Tweet Sentiment Analysis

SC1015 - Introduction to Data Science and Artificial Intelligence

Lab A127, Team 2
Tan Hee (U2220857H)
Lim Zheng Guang (U2221246G)
Kerwin Soon (U2223521J)

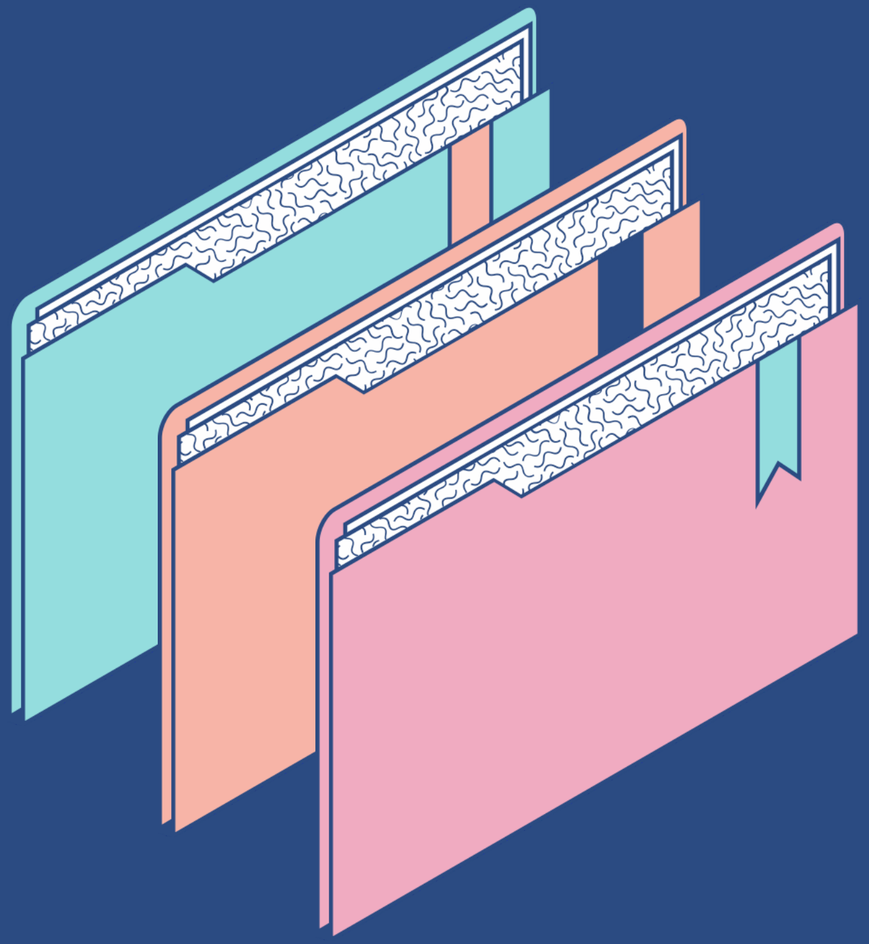
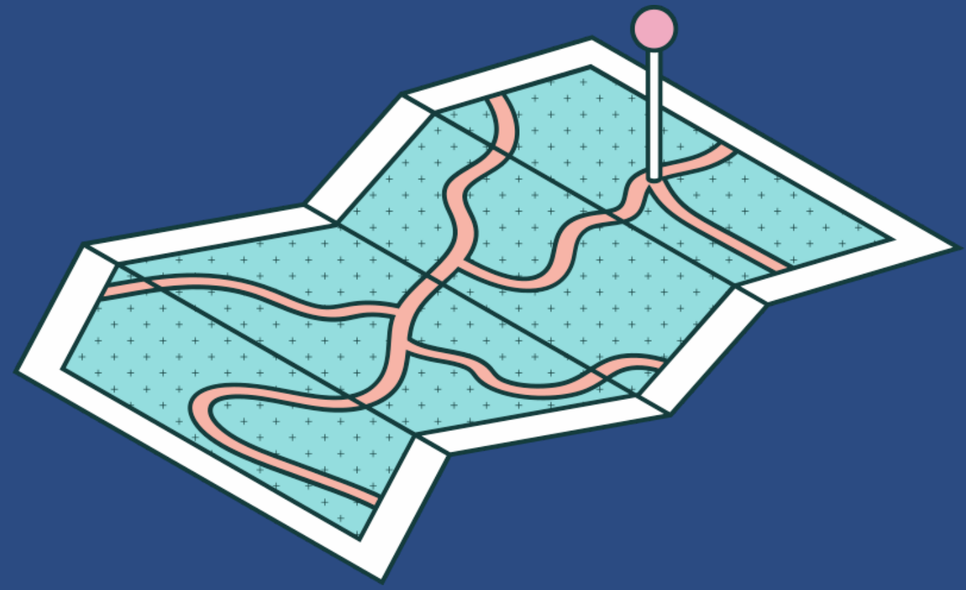


Table of Contents

1

Motivation and Problem Statement

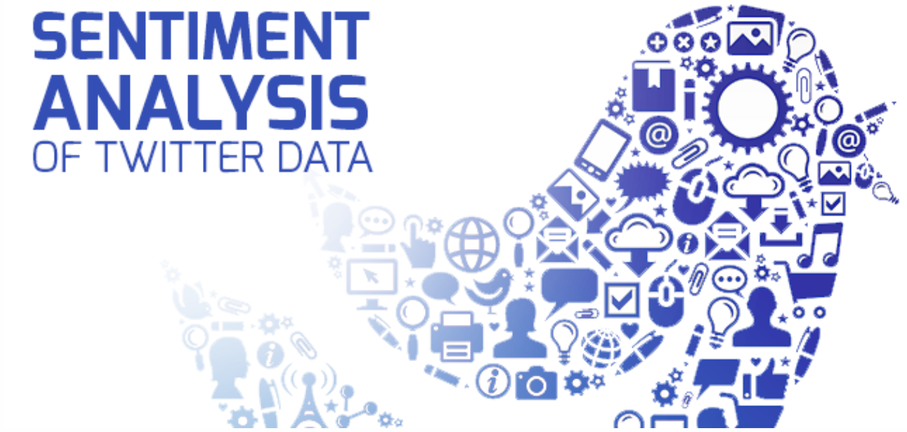


Our Motivation

We want to gain insights into people's daily lives by analyzing their tweets on the platform - whether they are being positive, negative or neutral based on the text they are tweeting.

