IMPORTING THE DEPENDENCIES OR LIBRARIES

In [1]:
```
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
```

-> Here, we have used these libraries for spam mail detection

- We have used NUMPY for creating an array.
- PANDAS is used for creating a data frame like load_data and mail_data. Pandas is used to put the data in a structured data frame.
- We have imported TRAIN_TEST_SPLIT function which is used to split data into training and test data
  - Where training data is used to train model, and,
  - Test data is used to evaluate model.
- We have imported TFIDFVECTORIZER, in order to transform the text into numeric values.
- We have imported a LOGISTIC REGRESSION MODEL,
- ACCURACY SCORE MODEL is used to find that how accurate our model is.

DATA COLLECTION & PRE-PROCESSING

In [2]:
```
# LOADING THE DATA FROM CSV FILE TO A PANDAS DATAFRAME

load_data = pd.read_csv(r'C:\Users\HP\OneDrive\Desktop\mail_prediction_project
```

In [3]:
```
print(load_data)
```

```
     Category                                            Message
0         ham  Go until jurong point, crazy.. Available only ...
1         ham                      Ok lar... Joking wif u oni...
2        spam  Free entry in 2 a wkly comp to win FA Cup fina...
3         ham  U dun say so early hor... U c already then say...
4         ham  Nah I don't think he goes to usf, he lives aro...
...       ...                                                ...
5567     spam  This is the 2nd time we have tried 2 contact u...
5568      ham               Will ü b going to esplanade fr home?
5569      ham  Pity, * was in mood for that. So...any other s...
5570      ham  The guy did some bitching but I acted like i'd...
5571      ham                         Rofl. Its true to its name

[5572 rows x 2 columns]
```

In [4]:
```
# REPLACE THE NULL VALUES WITH A NULL STRINGS

mail_data = load_data.where((pd.notnull(load_data)),'')
```

In [5]:
```
# PRINTING THE FIRST FIVE ROWS FROM THE DATA FRAME

mail_data.head()
```

Out[5]:

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

In [6]:
```
mail_data.tail()
```

Out[6]:

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

In [7]:
```
# CHECKING THE NUMBER OF ROWS AND COLUMNS IN DATA FRAME

mail_data.shape
```

Out[7]:
```
(5572, 2)
```

LABEL ENCODING

In [8]:
```
# label spam mail as 0; ham mail as 1;

mail_data.loc[mail_data['Category']=='spam', 'Category',]=0
mail_data.loc[mail_data['Category']=='ham', 'Category',]=1
```

spam = 0 ham = 1

In [9]:
```
#SEPARATING THE DATA AS TEXTS AND LABELS

x = mail_data['Message']

y = mail_data['Category']
```

In [10]:
```
print(x)
```

```
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...
                              ...
5567    This is the 2nd time we have tried 2 contact u...
5568                 Will ü b going to esplanade fr home?
5569    Pity, * was in mood for that. So...any other s...
5570    The guy did some bitching but I acted like i'd...
5571                           Rofl. Its true to its name
Name: Message, Length: 5572, dtype: object
```

In [11]:
```
print(y)
```

```
0       1
1       1
2       0
3       1
4       1
       ..
5567    0
5568    1
5569    1
5570    1
5571    1
Name: Category, Length: 5572, dtype: object
```

SPLITING THE DATA INTO TRAINING DATA AND TEST DATA

In [12]:
```
# for this we need to import 4 arrays..

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, rar
```

In [13]:
```
print(x.shape)
print(x_train.shape)
print(x_test.shape)
```

```
(5572,)
(4457,)
(1115,)
```

FEATURE EXTRACTION :-- In this we convert our text data into numerical

In [14]:
```
# TRANSFORM THE TEXT DATA TO FEATURE VECTORS THAT CAN BE USED AS INPUT TO THE

feature_extraction = TfidfVectorizer(min_df = 1, stop_words = 'english', lower

x_train_features = feature_extraction.fit_transform(x_train)
#THIS WILL FIT YOUR DATA and in parameter we have declared which we want to co

x_test_features = feature_extraction.transform(x_test)

# CONVERT Y_TRAIN AND Y_TEST VALUES AS INTEGERS

y_train = y_train.astype('int')
y_test = y_test.astype('int')
```

In [15]:
```
print(x_train)
```

```
3075                  Don know. I did't msg him recently.
1787    Do you know why god created gap between your f...
1614                            Thnx dude. u guys out 2nite?
4304                                         Yup i'm free...
3266    44 7732584351, Do you want a New Nokia 3510i c...
                              ...
789     5 Free Top Polyphonic Tones call 087018728737,...
968     What do u want when i come back?.a beautiful n...
1667    Guess who spent all last night phasing in and ...
3321    Eh sorry leh... I din c ur msg. Not sad alread...
1688    Free Top ringtone -sub to weekly ringtone-get ...
Name: Message, Length: 4457, dtype: object
```

