

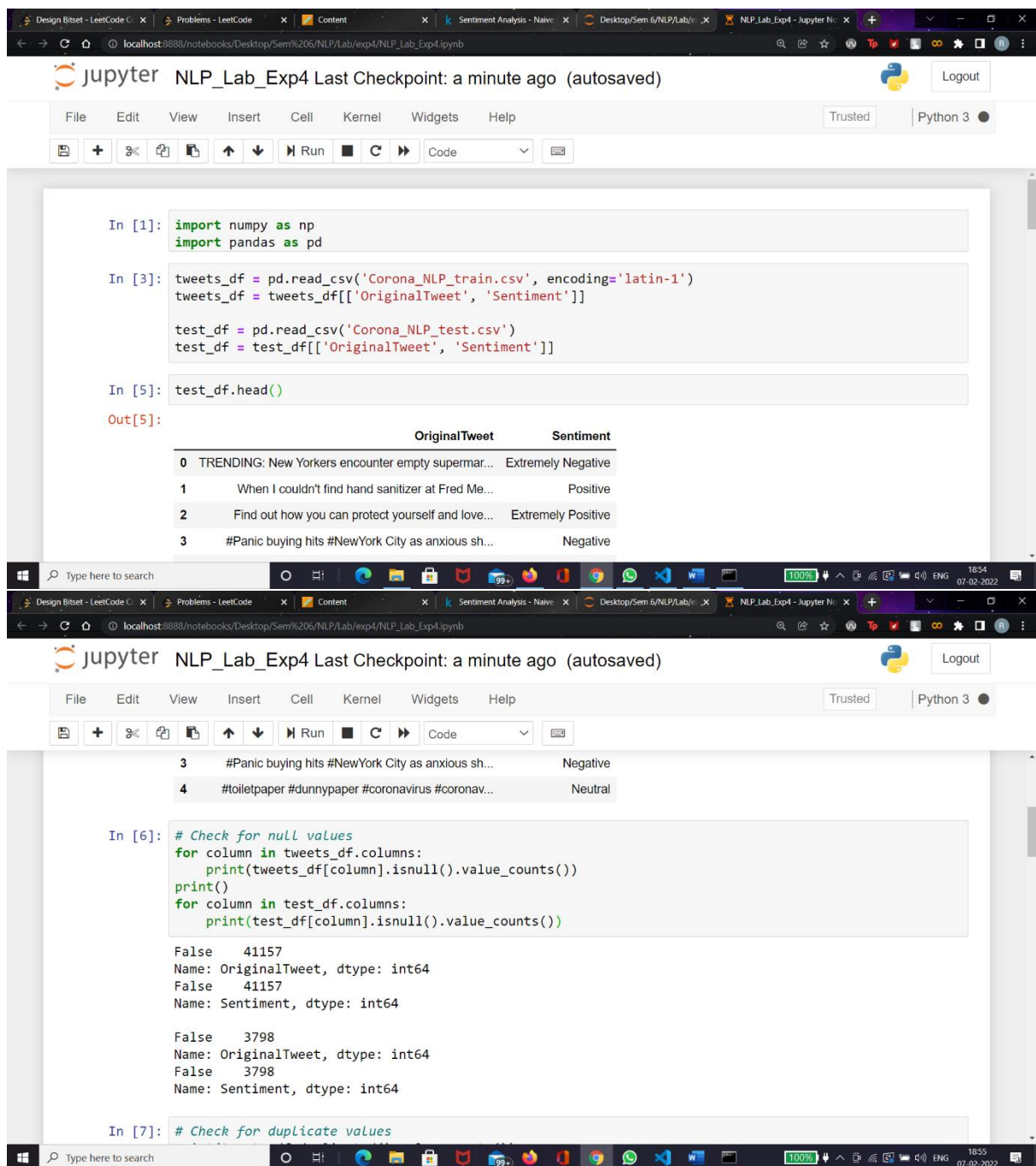
Rohan Nyati

500075940

R177219148

Batch – 5 (Ai & MI)

