```
#importing important libraries
import pandas as pd
```

```
# reading the dataset
df = pd.read_csv("/content/spam_ham_dataset.csv")
```

```
#finding the first five rows
df.head()
```

	Unnamed: 0	label	text	label_num
0	605	ham	Subject: enron methanol ; meter # : 988291\r\n...	0
1	2349	ham	Subject: hpl nom for january 9 , 2001\r\n(see...	0
2	3624	ham	Subject: neon retreat\r\nho ho ho , we ' re ar...	0
3	4685	spam	Subject: photoshop , windows , office . cheap ...	1
4	2030	ham	Subject: re : indian springs\r\nthis deal is t...	0

```
#dropping the unnecessary columns
df = df.drop(['Unnamed: 0','label_num'], axis=1)
```

```
#importing nltk and re libraries

import nltk
import re
```

```
#from nltk downloading stopwords

nltk.download('stopwords')
```

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

```
#finding the columns in given dataset
df.columns = ['lable', 'messages']
```

```
#finding the first 5 rows of dataset after removing unnecessary columns
df.head()
```

	lable	messages
0	ham	Subject: enron methanol ; meter # : 988291\r\n...
1	ham	Subject: hpl nom for january 9 , 2001\r\n(see...
2	ham	Subject: neon retreat\r\nho ho ho , we ' re ar...
3	spam	Subject: photoshop , windows , office . cheap ...
4	ham	Subject: re : indian springs\r\nthis deal is t...

```
df['messages'][0]
```

```
'Subject: enron methanol ; meter # : 988291\r\nthis is a follow up to the note i gave y
ou on monday , 4 / 3 / 00 { preliminary\r\nflow data provided by daren } .\r\nplease ov
erride pop ' s daily volume { presently zero } to reflect daily\r\nactivity you can obt
ain from gas control . \r\nthis change is needed asap for economics purposes . '
```

```
#importing stopwords from nltk.corpus
from nltk.corpus import stopwords
```

```
# importing porterstemmer from nltk.stem.porter
from nltk.stem.porter import PorterStemmer
```

```
ps = PorterStemmer()
```

```
df['messages'][30]
```

```
df['messages'][30]
```

```
'Subject: the houston expl dec 2000\r\ndarren :\r\nelizabeth hernandez fixed this deal
for me . i don ' t need for you to look into it . thanks anyway .\r\n- - - - - - - - -
- - - - - - - - - - - - - - forwarded by megan parker / corp / enron on 07 / 27 / 2001 01
: 46 pm - - - - - - - - - - - - - - - - - - - - - - - - - - -\r\nfrom : megan parker 07
/ 19 / 2001 10 : 27 am\r\nto : daren j farmer / enron @ enronxgate\r\ncc :\r\nsubject :
the houston expl dec 2000\r\ndaren :\r\ni ' m not sure if you can help me , but i have
a danny conner deal from december 2000 that has a price issue .\r\nwe were buying gas f
rom the houston exploration company on black marlin , high island 138 / hpl meter 98663
```

```
rev = re.sub('[^a-zA-Z]',' ', df['messages'][30])
```

```
rev
```

```
'Subject  the houston expl dec       darren    elizabeth hernandez fixed this deal for
me   i don   t need for you to look into it   thanks anyway
forwarded by megan parker   corp   enron on                        pm
from   megan parker                         am  to   daren j farmer   enron   enronxgate
cc    subject   the houston expl dec       daren    i   m not sure if you can help me
but i have a danny conner deal from december      that has a price issue    we were buy
ing gas from the houston exploration company on black marlin   high island       hpl me
```

```
rev = rev.lower()
```

```
rev
```

```
'subject  the houston expl dec       darren    elizabeth hernandez fixed this deal for
me   i don   t need for you to look into it   thanks anyway
forwarded by megan parker   corp   enron on                        pm
from   megan parker                         am  to   daren j farmer   enron   enronxgate
cc    subject   the houston expl dec       daren    i   m not sure if you can help me
but i have a danny conner deal from december      that has a price issue    we were buy
ing gas from the houston exploration company on black marlin   high island       hpl me
```

```
rev = [ps.stem(word) for word in rev if not word in stopwords.words('english')]
```

```
rev
```

```
['subjectimportantonlinebankingalertdearvaluedcitizensrbankmemberduetoconcernsforthesafetyandintegrityoftheonlinebankingcommunitywehavei
```

```
rev = ''.join(rev)
```

```
rev
```

```
's    u    b    j    e    c    t    i    m    p    o    r    t    a    n
t    o    n    l    i    n    e    b    a    n    k    i    n    g    a
l    e    r    t    d    e    a    r    v    a    l    u    e    d    c
i    t    i    z    e    n    s    r    b    a    n    k    m    e    m
b    e    r    d    u    e    t    o    c    o    n    c    e    r    n
s    f    o    r    t    h    e    s    a    f    e    t    y    a    n
d    i    n    t    e    g    r    i    t    y    o    f    t    h    e
o    n    l    i    n    e    b    a    n    k    i    n    g    c    o
```

```
#writing the function
corpus=[]
for i in range(0,len(df)):
  review = re.sub('[^a-zA-Z]','', df['messages'][i])  #replacing unnecessary symbols with space
  review = review.lower()  # lowering the review column
  review = review.split()  #splitting the data
  review =[ps.stem(word) for word in review if not word in stopwords.words('english')]
  review = ' '.join(review)
  corpus.append(review)
```

