```
import numpy as np
import pandas as pd


#Importing LOgisticRegression Model
from sklearn.linear_model import LogisticRegression


#Importing accuracy score function
from sklearn.metrics import accuracy_score


#Imorting train test split function
from sklearn.model_selection import train_test_split


#To convert all the strings formats into numeric
from sklearn.feature_extraction.text import TfidfVectorizer


#Importing the dataset
raw_mail_data = pd.read_csv('/content/mail_data.csv')
```

+ Code + Text

```
#Replacing all the null values with null string
mail_data = raw_mail_data.where((pd.notnull(raw_mail_data)),'')


#Printing the 5 rows
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...

```
#Checlking the rows and columns of the dataset
mail_data.shape
```

```
(5572, 2)
```

LABEL ENCODING

```
#Label spam email as 0 and ham mails as 1
mail_data.loc[mail_data['Category']=='spam','Category'] = 0
mail_data.loc[mail_data['Category']=='ham','Category'] = 1


#seperating the data as texts and label
X = mail_data['Message']
Y = 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 training data and testing data
X_train , X_test ,Y_train , Y_test = train_test_split(X, Y, test_size = 0.2, random_state = 3)
print(X.shape, X_train.shape, X_test.shape)
```

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

```
#Tranforming the text data to feature vectors that can be used as input to the logisctic regression
feature_extraction = TfidfVectorizer(min_df = 1, stop_words = 'english', lowercase = True)


X_train_num = feature_extraction.fit_transform(X_train)
X_test_num = 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')
```

Training the Model

```
model = LogisticRegression()


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

```
▾ LogisticRegression
LogisticRegression()
```

```
#Evaluating the Trained Model
prediction_on_training_data = model.predict(X_train_num)
#Finding the accuracy score on training data
accuracy_on_training_data = accuracy_score(Y_train,prediction_on_training_data)
print("Accuracy score on training data is",accuracy_on_training_data)
```

```
Accuracy score on training data is 0.9670181736594121
```

```
#Evaluating the Testing Model
prediction_on_testing_data = model.predict(X_test_num)
#Finding the accuracy score on training data
accuracy_on_testing_data = accuracy_score(Y_test,prediction_on_testing_data)
print("Accuracy score on testing data is",accuracy_on_testing_data)
```

```
Accuracy score on testing data is 0.9659192825112107
```

```
#Building a predective system
input_mail = ["Welcome to rummy circle, You've won a free login bonus of 2500"]
#Converting text to feature vectors (Numerical format)
input_mail_num = feature_extraction.transform(input_mail)


#Making Prediction
prediction = model.predict(input_mail_num)
if prediction[0] == 1:
  print("Ham Email")
else:
  print("Spam Email")
```

```
Spam Email
```