# Kagan ILTER

# Sentiment Analysis with 55.000 Tweets

Springboard Data Science Career Track

Capstone Project-2

# Problem Definition

The scope of this project is to predict the sentiment of a short twitter text and categorize them over 3 sentiments (positive, negative, neutral)


| Companies/ Organizations | Social Media Influencers | Ecommerce traders | Websites / applications |
| --- | --- | --- | --- |

# Data Information

- Dataset is acquired from https://data.world/crowdflower/sentiment-analysis-in-text
- 4 features and 55.000 data points/samples .
- Contains labels for the emotional content (such as happiness, sadness, and anger)
- Target Feature is ‘sentiment’

# Data Understanding and Preprocessing

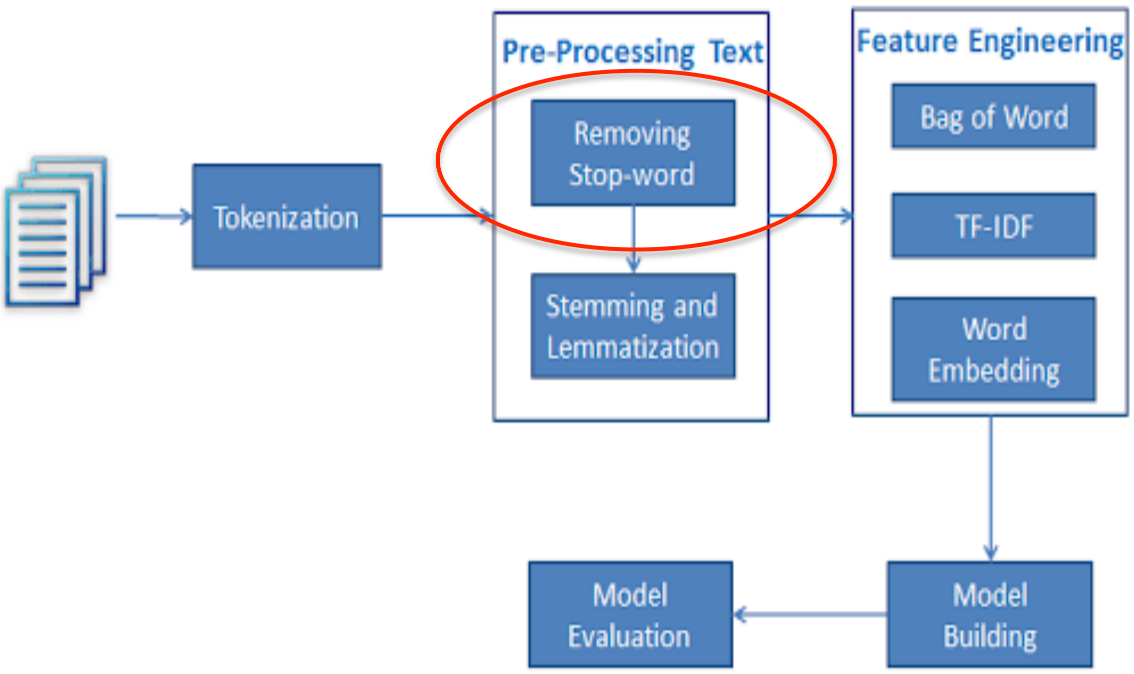
| | tweet_id | sentiment | author | content |
|---|---|---|---|---|
| 0 | 1956967341 | empty | xoshayzers | @tiffanylue i know i was listenin to bad habit earlier and i started freakin at his part =[ |
| 1 | 1956967666 | sadness | wannamama | Layin n bed with a headache ughhhh...waitin on your call... |
| 2 | 1956967696 | sadness | coolfunky | Funeral ceremony...gloomy friday... |
| 3 | 1956967789 | enthusiasm | czareaquino | wants to hang out with friends SOON! |
| 4 | 1956968416 | neutral | xkilljoyx | @dannycastillo We want to trade with someone who has Houston tickets, but no one will. |
| 5 | 1956968477 | worry | xxxPEACHESxxx | Re-pinging @ghostridah14: why didn't you go to prom? BC my bf didn't like my friends |
| 6 | 1956968487 | sadness | ShansBee | I should be sleep, but im not! thinking about an old friend who I want. but he's married now. damn, & he wants me 2! scandalous! |

