OPTIMIZING SPAM FILTERING IN MACHINE LEARNING

predicting based experiential learning program project

A project submitted by leader:

K.SUMITHAR

Team members:

K.SUMITHRA

S.KANAGALAKSHMI

R.SUVETHA

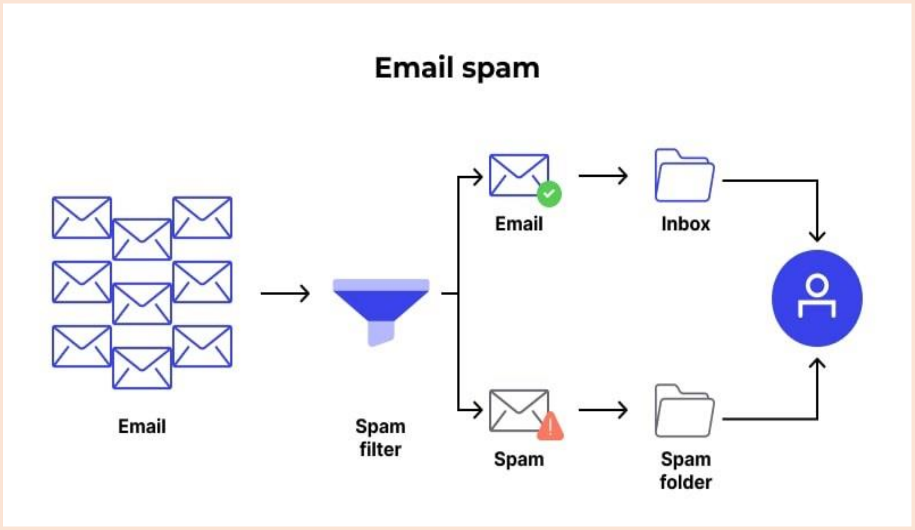
INTRUDUCTION

Overview:

Over recent years, as the popularity of mobile phone device 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 un solicited commercial advertisement (spams)being sent to mobile phones.

Due to spam SMS, mobile service providers suffer from some sort of financial problem as well as it reduces calling time for users. Unfortunately , if the user access 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 communication world and it growth 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.

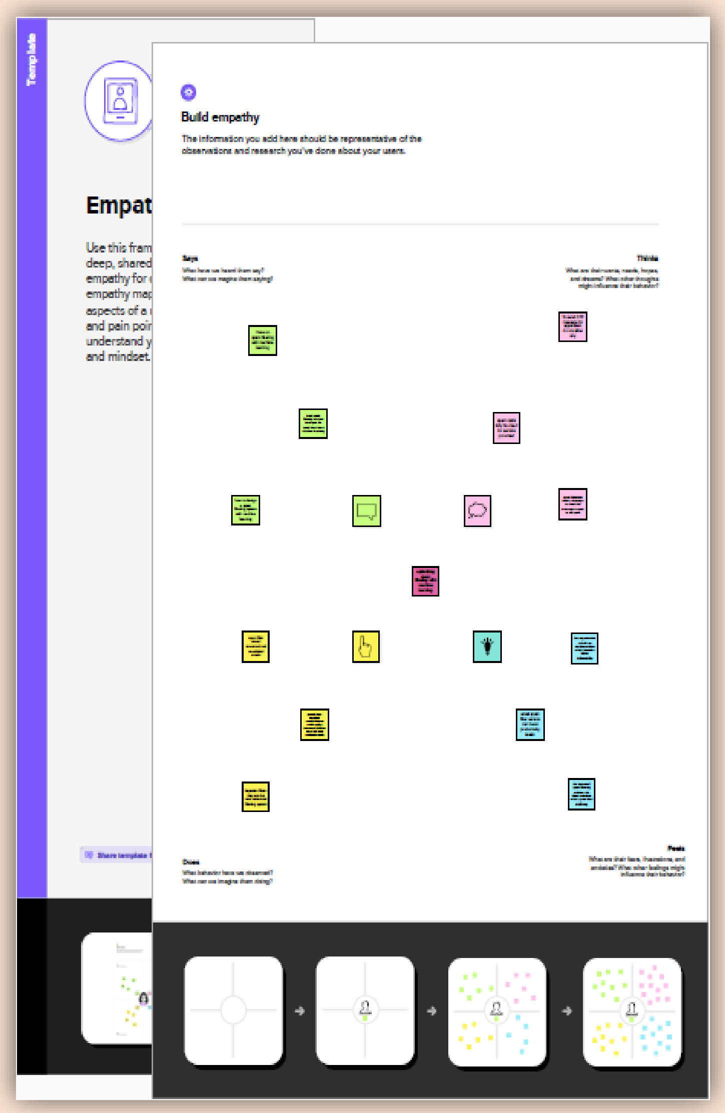


PURPOSE

- ❖ spam filters “heuristics” methods, which means that each email message is subjected to thousands of predefined rules (algorithm). Each rule assigns a numerical score to the probability of the message being spam, and if the score passes a certain threshold the email is flagged as spam and blocked from going further

- A spam filtering solution cannot be 100 percent effective. However, a business email system without spam filtering is highly vulnerable, if not unusable. It is important to stop as much spam as you can, to protect your network from the many possible risks: viruses, phishing attacks, compromised web links and other malicious content.
- Spam filters also protect your server form being overloaded with non-essential emails, and the worse problem of being infected with spam software that may turn them into spam servers themselves.
- By preventing spam emails from reaching your employees' mailboxes, spam filters give an additional layer of protection to your users, your network, and your business
- The spam step to maximizing the success of your bulk email is learning what a spam folder is and how it works. Rather then fearing spam filter and making them your enemy, you need to make them a tool to be used to your advantage.
- Rather , each mailbox, provider, company, and individual user can have their own configurable devices, software, algorithm and machine learning technology that will act as their spam filter.

EMPATHY MAP



Build empathy

The information you add here should be representative of the observations and research you've done about your users.

Says

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

Thinks

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

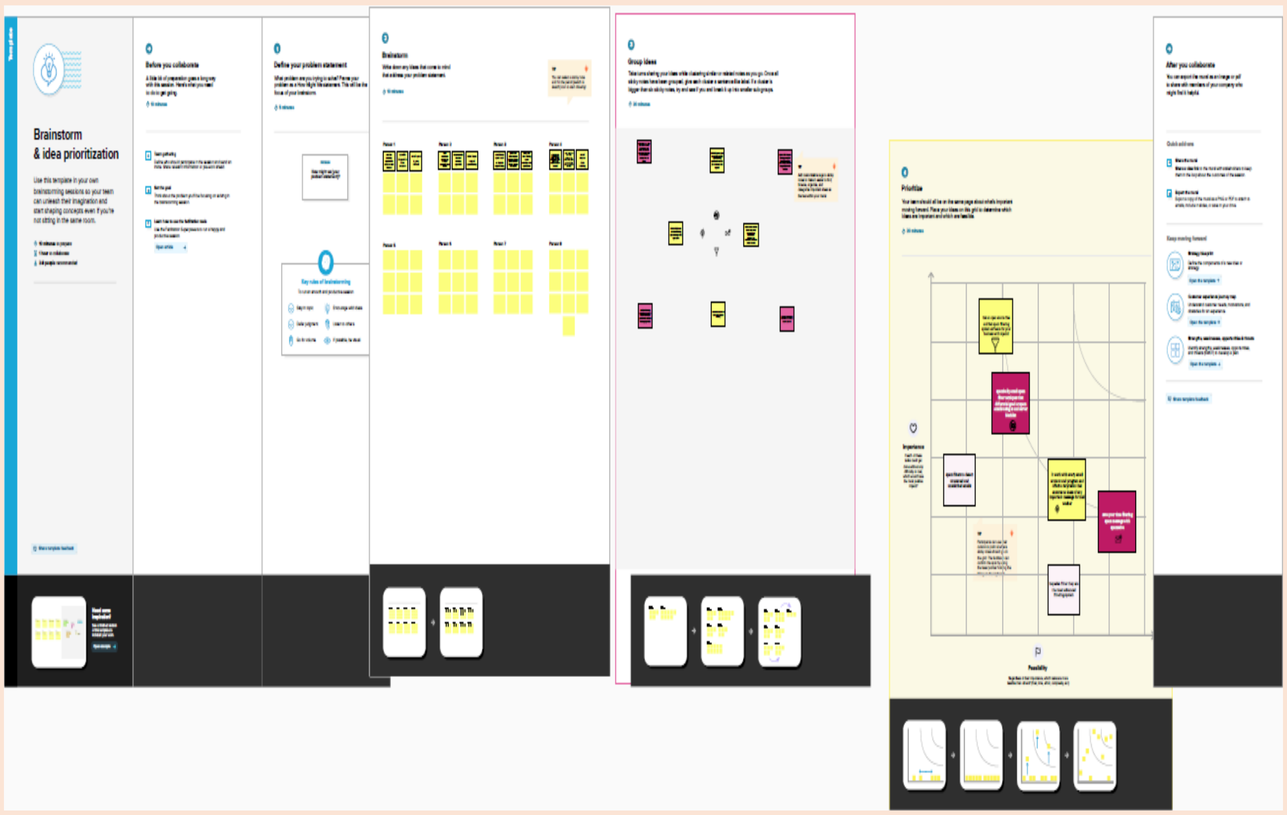
Does

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

Feels

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

BRAINSTROM



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.

