ID2221 – Data Intensive Computing

Final Report - Sentiment Analysis of Twitter Data for Prediction of Stock Movements

Course Coordinator:
Amir H. Payberah

Group 1337 (Intensive data)

Óttar Guðmundsson

Xin Ren

2018-10-17

Introduction

Twitter is a social network where people post short, 140-character, status messages called tweets. Because tweets are sent out continuously, Twitter is a great way to figure out how people feel about current events. For example, how people feel about the presidential candidates, how people feel about a movie to predict box office. In this project, we created a program that enables us to find out how people feel about Apple and its products thus to predict its stock movements.

We applied sentiment analysis on tweets. The total positive, negative and neutral emotions in tweets in a 4 hours period are calculated successively to predict the movement of next hour stock price. Then we compared the prediction with data from Yahoo finance to evaluate the accuracy of the program.

TwitterStreaming.scala

The functions of this module are as follows:

1. Connecting to Twitter Streaming API:

   We use Twitter4J library to connect to Twitter Streaming API. For authentication, we set 4 system properties named twitter4j.oauth.consumerKey/consumerSecret/accessToken/accessTokenSecret to the values we got from the Twitter APP we created before hand. Then we create a new TwitterStream interface instance with function TwitterStreamFactory().getInstance().

2. Filter on the Twitter Stream:

   We create a new FilterQuery class instance to filter on the Twitter Stream. With its track function we filter out tweets that contained any of following key strings: "Apple", "iPhone", "iPad", "iOS", "iWatch", "Macbook", "macOS", "AirPods", "Tim Cook", "Steve Jobs", "Mojave". And with its language function, we filter out tweets in English. We apply the FilterQuery to TwitterStream with TwitterStream's filter function.

3. Listen to Twitter Stream and action on tweets:

   We create a new StatusListener interface instance to define actions upon events from Twitter. We mainly take care of the event indicating a new tweet created, and for other events, we simply do printout. When a new tweet matches our filter defined previously, we check if it is a retweet, if yes, it is discarded to avoid duplications. If no, we get its text and apply our parse_text function to it. Twitter replaces all the URLs in tweets with short URLs in t.co domain. The parse_text function cleans up tweet text by replacing all these short URLs with string "URL" and all the user names mentioned with string "USER" and removing line separators and emojis. Then we check again if the new text contains any of the key strings we defined in the FilterQuery, if not, the tweet is discarded. Because Twitter API does filter before the URLs transformed to short URLs in t.co domain, we may get a tweet if any of our key strings is in the original URLs, but it won't be valuable for us if it does not contain our key strings anymore after URL transformation. If the text of a tweet still contains any of our key strings, we do sentiment analysis on the text with the function provided by SentimentAnalysisUtils.scala. Then we keep the timestamp of a tweet, its text and its sentiment analysis results as an array into a list. We create a Spark RDD with the array list and then create a DataFrame from the RDD with self defined schema. We write the DataFrame to local disk for every 10 tweets. We apply the StatusListener to TwitterStream with TwitterStream's addListener function.

SparkTwitterStreaming.scala

This file provides an alternative to process Twitter stream with Spark Streaming. It does everything TwitterStreaming.scala does, and more.

1. We define a Spark Streaming custom receiver called TwitterReceiver. After it is started, it receives tweets with Twitter4J library the same way as it is in TwitterStreaming.scala, and stores tweets into Spark memory. With StreamingContext.receiverStream function, the tweets stored in Spark memory form a DStream. Then we process the records in the DStream, transfer them to Dataframe and write them to local disk the same way as we do in TwitterStreaming.scala.
2. With Spark Streaming, we are able to apply windowing on Twitter stream. We set the window size to 3h and slide interval to 1h. With reduceByKeyAndWindow function, every hour we get the number of tweets of different attitudes from last 3 hours and then predict the stock movement for this hour. The window size and slide interval can be changed. However, the bigger the window size and slide interval, the more capacity required in the Spark Cluster to produce the predictions in time.

SentimentAnalysisUtils.scala

Sentiment Analysis is performed by using RNN components in Standford Core NLP. Standford Core NLP is a standard natural language software used for extracting various form of sentiments from large set of text. We define annotators property as "tokenize, ssplit, pos, lemma, parse, sentiment". Standford Core NLP gives a sentiment value for each sentence in a text. There are 5 different values: 0(very negative), 1(negative), 2(neutral), 3(positive) and 4(very positive). For each text, we produce 3 sentiment results:

1. Main Sentiment: the sentiment value of the longest sentence in the text.
2. Average Sentiment: the average sentiment value of all the sentences in the text.
3. Weighted Sentiment: the weighted sentiment value with sentence length as weight.

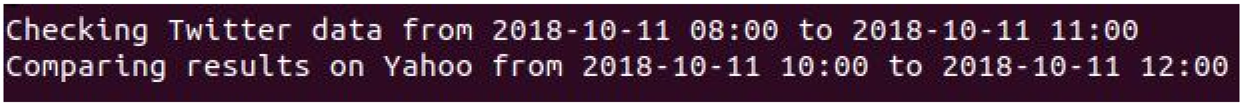
The Average and weighted sentiment values are rounded before mapped to corresponding string.

GetFinanceData.scala

This file consists of three functions, the main function that generates variables used to filter the data and printing out the results, the twitter function that filters data from the stream and the yahoo function that returns the actual results of the stocks.

Main

The main function can be called from the terminal, accepting 4 different parameters, a date, a time, a number and another number in this order. The date and the time together are converted into a timestamp variable indicating the start of twitter stream. The 3rd parameter indicating the length of twitter stream (window size) is converted into an int, representing hours being added to the former created timestamp, creating a second timestamp. These two timestamp variables will be used to filter the twitter stream. The 4th parameter indicating the length of predicted period (slide interval) is also converted into an int, but is used to generate two timestamps, one starting at the end of the twitter filter timestamp and the other with the value added to it. The default parameters used are “2018-10-11” , “08:00”, “3”, and “1” which will generate the filter seen in figure 1.



```
Checking Twitter data from 2018-10-11 08:00 to 2018-10-11 11:00
Comparing results on Yahoo from 2018-10-11 10:00 to 2018-10-11 12:00
```

Figure 1: A printout of the timestamp filter criteria. Note that the yahoo has the starting timestamp of 11:00 since we are using the closing price of that time as the opening price at hour 12:00.

These variables are furthermore used to call two functions, get_twitter_results and get_yahoo_finanace_intraday, and comparing the results of both of them. Finally, a printout to the terminal displays what type of sentiments correctly predicted the change in price based on their input dates as seen in the Results chapter.

Get_twitter_results

This function accepts the SparkSession and the timestamps used to filter the stream. The former recorded stream saved as a parquet file is read from a directory and registered as temporary table. Using spark sql, a new Dataframe (sqlDF) is created with Twitter timestamp casted to timestamp column. A filter is applied to select the data based on the timestamp parameters. Based on this dataframe, three other dataframes are created for each type of sentimental (main, average and weighted) by mapping the tuple index with the number 1. These frames are grouped by the sentiment and an aggregation function applied to sum all the numbers together, ordered from the highest appearing sentiment to the lowest. Finally the columns are renamed with appropriate naming and the highest sentiment of each type is collected and return to the main function.

Get_yahoo_finance_intraday

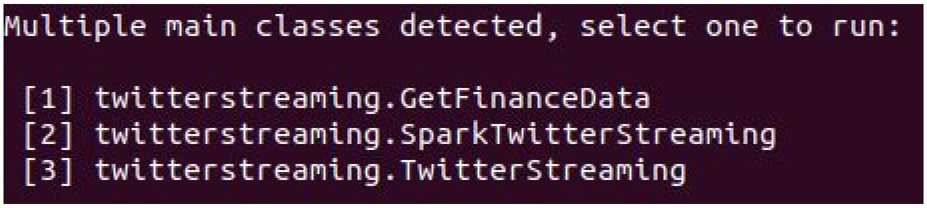
This function accepts the SparkSession, the two timestamps generated and a stock market company code as parameters. Additionally, it accepts the yahoo price interval format and how many days of data should be retrieved. The default value of those two are 1h and 7 days.

From these variables, a query string is constructed and used to retrieve a JSON string from the yahoo REST api. From this string, a Dataframe (df) is constructed and the columns traversed down to the closing price, volume and timestamps using the explode method provided for the df. Three columns are found and casted to arrays of appropriate type by mapping the elements to match the array type. These arrays are then zipped together and parallelized with new column header to construct a new Dataframe (newDF) that will be filtered according to the function input. After applying the filter to the newDF, the results collected represent the closing price at the start of the timestamp and the end of the timestamp. These collected variables are casted to doubles and the final closing price is subtracted from the starting one. Finally the function returns a string that’s either positive or negative, based on the change in price.

Note that the timestamp in tweets is in UTC, while it is in EDT in the yahoo data, so we add 4 hours time difference in the yahoo timestamp to convert it to UTC.

How to Run

Start a new terminal in the folder directory and run the commands “sbt package”, “sbt compile” to get the packages from sbt and compile the code. Next, run the command “sbt run”. Note that this will give two options after loading the project as seen in Figure 2.



```
Multiple main classes detected, select one to run:

 [1] twitterstreaming.GetFinanceData
 [2] twitterstreaming.SparkTwitterStreaming
 [3] twitterstreaming.TwitterStreaming
```

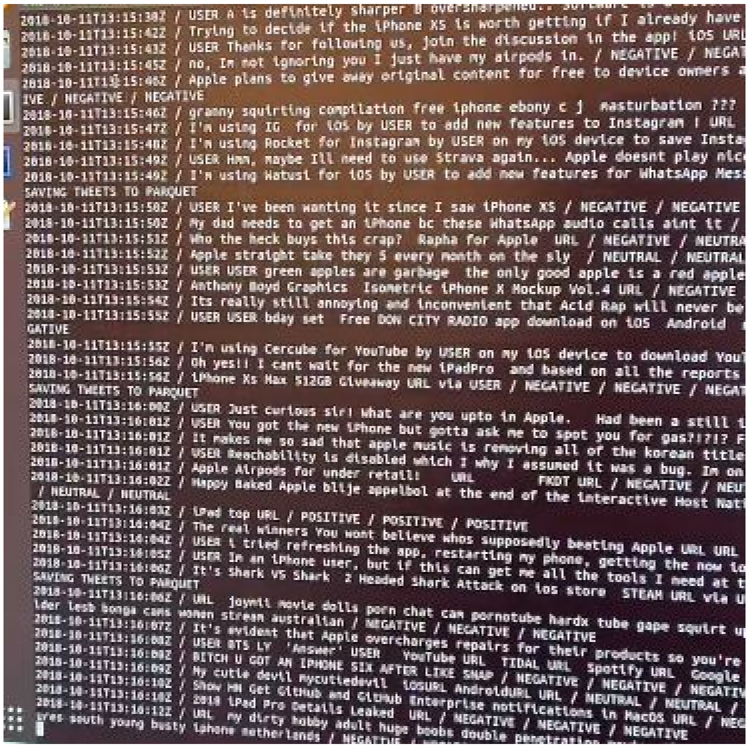
Figure 2: Available options of the project after loading.

By pressing 1, 2 or 3 and pressing Enter, either of the main classes in the project will run.

Selecting 1 will run the GetFinanceData main function, which will compare the results of the saved Twitter stream data with the Yahoo data available online. To run it with arguments, start sbt with following command: sbt "run <argument list>". After running the file, the terminal writes out the prediction of our analysis based on the tweets and the actual results. An example of the default printout can be seen in the results section.

Selecting 2 will run the SparkTwitterStreaming main function, which will listen to tweets related to Apple, filter and process them as described and saving them and their sentiments to a Parquet file. A printout of each tweet can be seen in the terminal as seen in Figure 3. And every hour, the predictions for this next hour will be printed separately for 3 different sentiment approaches as you can see in the Results chapter.

Selecting 3 will run the TwitterStreaming main function, which will listen to tweets related to Apple, filter and process them as described and saving them and their sentiments to a Parquet file. A printout of each tweet can be seen in the terminal as seen in Figure 3.

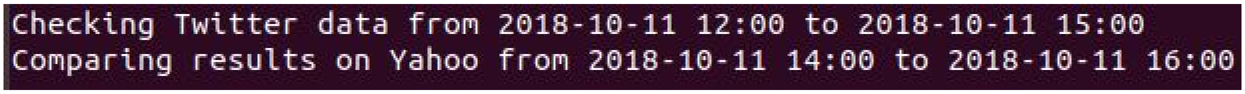


```
2018-10-11T13:15:38Z / USER A is definitely sharper & oversharpened.. Software is a
2018-10-11T13:15:42Z / Trying to decide if the iPhone XS is worth getting if I already have
2018-10-11T13:15:43Z / USER Thanks for following us, join the discussion in the app! iOS URL
2018-10-11T13:15:45Z / no, Im not ignoring you I just have my airpods in. / NEGATIVE / NEGAT
2018-10-11T13:15:46Z / Apple plans to give away original content for free to device owners a
IVE / NEGATIVE / NEGATIVE
2018-10-11T13:15:46Z / granny squirting compilation free iphone ebony c j  masturbation ???
2018-10-11T13:15:47Z / I'm using IG  for iOS by USER to add new features to Instagram ! URL
2018-10-11T13:15:48Z / I'm using Rocket for Instagram by USER on my iOS device to save Insta
2018-10-11T13:15:49Z / USER Hmm, maybe Ill need to use Strava again... Apple doesnt play nic
2018-10-11T13:15:49Z / I'm using Watusi for iOS by USER to add new features for WhatsApp Mes
SAVING TWEETS TO PARQUET
2018-10-11T13:15:50Z / USER I've been wanting it since I saw iPhone XS / NEGATIVE / NEGATIVE
2018-10-11T13:15:50Z / My dad needs to get an iPhone bc these WhatsApp audio calls aint it /
2018-10-11T13:15:51Z / Who the heck buys this crap?  Rapha for Apple  URL / NEGATIVE / NEUTRA
2018-10-11T13:15:52Z / Apple straight take they 5 every month on the sly  / NEUTRAL / NEUTRAL
2018-10-11T13:15:53Z / USER USER green apples are garbage  the only good apple is a red apple
2018-10-11T13:15:53Z / Anthony Boyd Graphics  Isometric iPhone X Mockup Vol.4 URL / NEGATIVE
2018-10-11T13:15:54Z / Its really still annoying and inconvenient that Acid Rap will never be
2018-10-11T13:15:55Z / USER USER bday set  Free DON CITY RADIO app download on iOS  Android
GATIVE
2018-10-11T13:15:55Z / I'm using Cercube for YouTube by USER on my iOS device to download You
2018-10-11T13:15:56Z / Oh yes!! I cant wait for the new iPadPro  and based on all the reports
2018-10-11T13:15:56Z / iPhone Xs Max 512GB Giveaway URL via USER / NEGATIVE / NEGATIVE / NEGAT
SAVING TWEETS TO PARQUET
2018-10-11T13:16:00Z / USER Just curious sir! What are you upto in Apple.   Had been a still
2018-10-11T13:16:01Z / USER You got the new iPhone but gotta ask me to spot you for gas?!?!? F
2018-10-11T13:16:01Z / It makes me so sad that apple music is removing all of the korean title
2018-10-11T13:16:01Z / USER Reachability is disabled which I why I assumed it was a bug. Im on
2018-10-11T13:16:01Z / Apple Airpods for under retail!    URL        FKDT URL / NEGATIVE / NEU
2018-10-11T13:16:02Z / Happy Baked Apple blije appelbol at the end of the interactive Host Nat
 / NEUTRAL / NEUTRAL
2018-10-11T13:16:03Z / iPad top URL / POSITIVE / POSITIVE / POSITIVE
2018-10-11T13:16:04Z / The real winners You wont believe whos supposedly beating Apple URL
2018-10-11T13:16:04Z / USER i tried refreshing the app, restarting my phone, getting the now lo
2018-10-11T13:16:05Z / USER Im an iPhone user, but if this can get me all the tools I need at t
2018-10-11T13:16:06Z / It's Shark VS Shark  2 Headed Shark Attack on ios store  STEAM URL via U
SAVING TWEETS TO PARQUET
2018-10-11T13:16:06Z / URL  joymii movie dolls porn chat cam pornotube hardx tube gape squirt u
lder lesb bonga cams women stream australian / NEGATIVE / NEGATIVE / NEGATIVE
2018-10-11T13:16:07Z / It's evident that Apple overcharges repairs for their products so you're
2018-10-11T13:16:08Z / USER BTS LY  'Answer' USER   YouTube URL  TIDAL URL  Spotify URL  Google
2018-10-11T13:16:08Z / BITCH U GOT AN IPHONE SIX AFTER LIKE SNAP / NEGATIVE / NEGATIVE / NEGATIV
2018-10-11T13:16:09Z / My cutie devil mycutiedevil  iOSURL AndroidURL URL / NEUTRAL / NEUTRAL /
2018-10-11T13:16:10Z / Show HN Get GitHub and GitHub Enterprise notifications in MacOS URL / NEG
2018-10-11T13:16:10Z / 2018 iPad Pro Details Leaked  URL / NEGATIVE / NEGATIVE / NEGATIVE
2018-10-11T13:16:12Z / URL  my dirty hobby adult huge boobs double penetration
yes south young busty iphone netherlands / NEGATIVE
```

Figure 3: The printout of the filtered twitter stream to the terminal, with the sentimental analysis. Notice that after every 10 tweets, a "SAVING TWEETS TO PARQUET" is printed out.

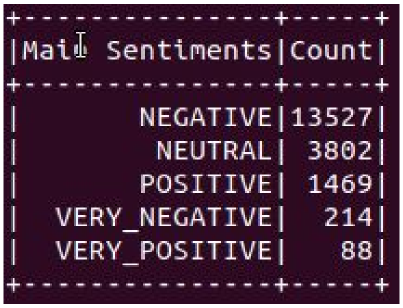
Results

The GetFinanceData class, ran with arguments "2018-10-11 12:00 3 1", after running TwitterStreaming.scala recording tweets from 12:00 UTC to 15:00 UTC to predict the stock movement from 15:00 UTC to 16:00 UTC, printed the following to the terminal.

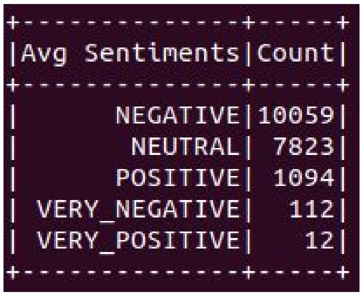


```
Checking Twitter data from 2018-10-11 12:00 to 2018-10-11 15:00
Comparing results on Yahoo from 2018-10-11 14:00 to 2018-10-11 16:00
```

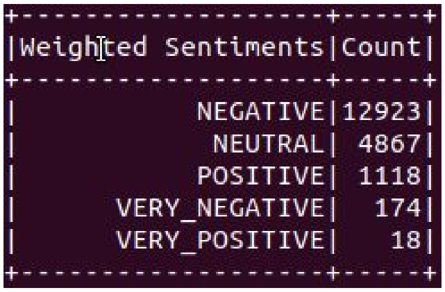
Continued, the get_twitter_results function reports the results of the sentimentals.



```
+---------------+-----+
|Main Sentiments|Count|
+---------------+-----+
|       NEGATIVE|13527|
|        NEUTRAL| 3802|
|       POSITIVE| 1469|
|  VERY_NEGATIVE|  214|
|  VERY_POSITIVE|   88|
+---------------+-----+
```

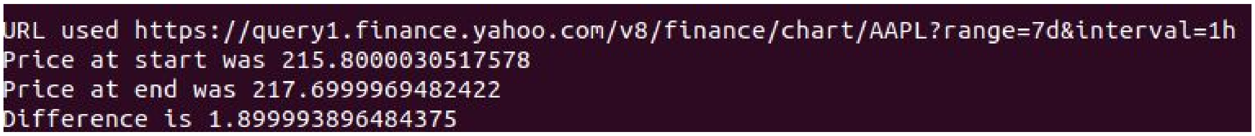


```
+--------------+-----+
|Avg Sentiments|Count|
+--------------+-----+
|      NEGATIVE|10059|
|       NEUTRAL| 7823|
|      POSITIVE| 1094|
| VERY_NEGATIVE|  112|
| VERY_POSITIVE|   12|
+--------------+-----+
```



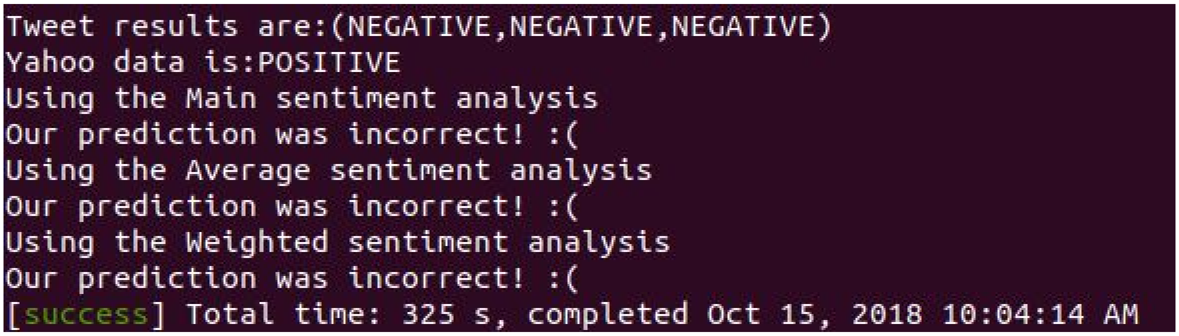
```
+-------------------+-----+
|Weighted Sentiments|Count|
+-------------------+-----+
|           NEGATIVE|12923|
|            NEUTRAL| 4867|
|           POSITIVE| 1118|
|      VERY_NEGATIVE|  174|
|      VERY_POSITIVE|   18|
+-------------------+-----+
```

Furthermore, the get_yahoo_finance_intraday printed out the url used to retrieve the data, displaying the change in price and returning negative or positive results



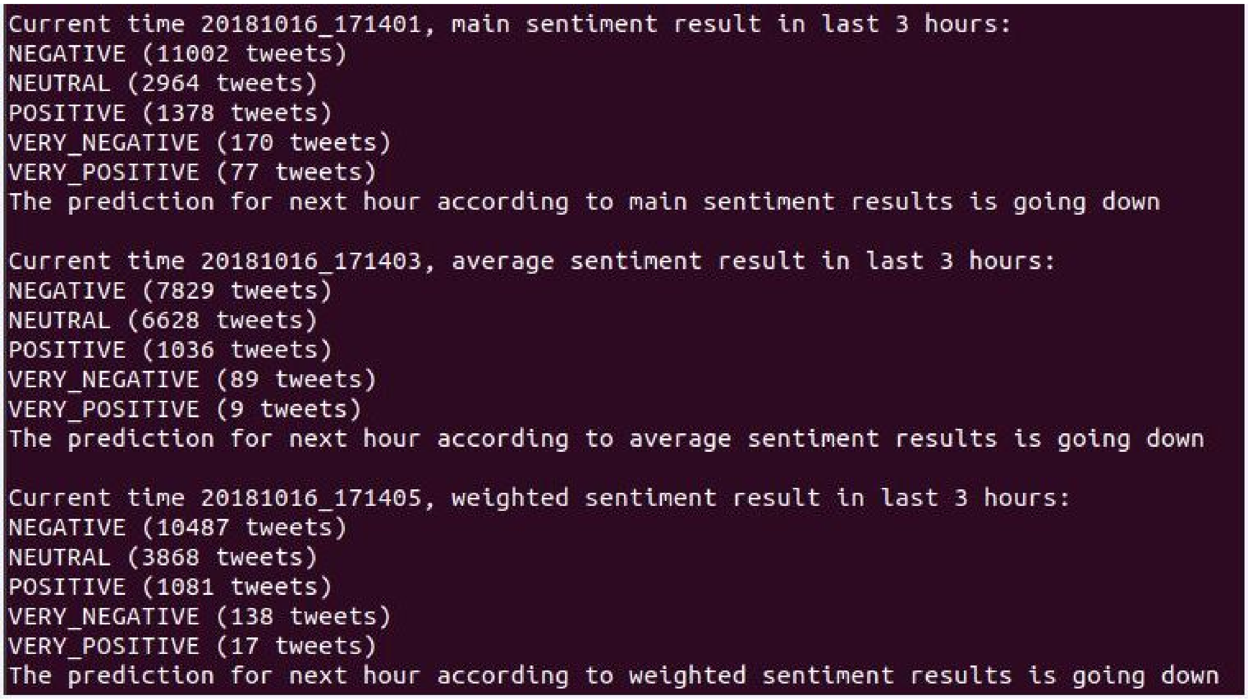
```
URL used https://query1.finance.yahoo.com/v8/finance/chart/AAPL?range=7d&interval=1h
Price at start was 215.8000030517578
Price at end was 217.6999969482422
Difference is 1.899993896484375
```