Anger, boredom, hate, worry, sadness : Negative
Happiness, fun, love, surprise, enthusiasm, relief : Positive
Empty, neutral : Neutral

# Data Understanding and Preprocessing

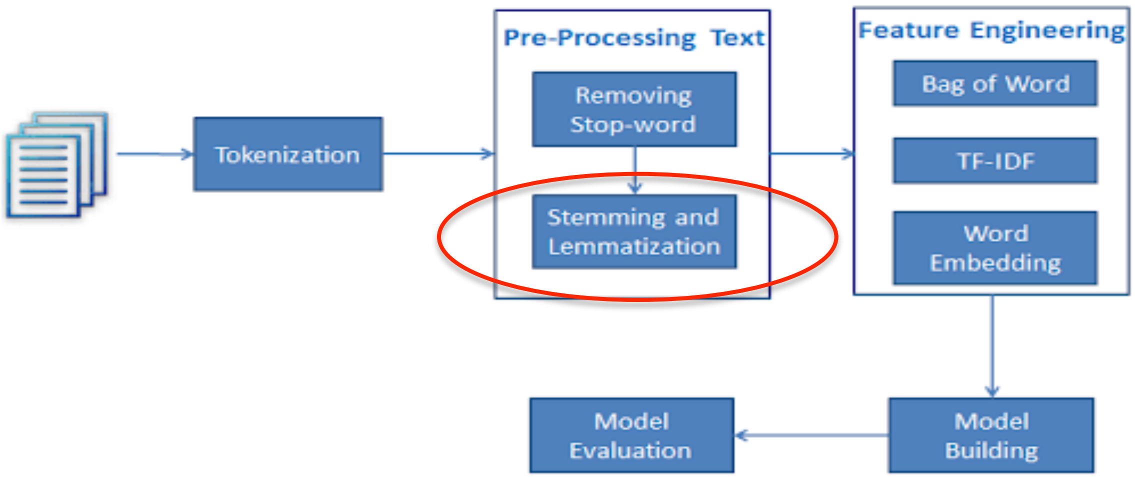
- <u>Cleaning the tweet texts :</u>

  - Removing special characters,
  - Punctuations,
  - Accented characters,
  - Html tags,
  - Spaces,
  - Tickers
  - Hyperlinks
  - Usernames
  - Stopwords



