## IPTIMIZING SPAM FILTERING WITH MACHINE LEARNING

1. INTODUCTION:

* It helps them to prepare for their future in the right and easy way.
* Technology plays a key role in helping educators to reach new levels with access to the up to date content, apps, and other helpful resources.
* Modern devices for students in education help them in preparing for the future on the right path.
* Educational technology also refers to the implementation of the latest technologies including software and hardware that enable different methods of modern teaching.
* In this way, both students and teachers have the access to the hundreds of online resources that helps provide mutual collaboration.

## OVERVIEW:

* Optimizing Spam Filtering with Machine Learning Over recent years, as the popularity of mobile phone devices has increased, Short Message Service (SMS) has grown into a multi-billion dollar industry. Due to Spam SMS, Mobile service providers suffer from some sort of financial problems as well as it reduces calling time for users.
* To avoid such Spam SMS people use white and black list of numbers. But this technique is not adequate to completely avoid Spam SMS. To tackle this problem it is needful to use a smarter technique which correctly identifies Spam SMS. Natural language processing technique is useful for Spam SMS identification.

Once model analyses the input the prediction is showcased on the UI To accomplish this, we have to complete all the activities listed below,

* Define Problem / Problem Understanding ○ Specify the business problem ○ Business requirements ○ Literature Survey ○ Social or Business Impact.
* Data Collection & Preparation ○ Collect the dataset ○ Data Preparation
* Exploratory Data Analysis ○ Descriptive statistical ○ Visual Analysis
* Model Building ○ Training the model in multiple algorithms ○ Testing the model
* Performance **Testing & Hyperparameter** Tuning ○ Testing model with multiple evaluation metrics ○ Comparing model accuracy before & after applying hyperparameter tuning
* Model Deployment ○ Save the best model ○ Integrate with Web Framework
* Project Demonstration & Documentation ○ Record explanation Video for project end to end solution ○ Project Documentation-Step by step project development procedure Project Structure: Create the Project folder which contains files as shown below
* We are building a flask application which needs HTML pages stored in the templates folder and a python script app.py for scripting. Milestone 1: Define Problem / Problem Understanding Activity 1: Specify the business problem Refer Project Description
* Activity 2: Business requirements A business requirement for an SMS spam classification system would include the ability to accurately identify and flag spam messages, protect customers from unwanted or harmful messages, and comply with industry regulations and laws regarding spam messaging. Additionally, the system should be able to handle a high volume of messages, integrate with existing systems and databases, and provide reporting and analysis capabilities to track performance and improve the system over time.
* Activity 3: Literature Survey (Student Will Write) project would involve researching and analysing existing studies, papers, and articles on the topic to gain a thorough understanding of the current state of SMS spam classification and to identify potential areas for improvement and future research.
* The survey would include looking at different methods and techniques used for identifying and flagging spam messages, such as machine learning algorithms, natural language processing, and rule-based systems.
* Additionally, the literature survey would review the current state of SMS spam and trends in the industry, as well as any existing laws and regulations related to spam messaging. The survey would also investigate the datasets and feature representations used in previous studies, which would help to determine the best approach for the current project.
* Furthermore, It would be important to check the pre-processing techniques used in the research to understand how to properly clean and prepare the data for the

classifier .

