A Project Report

OPTIMIZING SPAM FILTERING WITH MACHINE LEARNING

Team ID : NM2023TMID17394

Team Leader : MUTHUGANESH R

Team members : MUTHUSELVAN K

: NAGAPRIYA H

: NANDHAKUMAR P

INDEX

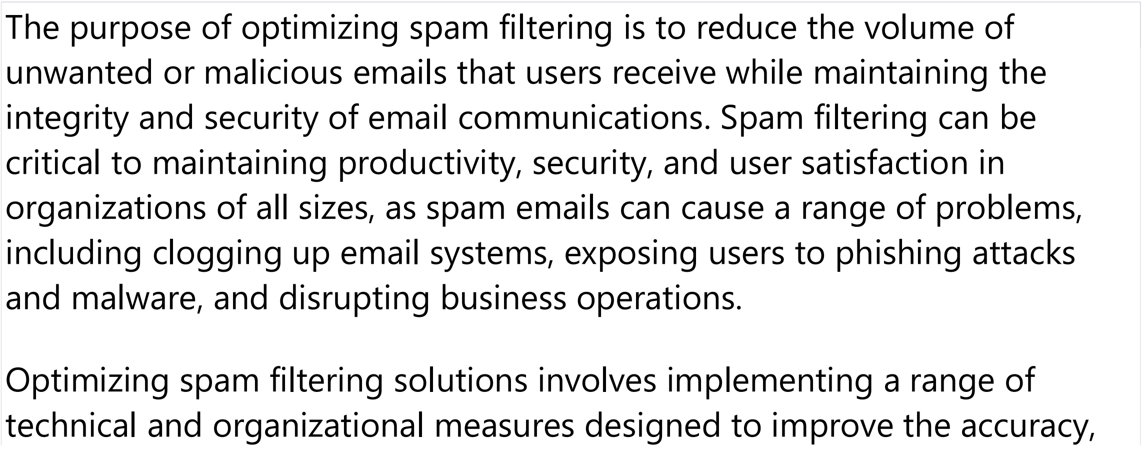
Optimizing Spam Filtering with Machine Learning

Introduction

1.1 Overview

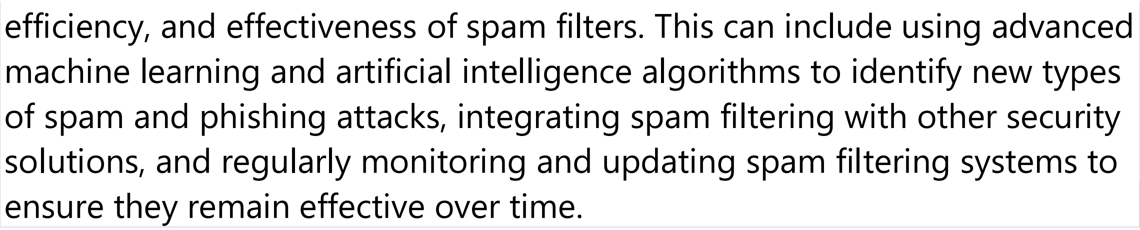
Over recent years, as the popularity of mobile phone devices has increased, Short Message Service (SMS) has grown into a multi-billion dollar industry. At the same time, reduction in the cost of messaging services has resulted in growth in unsolicited commercial advertisements (spams) being sent to mobile phones. Due to Spam SMS, Mobile service providers suffer from some sort of financial problems as well as it reduces calling time for users. Unfortunately, if the user accesses such Spam SMS they may face the problem of virus or malware. When SMS arrives at mobile it will disturb mobile user privacy and concentration. It may lead to frustration for the user. So Spam SMS is one of the major issues in the wireless communication world and it grows day by day. 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. It analyses text content and finds patterns which are used to identify Spam and Non-Spam SMS.

1.2 Purpose



The purpose of optimizing spam filtering is to reduce the volume of unwanted or malicious emails that users receive while maintaining the integrity and security of email communications. Spam filtering can be critical to maintaining productivity, security, and user satisfaction in organizations of all sizes, as spam emails can cause a range of problems, including clogging up email systems, exposing users to phishing attacks and malware, and disrupting business operations.

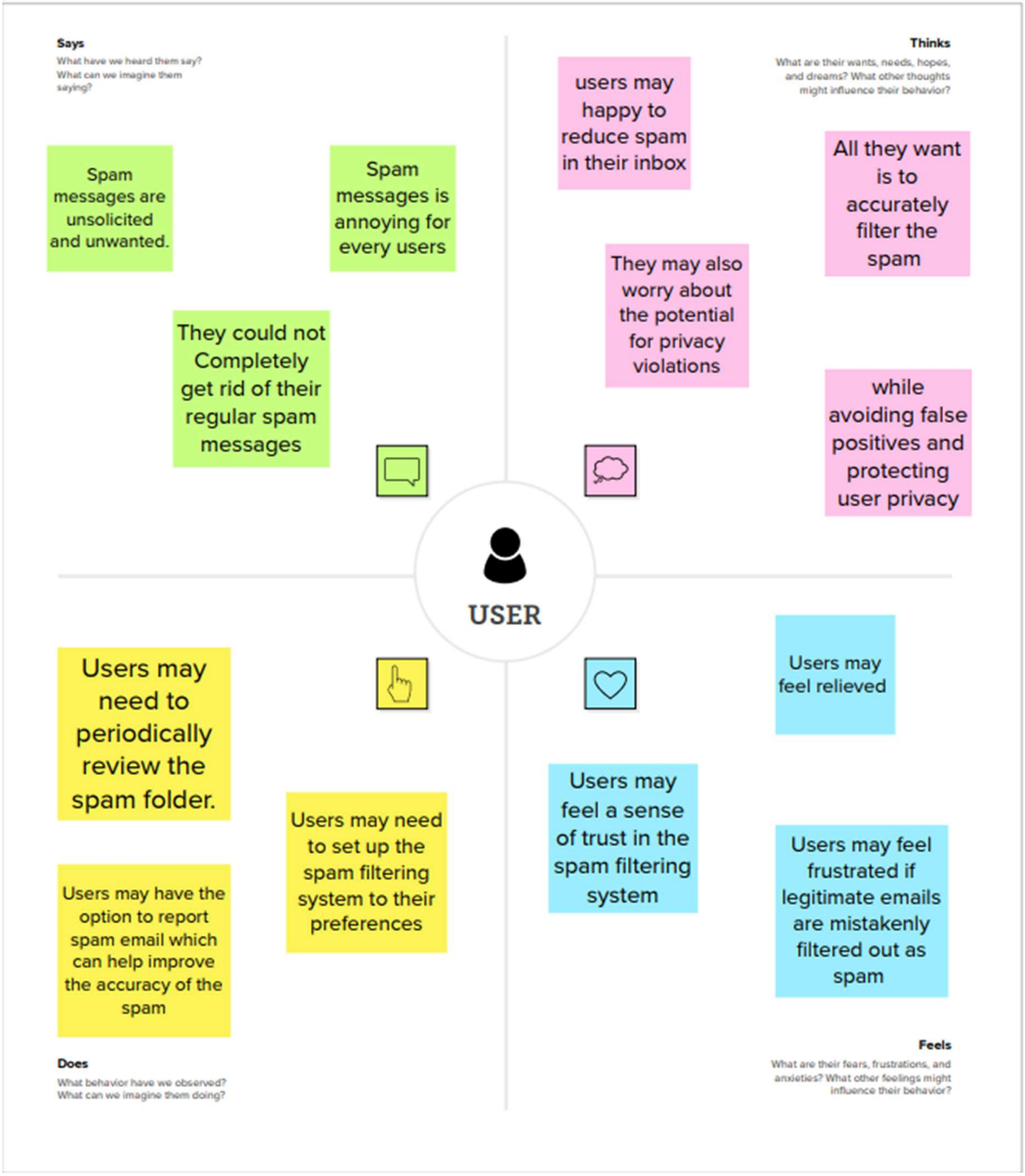
Optimizing spam filtering solutions involves implementing a range of technical and organizational measures designed to improve the accuracy,



efficiency, and effectiveness of spam filters. This can include using advanced machine learning and artificial intelligence algorithms to identify new types of spam and phishing attacks, integrating spam filtering with other security solutions, and regularly monitoring and updating spam filtering systems to ensure they remain effective over time.

Problem Definition & Design Thinking

2.1 Empathy Map



Says

What have we heard them say? What can we imagine them saying?

Spam messages are unsolicited and unwanted.

Spam messages is annoying for every users

They could not Completely get rid of their regular spam messages

Thinks

What are their wants, needs, hopes, and dreams? What other thoughts might influence their behavior?

users may happy to reduce spam in their inbox

All they want is to accurately filter the spam

They may also worry about the potential for privacy violations

while avoiding false positives and protecting user privacy

USER

Users may need to periodically review the spam folder.

Users may need to set up the spam filtering system to their preferences

Users may have the option to report spam email which can help improve the accuracy of the spam

Does

What behavior have we observed? What can we imagine them doing?

Users may feel relieved

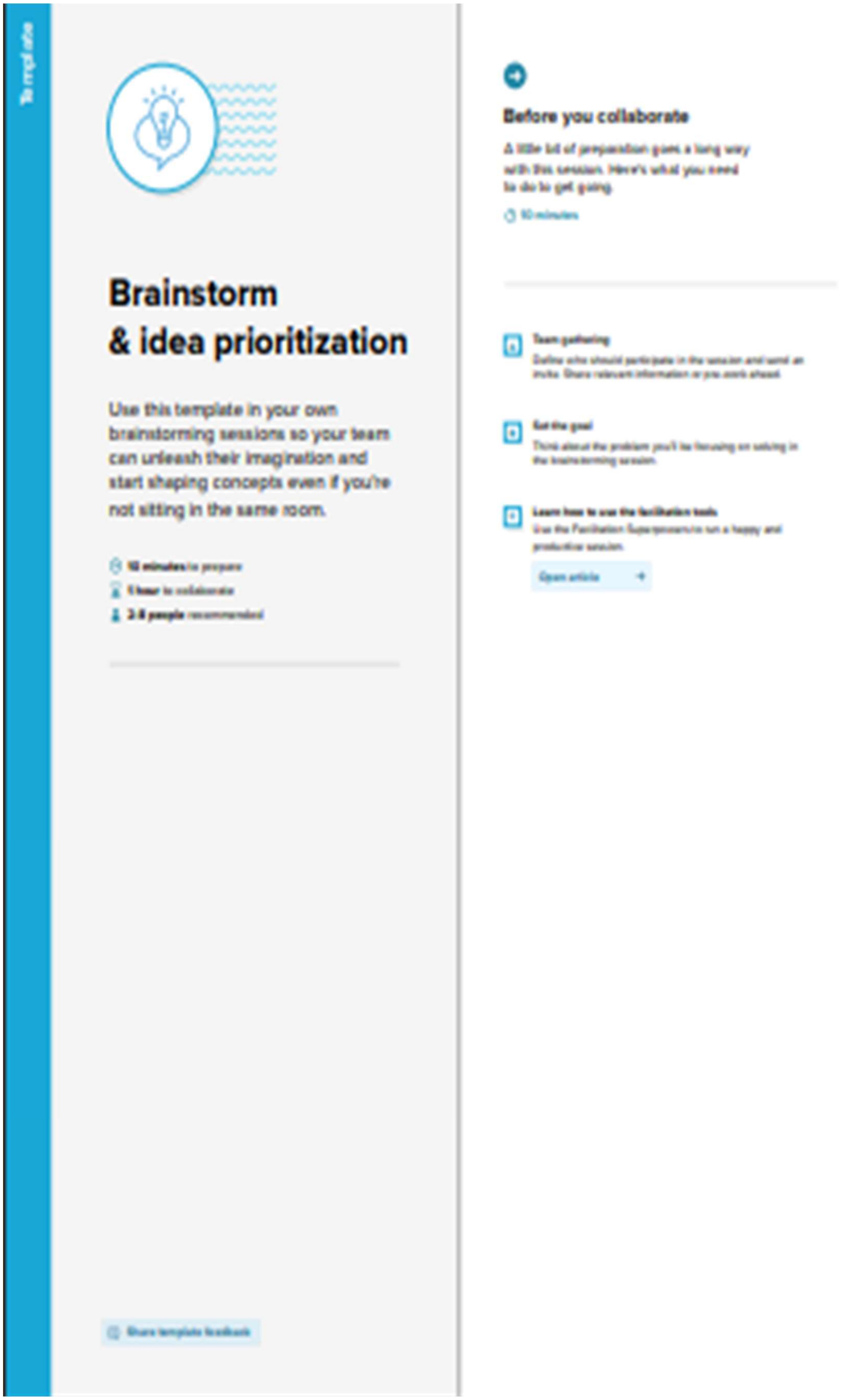
Users may feel a sense of trust in the spam filtering system

Users may feel frustrated if legitimate emails are mistakenly filtered out as spam

Feels

What are their fears, frustrations, and anxieties? What other feelings might influence their behavior?

2.1 Problem Definition & Design Thinking



Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare

1 hour to collaborate

2-8 people recommended

Before you collaborate

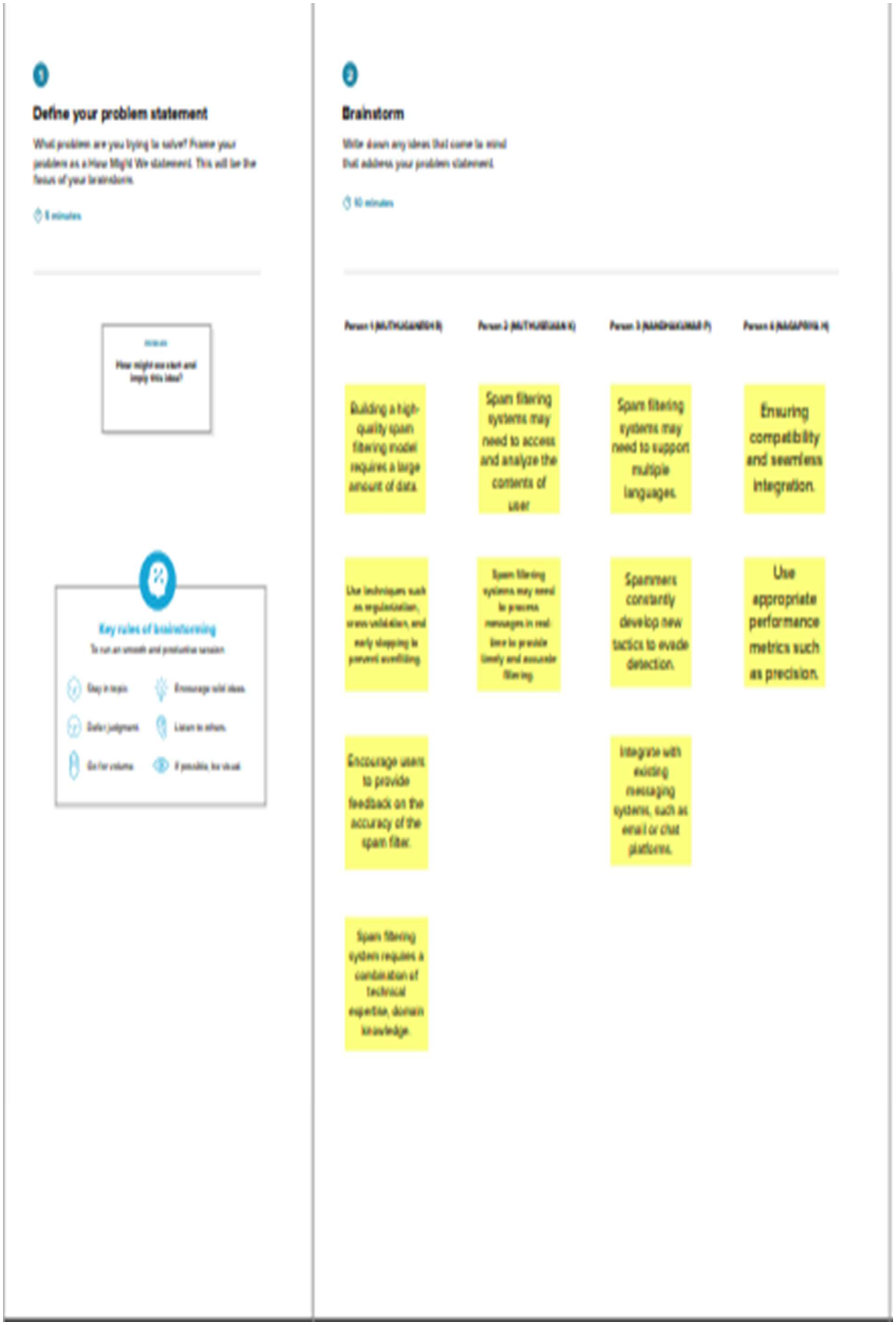
A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

Team gathering
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

Set the goal
Think about the problem you'll be focusing on solving in the brainstorming session.

