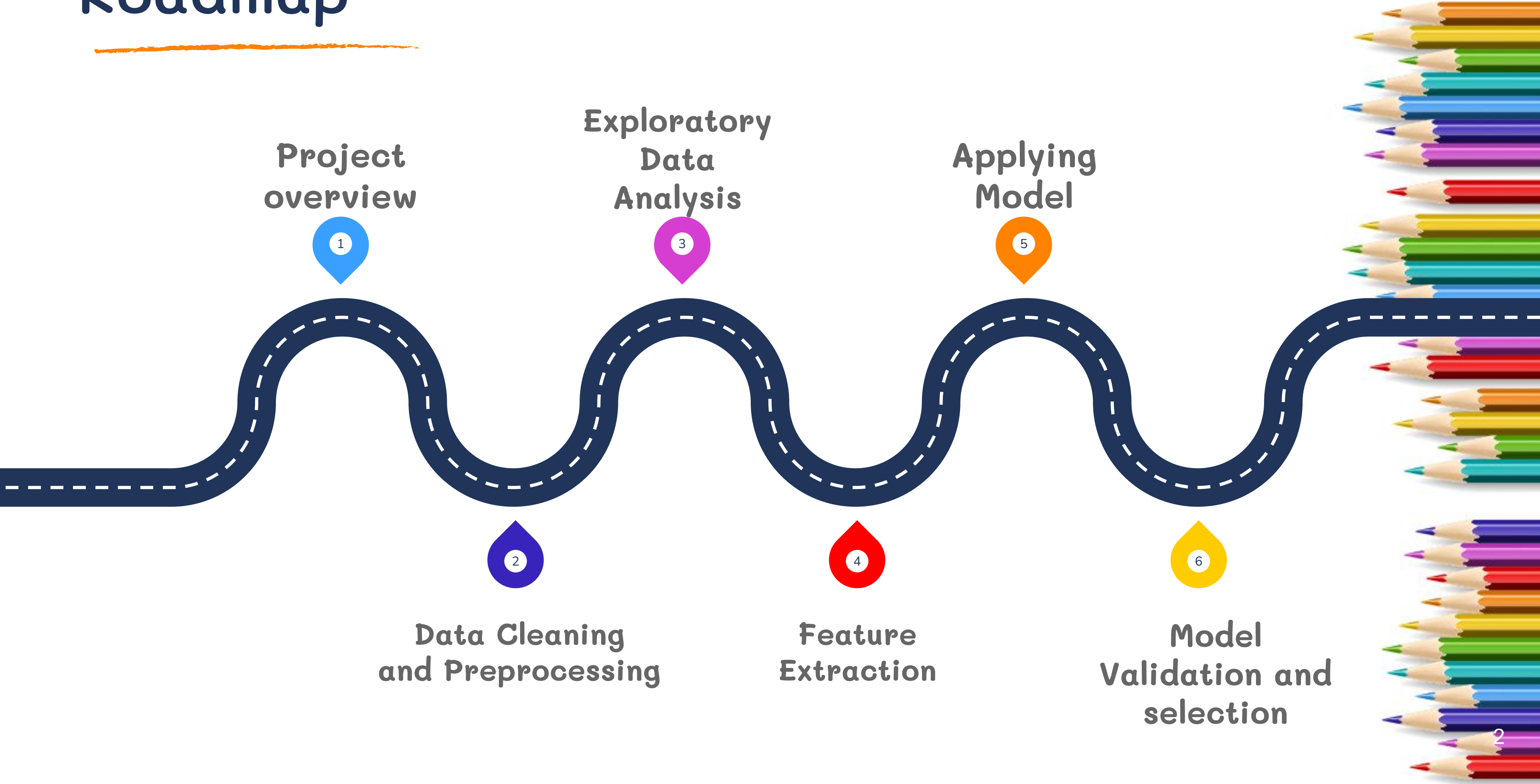
CAPSTONE PROJECT- 3

Coronavirus Tweet Sentiment Analysis

by
Gaurav Yogeshwar

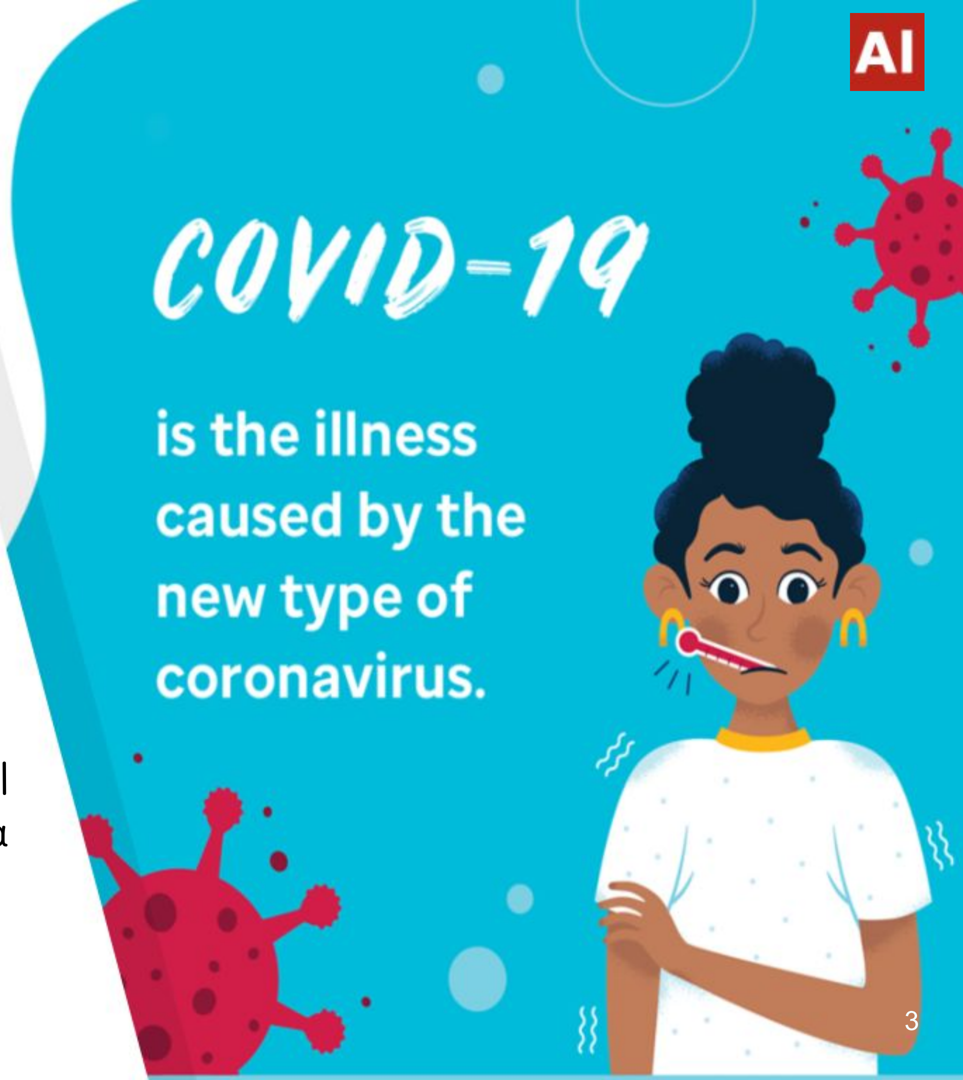
Roadmap


Project overview
1
Data Cleaning and Preprocessing
2
Exploratory Data Analysis
3
Feature Extraction
4
Applying Model
5
Model Validation and selection
6

Coronavirus

Coronavirus disease (COVID-19) is an infectious disease caused by the SARS-CoV-2 virus started in the year 2019.

WHO declared the outbreak a public health emergency of international concern on 30 January 2020 and a pandemic on 11 March 2020



Problem Description

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.

Data Summary

Location

Location of the tweet, it can be city , state or a country.

Tweet At

Timing of the tweet

Label

Sentiment of the tweet, our target variable that have four values.

Original Tweet

Original tweet, the text data.

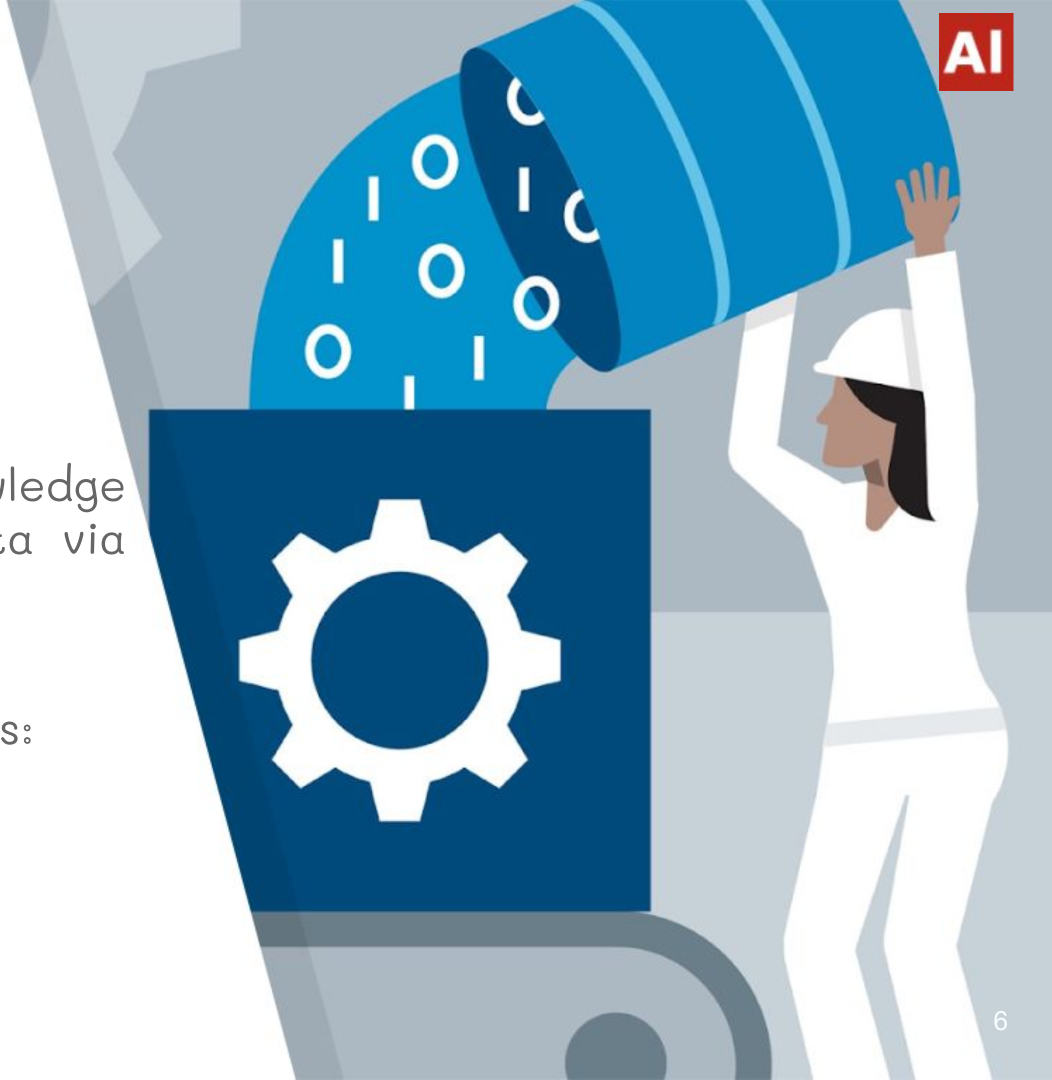
The names and usernames have been given codes to avoid any privacy concerns.

Data Cleaning, Preprocessing and Feature engineering

It is process of using domain knowledge to extract features from raw data via data mining technique.

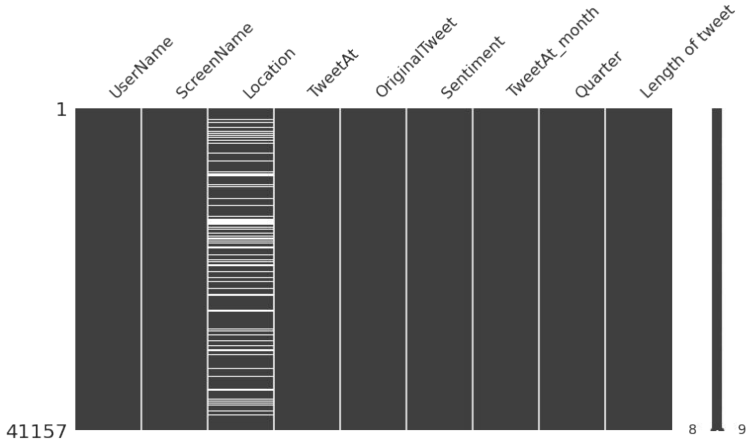
There are Three general approaches:

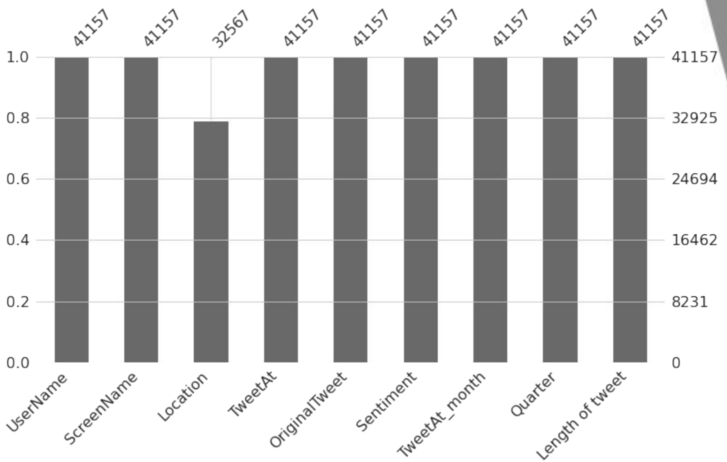
- Extracting Information
- Combining Information
- Transforming Information