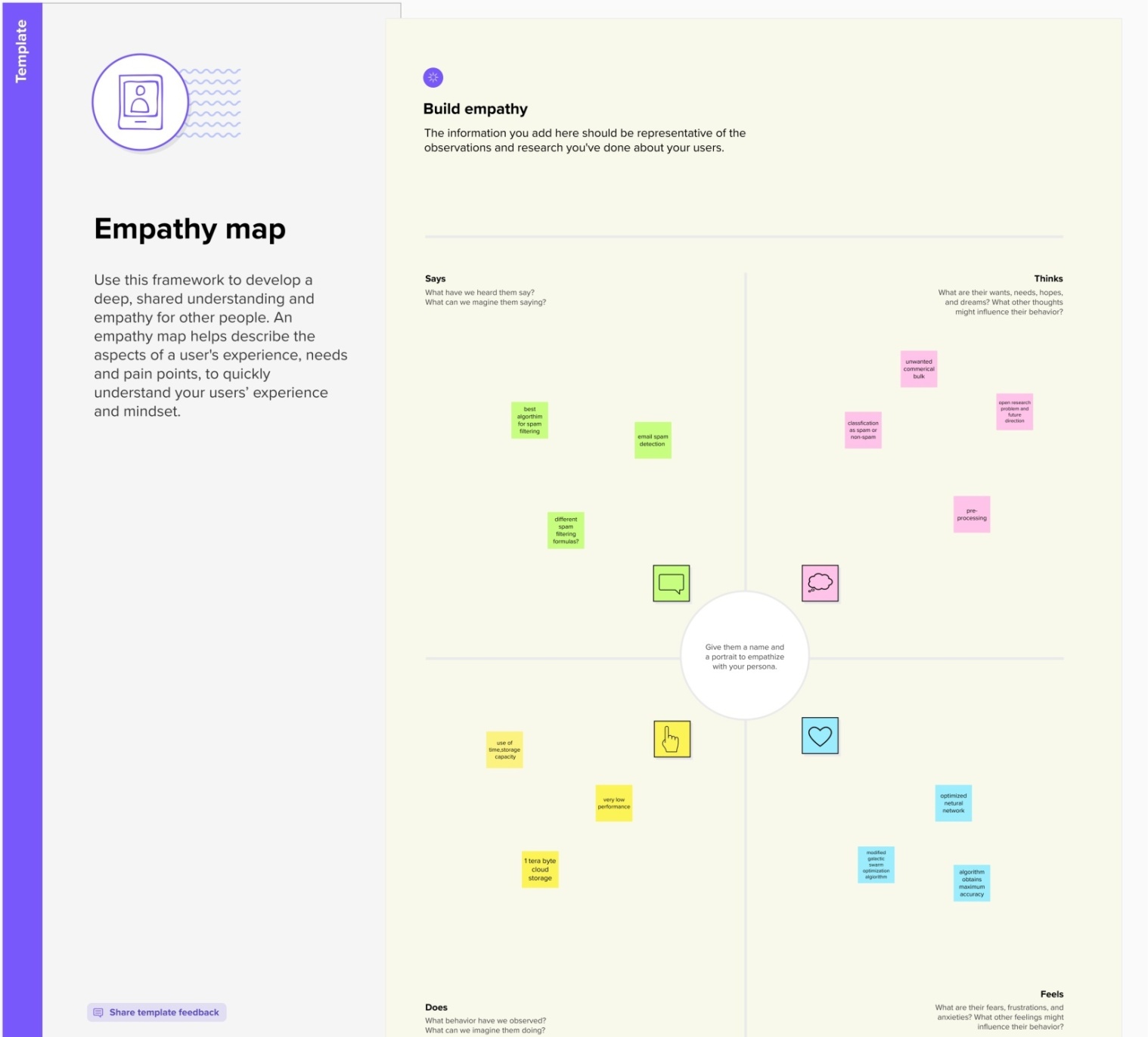
* Activity 4: Social or Business Impact. Social Impact:- it can help protect individuals from unwanted and potentially harmful messages
* . By accurately identifying and flagging spam messages, the system can help prevent these types of attacks and protect individuals from falling victim to them.
* Business Model/Impact:- it can help protect their customers and improve their reputation. Spam messages can harm a business`s reputation and lead to customer complaints and lost business. By accurately identifying and flagging spam messages, the system can help protect businesses and improve their customer`s trust.
* Activity 1: Collect the dataset There are many popular open sources for collecting the data. Link: https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset As the dataset is downloaded.
* Let us read and understand the data properly with the help of some visualisation techniques and some analysing techniques.