Learn how to use the facilitation tools
Use the Facilitation Superpowers to run a happy and productive session.



1

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM

How might we start and imply this idea?

Key rules of brainstorming

To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.

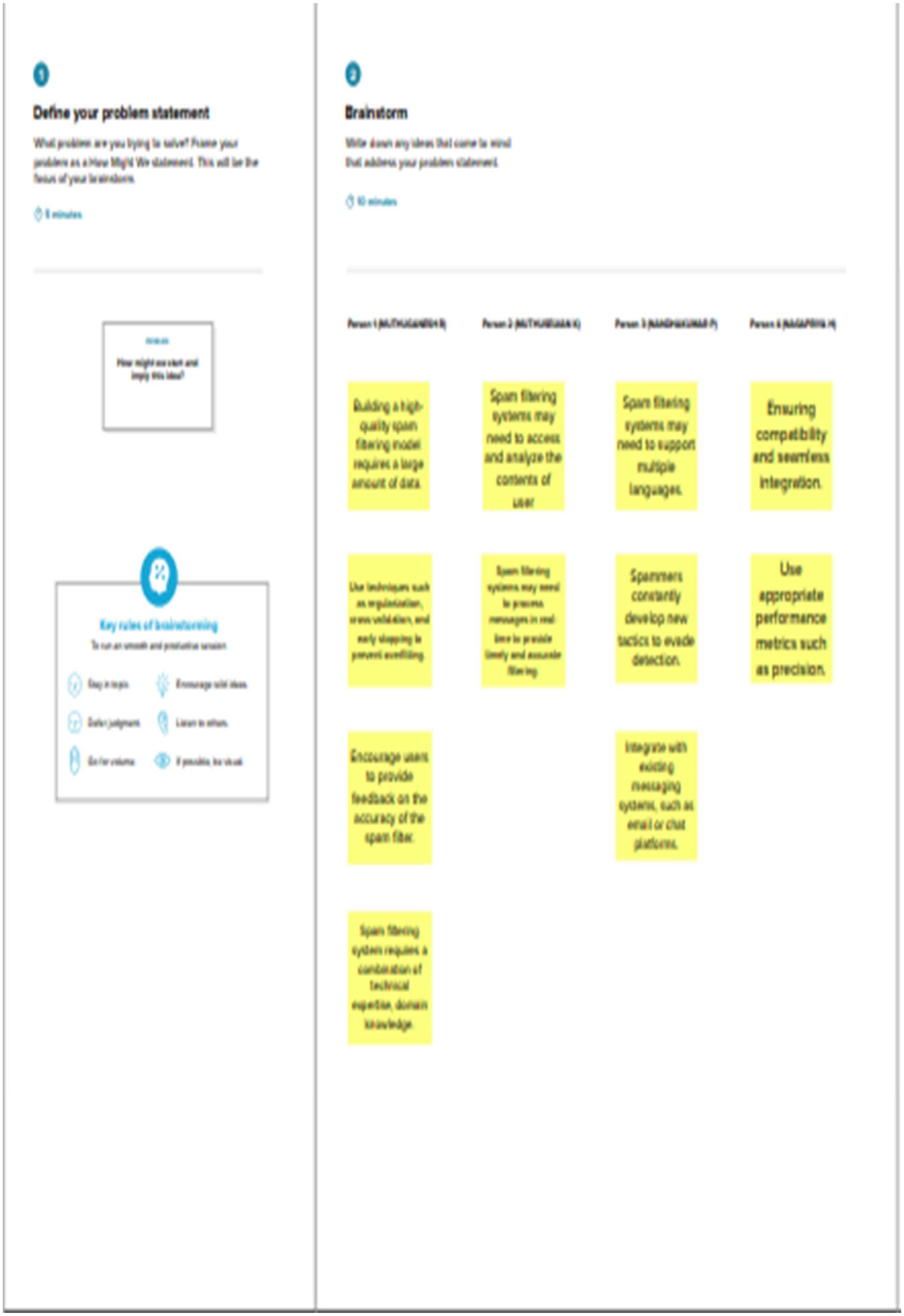
2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

Person 1 (MUTHUGANESH B)	Person 2 (MUTHUSELVAN K)	Person 3 (NANDHAKUMAR P)	Person 4 (NAGAPRIYA M)
Building a high-quality spam filtering model requires a large amount of data.	Spam filtering systems may need to access and analyze the contents of user	Spam filtering systems may need to support multiple languages.	Ensuring compatibility and seamless integration.
Use techniques such as regularization, cross-validation, and early stopping to prevent overfitting.	Spam filtering systems may need to process messages in real-time to provide timely and accurate filtering.	Spammers constantly develop new tactics to evade detection.	Use appropriate performance metrics such as precision.
Encourage users to provide feedback on the accuracy of the spam filter.		Integrate with existing messaging systems, such as email or chat platforms.	
Spam filtering system requires a combination of technical expertise, domain knowledge.			



1

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM

How might we start and imply this idea?

Key rules of brainstorming

To run an smooth and productive session

Stay in topic.

Encourage wild ideas.

Defer judgment.

Listen to others.

Go for volume.

If possible, be visual.

2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

Person 1 (MUTHUGANESH R)

Building a high-quality spam filtering model requires a large amount of data.

Use techniques such as regularization, cross-validation, and early stopping to prevent overfitting.

Encourage users to provide feedback on the accuracy of the spam filter.

Spam filtering system requires a combination of technical expertise, domain knowledge.

Person 2 (MUTHUSELVAN K)

Spam filtering systems may need to access and analyze the contents of user

Spam filtering systems may need to process messages in real-time to provide timely and accurate filtering.

Person 3 (NANDHAKUMAR P)

Spam filtering systems may need to support multiple languages.

Spammers constantly develop new tactics to evade detection.

Integrate with existing messaging systems, such as email or chat platforms.

Person 4 (NAGAPRIYA H)

Ensuring compatibility and seamless integration.

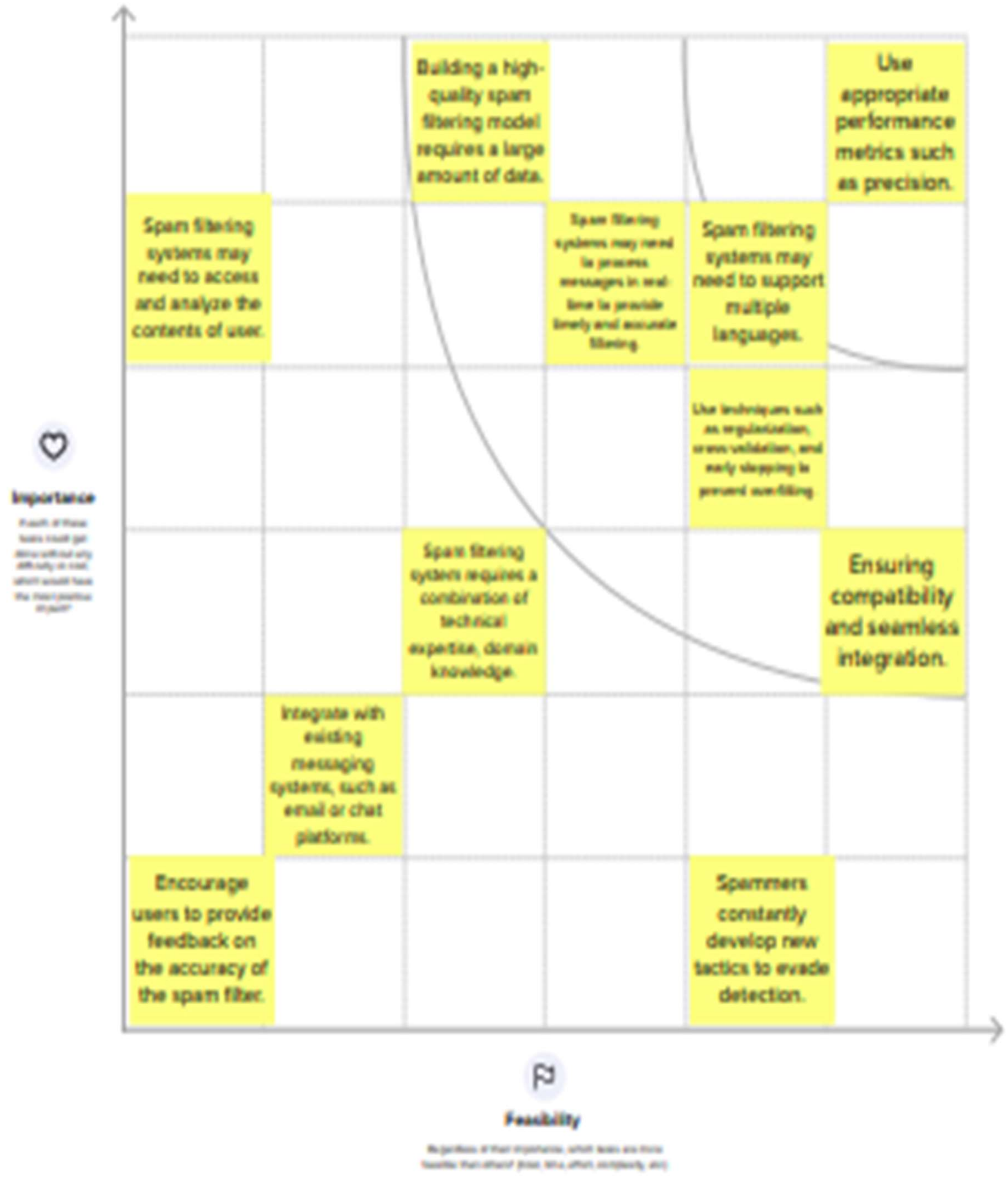
Use appropriate performance metrics such as precision.

