

# Capstone Project: Coronavirus Tweet Sentiment Analysis

# Content

1. Introduction.
2. Exploratory Data Analysis.
3. Data Preprocessing.
4. Vectorization.
5. Classification.
6. Evaluation.
7. Challenges.
8. Conclusion.
9. Q&A

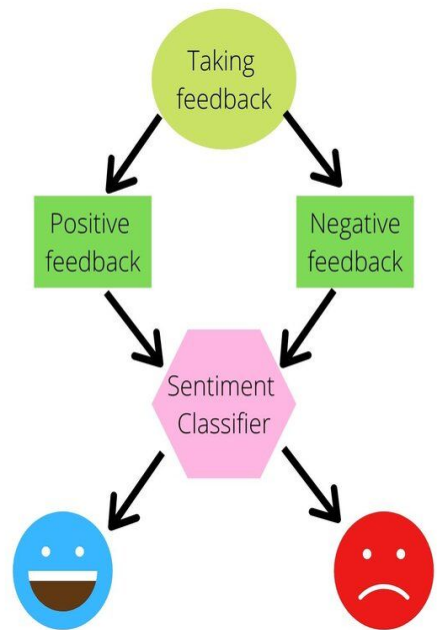


# Problem Statement

The challenge is 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.

We are given the following information:

1. Location
2. Tweet At
3. Original Tweet
4. Sentiment



# Introduction

- **Sentiment Analysis** is the process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer's attitude towards a particular topic is positive, negative, or neutral.
- **COVID-19** originally known as **Coronavirus Disease of 2019**, has been declared as a pandemic by **World Health Organization (WHO)** on **11th March 2020**.
- The study analyzes various types of tweets gathered during the pandemic times hence can be useful in policy making to safeguard the countries by demystifying the pertinent facts and information.

# Let's Guess Some Tweets: Negative, Neutral Or Positive?

- “Still shocked by the number of #Toronto supermarket employees working without some sort of mask. We all know by now, employees can be asymptomatic while spreading #coronavirus”.
- “Was at Supermarket today.Didn't buy toilet paper”.
- “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”.

# Data Summary

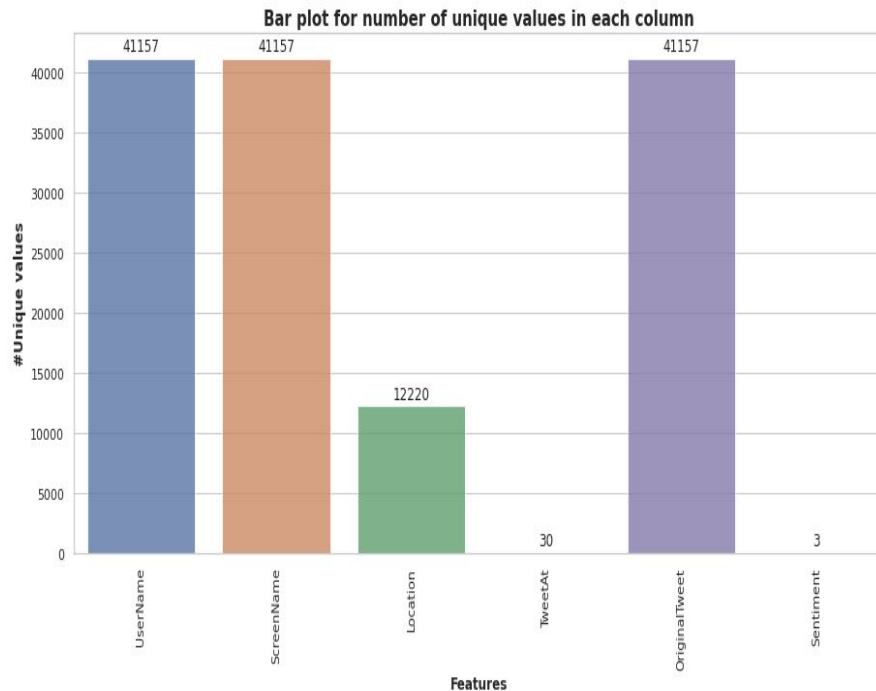
- The original dataset has 6 columns and 41157 rows.
- In order to analyse various sentiments, We require just two columns named Original Tweet and Sentiment.
- There are four types of sentiments- Extremely Negative, Negative, Neutral, Positive and Extremely Positive.

df.head()

	UserName	ScreenName	Location	TweetAt	OriginalTweet	Sentiment
0	3799	48751	London	16-03-2020	@MeNyrbie @Phil_Gahan @Chrisitv <a href="https://t.co/iFz9FAn2Pa">https://t.co/iFz9FAn2Pa</a> and <a href="https://t.co/xX6ghGFzCC">https://t.co/xX6ghGFzCC</a> and <a href="https://t.co/l2NlzdXNo8">https://t.co/l2NlzdXNo8</a>	Neutral
1	3800	48752	UK	16-03-2020	advice Talk to your neighbours family to exchange phone numbers create contact list with phone numbers of neighbours schools employer chemist GP set up online shopping accounts if poss adequate su...	Positive

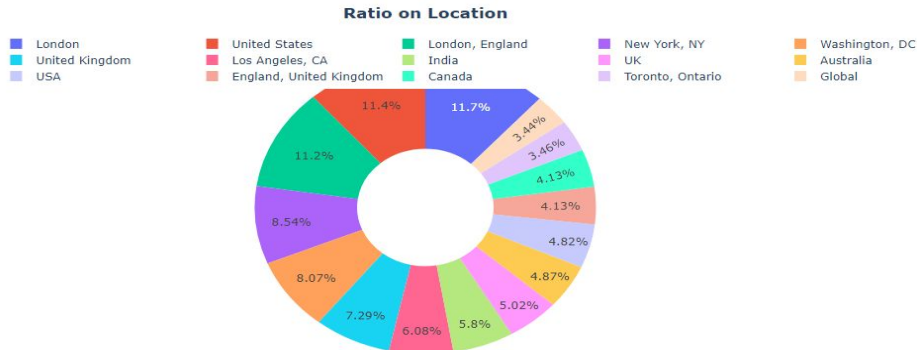
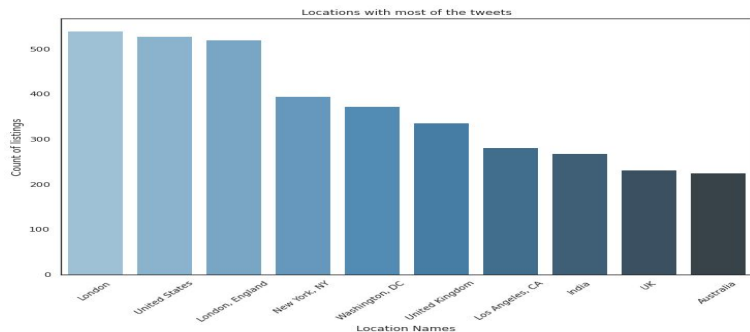
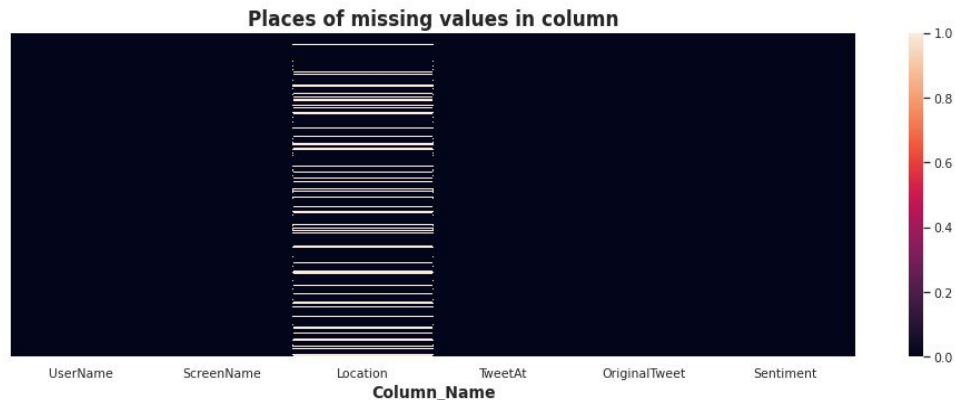
# Exploratory Data Analysis

- The columns such as “UserName” and “ScreenName” does not give any meaningful insights for our analysis.
- All tweets data collected from the months of March and April 2020.
- Bar plot shows us the number of unique values in each column.



# Exploratory Data Analysis: Location

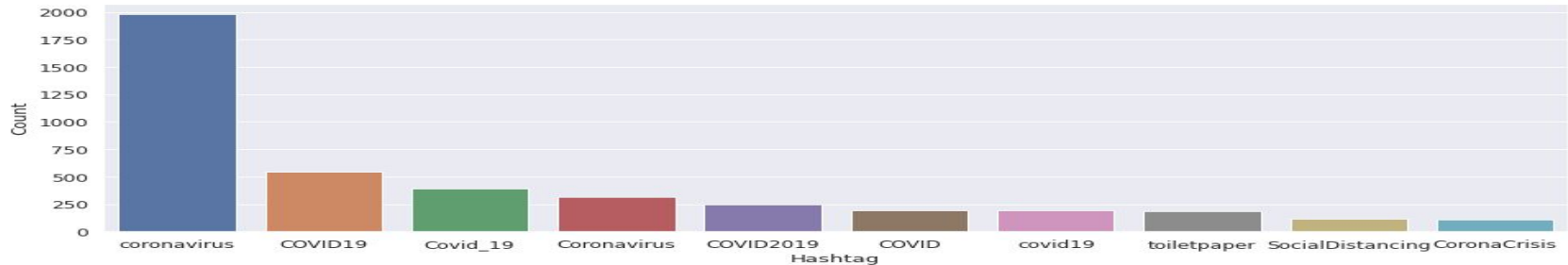
- There are 20.87%(8567) null values in various places of location column.
- Most of the tweets came from London followed by U.S.





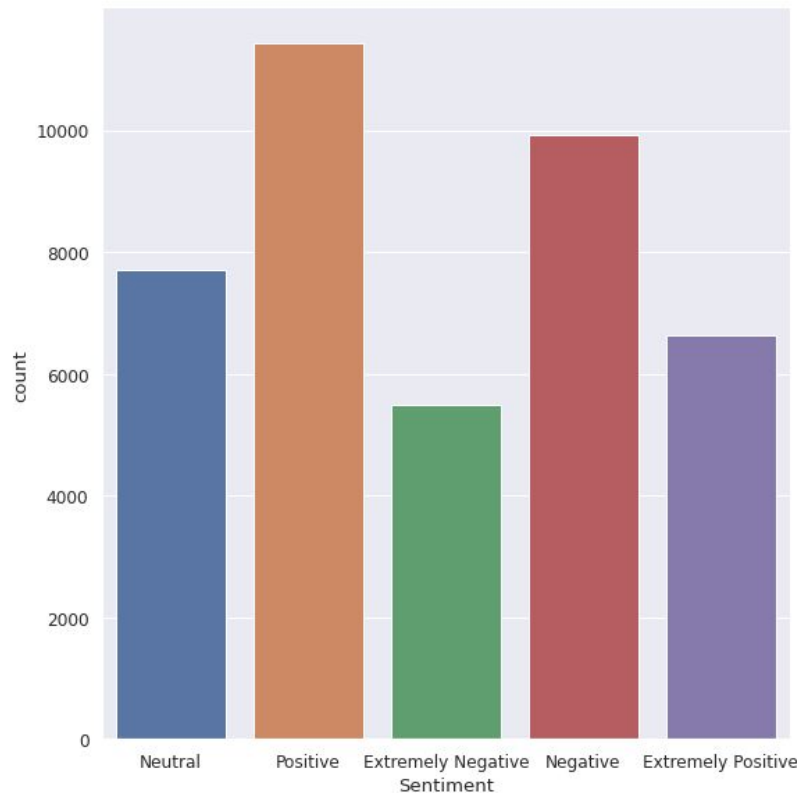
## EDA On “Original Tweet” Column.