Dataset

kaggle



Included in the dataset are 4 columns:

- textId `unique ID for each piece of text`
- text `the text of the tweet`
- selected text `the general sentiment of the tweet`
- sentiment `the text that supports the tweet's sentiment`

Dataset

In [4]:
```
df  = pd.read_csv("dataset/train.csv")
df.head()
```

Out[4]:

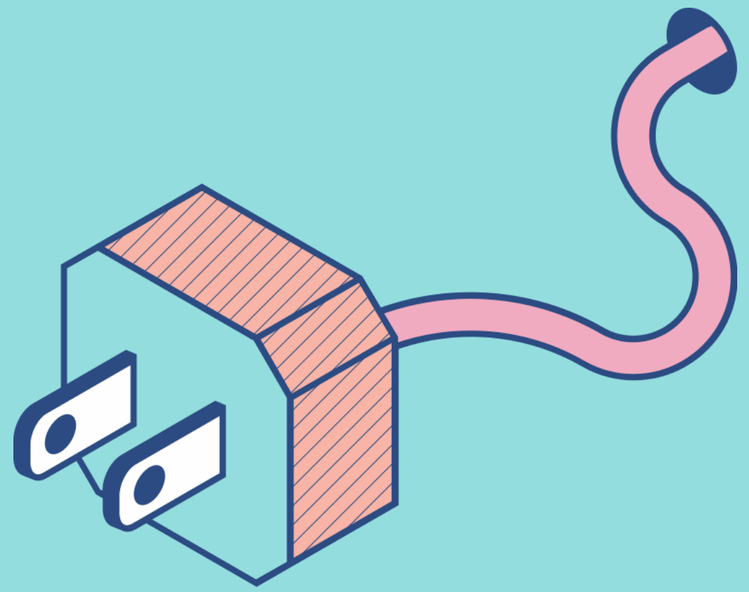
	textID	text	selected_text	sentiment
0	cb774db0d1	I'd have responded, if I were going	I'd have responded, if I were going	neutral
1	549e992a42	Sooo SAD I will miss you here in San Diego!!!	Sooo SAD	negative
2	088c60f138	my boss is bullying me...	bullying me	negative
3	9642c003ef	what interview! leave me alone	leave me alone	negative
4	358bd9e861	Sons of ****, why couldn`t they put them on t...	Sons of ****,	negative