* Note: There are a number of techniques for understanding the data.

### PURPOSE:

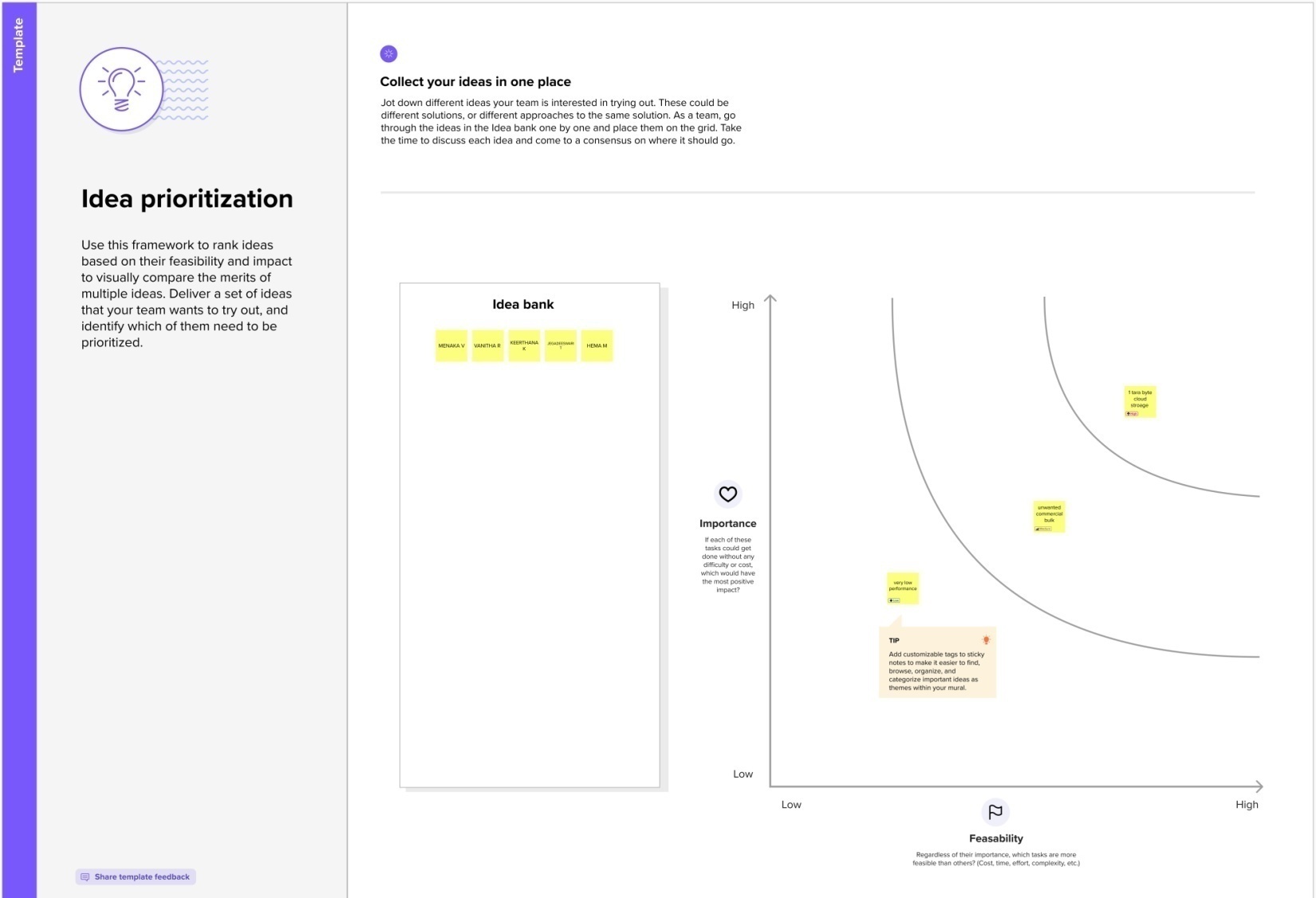
* Content filters analyze the text inside an email and use that information to decide whether or not to mark it as spam.
* The content of spam emails is often predictable, particularly because they tend to have the same basic objectives: offer deals, promote explicit material, or otherwise tap into human emotions, feelings, and desires, such as greed or fear.
* Content filters may search for words connected to money, such as “discount,” “limited time,” or “offer.” To trigger the filter, there typically would have to be multiple uses of the target word.
* Content filters may also examine an email for inappropriate language of a sexual nature that could indicate explicit content. In some campaigns, an attacker may use sexually explicit emails to lure users into opening the email and then clicking on malicious links.

