PROJECT REPORT

Project Report Template

1 **INTRODUCTION**

Overview

A brief description about your project

Purpose

The use of this project. What can be achieved using this.

2 **PROBLEM DEFINITION & DESIGN THINKING**

Empathy Map

Paste the empathy map screenshot

Ideation & Brainstorming Map

Paste the Ideation & brainstorming map screenshot

3 **RESULT**

Final findings (Output) of the project along with screenshots.

4 **ADVANTAGES & DISADVANTAGES**

List of advantages and disadvantages of the proposed solution

5 **APPLICATIONS**

The areas where this solution can be applied

6 **CONCLUSION**

Conclusion summarizing the entire work and findings.

7 **FUTURE SCOPE**

Enhancements that can be made in the future.

8 **APPENDIX**

A. Source Code

Attach the code for the solution built.

1. INTRODUCTION:

1.1 OVERVIEW:

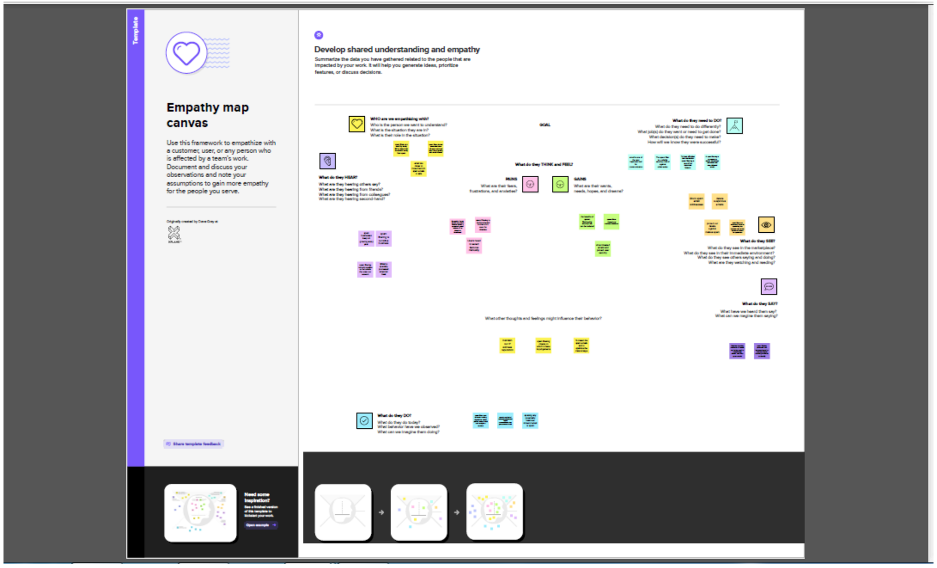
Over recent years, as the popularity of mobile phone devices has increased, Short Message Service (SMS) has grown into a multi-billion dollar industry. At the same time, reduction in the cost of messaging services has resulted in growth in unsolicited commercial advertisements (spams) being sent to mobile 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.

1.2 PURPOSE:

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.

2 . PROBLEM DEFINITION & DESIGN THINKING:

2.1 EMPATHY MAP:



Template

Empathy map canvas

Use this framework to empathize with a customer, user, or any person who is affected by a team's work. Document and discuss your observations and note your assumptions to gain more empathy for the people you serve.

Originally created by Dave Gray at XPLANE

Share template feedback

Need some inspiration? See a finished version of this template to kickstart your work. Open example

Develop shared understanding and empathy

Summarize the data you have gathered related to the people that are impacted by your work. It will help you generate ideas, prioritize features, or discuss decisions.

WHO are we empathizing with?
Who is the person we want to understand?
What is the situation they are in?
What is their role in the situation?

GOAL

What do they need to DO?
What do they need to do differently?
What job(s) do they want or need to get done?
What decision(s) do they need to make?
How will we know they were successful?

What do they HEAR?
What are they hearing others say?
What are they hearing from friends?
What are they hearing from colleagues?
What are they hearing second-hand?

What do they THINK and FEEL?

PAINS
What are their fears, frustrations, and anxieties?

GAINS
What are their wants, needs, hopes, and dreams?

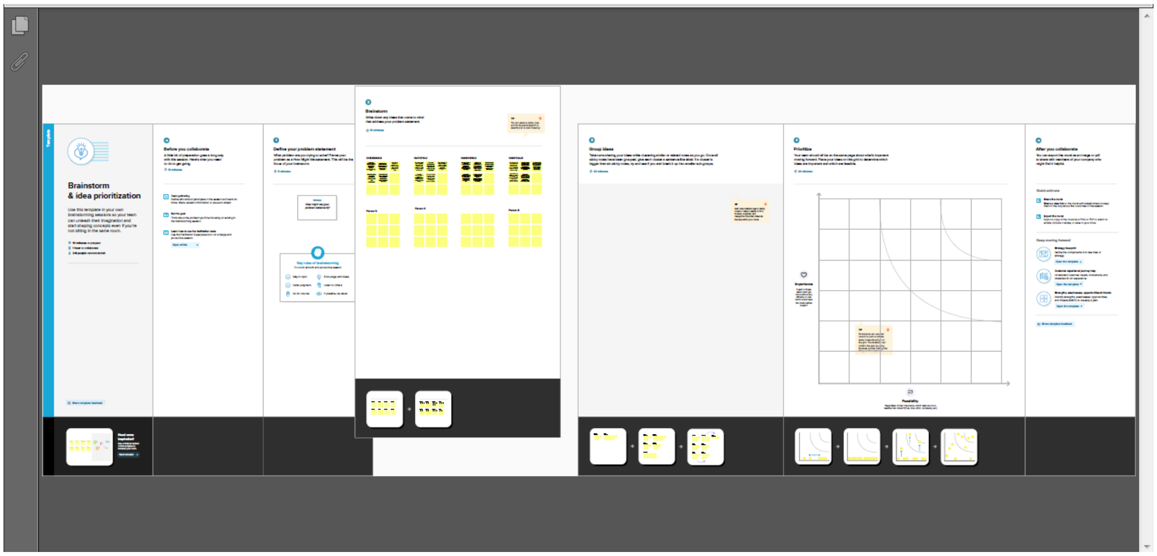
What do they SEE?
What do they see in the marketplace?
What do they see in their immediate environment?
What do they see others saying and doing?
What are they watching and reading?

What do they SAY?
What have we heard them say?
What can we imagine them saying?

What other thoughts and feelings might influence their behavior?

What do they DO?
What do they do today?
What behavior have we observed?
What can we imagine them doing?

