Spam Mail Prediction With Python

```
#Standard libraries for data analysis:
----------------------

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# sklearn modules for data preprocessi
ng--------------------------------
-
from sklearn.model_selection import t
rain_test_split
from sklearn.naive_bayes import Multi
nomialNB
from sklearn.feature_extraction.text
import CountVectorizer

#sklearn modules for Model Evaluation
& Improvement-----------------------
--
from sklearn.metrics import confusion
_matrix, accuracy_score, f1_score, p
```

```
ecision_score, recall_score, fbeta_sc
ore
from statsmodels.stats.outliers_influ
ence import variance_inflation_factor
from sklearn.model_selection import c
ross_val_score, GridSearchCV, Shuffle
Split, KFold

# from sklearn import feature_selectio
n
from sklearn import model_selection

# from sklearn import metrics
from sklearn.metrics import classific
ation_report, precision_recall_curve
from sklearn.metrics import auc, roc_
auc_score, roc_curve
from sklearn.metrics import make_scor
er, recall_score, log_loss
from sklearn.metrics import average_p
recision_score

#Standard libraries for data visualiza
tion--------------------
```

```
import seaborn as sn
from matplotlib import pyplot
import matplotlib.pyplot as plt
import matplotlib.pylab as pylab
import matplotlib
%matplotlib inline
color = sn.color_palette()
import matplotlib.ticker as mtick
from IPython.display import display
pd.options.display.max_columns = None
from pandas.plotting import scatter_m
atrix
from sklearn.metrics import roc_curve
```

Import Dataset

In [2]:

```
df = pd.read_csv("/kaggle/input/spam-
email-dataset/emails.csv")
df
```

	text	spam
0	Subject: naturally irresistible your corporate...	1
1	Subject: the stock trading gunslinger fanny i...	1
2	Subject: unbelievable new homes made easy im ...	1
3	Subject: 4 color printing special request add...	1
4	Subject: do not have money , get software cds ...	1
...	...	...
5723	Subject: re : research and development charges...	0
5724	Subject: re : receipts from visit jim , than...	0
5725	Subject: re : enron case study update wow ! a...	0
5726	Subject: re : interest david , please , call...	0
5727	Subject: news : aurora 5 . 2 update aurora ve...	0

5728 rows × 2 columns

In [3]:

```
df.info
```

Out[3]:

```
<bound method DataFrame.info of
text  spam
0     Subject: naturally irresistible
your corporate...     1
1     Subject: the stock trading guns
linger  fanny i...     1
2     Subject: unbelievable new homes
made easy  im ...     1
3     Subject: 4 color printing speci
al  request add...     1
4     Subject: do not have money , ge
t software cds ...     1
...
...   ...
5723  Subject: re : research and deve
lopment charges...     0
5724  Subject: re : receipts from vis
```

```
5724  Subject: re : receipts from vis
it  jim ,  than...      0
5725  Subject: re : enron case study
update  wow ! a...      0
5726  Subject: re : interest  david ,
please , call...      0
5727  Subject: news : aurora 5 . 2 up
date  aurora ve...      0

[5728 rows x 2 columns]>
```

In [4]:

```
df.dtypes
```

Out[4]:

```
text    object
spam     int64
dtype: object
```

In [5]:

```
df.isna().sum()
```

Out[5]:

```
text    0
spam    0
dtype: int64
```

In [6]:

```
df.duplicated().sum()
```

Out[6]:

```
33
```

In [7]:

```
# drop duplicate
df.drop_duplicates(inplace=True)
```

In [8]:

```
# Check Target Variable Distribution
df["spam"].value_counts()
```

```
df["spam"].value_counts()
```

Out[8]:

```
spam
0    4327
1    1368
Name: count, dtype: int64
```

In this case, we have class imbalance with few positives. In our business challenge, false negatives are costly. Hence let's keep an eye onto the Precision, Recall & F2 score besides accuracy

Handling Text Data

In [9]:

```
# clean the text
from nltk.corpus import stopwords
from nltk.tokenize import word_tokeni
```

```
ze
import re
```

In [10]:

```
def clean_text(text):
    text=text.lower()
    text=re.sub('[^a-z]',' ',text)
    text=re.sub('subject',' ',text)
    text=word_tokenize(text)
    text=[word for word in text if le
n(word)>1]
    return ' '.join(text)
clean_text('Data clean')
```

Out[10]:

```
'data clean'
```

In [11]:

```
df['text']=df['text'].apply(clean_tex
t)
```

In [12]:

```
df
```

Out[12]:

	text	spam
0	naturally irresistible your corporate identity...	1
1	the stock trading gunslinger fanny is merrill ...	1
2	unbelievable new homes made easy im wanting to...	1
3	color printing special request additional info...	1
4	do not have money get software cds from here s...	1
...	...	...
5723	re research and development charges to gpg her...	0
5724	re receipts from visit jim thanks again for th...	0
5725	re enron case study update wow all on the same...	0
5726	re interest david please call shirley crenshaw...	0
5727	news aurora update aurora version the fastest ...	0

Data Prepocessing

In [13]:

```
cv = CountVectorizer()
X = cv.fit_transform(df['text']).toar
ray()
y = df['spam']
```

In [14]:

```
X_train, X_test, y_train, y_test = tr
ain_test_split(X, y,stratify=y, test_
size=0.2, random_state=0)

#to resolve any class imbalance - use
stratify parameter.
print("Number transactions X_train da
taset: ", X_train.shape)
print("Number transactions y_train da
```

```
taset: ", y_train.shape)
print("Number transactions X_test dat
aset: ", X_test.shape)
print("Number transactions y_test dat
aset: ", y_test.shape)
```

```
Number transactions X_train dataset:
(4556, 33681)
Number transactions y_train dataset:
(4556,)
Number transactions X_test dataset:
(1139, 33681)
Number transactions y_test dataset:
(1139,)
```


THANK YOU