2 . PROBLEM DEFINITION & DESIGN THINKING :

### EMPATHY MAP:

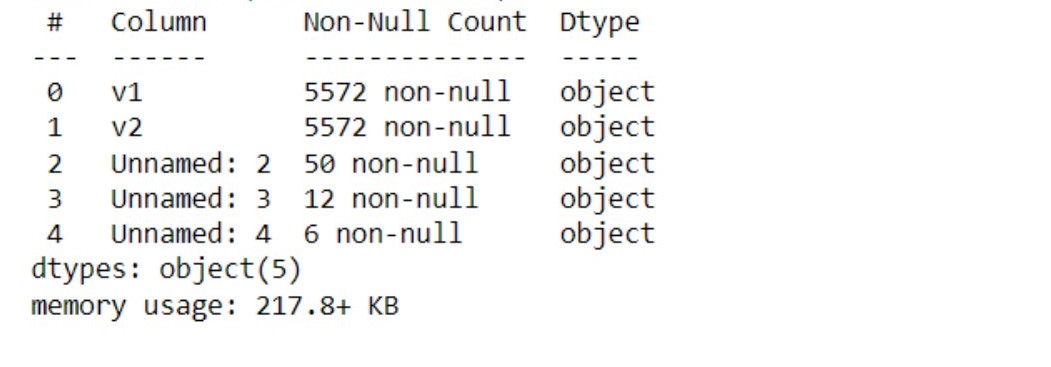


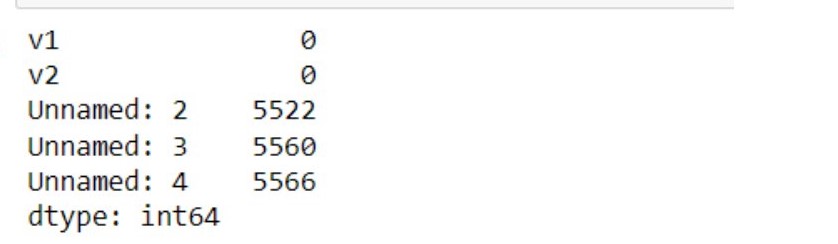
### Ideation & Brainstorming Map

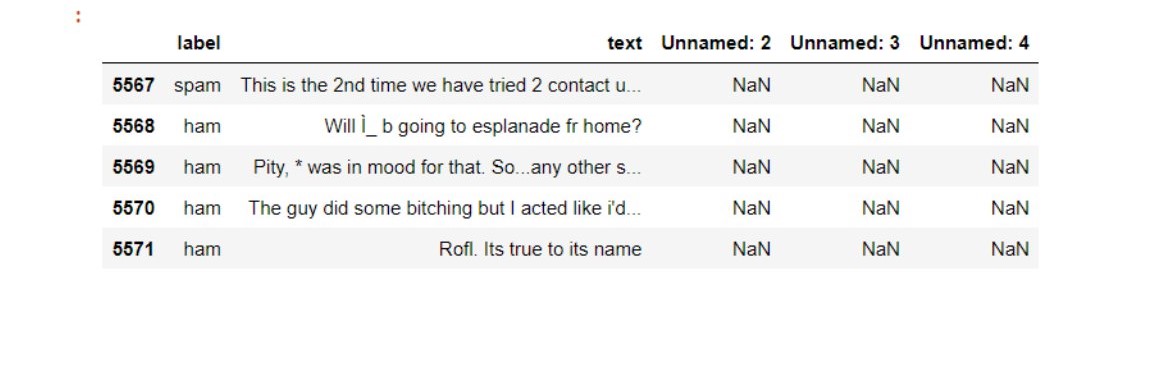


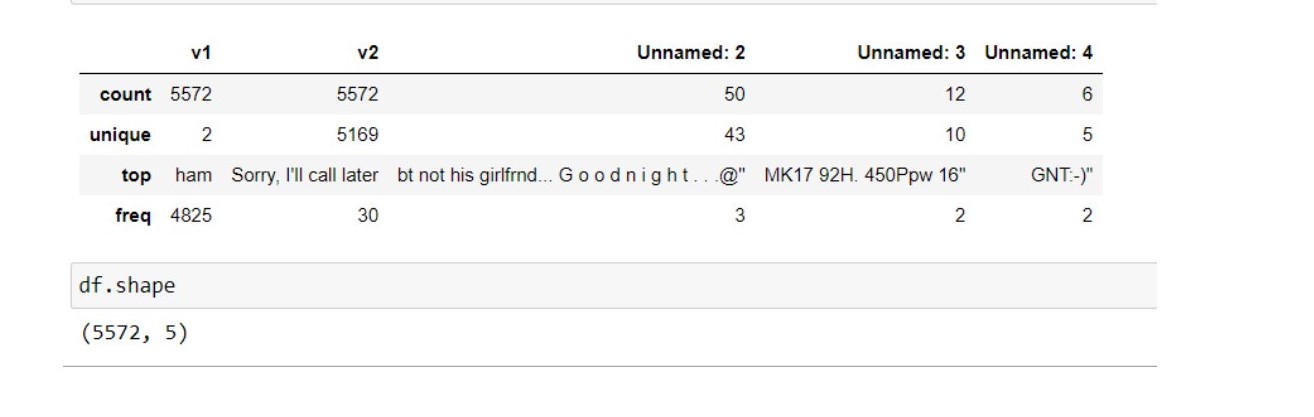
3. RESULT:

