code for Sentiment Analysis 210409183

January 26, 2023

#Initial data loading

[ ]:
```
!pip install tensorflow_addons
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Collecting tensorflow_addons
  Downloading tensorflow_addons-0.19.0-cp38-cp38-manylinux_2_17_x86_64.manylinux
2014_x86_64.whl (1.1 MB)
                                    1.1/1.1 MB
23.2 MB/s eta 0:00:00
Requirement already satisfied: packaging in /usr/local/lib/python3.8/dist-
packages (from tensorflow_addons) (21.3)
Requirement already satisfied: typeguard>=2.7 in /usr/local/lib/python3.8/dist-
packages (from tensorflow_addons) (2.7.1)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/usr/local/lib/python3.8/dist-packages (from packaging->tensorflow_addons)
(3.0.9)
Installing collected packages: tensorflow_addons
Successfully installed tensorflow_addons-0.19.0
```

[ ]:
```
from google.colab import drive
import numpy as np
import pandas as pd
```

[ ]:
```
drive.mount('/content/gdrive')
```

```
Mounted at /content/gdrive
```

[ ]:
```
original_data=pd.read_csv('/content/gdrive/MyDrive/Ml_project/Tweets_train.csv')
data_test=pd.read_csv('/content/gdrive/MyDrive/Ml_project/Tweets_test.
 ↪csv',encoding = "ISO-8859-1")
data_valid=pd.read_csv('/content/gdrive/MyDrive/Ml_project/Tweets_dev.
 ↪csv',encoding = "ISO-8859-1")
```

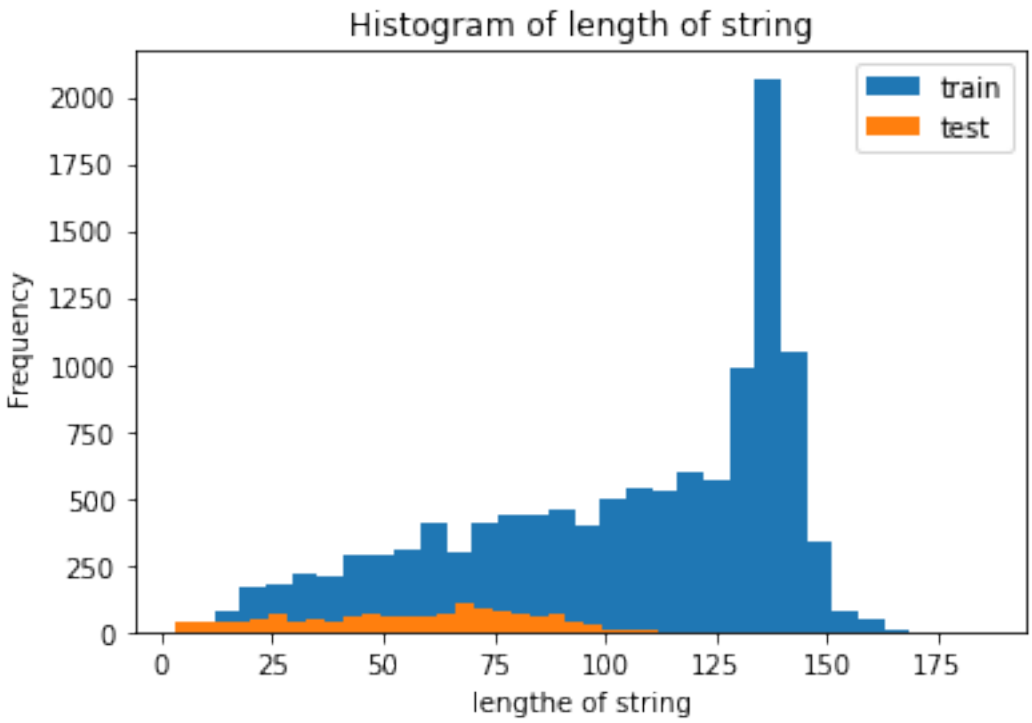
[ ]:
```
data_test
```

[ ]:
```
      Unnamed: 0            tweet_id  \
0              0  568107472260624384
1              1  568215698524246016
2              2  567842466851905536
3              3  568834824410148864
4              4  569590527349252096
…            …                  …
1310        1313  570060687164067840
1311        1314  570101371409559552
1312        1315  568572753403650049
1313        1316  567747769176432640
1314        1317  570011378091753472

                                                   text airline_sentiment
0     great job celebrating industry another reason …          positive
1            thanks taking upnotch leinenkugels norfolk           positive
2            put back hold hour completely unacceptable           negative
3                                    thank offer sorted           positive
4     wondering possible colleague andto get earlier…            neutral
…                                                   …                …
1310                         sorry disappointed kid job           negative
1311  stuck onplane dallas thats supposed going okc …          negative
1312  lost wallet flight yesterday houston bogota fi…          negative
1313  travelling pwm atl sunday flight got cancelled…          negative
1314                                             thank           positive

[1315 rows x 4 columns]
```

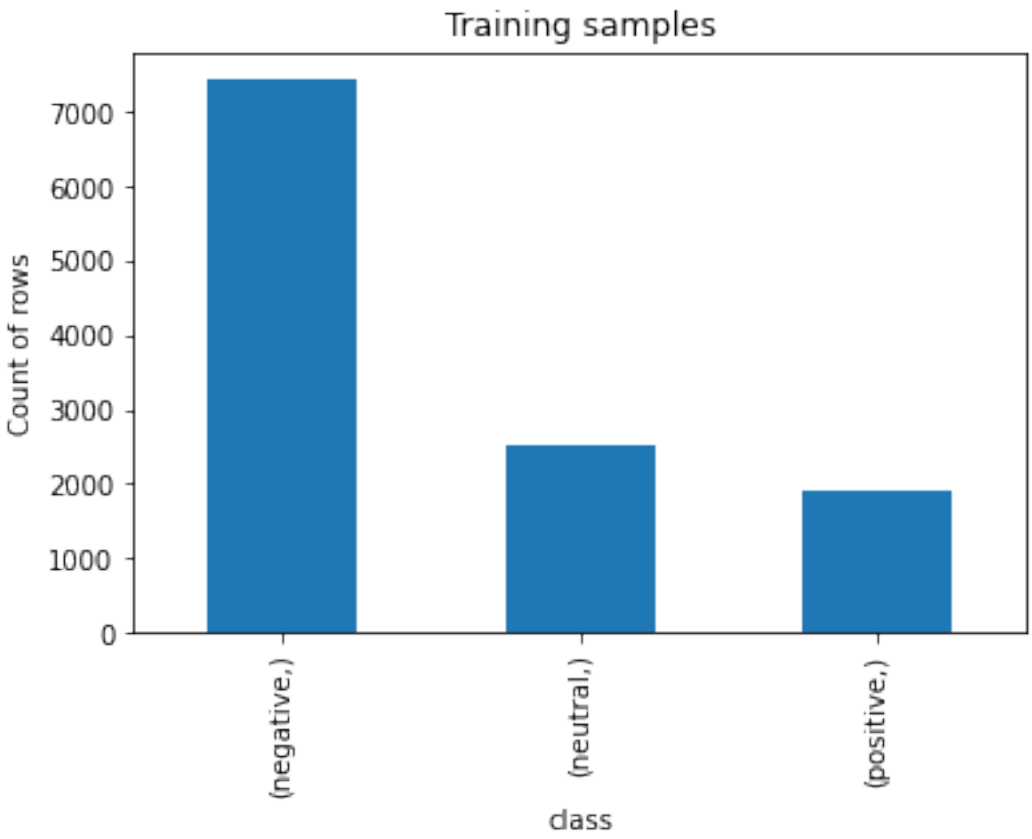
#Data Exploration

0.1 before balancing

[ ]:
```
import matplotlib.pyplot as plt

plt.hist(original_data.text.str.len(), bins=30, label='train')
plt.hist(data_test.text.str.len(), bins=30, label='test')
plt.legend()
plt.xlabel("lengthe of string")
plt.ylabel(" Frequency")
plt.title("Histogram of length of string")
plt.show()
```



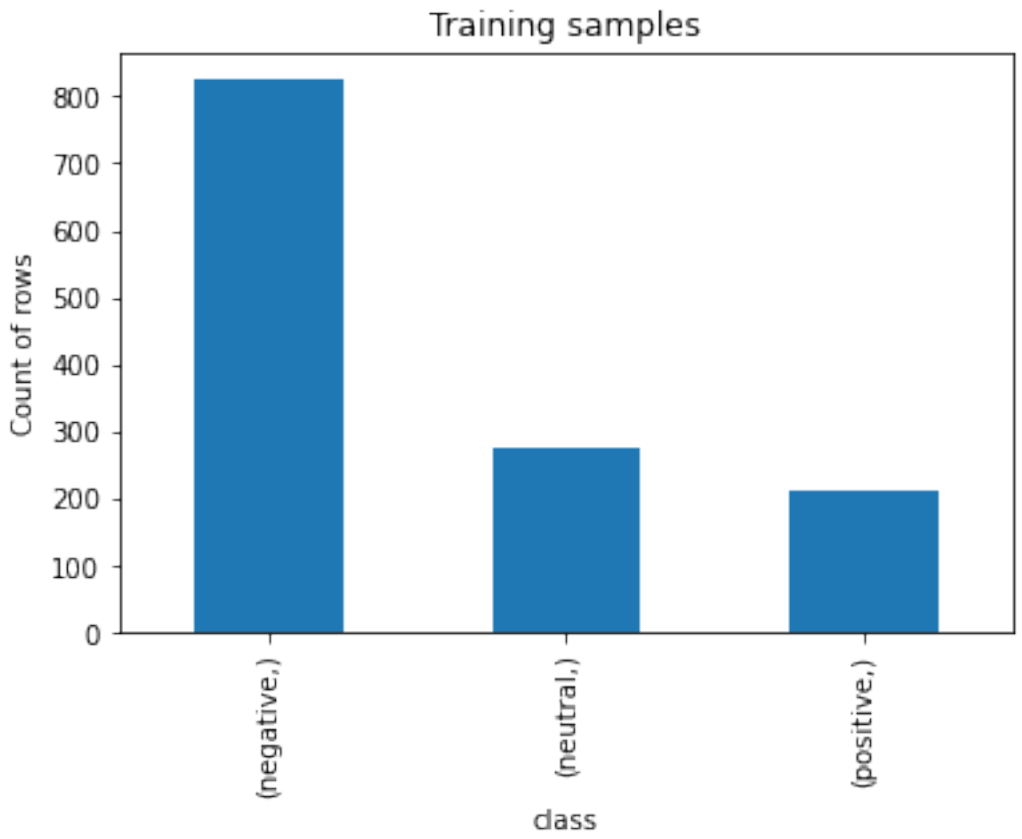
[ ]:
```
import pandas as pd
import matplotlib.pyplot as plt

k=pd.DataFrame(original_data['airline_sentiment'])
k.value_counts().plot(kind='bar')
plt.title('Training samples')
plt.ylabel('Count of rows')
plt.xlabel('class')
plt.show()
```



[ ]:
```
original_data['airline_sentiment'].value_counts()
```

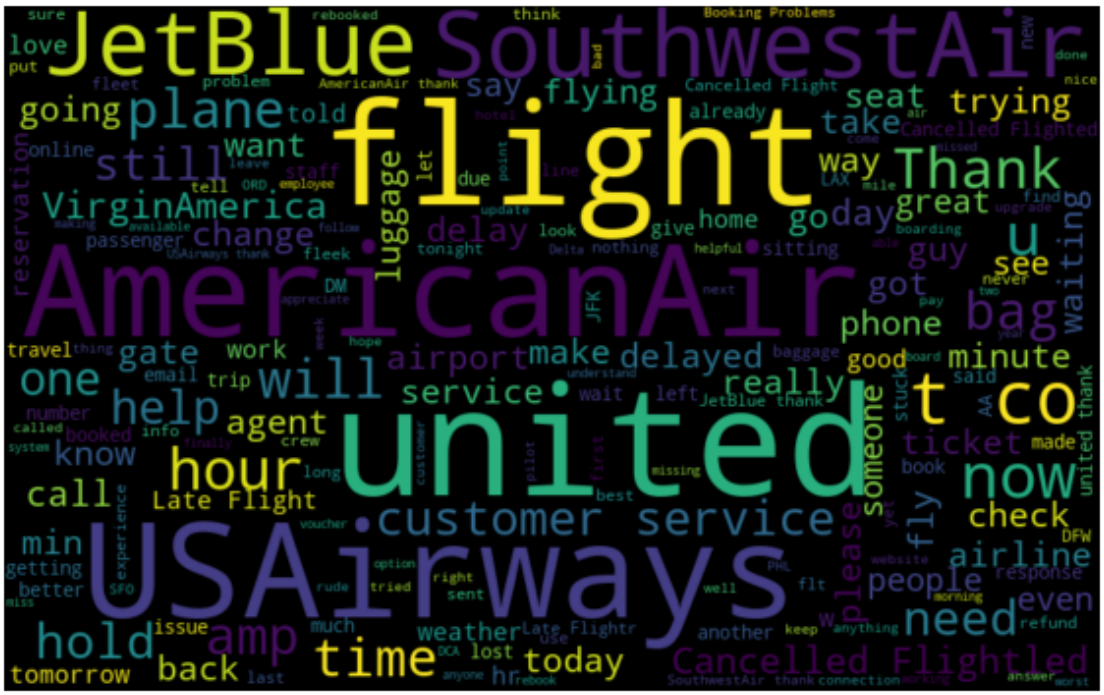
[ ]:
```
negative    7434
neutral     2510
positive    1914
Name: airline_sentiment, dtype: int64
```

[ ]:
```
import pandas as pd
import matplotlib.pyplot as plt

k=pd.DataFrame(data_test['airline_sentiment'])
k.value_counts().plot(kind='bar')
plt.title('Training samples')
plt.ylabel('Count of rows')
plt.xlabel('class')
plt.show()
```



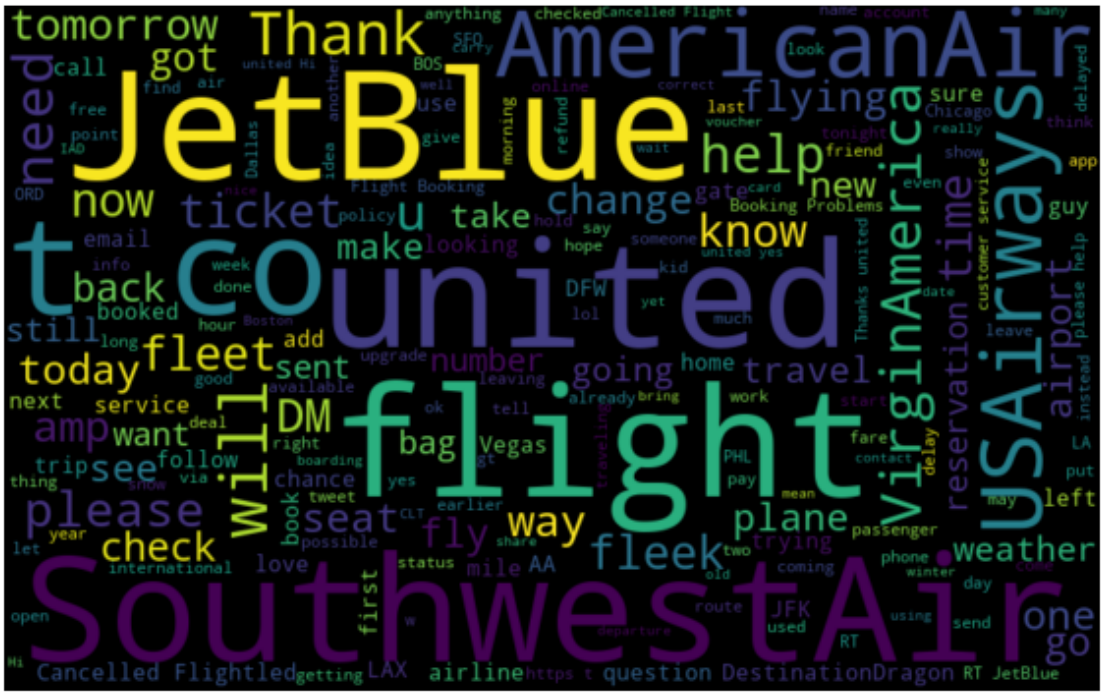
which word occured more?

