Twitter Sentiment Analysis

Team - D2

Sourabh Jain	01FE17BCS214
Usman Khan	01FE17BCS235
K L Vijeth	01FE17BCS247
Prathamesh Kulkarni	01FE17BCS250

Problem Statement

To perform sentiment analysis on twitter data (both on Real time and past tweets) using PySpark.

Introduction

- Sentiment analysis is the automated process of analyzing text data and sorting it into sentiments positive, negative, or neutral. Using sentiment analysis tools to analyze opinions in Twitter data can help companies understand how people are talking about their brand.
- Twitter boasts 330 million monthly active users, which allows businesses to reach a broad audience and connect with customers without intermediaries. On the downside, there's so much information that it's hard for brands to quickly detect negative social mentions that could harm their business.

Dataset

- Twitter dataset consisting of 16M Bitcoin tweets(4 GB) from Kaggle(9 attributes).
- Sentiment140 dataset(250 MB) consisting of 1.6 Million tweets for testing(6 attributes).

Frameworks/Technologies used

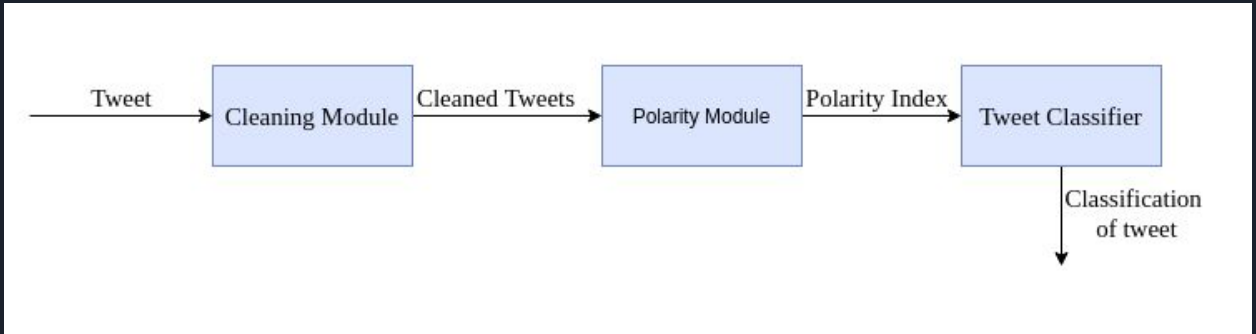
- Pyspark
- NLTK
- Pandas
- Regular expression
- Textblob
- BeautifulSoup
- Socket

Methodology

We used two different approaches with two different dataset. The two types of approaches are:

- Polarity based sentiment analysis
- Model based sentiment analysis

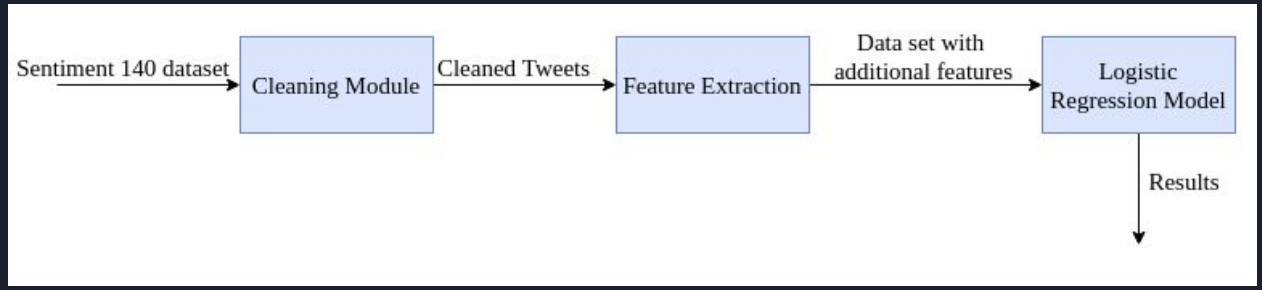
Polarity based Sentiment Analysis



Polarity based sentiment analysis

- Bitcoin dataset was used for analysis
- The tweets were cleaned using regular expressions and only english tweets were chosen.
- Textblob model was used to obtain a polarity index for each tweet.
- Tweets were classified as positive negative or neutral based on the polarity score.