- There are some words like 'coronavirus', 'grocery store', having the maximum frequency in our dataset.
- There are various #hashtags in tweets column. But they are almost same in all sentiments.

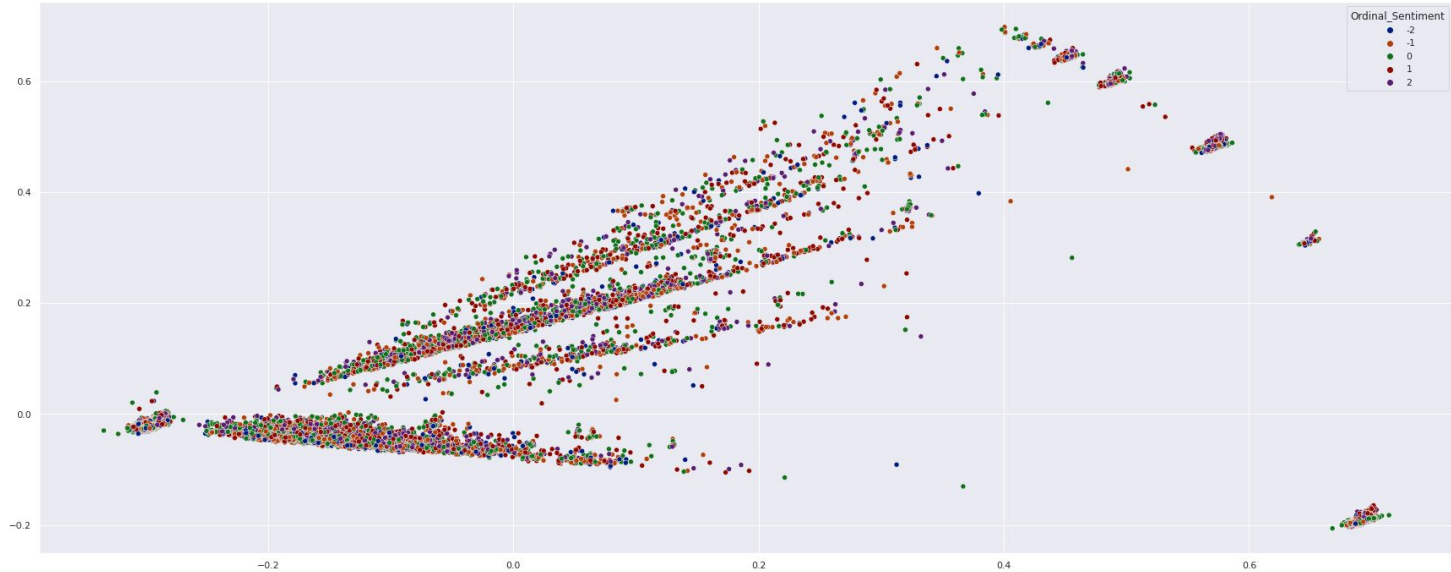


# EDA On Sentiment Column.

- Most of the peoples are having positive sentiments about various issues shows us their optimism during pandemic times.
- Very few people are having extremely negatives thoughts about Covid-19.



# Dimensionality reduction using PCA.



- We used PCA to reduce the features into two dimensions.

# Data Preprocessing

- The preprocessing of the text data is an essential step as it makes the raw text ready for mining.
- 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.

# Text Processing on Tweet

wewe



**Soraya Auer** @Soyalnk · 25 Mar 2020

...

Hi @BoConceptUK - are you raising your desk prices on purpose?? a desk for £899 is now £999, £1049 or higher - if so, this is considered profiteering and illegal #coronavirus #CoronavirusLockdown

User

HashTags

Punctuations

# Removing Tweeter Handle(@user)

As mentioned earlier, the tweets contain lots of twitter handles (@user). We will remove all these twitter handles from the data as they don't convey much information.

```
'hi - are you raising your desk prices on purpose?? a desk for â€899 is now â€999, â€1000  
own https://t.co/xwpwhwcg5x'
```

# Removing Hashtags(#)

We have analyzed that most of the tweets are like #coronavirus #covid-19 and this tweets are almost present in all the sentiments. So there is no use of keeping these hashtags in text. It will make the data noisy and which will affect accuracy of model.

**Before-**

---

```
- if so, this is considered profiteering and illegal #coronavirus #coronaviruslockd
```

**After-**

```
r - if so, this is considered profiteering and illegal https://t.co/xwpwhwcg5x'
```

# Removing links(https: / http:)

We are having twitter links in the data which are not useful for our Model. It will make our data noisy.

**Before -**

```
r - if so, this is considered profiteering and illegal https://t.co/xwpwhwcg5x'
```

**After -**

```
r - if so, this is considered profiteering and illegal'
```



# Removing Punctuations, Numbers, and Special Characters

As discussed, punctuations, numbers and special characters do not help much. It is better to remove them from the text just as we removed the twitter handles, links and hashtags.

**Before-**

```
a desk for â€899 is now â€999, â€1049 or higher - if so, this is considered profiteering and illegal'
```

**After-**

```
a desk for  is now  or higher if so this is considered profiteering and illegal'
```

# Removing Stopwords

Stop words are those words in natural language that have a very little meaning, such as "is", "an", "the", etc. To remove stop words from a sentence, you can divide your text into words and then remove the word if it exists in the list of stop words provided by NLTK.

**Before -** a desk for is now or higher if so this is considered profiteering and illegal'

**After -** 'raising desk prices purpose desk higher considered profiteering illegal'

# Stemming

- Stemming is a rule-based process of stripping the suffixes (“ing”, “ly”, “es”, “ed”, “s” etc) from a word.
- For example – “play”, “player”, “played”, “plays” and “playing” are the different variations of the word – “play”.

- Before -

'raising desk prices purpose desk higher considered profiteering illegal'

- After -

'rais desk price purpos desk higher consid profit illeg'

# Lemmatization

- Lemmatization is a more powerful operation, and it takes into consideration morphological analysis of the words. It returns the lemma which is the base form of all its inflectional forms.

- Before -

'raising desk prices purpose desk higher considered profiteering illegal'

- After -

'raise desk price purpose desk high consider profiteer illegal'

# Tokenization

- In tokenization we convert group of sentence into token . It is also called text segmentation or lexical analysis. It is basically splitting data into small chunk of words.
- Tokenization in python can be done by python NLTK library's `word_tokenize()` function.

# Vectorization

- We chose Count Vectorizer as our Vectorizer with minimum document frequency =10.
- It will create a sparse matrix of all words and the number of times they are present in a document.

# Classification

## Models Used:

1. Naive Bayes
2. Logistic Regression
3. Random Forest
4. XGBoost
5. Support Vector Machines
6. CatBoost
7. Stochastic Gradient Descent

# Naive Bayes

## Why Naive Bayes?

- Good accuracy for classification if the feature independence condition holds.
- Space and time effective.
- Can handle high dimensional data pretty well.
- A good baseline model.

### Multi class classification accuracy:

training accuracy Score : 0.6931511009870919

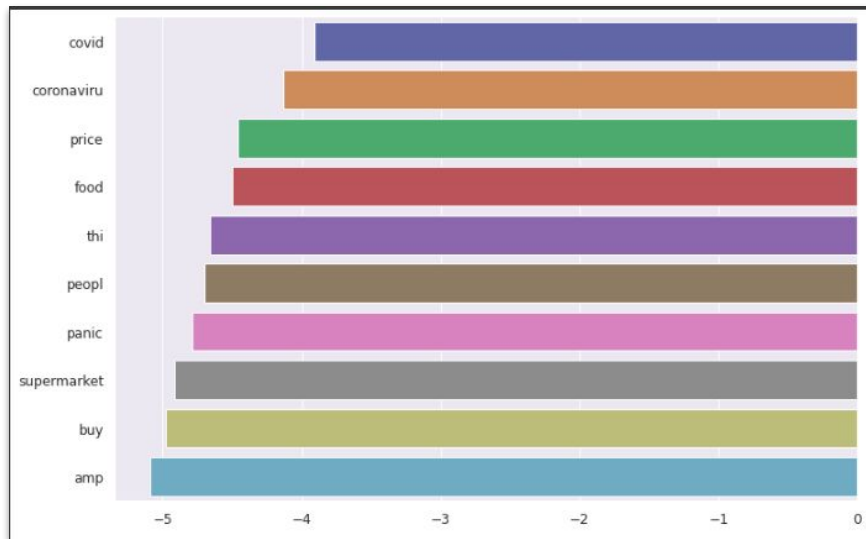
Validation accuracy Score : 0.47947035957240036



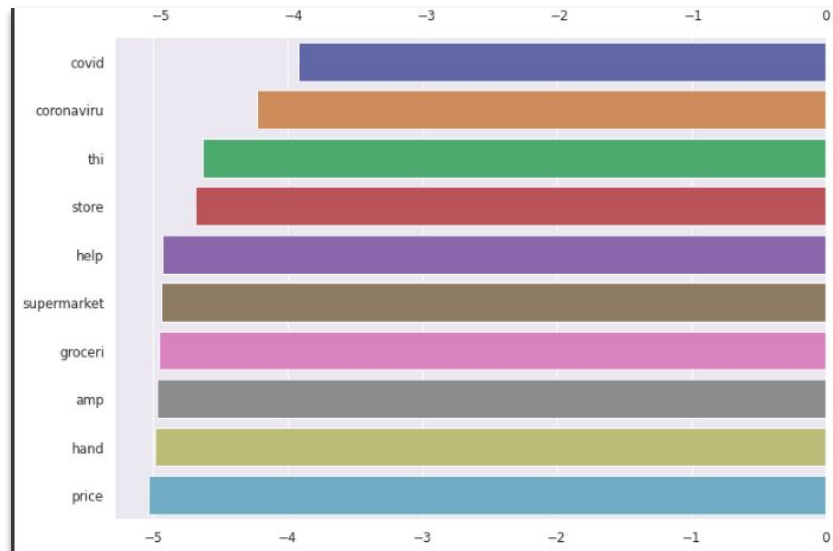
# Naive Bayes

## Feature Log Probabilities

### Neutral



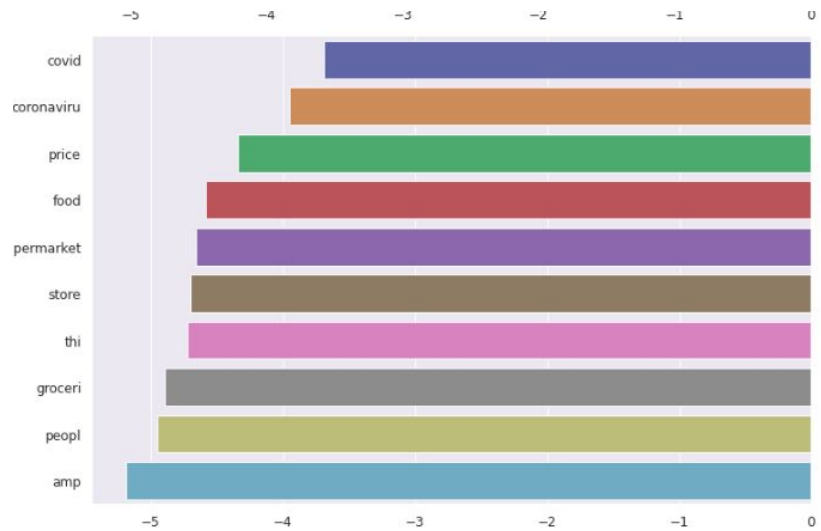
### Positive



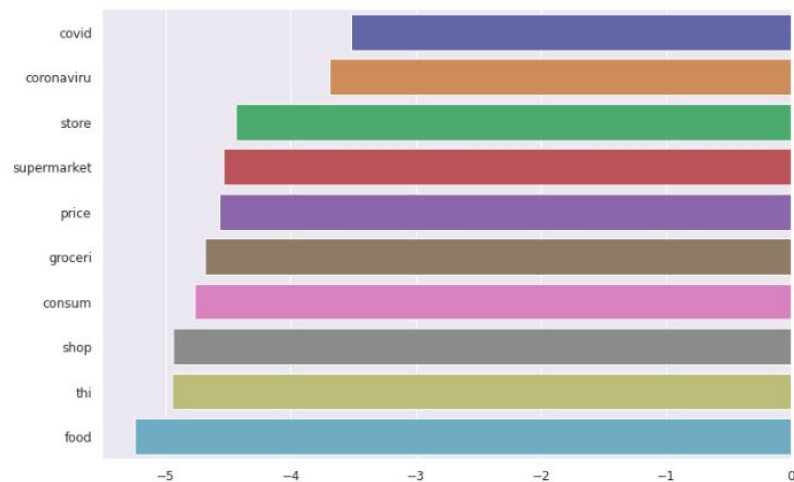
# Naive Bayes

## Feature Log Probabilities

### Extremely Negative



### Positive



# Naive Bayes

## What's the problem?

- Misclassifying samples to the similar groups because of same likelihood of words to be classified in a particular class.

## Solution:

- Binary Classification.

### Binary Classification accuracy :

training accuracy Score : 0.8585573272589218

Validation accuracy Score : 0.7916666666666666

# Logistic Regression

## Why Logistic Regression?

- Unlike Naive Bayes it makes no assumption about the feature independence.
- Logistic Regression with L1 regularization is well known for feature reduction.
- Fast to train.

## Binary Classification Accuracy:

```
Training accuracy Score      : 0.937798025816249  
Validation accuracy Score : 0.8594509232264335
```

# Random Forest

## Why Random Forest?

- Random Forest takes random samples and features to make train the model.
- Time taking, but Decision tree like model with less chance to overfit.

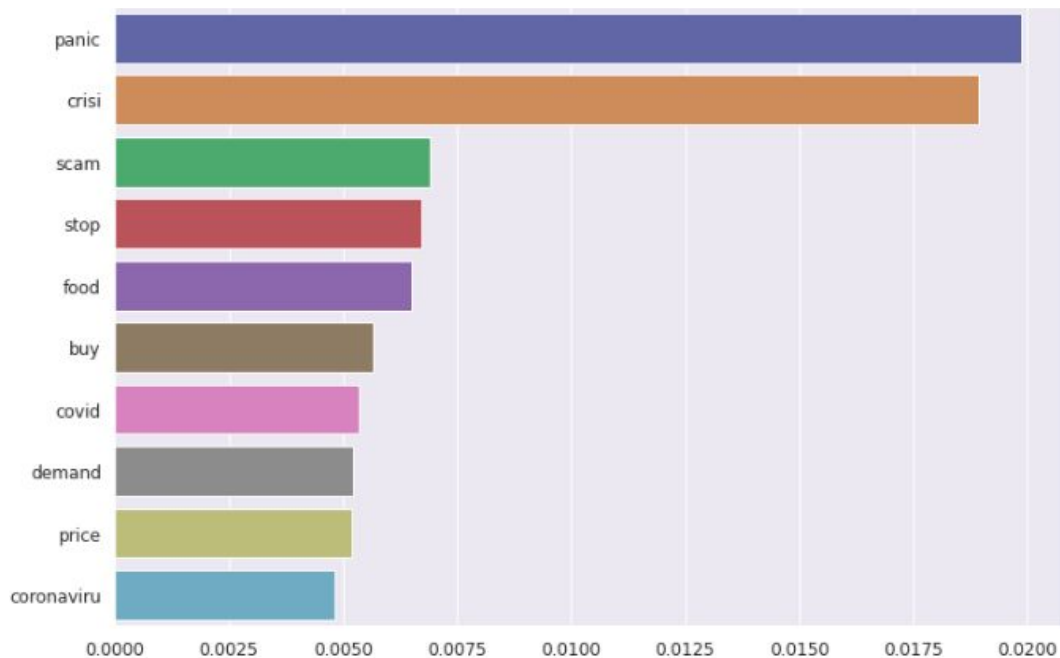
### Binary Classification accuracy:

Training accuracy Score : 0.9985725132877753

Validation accuracy Score : 0.8299319727891157

# Random Forest

## Feature Importance



# XGBoost

## Why XGB?

- Can be used with different objective functions.
- Handling missing values.
- Built in cross validation.

