# Analyzing Responses of Live Events Through Social Media Data

**KEDAR DEODHAR**

**JANVI PATEL**

**PRERAK TRIVEDI**

Date: April 17, 2020

—

Subject: Big Data

—

Professor Zeinab Noorian

## Contents

## 1. PROBLEM STATEMENT:

Enabling the real time analysis using Spark to obtain visualized data format and sentiments of the data fetched from the social media websites (specially YouTube and twitter) using python scripting language to get an overview of the ideas perceived by the users in regards to the context of the video or tweet.

## 2. MOTIVATION OF OUR WORK:

Our work is inspired by the websites and data graphs created by Google during the COVID-19 pandemic. This website provided live analysis of the cases being tested and their results on a live basis with the precision of a second. We thought that this idea would be of a great help to the streamers as well as viewers to get an idea on the real time basis to understand what the review is. Although there are comments that are visible which rain like cats and dogs, it is difficult to get an overall idea of the sentiments. Hence, it would be really an advantage to have something of this sort at hand.

As mentioned above, it can be of great assistance for the viewers and streamers. The streams can understand if the content being streamed is being reacted by hatred or is soothing to their viewers. It can be also used to provide an explanation for some misunderstandings that can occur due to the falsified information spread earlier or the misconceptions created in viewers mind. The viewers can get an overall idea of the emotions of many of the users who are watching the video. The user can think for himself for the reason of the conflict of ideas, if there is any, between the crowd and themselves.

Concluding, we expect the outcome of our project to have a great impact on the society. If taken positively, this can be helpful in rationalization and linearization of thoughts of the viewers and for the streamers to create and post a content that suffices the users' interest and prevent any misconceptions being created in users mind collectively.

## 3. APPROACH OVERVIEW:

We divided the problem statement in three parts, which are, streaming, storage and analytics. In the first part of the project, the data is streamed from two social media websites, naming, YouTube and Twitter. The storage is done using MongoDB and Spark for analytics. We will discuss all the three tasks in the below paragraphs. Please note that most of the coding has been carried out in Python scripting language. The tasks performed by us in this project are:

1) Streaming Twitter and YouTube data using APIs.
2) Sentiment Analysis in Python.
3) Stored data in MongoDB using PyMongo.
4) Mongo Spark Connector to connect spark and MongoDB.
5) Predictive, Aggregation and Visual Analysis in Spark using Databricks.

Sometimes people come across a question of preference where we must choose between the APIs and databases. Undoubtedly, the static file extensions (like CSV) are easy to use and store but the core idea of our project lied in the dynamic dataset which are susceptible to changes. The streaming concept and dynamic data could only be satisfied by the APIs hence the use of YouTube and twitter APIs.

## DATA COLLECTION:

For the data collection, we selected two sources (YouTube and Twitter). Initially this project was designed keeping in mind only YouTube but when we tried pulling data a couple of times, the YouTube API started restraining the free data collection. Upon research, we came to know that the YouTube API worked on two conditions:

- One call collects only 50 rows of information for free.
- Each row had a unique user

So, our idea to suffice the interest of our project, we decided to apply our idea on Twitter as well. The only difference that we could face when applying the project for the other websites would be the difference in name of columns and data collection patterns which could be managed by obtaining basic knowledge of python coding.

## DATA STORAGE:

The streamed data collected by APIs was stored in the MongoDB. We have learnt about three different storage technologies, MongoDB, DynamoDB and Cassandra in our lectures. We selected MongoDB as:

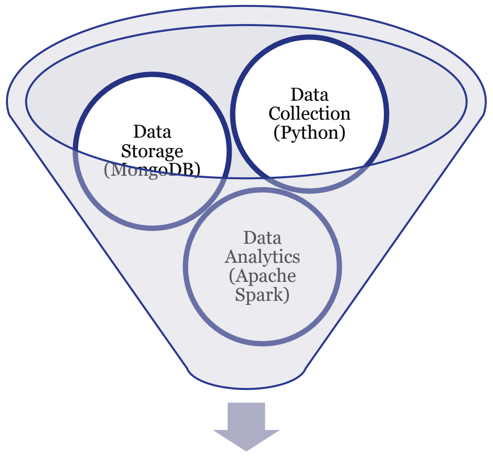
- It is very well connected with the Spark and is quite fast.
- The data shuffling between the structures is minimum.
- Faster insight for businesses with low cost maintenance and risk.
- The scaling and aggregation functions are very powerful.
- Reduction in latency.

Not only this, but a lot of the industrial applications also exist with the same combination of powerful analytics software (Apache Spark) and efficient database system (MongoDB). Moreover, we planned on using the assistance from the DataBricks[1] in working on Spark which strengthened our decision.

## DATA ANALYTICS:

Apache Spark can be considered as the sole technology that completely crushed most of the Big Data Analytics technologies that existed. The lectures conducted by the Professor pointed out two main advantages of the Apache Spark which did not let a single doubt arise in our mind about choosing this technology for the big data. The first advantage is the speed of Spark because of its in-memory data engine and the other advantage is its ease of use for the users.

[1] https://databricks.com/



Project

Furthermore, the support provided by the Apache Spark for the machine learning (MLib) and visualizations of the data from the SQL query is awesome. We have used a ready to use instance of Spark from the databricks so that we can have code as well as the output simultaneously.

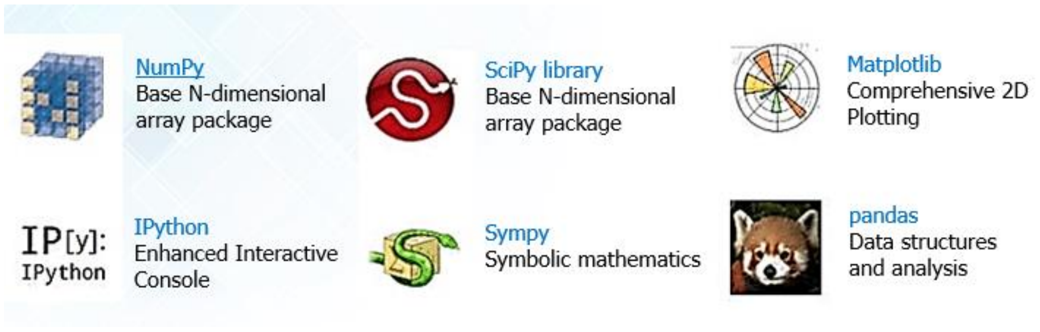
## 4. FRAMEWORKS:

This part of the report provides a detailed view of the frameworks (technologies) that we have used. We have covered their basics, explanation and our understanding of the requirement of the specific technologies.

### PYTHON:

Python is the main scripting language that we have used for our analysis. Although, this language does not require any explanation for its usage in the field of Data Analysis, we have tried to provide a summary of the ideas that we processed.

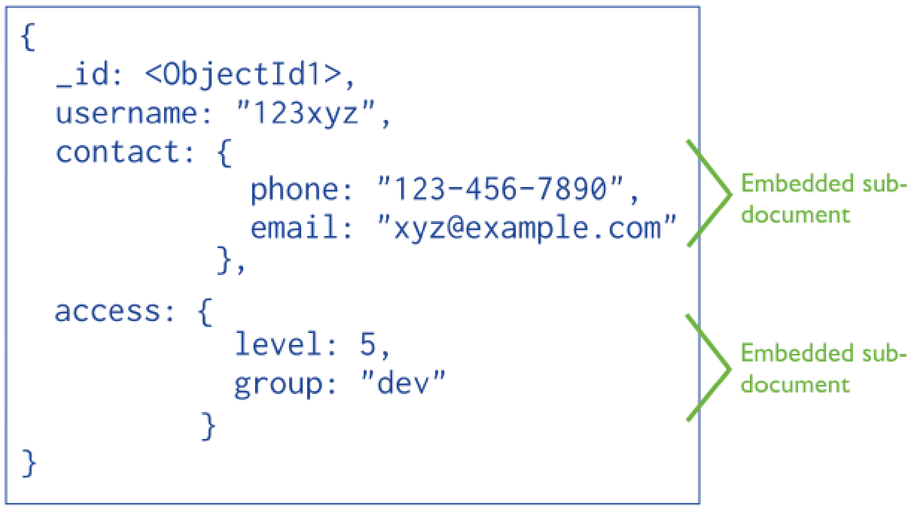
The first and the foremost advantage of the language usage is that it is open source which means that we can have unlimited help. The second reason is it's powerful libraries like Pandas, NumPy which ease the use of data frames, data frame operations along with logical and mathematical applications. A glimpse of the libraries which are widely used is provided below.


