In [22]:

**import** pandas **as** pd

**import** numpy **as** np

**import** seaborn **as** sns

**import** matplotlib.pyplot **as** plt

*# nltk*

**import** nltk

**import** nltk

nltk**.**download('stopwords')

**from** nltk.corpus **import** stopwords

**from** nltk.stem **import** SnowballStemmer

**import** re

**import** string

[nltk\_data] Downloading package stopwords to

[nltk\_data] C:\Users\pratiksha\AppData\Roaming\nltk\_data... [nltk\_data] Unzipping corpora\stopwords.zip.

In [13]:

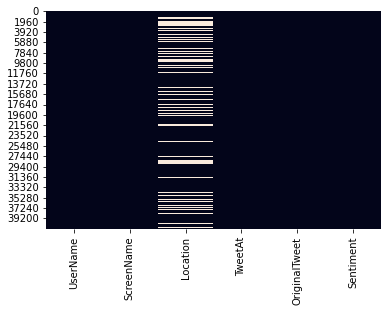
train**=**pd**.**read\_csv('Corona\_NLP\_train.csv',encoding**=**'latin-1') test**=**pd**.**read\_csv('Corona\_NLP\_test.csv',encoding**=**'latin-1')

In [14]:

Out[14]:

<AxesSubplot:>

sns**.**heatmap(train**.**isnull(),cbar**=False**)



In [15]:

train**.**head()

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Out[15]: | **UserName** | **ScreenName** | **Location** | **TweetAt** | **OriginalTweet** | **Sentiment** |
| **0** | 3799 | 48751 | London | 16-03-2020 | @MeNyrbie @Phil\_Gahan @Chrisitv https://t.co/i... | Neutral |
| **1** | 3800 | 48752 | UK | 16-03-2020 | advice Talk to your neighbours family to excha... | Positive |
| **2** | 3801 | 48753 | Vagabonds | 16-03-2020 | Coronavirus Australia: Woolworths to give elde... | Positive |
| **3** | 3802 | 48754 | NaN | 16-03-2020 | My food stock is not the only one which is emp... | Positive |
| **4** | 3803 | 48755 | NaN | 16-03-2020 | Me, ready to go at supermarket during the #COV... | Extremely Negative |

In [16]:

Out[16]:

<AxesSubplot:>

sns**.**heatmap(test**.**isnull(),cbar**=False**)



In [17]:

test**.**groupby('Sentiment')**.**count()

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Out[17]: | **Sentiment** | **UserName** | **ScreenName** | **Location** | **TweetAt** | **OriginalTweet** |
|  | **Extremely Negative** | 592 | 592 | 449 | 592 | 592 |
|  | **Extremely Positive** | 599 | 599 | 482 | 599 | 599 |
|  | **Negative** | 1041 | 1041 | 811 | 1041 | 1041 |
|  | **Neutral** | 619 | 619 | 497 | 619 | 619 |
|  | **Positive** | 947 | 947 | 725 | 947 | 947 |

In [18]:

*# char\_count*

train['char\_count'] **=** train['OriginalTweet']**.**apply(**lambda** x: len(str(x)))

*# mention\_count*

train['mention\_count'] **=** train['OriginalTweet']**.**apply(**lambda** x: len([c **for** c **in** str(x) **if** c **==** '@']))

In [19]:

*# char\_count*

test['char\_count'] **=** test['OriginalTweet']**.**apply(**lambda** x: len(str(x)))

*# mention\_count*

test['mention\_count'] **=** test['OriginalTweet']**.**apply(**lambda** x: len([c **for** c **in** str(x) **if** c **==** '@'])) test

Out[19]:

**UserName ScreenName Location TweetAt OriginalTweet Sentiment char\_count mention\_count**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **0** | 1 | 44953 | NYC 02-03- TRENDING: New Yorkers encounter empty Extremely 228 0  2020 supermar... Negative |
| **1** | 2 | 44954 | Seattle, WA 02-03- When I couldn't find hand sanitizer at Fred Positive 193 0  2020 Me... |
| **2** | 3 | 44955 | NaN 02-03- Find out how you can protect yourself and Extremely 73 0  2020 love... Positive |
|  | **3** | 4 | 44956 | Chicagoland 02-03- #Panic buying hits #NewYork City as anxious Negative 318 0  2020 sh... |

**4** 5 44957 Melbourne, Victoria

03-03-

2020

#toiletpaper #dunnypaper #coronavirus

#coronav...

Neutral 252 0

**...** ... ... ... ... ... ... ... ...

**3793** 3794 48746 Israel ?? 16-03-

2020

Meanwhile In A Supermarket in Israel --

People...

Positive 127 0

**3794**

3795

48747 Farmington, NM

16-03-

2020

Did you panic buy a lot of non-perishable

item...

Negative

213

0

**3795** 3796 48748 Haverford, PA 16-03-

2020

Asst Prof of Economics @cconces was on

@NBCPhi...

Neutral

**3796**

3797

48749

NaN

16-03- Gov need to do somethings instead of biar je 2020 r...

Extremely Negative

**3797** 3798 48750 Arlington,

Virginia

16-03-

2020

I and @ForestandPaper members are

committed to...

Extremely Positive

# 3798 rows × 8 columns

|  |  |
| --- | --- |
| 185 | 2 |
| 174 | 0 |
| 254 | 1 |

In [20]:

new **=** train["TweetAt"]**.**str**.**split("-", n **=** 2, expand **= True**) train['year']**=**new[2]**.**astype(int)

train['month']**=**new[1]**.**astype(int) train['day']**=**new[0]**.**astype(int)

train

Out[20]:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UserName ScreenName Location TweetAt OriginalTweet Sentiment** | **char\_count** | **mention\_count** | **year** | **month** | **day** |
| **0** 3799 48751 London 16-03- @MeNyrbie @Phil\_Gahan Neutral | 111 | 3 | 2020 | 3 | 16 |
| **1** 3800 48752 UK 16-03- advice Talk to your Positive | 237 | 0 | 2020 | 3 | 16 |
| **2** 3801 48753 Vagabonds 16-03- Coronavirus Australia: Positive | 131 | 0 | 2020 | 3 | 16 |
| **3** 3802 48754 NaN 16-03- My food stock is not the only Positive | 306 | 0 | 2020 | 3 | 16 |

2020 @Chrisitv https://t.co/i...

2020 neighbours family to excha...

2020 Woolworths to give elde...

**4** 3803 48755 NaN

2020

16-03-

2020

one which is emp...

Me, ready to go at supermarket during the

#COV...

Extremely 310 0 2020 3 16 Negative

**...**

...

...

...

...

...

...

...

...

...

... ...

**41152** 44951 89903 Wellington City,

New Zealand

14-04-

2020

Airline pilots offering to stock

supermarket s...

Neutral 102 0 2020 4 14

**41153**

44952

89904

NaN 14-04-

2020

Response to complaint not provided citing COVI...

Extremely Negative

138

0 2020

4 14

**41154** 44953 89905 NaN 14-04-

2020

You know itÂ s getting tough when @KameronWild...

Positive 136 2 2020 4 14

**41155**

44954

89906

NaN 14-04-

2020

Is it wrong that the smell of

hand sanitizer i...

Neutral

111

0 2020

4 14

**41156** 44955 89907 i love you so

much || he/him

14-04-

2020

@TartiiCat Well new/used Rift

S are going for ...

Negative 255 1 2020 4 14

# 41157 rows × 11 columns

In [21]:

new **=** test["TweetAt"]**.**str**.**split("-", n **=** 2, expand **= True**) test['year']**=**new[2]**.**astype(int)

test['month']**=**new[1]**.**astype(int) test['day']**=**new[0]**.**astype(int)

test

Out[21]:

**UserName ScreenName Location TweetAt OriginalTweet Sentiment char\_count mention\_count year month day**

**0** 1 44953 NYC 02-03-

2020

TRENDING: New Yorkers encounter empty supermar...

Extremely Negative

228 0 2020 3 2

**1**

2

44954 Seattle, WA 02-03-

2020

When I couldn't find hand sanitizer at Fred Me...

Positive

193

0 2020

3 2

**2** 3 44955 NaN 02-03-

2020

Find out how you can protect

yourself and love...

Extremely Positive

73 0 2020 3 2

**3**

4

44956 Chicagoland 02-03-

2020

#Panic buying hits #NewYork

City as anxious sh...

Negative

318

0 2020

3 2

**4** 5 44957 Melbourne, Victoria

**...** ... ... ... ... ... ... ... ... ... ... ...

03-03-

2020

#toiletpaper #dunnypaper #coronavirus #coronav...

Neutral 252 0 2020 3 3

**3793** 3794 48746 Israel ?? 16-03-

2020

**3794**

3795

48747 Farmington, 16-03-

NM 2020

Did you panic buy a lot of non-

perishable item...

