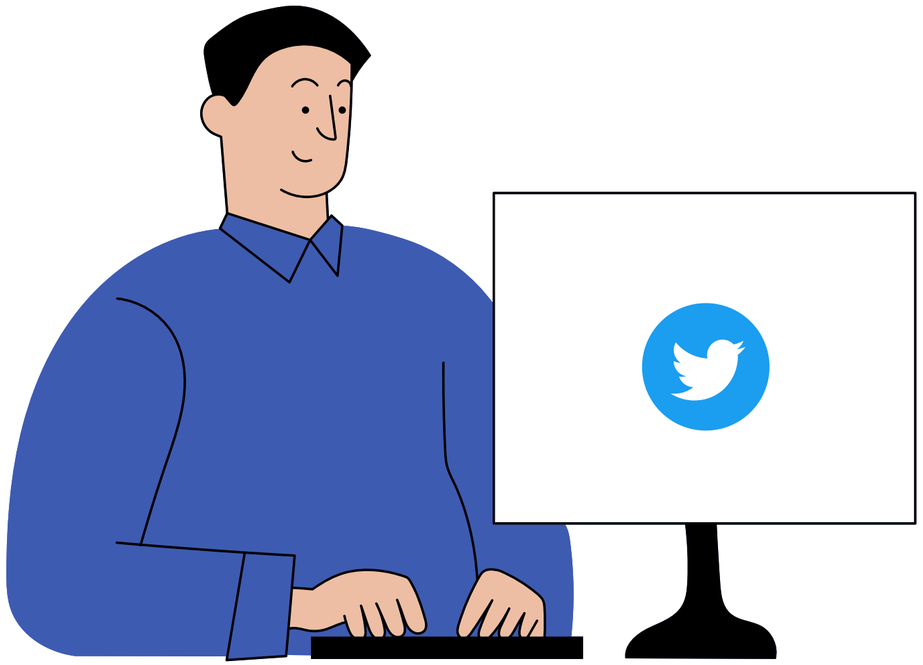
HIGH PERFORMANCE
MACHINE LEARNING

Twitter Sentiment Analysis

Prakyath Madadi (pm3140)
Akshit Kurani (ak9300)

Today's Agenda



- Introduction to the Session
- Executive summary
- Approach
- Main results
- Observations/conclusion
- GitHub Link

Introduction to Session

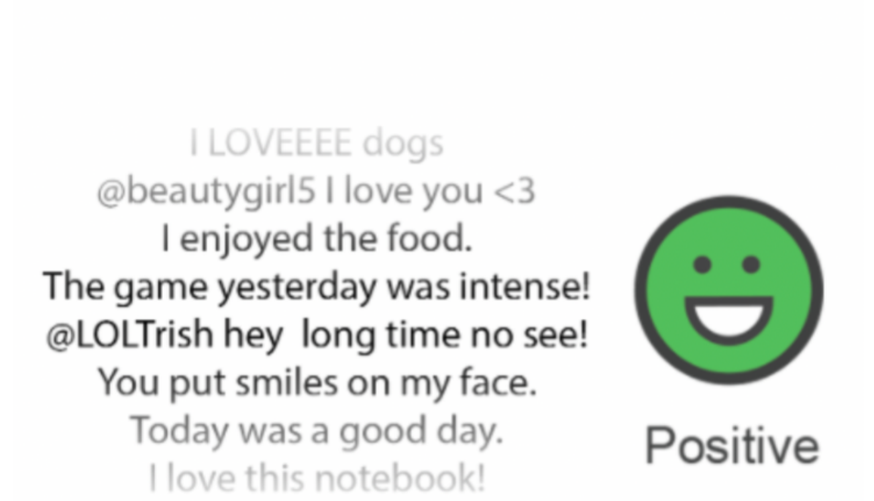
- With this age of social media and Twitter playing a vital role in elections, virtual currency, investment decisions, etc., it becomes extremely important to carry out an analysis of the tweets posted. However, they are unstructured, out of vocabulary, non-grammatical, acronyms, etc form.
- Sentiment Analysis is the text classification tool that analyses an incoming message and tells whether the underlying sentiment is positive, negative, or neutral.
- Hence, we have worked on it to carry out Twitter sentimental analysis to make it feasible for different applications which include prediction of opinions, mock results of games, political results, etc.

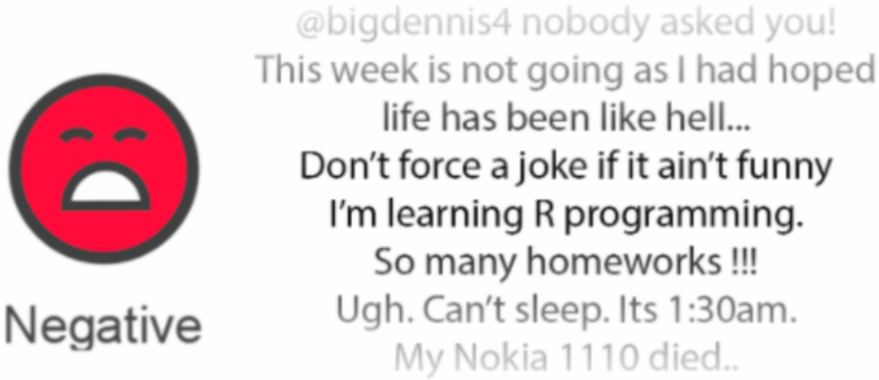
Executive Summary

- We carried out Twitter Sentimental Analysis using different tokenizers and ML algorithms, with the goal of benchmarking the accuracies and run times.

- The tokenizers included Bag of Words, TF-IDF, Word2vec, Doc2vec, BERT, etc

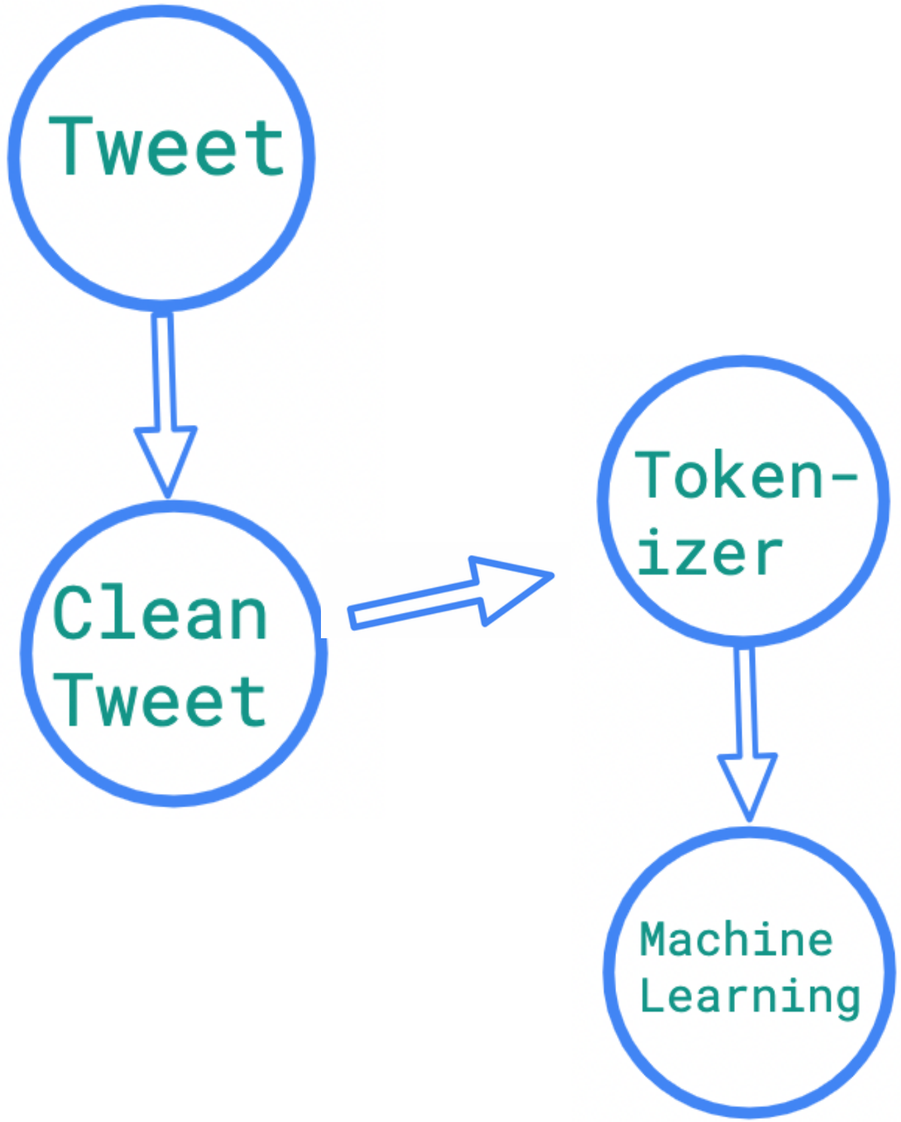
- We used different classifiers which included Logistic Regression, SVM, Random Forest, and Fine-Tune BERT classifier.