Before you collaborate

Define your problem statement

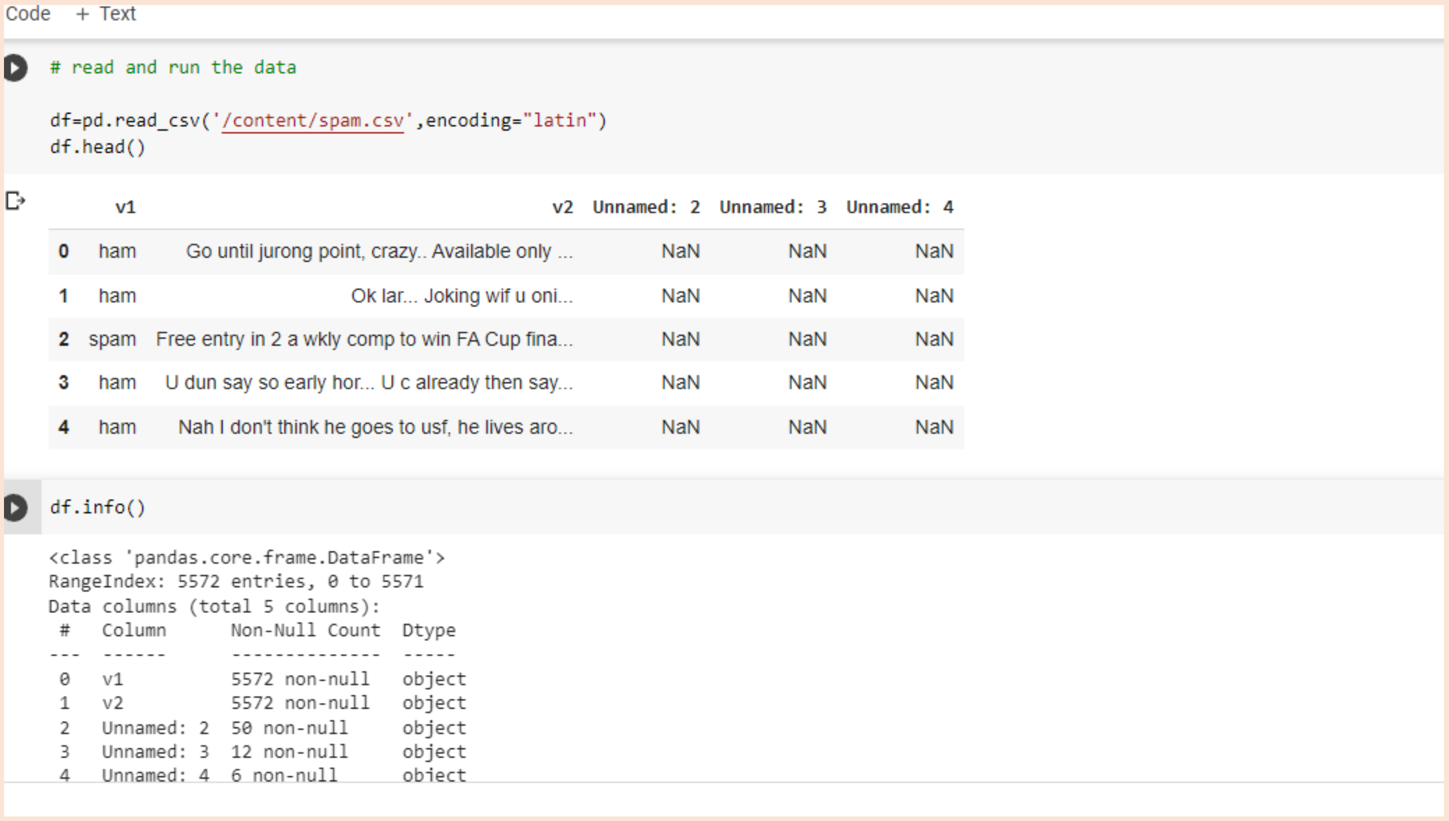
Brainstorm

Group ideas

Prioritize

After you collaborate

RESULT



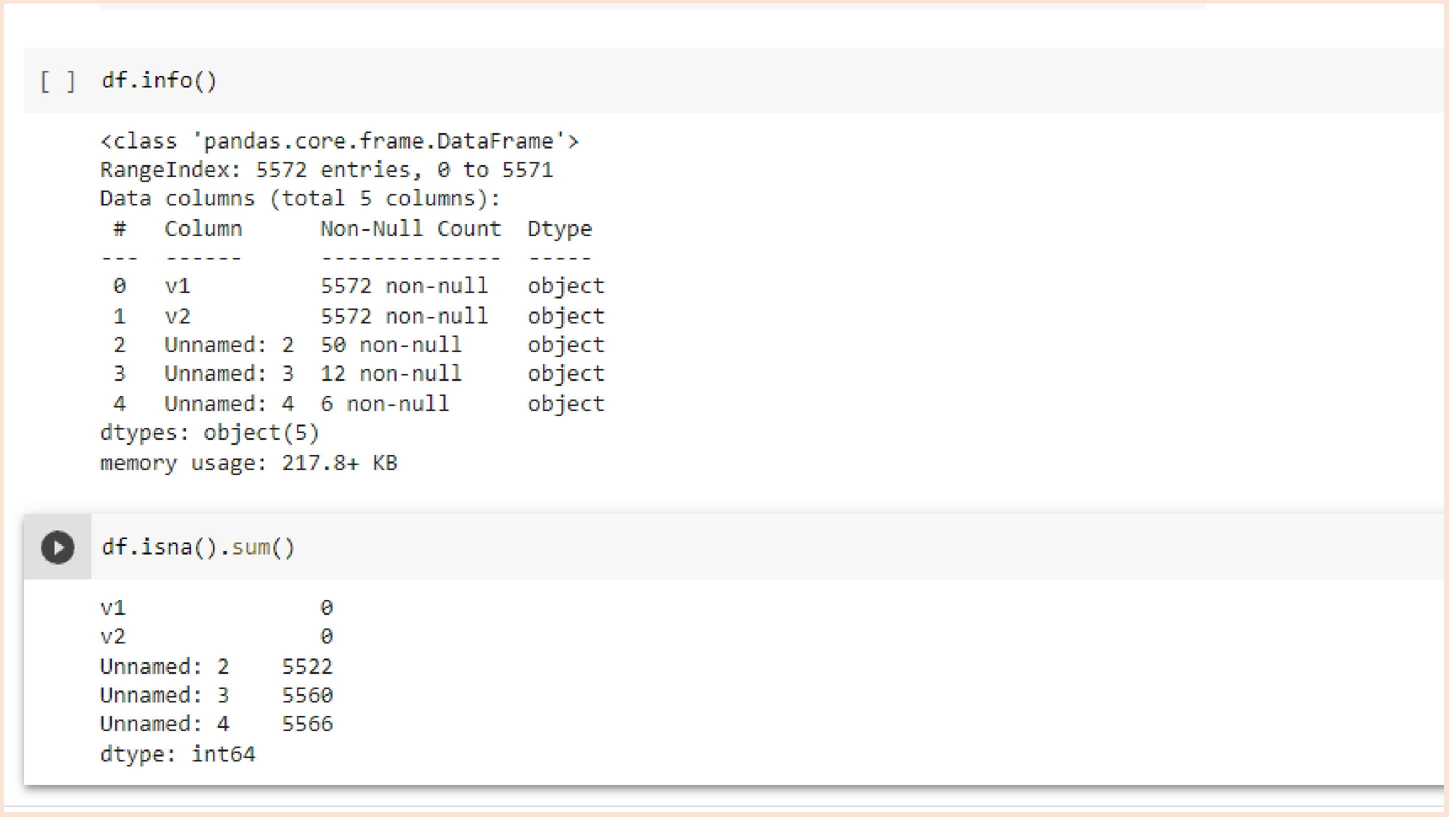
Code + Text

```
# read and run the data

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()
```

```
<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
```