Exploratory Data Analysis

Looking for missing values

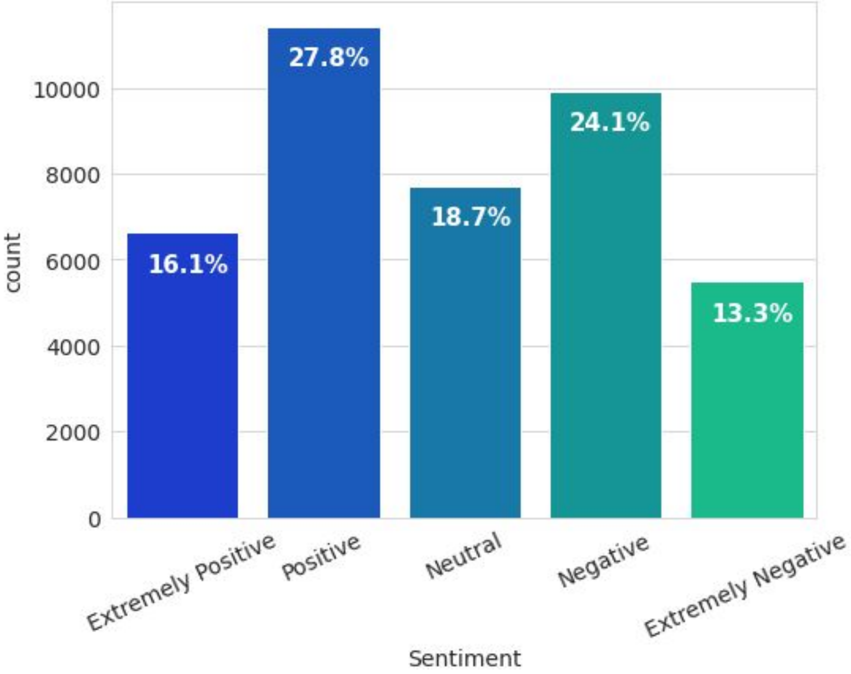




Insights:

- I have created few new features.
- Location feature have almost 20 percent missing values, other feature dont have any missing values.

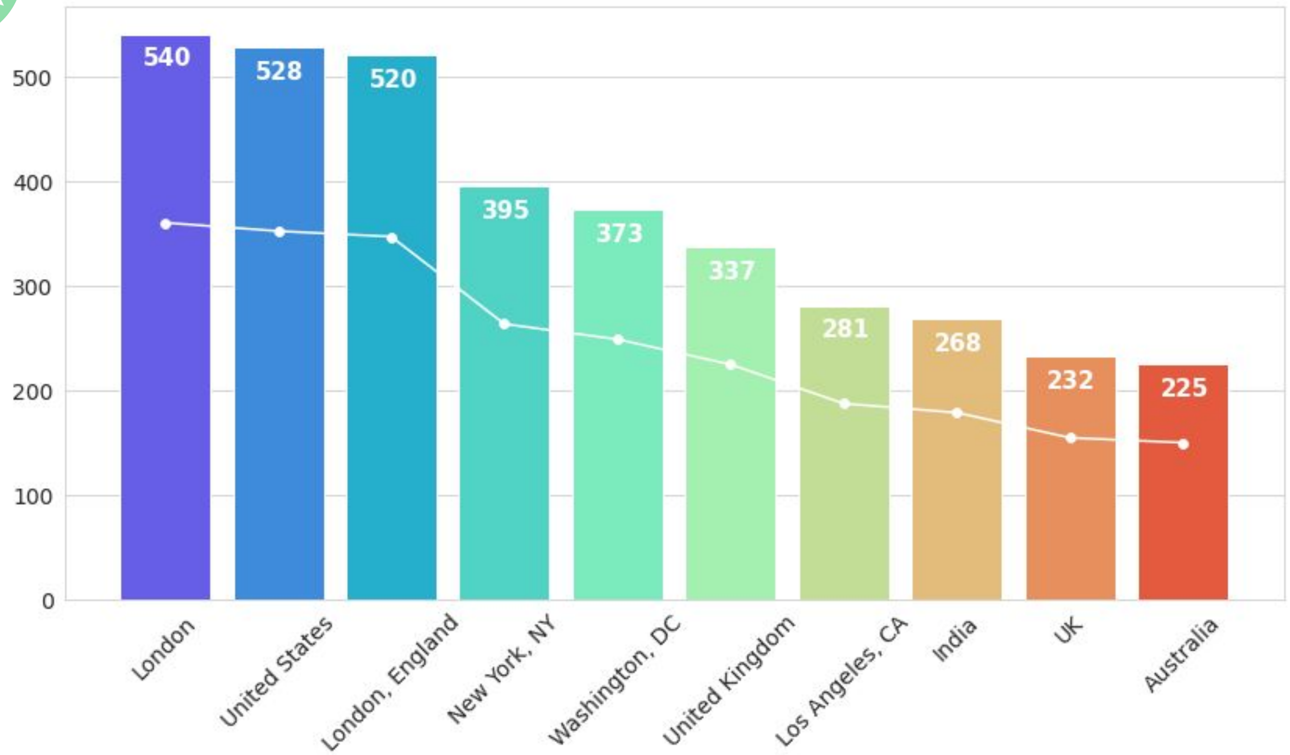
Display the Count and percentage of tweet per sentiment



Insights:

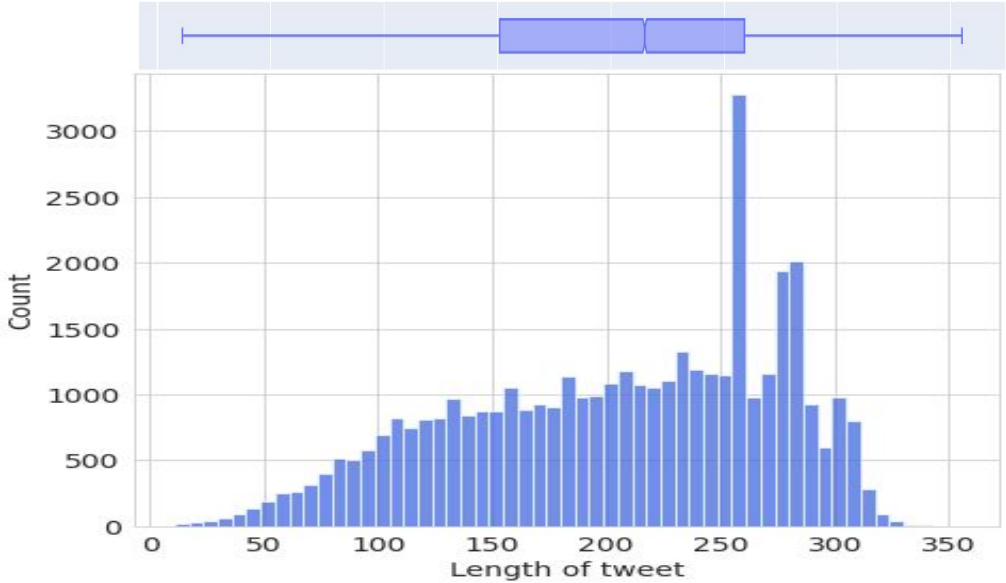
- All twitter sentiment is in significant numbers.
- There's more to tweets with a positive sentiment than a negative.

Bar plot for top 10 Locations



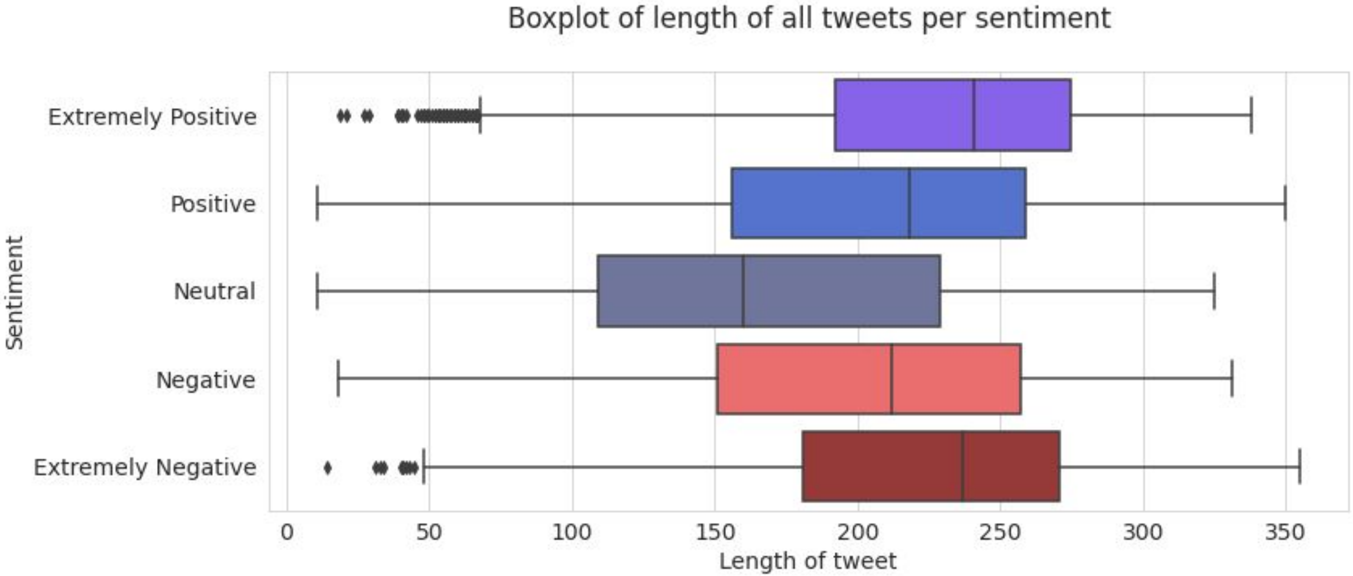
➢ As the location suggests, most of the places are from English speaking countries or country where people understand English, such as UK, USA, India, Canada, Australia etc., and among these most of them are also from the US and UK.

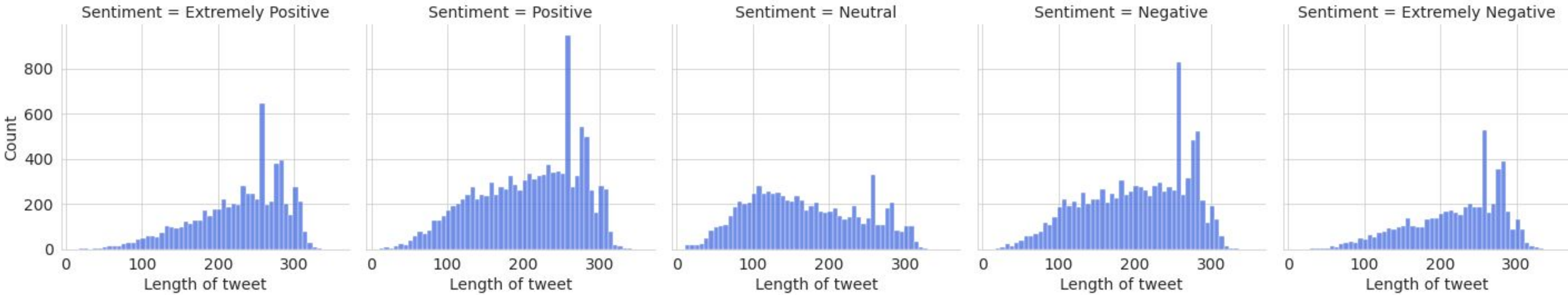
Box Plot and Histogram for Length of all Tweets

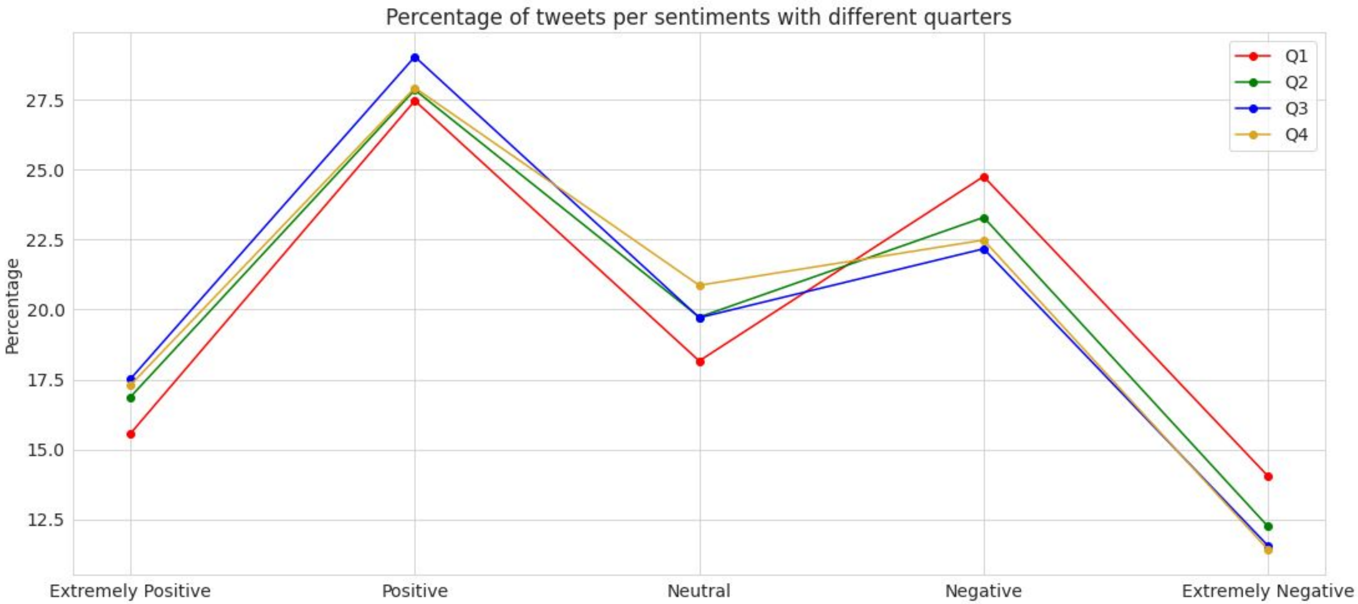


Insights:

- ➢ The boxplot and histogram shows that the length of the tweets is negatively skewed.
- ➢ The tallest tower is near 255.
- ➢ The second is near 280 as the Twitter official web page shows, which is the maximum limit of characters in a single tweet.


