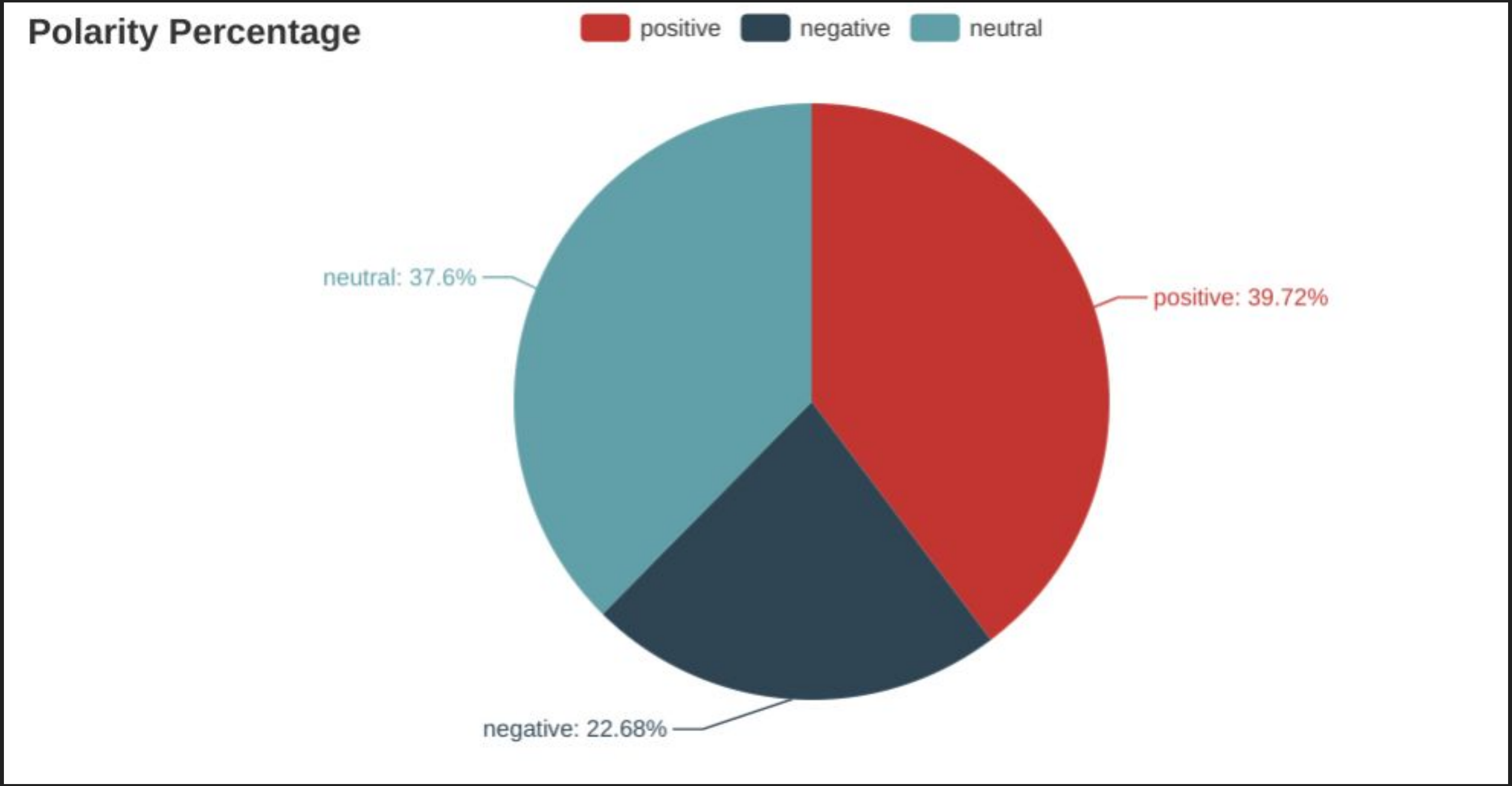
# Tweets Analysis with Spark

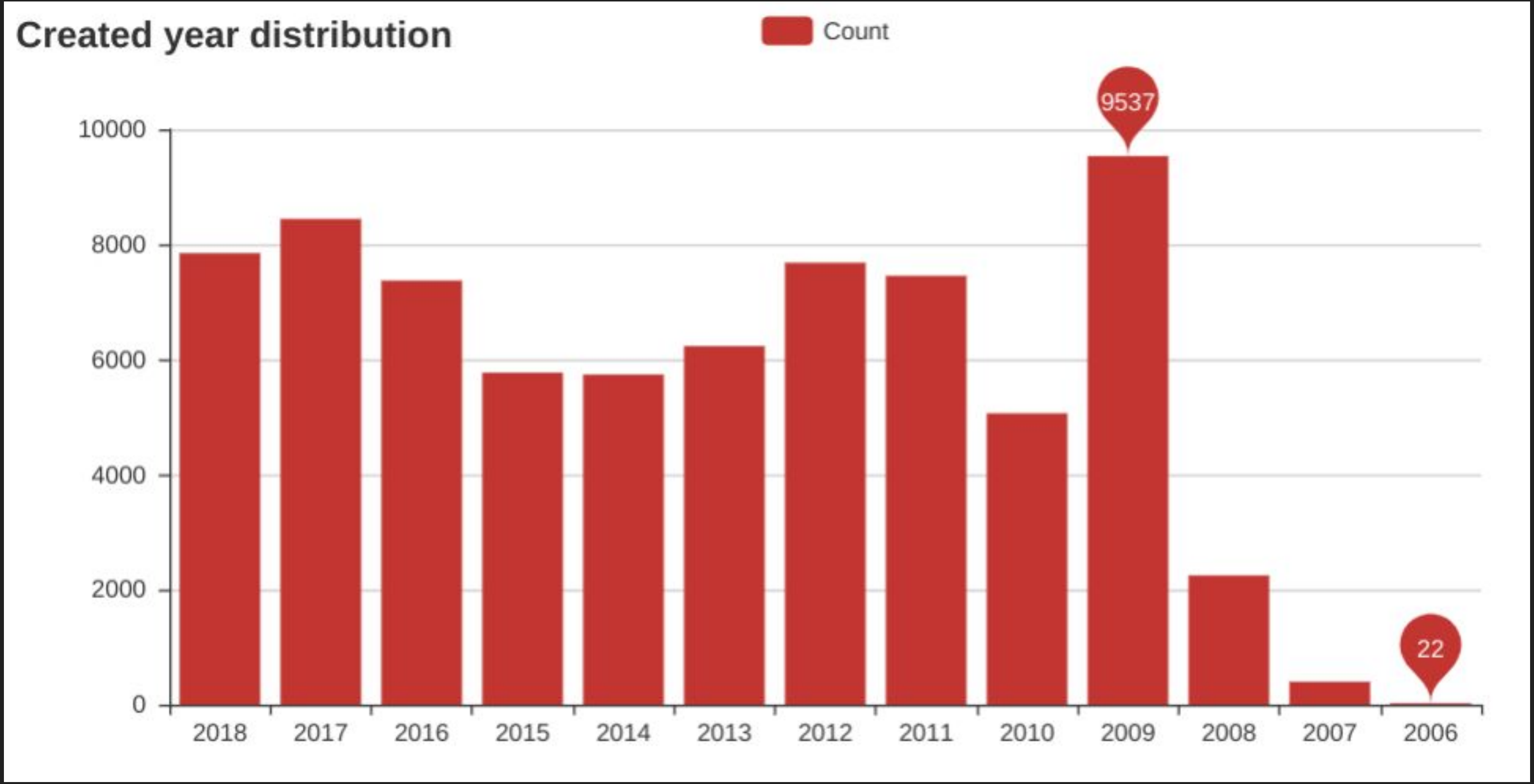
Harshil Patel (8), Matthew Boerner (1), Stephanie Retzke (10) and Yong Zheng (14)

Yong

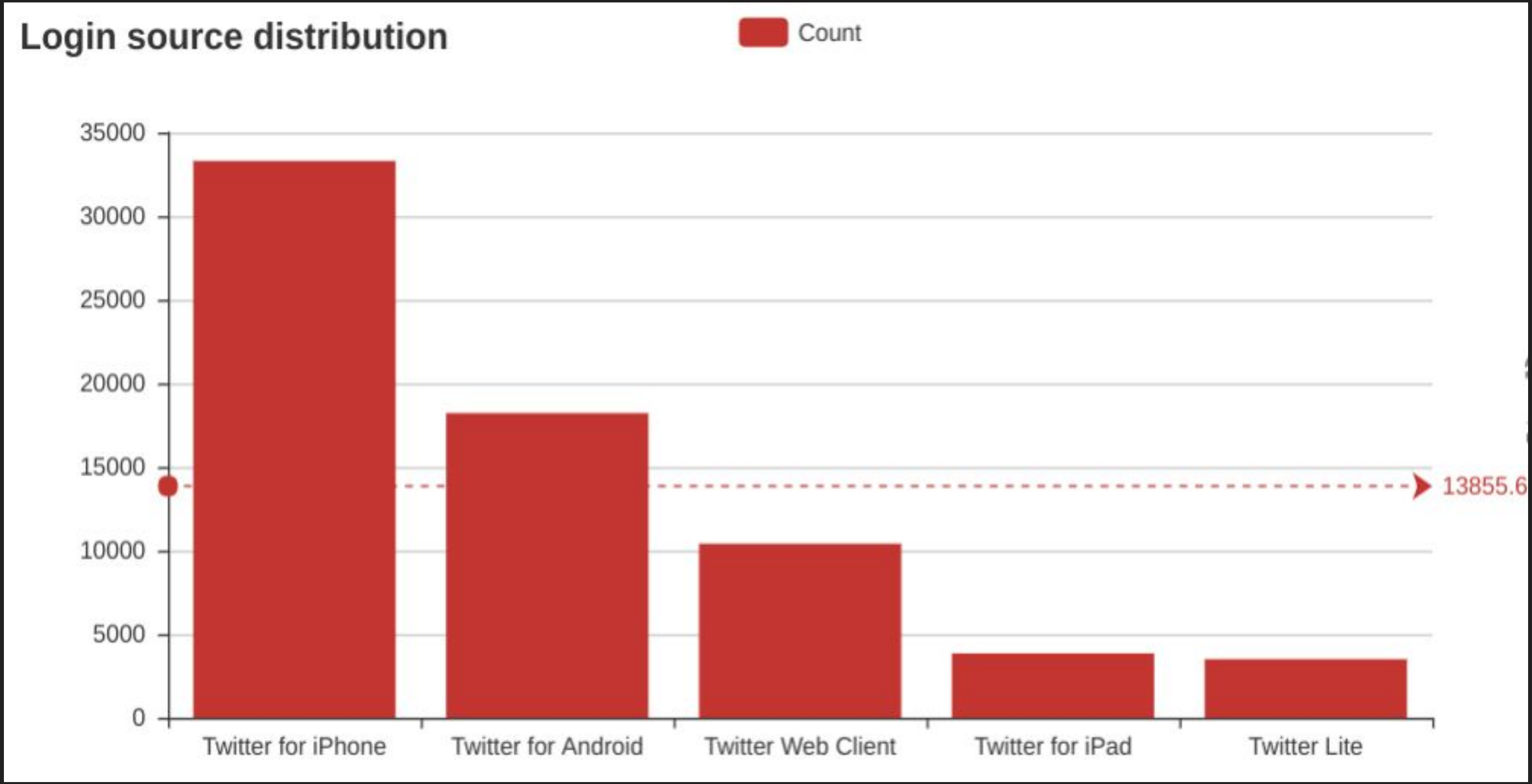
# Increment 1 - Query 1 - Polarity Percentage



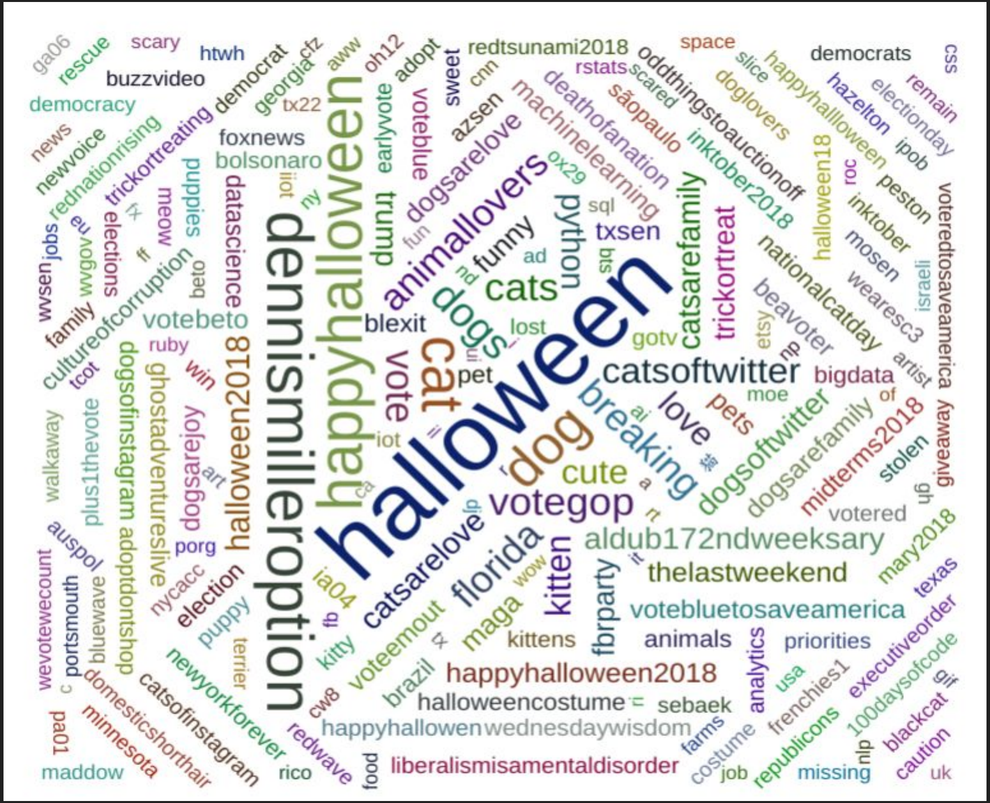
# Increment 1 - Query 2 - Created Year Distribution



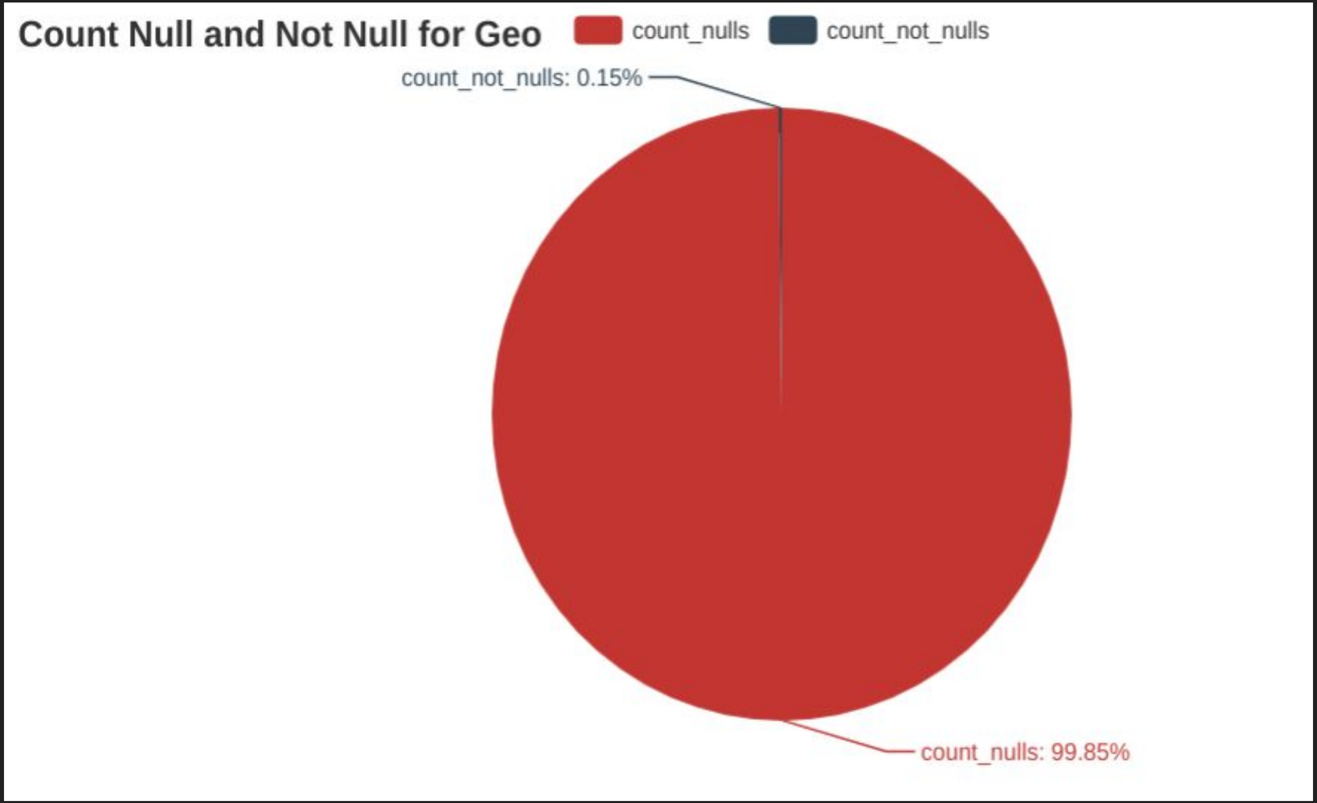
# Increment 1 - Query 3 - Login Source Distribution



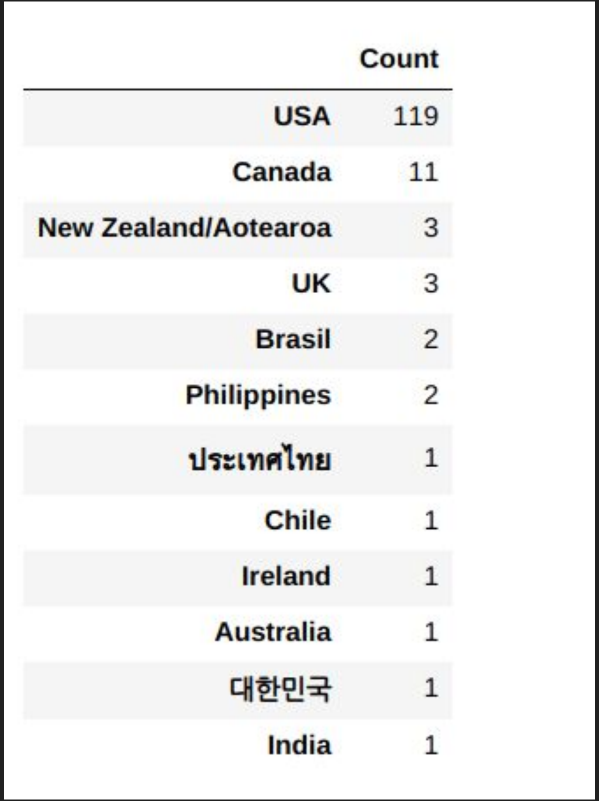
## Increment 1 - Query 4 - Word Cloud



## Increment 1 - Query 5 - With/Without Geo

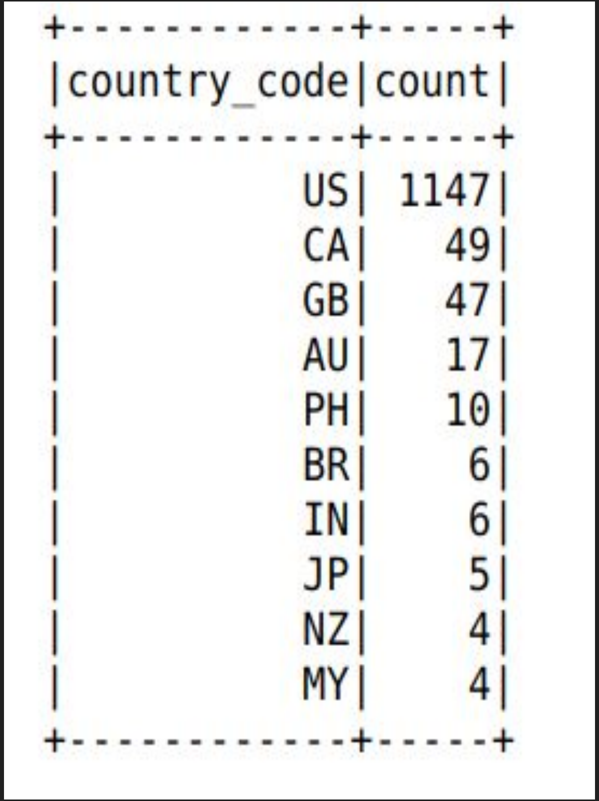


## Increment 1 - Query 6 - Country Distribution (Geo)



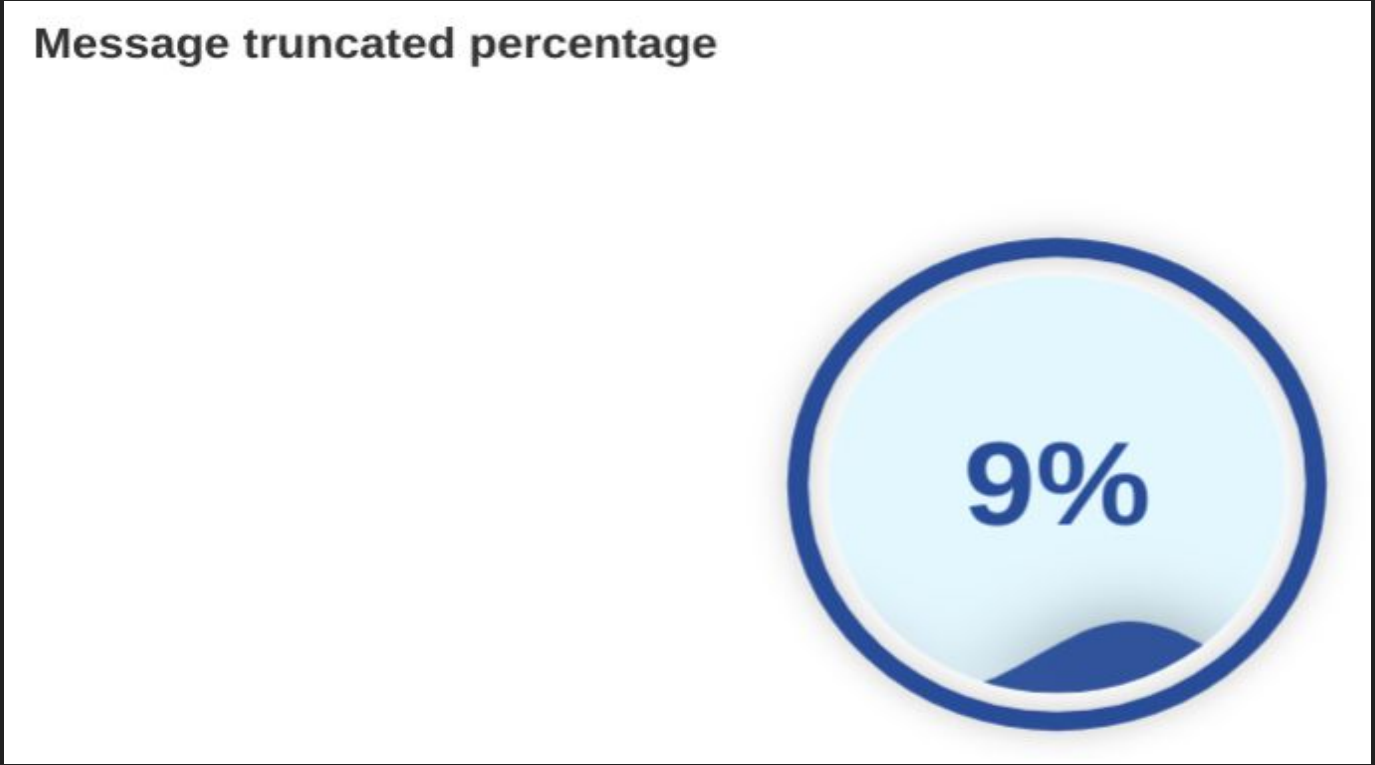
| | Count |
|---|---|
| USA | 119 |
| Canada | 11 |
| New Zealand/Aotearoa | 3 |
| UK | 3 |
| Brasil | 2 |
| Philippines | 2 |
| ประเทศไทย | 1 |
| Chile | 1 |
| Ireland | 1 |
| Australia | 1 |
| 대한민국 | 1 |
| India | 1 |

## Increment 1 - Query 7 - Country Distribution (Placed)

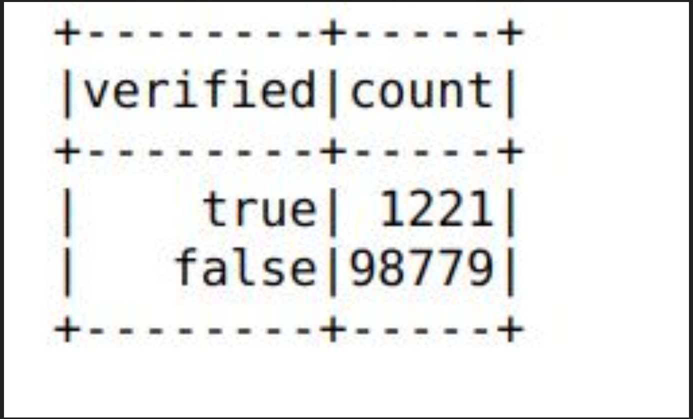


```
+------------+-----+
|country_code|count|
+------------+-----+
|          US| 1147|
|          CA|   49|
|          GB|   47|
|          AU|   17|
|          PH|   10|
|          BR|    6|
|          IN|    6|
|          JP|    5|
|          NZ|    4|
|          MY|    4|
+------------+-----+
```

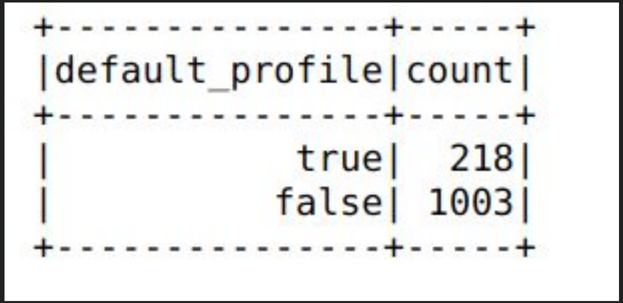
## Increment 1 - Query 8 - Long Message Percentage



# Increment 1 - Query 9 - With/Without Verified

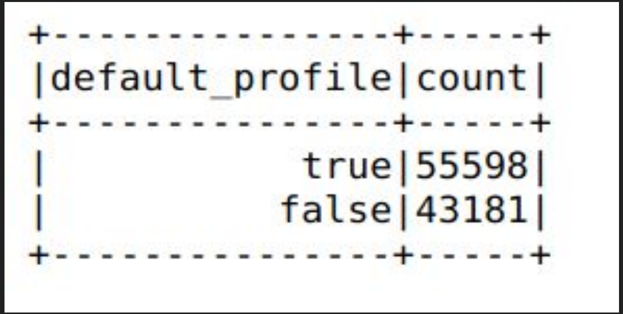


```
+--------+-----+
|verified|count|
+--------+-----+
|    true| 1221|
|   false|98779|
+--------+-----+
```



```
+---------------+-----+
|default_profile|count|
+---------------+-----+
|           true|  218|
|          false| 1003|
+---------------+-----+
```

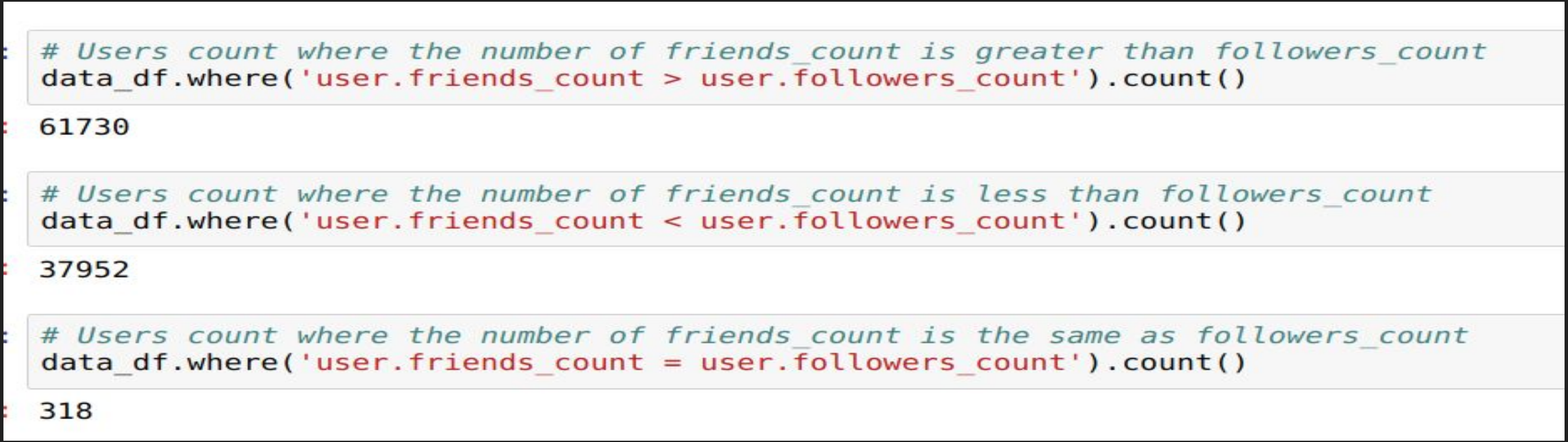
Verified Accounts



```
+---------------+-----+
|default_profile|count|
+---------------+-----+
|           true|55598|
|          false|43181|
+---------------+-----+
```

Unverified Accounts

## Increment 1 - Query 10 - Friends VS. Followers



```
# Users count where the number of friends_count is greater than followers_count
data_df.where('user.friends_count > user.followers_count').count()
```

```
61730
```

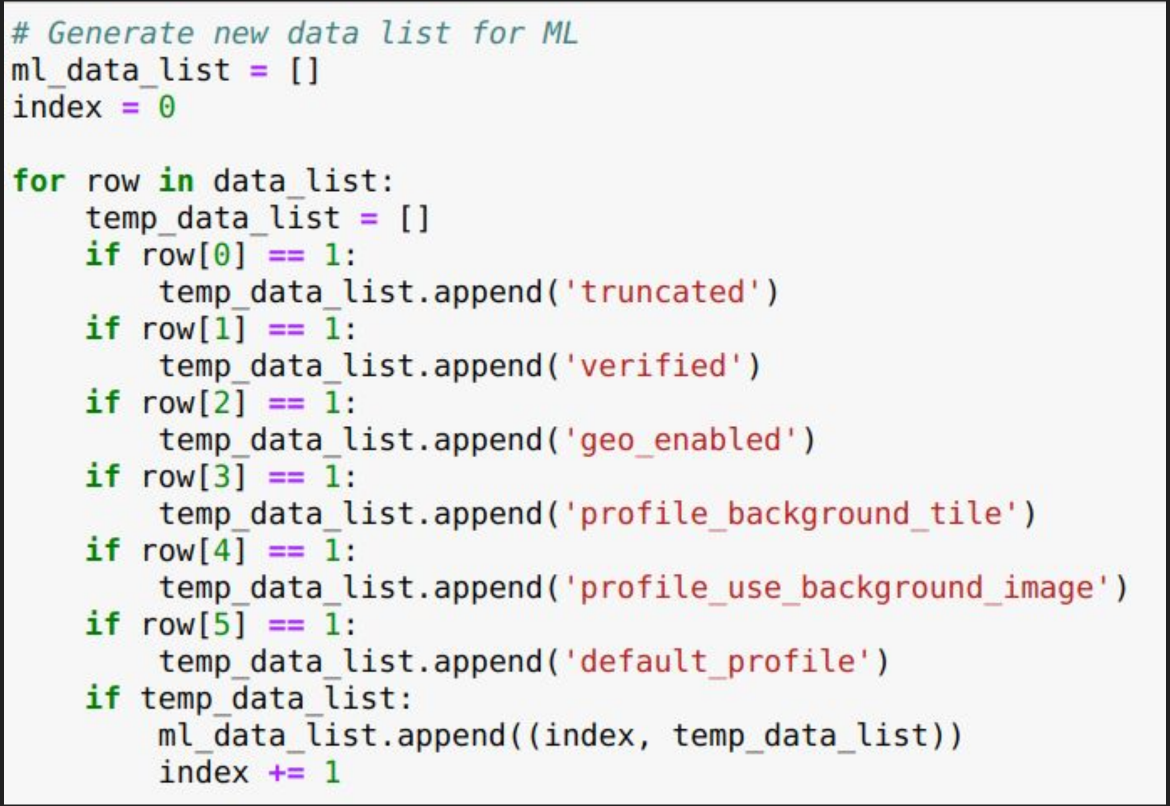
```
# Users count where the number of friends_count is less than followers_count
data_df.where('user.friends_count < user.followers_count').count()
```

```
37952
```

```
# Users count where the number of friends_count is the same as followers_count
data_df.where('user.friends_count = user.followers_count').count()
```

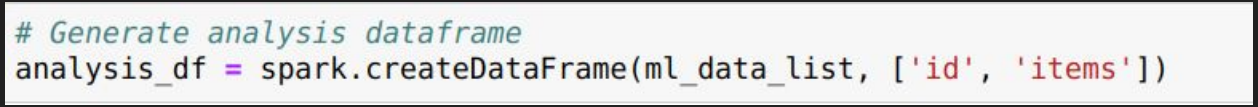
```
318
```

## Increment 2 - ML - Features

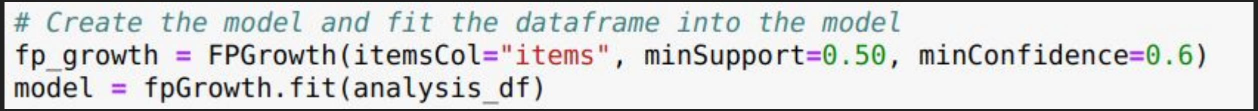


```python
# Generate new data list for ML
ml_data_list = []
index = 0

for row in data_list:
    temp_data_list = []
    if row[0] == 1:
        temp_data_list.append('truncated')
    if row[1] == 1:
        temp_data_list.append('verified')
    if row[2] == 1:
        temp_data_list.append('geo_enabled')
    if row[3] == 1:
        temp_data_list.append('profile_background_tile')
    if row[4] == 1:
        temp_data_list.append('profile_use_background_image')
    if row[5] == 1:
        temp_data_list.append('default_profile')
    if temp_data_list:
        ml_data_list.append((index, temp_data_list))
        index += 1
```

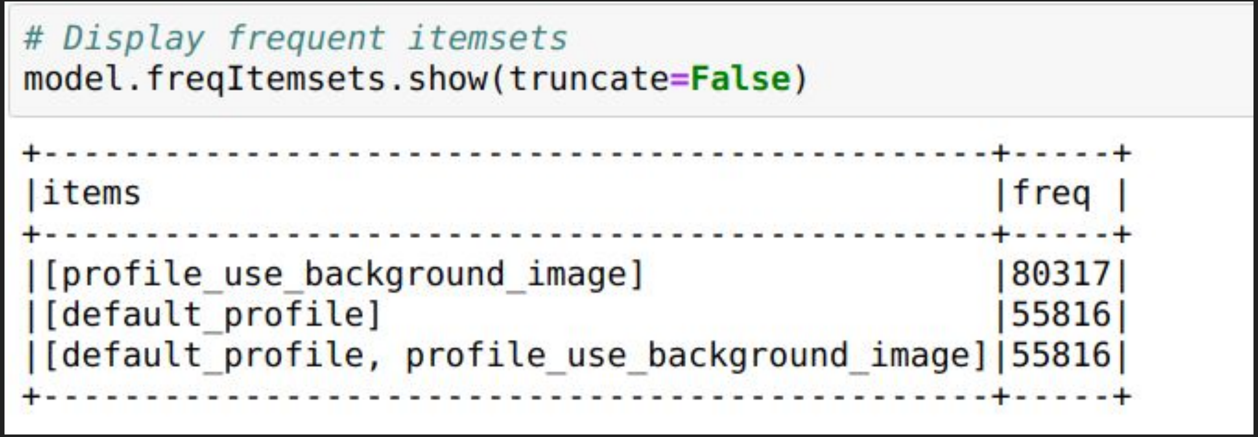
# Increment 2 - ML - Model



```python
# Generate analysis dataframe
analysis_df = spark.createDataFrame(ml_data_list, ['id', 'items'])
```



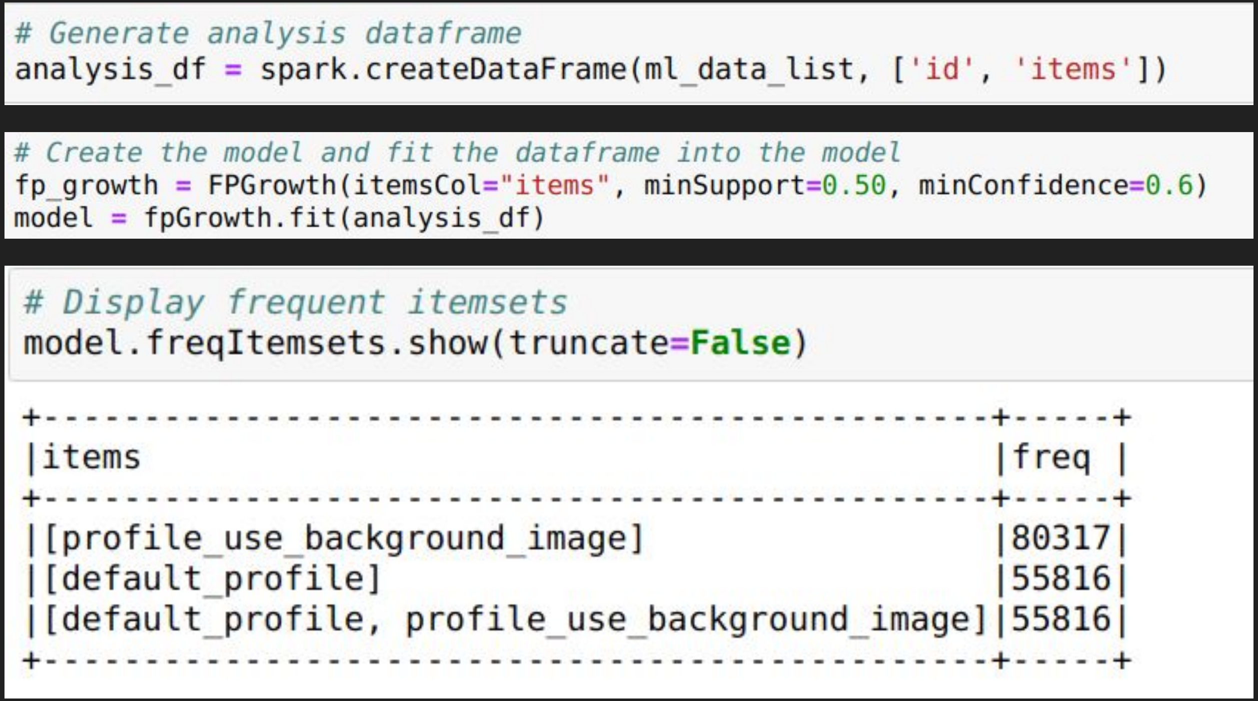
```python
# Create the model and fit the dataframe into the model
fp_growth = FPGrowth(itemsCol="items", minSupport=0.50, minConfidence=0.6)
model = fpGrowth.fit(analysis_df)
```



```python
# Display frequent itemsets
model.freqItemsets.show(truncate=False)
```

```
+---------------------------------------------+-----+
|items                                        |freq |
+---------------------------------------------+-----+
|[profile_use_background_image]               |80317|
|[default_profile]                            |55816|
|[default_profile, profile_use_background_image]|55816|
+---------------------------------------------+-----+
```

## Increment 2 - ML - Result



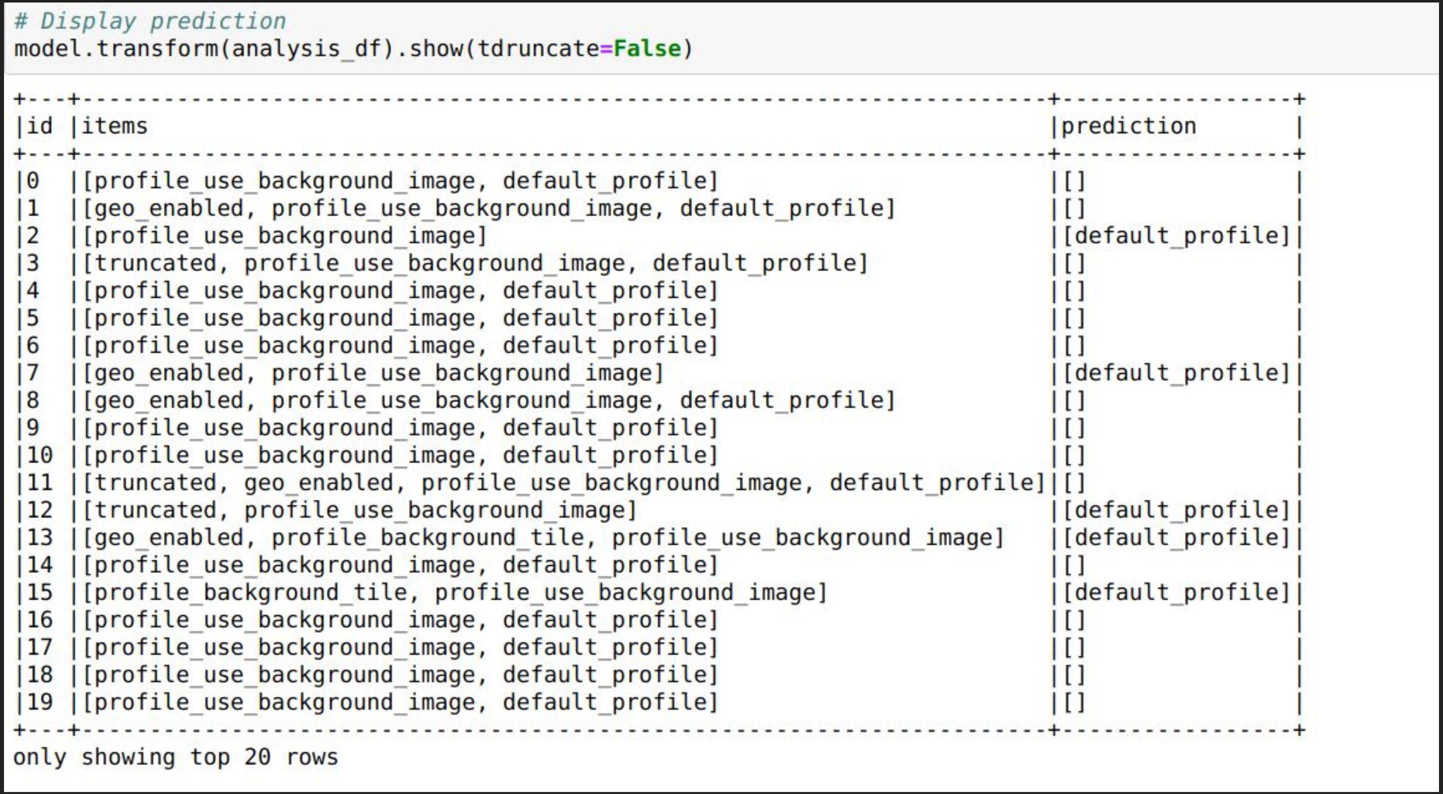
```
# Generate analysis dataframe
analysis_df = spark.createDataFrame(ml_data_list, ['id', 'items'])
```

```
# Create the model and fit the dataframe into the model
fp_growth = FPGrowth(itemsCol="items", minSupport=0.50, minConfidence=0.6)
model = fpGrowth.fit(analysis_df)
```

```
# Display frequent itemsets
model.freqItemsets.show(truncate=False)
```

```
+---------------------------------------------+-----+
|items                                        |freq |
+---------------------------------------------+-----+
|[profile_use_background_image]               |80317|
|[default_profile]                            |55816|
|[default_profile, profile_use_background_image]|55816|
+---------------------------------------------+-----+
```

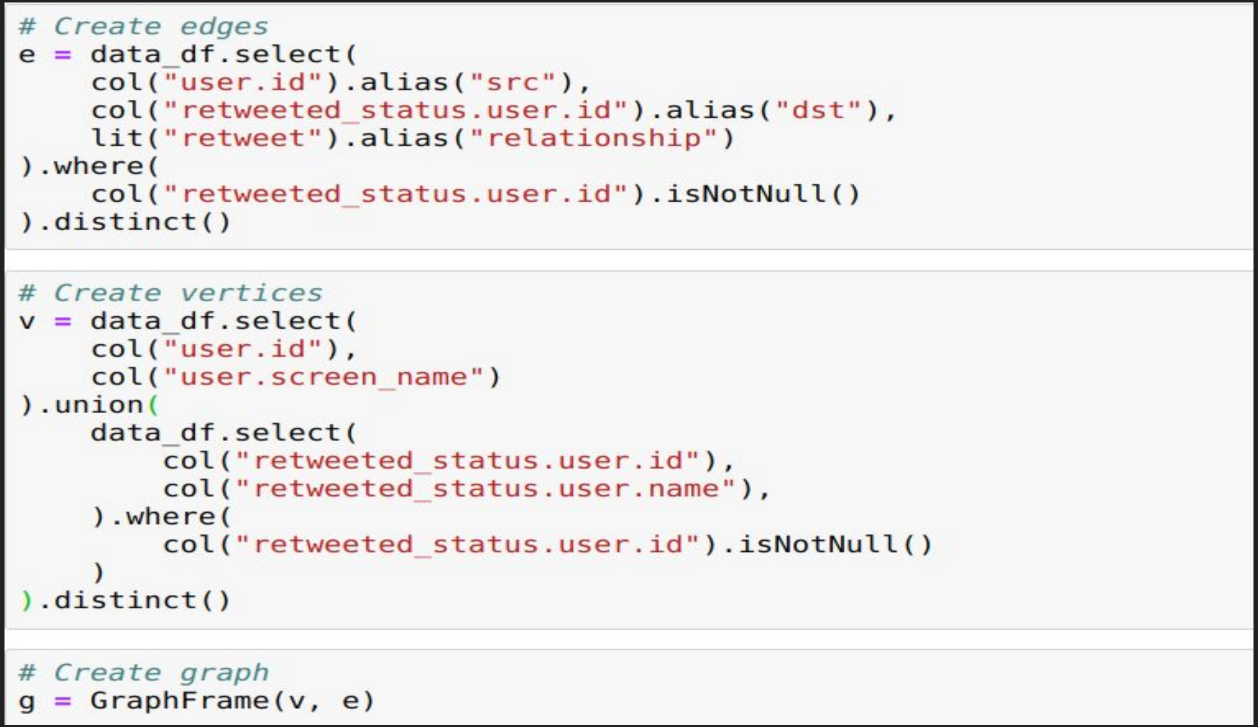
# Increment 2 - ML - Recommendations



```
# Display prediction
model.transform(analysis_df).show(tdruncate=False)
```

```
+---+------------------------------------------------------------------------+-----------------+
|id |items                                                                   |prediction       |
+---+------------------------------------------------------------------------+-----------------+
|0  |[profile_use_background_image, default_profile]                         |[]               |
|1  |[geo_enabled, profile_use_background_image, default_profile]            |[]               |
|2  |[profile_use_background_image]                                          |[default_profile]|
|3  |[truncated, profile_use_background_image, default_profile]              |[]               |
|4  |[profile_use_background_image, default_profile]                         |[]               |
|5  |[profile_use_background_image, default_profile]                         |[]               |
|6  |[profile_use_background_image, default_profile]                         |[]               |
|7  |[geo_enabled, profile_use_background_image]                             |[default_profile]|
|8  |[geo_enabled, profile_use_background_image, default_profile]            |[]               |
|9  |[profile_use_background_image, default_profile]                         |[]               |
|10 |[profile_use_background_image, default_profile]                         |[]               |
|11 |[truncated, geo_enabled, profile_use_background_image, default_profile]|[]               |
|12 |[truncated, profile_use_background_image]                               |[default_profile]|
|13 |[geo_enabled, profile_background_tile, profile_use_background_image]    |[default_profile]|
|14 |[profile_use_background_image, default_profile]                         |[]               |
|15 |[profile_background_tile, profile_use_background_image]                 |[default_profile]|
|16 |[profile_use_background_image, default_profile]                         |[]               |
|17 |[profile_use_background_image, default_profile]                         |[]               |
|18 |[profile_use_background_image, default_profile]                         |[]               |
|19 |[profile_use_background_image, default_profile]                         |[]               |
+---+------------------------------------------------------------------------+-----------------+
only showing top 20 rows
```

## Increment 2 - Graph - Graph Creation

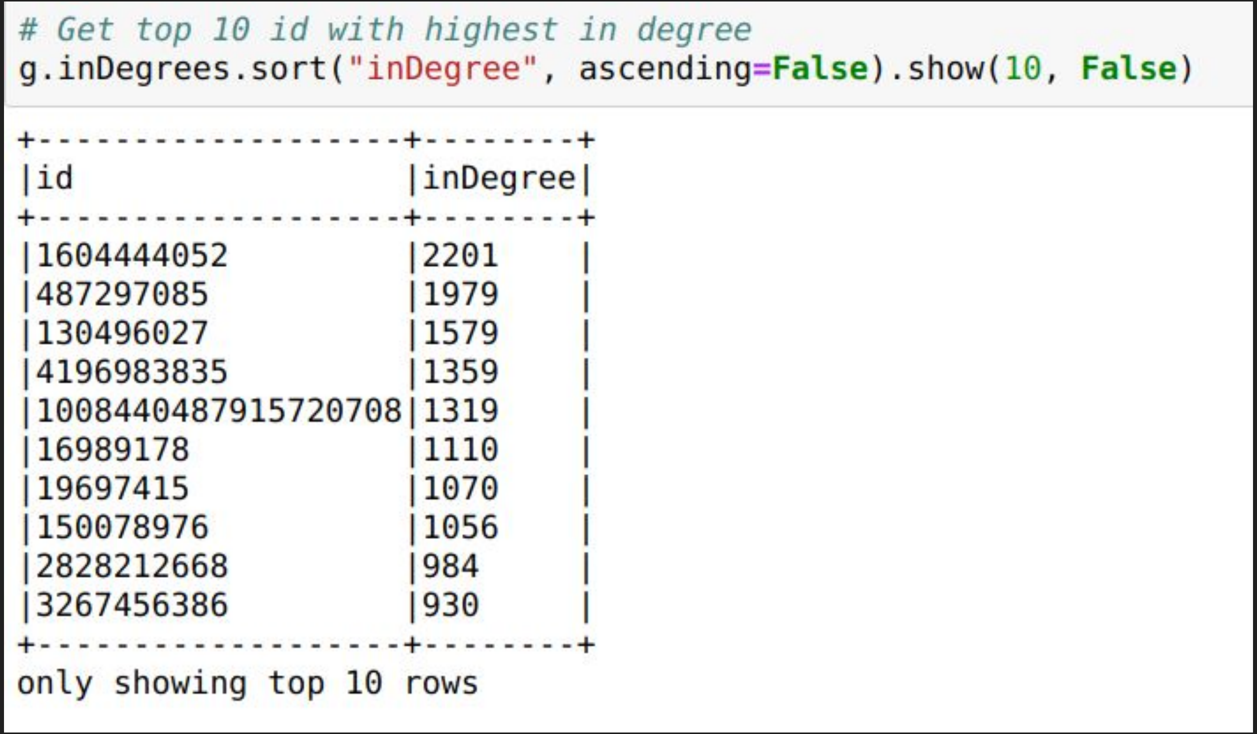


```
# Create edges
e = data_df.select(
    col("user.id").alias("src"),
    col("retweeted_status.user.id").alias("dst"),
    lit("retweet").alias("relationship")
).where(
    col("retweeted_status.user.id").isNotNull()
).distinct()
```

```
# Create vertices
v = data_df.select(
    col("user.id"),
    col("user.screen_name")
).union(
    data_df.select(
        col("retweeted_status.user.id"),
        col("retweeted_status.user.name"),
    ).where(
        col("retweeted_status.user.id").isNotNull()
    )
).distinct()
```

```
# Create graph
g = GraphFrame(v, e)
```