Negative

213

0 2020

3 16

Meanwhile In A Supermarket in

Israel -- People...

Positive 127 0 2020 3 16

**3795** 3796 48748 Haverford, PA 16-03-

2020

**3796**

3797

48749

NaN 16-03-

2020

Gov need to do somethings instead of biar je r...

Extremely Negative

174

0 2020

3 16

Asst Prof of Economics @cconces was on @NBCPhi...

Neutral 185 2 2020 3 16

**3797** 3798 48750 Arlington,

Virginia

16-03-

2020

I and @ForestandPaper members

are committed to...

Extremely Positive

254 1 2020 3 16

# 3798 rows × 11 columns

In [23]:

STOPWORDS **=** set(stopwords**.**words('english'))

**def** cleaning\_stopwords(text):

**return** " "**.**join([word **for** word **in** str(text)**.**split() **if** word **not in** STOPWORDS])

test['OriginalTweet'] **=** test['OriginalTweet']**.**apply(**lambda** text: cleaning\_stopwords(text))

train['OriginalTweet'] **=** train['OriginalTweet']**.**apply(**lambda** text: cleaning\_stopwords(text))

In [24]:

**def** cleaning\_URLs(data):

**return** re**.**sub('((www\.[^\s]+)|(https?://[^\s]+))',' ',data)

train['OriginalTweet']**=** train['OriginalTweet']**.**apply(**lambda** x: cleaning\_URLs(x))

test['OriginalTweet']**=** test['OriginalTweet']**.**apply(**lambda** x: cleaning\_URLs(x))

In [25]:

english\_punctuations **=** string**.**punctuation punctuations\_list **=** english\_punctuations **def** cleaning\_punctuations(text):

translator **=** str**.**maketrans('', '', punctuations\_list)

**return** text**.**translate(translator)

train['OriginalTweet']**=** train['OriginalTweet']**.**apply(**lambda** x: cleaning\_punctuations(x))

test['OriginalTweet']**=** test['OriginalTweet']**.**apply(**lambda** x: cleaning\_punctuations(x))

In [26]:

**def** cleaning\_numbers(data):

**return** re**.**sub('[0-9]+','', data)

train['OriginalTweet']**=** train['OriginalTweet']**.**apply(**lambda** x: cleaning\_numbers(x))

test['OriginalTweet']**=** test['OriginalTweet']**.**apply(**lambda** x: cleaning\_numbers(x))

In [27]:

train

Out[27]:

2020

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UserName ScreenName Location TweetAt OriginalTweet Sentiment** | **char\_count** | **mention\_count** | **year** | **month** | **day** |
| **0** 3799 48751 London 16-03- MeNyrbie PhilGahan Chrisitv Neutral | 111 | 3 | 2020 | 3 | 16 |
| **1** 3800 48752 UK 16-03- advice Talk neighbours family Positive | 237 | 0 | 2020 | 3 | 16 |
| **2** 3801 48753 Vagabonds 16-03- Coronavirus Australia Positive | 131 | 0 | 2020 | 3 | 16 |
| **3** 3802 48754 NaN 16-03- My food stock one empty Positive | 306 | 0 | 2020 | 3 | 16 |

2020 exchange phone n...

2020 Woolworths give elderly ...

2020

**4** 3803 48755 NaN 16-03-

2020

PLEASE panic THERE WIL...

Me ready go supermarket COVID outbreak Not Im ...

Extremely Negative

310 0 2020 3 16

**...** ... ... ... ... ... ... ... ... ... ... ...

**41152** 44951 89903 Wellington City,

New Zealand

14-04-

2020

Airline pilots offering stock

supermarket shel...

Neutral 102 0 2020 4 14

**41153**

44952

89904

NaN

14-04-

Response complaint

2020 provided citing COVID relat...

Extremely Negative

138

0 2020

4 14

**41154** 44953 89905 NaN

14-04-

2020

You know itÂ s getting Positive 136 2 2020 4 14 tough KameronWilds rati...

**41155**

44954

89906

NaN

14-04- Is wrong smell hand sanitizer

2020

starting turn on...

Neutral

111

0 2020

4 14

**41156** 44955 89907

i love you so much || he/him

14-04-

2020

TartiiCat Well newused Rift S Negative 255 1 2020 4 14 going Amazon rn...

# 41157 rows × 11 columns

In [28]:

**from** nltk.tokenize **import** RegexpTokenizer tokenizer **=** RegexpTokenizer(r'\w+')

train['OriginalTweet'] **=** train['OriginalTweet']**.**apply(tokenizer**.**tokenize)

test['OriginalTweet'] **=** test['OriginalTweet']**.**apply(tokenizer**.**tokenize)

In [29]:

st **=** nltk**.**PorterStemmer()

**def** stemming\_on\_text(data):

data **=** [st**.**stem(word) **for** word **in** data]

**return** data

train['OriginalTweet']**=** train['OriginalTweet']**.**apply(**lambda** x: stemming\_on\_text(x))

test['OriginalTweet']**=** test['OriginalTweet']**.**apply(**lambda** x: stemming\_on\_text(x))

# lm = nltk.WordNetLemmatizer() def lemmatizer\_on\_text(txt): txt = [lm.lemmatize(word) for word in txt] return txt train['OriginalTweet'] = train['OriginalTweet'].apply(lambda x: lemmatizer\_on\_text(x)) train

In [32]:

train['Sentiment']**=**train['Sentiment']**.**replace('Extremely Positive','Positive') train['Sentiment']**=**train['Sentiment']**.**replace('Extremely Negative','Negative')

test['Sentiment']**=**test['Sentiment']**.**replace('Extremely Positive','Positive') test['Sentiment']**=**test['Sentiment']**.**replace('Extremely Negative','Negative')

In [33]:

train1**=**train

train1**.**Sentiment**=**train1**.**Sentiment**.**replace('Neutral',1) train1**.**Sentiment**=**train1**.**Sentiment**.**replace('Positive',2) train1**.**Sentiment**=**train1**.**Sentiment**.**replace('Negative',0)

In [34]:

train**=**train**.**drop(columns**=**['ScreenName','Location','TweetAt','year']) test**=**test**.**drop(columns**=**['ScreenName','Location','TweetAt','year'])

In [35]:

Y\_train**=**train['Sentiment'] Y\_test**=**test['Sentiment']

X\_train**=**train**.**drop("Sentiment",axis**=**1) X\_test**=**test**.**drop("Sentiment",axis**=**1)

In [36]:

print(X\_train**.**shape,Y\_train**.**shape,X\_test**.**shape,Y\_test**.**shape)

(41157, 6) (41157,) (3798, 6) (3798,)

In [37]:

**from** tensorflow.keras.models **import** Sequential

**from** tensorflow.keras.layers **import** LSTM, GRU,SimpleRNN

**from** tensorflow.keras.layers **import** Dense, Activation, Dropout

**from** tensorflow.keras.layers **import** Embedding

**from** tensorflow.keras.layers **import** BatchNormalization

**from** tensorflow.keras.layers **import** LSTM, Activation, Dense, Dropout, Input, Embedding

**from** tensorflow.keras.models **import** Model

**from** tensorflow.keras.losses **import** SparseCategoricalCrossentropy

**from** tensorflow.keras.optimizers **import** RMSprop

**from** tensorflow.keras.preprocessing.text **import** Tokenizer

**from** tensorflow.keras.preprocessing **import** sequence

In [38]:

Y\_train

Out[38]: 0 1

|  |  |
| --- | --- |
| 1 | 2 |
| 2 | 2 |
| 3 | 2 |
| 4 | 0 |
| 41152 | ..  1 |
| 41153 | 0 |
| 41154 | 2 |
| 41155 | 1 |
| 41156 | 0 |
| Name: | Sentiment, Length: 41157, dtype: int64 |

In [39]:

X**=**X\_train**.**OriginalTweet Y**=**Y\_train

In [40]:

max\_len **=** 500

tok **=** Tokenizer(num\_words**=**2000) tok**.**fit\_on\_texts(X)

sequences **=** tok**.**texts\_to\_sequences(X)

sequences\_matrix **=** sequence**.**pad\_sequences(sequences,maxlen**=**max\_len) word\_index **=** tok**.**word\_index

sequences\_matrix

Out[40]:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| array([[ | 0, | 0, | 0, | ..., | 0, | 0, | 0], |
| [ | 0, | 0, | 0, | ..., | 41, | 1039, | 98], |
| [  ..., | 0, | 0, | 0, | ..., | 122, | 1, | 88], |
| [ | 0, | 0, | 0, | ..., | 28, | 29, | 301], |
| [ | 0, | 0, | 0, | ..., | 2, | 1, | 2], |
| [ | 0, | 0, | 0, | ..., | 772, | 1, | 255]]) |

In [41]:

**from** sklearn **import** preprocessing le **=** preprocessing**.**LabelEncoder() le**.**fit(Y)

Y**=**le**.**transform(Y)

In [42]:

**from** sklearn.model\_selection **import** train\_test\_split

X\_train, X\_test, Y\_train, Y\_test **=** train\_test\_split(sequences\_matrix, Y, test\_size**=**0.3, random\_state**=**2)

In [43]:

*# A simpleRNN without any pretrained embeddings and one dense layer*

model **=** Sequential()

model**.**add(Embedding(len(word\_index) **+** 1,

200,

input\_length**=**max\_len)) model**.**add(SimpleRNN(200))

model**.**add(Dense(3, activation**=**'sigmoid'))

model**.**compile(loss **=**SparseCategoricalCrossentropy(), optimizer**=**'adam', metrics**=**['accuracy'])

model**.**summary()

Model: "sequential"

In [45]:

model**.**fit(X\_train, Y\_train, epochs**=**10)

|  |  |  |
| --- | --- | --- |
|  | | =================================================================  Total params: 9,731,003  Trainable params: 9,731,003 |
| Non-trainable params: 0 |
|  |
| In | [44]: | **from** sklearn **import** preprocessing, decomposition, model\_selection, metrics, pipeline  **def** roc\_auc(predictions,target): |
|  | | '''  This methods returns the AUC Score when given the Predictions and Labels  '''  fpr, tpr, thresholds **=** metrics**.**roc\_curve(target, predictions) roc\_auc **=** metrics**.**auc(fpr, tpr)  **return** roc\_auc |

Layer (type) Output Shape Param #

=================================================================

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| embedding (Embedding) | (None, | 500, | 200) | 9650200 |
| simple\_rnn (SimpleRNN) | (None, | 200) |  | 80200 |
| dense (Dense) | (None, | 3) |  | 603 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Epoch 1/10  901/901 [==============================] | - 349s | 385ms/step | - loss: | 0.9197 | - accuracy: | 0.5457 |
| Epoch 2/10  901/901 [==============================] | - 316s | 350ms/step | - loss: | 0.6041 | - accuracy: | 0.7746 |
| Epoch 3/10  901/901 [==============================] | - 318s | 353ms/step | - loss: | 0.6724 | - accuracy: | 0.7378 |
| Epoch 4/10  901/901 [==============================] | - 322s | 357ms/step | - loss: | 0.6792 | - accuracy: | 0.7304 |
| Epoch 5/10  901/901 [==============================] | - 404s | 449ms/step | - loss: | 0.6281 | - accuracy: | 0.7602 |
| Epoch 6/10  901/901 [==============================] | - 357s | 397ms/step | - loss: | 0.5463 | - accuracy: | 0.8057 |
| Epoch 7/10  901/901 [==============================] | - 408s | 453ms/step | - loss: | 0.5544 | - accuracy: | 0.7994 |
| Epoch 8/10  901/901 [==============================] | - 393s | 436ms/step | - loss: | 0.7248 | - accuracy: | 0.6865 |
| Epoch 9/10  901/901 [==============================] | - 303s | 336ms/step | - loss: | 0.6069 | - accuracy: | 0.7656 |
| Epoch 10/10  901/901 [==============================] | - 286s | 317ms/step | - loss: | 0.5372 | - accuracy: | 0.8021 |

Out[45]:

In [47]:

Out[47]:

In [48]:

<keras.callbacks.History at 0x2815aeb4d90>

Y\_test

array([2, 0, 0, ..., 0, 2, 0], dtype=int64)

386/386 [==============================] - 40s 104ms/step

scores

scores **=** model**.**predict(X\_test)

In [49]:

Out[49]:

In [1]:

'''score=[]

for s in scores:

array([[0.17390123, 0.08241998, 0.9550262 ],

|  |  |  |
| --- | --- | --- |
| [0.8365786 , | 0.58865094, | 0.29535702], |
| [0.9677965 , | 0.38413322, | 0.43616864], |
| ...,  [0.59325635, | 0.34331757, | 0.8882675 ], |
| [0.07579926, | 0.07291647, | 0.98322797], |
| [0.8149449 , | 0.10628812, | 0.28367418]], dtype=float32) |

score+=[int(np.where(s == s.max())[0])] score'''

Out[1]:

In [51]:

'score=[]\nfor s in scores:\n score+=[int(np.where(s == s.max())[0])]\nscore'

Out[51]:

In [52]:

Out[52]:

In [ ]:

0.7577745383867833

**from** sklearn **import** metrics

acc**=**metrics**.**accuracy\_score(score,Y\_test) acc

len(score)

12348