Boxplot of length of all tweets per sentiment
Extremely Positive
Positive
Neutral
Negative
Extremely Negative
Sentiment
0
50
100
150
200
250
300
350
Length of tweet


Sentiment = Extremely Positive
Sentiment = Positive
Sentiment = Neutral
Sentiment = Negative
Sentiment = Extremely Negative
Count
800
600
400
200
0
100
200
300
Length of tweet



Insights:

- First quarter has the highest percentage of negative and extreme negative tweets.
- Third quarter has the highest percentage of positive and extreme positive tweets.
- Fourth quarter has the highest percentage of Neutral tweets.

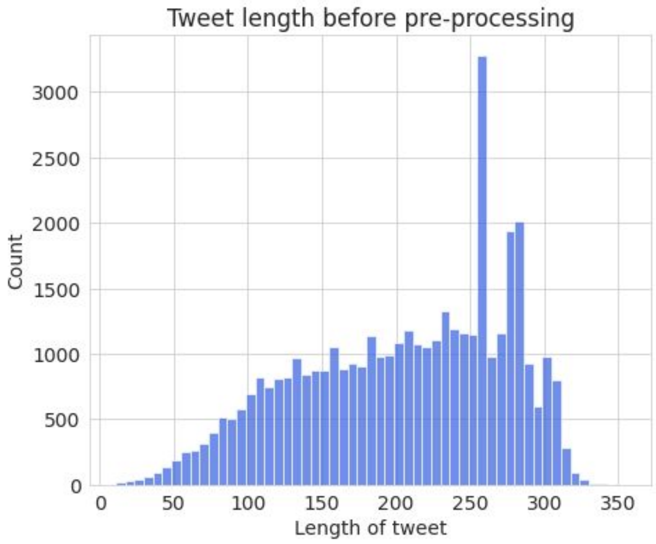
Text Preprocessing

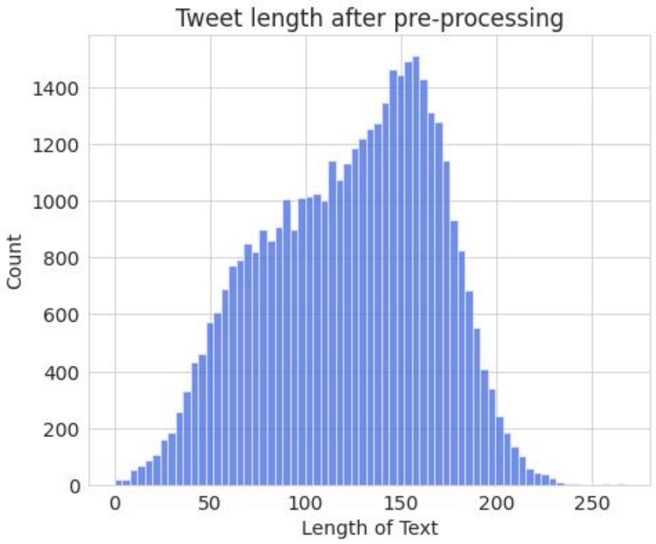
Text data is available to a great extent which is used to analyze and solve business problems. But before using the data for analysis or prediction, processing the data is important.

The various text preprocessing steps are:

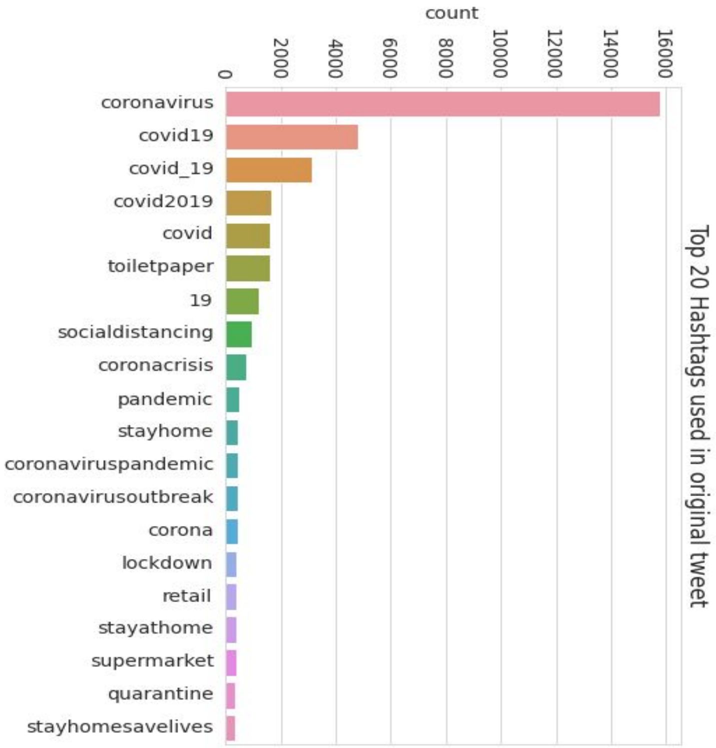
- ❖ Urls removal
- ❖ Tokenization
- ❖ Lower casing
- ❖ Punctuation removal
- ❖ Stop words removal
- ❖ Stemming or Lemmatization

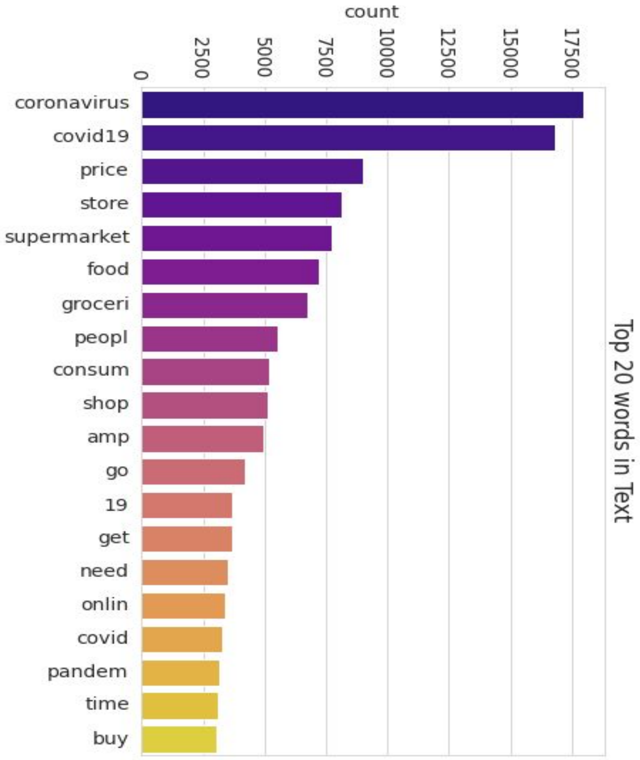
Effect on Text After Preprocessing





- ➢ The length and skewness are reduced after processing the original tweet.
- ➢ Disproportionate jumps are gone at specific length.





- Most of the hashtags are about coronavirus outbreak and pandemic, Social distancing, lockdown , staying at home etc..
- Due to the lockdown, people are also facing problems due to the closure of supermarkets, shortage of food, and running out of toilet papers.

Feature Extraction

1.Bag-of-Words:

The bag-of-words model converts text into fixed-length vectors by counting how many times each word appears.

	Word 1 Count	Word 2 Count	...	Word M
Message 1	0	1	...	0
Message 2	0	0	...	0
...	1	2	...	0
Message N	0	1	...	1

Eg..

Document	the	cat	sat	in	hat	with
the cat sat	1	1	1	0	0	0
the cat sat in the hat	2	1	1	1	1	0
the cat with the hat	2	1	0	0	1	1

Feature Extraction

2.TF-IDF :

Term Frequency measures how frequently a term occurs in a document.

$$tf(t,d) = \frac{Number\ of\ times\ term\ t\ appears\ in\ a\ document\ d}{Total\ number\ of\ terms\ in\ the\ document\ d}$$

$$\mathrm{tf}(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

Inverse Document Frequency, which measures how important a term is.

$$idf(t) = \log_e\left(\frac{Total\ number\ of\ documents}{Number\ of\ documents\ with\ term\ t\ in\ it}\right)$$

$$\mathrm{idf}(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

$$tf\text{-}idf(t,d,D) = tf(t,d) \cdot idf(t,d,D)$$

Applying models

We are applying overall 6 models..

1. Logistic Regression
2. Linear SVC
3. Multinomial NB
4. SGD Classifier
5. Decision Tree
6. Random Forest

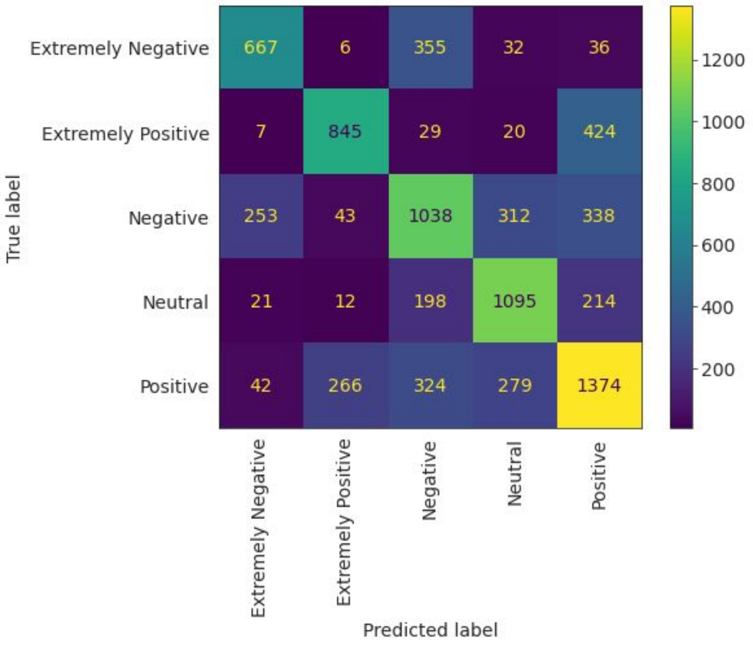
	Name	time_taken_sec	train_accuracy	test_accuracy
0	Logistic Regression	14.174774	0.810062	0.609842
3	SGD Classifier	0.759118	0.744258	0.573876
5	Random Forest	80.484173	0.999423	0.573390
	Linear SVC	14.409335	0.781717	0.552491
	Decision Tree	14.669399	0.999423	0.526853
	Multinomial NB	0.155201	0.610433	0.488578

Test Accuracy

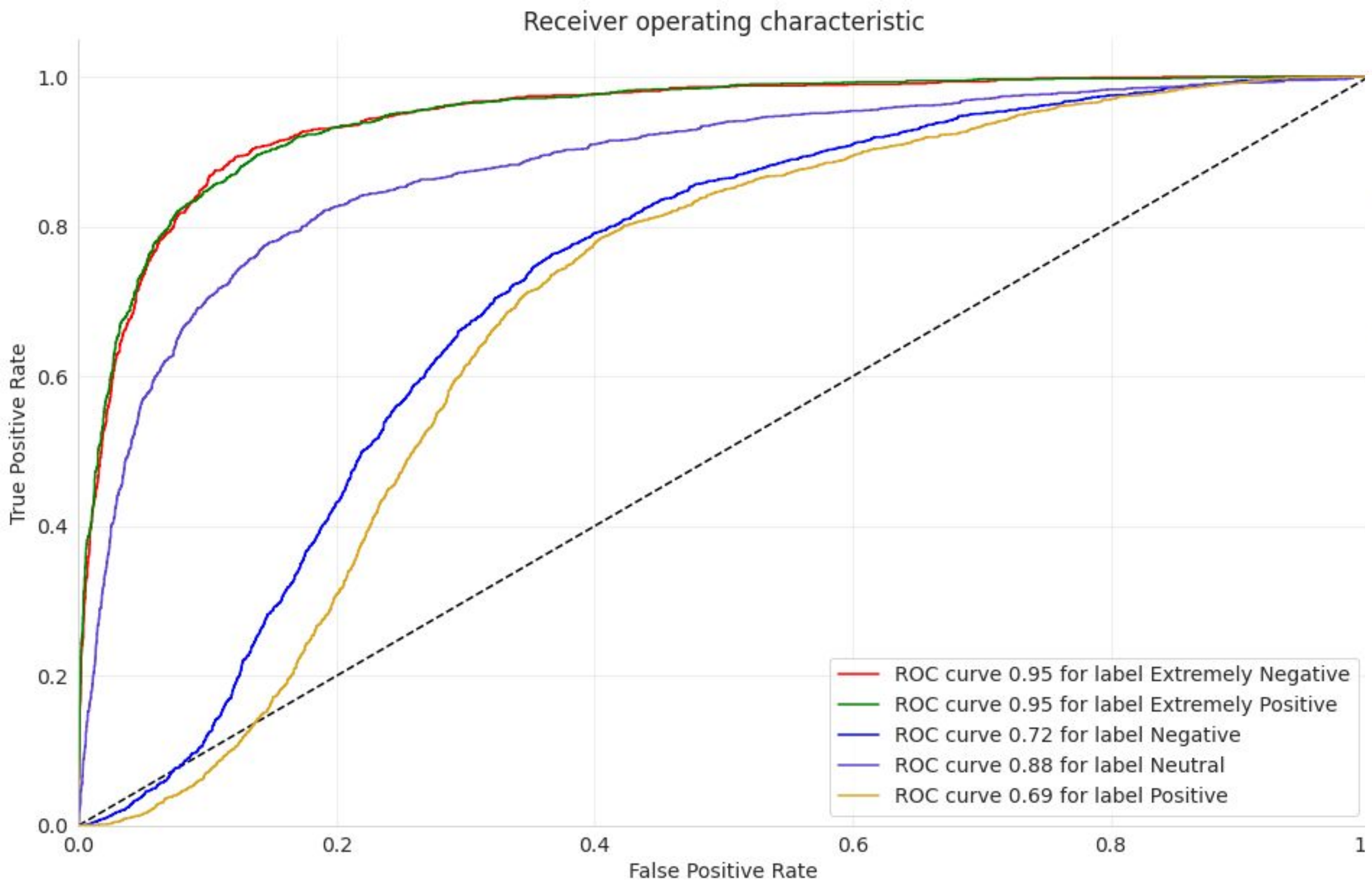
Confusion Matrix and Performance
Logistic regression

Logistic Regression

	precision	recall	f1-score	support
Extremely Negative	0.67	0.61	0.64	1096
Extremely Positive	0.72	0.64	0.68	1325
Negative	0.53	0.52	0.53	1984
Neutral	0.63	0.71	0.67	1540
Positive	0.58	0.60	0.59	2285
accuracy			0.61	8230
macro avg	0.63	0.62	0.62	8230
weighted avg	0.61	0.61	0.61	8230



Model have relatively high precision for Extremely Positive sentiment, and relatively high sensitivity for neutral sentiment.