[ ]:
```
word_bag = ' '.join([text for text in original_data["text"]])
from wordcloud import WordCloud
word_cloud = WordCloud(width=800, height=500, random_state=21,␣
 ↪max_font_size=110).generate(word_bag)
plt.figure(figsize=(10, 7))
plt.imshow(word_cloud, interpolation="bilinear")
plt.axis('off')
plt.title(" Plot representing the most occuring words ")
plt.show()
```



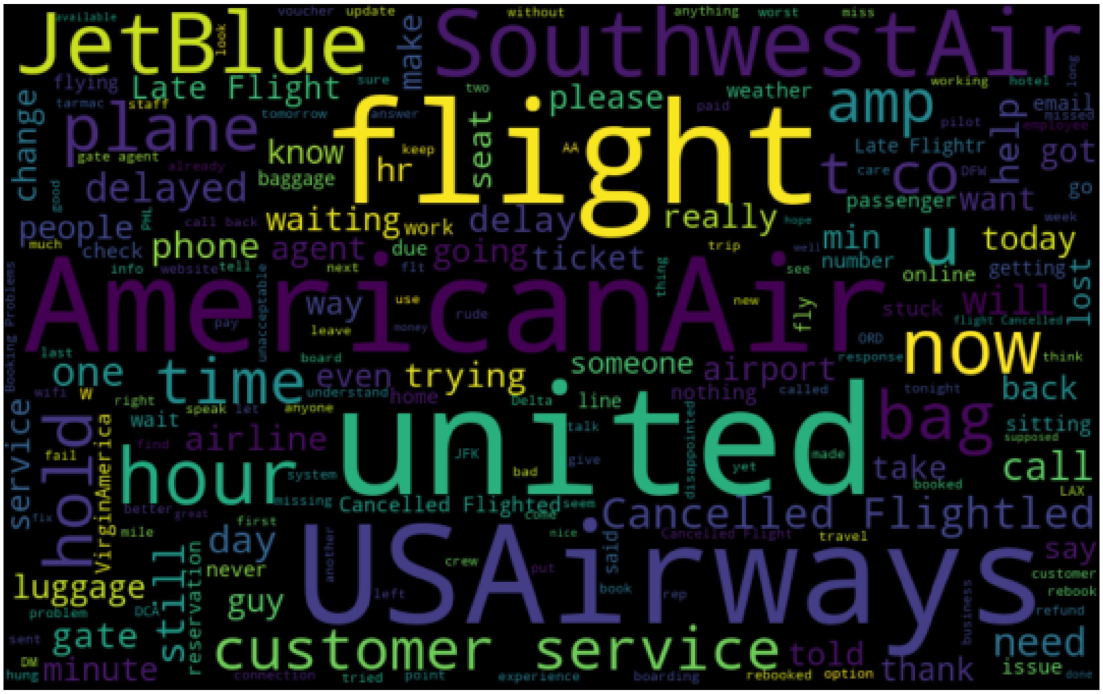

```
[ ]: neutr_word = ' '.join([text for text in␣
      ↪original_data['text'][original_data['airline_sentiment'] == 'neutral']])
     from wordcloud import WordCloud
     word_cloud = WordCloud(width=800, height=500, random_state=21,␣
      ↪max_font_size=110).generate(neutr_word)
     plt.figure(figsize=(10, 7))
     plt.imshow(word_cloud, interpolation="bilinear")
     plt.title(" Plot representing the most occuring words neutral class")
     plt.axis('off')
     plt.show()
```

Plot representing the most occuring words neutral class



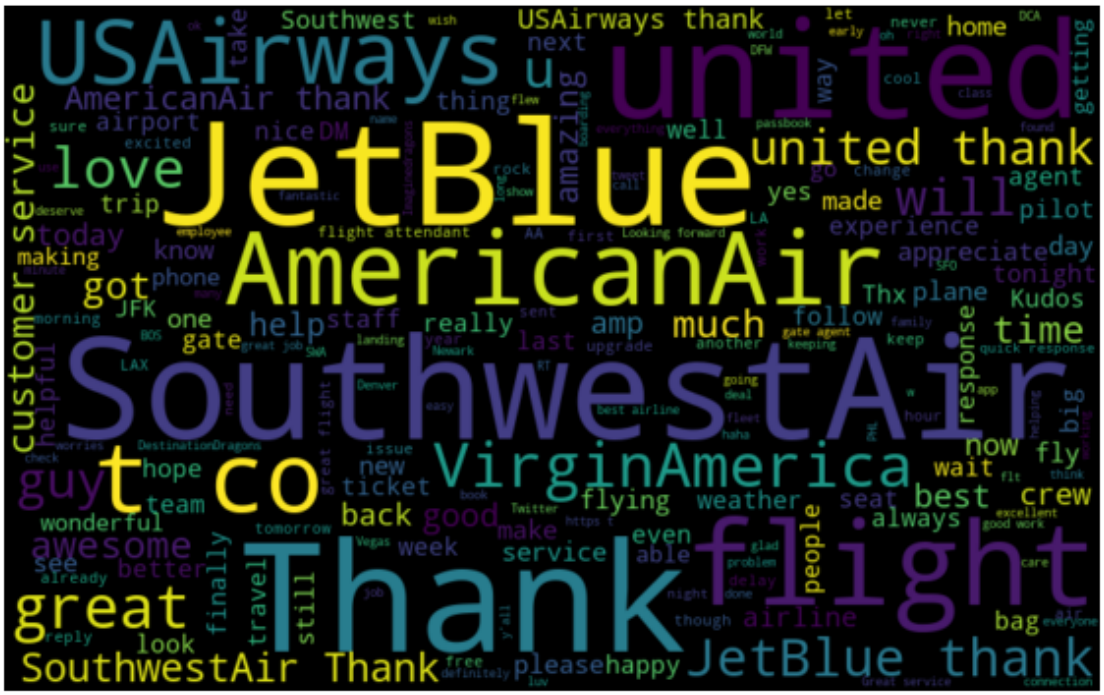

```
[ ]: negative_word = ' '.join([text for text in␣
      ↪original_data['text'][original_data['airline_sentiment'] == 'negative']])
     from wordcloud import WordCloud
     word_cloud = WordCloud(width=800, height=500, random_state=21,␣
      ↪max_font_size=110).generate(negative_word)
     plt.figure(figsize=(10, 7))
     plt.imshow(word_cloud, interpolation="bilinear")
     plt.title(" Plot representing the most occuring words negative class")
     plt.axis('off')
     plt.show()
```

Plot representing the most occuring words negative class




```
[ ]: positive_word = ' '.join([text for text in original_data['text'][original_data['airline_sentiment'] == 'positive']])
from wordcloud import WordCloud
word_cloud = WordCloud(width=800, height=500, random_state=21, max_font_size=110).generate(positive_word)
plt.figure(figsize=(10, 7))
plt.imshow(word_cloud, interpolation="bilinear")
plt.title(" Plot representing the most occuring words positive class")
plt.axis('off')
plt.show()
```

Plot representing the most occuring words positive class



Checking for missing values

[ ]:
```
original_data.isnull().sum()
```

[ ]:
```
tweet_id             0
text                 0
airline_sentiment    0
dtype: int64
```

#preprocessing

[ ]:
```
# Import stopwords
import nltk
nltk.download('omw-1.4')
from nltk.corpus import stopwords
nltk.download('stopwords')
nltk.download('wordnet')
stop_words_a = stopwords.words('english')
# Import textblob
from textblob import Word, TextBlob
custom_stopwords_a = ['']
def preprocess_tweets(tweet, custom_stopwords_a):
    processed_tweet_a = tweet
    processed_tweet_a.lower
```

```
    processed_tweet_a = " ".join(word for word in processed_tweet_a.split() if␣
 ↪word not in stop_words_a)
    processed_tweet_a = " ".join(word for word in processed_tweet_a.split() if␣
 ↪word not in custom_stopwords_a)
    processed_tweet_a = " ".join(Word(word).lemmatize() for word in␣
 ↪processed_tweet_a.split())
    return(processed_tweet_a)

original_data['text'] = original_data['text'].replace(r'#\w+|@\w+|https?:\/\/
 ↪\S+', '', regex=True) #url removing
original_data['text'] = original_data['text'].
 ↪replace(r'\s+[a-zA-Z]\s+','',regex=True) #single charector removing
original_data['text'] = original_data['text'].replace(r'[^A-Za-z ]+', '',␣
 ↪regex=True) # removing special charecters and numbers
original_data['text'] = original_data['text'].replace(r'\s+',' ',regex=True)#␣
 ↪removing multiple spaces
original_data['text'] = original_data['text'].str.lower() # lower

data_test['text'] = data_test['text'].replace(r'#\w+|@\w+|https?:\/\/\S+', '',␣
 ↪regex=True)
data_test['text'] = data_test['text'].replace(r'\s+[a-zA-Z]\s+','',regex=True)
data_test['text'] = data_test['text'].replace(r'[^A-Za-z ]+', '', regex=True)
data_test['text'] = data_test['text'].replace(r'\s+',' ',regex=True)
data_test['text'] = data_test['text'].str.lower() # lower


data_valid['text'] = data_valid['text'].replace(r'#\w+|@\w+|https?:\/\/\S+',␣
 ↪'', regex=True)#url removing
data_valid['text'] = data_valid['text'].replace(r'\s+[a-zA-Z]\s+','',regex=True)
data_valid['text'] = data_valid['text'].replace(r'[^A-Za-z ]+', '', regex=True)
data_valid['text'] = data_valid['text'].replace(r'\s+',' ',regex=True)
data_valid['text'] = data_valid['text'].str.lower() # lower


original_data['text']= original_data['text'].apply(lambda x:␣
 ↪preprocess_tweets(x, custom_stopwords_a))
data_test['text']= data_test['text'].apply(lambda x: preprocess_tweets(x,␣
 ↪custom_stopwords_a))
data_valid['text']= data_valid['text'].apply(lambda x: preprocess_tweets(x,␣
 ↪custom_stopwords_a))
```

```
[nltk_data] Downloading package omw-1.4 to /root/nltk_data…
[nltk_data]   Package omw-1.4 is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data…
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to /root/nltk_data…
[nltk_data]   Package wordnet is already up-to-date!
```

#empty string/null checking/single charecter Handling

[ ]:
```
original_data.drop(original_data[original_data['text'].apply(lambda x:␣
 ↪len(x)==0)].index,inplace=True)
original_data.drop(original_data[original_data['text'].apply(lambda x:␣
 ↪len(x)==1)].index,inplace=True)
original_data.drop(original_data[original_data['text'].apply(lambda x:␣
 ↪len(x)==2)].index,inplace=True)
original_data.dropna()
```

[ ]:
```
                tweet_id                                               text  \
0      569179849518161920                                  youre good thank
1      569835751275433984  way ruinvacation brother called night multiple…
2      568588936852799488  yes thankfully catering got loading frustrated…
3      569525116725567491  automated message isnt helpful impossible spea…
5      569617089155211265  downloaded app iphone notice drink coupon noth…
…                     …                                                  …
11853  570123872168574976  help u phone gate checkinbook travel client ca…
11854  570063683256242177  worst customer service line ive called time to…
11855  568032524749942784  grade tripflight timeliness cancelled flightat…
11856  569705813142409217  thanks vague canned response doesnt address issue
11857  569976114124349440  already airport hr late flightr still guy real…

      airline_sentiment
0              positive
1              negative
2              positive
3              negative
5               neutral
…                  …
11853          negative
11854          negative
11855          negative
11856          negative
11857          negative

[11834 rows x 3 columns]
```

[ ]:
```
data_valid.drop(data_valid[data_valid['text'].apply(lambda x: len(x)==0)].
 ↪index,inplace=True)
data_valid.drop(data_valid[data_valid['text'].apply(lambda x: len(x)==1)].
 ↪index,inplace=True)
data_valid.drop(data_valid[data_valid['text'].apply(lambda x: len(x)==2)].
 ↪index,inplace=True)
data_valid.dropna()
data_valid
```

[ ]:
```
      Unnamed: 0            tweet_id  \
0              0  570252000000000000
1              1  568173000000000000
2              2  569321000000000000
3              3  569503000000000000
4              4  568981000000000000
…            …                  …
1454        1459  569678000000000000
1455        1460  569882000000000000
1456        1461  568192000000000000
1457        1462  569775000000000000
1458        1463  569941000000000000

                                                   text airline_sentiment
0                                           need refund          negative
1     cancelled flightlations anddelay causing miss …          negative
2                         thanks much cant wait fly guy          positive
3     never frustrated conversation united cant spea…          negative
4     worst hold time crazy agent horrible accountab…          negative
…                                                   …               …
1454  didnt miss flight american airline gave ticket…          negative
1455  sitting hold hr flight cancelled flighted disc…          negative
1456                        hadgreat flight damion best          positive
1457                               aa return jfk thanks           neutral
1458  cangetflight change air delay causingmissed co…          negative

[1459 rows x 4 columns]
```

[ ]:
```
original_data['airline_sentiment'].value_counts()
```

[ ]:
```
negative    7430
neutral     2493
positive    1911
Name: airline_sentiment, dtype: int64
```

