


# Machine Learning presentation



# Team members:

Name:

ID:

Al Fayad Arnob

170042075

Rokeya Samanta Ruhee

170042064

Nowshadul Islam Nishad

170042066

Topic :

# **Twitter sentiment Extraction-Analysis, EDA and Model**

# Motivation

It is hard to tell whether the sentiment behind a specific tweet will impact a company or a person, brand to get viral, or to devastate a profit because it strikes a negative tone. Capturing sentiment in language is important in these times where decisions and reactions are created and updated in seconds.

# Challenges faced

1. Finding proper training Data Set with 3 types of sentiment classification
2. Inaccurate sentiment analysis data can prove catastrophic
3. Determining which models to use
4. Determining what to include and how to perform the EDA

# Existing other solution

page:6

## **1. NCSU Tweet Sentiment Visualization App**

NCSU Tweet Sentiment Visualization App is a cloud-based tool that allows users to perform sentiment analysis of Twitter posts based on keyword mentions.

## **2. Mention**

Mention is a cloud-based social media monitoring platform for businesses of all sizes.

## **3. Social Searcher**

Social Searcher is a cloud-based social media search engine for businesses of all sizes.

# Proposed Solution

page:7

A machine learning model which can

- Analyze the sentiment of the tweets
- Detect if it is positive or negative or neutral

We will be finding not the sentiment scores but the part of the tweet (word or phrase) that reflects the sentiment.

# Data Set

Collected the tweet data set from multiple datasources by tanulsingh077

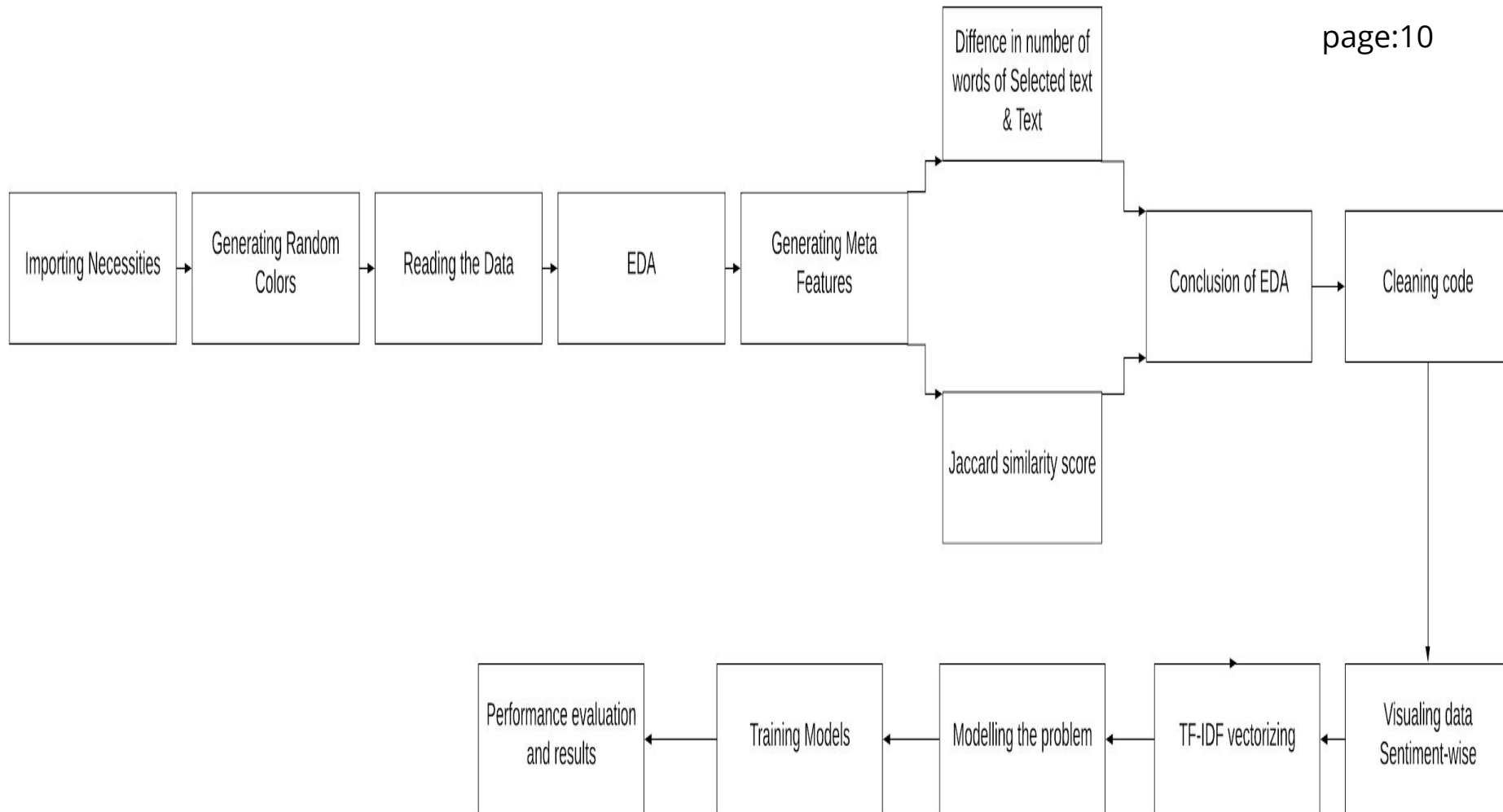
Columns in training dataset

- textID - unique ID for each piece of text
- text - the text of the tweet
- sentiment - the general sentiment of the tweet
- selected\_text - [train only] the text that supports the tweet's sentiment

We have **27486 tweets** in the train set and **3535 tweets** in the test set



# Experimental Setup



# Description

# EDA (exploratory data analysis)

## What do we currently Know :

We Know that selected\_text is a subset of text

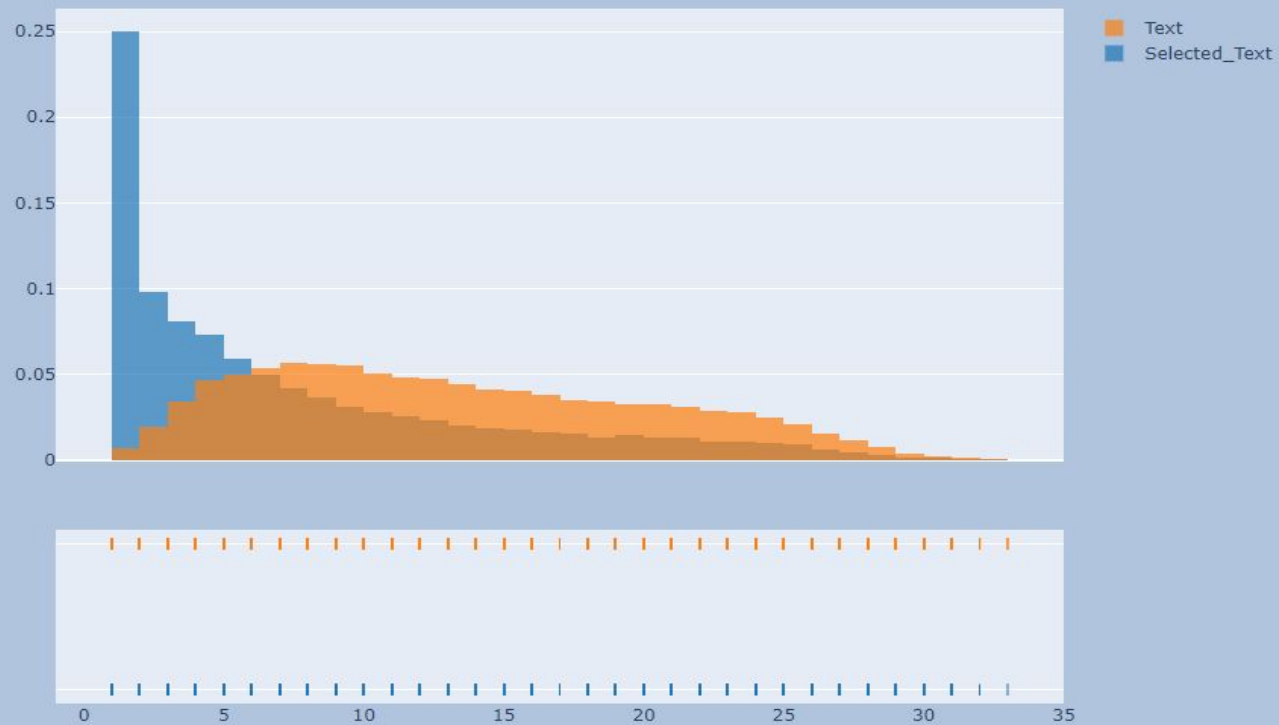
Selected\_text starts from between the words and thus selected\_texts don't always make sense