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27481 entries, 0 to 27480
```

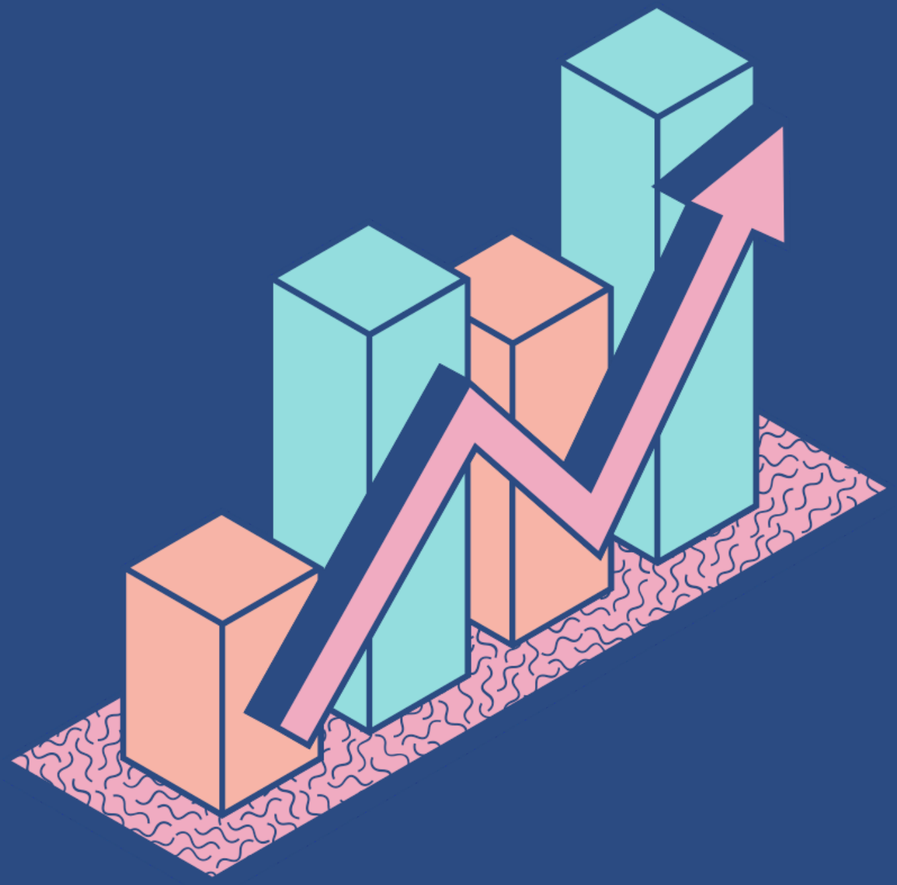
Problem Statement

To accurately predict and classify the sentiment of the tweet as positive, negative or neutral using different models



2

Exploratory Data Analysis

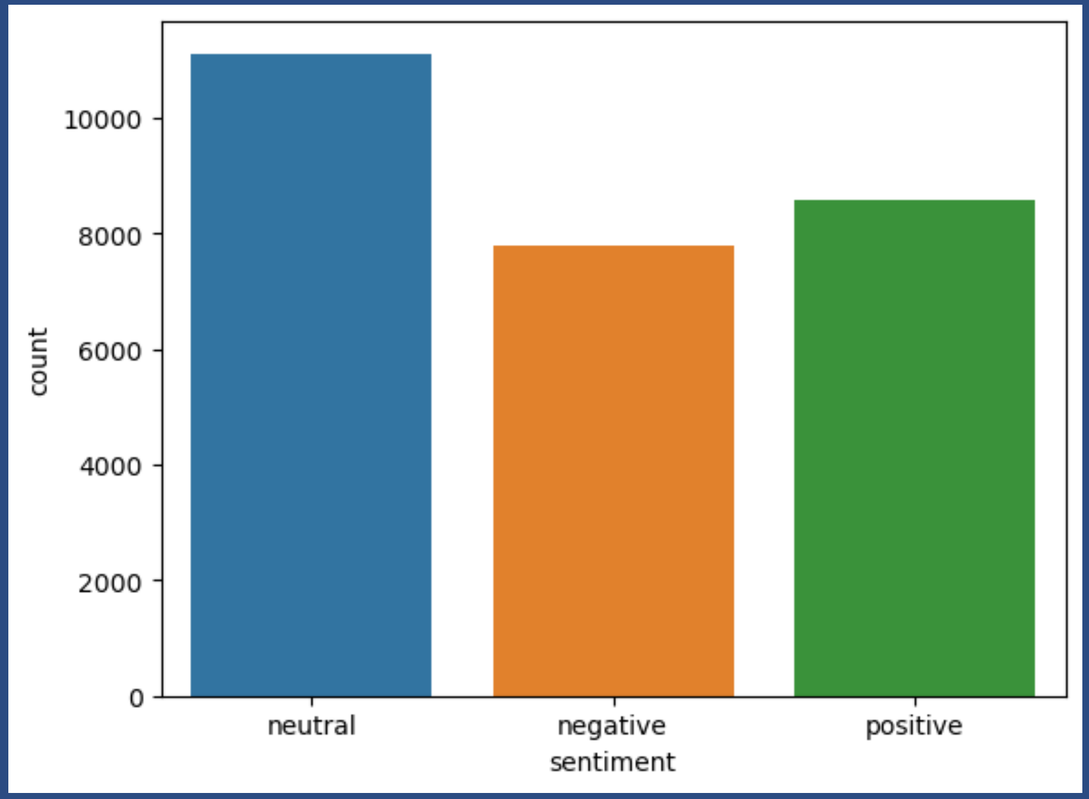


Exploratory Analysis & Analytic Visualization

Seaborn count plot

3 types of sentiment:

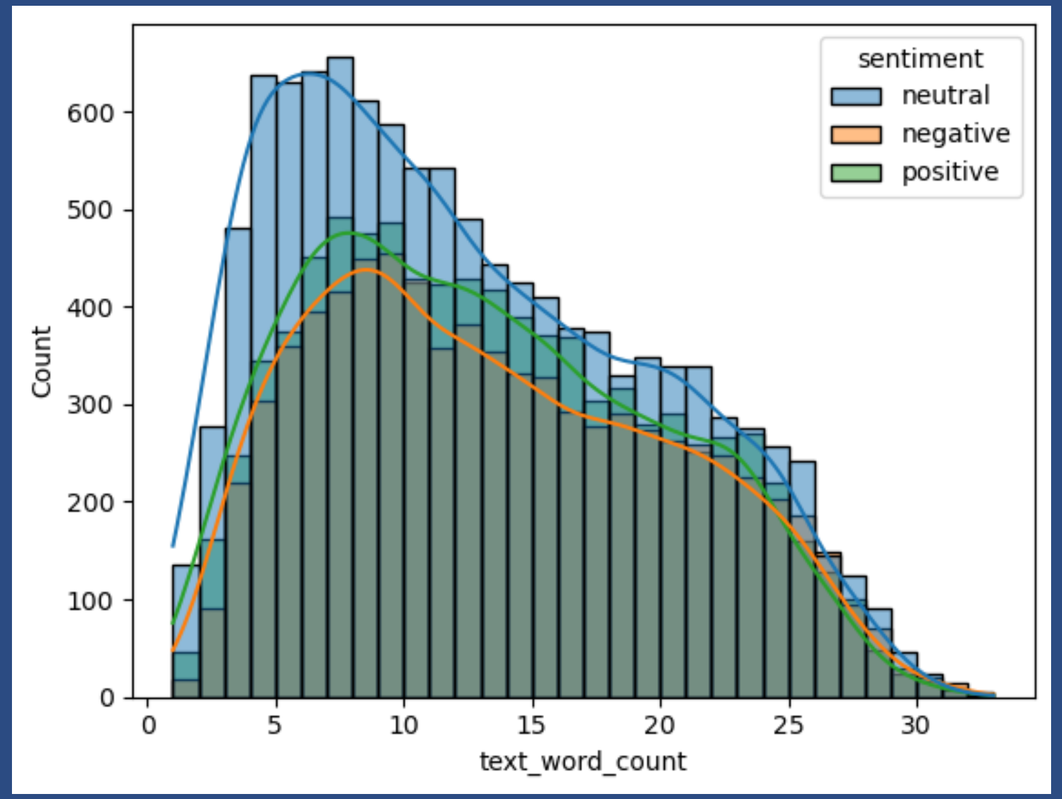
1. Neutral (Count: 11117)
2. Negative (Count: 7781)
3. Positive (Count: 8582)



Exploratory Analysis & Analytic Visualization

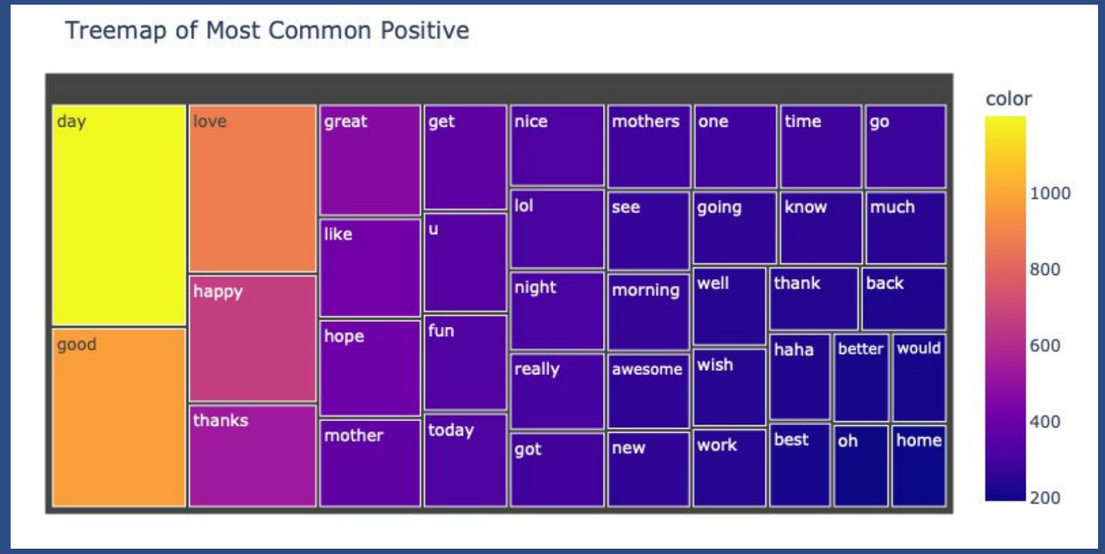
Seaborn histogram plot

- All 3 sentiments have similar distribution of words in a tweet
- Average distribution is 12 words in a tweet



Exploratory Analysis & Analytic Visualization

Treemap Charts



- The top 3 most common positive words in a tweet are "day", "good" and "love".
- "day" is 1179 of the total positive words
- "good" is 972 of the total positive words
- "love" is 872 of the total positive words

Exploratory Analysis & Analytic Visualization

Further Analysis of Unique Patterns
Tweets were taken around **mother's day** • 351 rows contain mother
A lot of **URLs** in the text • 1223 rows contain URLs
A lot of **censored words** replaced with **** in the dataset • 1000 rows contain ****

Exploratory Analysis & Analytic Visualization

Data Preprocessing Techniques

Tokenizing	Breaking a text based on the token (a meaningful unit of text)
Filtering Stop Words	Filter words you want to ignore of your text when processing it
Stemming	Reduce words to their root , which is the core part of a word
Lemmatizing	Reduce words their core meaning, but will give a complete English word that make sense on its own
Tagging parts of speech	Parts of speech is a grammatical term that deals with the roles words play when you use them together in sentences
Chunking	To identify phrases

Exploratory Analysis & Analytic Visualization

Data Preprocessing

3 techniques used:

1. tokenizing
2. filtering stop words
3. stemming

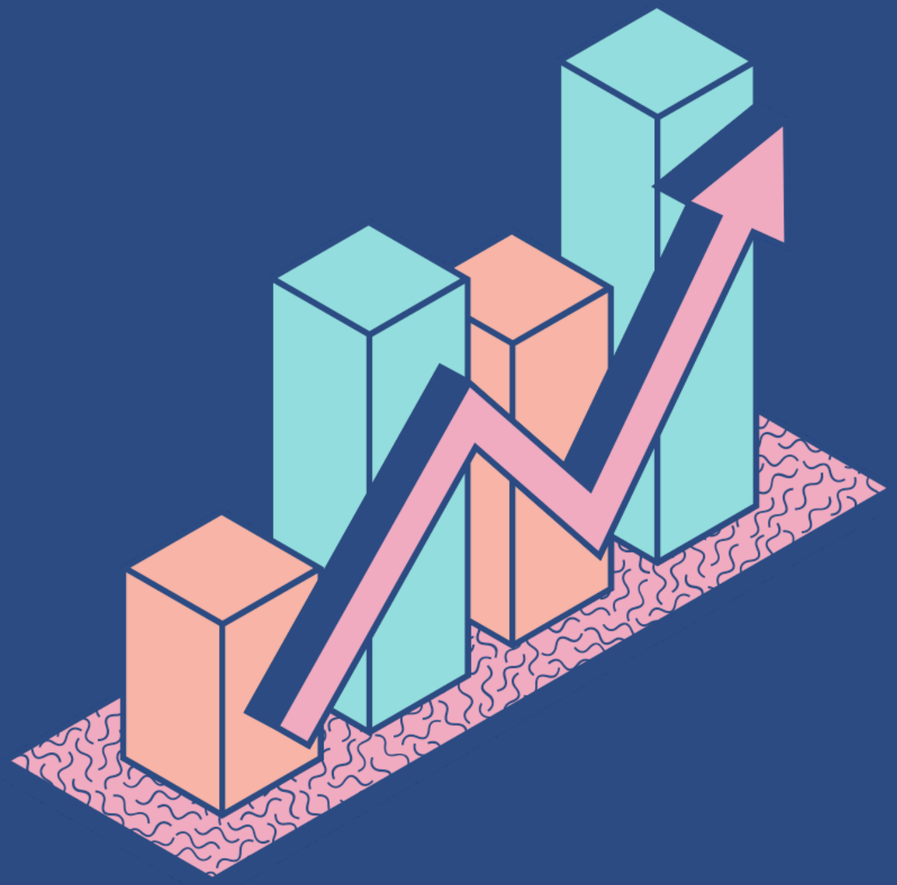
```
In [38]: print("Uncleaned: " + df[df.text.str.contains("http")].iloc[0].text)
         print()
         print("Cleaned: " + df[df.text.str.contains("http")].iloc[0].cleaned_text)

         Uncleaned: http://www.dotheboucy.com/smf - some shameless plugging for the best Rangers f

         Cleaned: shameless plug best ranger forum earth
```

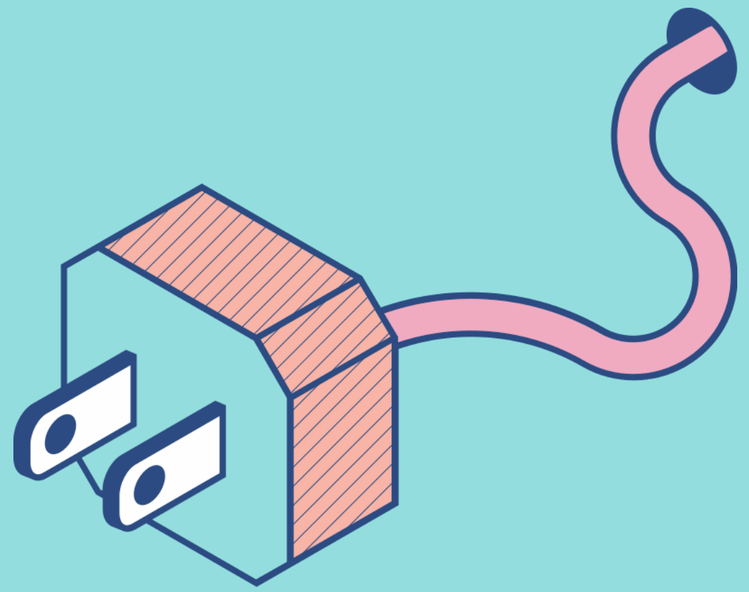
3

Predictive Models



Classification Task

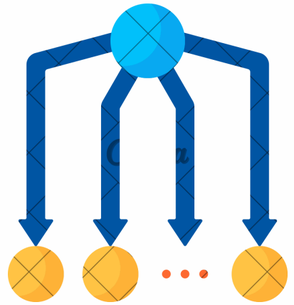
To accurately classify the sentiment of the tweet as positive, negative or neutral using different Machine Learning models



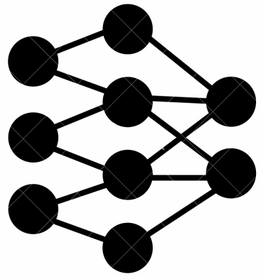
Lexicon-Based Approach

VADER
(Valence Aware Dictionary and sEntiment Reasoner)

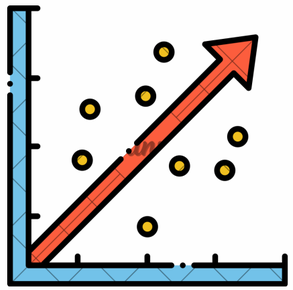
Machine Learning Approach



NAIVE BAYES



RECURRENT NEURAL NETWORK

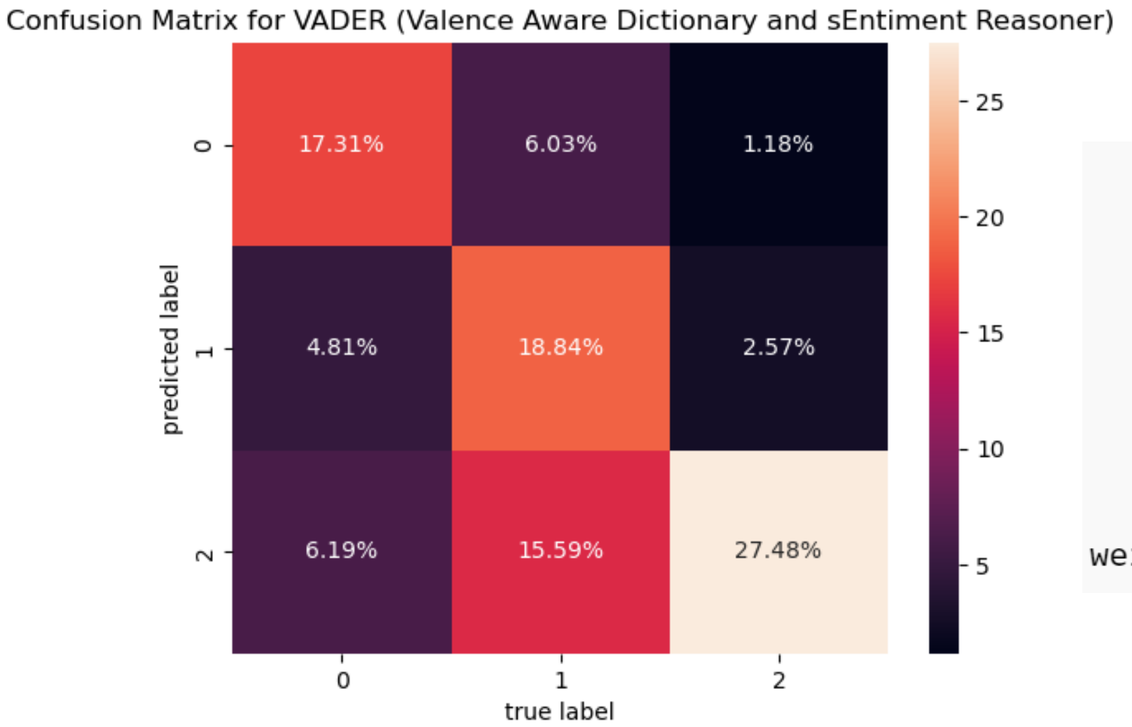


LINEAR SVC
(Support Vector Machine)

Lexicon-Based Sentiment Analysis with VADER (Valence Aware Dictionary and sEntiment Reasoner)

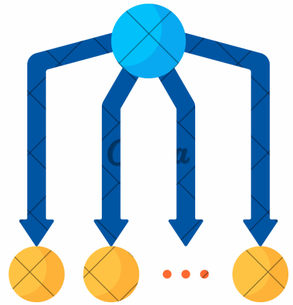
- VADER **does not** use Machine Learning
- Works by giving each word in a sentence a score based on **an internal valence dictionary**
- Aggregates the scores to provide a overall sentiment score for a sentence
- **Lower performance** of the predictor compared to more advanced models

Evaluation of Vader



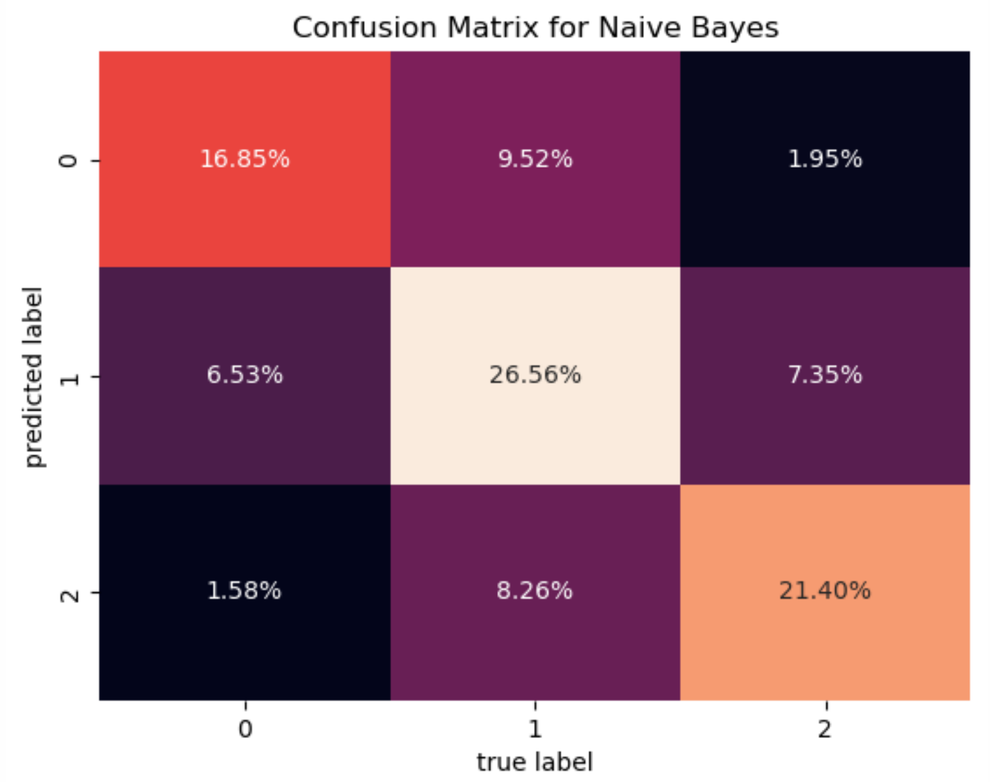
	precision	recall	f1-score	support
negative	0.71	0.61	0.66	7781
neutral	0.72	0.47	0.57	11117
positive	0.56	0.88	0.68	8582
accuracy			0.64	27480
macro avg	0.66	0.65	0.63	27480
weighted avg	0.66	0.64	0.63	27480

Naive Bayes Classifier



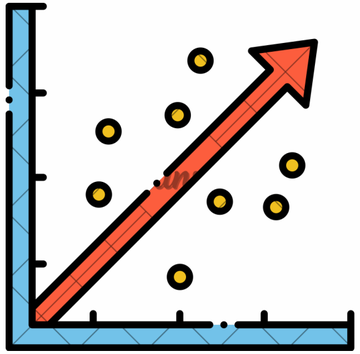
- It is a supervised machine learning algorithm, which is used for classification tasks, like text classification.
- It works by calculating probability of a given text belonging to a particular sentiment class, based on the frequency of words in text
- The algorithm is simple, efficient, and has been shown to perform well in sentiment analysis tasks.

Evaluation of Naive Bayes Classifier



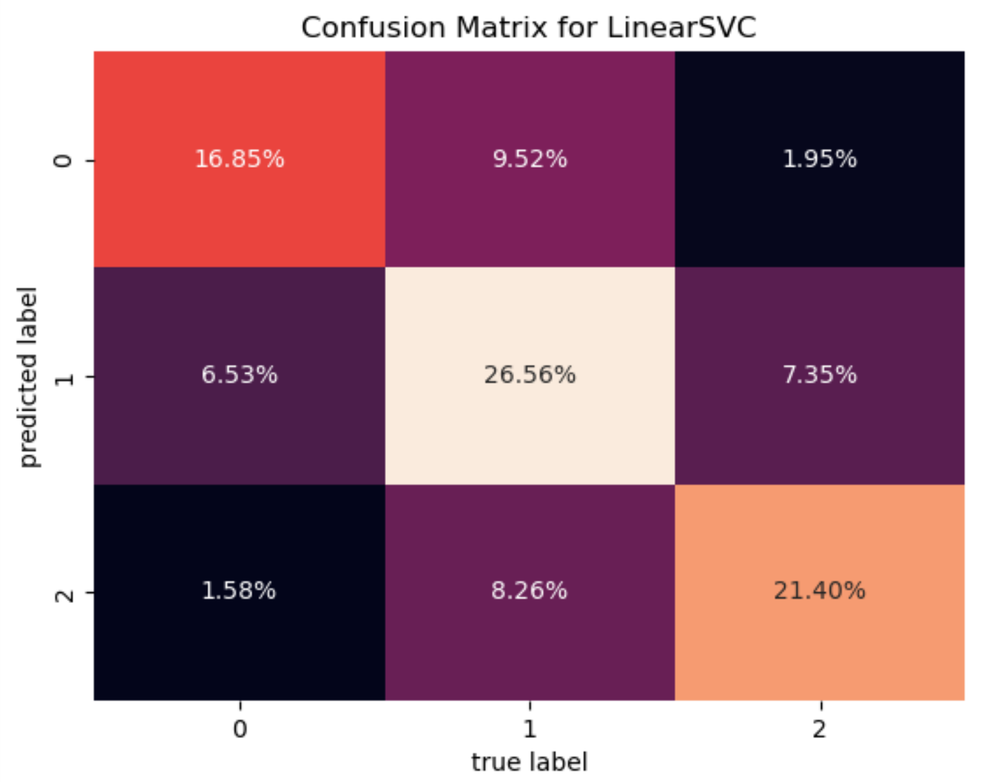
	precision	recall	f1-score	support
negative	0.67	0.60	0.63	1556
neutral	0.60	0.66	0.63	2223
positive	0.70	0.68	0.69	1717
accuracy			0.65	5496
macro avg	0.66	0.65	0.65	5496
weighted avg	0.65	0.65	0.65	5496

Linear SVC Classifier



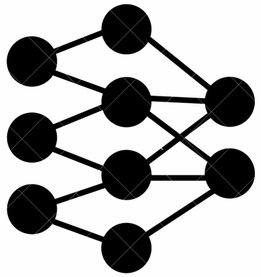
- Similar to Naive Bayes Classifier - used for classification tasks
- Linear SVC works by mapping data points to a high-dimensional space and then finding the optimal hyperplane that divides the data into two classes.
- It is less prone to overfitting compared to other classification algorithms which is why it is better than Naive Bayes Classifier.

Evaluation of Linear SVC Classifier



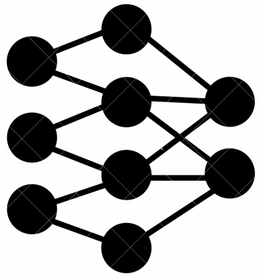
	precision	recall	f1-score	support
negative	0.68	0.61	0.65	1556
neutral	0.62	0.70	0.66	2223
positive	0.75	0.69	0.72	1717
accuracy			0.67	5496
macro avg	0.68	0.67	0.67	5496
weighted avg	0.68	0.67	0.67	5496

Recurrent Neural Network using Keras