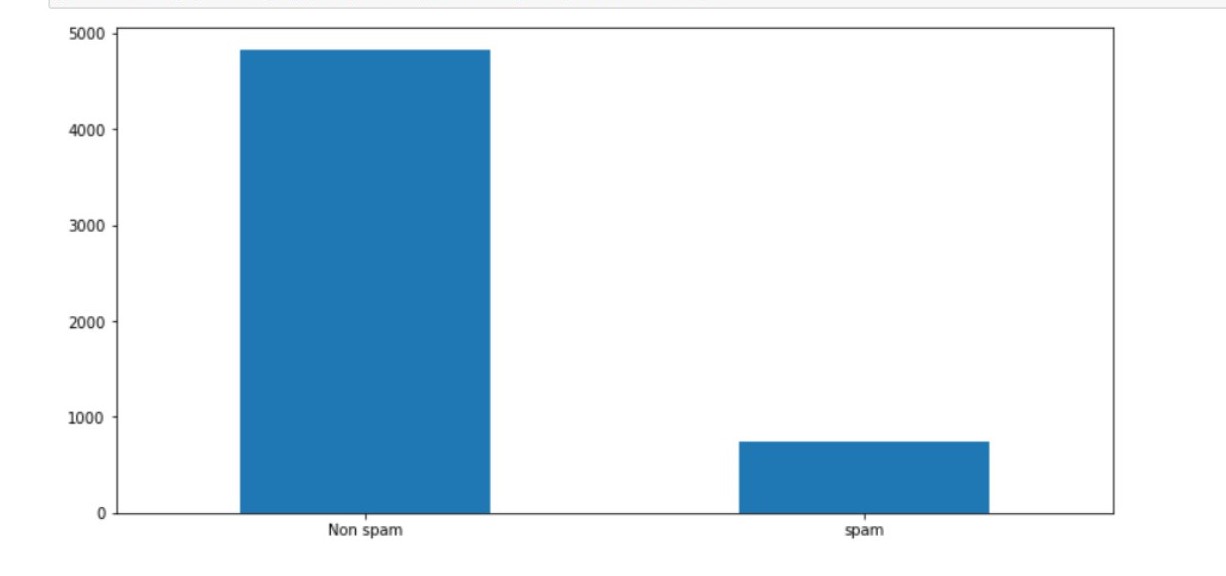


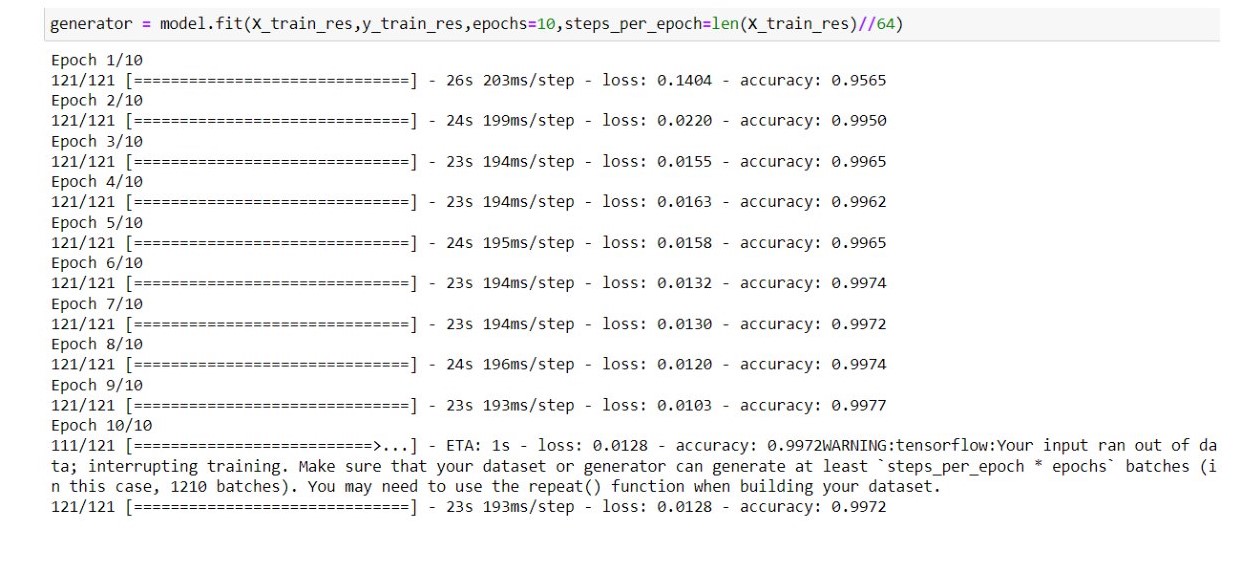




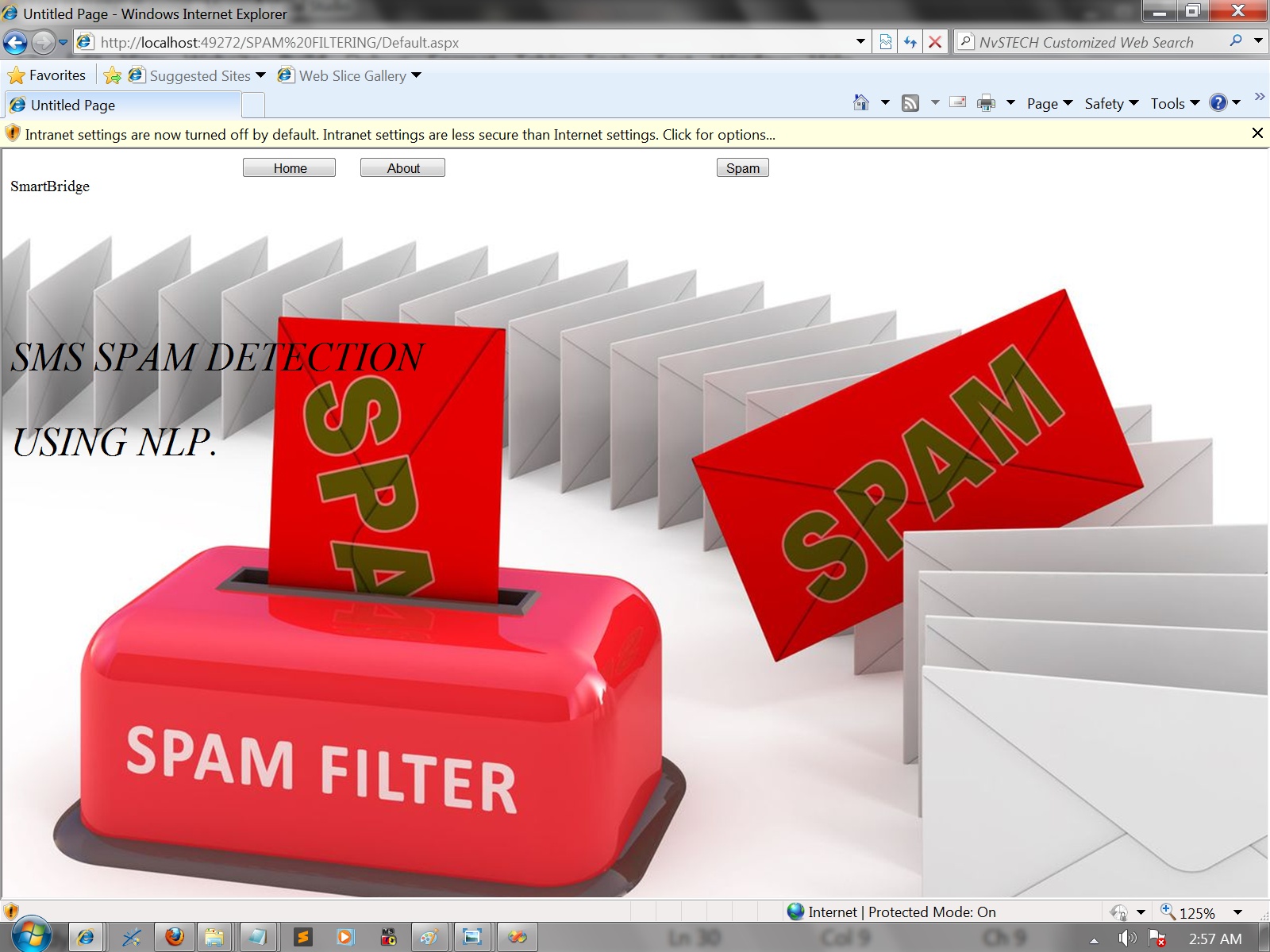








WEB BROWSER RESULT:-



4. ADVANTAGE & DISADVANTAGE:

### 4.1 Advantage:

* Improved Sales
* Time-Saving
* Cost-Effectiveness
* Enhanced Visibility
* Customer-Centric Marketing
* Identify Engaged Customers
* Complete Control Over Campaigns

### 4.2 Disadvantage:

* Risk of Spamming
* Likelihood of Unresponsiveness
* Ineffective Resource Utilization
* Damaged Company Reputation
* Undelivered Emails
* Website Blocking
* Data Protection
* Managing Mailing List

5. APPLICAION:

#### [email spam](https://www.techtarget.com/searchsecurity/definition/spam)

Email spam, also known as junk email, refers to unsolicited email messages, usually sent in bulk to a large list of recipients.

#### [spam filter](https://www.techtarget.com/searchsecurity/definition/spam-filter)

A spam filter is a program used to detect unsolicited, unwanted and virus-infected emails and prevent those messages from getting to a user's inbox.

#### [link spam](https://www.techtarget.com/whatis/definition/link-spam)

Link spam is the posting of out-of-context links on websites, discussion forums, blog comments, guestbooks or any other online venue that displays user comments.

#### [Application blacklisting](https://www.techtarget.com/searchsecurity/definition/application-blacklisting)

Application blacklisting, sometimes just referred to as blacklisting, is a network administration prevent the execution of undesirable programs.

#### [SMS spam (cell phone spam or short messaging service spam)](https://www.techtarget.com/searchmobilecomputing/definition/SMS-spam)

SMS spam (sometimes called cell phone spam) is any junk message delivered to a mobile phone as text messaging through the Short Message Service (SMS).

#### [storage filer](https://www.techtarget.com/searchstorage/definition/storage-filer)

A storage filer is a type of file server designed and programmed for high-volume data storage, backup and archiving.

6. CONCLUSION:

Microsoft Defender for Endpoint -- formerly Microsoft Defender Advanced Threat Protection or Windows Defender ATP -- is an endpoint security platform designed to help enterprise-class organizations prevent, detect. A large language model (LLM) is a type of artificial intelligence (AI) algorithm that uses deep learning techniques and massively large data sets to understand, summarize, generate and predict new content. Snapchat is a mobile app that allows users to send and receive "self-destructing" photos and videos. Photos and videos taken with the app are called snaps.

