

THIRUVALLUVAR UNIVERSITY
PERIYAR ARTS COLLEGE
CUDDALORE – 607001.



DEPARTMENT OF COMPUTER APPLICATIONS

MACHINE LEARNING WITH PYTHON

Project Title : Optimizing Spam Filtering with Machine Learning

Team Id : NM2023TMID35498

Team Leader: BALAMURUGAN R(tv10520CA11)

Team member: SARA VANAN S(tv10520CA39)

Team member: SUBASH R(tv10520CA44)

Team member: SRI BALAJI M(tv10520CA43)

S.NO	CONTENT	PAGE NO
01	Introduction	02
02	Problem Definition and Design Thinking	06
03	Result	08
04	Advantages and Disadvantages	10
05	Applications	13
06	Conclusion	16
07	Future Scope	18

INTRODUCTION

In recent times, unwanted commercial bulk emails called spam has become a huge problem on the internet. The person sending the spam messages is referred to as the spammer. Such a person gathers email addresses from different websites, chatrooms, and viruses. Spam prevents the user from making full and good use of time, storage capacity and [network bandwidth](#). The huge volume of spam mails flowing through the computer networks have destructive effects on the memory space of email servers, [communication bandwidth](#), CPU power and user time . The menace of spam email is on the increase on yearly basis and is responsible for over 77% of the whole global email traffic. Users who receive spam emails that they did not request find it very irritating. It is also resulted to untold financial loss to many users who have fallen victim of internet scams and other fraudulent practices of spammers who send emails pretending to be from reputable companies with the intention to persuade individuals to disclose sensitive personal information like passwords, Bank Verification Number (BVN) and [credit card numbers](#).

OVERVIEW:

Text messages are essential these days; however, spam texts have contributed negatively to the success of this communication mode. The compromised authenticity of such messages has given rise to several security breaches. Using spam messages, malicious links have been sent to either harm the system or obtain information detrimental to the user. Spam SMS messages as well as emails have been used as media for attacks such as masquerading and smishing (a phishing attack through text messaging), and this has threatened both the user and service providers.

Therefore, given the waves of attacks, the need to identify and remove these spam messages is important. This dissertation explores the process of text classification from data input to embedded representation of the words in vector form and finally the classification process. Therefore, we have applied different embedding methods

to capture both the linguistic and semantic meanings of words. Static embedding methods that are used include Word to Vector (Word2Vec) and Global Vectors (Glove), while for dynamic embedding the transfer learning of the Bidirectional Encoder Representations from Transformers (BERT) was employed.

For classification, both machine learning and deep learning techniques were used to build an efficient and sensitive classification model with good accuracy and low false positive rate. Our result established that the combination of BERT for embedding and machine learning for classification produced better classification results than other combinations.

PURPOSE:

Machine learning algorithms have been extensively applied in the field of spam filtering. Substantial work has been done to improve the effectiveness of spam filters for classifying emails as either ham (valid messages) or spam (unwanted messages) by means of ML classifiers. They have the ability to recognise distinctive characteristics of the contents of emails. Many significant work have been done in the field of spam filtering using techniques that does not possess the ability to adapt to different conditions; and on problems that are exclusive to some fields e.g. identifying messages that are hidden inside a stage image. Most of the machine learning algorithms used for classification of tasks were designed to learn about inactive objective groups.

PROBLEM DEFINITION AND DESIGN THINKING

Empathy map:

Says

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

They think that
it's very harmful
for open
calls/messages

Spam filters
designed to
identify
spam and
block messages

To avoid
spam
interruption
and feel with
free



Thinks

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

They need a
spam free
phone
calls, messages
in their mobile
and laptops

They have
frustration, the
cause of spam
calls, messages
and e-mails

The number of spam
emails is rapidly
increasing in
marketing, chain
communications,
stock market, politics,
and education

The loss of an
important email
that accidentally
gets deleted
along with the
plethora of spam



OPTIMIZING SPAM FILTERING WITH MACHINE LEARNING

Does

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

In recent times,
spam filters
designed to
identify
spam and
block messages



The implication of
this is that one out
of a thousand
messages received
in sending their
email spam filter

Spam filtering is
about as
reducing to the
lowest of the lowest
the volume of
unsolicited
emails



Spam is a
problem in
marketing,
chain communications,
stock market, politics,
and education

Spam is a
problem in
marketing,
chain communications,
stock market, politics,
and education

Spam is a
problem in
marketing,
chain communications,
stock market, politics,
and education

Feels

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

Ideation & Brainstorming map:

Brainstorm

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

🕒 10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!



Person 1

Spam Filter Word
Spam filters use a list of words that are known to be associated with spam. These words are often used in the subject line or body of the email. The filter will look for these words and if they are found, the email is marked as spam.

Spam Filter Algorithm
Spam filters use a variety of algorithms to determine if an email is spam. Some filters use a simple rule-based system, while others use more complex machine learning algorithms. The filter will analyze the content of the email and make a decision based on the results.

Spam Filter Training
Spam filters are trained using a large dataset of emails that have been labeled as either spam or not spam. The filter will learn from this data and make adjustments to its algorithm as it receives more data.

Spam Filter Updates

Person 2

Spam Filter Word
Spam filters use a list of words that are known to be associated with spam. These words are often used in the subject line or body of the email. The filter will look for these words and if they are found, the email is marked as spam.

Spam Filter Algorithm
Spam filters use a variety of algorithms to determine if an email is spam. Some filters use a simple rule-based system, while others use more complex machine learning algorithms. The filter will analyze the content of the email and make a decision based on the results.

Spam Filter Training
Spam filters are trained using a large dataset of emails that have been labeled as either spam or not spam. The filter will learn from this data and make adjustments to its algorithm as it receives more data.

Spam Filter Updates
Spam filters are updated regularly to keep up with the latest trends in spam. This is done by adding new words to the list and removing old ones that are no longer relevant.

Person 3

Spam Filter Word
Spam filters use a list of words that are known to be associated with spam. These words are often used in the subject line or body of the email. The filter will look for these words and if they are found, the email is marked as spam.

Spam Filter Algorithm
Spam filters use a variety of algorithms to determine if an email is spam. Some filters use a simple rule-based system, while others use more complex machine learning algorithms. The filter will analyze the content of the email and make a decision based on the results.

Spam Filter Training
Spam filters are trained using a large dataset of emails that have been labeled as either spam or not spam. The filter will learn from this data and make adjustments to its algorithm as it receives more data.