2.2 IDEATION & BRAINSTORMING MAP:



Template

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.

Share template feedback

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.

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.

Key rules of brainstorming

Brainstorm

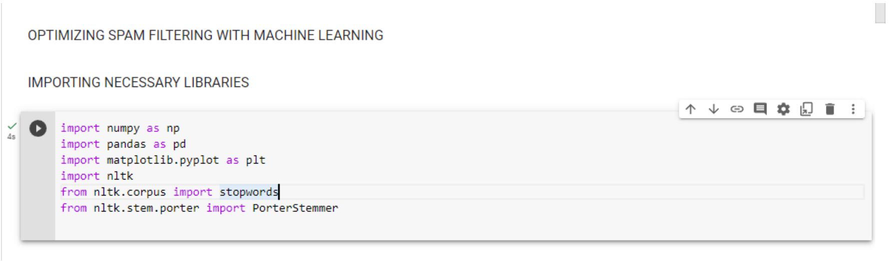
Write down any ideas that come to mind that address your problem statement.

Group ideas

Prioritize

After you collaborate

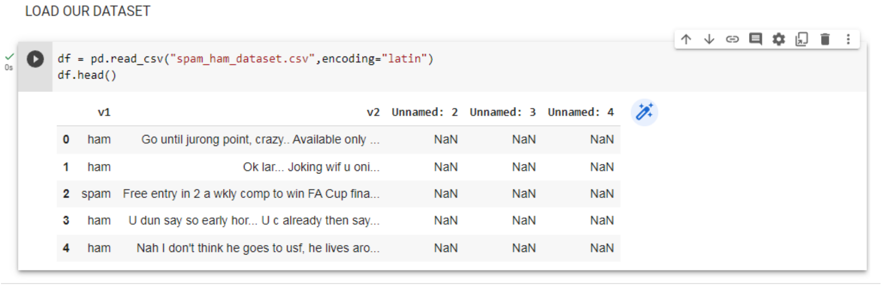
3. RESULT:



OPTIMIZING SPAM FILTERING WITH MACHINE LEARNING

IMPORTING NECESSARY LIBRARIES

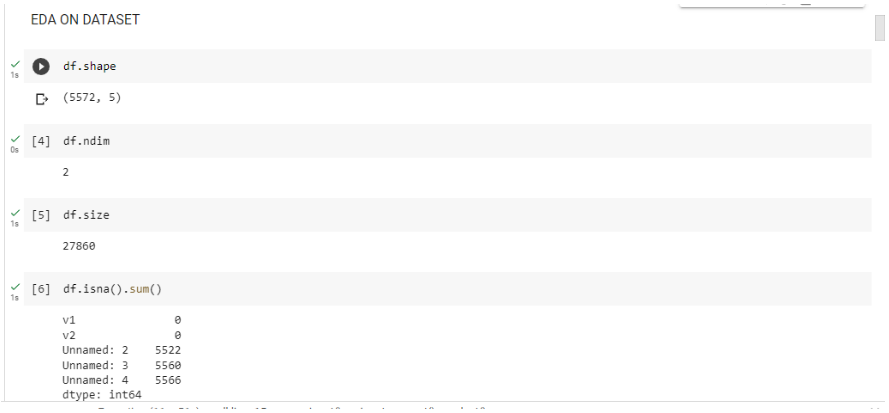
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import nltk
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
```



LOAD OUR DATASET

```
df = pd.read_csv("spam_ham_dataset.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



EDA ON DATASET

```
df.shape
```

```
(5572, 5)
```

```
[4] df.ndim
```

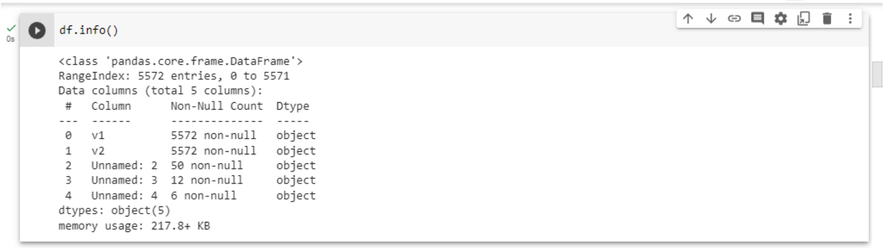
```
2
```

```
[5] df.size
```

```
27860
```

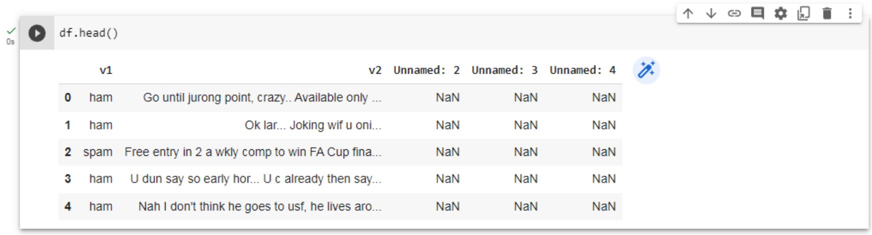
```
[6] df.isna().sum()
```

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



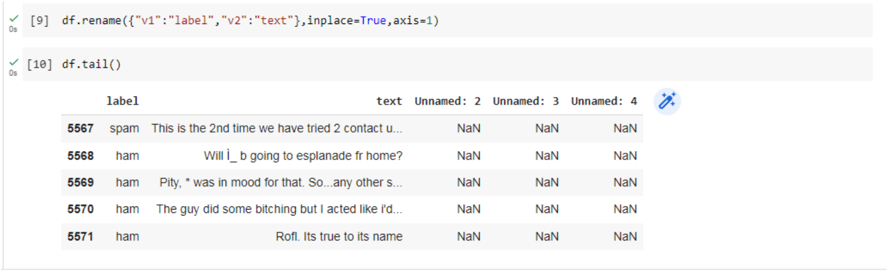
```
df.info()
```

```
<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.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

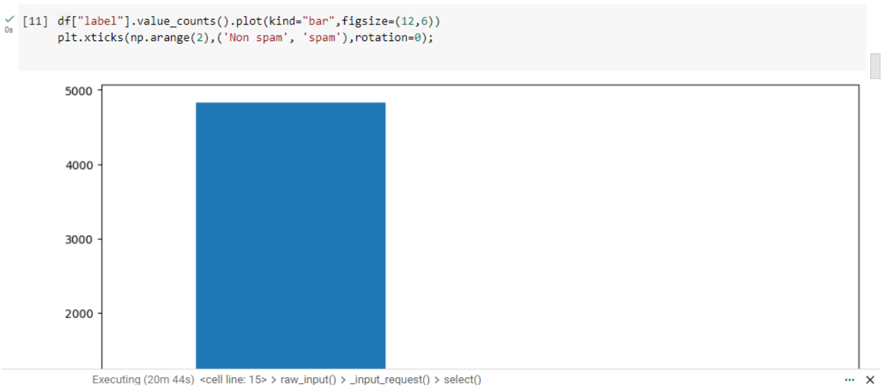