```
corpus
```

```
['subjectenronmethanolmeterthisisafollowuptothenoteigaveyouonmondaypreliminaryflowdataprovidedbydarenpleaseoverridepopsdailyvolumepres
 'subjecthplnomforjanuaryseeattachedfilehplnolxlshplnolxl',

 'subjectneonretreathohohowerearoundtothatmostwonderfultimeoftheyearneonleadersretreattimeiknowthatthistimeofyearisextremelyhecticandth

 'subjectphotoshopwindowsofficecheapmaintrendingabasementsdarerprudentlyfortuitousundergonelightheartedcharmorinocotasterrailroadafflue

 'subjectreindianspringsthisdealistobookthetecopvrrevenueitismyunderstandingthattecojustsendsusacheckihaventreceivedananswerastowhethe
 'subjectehronlinewebaddresschangethismessageisintendedforehronlineusersonlyduetoarecentchangetoehronlinetheurlakawebaddressforaccessi

 'subjectspringsavingscertificatetakeoffsavewhenyouuseourcustomerappreciationspringsavingscertificateatfootlockerladyfootlockerkidsfoo

 'subjectlookingformedicationwerethebestsourceitisdifficulttomakeourmaterialconditionbetterbythebestlawbutitiseasyenoughtoruinitbybadl

 'subjectnomsactualflowforweagreeforwardedbymelissajonestexasutilitiesonameileenpontononamtodavidavilalspenserchustucharliestonetexasu
 'subjectnominationsforoctseeattachedfilehplnlxlshplnlxl',

 'subjectvocablerndwordasceticismvcscbrandnewstockforyourattentionvocalscapeincthestocksymbolisvcscvcscwillbeourtopstockpickforthemont

 'subjectreportwffurattionbromestinstsiupiedpgstourriweasentlyresttonttopresyoutewconsofbencoyeefateryoustlyughtatumsandinencedsorepit
 'subjectenronhplactualsforaugusttecotapenronhplgasdailylshpllskicenron',

 'subjectvicodinnowbernehotboxcarnalbridecutwormdyadicguardiacontinuousborngremlinakincounterflowhereaftervocabularianpessimumyaoundec
 'subjecttenaskaivjulydarrenpleaseremovethepriceonthetenaskaivsaledealforjulyandenterthedemandfeetheamountshouldbethanksmegan',

 'subjectunderpricedissuewithhighreturnonequitystockreportdontsieeponthisstockthisisahotonecompanygamingtransactionsincstocksymbolggts

 'subjectrefirstdeliverywheeleroperatingvancedealhasbeencreatedandenteredinsitarabobvanceltaylorpmtorobertcottenhouectectccjuliemeyers

 'subjectswiftmayvolsseanfyicheckthepurchasefromswiftatthetailgatemeterandmakesuretonomthecorrectquantitymaryforwardedbymarypoormannae

 'subjectmetervariancesuacleanupdarenvancethetwometersbelowarenewandhaveunallocatableflowiwillneedapurchaseforeachofthempleaserespondw

 'subjectadditionalrecruitingimhappytointroducemollymageeasthenewestadditiontotheeopsrecruitingteamtoniandmollyhavedividedtheirrecruit

 'subjectfwercotloadcomparisonoriginalmessagefromgilbertsmithdougsenttuesdaymayamtotmartinenroncomsubjectercotloadcomparisontomhereisa

 'subjectmeterconcordechurchilloneyearrateforthisonewillbemmforvolumesgreaterthanmmdaypriceforvolumesmmdayorlesswillbemmplusapermonthm
 'subjecthplnomforjanuaryseeattachedfilehplnolxlshplnolxl',
 'subjectretenaskaivwehavereceivedallofthemoneyfromthespotsalesfortenaskaivinoctoberexceptforthetenaskaivsaleandthefeemegan',

 'subjectjumpintogainsubstantialgroundimmediatelyweareveryexcitedaboutthisnewupcomingstockabouttoexplodemontanaoilandgasincmogitoexplo

 'subjectreenronhplactualsforoctoberrevisionpleasenotethatthepricingallocationofvolumesforoctobershouldbechangedasfollowstecotapenronh

 'subjectregistrationconfirmationfromspinnercomthankyouforjoiningspinnercomthewebslargestsourceoffreestreamingmusicjustwantedtoconfirm

 'subjectaeptransitionitemsattachedisabriefmemooutlinesomeofthetranstionissueswithhpltoaepthisisthefirstdrafttheitilizeditemscurrently

 'subjectaninboundmessageforyouhasbeenquarantinedyouhavereceivedthismessagebecausesomeonehasattemptedtosendyouanemailfromoutsideofenro

 'subjectrevalerogasmarketingmetersitaraticketpleasezeroouttheVolumesuntilfurthernoticetheplantisscheduledtocomeuponmarchandwewilltrea

 'subjectthehoustonexpldecdarrenelizabethhernandezfixedthisdealformeidontneedforyoutolookintoitthanksanywayforwardedbymeganparkercorpe
 'subjectenronhplactualsforoctobertecotapenronhpliferclshpllskichpliferc',
```

```
stopwords.words('english')
```

```
['i',
 'me',
 'my',
 'myself',
 'we',
 'our',
 'ours',
 'ourselves',
 'you',
 "you're",
 "you've",
 "you'll",
 "you'd",
 'your',
 'yours',
 'yourself',
 'yourselves',
 'he',
 'him',
 'his',
 'himself',
 'she',
 "she's",
 'her',
 'hers',
 'herself',
 'it',
 "it's",
 'its',
 'itself',
 'they',
 'them',
 'their',
 'theirs',
 'themselves',
 'what',
 'which',
 'who',
 'whom',
 'this',
 'that',
 "that'll",
 'these',
 'those',
 'am',
 'is',
 'are',
 'was',
 'were',
 'be',
 'been',
 'being',
 'have',
 'has',
 'had',
 'having',
 'do',
 'does',
```

```
corpus[0]
```

```
     'subjectenronmethanolmeterthisisafollowuptothenoteigaveyouonmondaypreliminaryflowdatapr
     ovidedbydarenpleaseoverridepopsdailyvolumepresentlyzerotoreflectdailyactivityyoucanobta
     infromgascontrolthischangeisneededasanforeconomicspurpos'
```

```
#importing the countvectorizer from sklearn
from sklearn.feature_extraction.text import CountVectorizer
```

```
#since we are having more features in our data, so we are presizing to 2500
cv = CountVectorizer(max_features = 2500)
```

```
# creating the x variable
x = cv.fit_transform(corpus).toarray()
```

```
x[0]
```