[ ]:
```
data_test
```

[ ]:
```
      Unnamed: 0            tweet_id  \
0              0  568107472260624384
1              1  568215698524246016
2              2  567842466851905536
3              3  568834824410148864
4              4  569590527349252096
…            …                  …
1310        1313  570060687164067840
1311        1314  570101371409559552
1312        1315  568572753403650049
```

```
1313        1316  567747769176432640
1314        1317  570011378091753472

                                                   text airline_sentiment
0     great job celebrating industry another reason …          positive
1            thanks taking upnotch leinenkugels norfolk           positive
2            put back hold hour completely unacceptable           negative
3                                  thank offer sorted           positive
4     wondering possible colleague andto get earlier…           neutral
…                                                   …                …
1310                        sorry disappointed kid job           negative
1311  stuck onplane dallas thats supposed going okc …          negative
1312  lost wallet flight yesterday houston bogota fi…          negative
1313  travelling pwm atl sunday flight got cancelled…          negative
1314                                            thank           positive

[1315 rows x 4 columns]
```

1 Data Augmentation- start (execute only once)

1.1 Bert based model-transformer .

[ ]:
```
!pip install nlpaug
!pip install transformers
from tqdm.auto import tqdm
from sklearn.utils import shuffle
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Requirement already satisfied: nlpaug in /usr/local/lib/python3.8/dist-packages
(1.1.11)
Requirement already satisfied: gdown>=4.0.0 in /usr/local/lib/python3.8/dist-
packages (from nlpaug) (4.4.0)
Requirement already satisfied: requests>=2.22.0 in
/usr/local/lib/python3.8/dist-packages (from nlpaug) (2.25.1)
Requirement already satisfied: numpy>=1.16.2 in /usr/local/lib/python3.8/dist-
packages (from nlpaug) (1.21.6)
Requirement already satisfied: pandas>=1.2.0 in /usr/local/lib/python3.8/dist-
packages (from nlpaug) (1.3.5)
Requirement already satisfied: six in /usr/local/lib/python3.8/dist-packages
(from gdown>=4.0.0->nlpaug) (1.15.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.8/dist-
packages (from gdown>=4.0.0->nlpaug) (3.9.0)
Requirement already satisfied: tqdm in /usr/local/lib/python3.8/dist-packages
(from gdown>=4.0.0->nlpaug) (4.64.1)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.8/dist-
packages (from gdown>=4.0.0->nlpaug) (4.6.3)
```

```
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.8/dist-
packages (from pandas>=1.2.0->nlpaug) (2022.7)
Requirement already satisfied: python-dateutil>=2.7.3 in
/usr/local/lib/python3.8/dist-packages (from pandas>=1.2.0->nlpaug) (2.8.2)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.8/dist-packages (from requests>=2.22.0->nlpaug)
(2022.12.7)
Requirement already satisfied: chardet<5,>=3.0.2 in
/usr/local/lib/python3.8/dist-packages (from requests>=2.22.0->nlpaug) (4.0.0)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.8/dist-
packages (from requests>=2.22.0->nlpaug) (2.10)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/usr/local/lib/python3.8/dist-packages (from requests>=2.22.0->nlpaug) (1.24.3)
Requirement already satisfied: PySocks!=1.5.7,>=1.5.6 in
/usr/local/lib/python3.8/dist-packages (from requests>=2.22.0->nlpaug) (1.7.1)
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Requirement already satisfied: transformers in /usr/local/lib/python3.8/dist-
packages (4.25.1)
Requirement already satisfied: filelock in /usr/local/lib/python3.8/dist-
packages (from transformers) (3.9.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.8/dist-
packages (from transformers) (21.3)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.8/dist-
packages (from transformers) (4.64.1)
Requirement already satisfied: huggingface-hub<1.0,>=0.10.0 in
/usr/local/lib/python3.8/dist-packages (from transformers) (0.11.1)
Requirement already satisfied: tokenizers!=0.11.3,<0.14,>=0.11.1 in
/usr/local/lib/python3.8/dist-packages (from transformers) (0.13.2)
Requirement already satisfied: requests in /usr/local/lib/python3.8/dist-
packages (from transformers) (2.25.1)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.8/dist-
packages (from transformers) (1.21.6)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.8/dist-
packages (from transformers) (6.0)
Requirement already satisfied: regex!=2019.12.17 in
/usr/local/lib/python3.8/dist-packages (from transformers) (2022.6.2)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.8/dist-packages (from huggingface-
hub<1.0,>=0.10.0->transformers) (4.4.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/usr/local/lib/python3.8/dist-packages (from packaging>=20.0->transformers)
(3.0.9)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.8/dist-packages (from requests->transformers) (2022.12.7)
Requirement already satisfied: chardet<5,>=3.0.2 in
/usr/local/lib/python3.8/dist-packages (from requests->transformers) (4.0.0)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
```

```
/usr/local/lib/python3.8/dist-packages (from requests->transformers) (1.24.3)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.8/dist-
packages (from requests->transformers) (2.10)
```

[ ]:
```
import nlpaug.augmenter.word.context_word_embs as aug
augmenter = aug.ContextualWordEmbsAug(model_path='bert-base-uncased',␣
 ↪action="insert")
def augmentMyData(df, augmenter, repetitions, samples,p):
    augmented_texts = []
    # select only the minority class samples
    spam_df = df[df['airline_sentiment'] == p].reset_index(drop=True) # removes␣
 ↪unecessary index column
    for i in tqdm(np.random.randint(0, len(spam_df), samples)):
        # generating 'n_samples' augmented texts
        for _ in range(repetitions):
            augmented_text = augmenter.augment(spam_df['text'].iloc[i])
            augmented_texts.append(augmented_text)

    d = {
        'airline_sentiment': p,
        'text': augmented_texts
    }
    aug_df = pd.DataFrame(d)
    df = shuffle(df.append(aug_df).reset_index(drop=True))
    return df
```

```
Downloading:   0%|          | 0.00/28.0 [00:00<?, ?B/s]
Downloading:   0%|          | 0.00/570 [00:00<?, ?B/s]
Downloading:   0%|          | 0.00/232k [00:00<?, ?B/s]
Downloading:   0%|          | 0.00/466k [00:00<?, ?B/s]
Downloading:   0%|          | 0.00/440M [00:00<?, ?B/s]
```

[ ]:
```
new_df=pd.DataFrame(data={'text': original_data["text"],'airline_sentiment':␣
 ↪original_data["airline_sentiment"]})
```

[ ]:
```
aug_df = augmentMyData(new_df, augmenter,1,5520,'positive')
aug_df = augmentMyData(aug_df, augmenter,1,4924,'neutral')
```

```
  0%|          | 0/5520 [00:00<?, ?it/s]
  0%|          | 0/4924 [00:00<?, ?it/s]
```

[ ]:
```
import os
aug_df.to_csv('/content/gdrive/MyDrive/Ml_project/Tweets_train_new_bert.csv')
```

1.2 Text attack based

```
[ ]: !pip install textattack
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Collecting textattack
  Downloading textattack-0.3.8-py3-none-any.whl (418 kB)
                             418.7/418.7
KB 8.6 MB/s eta 0:00:00
Collecting lru-dict
  Downloading lru_dict-1.1.8-cp38-cp38-manylinux_2_5_x86_64.manylinux1_x86_64.ma
nylinux_2_17_x86_64.manylinux2014_x86_64.whl (29 kB)
Requirement already satisfied: nltk in /usr/local/lib/python3.8/dist-packages
(from textattack) (3.7)
Collecting num2words
  Downloading num2words-0.5.12-py3-none-any.whl (125 kB)
                            125.2/125.2 KB
16.0 MB/s eta 0:00:00
Collecting word2number
  Downloading word2number-1.1.zip (9.7 kB)
  Preparing metadata (setup.py) … done
Collecting terminaltables
  Downloading terminaltables-3.1.10-py2.py3-none-any.whl (15 kB)
Collecting pycld2
  Downloading pycld2-0.41.tar.gz (41.4 MB)
                             41.4/41.4 MB
17.1 MB/s eta 0:00:00
  Preparing metadata (setup.py) … done
Requirement already satisfied: pandas>=1.0.1 in /usr/local/lib/python3.8/dist-
packages (from textattack) (1.3.5)
Requirement already satisfied: torch!=1.8,>=1.7.0 in
/usr/local/lib/python3.8/dist-packages (from textattack) (1.13.1+cu116)
Collecting pinyin==0.4.0
  Downloading pinyin-0.4.0.tar.gz (3.6 MB)
                             3.6/3.6 MB
90.0 MB/s eta 0:00:00
  Preparing metadata (setup.py) … done
Requirement already satisfied: editdistance in /usr/local/lib/python3.8/dist-
packages (from textattack) (0.5.3)
Requirement already satisfied: scipy>=1.4.1 in /usr/local/lib/python3.8/dist-
packages (from textattack) (1.7.3)
Collecting flair
  Downloading flair-0.11.3-py3-none-any.whl (401 kB)
                            401.9/401.9 KB
39.3 MB/s eta 0:00:00
Collecting transformers>=4.21.0
```

```
  Downloading transformers-4.25.1-py3-none-any.whl (5.8 MB)
                             5.8/5.8 MB
88.6 MB/s eta 0:00:00
Collecting bert-score>=0.3.5
  Downloading bert_score-0.3.12-py3-none-any.whl (60 kB)
                             60.8/60.8 KB
8.1 MB/s eta 0:00:00
Collecting lemminflect
  Downloading lemminflect-0.2.3-py3-none-any.whl (769 kB)
                            769.7/769.7 KB
42.4 MB/s eta 0:00:00
Collecting OpenHowNet
  Downloading OpenHowNet-2.0-py3-none-any.whl (18 kB)
Requirement already satisfied: filelock in /usr/local/lib/python3.8/dist-
packages (from textattack) (3.9.0)
Collecting datasets==2.4.0
  Downloading datasets-2.4.0-py3-none-any.whl (365 kB)
                            365.7/365.7 KB
21.8 MB/s eta 0:00:00
Collecting language-tool-python
  Downloading language_tool_python-2.7.1-py3-none-any.whl (34 kB)
Requirement already satisfied: jieba in /usr/local/lib/python3.8/dist-packages
(from textattack) (0.42.1)
Requirement already satisfied: numpy>=1.21.0 in /usr/local/lib/python3.8/dist-
packages (from textattack) (1.21.6)
Requirement already satisfied: PySocks!=1.5.7,>=1.5.6 in
/usr/local/lib/python3.8/dist-packages (from textattack) (1.7.1)
Requirement already satisfied: tqdm in /usr/local/lib/python3.8/dist-packages
(from textattack) (4.64.1)
Requirement already satisfied: more-itertools in /usr/local/lib/python3.8/dist-
packages (from textattack) (9.0.0)
Requirement already satisfied: click<8.1.0 in /usr/local/lib/python3.8/dist-
packages (from textattack) (7.1.2)
Collecting xxhash
  Downloading
xxhash-3.2.0-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (213 kB)
                            213.0/213.0 KB
24.4 MB/s eta 0:00:00
Requirement already satisfied: packaging in /usr/local/lib/python3.8/dist-
packages (from datasets==2.4.0->textattack) (21.3)
Collecting multiprocess
  Downloading multiprocess-0.70.14-py38-none-any.whl (132 kB)
                            132.0/132.0 KB
16.8 MB/s eta 0:00:00
Collecting dill<0.3.6
  Downloading dill-0.3.5.1-py2.py3-none-any.whl (95 kB)
                             95.8/95.8 KB
13.3 MB/s eta 0:00:00
```

```
Requirement already satisfied: pyarrow>=6.0.0 in
/usr/local/lib/python3.8/dist-packages (from datasets==2.4.0->textattack)
(9.0.0)
Requirement already satisfied: aiohttp in /usr/local/lib/python3.8/dist-packages
(from datasets==2.4.0->textattack) (3.8.3)
Collecting huggingface-hub<1.0.0,>=0.1.0
  Downloading huggingface_hub-0.11.1-py3-none-any.whl (182 kB)
                         182.4/182.4 KB
19.0 MB/s eta 0:00:00
Requirement already satisfied: fsspec[http]>=2021.11.1 in
/usr/local/lib/python3.8/dist-packages (from datasets==2.4.0->textattack)
(2022.11.0)
Collecting responses<0.19
  Downloading responses-0.18.0-py3-none-any.whl (38 kB)
Requirement already satisfied: requests>=2.19.0 in
/usr/local/lib/python3.8/dist-packages (from datasets==2.4.0->textattack)
(2.25.1)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.8/dist-
packages (from bert-score>=0.3.5->textattack) (3.2.2)
Requirement already satisfied: python-dateutil>=2.7.3 in
/usr/local/lib/python3.8/dist-packages (from pandas>=1.0.1->textattack) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.8/dist-
packages (from pandas>=1.0.1->textattack) (2022.7)
Requirement already satisfied: typing-extensions in
/usr/local/lib/python3.8/dist-packages (from torch!=1.8,>=1.7.0->textattack)
(4.4.0)
Requirement already satisfied: regex!=2019.12.17 in
/usr/local/lib/python3.8/dist-packages (from transformers>=4.21.0->textattack)
(2022.6.2)
Collecting tokenizers!=0.11.3,<0.14,>=0.11.1
  Downloading
tokenizers-0.13.2-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (7.6
MB)
                           7.6/7.6 MB
92.7 MB/s eta 0:00:00
Requirement already satisfied: pyyaml>=5.1 in
/usr/local/lib/python3.8/dist-packages (from transformers>=4.21.0->textattack)
(6.0)
Collecting pptree
  Downloading pptree-3.1.tar.gz (3.0 kB)
  Preparing metadata (setup.py) … done
Requirement already satisfied: tabulate in /usr/local/lib/python3.8/dist-
packages (from flair->textattack) (0.8.10)
Collecting conllu>=4.0
  Downloading conllu-4.5.2-py2.py3-none-any.whl (16 kB)
Collecting segtok>=1.5.7
  Downloading segtok-1.5.11-py3-none-any.whl (24 kB)
Collecting mpld3==0.3
```

```
  Downloading mpld3-0.3.tar.gz (788 kB)
                           788.5/788.5 KB
57.3 MB/s eta 0:00:00
  Preparing metadata (setup.py) … done
Collecting ftfy
  Downloading ftfy-6.1.1-py3-none-any.whl (53 kB)
                           53.1/53.1 KB
5.7 MB/s eta 0:00:00
Requirement already satisfied: gdown==4.4.0 in
/usr/local/lib/python3.8/dist-packages (from flair->textattack) (4.4.0)
Requirement already satisfied: gensim>=3.4.0 in /usr/local/lib/python3.8/dist-
packages (from flair->textattack) (3.6.0)
Collecting sqlitedict>=1.6.0
  Downloading sqlitedict-2.1.0.tar.gz (21 kB)
  Preparing metadata (setup.py) … done
Collecting deprecated>=1.2.4
  Downloading Deprecated-1.2.13-py2.py3-none-any.whl (9.6 kB)
Collecting bpemb>=0.3.2
  Downloading bpemb-0.3.4-py3-none-any.whl (19 kB)
Collecting wikipedia-api
  Downloading Wikipedia_API-0.5.8-py3-none-any.whl (13 kB)
Collecting hyperopt>=0.2.7
  Downloading hyperopt-0.2.7-py2.py3-none-any.whl (1.6 MB)
                           1.6/1.6 MB
60.5 MB/s eta 0:00:00
Requirement already satisfied: lxml in /usr/local/lib/python3.8/dist-
packages (from flair->textattack) (4.9.2)
Collecting langdetect
  Downloading langdetect-1.0.9.tar.gz (981 kB)
                           981.5/981.5 KB
32.1 MB/s eta 0:00:00
  Preparing metadata (setup.py) … done
Collecting janome
  Downloading Janome-0.4.2-py2.py3-none-any.whl (19.7 MB)
                           19.7/19.7 MB
64.8 MB/s eta 0:00:00
Collecting sentencepiece==0.1.95
  Downloading sentencepiece-0.1.95-cp38-cp38-manylinux2014_x86_64.whl (1.2 MB)
                           1.2/1.2 MB
72.3 MB/s eta 0:00:00
Collecting konoha<5.0.0,>=4.0.0
  Downloading konoha-4.6.5-py3-none-any.whl (20 kB)
Requirement already satisfied: scikit-learn>=0.21.3 in
/usr/local/lib/python3.8/dist-packages (from flair->textattack) (1.0.2)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.8/dist-
packages (from gdown==4.4.0->flair->textattack) (4.6.3)
Requirement already satisfied: six in /usr/local/lib/python3.8/dist-packages
(from gdown==4.4.0->flair->textattack) (1.15.0)
```

```
Requirement already satisfied: joblib in /usr/local/lib/python3.8/dist-packages
(from nltk->textattack) (1.2.0)
Collecting docopt>=0.6.2
  Downloading docopt-0.6.2.tar.gz (25 kB)
  Preparing metadata (setup.py) … done
Requirement already satisfied: setuptools in /usr/local/lib/python3.8/dist-
packages (from OpenHowNet->textattack) (57.4.0)
Collecting anytree
  Downloading anytree-2.8.0-py2.py3-none-any.whl (41 kB)
                           41.7/41.7 KB
2.9 MB/s eta 0:00:00
Requirement already satisfied: wrapt<2,>=1.10 in
/usr/local/lib/python3.8/dist-packages (from
deprecated>=1.2.4->flair->textattack) (1.14.1)
Requirement already satisfied: frozenlist>=1.1.1 in
/usr/local/lib/python3.8/dist-packages (from
aiohttp->datasets==2.4.0->textattack) (1.3.3)
Requirement already satisfied: aiosignal>=1.1.2 in
/usr/local/lib/python3.8/dist-packages (from
aiohttp->datasets==2.4.0->textattack) (1.3.1)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.8/dist-
packages (from aiohttp->datasets==2.4.0->textattack) (22.2.0)
Requirement already satisfied: charset-normalizer<3.0,>=2.0 in
/usr/local/lib/python3.8/dist-packages (from
aiohttp->datasets==2.4.0->textattack) (2.1.1)
Requirement already satisfied: async-timeout<5.0,>=4.0.0a3 in
/usr/local/lib/python3.8/dist-packages (from
aiohttp->datasets==2.4.0->textattack) (4.0.2)
Requirement already satisfied: multidict<7.0,>=4.5 in
/usr/local/lib/python3.8/dist-packages (from
aiohttp->datasets==2.4.0->textattack) (6.0.4)
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.8/dist-
packages (from aiohttp->datasets==2.4.0->textattack) (1.8.2)
Requirement already satisfied: smart-open>=1.2.1 in
/usr/local/lib/python3.8/dist-packages (from gensim>=3.4.0->flair->textattack)
(6.3.0)
Requirement already satisfied: future in /usr/local/lib/python3.8/dist-packages
(from hyperopt>=0.2.7->flair->textattack) (0.16.0)
Collecting py4j
  Downloading py4j-0.10.9.7-py2.py3-none-any.whl (200 kB)
                           200.5/200.5 KB
12.3 MB/s eta 0:00:00
Requirement already satisfied: networkx>=2.2 in
/usr/local/lib/python3.8/dist-packages (from hyperopt>=0.2.7->flair->textattack)
(3.0)
Requirement already satisfied: cloudpickle in /usr/local/lib/python3.8/dist-
packages (from hyperopt>=0.2.7->flair->textattack) (2.2.0)
Collecting importlib-metadata<4.0.0,>=3.7.0
```

```
  Downloading importlib_metadata-3.10.1-py3-none-any.whl (14 kB)
Collecting overrides<4.0.0,>=3.0.0
  Downloading overrides-3.1.0.tar.gz (11 kB)
  Preparing metadata (setup.py) … done
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in
/usr/local/lib/python3.8/dist-packages (from matplotlib->bert-
score>=0.3.5->textattack) (3.0.9)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.8/dist-packages (from matplotlib->bert-
score>=0.3.5->textattack) (1.4.4)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.8/dist-
packages (from matplotlib->bert-score>=0.3.5->textattack) (0.11.0)
Requirement already satisfied: chardet<5,>=3.0.2 in
/usr/local/lib/python3.8/dist-packages (from
requests>=2.19.0->datasets==2.4.0->textattack) (4.0.0)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/usr/local/lib/python3.8/dist-packages (from
requests>=2.19.0->datasets==2.4.0->textattack) (1.24.3)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.8/dist-
packages (from requests>=2.19.0->datasets==2.4.0->textattack) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.8/dist-packages (from
requests>=2.19.0->datasets==2.4.0->textattack) (2022.12.7)
Collecting urllib3<1.27,>=1.21.1
  Downloading urllib3-1.26.14-py2.py3-none-any.whl (140 kB)
                           140.6/140.6 KB
18.4 MB/s eta 0:00:00
Requirement already satisfied: threadpoolctl>=2.0.0 in
/usr/local/lib/python3.8/dist-packages (from scikit-
learn>=0.21.3->flair->textattack) (3.1.0)
Requirement already satisfied: wcwidth>=0.2.5 in /usr/local/lib/python3.8/dist-
packages (from ftfy->flair->textattack) (0.2.5)
Collecting multiprocess
  Downloading multiprocess-0.70.13-py38-none-any.whl (131 kB)
                           131.4/131.4 KB
16.4 MB/s eta 0:00:00
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.8/dist-
packages (from importlib-
metadata<4.0.0,>=3.7.0->konoha<5.0.0,>=4.0.0->flair->textattack) (3.11.0)
Building wheels for collected packages: pinyin, mpld3, pycld2, word2number,
docopt, sqlitedict, langdetect, pptree, overrides
  Building wheel for pinyin (setup.py) … done
  Created wheel for pinyin: filename=pinyin-0.4.0-py3-none-any.whl size=3630495
sha256=548ae5e204f477089e0bc30523e05427992bd0625d21aef764cd07b212c4f965
  Stored in directory: /root/.cache/pip/wheels/d1/2a/d9/9c0f787a4d55f9a9eca26d32
2eafbe083bab41cb9bffb2e6e8
  Building wheel for mpld3 (setup.py) … done
  Created wheel for mpld3: filename=mpld3-0.3-py3-none-any.whl size=116702
```

```
sha256=f44fb118f91cb3f5fd755b18703cf0ab36424fb1fac69bd955b6f62cbc1a4ecb
  Stored in directory: /root/.cache/pip/wheels/3d/9f/9d/d806a20bd97bc7076d724fa3
e69fa5be61836ba16b2ffa6126
  Building wheel for pycld2 (setup.py) … done
  Created wheel for pycld2: filename=pycld2-0.41-cp38-cp38-linux_x86_64.whl
size=9917393
sha256=80dcf5286492df4e7ef343f4d0f0ed49cfd5c2f741bc51a66d2db388d1e44e58
  Stored in directory: /root/.cache/pip/wheels/2b/3a/82/d990040cbe6c3527732e931e
2925785e83fe9aaa5a11c313ca
  Building wheel for word2number (setup.py) … done
  Created wheel for word2number: filename=word2number-1.1-py3-none-any.whl
size=5582
sha256=ce2af6fc5147d8a9b5429efd9eac7cc8b54e056d541e042b318e8189c461f746
  Stored in directory: /root/.cache/pip/wheels/cb/f3/5a/d88198fdeb46781ddd7e7f26
53061af83e7adb2a076d8886d6
  Building wheel for docopt (setup.py) … done
  Created wheel for docopt: filename=docopt-0.6.2-py2.py3-none-any.whl
size=13723
sha256=ad6a74db92ac36a2fd872110a4bf29732d78c671fcd7356f306f64bab6583e5a
  Stored in directory: /root/.cache/pip/wheels/56/ea/58/ead137b087d9e326852a8513
51d1debf4ada529b6ac0ec4e8c
  Building wheel for sqlitedict (setup.py) … done
  Created wheel for sqlitedict: filename=sqlitedict-2.1.0-py3-none-any.whl
size=16869
sha256=0a39db83177dced8d58cf872bfc2fa5f4a1d3505057e0e645328507381aacd58
  Stored in directory: /root/.cache/pip/wheels/04/c6/16/46e174009277f9bccdaa7215
a243939d2f70180804b249bf3a
  Building wheel for langdetect (setup.py) … done
  Created wheel for langdetect: filename=langdetect-1.0.9-py3-none-any.whl
size=993242
sha256=70185a6a50210e407a0b2fa2cd280114d8a692b996b62069c902fdc32e142c23
  Stored in directory: /root/.cache/pip/wheels/13/c7/b0/79f66658626032e78fc1a831
03690ef6797d551cb22e56e734
  Building wheel for pptree (setup.py) … done
  Created wheel for pptree: filename=pptree-3.1-py3-none-any.whl size=4629
sha256=193f2bae0aae29ac240fc59e0af8fd378676da6c2a74a7d2afae927c0c9e96c9
  Stored in directory: /root/.cache/pip/wheels/e1/8b/30/5b20240d3d13a9dfafb6a6dd
49d1b541c86d39812cb3690edf
  Building wheel for overrides (setup.py) … done
  Created wheel for overrides: filename=overrides-3.1.0-py3-none-any.whl
size=10187
sha256=cded778845325184f23211dc59854b24d78c76e9a0f7cb3c36800fd0842650a5
  Stored in directory: /root/.cache/pip/wheels/6a/4f/72/28857f75625b263e2e3f5ab2
fc4416c0a85960ac6485007eaa
Successfully built pinyin mpld3 pycld2 word2number docopt sqlitedict langdetect
pptree overrides
Installing collected packages: word2number, tokenizers, sqlitedict,
sentencepiece, pycld2, py4j, pptree, pinyin, overrides, mpld3, lru-dict, janome,
```

```
docopt, xxhash, urllib3, terminaltables, segtok, num2words, lemminflect,
langdetect, importlib-metadata, ftfy, dill, deprecated, conllu, anytree,
multiprocess, hyperopt, wikipedia-api, responses, OpenHowNet, language-tool-
python, konoha, huggingface-hub, bpemb, transformers, datasets, flair, bert-
score, textattack
  Attempting uninstall: urllib3
    Found existing installation: urllib3 1.24.3
    Uninstalling urllib3-1.24.3:
      Successfully uninstalled urllib3-1.24.3
  Attempting uninstall: importlib-metadata
    Found existing installation: importlib-metadata 6.0.0
    Uninstalling importlib-metadata-6.0.0:
      Successfully uninstalled importlib-metadata-6.0.0
  Attempting uninstall: dill
    Found existing installation: dill 0.3.6
    Uninstalling dill-0.3.6:
      Successfully uninstalled dill-0.3.6
  Attempting uninstall: hyperopt
    Found existing installation: hyperopt 0.1.2
    Uninstalling hyperopt-0.1.2:
      Successfully uninstalled hyperopt-0.1.2
ERROR: pip's dependency resolver does not currently take into account all
the packages that are installed. This behaviour is the source of the following
dependency conflicts.
markdown 3.4.1 requires importlib-metadata>=4.4; python_version < "3.10", but
you have importlib-metadata 3.10.1 which is incompatible.
gym 0.25.2 requires importlib-metadata>=4.8.0; python_version < "3.10", but you
have importlib-metadata 3.10.1 which is incompatible.
Successfully installed OpenHowNet-2.0 anytree-2.8.0 bert-score-0.3.12
bpemb-0.3.4 conllu-4.5.2 datasets-2.4.0 deprecated-1.2.13 dill-0.3.5.1
docopt-0.6.2 flair-0.11.3 ftfy-6.1.1 huggingface-hub-0.11.1 hyperopt-0.2.7
importlib-metadata-3.10.1 janome-0.4.2 konoha-4.6.5 langdetect-1.0.9 language-
tool-python-2.7.1 lemminflect-0.2.3 lru-dict-1.1.8 mpld3-0.3
multiprocess-0.70.13 num2words-0.5.12 overrides-3.1.0 pinyin-0.4.0 pptree-3.1
py4j-0.10.9.7 pycld2-0.41 responses-0.18.0 segtok-1.5.11 sentencepiece-0.1.95
sqlitedict-2.1.0 terminaltables-3.1.10 textattack-0.3.8 tokenizers-0.13.2
transformers-4.25.1 urllib3-1.26.14 wikipedia-api-0.5.8 word2number-1.1
xxhash-3.2.0
```

[ ]:
```
from textattack.augmentation import WordNetAugmenter
```

```
textattack: Updating TextAttack package dependencies.
textattack: Downloading NLTK required packages.
[nltk_data] Downloading package averaged_perceptron_tagger to
```

```
[nltk_data]     /root/nltk_data…
[nltk_data]   Unzipping taggers/averaged_perceptron_tagger.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data…
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package omw to /root/nltk_data…
[nltk_data] Downloading package universal_tagset to /root/nltk_data…
[nltk_data]   Unzipping taggers/universal_tagset.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data…
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data…
[nltk_data]   Unzipping tokenizers/punkt.zip.
/usr/local/lib/python3.8/dist-packages/torch/cuda/__init__.py:497: UserWarning:
Can't initialize NVML
  warnings.warn("Can't initialize NVML")
```

[ ]:
```
from sklearn.utils import shuffle
augmented_texts = []
df=[]
df=original_data # origianl_dat:train data
df=df.drop(labels=['tweet_id'], axis=1)
wordnet_aug = WordNetAugmenter()
temp=df[df["airline_sentiment"]=='positive']
l=i=0
while l<=5519:
 if i>=1911:
   i=0
 k=temp["text"].iloc[i]
 augmented_text = wordnet_aug.augment(k)
 augmented_texts.append(augmented_text)
 d = {
      'airline_sentiment': 'positive',
      'text': augmented_texts
     }
 aug_df = pd.DataFrame(d)
 i+=1
 l+=1
df = shuffle(df.append(aug_df).reset_index(drop=True))
```

```
[nltk_data] Downloading package omw-1.4 to /root/nltk_data…
[nltk_data]   Package omw-1.4 is already up-to-date!
```

[ ]:
```
augmented_texts = []
wordnet_aug = WordNetAugmenter()
temp=df[df["airline_sentiment"]=='neutral']
l=i=0
while l<=4939:# denote how many neutral strings to be generated
 if i>=2490:
```

```
  i=0
 k=temp["text"].iloc[i]
 augmented_text = wordnet_aug.augment(k)
 augmented_texts.append(augmented_text)
 d = {
      'airline_sentiment': 'neutral',
      'text': augmented_texts
     }
 aug_df = pd.DataFrame(d)
 i+=1
 l+=1
df = shuffle(df.append(aug_df).reset_index(drop=True))
```

```
[nltk_data] Downloading package omw-1.4 to /root/nltk_data…
[nltk_data]   Package omw-1.4 is already up-to-date!
```

[ ]:
```
df['airline_sentiment'].value_counts()
```

[ ]:
```
neutral     7433
positive    7431
negative    7430
Name: airline_sentiment, dtype: int64
```

[ ]:
```
#saving augmented df , preprocessed test and valid data
import os
df.to_csv('/content/gdrive/MyDrive/Ml_project/Tweets_test_new_word_attack.csv')
```

2 Data augmentation -end

3 balanced training data loading from google cloud

[ ]:
```
data=pd.read_csv('/content/gdrive/MyDrive/Ml_project/
 ↪Tweets_test_new_word_attack.csv')
data['text'] = data['text'].replace(r'[^A-Za-z ]+', '', regex=True)
```

[ ]:
```
x_train=data['text']
y_train=data['airline_sentiment']
x_test=data_test['text']
y_test=data_test['airline_sentiment']
x_valid=data_valid['text']
y_valid=data_valid['airline_sentiment']
```

[ ]:
```
#final checking
for i in range(0,2):
 print(x_train[x_train.apply(lambda x: len(x)==i)])
```

```
for i in range(0,2):
 print(x_test[x_test.apply(lambda x: len(x)==i)])

for i in range(0,2):
 print(x_valid[x_valid.apply(lambda x: len(x)==i)])
```

```
Series([], Name: text, dtype: object)
Series([], Name: text, dtype: object)
Series([], Name: text, dtype: object)
Series([], Name: text, dtype: object)
Series([], Name: text, dtype: object)
Series([], Name: text, dtype: object)
```

[ ]:
```
p=[y_train,y_test,y_valid]

for p in p:
  print("class wise count of {}".format(c))
  print(p.value_counts())
```

```
class wise count of ['y_train', 'y_test', 'y_valid']
neutral     7433
positive    7431
negative    7430
Name: airline_sentiment, dtype: int64
class wise count of ['y_train', 'y_test', 'y_valid']
negative    825
neutral     277
positive    213
Name: airline_sentiment, dtype: int64
class wise count of ['y_train', 'y_test', 'y_valid']
negative    918
neutral     306
positive    235
Name: airline_sentiment, dtype: int64
```

Preprocessing stage 2

[ ]:
```
import keras
num_classes=3
# Using map function
y_train = y_train.map({'positive': 1, 'negative': 2, 'neutral' : 0})
y_test = y_test.map({'positive': 1, 'negative': 2, 'neutral' : 0})
y_valid = y_valid.map({'positive': 1, 'negative': 2, 'neutral' : 0})
y_train_new=np.array(y_train)
y_test_new=np.array(y_test)
y_valid_new=np.array(y_valid)
```

[ ]:
```
y_train_new_tf = keras.utils.to_categorical(y_train_new, num_classes)
y_test_new_tf = keras.utils.to_categorical(y_test_new, num_classes)
y_valid_new_tf = keras.utils.to_categorical(y_valid_new, num_classes)
```

[ ]:
```
from keras.preprocessing import sequence
from keras.metrics import FalseNegatives, Precision, Recall, TruePositives,␣
 ↪Accuracy, TrueNegatives, FalsePositives
from tensorflow_addons.metrics import F1Score
from pandas._libs.algos import pad_2d_inplace
from keras.preprocessing.text import Tokenizer
from keras_preprocessing.sequence import pad_sequences
tokenizer=Tokenizer(10000)
tokenizer_test=Tokenizer(10000)
tokenizer.fit_on_texts(x_train)
sequences=tokenizer.texts_to_sequences(x_train)
tokenizer_test.fit_on_texts(x_test)
sequences_test=tokenizer_test.texts_to_sequences(x_test)
tokenizer_test.fit_on_texts(x_valid)
sequences_valid=tokenizer_test.texts_to_sequences(x_valid)
word_index=tokenizer.word_index
word_index_test=tokenizer_test.word_index
x_train_new=pad_sequences(sequences,100)
x_test_new=pad_sequences(sequences_test,100)
x_valid_new=pad_sequences(sequences_valid,100)
```

[ ]:
```
x_test_new.shape
```

[ ]:
```
(1464, 100)
```

[ ]:
```
y_train_new.shape
```

[ ]:
```
(11858,)
```

4 Bag of words using countvectorizer

[ ]:
```
# input data preparation
from sklearn.feature_extraction.text import CountVectorizer
train_data_mnb = x_train
train_labels_mnb =y_train

valid_data_mnb=x_valid
valid_labels_mnb=y_valid

test_data_mnb=x_test
test_labels_mnb=y_test
# vectorizing
```

```
vector = CountVectorizer(stop_words='english',lowercase=1)
train_vectors = vector.fit_transform(train_data_mnb)
test_vectors = vector.transform(test_data_mnb)
valid_vectors = vector.transform(valid_data_mnb)
```

#Tf-idf vectorization

[ ]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
# input data preparation
train_data_tfidf = x_train
train_labels_tfidf =y_train

test_data_tfidf=x_test
test_labels_tfidf=y_test
# vectorizing
vector = TfidfVectorizer()
train_vectors_tfidf = vector.fit_transform(train_data_tfidf)
test_vectors_tfidf = vector.transform(test_data_tfidf)
```

5 Bidirectional LSTM Network

5.1 Embedding leading using pretrained GloVe method

Download the embedding and download it to /content/gdrive/MyDrive/Ml_project/ before loading the embedding.

[ ]:
```
#parsing the GloVe Word-embedding file
glove_dir="/content/gdrive/MyDrive/Ml_project/glove.twitter.27B.100d.txt"
embeddings_index={}
g_f=open(glove_dir)
for line in g_f:
  values=line.split()
  word=values[0]
  coef=np.asarray(values[1:], dtype='float32')
  embeddings_index[word]=coef
g_f.close()
print('Found {} word vectors.'.format(len(embeddings_index)) )
```

```
Found 1193514 word vectors.
```

[ ]:
```
#preparing the GloVe word-embeddings matrix
embedding_dim=100
max_words=10000
embedding_matrix=np.zeros((max_words,embedding_dim))
for w, h in word_index.items():
  if h<max_words:
    embedding_vector=embeddings_index.get(w)
```

```
    if embedding_vector is not None:
      embedding_matrix[h]=embedding_vector
```

5.2 definying the LSTM model

[ ]:
```
#definying model
from keras.models import Sequential
from keras.layers import Embedding,Flatten,Dense,Dropout,LSTM,Bidirectional
model_2=Sequential()
model_2.add(Embedding(max_words,embedding_dim,input_length=100))
#model_1.add(Flatten())
model_2.add(Bidirectional(LSTM(64,dropout=0.5,recurrent_dropout=0.
 ↪5,return_sequences=0)))
model_2.add(Dense(1024,activation='relu'))
model_2.add(Dropout(0.5))
model_2.add(Dense(1024,activation='relu'))
model_2.add(Dropout(0.5))
model_2.add(Dense(128,activation='relu'))
model_2.add(Dropout(0.5))
model_2.add(Dense(3,activation='softmax'))
model_2.layers[0].set_weights([embedding_matrix])
model_2.layers[0].trainable=False # embedding should not be trained
model_2.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 100, 100)          1000000

 bidirectional (Bidirectiona  (None, 128)              84480
 l)

 dense (Dense)               (None, 1024)              132096

 dropout (Dropout)           (None, 1024)              0

 dense_1 (Dense)             (None, 1024)              1049600

 dropout_1 (Dropout)         (None, 1024)              0

 dense_2 (Dense)             (None, 128)               131200

 dropout_2 (Dropout)         (None, 128)               0

 dense_3 (Dense)             (None, 3)                 387
```

```
=================================================================
Total params: 2,397,763
Trainable params: 1,397,763
Non-trainable params: 1,000,000
_________________________________________________________________
```

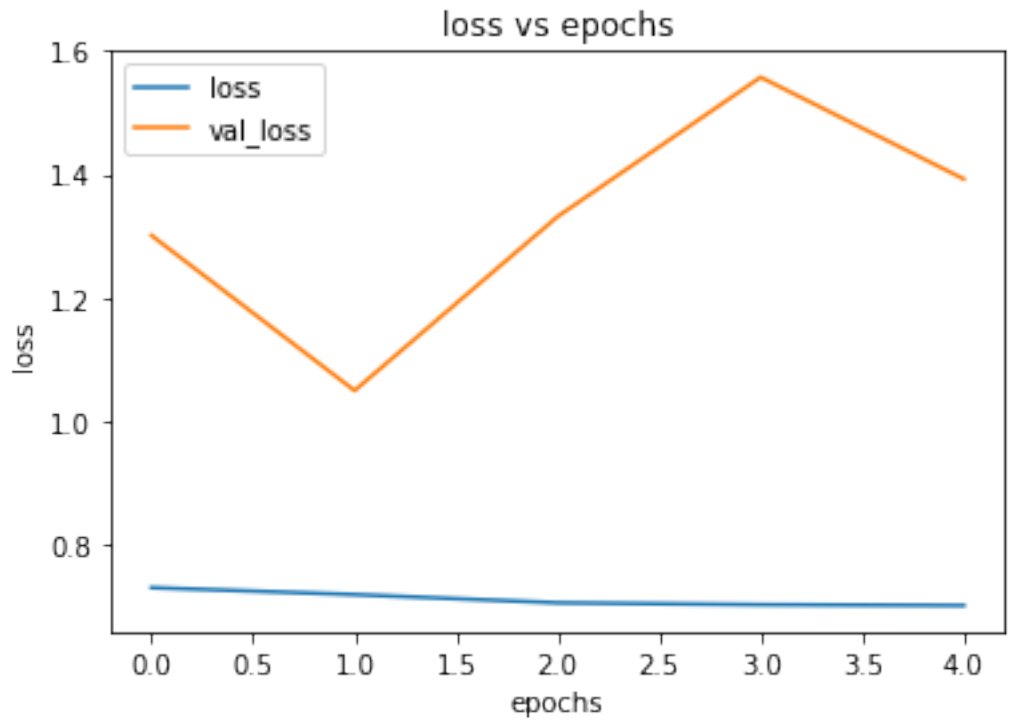
5.3 Training

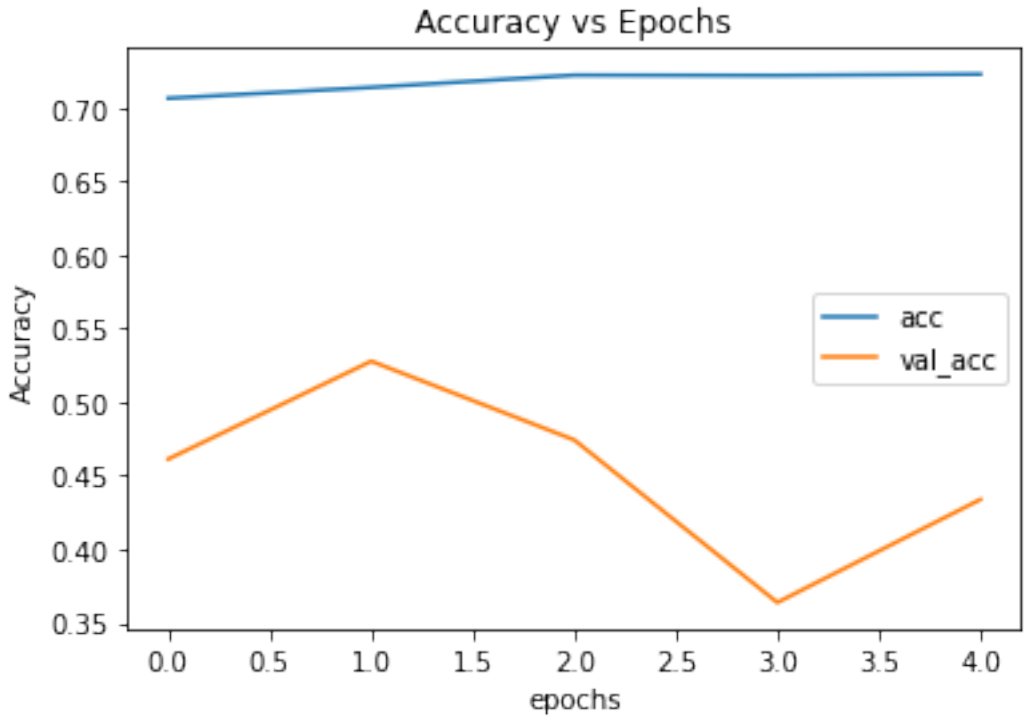
[ ]:
```
from sklearn.metrics import f1_score
from tensorflow_addons.metrics import F1Score
metrics= [Precision(), Recall(),'acc',F1Score(num_classes=3)]
# compile the model
model_2.compile(optimizer=keras.optimizers.RMSprop(),␣
 ↪loss='categorical_crossentropy', metrics=metrics)