```
[9] df.rename({"v1":"label","v2":"text"},inplace=True,axis=1)
```

```
[10] df.tail()
```

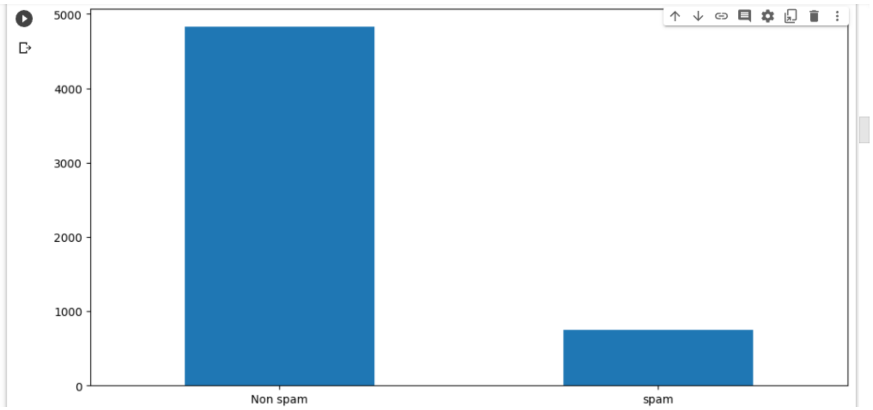
	label	text	Unnamed: 2	Unnamed: 3	Unnamed: 4
5567	spam	This is the 2nd time we have tried 2 contact u...	NaN	NaN	NaN
5568	ham	Will Ì_ b going to esplanade fr home?	NaN	NaN	NaN
5569	ham	Pity, * was in mood for that. So...any other s...	NaN	NaN	NaN
5570	ham	The guy did some bitching but I acted like i'd...	NaN	NaN	NaN
5571	ham	Rofl. Its true to its name	NaN	NaN	NaN

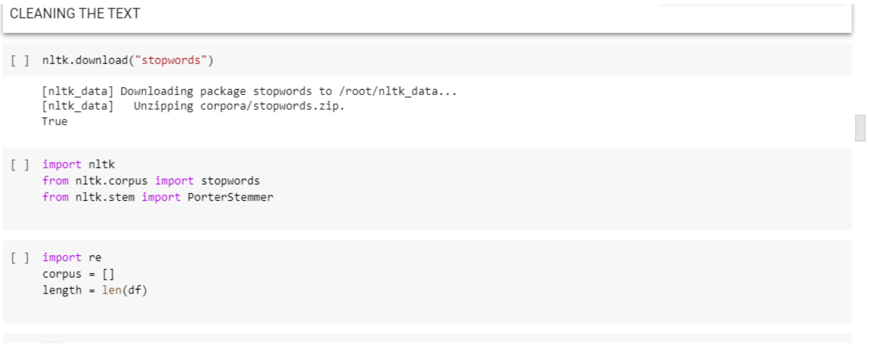
LET'S VISUALIZE THE COLUMN LABEL



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

Executing (20m 44s) <cell line: 15> > raw_input() > _input_request() > select()





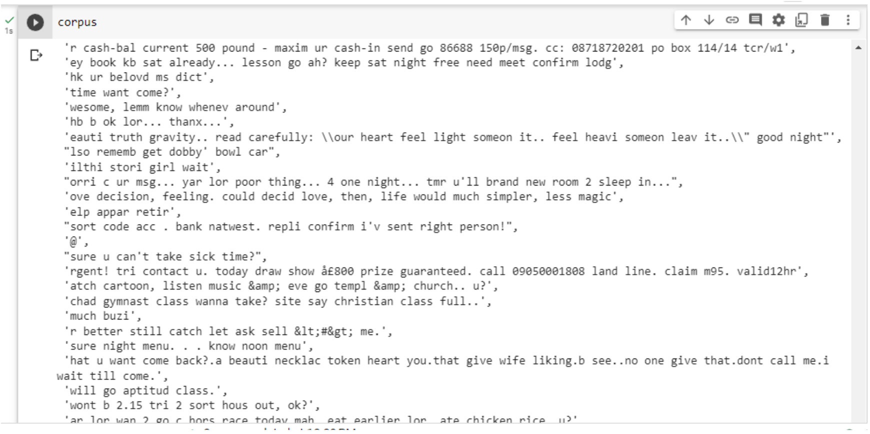
CLEANING THE TEXT

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

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

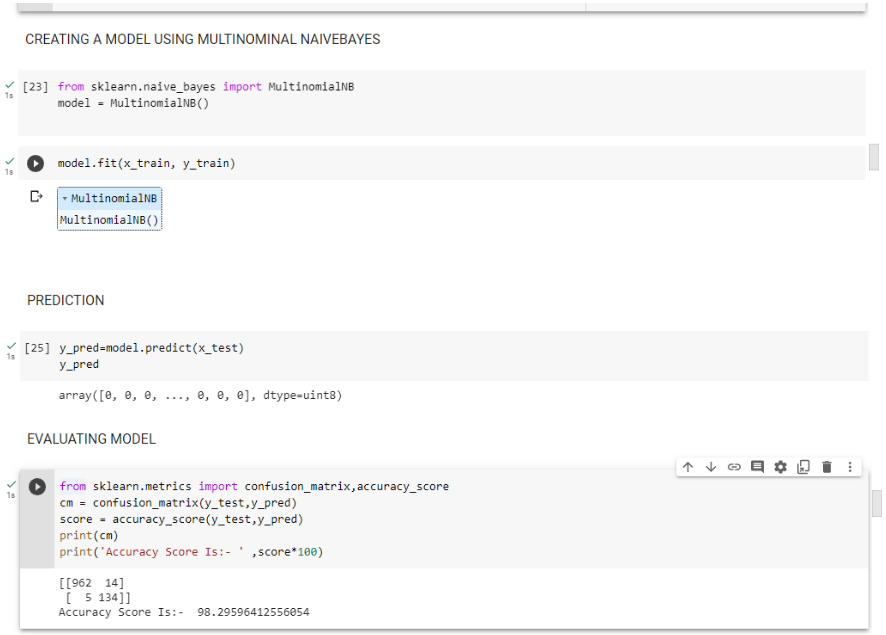
```
[ ] import nltk
    from nltk.corpus import stopwords
    from nltk.stem import PorterStemmer
```

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



