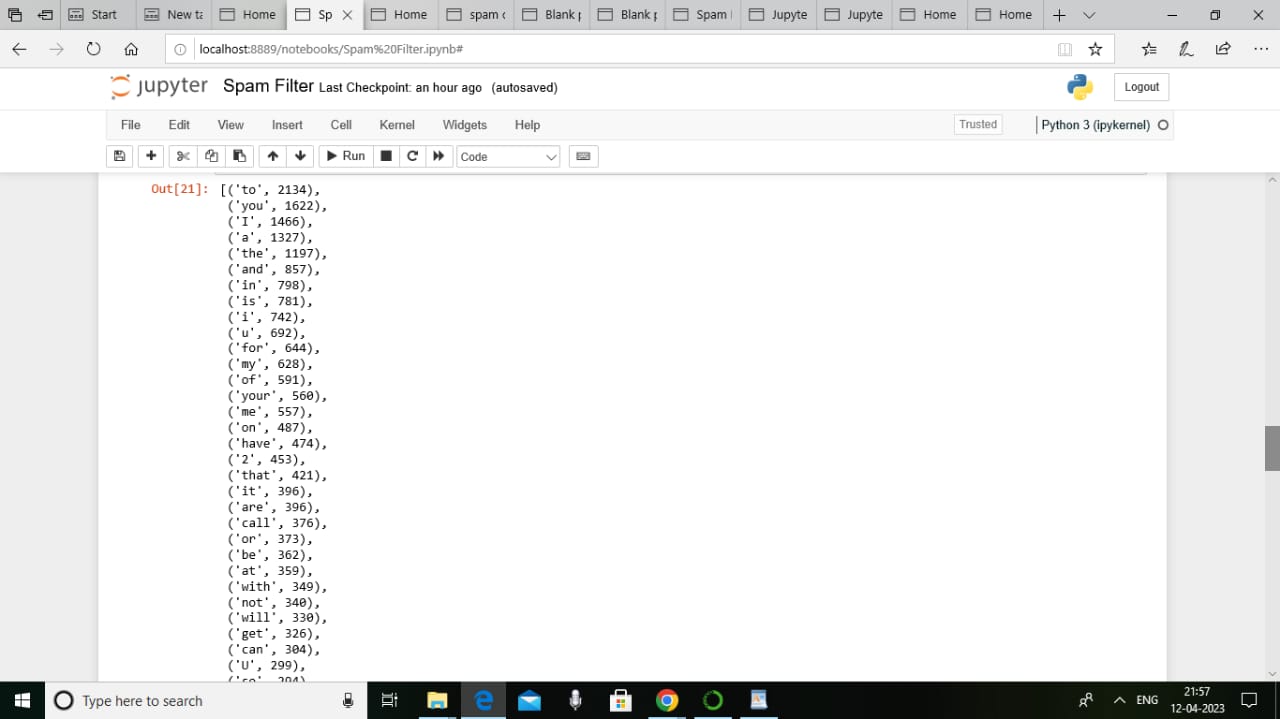
****