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

```
# finding the shape of x variable
x.shape
```

```
     (5171, 2500)
```

```
df['lable']
```

```
     0        ham
     1        ham
     2        ham
     3       spam
     4        ham
             ...
     5166     ham
     5167     ham
     5168     ham
     5169     ham
     5170    spam
     Name: lable, Length: 5171, dtype: object
```

```
#creating the y variable by using dummies
y = pd.get_dummies(df['lable'], drop_first= True)
```

```
x
```

```
     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]])
```

```
y
```

	spam
0	0
1	0
2	0
3	1
4	0
...	...
5166	0
5167	0
5168	0
5169	0
5170	1

5171 rows × 1 columns

```
#train test spilit from sklearn
from sklearn.model_selection import train_test_split
```

```
x_train, x_test, y_train, y_test = train_test_split(
...     x, y, test_size=0.33, random_state=42)
```

```
#model building
#here we are creating the model of Gaussian NB from sklearn naive bayes

from sklearn.naive_bayes import GaussianNB
```

```
model1 = GaussianNB()
```

```
#fitting our model in to our dataset

model1.fit(x_train, y_train)
```

```
/usr/local/lib/python3.8/dist-packages/sklearn/utils/validation.py:993: DataConversionWarning: A column-vector y was passed when a 1d ar
  y = column_or_1d(y, warn=True)
GaussianNB()
```

```
#predicting the y_test.
y_pre1 = model1.predict(x_test)
```

```
#importing accuracy_score from sklearn metrics

from sklearn.metrics import accuracy_score
```

```
#predicted value
GS = accuracy_score(y_test, y_pre1)
```

```
GS
```

```
0.3942589338019918
```

```
# from gaussian naive bayes, we got 39%, which is low. so we can check with another model accuracy
# building the another model MultinomialNB
>>> from sklearn.naive_bayes import MultinomialNB
```

```
model2 = MultinomialNB()
```

```
# fitting out model in to MultinomialNB
model2.fit(x_train, y_train)
```

```
    /usr/local/lib/python3.8/dist-packages/sklearn/utils/validation.py:993: DataConversionWarning: A column-vector y was passed when a 1d ar
      y = column_or_1d(y, warn=True)
    MultinomialNB()
```

```
#predicted the y_test
y_pred2 = model2.predict(x_test)
```

```
#finding the accuracy score of multinomial NB
MGNS = accuracy_score(y_test, y_pred2)
```

```
MGNS
```

```
    0.7387229056824839
```

```
#observations: Here we got 73% of accuracy

#Conclusions:After comparing the above 2 models, we can choose Multinomial NB is the best model for our dataset.
```