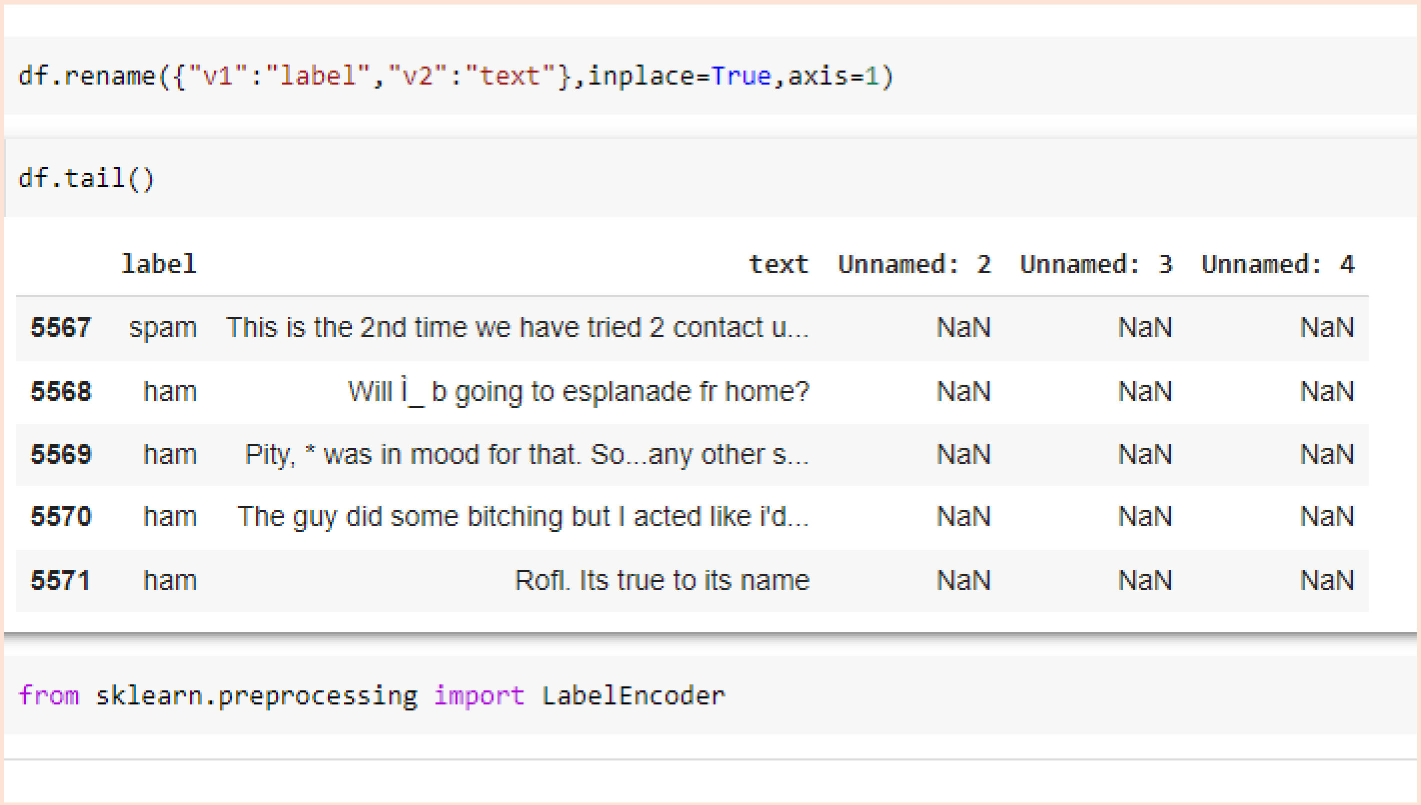


```
[ ] df.info()
```

```
<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
```

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

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

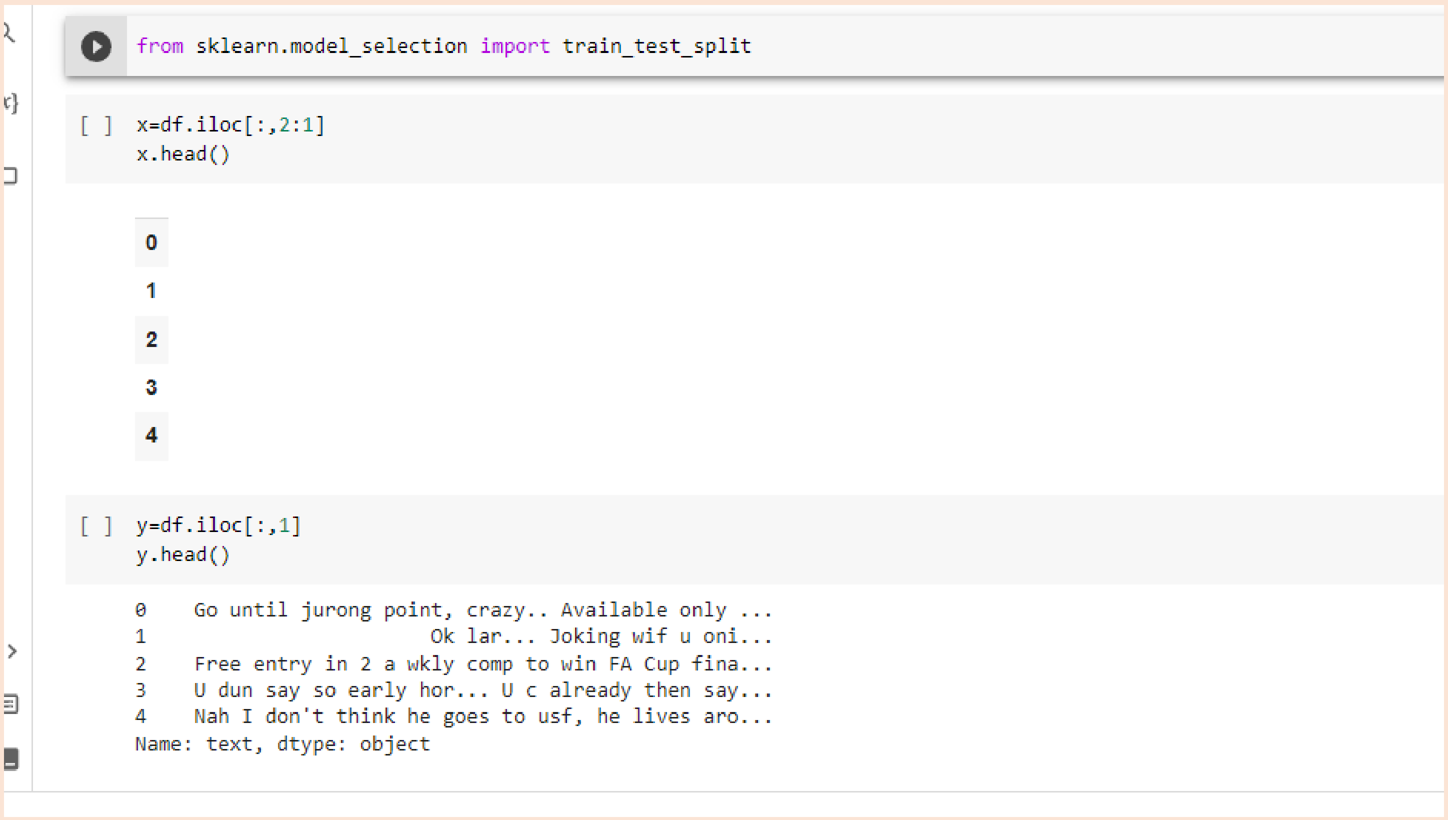


```
df.rename({"v1":"label","v2":"text"},inplace=True,axis=1)
```

```
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
5570	ham	The guy did some bitching but I acted like i'd...	NaN	NaN	NaN
5571	ham	Rofl. Its true to its name	NaN	NaN	NaN

```
from sklearn.preprocessing import LabelEncoder
```



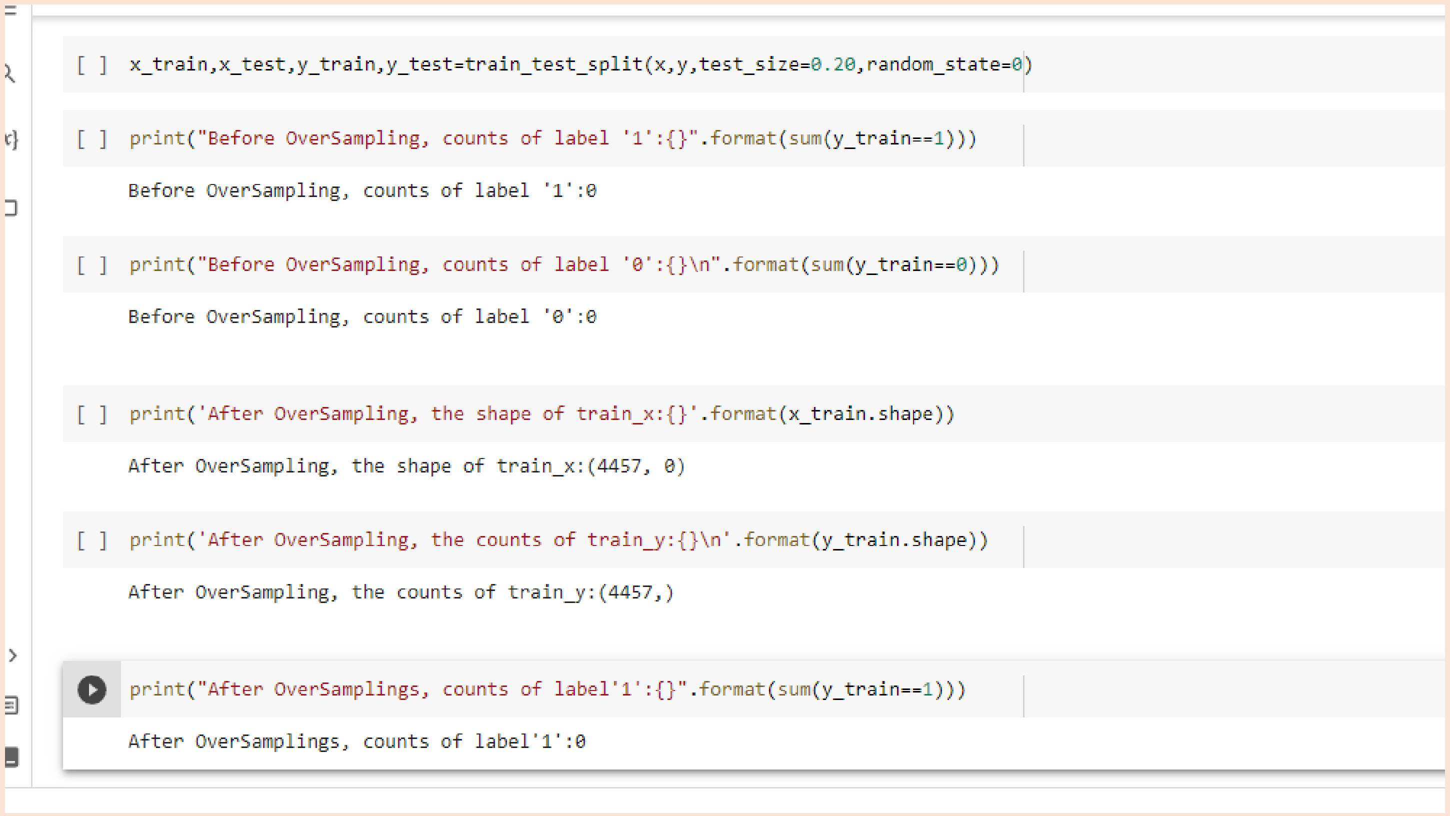
```
from sklearn.model_selection import train_test_split
```

```
[ ] x=df.iloc[:,2:1]
    x.head()
```

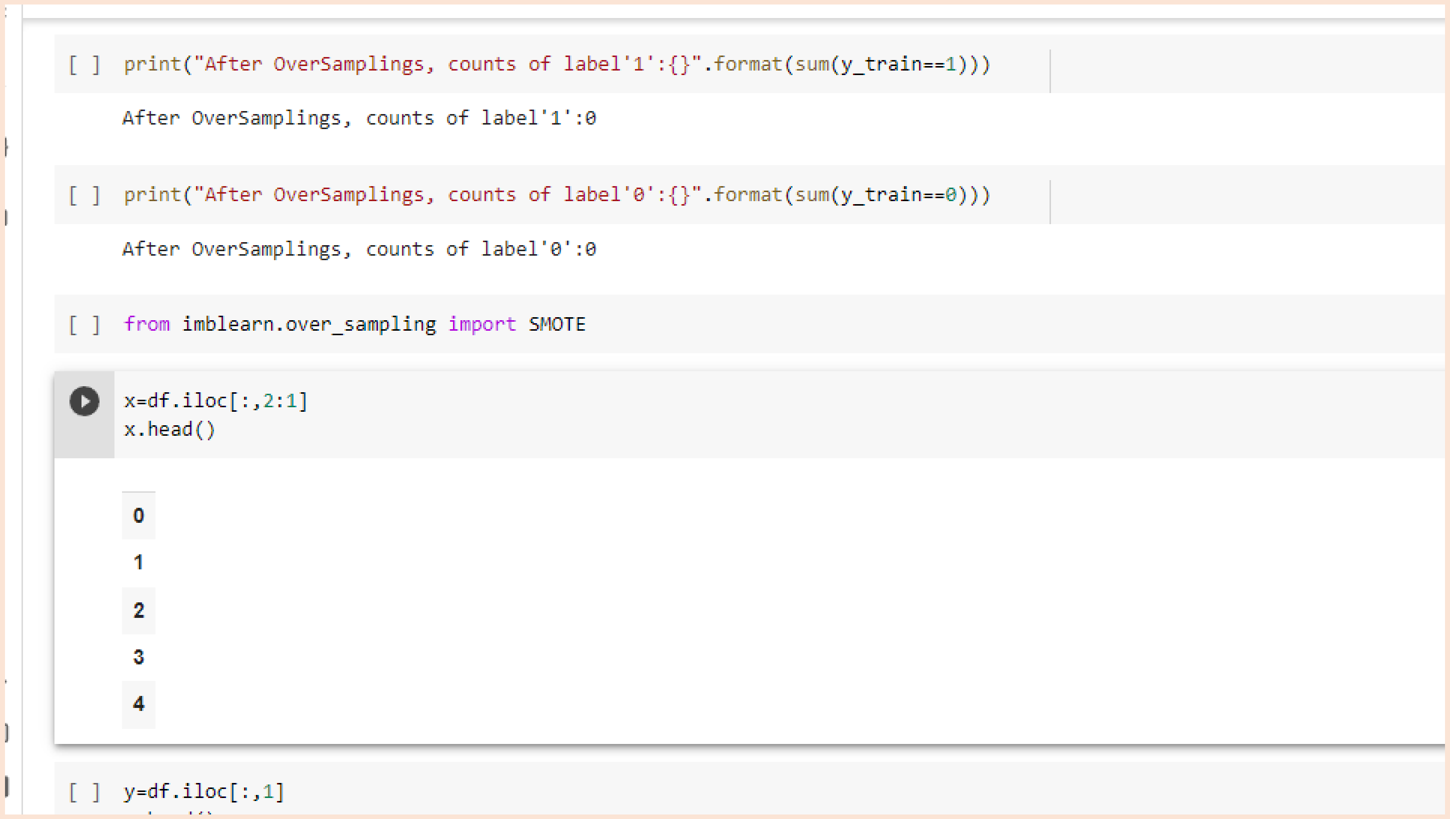
```
0
1
2
3
4
```

```
[ ] y=df.iloc[:,1]
    y.head()
```

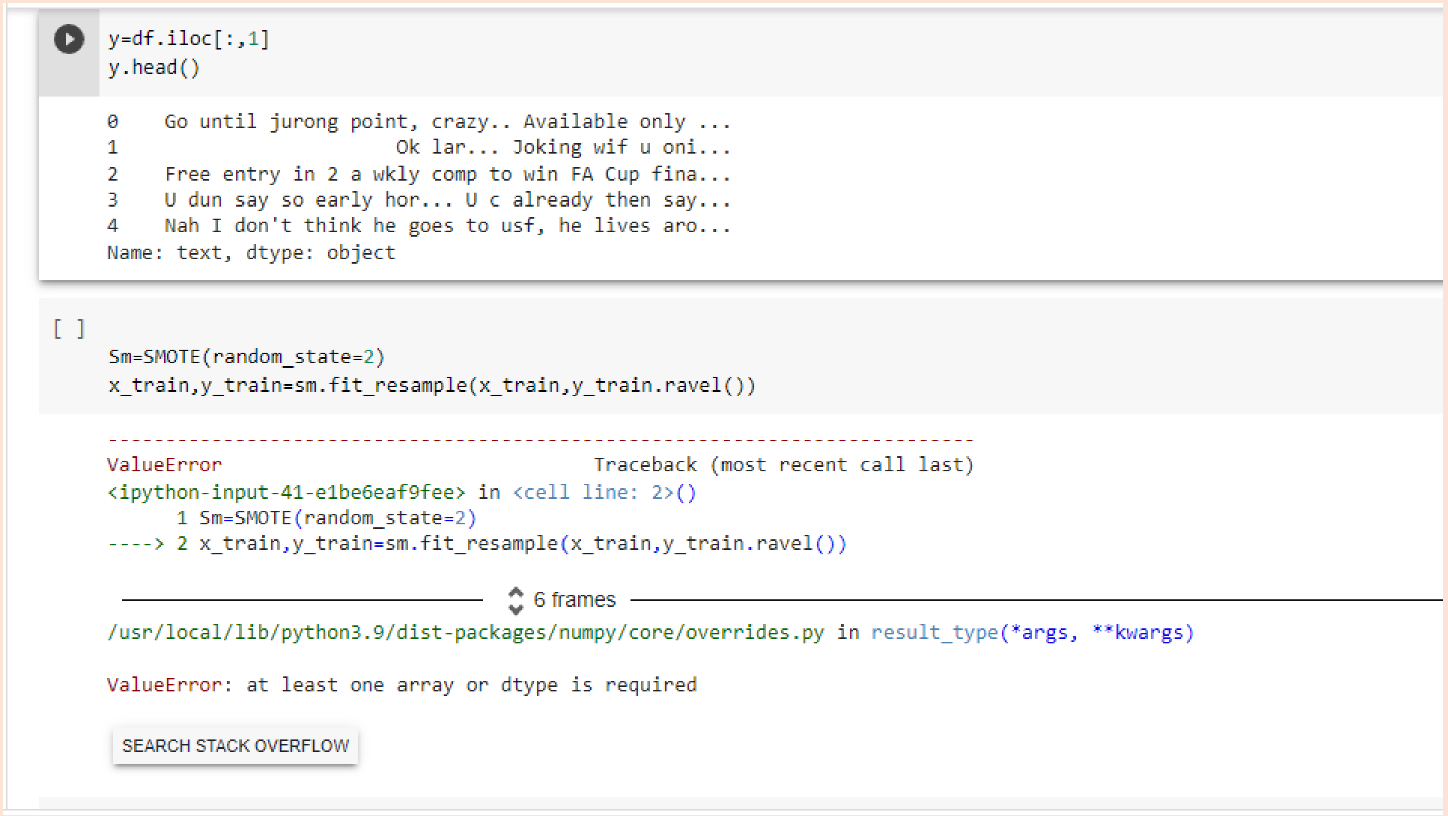
```
0    Go until jurong point, crazy.. Available only ...
1                        Ok lar... Joking wif u oni...
2    Free entry in 2 a wkly comp to win FA Cup fina...
3    U dun say so early hor... U c already then say...
4    Nah I don't think he goes to usf, he lives aro...
Name: text, dtype: object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



```
[ ] x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)

[ ] print("Before OverSampling, counts of label '1':{}".format(sum(y_train==1)))

    Before OverSampling, counts of label '1':0

[ ] print("Before OverSampling, counts of label '0':{}\n".format(sum(y_train==0)))

    Before OverSampling, counts of label '0':0


[ ] print('After OverSampling, the shape of train_x:{}'.format(x_train.shape))

    After OverSampling, the shape of train_x:(4457, 0)

[ ] print('After OverSampling, the counts of train_y:{}\n'.format(y_train.shape))

    After OverSampling, the counts of train_y:(4457,)


    print("After OverSamplings, counts of label'1':{}".format(sum(y_train==1)))

    After OverSamplings, counts of label'1':0
```



```
[ ] print("After OverSamplings, counts of label'1':{}".format(sum(y_train==1)))

    After OverSamplings, counts of label'1':0

[ ] print("After OverSamplings, counts of label'0':{}".format(sum(y_train==0)))

    After OverSamplings, counts of label'0':0

[ ] from imblearn.over_sampling import SMOTE

    x=df.iloc[:,2:1]
    x.head()

    0
    1
    2
    3
    4

[ ] y=df.iloc[:,1]
```

