Spam Mail Prediction

Step-1: Importing the Libraries

In [13]:
```
import numpy as np
import pandas as pd

# warnings removal
import warnings
warnings.filterwarnings("ignore")


from sklearn.model_selection import train_test_split

from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from xgboost import XGBClassifier

from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
```

Step-2: Load the data set

In [14]:
```
mail_data = pd.read_csv("C:/Users/Hi/Downloads/mail_data.csv")
```

In [15]:
```
mail_data
```

Out[15]:

	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 × 2 columns

In [16]:
```
## shape

print("The num of rows (observation) is", data.shape[0], '\n', 'The num of columns (variables) is', data.shape[1]
```

```
The num of rows (observation) is 5572
 The num of columns (variables) is 2
```

In [17]:
```
mail_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 2 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   Category  5572 non-null   object
 1   Message   5572 non-null   object
dtypes: object(2)
memory usage: 87.2+ KB
```

Step-3 Preprocessing of the data

In [8]:
```
mail_data.isnull().sum()
```

Out[8]:
```
Category    0
Message     0
dtype: int64
```

In [9]:
```
# replace the null values with a null string
mail_data1 = data.where((pd.notnull(data)),'')
```

In [10]:
```
mail_data1.head()
```

Out[10]:

	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 [18]:
```
# 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
```

In [23]:
```
mail_data.Category.value_counts()
```

Out[23]:
```
1    4825
0     747
Name: Category, dtype: int64
```

In [28]:
```
# separating the data as texts and label

x = mail_data['Message']

y = mail_data['Category']
```

In [29]:
```
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 [30]:
```
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
```

In [37]:
```
# convert y values as integers
y= y.astype('int')
```

In [38]:
```
y
```

Out[38]:
```
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: int32
```

Step-4: splitting the data into Train & Test

In [39]:
```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=3,stratify = y)
```

In [40]:
```
print(x.shape)
print(x_train.shape)
print(x_test.shape)
```

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

In [41]:
```
# transform the text data to feature vectors
from sklearn.feature_extraction.text import TfidfVectorizer

feature_extraction = TfidfVectorizer(min_df=1, stop_words = 'english', lowercase=True)
x_train_features = feature_extraction.fit_transform(x_train)
x_test_features = feature_extraction.transform(x_test)
```

Step-5: Training the Model

Logistic Regression

In [45]:
```
# Create model
lr = LogisticRegression()
```

In [46]:
```
# model fit
lr.fit(x_train_features,y_train)
```

Out[46]:
```
LogisticRegression
LogisticRegression()
```

In [47]:
```
# test set predict
test_pred_lr = lr.predict(x_test_features)
# train set predict
train_pred_lr = lr.predict(x_train_features)
```

In [48]:
```
print(classification_report(y_train, train_pred_lr))
print()
print(classification_report(y_test,test_pred_lr))
```

```
              precision    recall  f1-score   support

           0       0.99      0.76      0.86       598
           1       0.96      1.00      0.98      3859

    accuracy                           0.97      4457
   macro avg       0.98      0.88      0.92      4457
weighted avg       0.97      0.97      0.97      4457


              precision    recall  f1-score   support

           0       0.99      0.80      0.88       149
           1       0.97      1.00      0.98       966

    accuracy                           0.97      1115
   macro avg       0.98      0.90      0.93      1115
weighted avg       0.97      0.97      0.97      1115
```

In [49]:
```
print("Training Accuracy", accuracy_score(y_train, train_pred_lr))
print()
print("Test Accuracy", accuracy_score(y_test,test_pred_lr))
```

```
Training Accuracy 0.9670181736594121

Test Accuracy 0.9721973094170404
```

Step:6 Building a Predictive System

In [52]:
```
input_mail = ["I've been searching for the right words to thank you for this breather. I promise i wont take your

## convert text to feature vectors

input_data_features = feature_extraction.transform(input_mail)

## making a prediction

prediction = lr.predict(input_data_features)
print(prediction)


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

else:
    print("ham")
```

```
[1]
spam
```

In [ ]: