PRINCIPLES OF BIG DATA MANAGEMENT

Phase 2 Report

Twitter Data Analysis (Avengers)

BY
Neha Navgale (16286910)
Naveena Madepally (16280573)
Dharani Muli (16286306)

Project Objective:

1. To collect tweets related to Avengers.
2. To analyze the collected tweets and develop interesting analysis on it using Spark SQL.
3. Create Visualizations on the analysis.

Installation and setup:

Hadoop

Spark

Python

Angular

Design and Implementation

Backend

1. Designed Python Flask application and used tweepy library to fetch Tweets (approx. 50K) from Twitter's Streaming APIs. We have fetched tweets for **Avengers**.
2. Performed analysis on Tweets data and came up with 10 queries to extract meaningful information from tweets.
3. Used pyspark library to create tweets view in Spark SQL, executed queries on view and generated the output.
4. Output is sent in json format to Angular 7 frontend application to draw visualization.

Frontend

1. Designed Angular 7 application to visualize all the executed queries.
2. Used am4charts library to draw different graphs.

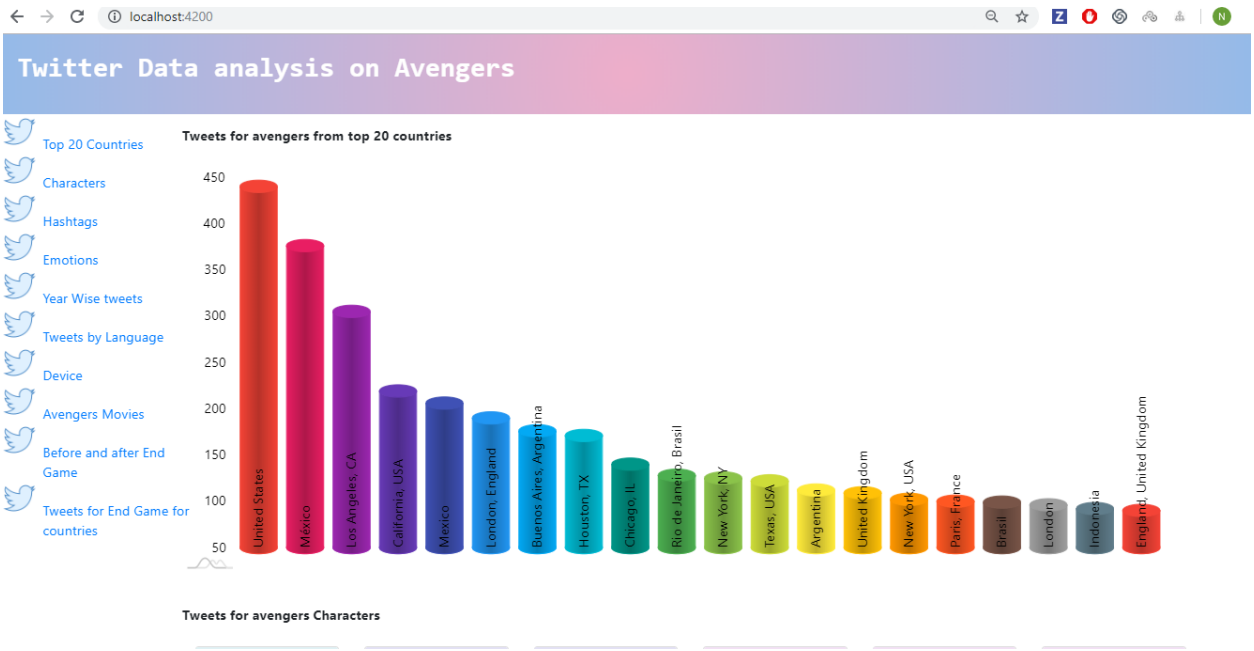
Testing

1. Performed Unit testing on both Frontend and backend application.
2. Performed integration testing of frontend and backend.

Queries

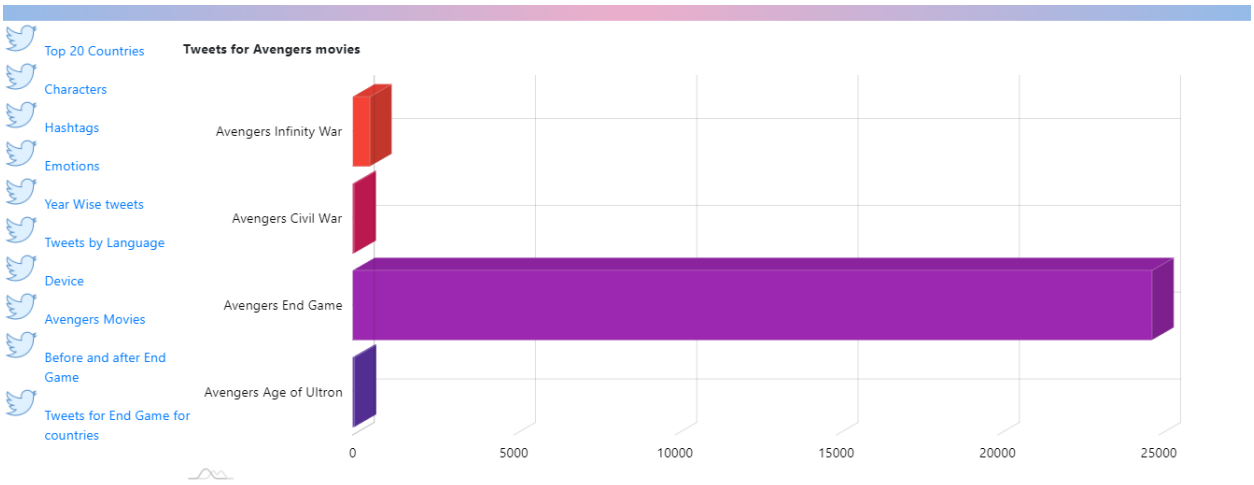
1. **Number of tweets for Avengers from different location**

```
df = spark.read.json("static/Tweets/NewAvengersTweets.json")
df.createOrReplaceTempView("tweets")
df.printSchema()

query1 = spark.sql("select count(*) as count, user.location from tweets where user.location is not null group by "
                   "user.location order by count desc Limit 20")
query1.show()
pd = query1.toPandas()
pd.to_csv('static/Output/byCountry.csv', index=False)
```



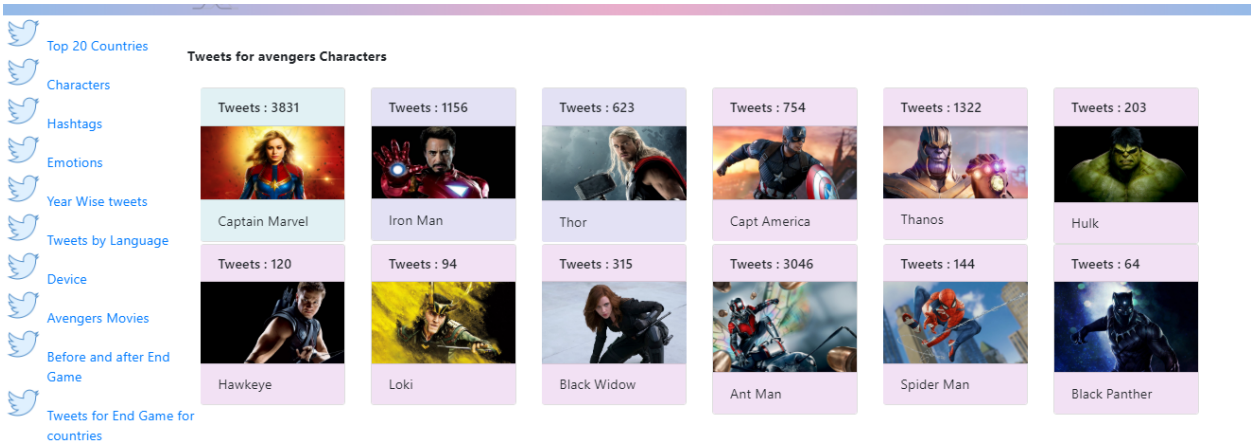
2. Number of Tweets for all Avenger Movies

```
df = spark.read.json("static/Tweets/NewAvengersTweets.json")
df.createOrReplaceTempView("tweets")
df.printSchema()

query2 = spark.sql("SELECT COUNT(*) AS NumberOfTweets, 'Avengers Infinity War' as Movie FROM tweets where upper(text) LIKE '%INFINITY%' "
                   "UNION SELECT COUNT(*) AS NumberOfTweets,'Avengers Age of Ultron' as Movie FROM tweets where upper(text) LIKE '%ULTRON%' "
                   "UNION SELECT COUNT(*) AS NumberOfTweets, 'Avengers Civil War' as Movie FROM tweets where upper(text) LIKE '%CIVIL%' "
                   "UNION SELECT COUNT(*) AS NumberOfTweets, 'Avengers End Game' as Movie FROM tweets where upper(text) LIKE '%END%'")
query2.show()
pd = query2.toPandas()
pd.to_csv('static/Output/byMovie.csv', index=False)
```



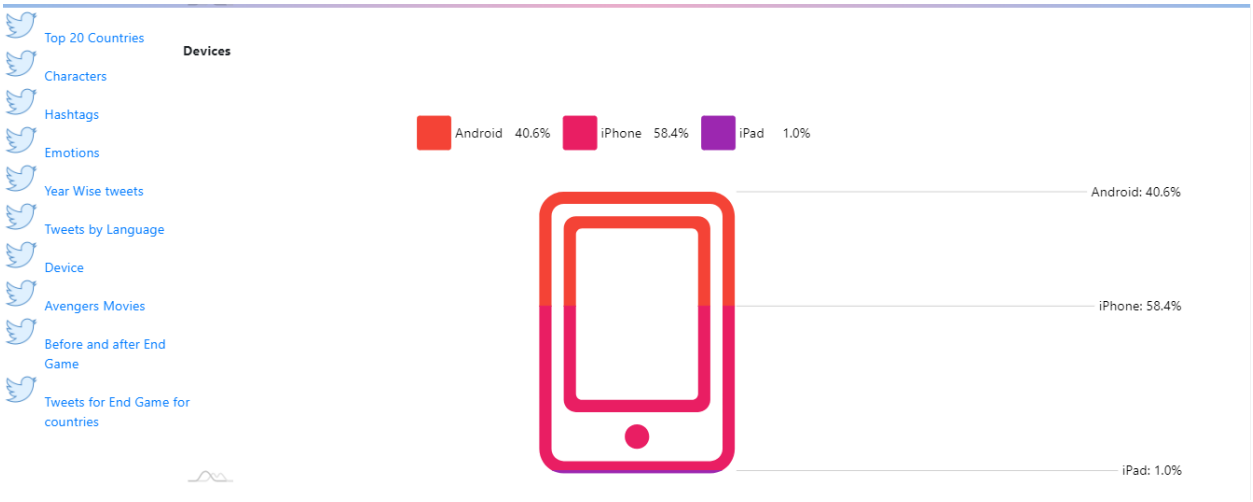
3. Number of tweets for each characters of Avengers

```
df = spark.read.json("static/Tweets/NewAvengersTweets.json")
df.createOrReplaceTempView("tweets")
df.printSchema()

query3 = spark.sql("SELECT COUNT(*) AS NumberOfTweets, 'Captain America' as Character FROM tweets "
                   "where upper(text) LIKE '%AMERICA%' UNION SELECT COUNT(*) AS NumberOfTweets,'Thanos' as Character FROM tweets "
                   "where upper(text) LIKE '%THANOS%' UNION SELECT COUNT(*) AS NumberOfTweets, 'Black Panther' as Character FROM tweets "
                   "where upper(text) LIKE '%PANTHER%' UNION SELECT COUNT(*) AS NumberOfTweets, 'Rogg' as Character FROM tweets "
                   "where upper(text) LIKE '%ROGG%' UNION SELECT COUNT(*) AS NumberOfTweets, 'Spider-Man' as Character FROM tweets "
                   "where upper(text) LIKE '%SPIDER%' UNION SELECT COUNT(*) AS NumberOfTweets, 'Iron Man' as Character FROM tweets "
                   "where upper(text) LIKE '%IRON%' UNION SELECT COUNT(*) AS NumberOfTweets, 'Black Widow' as Character FROM tweets "
                   "where upper(text) LIKE '%WIDOW%' UNION SELECT COUNT(*) AS NumberOfTweets, 'Hulk' as Character FROM tweets "
                   "where upper(text) LIKE '%HULK%' UNION SELECT COUNT(*) AS NumberOfTweets, 'Thor' as Character FROM tweets "
                   "where upper(text) LIKE '%THOR%' UNION SELECT COUNT(*) AS NumberOfTweets, 'Hawkeye' as Character FROM tweets "
                   "where upper(text) LIKE '%HAWKEYE%' UNION SELECT COUNT(*) AS NumberOfTweets, 'Deadpool' as Character FROM tweets "
                   "where upper(text) LIKE '%DEADPOOL%' UNION SELECT COUNT(*) AS NumberOfTweets, 'Loki' as Character FROM tweets "
                   "where upper(text) LIKE '%LOKI%' UNION SELECT COUNT(*) AS NumberOfTweets, 'Captain Marvel' as Character FROM tweets "
                   "where upper(text) LIKE '%MARVEL%' UNION SELECT COUNT(*) AS NumberOfTweets, 'Ant-Man' as Character FROM tweets "
                   "where upper(text) LIKE '%ANT%' UNION SELECT COUNT(*) AS NumberOfTweets, 'Nebula' as Character FROM tweets "
                   "where upper(text) LIKE '%NEBULA%' UNION SELECT COUNT(*) AS NumberOfTweets, 'Gamora' as Character FROM tweets "
                   "where upper(text) LIKE '%GAMORA%' UNION SELECT COUNT(*) AS NumberOfTweets, 'Groot' as Character FROM tweets "
                   "where upper(text) LIKE '%GROOT%' UNION SELECT COUNT(*) AS NumberOfTweets, 'Nick Fury' as Character FROM tweets "
                   "where upper(text) LIKE '%NICK%'")
query3.show()
pd = query3.toPandas()
pd.to_csv('static/Output/byCharacter.csv', index=False)
```



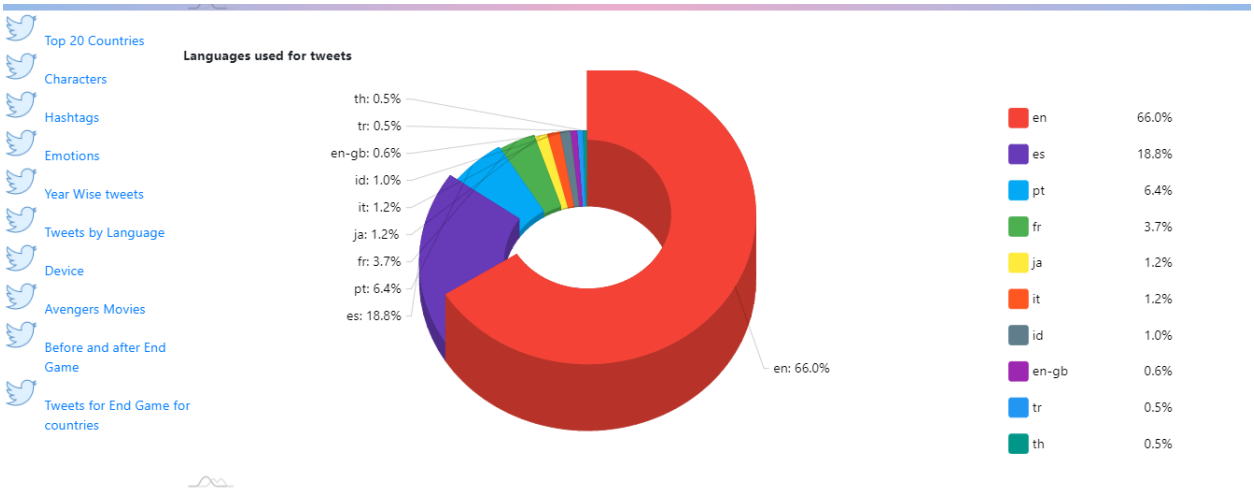
4. Number of tweets posted from different device

```
df = spark.read.json("static/Tweets/NewAvengersTweets.json")
df.createOrReplaceTempView("tweets")
df.printSchema()

query4 = spark.sql("SELECT COUNT(*) AS Count, 'iPhone' as Device FROM tweets where source LIKE '%iPhone%' UNION SELECT COUNT(*) "
                   "AS Count,'Android' as Device FROM tweets where source LIKE '%Android%' UNION SELECT COUNT(*) "
                   "AS Count,'iPad' as Device FROM tweets where source LIKE '%iPad%'")
query4.show()
pd = query4.toPandas()
pd.to_csv('static/Output/byDevice.csv', index=False)
```



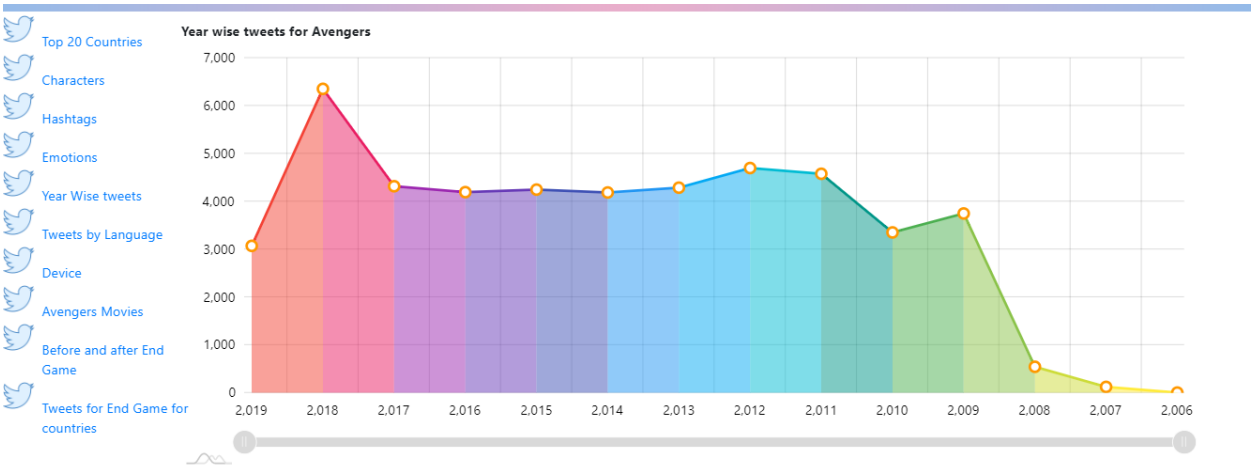
5. Number of tweets in different language

```
df = spark.read.json("static/Tweets/NewAvengersTweets.json")
df.createOrReplaceTempView("tweets")
df.printSchema()

query5 = spark.sql("SELECT user.lang, count(*) AS Lang_Count FROM tweets where user.lang is "
                   "not null group by user.lang order by Lang_Count desc LIMIT 10")
query5.show()
pd = query5.toPandas()
pd.to_csv('static/Output/byLanguage.csv', index=False)
```



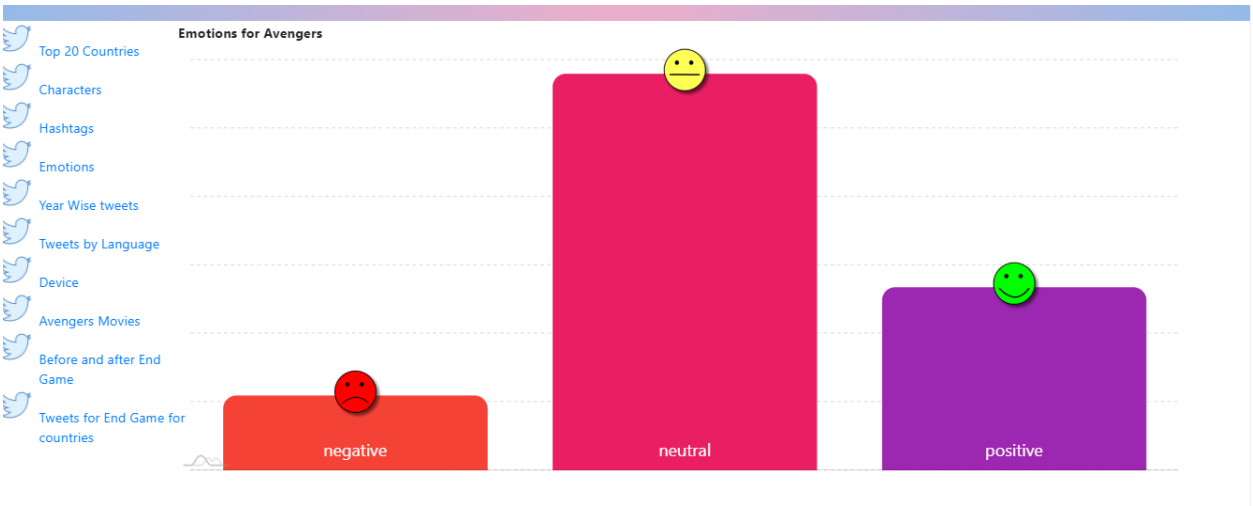
6. Trend of tweets over the years

```
df = spark.read.json("static/Tweets/NewAvengersTweets.json")
df.createOrReplaceTempView("tweets")
df.printSchema()

year = df.select(date_format((from_unixtime(unix_timestamp('user.created_at', 'EEE MMM dd HH:mm:ss ZZZZZ yyyy')).alias('date')),
                             'MM/dd/yyyy').alias('date'))
year.printSchema()
year.createOrReplaceTempView("year")

query6 = spark.sql("select count(*) as NumberOfTweets, substr(date, 7, 11) as MovieYear from year group by substr(date, 7, 11) "
                   "order by substr(date, 7, 11) desc")
pd = query6.toPandas()
pd.to_csv('static/Output/byYear.csv', index=False)
query6.show()
```



7. Sentiment Analysis on Avengers

```
df = spark.read.json("static/Tweets/NewAvengersTweets.json")
df.createOrReplaceTempView("tweets")
df.printSchema()

query7 = spark.sql("SELECT text FROM tweets")
i=0
positive=0
neutral=0
negative=0

for t in query7.select("text").collect():
    i=i+1
    analysis = TextBlob(str((t.text).encode('ascii', 'ignore')))
    if (analysis.sentiment.polarity<0):
        negative=negative+1
        # print(i," in negative")
    elif(analysis.sentiment.polarity==0.0):
        neutral=neutral+1
        # print(i," in neutral")
    elif(analysis.sentiment.polarity>0):
        positive=positive+1
        # print(i," in positive")

print(negative)
print(neutral)
print(positive)

sentiment = {'Sentiment': ['negative', 'neutral', 'positive'], 'Count': [negative, neutral, positive]}
sentiment = pd.DataFrame(data=sentiment)
sentiment.to_csv('static/Output/sentiment.csv', index=False)
```

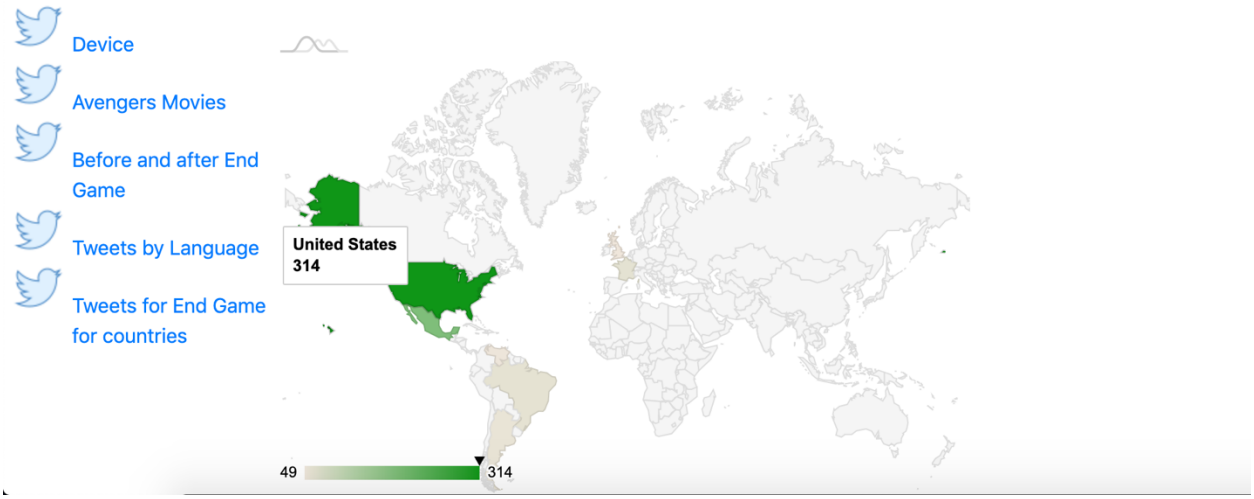


8. Hashtags used for Avengers

```
df = spark.read.json("static/Tweets/NewAvengersTweets.json")
df.createOrReplaceTempView("tweets")
df.printSchema()

hashtag = df.select('text', regexp_extract(col('text'), '#(\w+)', 1).alias('hashtag'))
hashtag.createOrReplaceTempView("hashtag")

query8 = spark.sql("SELECT count(*) as count, hashtag from hashtag where hashtag is not null group by hashtag")
query8.show()
pd = query8.toPandas()
pd.to_csv('static/Output/hashtag.csv', index=False)
```

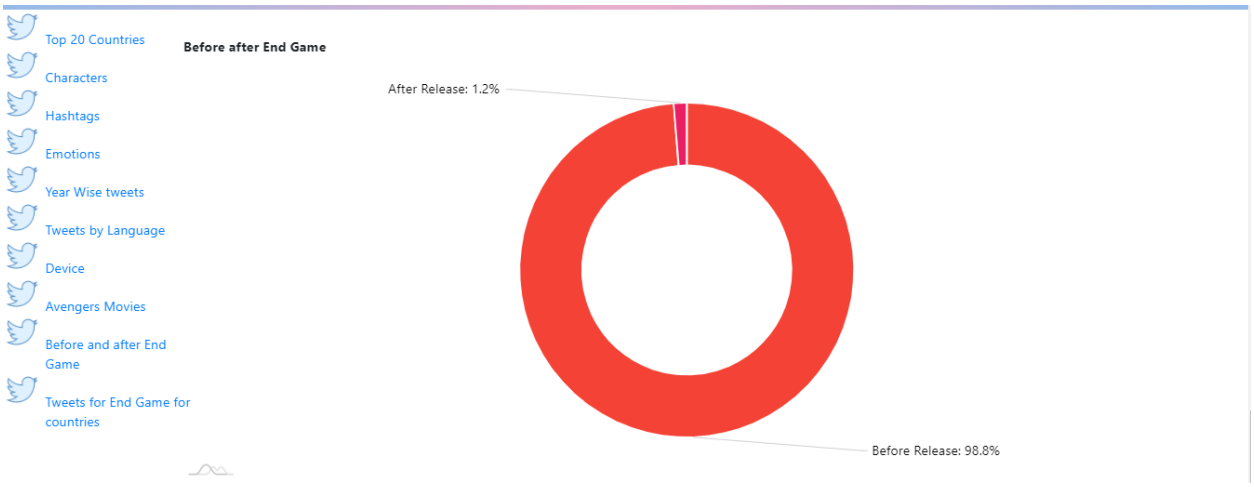
9. Number of tweets for Movie ‘Avengers: End Game’ from different Locations

```
df = spark.read.json("static/Tweets/NewAvengersTweets.json")
df.createOrReplaceTempView("tweets")
df.printSchema()

query9 = spark.sql("SELECT COUNT(*) AS NumberOfTweets, 'Avengers End Game' as Movie, user.location as Location "
                   "FROM tweets where upper(text) LIKE '%END%' and user.location is not null group by user.location "
                   "ORDER BY NumberOfTweets DESC LIMIT 20")
query9.show()
pd = query9.toPandas()
pd.to_csv('static/Output/byEndGameAndLocation.csv', index=False)
```



10. Number of tweets before and after the release of ‘Avengers: End Game’

```
df = spark.read.json("static/Tweets/NewAvengersTweets.json")
df.createOrReplaceTempView("tweets")
df.printSchema()

Movieyear = df.select('text', date_format((from_unixtime(unix_timestamp('user.created_at', 'EEE MMM dd HH:mm:ss ZZZZZ yyyy')).alias('date')),
                                          'MM/dd/yyyy').alias('date'))
Movieyear.printSchema()
Movieyear.createOrReplaceTempView("movieYear")
#
query10 = spark.sql("SELECT count(date) as NumberOfTweets, 'After Release' As Avengers from movieYear where upper(text) "
                    "like '%END%' and date in ('04/22/2019', '04/23/2019', '04/24/2019', '04/25/2019', '04/26/2019') "
                    "UNION SELECT count(date) as NumberOfTweets, 'Before Release' As Avengers from movieYear "
                    "where upper(text) like '%END%' and date not in ('04/22/2019', '04/23/2019', '04/24/2019', '04/25/2019', '04/26/2019')")
query10.show()
pd = query10.toPandas()
pd.to_csv('static/Output/beforeAndAfterRelease.csv', index=False)
```



CODE LINK:

https://github.com/NehaNavgale/TwitterDataAnalysis/tree/master/Phase-2