Finally, the main function compared the results of both functions and reported back on if analysis was indeed correct or not.



```
Tweet results are:(NEGATIVE,NEGATIVE,NEGATIVE)
Yahoo data is:POSITIVE
Using the Main sentiment analysis
Our prediction was incorrect! :(
Using the Average sentiment analysis
Our prediction was incorrect! :(
Using the Weighted sentiment analysis
Our prediction was incorrect! :(
[success] Total time: 325 s, completed Oct 15, 2018 10:04:14 AM
```

We also ran SparkTwitterStreaming.scala for more than 3 hours, and as expected, it used 3 hours tweets to predict stock movement for next hour and printed predictions in following format every hour:



```
Current time 20181016_171401, main sentiment result in last 3 hours:
NEGATIVE (11002 tweets)
NEUTRAL (2964 tweets)
POSITIVE (1378 tweets)
VERY_NEGATIVE (170 tweets)
VERY_POSITIVE (77 tweets)
The prediction for next hour according to main sentiment results is going down

Current time 20181016_171403, average sentiment result in last 3 hours:
NEGATIVE (7829 tweets)
NEUTRAL (6628 tweets)
POSITIVE (1036 tweets)
VERY_NEGATIVE (89 tweets)
VERY_POSITIVE (9 tweets)
The prediction for next hour according to average sentiment results is going down

Current time 20181016_171405, weighted sentiment result in last 3 hours:
NEGATIVE (10487 tweets)
NEUTRAL (3868 tweets)
POSITIVE (1081 tweets)
VERY_NEGATIVE (138 tweets)
VERY_POSITIVE (17 tweets)
The prediction for next hour according to weighted sentiment results is going down
```

Discussion for further improvements

As we can see from the result, the prediction was incorrect. It means the program can be improved to improve the accuracy. Due to the limited time in this project, we leave the improvements to the future.

Improvement on the Twitter filter or usage of NLP to filter

To analyze the stream, we use the products related to the company (macBook, iPhone) and of course Apple, the company name. However, as fall has started and the apple season with it, some tweets related to the fruit and the act of picking them slip through our filter. For example

“1st Home made Apple Pie of the Season #Apple #ApplePicking”

managed to slip through the stream and was classified as neutral. This did not have a major effect on our results, but we acknowledge that the project could be improved by enhancing the filter by filtering out tweets that had words like pies or picking. More advanced features could also be applied such as NLP context analysis but this would require way more expertise in machine learning.

Incorrection of the NLP

As the TwitterStreaming.scala and SparkTwitterStreaming.scala printed out tweets and their classified sentiments, we noticed that a dozen of tweets were incorrectly classified according to our judgement. For example, two comments received in the stream were classified as negative, even though we both agree that they are pretty positive towards Apple.

“Not going to lie, its a good phone but Ill never switch from away from iphone “

and

“Companies like Amazon, Apple, Walmart… etc are great companies and are not going away. Long term they will come back and come back fast I think”

All 3 different sentiment analysis approaches classified them as negative. After multiple attempts to tweak the sentimental analysis, we noticed that for some cases the sentiment changed from negative to positive as some words were changed from upper case to lower case. Capital letters seemed to affect the analysis, which requires further research.

Considering more factors

Twitter is a social network where people can follow or be followed by others. People's influence depends on how many followers they have. Considering this, we can introduce a weight mechanism on the tweets, i.e. the more followers a person has, the more his/her tweets affect our prediction. Also a company's stock movement is not only decided by tweets about it and its products, also it may depend on the news regarding its competitors, weather which can affect stock traders' emotion, global economy situation and so on. We can improve the accuracy by considering more factors.

Tuning the window size and sliding interval

After above improvements done, we can then tune the window size and sliding interval to find the ones that provide best accuracy.

Conclusions

Data-Intensive Computing is a great course where we have learnt lots of softwares that can be used to store and process different data in big amount. With this project, we got more familiar with Spark Streaming and Spark SQL, and we learnt how to access Twitter Streaming API with Twitter4j Library, how to do sentiment analysis with Stanford CoreNLP, and how to access Yahoo Finance Data with its REST API.