Capstone project

Sentiment Analysis : Predicting sentiment of COVID-19 tweets

By
Pushkar srivastava
Rahul pandey

Point of discussion

- Introduction
- Problem statement
- Exploratory Data Analysis
- Looking For Null Values
- Values In Each Feature Column
- Top 10 locations of tweet
- Sentiments Countplot
- Data Preprocessing
- Story Generation and Visualization from Tweets
- Understanding the impact of Hashtags on tweets sentiment

Point of discussion

- Extracting Features from Cleaned Tweets
- Spitting Our Dataset into Training And Testing Dataset
- Countvectorizer
- Building Classification Models
- All the multiclass models test accuracy in descending order
- Evaluation of all binary classification models
- Conclusion

Introduction

We all have gone through the unprecedented time of the Coronavirus pandemic. Some people lost their lives, but many of us successfully defeated this new strain i.e. Covid-19. The virus was declared a pandemic by World Health Organization on 11th March 2020. We will analyze various types of “Tweets” gathered during pandemic times. The study can be helpful for different stakeholders.

In this project, we are going to predict the Sentiments of COVID-19 tweets. The data gathered from the Tweeter and I’m going to use Python environment to implement this project.

Problem statement

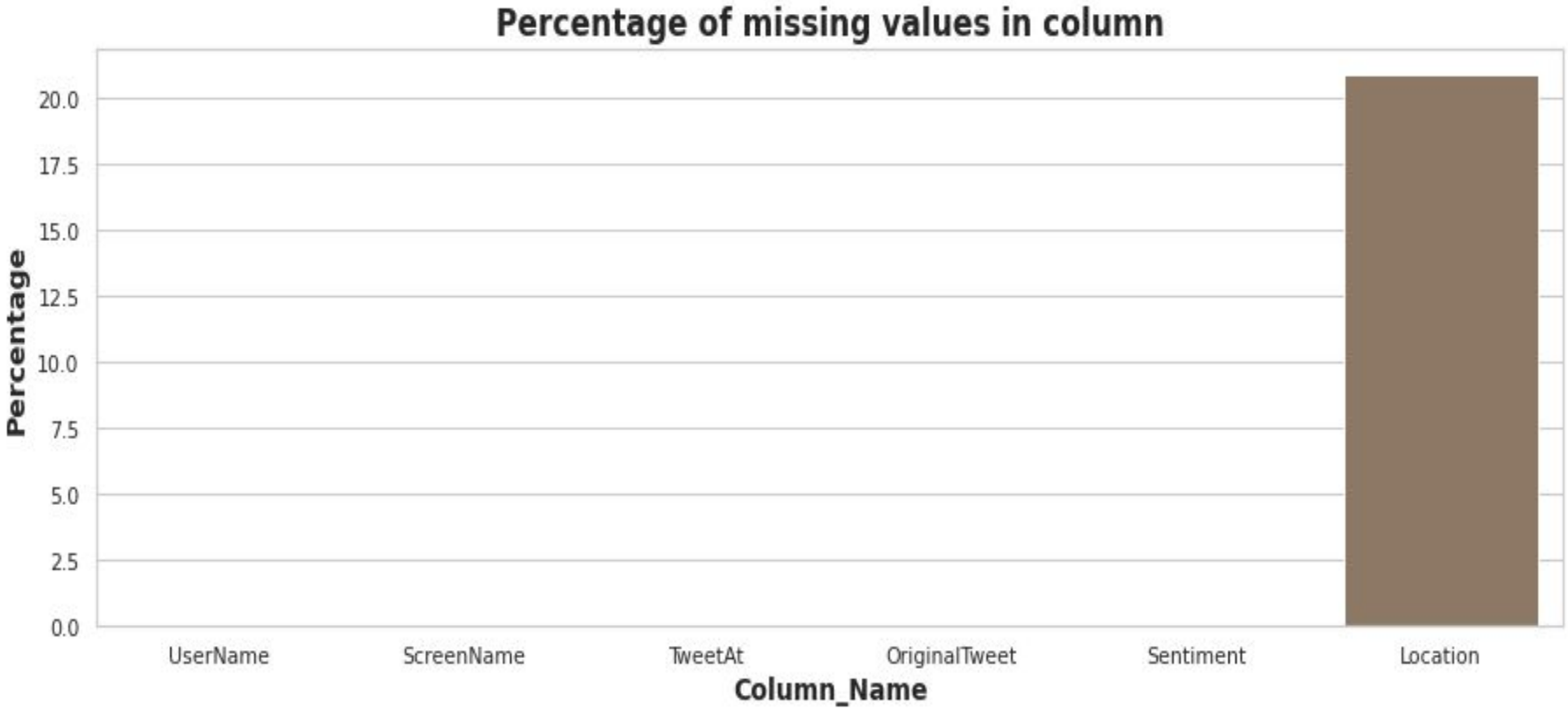
We are going to build a classification model to predict the sentiment of COVID-19 tweets.The tweets have been pulled from Twitter and manual tagging has been done then.

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

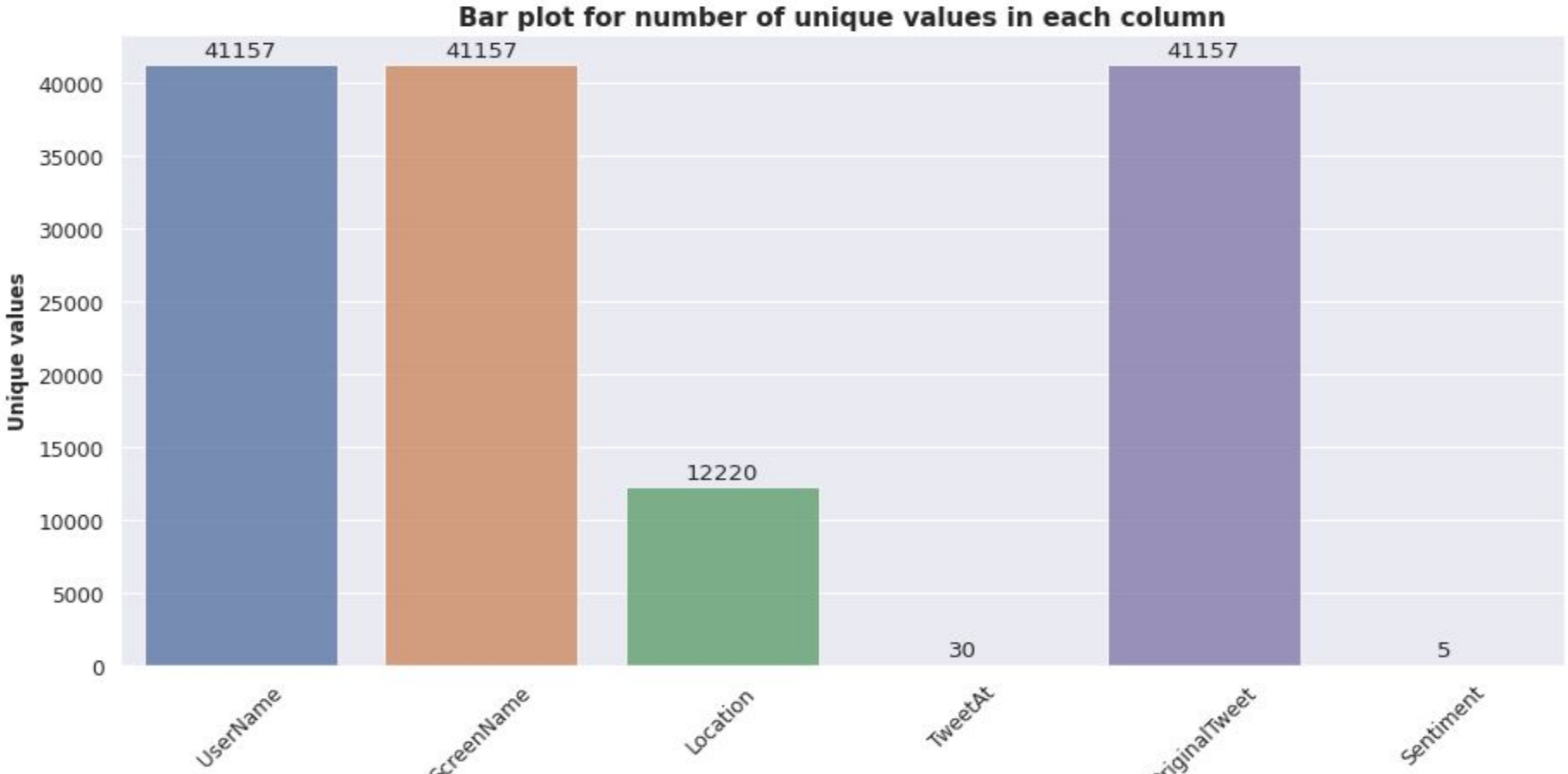
Exploratory Data Analysis

Looking For Null Values

Only Location column contains 20.8% null values. But this column is not useful for our sentiment analysis. Hence we will neglect these null values and focus on rest of the features

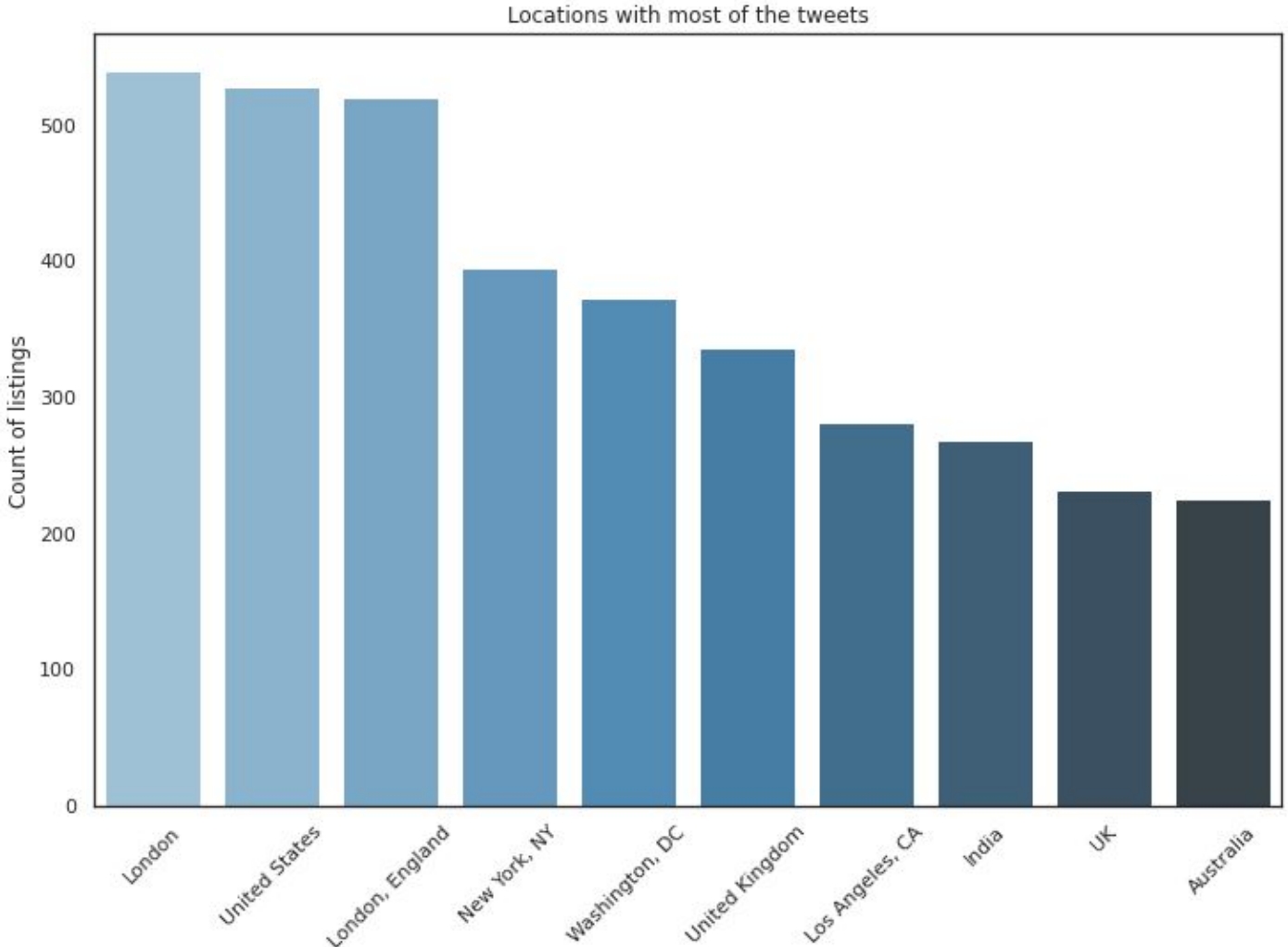

Percentage of missing values in column
Percentage
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
UserName
ScreenName
TweetAt
OriginalTweet
Sentiment
Location
Column_Name

Values In Each Feature Columns

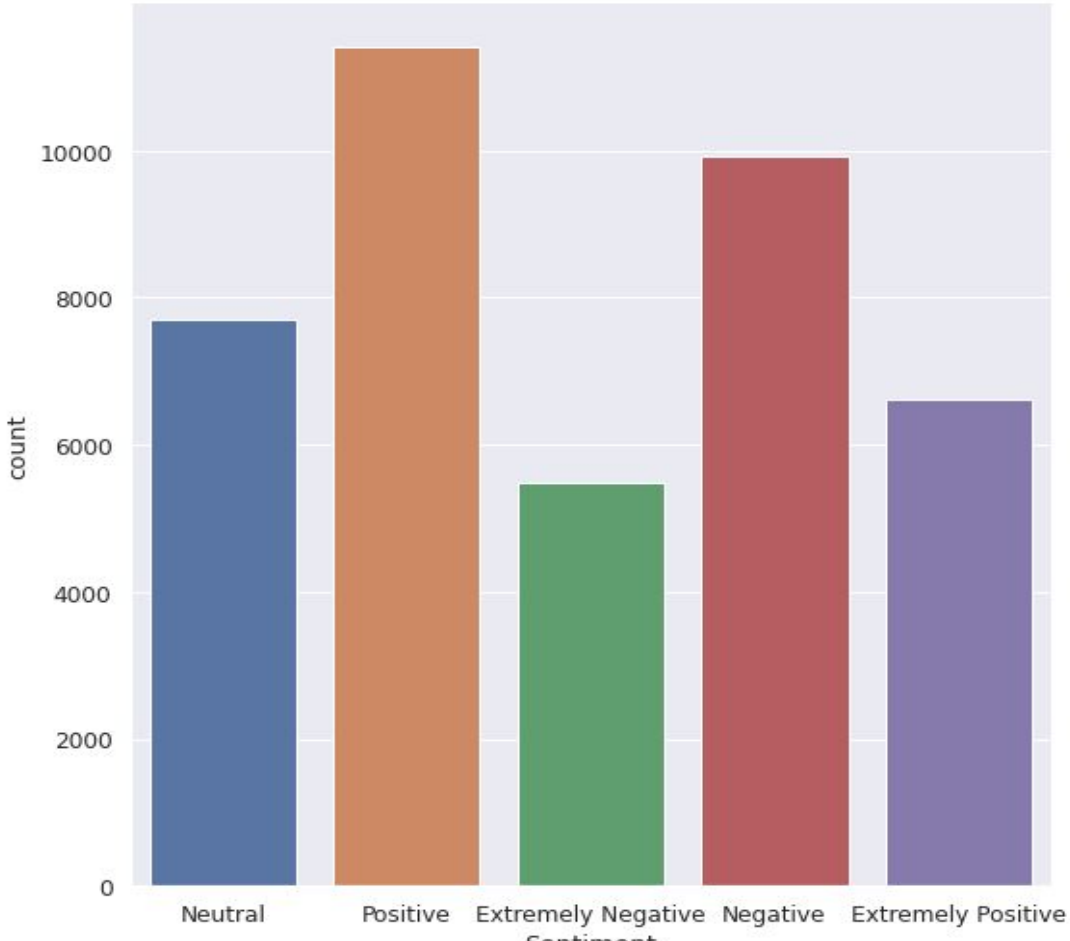


Top 10 locations of tweet

London -	540
United States-	528
London, England-	520
New York, NY-	395
Washington DC -	373
United Kingdom-	337
Los Angeles, CA-	281
India-	268
UK-	232
Australia-	225


Locations with most of the tweets
Count of listings
0
100
200
300
400
500
London
United States
London, England
New York, NY
Washington, DC
United Kingdom
Los Angeles, CA
India
UK
Australia

Sentiments Countplot



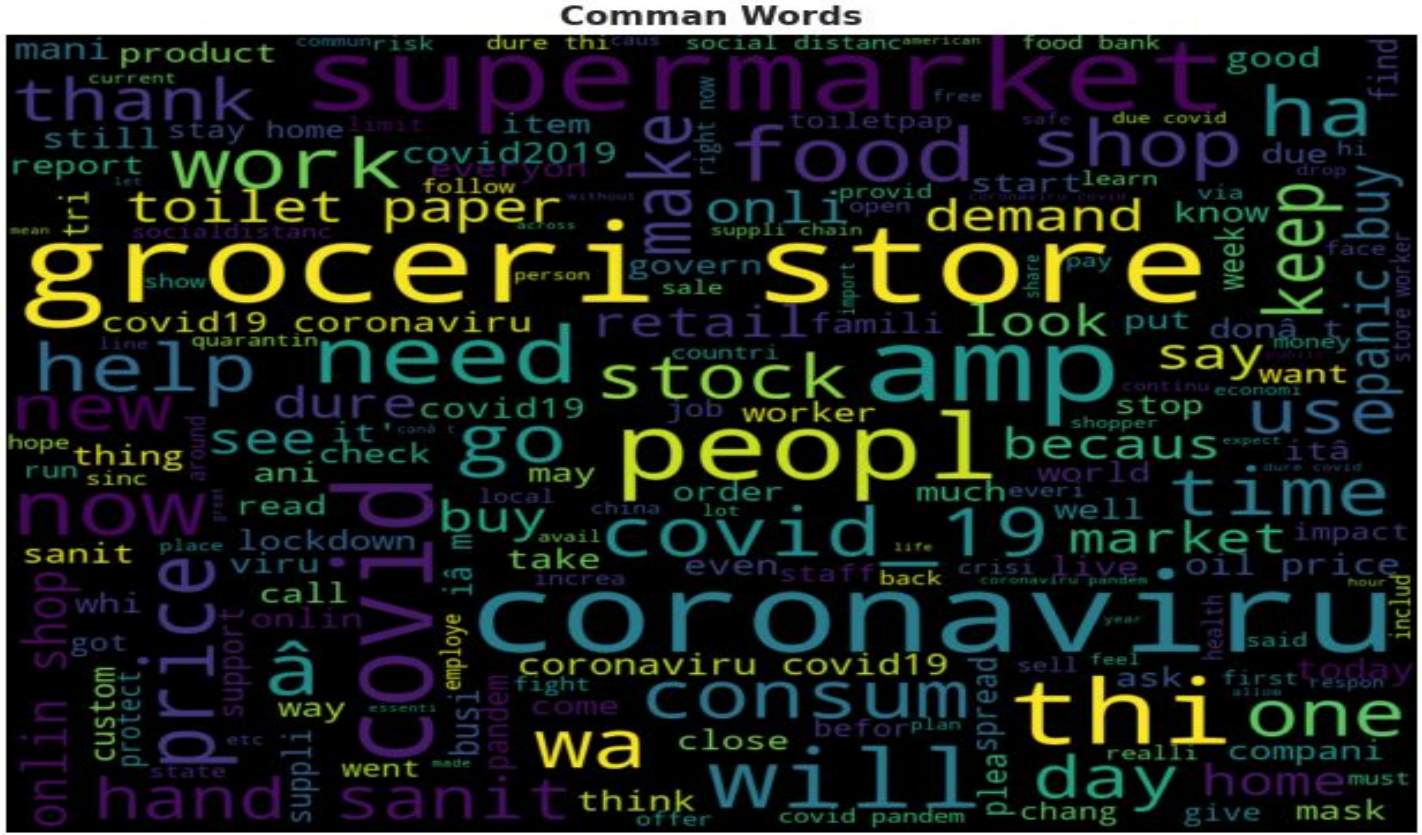
Data Preprocessing

- **Removing @user from Tweets**
- **Removed HTTP And URLS from Tweets**
- **Removing Punctuation, Numbers, and Special Characters**
- **Removing Short Words**
- **Tokenization**
- **Stemming**