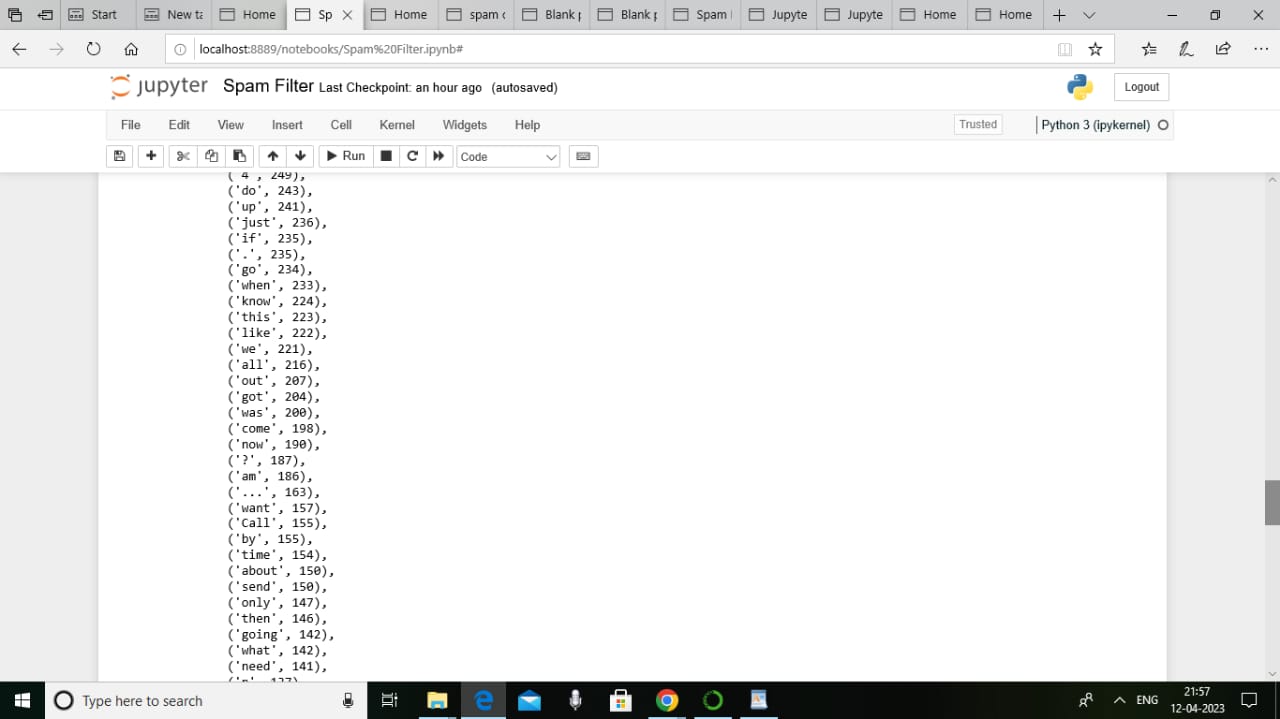
****

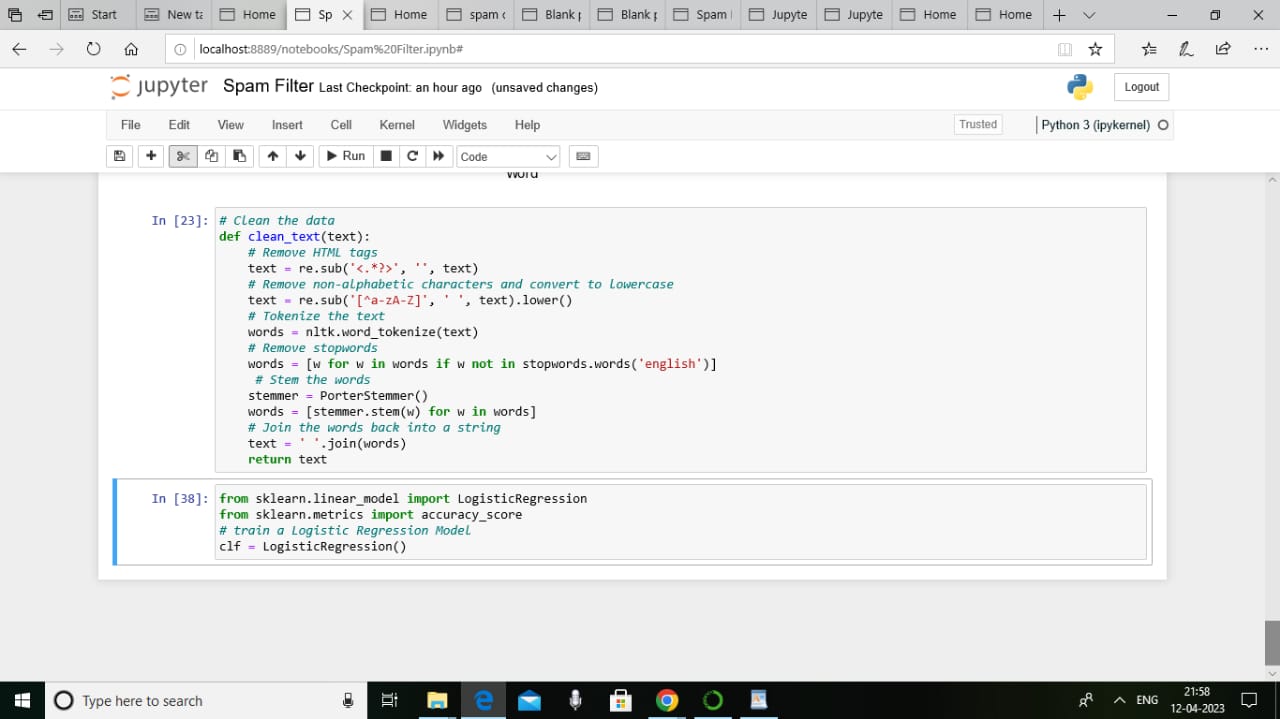