In [16]:
```python
print(x_train_features)
```

```
  (0, 5413)	0.6198254967574347
  (0, 4456)	0.4168658090846482
  (0, 2224)	0.413103377943378
  (0, 3811)	0.34780165336891333
  (0, 2329)	0.38783870336935383
  (1, 4080)	0.18880584110891163
  (1, 3185)	0.29694482957694585
  (1, 3325)	0.31610586766078863
  (1, 2957)	0.3398297002864083
  (1, 2746)	0.3398297002864083
  (1, 918)	0.22871581159877646
  (1, 1839)	0.2784903590561455
  (1, 2758)	0.3226407885943799
  (1, 2956)	0.33036995955537024
  (1, 1991)	0.33036995955537024
  (1, 3046)	0.2503712792613518
  (1, 3811)	0.17419952275504033
  (2, 407)	0.509272536051008
  (2, 3156)	0.4107239318312698
  (2, 2404)	0.45287711070606745
  (2, 6601)	0.6056811524587518
  (3, 2870)	0.5864269879324768
  (3, 7414)	0.8100020912469564
  (4, 50)	0.23633754072626942
  (4, 5497)	0.15743785051118356
  :	:
  (4454, 4602)	0.2669765732445391
  (4454, 3142)	0.32014451677763156
  (4455, 2247)	0.37052851863170466
  (4455, 2469)	0.35441545511837946
  (4455, 5646)	0.33545678464631296
  (4455, 6810)	0.29731757715898277
  (4455, 6091)	0.23103841516927642
  (4455, 7113)	0.30536590342067704
  (4455, 3872)	0.3108911491788658
  (4455, 4715)	0.30714144758811196
  (4455, 6916)	0.19636985317119715
  (4455, 3922)	0.31287563163368587
  (4455, 4456)	0.24920025316220423
  (4456, 141)	0.292943737785358
  (4456, 647)	0.30133182431707617
  (4456, 6311)	0.30133182431707617
  (4456, 5569)	0.4619395404299172
  (4456, 6028)	0.21034888000987115
  (4456, 7154)	0.24083218452280053
  (4456, 7150)	0.3677554681447669
  (4456, 6249)	0.17573831794959716
  (4456, 6307)	0.2752760476857975
  (4456, 334)	0.2220077711654938
  (4456, 5778)	0.16243064490100795
  (4456, 2870)	0.31523196273113385
```

Training the Model

logistic regression model

In [17]:
```
model = LogisticRegression()
```

In [18]:
```
# TRAINING THE LOGISTIC REGRESSION MODEL WITH THE TRAINING DATA

model.fit(x_train_features, y_train)
```

Out[18]:
```
LogisticRegression()
```

Evaluating the trained Model

In [19]:
```
# PREDICTION ON TRAINING DATA

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

In [20]:
```
print('Accuracy on training data : ',accuracy_on_training_data)
```

```
Accuracy on training data :  0.9670181736594121
```

In [21]:
```
# 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 [22]:
```
print('Accuracy on test data : ', accuracy_on_test_data)
```

```
Accuracy on test data :  0.9659192825112107
```

BUILDING A PREDICTIVE SYSTEM FOR NEW MAILS

In [23]:
```
input_mail = ["U dun say so early hor... U c already then say..."]

# CONVERT TEXT TO FEATURE VECCTORS

input_data_feature = feature_extraction.transform(input_mail)

# MAKING PREDICTION

prediction = model.predict(input_data_feature)

print(prediction)

# IF IT IS HAM THEN IT SHOULD BE RETURN 1.
# ELSE IF IT IS SPAM IT SHOULD BE RETURN 0.

if (prediction[0]==1):
    print("Ham mail")

else:
    print("Spam mail")
```

```
[1]
Ham mail
```

In [24]:
```
input_mail = ["Had your mobile 11 months or more? U R entitled to Update to th

# CONVERT TEXT TO FEATURE VECCTORS

input_data_feature = feature_extraction.transform(input_mail)

# MAKING PREDICTION

prediction = model.predict(input_data_feature)

print(prediction)

# IF IT IS HAM THEN IT SHOULD BE RETURN 1.
# ELSE IF IT IS SPAM IT SHOULD BE RETURN 0.

if (prediction[0]==1):
    print("Ham mail")

else:
    print("Spam mail")
```

```
[0]
Spam mail
```