### Binary Accuracy Score:

Training accuracy Score : 0.7434776006074412

Validation accuracy Score : 0.7395529640427599

# Support Vector Machines

## Why Support Vector Classifier?

- It is well known to handle high dimensional data.
- It allows misclassification as well with soft margins.

### Binary Classification accuracy:

Training accuracy Score : 0.9569020501138952

Validation accuracy Score : 0.8456025267249757



# CatBoost

## Why Support Vector Classifier?

- It is good in handling sophisticated categorical features.
- Uses symmetric trees, which result in a Fast Inference.

### For multiple classes:

```
Training accuracy Score      : 0.6703720577069097  
Validation accuracy Score : 0.6203838678328474
```

### For binary classes:

```
Training accuracy Score      : 0.8840091116173121  
Validation accuracy Score : 0.8521622934888241
```

# Stochastic Gradient Descent

## Why SGD?

- It is neural network based.
- It converges comparatively faster for large datasets.
- It fits one sample at a time.
- Computationally Fast.

## Binary Classification Accuracy:

Training accuracy Score : 0.9350949126803341

Validation accuracy Score : 0.8624878522837707

# Evaluation

## Multi-class Classification

Model

Test accuracy

CatBoost

62.0%

Logistic Regression

61.8%

Support Vector Machines

60.7%

Stochastic Gradient Descent

57.3%

Random Forest

56.0%

XGBoost

48.7%

Naive Bayes

47.9%

## Binary Classification

Model

Test accuracy

Stochastic Gradient Descent

86.2%

Logistic Regression

85.9%

CatBoost

85.2%

Support Vector Machines

84.6%

Random Forest

82.9%

Naive Bayes

79.2%

XGBoost

74.0%

# Evaluation (contd.)



## Multi-class Classification Winner - CatBoost

	precision	recall	f1-score	support
Extremely Negative	0.54	0.70	0.61	843
Extremely Positive	0.56	0.76	0.65	974
Negative	0.53	0.58	0.56	1813
Neutral	0.81	0.60	0.69	2058
Positive	0.64	0.58	0.61	2544
accuracy			0.62	8232
macro avg	0.62	0.65	0.62	8232
weighted avg	0.64	0.62	0.62	8232



## Binary Classification Winner- Stochastic Grad. Descent

	precision	recall	f1-score	support
0	0.78	0.84	0.81	2882
1	0.91	0.88	0.89	5350
accuracy			0.86	8232
macro avg	0.85	0.86	0.85	8232
weighted avg	0.87	0.86	0.86	8232

# Challenges

- Locations being too many/unformatted/irrelevant
- Sarcastic tweets
- Advertisements tagged as positive
- Computation time/crashes

# Conclusion

- For multiclass classification, the best model for this dataset would be CatBoost
- For binary classification, the best model for this dataset would be Stochastic Gradient Descent

**To end it on a lighter note, a few funny tweets we came across:**

I used to spin toilet paper like I was spinning the wheel on Wheel of Fortune! Now when I use it I look like I'm cracking a safe!

People posting and sharing photos of half to completely empty shelves calling those people "dumb" or "idiots."  
All while shopping at the grocery store. |

I just want a COVID-19 online shopping sale . F\*\*\* your concerns . Give me the promo codes  
The #coronavirus can now give you diarrhea. Toilet paper hoarding is now officially worth it

If it wasn't for social distancing, I'd have hugged the grocery store worker. We found toilet paper at #Marianos!!

# Q & A