```
corpus
```

```
 'r cash-bal current 500 pound - maxim ur cash-in send go 86688 150p/msg. cc: 08718720201 po box 114/14 tcr/w1',
 'ey book kb sat already... lesson go ah? keep sat night free need meet confirm lodg',
 'hk ur belovd ms dict',
 'time want come?',
 'wesome, lemm know whenev around',
 'hb b ok lor... thanx...',
 'eauti truth gravity.. read carefully: \\our heart feel light someon it.. feel heavi someon leav it..\\" good night"',
 "lso rememb get dobby' bowl car",
 'ilthi stori girl wait',
 "orri c ur msg... yar lor poor thing... 4 one night... tmr u'll brand new room 2 sleep in...",
 'ove decision, feeling. could decid love, then, life would much simpler, less magic',
 'elp appar retir',
 "sort code acc . bank natwest. repli confirm i'v sent right person!",
 '@',
 "sure u can't take sick time?",
 'rgent! tri contact u. today draw show å£800 prize guaranteed. call 09050001808 land line. claim m95. valid12hr',
 'atch cartoon, listen music &amp; eve go templ &amp; church.. u?',
 'chad gymnast class wanna take? site say christian class full..',
 'much buzi',
 'r better still catch let ask sell &lt;#&gt; me.',
 'sure night menu. . . know noon menu',
 'hat u want come back?.a beauti necklac token heart you.that give wife liking.b see..no one give that.dont call me.i
wait till come.',
 'will go aptitud class.',
 'wont b 2.15 tri 2 sort hous out, ok?',
 'ar lor wan 2 go c hors race today mah. eat earlier lor. ate chicken rice. u?'
```



CREATING A MODEL USING MULTINOMINAL NAIVEBAYES

```
[23] from sklearn.naive_bayes import MultinomialNB
     model = MultinomialNB()
```

```
model.fit(x_train, y_train)

▾ MultinomialNB
MultinomialNB()
```

PREDICTION

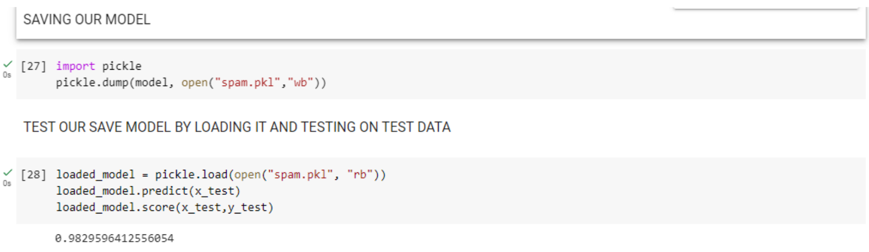
```
[25] y_pred=model.predict(x_test)
     y_pred

     array([0, 0, 0, ..., 0, 0, 0], dtype=uint8)
```

EVALUATING MODEL

```
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:- ' ,score*100)

[[962  14]
 [  5 134]]
Accuracy Score Is:-  98.29596412556054
```



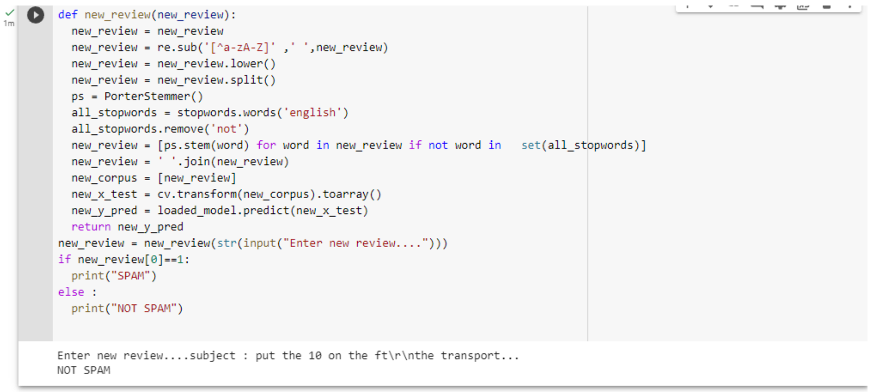
SAVING OUR MODEL

```
[27] import pickle
     pickle.dump(model, open("spam.pkl","wb"))
```

TEST OUR SAVE MODEL BY LOADING IT AND TESTING ON TEST DATA

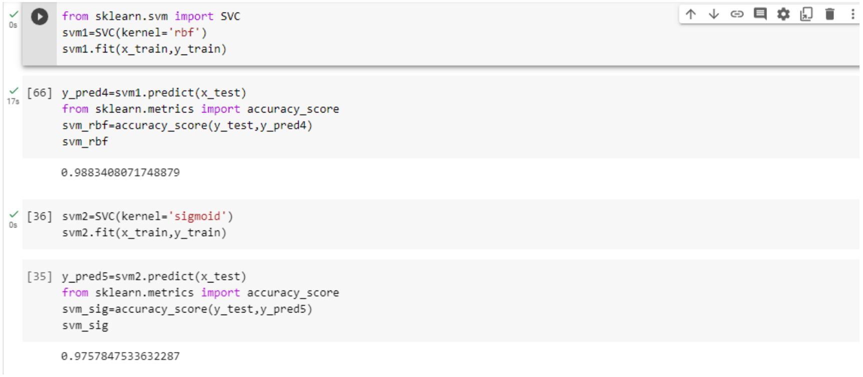
```
[28] loaded_model = pickle.load(open("spam.pkl", "rb"))
     loaded_model.predict(x_test)
     loaded_model.score(x_test,y_test)

     0.9829596412556054
```



```
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()
  new_y_pred = loaded_model.predict(new_x_test)
  return new_y_pred
new_review = new_review(str(input("Enter new review....")))
if new_review[0]==1:
  print("SPAM")
else :
  print("NOT SPAM")

Enter new review....subject : put the 10 on the ft\r\nthe transport...
NOT SPAM
```