#### 

#### [packet filtering](https://www.techtarget.com/searchnetworking/definition/packet-filtering)

#### [black box AI](https://www.techtarget.com/whatis/definition/black-box-AI)

#### [ACK (acknowledgement)](https://www.techtarget.com/searchnetworking/definition/ACK)

#### [knowledge-based systems (KBSes)](https://www.techtarget.com/searchcio/definition/knowledge-based-systems-KBS)

#### [Multiprotocol Label Switching (MPLS)](https://www.techtarget.com/searchnetworking/definition/Multiprotocol-Label-Switching-MPLS)

In some digital communication protocols, ACK -- short for 'acknowledgement' -- refers to a signal that a device sends to indicate that data has been received successfully.Artificial intelligence is the simulation of human intelligence processes by machines, especially computer systemsKnowledge-based systems (KBSes) are computer programs that use a centralized repository of data known as a knowledge base to provide a method for problem-solving.

7. FUTURE SCOPE:

* **Features of a spam filter service are presented  below in no specific order of priority**.

## Greylisting

spam filter service will block the delivery of emails from known sources of spam. Greylisting is used to prevent the delivery of emails from previously unknown sources of spam. This is achieved by returning non-whitelisted emails to their originating servers with a request for the email to be resent. Spammers´ servers often have the mail retry function disabled, and the spam email rarely comes back.

## Accelerated Filtering Rates

With fewer spam emails going beyond the Greylisting process, the rate at which remaining emails are filtered is significantly accelerated. This mitigates any delays attributable to the Greylisting process and reduces the administrative burden of whitelisting multiple “trusted” senders in order to avoid delays in the delivery of their emails.

## Outbound Email Scanning

Outbound email scanning is a valuable security tool for identifying business email accounts that may have been compromised and are being used as “trusted” senders to distribute spam or malware. Not only might this save a business from an internal threat, but also protect their reputation with business partners and other important entities.

## Advanced Anti-Virus Software

Most anti-virus software works by scanning devices and networks for the presence of malware after it has been deployed. The best spam filtering service will scan inbound and outbound emails and their attachments for malware as they flow through the mail server in order to identify threats before they have the opportunity to be deployed.