## Experiment – 4



The screenshot displays a Jupyter Notebook titled "NLP\_Lab\_Exp4" with the following content:

```
In [1]: import numpy as np
import pandas as pd

In [3]: tweets_df = pd.read_csv('Corona_NLP_train.csv', encoding='latin-1')
tweets_df = tweets_df[['OriginalTweet', 'Sentiment']]

test_df = pd.read_csv('Corona_NLP_test.csv')
test_df = test_df[['OriginalTweet', 'Sentiment']]

In [5]: test_df.head()
```

Out[5]:

	OriginalTweet	Sentiment
0	TRENDING: New Yorkers encounter empty supermar...	Extremely Negative
1	When I couldn't find hand sanitizer at Fred Me...	Positive
2	Find out how you can protect yourself and love...	Extremely Positive
3	#Panic buying hits #NewYork City as anxious sh...	Negative

```
3    #Panic buying hits #NewYork City as anxious sh... Negative
4    #toiletpaper #dunnypaper #coronavirus #coronav...
```

```
In [6]: # Check for null values
for column in tweets_df.columns:
    print(tweets_df[column].isnull().value_counts())
print()
for column in test_df.columns:
    print(test_df[column].isnull().value_counts())

False    41157
Name: OriginalTweet, dtype: int64
False    41157
Name: Sentiment, dtype: int64

False    3798
Name: OriginalTweet, dtype: int64
False    3798
Name: Sentiment, dtype: int64

In [7]: # Check for duplicate values
```

Design Bitset - LeetCode x Problems - LeetCode x Content x Sentiment Analysis - Naive x Desktop/Sem 6/NLP/Lab/ x NLP\_Lab\_Exp4 - Jupyter N x

localhost:8888/notebooks/Desktop/Sem%206/NLP/Lab/exp4/NLP\_Lab\_Exp4.ipynb

jupyter NLP\_Lab\_Exp4 Last Checkpoint: a minute ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [7]:

```
# Check for duplicate values
print(tweets_df.duplicated().value_counts())
print()
print(test_df.duplicated().value_counts())
```

False 41157  
dtype: int64

False 3798  
dtype: int64

In [8]:

```
import nltk
from nltk.corpus import stopwords
#from nltk.stem import PorterStemmer
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import word_tokenize
import string

nltk.download('stopwords')
#ps = PorterStemmer()
lemmatizer = WordNetLemmatizer()
```

jupyter NLP\_Lab\_Exp4 Last Checkpoint: a minute ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [9]:

```
!pip install pyspellchecker
from spellchecker import SpellChecker
spell = SpellChecker()
```

Collecting pyspellchecker  
Downloading <https://files.pythonhosted.org/packages/b4/e3/64a6a11f885d2f95a680e5d7bfa6aee3e3eb5f7671ff5bba0a80cd890fb3/pyspellchecker-0.6.3-py3-none-any.whl> (2.7MB)  
Installing collected packages: pyspellchecker  
Successfully installed pyspellchecker-0.6.3

In [10]:

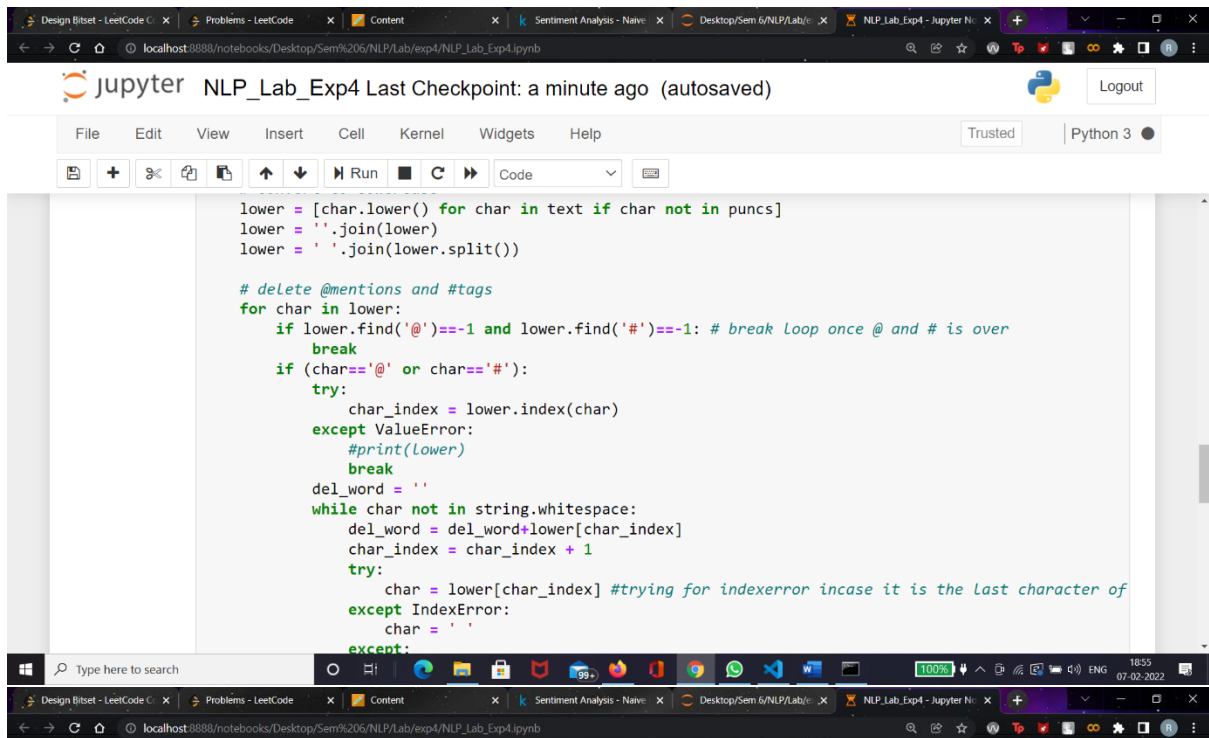
```
puncs_ = string.punctuation.replace('@','')
puncs = puncs_.replace('#','')
puncs
```

Out[10]: '!"\$%&'()\*+,-./:;<=>?[\\]^\_`{|}~'

In [11]:

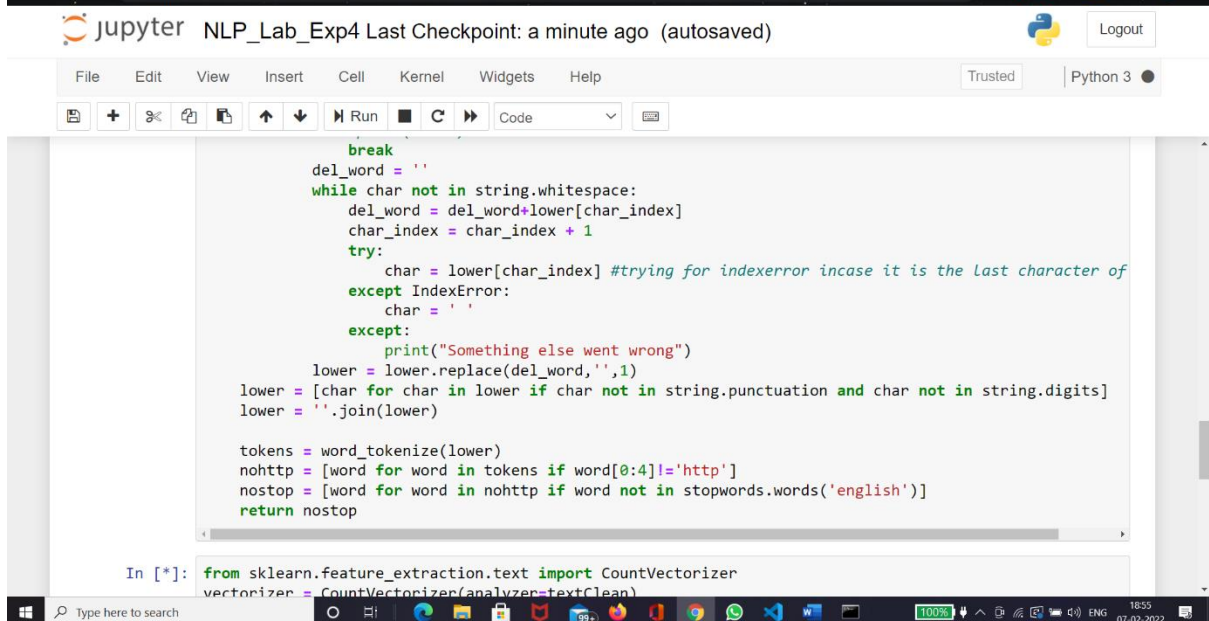
```
s = ' @ the springs theatre httpstcoaertookvav'
mytext = " ".join(s.split(" "))
mytext
```

Out[11]: ' @ the springs theatre httpstcoaertookvav'



```
lower = [char.lower() for char in text if char not in puncs]
lower = ''.join(lower)
lower = ' '.join(lower.split())

# delete @mentions and #tags
for char in lower:
    if lower.find('@')==1 and lower.find('#')==1: # break loop once @ and # is over
        break
    if (char=='@' or char=='#'):
        try:
            char_index = lower.index(char)
        except ValueError:
            #print(lower)
            break
        del_word = ''
        while char not in string.whitespace:
            del_word = del_word+lower[char_index]
            char_index = char_index + 1
        try:
            char = lower[char_index] #trying for indexerror incase it is the last character of
        except IndexError:
            char = ' '
        except:
```



```
        break
        del_word = ''
        while char not in string.whitespace:
            del_word = del_word+lower[char_index]
            char_index = char_index + 1
        try:
            char = lower[char_index] #trying for indexerror incase it is the last character of
        except IndexError:
            char = ' '
        except:
            print("Something else went wrong")
            lower = lower.replace(del_word, '', 1)
    lower = [char for char in lower if char not in string.punctuation and char not in string.digits]
    lower = ' '.join(lower)

    tokens = word_tokenize(lower)
    nohttp = [word for word in tokens if word[0:4]!='http']
    nostop = [word for word in nohttp if word not in stopwords.words('english')]
    return nostop

In [*]: from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer(analyzer=TextCleaner)
```

Design Bitset - LeetCode x Problems - LeetCode x Content x Sentiment Analysis - Naive x Desktop/Sem 6/NLP/Lab/ NLP\_Lab\_Exp4 - Jupyter N... x

localhost:8888/notebooks/Desktop/Sem%206/NLP/Lab/exp4/NLP\_Lab\_Exp4.ipynb

### jupyter NLP\_Lab\_Exp4 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

lower = .join(lower)

```
tokens = word_tokenize(lower)
nohttp = [word for word in tokens if word[0:4]!='http']
nostop = [word for word in nohttp if word not in stopwords.words('english')]
return nostop
```

In [17]: `from sklearn.feature_extraction.text import CountVectorizer`  
`vectorizer = CountVectorizer(analyzer=textClean)`  
`message = vectorizer.fit_transform(tweets_df['OriginalTweet'])`  
`message.shape`

Out[17]: (41157, 39097)

In [16]: `import nltk`  
`nltk.download('punkt')`

[nltk\_data] Downloading package punkt to

Type here to search

Design Bitset - LeetCode x Problems - LeetCode x Content x Sentiment Analysis - Naive x Desktop/Sem 6/NLP/Lab/ NLP\_Lab\_Exp4 - Jupyter N... x

localhost:8888/notebooks/Desktop/Sem%206/NLP/Lab/exp4/NLP\_Lab\_Exp4.ipynb

### jupyter NLP\_Lab\_Exp4 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [19]: `from sklearn.model_selection import train_test_split`  
`xtrain, xtest, ytrain, ytest = train_test_split(message, tweets_df.Sentiment, test`  
`from sklearn.naive_bayes import MultinomialNB`  
`classifier = MultinomialNB().fit(xtrain, ytrain)`

In [20]: `from sklearn.metrics import classification_report, confusion_matrix, accuracy_sc`  
`pred = classifier.predict(xtrain)`  
`print(classification_report(ytrain, pred))`  
`print()`  
`print("Confusion Matrix: \n", confusion_matrix(ytrain, pred))`  
`print("Accuracy: \n", accuracy_score(ytrain, pred))`

	precision	recall	f1-score	support
Extremely Negative	0.88	0.66	0.76	4387
Extremely Positive	0.83	0.72	0.77	5293

Type here to search

Design Bitset - LeetCode x Problems - LeetCode x Content x Sentiment Analysis - Naive x Desktop/Sem 6/NLP/Lab/ NLP\_Lab\_Exp4 - Jupyter N... x

localhost:8888/notebooks/Desktop/Sem%206/NLP/Lab/exp4/NLP\_Lab\_Exp4.ipynb



Design Bitset - LeetCode x Problems - LeetCode x Content x Sentiment Analysis - Naive x Desktop/Sem 6/NLP/Lab/ x NLP\_Lab\_Exp4 - Jupyter N x

localhost:8888/notebooks/Desktop/Sem%206/NLP/Lab/exp4/NLP\_Lab\_Exp4.ipynb

### jupyter NLP\_Lab\_Exp4 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

```
print("Confusion Matrix: \n", confusion_matrix(ytrain, pred))
print("Accuracy: \n", accuracy_score(ytrain, pred))
```

	precision	recall	f1-score	support
Extremely Negative	0.88	0.66	0.76	4387
Extremely Positive	0.83	0.72	0.77	5293
Negative	0.70	0.78	0.74	7931
Neutral	0.93	0.55	0.69	6187
Positive	0.63	0.87	0.73	9127
accuracy			0.73	32925
macro avg	0.79	0.71	0.74	32925
weighted avg	0.77	0.73	0.73	32925

Confusion Matrix:  
[[2909 32 1033 36 377]

Type here to search

Design Bitset - LeetCode x Problems - LeetCode x Content x Sentiment Analysis - Naive x Desktop/Sem 6/NLP/Lab/ x NLP\_Lab\_Exp4 - Jupyter N x

localhost:8888/notebooks/Desktop/Sem%206/NLP/Lab/exp4/NLP\_Lab\_Exp4.ipynb

### jupyter NLP\_Lab\_Exp4 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

accuracy			0.73	32925
macro avg	0.79	0.71	0.74	32925
weighted avg	0.77	0.73	0.73	32925

Confusion Matrix:  
[[2909 32 1033 36 377]  
[ 21 3791 166 20 1295]  
[ 257 152 6180 110 1232]  
[ 59 139 887 3372 1730]  
[ 64 477 575 99 7912]]

Accuracy:  
0.7339104024297646

```
In [21]: from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
pred = classifier.predict(xtest)
print(classification_report(ytest, pred))
print()
```

Type here to search

Design Bitset - LeetCode x Problems - LeetCode x Content x Sentiment Analysis - Naive x Desktop/Sem 6/NLP/Lab/ x NLP\_Lab\_Exp4 - Jupyter N x

localhost:8888/notebooks/Desktop/Sem%206/NLP/Lab/exp4/NLP\_Lab\_Exp4.ipynb

Design Bitset - LeetCode x Problems - LeetCode x Content x Sentiment Analysis - Naive x Desktop/Sem 6/NLP/Lab/ x NLP\_Lab\_Exp4 - Jupyter N x

localhost:8888/notebooks/Desktop/Sem%206/NLP/Lab/exp4/NLP\_Lab\_Exp4.ipynb

### jupyter NLP\_Lab\_Exp4 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 Logout

Run Code

Accuracy:  
0.7339104024297646

```
In [21]: from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
pred = classifier.predict(xtest)
print(classification_report(ytest, pred))
print()
print("Confusion Matrix: \n", confusion_matrix(ytest, pred))
print("Accuracy: \n", accuracy_score(ytest, pred))
```

	precision	recall	f1-score	support
Extremely Negative	0.59	0.39	0.47	1094
Extremely Positive	0.56	0.44	0.49	1331
Negative	0.44	0.50	0.47	1986
Neutral	0.67	0.34	0.45	1526
Positive	0.41	0.61	0.49	2295

Type here to search

Design Bitset - LeetCode x Problems - LeetCode x Content x Sentiment Analysis - Naive x Desktop/Sem 6/NLP/Lab/ x NLP\_Lab\_Exp4 - Jupyter N x

localhost:8888/notebooks/Desktop/Sem%206/NLP/Lab/exp4/NLP\_Lab\_Exp4.ipynb

### jupyter NLP\_Lab\_Exp4 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 Logout

Run Code

Confusion Matrix:  
[[ 422 8 511 26 127]  
[ 7 589 78 23 634]  
[ 209 72 1000 81 624]  
[ 30 44 313 518 621]  
[ 48 336 395 123 1393]]

Accuracy:  
0.47643343051506315

```
In [22]: test_df.shape
```

Out[22]: (3798, 2)

```
In [23]: message2 = vectorizer.transform(test_df['OriginalTweet'])
message2.shape
```

Out[23]: (3798, 39097)

Type here to search

Design Bitset - LeetCode x Problems - LeetCode x Content x Sentiment Analysis - Naive x Desktop/Sem 6/NLP/Lab/ x NLP\_Lab\_Exp4 - Jupyter N x

localhost:8888/notebooks/Desktop/Sem%206/NLP/Lab/exp4/NLP\_Lab\_Exp4.ipynb

Design Bitset - LeetCode x Problems - LeetCode x Content x Sentiment Analysis - Naive x Desktop/Sem 6/NLP/Lab/ x NLP\_Lab\_Exp4 - Jupyter N x

localhost:8888/notebooks/Desktop/Sem%206/NLP/Lab/exp4/NLP\_Lab\_Exp4.ipynb

### jupyter NLP\_Lab\_Exp4 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [22]: test\_df.shape

Out[22]: (3798, 2)

In [23]: message2 = vectorizer.transform(test\_df['OriginalTweet'])  
message2.shape

Out[23]: (3798, 39097)

In [24]: message2

Out[24]: <3798x39097 sparse matrix of type '<class 'numpy.int64'>' with 58205 stored elements in Compressed Sparse Row format>

In [25]: pred = classifier.predict(message2)  
print(classification\_report(test\_df.Sentiment, pred))  
print()  
print("Confusion Matrix: \n", confusion\_matrix(test\_df.Sentiment, pred))

Type here to search

Design Bitset - LeetCode x Problems - LeetCode x Content x Sentiment Analysis - Naive x Desktop/Sem 6/NLP/Lab/ x NLP\_Lab\_Exp4 - Jupyter N x

localhost:8888/notebooks/Desktop/Sem%206/NLP/Lab/exp4/NLP\_Lab\_Exp4.ipynb

jupyter NLP\_Lab\_Exp4 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Out[24]: <3798x39097 sparse matrix of type '<class 'numpy.int64'>' with 58205 stored elements in Compressed Sparse Row format>

In [25]: pred = classifier.predict(message2)  
print(classification\_report(test\_df.Sentiment, pred))  
print()  
print("Confusion Matrix: \n", confusion\_matrix(test\_df.Sentiment, pred))  
print("Accuracy: \n", accuracy\_score(test\_df.Sentiment, pred))

	precision	recall	f1-score	support
Extremely Negative	0.59	0.30	0.39	592
Extremely Positive	0.63	0.35	0.45	599
Negative	0.44	0.51	0.47	1041
Neutral	0.66	0.21	0.31	619
Positive	0.37	0.69	0.48	947
accuracy			0.45	3798
macro avg	0.54	0.41	0.42	3798

Type here to search

Design Bitset - LeetCode x Problems - LeetCode x Content x Sentiment Analysis - Naive x Desktop/Sem 6/NLP/Lab/ x NLP\_Lab\_Exp4 - Jupyter N x

localhost:8888/notebooks/Desktop/Sem%206/NLP/Lab/exp4/NLP\_Lab\_Exp4.ipynb

Design Bitset - LeetCode x Problems - LeetCode x Content x Sentiment Analysis - Naive x Desktop/Sem 6/NLP/Lab/ NLP\_Lab\_Exp4 - Jupyter N...

localhost:8888/notebooks/Desktop/Sem%206/NLP/Lab/exp4/NLP\_Lab\_Exp4.ipynb

# jupyter NLP\_Lab\_Exp4 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

Extremely Negative	0.59	0.30	0.39	592
Extremely Positive	0.63	0.35	0.45	599
Negative	0.44	0.51	0.47	1041
Neutral	0.66	0.21	0.31	619
Positive	0.37	0.69	0.48	947
accuracy			0.45	3798
macro avg	0.54	0.41	0.42	3798
weighted avg	0.51	0.45	0.43	3798

Confusion Matrix:

```
[[175  5 328  4  80]
 [  5 211  34  2 347]
 [ 83  21 529 38 370]
 [  9  15 158 127 310]
 [ 26  84 164  21 652]]
```

Accuracy:

0.4460242232754081

Type here to search

100% 19:15 07-02-2022