IBM NAAN MUDHALVAN

PHASE 5 SUBMISSION

SPAM CLASSIFIER USING AI

AGENGA...

- *Introduction*
- *Problem Statements*
- *Dataset*
- *Python Codes using Machine Learning*
- *Conclusion*

INTRODUCTION

In today’s globalized world,email is a primary source of communication.This communication can vary from personal,business, corporate to government.SPAM emails,also known as junk mail involves nearly identical messages sent to numerous recipients by email.We try to identify patterns using Data-mining classification algorithms to enable us classifythe emails as HAM or SPAM.

PROBLEM STATEMENT

Unlike emails,which have a variety of large datasets available,real databases for SMS spams are very limited.Additionally,due to small length of text messages,the number of features that can be used for their classification is far smaller than the corresponding number in emails

DATASET

```
#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
```

```
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

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()
```

```
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()
```

```
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

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
```

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
```

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,)
```

SIMPLE CODING USING MACHINE LEARNING

Introducing ChatGPT | Spam Classifier ML Code | Welcome To Colaboratory - Cola | Untitled0.ipynb - Colaboratory

colab.research.google.com/drive/1cjm4--LvumkFjK0NGJ44WEzexp8zbv0o#scrollTo=m_0BVQkxXjM-

Untitled0.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

```python
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, classification_report

# Sample data: You should replace this with your dataset
# Create two lists, one containing messages and the other labels (1 for spam, 0 for non-spam)
messages = [
    "Congratulations, you've won a prize!",
    "Hello, can we schedule a meeting tomorrow?",
    "Click here to claim your reward!",
    "Don't forget our meeting today at 3 PM.",
]
labels = [1, 0, 1, 0]

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(messages, labels, test_size=0.2, random_state=42)

# Create a CountVectorizer to convert text into a bag of words representation
vectorizer = CountVectorizer()
X_train = vectorizer.fit_transform(X_train)
X_test = vectorizer.transform(X_test)

# Train a Multinomial Naive Bayes classifier
classifier = MultinomialNB()
classifier.fit(X_train, y_train)
```

0s completed at 6:42 PM

NED - BAN Live | Search | ENG IN | 18:42 28-10-2023

Introducing ChatGPT | Spam Classifier ML Code | Welcome To Colaboratory - Cola | Untitled0.ipynb - Colaboratory

colab.research.google.com/drive/1cjm4--LvumkFjK0NGJ44WEzexp8zbv0o#scrollTo=m_0BVQkxXjM-

Untitled0.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

```python
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(messages, labels, test_size=0.2, random_state=42)

# Create a CountVectorizer to convert text into a bag of words representation
vectorizer = CountVectorizer()
X_train = vectorizer.fit_transform(X_train)
X_test = vectorizer.transform(X_test)

# Train a Multinomial Naive Bayes classifier
classifier = MultinomialNB()
classifier.fit(X_train, y_train)

# Make predictions on the test data
y_pred = classifier.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)

print(f"Accuracy: {accuracy}")
print(report)
```

0s completed at 6:42 PM

NED - BAN Live

Search

ENG IN

18:42 28-10-2023

Introducing ChatGPT | Spam Classifier ML Code | Welcome To Colaboratory - Cola | Untitled0.ipynb - Colaboratory

colab.research.google.com/drive/1cjm4--LvumkFjK0NGJ44WEzexp8zbv0o#scrollTo=m_0BVQkxXjM-

Untitled0.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

```python
print(f"Accuracy: {accuracy}")
print(report)
```

```
Accuracy: 1.0
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         1

    accuracy                           1.00         1
   macro avg       1.00      1.00      1.00         1
weighted avg       1.00      1.00      1.00         1
```