Spam Filter Updates
Spam filters are updated regularly to keep up with the latest trends in spam. This is done by adding new words to the list and removing old ones that are no longer relevant.

Person 4

Spam Filter Word
Spam filters use a list of words that are known to be associated with spam. These words are often used in the subject line or body of the email. The filter will look for these words and if they are found, the email is marked as spam.

Spam Filter Algorithm
Spam filters use a variety of algorithms to determine if an email is spam. Some filters use a simple rule-based system, while others use more complex machine learning algorithms. The filter will analyze the content of the email and make a decision based on the results.

Spam Filter Training
Spam filters are trained using a large dataset of emails that have been labeled as either spam or not spam. The filter will learn from this data and make adjustments to its algorithm as it receives more data.

Spam Filter Updates
Spam filters are updated regularly to keep up with the latest trends in spam. This is done by adding new words to the list and removing old ones that are no longer relevant.

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.



RESULT

Spam Detector For Emails!!

Enter Your Message Here

Predict

Great! This is NOT a spam message.

Advantages and Disadvantages

Advantages:

Protection against Viruses

Spam emails are not just innocent marketing tools they can be carriers of dangerous computer viruses. Just one click on the wrong email can debilitate your network. Filters can provide a great firewall.

Keeping Hackers at Bay

In addition to dangerous viruses, hackers can also gain access to your system through a benign-looking email. A filter that blocks spam emails from reaching your inbox can save your important data.

Saving Time

Spam filtering can save time. Business employees do not have to go through numerous emails to decide which ones are spam, as sometimes that can be hard to decide. The time saved can be used to increase productivity.

Keeping your Reputation Intact

Spam filters can help keep a company maintains its reputation. They can block viruses from reaching consumers' data and prevent any spam mail from accidentally being forwarded to consumers.

Customized Services

Anti-spam software and programs can be tailored to your needs. You can create a blacklist of email addresses that often send you spam. A whitelist contains all the email addresses of your important associates.

Lets you Be Sure

Many anti-spam filters offer the service of keeping the spam emails saved for a few days. It allows you to make sure that no useful emails are being deleted together with the junk mail.

Disadvantages:

Most people naturally become nervous as they see they got an email that is unwanted and that is instantly labeled as being spam. If you are a heavy email user there is a pretty good possibility you actually get hundreds of spam emails per week, or even per day, based on activity.

The common approach is to select the unwanted messages and then delete them. However, this is not always something that you want to do. What happens if you mistakenly identified a message as being spam? If this is the case, it is possible you will end up with some serious problems. As an example, if the boss sends you an email and you do not even open it, you might get fired.

Why Not Use Spam Filters

The biggest disadvantage of using an email filter is that you may end up with messages being identified as being spam through a mistake of the algorithm that is used.

While missing out on important emails is a nuisance, we need to think about the fact that you can also miss the same emails if you receive a lot of spam. How can you see that message from the boss if there are hundreds of emails sent every single day? You can be highly attentive and still miss out on some emails.

Application

When reading spam email to obtain keywords to create application level filters, remember to keep your *Internet* connection turned off or firewall locked so that images in the email don't display and broadcast your address availability back to the spammers (fortunately some email applications like *Thunderbird* have this protection built-in). The following guidelines describe how to set up application-level spam filters:

- Mailbox. Create a special mailbox called "Junk-filter", or something similar, into which you will direct the filtered mail instead of deleting it. Then every once in awhile scan the junk mailbox to gauge the success of your filters and make sure no legitimate email is being trapped. You can also use this archive to help tune your filters as described under *Selection* below.
- Filters. Now you are ready to set up a set of filters to recognize common spam keywords and then transfer the email into the Junk-filter mailbox. First set up an initial set of filters based on the spam you have been getting most recently, and constructed to trap the most common spam keywords and phrases.
- Maintenance. After you set up your initial set of filters, monitor the spam you still receive and add a couple of rules each time you check your mail to continually increase the efficiency of your filter engine. Add a few rules each time to catch the most common examples still making it through the filter engine. Over time, the amount of spam that makes it through will become less and less, and will be of increasingly unusual nature (ex: weird subject lines) that they will be easily recognizable as spam by the human eye and easily deleted.
- Selection. There are two basic goals when drafting a spam filter: make the rule broad enough to be effective at catching spam, and make the rule specific enough to avoid trapping legitimate email by mistake. The following guidelines assist in creation of good spam filter rules:

- Tuning. You can occasionally go through your trash, which consists largely of spam you had to delete manually, sort the mailbox by subject, and look for common patterns and keywords. You can then create a few new filters to catch similar email, fine-tuning your engine to catch more of the spam that have been escaping your filter engine.
- Efficiency. If you wonder if a rule is worth creating, you can use the filter mailbox as a useful archive to test the rule. Search the spam mailbox for the filter condition you are considering. If none (or very few) of the spam you've so far received match the condition, the rule is probably ineffective and not worthwhile.
- Safety. Always use guard rules as described above to minimize the chance of trapping legitimate email. However, you still want to remain accessible to the world and new correspondents.
- Fields. With most email applications, filters can target the sender, subject, message body, and other fields. However, because most fields can be faked, the subject and message body are the best for spam filters

CONCLUSION

we reviewed machine learning approaches and their application to the field of spam filtering. A review of the state of the art algorithms been applied for classification of messages as either spam or ham is provided. The attempts made by different researchers to solving the problem of spam through the use of machine learning classifiers was discussed. The evolution of spam messages over the years to evade

filters was examined. The basic architecture of email spam filter and the processes involved in filtering spam emails were looked into. The paper surveyed some of the publicly available datasets and performance metrics that can be used to measure the effectiveness of any spam filter. The challenges of the machine learning algorithms in efficiently handling the menace of spam was pointed out and comparative studies of the machine learning technics available in literature was done. We also revealed some open research problems associated with spam filters. In general, the figure and volume of literature we reviewed shows that significant progress have been made and will still be made in this field. Having discussed the open problems in spam filtering, further research to enhance the effectiveness of spam filters need to be done. This will make the development of spam filters to continue to be an active research field for academicians and industry practitioners researching machine learning techniques for effective spam filtering. Our hope is that research students will use this paper as a spring board for doing qualitative research in spam filtering using machine learning, deep learning and deep adversarial learning algorithms.

FUTURE SCOPE

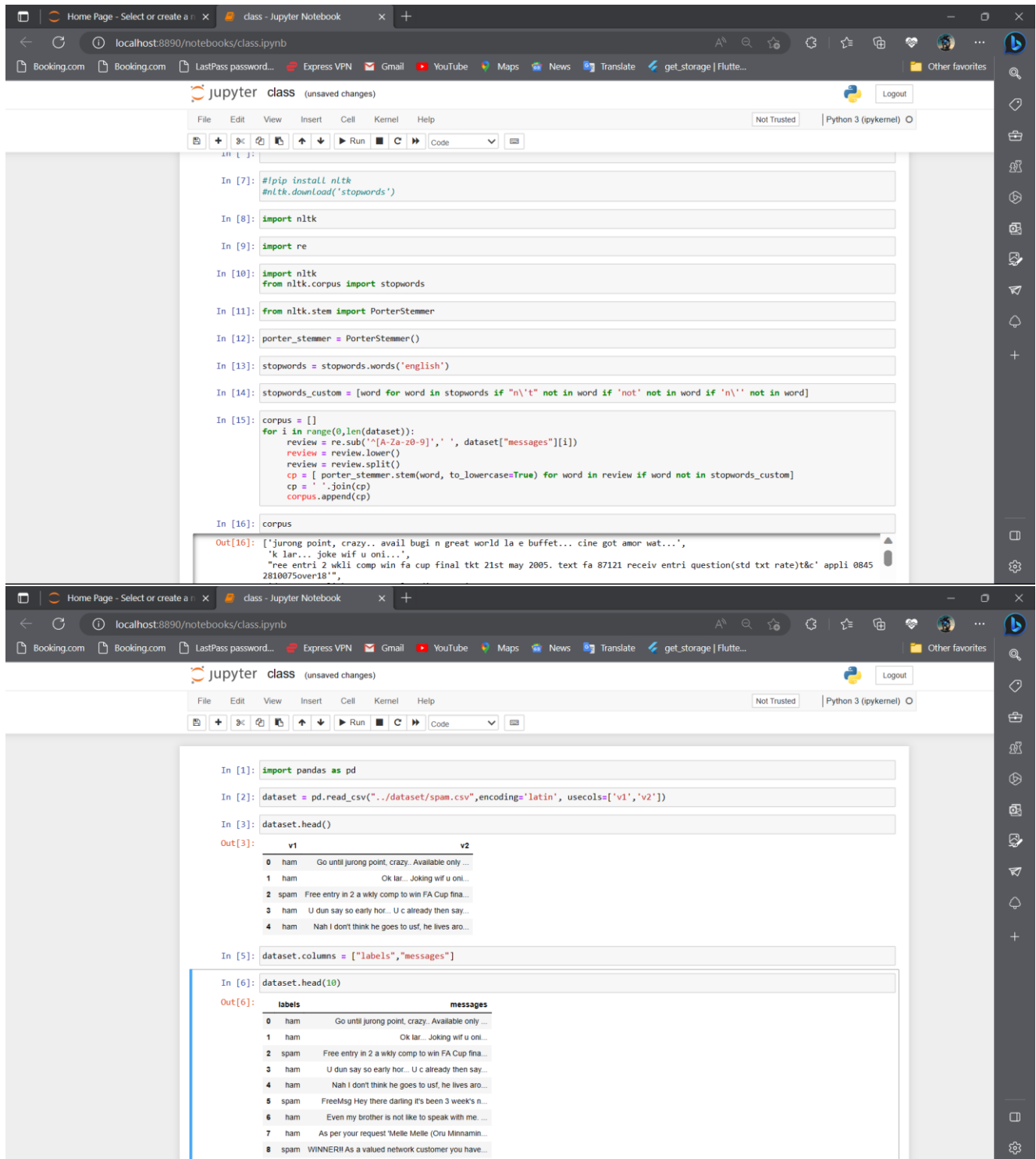
Efficient pattern detection in spam mail filtering plays crucial role. Using RFD model spam detection gives the spam patterns, non –spam patterns and general patterns which easily identify the whether the mail is spam or ham. The current method which uses the pattern detection method does not include the general patterns. RFD gives the general patterns of which user can decide to determine whether he wants to put the mail as spam or non-spam to avoid the loss of important mails. The images which are in forms of spams are also detected using File Properties, Histogram and Hough Transform. The current proposed system is for English language mails but as future scope we can design the system for multiple languages.

Hence there is scope for complete automation of spam detection systems with maximum efficiency. With grow- ing popularity of online stores, the competition also increases.

Appendix

Source code

class.ipynb



The image shows two screenshots of a Jupyter Notebook interface. The top screenshot displays the first code cell, which includes the following code:

```
In [7]: #!pip install nltk
#!nlk.download('stopwords')

In [8]: import nltk

In [9]: import re

In [10]: import nltk
from nltk.corpus import stopwords

In [11]: from nltk.stem import PorterStemmer

In [12]: porter_stemmer = PorterStemmer()

In [13]: stopwords = stopwords.words('english')

In [14]: stopwords_custom = [word for word in stopwords if 'n' not in word and 't' not in word and 'l' not in word and 's' not in word]

In [15]: corpus = []
for i in range(0, len(dataset)):
    review = re.sub('[A-Za-z0-9]', ' ', dataset["messages"][i])
    review = review.lower()
    review = review.split()
    cp = [porter_stemmer.stem(word, to_lowercase=True) for word in review if word not in stopwords_custom]
    cp = ' '.join(cp)
    corpus.append(cp)

In [16]: corpus
```

The output of the last cell is:

```
Out[16]: ['jurong point, crazy.. avail bugi n great world la e buffet... cine got amor wat...',
'k lar... joke wif u oni...',
'ree entri 2 wkli comp win fa cup final tkt 21st may 2005. text fa 87121 receiv entri question(std txt rate)t&c' appli 0845
2810075over18']
```

The bottom screenshot displays the second code cell, which includes the following code:

```
In [1]: import pandas as pd

In [2]: dataset = pd.read_csv("../dataset/spam.csv", encoding='latin', usecols=['v1', 'v2'])

In [3]: dataset.head()

Out[3]:
```

	v1	v2
0	ham	Go until jurong point, crazy.. Available only...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I dont think he goes to usf, he lives aro...