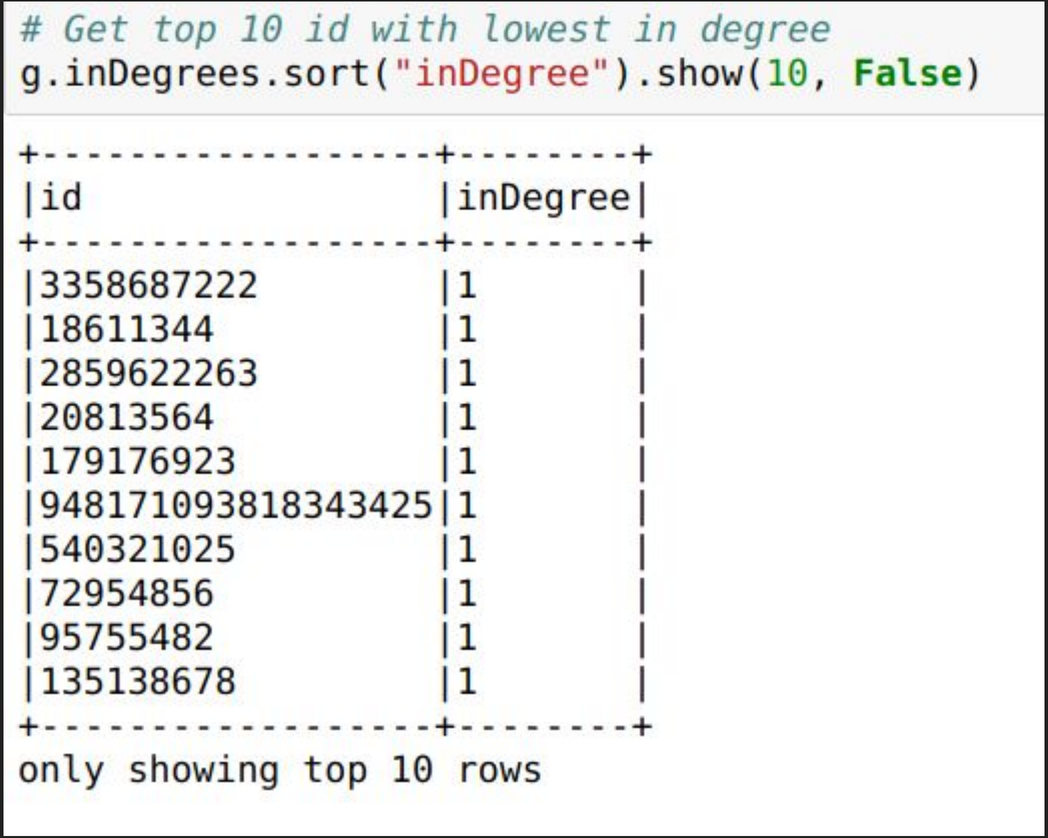
## Increment 2 - Graph - In-degree



```
# Get top 10 id with highest in degree
g.inDegrees.sort("inDegree", ascending=False).show(10, False)
```

```
+-------------------+--------+
|id                 |inDegree|
+-------------------+--------+
|1604444052         |2201    |
|487297085          |1979    |
|130496027          |1579    |
|4196983835         |1359    |
|1008440487915720708|1319    |
|16989178           |1110    |
|19697415           |1070    |
|150078976          |1056    |
|2828212668         |984     |
|3267456386         |930     |
+-------------------+--------+
only showing top 10 rows
```

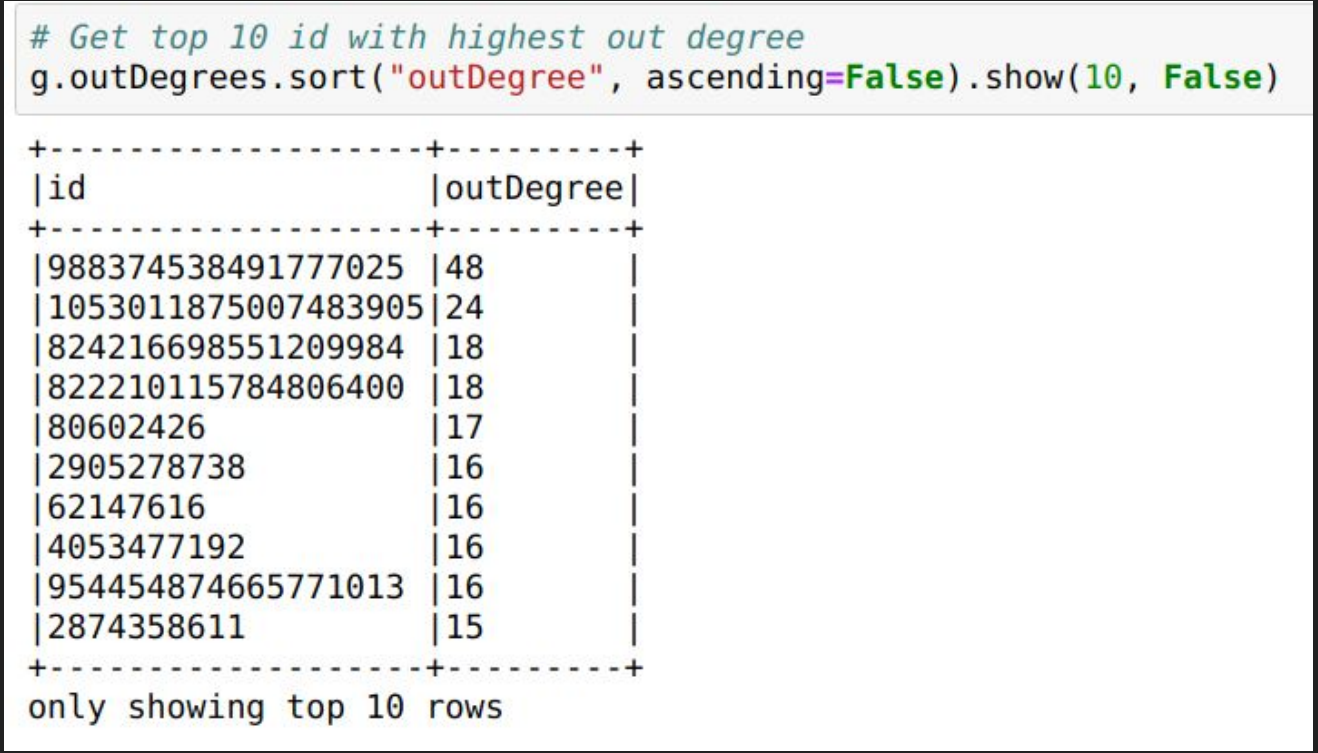
## Increment 2 - Graph - In-degree (sorted)



```
# Get top 10 id with lowest in degree
g.inDegrees.sort("inDegree").show(10, False)
```

```
+------------------+--------+
|id                |inDegree|
+------------------+--------+
|3358687222        |1       |
|18611344          |1       |
|2859622263        |1       |
|20813564          |1       |
|179176923         |1       |
|948171093818343425|1       |
|540321025         |1       |
|72954856          |1       |
|95755482          |1       |
|135138678         |1       |
+------------------+--------+
only showing top 10 rows
```

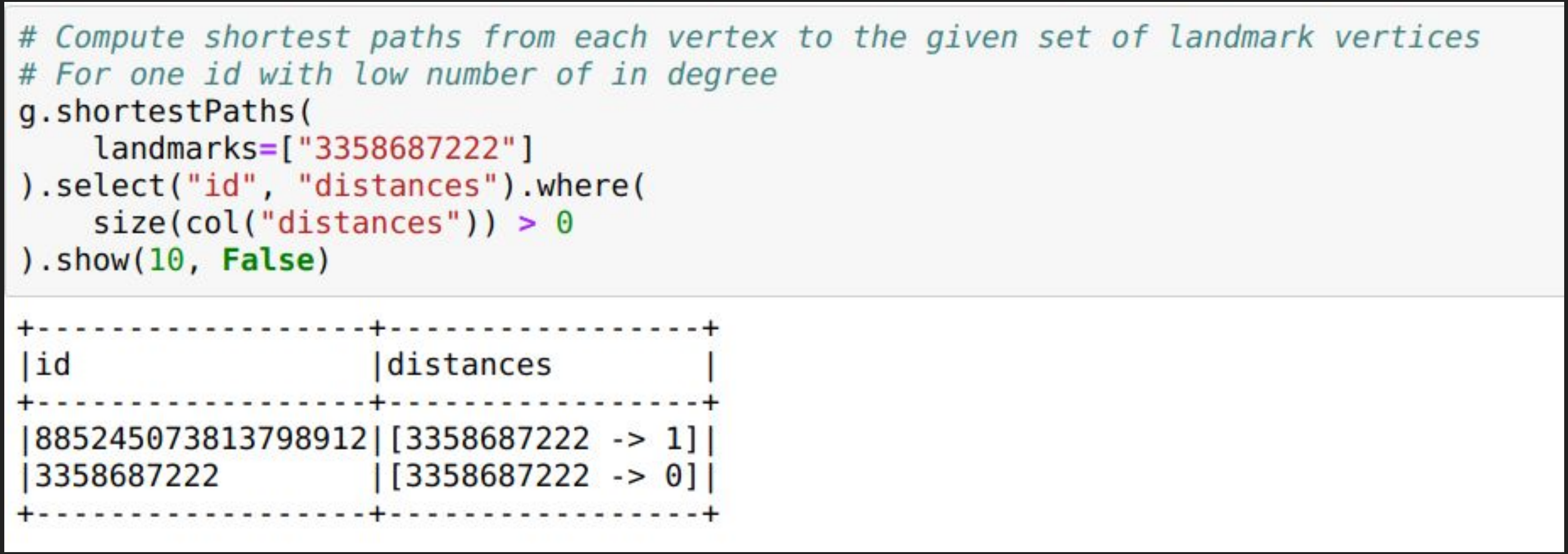
# Increment 2 - Graph - Out-degree



```
# Get top 10 id with highest out degree
g.outDegrees.sort("outDegree", ascending=False).show(10, False)
```

```
+-------------------+---------+
|id                 |outDegree|
+-------------------+---------+
|988374538491777025 |48       |
|1053011875007483905|24       |
|824216698551209984 |18       |
|822210115784806400 |18       |
|80602426           |17       |
|2905278738         |16       |
|62147616           |16       |
|4053477192         |16       |
|954454874665771013 |16       |
|2874358611         |15       |
+-------------------+---------+
only showing top 10 rows
```

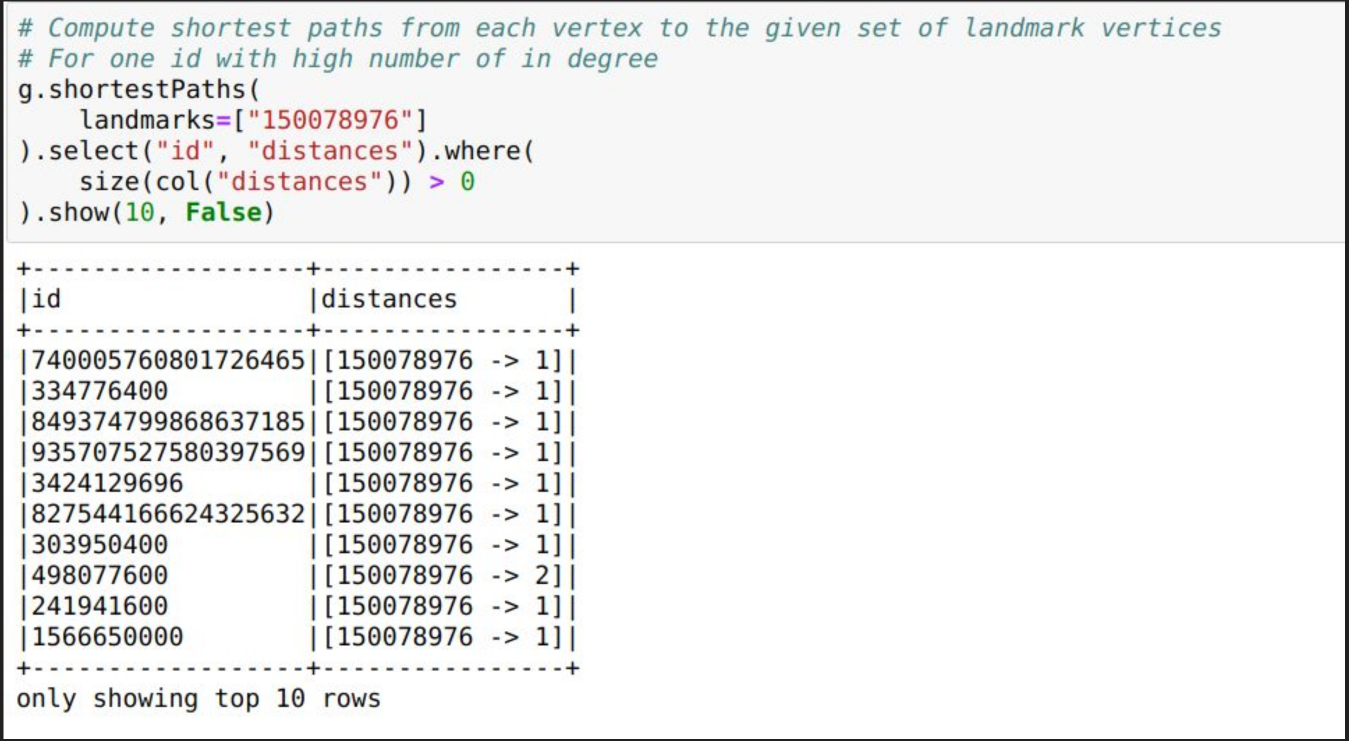
# Increment 2 - Graph - Shortest Paths  - Less Popular Landmarks



```python
# Compute shortest paths from each vertex to the given set of landmark vertices
# For one id with low number of in degree
g.shortestPaths(
    landmarks=["3358687222"]
).select("id", "distances").where(
    size(col("distances")) > 0
).show(10, False)
```

```
+------------------+-----------------+
|id                |distances        |
+------------------+-----------------+
|885245073813798912|[3358687222 -> 1]|
|3358687222        |[3358687222 -> 0]|
+------------------+-----------------+
```

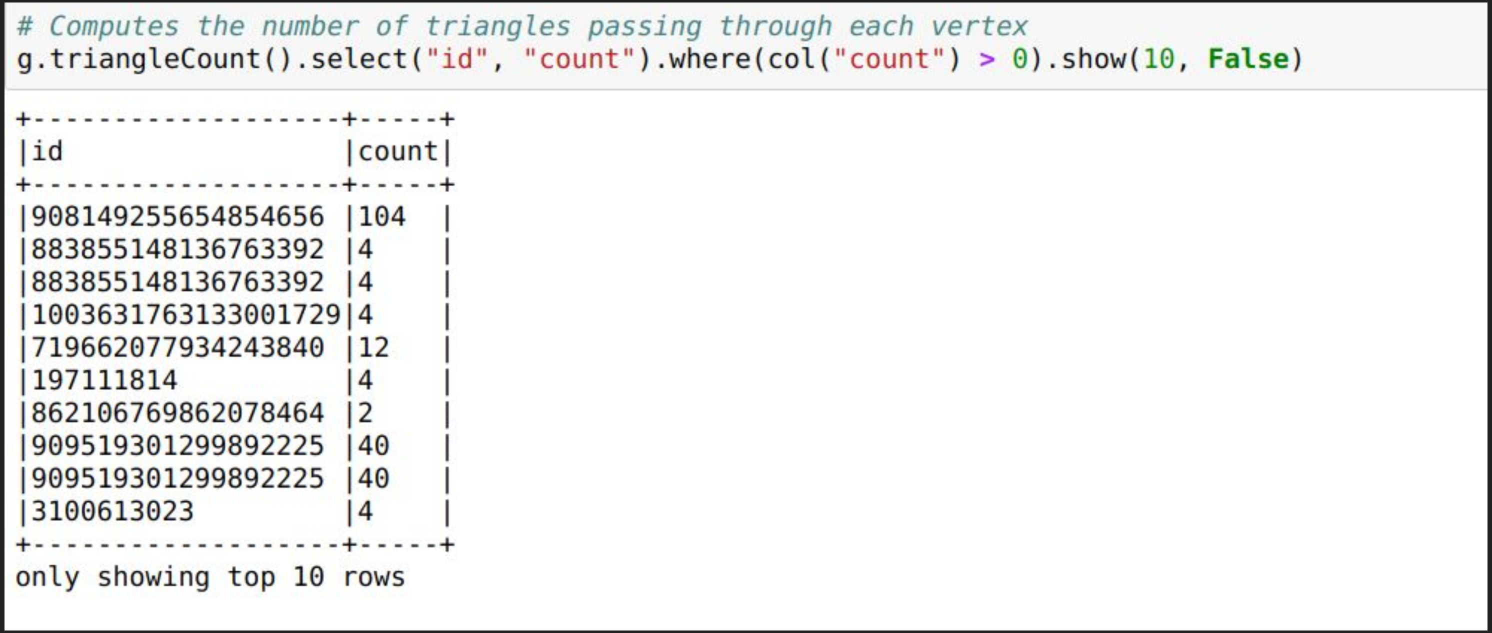
## Increment 2 - Graph - Shortest Paths  - Popular Landmarks



```
# Compute shortest paths from each vertex to the given set of landmark vertices
# For one id with high number of in degree
g.shortestPaths(
    landmarks=["150078976"]
).select("id", "distances").where(
    size(col("distances")) > 0
).show(10, False)
```

```
+------------------+----------------+
|id                |distances       |
+------------------+----------------+
|740005760801726465|[150078976 -> 1]|
|334776400         |[150078976 -> 1]|
|849374799868637185|[150078976 -> 1]|
|935707527580397569|[150078976 -> 1]|
|3424129696        |[150078976 -> 1]|
|827544166624325632|[150078976 -> 1]|
|303950400         |[150078976 -> 1]|
|498077600         |[150078976 -> 2]|
|241941600         |[150078976 -> 1]|
|1566650000        |[150078976 -> 1]|
+------------------+----------------+
only showing top 10 rows
```

# Increment 2 - Graph - Shortest Paths  - Triangle Count

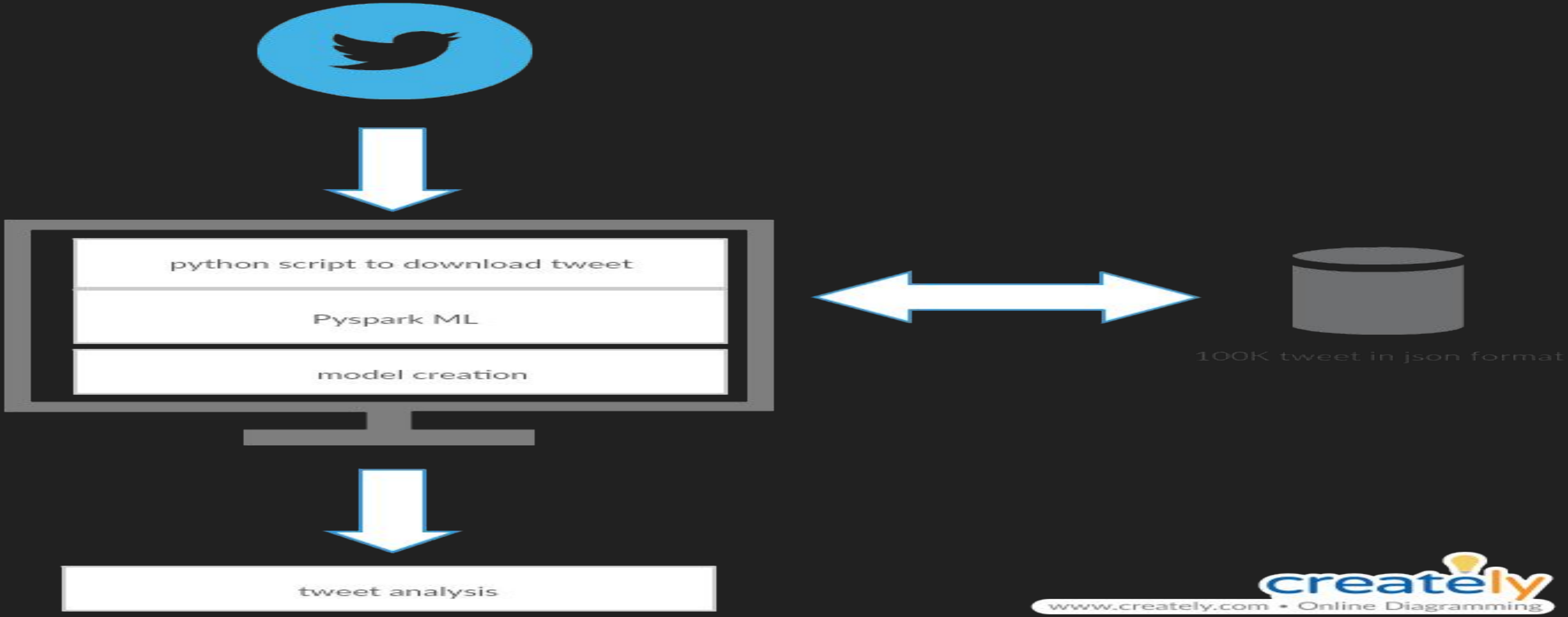


```
# Computes the number of triangles passing through each vertex
g.triangleCount().select("id", "count").where(col("count") > 0).show(10, False)
```

```
+-------------------+-----+
|id                 |count|
+-------------------+-----+
|908149255654854656 |104  |
|883855148136763392 |4    |
|883855148136763392 |4    |
|1003631763133001729|4    |
|719662077934243840 |12   |
|197111814          |4    |
|862106769862078464 |2    |
|909519301299892225 |40   |
|909519301299892225 |40   |
|3100613023         |4    |
+-------------------+-----+
only showing top 10 rows
```

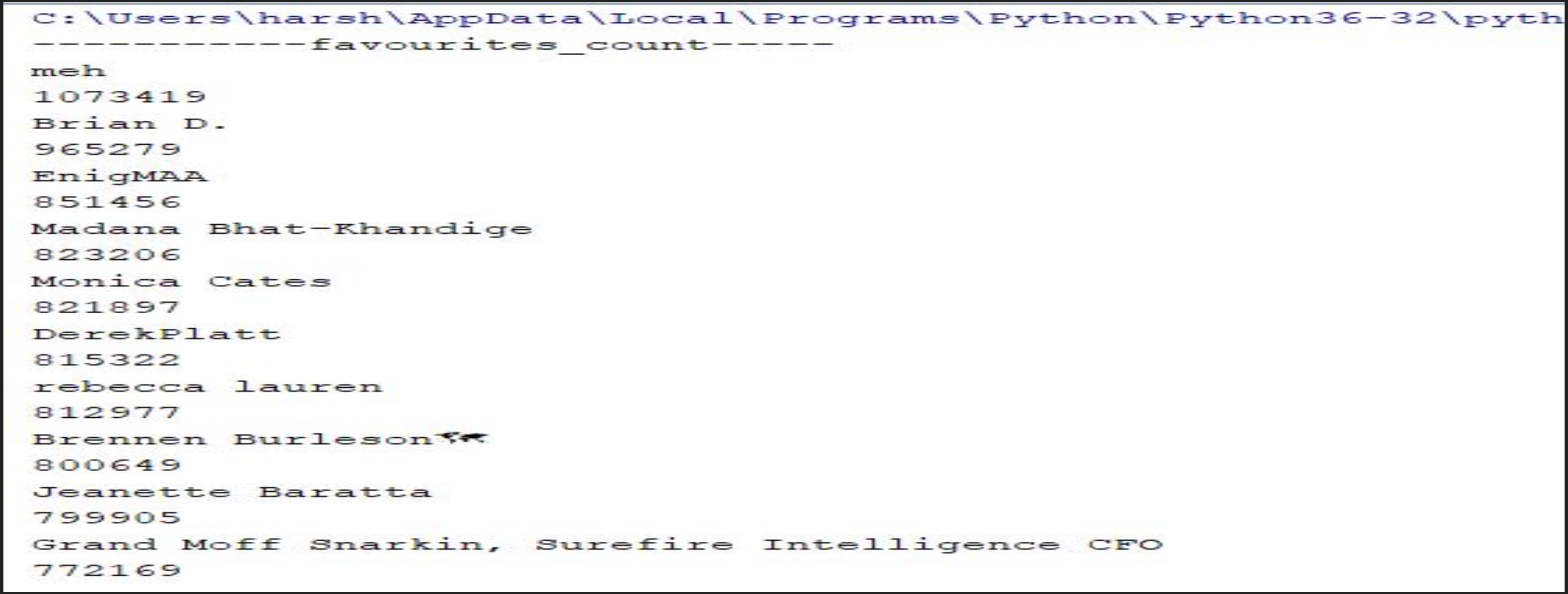
# Harshil

## Model



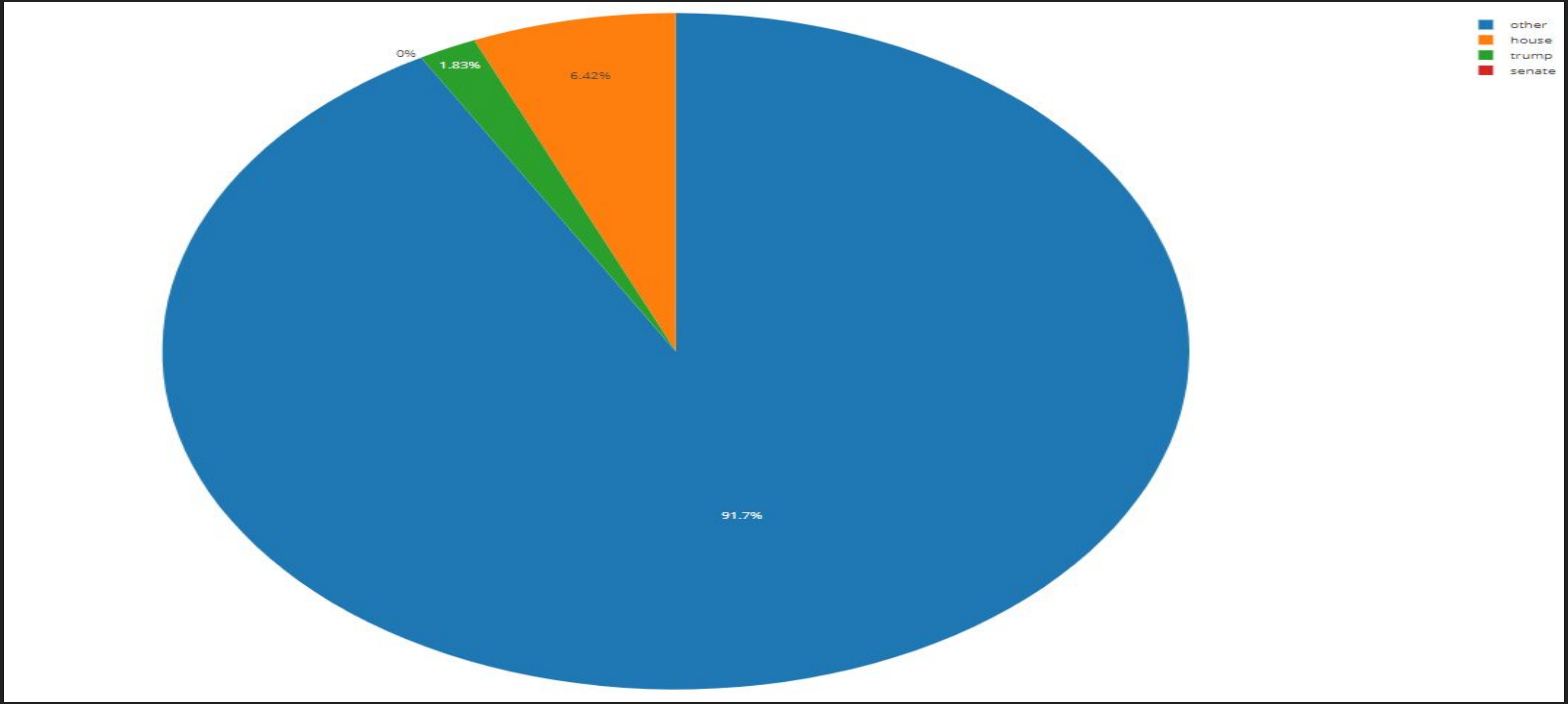
python script to download tweet
Pyspark ML
model creation
100K tweet in json format
tweet analysis
creately
www.creately.com • Online Diagramming