# Data Understanding and Preprocessing

- **Stemming and lemmatization:**



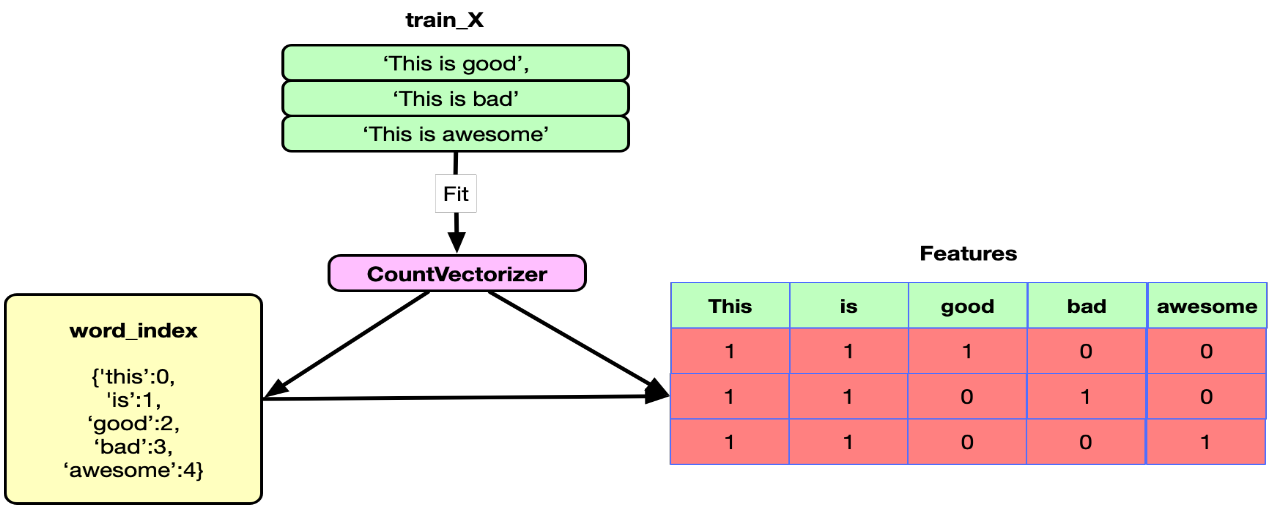

- **Missing Values, outliers, duplicates:**

  - No missing values
  - 173 repeating duplicates

# Data Understanding and Preprocessing

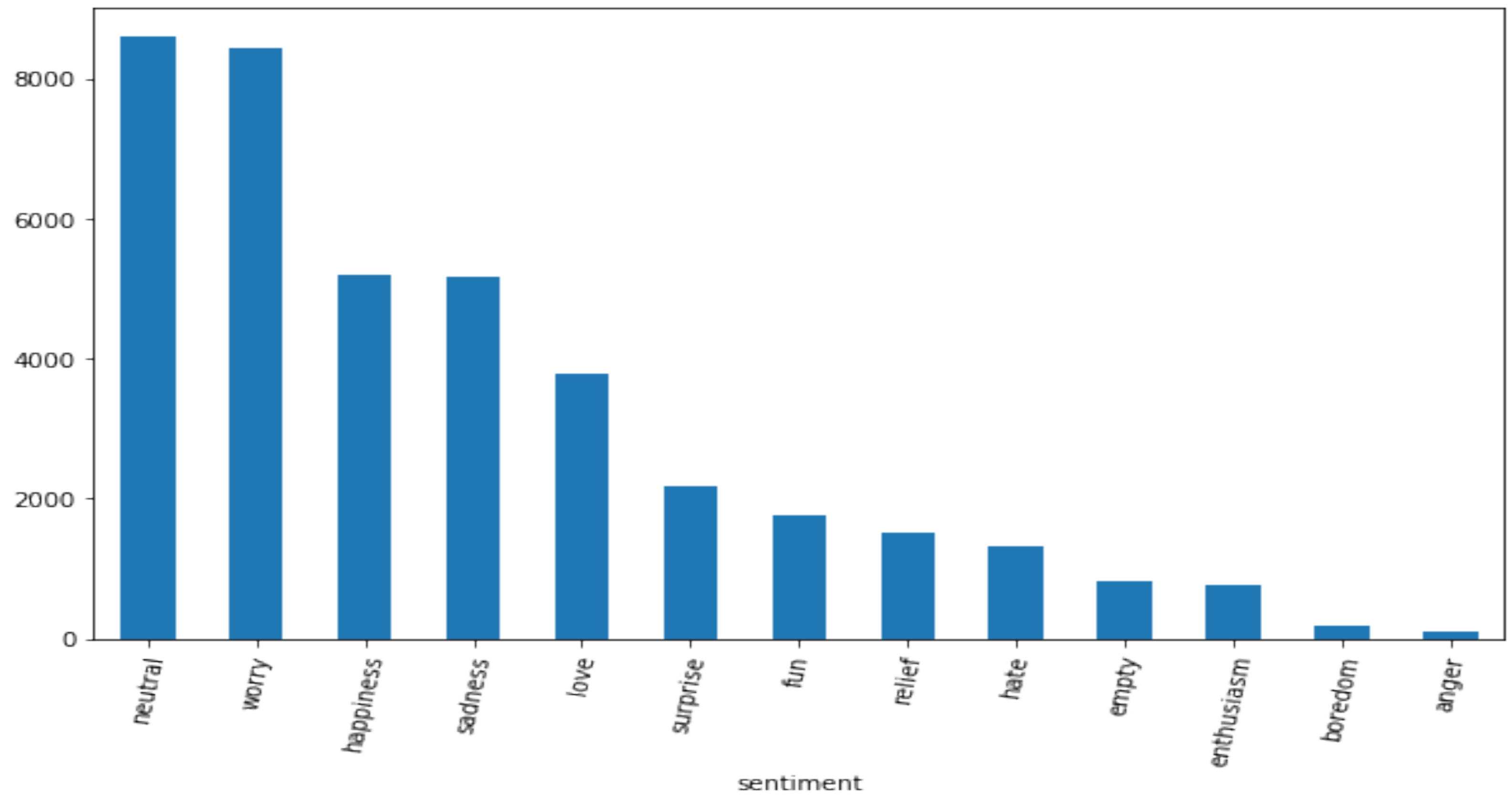
➢ <u>Bag of Words (Plus n-grams) (CountVectorizing in ScikitLearn):</u>

- A mathematical model to represent unstructured text (or any other data) as numeric vectors



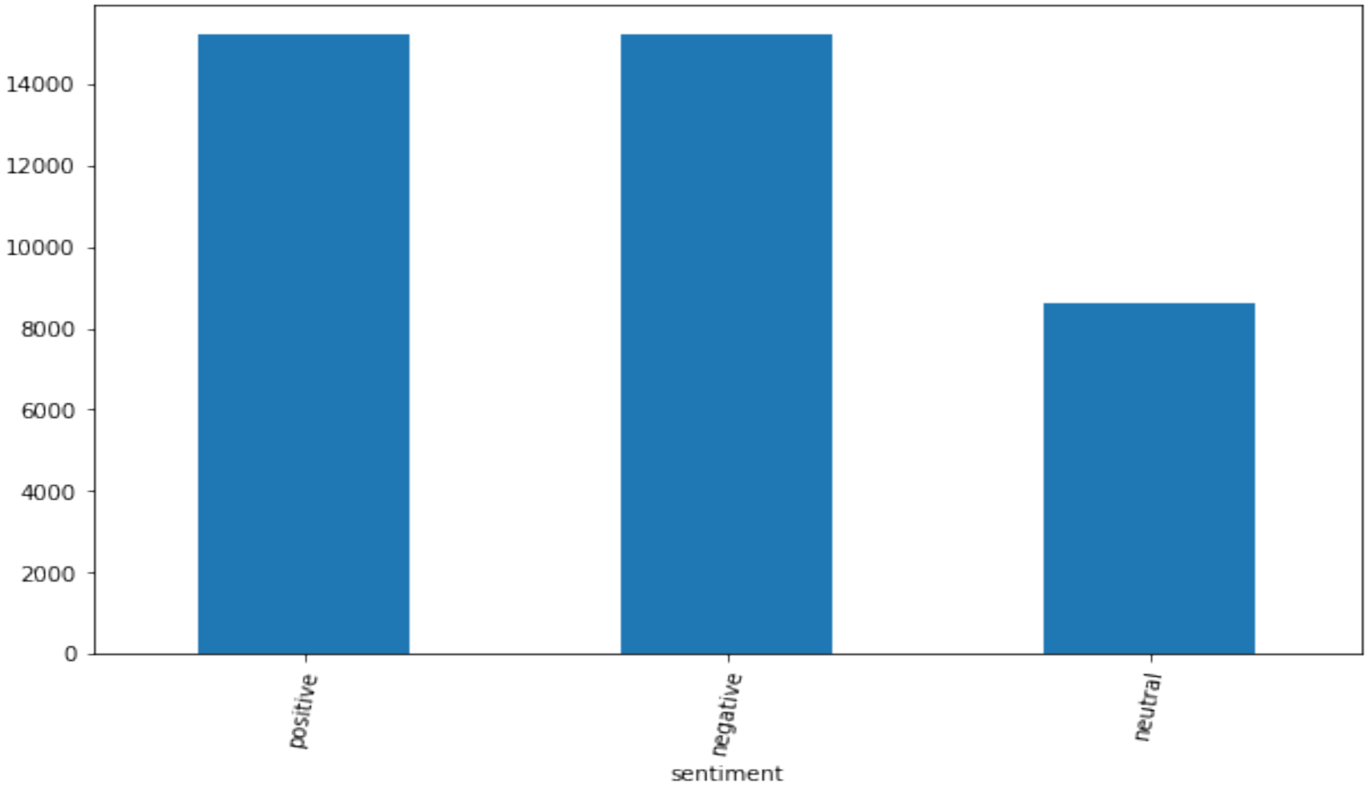

| you | youll | your | youre | youtube | youve | yr | yum | yummy | ûï |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