Story Generation and Visualization from Tweets

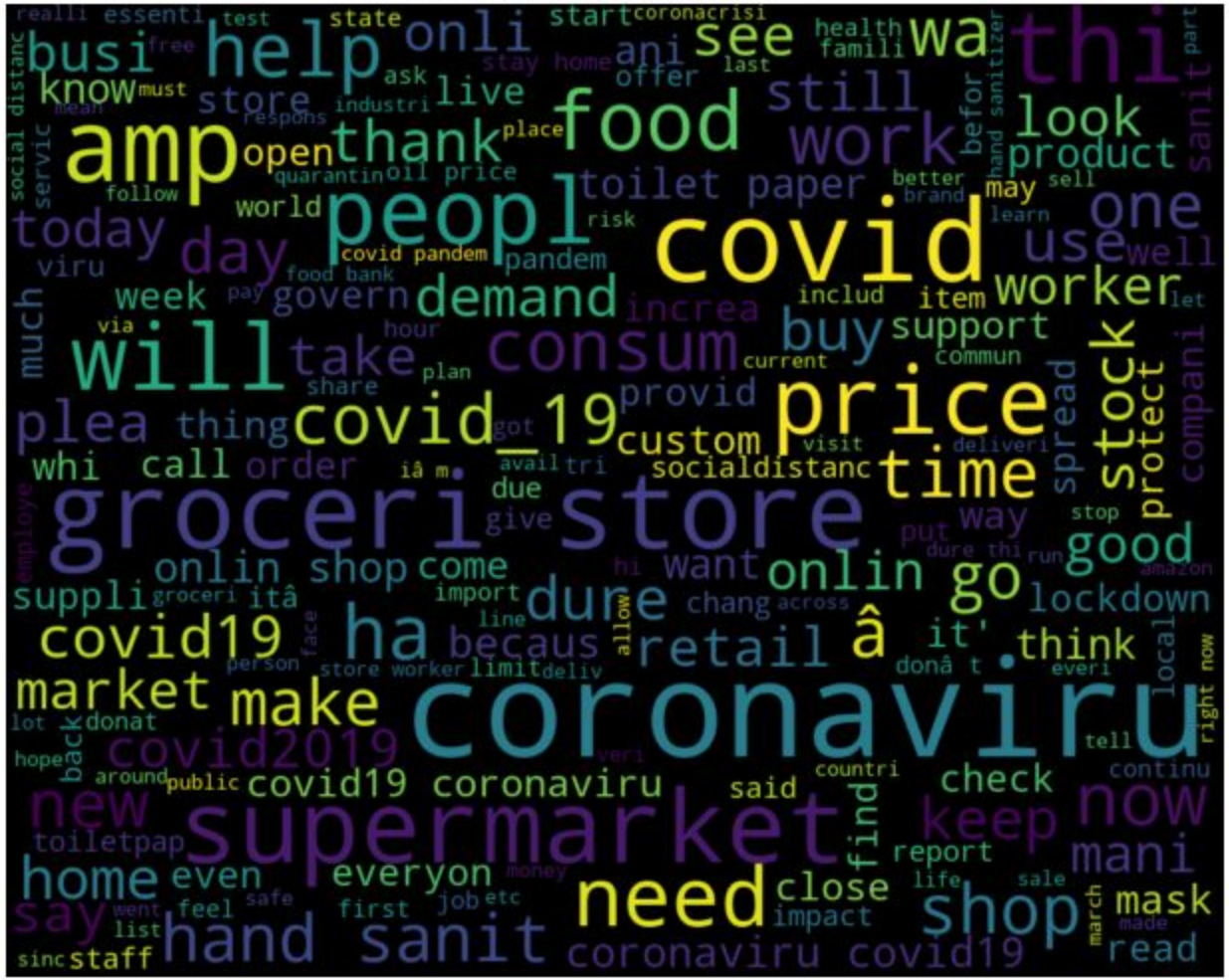
Understanding the common words used in the tweets:

WordCloud

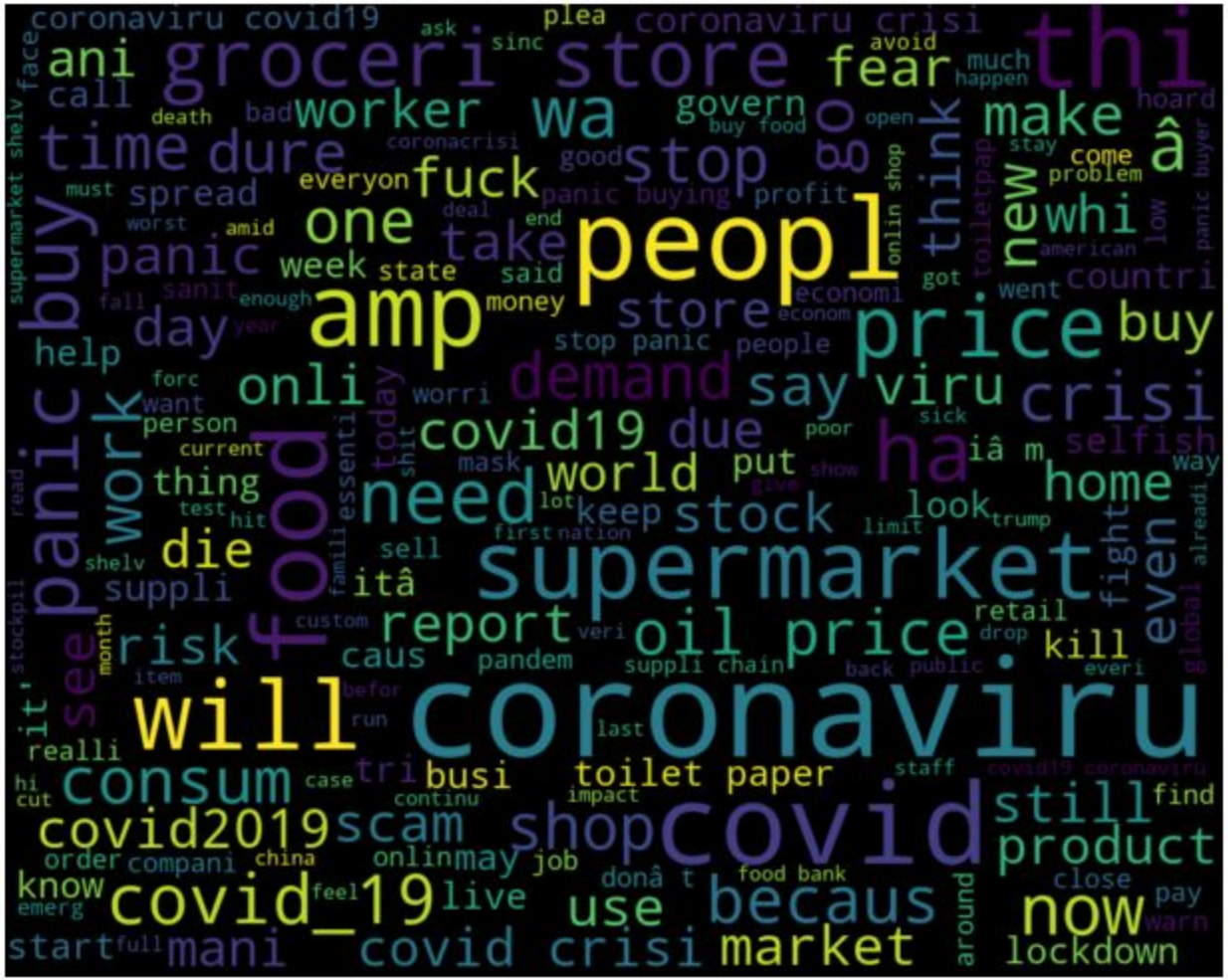


Comman Extremely Positive Words

Comman Positive Words

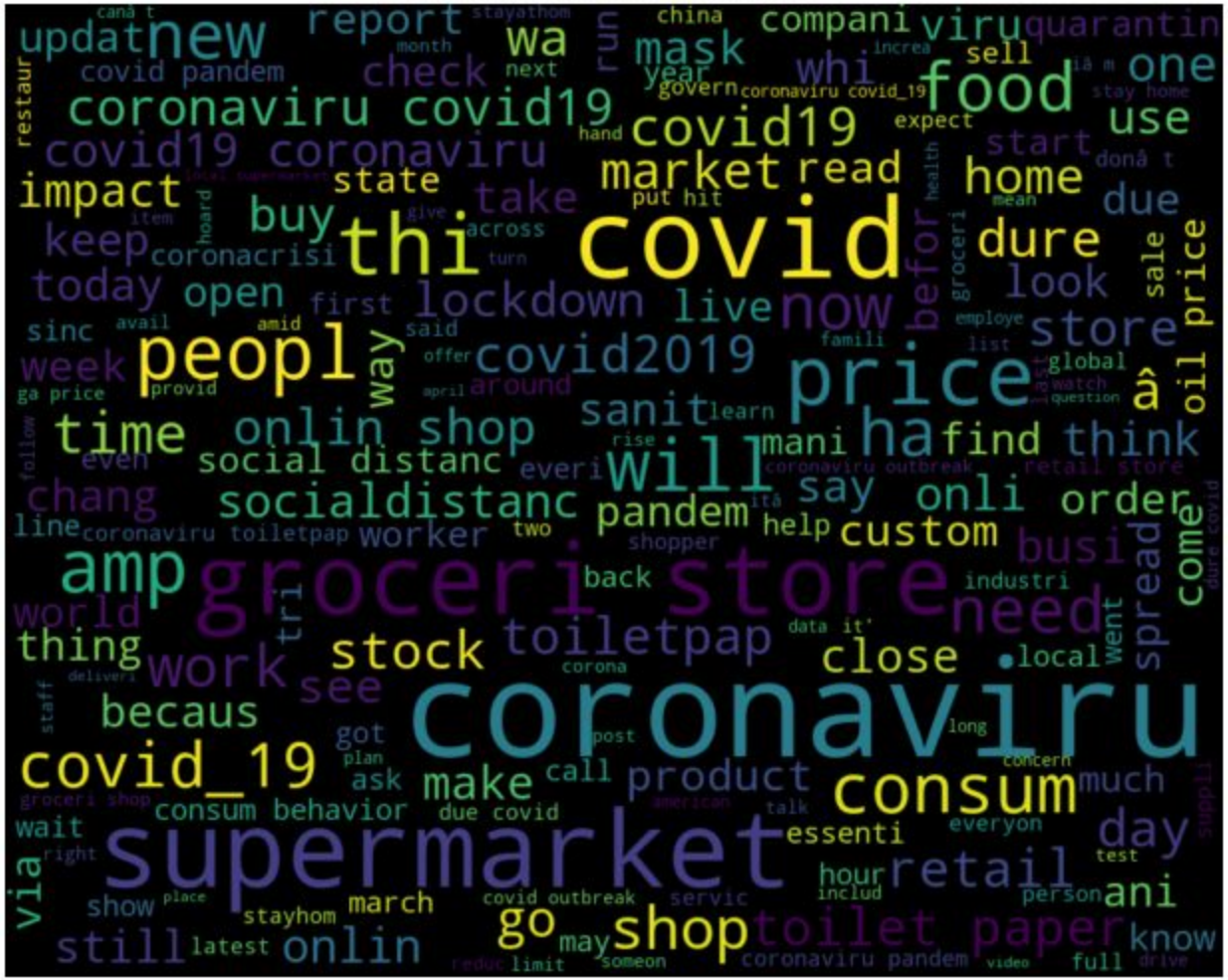


Comman Extremely Negative Words



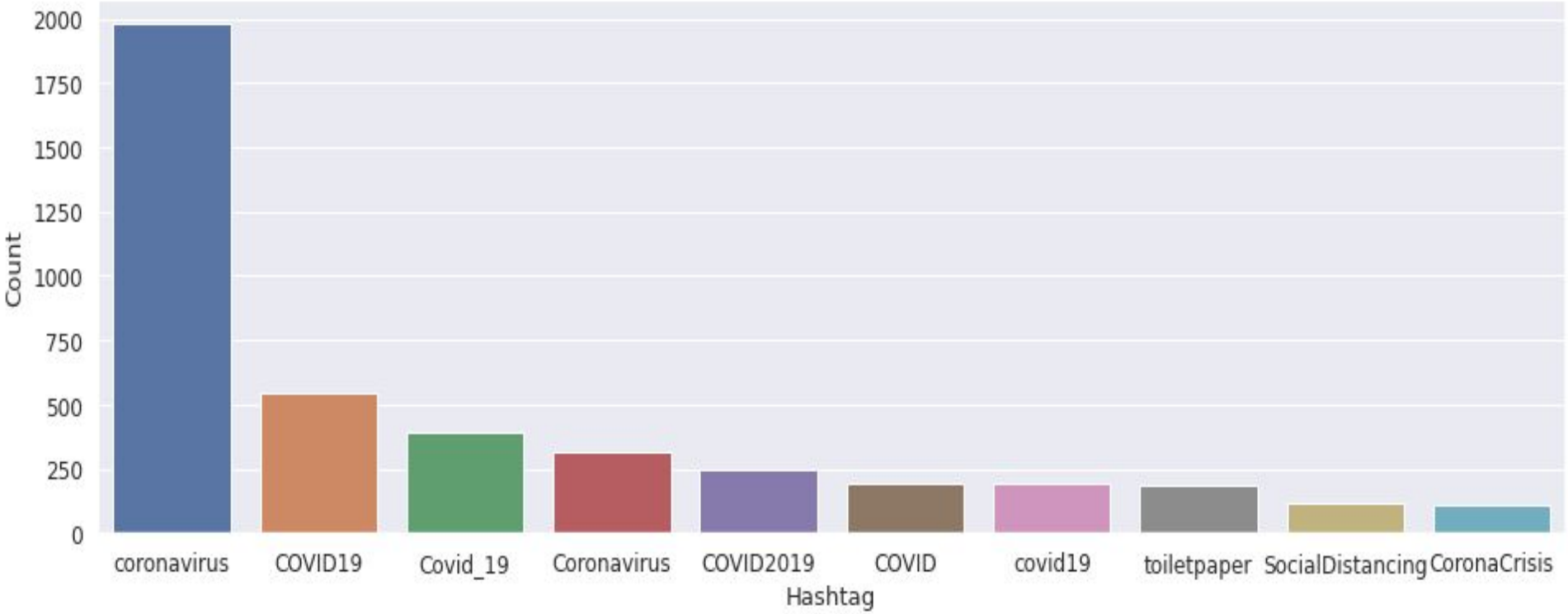
Comman Negative Words

Comman Neutral Words

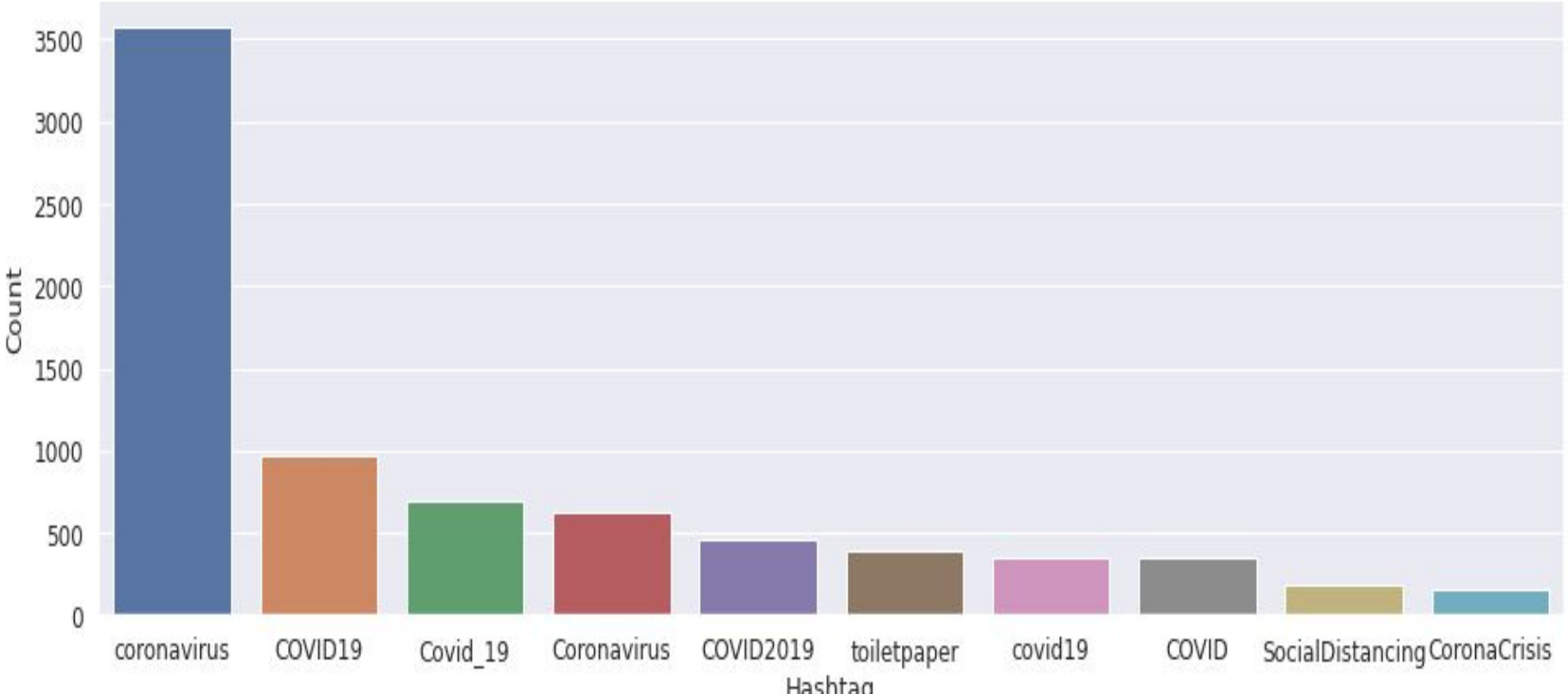


Understanding the impact of Hashtags on tweets sentiment

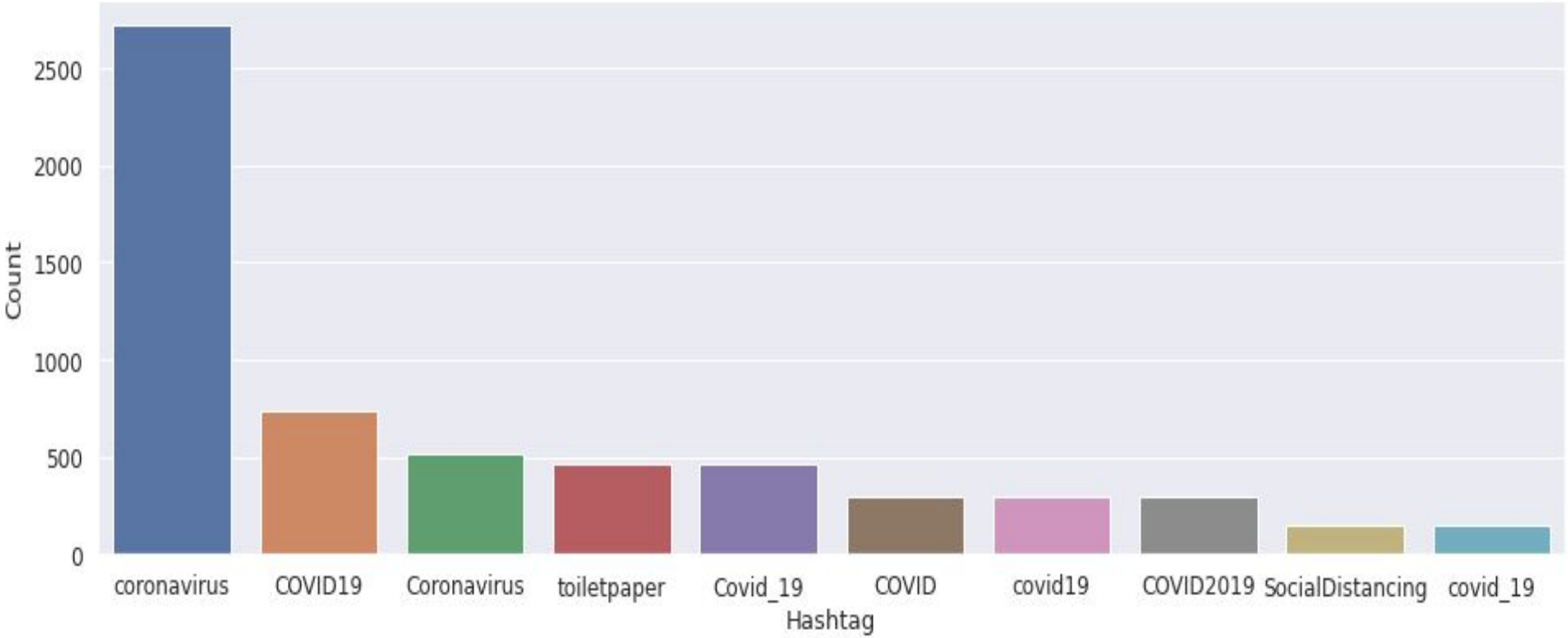