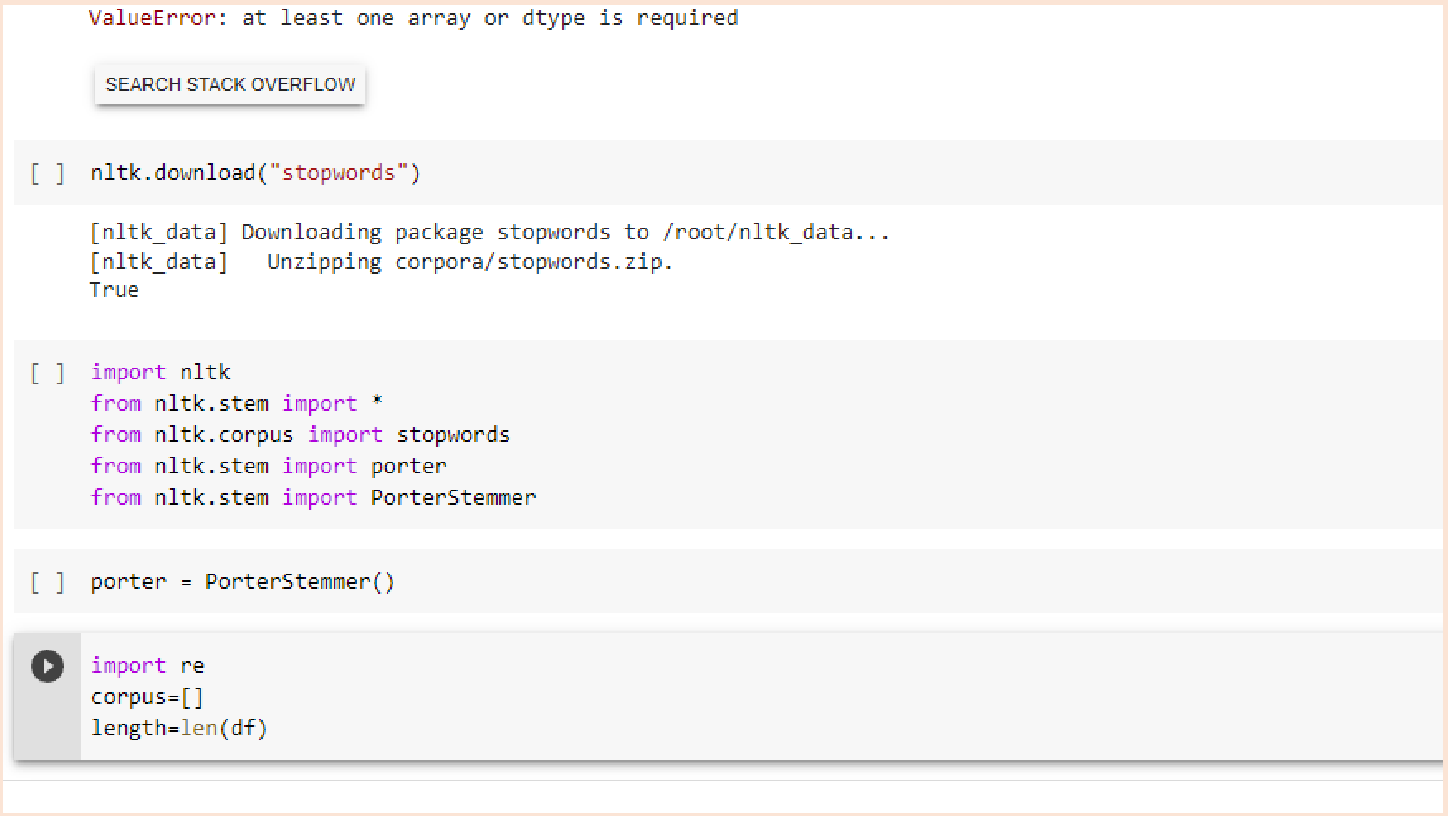


```
y=df.iloc[:,1]
y.head()
```

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

```
Sm=SMOTE(random_state=2)
x_train,y_train=sm.fit_resample(x_train,y_train.ravel())
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-41-e1be6eaf9fee> in <cell line: 2>()
      1 Sm=SMOTE(random_state=2)
----> 2 x_train,y_train=sm.fit_resample(x_train,y_train.ravel())

/usr/local/lib/python3.9/dist-packages/numpy/core/overrides.py in result_type(*args, **kwargs)

ValueError: at least one array or dtype is required
```



```
ValueError: at least one array or dtype is required
```

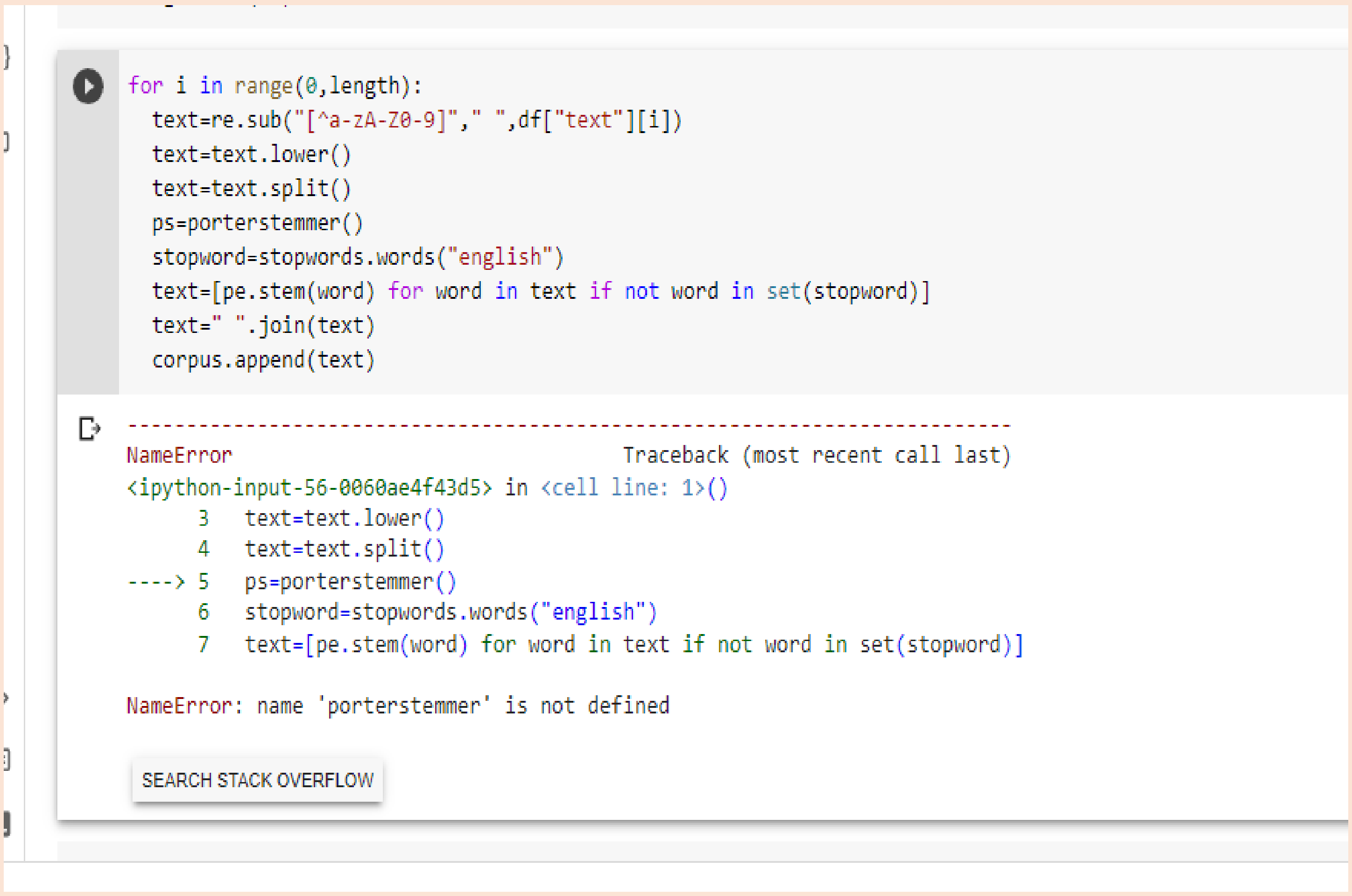
```
nltk.download("stopwords")
```

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

```
import nltk
from nltk.stem import *
from nltk.corpus import stopwords
from nltk.stem import porter
from nltk.stem import PorterStemmer
```