4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes



Building a high-quality spam filtering model requires a large amount of data.

Use appropriate performance metrics such as precision.

Spam filtering systems may need to access and analyze the contents of user.

Spam filtering systems may need to process messages in real-time to provide timely and accurate filtering.

Spam filtering systems may need to support multiple languages.

Use techniques such as regularization, cross-validation, and early stopping to prevent overfitting.

Spam filtering system requires a combination of technical expertise, domain knowledge.

Ensuring compatibility and seamless integration.

Integrate with existing messaging systems, such as email or chat platforms.

Encourage users to provide feedback on the accuracy of the spam filter.

Spammers constantly develop new tactics to evade detection.

Importance

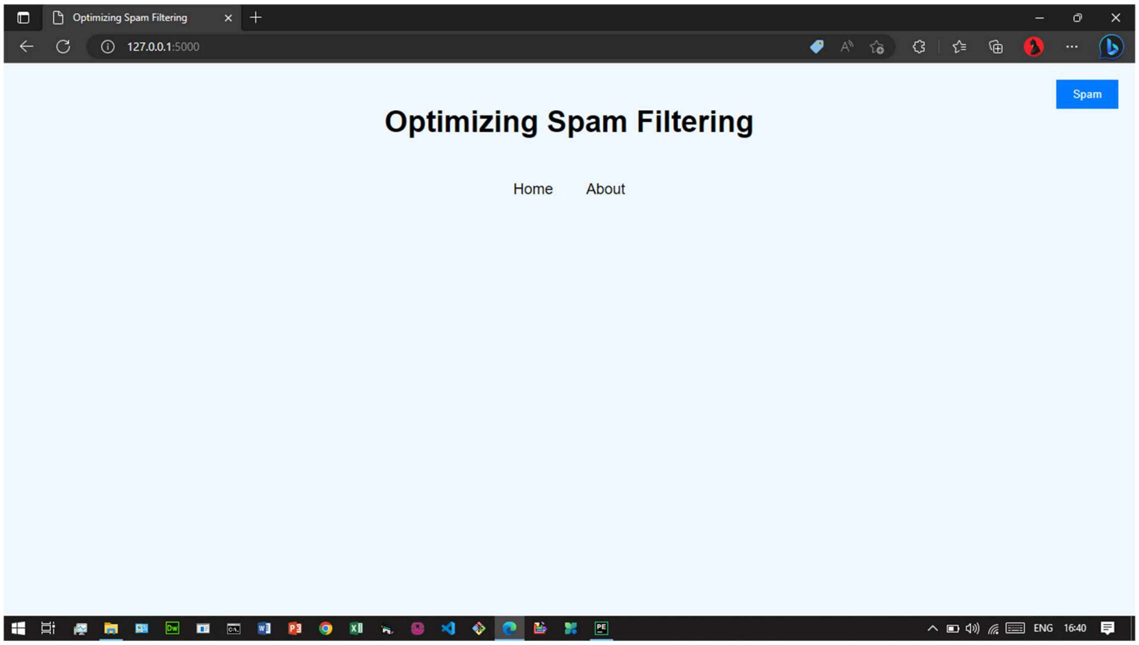
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Feasibility

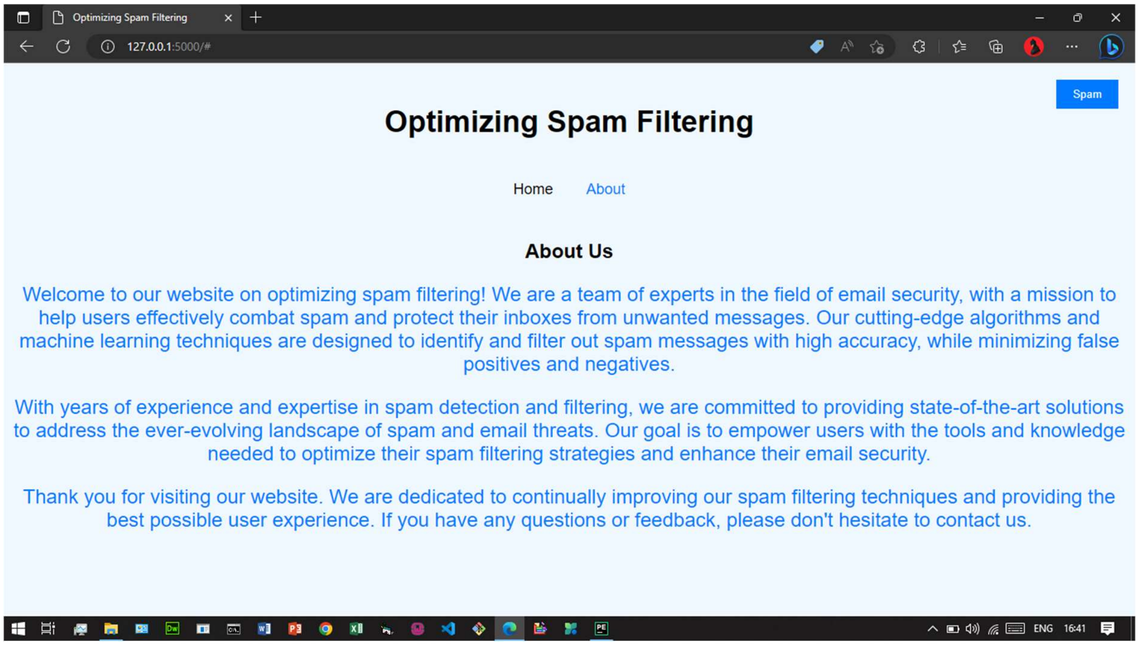
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

Result

Output:



Optimizing Spam Filtering



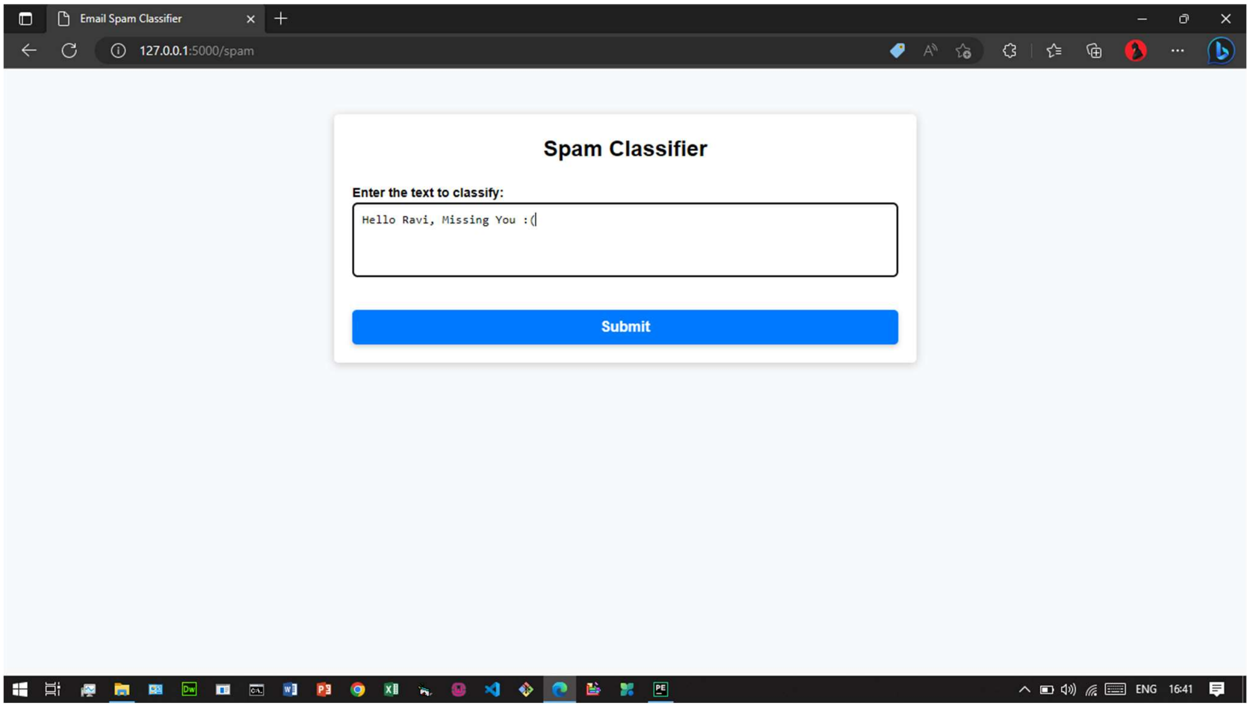
Optimizing Spam Filtering

About Us

Welcome to our website on optimizing spam filtering! We are a team of experts in the field of email security, with a mission to help users effectively combat spam and protect their inboxes from unwanted messages. Our cutting-edge algorithms and machine learning techniques are designed to identify and filter out spam messages with high accuracy, while minimizing false positives and negatives.

With years of experience and expertise in spam detection and filtering, we are committed to providing state-of-the-art solutions to address the ever-evolving landscape of spam and email threats. Our goal is to empower users with the tools and knowledge needed to optimize their spam filtering strategies and enhance their email security.

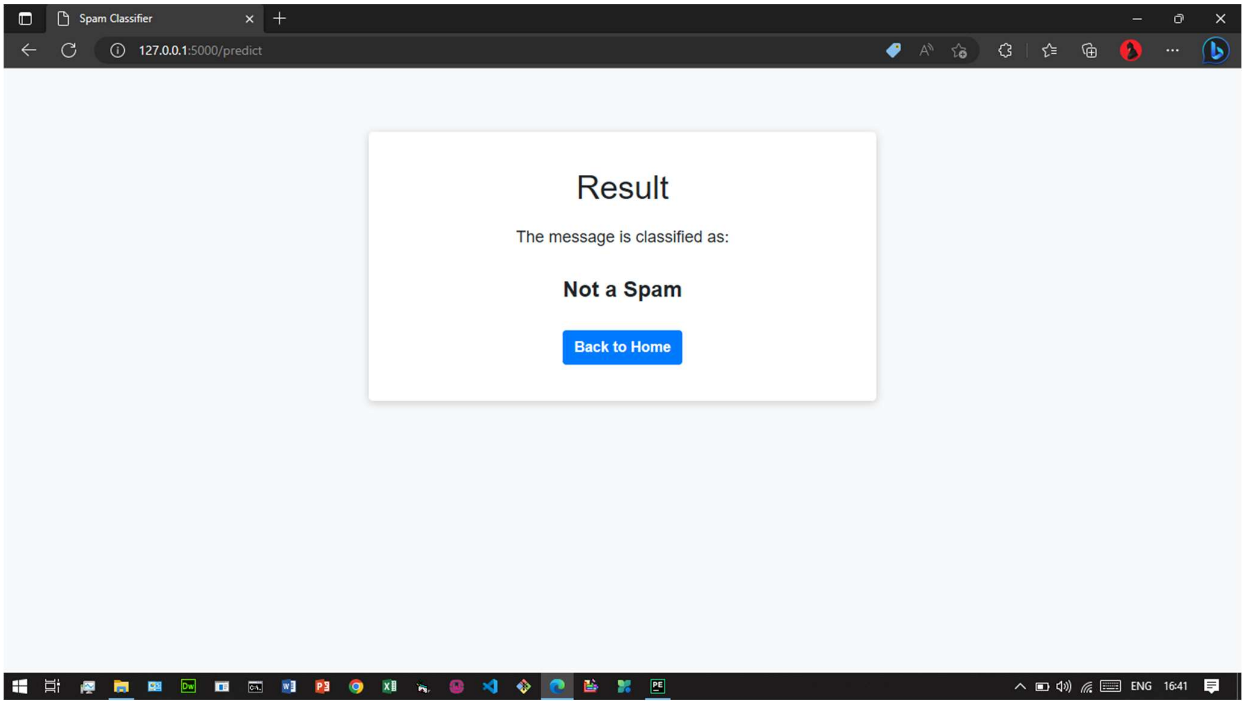
Thank you for visiting our website. We are dedicated to continually improving our spam filtering techniques and providing the best possible user experience. If you have any questions or feedback, please don't hesitate to contact us.



Spam Classifier

Enter the text to classify:

Hello Ravi, Missing You :(



Result

The message is classified as:

Not a Spam

ADVANTAGES & DISADVANTAGES

Advantages

1. Improved user experience: By filtering out spam emails, users' inboxes become less cluttered, making it easier to find important messages. This improves the overall user experience and saves time.
2. Increased productivity: Spam filtering helps employees to focus on important tasks rather than wasting time sorting through spam messages.
3. Enhanced security: Spam emails can contain malicious links or attachments that can harm your computer system. By filtering out spam, you reduce the risk of falling victim to phishing attacks or malware infections.
4. Cost savings: By reducing the amount of spam that reaches your inbox, you can save on storage and bandwidth costs associated with storing and downloading these messages.
5. Regulatory compliance: Some industries are required to maintain certain levels of email security and privacy. By implementing effective spam filtering, you can ensure that you meet these requirements and avoid potential fines or penalties.

Overall, optimizing spam filtering can provide significant advantages in terms of user experience, productivity, security, cost savings, and regulatory compliance.

Disadvantages

1. False positives: Overly aggressive spam filtering can sometimes result in legitimate emails being marked as spam and filtered out. This can lead to missed opportunities and frustration for users who expect to receive certain emails.
2. Complexity: Spam filtering can be a complex process that requires significant resources and expertise to implement and maintain. This can be a challenge for smaller organizations or those with limited IT resources.
3. Cost: Implementing effective spam filtering solutions can be costly, particularly for larger organizations. This can include hardware and software costs, as well as ongoing maintenance and support expenses.
4. User dissatisfaction: Some users may feel that spam filtering is too restrictive and may prefer to receive all emails, even if it means sorting through a large number of spam messages. This can lead to dissatisfaction and reduced productivity for these users.

Overall, while the benefits of optimizing spam filtering are significant, there are some potential downsides to consider as well. It is important to balance the need for effective spam filtering with the potential impact on users and the organization as a whole.

APPLICATION

Application of Spam Filtering:

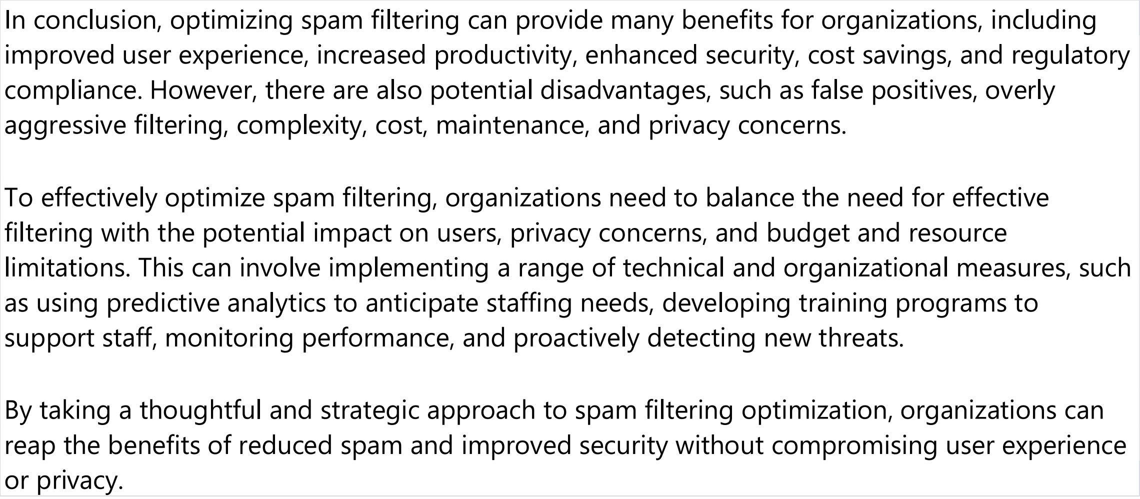
The application of optimizing spam filtering is relevant in a variety of settings, including:

1. Businesses: Spam filtering is critical for businesses of all sizes, as unwanted or malicious emails can disrupt operations, harm productivity, and compromise sensitive information. By optimizing spam filtering, businesses can reduce the risk of email-based threats, protect their assets, and maintain a secure and productive work environment.
2. Educational institutions: Spam filtering is important in educational institutions, where students and staff rely heavily on email for communication and collaboration. Optimizing spam filtering can help reduce the volume of spam emails, prevent phishing attacks, and protect the privacy of students and staff.
3. Government agencies: Government agencies handle sensitive and classified information, making them prime targets for email-based attacks. By optimizing spam filtering, government agencies can reduce the risk of email-based threats, protect national security, and ensure the integrity of communications.
4. Healthcare organizations: Healthcare organizations handle sensitive patient information, making them vulnerable to email-based attacks. Optimizing spam filtering can help protect patient privacy, prevent data breaches, and maintain compliance with healthcare regulations.
5. Individuals: Spam filtering is also important for individual users, who rely on email for personal and professional communication. By optimizing spam filtering, individuals can reduce the risk of phishing attacks, protect their privacy, and keep their email inbox organized and manageable.

Overall, optimizing spam filtering is applicable in a wide range of settings and can help ensure the security, privacy, and productivity of email communications.

CONCLUSION

Conclusion:



In conclusion, optimizing spam filtering can provide many benefits for organizations, including improved user experience, increased productivity, enhanced security, cost savings, and regulatory compliance. However, there are also potential disadvantages, such as false positives, overly aggressive filtering, complexity, cost, maintenance, and privacy concerns.

To effectively optimize spam filtering, organizations need to balance the need for effective filtering with the potential impact on users, privacy concerns, and budget and resource limitations. This can involve implementing a range of technical and organizational measures, such as using predictive analytics to anticipate staffing needs, developing training programs to support staff, monitoring performance, and proactively detecting new threats.

By taking a thoughtful and strategic approach to spam filtering optimization, organizations can reap the benefits of reduced spam and improved security without compromising user experience or privacy.

FUTURE SCOPE

Future Scope:

The future scope of optimizing spam filtering is promising, as new technologies and approaches continue to emerge that can enhance the effectiveness and efficiency of spam filtering solutions. Some potential areas of future development include:

1. Artificial intelligence and machine learning: By leveraging AI and machine learning, spam filters can become smarter and more sophisticated over time. These technologies can help filters learn to identify new types of spam and phishing attacks, adapt to changing email patterns and behaviors, and improve accuracy and performance.
2. Big data analytics: With the massive amounts of data generated by email systems, big data analytics can be used to identify patterns and trends in spam email and help organizations make more informed decisions about how to optimize their spam filtering solutions.
3. Cloud-based spam filtering: As more organizations move their IT infrastructure to the cloud, cloud-based spam filtering solutions are becoming more popular. These solutions offer scalability, flexibility, and cost savings, making them an attractive option for organizations of all sizes.
4. Integration with other security solutions: Spam filtering can be integrated with other security solutions, such as anti-virus and anti-malware software, to provide a more comprehensive defense against email-based threats.
5. Blockchain-based spam filtering: Blockchain technology can be used to create decentralized spam filtering solutions that are resistant to tampering and manipulation. These solutions can offer greater security and transparency, particularly in high-risk environments.

Overall, the future of optimizing spam filtering looks bright, with new technologies and approaches emerging that can help organizations improve their email security and enhance user experience. By staying up-to-date on the latest developments and best practices, organizations can continue to optimize their spam filtering solutions to meet the evolving needs of their users and business operations.

APPENDIX

Source Code

1. app.py(Source Code)

```
from flask import Flask, render_template, request
import pickle
import numpy as np
import re
import nltk
from nltk.stem.porter import PorterStemmer
from nltk.corpus import stopwords
from tensorflow.keras.models import load_model
import os

load_model = load_model('spam.h5')
cv = pickle.load(open('cv1.pkl','rb'))
app = Flask(__name__)

@app.route('/')
def home():
  return render_template('home.html')

@app.route('/Spam', methods=['POST','GET'])
def prediction():
  return render_template('spam.html')

@app.route('/predict', methods=['POST'])
```

```
def predict():
  if request.method =='POST':
     message=request.form['message']
     data=message

  new_review = str(data)
  print(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 = load_model.predict(new_X_test)
  new_X_pred = np.where(new_y_pred>0.5,1,0)
  return new_X_pred
  if new_review[0][0]==1:
    return render_template('result.html', prediction="Spam")
  else :
    return render_template('result.html', prediction="Not a
Spam")

if __name__ == "__main__":
    #app.run(host='0.0.0.0',port=8000,debug=True)  #running
the app
    port=int(os.environ.get('PORT',8000))
    app.run(debug=False)
```

2.spam_filtering_py

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import nltk
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
import re
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from imblearn.over_sampling import SMOTE
import pickle
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import MultinomialNB
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from sklearn.metrics import confusion_matrix,
accuracy_score, classification_report

df = pd.read_csv("spam.csv",encoding="latin")
df.head()

df.info()
df.isna().sum()

df.rename({"v1":"label", "v2":"text"},inplace=True,axis=1)
df.tail()
```

```
le=LabelEncoder()
df['label']=le.fit_transform(df['label'])

nltk.download("stopwords")
corpus = []
length=len(df)
ps = PorterStemmer()

for i in range(0,length):
    text = df['text'][i]
    text = re.sub('[^a-zA-Z]', ' ', text)
    text=text.lower()
    text = text.split()
    pe=PorterStemmer()
    stopword = stopwords.words("english")
    text = [ps.stem(word) for word in text if not word in
set(stopword)]
    text = ' '.join(text)
    corpus.append(text)
corpus

cv = CountVectorizer(max_features = 35000)
X = cv.fit_transform(corpus).toarray()
y = df['label'].values

X_train, X_test, y_train, y_test = train_test_split( X, y,
test_size = 0.20, random_state = 0)
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 '0':
{}".format(sum(y_train_res == 0)))
pickle.dump(cv, open('cv1.pkl', 'wb'))

df.describe()

df.shape

df["label"].value_counts().plot(kind="bar", figsize=(12,6))
plt.xticks(np.arange(2),('Non spam','spam'), rotation=0);

sc = StandardScaler()
x_bal = sc.fit_transform(X)

x_bal =pd.DataFrame(x_bal)

model = MultinomialNB()
model.fit(X_train_res,y_train_res)

model = Sequential()

X_train.shape

model.add(Dense(units =
X_train_res.shape[1],activation="relu",kernel_initializer="ra
ndom_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 = 1,activation="sigmoid"))
model.compile(optimizer="adam",loss="binary_crossentropy
",metrics=['accuracy'])
generator =
model.fit(X_train_res,y_train_res,epochs=10,steps_per_epoc
h=len(X_train_res)//64)

y_pred = model.predict(X_test)
y_pred

y_pr = np.where(y_pred>0.5,1,0)
y_test

cm = confusion_matrix(y_test, y_pr)
score = accuracy_score(y_test, y_pr)
print(cm)
print('Accuracy Score 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 = 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...")))
threshold = 0.5
y_pred = (y_pred > threshold).astype(int)

cm = confusion_matrix(y_test, y_pred)
score = accuracy_score(y_test, y_pred)
print(cm)
print('Accuracy Score Is:- ',score*100)

model.save('spam.h5')
```

3. index.html

```
<!DOCTYPE html>

<html>

<head>

   <title>Optimizing Spam Filtering</title>

   <style>

   body {

          font-family: Arial, sans-serif;

          margin: 0;
```

```
    padding: 0;
    background-color: #f0fff0;

}
.header {
    text-align: center;
    margin-top: 50px;
}
.header h1 {
    font-size: 36px;
}
.nav {
    display: flex;
    justify-content: center;
    margin-top: 50px;
}
.nav a {
    margin: 0 20px;
    text-decoration: none;
    color: #000;
    font-size: 18px;
}
.nav a:hover {
```