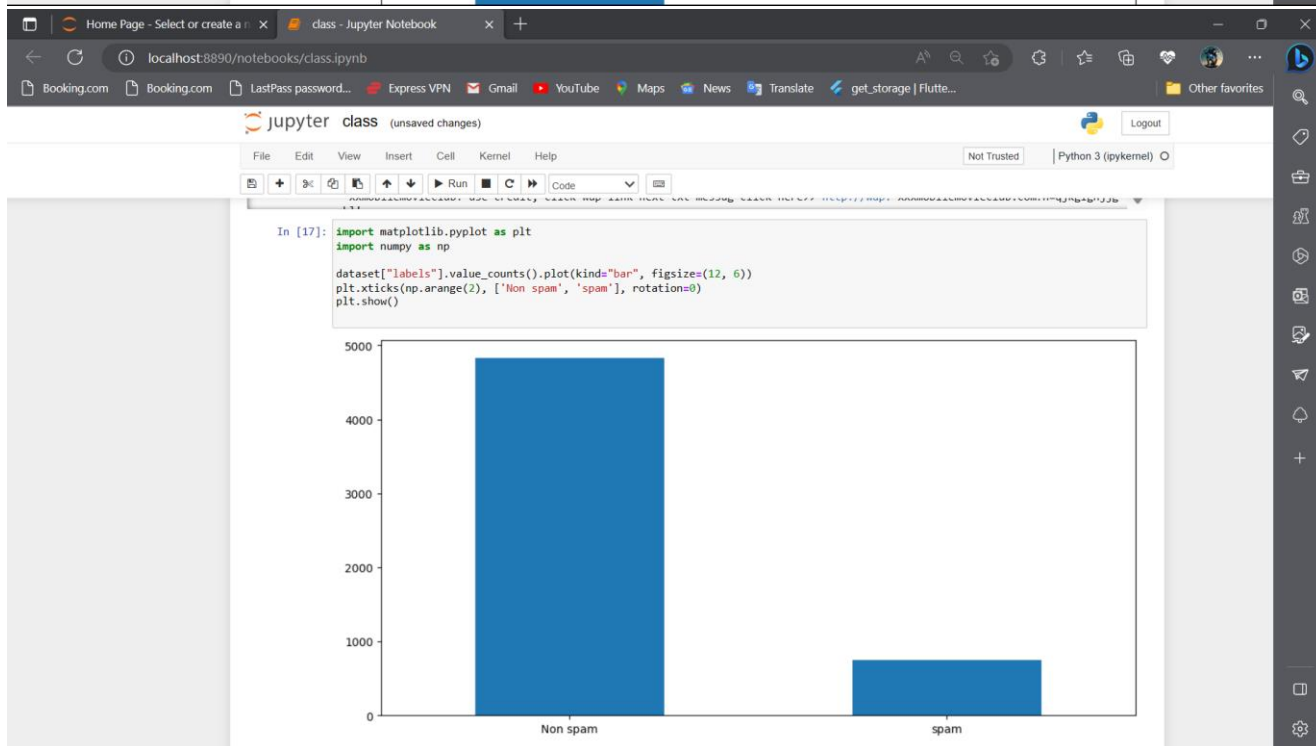
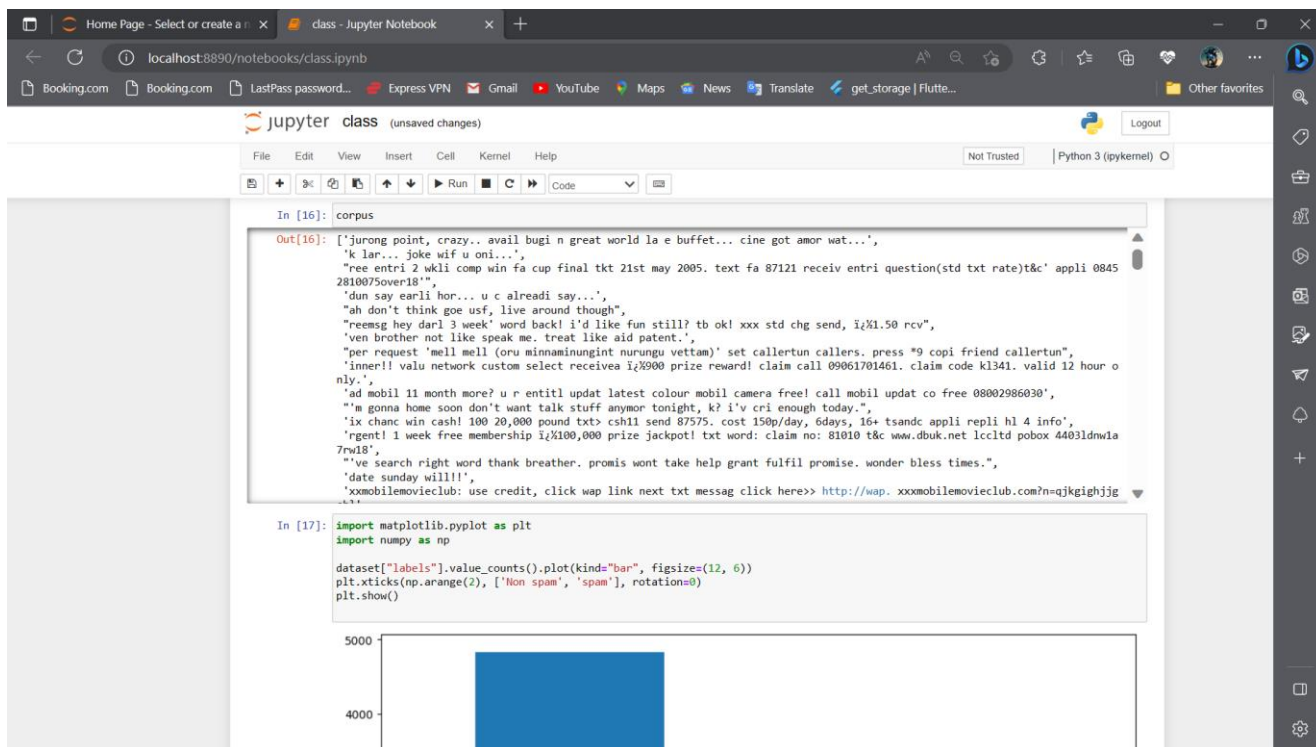
The output of the last cell is:

```
In [5]: dataset.columns = ["labels", "messages"]

In [6]: dataset.head(10)

Out[6]:
```

	labels	messages
0	ham	Go until jurong point, crazy.. Available only...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I dont think he goes to usf, he lives aro...
5	spam	FreeMsg Hey there darling it's been 3 week's n...
6	ham	Even my brother is not like to speak with me. ...
7	ham	As per your request 'Melle Melle (Oru Minnamin...
8	spam	WINNER!! As a valued network customer you have...
9	spam	Urgent! M... 10 months to go now!! U.D. Bu...