## Phishing Protection/Malicious URL Blocking

This feature compares links in the body of emails against a database of malicious URLs known to have been included in previous phishing emails. Malicious URL blocking can prevent the most carefully-crafted spear phishing attack from reaching its destination, and adds the sender´s IP address to the database in order to block future attempts using a different URL.

## Data Leak Protection

Data leak protection is a relatively new development in email filtering. It allows the tagging of user-defined keywords so that if an outbound email containing a tagged keyword passes through the filtering service, the email is flagged or blocked. This feature is designed to stop the leak of confidential data, either to a malicious actor or from a malicious insider.

## Ease of Operation

The best email filtering service in the world may not be of value to businesses if it is too complicated to configure. Therefore, whatever solution is chosen to protect businesses from excessive levels of spam emails and the threats that accompany them must be user-friendly for customizing the service to the business´s specific security requirements.

## Adding an Email Filtering Service to Office 365

Microsoft Office 365 includes some email security features as standard in its Exchange Online Protection (EOP) feature; however, many businesses find the basic level of protection provided by Microsoft’s email filtering service falls short of requirements. The email filter in Office 365 does a reasonable job of blocking spam from know malicious sources, but it is not so effective at blocking phishing emails and zero-day malware threats, which often sneak past EOP.

To improve protection you need to use a third-party email filtering service to add another layer of security to Office 365. In contrast to Office 365’s standard email filtering service, SpamTitan uses machine learning techniques to identify new phishing threats that have not been seen before. Greylisting is used to identify new sources of spam and phishing emails, and sandboxing is used to isolate zero-day malware threats in safe environments. These features are not provided in Microsoft’s EOP email filtering service.

8.APPENDIX:

* Lack of effective strategy to handle the threats to the security of the spam filters. Such an attack can be causative or exploratory, targeted or indiscriminate attack.
* The inability of the current spam filtering techniques to effectively deal with the concept drift phenomenon.
* Majority of the existing email spam filters does not possess the capacity to incrementally learn in real-time. Conventional spam email classification techniques are no longer viable to cope in real time environment that is characterised by evolving data streams and concept drift.
* Failure of many spam filters to reduce their false positive rate.
* Development of more efficient image spam filters. Most spam filters can only classify spam messages that are text. However, many savvy spammers send spam email as text embedded in an image (stego image) thereby making the spam email to evade detection from filters.
* The need to develop adapted, scalable, and integrated filters by applying ontology and semantic web to spam email filtering.
* Lack of filters that have the capacity to dynamically update the feature space. Majority of the existing spam filters are unable to incrementally add or delete features without re-creating the model totally to keep abreast of current trends in email spam filtering.

## Source code:

**Activity 1.1: Importing the libraries**

import numpy as np # scientific computation

import pandas as pd # loading dataset file

import matplotlib.pyplot as plt = visualization

import nltk # Preprocessing our text

from nltk.corpus import stopwords # removing all the stop words

from nltk.stem.porter import PorterStemmer # stemming of words

**Activity 1.2: Read the Dataset**

#Load our dataset

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

df.head()

**Activity 2: Data Preparation**

**Activity 2.1: Handling missing values**

* #Give concise summary of a DataFrame

df.info()

* #Returns the sum fo all na values

df.isna().sum()

* df.rename({"v1":"label","v2":"text"}, inplace=True, axis=1)
* # bottom 5 rows of the dataframe

df.tail()

**Activity 2.2: Handling Categorical Values**

from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

df['label'] = le.fit\_transform(df ['label'])

**Activity 2.3:Handling Imbalance Data**

from sklearn.model\_selection import train\_test\_split

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size = 0.20, random\_state= 0)

### Given data is imbalanced one, we are balancing the data

print("Before Oversampling, counts of label '1': ()".format(sun(y train = 1))) print("Before Oversampling, counts of label 'o': () \n".format(sunty train 0)))

from imblearn.over\_sampling import SMOTE

Sm=SMOTE (random\_state= 2)

x\_train\_res, y\_train\_res sm.fit\_resample(x\_train, y\_train.ravel())

print('After OverSampling, the shape of train X: {}'.format(x\_train\_res.shape))

print("After OverSampling, the shape of train\_y: ( \n'.format(y\_train\_res.shape))

print("After Oversampling, counts of label '1': {}".format(sum(y\_train\_res == 1)))

print("After Oversampling, counts of label 'o': ()".format(sum(y train res == 0)))

**Activity 2.3: Cleaning the text data**

nltk.download(“stopwords”)

[nltk\_data] Downloading package stopwords to

[nltk\_data]C:\Users\smart \AppData\Roaming\nltk\_data...