Model based Sentiment Analysis

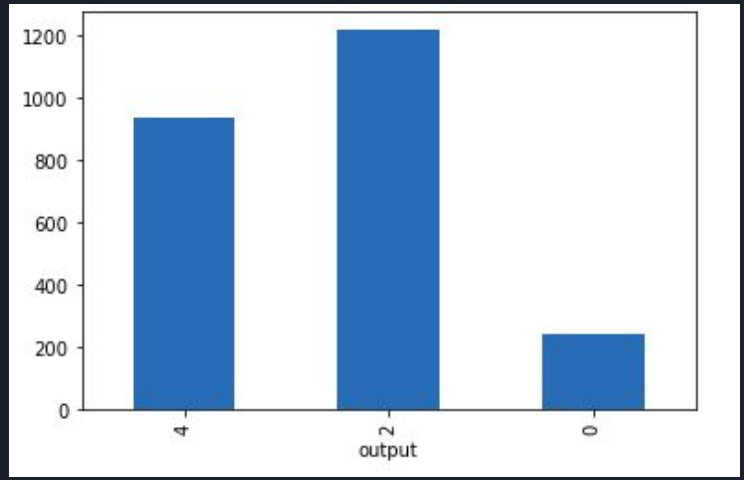


Model based Sentiment Analysis

- Hashing TF + IDF + Logistic Regression Model is built on sentiment analysis dataset
- Dataset is divided into a training set and a testing set.
- Data is sent to pipeline containing functions such as tokenizer, hashtf, idf, label_stringIdx.
- Data with extracted features is now supplied to a logistic regression model.
- Later test set is used to predict the sentiment of the tweets.

Results(Polarity based sentiment analysis)

text	englishornot	cleantweet	polarity	output
Cardano: Digitize...	1	Cardano Digitize ...	0.0	2
Another Test twee...	1	Another Test twee...	0.0	2
Current Crypto Pr...	1	Current Crypto Pr...	0.0	2
Spiv (Nosar Baz):...	1	Spiv Nosar Baz BI...	0.0	2
@nwoodfine We hav...	1	We have been buil...	0.2	4
CHANGE IS COMING....	1	CHANGE IS COMING ...	0.2	4
One of the useful...	1	One of the useful...	0.3	4
BestMixer has bee...	1	BestMixer has bee...	0.0	2
Invested my Life ...	1	Invested my Life ...	0.0	2
Bitcoin Price Hit...	1	Bitcoin Price Hit...	0.14818181	4
#Countdown #Comin...	1	Countdown ComingS...	-0.2	0
You have roughly ...	1	You have roughly ...	-0.033333335	0
Running bitcoin	1	Running bitcoin	0.0	2
Crypto prices ros...	1	Crypto prices ros...	0.6	4
Bitcoin SV (BSV) ...	1	Bitcoin SV BSV Pr...	0.0	2
Bitcoin Whales Mo...	1	Bitcoin Whales Mo...	0.125	4
As Intel $INTC Va...	1	As Intel INTC Val...	0.6	4
This is true. Cry...	1	This is true Cryp...	0.24318182	4
@APompliano If by...	1	If by Bitcoin you...	-0.104166664	0
Pls save cuttoff ...	1	Pls save cuttoff ...	0.0	2

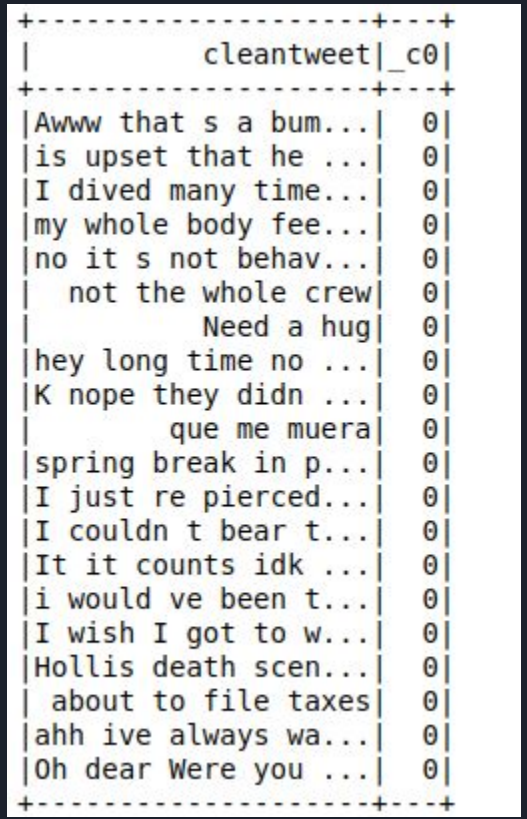


Results(Polarity based sentiment analysis)

- Execution time for 10000 rows:
  - Traditional data processing using pandas: 120.12 seconds
  - Big-data processing using Pyspark: 81.41 seconds
- Execution time for entire dataset using pyspark: 420.31

Results(Model based sentiment analysis)

Cleaning tweets



cleantweet	_c0
Awww that s a bum...	0
is upset that he ...	0
I dived many time...	0
my whole body fee...	0
no it s not behav...	0
not the whole crew	0
Need a hug	0
hey long time no ...	0
K nope they didn ...	0
que me muera	0
spring break in p...	0
I just re pierced...	0
I couldn t bear t...	0
It it counts idk ...	0
i would ve been t...	0
I wish I got to w...	0
Hollis death scen...	0
about to file taxes	0
ahh ive always wa...	0
Oh dear Were you ...	0

Results(Model based sentiment analysis)

Feature extraction

```
+--------------------+---+--------------------+--------------------+--------------------+-----+
|          cleantweet|_c0|               words|                  tf|            features|label|
+--------------------+---+--------------------+--------------------+--------------------+-----+
|Awww that s a bum...|  0|[awww, that, s, a...|(65536,[18354,216...|(65536,[18354,216...|  0.0|
|is upset that he ...|  0|[is, upset, that,...|(65536,[1981,3085...|(65536,[1981,3085...|  0.0|
|I dived many time...|  0|[i, dived, many, ...|(65536,[2548,2888...|(65536,[2548,2888...|  0.0|
|my whole body fee...|  0|[my, whole, body,...|(65536,[1880,9243...|(65536,[1880,9243...|  0.0|
|no it s not behav...|  0|[no, it, s, not, ...|(65536,[1968,8538...|(65536,[1968,8538...|  0.0|
+--------------------+---+--------------------+--------------------+--------------------+-----+
only showing top 5 rows

Total Time = 50.00811433792114
```

Results(Model Based sentiment analysis)

```
import time

start=time.time()

from pyspark.ml.classification import LogisticRegression
lr = LogisticRegression(maxIter=100)
lrModel = lr.fit(train_df)
predictions = lrModel.transform(val_df)

from pyspark.ml.evaluation import BinaryClassificationEvaluator
evaluator = BinaryClassificationEvaluator(rawPredictionCol="rawPrediction")
evaluator.evaluate(predictions)

end=time.time()

print("Total Time =",end-start)
```

```
Total Time = 246.10463690757751
```

```
import time

start=time.time()

accuracy = predictions.filter(predictions.label == predictions.prediction).count() / float(val_set.count())
print(accuracy)

end=time.time()

print("Total Time =",end-start)
```

```
0.7525
Total Time = 38.19857215881348
```

Model Results

An accuracy of 75.25% was obtained by using logistic regression.

Conclusion and future scope

- Big data processing techniques are significantly more efficient than traditional processing techniques while handling large volumes of data.
- With better and more powerful machine a more complex model could be applied to obtain higher accuracy.

References

- Geetika Gautam, Divakar Yadav. (2014). Sentiment Analysis of Twitter Data Using Machine Learning Approaches and Semantic Analysis. IEEE 2014
- https://towardsdatascience.com/another-twitter-sentiment-analysis-bb5b01ebad90
- https://www.tutorialspoint.com/pyspark/
- https://medium.com/@aieeshashafique/exploratory-data-analysis-using-pyspark-dataframe-in-python-bd55c02a2852
- https://databricks.com/glossary/pyspark
- https://monkeylearn.com/blog/what-is-tf-idf/#:~:text=TF%2DIDF%20is%20a%20statistical,across%20a%20set%20of%20documents.
- Datasets:
    - http://help.sentiment140.com/for-students
    - https://www.kaggle.com/alaix14/bitcoin-tweets-20160101-to-20190329

Thank You