NumPy
Base N-dimensional array package
SciPy library
Base N-dimensional array package
Matplotlib
Comprehensive 2D Plotting
IP[y]: IPython
IPython
Enhanced Interactive Console
Sympy
Symbolic mathematics
pandas
Data structures and analysis

## MONGODB:

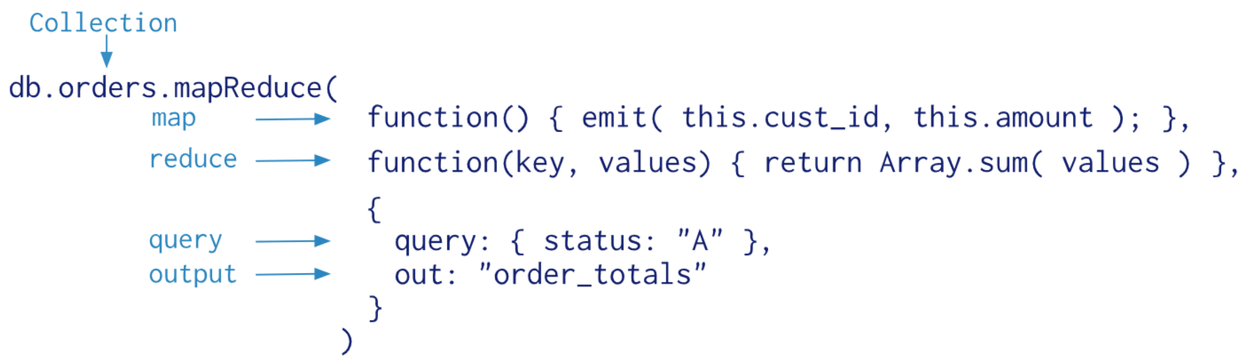
MongoDB is a database storage system that stores that data in something which is like JSON document. We have used python version 3 to perform all our coding. Python connection to MongoDB is possible using the pymongo connector which allows interaction between the MongoDB and Python. We have performed all our tasks on the AWS EC2 machine.

As mentioned earlier, the MongoDB stores data in JSON like documents whose structure is mentioned in the diagram below. This contains the information of a single data whose id, username is in single braces of a document, but the contact details contain multiple sets of information hence it is under another curly braces which means it is further embedded in the document. In short as the subset of the dataset increases in the input, it is filed by as a subset.



```
{
  _id: <ObjectId1>,
  username: "123xyz",
  contact: {
              phone: "123-456-7890",
              email: "xyz@example.com"
           },
  access: {
             level: 5,
             group: "dev"
          }
}
```

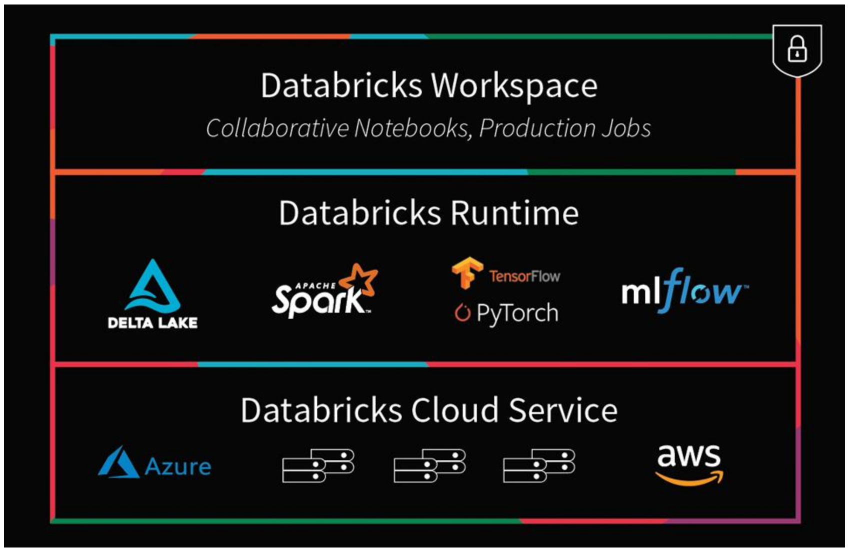
We have used the MapReduce feature for aggregations in the MongoDB. These aggregations helped us to produce a concise and understandable output. Like other aggregation operations, map-reduce can specify a query condition to select the input documents as well as sort and limit the results.



```
db.orders.mapReduce(
                     function() { emit( this.cust_id, this.amount ); },
                     function(key, values) { return Array.sum( values ) },
                     {
                       query: { status: "A" },
                       out: "order_totals"
                     }
                   )
```

## SPARK:

Apache Spark is one of the very few big data processing platforms that can perform tasks on big datasets. It can also distribute the tasks autonomously or using the computing tools and rules defined by the user. The ease of coding is available in Spark using the APIs which enhance the coding experience by enabling the user to use the language they like for coding and the API converts it in Spark Understandable Language. These APIs support most of the languages like Python, R, Java and Scala. Moreover, Apache has found its place in many multinational companies which are the top league players in the field of Big Data.

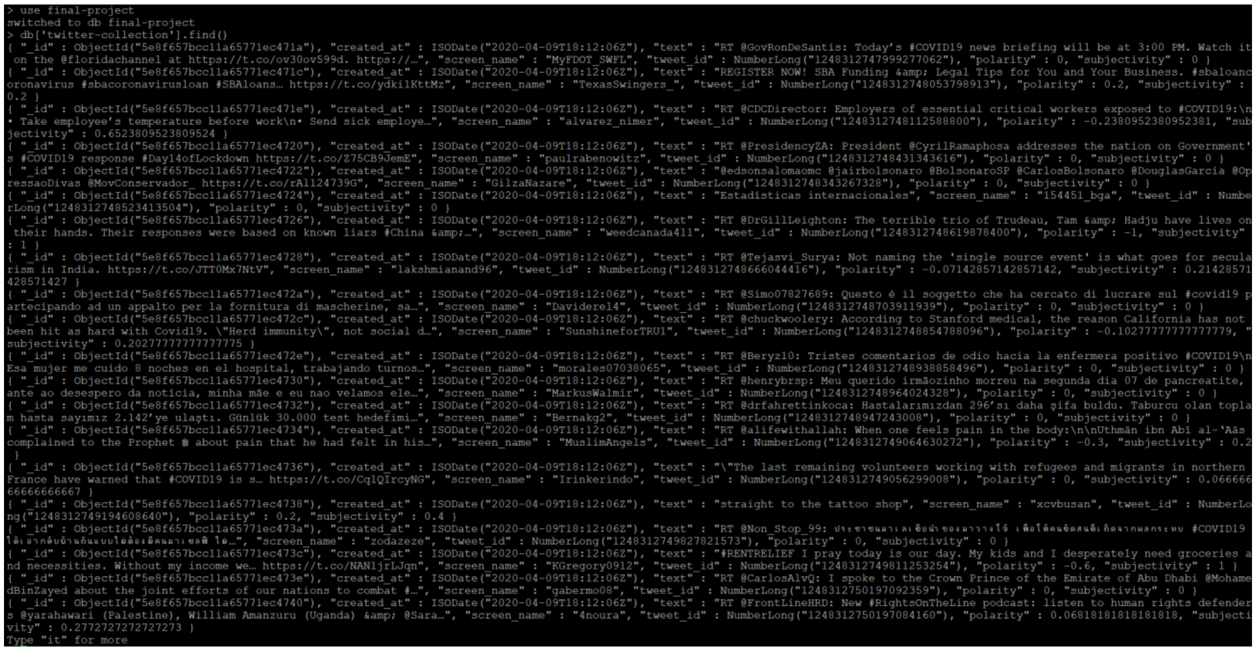


The application of Apache spark is divided in two main frameworks: a driver and an executor. The driver converts the user code into chunks which is distributed as tasks to various nodes whereas the executor keeps an eye on the nodes and directs them to execute the distributed tasks. A mediator which can be thought as cluster is necessary between the two components. In our case, we have used the Data Bricks instance for Spark.

## 5. IMPLEMENTATION:

Python is one of the most common languages used for the streaming and data analysis. The main advantages of python are its libraries and cross platform connectors that help us perform most of the tasks with ease. We have used pymongo, pyspark and YouTube API from Google developer. The implementation of ideas carried out:

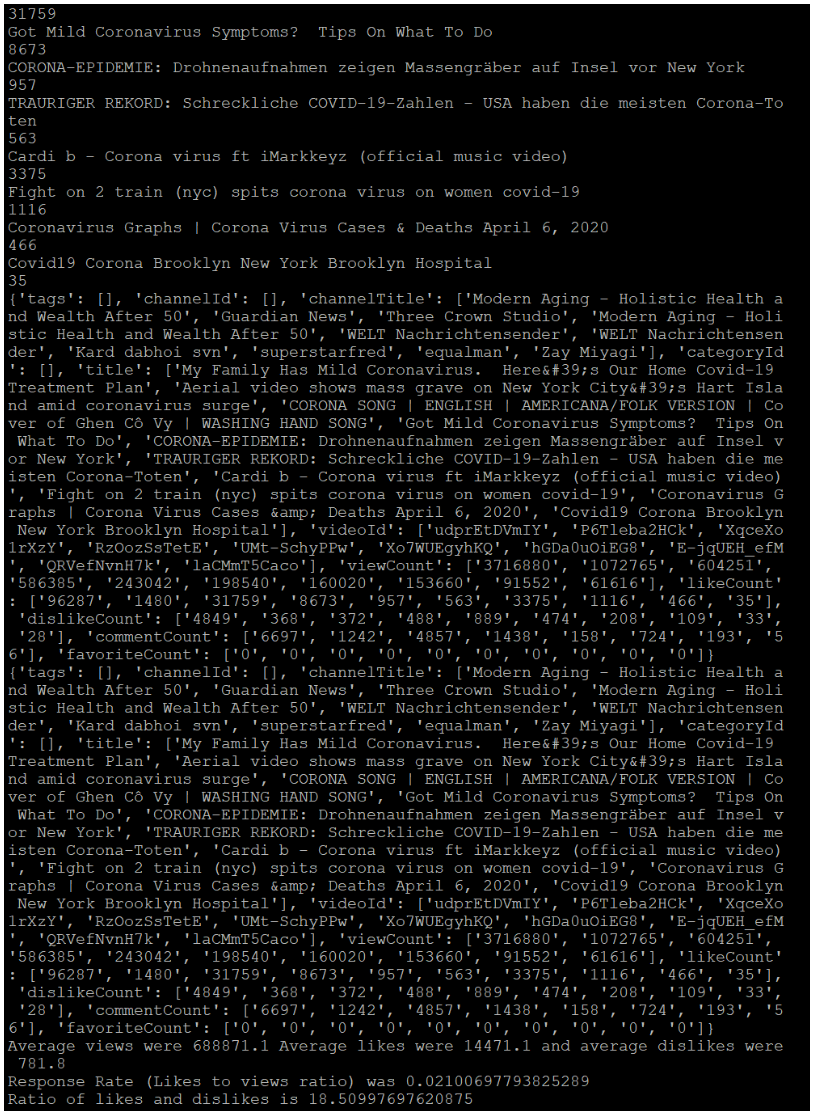
Firstly, we had created a database in Mongo DB and then using python scripting nano editor we have streamed the data by using YouTube and twitter API. The API which we had used are “Text Blob” and “API client” for twitter and YouTube respectively. Secondly, in the code we had included the sentiment analysis in the form of subjectivity and polarity. Subjectivity refers to normalized value and tells us that whether the emotions are generalized or private view. Polarity refers to negative emotions and positive emotions in standardized values which means that negative value will refer to negative emotion and positive value will refer to positive emotion. Moreover, a higher value would mean that emotion is very strong. The figure 1 provides a glimpse of the twitter data that we pulled which was around 100,000.



```
> use final-project
switched to db final-project
> db['twitter-collection'].find()
{ "_id" : ObjectId("5e8f657bcc11a65771ec471a"), "created_at" : ISODate("2020-04-09T18:12:06Z"), "text" : "RT @GovRonDeSantis: Today’s #COVID19 news briefing will be at 3:00 PM. Watch it
 on the @floridachannel at https://t.co/ov30ov599d. https://…", "screen_name" : "MyFDOT_SWFL", "tweet_id" : NumberLong("1248312747999277062"), "polarity" : 0, "subjectivity" : 0 }
{ "_id" : ObjectId("5e8f657bcc11a65771ec471c"), "created_at" : ISODate("2020-04-09T18:12:06Z"), "text" : "REGISTER NOW! SBA Funding &amp; Legal Tips for You and Your Business. #sbaloanc
oronavirus #sbacoronavirusloan #SBAloans… https://t.co/ydkilKttMz", "screen_name" : "TexasSwingers_", "tweet_id" : NumberLong("1248312748053798913"), "polarity" : 0.2, "subjectivity" :
0.2 }
{ "_id" : ObjectId("5e8f657bcc11a65771ec471e"), "created_at" : ISODate("2020-04-09T18:12:06Z"), "text" : "RT @CDCDirector: Employers of essential critical workers exposed to #COVID19:\n
• Take employee’s temperature before work\n• Send sick employe…", "screen_name" : "alvarez_nimer", "tweet_id" : NumberLong("1248312748112588800"), "polarity" : -0.2380952380952381, "sub
jectivity" : 0.6523809523809524 }
{ "_id" : ObjectId("5e8f657bcc11a65771ec4720"), "created_at" : ISODate("2020-04-09T18:12:06Z"), "text" : "RT @PresidencyZA: President @CyrilRamaphosa addresses the nation on Government'
s #COVID19 response #Day14ofLockdown https://t.co/Z75CB9JemE", "screen_name" : "paulrabenowitz", "tweet_id" : NumberLong("1248312748431343616"), "polarity" : 0, "subjectivity" : 0 }
{ "_id" : ObjectId("5e8f657bcc11a65771ec4722"), "created_at" : ISODate("2020-04-09T18:12:06Z"), "text" : "@edsonsalomaomc @jairbolsonaro @BolsonaroSP @CarlosBolsonaro @DouglasGarcia @Op
ressaoDivas @MovConservador_ https://t.co/rAl124739G", "screen_name" : "GilzaNazare", "tweet_id" : NumberLong("1248312748343267328"), "polarity" : 0, "subjectivity" : 0 }
{ "_id" : ObjectId("5e8f657bcc11a65771ec4724"), "created_at" : ISODate("2020-04-09T18:12:06Z"), "text" : "Estadísticas internacionales", "screen_name" : "154451_bga", "tweet_id" : Numbe
rLong("1248312748523413504"), "polarity" : 0, "subjectivity" : 0 }
{ "_id" : ObjectId("5e8f657bcc11a65771ec4726"), "created_at" : ISODate("2020-04-09T18:12:06Z"), "text" : "RT @DrGillLeighton: The terrible trio of Trudeau, Tam &amp; Hadju have lives on
 their hands. Their responses were based on known liars #China &amp;…", "screen_name" : "weedcanada411", "tweet_id" : NumberLong("1248312748619878400"), "polarity" : -1, "subjectivity"
: 1 }
{ "_id" : ObjectId("5e8f657bcc11a65771ec4728"), "created_at" : ISODate("2020-04-09T18:12:06Z"), "text" : "RT @Tejasvi_Surya: Not naming the 'single source event' is what goes for secula
rism in India. https://t.co/JTT0Mx7NtV", "screen_name" : "lakshmianand96", "tweet_id" : NumberLong("1248312748666044416"), "polarity" : -0.07142857142857142, "subjectivity" : 0.21428571
428571427 }
{ "_id" : ObjectId("5e8f657bcc11a65771ec472a"), "created_at" : ISODate("2020-04-09T18:12:06Z"), "text" : "RT @Simo07827689: Questo è il soggetto che ha cercato di lucrare sul #covid19 p
artecipando ad un appalto per la fornitura di mascherine, sa…", "screen_name" : "Daviderel4", "tweet_id" : NumberLong("1248312748703911939"), "polarity" : 0, "subjectivity" : 0 }
{ "_id" : ObjectId("5e8f657bcc11a65771ec472c"), "created_at" : ISODate("2020-04-09T18:12:06Z"), "text" : "RT @chuckwoolery: According to Stanford medical, the reason California has not
been hit as hard with Covid19. \"Herd immunity\", not social d…", "screen_name" : "SunshineforTRU1", "tweet_id" : NumberLong("1248312748854788096"), "polarity" : -0.10277777777777779, "
subjectivity" : 0.20277777777777775 }
{ "_id" : ObjectId("5e8f657bcc11a65771ec472e"), "created_at" : ISODate("2020-04-09T18:12:06Z"), "text" : "RT @Beryz10: Tristes comentarios de odio hacia la enfermera positivo #COVID19\n
Esa mujer me cuido 8 noches en el hospital, trabajando turnos…", "screen_name" : "morales07038065", "tweet_id" : NumberLong("1248312748938858496"), "polarity" : 0, "subjectivity" : 0 }
{ "_id" : ObjectId("5e8f657bcc11a65771ec4730"), "created_at" : ISODate("2020-04-09T18:12:06Z"), "text" : "RT @henrybrsp: Meu querido irmãozinho morreu na segunda dia 07 de pancreatite,
ante ao desespero da notícia, minha mãe e eu nao velamos ele…", "screen_name" : "MarkusWalmir", "tweet_id" : NumberLong("1248312748964024328"), "polarity" : 0, "subjectivity" : 0 }
{ "_id" : ObjectId("5e8f657bcc11a65771ec4732"), "created_at" : ISODate("2020-04-09T18:12:06Z"), "text" : "RT @drfahrettinkoca: Hastalarımızdan 296’sı daha şifa buldu. Taburcu olan topla
m hasta sayımız 2.142’ye ulaştı. Günlük 30.000 test hedefimi…", "screen_name" : "Bernakg2", "tweet_id" : NumberLong("1248312748947243008"), "polarity" : 0, "subjectivity" : 0 }
{ "_id" : ObjectId("5e8f657bcc11a65771ec4734"), "created_at" : ISODate("2020-04-09T18:12:06Z"), "text" : "RT @alifewithallah: When one feels pain in the body:\n\nUthmān ibn Abī al-‘Aās
complained to the Prophet ﷺ about pain that he had felt in his…", "screen_name" : "MuslimAngels", "tweet_id" : NumberLong("1248312749064630272"), "polarity" : -0.3, "subjectivity" : 0.2
 }
{ "_id" : ObjectId("5e8f657bcc11a65771ec4736"), "created_at" : ISODate("2020-04-09T18:12:06Z"), "text" : "\"The last remaining volunteers working with refugees and migrants in northern
France have warned that #COVID19 is s… https://t.co/CqlQIrcyNG", "screen_name" : "Irinkerindo", "tweet_id" : NumberLong("1248312749056299008"), "polarity" : 0, "subjectivity" : 0.066666
66666666667 }
{ "_id" : ObjectId("5e8f657bcc11a65771ec4738"), "created_at" : ISODate("2020-04-09T18:12:06Z"), "text" : "straight to the tattoo shop", "screen_name" : "xcvbusan", "tweet_id" : NumberLo
ng("1248312749194608640"), "polarity" : 0.2, "subjectivity" : 0.4 }
{ "_id" : ObjectId("5e8f657bcc11a65771ec473a"), "created_at" : ISODate("2020-04-09T18:12:06Z"), "text" : "RT @Non_Stop_99: ประชาชนมาเลเซียนำของมาวางไว้ เพื่อให้คนขัดสนที่เดือดร้อนจากผลกระทบ #COVID19
ได้เอากลับบ้านกันแบบไม่ต้องมีคนมาเซลฟี่ ไอ…", "screen_name" : "zodazeze", "tweet_id" : NumberLong("1248312749827821573"), "polarity" : 0, "subjectivity" : 0 }
{ "_id" : ObjectId("5e8f657bcc11a65771ec473c"), "created_at" : ISODate("2020-04-09T18:12:06Z"), "text" : "#RENTRELIEF I pray today is our day. My kids and I desperately need groceries a
nd necessities. Without my income we… https://t.co/NANljrLJqn", "screen_name" : "KGregory0912", "tweet_id" : NumberLong("1248312749811253254"), "polarity" : -0.6, "subjectivity" : 1 }
{ "_id" : ObjectId("5e8f657bcc11a65771ec473e"), "created_at" : ISODate("2020-04-09T18:12:06Z"), "text" : "RT @CarlosAlvQ: I spoke to the Crown Prince of the Emirate of Abu Dhabi @Mohame
dBinZayed about the joint efforts of our nations to combat #…", "screen_name" : "gabermo08", "tweet_id" : NumberLong("1248312750197092359"), "polarity" : 0, "subjectivity" : 0 }
{ "_id" : ObjectId("5e8f657bcc11a65771ec4740"), "created_at" : ISODate("2020-04-09T18:12:06Z"), "text" : "RT @FrontLineHRD: New #RightsOnTheLine podcast: listen to human rights defender
s @yarahawari (Palestine), William Amanzuru (Uganda) &amp; @Sara…", "screen_name" : "4noura", "tweet_id" : NumberLong("1248312750197084160"), "polarity" : 0.06818181818181818, "subjecti
vity" : 0.2772727272727273 }
Type "it" for more
```

*Figure 1: Twitter Data*

Now, for visualizing and forming function we had used pyspark by installing it in AWS instance and performing some aggregations in it. We successfully done the aggregation part, but our effort lagged in visualization part. So, we decided to take that part in data bricks and fetch the mongo data into data bricks pyspark. Moreover, we have used Mongo Spark Connector to connect spark and MongoDB and successfully fetched the streamed data into pyspark. To read the data we had used “spark.read” which stores the data in “pysaprk.sql.dataframe” format which allows easy manipulation of the data, since it can be used as simple pandas data. We used “format” function to specify the type of source files and we used “option” method to specify the URL(in our case it was the ip address of our AWS ec2 instance) from where the data is to be fetched.



```
31759
Got Mild Coronavirus Symptoms?  Tips On What To Do
8673
CORONA-EPIDEMIE: Drohnenaufnahmen zeigen Massengräber auf Insel vor New York
957
TRAURIGER REKORD: Schreckliche COVID-19-Zahlen - USA haben die meisten Corona-To
ten
563
Cardi b - Corona virus ft iMarkkeyz (official music video)
3375
Fight on 2 train (nyc) spits corona virus on women covid-19
1116
Coronavirus Graphs | Corona Virus Cases & Deaths April 6, 2020
466
Covid19 Corona Brooklyn New York Brooklyn Hospital
35
{'tags': [], 'channelId': [], 'channelTitle': ['Modern Aging - Holistic Health a
nd Wealth After 50', 'Guardian News', 'Three Crown Studio', 'Modern Aging - Holi
stic Health and Wealth After 50', 'WELT Nachrichtensender', 'WELT Nachrichtensen
der', 'Kard dabhoi svn', 'superstarfred', 'equalman', 'Zay Miyagi'], 'categoryId
': [], 'title': ['My Family Has Mild Coronavirus.  Here&#39;s Our Home Covid-19
Treatment Plan', 'Aerial video shows mass grave on New York City&#39;s Hart Isla
nd amid coronavirus surge', 'CORONA SONG | ENGLISH | AMERICANA/FOLK VERSION | Co
ver of Ghen Cô Vy | WASHING HAND SONG', 'Got Mild Coronavirus Symptoms?  Tips On
 What To Do', 'CORONA-EPIDEMIE: Drohnenaufnahmen zeigen Massengräber auf Insel v
or New York', 'TRAURIGER REKORD: Schreckliche COVID-19-Zahlen - USA haben die me
isten Corona-Toten', 'Cardi b - Corona virus ft iMarkkeyz (official music video)
', 'Fight on 2 train (nyc) spits corona virus on women covid-19', 'Coronavirus G
raphs | Corona Virus Cases &amp; Deaths April 6, 2020', 'Covid19 Corona Brooklyn
 New York Brooklyn Hospital'], 'videoId': ['udprEtDVmIY', 'P6Tleba2HCk', 'XqceXo
1rXzY', 'RzOozSsTetE', 'UMt-SchyPPw', 'Xo7WUEgyhKQ', 'hGDa0uOiEG8', 'E-jqUEH_efM
', 'QRVefNvnH7k', 'laCMmT5Caco'], 'viewCount': ['3716880', '1072765', '604251',
'586385', '243042', '198540', '160020', '153660', '91552', '61616'], 'likeCount'
: ['96287', '1480', '31759', '8673', '957', '563', '3375', '1116', '466', '35'],
 'dislikeCount': ['4849', '368', '372', '488', '889', '474', '208', '109', '33',
 '28'], 'commentCount': ['6697', '1242', '4857', '1438', '158', '724', '193', '5
6'], 'favoriteCount': ['0', '0', '0', '0', '0', '0', '0', '0', '0', '0']}
{'tags': [], 'channelId': [], 'channelTitle': ['Modern Aging - Holistic Health a
nd Wealth After 50', 'Guardian News', 'Three Crown Studio', 'Modern Aging - Holi
stic Health and Wealth After 50', 'WELT Nachrichtensender', 'WELT Nachrichtensen
der', 'Kard dabhoi svn', 'superstarfred', 'equalman', 'Zay Miyagi'], 'categoryId
': [], 'title': ['My Family Has Mild Coronavirus.  Here&#39;s Our Home Covid-19
Treatment Plan', 'Aerial video shows mass grave on New York City&#39;s Hart Isla
nd amid coronavirus surge', 'CORONA SONG | ENGLISH | AMERICANA/FOLK VERSION | Co
ver of Ghen Cô Vy | WASHING HAND SONG', 'Got Mild Coronavirus Symptoms?  Tips On
 What To Do', 'CORONA-EPIDEMIE: Drohnenaufnahmen zeigen Massengräber auf Insel v
or New York', 'TRAURIGER REKORD: Schreckliche COVID-19-Zahlen - USA haben die me
isten Corona-Toten', 'Cardi b - Corona virus ft iMarkkeyz (official music video)
', 'Fight on 2 train (nyc) spits corona virus on women covid-19', 'Coronavirus G
raphs | Corona Virus Cases &amp; Deaths April 6, 2020', 'Covid19 Corona Brooklyn
 New York Brooklyn Hospital'], 'videoId': ['udprEtDVmIY', 'P6Tleba2HCk', 'XqceXo
1rXzY', 'RzOozSsTetE', 'UMt-SchyPPw', 'Xo7WUEgyhKQ', 'hGDa0uOiEG8', 'E-jqUEH_efM
', 'QRVefNvnH7k', 'laCMmT5Caco'], 'viewCount': ['3716880', '1072765', '604251',
'586385', '243042', '198540', '160020', '153660', '91552', '61616'], 'likeCount'
: ['96287', '1480', '31759', '8673', '957', '563', '3375', '1116', '466', '35'],
 'dislikeCount': ['4849', '368', '372', '488', '889', '474', '208', '109', '33',
 '28'], 'commentCount': ['6697', '1242', '4857', '1438', '158', '724', '193', '5
6'], 'favoriteCount': ['0', '0', '0', '0', '0', '0', '0', '0', '0', '0']}
Average views were 688871.1 Average likes were 14471.1 and average dislikes were
 781.8
Response Rate (Likes to views ratio) was 0.02100697793825289
Ratio of likes and dislikes is 18.50997697620875
```

*Figure 2: YouTube Data*

Now, for visualization and analysis of data we have created some graphs such as bar graph, pie chart and line graph for simplified overview and comparison between data. To over simply the data we have done binning for polarity column in which it will show that if value is less than -0.6 than it will be categorized as very negative, a value between  -0.1 to -0.6 will be negative, and same for the positive scale of the polarity. Furthermore, remaining values which are nearer to zero will be categorized as neutral. Let us explain our analysis results that we have done on Spark.

Furthermore, to expand our project we have applied machine learning algorithm on our YouTube data. To know the response of viewers on a video, the number of likes is a good indicator.

## 6. EVALUATION:

Figure 3 represents the analogy between the subjectivity and polarity of the scores. The positive emotion represents a positive score. The below depicts that the scores with the high score in negative scale is more indulged towards subjectivity meaning that there are more comments on videos that have negative words or emotions. The comments might be a subject to positive or negative ideology of people.

```
df.show()
display(df.select("subjectivity", "polarity"))
```

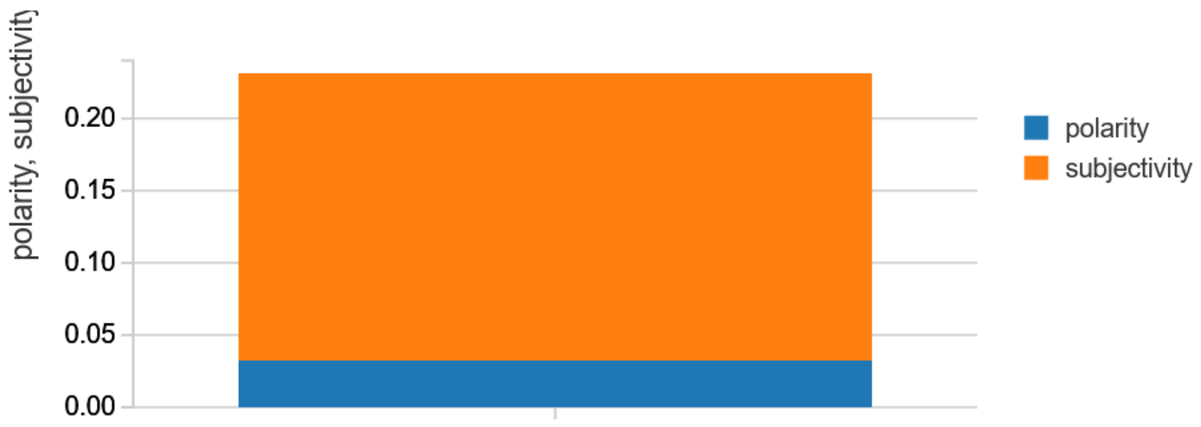
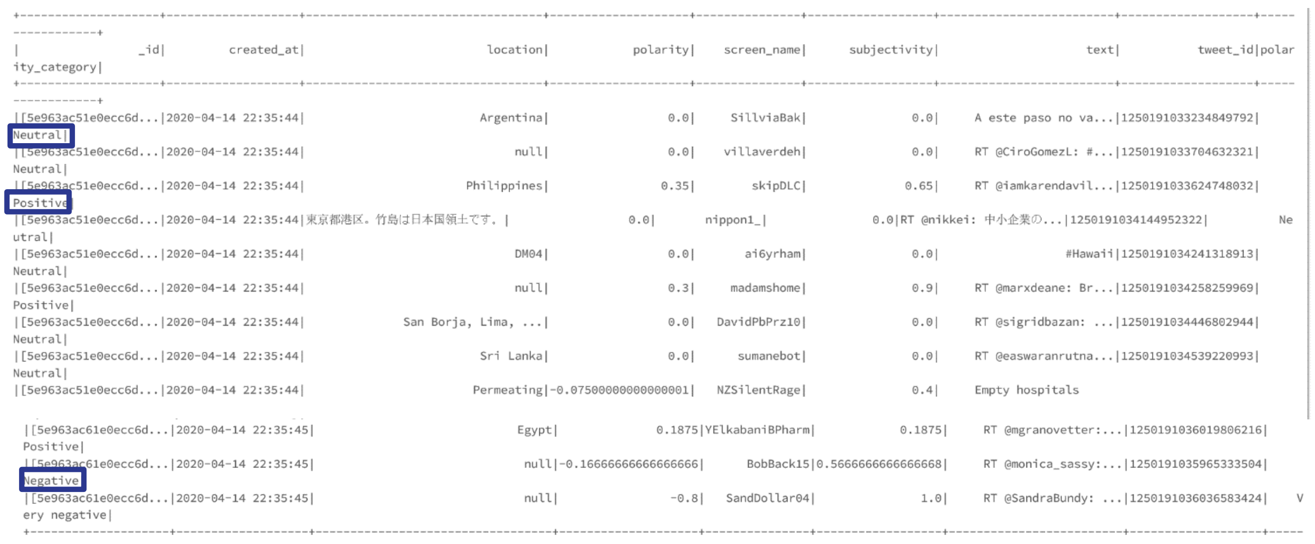


*Figure 3: Subjectivity and Polarity*

Figure 4 represents the binning output in which we have given different criteria for each group of values such as if value is less than -0.6 than it will be categorized as very negative, a value between  -0.1 to -0.6 will be negative, and same for the positive scale of the polarity.



```
+--------------------+-------------------+----------------------------------+--------------------+---------------+------------------+-------------------------+--------------------+-----
------------+
|                 _id|         created_at|                          location|            polarity|    screen_name|      subjectivity|                     text|            tweet_id|polar
ity_category|
+--------------------+-------------------+----------------------------------+--------------------+---------------+------------------+-------------------------+--------------------+-----
------------+
|[5e963ac51e0ecc6d...|2020-04-14 22:35:44|                         Argentina|                 0.0|     SillviaBak|               0.0|   A este paso no va...|1250191033234849792|
Neutral|
|[5e963ac51e0ecc6d...|2020-04-14 22:35:44|                              null|                 0.0|    villaverdeh|               0.0|   RT @CiroGomezL: #...|1250191033704632321|
Neutral|
|[5e963ac51e0ecc6d...|2020-04-14 22:35:44|                       Philippines|                0.35|        skipDLC|              0.65|   RT @iamkarendavil...|1250191033624748032|
Positive|
|[5e963ac51e0ecc6d...|2020-04-14 22:35:44|東京都港区。竹島は日本国領土です。|                 0.0|       nippon1_|               0.0|RT @nikkei: 中小企業の...|1250191034144952322|          Ne
utral|
|[5e963ac51e0ecc6d...|2020-04-14 22:35:44|                              DM04|                 0.0|        ai6yrham|               0.0|                  #Hawaii|1250191034241318913|
Neutral|
|[5e963ac51e0ecc6d...|2020-04-14 22:35:44|                              null|                 0.3|      madamshome|               0.9|   RT @marxdeane: Br...|1250191034258259969|
Positive|
|[5e963ac51e0ecc6d...|2020-04-14 22:35:44|                San Borja, Lima, ...|                 0.0|   DavidPbPrz10|               0.0|   RT @sigridbazan: ...|1250191034446802944|
Neutral|
|[5e963ac51e0ecc6d...|2020-04-14 22:35:44|                         Sri Lanka|                 0.0|      sumanebot|               0.0|   RT @easwaranrutna...|1250191034539220993|
Neutral|
|[5e963ac51e0ecc6d...|2020-04-14 22:35:44|                         Permeating|-0.07500000000000001|   NZSilentRage|               0.4|   Empty hospitals

 |[5e963ac61e0ecc6d...|2020-04-14 22:35:45|                             Egypt|              0.1875|YElkabaniBPharm|            0.1875|   RT @mgranovetter:...|1250191036019806216|
 Positive|
 |[5e963ac61e0ecc6d...|2020-04-14 22:35:45|                              null|-0.16666666666666666|      BobBack15|0.5666666666666668|   RT @monica_sassy:...|1250191035965333504|
 Negative|
 |[5e963ac61e0ecc6d...|2020-04-14 22:35:45|                              null|                -0.8|   SandDollar04|               1.0|   RT @SandraBundy: ...|1250191036036583424|    V
 ery negative|
 +--------------------+-------------------+----------------------------------+--------------------+---------------+------------------+-------------------------+--------------------+-----
```

*Figure 4: Spark Dataframe*

Figure 5 represents the analogy between the subjectivity and polarity of the scores. The positive emotion represents a positive score. The below depicts that the scores with the high score in negative scale is more indulged towards subjectivity meaning that there are more comments on videos that have negative words or emotions. The comments might be a subject to positive or negative ideology of people.

Subjectivity score for each polarity category

```
display(bucketed.select("polarity_category","subjectivity"))
```

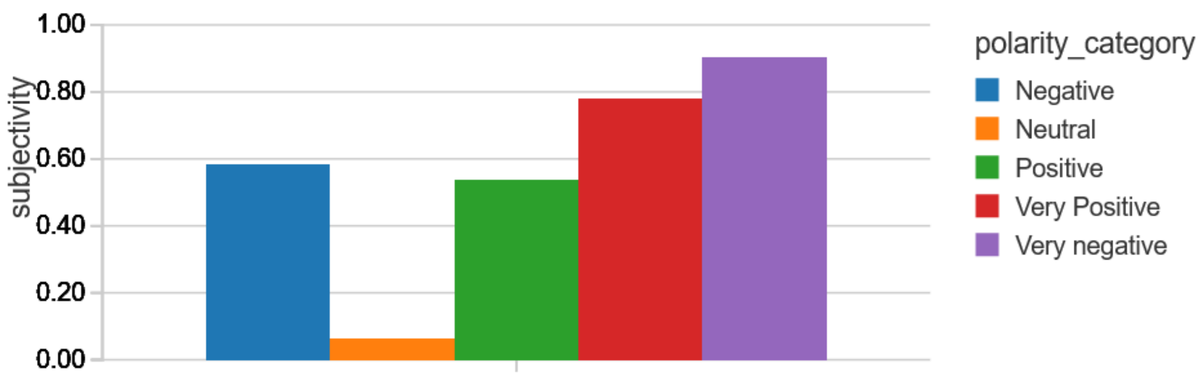


*Figure 5: Subjectivity vs Polarity*

Figure 6 illustrates the variation in the polarity based on the comments. This graph gives us an overview of how the polarity scores vary. We can clearly understand that the graph varies a lot so the comments have the positive as well as the negative emotion words.

Checking variation in polarity overtime

```
display(bucketed.select("created_at","polarity"))
```

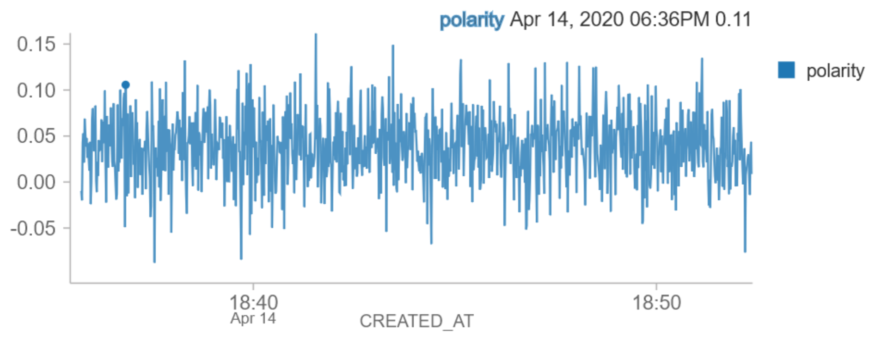


*Figure 6: Variation in Polarity*

Figure 7 depicts the average view count with repsect to channel title. However, here we can see that michael anderon consist hughest view count compare to two other channels.

## Average of view count based on channel title

```
yt_grouped = yt_df.select("viewCount","channelTitle").groupBy("channelTitle").avg().limit(3)
display(yt_grouped)
```

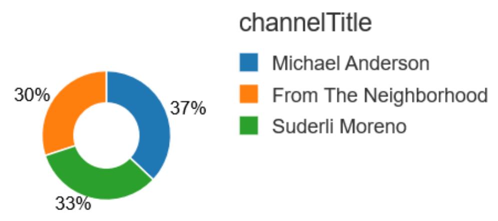


*Figure 7: Percentage of Viewers*

Figure 8 illustrates the comparison between different counts such as dislike, like and commnet count. Here, “like count” contibutes maximum count with 100k of likes comapred to dislike and comment counts.

```
yt_ld_ratio = yt_df.select("likeCount", "dislikeCount","commentCount", "viewCount")
display(yt_ld_ratio)
```

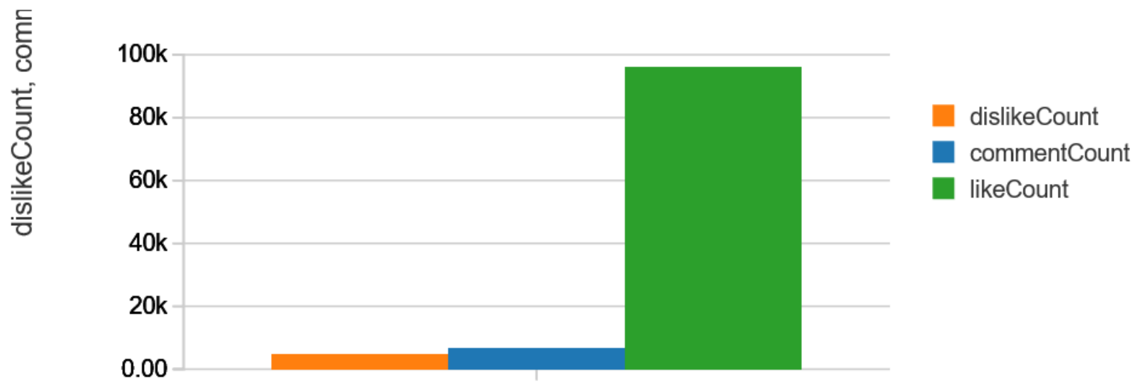


*Figure 8: Counts of likes, dislikes and comments*

Figure 9 shows that dislike and comment counts are compared based on YouTube channel title. From this graph we can see that the comment count and the dislikes for each of the top-notch channels data obtained by the streaming.

Comparing dislike count and comment count for youtube channel title

```
views_likes_comments = yt_df.select("likeCount", "viewCount", "commentCount","dislikeCount","channelTitle")
vlc_limited = views_likes_comments.limit(5)
display(vlc_limited)
```

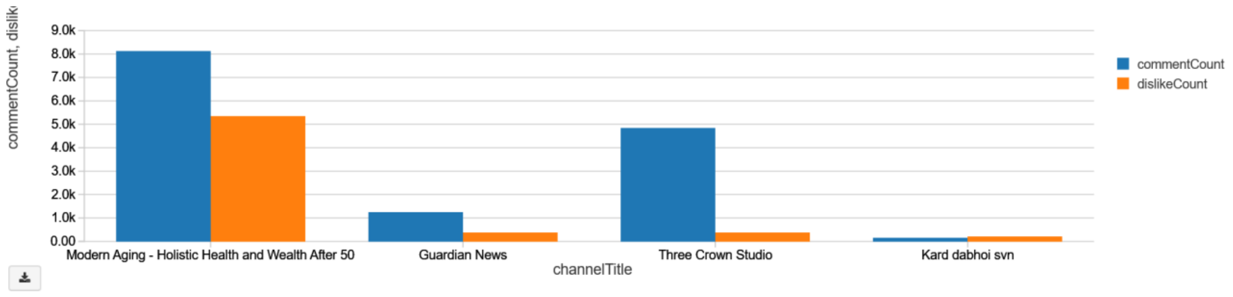


*Figure 9: Relation between Comments and Dislikes*

Figure 10 represents the relationship between the views and their counts based on likes, dislikes and comments. The users with more likes show a drastic change in the viewcount however, it seems that there are not much people who tend to reject the

```
display(views_likes_comments)
```

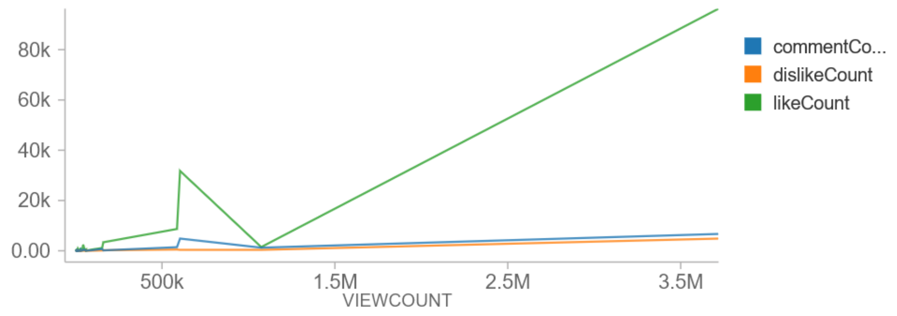
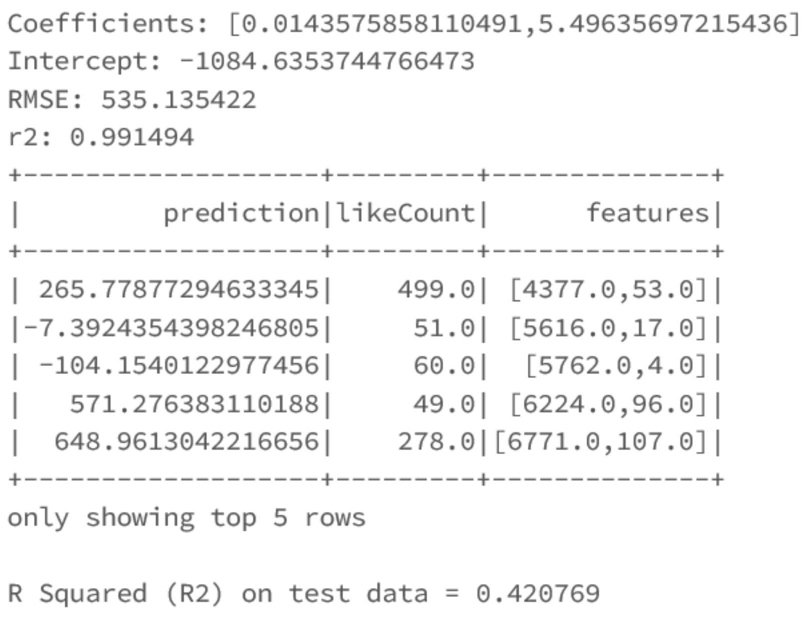


*Figure 10: Relationships between View Counts and Likes*

We applied linear regression method with training and test data, where training data would be 70% and test data would be 30%. The model predicted how many likes a video would get based on other parameters. Figure 11 shows the accuracy achieved on the test data was around 42.08% however on training data it is 99.15%. For every 100 views, a YouTuber is expected to get 1 like. The correlation coefficient of the data is around 65% which means that the attributes between the test and training where predicted 65 times correctly out of every 100 times.



```
Coefficients: [0.0143575858110491,5.49635697215436]
Intercept: -1084.6353744766473
RMSE: 535.135422
r2: 0.991494
+------------------+---------+--------------+
|        prediction|likeCount|      features|
+------------------+---------+--------------+
| 265.77877294633345|    499.0| [4377.0,53.0]|
|-7.3924354398246805|     51.0| [5616.0,17.0]|
| -104.1540122977456|     60.0|  [5762.0,4.0]|
|   571.276383110188|     49.0| [6224.0,96.0]|
|  648.9613042216656|    278.0|[6771.0,107.0]|
+------------------+---------+--------------+
only showing top 5 rows

R Squared (R2) on test data = 0.420769
```

*Figure 11: Linear Regression Predictions*

For the interactive experience of the pynotebook of spark please follow the link: https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/3625458541459463/1638528181175261/974204641355987/latest.html

## WORK DISTRIBUTION:

**Please note** that we stay together in the same house, we have done most of the part together as a team. Since the Professor has asked for an explicit list of tasks performed, the table below provides rough idea about the distribution of workflow.

| Name and Student ID of the Group Member | Data Collection and Storage using Python and MongoDB | Data Analysis and visualization (Apache Spark) | Extra Tasks |
|---|---|---|---|
| Kedar Deodhar | Data collection of YouTube API and cleaning of data. | YouTube data Visualization, Machine Learning in spark | Coding and Presentation |
| Janvi Patel | Data collection of twitter API, sentiment analysis of gathered data. | Twitter Visualization analysis and binning of data using Apache Spark. | Visualization and Py-Notebook Presentation |
| Prerak Trivedi | Cleaning of Twitter data and sentiment analysis of YouTube data. | Mongo Spark Connector, YouTube data visualization | Report and Presentation |

## 7. POSSIBLE FUTURE WORK:

### LIMITATIONS:

- YouTube API client allows only 50 videos per request on free basis. We did not have enough funds at our disposal to get a corporate access and pull huge rows of dataset.

- There were no live events occurring due to COVID-19. Hence, we worked on the API based data.

- There are some key entities in tweepy API response which are often missing at random or unstructured which creates errors while attempting to obtain analysis. For instance, the location column consists of values like Toronto, Ontario or Ontario or Toronto or findmetoronto@gmail.com (which is invalid).

- At the end of our project(pynotebook), we have attempted to predict a relationship between "LikeCounts" based on other sensible and useful criteria's. However, since the sample size was small, despite of selecting 30% data for testing the model there is slight chance of overfitting.

### POSSIBLE IMPROVISATIONS:

- Using enough computational power, we can provide real time analysis of the dataset as it is being streamed which can be helpful.

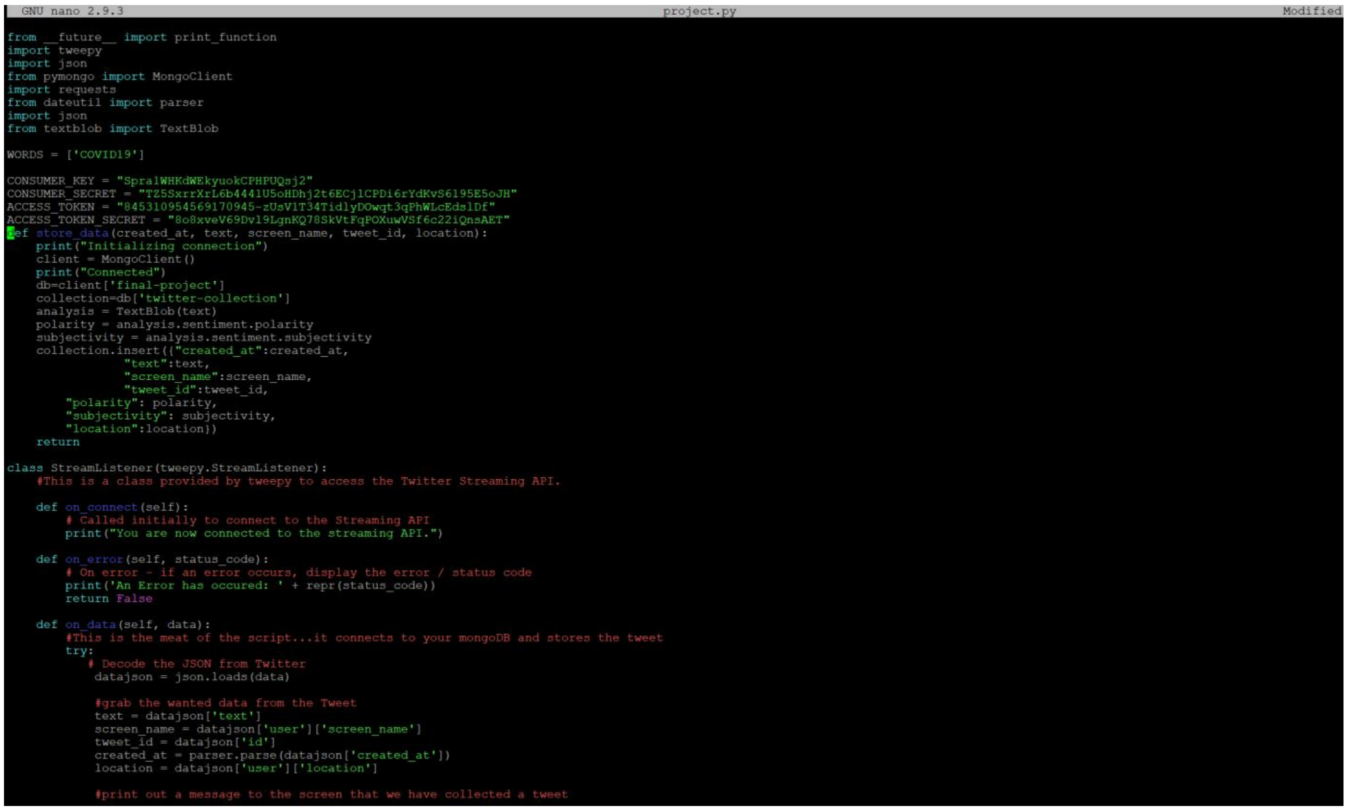
- The machine learning applied would work great with the more available samples.

- If we would be allowed to use other software for data cleaning or a strict data entering scheme, then the analysis would be appropriate as the information extracted from the twitter were missing.

### LINK TO VIEW THE PRESENTATION VIDEO:

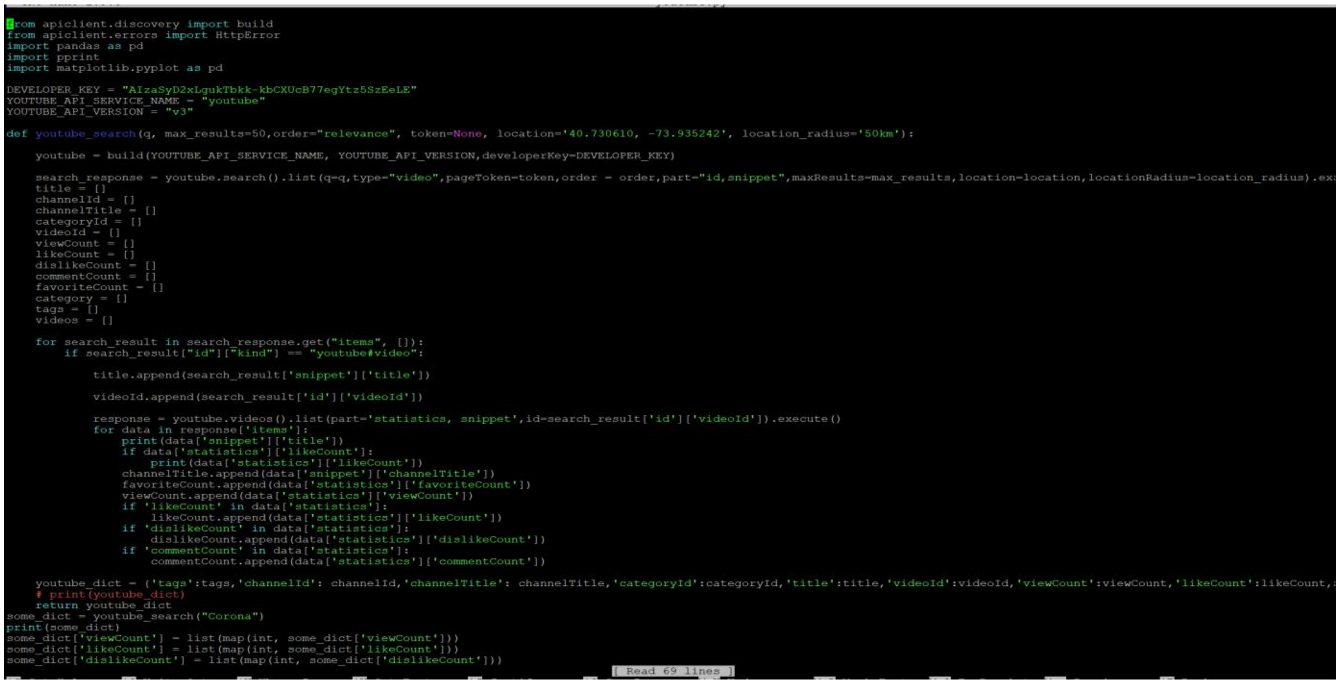
## 8. APPENDIX:

