



I did the horizontal bar on kibana with y-axis as the value of sentiment which ranges from 0-4(0 means that it is very negative,1 means that it is negative, 2 means that it is neutral, 3 means that it is positive, 4 means that it is very positive. I analysed the sentiment(Y-axis) with number of tweets count(in X-axis) and I did the analysis for the last 15 minutes .So, As it is live streaming as real time data streaming so the plots vary but most of the time.

I observed that most of the tweets are negative and very less number of tweets like 2% are very negative and nearly 0.1% are very positive. I analyzed the sentiment for last 15 minutes in first visualization and last 1 week in second visualization.

I runned the command

**spark-submit --class streaming.TwitterProducer /Users/likhithakommineni/Downloads/SparkStreamingTwitterExample/target/twitter-1.0-jar-with-dependencies.jar /Users/likhithakommineni/Downloads/SparkStreamingTwitterExample/src/main/resources/twitter.txt twitter "iphone"**

to start my application

Basically the application requires minimum of 3 arguments .(the 3 arguments are the file having the twitter credentials as the credentials written in application.properties is not working.So,I wrote a file and used that to authenticate .you can pass that path from command line having the file in format of

consumerkey \*\*\*\*

consumerSecret \*\*\*\*

accessToken \*\*\*\*\*

accessTokenSecret \*\*\*\*\*\*.

There should be one space between key and value and every key value pair should be in new line.

Second argument is kafka topic . Here it is twitter in above example. after that all other arguments are the keywords u can pass to filter the tweets you get from twitter (example: “iphone” , “technology”, “innovation”)

Spark-submit –class streaming.TwitterProducer(class you want to run in scala) /Users/likhithakommineni/Downloads/SparkStreamingTwitterExample/target/twitter-1.0-jar-with-dependencies.jar(the output generated jar file path) /Users/likhithakommineni/Downloads/SparkStreamingTwitterExample/src/main/resources/twitter.txt(this is the path of the twitter credentials file which has the authentication credentials for logging to your path) twitter(kafka topic which is mandatory) “iphone”(the keywords that you want to filter the tweet with in twitter)

You can give as many keywords as possible as arguments to filter the twitter data with

After running the application the kafka producer is started. So , no we need to start

Confluent by going to the confluent folder and it can be started easily by going to the bin subdirectory and running the following command:

./confluent start schema-registry

Nowyou need to go to the kafka folder in your Mac or windows and then run command

bin/kafka-topics.sh --list --zookeeper localhost:2181

to verify the kafkatopic in list

the kafkatopic is topic you gave while running the spark application above like twitter in example (**spark-submit --class streaming.TwitterProducer /Users/likhithakommineni/Downloads/SparkStreamingTwitterExample/target/twitter-1.0-jar-with-dependencies.jar /Users/likhithakommineni/Downloads/SparkStreamingTwitterExample/src/main/resources/twitter.txt twitter "iphone"** )

Now you can run the below command

bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic test --from-beginning

to see the messages you receive from producer.

Visualizing the data using Elasticsearch and Kibana

Elasticsearch using the following command in the elasticsearch-5.6.8/bin directory

Run command

**./elasticsearch**

Kibana using the following command in the kibana-5.6.8-darwin-x86\_64/bin directory:

Run command

**./kibana**

Logstash: Go to the logstash-5.6.8 directory and create a file logstash-simple.conf with following content:

input {

kafka {

bootstrap\_servers => "localhost:9092"

topics => ["twitter"]

key\_deserializer\_class => "org.apache.kafka.common.serialization.StringDeserializer"

value\_deserializer\_class => "org.apache.kafka.common.serialization.IntegerDeserializer"

}

}

output {

elasticsearch {

hosts => ["localhost:9200"]

index => "sentiment"

}

}Then run the following command

bin/logstash -f logstash-simple.conf

This sets up the right pipeline between Kafka and Elasticsearch.

If everything is set up properly, you should be able to go to <http://localhost:5601>

Here u can go to management and create index pattern with name as sentiment

Now go to visualization tab . I selected the horizontal bar visualization. There on X-axis u can add aggregation and select filters where u can select filters by message field on X-axis(giving filters as message is 1,message is 2, message is 3, message is 4, message is 0) here message indicates the sentiment of the tweets and u can aggregate by the count where you can see the output .The count of tweets is on X-axis and the sentiment is on Y-axis.