```
from sklearn.svm import SVC
svm1=SVC(kernel='rbf')
svm1.fit(x_train,y_train)
```

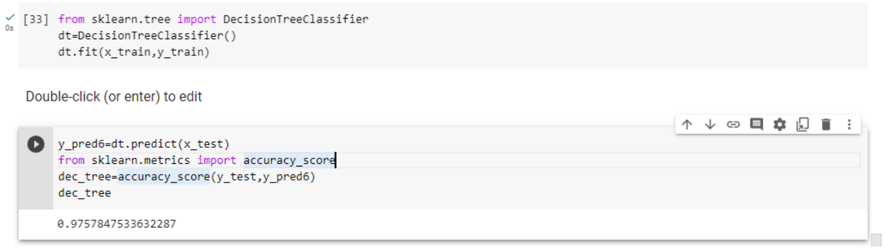
```
[66] y_pred4=svm1.predict(x_test)
     from sklearn.metrics import accuracy_score
     svm_rbf=accuracy_score(y_test,y_pred4)
     svm_rbf

     0.9883408071748879
```

```
[36] svm2=SVC(kernel='sigmoid')
     svm2.fit(x_train,y_train)
```

```
[35] y_pred5=svm2.predict(x_test)
     from sklearn.metrics import accuracy_score
     svm_sig=accuracy_score(y_test,y_pred5)
     svm_sig

     0.9757847533632287
```

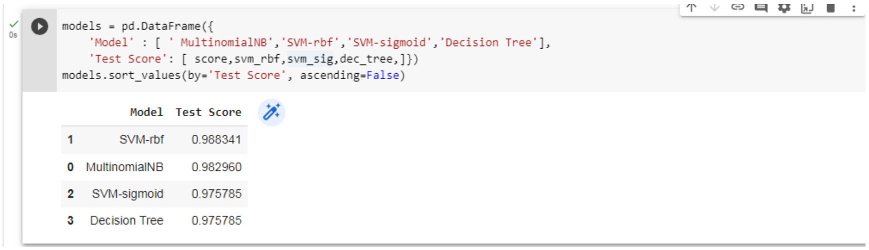


```
[33] from sklearn.tree import DecisionTreeClassifier
     dt=DecisionTreeClassifier()
     dt.fit(x_train,y_train)
```

Double-click (or enter) to edit

```
y_pred6=dt.predict(x_test)
from sklearn.metrics import accuracy_score
dec_tree=accuracy_score(y_test,y_pred6)
dec_tree

0.9757847533632287
```



```
models = pd.DataFrame({
    'Model' : [ ' MultinomialNB','SVM-rbf','SVM-sigmoid','Decision Tree'],
    'Test Score': [ score,svm_rbf,svm_sig,dec_tree,]})
models.sort_values(by='Test Score', ascending=False)
```

	Model	Test Score
1	SVM-rbf	0.988341
0	MultinomialNB	0.982960
2	SVM-sigmoid	0.975785
3	Decision Tree	0.975785

4. ADVANTAGES:

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

- Spam emails are almost always commercial and driven by a financial motive. Spammers try to promote and sell questionable goods, make false claims and deceive recipients into believing something that's not true. The most popular spam subjects include the following: pharmaceuticals.

- It has a broadcasted, rather than targeted, message. It suits the purposes of the sender rather than the receiver. Most important, the message is distributed without the explicit permission of the recipients.

- Share data more easily and efficiently

DISADVANTAGES:

- Unsolicated commercial email spam
- Impinges on the privacy of individual internet users
- Time consuming reading and deleting the messages
- Spam is a violation of Internet etiquette
- Thousands of spam emails may reach inboxes before a spammer’s email address,IP or domain is blacklisted.
- Spam filtering is machine based so there is a room for mistakes called false positives
- Filters are cumbersome to disable and to override

5.APPLICATIONS:

In machine learning, spam filtering protocols use instance-based or memory-based learning methods to identify and classify incoming spam emails based on their resemblance to stored training examples of spam emails.

For this project create two HTML files namely

- index.html
- spam.html
- result.html

6. CONCLUSION:

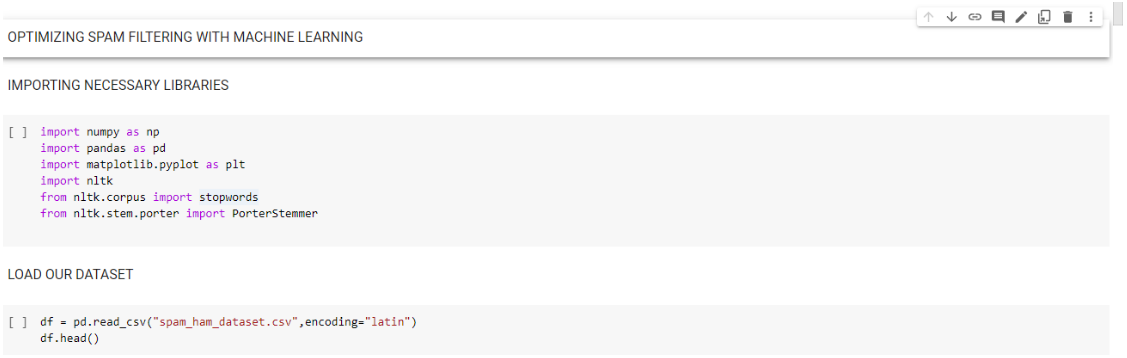
For many companies and individuals, spam is an annoyance and undesired expense. Many products and services are available to help avoid spam. Only by using these tools can we help to stem the tide of the ever-increasing unsolicited e-mails that reach our inboxes every day.

7. FUTURE SCOPE:

The algorithms developed so far have not been able to remove the requirement of manual checking of the reviews. Hence there is scope for complete automation of spam detection systems with maximum efficiency. With grow- ing popularity of online stores, the competition also increases.

8.APPENDIX:

A.SOURCE CODE:



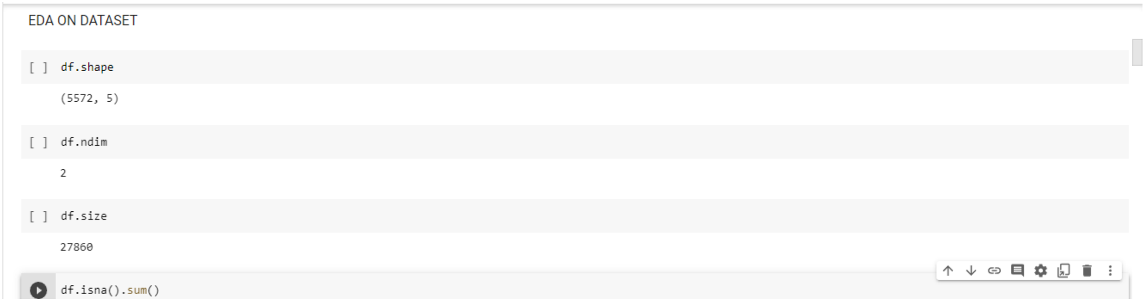
OPTIMIZING SPAM FILTERING WITH MACHINE LEARNING

IMPORTING NECESSARY LIBRARIES

