Sentiment Analysis Documentation

Dataset: It's a Twitter data includes 1600000 tweets; half has positive sentiment and negative for the rest.
https://www.kaggle.com/kazanova/sentiment140

Naïve Byes as Classic NLP Approach:

Using Python and NLTK only

1- Read data as a Data Frame using Pandas.

2- Remove punctuations, stop words, URL's and mentions using NLTK

3- Tokenize each tweet to make stemming on each word to get its root

4- Finally call NLTK Naïve Bayes classifier and pass data for it.

5- Calculate accuracy on new data.

Accuracy: was **55.2%** on 10,000 tweets.

Logistic Regression as ML Approach:

Using Python, NLTK, Pandas and Scikit-Learn

1- Read data as a Data Frame using Pandas.

2- Remove punctuations, stop words, URL's and mentions using NLTK

3- Tokenize each tweet to make stemming on each word to get its root

4- Using CountVectorizer convert data to numeric features for each word

5- Finally call scikit-learn logistic regression classifier and pass data for it.

6- Calculate confusion matrix
Accuracy: was **72.8%** on 10,000 tweets.

Deep Learning Approach:

Using Python, NLTK, Pandas, Keras and matplotlib

1- Read data as a Data Frame using Pandas.
2- Remove punctuations, stop words, URL’s and mentions using NLTK
3- Tokenize each tweet to get most common words and build word dictionary
4- Convert each word in dictionary to sequence of number
5- Convert each class to numeric encoding
6- Build network based on sequential model:
   (Dense:64, Activation: Relu, input: words_num) (Dense:64, Activation: Relu) (Dense:2, Activation: softmax)
7- Compare loss of Validation and training by visualization.
8- add a regularization parameters L2 to handle it to handle overfitting.

   Accuracy: was **71%** on just 10,000 tweets.