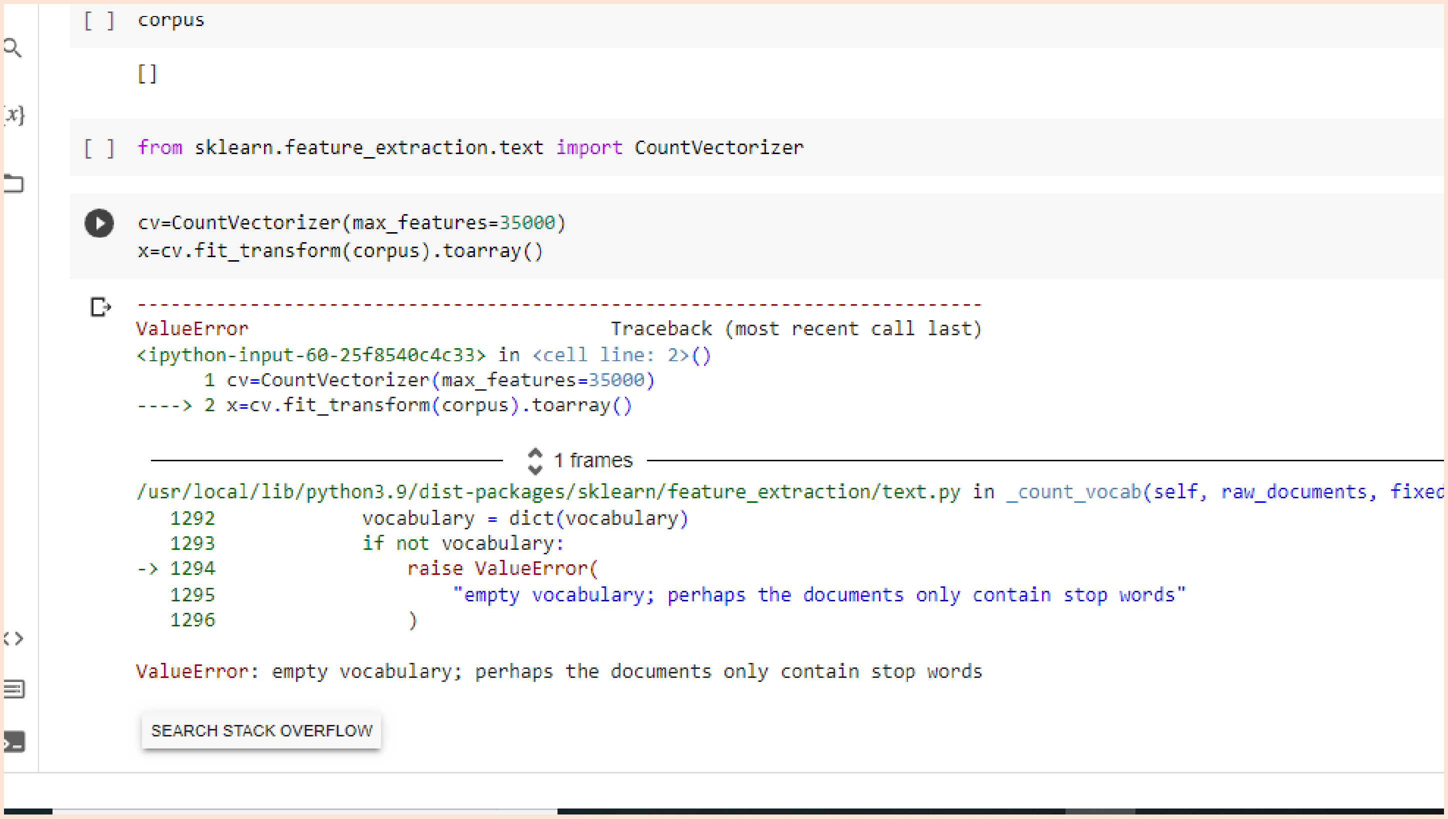
```
porter = PorterStemmer()
```

```
import re
corpus=[]
length=len(df)
```



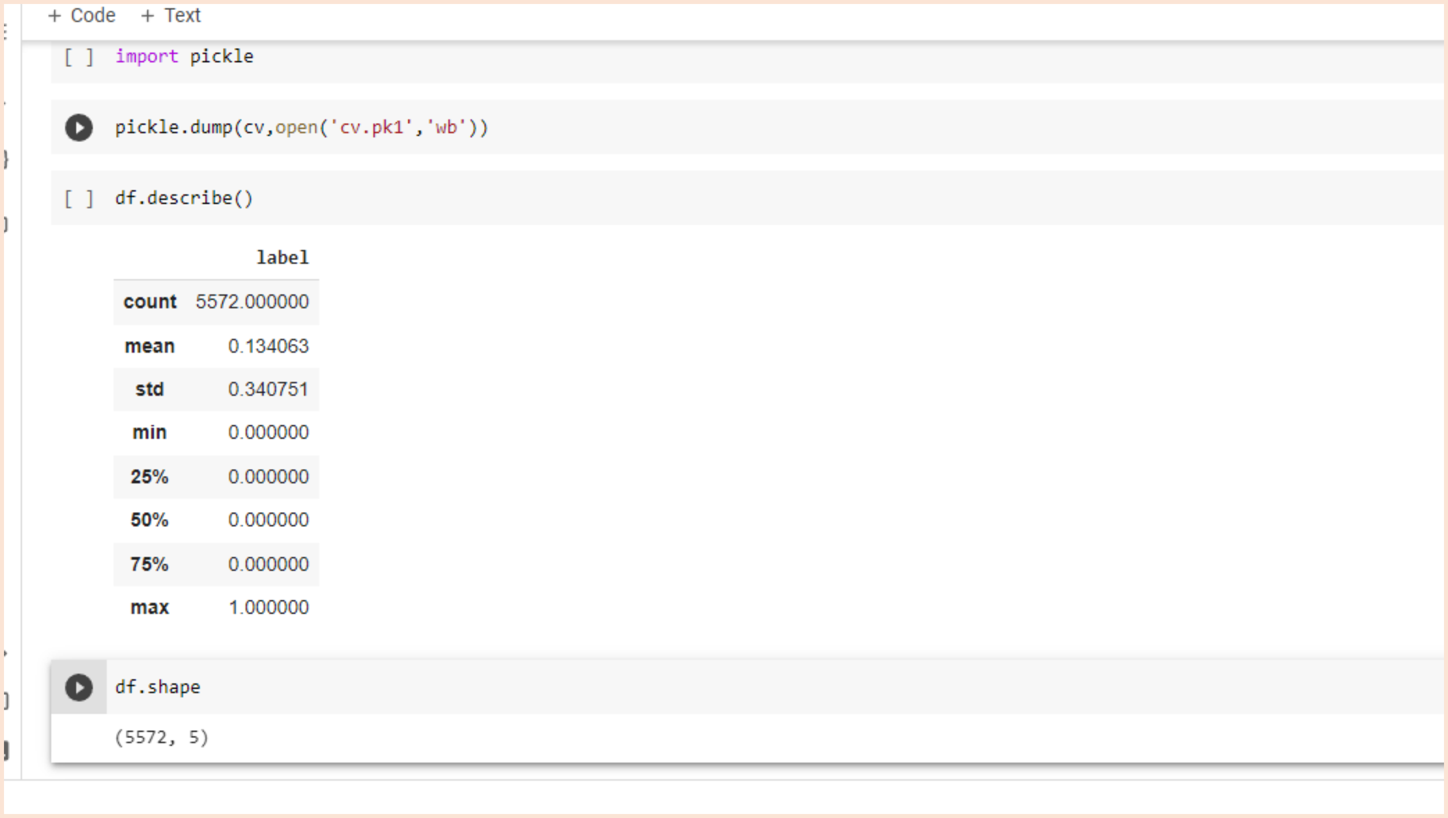
```
for i in range(0,length):
  text=re.sub("[^a-zA-Z0-9]"," ",df["text"][i])
  text=text.lower()
  text=text.split()
  ps=porterstemmer()
  stopword=stopwords.words("english")
  text=[pe.stem(word) for word in text if not word in set(stopword)]
  text=" ".join(text)
  corpus.append(text)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-56-0060ae4f43d5> in <cell line: 1>()
      3   text=text.lower()
      4   text=text.split()
----> 5   ps=porterstemmer()
      6   stopword=stopwords.words("english")
      7   text=[pe.stem(word) for word in text if not word in set(stopword)]

NameError: name 'porterstemmer' is not defined
```



```
[ ] corpus

    []

[ ] from sklearn.feature_extraction.text import CountVectorizer

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

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-60-25f8540c4c33> in <cell line: 2>()
      1 cv=CountVectorizer(max_features=35000)
----> 2 x=cv.fit_transform(corpus).toarray()

                              1 frames
/usr/local/lib/python3.9/dist-packages/sklearn/feature_extraction/text.py in _count_vocab(self, raw_documents, fixed
   1292             vocabulary = dict(vocabulary)
   1293             if not vocabulary:
-> 1294                 raise ValueError(
   1295                     "empty vocabulary; perhaps the documents only contain stop words"
   1296                 )

ValueError: empty vocabulary; perhaps the documents only contain stop words
```