The screenshot shows a Jupyter Notebook titled 'class - Jupyter Notebook' running on a local host. The notebook has a menu bar with File, Edit, View, Insert, Cell, Kernel, and Help. Below the menu bar is a toolbar with icons for file operations, running cells, and other functions. The notebook content shows the following code and output:

```
In [18]: y = pd.get_dummies(dataset["labels"], drop_first=True)

In [19]: y
```

Out[19]:

	spam
0	0
1	0
2	1
3	0
4	0
...	...
5567	1
5568	0
5569	0
5570	0
5571	0

5572 rows x 1 columns

```
In [20]: #split to train test
from sklearn.model_selection import train_test_split

In [21]: X_train, X_test, y_train, y_test = train_test_split(corpus, y, test_size=0.2)

In [22]: X_train
```

Out[22]:

```
['reat. hope use connect mode men also co never know old friend lead today',
'ol yes. friendship hang thread caus u won't buy stuff.',
'm come back thursday. yay. gonna ok get money. cheers. oh yeah you. everyth alright. how school. call work",
'ake small dose tablet fever",
'maolnic 1',
```

The screenshot shows the same Jupyter Notebook interface, now showing the vectorization of the training data. The notebook content shows the following code and output:

```
In [23]: y_train
```

Out[23]:

	spam
4269	0
307	0
5308	0
1470	0
1670	0
...	...
1199	0
5058	1
9231	0
1256	0
4412	0

4457 rows x 1 columns

```
In [24]: from sklearn.feature_extraction.text import CountVectorizer

In [25]: cv = CountVectorizer()
X_train = cv.fit_transform(X_train).toarray()

In [26]: X_train
```

Out[26]:

```
array([[0, 0, 0, ..., 0, 0, 0],
[0, 0, 0, ..., 0, 0, 0],
[0, 0, 0, ..., 0, 0, 0],
...,
[0, 0, 0, ..., 0, 0, 0],
[0, 0, 0, ..., 0, 0, 0],
[0, 0, 0, ..., 0, 0, 0]], dtype=int64)
```

```
In [26]: X_train
Out[26]: array([[0, 0, 0, ..., 0, 0, 0],
               [0, 0, 0, ..., 0, 0, 0],
               [0, 0, 0, ..., 0, 0, 0],
               ...,
               [0, 0, 0, ..., 0, 0, 0],
               [0, 0, 0, ..., 0, 0, 0],
               [0, 0, 0, ..., 0, 0, 0]], dtype=int64)

In [27]: X_test = cv.transform(X_test).toarray()

In [28]: X_test.shape
Out[28]: (1115, 7712)

In [29]: X_train.shape
Out[29]: (4457, 7712)

In [30]: y_train.shape
Out[30]: (4457, 1)

In [31]: y_test.shape
Out[31]: (1115, 1)

In [32]: from sklearn.ensemble import RandomForestClassifier

In [33]: rfc = RandomForestClassifier()

In [34]: y_pre = rfc.fit(X_train, y_train)

C:\Users\Murugavel Dev\AppData\Local\Temp\ipykernel_14004\3720719451.py:1: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
```

```
In [35]: pred = rfc.predict(X_test)

In [36]: from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

In [37]: accuracy_score(y_test, pred)
Out[37]: 0.9766816143497757

In [38]: confusion_matrix(y_test, pred)
Out[38]: array([[966,  0],
               [ 26, 123]], dtype=int64)

In [39]: print(classification_report(y_test, pred))
              precision    recall  f1-score   support

     0       0.97       1.00       0.99       966
     1       1.00       0.83       0.90       149

 accuracy      0.99      0.91      0.95      1115
 macro avg     0.99      0.91      0.95      1115
 weighted avg   0.98      0.98      0.98      1115

In [40]: import os
os.chdir(os.path.dirname(os.path.dirname(os.getcwd())))

In [41]: import sys
sys.path.insert(0, '../SMS Spam Classifier/entity')

In [42]: from entity.sms_spam_classifier import SMSSPAMClassifier

In [43]: sms_spam = SMSSPAMClassifier(cv=cv, rfc=rfc, sp_word=stopwords_custom)
```

```

0      0.97      1.00      0.99      966
1      1.00      0.83      0.90      149

accuracy      0.99      0.91      0.98      1115
macro avg      0.99      0.91      0.95      1115
weighted avg   0.98      0.98      0.98      1115

In [40]: import os
os.chdir(os.path.dirname(os.path.dirname(os.getcwd())))

In [41]: import sys
sys.path.insert(0, '../SMSSpamClassifier/entity/')

In [42]: from entity.sms_spam_classifier import SMSSpamClassifier

In [43]: sms_spam = SMSSpamClassifier(cv=cv, rfc=rfc, sp_word=stopwords_custom)

In [44]: value = sms_spam.predict("WINNER!! As a valued network customer you have been selected to receive a 900 prize reward! To claim c
[inner!! valu network custom select receivea 900 prize reward! claim call 09061701461. claim code kl341. valid 12 hour']

In [45]: value
Out[45]: array([1], dtype=uint8)

In [46]: import pickle

In [47]: with open('smsspamclassifier.pkl', 'wb') as pickle_output:
pickle.dump(sms_spam, pickle_output)

In [ ]:

```

jupyter class (unsaved changes) Logout

File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

```

In [21]: X_train, X_test, y_train, y_test = train_test_split(corpus, y, test_size=0.2)

In [22]: X_train
Out[22]: ['reat. hope use connect mode men also co never know old friend lead today',
'ol yes. friendship hang thread caus u won't buy stuff.',
'm come back thursday. yay. gonna ok get money. cheers. oh yeah you. everyth alright. how school. call work",
'ake small dose tablet fever',
'maolnic 1',
'queeeeezel! christma hug.. u lik frndshp den hug back.. u get 3 u r cute:) 6 u r luvd:* 9 u r lucky;) none? peopl hate
u:',
'f don't, prize go anoth customer. t&c www.t-c.biz 18+ 150p/min polo ltd suit 373 london wij 6hl pleas call back busi",
'r u doing?',
'hey r give second chanc rahul dengra.',
'uncl atlanta. wish guy great semester.',
'nt worry...us ice piec cloth pack.also take 2 tablets.',
'eah not, gang read!',
'i2Xi2X dun wan watch infern affair?',
'let math. not good it.',
'ree game. get rayman golf 4 free o2 game arcade. 1st get ur game settings. repli post, save & activ8. press 0 key arcade.
termsappli',
'ongrats! 2 mobil 3g videophon r yours. call 09061744553 now! videochat wid ur mates, play java games, dload polyh music, n
olin rentl. bx420. ip4. 5we. 150pm',
...
In [23]: y_train
Out[23]:
   spam
4269   0
307    0
5308   0
1470   0
1670   0
...
1199   0
5058   1

```


About.html

```
about.html - Notepad
File Edit View

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Find Spam message</title>
  <link rel="stylesheet" href="static/css/about.css">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-18mE4kbWq78iYhF1dvKuhFTA6auU8tT94WrHftjDbrCEXS"
</head>
<body>
  <div class="container-fluid">
    <!--SET NAV BAR-->
    <nav class="navbar navbar-expand-sm navbar-light bg-dark text-white fw-bold">
      <div class="container">
        <a class="navbar-brand text-light" href="{url_for('home')}}">
          
          SPAM FILTER
        </a>
        <button
          class="navbar-toggler bg-light"
          type="button"
          data-bs-toggle="collapse"
          data-bs-target="#navbarNav"
          aria-controls="navbarNav"
          aria-expanded="false"
          aria-label="Toggle navigation"
        >
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
          <ul class="navbar-nav">
            <li class="nav-item">
              <a
                class="nav-link active text-decoration-none text-white"
                href="#"
              >
                DEV</a>
            </li>
            <li class="nav-item">
              <a
                class="nav-link text-decoration-none text-white"
                href="{url_for('home')}}"
              >
                Home</a>
            </li>
          </ul>
        </div>
      </div>
    </nav>
  </div>
</body>
</html>
```

```
about.html - Notepad
File Edit View

  <div class="card-body">
    <p> Spam messages are unsolicited messages sent in bulk, typically for commercial purposes, and can include emails, text messages, and social media messages. </p>
    <p> Spam messages are usually generated automatically by spamming software, and they can be difficult to filter or block. However, many email services and mail providers offer spam filters to help protect your inbox. </p>
  </div>
</div>
<div class="hr bg-white" />
<!--CARD TWO-->
<div class="container" style="margin-top:5%; margin-bottom:10%">
  <div class="row">
    <div class="col-md-6">
      <div class="card">
        <div class="card-header text-white">SPAM MESSAGE TIPS</div>
        <div class="card-body">
          <ul class="list">
            <li>Be cautious about sharing your personal information online, especially on websites or platforms that you are not familiar with.</li>
            <li>Use spam filters and antivirus software on your computer and mobile devices.</li>
            <li>Don't respond to or click on any links in spam messages, as they could be malicious or lead to phishing scams.</li>
            <li>Block or report any suspicious messages or accounts that send you spam.</li>
            <li>Avoid opening attachments or downloading files from unknown sources.</li>
          </ul>
        </div>
      </div>
    </div>
    <div class="col-md-6">
      <div class="card">
        <div class="card-body">
          
        </div>
      </div>
    </div>
  </div>
</div>
</div>
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js" integrity="sha384-7zCNj/Iq95wo16oMtf5kbZ9ccEh31e0z1HGyDuCQ6wgnyJNSydrPa03rtR1zdB"
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js" integrity="sha384-QJHtvGhmr9XOJPi66hYYVuuG+200K9T+2nN4kZFN1RtK3zEFEIsxhlmwL1/YESvPz13" cro
</body>
</html>
```

Home.html

```
home.html - Notepad
File Edit View

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Find Spam message</title>
  <link rel="stylesheet" href="static/css/main.css">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-18mE4kbWq78iYhF1dvKuhFTA06auU8tT94WrHftjDbrCEXS"
</head>
<body style="background-image:url('static/images/home.jpg');">
  <div class="container-fluid">
    <!--SET NAV BAR-->
    <nav class="navbar navbar-expand-xl navbar-light bg-dark text-warning fw-bolder">
      <div class="container">
        <a class="navbar-brand text-light" href="{{url_for('home')}}">
          
          SPAM FILTER
        </a>
        <button
          class="navbar-toggler bg-light"
          type="button"
          data-bs-toggle="collapse"
          data-bs-target="#navbarNav"
          aria-controls="navbarNav"
          aria-expanded="false"
          aria-label="Toggle navigation"
        >
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
          <ul class="navbar nav">
            <li class="nav-item">
              <a
                class="nav-link text-decoration-none text-reset"
                href="{{url_for('home')}}">
                Home</a>
            </li>
            <li class="nav-item">
              <a
                class="nav-link text-decoration-none text-reset"
                href="{{url_for('index')}}">
                Spam</a>
            </li>
          </ul>
        </div>
      </div>
    </nav>
  </div>
  <div class="content">
    <div class="text">
      <p><strong> SMS <strong> <strong class="text-danger"> SPAM </strong>DETECTION </strong></p>
    </div>
    <div class="text">
      <a href="{{url_for('index')}}" class="btn btn-primary btn-lg active" role="button" aria-pressed="true">CLICK HERE ></a>
    </div>
    <div class="text-left pg">
      <ul class="list-unstyled fw-light">
        <li>SMS spam detection is an important tool for protecting mobile</li>
        <li>phone users from unwanted and potentially harmful messages, </li>
        <li>and it relies on a combination of sophisticated algorithms and </li>
        <li>techniques to effectively identify and filter out these messages.</li>
      </ul>
    </div>
  </div>
</div>
</div>
</div>
</div>
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js" integrity="sha384-7+zCnJ/Iq95wo16oMtfkskbZ9ccEh31e0z1HGyDuCQ6wgnyJNSydrPa03rtR1zdB"
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js" integrity="sha384-QJHtvGhmr9X0PiI6YVutG+200K9T+2nN4kZFN1RtK3zEFEIsxhlmwL1/YESvpZ13" cro
</body>
</html>
```

```
home.html - Notepad
File Edit View

      </li>
      <li class="nav-item">
        <a
          class="nav-link text-decoration-none text-reset"
          href="{{url_for('index')}}">
          Spam</a>
      </li>
      <li class="nav-item">
        <a
          class="nav-link text-decoration-none text-reset"
          href="{{url_for('about')}}">
          About</a>
      </li>
    </ul>
  </div>
</div>
</nav>
<div>
</div>
<div class="content">
  <div class="text">
    <p><strong> SMS <strong> <strong class="text-danger"> SPAM </strong>DETECTION </strong></p>
  </div>
  <div class="text">
    <a href="{{url_for('index')}}" class="btn btn-primary btn-lg active" role="button" aria-pressed="true">CLICK HERE ></a>
  </div>
  <div class="text-left pg">
    <ul class="list-unstyled fw-light">
      <li>SMS spam detection is an important tool for protecting mobile</li>
      <li>phone users from unwanted and potentially harmful messages, </li>
      <li>and it relies on a combination of sophisticated algorithms and </li>
      <li>techniques to effectively identify and filter out these messages.</li>
    </ul>
  </div>
</div>
</div>
</div>
</div>
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js" integrity="sha384-7+zCnJ/Iq95wo16oMtfkskbZ9ccEh31e0z1HGyDuCQ6wgnyJNSydrPa03rtR1zdB"
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js" integrity="sha384-QJHtvGhmr9X0PiI6YVutG+200K9T+2nN4kZFN1RtK3zEFEIsxhlmwL1/YESvpZ13" cro
</body>
</html>
```