Receiver operating characteristic
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1
ROC curve 0.95 for label Extremely Negative
ROC curve 0.95 for label Extremely Positive
ROC curve 0.72 for label Negative
ROC curve 0.88 for label Neutral
ROC curve 0.69 for label Positive

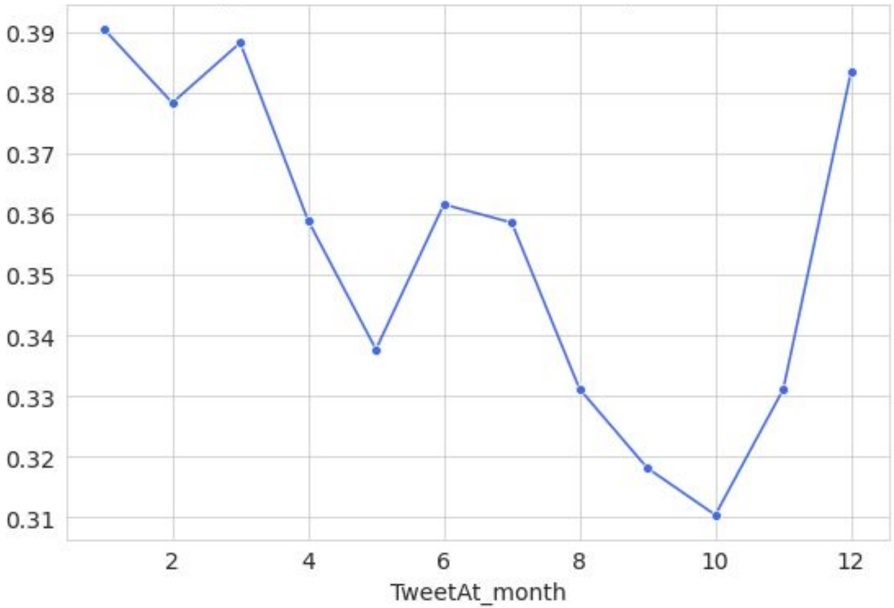
Multi to Binary class categorical variable

- We can see that since the target variable has five classes and the accuracy is not good we will convert it to a binary class.

- I will divide the sentiment into two parts, negative and non negative sentiments, negative means overall negative sentiment.

- It was the time of the corona pandemic so we will see how many people are full of negativity and panic.

Negative Sentiment Tweet Rate per month



Over time, the trend of tweets with negative sentiment is decreasing except in the last two months.

Tag	time_taken_sec	train_accuracy	test_accuracy
SGD Classifier cv	0.153503	0.905304	0.861118
Logistic Regression cv	1.121075	0.904818	0.857959
Linear SVC tfidf	0.196537	0.905213	0.855650
Linear SVC cv	2.790805	0.912170	0.848117
Logistic Regression tfidf	0.751916	0.873375	0.844836
SGD Classifier tfidf	0.071170	0.857364	0.834508
Random Forest cv	48.311036	0.999666	0.831106
Random Forest tfidf	46.300168	0.999635	0.825030
Multinomial NB cv	0.020718	0.813677	0.788821
Multinomial NB tfidf	0.023921	0.807905	0.781409
Decision Tree cv	13.981791	0.999666	0.766464
Decision Tree tfidf	18.539158	0.999635	0.758445

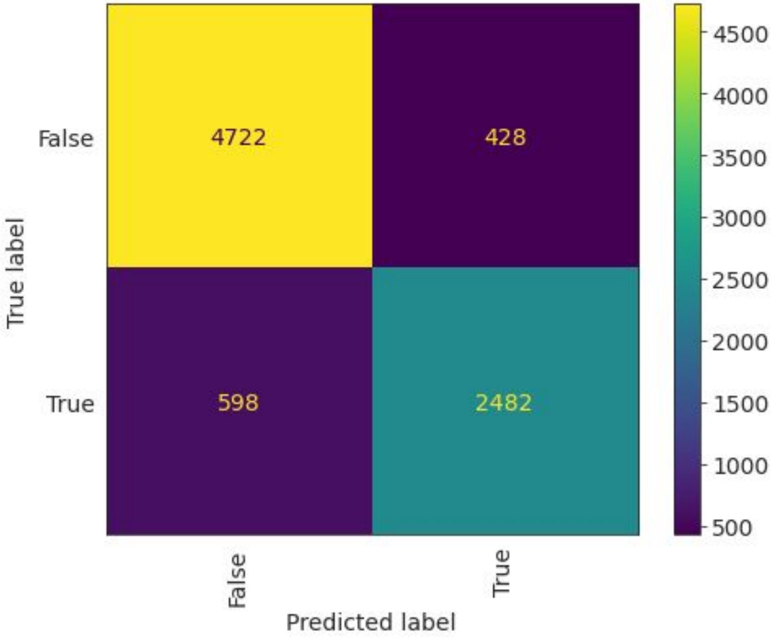
Test Accuracy with Binary variables

Hyperparameter Tuning

- Stochastic Gradient Descent Classifier with Countvectorizer has the best performance among all the models.

- Thus I am choosing it alongside Linear SVC with tf-idf vectorizer

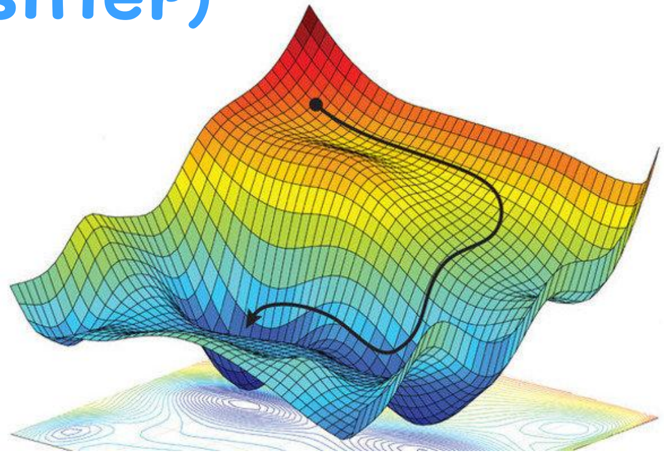
Confusion Matrix and Performance



Stochastic Gradient Descent Classifier

	precision	recall	f1-score	support
False	0.89	0.92	0.90	5150
True	0.85	0.81	0.83	3080
accuracy			0.88	8230
macro avg	0.87	0.86	0.87	8230
weighted avg	0.87	0.88	0.87	8230

Best Parameters (SGD Classifier)



We had chosen Random Forest Classifier for our prediction and best hyperparameters obtained are as below.