+ Code + Text

```
[ ] import pickle
```

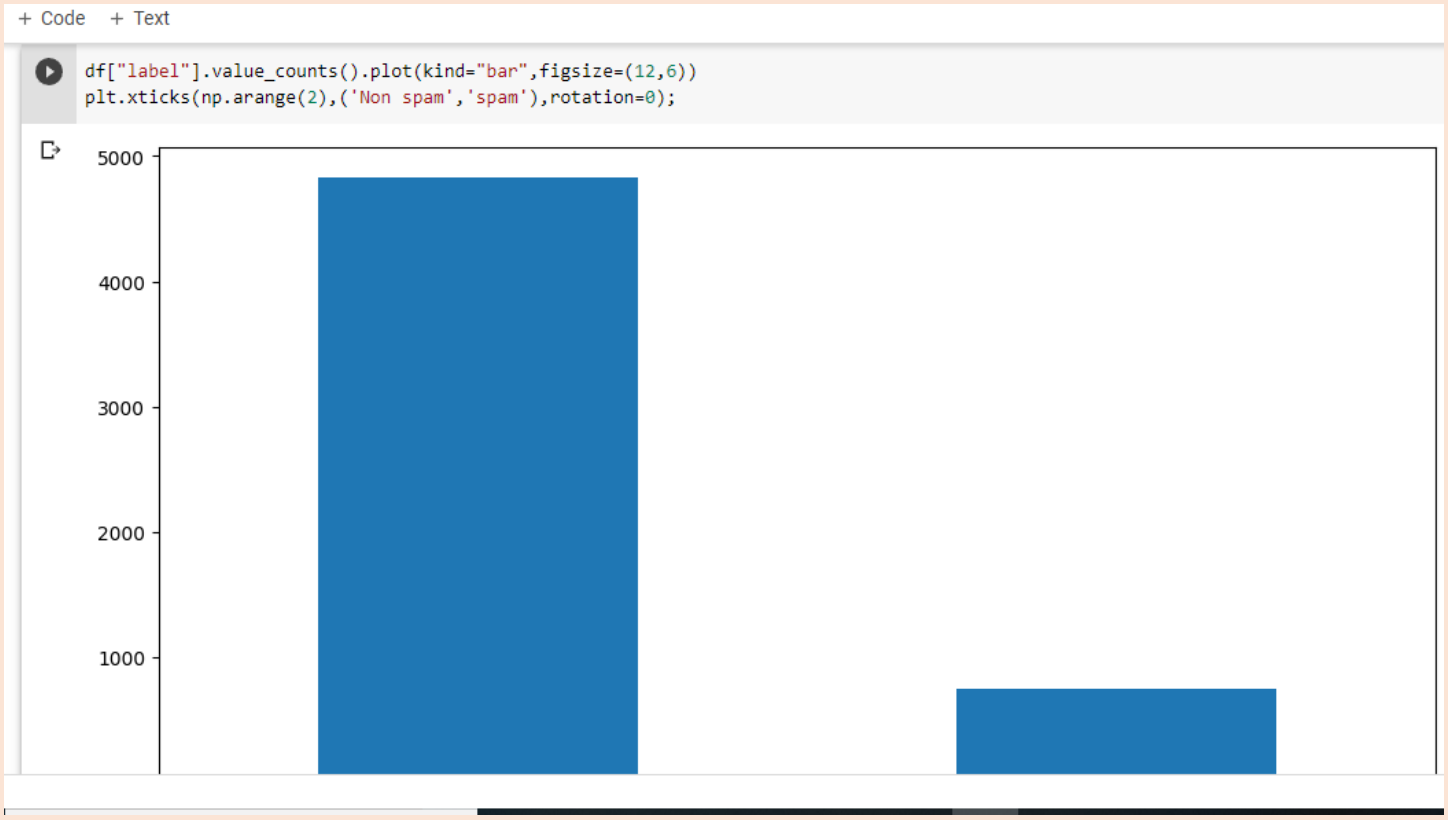
```
pickle.dump(cv,open('cv.pk1','wb'))
```

```
[ ] 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

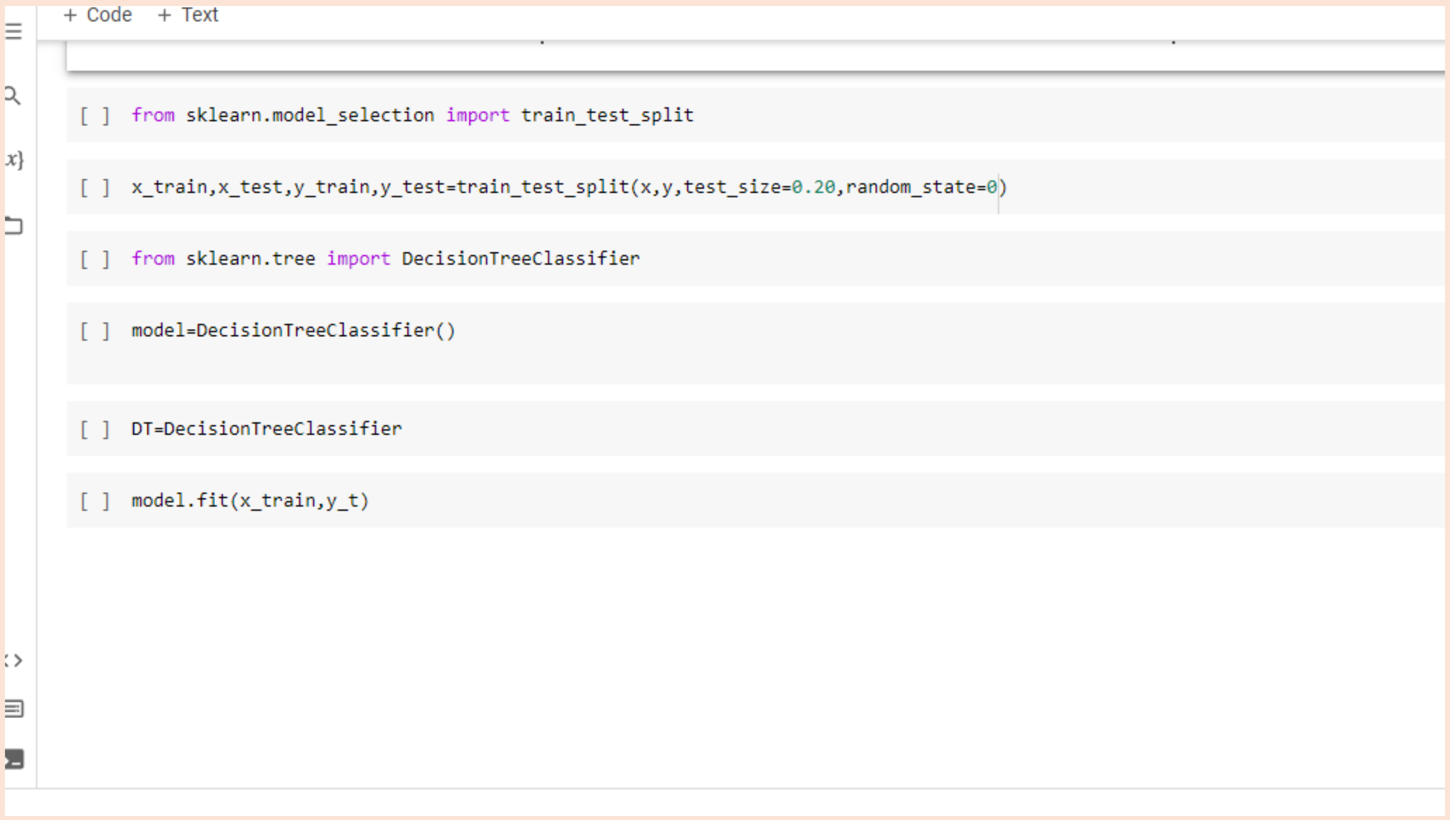
```
df.shape
```

```
(5572, 5)
```



+ Code + Text

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



```
[ ] 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)
```

```
[ ] from sklearn.tree import DecisionTreeClassifier
```

```
[ ] model=DecisionTreeClassifier()
```

```
[ ] DT=DecisionTreeClassifier
```

```
[ ] model.fit(x_train,y_t)
```

Advantages and disadvantages

Spam filtering in advantages:

Spam is most well-know for spreading viruses and scams to unwitting people across the internet, but it can actually cause plenty of problems for the modern business. This is why effective spam filtering, like securence spam filtering, is an important part of running a successful business in the 21st century. Here are just a few reasons why spam filtering is important for not only keeping you safe from viruses, but also for helping your company be more effective and successful.

1.It streamlines inboxes

- ❖ The average office workers receives roughly 121 emails per day, half of which are estimated to be spam. But even at 60 emails a day, it is easy to lose important communication to the sheer number that are coming in.
- ❖ This is one of the secret benefits of spam filtering that people do not know about: it simply streamline your inbox. With less garbage coming into your inbox, you can actually go through your emails more effectively and stay in touch with those who matter.

2.Protect Against Malware

- ❖ Malware, viruses, and other forms of malicious attacks are heading to people's email inboxes every days. some of these can be easily weeded out by your internet provider's own spam filters, but spam gets smarter everyday.
- ❖ Smarter spam gets into more inboxes, which makes it more likely to be opened and more likely to cause harm. With spam tactics that are being used today so you can ensure that your email inboxes stay free of harmful messages .

3.keep you compliant

- ❖ Many small and medium sized businesses are losing out on important client toady because their cybersecurity is not up to par. spam filtering is a major part of any cybersecurity plan, and it You compliant with the wishes and demands of companies and agencies that are concerned about their information.

- ❖ Without proper spam filtering, you could unwittingly put spyware in your emails and break security protocols the result could be a loss of business, reputation, and ultimately income.

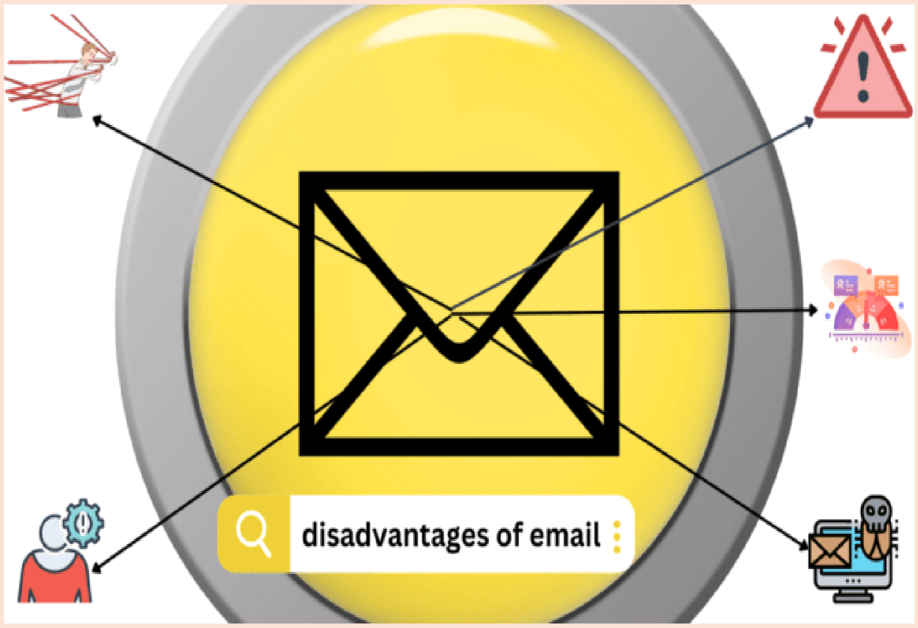
4.It saves you money

- ❖ Everyday, someone falls prey to a phishing scam, a particular kind of spam-based schema where someone thinks they are getting a legitimate emails and ends up divulging credit card information.

Disadvantage of spam filtering

- ❖
- ❖ There are many people who will become nervous when they receive a wanted mail which they usually considered as spam. These people are getting a plenty of mails.
- ❖ What they do with these is they simply arrange all the unwanted message and wanted message and throw it to their waste bin. To send a paper we need to cut a three and when throwing it away we are polluting are environment.
- ❖ Here is the benefits of the electronic mailing system. In this one can easily keep the mail they want and can delete the unwanted. Theres message will disappear the moment we deleted it and there will not be a trail of their existence. By this way we are not only saving our time and money but also we are protecting our environment.

- The mailbox is the place where people use to communicate with others and they try to keep it as good possible. We can't predict when the evils like spam will come into our inbox. But sometimes we can get interesting offers and will be able to discover different areas of internet which never heard of. In the future this can become our favorite hobbies .
- if my opinion spam mails are good they will take us to a world that we never have been. But some people think that the internet lost its purity. So they complained about spam to there is and force them to take further action against the spam. These kinds of action make servers to dual check for piracy and use highly encrypted lines for any communication. Most are now providing filters to catch spam mails.
- The internet is very amazing because it never make barriers for people to communicate even if the government and large organization block them. We should try our best keep spam message away by cleaning our inbox that contain spam message.



Application

- spam filtering applications are software programs designed to identify and block unwanted and unsolicited message, also know as spam, from entering a user's inbox. some popular spam filtering application include:
  - **GMAILS**: Gmail's spam filtering technology automatically identifies and filters spam message into a separate spam folder, so that they don't clutter the inbox.
  - **SPAM ASSASSIN**: Spam assassin is an open-source spam filtering application that uses a variety of techniques to identify and source spam message, including text analysis, header analysis, and Bayesian filtering.
  - **SPAMFIGHTER**: spam fighter is a spam filtering application that integrates with outlook, outlook express, and window mall and uses a combination of advanced algorithms and use-defined filter to identify and block spam messages.
  - **KASPERSKY ANTI-SPAM**: Kaspersky anti-spam is a comprehensive spam filtering solution that integrates with Microsoft exchange, lotus domino, and other email server, and uses a combination of signature-based and heuristic analysis to identify and block spam message.
  - **BARRACUDA SPAM FIREWALL**: The barracuda spam firewall is

hardware-based spam filtering solution that integrates with a range of email servers, and uses a combination of signature-based and behavioral analyses to identify and block spam message.

Conclusion

Summarizing above-listed, we obtain the following conclusion

- *So, spammers constantly change external sighs of w-mails to skip spam filtering system, there arises a need for adaptive filtering system, which should have the ability to react quickly to the changes and provides fast and qualitative self-tuning in accordance with a new set of feature.*
- *Since the filter are trained on a very limited number of message that come only to a specific user or a special mail provider, the quality of filtering in the existing server client and server filtering system is rather lo. But it can be improved if to apply the hybrid filtration system in other word the complex hierarchical in the identification of the filtering errors and the appropriate setting of filtering at(each level organization level, mail provider level).*
- *Therefore it is quite perspective for solving this problem , the combination model on two widespread approaches as using the personal e-mail classification model on a server side solution. Development of set side personalized e-mail filtering system that use the learning -based classification algorithm based on data mining methods is a very perspective direction.*

This statement is supported by the following:

- *Personalized server side filtering system are preferable than the client side solutions, because provide universal access to an e-mail, reduce expenses, which is very important for corporate user.*
- *Personalized server side filtering systems are more preferably because of greater accuracy and fewer errors in comparison with general model;*
- *Personalized server side filtering system offered in author's another paper[59] bases on the universal Declaration of human rights and has a universal character, can be applied in all countries;*
- *Learning -based algorithms used in personalized server side filtering system exceed traditional ones because of a number of fundamental qualities (quality of filtering, the absence of updates, independent from external knowledge bases).*

Future scope

- ❖ Email spam has been the focus of studies for a long time. Though there are many different technique to back spam email message to reach users inbox, filtering is the most commonly used mechanism and has gained success to some extent.
- ❖ Give the large number of usage of email worldwide, email spam is still plentiful and scale of the problem is enormous. Researchers and organization make the files smart and self-learning but spammers are a step ahead.
- ❖ They keep on finding techniques to devices the filter and their learning mechanism. Hence the problem still remains giving scope for researchers to work in the area. This work is an effort in the same scope to reduce false negatives/spam in the inbox are user which has deceived the organizational filters. It is observed that this further filtering by training the filter with user specific data did make a difference in the amount of false positives.
- ❖ Future work involves creating the feature sets including creating domain specific keyword and list of organization which can be fed to the filter, conducting experiments and then observing the results to record the improvements.

- As a software developer, email is one of the very important tools for the communication.to have effective communication, spam filtering is one of the important features.

- For this email spamming data set, it is distributed by spam assassin, you can click in to data set. There are a few categories of the data , you can read the readme.html to get more background information on the data.

Appendix

A.SOURCE CODE:

Importing the libraries:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report,confusion_matrix
```

Read the Dataset

```
       # read and run the data


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

Handling missing values

```
      df.info()

   df.isna().sum()

   df.describe(include='all')

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

  df.tail()
```

Handling categorical values

```
 from sklearn.preprocessing import LabelEncoder

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

```
# independent variable
   x=df.iloc[:,1:2]
   x.head()
   y=df.iloc[:,1:]

   y.head()
```

Handling imbalance data

```
# split training& testing

from sklearn.model_selection import train_test_split
xtrian,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.20,random_state=0)
print(ytrain.shape)
print(ytest.shape)
#model building
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x,y)
print("Before OverSampling, counts of label '1': {}".format(sum(y_train == 1)))
print("Before OverSampling, counts of label '0':{} \n".format(sum(y_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())
```

Cleaning the text data

```
 import nltk

from nltk.corpus import stopwords
nltk.download("stopwords")
import re
corpus=[]
length=len(df)
from nltk.stem import porterstemmer
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=stopword.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('c1.pk1','wb')
```

Descriptive statistical

Universal analysis

```
    df.describe()

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

Scaling the data

```
sc=StandardScaler()
x=bal=sc.fit_transform(x_bal)
x_bal=pd.DataFrame(x_bal,columns=names)
```

Splitting data into train and test

```
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)
```

Training the model in multiple algorithms

Decision tree model

```
# decision tree
from sklearn.tree import DecisionTreeClassifier
dt.DecisionTreeClassifier()
model=DecisionTreeClassifier()
model.fit(X_train_res,y_train_res)
```

Random forest model

```
model=DecisionTreeClassifier()
model.fit(X_train_res,y_train_res)
from sklearn.ensemble import RandomForestClassifier
model1=RandomForestClassifier()
model1.fit(x,y)
```

Naive bayes model

```
from sklearn.Navie_Bayes import MultinomialNB
model.fit(x,y)
model.fit(x_train_res, y_train_res)
# dependant variable

y=df.iloc[:,4:]
y.head()
```

ANN model

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Dense
model.add(Dense(6,activation="relu"))
model.add(Dense(units=100,activation="rule"))
model.add(Dense(units=100,activation="rule"))
model.add(Dense(units=1,activation="sigmoid"))
model.complie(optimizer="rmsprop",loss="accuracy",metrics=["accuracy"])
model.fit(xtrain, ytrain, batch_size=2,epochs=20)
```

Testing the model

```
y_pred=model.predict(x_test)
y_pred
y_pr=np.where(y_pred>0.5,1,0)
y_test
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*10)
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....")))
```

Testing model with multiple evaluation metrices

Compare the model

```
from sklearn.metrics import confusion_matrix,accuracy_score,classification_report
from sklearn.metrics import confusion_matrix,accuracy_score
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_pred1)
score1=accuracy_score(y_test,y_pred1)
print(cm)
print('Accuracy Score Is:- ',score1*100)
cm1=confusion_matrix(y_test, y_pred1)
score1=accuracy_score(y_test,y_pred1)
print(cm)
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)
```

comparing the model accuracy before & after applying hyperparameter tuning

```
from sklearn.metrics import confusion_matrix,accuracy_score
cm=confusion_matrix(y_test, y_pr)
score=accuray_score(y_test, y_pr)
print(cm)
print('Accuracy Score Is: ',score*100)
```