0s completed at 6:42 PM

NED - BAN Live

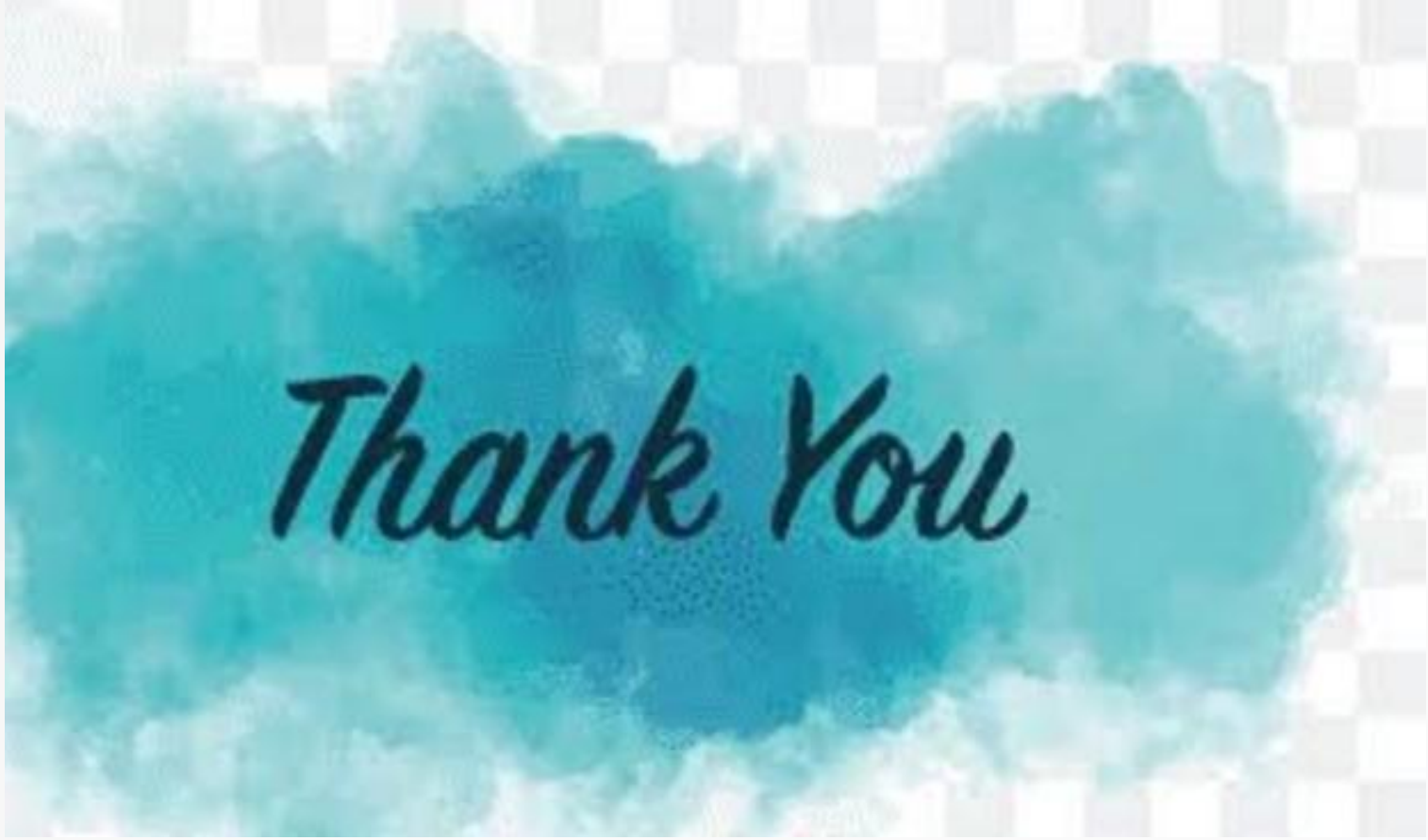
Search

ENG IN

18:42 28-10-2023

CONCLUSION

By accurately identifying and filtering spam,individuals and organizations can focus on important emails and mitigate potential risks associated with malicious content.In conclusion,email spam detection using machine learning offers a promising solution to the pervasive problem of unwanted and harmful emails.


Thank You