# Data Visualization




The distribution of emotions in the data set (imbalanced)

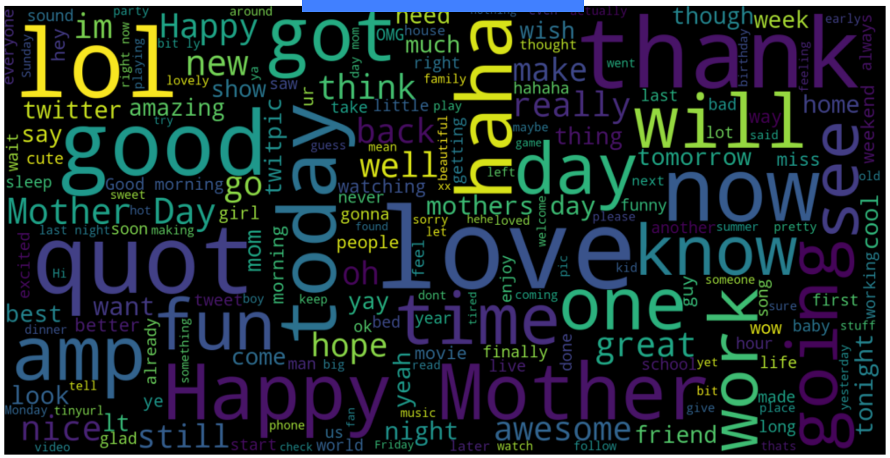
# Data Visualization



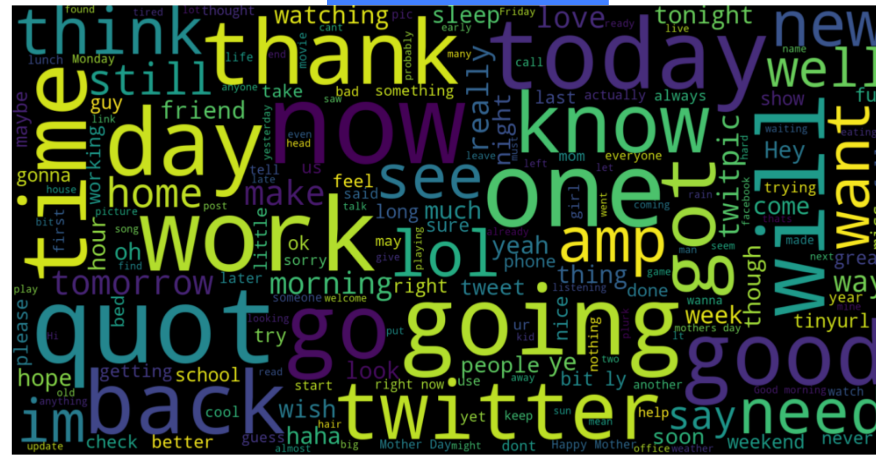

The distribution of 3 sentiments

# Data Visualization

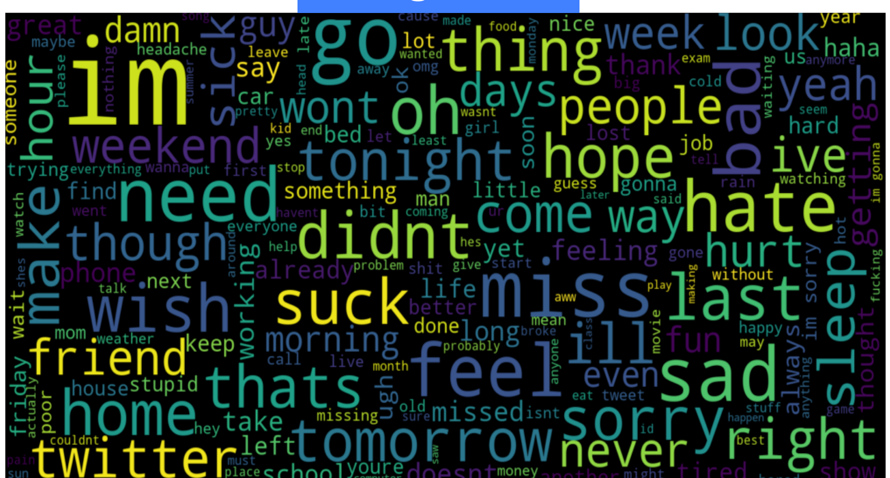
Positive




Neutral

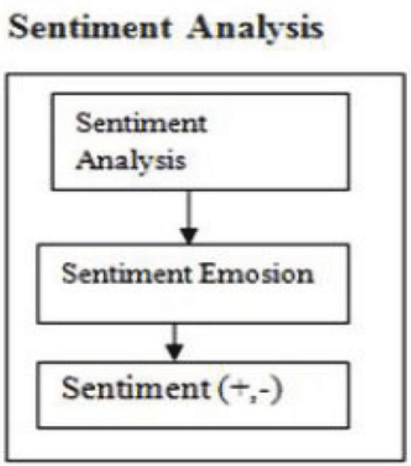


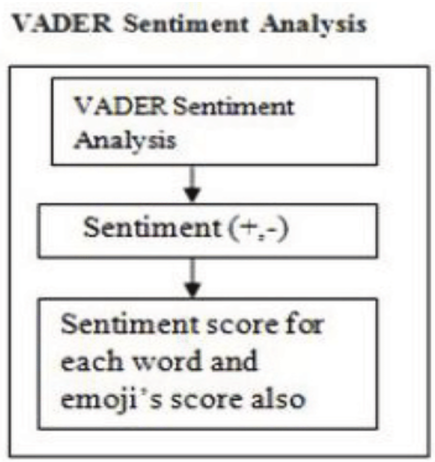

Negative



# Data Comparison with NLTK Vader

- NLTK Vader is a parsimonious rule-based model for sentiment analysis of social media text. With Vader, we can compare our dataset's classification with the Vader classification



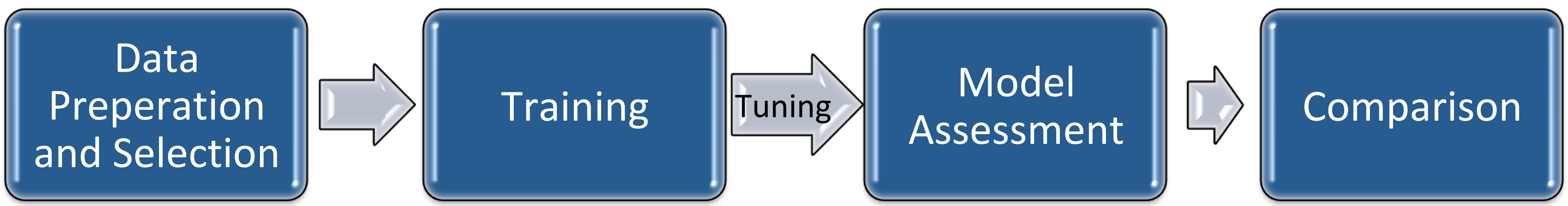

- If the Vader compound result is lower than -0.05, the text is categorized as negative sentiment. Higher than 0.05 is a positive sentiment.

- As a result of Vader comparison, we realized that our dataset's classification is better than Vader's classification.

# Predictive Modeling

# Machine Learning Models

- Supervised learning multi-class classification




- Subset of whole data: 30%
    - Logistic regression,
    - Naive bayes,
    - Linear svm,
    - Random forest,
    - Gradient boosting,
    - Xgboosting
    - Deep learning

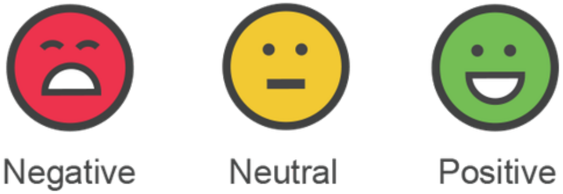
# Machine Learning Models

- **Logistic Regression for 13 emotions**

  Logistic Regression is one of the basic and popular algorithm to solve a classification problem. Because of the imbalanced features, the accuracy is low.