# fit the model
history_2 = model_2.fit(x_train_new, y_train_new_tf,␣
 ↪epochs=3,verbose=1,validation_data=(x_valid_new,y_valid_new_tf),batch_size=32)
```

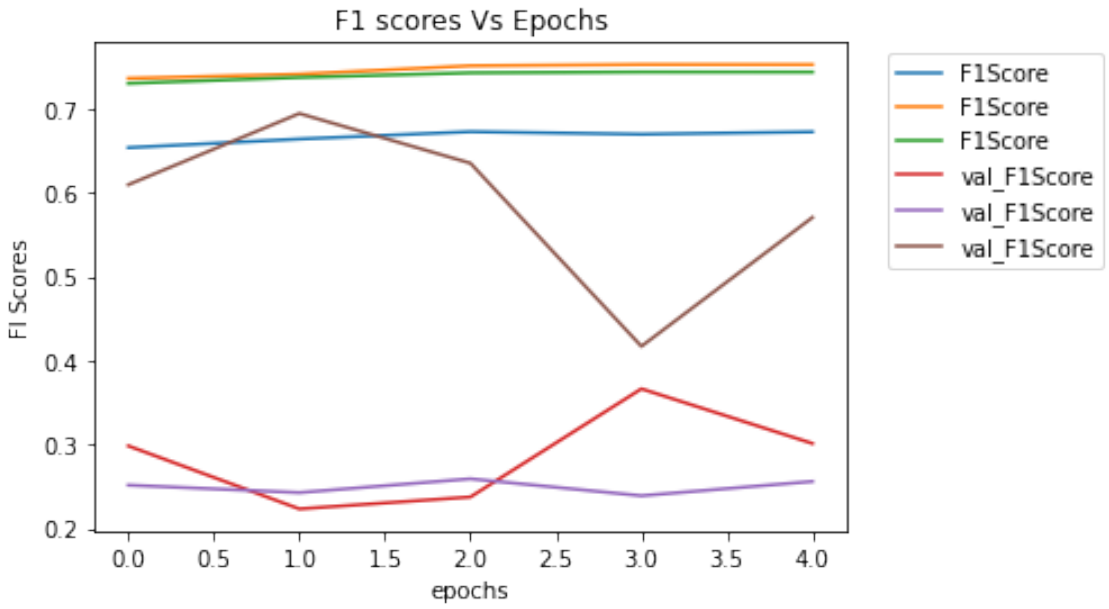
```
Epoch 1/5
697/697 [==============================] - 201s 276ms/step - loss: 0.7290 -
precision_1: 0.7597 - recall_1: 0.6170 - acc: 0.7062 - f1_score: 0.7065 -
val_loss: 1.3007 - val_precision_1: 0.4887 - val_recall_1: 0.3557 - val_acc:
0.4613 - val_f1_score: 0.3862
Epoch 2/5
697/697 [==============================] - 191s 274ms/step - loss: 0.7177 -
precision_1: 0.7632 - recall_1: 0.6244 - acc: 0.7136 - f1_score: 0.7139 -
val_loss: 1.0489 - val_precision_1: 0.6112 - val_recall_1: 0.3825 - val_acc:
0.5278 - val_f1_score: 0.3865
Epoch 3/5
697/697 [==============================] - 191s 274ms/step - loss: 0.7044 -
precision_1: 0.7737 - recall_1: 0.6351 - acc: 0.7219 - f1_score: 0.7220 -
val_loss: 1.3314 - val_precision_1: 0.5045 - val_recall_1: 0.3838 - val_acc:
0.4743 - val_f1_score: 0.3770
Epoch 4/5
697/697 [==============================] - 200s 286ms/step - loss: 0.7014 -
precision_1: 0.7754 - recall_1: 0.6312 - acc: 0.7217 - f1_score: 0.7218 -
val_loss: 1.5576 - val_precision_1: 0.3595 - val_recall_1: 0.2735 - val_acc:
0.3639 - val_f1_score: 0.3405
Epoch 5/5
697/697 [==============================] - 195s 280ms/step - loss: 0.7000 -
precision_1: 0.7730 - recall_1: 0.6333 - acc: 0.7226 - f1_score: 0.7227 -
val_loss: 1.3925 - val_precision_1: 0.4847 - val_recall_1: 0.2714 - val_acc:
0.4339 - val_f1_score: 0.3757
```

5.4 Results

[ ]:
```
import matplotlib.pyplot as plt
ax=plt.plot(history_2.history['loss'], label='loss')
ax=plt.plot(history_2.history['val_loss'], label='val_loss')
plt.xlabel("epochs")
plt.ylabel("loss")
plt.title("loss vs epochs")
plt.legend()
plt.show()

plt.plot(history_2.history['acc'], label='acc')
plt.plot(history_2.history['val_acc'], label='val_acc')
plt.xlabel("epochs")
plt.ylabel("Accuracy")
plt.title("Accuracy vs Epochs")
plt.legend(loc="center right")
plt.show()


plt.plot(history_2.history['f1_score'], label='F1Score')
plt.plot(history_2.history['val_f1_score'], label='val_F1Score')
plt.xlabel("epochs")
plt.ylabel("Fl Scores")
plt.title(" F1 scores Vs Epochs ")
plt.legend(bbox_to_anchor=(1.04, 1), loc="upper left")
plt.show()
```


loss vs epochs
loss
val_loss
1.6
1.4
1.2
1.0
0.8
loss
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
epochs


Accuracy vs Epochs
0.70
0.65
0.60
0.55
0.50
0.45
0.40
0.35
Accuracy
acc
val_acc
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
epochs



[ ]:
```
from sklearn.metrics import precision_score,␣
↪recall_score,f1_score,accuracy_score
prediction_bnn=np.argmax(model_2.predict_on_batch(x_test_new),axis=-1)
from sklearn.metrics import classification_report, confusion_matrix
# print classification report
print(classification_report(y_test_new, prediction_bnn))
```

```
              precision    recall  f1-score   support

           0       0.33      0.38      0.35       277
           1       0.20      0.32      0.25       213
           2       0.72      0.57      0.63       825

    accuracy                           0.49      1315
   macro avg       0.42      0.42      0.41      1315
weighted avg       0.55      0.49      0.51      1315
```

[ ]:
```
model_2.save('/content/gdrive/MyDrive/Ml_project/
↪bi_lstm64_dns1024_1024_128acc56.h5')
```

6 loading earlier version of trained model from google drive-LSTM (Additional Work)

[ ]:
```
from tensorflow import keras
model_l = keras.models.load_model('/content/gdrive/MyDrive/Ml_project/
 ↪bi_lstm64_dns1024_1024_128acc56.h5')
```

[ ]:
```
model_l.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 100, 100)          1000000

 bidirectional (Bidirectiona  (None, 128)              84480
 l)

 dense (Dense)               (None, 1024)              132096

 dropout (Dropout)           (None, 1024)              0

 dense_1 (Dense)             (None, 1024)              1049600

 dropout_1 (Dropout)         (None, 1024)              0

 dense_2 (Dense)             (None, 128)               131200

 dropout_2 (Dropout)         (None, 128)               0

 dense_3 (Dense)             (None, 3)                 387

=================================================================
Total params: 2,397,763
Trainable params: 1,397,763
Non-trainable params: 1,000,000
_________________________________________________________________
```

[ ]:
```
from sklearn.metrics import precision_score,␣
 ↪recall_score,f1_score,accuracy_score
prediction_bnn_l=np.argmax(model_l.predict_on_batch(x_test_new),axis=-1)
from sklearn.metrics import classification_report, confusion_matrix
# print classification report
print(classification_report(y_test_new, prediction_bnn_l))
```

```
              precision    recall  f1-score   support
```

```
              0       0.33      0.37      0.35       277
              1       0.23      0.25      0.24       213
              2       0.71      0.66      0.68       825

    accuracy                            0.53      1315
   macro avg       0.42      0.43      0.42      1315
weighted avg       0.55      0.53      0.54      1315
```

No siginificant change, so decided to go with the earlier version

7 Multinomial Naive Bayes

7.1 hyper parameter tuning

[ ]:
```
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import classification_report, confusion_matrix


#Hyper parameter tuning and fitting the best hyperparameters

parameters = {'alpha': np.arange(0.001,0.02,0.001)}
model_nb=MultinomialNB()
clf = GridSearchCV(model_nb, parameters,verbose=3,scoring='accuracy',cv=10)
clf.fit(train_vectors,train_labels_mnb)
print(clf.best_params_)
print(classification_report(test_labels_mnb, clf.predict(test_vectors)))
```

```
Fitting 10 folds for each of 19 candidates, totalling 190 fits
[CV 1/10] END …alpha=0.001;, score=0.857 total time=   0.0s
[CV 2/10] END …alpha=0.001;, score=0.851 total time=   0.0s
[CV 3/10] END …alpha=0.001;, score=0.847 total time=   0.0s
[CV 4/10] END …alpha=0.001;, score=0.853 total time=   0.0s
[CV 5/10] END …alpha=0.001;, score=0.852 total time=   0.0s
[CV 6/10] END …alpha=0.001;, score=0.858 total time=   0.0s
[CV 7/10] END …alpha=0.001;, score=0.846 total time=   0.0s
[CV 8/10] END …alpha=0.001;, score=0.850 total time=   0.0s
[CV 9/10] END …alpha=0.001;, score=0.843 total time=   0.0s
[CV 10/10] END …alpha=0.001;, score=0.864 total time=   0.0s
[CV 1/10] END …alpha=0.002;, score=0.857 total time=   0.0s
[CV 2/10] END …alpha=0.002;, score=0.852 total time=   0.0s
[CV 3/10] END …alpha=0.002;, score=0.847 total time=   0.0s
[CV 4/10] END …alpha=0.002;, score=0.853 total time=   0.0s
[CV 5/10] END …alpha=0.002;, score=0.854 total time=   0.0s
[CV 6/10] END …alpha=0.002;, score=0.858 total time=   0.0s
[CV 7/10] END …alpha=0.002;, score=0.846 total time=   0.0s
[CV 8/10] END …alpha=0.002;, score=0.852 total time=   0.0s
```

```
[CV 9/10] END …alpha=0.002;, score=0.843 total time=   0.0s
[CV 10/10] END …alpha=0.002;, score=0.864 total time=   0.0s
[CV 1/10] END …alpha=0.003;, score=0.858 total time=   0.0s
[CV 2/10] END …alpha=0.003;, score=0.853 total time=   0.0s
[CV 3/10] END …alpha=0.003;, score=0.847 total time=   0.0s
[CV 4/10] END …alpha=0.003;, score=0.852 total time=   0.0s
[CV 5/10] END …alpha=0.003;, score=0.854 total time=   0.0s
[CV 6/10] END …alpha=0.003;, score=0.857 total time=   0.0s
[CV 7/10] END …alpha=0.003;, score=0.846 total time=   0.0s
[CV 8/10] END …alpha=0.003;, score=0.852 total time=   0.0s
[CV 9/10] END …alpha=0.003;, score=0.843 total time=   0.0s
[CV 10/10] END …alpha=0.003;, score=0.863 total time=   0.0s
[CV 1/10] END …alpha=0.004;, score=0.857 total time=   0.0s
[CV 2/10] END …alpha=0.004;, score=0.853 total time=   0.0s
[CV 3/10] END …alpha=0.004;, score=0.847 total time=   0.0s
[CV 4/10] END …alpha=0.004;, score=0.852 total time=   0.0s
[CV 5/10] END …alpha=0.004;, score=0.853 total time=   0.0s
[CV 6/10] END …alpha=0.004;, score=0.858 total time=   0.0s
[CV 7/10] END …alpha=0.004;, score=0.845 total time=   0.0s
[CV 8/10] END …alpha=0.004;, score=0.852 total time=   0.0s
[CV 9/10] END …alpha=0.004;, score=0.843 total time=   0.0s
[CV 10/10] END …alpha=0.004;, score=0.864 total time=   0.0s
[CV 1/10] END …alpha=0.005;, score=0.857 total time=   0.0s
[CV 2/10] END …alpha=0.005;, score=0.853 total time=   0.0s
[CV 3/10] END …alpha=0.005;, score=0.847 total time=   0.0s
[CV 4/10] END …alpha=0.005;, score=0.852 total time=   0.0s
[CV 5/10] END …alpha=0.005;, score=0.853 total time=   0.0s
[CV 6/10] END …alpha=0.005;, score=0.858 total time=   0.0s
[CV 7/10] END …alpha=0.005;, score=0.846 total time=   0.0s
[CV 8/10] END …alpha=0.005;, score=0.852 total time=   0.0s
[CV 9/10] END …alpha=0.005;, score=0.843 total time=   0.0s
[CV 10/10] END …alpha=0.005;, score=0.864 total time=   0.0s
[CV 1/10] END …alpha=0.006;, score=0.857 total time=   0.0s
[CV 2/10] END …alpha=0.006;, score=0.853 total time=   0.0s
[CV 3/10] END …alpha=0.006;, score=0.847 total time=   0.0s
[CV 4/10] END …alpha=0.006;, score=0.852 total time=   0.0s
[CV 5/10] END …alpha=0.006;, score=0.854 total time=   0.0s
[CV 6/10] END …alpha=0.006;, score=0.858 total time=   0.0s
[CV 7/10] END …alpha=0.006;, score=0.845 total time=   0.0s
[CV 8/10] END …alpha=0.006;, score=0.852 total time=   0.0s
[CV 9/10] END …alpha=0.006;, score=0.843 total time=   0.0s
[CV 10/10] END …alpha=0.006;, score=0.864 total time=   0.0s
[CV 1/10] END …alpha=0.007;, score=0.857 total time=   0.0s
[CV 2/10] END …alpha=0.007;, score=0.853 total time=   0.0s
[CV 3/10] END …alpha=0.007;, score=0.846 total time=   0.0s
[CV 4/10] END …alpha=0.007;, score=0.852 total time=   0.0s
[CV 5/10] END …alpha=0.007;, score=0.853 total time=   0.0s
[CV 6/10] END …alpha=0.007;, score=0.857 total time=   0.0s
```

```
[CV 7/10] END …alpha=0.007;, score=0.845 total time=   0.0s
[CV 8/10] END …alpha=0.007;, score=0.852 total time=   0.0s
[CV 9/10] END …alpha=0.007;, score=0.842 total time=   0.0s
[CV 10/10] END …alpha=0.007;, score=0.864 total time=   0.0s
[CV 1/10] END …alpha=0.008;, score=0.857 total time=   0.0s
[CV 2/10] END …alpha=0.008;, score=0.853 total time=   0.0s
[CV 3/10] END …alpha=0.008;, score=0.847 total time=   0.0s
[CV 4/10] END …alpha=0.008;, score=0.852 total time=   0.0s
[CV 5/10] END …alpha=0.008;, score=0.853 total time=   0.0s
[CV 6/10] END …alpha=0.008;, score=0.857 total time=   0.0s
[CV 7/10] END …alpha=0.008;, score=0.844 total time=   0.0s
[CV 8/10] END …alpha=0.008;, score=0.853 total time=   0.0s
[CV 9/10] END …alpha=0.008;, score=0.841 total time=   0.0s
[CV 10/10] END …alpha=0.008;, score=0.864 total time=   0.0s
[CV 1/10] END …alpha=0.009000000000000001;, score=0.857 total time=   0.0s
[CV 2/10] END …alpha=0.009000000000000001;, score=0.853 total time=   0.0s
[CV 3/10] END …alpha=0.009000000000000001;, score=0.847 total time=   0.0s
[CV 4/10] END …alpha=0.009000000000000001;, score=0.853 total time=   0.0s
[CV 5/10] END …alpha=0.009000000000000001;, score=0.854 total time=   0.0s
[CV 6/10] END …alpha=0.009000000000000001;, score=0.857 total time=   0.0s
[CV 7/10] END …alpha=0.009000000000000001;, score=0.843 total time=   0.0s
[CV 8/10] END …alpha=0.009000000000000001;, score=0.852 total time=   0.0s
[CV 9/10] END …alpha=0.009000000000000001;, score=0.841 total time=   0.0s
[CV 10/10] END …alpha=0.009000000000000001;, score=0.864 total time=   0.0s
[CV 1/10] END …alpha=0.010000000000000002;, score=0.857 total time=   0.0s
[CV 2/10] END …alpha=0.010000000000000002;, score=0.853 total time=   0.0s
[CV 3/10] END …alpha=0.010000000000000002;, score=0.847 total time=   0.0s
[CV 4/10] END …alpha=0.010000000000000002;, score=0.852 total time=   0.0s
[CV 5/10] END …alpha=0.010000000000000002;, score=0.854 total time=   0.0s
[CV 6/10] END …alpha=0.010000000000000002;, score=0.857 total time=   0.0s
[CV 7/10] END …alpha=0.010000000000000002;, score=0.843 total time=   0.0s
[CV 8/10] END …alpha=0.010000000000000002;, score=0.852 total time=   0.0s
[CV 9/10] END …alpha=0.010000000000000002;, score=0.841 total time=   0.0s
[CV 10/10] END …alpha=0.010000000000000002;, score=0.864 total time=   0.0s
[CV 1/10] END …alpha=0.011;, score=0.856 total time=   0.0s
[CV 2/10] END …alpha=0.011;, score=0.854 total time=   0.0s
[CV 3/10] END …alpha=0.011;, score=0.847 total time=   0.0s
[CV 4/10] END …alpha=0.011;, score=0.853 total time=   0.0s
[CV 5/10] END …alpha=0.011;, score=0.854 total time=   0.0s
[CV 6/10] END …alpha=0.011;, score=0.856 total time=   0.0s
[CV 7/10] END …alpha=0.011;, score=0.843 total time=   0.0s
[CV 8/10] END …alpha=0.011;, score=0.852 total time=   0.0s
[CV 9/10] END …alpha=0.011;, score=0.842 total time=   0.0s
[CV 10/10] END …alpha=0.011;, score=0.864 total time=   0.0s
[CV 1/10] END …alpha=0.012;, score=0.857 total time=   0.0s
[CV 2/10] END …alpha=0.012;, score=0.854 total time=   0.0s
[CV 3/10] END …alpha=0.012;, score=0.848 total time=   0.0s
[CV 4/10] END …alpha=0.012;, score=0.853 total time=   0.0s
```

```
[CV 5/10] END …alpha=0.012;, score=0.854 total time=   0.0s
[CV 6/10] END …alpha=0.012;, score=0.856 total time=   0.0s
[CV 7/10] END …alpha=0.012;, score=0.843 total time=   0.0s
[CV 8/10] END …alpha=0.012;, score=0.852 total time=   0.0s
[CV 9/10] END …alpha=0.012;, score=0.842 total time=   0.0s
[CV 10/10] END …alpha=0.012;, score=0.864 total time=   0.0s
[CV 1/10] END …alpha=0.013000000000000001;, score=0.857 total time=   0.0s
[CV 2/10] END …alpha=0.013000000000000001;, score=0.853 total time=   0.0s
[CV 3/10] END …alpha=0.013000000000000001;, score=0.848 total time=   0.0s
[CV 4/10] END …alpha=0.013000000000000001;, score=0.853 total time=   0.0s
[CV 5/10] END …alpha=0.013000000000000001;, score=0.854 total time=   0.0s
[CV 6/10] END …alpha=0.013000000000000001;, score=0.856 total time=   0.0s
[CV 7/10] END …alpha=0.013000000000000001;, score=0.843 total time=   0.0s
[CV 8/10] END …alpha=0.013000000000000001;, score=0.852 total time=   0.0s
[CV 9/10] END …alpha=0.013000000000000001;, score=0.842 total time=   0.0s
[CV 10/10] END …alpha=0.013000000000000001;, score=0.864 total time=   0.0s
[CV 1/10] END …alpha=0.014000000000000002;, score=0.856 total time=   0.0s
[CV 2/10] END …alpha=0.014000000000000002;, score=0.852 total time=   0.0s
[CV 3/10] END …alpha=0.014000000000000002;, score=0.848 total time=   0.0s
[CV 4/10] END …alpha=0.014000000000000002;, score=0.853 total time=   0.0s
[CV 5/10] END …alpha=0.014000000000000002;, score=0.854 total time=   0.0s
[CV 6/10] END …alpha=0.014000000000000002;, score=0.856 total time=   0.0s
[CV 7/10] END …alpha=0.014000000000000002;, score=0.843 total time=   0.0s
[CV 8/10] END …alpha=0.014000000000000002;, score=0.851 total time=   0.0s
[CV 9/10] END …alpha=0.014000000000000002;, score=0.842 total time=   0.0s
[CV 10/10] END …alpha=0.014000000000000002;, score=0.864 total time=   0.0s
[CV 1/10] END …alpha=0.015;, score=0.856 total time=   0.0s
[CV 2/10] END …alpha=0.015;, score=0.852 total time=   0.0s
[CV 3/10] END …alpha=0.015;, score=0.848 total time=   0.0s
[CV 4/10] END …alpha=0.015;, score=0.853 total time=   0.0s
[CV 5/10] END …alpha=0.015;, score=0.854 total time=   0.0s
[CV 6/10] END …alpha=0.015;, score=0.856 total time=   0.0s
[CV 7/10] END …alpha=0.015;, score=0.843 total time=   0.0s
[CV 8/10] END …alpha=0.015;, score=0.851 total time=   0.0s
[CV 9/10] END …alpha=0.015;, score=0.842 total time=   0.0s
[CV 10/10] END …alpha=0.015;, score=0.863 total time=   0.0s
[CV 1/10] END …alpha=0.016;, score=0.856 total time=   0.0s
[CV 2/10] END …alpha=0.016;, score=0.852 total time=   0.0s
[CV 3/10] END …alpha=0.016;, score=0.848 total time=   0.0s
[CV 4/10] END …alpha=0.016;, score=0.853 total time=   0.0s
[CV 5/10] END …alpha=0.016;, score=0.854 total time=   0.0s
[CV 6/10] END …alpha=0.016;, score=0.856 total time=   0.0s
[CV 7/10] END …alpha=0.016;, score=0.843 total time=   0.0s
[CV 8/10] END …alpha=0.016;, score=0.851 total time=   0.0s
[CV 9/10] END …alpha=0.016;, score=0.842 total time=   0.0s
[CV 10/10] END …alpha=0.016;, score=0.863 total time=   0.0s
[CV 1/10] END …alpha=0.017;, score=0.856 total time=   0.0s
[CV 2/10] END …alpha=0.017;, score=0.852 total time=   0.0s
```

```
[CV 3/10] END …alpha=0.017;, score=0.848 total time=   0.0s
[CV 4/10] END …alpha=0.017;, score=0.853 total time=   0.0s
[CV 5/10] END …alpha=0.017;, score=0.854 total time=   0.0s
[CV 6/10] END …alpha=0.017;, score=0.855 total time=   0.0s
[CV 7/10] END …alpha=0.017;, score=0.843 total time=   0.0s
[CV 8/10] END …alpha=0.017;, score=0.851 total time=   0.0s
[CV 9/10] END …alpha=0.017;, score=0.843 total time=   0.0s
[CV 10/10] END …alpha=0.017;, score=0.863 total time=   0.0s
[CV 1/10] END …alpha=0.018000000000000002;, score=0.856 total time=   0.0s
[CV 2/10] END …alpha=0.018000000000000002;, score=0.852 total time=   0.0s
[CV 3/10] END …alpha=0.018000000000000002;, score=0.848 total time=   0.0s
[CV 4/10] END …alpha=0.018000000000000002;, score=0.854 total time=   0.0s
[CV 5/10] END …alpha=0.018000000000000002;, score=0.854 total time=   0.0s
[CV 6/10] END …alpha=0.018000000000000002;, score=0.856 total time=   0.0s
[CV 7/10] END …alpha=0.018000000000000002;, score=0.843 total time=   0.0s
[CV 8/10] END …alpha=0.018000000000000002;, score=0.851 total time=   0.0s
[CV 9/10] END …alpha=0.018000000000000002;, score=0.843 total time=   0.0s
[CV 10/10] END …alpha=0.018000000000000002;, score=0.863 total time=   0.0s
[CV 1/10] END …alpha=0.019000000000000003;, score=0.856 total time=   0.0s
[CV 2/10] END …alpha=0.019000000000000003;, score=0.852 total time=   0.0s
[CV 3/10] END …alpha=0.019000000000000003;, score=0.848 total time=   0.0s
[CV 4/10] END …alpha=0.019000000000000003;, score=0.854 total time=   0.0s
[CV 5/10] END …alpha=0.019000000000000003;, score=0.854 total time=   0.0s
[CV 6/10] END …alpha=0.019000000000000003;, score=0.856 total time=   0.0s
[CV 7/10] END …alpha=0.019000000000000003;, score=0.843 total time=   0.0s
[CV 8/10] END …alpha=0.019000000000000003;, score=0.851 total time=   0.0s
[CV 9/10] END …alpha=0.019000000000000003;, score=0.843 total time=   0.0s
[CV 10/10] END …alpha=0.019000000000000003;, score=0.864 total time=   0.0s
{'alpha': 0.006}
              precision    recall  f1-score   support

           0       0.52      0.48      0.50       277
           1       0.54      0.60      0.57       213
           2       0.83      0.83      0.83       825

    accuracy                           0.72      1315
   macro avg       0.63      0.64      0.63      1315
weighted avg       0.72      0.72      0.72      1315
```

[ ]:
```
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.model_selection import cross_validate
nb_model=MultinomialNB(alpha=0.006)
cv_results = cross_validate(nb_model,train_vectors,train_labels_mnb,␣
 ↪cv=5,verbose=1,return_estimator=True)
accuracies_nb = cv_results['test_score']
```

```
accuracies_nb=list(accuracies_nb)
nb_models=cv_results['estimator']
nb_model = nb_models[accuracies_nb.index(max(accuracies_nb))]
predictions_nb = nb_model.predict(test_vectors)

# print classification report
print(classification_report(test_labels_mnb, predictions_nb))
```

```
              precision    recall  f1-score   support

           0       0.53      0.51      0.52       277
           1       0.53      0.60      0.56       213
           2       0.83      0.82      0.83       825

    accuracy                           0.72      1315
   macro avg       0.63      0.64      0.64      1315
weighted avg       0.72      0.72      0.72      1315
```

```
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done   5 out of   5 | elapsed:    0.1s finished
```

comments: Trained using different vectorizer method, tf-idf, bag of words,

8 Random Forest

8.1 Hyper parameter tuning and training the best Model

[ ]:
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import classification_report, confusion_matrix


#Hyper parameter tuning and fitting the best hyperparameters
#
parameters = {'criterion': ('gini','entropy'),'max_depth':np.arange(30,35)}
model_rf=RandomForestClassifier()
clf = GridSearchCV(model_rf, parameters,verbose=3,scoring='accuracy',cv=3)
clf.fit(x_train_new,y_train_new)
print(clf.best_params_)
print(classification_report(y_test_new, clf.predict(x_test_new)))
```

```
Fitting 3 folds for each of 10 candidates, totalling 30 fits
[CV 1/3] END …criterion=gini, max_depth=30;, score=0.788 total time=   4.4s
[CV 2/3] END …criterion=gini, max_depth=30;, score=0.802 total time=   3.2s
[CV 3/3] END …criterion=gini, max_depth=30;, score=0.796 total time=   3.1s
[CV 1/3] END …criterion=gini, max_depth=31;, score=0.791 total time=   3.1s
```

```
[CV 2/3] END …criterion=gini, max_depth=31;, score=0.797 total time=   3.1s
[CV 3/3] END …criterion=gini, max_depth=31;, score=0.795 total time=   3.1s
[CV 1/3] END …criterion=gini, max_depth=32;, score=0.784 total time=   3.1s
[CV 2/3] END …criterion=gini, max_depth=32;, score=0.802 total time=   3.1s
[CV 3/3] END …criterion=gini, max_depth=32;, score=0.793 total time=   3.1s
[CV 1/3] END …criterion=gini, max_depth=33;, score=0.787 total time=   3.1s
[CV 2/3] END …criterion=gini, max_depth=33;, score=0.803 total time=   3.1s
[CV 3/3] END …criterion=gini, max_depth=33;, score=0.798 total time=   3.1s
[CV 1/3] END …criterion=gini, max_depth=34;, score=0.789 total time=   3.1s
[CV 2/3] END …criterion=gini, max_depth=34;, score=0.803 total time=   3.1s
[CV 3/3] END …criterion=gini, max_depth=34;, score=0.794 total time=   3.2s
[CV 1/3] END …criterion=entropy, max_depth=30;, score=0.786 total time=   3.9s
[CV 2/3] END …criterion=entropy, max_depth=30;, score=0.804 total time=   3.8s
[CV 3/3] END …criterion=entropy, max_depth=30;, score=0.792 total time=   3.8s
[CV 1/3] END …criterion=entropy, max_depth=31;, score=0.786 total time=   3.8s
[CV 2/3] END …criterion=entropy, max_depth=31;, score=0.800 total time=   3.8s
[CV 3/3] END …criterion=entropy, max_depth=31;, score=0.796 total time=   3.8s
[CV 1/3] END …criterion=entropy, max_depth=32;, score=0.786 total time=   3.8s
[CV 2/3] END …criterion=entropy, max_depth=32;, score=0.806 total time=   3.8s
[CV 3/3] END …criterion=entropy, max_depth=32;, score=0.793 total time=   3.8s
[CV 1/3] END …criterion=entropy, max_depth=33;, score=0.784 total time=   3.8s
[CV 2/3] END …criterion=entropy, max_depth=33;, score=0.803 total time=   3.8s
[CV 3/3] END …criterion=entropy, max_depth=33;, score=0.795 total time=   3.8s
[CV 1/3] END …criterion=entropy, max_depth=34;, score=0.790 total time=   3.8s
[CV 2/3] END …criterion=entropy, max_depth=34;, score=0.802 total time=   3.8s
[CV 3/3] END …criterion=entropy, max_depth=34;, score=0.795 total time=   3.7s
{'criterion': 'entropy', 'max_depth': 34}
              precision    recall  f1-score   support

           0       0.36      0.28      0.32       277
           1       0.26      0.23      0.25       213
           2       0.73      0.80      0.76       825

    accuracy                           0.60      1315
   macro avg       0.45      0.44      0.44      1315
weighted avg       0.58      0.60      0.59      1315
```

8.2 Results

[ ]:
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.model_selection import cross_validate
rf_model=RandomForestClassifier(criterion='entropy',max_depth=34)
rf_model.fit(x_train_new,y_train_new)
# print classification report
print(classification_report(y_test_new, rf_model.predict(x_test_new)))
```

```
              precision    recall  f1-score   support

           0       0.38      0.28      0.32       277
           1       0.27      0.24      0.26       213
           2       0.72      0.80      0.76       825

    accuracy                           0.60      1315
   macro avg       0.46      0.44      0.45      1315
weighted avg       0.58      0.60      0.59      1315
```

9 K-means (unsupervised learning)

[ ]:
```
#kmeans classifier unsupervised
from sklearn.cluster import KMeans
from sklearn.metrics import classification_report, confusion_matrix
# Apply the k-means algorithm
k_model = KMeans(n_clusters=3)
k_model.fit(train_vectors_tfidf)
print(classification_report(test_labels_mnb, k_model.
 ↪predict(test_vectors_tfidf)))
```