### Twitter Code:



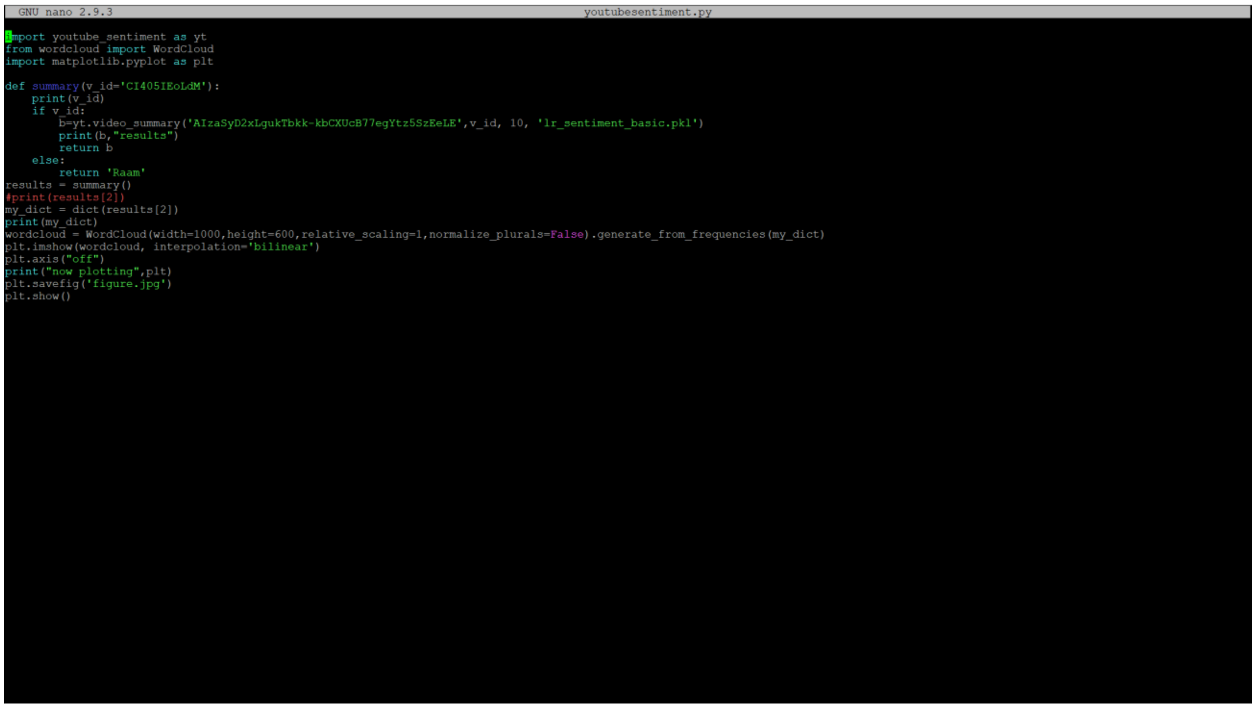
```
GNU nano 2.9.3                                  project.py                                  Modified

from __future__ import print_function
import tweepy
import json
from pymongo import MongoClient
import requests
from dateutil import parser
import json
from textblob import TextBlob

WORDS = ['COVID19']

CONSUMER_KEY = "Spra1WHKdWEkyuokCPHPUQsj2"
CONSUMER_SECRET = "TZ5SxrrXrL6b4441U5oHDhj2t6ECj1CPDi6rYdKvS6195E5oJH"
ACCESS_TOKEN = "845310954569170945-zUsV1T34TidlyDOwqt3qPhWLcEds1Df"
ACCESS_TOKEN_SECRET = "8o8xveV69Dv19LgnKQ78SkVtFqPOXuwVSf6c22iQnsAET"
def store_data(created_at, text, screen_name, tweet_id, location):
    print("Initializing connection")
    client = MongoClient()
    print("Connected")
    db=client['final-project']
    collection=db['twitter-collection']
    analysis = TextBlob(text)
    polarity = analysis.sentiment.polarity
    subjectivity = analysis.sentiment.subjectivity
    collection.insert({"created_at":created_at,
                "text":text,
                "screen_name":screen_name,
                "tweet_id":tweet_id,
        "polarity": polarity,
        "subjectivity": subjectivity,
        "location":location})
    return

class StreamListener(tweepy.StreamListener):
    #This is a class provided by tweepy to access the Twitter Streaming API.

    def on_connect(self):
        # Called initially to connect to the Streaming API
        print("You are now connected to the streaming API.")

    def on_error(self, status_code):
        # On error - if an error occurs, display the error / status code
        print('An Error has occured: ' + repr(status_code))
        return False

    def on_data(self, data):
        #This is the meat of the script...it connects to your mongoDB and stores the tweet
        try:
           # Decode the JSON from Twitter
            datajson = json.loads(data)

            #grab the wanted data from the Tweet
            text = datajson['text']
            screen_name = datajson['user']['screen_name']
            tweet_id = datajson['id']
            created_at = parser.parse(datajson['created_at'])
            location = datajson['user']['location']

            #print out a message to the screen that we have collected a tweet
```

## YouTube Code:



```
from apiclient.discovery import build
from apiclient.errors import HttpError
import pandas as pd
import pprint
import matplotlib.pyplot as pd

DEVELOPER_KEY = "AIzaSyD2xLgukTbkk-kbCXUcB77egYtz5SzEeLE"
YOUTUBE_API_SERVICE_NAME = "youtube"
YOUTUBE_API_VERSION = "v3"

def youtube_search(q, max_results=50,order="relevance", token=None, location='40.730610, -73.935242', location_radius='50km'):

    youtube = build(YOUTUBE_API_SERVICE_NAME, YOUTUBE_API_VERSION,developerKey=DEVELOPER_KEY)

    search_response = youtube.search().list(q=q,type="video",pageToken=token,order = order,part="id,snippet",maxResults=max_results,location=location,locationRadius=location_radius).ex
    title = []
    channelId = []
    channelTitle = []
    categoryId = []
    videoId = []
    viewCount = []
    likeCount = []
    dislikeCount = []
    commentCount = []
    favoriteCount = []
    category = []
    tags = []
    videos = []

    for search_result in search_response.get("items", []):
        if search_result["id"]["kind"] == "youtube#video":

            title.append(search_result['snippet']['title'])

            videoId.append(search_result['id']['videoId'])

            response = youtube.videos().list(part='statistics, snippet',id=search_result['id']['videoId']).execute()
            for data in response['items']:
                print(data['snippet']['title'])
                if data['statistics']['likeCount']:
                    print(data['statistics']['likeCount'])
                channelTitle.append(data['snippet']['channelTitle'])
                favoriteCount.append(data['statistics']['favoriteCount'])
                viewCount.append(data['statistics']['viewCount'])
                if 'likeCount' in data['statistics']:
                    likeCount.append(data['statistics']['likeCount'])
                if 'dislikeCount' in data['statistics']:
                    dislikeCount.append(data['statistics']['dislikeCount'])
                if 'commentCount' in data['statistics']:
                    commentCount.append(data['statistics']['commentCount'])

    youtube_dict = {'tags':tags,'channelId': channelId,'channelTitle': channelTitle,'categoryId':categoryId,'title':title,'videoId':videoId,'viewCount':viewCount,'likeCount':likeCount,
    # print(youtube_dict)
    return youtube_dict
some_dict = youtube_search("Corona")
print(some_dict)
some_dict['viewCount'] = list(map(int, some_dict['viewCount']))
some_dict['likeCount'] = list(map(int, some_dict['likeCount']))
some_dict['dislikeCount'] = list(map(int, some_dict['dislikeCount']))
                                                  [ Read 69 lines ]
```

## YouTube Sentiment Code:



```
  GNU nano 2.9.3                                                  youtubesentiment.py

import youtube_sentiment as yt
from wordcloud import WordCloud
import matplotlib.pyplot as plt

def summary(v_id='CI405IEoLdM'):
    print(v_id)
    if v_id:
        b=yt.video_summary('AIzaSyD2xLgukTbkk-kbCXUcB77egYtz5SzEeLE',v_id, 10, 'lr_sentiment_basic.pkl')
        print(b,"results")
        return b
    else:
        return 'Raam'
results = summary()
#print(results[2])
my_dict = dict(results[2])
print(my_dict)
wordcloud = WordCloud(width=1000,height=600,relative_scaling=1,normalize_plurals=False).generate_from_frequencies(my_dict)
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
print("now plotting",plt)
plt.savefig('figure.jpg')
plt.show()
```