[nltk\_data]Package stopwords is already up-to-datel

import nitk

from nltk.corpus import stopwords

from nltk.stem import Porterstemmer

isport re

corpus = []

length = len(df)

for i in range(0, length):

text = re.sub("[^a-zA-Z0-9]","",df["text"][i])

text= text.lower()

text = text.split()

pe = PorterStemmer ()

stopword stopwords.words("english")

text = [pe.stem(word) for word in text if not word in set(stopword)]

text = “ “.join(text)

corpus.append(text)

corpus

from = sklearn.feature extraction.text import Countvectorizer

cv = Countvectorizer (max \_features=35000)

X = cv.fit\_ transform(corpus).toarray()

Import pickle

Pickle.dump(cv, open(‘cv1.pkl’,’wb’))

**Milestone 3: Exploratory Data Analysis**

**Activity 1: Descriptive statistical**

df.describe()

**Activity 2: Visual analysis**

**Activity 2.1 Univaariate analysis**

df["label").value\_counts().plot(kind="bar", figsize=(12,6))

plt.xticks(np.arange(2), ('Non spam', 'spam), rotation=e);

from sklearn.model\_selection import train\_test\_split

X\_train, X\_trst, y\_train = train\_test\_split(x,y, test\_size = 0.20, random\_state = 0)

**Milestone 4: Model Building**

**Activity 1.1 Decision tree model:**

from sklearn.tree import DecisionTreeclassifier

model = DecisionTreeClassifier ()

model.fit (X\_train res, y\_train\_res)

DecisionTreeclassifier

DecisionTreeClassifier()

**Activity 1.2 Random farest model:**

from sklearn.ensemble import Random Forestclassifier

nodel1 =RandomForestclassifier()

model1.fit (X\_train\_res, y\_train res)

RandomForestClassifier

RandomForestclassifier()

**Activity 1.3**

From sklearn.naive\_bayes import MultinomialNB

Model=MultinomialNB ()

model.fit (X\_train\_res, y\_train\_res)

MultinomialNB

model = MultinomialNB ()

**Activity 1.5**

From tensorflow.keras.models sequential

From tensorflow.keras.layers import Dense

model = Sequential()

X train.shape

(4457, 7163)

model.add(Dense(units = X\_train res.shape[1],activation="relu",kernel\_ initializer="randon\_ uniform"))

model.add(Dense(units=100, activation="relu",kernel\_initializer="random \_uniform"))

model.add(Dense(units=100, activation="relu",kernel\_initializer="random \_uniform"))

model.add (Dense (units=I, activation="sigmoid"))

mOdel.compile(optimizer="adan", loss="binary crossentropy",metrics=['accuracy' ])

generator = model.fit(X\_train res,y\_ train\_ res,epochs=10, steps\_per\_epoch=len(X\_train\_res)//64)

**Activity2: Testing the model:**

Y\_pred

y\_pred=model.predict (x\_test)

y\_pr = np.where (y pred>0.5, 1,0)

y\_test

array( [0, 8, 0, ..., 0, 0, 0])

from sklearn . metrics import confusion \_matrix, accuracy\_Score

cm = confusion\_matrix(y\_test, y\_pr)

score = accuracy\_score(y\_test,y\_pr)

print (cm)

print('ACcuracy 5core Is:- ‘ ,score\*100)

def new review(new\_review):

new\_ review= new-review

new\_review = re.sub(^a-zA-Z]’, ‘ ‘, new\_review)

new\_review = new\_review.lower()

new\_review = new\_ review.split()

ps = Porterstemmer()

all\_stopwords = stopwords . words ('english’)

all \_stopwords .remove('not')

new\_review = [ps.stem(word) for word in new\_review if not word in set(all\_stopwords)]

new\_review =’ ‘.join (new\_review)

new\_corpus = [new \_review]

new\_ X\_ test = Cv.transform(new\_corpus).toarray()

print(new\_X\_test)

new \_y\_pred = loaded model.predict (new\_x\_ test)

print(new\_y\_ pred)

new \_X \_pred = np.where(new \_y\_ pred>0.5,1,0)

return new\_ y\_ pred

new\_review = new\_review(str(input ("Enter new review...”)))

Milesone5:

Activity 1.1

from sklearn.metrics import confusion\_matrix, accuracy\_score, classification\_report

cm = confusion\_matrix(y\_test, y\_pred)

Score = accuracy\_ score (y\_ test,y\_ pred)

print (cm)

print('Accuracy Score Is Naive Bayes:- ‘ ,score\*100)

cm = confusion\_matrix(y\_ test, y\_pred)

score = accuracy\_score(y\_test,y\_pred)

print (cm)

print('Accuracy Score Is:- ‘ ,score\*100)

cm1 = confusion\_matrix(y\_test, y\_predi)

scorel = accuracy\_score (y\_test,y\_pred1)

print (cm1)

print('Accuracy Score Is:- ‘ ,score1\*100)

from sklearn .metrics import confusion\_matrix, accuracy\_score

cm = confusion\_matrix(y\_test, y\_pr)

Score = accuracy\_score(y\_test,y\_pr)

print (cm)

print(‘ ACcuracy Score Is:- ‘, Score\*100)

Activity 2:

from sklearn .metrics import confusion\_matrix, accuracy\_score

cm = confusion\_matrix(y\_test, y\_pr)

Score = accuracy\_score(y\_test,y\_pr)

print (cm)

print(‘ ACcuracy Score Is:- ‘, Score\*100)

Milestone 6: Model Deployment

Activity :

Model.save(‘spam.h5’)