Index.html

```
index.html - Notepad
File Edit View

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Find Spam message</title>
  <link rel="stylesheet" href="static/css/style.css">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-18mE4kWBq78iYhF1dVkuHFTAU6au08tT94wrHftjDbrCEXS" />
</head>
<body>
  <div class="container">
    <!--SET NAV BAR-->
    <nav class="navbar navbar-expand-xl navbar-light bg-white text-dark fw-bold">
      <div class="container">
        <a class="navbar-brand text-dark" href="{url_for('home')}">
          
          SPAM FILTER
        </a>
        <button
          class="navbar-toggler"
          type="button"
          data-bs-toggle="collapse"
          data-bs-target="#navbarNav"
          aria-controls="navbarNav"
          aria-expanded="false"
          aria-label="Toggle navigation"
        >
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
          <ul class="navbar-nav">
            <li class="nav-item">
              <a
                class="nav-link active text-decoration-none text-reset"
                href="#"
              >
                DEV</a>
            </li>
            <li class="nav-item">
              <a
                class="nav-link text-decoration-none text-reset"
                href="{url_for('home')}"
              >
                Home</a>
            </li>
          </ul>
        </div>
      </div>
    </nav>
  </div>
</body>
</html>
```

```
index.html - Notepad
File Edit View

    <a
      class="nav-link text-decoration-none text-reset"
      href="{url_for('index')}"
    >
      Spam</a>
    </li>
    <li class="nav-item">
      <a
        class="nav-link text-decoration-none text-reset"
        href="{url_for('about')}"
      >
        About</a>
      </li>
    </ul>
  </div>
</div>
</nav>

<!--SET OUTPUT CARD-->
<div class="card container bg-white text-dark" style="height: 25rem;">
  <div class="card-body">
    <h5 class="card-title">ENTER MESSAGE</h5>
    <form method="post" action="{url_for('index')}">
      <div class="mb-3">
        <label for="exampleInputEmail" class="form-label fw-bold">Enter message to classify ham or spam</label>
        <input type="text" class="form-control" name="review_msg" id="id_message" placeholder="Enter your message">
      </div>
      <div class="d-flex align-items-center">
        <button type="submit" class="btn btn-primary">Check the Message</button>
      </div>
    </form>
    <div class="fs-6 fw-bold text-center">
      The message you have entered is {{message_status}}
    </div>
  </div>
</div>
</div>
</div>

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js" integrity="sha384-7+zCNj/IqJ95wo16oMtfsKbZ9ccEh31e0z1HGyDuCQ6wgnyJNSYdrPa03rtR1zdB" />
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js" integrity="sha384-QJHtvGhmr9XOIdI6YUutG+2Q0K9t+2nN4kzFNIrK3zEFIEIshxh1mW15/YESvpZ13" />
</script>
</body>
</html>
```

App.py

```
File Edit Selection View Go Run Terminal Help • app.py - Visual Studio Code
C:\Users\HP\Desktop> All Projects > COMPLETED > Project 6 > Flask > app.py > index

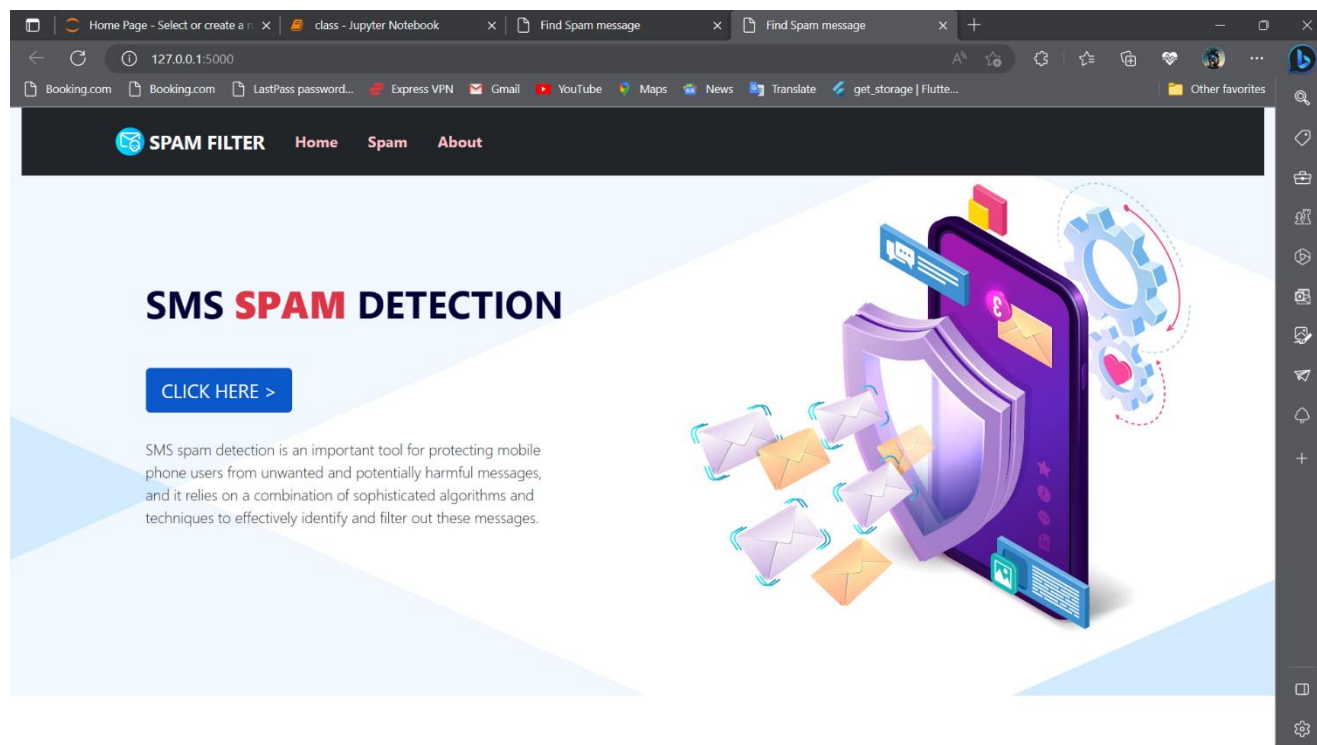
25 def about():
26     return render_template("about.html")
27 @app.route("/spam", methods=['GET','POST'])
28 def index():
29     message_status = None
30     if request.method == "POST":
31         message_status = "Ham"
32
33
34     status = flask.model.predict(request.form.get('review_msg'))
35
36     if status[0] == 1:
37         message_status = "Spam"
38
39     #return status
40
41     return render_template('index.html',message_status=message_status)
42
43
44
45 if __name__ == "__main__":
46     app.run(debug=True)
47
```

```
File Edit Selection View Go Run Terminal Help • app.py - Visual Studio Code
C:\Users\HP\Desktop> All Projects > COMPLETED > Project 6 > Flask > app.py > index

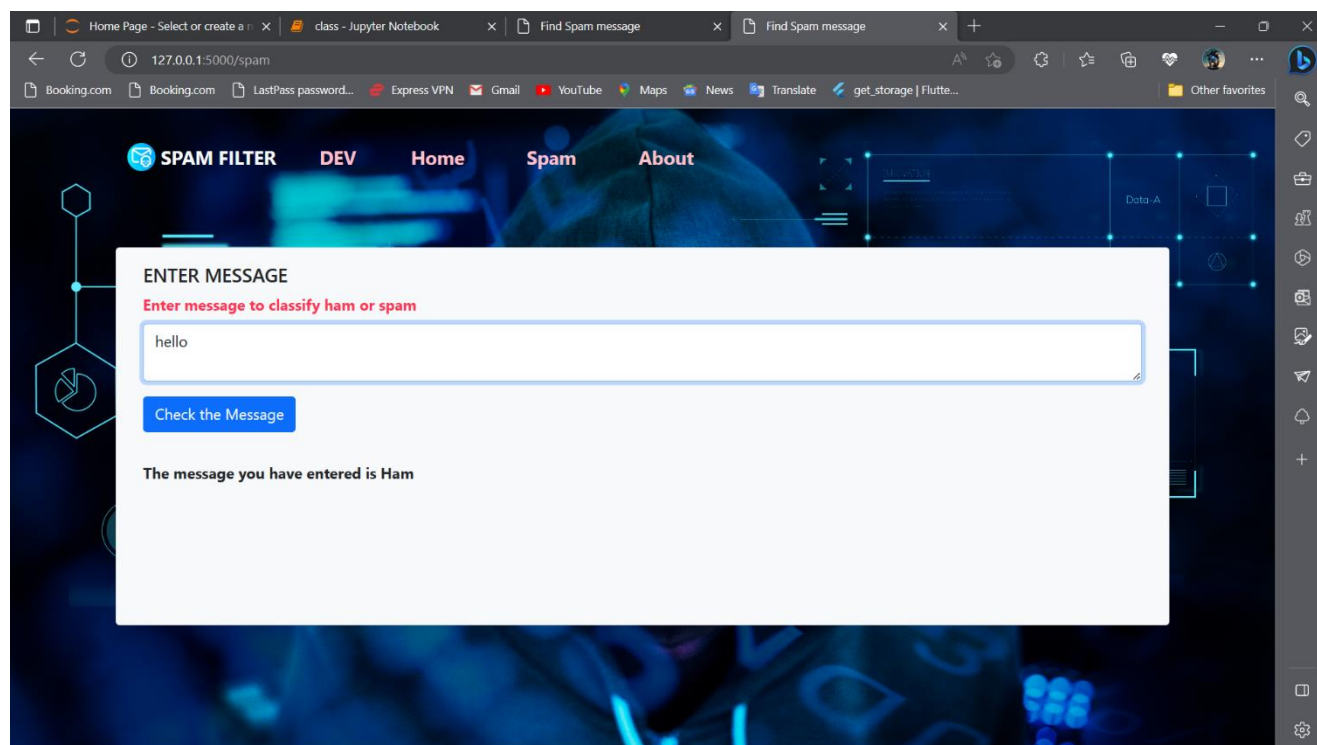
1 #from crypt import methods
2 from os import stat
3 import flask
4 from flask import Flask, current_app, render_template, request,session
5 import pickle
6 from entity.sms_spam_classifier import SMSSPAMClassifier
7 import sys
8 import nltk
9 sys.path.insert(0, '../SMSSpamClassifier/entity/')
10 nltk.download('stopwords')
11 app = Flask(__name__)
12 app.secret_key = "mlmodel"
13 def load_classifier() -> SMSSPAMClassifier:
14     sms_classifier = None
15     with open('smsspamclassifier.pkl','rb') as picker_reader:
16         sms_classifier = pickle.load(picker_reader)
17     return sms_classifier
18 ctx = app.app_context()
19 flask.model = load_classifier()
20 ctx.push()
21 @app.route("/")
22 def home():
23     return render_template("home.html")
24 @app.route("/about")
25 def about():
26     return render_template("about.html")
27 @app.route("/spam", methods=['GET','POST'])
28 def index():
29     message_status = None
30     if request.method == "POST":
31         message_status = "Ham"
```

Result and outputs :

Home page



Predict page



About page