## Get top 10 user which have most favourites count

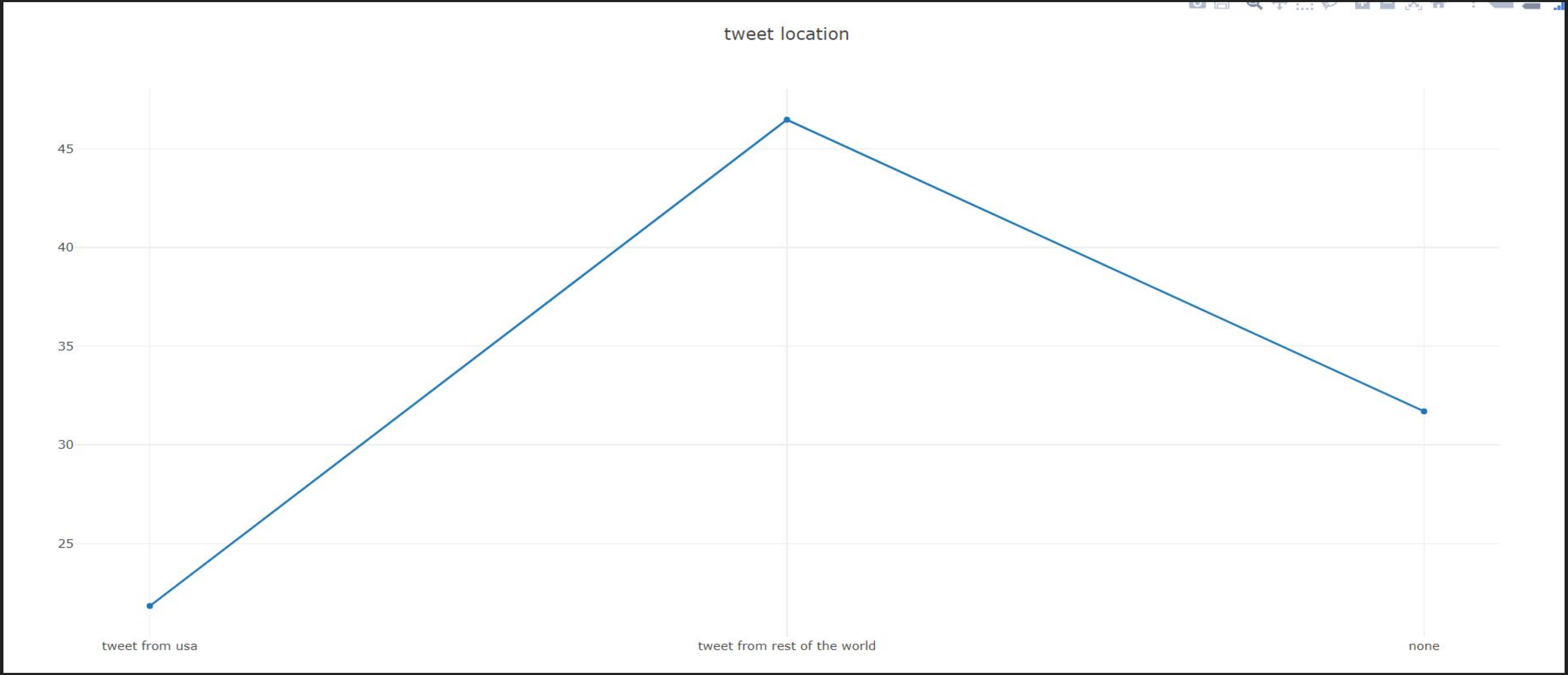


```
C:\Users\harsh\AppData\Local\Programs\Python\Python36-32\pyth
-----------favourites_count-----
meh
1073419
Brian D.
965279
EnigMAA
851456
Madana Bhat-Khandige
823206
Monica Cates
821897
DerekPlatt
815322
rebecca lauren
812977
Brennen Burleson
800649
Jeanette Baratta
799905
Grand Moff Snarkin, Surefire Intelligence CFO
772169
```

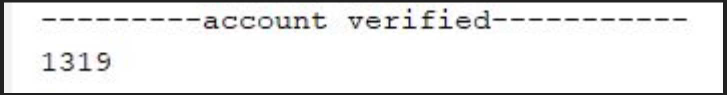
## Find the tracks in the tweet and shows its distribution



## Show tweets generated from united state and other than united states

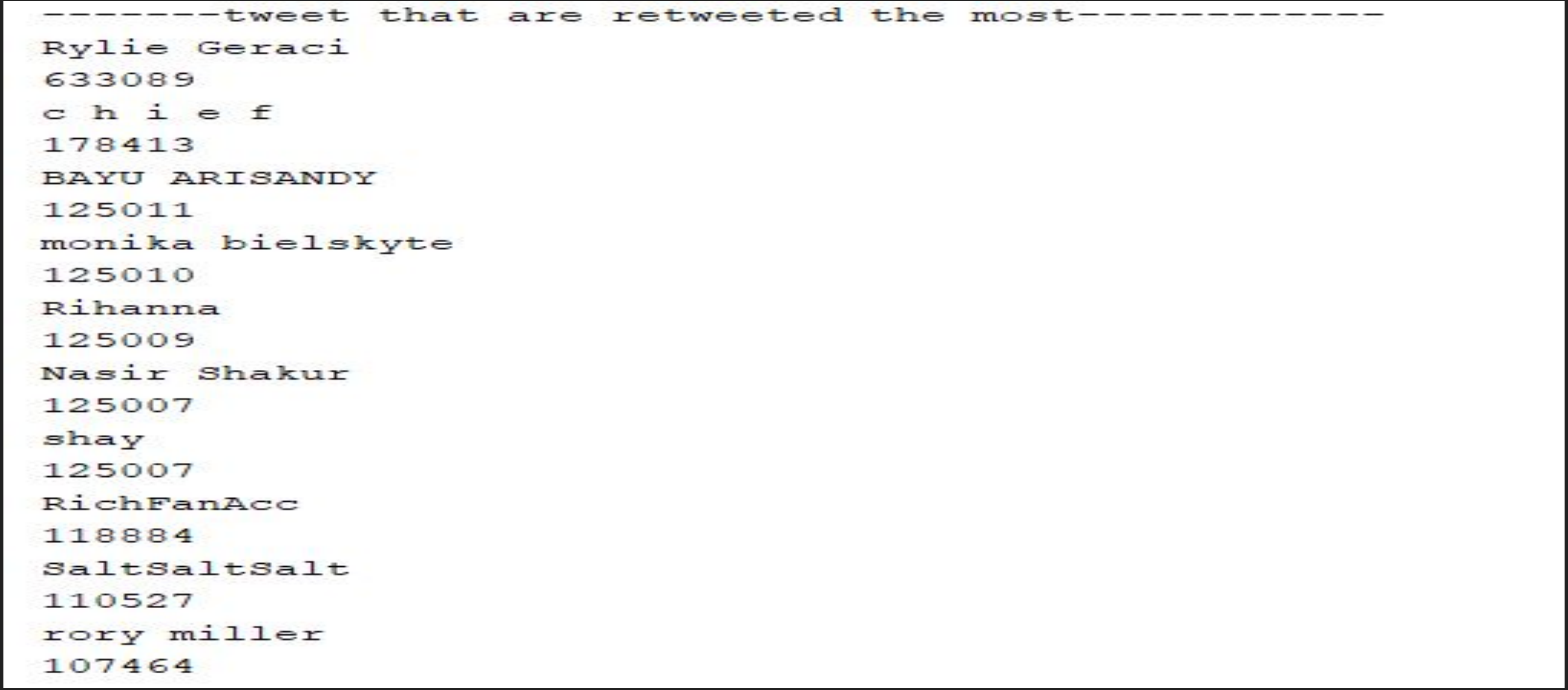


## Get the count of the user which have account verified



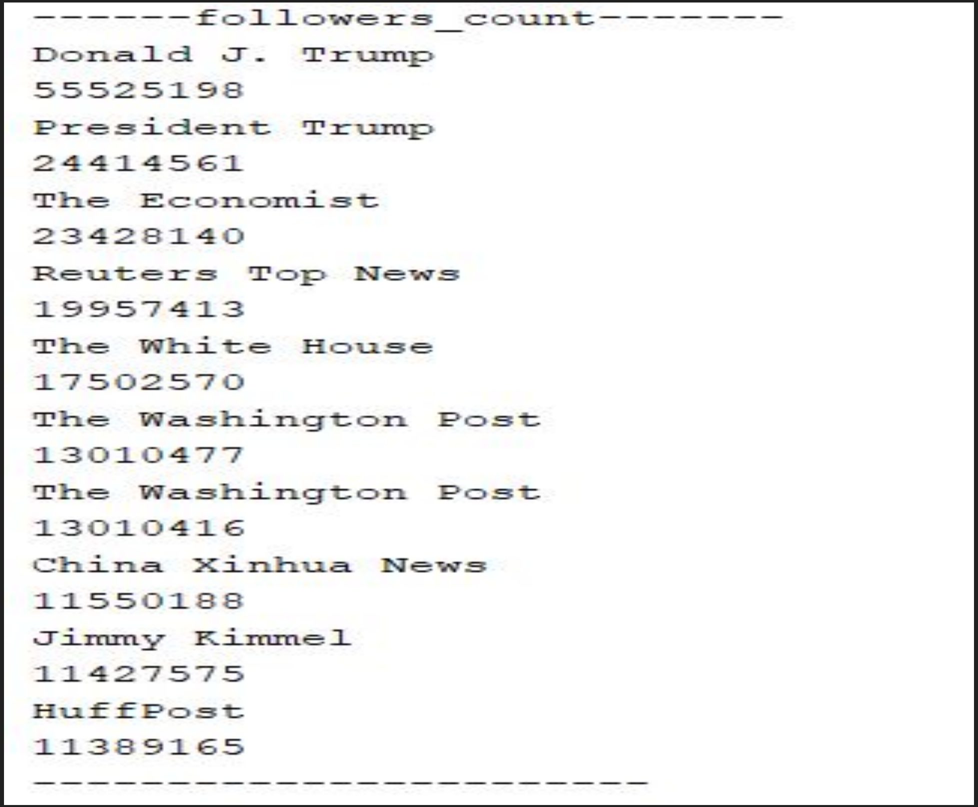
```
---------account verified-----------
1319
```

## Get top 10 tweet that are retweeted the most



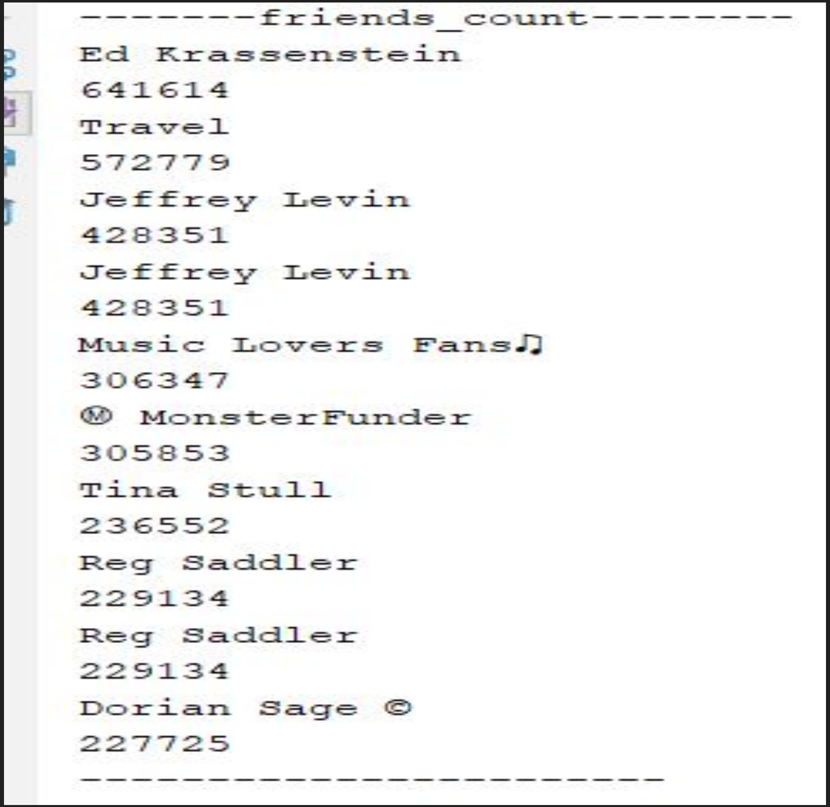
```
-------tweet that are retweeted the most------------
Rylie Geraci
633089
c h i e f
178413
BAYU ARISANDY
125011
monika bielskyte
125010
Rihanna
125009
Nasir Shakur
125007
shay
125007
RichFanAcc
118884
SaltSaltSalt
110527
rory miller
107464
```

## Get top 10 user which have most followers count



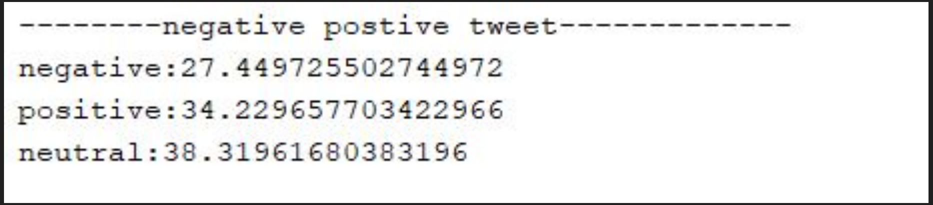
```
------followers_count-------
Donald J. Trump
55525198
President Trump
24414561
The Economist
23428140
Reuters Top News
19957413
The White House
17502570
The Washington Post
13010477
The Washington Post
13010416
China Xinhua News
11550188
Jimmy Kimmel
11427575
HuffPost
11389165
-----------------------
```

## Get top 10 user which have most friends count



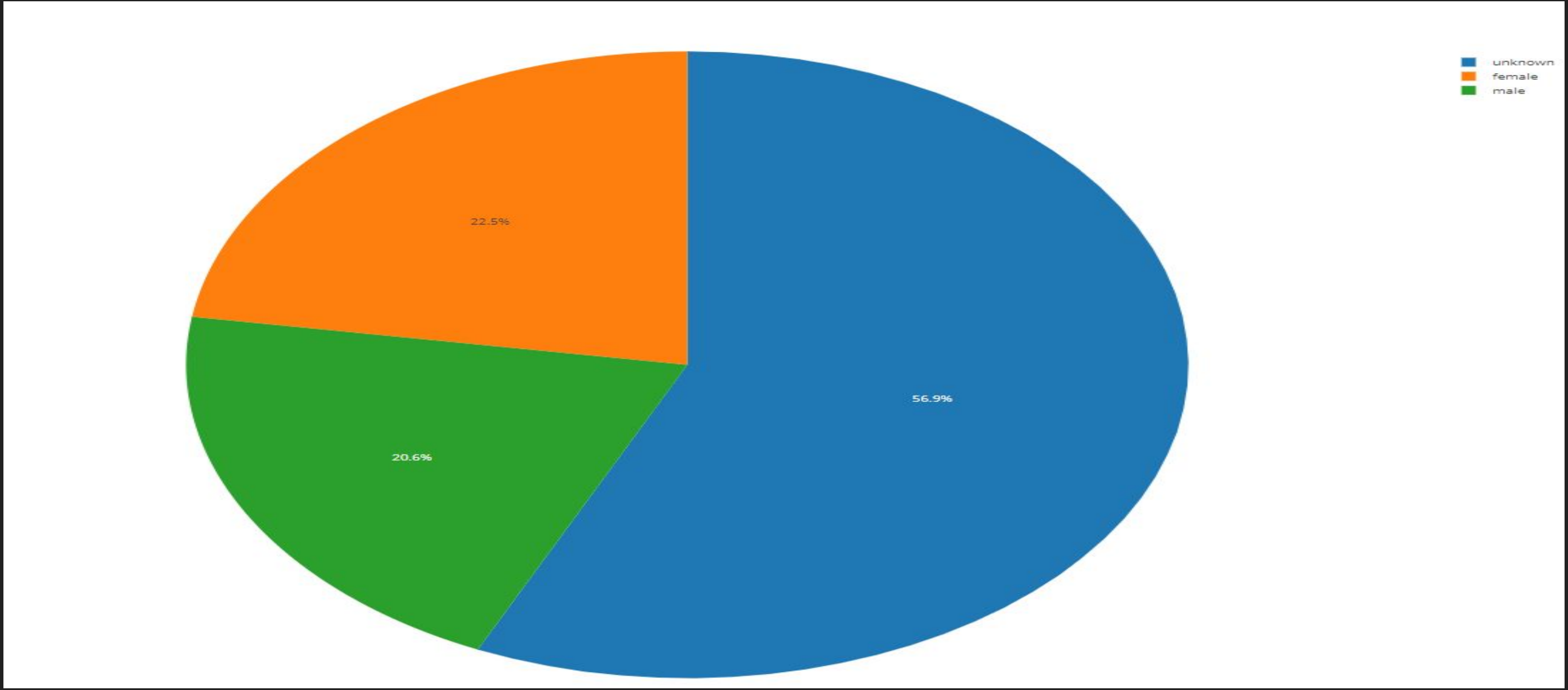
```
-------friends_count--------
Ed Krassenstein
641614
Travel
572779
Jeffrey Levin
428351
Jeffrey Levin
428351
Music Lovers Fans♫
306347
Ⓜ MonsterFunder
305853
Tina Stull
236552
Reg Saddler
229134
Reg Saddler
229134
Dorian Sage ©
227725
-----------------------
```

## Get percentage of tweet based on negative, positive, neutral

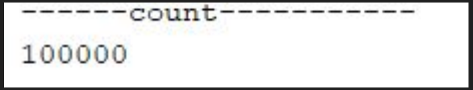


```
--------negative postive tweet-------------
negative:27.449725502744972
positive:34.229657703422966
neutral:38.31961680383196
```

## Find  gender of the user based on their name

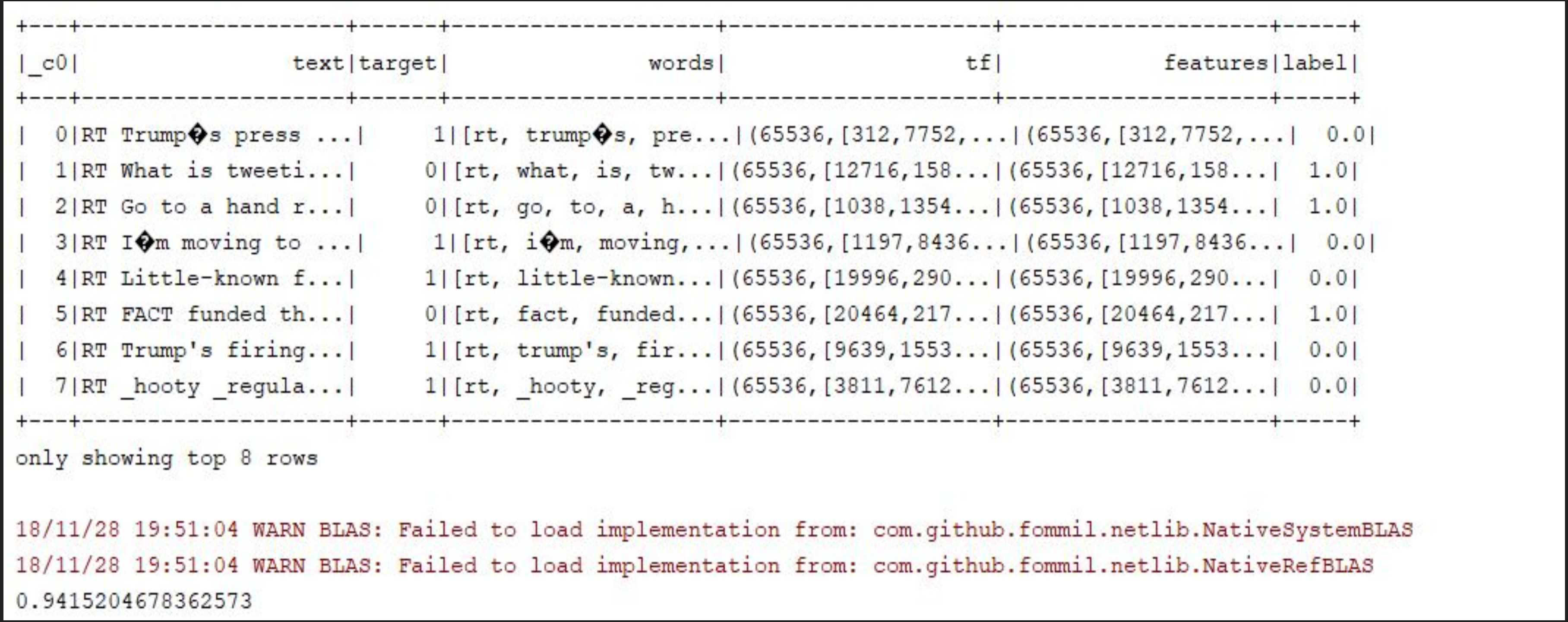


## Get the count of tweets



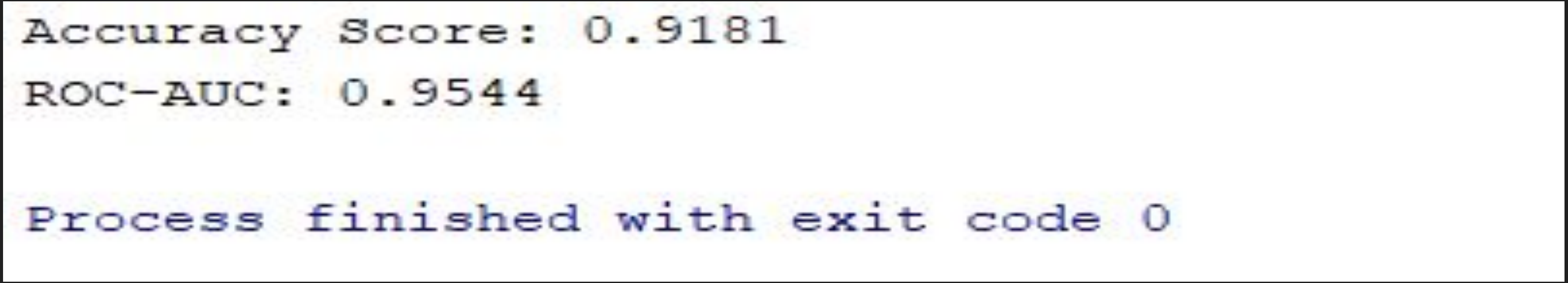
```
------count-----------
100000
```

## HashingTF + IDF + Logistic Regression



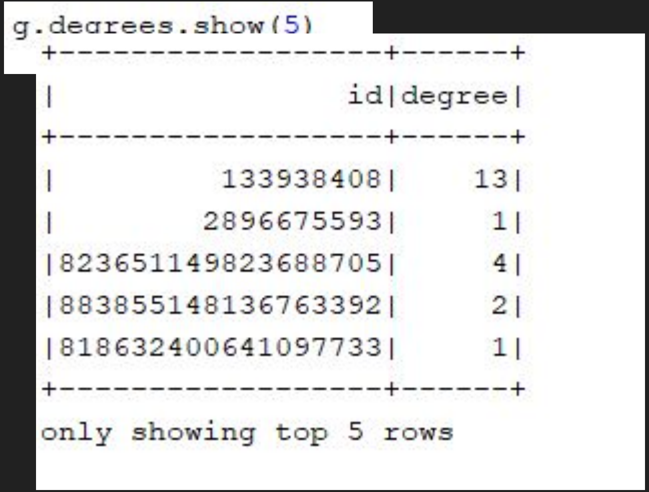
```
+---+--------------------+------+--------------------+--------------------+--------------------+-----+
|_c0|                text|target|               words|                  tf|            features|label|
+---+--------------------+------+--------------------+--------------------+--------------------+-----+
|  0|RT Trump�s press ...|     1|[rt, trump�s, pre...|(65536,[312,7752,...|(65536,[312,7752,...|  0.0|
|  1|RT What is tweeti...|     0|[rt, what, is, tw...|(65536,[12716,158...|(65536,[12716,158...|  1.0|
|  2|RT Go to a hand r...|     0|[rt, go, to, a, h...|(65536,[1038,1354...|(65536,[1038,1354...|  1.0|
|  3|RT I�m moving to ...|     1|[rt, i�m, moving,...|(65536,[1197,8436...|(65536,[1197,8436...|  0.0|
|  4|RT Little-known f...|     1|[rt, little-known...|(65536,[19996,290...|(65536,[19996,290...|  0.0|
|  5|RT FACT funded th...|     0|[rt, fact, funded...|(65536,[20464,217...|(65536,[20464,217...|  1.0|
|  6|RT Trump's firing...|     1|[rt, trump's, fir...|(65536,[9639,1553...|(65536,[9639,1553...|  0.0|
|  7|RT _hooty _regula...|     1|[rt, _hooty, _reg...|(65536,[3811,7612...|(65536,[3811,7612...|  0.0|
+---+--------------------+------+--------------------+--------------------+--------------------+-----+
only showing top 8 rows

18/11/28 19:51:04 WARN BLAS: Failed to load implementation from: com.github.fommil.netlib.NativeSystemBLAS
18/11/28 19:51:04 WARN BLAS: Failed to load implementation from: com.github.fommil.netlib.NativeRefBLAS
0.9415204678362573
```

## CountVectorizer + IDF + Logistic Regression

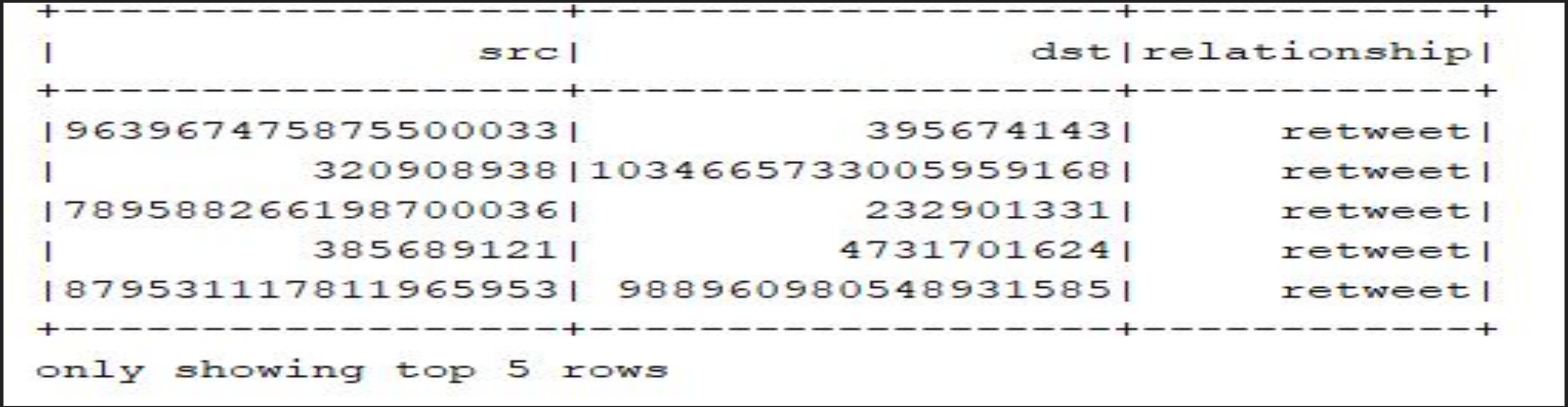


```
Accuracy Score: 0.9181
ROC-AUC: 0.9544


Process finished with exit code 0
```

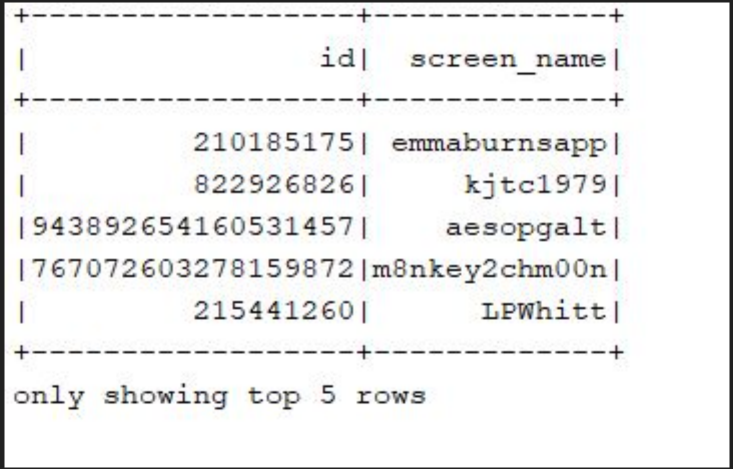
## Graph



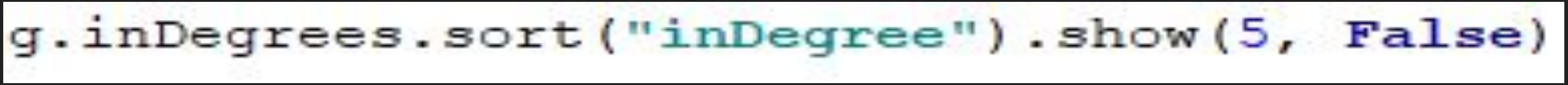
```
g.degrees.show(5)
+------------------+------+
|                id|degree|
+------------------+------+
|         133938408|    13|
|        2896675593|     1|
|823651149823688705|     4|
|883855148136763392|     2|
|818632400641097733|     1|
+------------------+------+
only showing top 5 rows
```



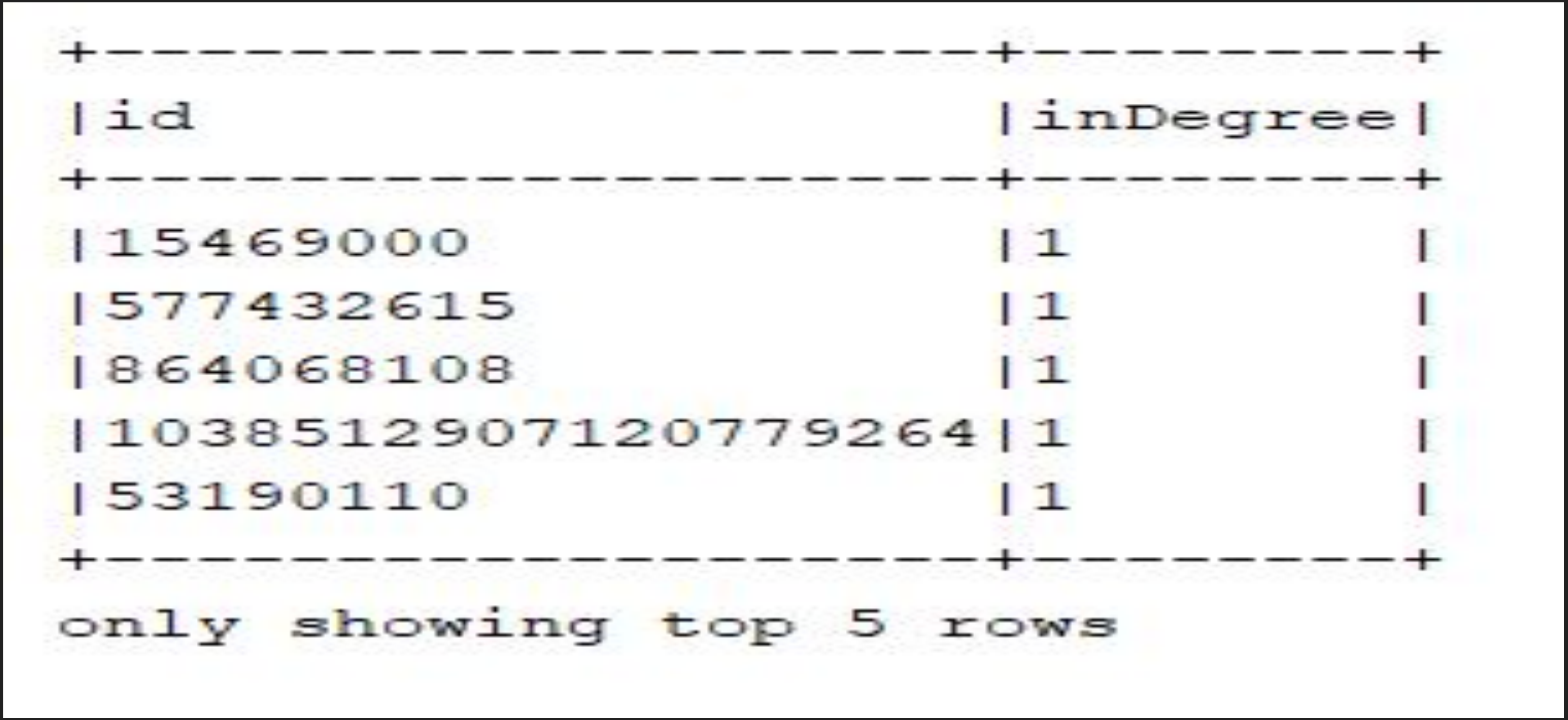
```
+-------------------+-------------------+------------+
|                src|                dst|relationship|
+-------------------+-------------------+------------+
|963967475875500033|          395674143|    retweet|
|          320908938|1034665733005959168|    retweet|
|789588266198700036|          232901331|    retweet|
|          385689121|         4731701624|    retweet|
|879531117811965953| 988960980548931585|    retweet|
+-------------------+-------------------+------------+
only showing top 5 rows
```



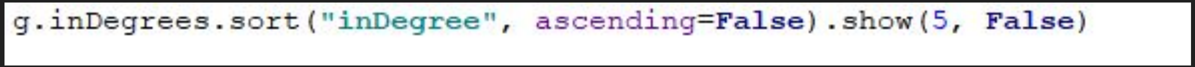
```
+------------------+------------+
|                id| screen_name|
+------------------+------------+
|         210185175| emmaburnsapp|
|         822926826|    kjtc1979|
|943892654160531457|   aesopgalt|
|767072603278159872|m8nkey2chm00n|
|         215441260|     LPWhitt|
+------------------+------------+
only showing top 5 rows
```



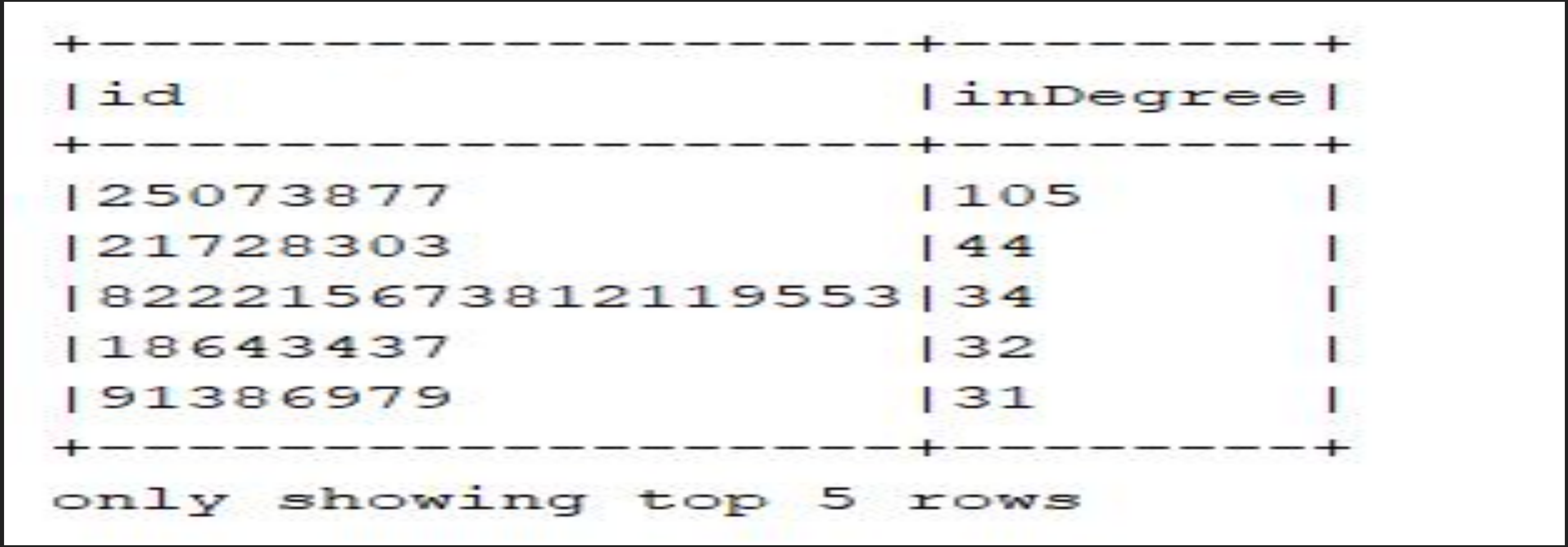
```
g.inDegrees.sort("inDegree").show(5, False)
```



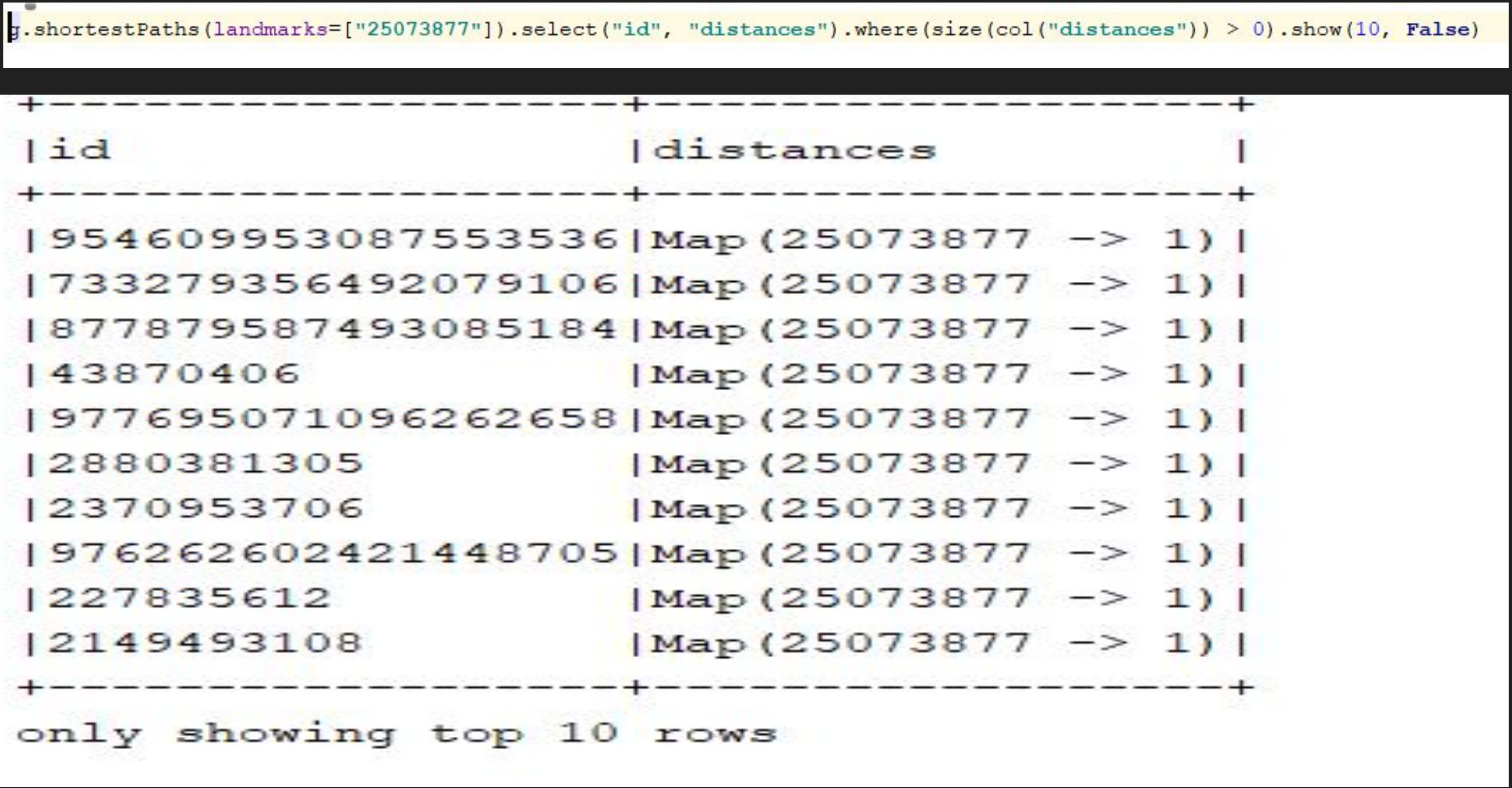
```
+-------------------+--------+
|id                 |inDegree|
+-------------------+--------+
|15469000           |1       |
|577432615          |1       |
|864068108          |1       |
|1038512907120779264|1       |
|53190110           |1       |
+-------------------+--------+
only showing top 5 rows
```



```
g.inDegrees.sort("inDegree", ascending=False).show(5, False)
```



```
+------------------+--------+
|id                |inDegree|
+------------------+--------+
|25073877          |105     |
|21728303          |44      |
|822215673812119553|34      |
|18643437          |32      |
|91386979          |31      |
+------------------+--------+
only showing top 5 rows
```



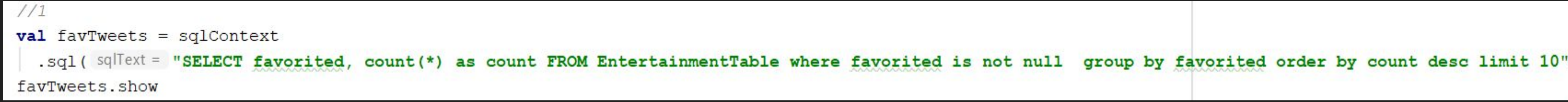
```
g.shortestPaths(landmarks=["25073877"]).select("id", "distances").where(size(col("distances")) > 0).show(10, False)
```

```
+-------------------+-------------------+
|id                 |distances          |
+-------------------+-------------------+
|954609953087553536|Map(25073877 -> 1)|
|733279356492079106|Map(25073877 -> 1)|
|877879587493085184|Map(25073877 -> 1)|
|43870406           |Map(25073877 -> 1)|
|977695071096262658|Map(25073877 -> 1)|
|2880381305         |Map(25073877 -> 1)|
|2370953706         |Map(25073877 -> 1)|
|976262602421448705|Map(25073877 -> 1)|
|227835612          |Map(25073877 -> 1)|
|2149493108         |Map(25073877 -> 1)|
+-------------------+-------------------+
only showing top 10 rows
```

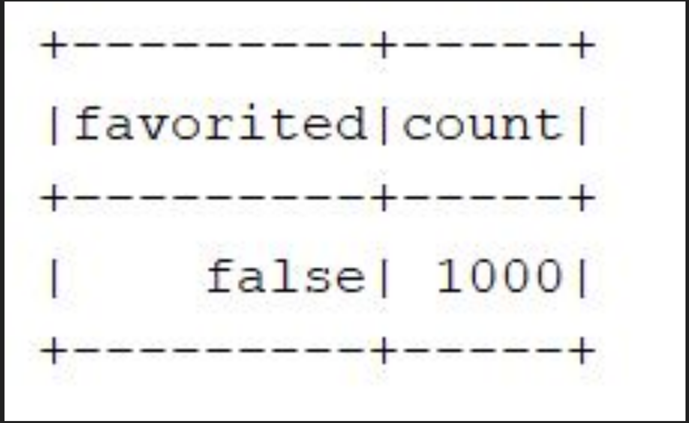
Stephanie

## Increment 1

Query 1: Count how many tweets were favorited and those that were not



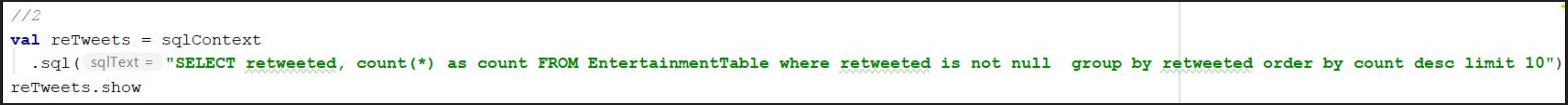
```
//1
val favTweets = sqlContext
  .sql( sqlText = "SELECT favorited, count(*) as count FROM EntertainmentTable where favorited is not null  group by favorited order by count desc limit 10"
favTweets.show
```



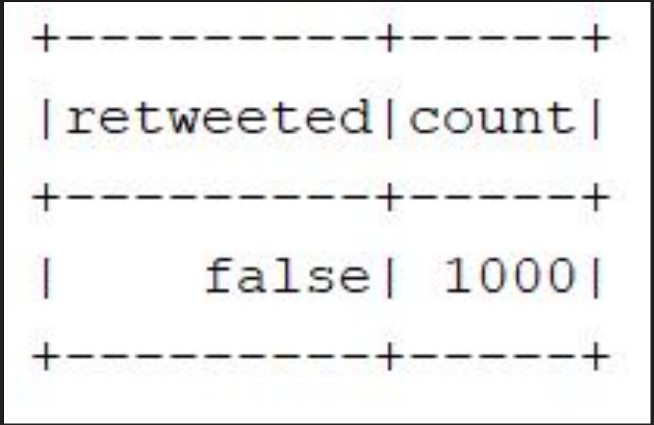
```
+---------+-----+
|favorited|count|
+---------+-----+
|    false| 1000|
+---------+-----+
```

# Increment 1

Query 2:  Count how many tweets were retweeted and those that were not



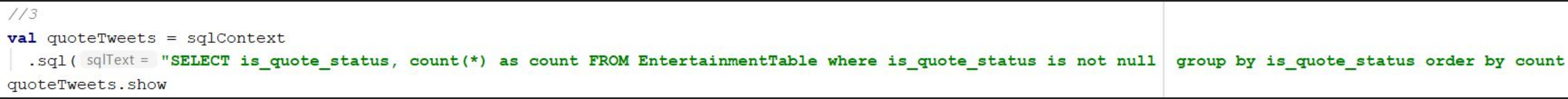
```
//2
val reTweets = sqlContext
  .sql( sqlText = "SELECT retweeted, count(*) as count FROM EntertainmentTable where retweeted is not null  group by retweeted order by count desc limit 10")
reTweets.show
```



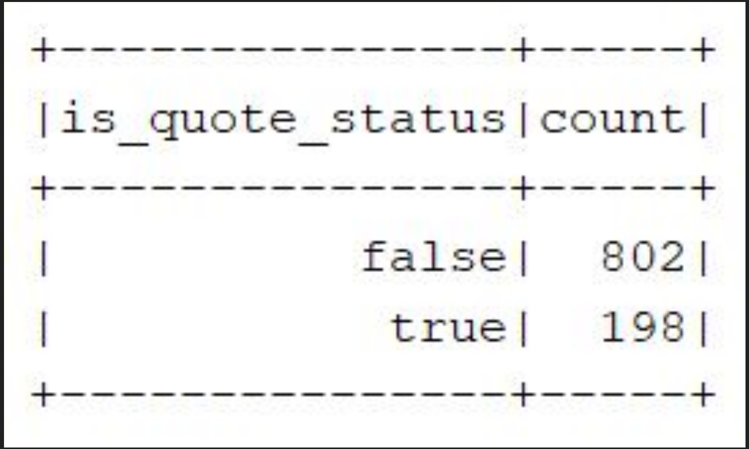
```
+---------+-----+
|retweeted|count|
+---------+-----+
|    false| 1000|
+---------+-----+
```

## Increment 1

Query 3:  Count how many tweets were quotes and those that were not



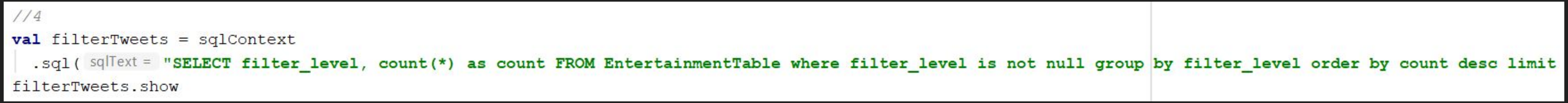
```
//3
val quoteTweets = sqlContext
  .sql( sqlText = "SELECT is_quote_status, count(*) as count FROM EntertainmentTable where is_quote_status is not null  group by is_quote_status order by count
quoteTweets.show
```



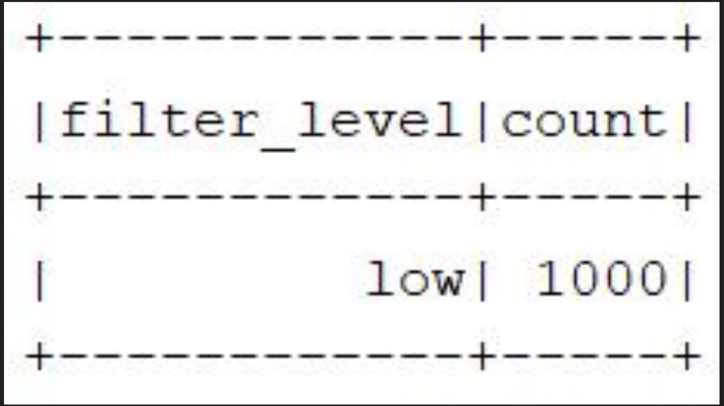
```
+---------------+-----+
|is_quote_status|count|
+---------------+-----+
|          false|  802|
|           true|  198|
+---------------+-----+
```

# Increment 1

Query 4:  See what filter levels people use and how many



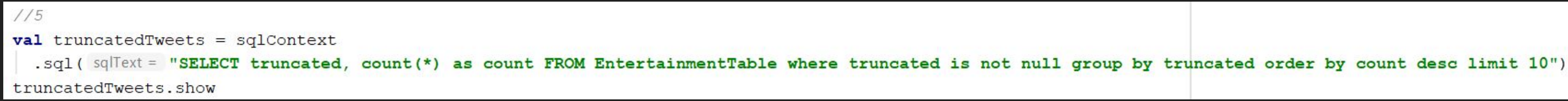
```
//4
val filterTweets = sqlContext
  .sql( sqlText = "SELECT filter_level, count(*) as count FROM EntertainmentTable where filter_level is not null group by filter_level order by count desc limit
filterTweets.show
```



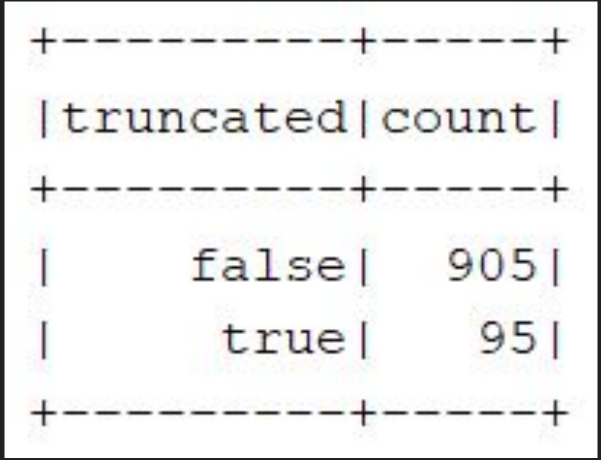
```
+------------+-----+
|filter_level|count|
+------------+-----+
|         low| 1000|
+------------+-----+
```

## Increment 1

Query 5:  Count how many tweets were truncated and those that were not



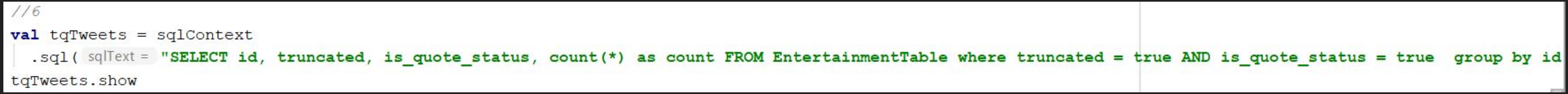
```
//5
val truncatedTweets = sqlContext
  .sql( sqlText = "SELECT truncated, count(*) as count FROM EntertainmentTable where truncated is not null group by truncated order by count desc limit 10")
truncatedTweets.show
```



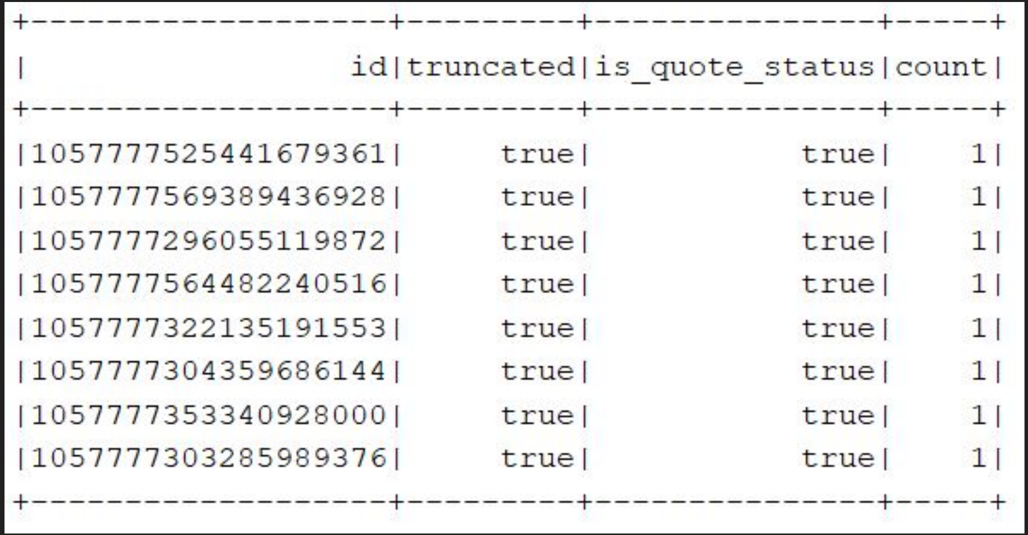
```
+---------+-----+
|truncated|count|
+---------+-----+
|    false|  905|
|     true|   95|
+---------+-----+
```

# Increment 1

Query 6:  Count how many tweets were quotes and truncated



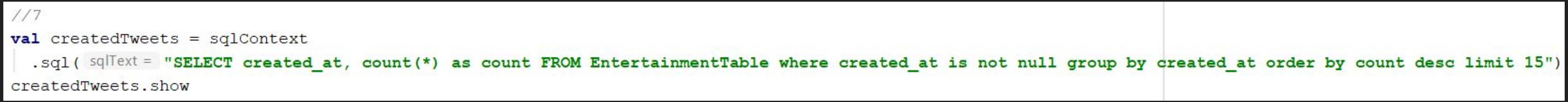
```
//6
val tqTweets = sqlContext
  .sql( sqlText = "SELECT id, truncated, is_quote_status, count(*) as count FROM EntertainmentTable where truncated = true AND is_quote_status = true  group by id
tqTweets.show
```



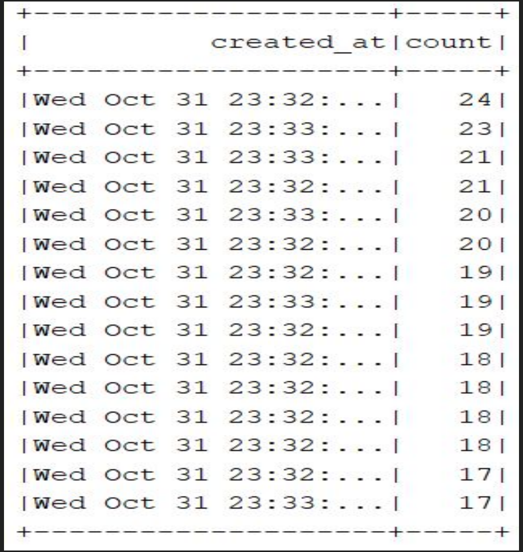
```
+-------------------+---------+---------------+-----+
|                 id|truncated|is_quote_status|count|
+-------------------+---------+---------------+-----+
|105777525441679361|     true|           true|    1|
|105777569389436928|     true|           true|    1|
|105777296055119872|     true|           true|    1|
|105777564482240516|     true|           true|    1|
|105777322135191553|     true|           true|    1|
|105777304359686144|     true|           true|    1|
|105777353340928000|     true|           true|    1|
|105777303285989376|     true|           true|    1|
+-------------------+---------+---------------+-----+
```

# Increment 1

Query 7:  Count how many tweets were created at the same time



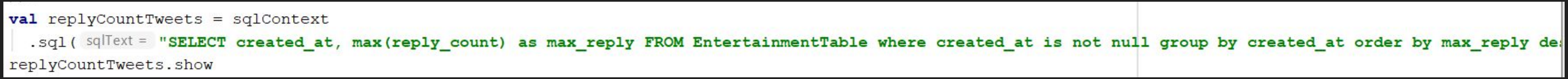
```
//7
val createdTweets = sqlContext
  .sql( sqlText = "SELECT created_at, count(*) as count FROM EntertainmentTable where created_at is not null group by created_at order by count desc limit 15")
createdTweets.show
```



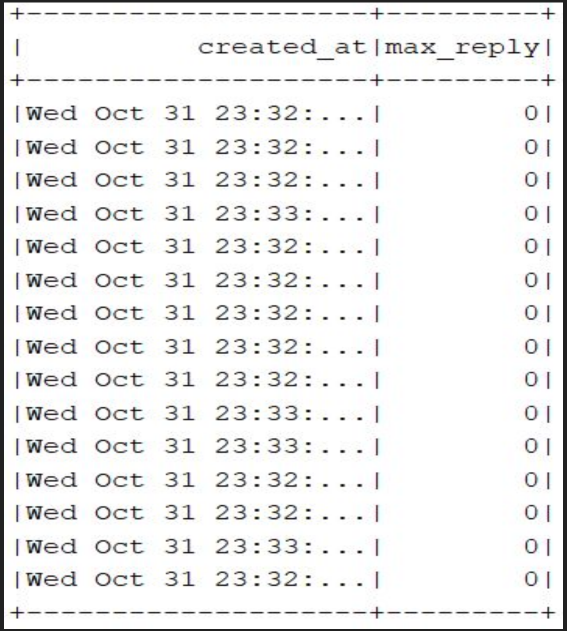
```
+--------------------+-----+
|          created_at|count|
+--------------------+-----+
|Wed Oct 31 23:32:...|   24|
|Wed Oct 31 23:33:...|   23|
|Wed Oct 31 23:33:...|   21|
|Wed Oct 31 23:32:...|   21|
|Wed Oct 31 23:33:...|   20|
|Wed Oct 31 23:32:...|   20|
|Wed Oct 31 23:32:...|   19|
|Wed Oct 31 23:33:...|   19|
|Wed Oct 31 23:32:...|   19|
|Wed Oct 31 23:32:...|   18|
|Wed Oct 31 23:32:...|   18|
|Wed Oct 31 23:32:...|   18|
|Wed Oct 31 23:32:...|   18|
|Wed Oct 31 23:32:...|   17|
|Wed Oct 31 23:33:...|   17|
+--------------------+-----+
```

# Increment 1

Query 8:  Find the max replies a tweet had at a certain time



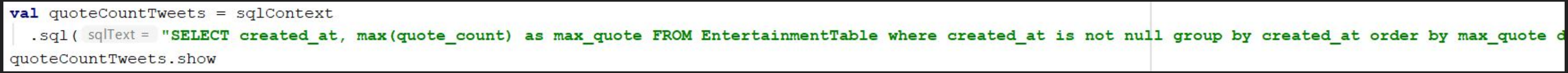
```
val replyCountTweets = sqlContext
  .sql( sqlText = "SELECT created_at, max(reply_count) as max_reply FROM EntertainmentTable where created_at is not null group by created_at order by max_reply de
replyCountTweets.show
```



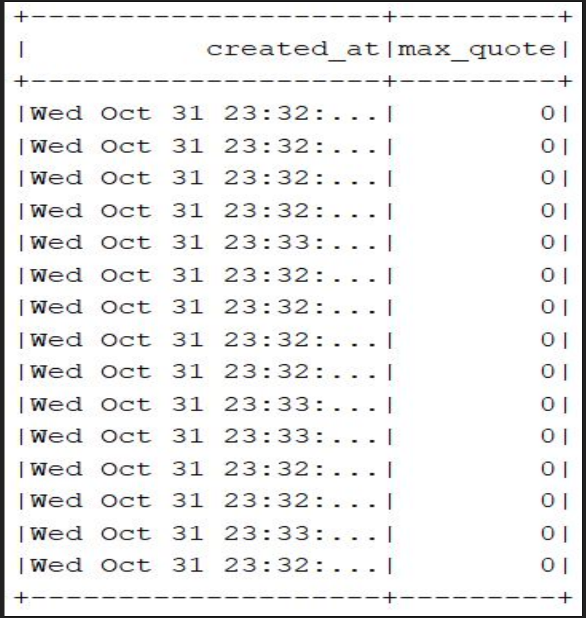
```
+--------------------+---------+
|          created_at|max_reply|
+--------------------+---------+
|Wed Oct 31 23:32:...|        0|
|Wed Oct 31 23:32:...|        0|
|Wed Oct 31 23:32:...|        0|
|Wed Oct 31 23:33:...|        0|
|Wed Oct 31 23:32:...|        0|
|Wed Oct 31 23:32:...|        0|
|Wed Oct 31 23:32:...|        0|
|Wed Oct 31 23:32:...|        0|
|Wed Oct 31 23:32:...|        0|
|Wed Oct 31 23:33:...|        0|
|Wed Oct 31 23:33:...|        0|
|Wed Oct 31 23:32:...|        0|
|Wed Oct 31 23:32:...|        0|
|Wed Oct 31 23:33:...|        0|
|Wed Oct 31 23:32:...|        0|
+--------------------+---------+
```

# Increment 1

Query 9:  Find the max quotes a tweet had at a certain time



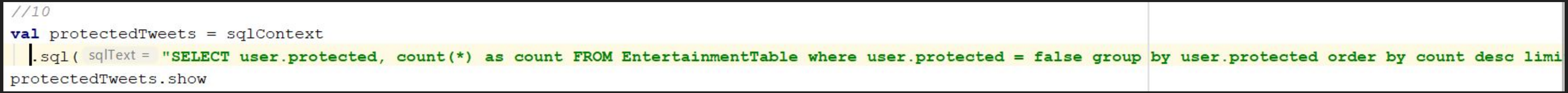
```
val quoteCountTweets = sqlContext
  .sql( sqlText = "SELECT created_at, max(quote_count) as max_quote FROM EntertainmentTable where created_at is not null group by created_at order by max_quote d
quoteCountTweets.show
```



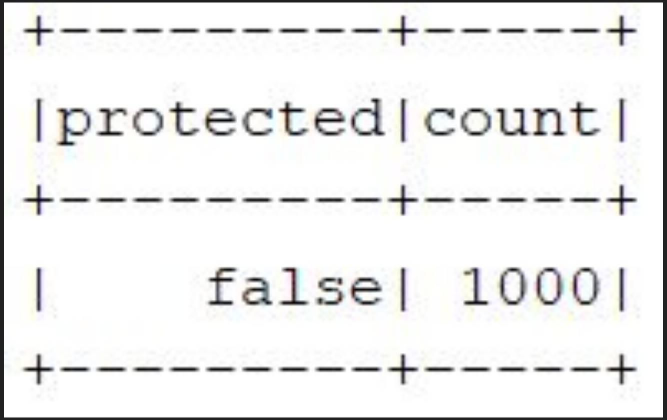
```
+--------------------+---------+
|          created_at|max_quote|
+--------------------+---------+
|Wed Oct 31 23:32:...|        0|
|Wed Oct 31 23:32:...|        0|
|Wed Oct 31 23:32:...|        0|
|Wed Oct 31 23:32:...|        0|
|Wed Oct 31 23:33:...|        0|
|Wed Oct 31 23:32:...|        0|
|Wed Oct 31 23:32:...|        0|
|Wed Oct 31 23:32:...|        0|
|Wed Oct 31 23:32:...|        0|
|Wed Oct 31 23:33:...|        0|
|Wed Oct 31 23:33:...|        0|
|Wed Oct 31 23:32:...|        0|
|Wed Oct 31 23:32:...|        0|
|Wed Oct 31 23:33:...|        0|
|Wed Oct 31 23:32:...|        0|
+--------------------+---------+
```

# Increment 1

Query 10:  Count how many users are protected and those that are not

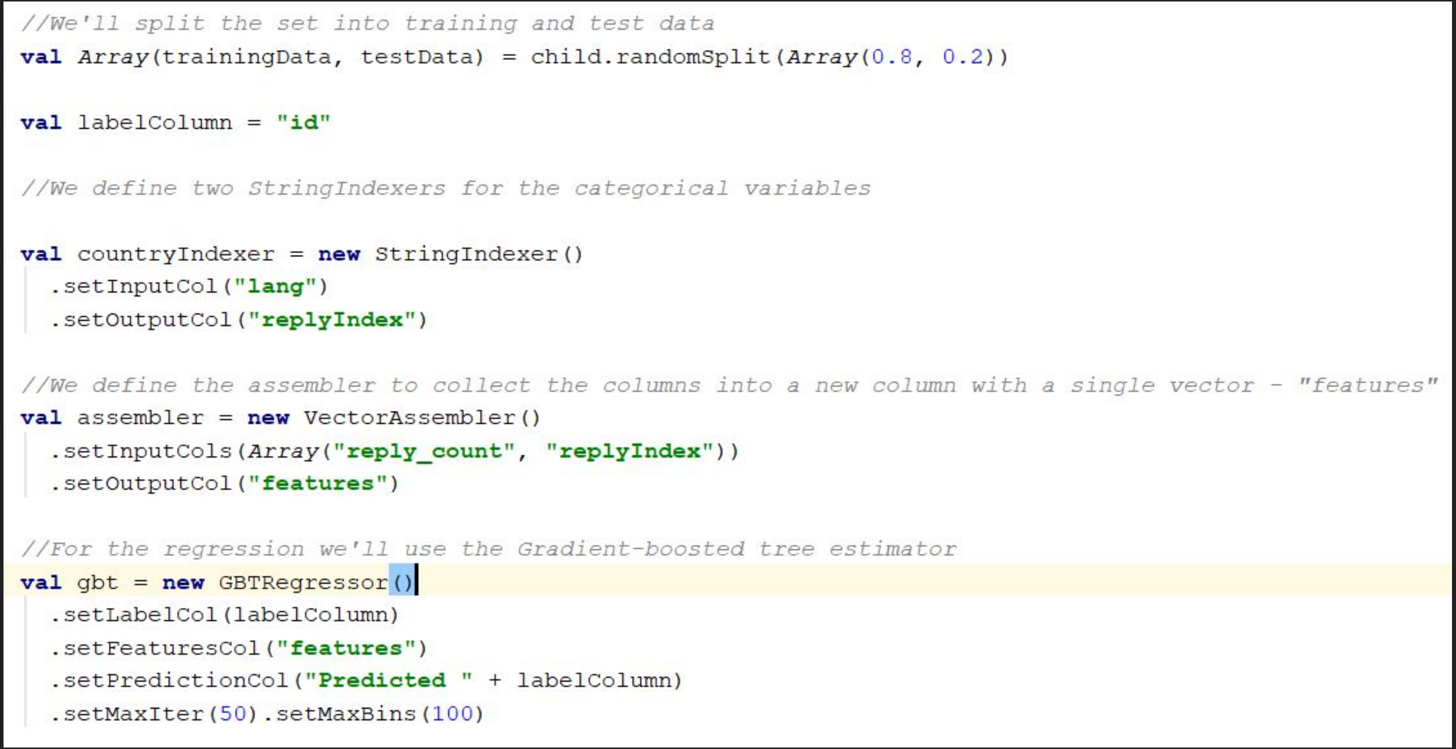


```
//10
val protectedTweets = sqlContext
  .sql( sqlText = "SELECT user.protected, count(*) as count FROM EntertainmentTable where user.protected = false group by user.protected order by count desc limi
protectedTweets.show
```



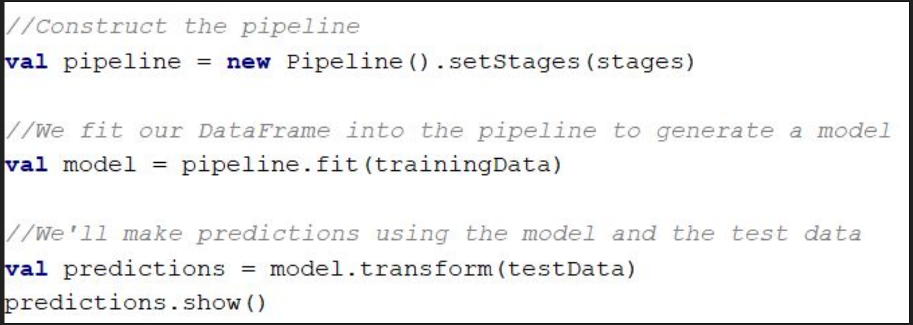
```
+---------+-----+
|protected|count|
+---------+-----+
|    false| 1000|
+---------+-----+
```

# Increment 2 - Machine Learning

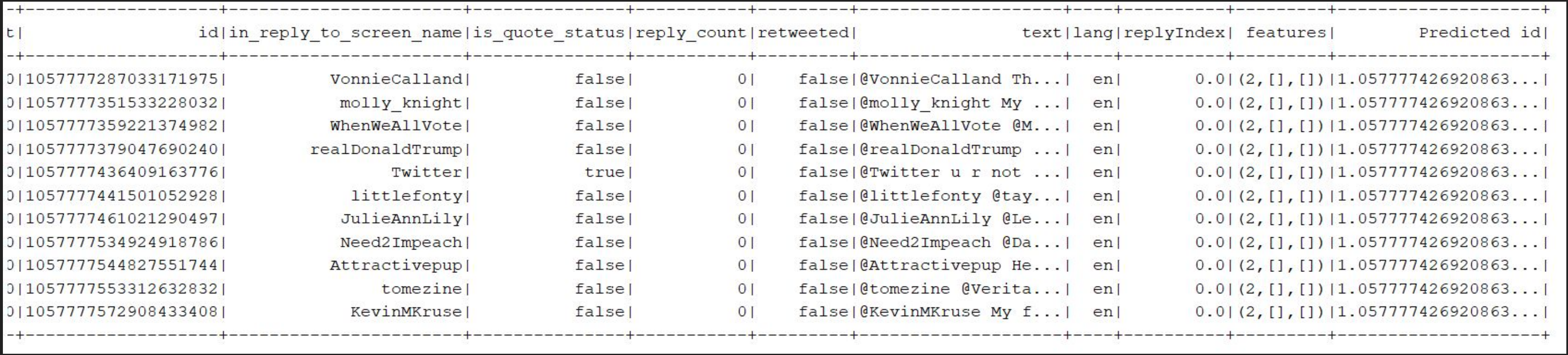


```
//We'll split the set into training and test data
val Array(trainingData, testData) = child.randomSplit(Array(0.8, 0.2))

val labelColumn = "id"

//We define two StringIndexers for the categorical variables

val countryIndexer = new StringIndexer()
  .setInputCol("lang")
  .setOutputCol("replyIndex")

//We define the assembler to collect the columns into a new column with a single vector - "features"
val assembler = new VectorAssembler()
  .setInputCols(Array("reply_count", "replyIndex"))
  .setOutputCol("features")

//For the regression we'll use the Gradient-boosted tree estimator
val gbt = new GBTRegressor()
  .setLabelCol(labelColumn)
  .setFeaturesCol("features")
  .setPredictionCol("Predicted " + labelColumn)
  .setMaxIter(50).setMaxBins(100)
```

## Increment 2 - Machine Learning

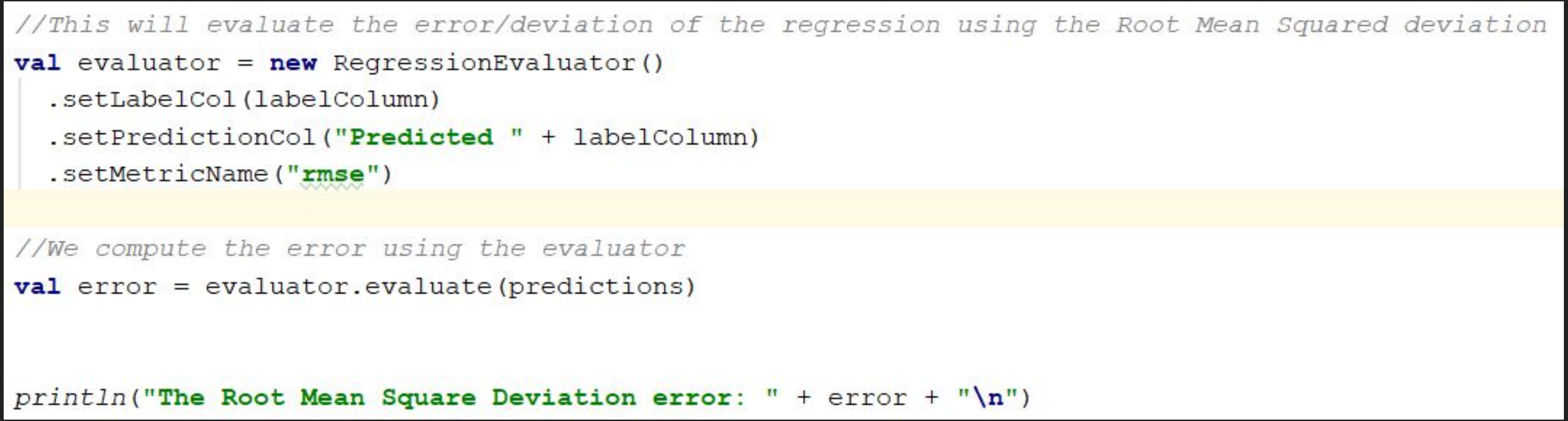


```scala
//Construct the pipeline
val pipeline = new Pipeline().setStages(stages)

//We fit our DataFrame into the pipeline to generate a model
val model = pipeline.fit(trainingData)

//We'll make predictions using the model and the test data
val predictions = model.transform(testData)
predictions.show()
```

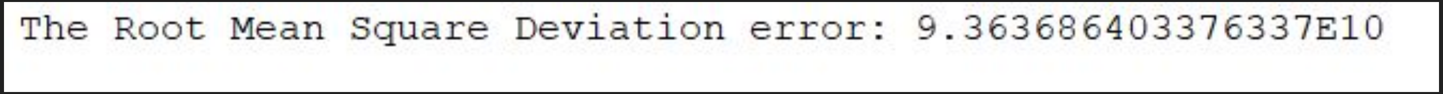


```
-+-------------------+----------------------+----------------+-----------+---------+--------------------+----+----------+---------+-------------------+
t|                 id|in_reply_to_screen_name|is_quote_status|reply_count|retweeted|                text|lang|replyIndex| features|       Predicted id|
-+-------------------+----------------------+----------------+-----------+---------+--------------------+----+----------+---------+-------------------+
0|1057777287033171975|          VonnieCalland|           false|          0|    false|@VonnieCalland Th...|  en|       0.0|(2,[],[])|1.05777742692086...|
0|1057777351533228032|           molly_knight|           false|          0|    false|@molly_knight My ...|  en|       0.0|(2,[],[])|1.05777742692086...|
0|1057777359221374982|          WhenWeAllVote|           false|          0|    false|@WhenWeAllVote @M...|  en|       0.0|(2,[],[])|1.05777742692086...|
0|1057777379047690240|        realDonaldTrump|           false|          0|    false|@realDonaldTrump ...|  en|       0.0|(2,[],[])|1.05777742692086...|
0|1057777436409163776|                Twitter|            true|          0|    false|@Twitter u r not ...|  en|       0.0|(2,[],[])|1.05777742692086...|
0|1057777441501052928|            littlefonty|           false|          0|    false|@littlefonty @tay...|  en|       0.0|(2,[],[])|1.05777742692086...|
0|1057777461021290497|            JulieAnnLily|           false|          0|    false|@JulieAnnLily @Le...|  en|       0.0|(2,[],[])|1.05777742692086...|
0|1057777534924918786|            Need2Impeach|           false|          0|    false|@Need2Impeach @Da...|  en|       0.0|(2,[],[])|1.05777742692086...|
0|1057777544827551744|          Attractivepup|           false|          0|    false|@Attractivepup He...|  en|       0.0|(2,[],[])|1.05777742692086...|
0|1057777553312632832|               tomezine|           false|          0|    false|@tomezine @Verita...|  en|       0.0|(2,[],[])|1.05777742692086...|
0|1057777572908433408|             KevinMKruse|           false|          0|    false|@KevinMKruse My f...|  en|       0.0|(2,[],[])|1.05777742692086...|
-+-------------------+----------------------+----------------+-----------+---------+--------------------+----+----------+---------+-------------------+
```

## Increment 2 - Machine Learning

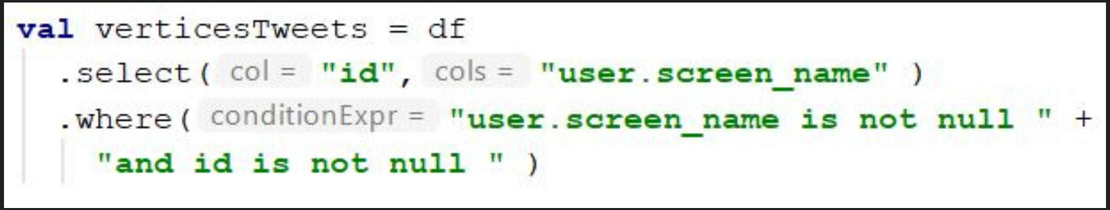


```
//This will evaluate the error/deviation of the regression using the Root Mean Squared deviation
val evaluator = new RegressionEvaluator()
  .setLabelCol(labelColumn)
  .setPredictionCol("Predicted " + labelColumn)
  .setMetricName("rmse")

//We compute the error using the evaluator
val error = evaluator.evaluate(predictions)


println("The Root Mean Square Deviation error: " + error + "\n")
```

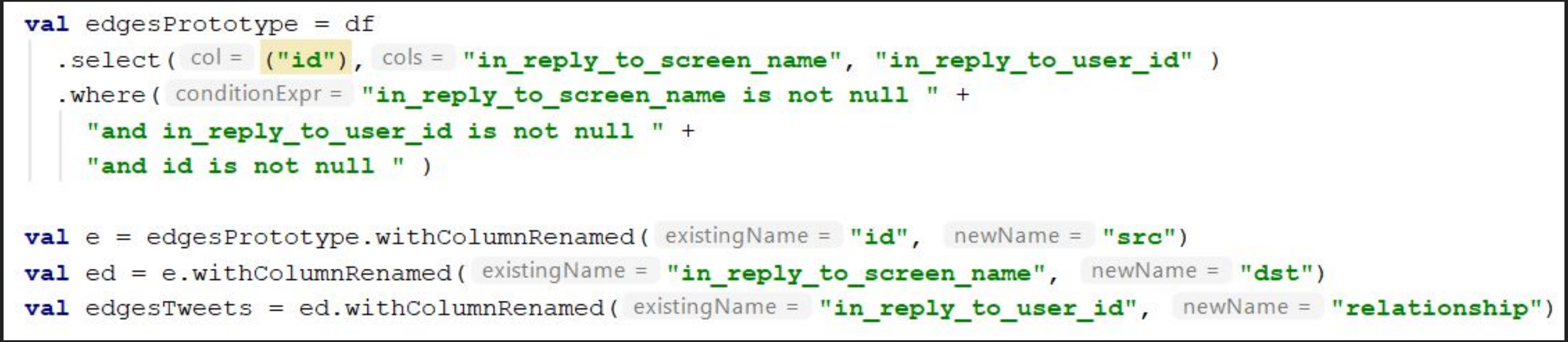


```
The Root Mean Square Deviation error: 9.36368640337633 7E10
```

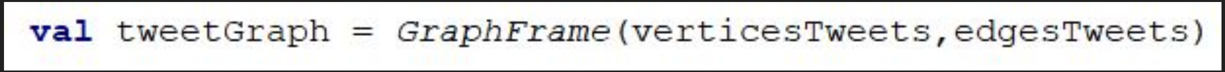
## Increment 2 - GraphFrame



```
val verticesTweets = df
  .select( col = "id", cols = "user.screen_name" )
  .where( conditionExpr = "user.screen_name is not null " +
    "and id is not null " )
```

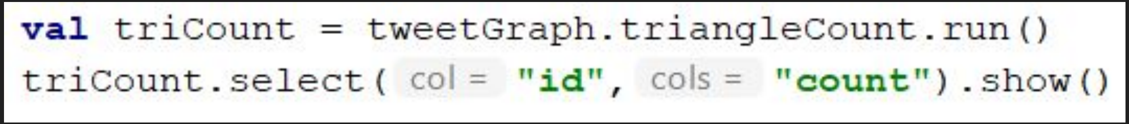


```
val edgesProtoype = df
  .select( col = ("id"), cols = "in_reply_to_screen_name", "in_reply_to_user_id" )
  .where( conditionExpr = "in_reply_to_screen_name is not null " +
    "and in_reply_to_user_id is not null " +
    "and id is not null " )

val e = edgesProtoype.withColumnRenamed( existingName = "id",  newName = "src")
val ed = e.withColumnRenamed( existingName = "in_reply_to_screen_name",  newName = "dst")
val edgesTweets = ed.withColumnRenamed( existingName = "in_reply_to_user_id",  newName = "relationship")
```

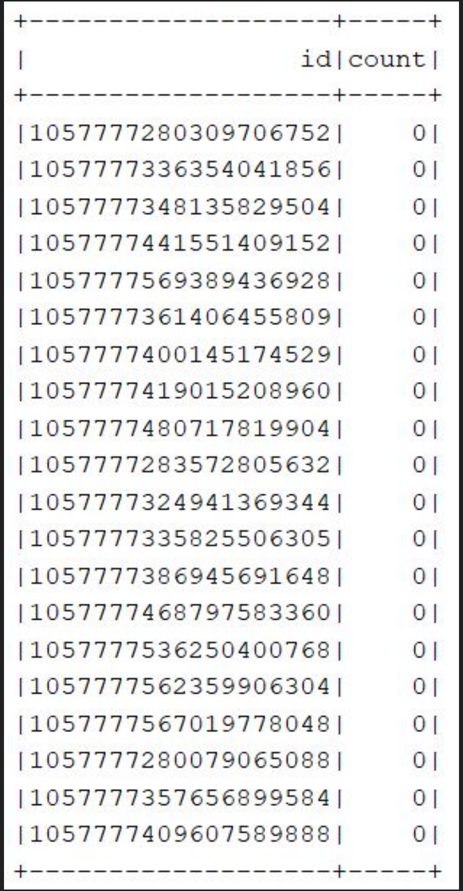


```
val tweetGraph = GraphFrame(verticesTweets,edgesTweets)
```

## Increment 2 - GraphFrame

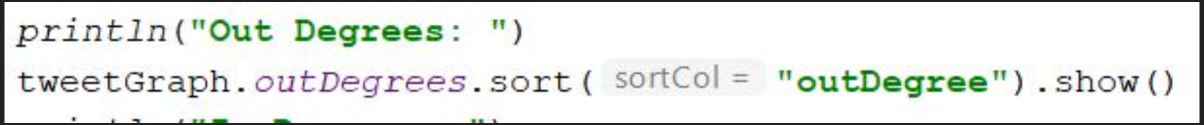


```scala
val triCount = tweetGraph.triangleCount.run()
triCount.select( col = "id", cols = "count").show()
```

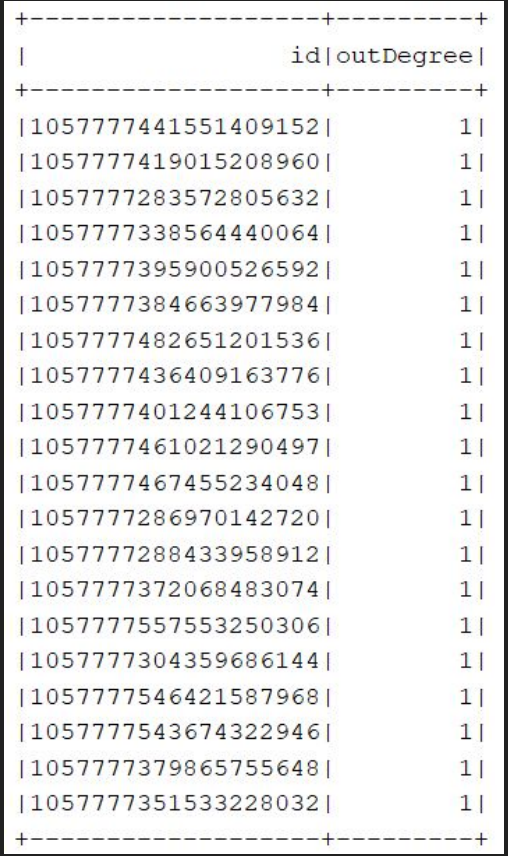


```
+-------------------+-----+
|                 id|count|
+-------------------+-----+
|1057777280309706752|    0|
|1057777336354041856|    0|
|1057777348135829504|    0|
|1057777441551409152|    0|
|1057777569389436928|    0|
|1057777361406455809|    0|
|1057777400145174529|    0|
|1057777419015208960|    0|
|1057777480717819904|    0|
|1057777283572805632|    0|
|1057777324941369344|    0|
|1057777335825506305|    0|
|1057777386945691648|    0|
|1057777468797583360|    0|
|1057777536250400768|    0|
|1057777562359906304|    0|
|1057777567019778048|    0|
|1057777280079065088|    0|
|1057777357656899584|    0|
|1057777409607589888|    0|
+-------------------+-----+
```

# Increment 2 - GraphFrame

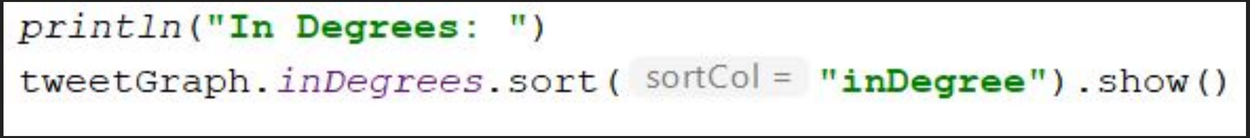


```
println("Out Degrees: ")
tweetGraph.outDegrees.sort( sortCol = "outDegree").show()
```

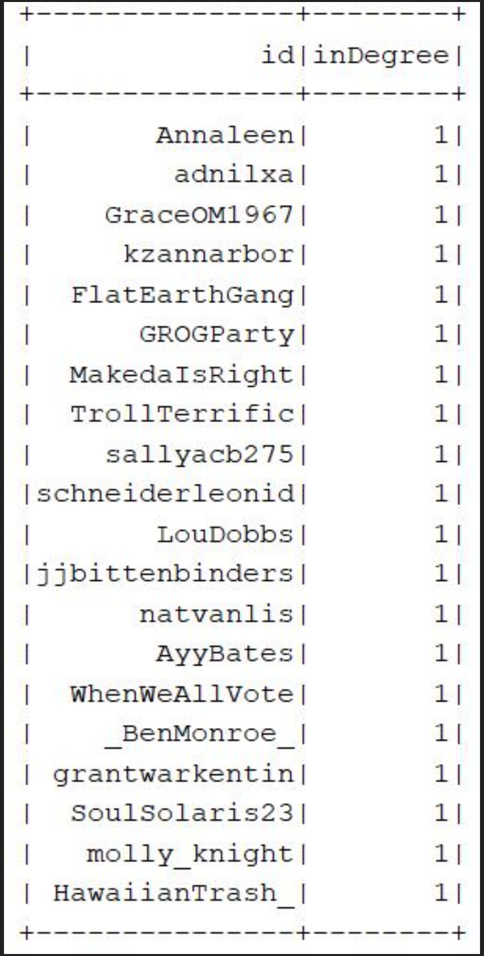


```
+-------------------+---------+
|                 id|outDegree|
+-------------------+---------+
|1057777441551409152|        1|
|1057777419015208960|        1|
|1057777283572805632|        1|
|1057777338564440064|        1|
|1057777395900526592|        1|
|1057777384663977984|        1|
|1057777482651201536|        1|
|1057777436409163776|        1|
|1057777401244106753|        1|
|1057777461021290497|        1|
|1057777467455234048|        1|
|1057777286970142720|        1|
|1057777288433958912|        1|
|1057777372068483074|        1|
|1057777557553250306|        1|
|1057777304359686144|        1|
|1057777546421587968|        1|
|1057777543674322946|        1|
|1057777379865755648|        1|
|1057777351533228032|        1|
+-------------------+---------+
```

# Increment 2 - GraphFrame

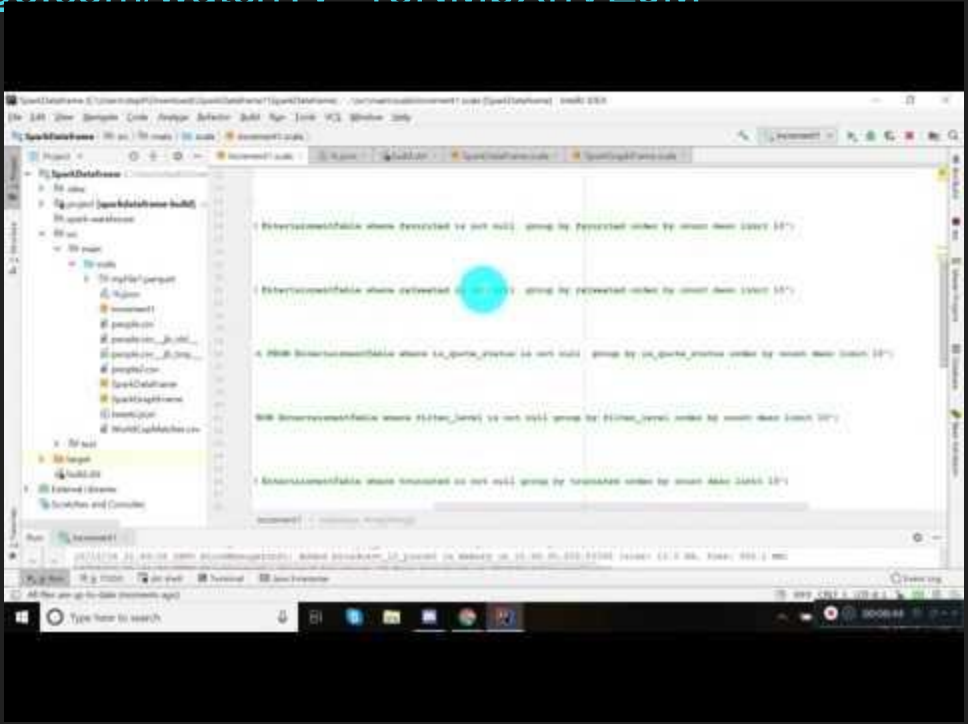


```
println("In Degrees: ")
tweetGraph.inDegrees.sort( sortCol = "inDegree").show()
```



```
+---------------+--------+
|             id|inDegree|
+---------------+--------+
|       Annaleen|       1|
|        adnilxa|       1|
|     GraceOM1967|       1|
|      kzannarbor|       1|
|   FlatEarthGang|       1|
|        GROGParty|       1|
|   MakedaIsRight|       1|
|   TrollTerrific|       1|
|     sallyacb275|       1|
|schneiderleonid|       1|
|       LouDobbs|       1|
|jjbittenbinders|       1|
|       natvanlis|       1|
|       AyyBates|       1|
|   WhenWeAllVote|       1|
|     _BenMonroe_|       1|
|  grantwarkentin|       1|
|   SoulSolaris23|       1|
|    molly_knight|       1|
|  HawaiianTrash_|       1|
+---------------+--------+
```

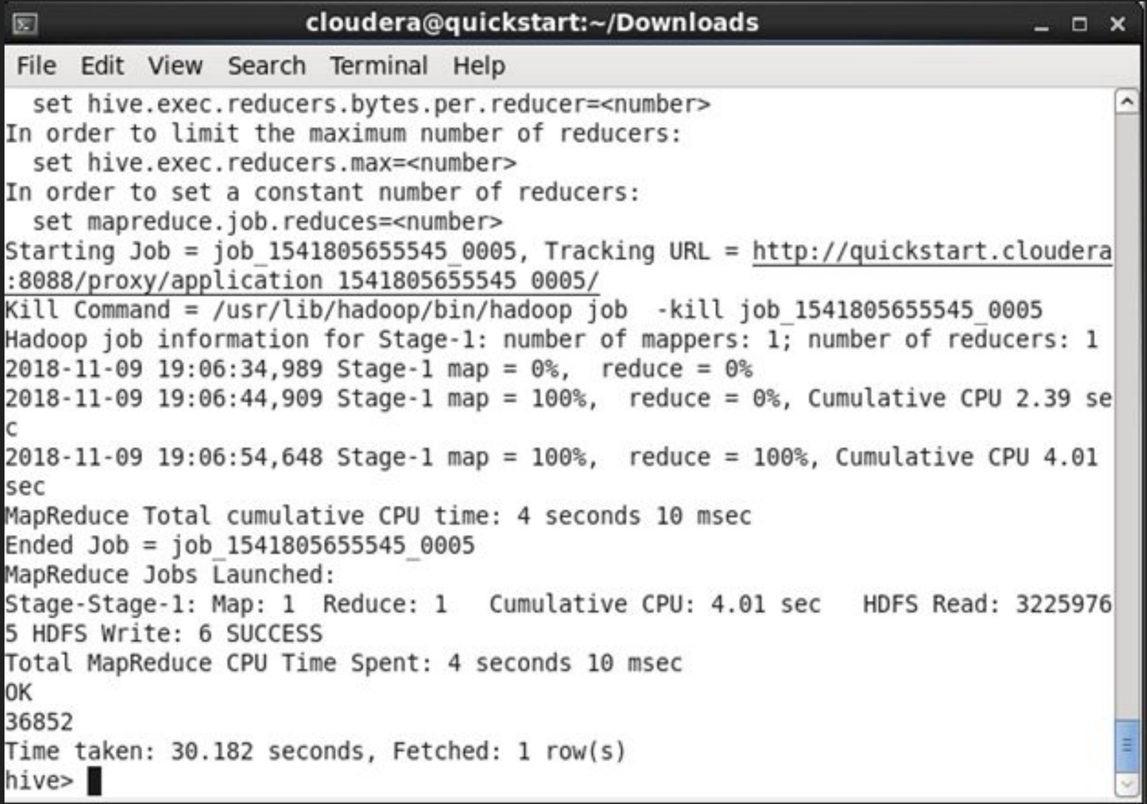
[https://www.youtube.com/watch?v=18NMbXnVEJM](https://www.youtube.com/watch?v=18NMbXnVEJM)



```
( EntertainmentTable where favorited is not null  group by favorited order by count desc limit 10")

( EntertainmentTable where retweeted [illegible]  group by retweeted order by count desc limit 10")

+ FROM EntertainmentTable where is_quote_status is not null  group by is_quote_status order by count desc limit 10")

ROM EntertainmentTable where filter_level is not null group by filter_level order by count desc limit 10")

( EntertainmentTable where truncated is not null group by truncated order by count desc limit 10")
```

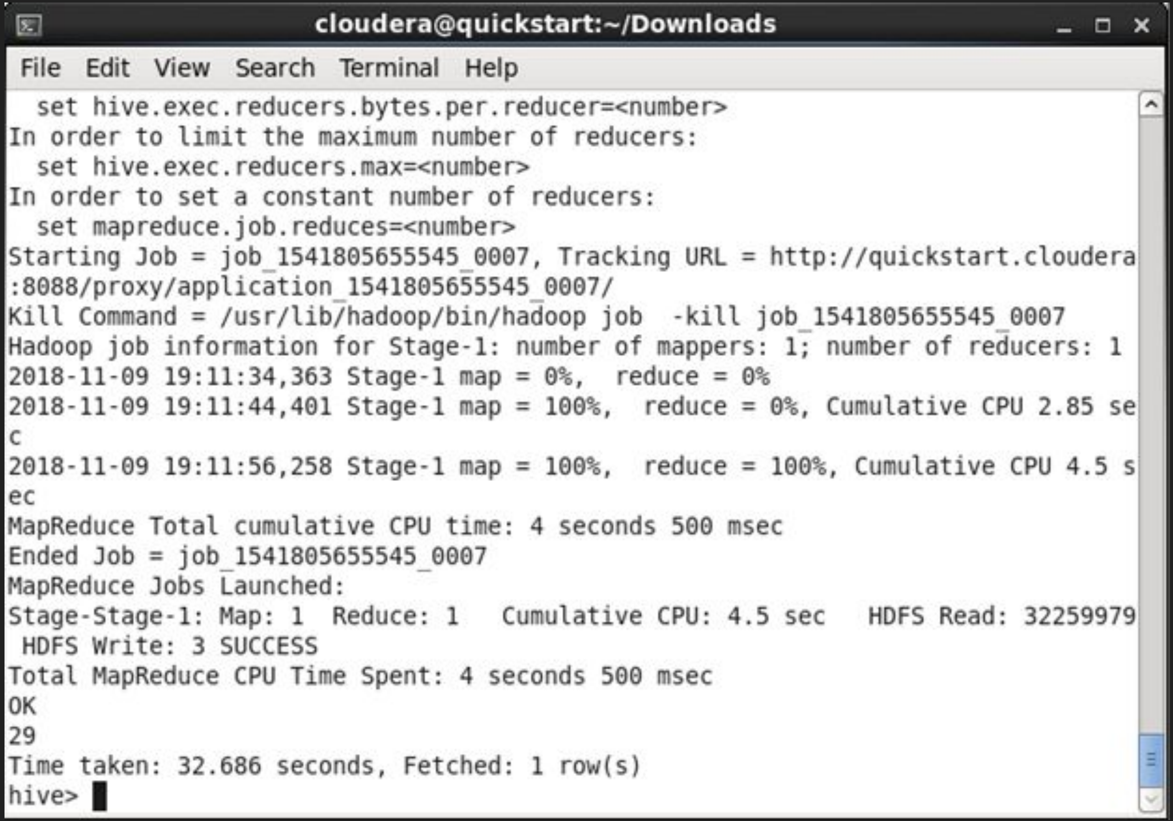
Shaun

## Increment 1 - Query 1 - Favorites All Tweet


cloudera@quickstart:~/Downloads
File Edit View Search Terminal Help

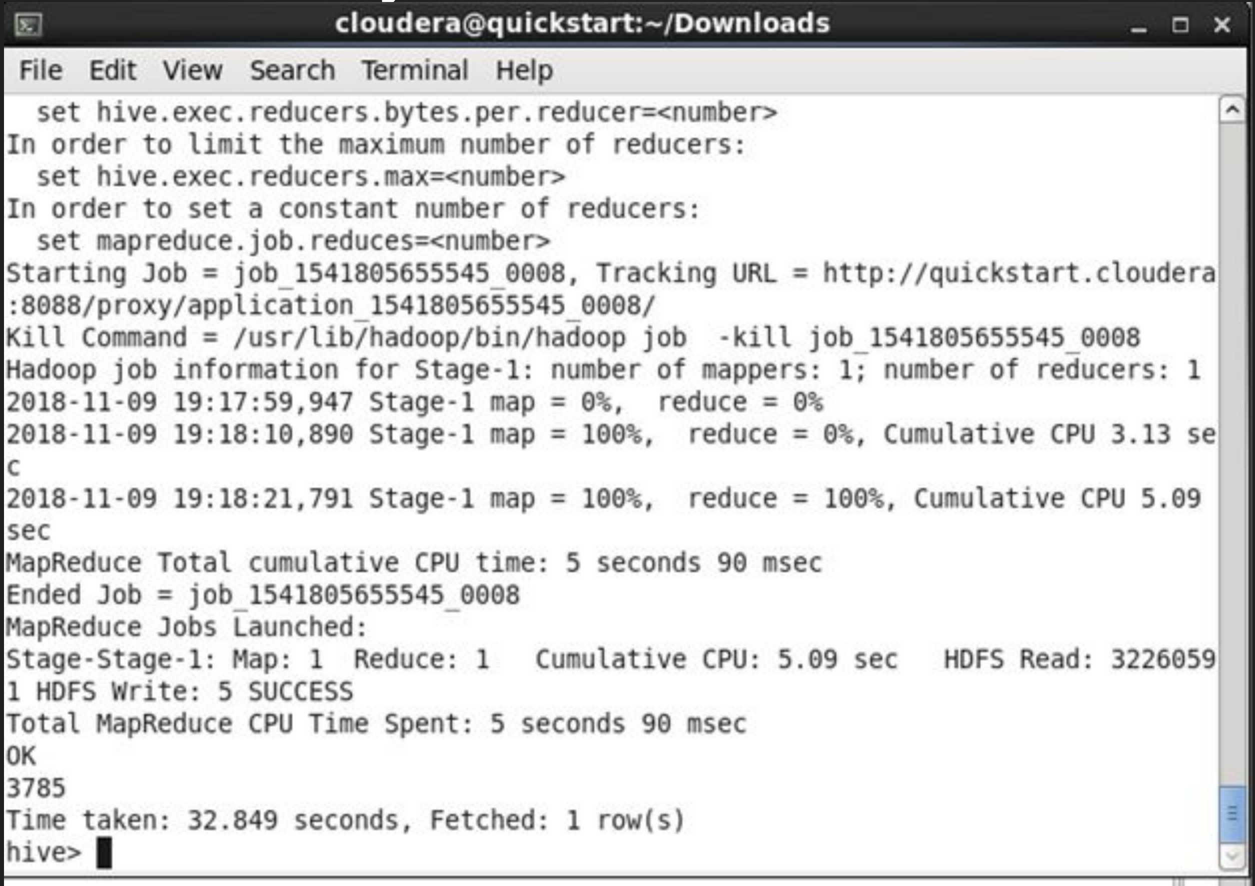
```
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1541805655545_0005, Tracking URL = http://quickstart.cloudera
:8088/proxy/application_1541805655545_0005/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1541805655545_0005
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-11-09 19:06:34,989 Stage-1 map = 0%,  reduce = 0%
2018-11-09 19:06:44,909 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.39 se
c
2018-11-09 19:06:54,648 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.01
sec
MapReduce Total cumulative CPU time: 4 seconds 10 msec
Ended Job = job_1541805655545_0005
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.01 sec   HDFS Read: 3225976
5 HDFS Write: 6 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 10 msec
OK
36852
Time taken: 30.182 seconds, Fetched: 1 row(s)
hive> 
```

# Increment 1 - Query 2 - Tweets Containing Trump


cloudera@quickstart:~/Downloads
File Edit View Search Terminal Help

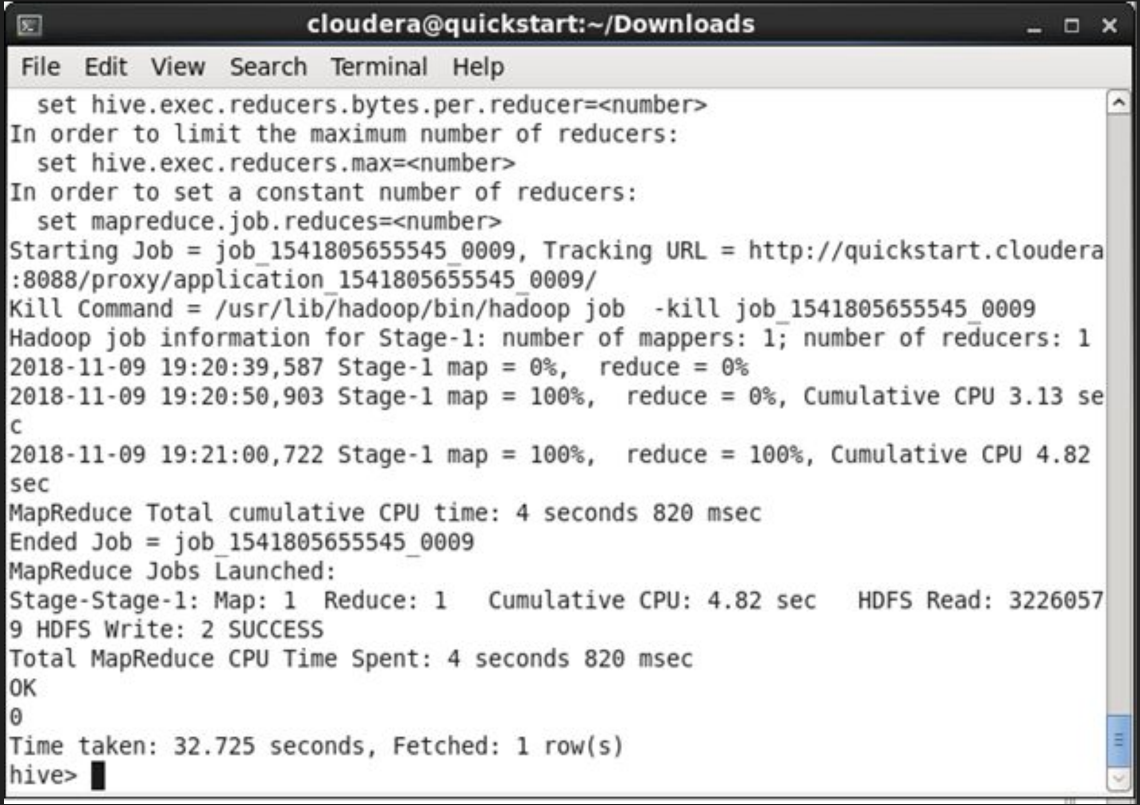
```
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1541805655545_0007, Tracking URL = http://quickstart.cloudera
:8088/proxy/application_1541805655545_0007/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1541805655545_0007
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-11-09 19:11:34,363 Stage-1 map = 0%,  reduce = 0%
2018-11-09 19:11:44,401 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.85 se
c
2018-11-09 19:11:56,258 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.5 s
ec
MapReduce Total cumulative CPU time: 4 seconds 500 msec
Ended Job = job_1541805655545_0007
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.5 sec   HDFS Read: 32259979
 HDFS Write: 3 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 500 msec
OK
29
Time taken: 32.686 seconds, Fetched: 1 row(s)
hive> 
```