```
model = Sequential()
model.add(layers.Embedding(max_words, 20)) #The embedding layer
model.add(layers.LSTM(15,dropout=0.5)) #Our LSTM layer
model.add(layers.Dense(3,activation='softmax')) #Our ouput layer
```

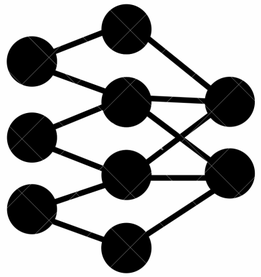
Recurrent Neural Network using Keras



Embedding Layer

- To learn word embeddings from scratch
- Converts our wordy sentences into dense vectors

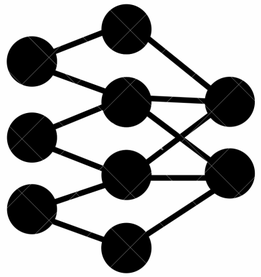
Recurrent Neural Network using Keras



LSTM Layer

- Type of RNN especially performant in text classification tasks
- Long Short Term Memory networks memorises context of words
- We utilised the dropout rate to reduce overfitting

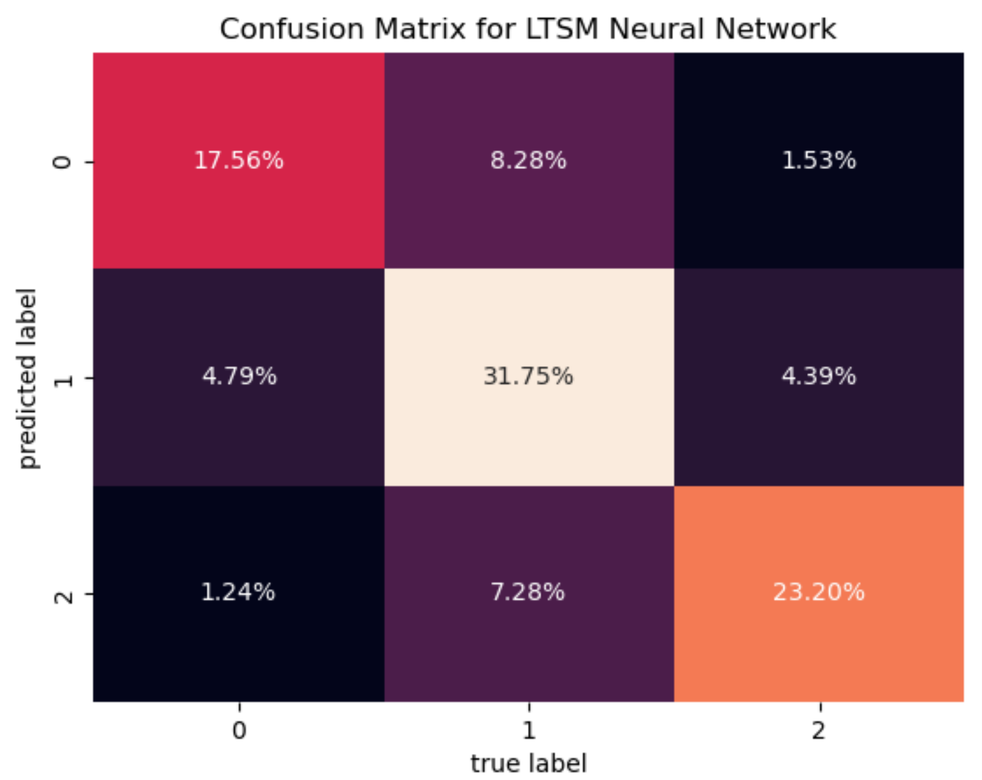
Recurrent Neural Network using Keras



Dense Layer

- Output layer with **3 output units** to represent our 3 classes
- Uses **softmax** activation to collect **probabilistic** scores for each class

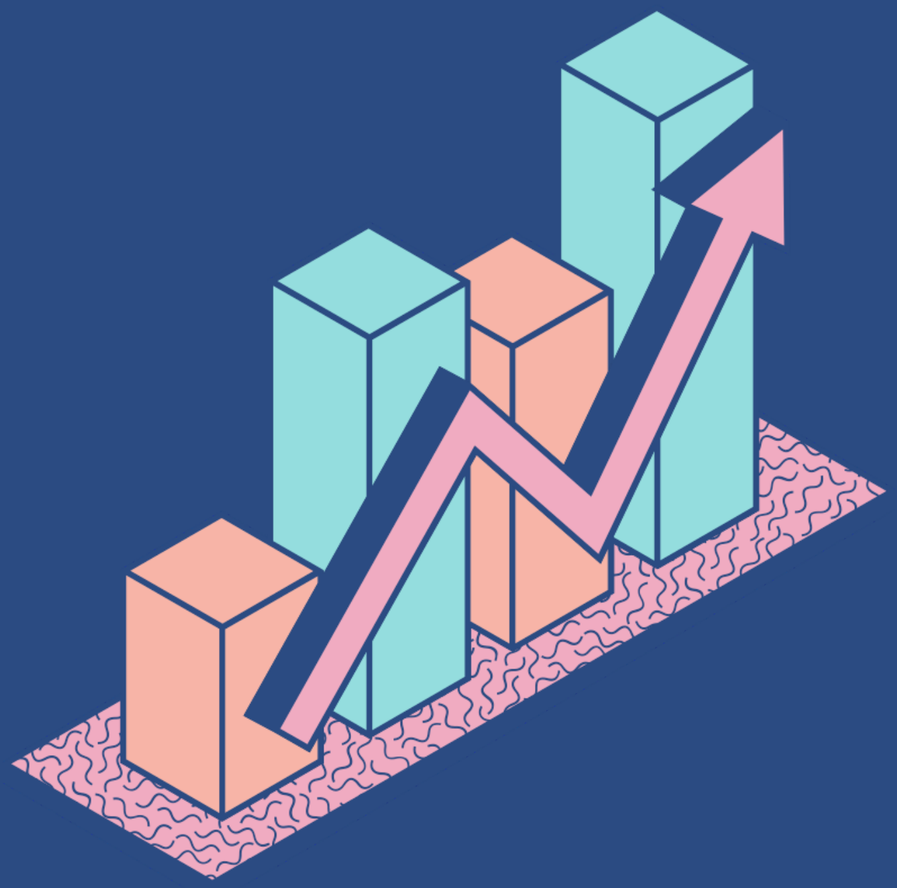
Evaluation of RNN



	precision	recall	f1-score	support
0	0.74	0.64	0.69	1504
1	0.67	0.78	0.72	2249
2	0.80	0.73	0.76	1743
accuracy			0.73	5496
macro avg	0.74	0.72	0.72	5496
