Oasis Infobyte

Task 4. EMAIL SPAM detection using Machine Learning

We've all been the recipient of spam emails before. Spam mail, or junk mail, is a type of email that is sent to a massive number of users at one time, frequently containing cryptic messages, scams, or most dangerously, phishing content.

In this Project, use Python to build an email spam detector. Then, use machine learning to the spam detector to recognize and classify emails into spam and non-spam. Let's get started!

1. Import all necessary

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

2. Import dataframe

In [2]:
```
df = pd.read_csv('spam.csv',encoding="ISO-8859-1")
df
```

Out[2]:

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN
...	...	...	...	...	...
5567	spam	This is the 2nd time we have tried 2 contact u...	NaN	NaN	NaN
5568	ham	Will Ì_ b going to esplanade fr home?	NaN	NaN	NaN
5569	ham	Pity, * was in mood for that. So...any other s...	NaN	NaN	NaN
5570	ham	The guy did some bitching but I acted like i'd...	NaN	NaN	NaN
5571	ham	Rofl. Its true to its name	NaN	NaN	NaN

5572 rows × 5 columns

In [3]:
```
df = df.drop(["Unnamed: 2","Unnamed: 3","Unnamed: 4"], axis=1)
```

In [4]:
```
df.rename(columns = {"v1" : "output", "v2":"Message"},inplace = True)
df.head()
```

Out[4]:

	output	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...

5. Summery of data

In [5]:
```
df.isnull().sum()
```

Out[5]:
```
output     0
Message    0
dtype: int64
```

4. Check for Dublicate row

In [6]:
```
df.duplicated().sum()
```

Out[6]:
```
403
```

In [7]:
```
df = df.drop_duplicates(keep = 'first')
```

In [8]:
```
df.duplicated().sum()
```

Out[8]:
```
0
```

5. Summery of data

In [9]:
```
df.describe()
```

Out[9]:

	output	Message
count	5169	5169
unique	2	5169
top	ham	Go until jurong point, crazy.. Available only ...
freq	4516	1

6. Check column name

In [10]:
```
df.columns
```

Out[10]:
```
Index(['output', 'Message'], dtype='object')
```

7.Shape of dataset

In [11]:
```
df.shape
```

Out[11]:
```
(5169, 2)
```

8. Check the datatype

In [12]:
```
df.dtypes
```

Out[12]:
```
output     object
Message    object
dtype: object
```

9. LABEL ENCODING

In [13]:
```
df.columns
```

Out[13]:
```
Index(['output', 'Message'], dtype='object')
```

In [14]:
```
df["output"].unique()
```

Out[14]:
```
array(['ham', 'spam'], dtype=object)
```

In [15]:
```
df.replace({"spam" : 0 , "ham" : 1 } , inplace=True)
```

10. Splitting the data

In [16]:
```
x =  df["Message"]
y =df["output"]
```

In [17]:
```
x
```

Out[17]:
```
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 Ì_ 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: 5169, dtype: object
```

In [18]:
```
y
```

Out[18]:
```
0       1
1       1
2       0
3       1
4       1
       ..
5567    0
5568    1
5569    1
5570    1
5571    1
Name: output, Length: 5169, dtype: int64
```

11. training and testing data

In [19]:
```
from sklearn.model_selection import train_test_split

x_train, x_test, y_train,y_test = train_test_split(x,y,test_size = 0.20 )
```

In [20]:
```
x_train
```

Out[20]:
```
1311    U r too much close to my heart. If u go away i...
4923    We can go 4 e normal pilates after our intro...
5402    Hi babe its me thanks for coming even though i...
4383    Thanks honey but still haven't heard anything ...
3524    I not free today i haf 2 pick my parents up to...
                              ...
4444                         2 celebrate my båÕday, y else?
3849    I to am looking forward to all the sex cuddlin...
4609                        Just glad to be talking to you.
4574       Not directly behind... Abt 4 rows behind Ì_...
4443                         COME BACK TO TAMPA FFFFUUUUUUU
Name: Message, Length: 4135, dtype: object
```

In [21]:
```
x_test
```

Out[21]:
```
3375    Good afternon, my love. How are today? I hope ...
233      Yes:)here tv is always available in work place..
3358    Sorry I missed your call let's talk when you h...
193     It will stop on itself. I however suggest she ...
1035    Hello baby, did you get back to your mom's ? A...
                              ...
3758    GOD ASKED, \What is forgiveness?\" A little ch...
2160    No. Its not specialisation. Can work but its s...
3177                    K k :-):-) then watch some films.
4055    Ha ha nan yalrigu heltini..Iyo kothi chikku, u...
908                         WHITE FUDGE OREOS ARE IN STORES
Name: Message, Length: 1034, dtype: object
```

In [22]:
```
y_train
```

Out[22]:
```
1311    1
4923    1
5402    1
4383    1
3524    1
       ..
4444    1
3849    1
4609    1
4574    1
4443    1
Name: output, Length: 4135, dtype: int64
```

In [23]:
```
y_test
```

Out[23]:
```
3375    1
233     1
3358    0
193     1
1035    1
       ..
3758    1
2160    1
3177    1
4055    1
908     1
Name: output, Length: 1034, dtype: int64
```

12. Feature extraction

In [24]:
```
from sklearn.feature_extraction.text import CountVectorizer

cv = CountVectorizer()
x_train  = cv.fit_transform(x_train.values)
```

In [25]:
```
x_train.toarray()
```

Out[25]:
```
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]], dtype=int64)
```

In [ ]:
```

```

13. Model Fitting

In [26]:
```
from sklearn.naive_bayes import MultinomialNB

model = MultinomialNB()

model.fit(x_train,y_train)
```

Out[26]:
```
MultinomialNB()
```

14 . check model accuracy

In [27]:
```
from sklearn.metrics import confusion_matrix , recall_score , precision_score
from sklearn.metrics import accuracy_score
```

In [28]:
```
mail_ham = ['Same. Wana plan a trip sometme then']
mail_ham_count = cv.transform(mail_ham)
y_pred = model.predict(mail_ham_count)
y_pred
```

Out[28]:
```
array([1], dtype=int64)
```

finding accuracy of the training dataset

In [29]:
```
 model.score(x_train ,y_train)
```

Out[29]:
```
0.9937122128174123
```

#finding accuracy of the test dataset

In [30]:
```
x_test_count = cv.transform(x_test)
model.score(x_test_count,y_test)
```

Out[30]:
```
0.9845261121856866
```

In [ ]:

In [ ]:

In [ ]:

In [ ]: