Capstone Project - 3

CoronaVirus Tweet Sentimental Analysis

Index

1) Aim of Our Project
2) Why Sentimental Analysis?
3) Data Summary
4) Exploratory data Analysis
5) Insights from EDA
6) What and Why Preprocessing?
7) Model Training
8) Count Vectorisation Method
9) Tf-IDF Method
10) Comparison of both method
11) Conclusion

Aim of our project

This challenge asks you to build a classification model to predict the sentiment of COVID-19 tweets.The tweets have been pulled from Twitter and manual tagging has been done then.

What is Sentimental Analysis?

Sentiment analysis (or **opinion mining**) is a natural language processing technique used to determine whether data is positive, negative or neutral.

Sentiment analysis is often performed on textual data to help businesses monitor brand and product sentiment in customer feedback, and understand customer needs.

Data Summary

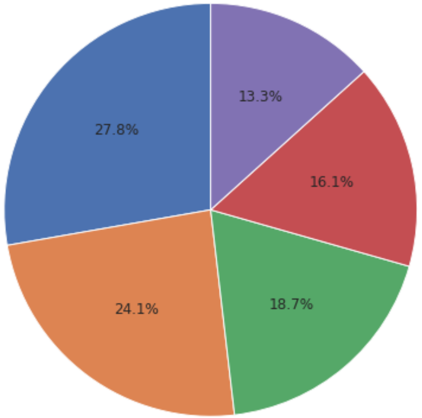
The following images gives basic detail about our data.
- Total Number of rows are 41156
- Data of type int64(2) and Object(4) are only present
- Location column has some null values.
- About 20.8% null values are present in location column, whereas all other columns are clean.

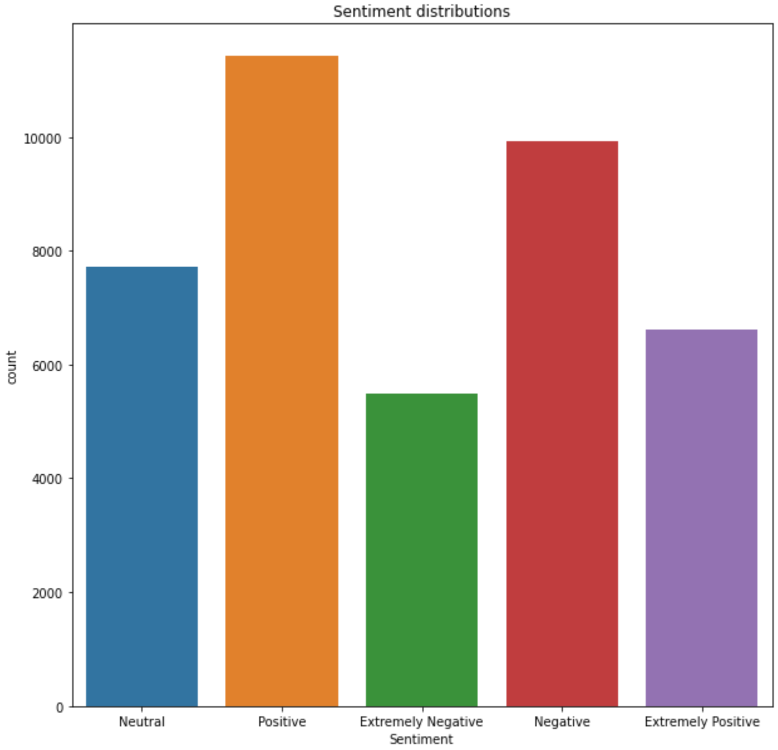
	Null Percentage
UserName	0.000000
ScreenName	0.000000
Location	20.871298
TweetAt	0.000000
OriginalTweet	0.000000
Sentiment	0.000000

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 41157 entries, 0 to 41156
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   UserName       41157 non-null  int64
 1   ScreenName     41157 non-null  int64
 2   Location       32567 non-null  object
 3   TweetAt        41157 non-null  object
 4   OriginalTweet  41157 non-null  object
 5   Sentiment      41157 non-null  object
dtypes: int64(2), object(4)
memory usage: 1.9+ MB
```

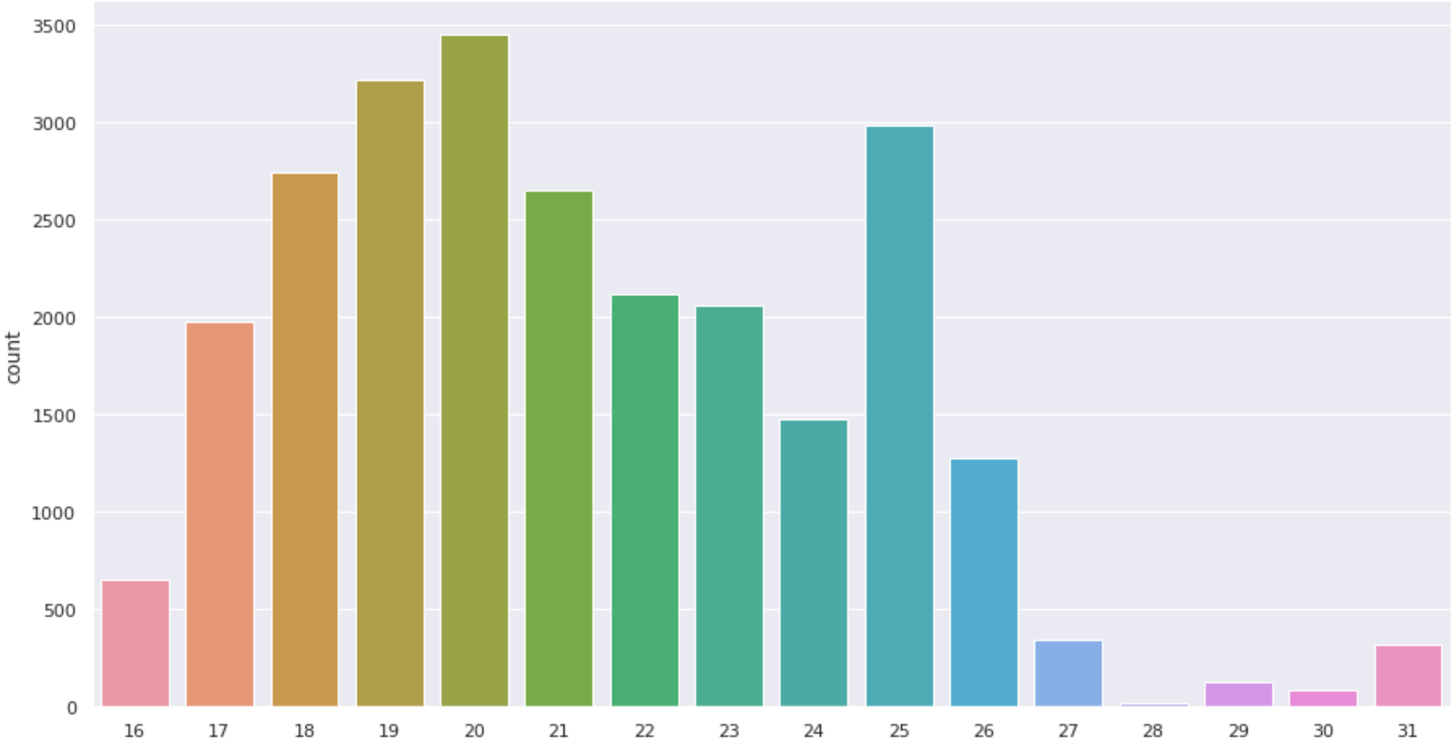
Exploratory Data Analysis

Sentimental Distribution

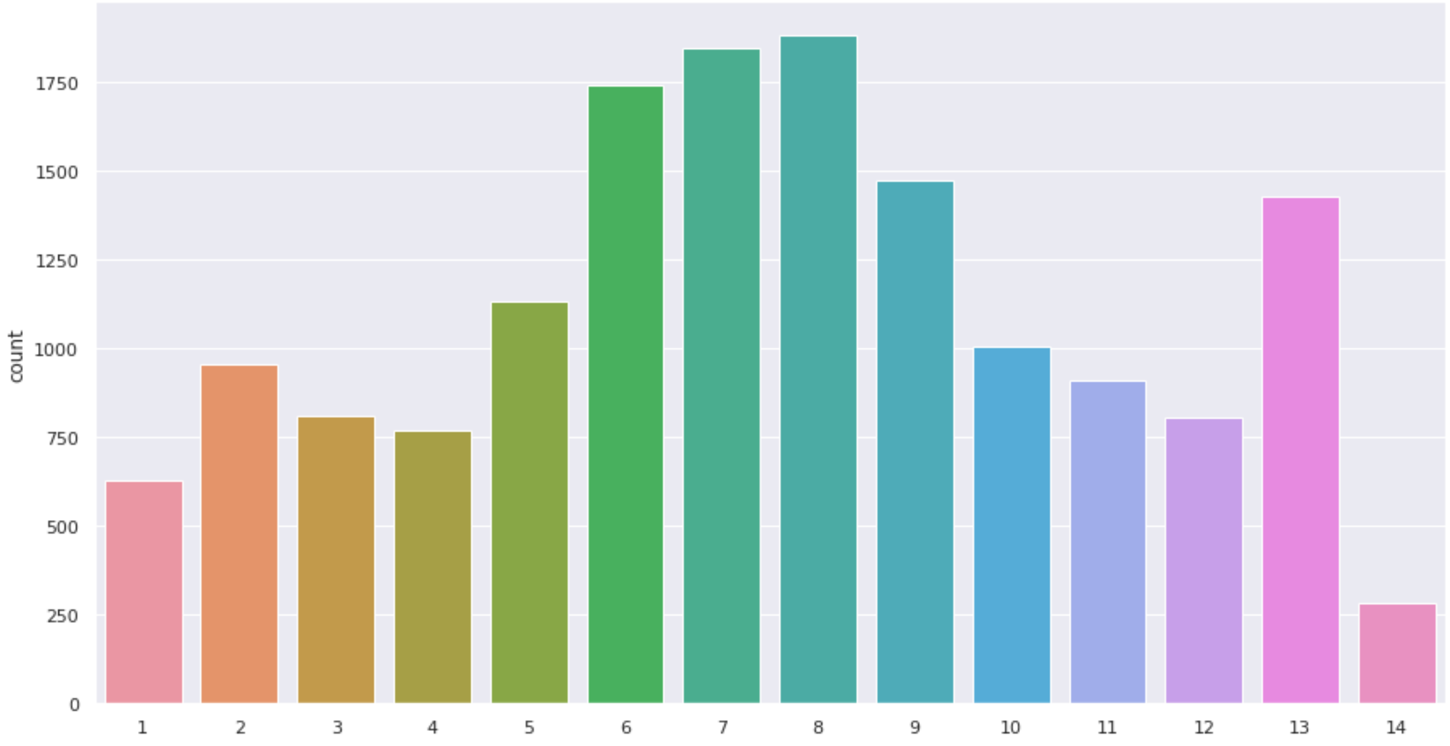




Tweet distribution in March



Tweet distribution in April



Top 10 locations which has most tweets from them.

	Location	Location_Count
0	London	9130
1	United States	528
2	London, England	520
3	New York, NY	395
4	Washington, DC	373
5	United Kingdom	337
6	Los Angeles, CA	281
7	India	268
8	UK	232
9	Australia	225

Top 10 dates on which maximums tweets have been made.

	Date	Count of Tweets
0	16-03-2020	3448
1	17-03-2020	3215
2	18-03-2020	2979
3	19-03-2020	2742
4	20-03-2020	2653
5	21-03-2020	2114
6	22-03-2020	2062
7	23-03-2020	1977
8	24-03-2020	1881
9	25-03-2020	1843

Insights from EDA

- 1) From sentimental distribution, it was clear that 24.1% people tweets something positive, which is help-full.
- 2) There are a total of 12220 uniques locations of tweets.
- 3) London has most active people who tweets related to covid-19.
- 4) India Ranks 7th in most active locations with an average of 4.4 tweets related to corona per day.
- 5) 16 MARCH 2020 had most number of tweets related to corona, with main as lockdown in most country had started on that day.
- 6)

Preprocessing of data

The preprocessing of the text data is an essential step as it makes the raw text ready for mining, i.e., it becomes easier to extract information from the text and apply machine learning algorithms to it. If we skip this step then there is a higher chance that you are working with noisy and inconsistent data.

The objective of this step is to clean noise those are less relevant to find the sentiment of tweets such as punctuation, special characters, numbers, and terms which don’t carry much weightage in context to the text.

	OriginalTweet	Sentiment	Processed_text
0	@MeNyrbie @Phil_Gahan @Chrisitv https://t.co/i...	Neutral	menyrbi philgahan chrisitv
1	advice Talk to your neighbours family to excha...	Positive	advic talk neighbour famili exchang phone numb...
2	Coronavirus Australia: Woolworths to give elde...	Positive	coronaviru australia woolworth give elderli di...
3	My food stock is not the only one which is emp...	Positive	food stock one empti pleas dont panic enough f...
4	Me, ready to go at supermarket during the #COV...	Extremely Negative	readi go supermarket covid19 outbreak im paran...

Model Training

A) Count Vectorizer Method

In order to use textual data for predictive modeling, the text must be parsed to remove certain words – this process is called **tokenization**. These words need to then be encoded as integers, or floating-point values, for use as inputs in machine learning algorithms.

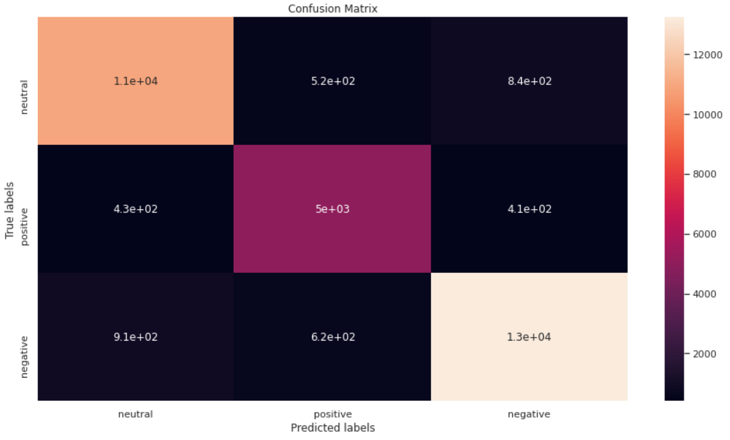
Scikit-learn's `CountVectorizer` is used to convert a collection of text documents to a vector of term/token counts. It also enables the pre-processing of text data prior to generating the vector representation. This functionality makes it a highly flexible feature representation module for text

Confusion Matrix for train and Test set

- Train Set

30/07/21, 4:28 PM

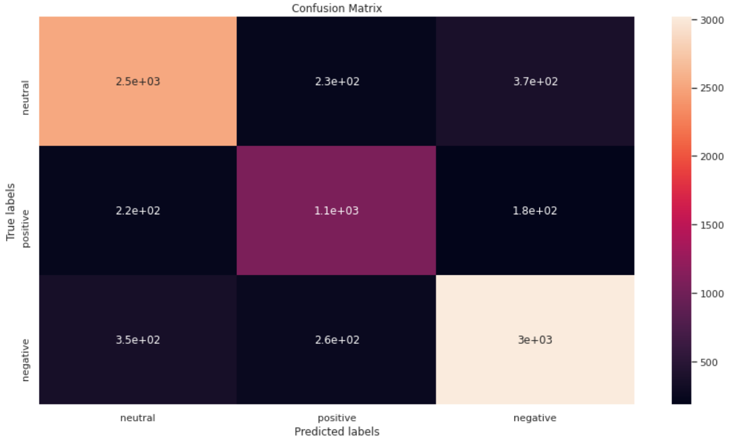
```
[[10959   519   837]
 [  426  5011   412]
 [  914   624 13223]]
[Text(0, 0.5, 'neutral'), Text(0, 1.5, 'positive'), Text(0, 2.5, 'negative')]
```



- Test Set

30/07/21, 4:30 PM

```
[[2523  229  374]
 [ 222 1075  185]
 [ 354  255 3015]]
[Text(0, 0.5, 'neutral'), Text(0, 1.5, 'positive'), Text(0, 2.5, 'negative')]
```



Classification Report of Train and Test Set

Train set report

	precision	recall	f1-score	support
-1	0.89	0.89	0.89	12299
0	0.86	0.81	0.83	6154
1	0.90	0.91	0.90	14472
accuracy			0.89	32925
macro avg	0.88	0.87	0.88	32925
weighted avg	0.89	0.89	0.89	32925

Test set report

	precision	recall	f1-score	support
-1	0.81	0.81	0.81	3099
0	0.73	0.69	0.71	1559
1	0.83	0.84	0.84	3574
accuracy			0.80	8232
macro avg	0.79	0.78	0.79	8232
weighted avg	0.80	0.80	0.80	8232

B) Tf-IDF Method

This is another method which is based on the frequency method but it is different to the bag-of-words approach in the sense that it takes into account, not just the occurrence of a word in a single document (or tweet) but in the entire corpus.

TF-IDF works by penalizing the common words by assigning them lower weights while giving importance to words which are rare in the entire corpus but appear in good numbers in few documents.

Let's have a look at the important terms related to TF-IDF:

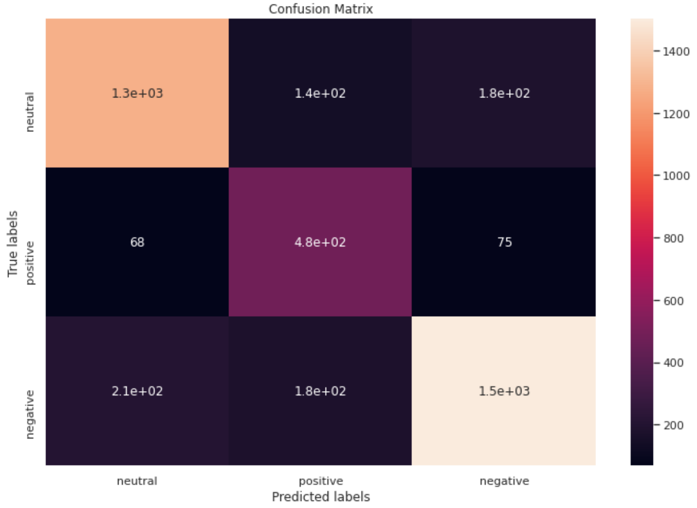
- TF = (Number of times term t appears in a document)/(Number of terms in the document)
- IDF = log(N/n), where, N is the number of documents and n is the number of documents a term t has appeared in.
- TF-IDF = TF*IDF

Confusion matrix for train and test data

- Test Confusion Matrix

30/07/21, 4:42 PM

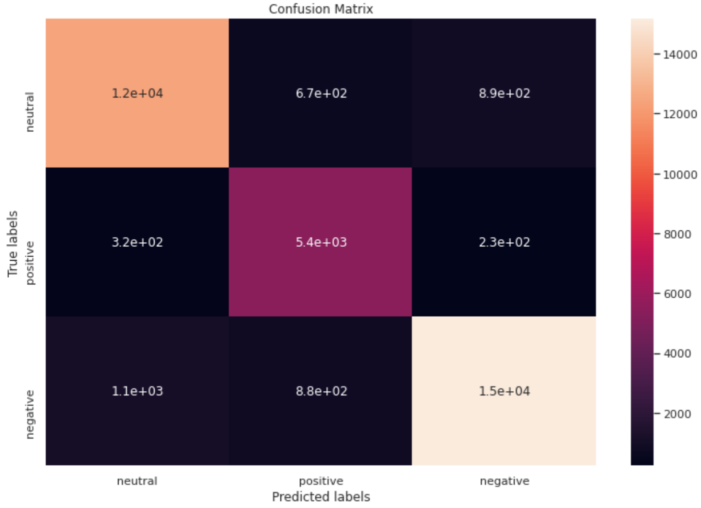
[Text(0, 0.5, 'neutral'), Text(0, 1.5, 'positive'), Text(0, 2.5, 'negative')]



- Train Confusion Matrix

30/07/21, 4:43 PM

[Text(0, 0.5, 'neutral'), Text(0, 1.5, 'positive'), Text(0, 2.5, 'negative')]



Classification report for train and test data

Train set

	precision	recall	f1-score	support
Negative	0.89	0.90	0.89	13846
Neutral	0.91	0.78	0.84	6910
Positive	0.89	0.93	0.91	16285
accuracy			0.89	37041
macro avg	0.89	0.87	0.88	37041
weighted avg	0.89	0.89	0.89	37041

Test Set

	precision	recall	f1-score	support
Negative	0.80	0.82	0.81	1552
Neutral	0.77	0.60	0.67	803
Positive	0.80	0.85	0.82	1761
accuracy			0.79	4116
macro avg	0.79	0.76	0.77	4116
weighted avg	0.79	0.79	0.79	4116