```
              precision    recall  f1-score   support

           0       0.10      0.03      0.05       277
           1       0.93      0.07      0.12       213
           2       0.66      0.97      0.78       825

    accuracy                           0.62      1315
   macro avg       0.56      0.35      0.32      1315
weighted avg       0.59      0.62      0.52      1315
```

10 SVC

[ ]:
```
from sklearn.svm import SVC
from sklearn.metrics import classification_report, confusion_matrix
# Apply the k-means algorithm
svc_model = SVC()
svc_model.fit(train_vectors_tfidf[:9000],train_labels_tfidf[:9000])
print(classification_report(test_labels_mnb, svc_model.
 ↪predict(test_vectors_tfidf)))
```

```
              precision    recall  f1-score   support

           0       0.50      0.58      0.54       277
```

```
           1       0.67      0.67      0.67       213
           2       0.86      0.82      0.84       825

    accuracy                           0.74      1315
   macro avg       0.68      0.69      0.68      1315
weighted avg       0.75      0.74      0.75      1315
```

#Ensambled method using svc and naive bayes

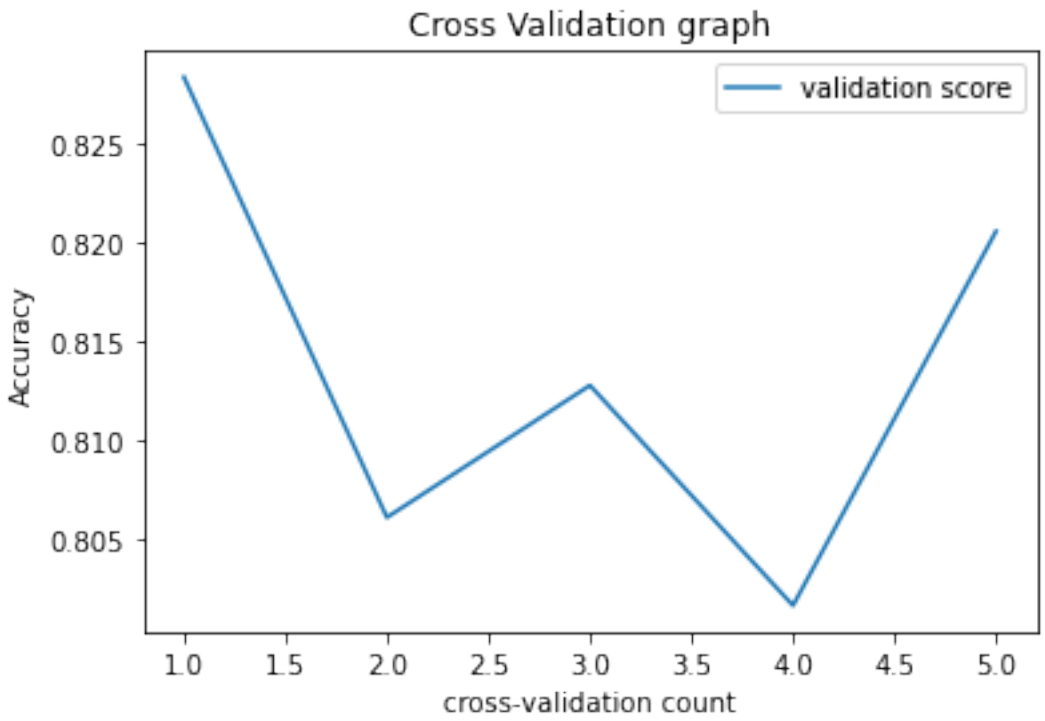
[ ]:
```
from sklearn.ensemble import StackingClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.model_selection import cross_validate

svc= svc_model
nb=nb_model
models = [('nb',nb),('svc',svc)]

stacking_model = LogisticRegression(solver='sag',random_state=0,max_iter=200)
stacked_model = StackingClassifier( estimators = models,final_estimator =␣
 ↪stacking_model)
cv_results_s = cross_validate(stacked_model, train_vectors_tfidf[:
 ↪9000],train_labels_tfidf[:9000], cv=5,verbose=1,return_estimator=True)
```

```
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done   5 out of   5 | elapsed:  3.1min finished
```

Cross-validation graph

[ ]:
```
import matplotlib.pyplot as plt

d={'t':[1,2,3,4,5],'validation score':cv_results_s['test_score'][:5]}
pd.DataFrame(d).plot.line('t','validation score')
plt.title("Cross Validation graph")
plt.xlabel("cross-validation count")
plt.ylabel("Accuracy")
```

[ ]:
```
Text(0, 0.5, 'Accuracy')
```



[ ]:
```
cv_results_s['test_score'][:5]
```

[ ]:
```
array([0.82833333, 0.80611111, 0.81277778, 0.80166667, 0.82055556])
```

[ ]:
```
pre_st=cv_results_s['estimator'][3].predict(test_vectors_tfidf)
```

[ ]:
```
print(classification_report(test_labels_mnb, pre_st))
```

```
              precision    recall  f1-score   support

           0       0.53      0.54      0.54       277
           1       0.64      0.70      0.67       213
           2       0.86      0.83      0.84       825

    accuracy                           0.75      1315
   macro avg       0.67      0.69      0.68      1315
weighted avg       0.75      0.75      0.75      1315
```

#(Additional work done)

#checking using lazy classifier- for getting best classifier using K-mean algorithm

[ ]:
```
!pip install lazypredict
from lazypredict.Supervised import LazyClassifier
clf=LazyClassifier(verbose=0,ignore_warnings=True,custom_metric=None)
models,prediction=clf.fit(x_train_new[:1000],x_test_new,y_train_new[:
  ↪1000],y_test_new)
print(models)
```

```
100%|      | 29/29 [00:10<00:00,  2.81it/s]
```

Model	Accuracy	Balanced Accuracy	ROC AUC	F1 Score
XGBClassifier	0.47	0.46	None	0.50
RandomForestClassifier	0.52	0.46	None	0.54
BernoulliNB	0.40	0.43	None	0.43
ExtraTreesClassifier	0.47	0.43	None	0.50
BaggingClassifier	0.43	0.42	None	0.46
AdaBoostClassifier	0.45	0.42	None	0.48
PassiveAggressiveClassifier	0.37	0.41	None	0.39
Perceptron	0.34	0.40	None	0.36
NuSVC	0.36	0.40	None	0.37
KNeighborsClassifier	0.36	0.40	None	0.38
LGBMClassifier	0.43	0.39	None	0.46
LinearDiscriminantAnalysis	0.28	0.39	None	0.27
SGDClassifier	0.34	0.38	None	0.36
ExtraTreeClassifier	0.39	0.38	None	0.43
RidgeClassifierCV	0.29	0.38	None	0.29
RidgeClassifier	0.29	0.38	None	0.29
LogisticRegression	0.27	0.38	None	0.27
DecisionTreeClassifier	0.39	0.37	None	0.43
LinearSVC	0.28	0.37	None	0.28
CalibratedClassifierCV	0.26	0.37	None	0.26
SVC	0.20	0.37	None	0.13
NearestCentroid	0.25	0.36	None	0.23
LabelSpreading	0.32	0.35	None	0.35
LabelPropagation	0.32	0.35	None	0.35
QuadraticDiscriminantAnalysis	0.20	0.35	None	0.13
GaussianNB	0.17	0.33	None	0.07
DummyClassifier	0.21	0.33	None	0.07

Model	Time Taken
XGBClassifier	1.84
RandomForestClassifier	0.82
BernoulliNB	0.06
ExtraTreesClassifier	0.63
BaggingClassifier	0.11
AdaBoostClassifier	0.23
PassiveAggressiveClassifier	0.06

```
Perceptron                          0.07
NuSVC                               0.66
KNeighborsClassifier                0.18
LGBMClassifier                      1.15
LinearDiscriminantAnalysis          0.17
SGDClassifier                       0.21
ExtraTreeClassifier                 0.03
RidgeClassifierCV                   0.14
RidgeClassifier                     0.06
LogisticRegression                  0.21
DecisionTreeClassifier              0.05
LinearSVC                           0.46
CalibratedClassifierCV              1.69
SVC                                 0.47
NearestCentroid                     0.05
LabelSpreading                      0.35
LabelPropagation                    0.19
QuadraticDiscriminantAnalysis       0.12
GaussianNB                          0.03
DummyClassifier                     0.04
```

11 BERT-Transformer based model

[ ]:
```
!pip install torch==1.8.1+cu111 torchvision==0.9.1+cu111 torchaudio===0.8.1 -f␣
↪https://download.pytorch.org/whl/torch_stable.html

!pip install transformers requests beautifulsoup4 pandas numpy
from transformers import AutoTokenizer, AutoModelForSequenceClassification
import torch
import requests
from bs4 import BeautifulSoup
import re
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Looking in links: https://download.pytorch.org/whl/torch_stable.html
Requirement already satisfied: torch==1.8.1+cu111 in
/usr/local/lib/python3.8/dist-packages (1.8.1+cu111)
Requirement already satisfied: torchvision==0.9.1+cu111 in
/usr/local/lib/python3.8/dist-packages (0.9.1+cu111)
Requirement already satisfied: torchaudio===0.8.1 in
/usr/local/lib/python3.8/dist-packages (0.8.1)
Requirement already satisfied: typing-extensions in
/usr/local/lib/python3.8/dist-packages (from torch==1.8.1+cu111) (4.4.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.8/dist-packages
```

```
(from torch==1.8.1+cu111) (1.21.6)
Requirement already satisfied: pillow>=4.1.1 in /usr/local/lib/python3.8/dist-
packages (from torchvision==0.9.1+cu111) (7.1.2)
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Requirement already satisfied: transformers in /usr/local/lib/python3.8/dist-
packages (4.25.1)
Requirement already satisfied: requests in /usr/local/lib/python3.8/dist-
packages (2.25.1)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.8/dist-
packages (4.6.3)
Requirement already satisfied: pandas in /usr/local/lib/python3.8/dist-packages
(1.3.5)
Requirement already satisfied: numpy in /usr/local/lib/python3.8/dist-packages
(1.21.6)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.8/dist-
packages (from transformers) (4.64.1)
Requirement already satisfied: huggingface-hub<1.0,>=0.10.0 in
/usr/local/lib/python3.8/dist-packages (from transformers) (0.11.1)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.8/dist-
packages (from transformers) (6.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.8/dist-
packages (from transformers) (21.3)
Requirement already satisfied: filelock in /usr/local/lib/python3.8/dist-
packages (from transformers) (3.9.0)
Requirement already satisfied: tokenizers!=0.11.3,<0.14,>=0.11.1 in
/usr/local/lib/python3.8/dist-packages (from transformers) (0.13.2)
Requirement already satisfied: regex!=2019.12.17 in
/usr/local/lib/python3.8/dist-packages (from transformers) (2022.6.2)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.8/dist-
packages (from requests) (2.10)
Requirement already satisfied: chardet<5,>=3.0.2 in
/usr/local/lib/python3.8/dist-packages (from requests) (4.0.0)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/usr/local/lib/python3.8/dist-packages (from requests) (1.24.3)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.8/dist-packages (from requests) (2022.12.7)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.8/dist-
packages (from pandas) (2022.7)
Requirement already satisfied: python-dateutil>=2.7.3 in
/usr/local/lib/python3.8/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.8/dist-packages (from huggingface-
hub<1.0,>=0.10.0->transformers) (4.4.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/usr/local/lib/python3.8/dist-packages (from packaging>=20.0->transformers)
(3.0.9)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.8/dist-
```

```
packages (from python-dateutil>=2.7.3->pandas) (1.15.0)
```

[ ]:
```
tokenizer = AutoTokenizer.from_pretrained('nlptown/
 ↪bert-base-multilingual-uncased-sentiment')

model = AutoModelForSequenceClassification.from_pretrained('nlptown/
 ↪bert-base-multilingual-uncased-sentiment')
```

```
Downloading:   0%|          | 0.00/39.0 [00:00<?, ?B/s]
Downloading:   0%|          | 0.00/953 [00:00<?, ?B/s]
Downloading:   0%|          | 0.00/872k [00:00<?, ?B/s]
Downloading:   0%|          | 0.00/112 [00:00<?, ?B/s]
Downloading:   0%|          | 0.00/669M [00:00<?, ?B/s]
```

[ ]:
```
data[data["airline_sentiment"]=="neutral"].head(10)
```

[ ]:
```
              tweet_id                                                text  \
5   569617089155211265  @AmericanAir just downloaded the app for iPhon…
7   570287303681294337  @JetBlue I'm not sure if you can do anything t…
17  568586231409475584  @SouthwestAir seems like you could make more m…
24  568095662262358016  @united this is me and my partners first trip …
27  568492682152189952  @VirginAmerica partners with @Visa Checkout as…
34  568077686507229184  @USAirways just realized my @AmericanAir advan…
47  567802111321444352  @united do you think there Will problems at Ne…
48  568812654346653697  @JetBlue OK cool.  I need to listen to some Dr…
49  567935527481188352   @JetBlue Anywhere warm cause its freezing in NYC
51  568804534845345792  @USAirways can I book using some sort of breav…

   airline_sentiment
5            neutral
7            neutral
17           neutral
24           neutral
27           neutral
34           neutral
47           neutral
48           neutral
49           neutral
51           neutral
```

[ ]:
```
tokens = tokenizer.encode("@JetBlue OK cool. I need to listen to some Dre and␣
 ↪Snoop en route to LA. That would have been a shame.", return_tensors='pt')
result = model(tokens)
result.logits
int(torch.argmax(result.logits))+1
```

[ ]: 3

[ ]:
```
def sentiment_score(review):
    tokens = tokenizer.encode(review, return_tensors='pt')
    result = model(tokens)
    return int(torch.argmax(result.logits))+1
```

[ ]:
```
prediction=x_test[:100].apply(lambda x: sentiment_score(x))
```

[ ]:
```
predictions= prediction.map({5: 1, 1: 2, 2 : 0,3:0,4:0})
predictions=np.array(predictions)
```

[ ]:
```
from sklearn.metrics import precision_score,␣
 ↪recall_score,f1_score,accuracy_score
accuracy = accuracy_score(y_test_new[:100], predictions)
print('Accuracy:', accuracy)
print("F1 zscore(Macro) =  ",f1_score(y_test_new[:
 ↪100],predictions,average="macro"))
print("F1 zscore =  ",f1_score(y_test_new[:100],predictions,average=None))
```

```
Accuracy: 0.7
F1 zscore(Macro) =   0.5242938856891863
F1 zscore =   [0.13793103 0.58823529 0.84671533]
```