```
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
```

LOAD OUR DATASET

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



EDA ON DATASET

```
df.shape
```

```
(5572, 5)
```

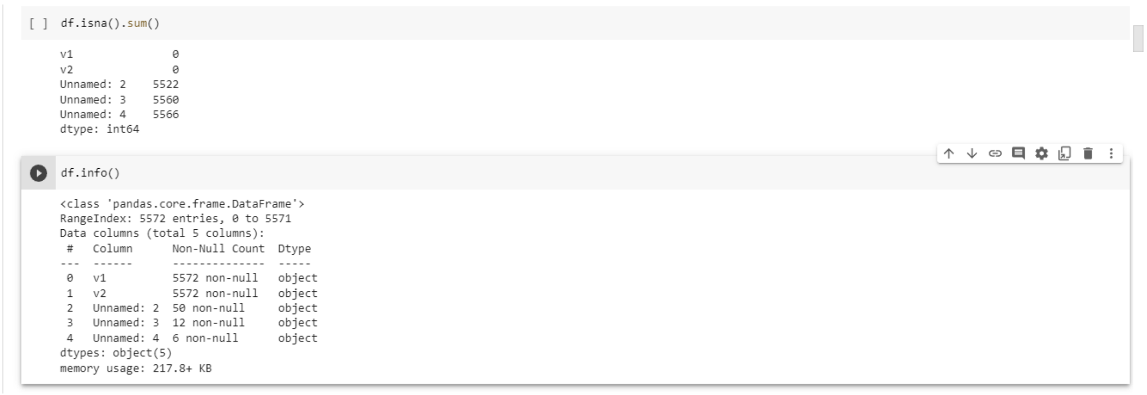
```
df.ndim
```

```
2
```

```
df.size
```

```
27860
```

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

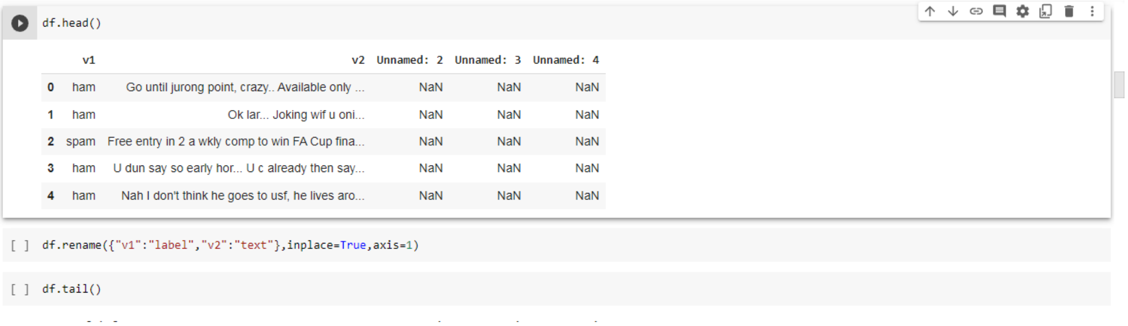


```
df.isna().sum()
```

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

```
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
```

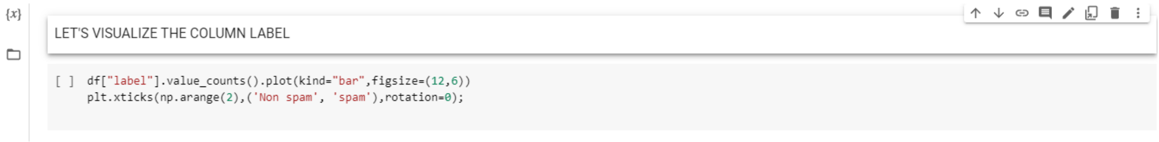


```
df.head()
```

	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

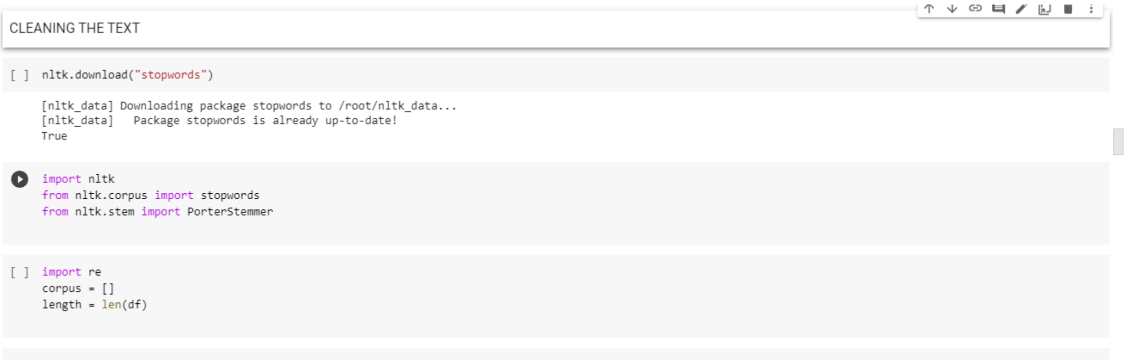
```
[ ] df.rename({"v1":"label","v2":"text"},inplace=True,axis=1)
```

```
[ ] df.tail()
```



LET'S VISUALIZE THE COLUMN LABEL

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



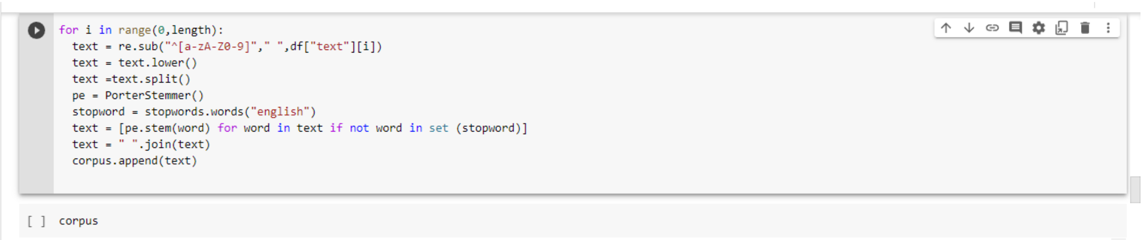
CLEANING THE TEXT

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

    [nltk_data] Downloading package stopwords to /root/nltk_data...
    [nltk_data]   Package stopwords is already up-to-date!
    True
```

