
Python Programming
Email Spam Classification
A POPULAR CLASSIFICATION PROBLEM
SPAM

```
[ ] import numpy as np
    import pandas as pd
    from sklearn.model_selection impor
    from sklearn.feature_extraction.te
    from sklearn.linear_model import L
    from sklearn.metrics import accura
```

Data Collection & Pre-Processing

```
[ ] # loading the data from csv file t
    raw_mail_data = pd.read_csv('/cont
```

```
[ ] print(raw_mail_data)
```

```
     Category
0         ham  Go until jurong p
1         ham
2        spam  Free entry in 2 a
3         ham  U dun say so earl
4         ham  Nah I don't think
...       ...
5567     spam  This is the 2nd t
5568      ham              Will
5569      ham  Pity, * was in mo
5570      ham  The guy did some
5571      ham

[5572 rows x 2 columns]
```

```
[ ] # replace the null values with a r
    mail_data = raw_mail_data.where((p
```

```
[ ] # printing the first 5 rows of the
    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 number of rows and
    mail_data.shape
```

```
(5572, 2)
```

Label Encoding

```
[ ] # label spam mail as 0;  ham mail
```

```
[ ] # label spam mail as 0;  ham mail

    mail_data.loc[mail_data['Category'
    mail_data.loc[mail_data['Category'
```

spam - 0

ham - 1

```
[ ] # separating the data as texts anc

    X = mail_data['Message']

    Y = mail_data['Category']
```

```
[ ] print(X)
```

```
0       Go until jurong point, c
1                            Ok l
2       Free entry in 2 a wkly c
3       U dun say so early hor..
4       Nah I don't think he goe
                               ..
5567    This is the 2nd time we
5568                 Will ü b go
5569    Pity, * was in mood for
5570    The guy did some bitchin
5571                           R
Name: Message, Length: 5572, dty
```

```
[ ] print(Y)
```

```
[ ] 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, dt
```

Splitting the data into training data & test data

```
[ ] X_train, X_test, Y_train, Y_test =
```

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

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

Feature Extraction

```
[ ] # transform the text data to featu

    feature_extraction = TfidfVectoriz

    X_train_features = feature_extract
    X_test_features = feature_extracti

    # convert Y_train and Y_test value

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

```
[ ] print(X_train)
```

```
[ ] print(X_train_features)
```

Training the Model

Logistic Regression

```
[ ] model = LogisticRegression()
```

```
[ ] # training the Logistic Regressior
    model.fit(X_train_features, Y_trai
```

```
LogisticRegression(C=1.0,
class_weight=None, dual=False,
fit_intercept=True,
```

```
[ ] model = LogisticRegression()
```

```
[ ] # training the Logistic Regressior
    model.fit(X_train_features, Y_trai
```

```
LogisticRegression(C=1.0,
class_weight=None, dual=False,
fit_intercept=True,

intercept_scaling=1,
l1_ratio=None, max_iter=100,

multi_class='auto',
n_jobs=None, penalty='l2',

random_state=None,
solver='lbfgs', tol=0.0001,
verbose=0,

warm_start=False)
```

Evaluating the trained model

```
[ ] # prediction on training data

    prediction_on_training_data = mode
    accuracy_on_training_data = accura
```

```
[ ] print('Accuracy on training data :
```

```
Accuracy on training data :  0.9
```

```
[ ] input_mail = ["I've been searching

    # convert text to feature vectors
    input_data_features = feature_extr

    # making prediction

    prediction = model.predict(input_c
    print(prediction)


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

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


THANK YOU