

CSEE5590-0001/490-0003: Big Data Programming

Increment 1 Report

Project Title: Spark ETL and Sentiment Analysis

Team Members:

- Neeraj Padarshi - 19
- Hiresh Jakkala Bhaskar- 11
- Hari Y – 29

Goals and Objectives:

Motivation:

In this data-driven world, handling data has become vital in the decision-making process in many industries such as Telecom, Banking, Financial and Health sector servicing industries. Managing the sheer volumes of data and getting insights from it would be the main factors. One of the amazing frameworks that can handle big data in real-time and perform different analysis, using Apache Spark.

Objectives:

Our Project's main idea is to do the ETL process using Spark Streaming and implementing the machine learning concepts on this real-time data. The source of our system is Twitter data and we would be using Streaming Content which is real-time processing of data, by using streaming API we would be collecting the data in a near real-time process for a set of defined keywords. Then we would be performing the transformations on the streaming set of RDD's and load the data into the Hive system which is similar to basic ETL process. Also, we would be performing the EDA on twitter data while capturing the context of the data. Our project would also highlight the Sentiment Analysis System where we populate real-time sentiments for the tweets. It also identifies the major keyword factors for a tweet to be categorized into positive or negative sentiment.

Significance:

For sentiment analysis we are using an existing ML tool(TextBlob) to predict the sentiment of the tweets and based on that we will analyse the keywords which are vital in the prediction of sentiment and we will train a new ML model with that analysis to predict the sentiment of tweets which will increase the accuracy of the prediction.

Features:

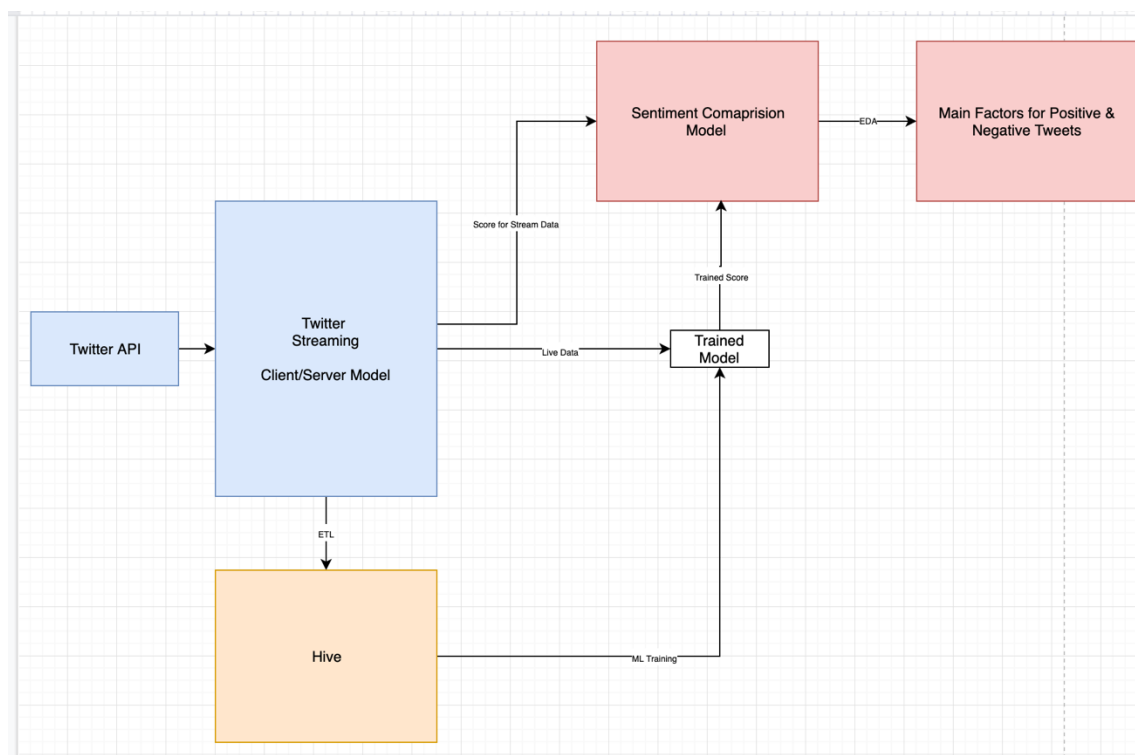
The project's features includes collecting real-time tweets from twitter streaming API, performing ETL (Pre-Processing the data, Extracting necessary information and loading the data in to the Hive), and using TextBlob, predict the sentiment for each tweet and with that train a new ML model with tweets as input and sentiment as output, to enhance the prediction of sentiment for each tweet.