Accuracy: 0.2208

- **Logistic Regression for 3 sentiments**




Accuracy: 0.6158

# Machine Learning Models

- **Linear SVC with Count Vectorizing**

The objective of a Linear SVC (Support Vector Classifier) is to fit to the data providde, returning a "best fit" hyperplane that divides, or categorizes ourdata.

Accuracy: 0.6156

- **Naïve Bayes with Count Vectorizing**

Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

Accuracy: 0.5667

# Machine Learning Models

- **Random Forest with Count Vectorizing**

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. We used random forest with a 'balanced' class weight

Accuracy: 0.6156

- **Gradient Boosting with Count Vectorizing**

Gradient boosting classifiers are a group of machine learning algorithms that combine many weak learning models together to create a strong predictive model.

Accuracy: 0.5488

# Machine Learning Models

➢ **Word frequencies with Tf-Idf:**

TF-IDF are word frequency scores that try to highlight words that are more interesting, e.g. frequent in a document but not across documents.

➢ **Logistic Regression with Tf-idf**

Accuracy: 0.5488

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| negative | 0.71 | 0.72 | 0.72 | 6124 |
| neutral | 0.43 | 0.41 | 0.42 | 2917 |
| positive | 0.61 | 0.62 | 0.61 | 4317 |
| accuracy | | | 0.62 | 13358 |
| macro avg | 0.58 | 0.58 | 0.58 | 13358 |
| weighted avg | 0.62 | 0.62 | 0.62 | 13358 |

# Machine Learning Models

- **Pipeline**

A pipeline consists of a chain of processing elements (processes, threads, coroutines, functions, etc.), arranged so that the output of each element is the input of the next. We used scikit-learn pipeline models.

We used :

- Count Vectorizer
- Tr-Idf Transformer
- Multimnomial  Naïve Bayes model as a classifier

Accuracy: 0.5841

# Machine Learning Models

- **Grid Search**

We applied a grid-searching model for scanning the data to configure optimal parameters for our model. With the best parameters below, the accuracy is:

```
GridSearchCV:
Best score: 0.593
Best parameters set:
        clf__alpha: 1.0
        vect__max_df: 0.7
        vect__min_df: 10
        vect__stop_words: 'english'
```

Accuracy: 0.5946

# Machine Learning Models

- **Deep Learning Algorithm**

Deep Learning (which includes Recurrent Neural Networks, Convolution neural Networks and others) is an important type of Machine Learning approach.

We used Keras deep learning frame and Tensorflow in our NLP text classification model.

```
Epoch 1/5
 - 336s - loss: 0.8706 - acc: 0.5974
Epoch 2/5
 - 333s - loss: 0.7893 - acc: 0.6472
Epoch 3/5
 - 333s - loss: 0.7533 - acc: 0.6618
Epoch 4/5
 - 333s - loss: 0.7174 - acc: 0.6812
Epoch 5/5
 - 333s - loss: 0.6860 - acc: 0.6964
```

Accuracy: 0.6964

# Conclusion

- We have chosen a difficult and informal text in order to make it harder to analyze the sentiment (general text, not a review or a sentiment pool data)

- The distribution of the features are not balanced. In order to deal with this problem, we concatenated additional 15 thousands row data set, we limited the categorization, we applied random forest hyper-tuning and feature engineering.

- We used random forest model to balance the distribution and Grid Search with 5-fold cross validation technique to deal with the overfitting problem. Because of the reasons mentioned above, our best score is 0.68 after applying Tf-idf and deep learning algorithm.

- Most important things for an effective sentiment analysis of short social media texts are data preprocessing, feature engineering and choosing the best model

# Future Works

- For increasing the accuracy of our model, it is very important to find additional balanced data sets, and applying effective feature engineering techniques.

- Applying Word2vec or Phrase modeling could also improves the model.

- Categorization of sentiments in 2 classes (such as bad or not bad) could also give higher results.

- Run time for these kinds of data with decision tree models and deep learning models is relatively long. Decreasing the run time with more efficient computers or cloud systems could be used for increasing the effectiveness.

# Thank You