Approach

- First, we take the tweet and process(clean) it, and then pass it through a tokenizer to form vectors.
- Then, we apply different Machine Learning Algorithms to the vectors to classify the tweet.
- We noted the accuracies and training time for different algorithms, to find the best one.
- We also scaled the data from size 100 to 100k tweets, (we also tried for 1M) to find the effect of data size on accuracy and time.



Results

Data Size - 100, 000 tweets

Tokenizer	Training Time			Accuracy			F1 Score		
	Logistic	SVM	RF	Logistic	SVM	RF	Logistic	SVM	RF
TF-IDF	1.45s	13min	4.2min	**73.265**	67.12	72.46	**76.224**	72.4	75.4
Bag of Words	0.5s	6min	1.5min	70.16	53.64	68.64	74.43	67.83	72.12
Word2vec	0.91s	**48.5 min**	1.7min	69.81	50.47	63.54	73.34	66.57	71.39
Doc2vec	1.47s	43.4 min	1.8min	52.09	49.7	51.26	**65.9**	66.4	66
Bert	0.84s	48.2 min	**22sec**	50.46	**50.45**	54.94	67.06	67	66.5

Results

BERT Classifier

Data size	**Training Time** (per epoch)	**Accuracy**	**F1 Score**
100	<1 sec	60	66.6
1000	7.2 sec	75	69.13
10k	1.5 min	75.8 val accuracy	75.5
100k	5.5 min	78.5 val accuracy	-
1M	~500 min	~80 val accuracy ~97 train accuracy	-

Results

Time for Tokenizing

Data/Tokenizer	TF_IDF	Bag of Words	Word2vec	Doc2vec	BERT
100	5.43 ms	2.8 ms	0.571 sec	0.718 sec	0.03 sec
1000	13.6 ms	12.5 ms	3.1 sec	3.39 sec	0.31 sec
10k	74.9 ms	80.1 ms	64 sec	97 sec	3.05 sec
100k	0.704 sec	0.718 sec	14 min 42 sec	16 min	31.54 sec
1M	-	-	-	-	~310.27 sec

Observations/Conclusion

- We observed that the BERT classifier gives us the maximum accuracy but takes up more time comparatively. (RF, Logistic, etc)
- Logistic Regression takes up the least amount of time comparatively and gives us great accuracy.
- TF-IDF tokenizer takes the least amount of time and also gives better accuracies on ML algorithms.
- The best tokenizer and classifier combination is the BERT tokenizer combined with the BERT classifier.

GITHUB LINK

https://github.com/prakyath-04/Twitter_Sentiment_Analysis

That's a wrap!

Any Questions?

May 14, 2022