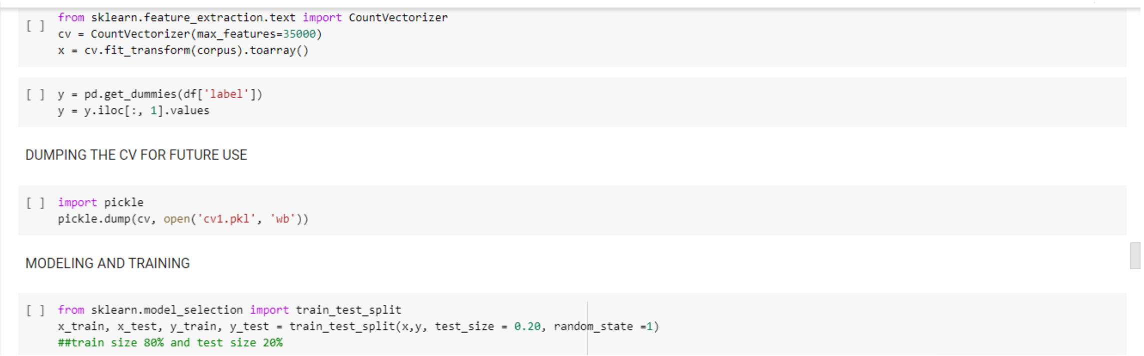
```
import nltk
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
```

```
[ ] 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()
  pe = PorterStemmer()
  stopword = stopwords.words("english")
  text = [pe.stem(word) for word in text if not word in set (stopword)]
  text = " ".join(text)
  corpus.append(text)
```

```
[ ] corpus
```



```
[ ]  from sklearn.feature_extraction.text import CountVectorizer
     cv = CountVectorizer(max_features=35000)
     x = cv.fit_transform(corpus).toarray()
```

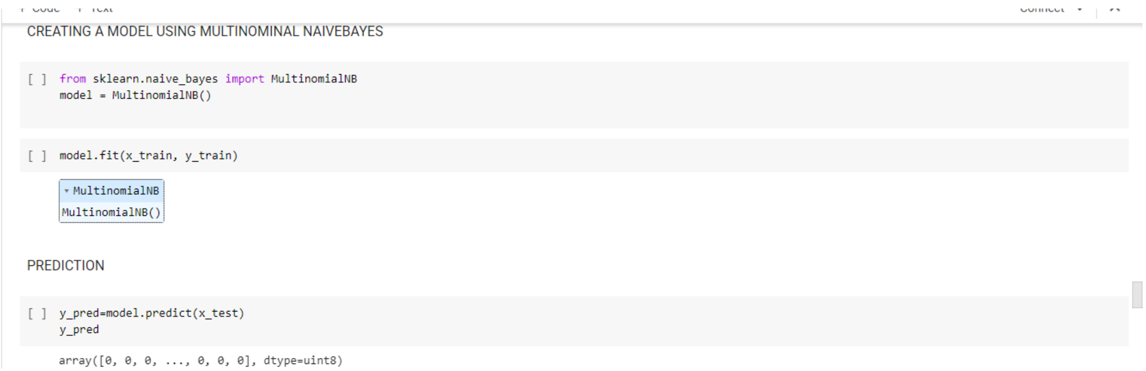
```
[ ]  y = pd.get_dummies(df['label'])
     y = y.iloc[:, 1].values
```

DUMPING THE CV FOR FUTURE USE

```
[ ]  import pickle
     pickle.dump(cv, open('cv1.pkl', 'wb'))
```

MODELING AND TRAINING

```
[ ]  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 =1)
     ##train size 80% and test size 20%
```



CREATING A MODEL USING MULTINOMINAL NAIVEBAYES

```
[ ]  from sklearn.naive_bayes import MultinomialNB
     model = MultinomialNB()
```

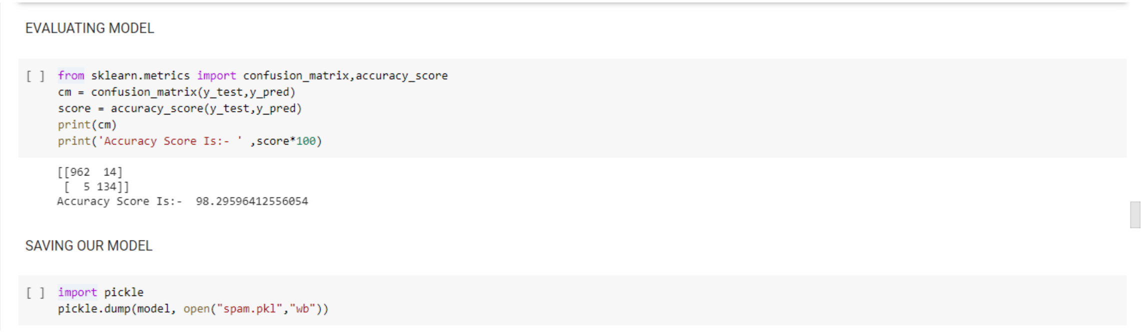
```
[ ]  model.fit(x_train, y_train)
```

```
▾ MultinomialNB
MultinomialNB()
```

PREDICTION

```
[ ]  y_pred=model.predict(x_test)
     y_pred
```

```
array([0, 0, 0, ..., 0, 0, 0], dtype=uint8)
```



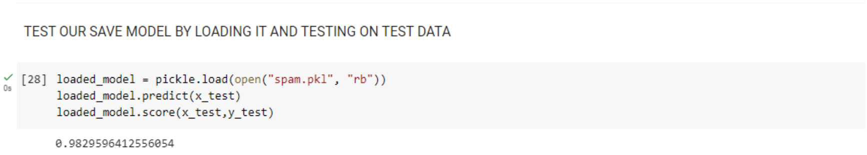
EVALUATING MODEL

```
[ ]  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:- ' ,score*100)
```

```
[[962  14]
 [  5 134]]
Accuracy Score Is:-  98.29596412556054
```

