OUTPUT

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)
```

```
[ ] df.ndim
```

```
2
```

```
[ ] df.size
```

```
27860
```

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

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

```
[ ] 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
```

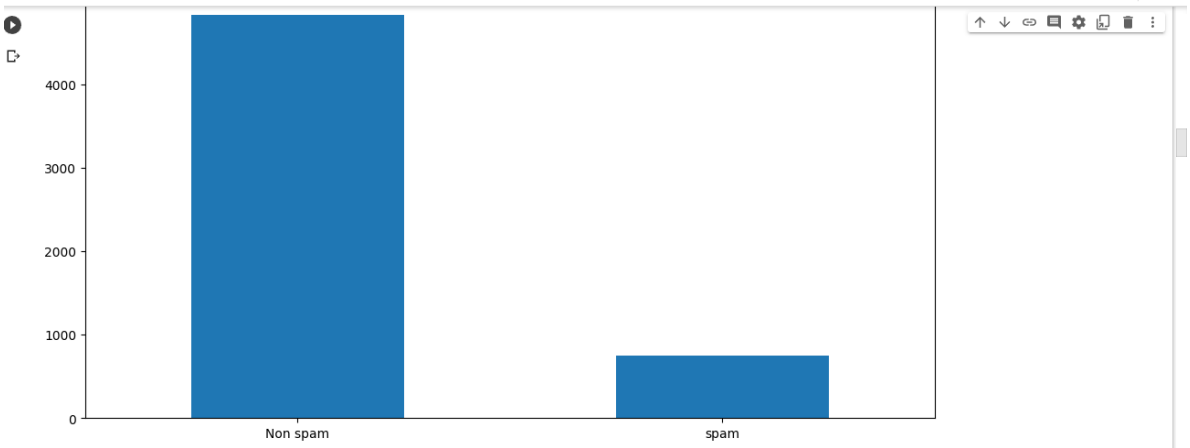
```
df.head()
```

	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()
```

	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 İ_ 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



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
```

```
['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 08452810075over18'",
 'dun say earli hor... u c alreadi say...',
 'ah think goe usf, live around though',
 "reemsg hey darl 3 week' word back! i'd like fun still? tb ok! xxx std chg send, å£1.50 rcv",
 'ven brother like speak me. treat like aid patent.',
 "per request 'mell mell (oru minnaminungint nurungu vettam)' set callertun callers. press *9 copi friend callertun",
 'inner!! valu network custom select receivea å£900 prize reward! claim call 09061701461. claim code kl341. valid 12 hour only.',
 'ad mobil 11 month more? u r entitl updat latest colour mobil camera free! call mobil updat co free 08002986030',
 "'m gonna home soon want talk stuff anymor tonight, k? i'v cri enough today.",
 'ix chanc win cash! 100 20,000 pound txt> csh11 send 87575. cost 150p/day, 6days, 16+ tsandc appli repli hl 4 info',
 'rgent! 1 week free membership å£100,000 prize jackpot! txt word: claim no: 81010 t&c www.dbuk.net lccltd pobox 4403ldnw1a7rw18',
 "'ve search right word thank breather. promis wont take help grant fulfil promise. wonder bless times.",
 'date sunday will!!',
 'xxmobilemovieclub: use credit, click wap link next txt messag click here>> http://wap. xxxmobilemovieclub.com?n=qjkgighjjgcbl',
 "h k...i'm watch here:)",
 'h u rememb 2 spell name... ye did. v naughti make v wet.',
 'ine thatåð way u feel. thatåð way gota b',
 'ngland v macedonia - dont miss goals/team news. txt ur nation team 87077 eg england 87077 try:wales, scotland 4txt/ì¼1.20 poboxox36504w45wq 16+',
 'serious spell name?',
 '\x89û÷m go tri 2 month ha ha joke',
 'ì_ pay first lar... da stock comin...',
 'ft finish lunch go str lor. ard 3 smth lor. u finish ur lunch already?',
 'fffffffff. alright way meet sooner?',
```

CREATING A MODEL USING MULTINOMINAL NAIVEBAYES

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

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

```
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)
```

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