Extremely Positive



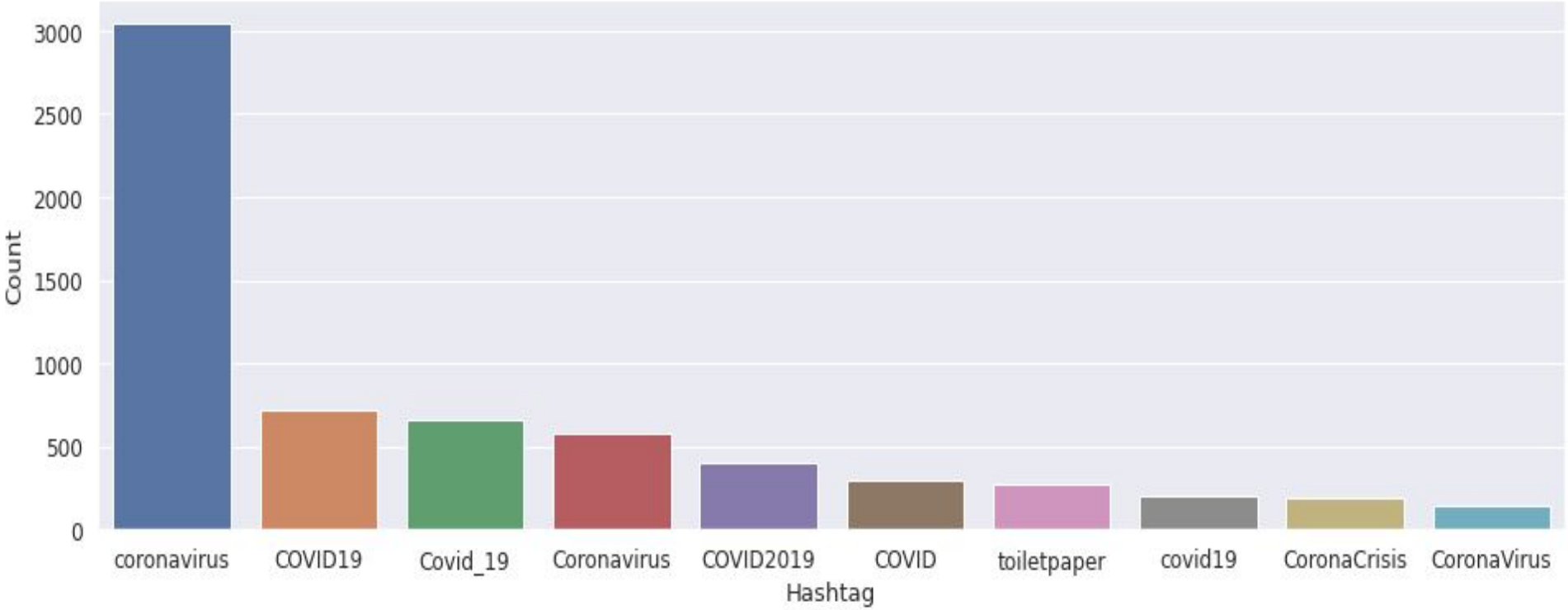
Positive


3500
3000
2500
2000
1500
1000
500
0
Count
coronavirus
COVID19
Covid_19
Coronavirus
COVID2019
toiletpaper
covid19
COVID
SocialDistancing
CoronaCrisis
Hashtag

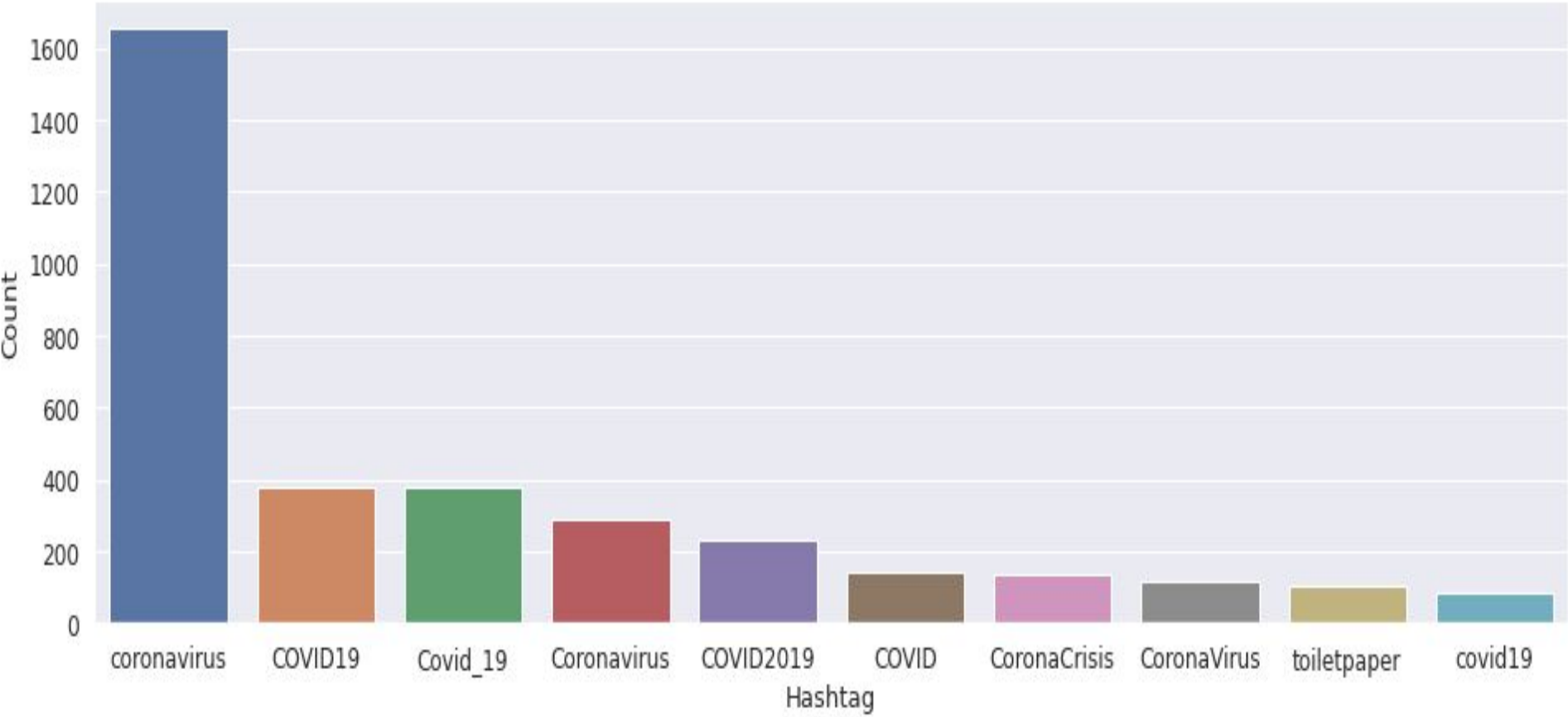
Extremely Negative



Negative


3000
2500
2000
1500
1000
500
0
Count
coronavirus
COVID19
Covid_19
Coronavirus
COVID2019
COVID
toiletpaper
covid19
CoronaCrisis
CoronaVirus
Hashtag

Neutral


1600
1400
1200
1000
800
600
400
200
0
Count
coronavirus
COVID19
Covid_19
Coronavirus
COVID2019
COVID
CoronaCrisis
CoronaVirus
toiletpaper
covid19
Hashtag

Removing Stopwords

A stop word is a commonly used word (such as “the”, “a”, “an”, “in”) that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query.