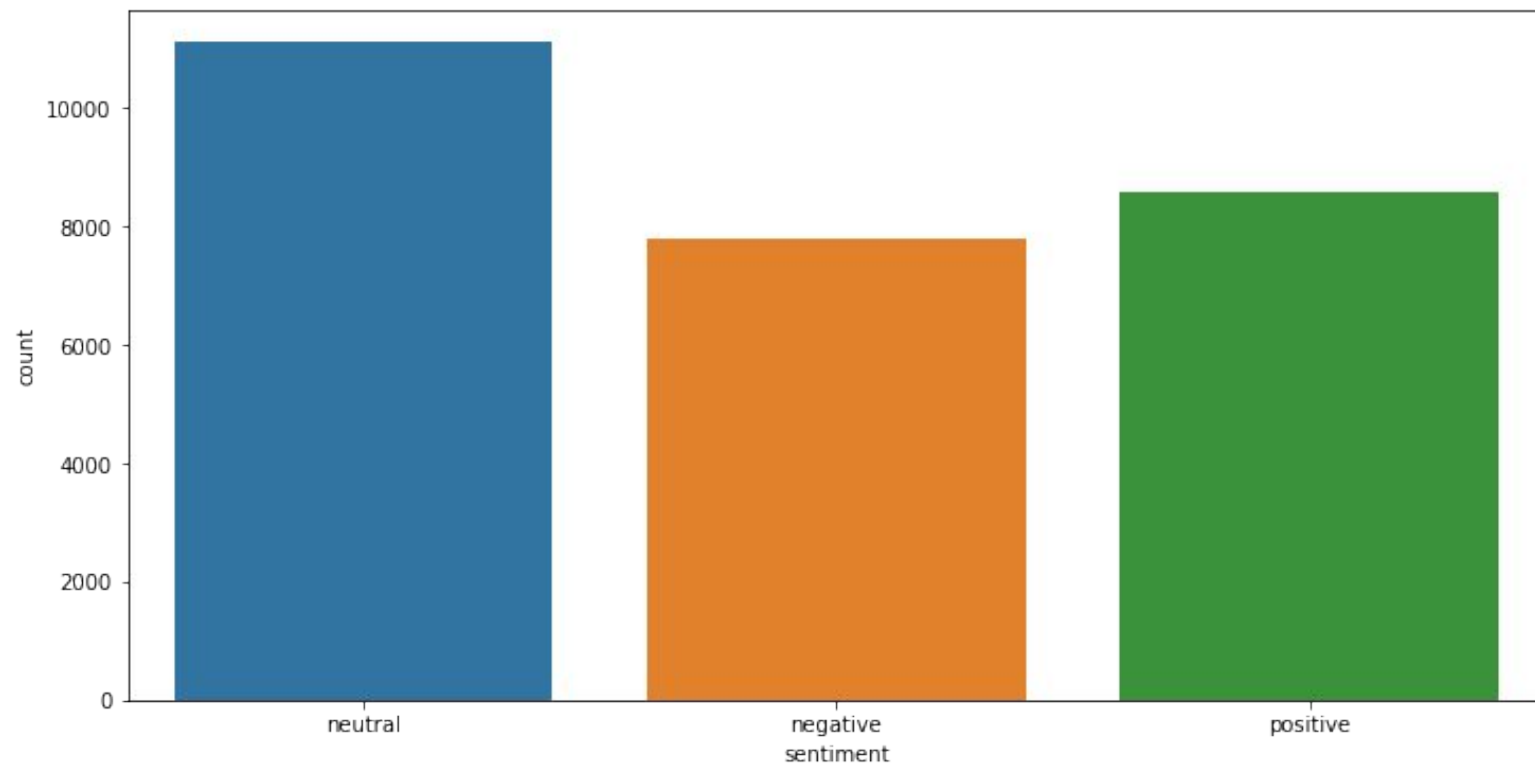
**Generating these meta features - Number of words in selected text and main text, Length of words in text and selected text would not be useful**

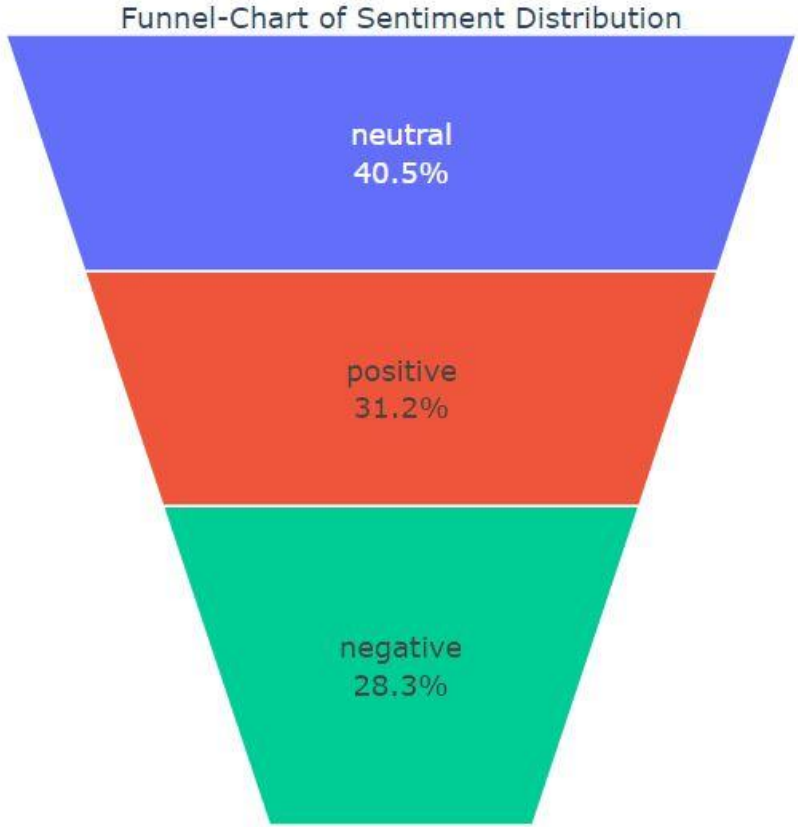
in the context of this presentation where we have to predict selected\_text which is a subset of text, more useful features to generate would be :-

- **Difference In Number Of words of Selected\_text and Text**
- **Jaccard Similarity Scores between text and Selected\_text**

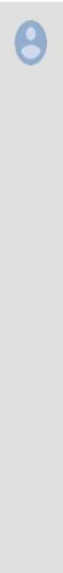
Distribution of Number Of words





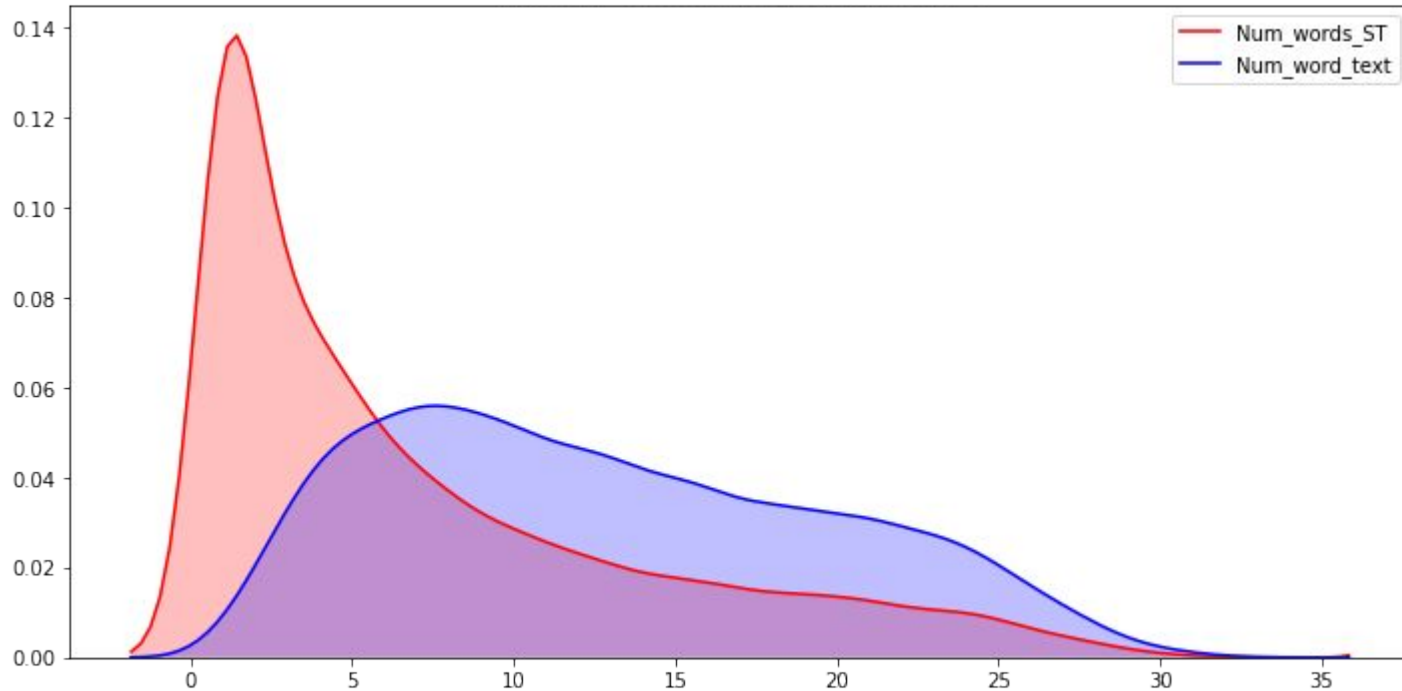


# Jaccard similarity scores

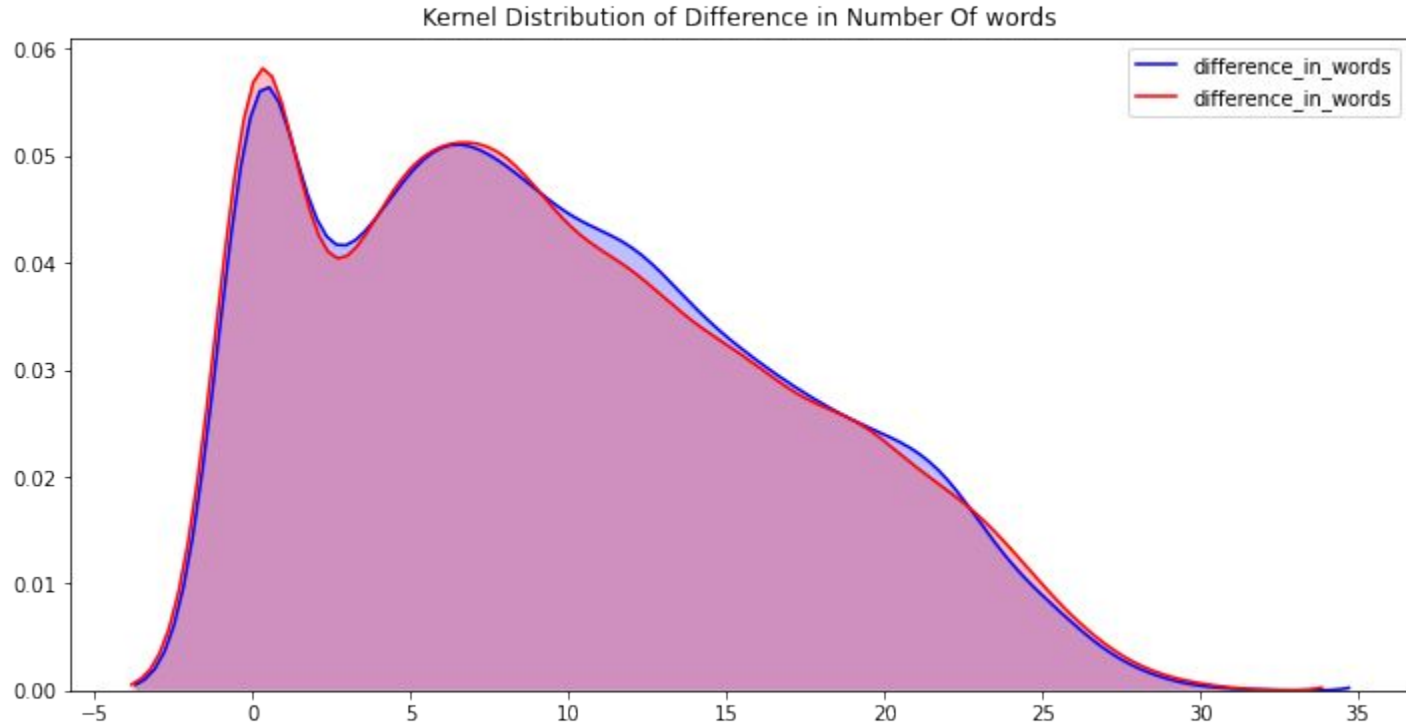


	textID	text	selected_text	sentiment	jaccard_score	Num_words_ST	Num_word_text	difference_in_words
0	cb774db0d1	I'd have responded, if I were going	I'd have responded, if I were going	neutral	1.000000	7	7	0
1	549e992a42	Sooo SAD I will miss you here in San Diego!!!	Sooo SAD	negative	0.200000	2	10	8
2	088c60f138	my boss is bullying me...	bullying me	negative	0.166667	2	5	3
3	9642c003ef	what interview! leave me alone	leave me alone	negative	0.600000	3	5	2
4	358bd9e861	Sons of ****, why couldn't they put them on t...	Sons of ****,	negative	0.214286	3	14	11