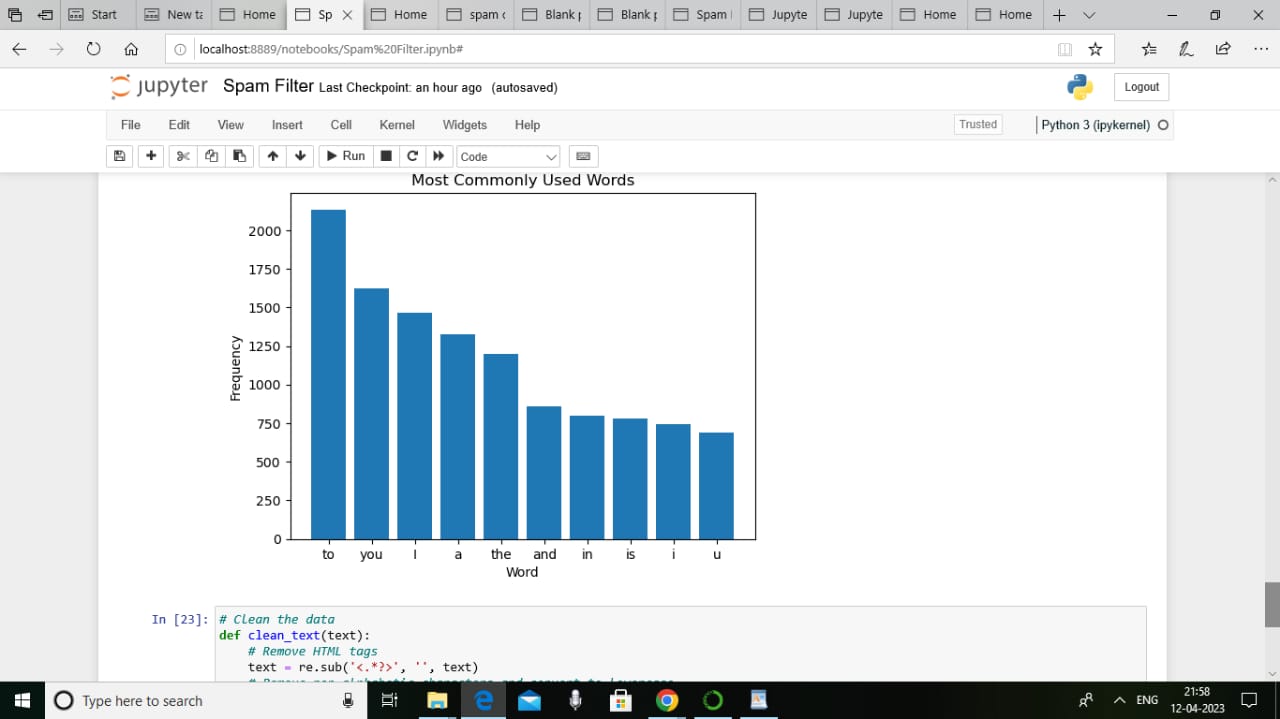
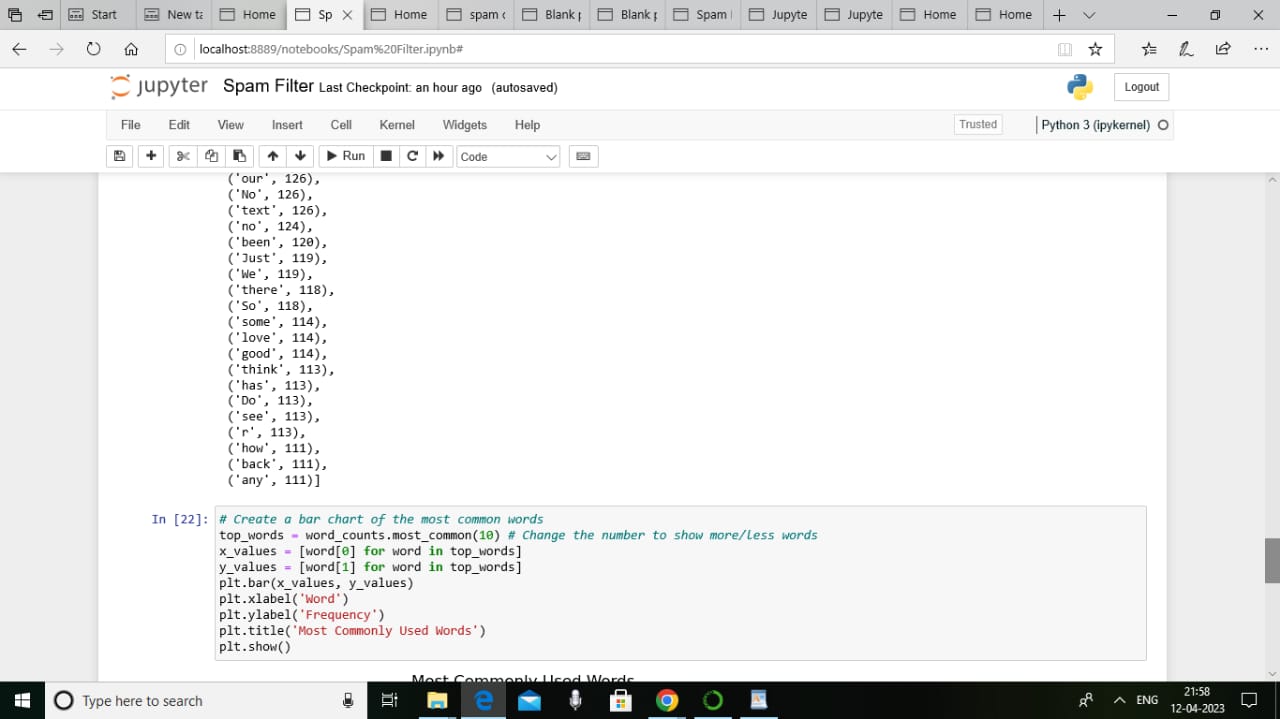