```
            color: #007bff;
        }
        .spam-button {
            position: absolute;
            top: 20px;
            right: 20px;
        }
        .spam-button button {
            background-color: #007bff;
            color: #fff;
            border: none;
            padding: 10px 20px;
            cursor: pointer;
        }
        .spam-button button:hover {
            background-color: #0056b3;
        }
    </style>
</head>
<body>
    <header class="header">
        <h1>Optimizing Spam Filtering</h1>
    </header>
```

```
   <nav class="nav">
      <a href="index.html">Home</a>
      <a href="about.html">About</a>
   </nav>
   <div class="spam-button">
      <button onclick="location.href='spam.html'">Spam</button>
   </div>
</body>
</html>
```

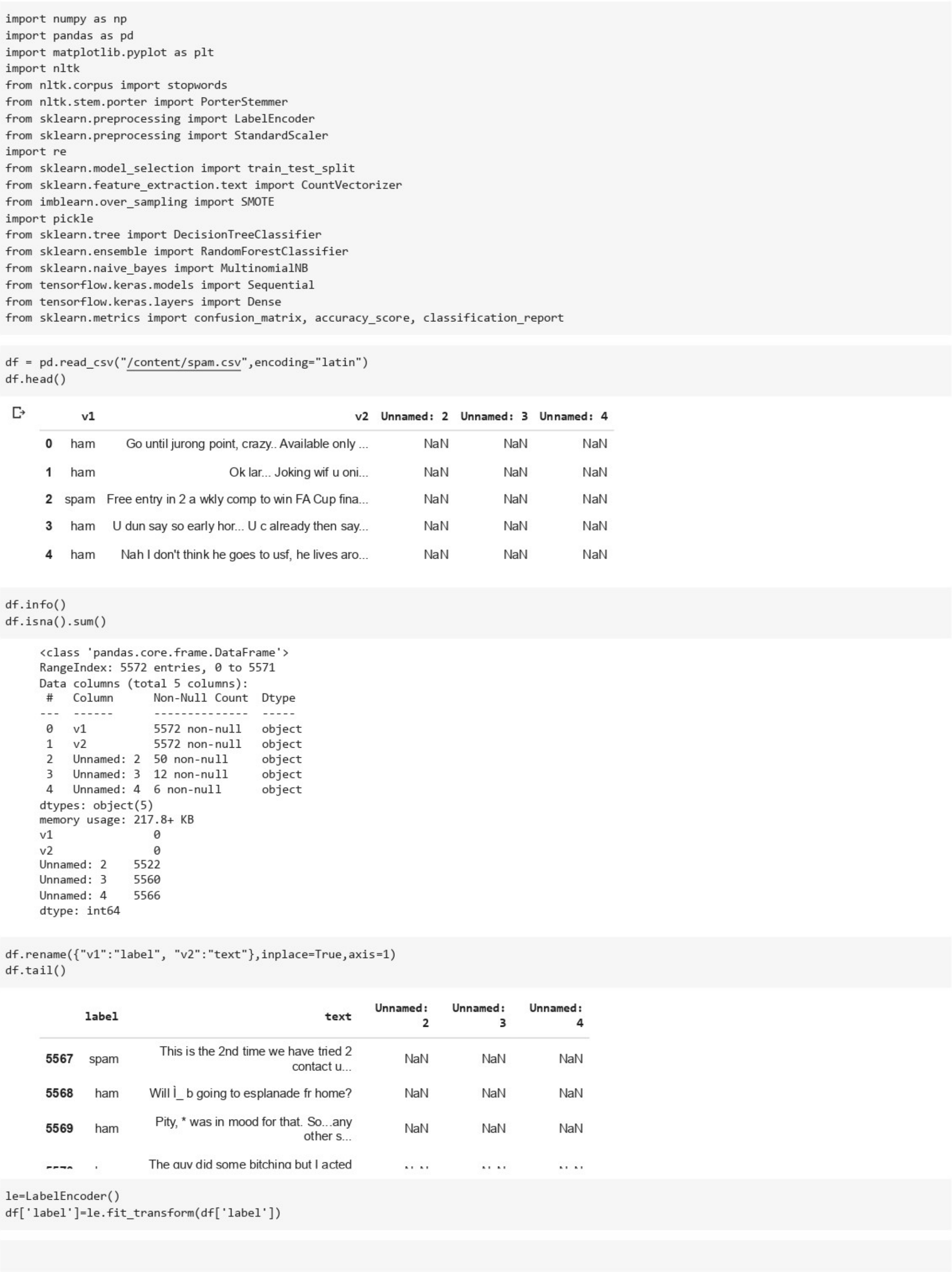
4. spam.html

```
 <!DOCTYPE html>
<html>
<head>
   <title>Spam Classifier</title>
   <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.
min.css">
   <style>
      .jumbotron {
            background-color: #f8f9fa;
            margin-top: 50px;
        }
    </style>
</head>
<body>
    <div class="container">
        <div class="jumbotron">
            <h1 class="display-4">Spam Classifier</h1>
            <p class="lead">Enter a message to classify:</p>
            <form action="/predict" method="POST">
                <div class="form-group">
                    <textarea class="form-control" rows="5" id="message" name="message" required></textarea>
                </div>
                <button type="submit" class="btn btn-primary">Submit</button>
            </form>
        </div>
    </div>
</body>
</html>
```

5. result.html

```
<!DOCTYPE html>
<html>
<head>
   <title>Spam Classifier</title>
   <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/b
ootstrap.min.css">
   <style>
      .jumbotron {
         background-color: #f8f9fa;
         margin-top: 50px;
      }
   </style>
</head>
<body>
   <div class="container">
      <div class="jumbotron">
         <h1 class="display-4">Result</h1>
         <p class="lead">The message is classified as:</p>
         <h2>{{ prediction }}</h2>
         <a href="/" class="btn btn-primary">Back to
Home</a>
      </div>
   </div>
</body>
</html>
```

6. Spam_Filtering_py.ipynb



```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import nltk
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
import re
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from imblearn.over_sampling import SMOTE
import pickle
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import MultinomialNB
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
```

```
df = pd.read_csv("/content/spam.csv",encoding="latin")
df.head()
```

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN

```
df.info()
df.isna().sum()
```

```
<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
v1               0
v2               0
Unnamed: 2    5522
Unnamed: 3    5560
Unnamed: 4    5566
dtype: int64
```

```
df.rename({"v1":"label", "v2":"text"},inplace=True,axis=1)
df.tail()
```

	label	text	Unnamed: 2	Unnamed: 3	Unnamed: 4
5567	spam	This is the 2nd time we have tried 2 contact u...	NaN	NaN	NaN
5568	ham	Will Ì_ b going to esplanade fr home?	NaN	NaN	NaN
5569	ham	Pity, * was in mood for that. So...any other s...	NaN	NaN	NaN
		The guy did some bitching but I acted			