Penalty = L1

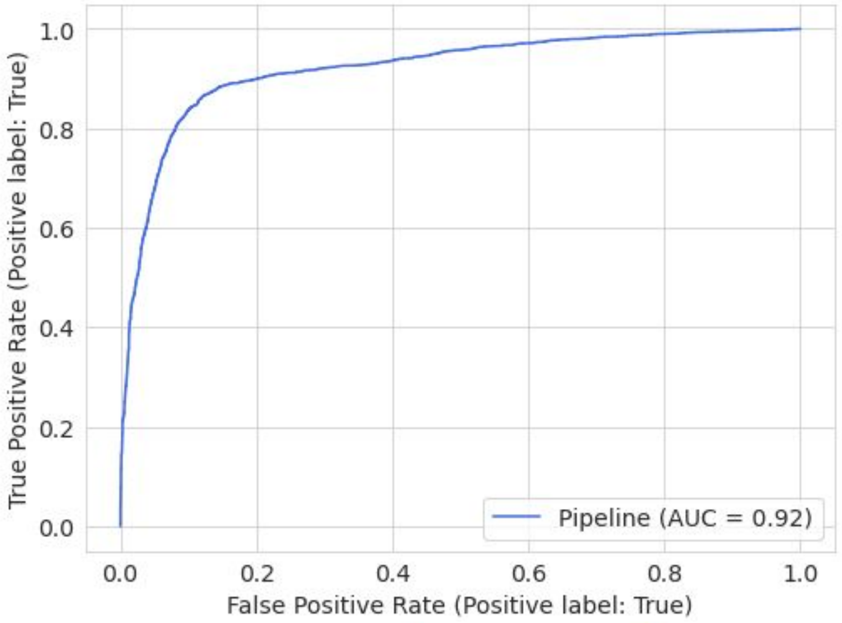
Max_iter = 100

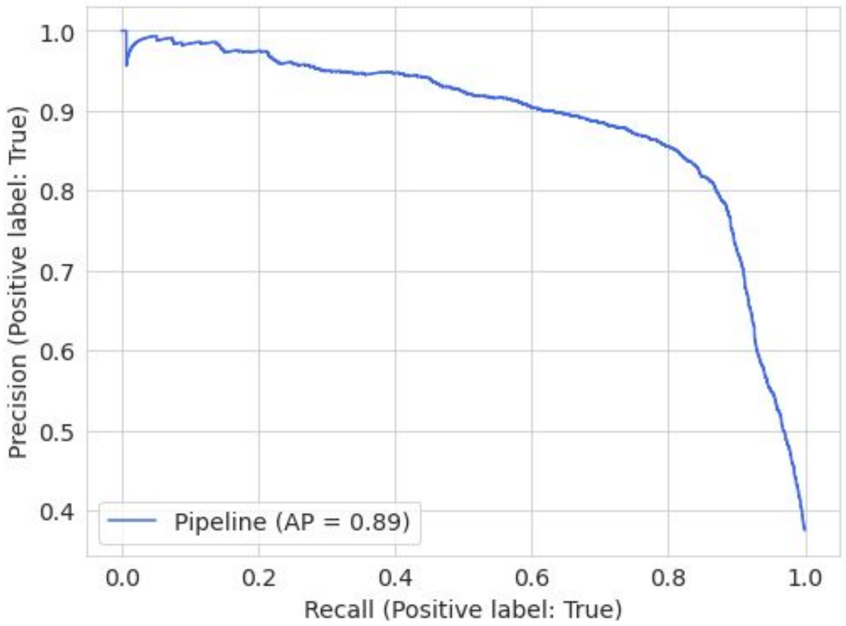
Loss= hing

epsilon=0.1

fit_intercept=True

ROC AUC and Precision vs Recall

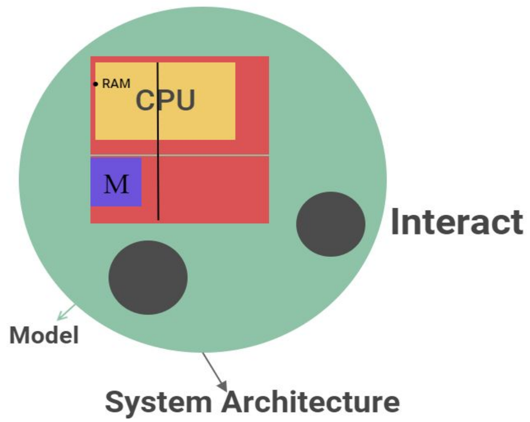




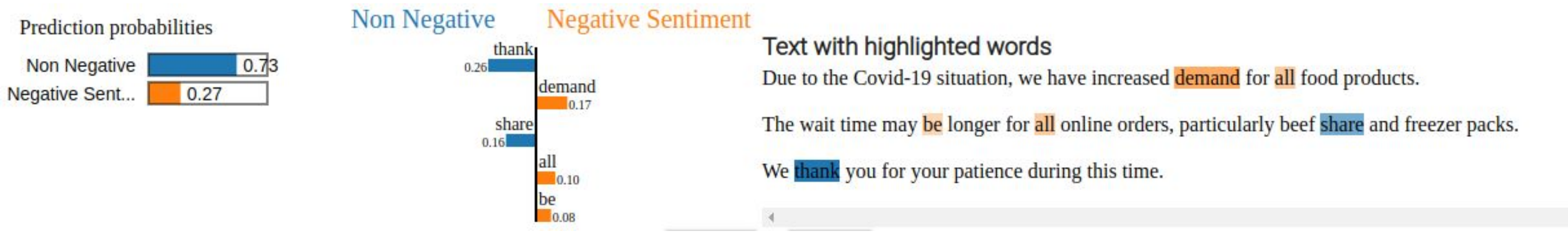
Model Persistence (Saving and Loading a Model)

Model persistence is the ability to save and load the machine learning model. It is desirable to have a way to persist the model for future use without having to retrain.

Joblib belongs to the python machine learning package — scikit-Learn. It is more efficient on objects that carry large numpy arrays and can be used instead of a pickle module for saving the model.



LIME



Prediction probabilities
Non Negative 0.73
Negative Sent... 0.27
Non Negative
Negative Sentiment
thank 0.26
demand 0.17
share 0.16
all 0.10
be 0.08
Text with highlighted words
Due to the Covid-19 situation, we have increased demand for all food products.
The wait time may be longer for all online orders, particularly beef share and freezer packs.
We thank you for your patience during this time.

Prediction probabilities
Non Negative 0.03
Negative Sent... 0.97
Non Negative
Negative Sentiment
panic 0.28
fear 0.26
racism 0.05
the 0.02
outbreak 0.01
Text with highlighted words
"Everything weÂre seeing in the current COVID-19 outbreak has been seen before in previous epidemics and pandemics; the rise of fear, racism, panic buying of food and medicines, conspiracy theories, the proliferation of quack cures" https://t.co/Pr8NpKX41A

Thank You!!!

Any questions?