weighted avg	0.73	0.73	0.73	5496

4

Conclusion



Conclusion

1 STEP

Our Problem

Finding a dataset and coming up with a problem statment

2 STEP

Exploratory Data Analysis

Finding trends in the text data using visualization

3 STEP

Data Preprocessing

Cleaning and preparing the text data using new tools

4 STEP

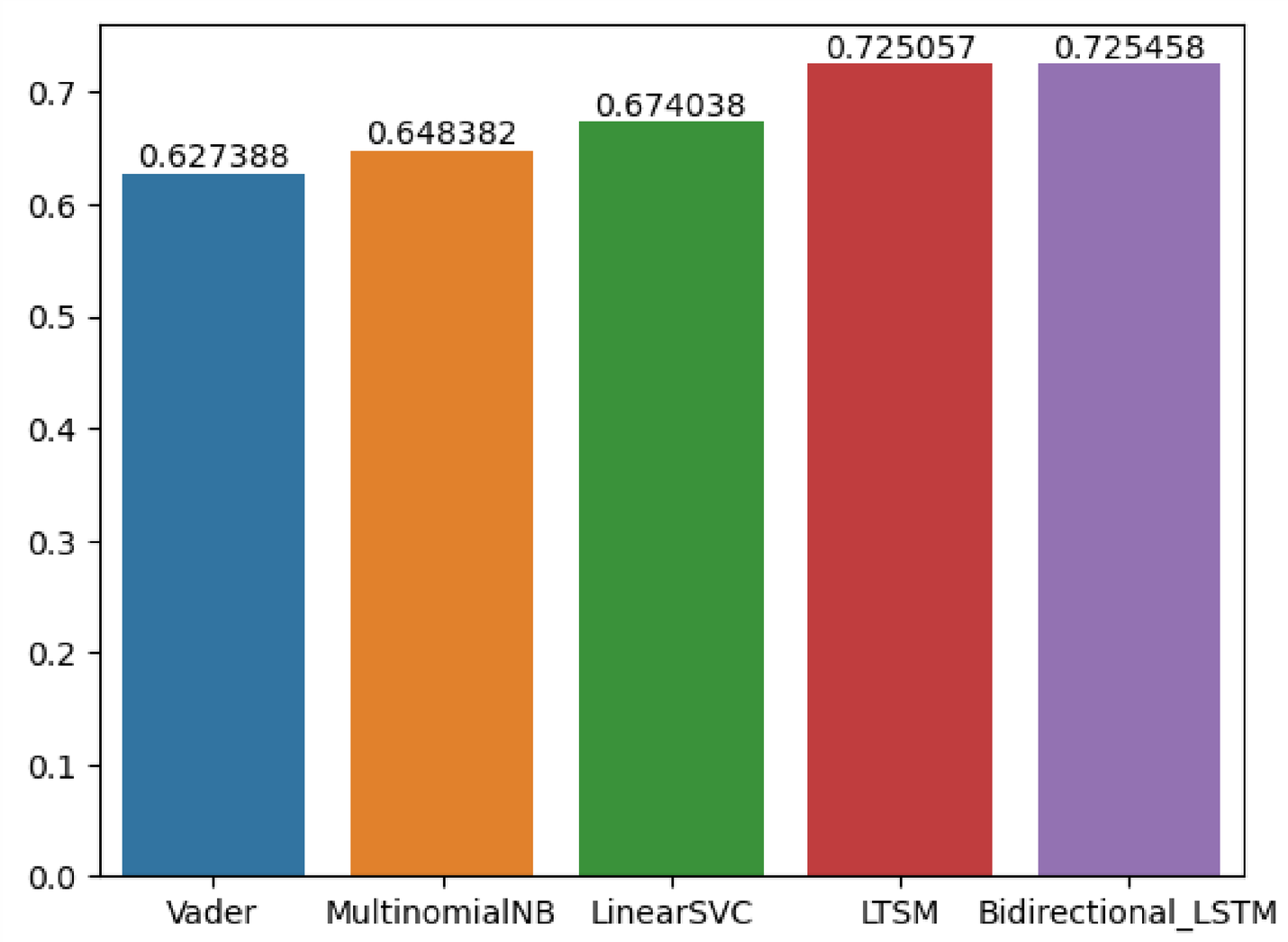
Machine Learning Model Training

Researching and preparing text classification models

5 STEP

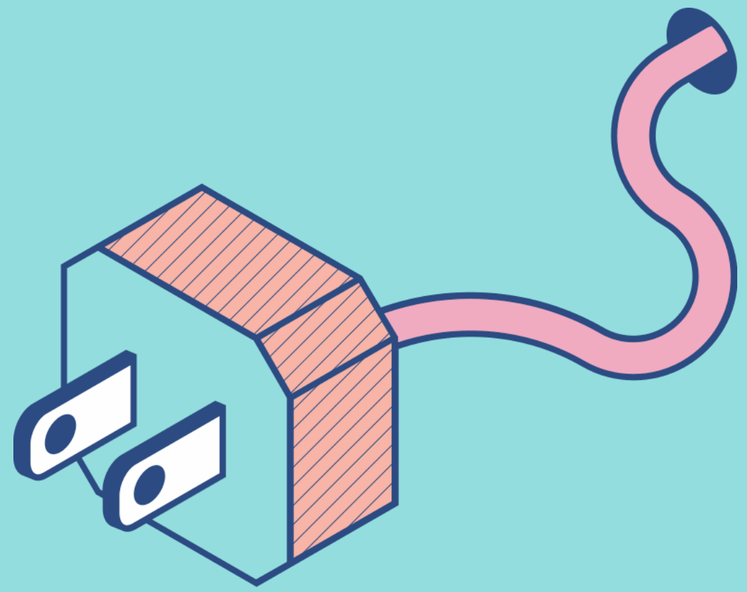
Model Evaluation

Interpreting the results of the model predictions and scoring them


0.627388
0.648382
0.674038
0.725057
0.725458
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Vader
MultinomialNB
LinearSVC
LTSM
Bidirectional_LSTM

AI Predicted Tweet sentiments with 72% Accuracy

Sentiment Analysis is a tedious task for Humans. With the aid of Machine Learning and Artificial Intelligence, we can accurately predict sentiments of a huge number amount of text, creating a **stepping stone** for a vast range of applications such as **Social Media Monitoring** and **Cyber Bullying Detection**.



Thank You!