Stop words are available in abundance in any human language. By removing these words, we remove the low-level information from our text in order to give more focus to the important information.

Spitting Our Dataset into Training And Testing Dataset

We need to split a dataset into train and test sets to evaluate how well our machine learning model performs. The train set is used to fit the model, and the statistics of the train set are known. The second set is called the test data set.

Spitting Our Dataset into Training And Testing Dataset (For Multiclass Classification)

train shape : (32925, 2)
valid shape : (8232, 2)

Countvectorizer

We used Countvectorizer to transform a given text into a vector based on the frequency (count) of each word that occurs in the entire text.

Building Classification Models

There are five types of sentiments so we have to train our models so that they can give us the correct label for the test dataset. I am going to built different models like Naive Bayes, Logistic Regression, Random Forest, XGBoost, Support Vector Machines, CatBoost, and Stochastic Gradient Descent.

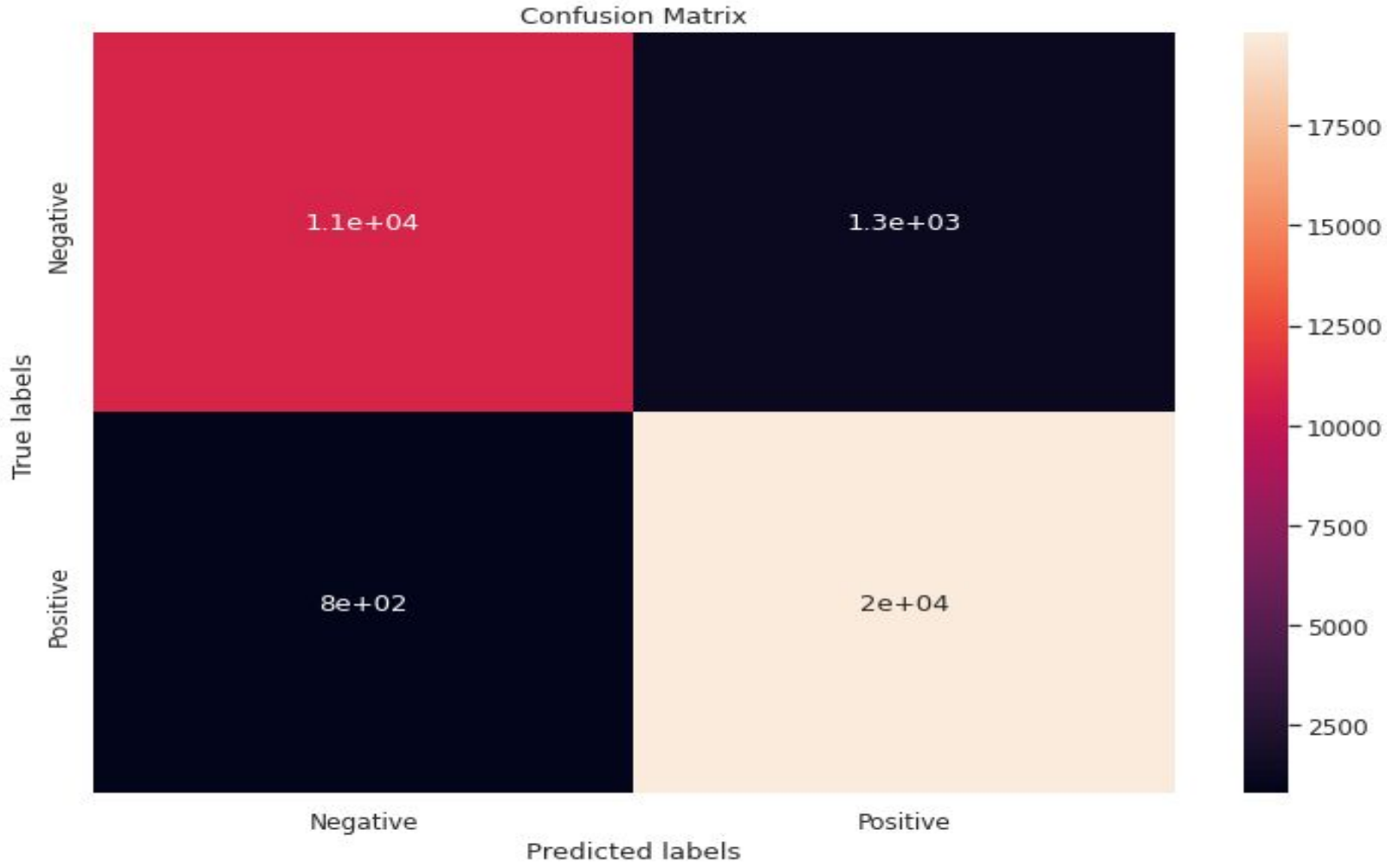
All the multiclass models test accuracy in descending order

Model	test accuracy
CatBoost-	0.620384
Logistic Regression-	0.617954
Support Vector Machines-	0.607264
Stochastic Gradient Descent -	0.572643
Random Forest-	0.560860
XGBoost-	0.486880
Naive Bayes-	0.479470

EVALUATION OF ALL BINARY CLASSIFICATION MODELS

All the model test accuracy by descending order

ModelTest	accuracy
Stochastic Gradient Descent	0.8624881
Logistic Regression	0.8594516
CatBoost	0.8507050
Support Vector Machines	0.8456032
Random Forest	0.8324833
Naive Bayes	0.7916675
XGBoost	0.739553


Confusion Matrix
1.1e+04
1.3e+03
8e+02
2e+04
Negative
Positive
True labels
Negative
Positive
Predicted labels
17500
15000
12500
10000
7500
5000
2500

Concluson

We focused on sentiment analysis for sentence labelling. We described the preprocessing steps, and pipeline steps within which text normalization and model cross-validation is included, and performance has been measured using balanced accuracy, f1 score etc. We used “Stemming” instead of Lemmatization to reduce dimensions, for the same reason we haven’t tried tf-idf or term frequency vectorizer. We concentrated on feeding our model with word count information. We assume, in the case of binary classification, we can further improve this score.

Q/A