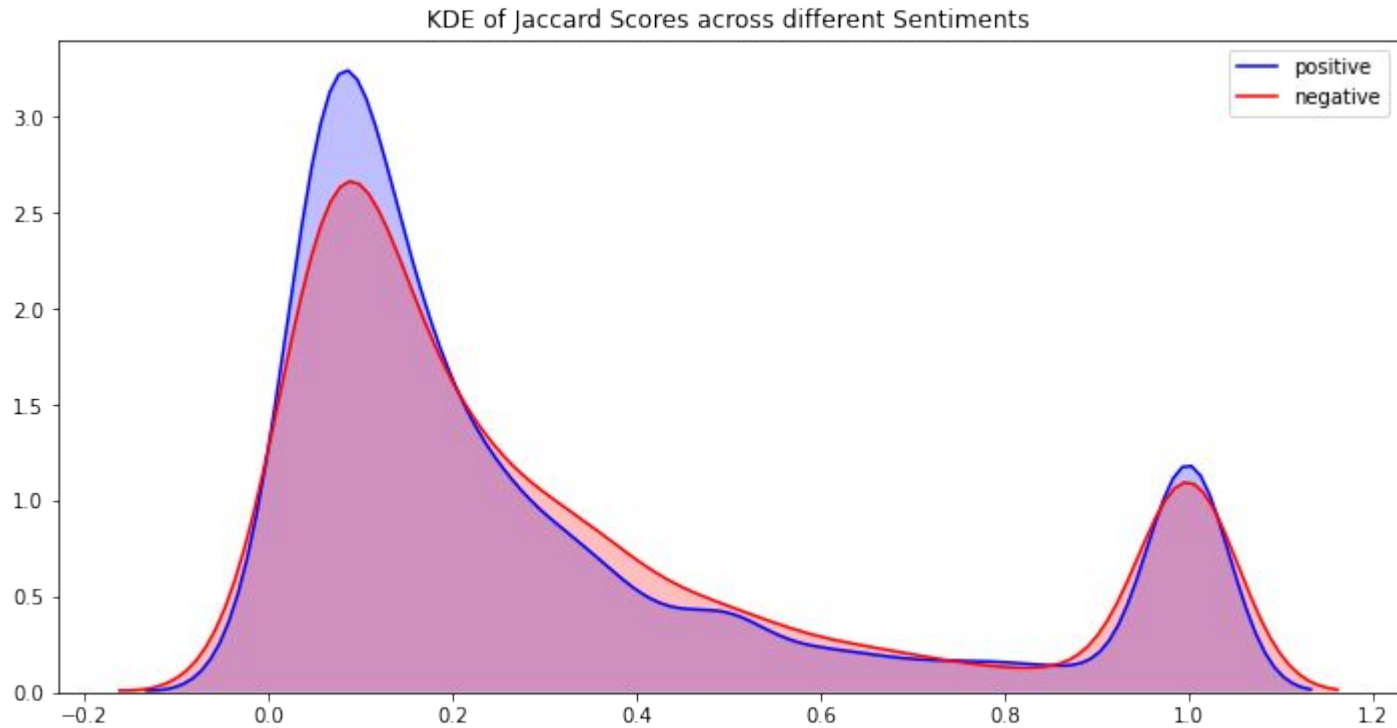




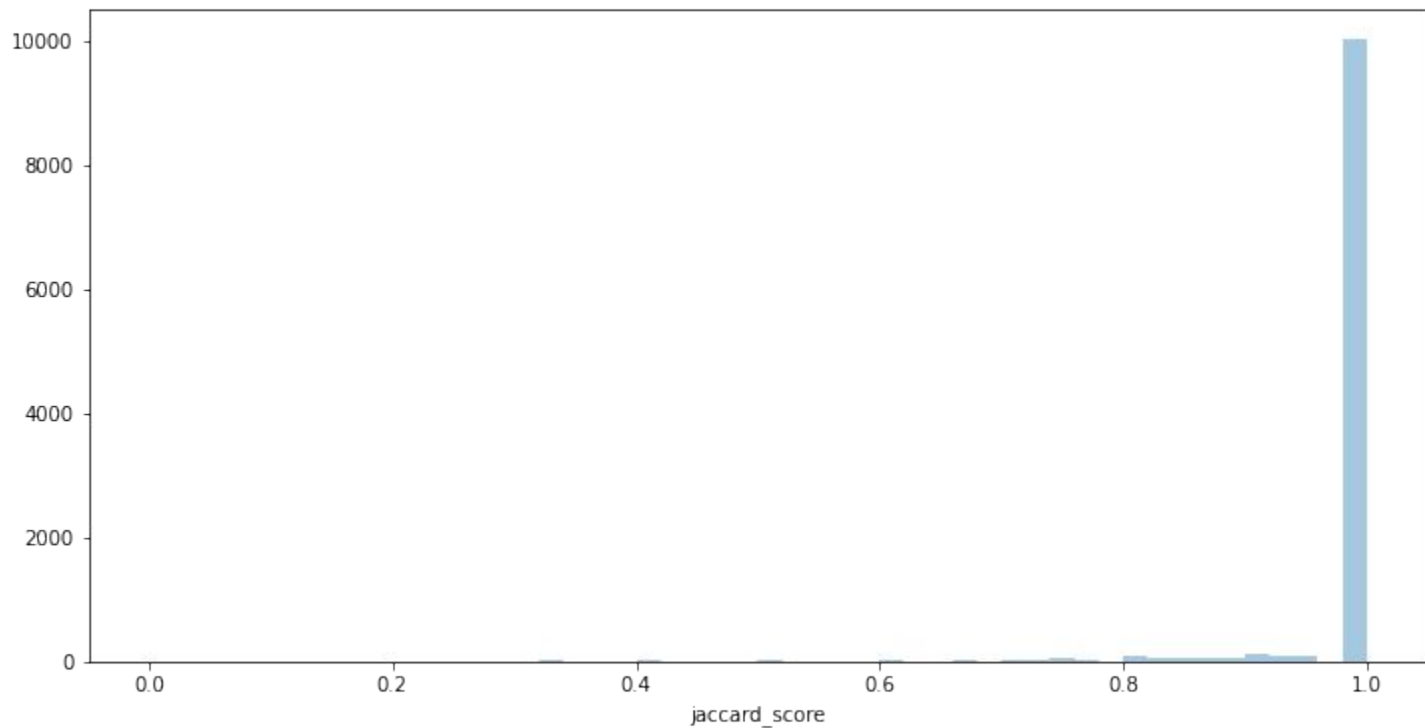
The number of words plot is really interesting ,the tweets having number of words greater than 25 are very less and thus the number of words distribution plot is right skewed



I was not able to plot kde plot for neutral tweets because most of the values for difference in number of words were zero. Red = neg , Blue = Pos



there is a peak for negative and positive plot around score of 1



Distribution plot for neutral tweets



	textID	text	selected_text	sentiment	jaccard_score	Num_words_ST	Num_word_text	difference_in_words
68	fa2654e730	Chilliin	Chilliin	positive	1.0	1	1	0
80	bbbc46889b	THANK YYYYYYYYYO000000000UUUUU!	THANK YYYYYYYYYO000000000UUUUU!	positive	1.0	2	2	0
170	f3d95b57b1	good morning	good morning	positive	1.0	2	2	0
278	89d5b3f0b5	Thanks	Thanks	positive	1.0	1	1	0
429	a78ef3e0d0	Goodmorning	Goodmorning	positive	1.0	1	1	0
...	...	...	...	...	...	...	...	...
26689	e80c242d6a	Goodnight;	Goodnight;	positive	1.0	1	1	0
26725	aad244f37d	*hug*	*hug*	positive	1.0	1	1	0
26842	a46571fe12	congrats!	congrats!	positive	1.0	1	1	0
26959	49a942e9b1	Happy birthday.	Happy birthday.	positive	1.0	2	2	0
27292	47c474aaf1	Good choice	Good	positive	0.5	1	2	1

207 rows × 8 columns

text is used as selected text mostly when word length is less than 3

# Conclusion of EDA

We can see some interesting trends here:

- **Positive and negative** tweets have **high kurtosis** and thus values are **concentrated in two regions narrow and high density**
- **Neutral tweets** have a **low kurtosis** value and their is bump **in density near values of 1**
- We can see from the **jaccard score plot** that there is **peak for negative and positive plot around score of 1** .That means there is a **cluster of tweets** where there is a **high similarity** between text and selected texts ,if we can find those clusters then we can predict text for selected texts for those tweets irrespective of segment
- Thus its clear that most of the times , **text is used as selected text**.We can **improve this by preprocessing** the text which have word length less than 3

# Cleaning the data

- Make text lowercase.
- Remove text in square brackets.
- Remove links.
- Remove punctuation.
- Remove words containing numbers.
- Replacing same letters in sequence more than twice into just double letters.
- Lemmatizing the words.

# Most common words sentiment - wise



## Most Common Positive words

## Most Common Negative words





# Modeling and Training

We're creating 3 different types of model for our sentiment analysis problem:

- **Bernoulli Naive Bayes (BernoulliNB)**
- **Linear Support Vector Classification (LinearSVC)**
- **Logistic Regression (LR)**

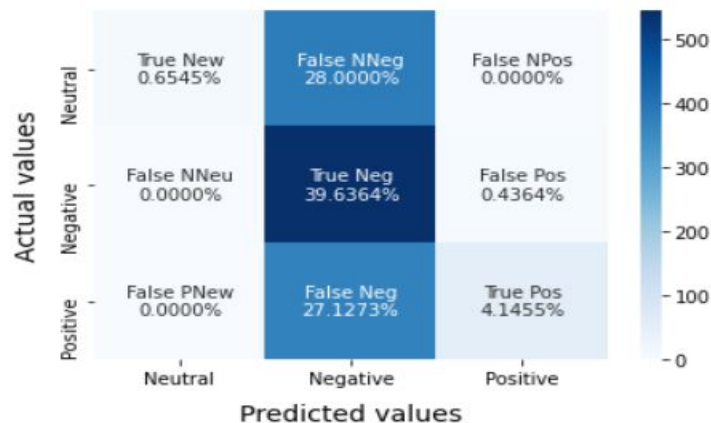
Since our dataset is not **skewed**, i.e. it has equal number of **Positive and Negative** Predictions. We're choosing **Accuracy** as our evaluation metric. Furthermore, we're plotting the **Confusion Matrix** to get an understanding of how our model is performing on both classification types