```
le=LabelEncoder()
df['label']=le.fit_transform(df['label'])
```

```
nltk.download("stopwords")
corpus = []
length=len(df)
ps = PorterStemmer()
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

```
for i in range(0,length):
    text = df['text'][i]
    text = re.sub('[^a-zA-Z]', ' ', text)
    text=text.lower()
    text = text.split()
    pe=PorterStemmer()
    stopword = stopwords.words("english")
    text = [ps.stem(word) for word in text if not word in set(stopword)]
    text = ' '.join(text)
    corpus.append(text)
corpus
```

```
['go jurong point crazi avail bugi n great world la e buffet cine got amor wat',
 'ok lar joke wif u oni',
 'free entri wkli comp win fa cup final tkt st may text fa receiv entri question std txt rate c appli',
 'u dun say earli hor u c alreadi say',
 'nah think goe usf live around though',
 'freemsg hey darl week word back like fun still tb ok xxx std chg send rcv',
 'even brother like speak treat like aid patent',
 'per request mell mell oru minnaminungint nurungu vettam set callertun caller press copi friend callertun',
 'winner valu network custom select receivea prize reward claim call claim code kl valid hour',
 'mobil month u r entitl updat latest colour mobil camera free call mobil updat co free',
 'gonna home soon want talk stuff anymor tonight k cri enough today',
 'six chanc win cash pound txt csh send cost p day day tsandc appli repli hl info',
 'urgent week free membership prize jackpot txt word claim c www dbuk net lccltd pobox ldnw rw',
 'search right word thank breather promis wont take help grant fulfil promis wonder bless time',
 'date sunday',
 'xxxmobilemovieclub use credit click wap link next txt messag click http wap xxxmobilemovieclub com n qjkgighjjgcbl',
 'oh k watch',
 'eh u rememb spell name ye v naughti make v wet',
 'fine way u feel way gota b',
 'england v macedonia dont miss goal team news txt ur nation team eg england tri wale scotland txt poboxox w wq',
 'serious spell name',
 'go tri month ha ha joke',
 'pay first lar da stock comin',
 'aft finish lunch go str lor ard smth lor u finish ur lunch alreadi',
 'ffffffffff alright way meet sooner',
 'forc eat slice realli hungri tho suck mark get worri know sick turn pizza lol',
 'lol alway convinc',
 'catch bu fri egg make tea eat mom left dinner feel love',
 'back amp pack car let know room',
 'ahhh work vagu rememb feel like lol',
 'wait still clear sure sarcast x want live us',
 'yeah got v apologet n fallen actin like spoilt child got caught till go badli cheer',
 'k tell anyth',
 'fear faint housework quick cuppa',
 'thank subscript rington uk mobil charg month pleas confirm repli ye repli charg',
 'yup ok go home look time msg xuhui go learn nd may lesson',
 'oop let know roommat done',
 'see letter b car',
 'anyth lor u decid',
 'hello saturday go text see decid anyth tomo tri invit anyth',
 'pl go ahead watt want sure great weekend abiola',
 'forget tell want need crave love sweet arabian steed mmmmmm yummi',
 'rodger burn msg tri call repli sm free nokia mobil free camcord pleas call deliveri tomorrow',
 'see',
 'great hope like man well endow lt gt inch',
 'call messag miss call',
 'get hep b immunis nigeria',
 'fair enough anyth go',
 'yeah hope tyler could mayb ask around bit',
 'u know stubborn even want go hospit kept tell mark weak sucker hospit weak sucker',
 'think first time saw class',
 'gram usual run like lt gt half eighth smarter though get almost whole second gram lt gt',
 'k fyi x ride earli tomorrow morn crash place tonight',
 'wow never realiz embarass accomod thought like sinc best could alway seem happi cave sorri give sorri offer sorri room embarass',
 'sm ac sptv new jersey devil detroit red wing play ice hockey correct incorrect end repli end sptv',
 'know mallika sherawat yesterday find lt url gt',
 'congrat year special cinema pass call c suprman v matrix starwar etc free bx ip pm dont miss',
 'sorri call later meet',
```

```
cv = CountVectorizer(max_features = 35000)
X = cv.fit_transform(corpus).toarray()
```

```
y = df['label'].values

X_train, X_test, y_train, y_test = train_test_split( X, y, test_size = 0.20, random_state = 0)
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 '0': {}".format(sum(y_train_res == 0)))
pickle.dump(cv, open('cv1.pkl', 'wb'))
```

```
After Oversampling, the shape of train X: (7752, 6221)
After Oversampling, the shape of train y: (7752,)

After Oversampling, counts of label '1': 3876
After oversampling, counts of label '0': 3876
```

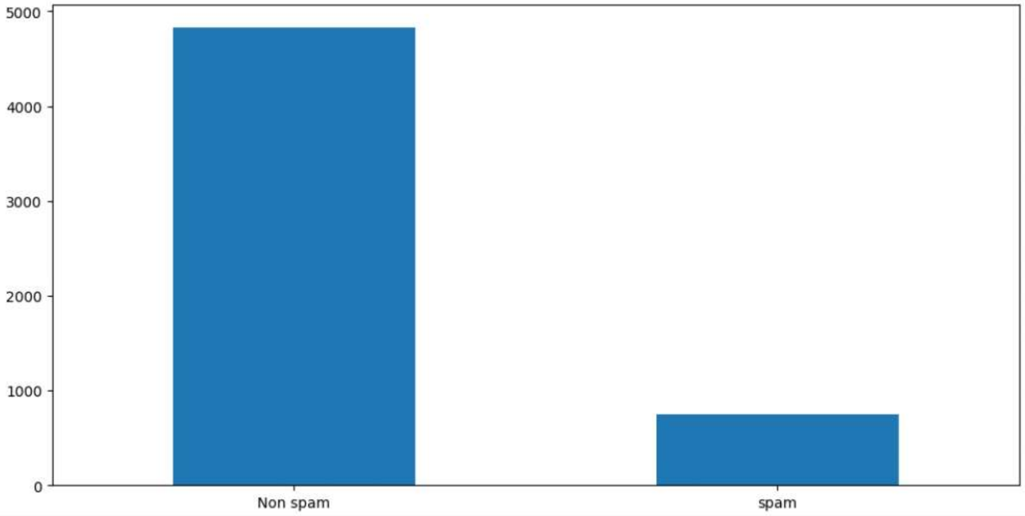
```
df.describe()
```

	label
count	5572.000000
mean	0.134063
std	0.340751
min	0.000000
25%	0.000000
50%	0.000000
75%	0.000000
max	1.000000

```
df.shape
```

```
(5572, 5)
```

```
df["label"].value_counts().plot(kind="bar", figsize=(12,6))
plt.xticks(np.arange(2),('Non spam','spam'), rotation=0);
```



```
sc = StandardScaler()
x_bal = sc.fit_transform(X)
x_bal =pd.DataFrame(x_bal)
```

```
model = MultinomialNB()
model.fit(X_train_res,y_train_res)
```

```
▾ MultinomialNB
MultinomialNB()
```

```
model = Sequential()
```

```
X_train.shape
```

```
(4457, 6221)
```

```
model.add(Dense(units = X_train_res.shape[1],activation="relu",kernel_initializer="random_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 = 1,activation="sigmoid"))
model.compile(optimizer="adam",loss="binary_crossentropy",metrics=['accuracy'])
generator = model.fit(X_train_res,y_train_res,epochs=10,steps_per_epoch=len(X_train_res)//64)
```

```
Epoch 1/10
121/121 [==============================] - 97s 790ms/step - loss: 0.1486 - accuracy: 0.9514
Epoch 2/10
121/121 [==============================] - 108s 896ms/step - loss: 0.0253 - accuracy: 0.9940
Epoch 3/10
121/121 [==============================] - 97s 799ms/step - loss: 0.0176 - accuracy: 0.9962
Epoch 4/10
121/121 [==============================] - 96s 794ms/step - loss: 0.0161 - accuracy: 0.9964
Epoch 5/10
121/121 [==============================] - 107s 882ms/step - loss: 0.0137 - accuracy: 0.9971
Epoch 6/10
121/121 [==============================] - 99s 817ms/step - loss: 0.0134 - accuracy: 0.9972
Epoch 7/10
121/121 [==============================] - 99s 816ms/step - loss: 0.0144 - accuracy: 0.9969
Epoch 8/10
121/121 [==============================] - 97s 806ms/step - loss: 0.0140 - accuracy: 0.9968
Epoch 9/10
121/121 [==============================] - 99s 817ms/step - loss: 0.0136 - accuracy: 0.9972
Epoch 10/10
111/121 [==========================>...] - ETA: 8s - loss: 0.0114 - accuracy: 0.9976WARNING:tensorflow:Your input ran out of data; inter
121/121 [==============================] - 91s 749ms/step - loss: 0.0114 - accuracy: 0.9976
```

```
y_pred = model.predict(X_test)
y_pred
```

```
35/35 [==============================] - 3s 78ms/step
array([[6.9499037e-16],
       [1.2465079e-03],
       [3.0413420e-19],
       ...,
       [5.8589937e-09],
       [2.2677074e-19],
       [2.5755884e-10]], dtype=float32)
```

```
y_pr = np.where(y_pred>0.5,1,0)
y_test
```

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

```
cm = confusion_matrix(y_test, y_pr)
score = accuracy_score(y_test, y_pr)
print(cm)
print('Accuracy Score Is:- ',score*100)
```

```
[[942   7]
 [ 20 146]]
Accuracy Score Is:-  97.57847533632287
```

```
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 = 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...")))
threshold = 0.5
y_pred = (y_pred > threshold).astype(int)
```

```
Enter New Review...hello fg hou hkl
[[0 0 0 ... 0 0 0]]
1/1 [==============================] - 0s 43ms/step
[[0.99604416]]
```

```
cm = confusion_matrix(y_test, y_pred)
score = accuracy_score(y_test, y_pred)
print(cm)
print('Accuracy Score Is:- ',score*100)
```

```
[[942   7]
 [ 20 146]]
Accuracy Score Is:-  97.57847533632287
```

```
model.save('spam.h5')
```