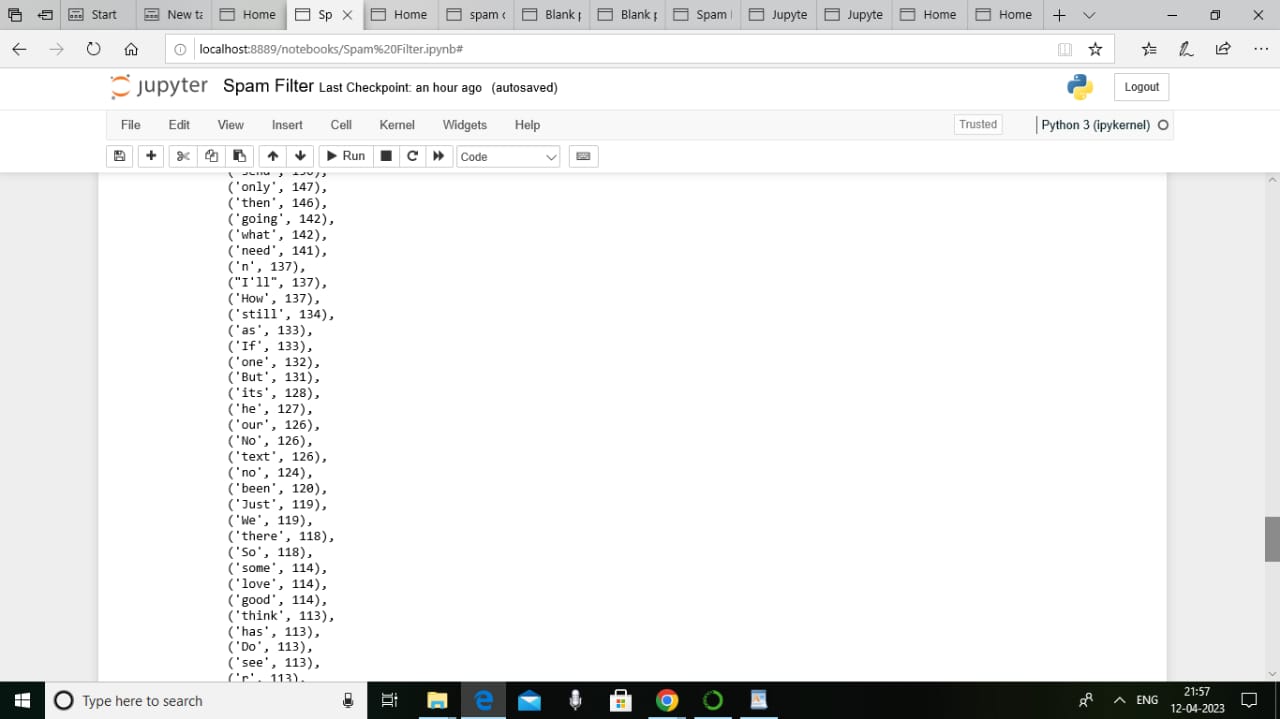
****