Increment - 1

Dataset:

We are using twitter streaming API to collect near real-time tweets on a specific set of keywords. Each tweet is in JSON format, which is a key-value pairs. It has various information about a tweet like tweet text, who tweeted it, user information, location where the tweet originated, source of the tweet like Android, Iphone etc. We are collecting real time data from streaming API using Spark streaming context with a window of 20 seconds and the streaming API was able to download 500kb of data in which we are filtering for keywords like sports, cricket, football, hockey and we are extract ~80kb from that.

Detail design of Features:



Initially, we have created an account in Twitter Developers API. From the provided API tokens and credentials, we downloaded the tweets using the Spark Streaming application. Our project has a Client/Server kind of model where we got tweets using the Tweepy PY library which acted as the Server stage of our application. Now, we have utilized the Spark Streaming application to send the request and, on the success, the application would receive the tweets from the server on a window-based model.

Once the client receives the tweets on a windowing-based model then we will be mainly performing two operations. Firstly, on the stream data, we would show the sentiment on fly for each tweet that is streamed. Secondly, we would be storing the tweets into HIVE system after performing the set of transformations on the streamed tweets. After storing the data in HIVE, we would be building a classification model by training on the stored tweets. Once, the model is trained then we would run the model on real-time tweets and predict the sentiment.

The next stage of our project is to compare the scores provided by the model and the direct sentiment scores which were provided on the tweets. It also identifies the major keyword factors for a tweet to be categorized into positive or negative sentiment.

Analysis for Increment-1:

For downloading twitter data, we have created a twitter developer account, and we used tweepy python library and spark streaming context for downloading tweets. Each tweets if a raw json data, which has complex json formats as well, we pre-processed the data and converted the complex json to a simple text format which is each to handle and sent that to client side program through socket, from client side we transformed them in to table structure on which we can write hive queries to do some analysis. The analysis results are explained briefly in preliminary results.

Implementation:

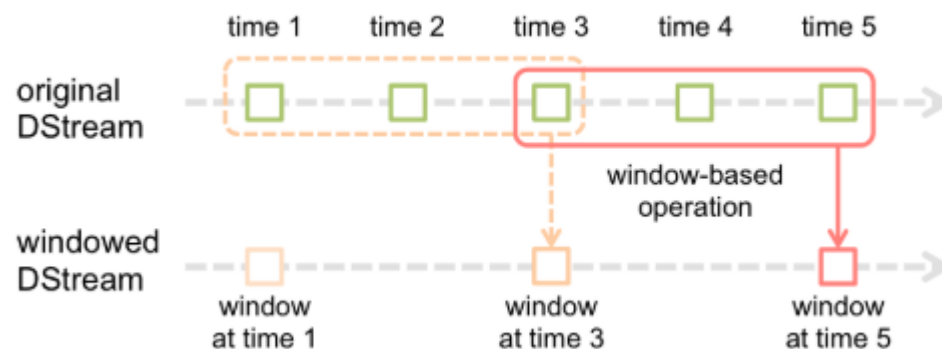
- **Client-Serve Implementation:**

For downloading the streaming twitter tweets, we followed client-serve approach,

In server side, we used tweepy library for connecting to twitter streaming API with required credentials which we got from twitter developer account. We used socket connection to transfer data from server to client. In server, the data which we got from twitter streaming API is in JSON format, which we pre-processed it in server side and sends simple text data

with required fields to the client side. The tweepy will keep on streaming data from twitter and puts the data in to the socket.

In client side, we used spark streaming context to collect the data from the socket, the spark streaming context is configured with a window length of 20 time units and sliding-interval of 20 seconds, which means for every 20 seconds spark reader will read next 20 windows of data.



- **Pre- Processing Data:**

Once the data is observed by the client, we are applying map and reduce for transforming the data in to table form and storing it in a hive table. From the hive, it is easier to write Spark SQL queries and to perform some analysis on the data.

- **Tweet Analysis:**

On the pre-processed data,

1. Finding the top 10 hash tags in the timeframe.
2. Finding the count of positive and negative tweets using Decision Rule analysis.
3. Comparing the number of tweets tweeted for different games like cricket, football etc.
4. Finding the most used URL's in tweets
5. Comparing the number of tweets tweeted from different sources like iPhone, Android, Web App etc.
6. Analysing the number of tweets originated from different locations.

Preliminary Results:

- Server Side Streaming:
 - We are able to connect to twitter streaming API and download the data in JSON format.

sendData(c)

b" Can't stop watching" - Lee Johnson's reaction to 'Better than sex' question goes viral after... #BristolCityFC <https://t.co/Km070Sy7HS>

b'RT @MySuburbanLife: Football: "I think the defense has that confidence and swagger." \n\nTwo-way heroics of Tyler Morris (8tylermorris2503), dxe2\x80\xa6

b'RT @SaddickAdams: GFA Boss @Kurtokraku.\n\nWas a footballer\n\nFormed his own club at age 17\n\nReported on the game as a sports journalist\n\nStud\x80\xa6

b'Ohio State vs Mercyhurst - NCAA Men's Hockey LIVE STREAM\n\nhttps://t.co/95c7kaeige <https://t.co/7PWTLLISzTR>

b'RT @JaredStillman: I am convinced Scott Satterfield is a BIG TIME college football coach. He will be ACC Coach of the Year. I think he shou\x80\xa6

b'RT @BWFScore: YONEX French Open 2019\n\nMD - Semi final\n\n\x80\x9f\x87\xae\x80\x9f\x87\x8b3Satwiksairaj RANK IREDDY\x80\x9f\x8f\x85\n\n21 25 \x80\x9f\x87\xae\x80\x9f\x87\x8b3Chirag SHETTY\x80\x9f\x8f\x85\n\n11 23 \x80\x9f\x87\xaf\x80\x9f\x87\x8b5Hiroyuki ENDO\x80\x9f\x87\xae\x80\x9f\x87\x8b5

b'RT @kieronafvfc: I can\x80\x9f wait. There are still some tickets available. Come & enjoy an afternoon of legends football @BedfordTown <https://\x80\xa6>

b'Alex Neil hails Preston "fight" after Blackburn comeback #PNE <https://t.co/CYISoAHj9u>

b'RT @KFOX14: .@EPMustangsFB won 42-18 vs @Bowie_bears. @ELPASO_TSD @RomanocBSA @PatrickKFOX14 <https://t.co/cqxP4pf0d5> <https://t.co/9dc9El3u3a>

b'esport Tardaron des \x80\x9f para venderlos.'

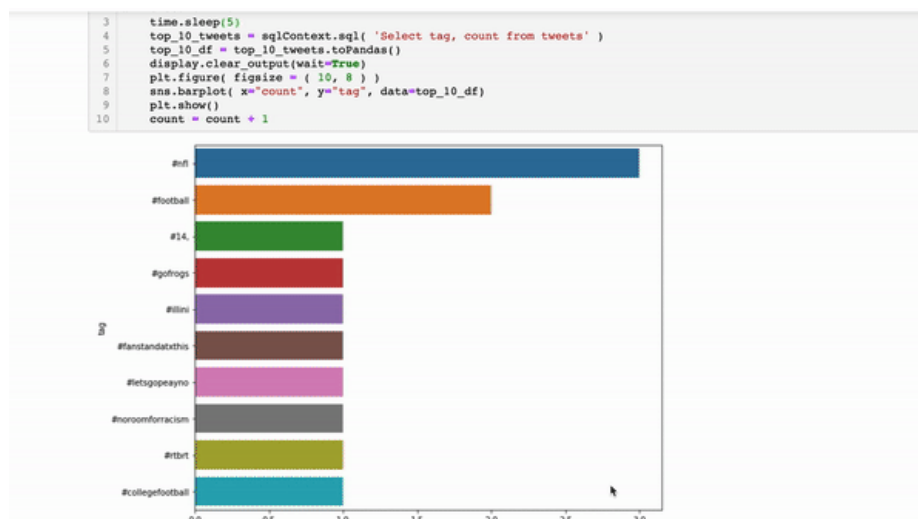
b'With my running buddy for game 2 of the fall sponsorship/partnership run promoting the industry with Kansas Strong I\x80\xa6

b'Clocks to back to normal meaning I have an extra hour to suffer the defeat and bloody VAR decision today. Football i

- Client Side Analysis:

We have performed some analysis on the pre-processed data as explained below,

- Finding the top 10 hash tags in the timeframe



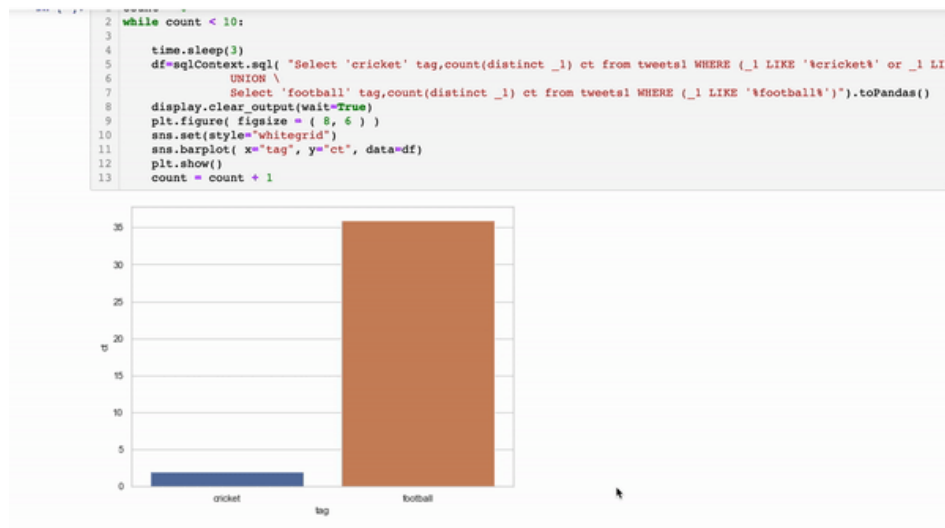
The above graph describes the top 10 tags which have a number of counts during the window frame. So, we can clearly see that the tags which are #WordSeries and #FootBall has more number of tweets.

- Finding the count of positive and negative tweets using Decision Rule analysis



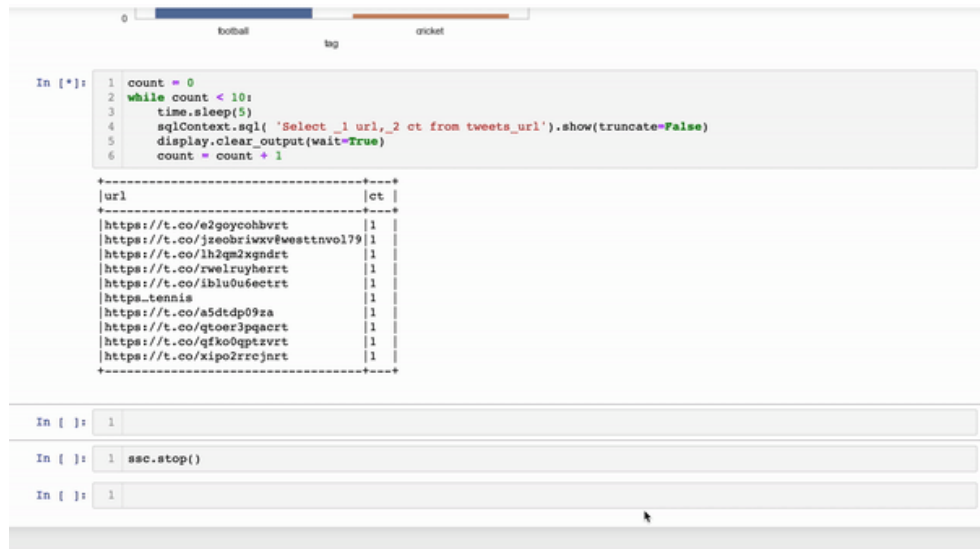
We have used the decision rule-based model, to segregate the tweets which would fall under the positive and negative sentiments. We have taken a set of words that would mainly separate the tweets. So, we can clearly say that negative tweets are more than positive tweets.

- Comparing the number of tweets tweeted for different games like cricket, football etc



The above graph describes the game tweets which has more number of counts during the window frame. So we can clearly see that there number of tweets for the football game is more.

- Finding the most used URL's in tweets



```

In [ ]: 1 count = 0
        2 while count < 10:
        3     time.sleep(5)
        4     sqlContext.sql('Select _1 url,_2 ct from tweets_url').show(truncate=False)
        5     display.clear_output(wait=True)
        6     count = count + 1

url      ct
-----
https://t.co/e2goycohbvrt 1
https://t.co/jzeobriwxv8westtnvol79 1
https://t.co/lh2gm2xgndrt 1
https://t.co/rwelruiyhertr 1
https://t.co/iblu0u6ectrt 1
https://t.co/as5tdp09za 1
https://t.co/qtoer3pqacrt 1
https://t.co/qfko0qptzvrtr 1
https://t.co/xipo2zrcjnrtr 1

```

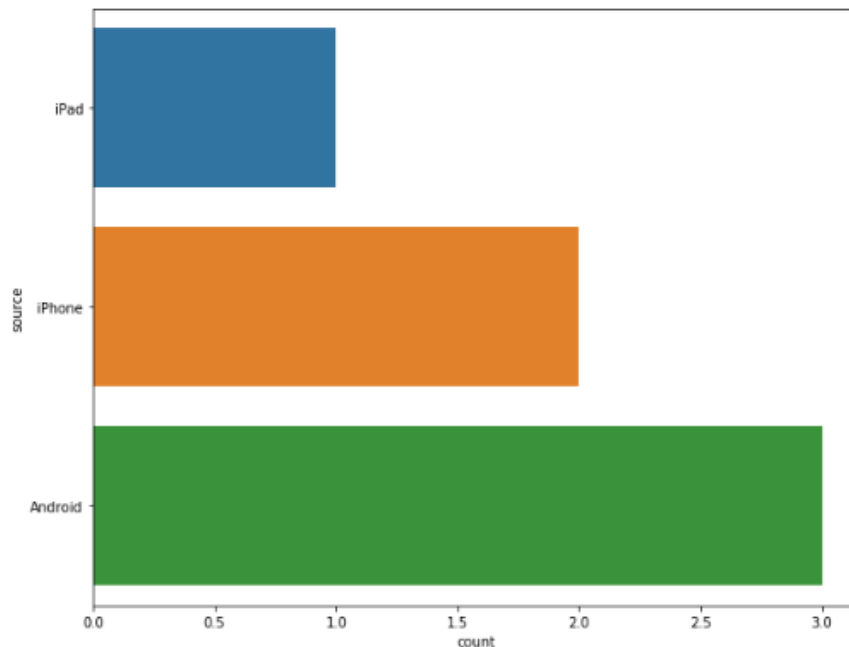
```

In [ ]: 1
In [ ]: 1 ssc.stop()
In [ ]: 1

```

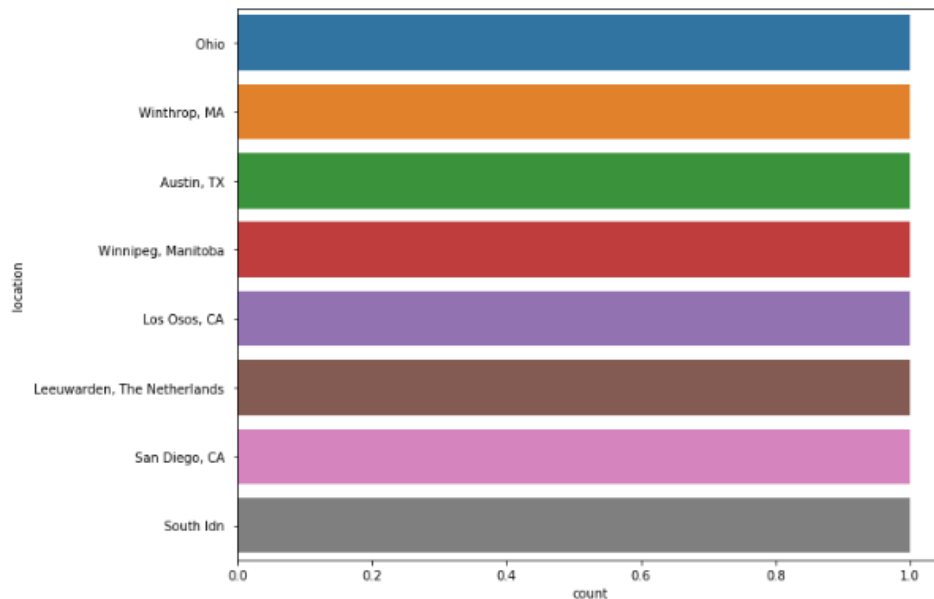
We have extracted the URL's from the tweets and performed the count on these URLs. So the above table displays the distinct URLs present in the tweets during the time frame.

- Comparing the number of tweets tweeted from different sources like iPhone, Android, Web App etc.



The above plot is for comparing the number of tweets tweeted from a particular source. From the above plot we can infer that more tweets are originated from Android devices.

- Analysing the number of tweets originated from different locations.



From the above plot we can conclude that more tweets are tweeted from ohio and Texas.

Project Management

Implementation status report

- **Work completed:**
 - **Downloading twitter streaming data**
 - Researching various ways of implementation **done by Hari**
 - Server side implementation **done by Hiresh**
 - Client side implementation **done by Neeraj**
 - **Analysis of Data:**
 - **Neeraj** - Finding the top 10 hash tags in the timeframe, Finding the count of positive and negative tweets using Decision Rule analysis.
 - **Hari** - Comparing the number of tweets tweeted for different games like cricket, football etc, Finding the most used URL's in tweets
 - **Hiresh** - Comparing the number of tweets tweeted from different sources like iPhone, Android, Web App etc, Analysing the number of tweets originated from different locations.

- **Contributions (members/percentage)**
 - **Neeraj** – 35%
 - **Hiresh** – 35%
 - **Hari** – 30%
- **Work to be completed**
 - **Description**
 - Doing tweets ETL to HIVE system. Identifying the columns which needed to be loaded into HIVE and performing the transformations on the selected attributes.
 - Training the model and predicting the score on the Near Real-Time tweets
 - Utilizing the pre-trained models and providing the Sentiment Score for the Near Real-Time tweets.
 - Comparing the differences between the scores provided the pre-trained model and trained model.
 - Identifying the main factors for a tweet to be categorized to a positive and negative tweet.
 - **Responsibilities**
 - **Neeraj** – Performing the ETL and transformation of tweets to Hive system and training the model
 - **Hiresh** - Using the pre-trained model and providing the scores for the tweets. Comparing the scores between the models.
 - **Hari** – Doing ETL on the classified data and identifying the main factors for categorizing the tweets.

References/Bibliography

- <https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/intro-to-tweet-json>
- <https://spark.apache.org/docs/latest/streaming-programming-guide.html>
- <https://www.rittmanmead.com/blog/2017/01/getting-started-with-spark-streaming-with-python-and-kafka/>
- <https://towardsdatascience.com/almost-real-time-twitter-sentiment-analysis-with-tweep-vader-f88ed5b93b1c>