# Performance Evaluation and Results

# NB Bernoulli Model

	precision	recall	f1-score	support
negative	1.00	0.02	0.04	394
neutral	0.42	0.99	0.59	551
positive	0.90	0.13	0.23	430
accuracy			0.44	1375
macro avg	0.77	0.38	0.29	1375
weighted avg	0.74	0.44	0.32	1375

Confusion Matrix

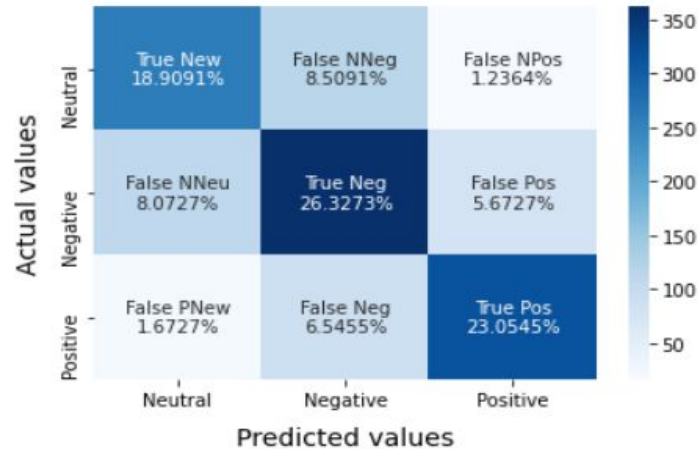


# Linear SVC Model




	precision	recall	f1-score	support
negative	0.66	0.66	0.66	394
neutral	0.64	0.66	0.65	551
positive	0.77	0.74	0.75	430
accuracy			0.68	1375
macro avg	0.69	0.68	0.69	1375
weighted avg	0.68	0.68	0.68	1375

Confusion Matrix



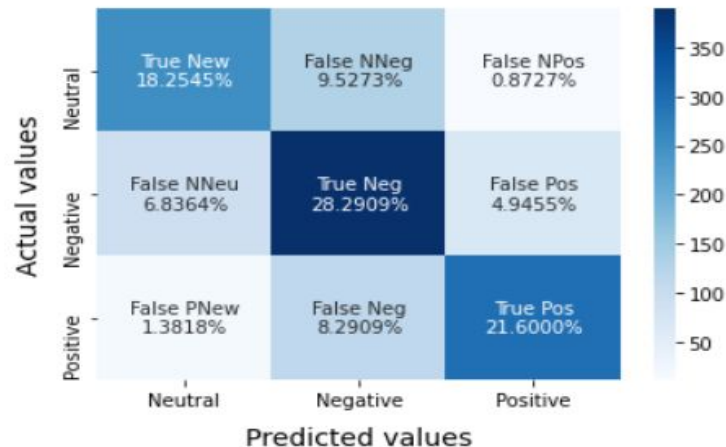


# Logistic Regression Model



	precision	recall	f1-score	support
negative	0.69	0.64	0.66	394
neutral	0.61	0.71	0.66	551
positive	0.79	0.69	0.74	430
accuracy			0.68	1375
macro avg	0.70	0.68	0.68	1375
weighted avg	0.69	0.68	0.68	1375

Confusion Matrix



# Conclusion

We can clearly see that the **Logistic Regression Model** performs the best out of all the different models that we tried. It achieves nearly **68% accuracy** while classifying the sentiment of a tweet. Also it had a **higher recall and f1 score**.

Hence we used this model to predict the sentences.

Although it should also be noted that the **BernoulliNB Model** is the fastest to train and predict on. But It achieves **44% accuracy** while classifying and also showed the **lowest recall and f1 score** for **negative and positive**.

However, as these models are to be trained with larger datasets, there were no dataset found with 3 types of sentiment classification, there is a scope in future to train our models with larger dataset and reduce underfitting and improve accuracy.

**We are here facing High Bias and Low Variance**

To tackle this we **need more data** and a **good balance** between data and features

# Future Improvements

We're using **PICKLE** to save **Vectoriser and BernoulliNB, Logistic Regression Model** for later use. To use the model for **Sentiment Prediction** we need to import the **Vectoriser** and **LR Model** using **Pickle**.

Thus we can **use larger data sets** in the future and see if we can improve the accuracies in these 3 models

**Also , we can try different algorithms and models such as Bert and Spacy to improve the confusion matrix and performance**



# *Thank you!*

[https://github.com/SamantaRuhee/Machine-Learning?fbclid=IwAR2f\\_QfcUT2V5UHOH7K-N8qnezWwT9WsONYwFnjuPdJ88vYhnA7hMaMKiL0](https://github.com/SamantaRuhee/Machine-Learning?fbclid=IwAR2f_QfcUT2V5UHOH7K-N8qnezWwT9WsONYwFnjuPdJ88vYhnA7hMaMKiL0)

## Lets move on to the demonstration

