In [155]:
```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

In [156]:
```
# loading the data from csv file too a panda Dataframe
raw_mail_data = pd.read_csv(r"C:\Users\Kamal Kant\OneDrive\Documents\spam_ham_c
```

In [157]:
```
raw_mail_data.head()
```

Out[157]:

	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

In [158]:
```
print(raw_mail_data)
```

```
      Unnamed: 0 label                                               text  \
0            605   ham  Subject: enron methanol ; meter # : 988291\r\n...
1           2349   ham  Subject: hpl nom for january 9 , 2001\r\n( see...
2           3624   ham  Subject: neon retreat\r\nho ho ho , we ' re ar...
3           4685  spam  Subject: photoshop , windows , office . cheap ...
4           2030   ham  Subject: re : indian springs\r\nthis deal is t...
...          ...   ...                                                ...
5166        1518   ham  Subject: put the 10 on the ft\r\nthe transport...
5167         404   ham  Subject: 3 / 4 / 2000 and following noms\r\nhp...
5168        2933   ham  Subject: calpine daily gas nomination\r\n>\r\n...
5169        1409   ham  Subject: industrial worksheets for august 2000...
5170        4807  spam  Subject: important online banking alert\r\ndea...

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

[5171 rows x 4 columns]
```

In [159]:
```
#replace the null values with null string
mail_data = raw_mail_data.where((pd.notnull(raw_mail_data)),'')
```

In [160]:
```
mail_data.head()
```

Out[160]:

	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

In [161]:
```
# checking the number of columns in data frame
mail_data.shape
```

Out[161]:
```
(5171, 4)
```

In [162]:
```
# label spam mail as 0: ham mail as 1:
mail_data.loc[mail_data['label'] == 'spam', 'label',] = 0
mail_data.loc[mail_data['label'] == 'ham', 'label',] = 1
```

In [163]:
```
# seprating the dat as text and labels
x = mail_data['text']
y = mail_data['label']
```

In [164]:
```
print(x)
```

```
0       Subject: enron methanol ; meter # : 988291\r\n...
1       Subject: hpl nom for january 9 , 2001\r\n( see...
2       Subject: neon retreat\r\nho ho ho , we ' re ar...
3       Subject: photoshop , windows , office . cheap ...
4       Subject: re : indian springs\r\nthis deal is t...
                              ...
5166    Subject: put the 10 on the ft\r\nthe transport...
5167    Subject: 3 / 4 / 2000 and following noms\r\nhp...
5168    Subject: calpine daily gas nomination\r\n>\r\n...
5169    Subject: industrial worksheets for august 2000...
5170    Subject: important online banking alert\r\ndea...
Name: text, Length: 5171, dtype: object
```

In [165]:
```
print(y)
```

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

In [166]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random
```

In [167]:
```
print(x.shape)
print(x_train.shape)
print(x_test.shape)
```

```
(5171,)
(4136,)
(1035,)
```

In [168]:
```
# transform the text data to feature vectors that can be use as input
feature_extraction = TfidfVectorizer(min_df = 1)

x_train_features = feature_extraction.fit_transform(x_train)
x_test_features = feature_extraction.transform(x_test)

# convert the value of y_train and y_test into integers
y_train = y_train.astype('int')
y_test = y_test.astype('int')
```

In [169]:
```
print(x_train)
```

```
1667    Subject: neon for march 28\r\nhere is the neon...
1951    Subject: hpl nom for february 1 , 2001\r\n( se...
4659    Subject: enron / hpl actuals for july 20 , 200...
4339    Subject: natural gas nomination for 04 / 01\r\...
2264    Subject: first delivery - wheeler operating\r\...
                              ...
3335    Subject: \r\nto _ cc _ default _ handler\r\nsu...
1099    Subject: s 709101 - 04 / 03 / 01\r\ndaren , bp...
2514    Subject: viagra _ cialis _ levitra _ ambien _ ...
3606    Subject: panenergy marketing march 2000 produc...
2575    Subject: important information about united he...
Name: text, Length: 4136, dtype: object
```

In [170]:
```
print(x_train_features)
```

```
  (0, 15397)    0.220130809090187
  (0, 43560)    0.18242554738893527
  (0, 20304)    0.2732677361768187
  (0, 17754)    0.2989092821232026
  (0, 40392)    0.1610302770559178
  (0, 26081)    0.3012303644207342
  (0, 40425)    0.06493500764738999
  (0, 23890)    0.0828929050888299
  (0, 21570)    0.1398241512858935
  (0, 1411)     0.3587839879514815
  (0, 27331)    0.3709133678348265
  (0, 18952)    0.1294403145442609
  (0, 29565)    0.5671081282538426
  (0, 39266)    0.047572750308263516
  (1, 44560)    0.3590416750320136
  (1, 1016)     0.6204359599798661
  (1, 22204)    0.5289113861442295
  (1, 18472)    0.1738880806400928
  (1, 6842)     0.14832769661486625
  (1, 36981)    0.1415516303443518
  (1, 995)      0.15125104806276704
  (1, 18216)    0.23527225253071396
  (1, 29926)    0.17133984042638714
  (1, 22198)    0.13590590758074877
  (1, 18952)    0.07237506166757512
  :     :
  (4135, 21272) 0.06861374865350543
  (4135, 30862) 0.08192987881332453
  (4135, 31027) 0.03976451177749149
  (4135, 5554)  0.03723701344200261
  (4135, 30708) 0.018134197869227795
  (4135, 9745)  0.0371187526687948
  (4135, 5836)  0.09353819288156556
  (4135, 8123)  0.051332134814522465
  (4135, 6383)  0.025322337224710008
  (4135, 40872) 0.14199449606248007
  (4135, 14105) 0.025179738267686356
  (4135, 19289) 0.046521746060971876
  (4135, 40537) 0.08554262782784067
  (4135, 13558) 0.026481238834695636
  (4135, 32566) 0.023375726025729283
  (4135, 6776)  0.060760767216537524
  (4135, 240)   0.01490015250568432
  (4135, 992)   0.012902486788303264
  (4135, 16964) 0.06596966567544651
  (4135, 995)   0.01658252921375621
  (4135, 40425) 0.12737982264917766
  (4135, 23890) 0.05081467191023478
  (4135, 1411)  0.021994030266960587
  (4135, 18952) 0.023804692724216793
  (4135, 39266) 0.0058325708242253005
```

In [171]:
```
# Traning model
# Logistic Regression
```

In [172]:
```
model = LogisticRegression()
model.fit(x_train_features, y_train)
```

Out[172]:
```
LogisticRegression()
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [173]:
```
# Evaluating on traning model

# prediction on traning model

prediction_on_training_data = model.predict(x_train_features)
accuracy_on_traning_data = accuracy_score(y_train, prediction_on_training_data)
```

In [174]:
```
print('Accuracy on traning data : ', accuracy_on_traning_data)
```

```
Accuracy on traning data :  0.9944390715667312
```

In [177]:
```
# prediction on test data
prediction_on_test_data = model.predict(x_test_features)
accuracy_on_test_data = accuracy_score(y_test, prediction_on_test_data)
```

In [178]:
```
print('Accuracy on test data : ', accuracy_on_test_data)
```

```
Accuracy on test data :  0.9845410628019323
```

In [179]:
```
# building a predictive system
input_mail = ["i have been searching for the right text to thank you for this b
input_data_features = feature_extraction.transform(input_mail)
prediction = model.predict(input_data_features)
print(prediction)
if (prediction[0]==1):
    print('ham mail')
else:
    print('spam mail')
```

```
[1]
ham mail
```

In [ ]:

In [ ]: