```
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
```

```
#loading the csv data into pandas data frame
raw_mail_data=pd.read_csv('/content/mail_data.csv')
```

```
#checking the first five rows from the given data set
raw_mail_data.head()
```

```
   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...
```

```
#checking the last fice rows from the given data set
raw_mail_data.tail()
```

	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?

```
#checking the basic information from the given data set
raw_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
```

```
#checking the numbers of rows and columns
raw_mail_data.shape
```

```
(5572, 2)
```

Label encoding

```
# label spam mail as 0;  ham mail as 1;

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

Spam - 0 Ham - 1

```
#separating the data part into text and labels
X=raw_mail_data['Message']
```

```
Y=raw_mail_data['Category']
```

```
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
```

```
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
```

```
#splitting the data into train and test split
X_train,X_test,Y_train,Y_test = train_test_split(X, Y, test_size=0.2, random_state=3)
```

```
print(X)
print(X_test.shape)
print(X_train.shape)
```

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

Feature Extraction

```
# transform the text data to feature vectors that can be used as input to the Logistic regression

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)

# convert Y_train and Y_test values as integers

Y_train = Y_train.astype('int')
Y_test = Y_test.astype('int')
```

```
model = LogisticRegression()
```

```
# training the Logistic Regression model with the training data
model.fit(X_train_features, Y_train)
```

```
     LogisticRegression()
```

```
# prediction on training data

prediction_on_training_data = model.predict(X_train_features)
accuracy_on_training_data = accuracy_score(Y_train, prediction_on_training_data)


print('Accuracy on training data : ', accuracy_on_training_data)
```

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

```
input_mail = ["I've been searching for the right words to thank you for this breather. I promise i wont take your help for granted an

# convert text to feature vectors
input_data_features = feature_extraction.transform(input_mail)

# making prediction

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


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

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

```
     [1]
     Ham mail
```

● ✕