****

****

****

****

****

**4. ADVANTAGES & DISADVANTAGES**

**(List of advantages and disadvantages of the proposed solution).**

**ADVANTAGES**

* Spamming refers to the practice of sending unsolicited messages especially advertising messages to a group of recipients. Unsolicited messages mean the recipient did not grant permission for those message to be sent.
* **Blocking Spam**  
  Some [**anti-spam**](https://www.comodo.com/business-security/email-security/antispam-gateway.php) solutions not only block specific email addresses but also search for subject lines and text in the email messages. You can customize it to block incoming emails based on senders, and even if your email address is not in the recipient field.
* **Quarantining Spam**  
  Anti-spam filters automatically quarantine the spam emails, ensuring your inbox is spam free. Quarantined emails are kept for a fixed number of days and then discarded. During that period, you can check and recover any legitimate email that may have been quarantined.
* **Automatic Filter Updates**  
  Most of the [antivirus software](https://antivirus.comodo.com/) comes with automatic filter update feature for timely detection of new [types of Malware](https://antivirus.comodo.com/blog/computer-safety/malware-vs-viruses-whats-difference/) threats. Automatic updates not only helps the anti-spam software to stay up-to-date, but it also helps secure your system from new kinds of Malware.
* **Monitoring Multiple Accounts**  
  With this feature, you can monitor and filter out spam from multiple accounts. You can filter your home email from work email, and vice versa.

**DISADVANTAGES**

* Thousands of spam emails may reach Inboxes before a spammer's email address, IP or domain is blacklisted. Spam filtering is machine-based so there is a room for mistakes called “false positives.” Bayesian filters may be fooled by spammers, e.g. in a case of using large blocks of legitimate text.

**5. APPLICATIONS**

(**THE** **AREAS WHERE THIS SOLUTION CAN BE APPLIED)**

* Email spam filter software, also known as email anti-spam software, is used to prevent malicious content from being delivered by email and to prevent sensitive information from being leaked through email.
* The software uses AI and machine learning to scan emails and attachments for potential threats and then blocks or quarantines the delivery of suspicious messages.
* An email’s header, sender and recipient IP addresses, message content, attachments, and links are evaluated for questionable characteristics.
* Email viruses, malware, malicious links, phishing attacks, spoofing/impersonation, random ware, and spam are identified.
* Organizations and businesses use email filtering software to stop phishing attacks and the inadvertent downloading of malware.
* The intent of both phishing attacks and malware is to take advantage of an email recipient’s lack of awareness to gain access to proprietary information and resources.
* This software helps to prevent compromising a company’s confidential data and its IT infrastructure.
* Email administrators can whitelist known senders and approved content providers, and blacklist bad actors.
* Email filtering supports the routing of email into a variety of designated categories and folders for wanted and unwanted emails such as bulk mail and junk mail.
* Increasingly, artificial neural networks are being used to identify personal preferences when organizing email inboxes.
* Email filtering software tools provide encryption and scanning for outgoing emails.
* Data Loss Prevention (DLP) features prevent the inadvertent or intentional leaking of sensitive data from a business.
* Account takeover protection prevents hacked accounts from sending out restricted information. This preserves an organization’s reputation and integrity by halting the distribution of questionable content and confidential information.