SAVING OUR MODEL

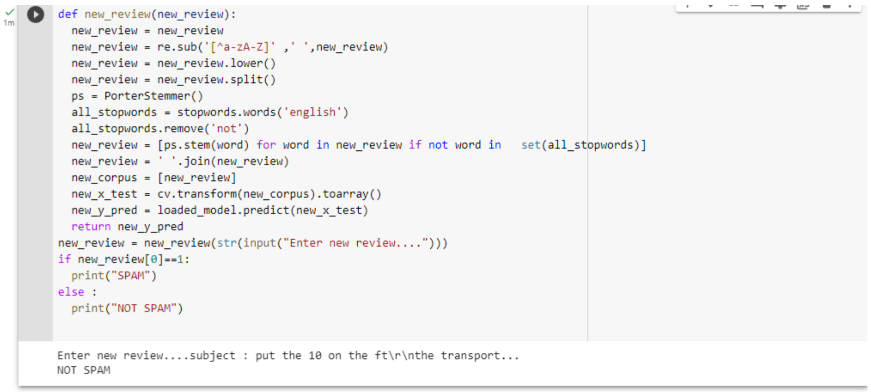
```
[ ]  import pickle
     pickle.dump(model, open("spam.pkl","wb"))
```



TEST OUR SAVE MODEL BY LOADING IT AND TESTING ON TEST DATA

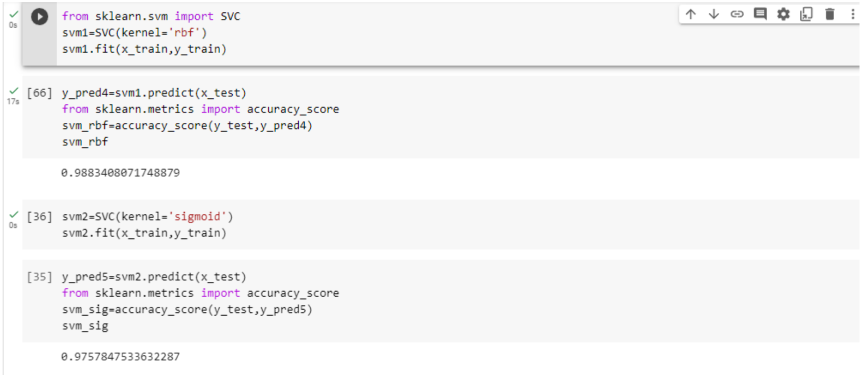
```
[28] loaded_model = pickle.load(open("spam.pkl", "rb"))
     loaded_model.predict(x_test)
     loaded_model.score(x_test,y_test)
```

```
0.9829596412556054
```



```
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()
  new_y_pred = loaded_model.predict(new_x_test)
  return new_y_pred
new_review = new_review(str(input("Enter new review...."))) 
if new_review[0]==1:
  print("SPAM")
else :
  print("NOT SPAM")
```

```
Enter new review....subject : put the 10 on the ft\r\nthe transport...
NOT SPAM
```



```
from sklearn.svm import SVC
svm1=SVC(kernel='rbf')
svm1.fit(x_train,y_train)
```

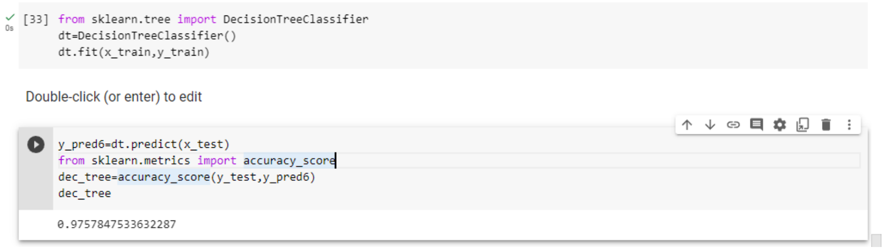
```
[66] y_pred4=svm1.predict(x_test)
     from sklearn.metrics import accuracy_score
     svm_rbf=accuracy_score(y_test,y_pred4)
     svm_rbf

     0.9883408071748879
```

```
[36] svm2=SVC(kernel='sigmoid')
     svm2.fit(x_train,y_train)
```

```
[35] y_pred5=svm2.predict(x_test)
     from sklearn.metrics import accuracy_score
     svm_sig=accuracy_score(y_test,y_pred5)
     svm_sig

     0.9757847533632287
```

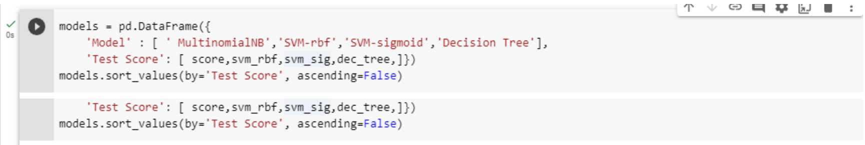


```
[33] from sklearn.tree import DecisionTreeClassifier
     dt=DecisionTreeClassifier()
     dt.fit(x_train,y_train)
```

Double-click (or enter) to edit

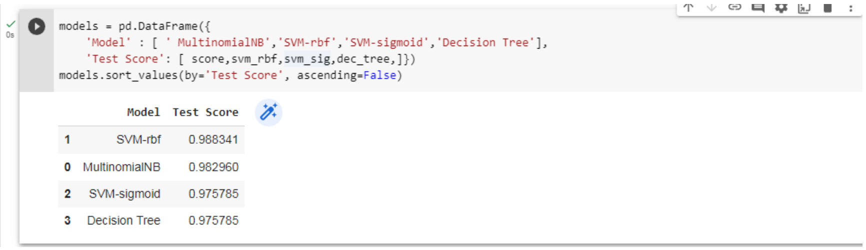
```
y_pred6=dt.predict(x_test)
from sklearn.metrics import accuracy_score
dec_tree=accuracy_score(y_test,y_pred6)
dec_tree

0.9757847533632287
```



```
models = pd.DataFrame({
    'Model' : [ ' MultinomialNB','SVM-rbf','SVM-sigmoid','Decision Tree'],
    'Test Score': [ score,svm_rbf,svm_sig,dec_tree,]})
models.sort_values(by='Test Score', ascending=False)
```

```
    'Test Score': [ score,svm_rbf,svm_sig,dec_tree,]})
models.sort_values(by='Test Score', ascending=False)
```



```
models = pd.DataFrame({
    'Model' : [ ' MultinomialNB','SVM-rbf','SVM-sigmoid','Decision Tree'],
    'Test Score': [ score,svm_rbf,svm_sig,dec_tree,]})
models.sort_values(by='Test Score', ascending=False)
```

	Model	Test Score
1	SVM-rbf	0.988341
0	MultinomialNB	0.982960
2	SVM-sigmoid	0.975785
3	Decision Tree	0.975785