**6. CONCLUSION**

**(CONCLUSION SUMMARIZING THE ENTIRE WORK & FINDINGS).**

In this paper we proposed a new approach for a spam detection filter. Messages are classified with the KNN algorithm based on a set of features extracted from the email’s properties and content. The train set is resampled to the most appropriate size and positive class distribution determined by several experiments. The system performs a constant update of the data set and the list of most frequently words that appear in the messages. Also a feedback option regarding misclassified messages is implemented and recommended to be performed at a certain rate depending on the resources available. We adapted the filter to each user’s particularities. Further development would be to add the email’s sending date as an attribute and change the KNN algorithm to set greater weights for the emails sent in the same time frame. Also performing a feature selection on the extracted attributes would be another direction for further possible. The existing spam detection techniques, their current applications and their limitations have been highlighted. It has been seen that even the existing techniques consist of certain loop holes and none of the methods is completely effective in itself. Review spam detection is indeed a hard task but it requires continuous research and development in this field.

**7. FURTHER WORK**

**( ENCHANCEMENTS THAT CAN BE MADE IN THE FUTURE).**

Review spam detection is essential since it can ensure justice for the sellers and retain the trust of the buyer on the online stores. The algorithms developed so far have not been able to remove the requirement of manual checking of the reviews. Hence there is scope for complete automation of spam detection systems with maximum efficiency. With growing popularity of online stores, the competition also increases. The spammers get smarter day by day and spam reviews become untraceable. It is necessary to identify the spamming techniques in order to produce counter algorithms.

**8. APPENDIX**

**A. SOURCE CODE**

**(Attach the code for the solution built).**

**import pandas as pd**

**import re**

**import nltk**

**from nltk.corpus import stopwords**

**from nltk.stem import PorterStemmer**

**from sklearn.feature\_extraction.text**

**import CountVectorizer**

**import seaborn as sns**

**import matplotlib.pyplot as plt**

**from tqdm.auto import tqdm**

**import time**

**data=pd.read\_csv("spam.csv",encoding = "latin")**

**data.head()**

**data.info()**

**data.tail()**

**data.isna().any()**

**data.isna().sum()**

**data['v2'].nunique()**

**data.shape**

**data['v2'].drop\_duplicates(inplace = True)**

**data**

**# Create a bar plot of the class distribution**

**class\_counts = data['v1'].value\_counts()**

**class\_counts.plot(kind='bar')**

**plt.title('Class Distribution of Spam/Ham')**

**plt.xlabel('Spam/Ham')**

**plt.ylabel('Number of Mails')**

**plt.show()**

**from collections import Counter**

**import re**

**import nltk**

**from nltk.corpus import stopwords**

**# Concatenate all tweet texts into a single string**

**all\_text = ' '.join(data['v2'].values)**

**# Remove URLs, mentions, and hashtags from the text**

**all\_text = re.sub(r'http\S+', '', all\_text)**

**all\_text = re.sub(r'@\S+', '', all\_text)**

**all\_text = re.sub(r'#\S+', '', all\_text)**

**# Split the text into individual words**

**words = all\_text.split()**

**# Count the frequency of each word**

**word\_counts = Counter(words)**

**top\_words = word\_counts.most\_common(100)**

**top\_words**

**# Create a bar chart of the most common words**

**top\_words = word\_counts.most\_common(10) # Change the number to show more/less words**

**x\_values = [word[0] for word in top\_words]**

**y\_values = [word[1] for word in top\_words]**

**plt.bar(x\_values, y\_values)**

**plt.xlabel('Word')**

**plt.ylabel('Frequency')**

**plt.title('Most Commonly Used Words')**

**plt.show()**

**# Clean the data**

**def clean\_text(text):**

**# Remove HTML tags**

**text = re.sub('<.\*?>', '', text)**

**# Remove non-alphabetic characters and convert to lowercase**

**text = re.sub('[^a-zA-Z]', ' ', text).lower()**

**# Tokenize the text**

**words = nltk.word\_tokenize(text)**

**# Remove stopwords**

**words = [w for w in words if w not in stopwords.words('english')]**

**# Stem the words**

**stemmer = PorterStemmer()**

**words = [stemmer.stem(w) for w in words]**

**# Join the words back into a string**

**text = ' '.join(words)**

**return text**

**from sklearn.linear\_model import LogisticRegression**

**from sklearn.metrics import accuracy\_score**

**# train a Logistic Regression Model**

**clf = LogisticRegression()**