## Increment 1 - Query 3 - Tweets from Northern Hemosphere


cloudera@quickstart:~/Downloads
File Edit View Search Terminal Help

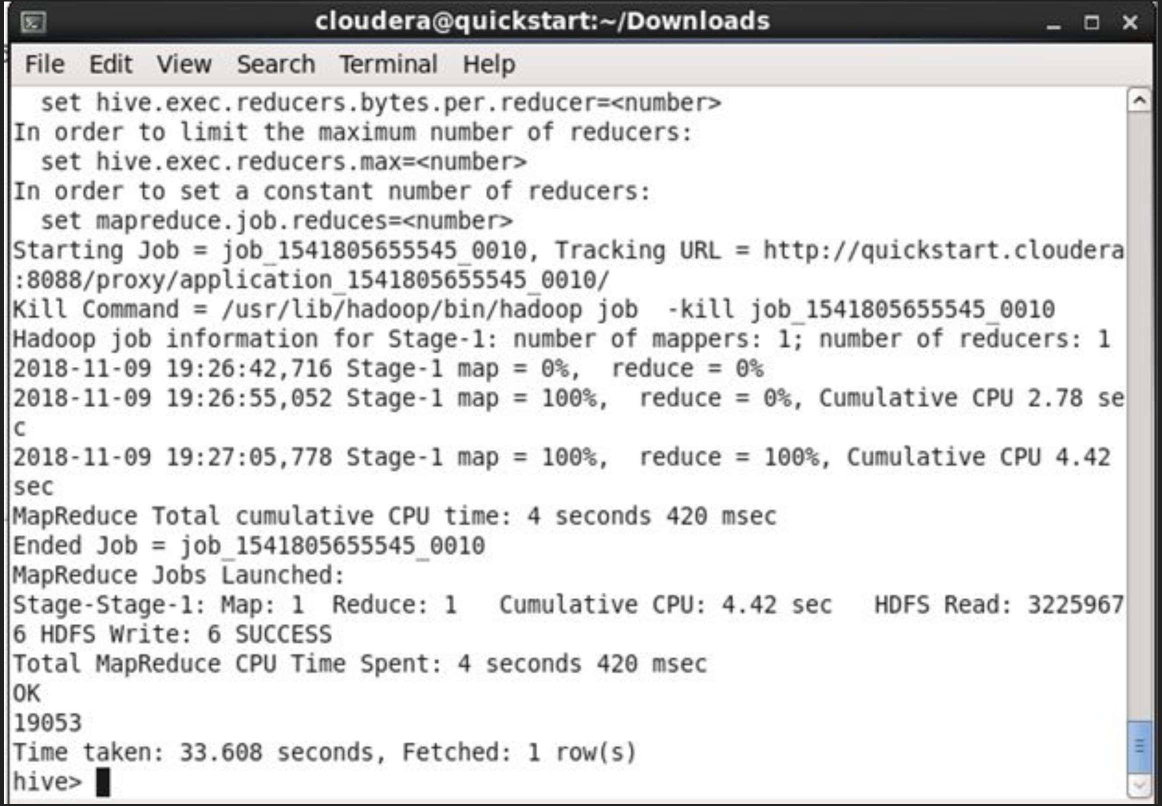
```
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1541805655545_0008, Tracking URL = http://quickstart.cloudera
:8088/proxy/application_1541805655545_0008/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1541805655545_0008
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-11-09 19:17:59,947 Stage-1 map = 0%,  reduce = 0%
2018-11-09 19:18:10,890 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.13 se
c
2018-11-09 19:18:21,791 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 5.09
sec
MapReduce Total cumulative CPU time: 5 seconds 90 msec
Ended Job = job_1541805655545_0008
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 5.09 sec   HDFS Read: 3226059
1 HDFS Write: 5 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 90 msec
OK
3785
Time taken: 32.849 seconds, Fetched: 1 row(s)
hive>
```

# Increment 1 - Query 4 - Tweets from Southern Hemosphere


cloudera@quickstart:~/Downloads
File Edit View Search Terminal Help

```
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1541805655545_0009, Tracking URL = http://quickstart.cloudera
:8088/proxy/application_1541805655545_0009/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1541805655545_0009
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-11-09 19:20:39,587 Stage-1 map = 0%,  reduce = 0%
2018-11-09 19:20:50,903 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.13 se
c
2018-11-09 19:21:00,722 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.82
sec
MapReduce Total cumulative CPU time: 4 seconds 820 msec
Ended Job = job_1541805655545_0009
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.82 sec   HDFS Read: 3226057
9 HDFS Write: 2 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 820 msec
OK
0
Time taken: 32.725 seconds, Fetched: 1 row(s)
hive> 
```

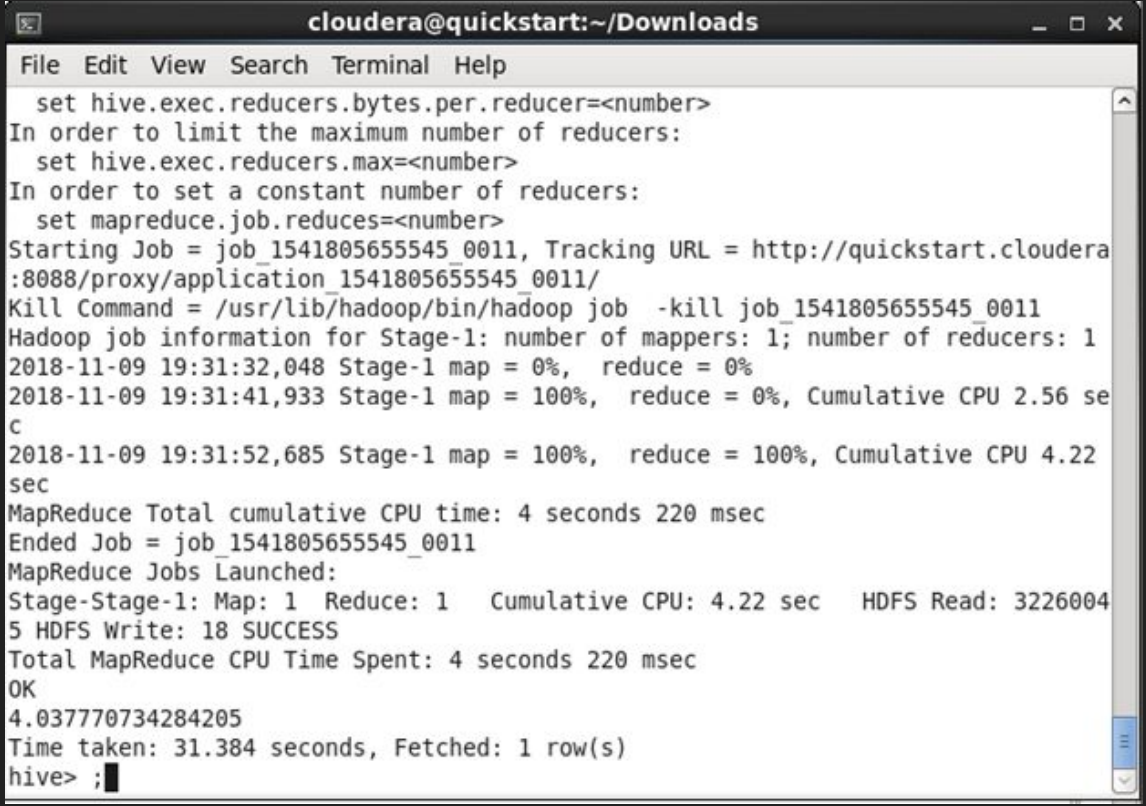
# Increment 1 - Query 5 - Contain “the”



cloudera@quickstart:~/Downloads

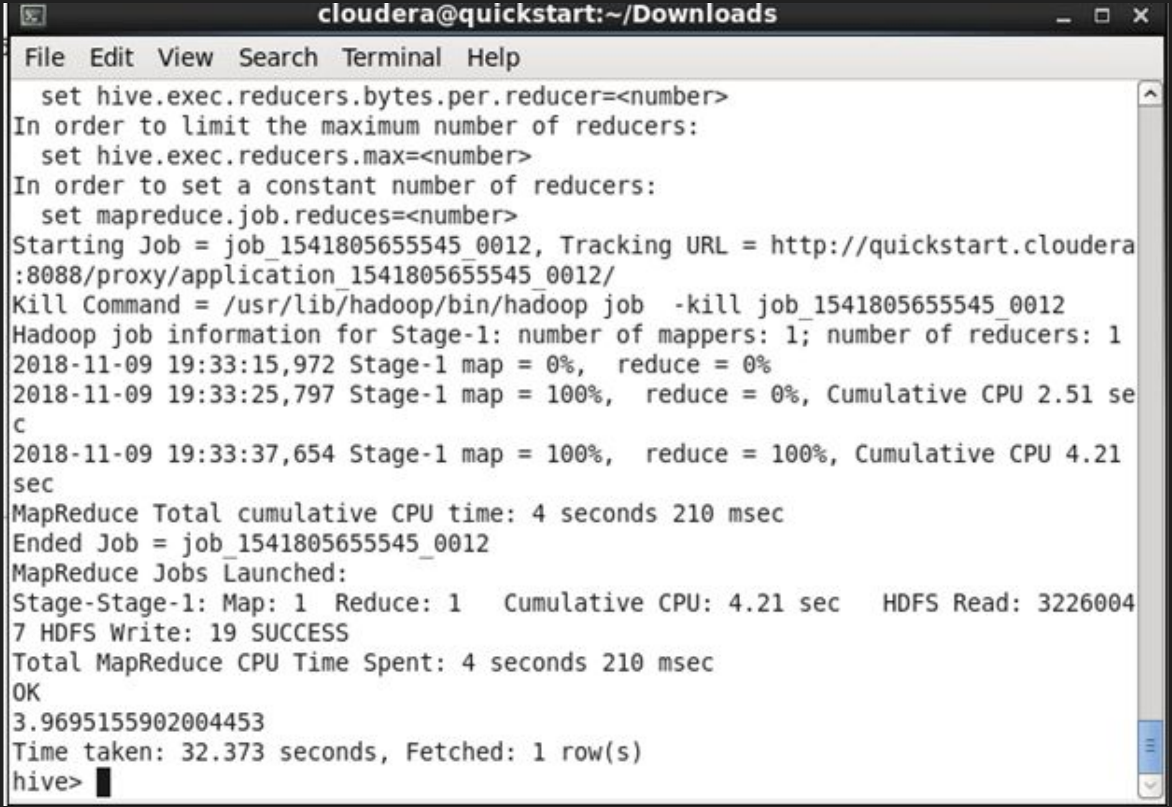
```
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1541805655545_0010, Tracking URL = http://quickstart.cloudera
:8088/proxy/application_1541805655545_0010/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1541805655545_0010
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-11-09 19:26:42,716 Stage-1 map = 0%,  reduce = 0%
2018-11-09 19:26:55,052 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.78 se
c
2018-11-09 19:27:05,778 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.42
sec
MapReduce Total cumulative CPU time: 4 seconds 420 msec
Ended Job = job_1541805655545_0010
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.42 sec   HDFS Read: 3225967
6 HDFS Write: 6 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 420 msec
OK
19053
Time taken: 33.608 seconds, Fetched: 1 row(s)
hive>
```

## Increment 1 - Query 6 - Average Lat.


cloudera@quickstart:~/Downloads
File Edit View Search Terminal Help

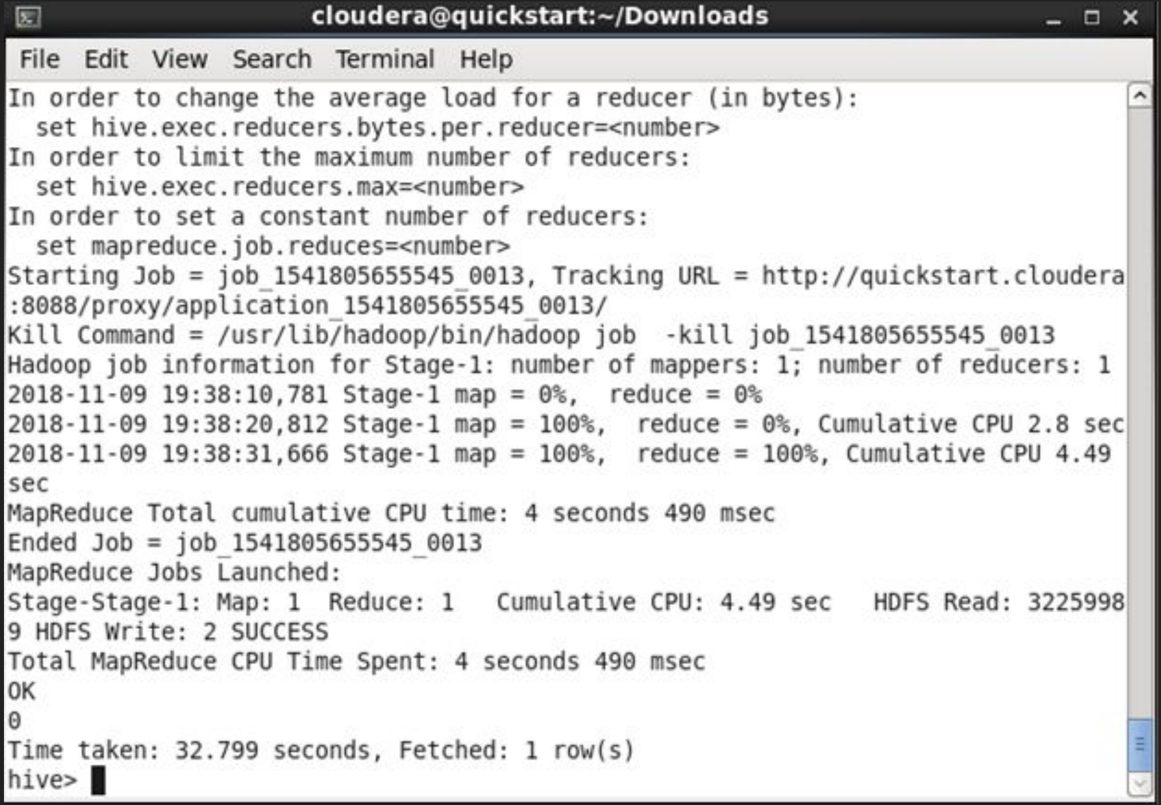
```
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1541805655545_0011, Tracking URL = http://quickstart.cloudera
:8088/proxy/application_1541805655545_0011/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1541805655545_0011
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-11-09 19:31:32,048 Stage-1 map = 0%,  reduce = 0%
2018-11-09 19:31:41,933 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.56 se
c
2018-11-09 19:31:52,685 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.22
sec
MapReduce Total cumulative CPU time: 4 seconds 220 msec
Ended Job = job_1541805655545_0011
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.22 sec   HDFS Read: 3226004
5 HDFS Write: 18 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 220 msec
OK
4.037770734284205
Time taken: 31.384 seconds, Fetched: 1 row(s)
hive> ;
```

# Increment 1 - Query 7 - Average Longitude


cloudera@quickstart:~/Downloads

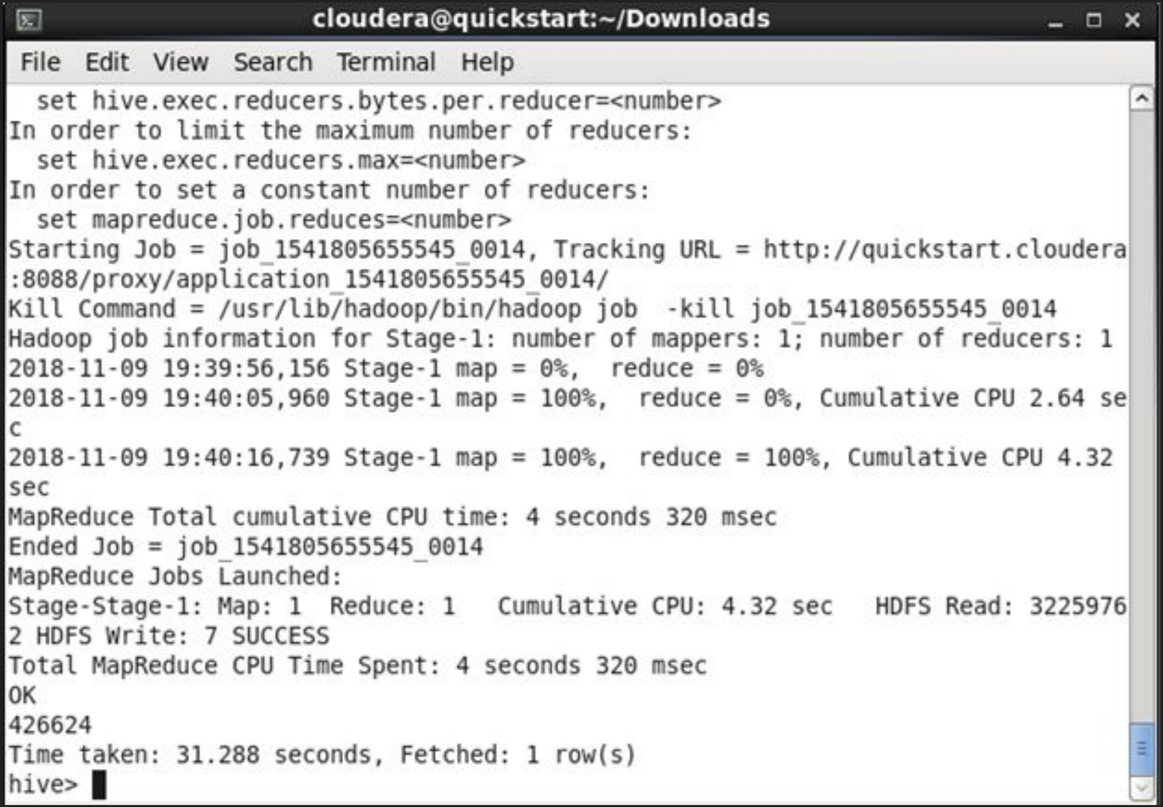
```
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1541805655545_0012, Tracking URL = http://quickstart.cloudera
:8088/proxy/application_1541805655545_0012/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1541805655545_0012
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-11-09 19:33:15,972 Stage-1 map = 0%,  reduce = 0%
2018-11-09 19:33:25,797 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.51 se
c
2018-11-09 19:33:37,654 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.21
sec
MapReduce Total cumulative CPU time: 4 seconds 210 msec
Ended Job = job_1541805655545_0012
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.21 sec   HDFS Read: 3226004
7 HDFS Write: 19 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 210 msec
OK
3.9695155902004453
Time taken: 32.373 seconds, Fetched: 1 row(s)
hive> 
```

## Increment 1 - Query 8 - Contain “User” and “Generated”


cloudera@quickstart:~/Downloads
File Edit View Search Terminal Help

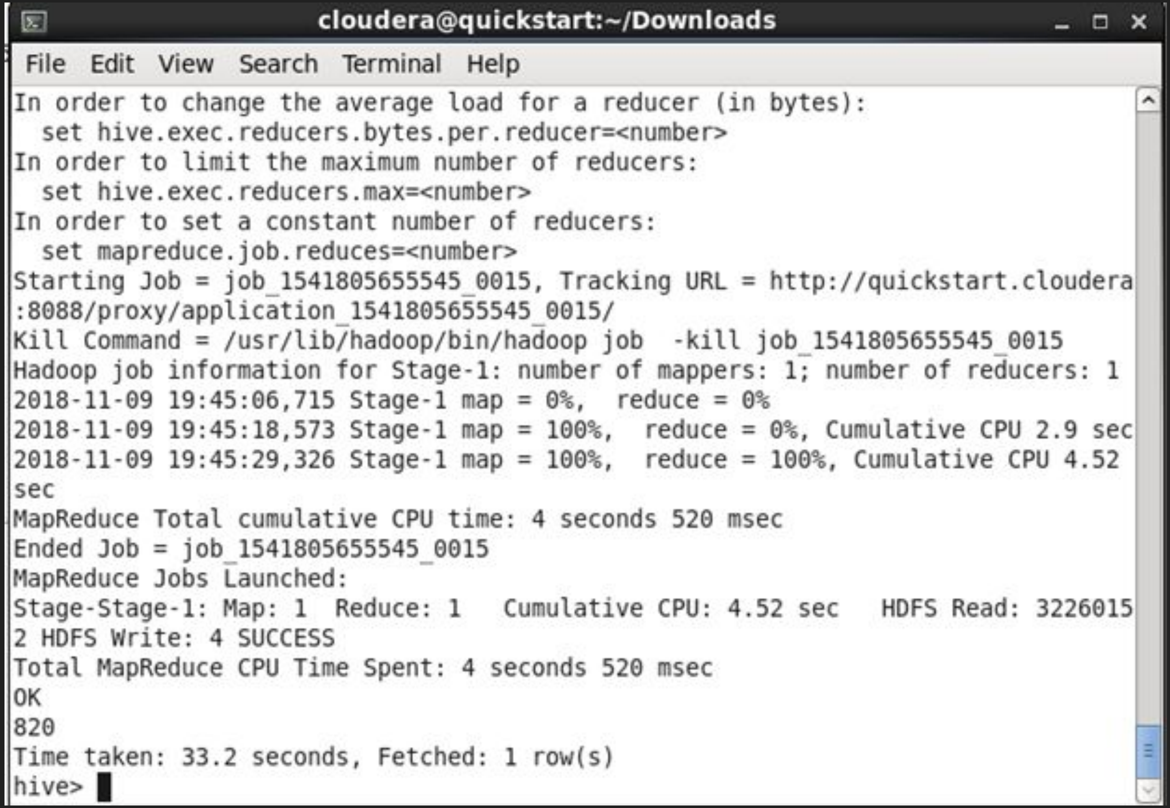
```
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1541805655545_0013, Tracking URL = http://quickstart.cloudera
:8088/proxy/application_1541805655545_0013/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1541805655545_0013
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-11-09 19:38:10,781 Stage-1 map = 0%,  reduce = 0%
2018-11-09 19:38:20,812 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.8 sec
2018-11-09 19:38:31,666 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.49
sec
MapReduce Total cumulative CPU time: 4 seconds 490 msec
Ended Job = job_1541805655545_0013
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.49 sec   HDFS Read: 3225998
9 HDFS Write: 2 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 490 msec
OK
0
Time taken: 32.799 seconds, Fetched: 1 row(s)
hive> 
```

## Increment 1 - Query 9 - Total Retweets


cloudera@quickstart:~/Downloads
File Edit View Search Terminal Help

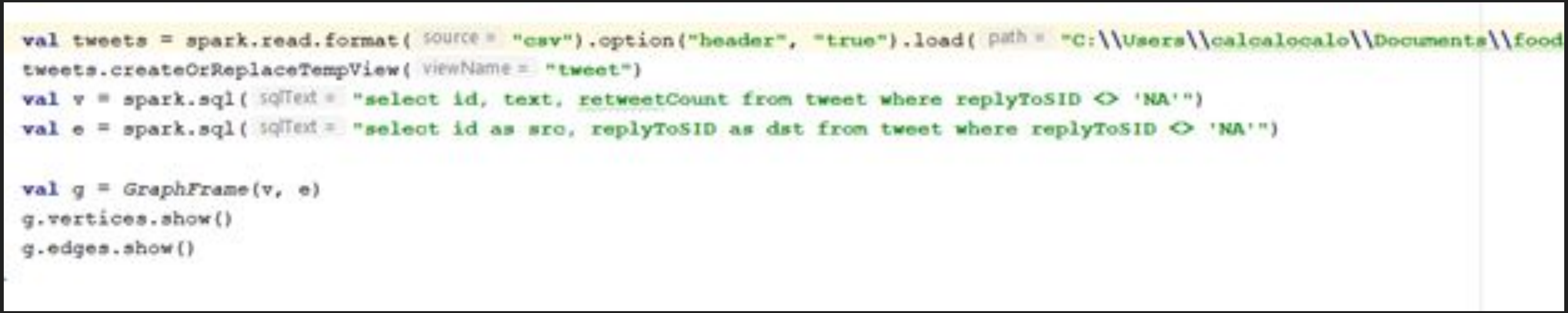
```
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1541805655545_0014, Tracking URL = http://quickstart.cloudera
:8088/proxy/application_1541805655545_0014/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1541805655545_0014
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-11-09 19:39:56,156 Stage-1 map = 0%,  reduce = 0%
2018-11-09 19:40:05,960 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.64 se
c
2018-11-09 19:40:16,739 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.32
sec
MapReduce Total cumulative CPU time: 4 seconds 320 msec
Ended Job = job_1541805655545_0014
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.32 sec   HDFS Read: 3225976
2 HDFS Write: 7 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 320 msec
OK
426624
Time taken: 31.288 seconds, Fetched: 1 row(s)
hive> 
```

# Increment 1 - Query 10 - Longest Tweet


cloudera@quickstart:~/Downloads

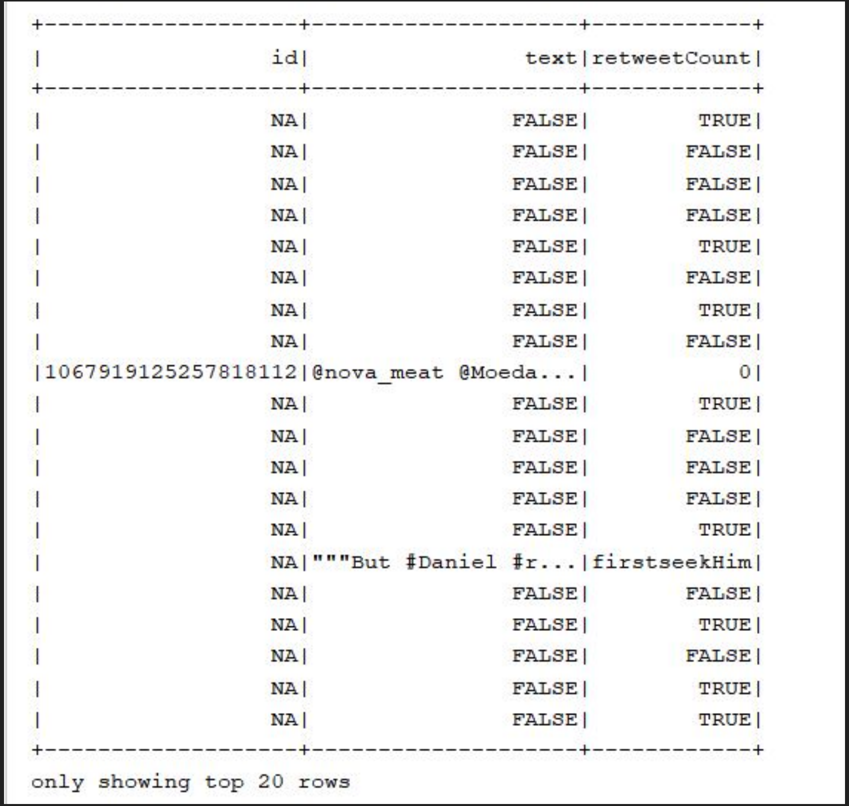
```
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1541805655545_0015, Tracking URL = http://quickstart.cloudera
:8088/proxy/application_1541805655545_0015/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1541805655545_0015
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-11-09 19:45:06,715 Stage-1 map = 0%,  reduce = 0%
2018-11-09 19:45:18,573 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.9 sec
2018-11-09 19:45:29,326 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.52
sec
MapReduce Total cumulative CPU time: 4 seconds 520 msec
Ended Job = job_1541805655545_0015
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.52 sec   HDFS Read: 3226015
2 HDFS Write: 4 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 520 msec
OK
820
Time taken: 33.2 seconds, Fetched: 1 row(s)
hive>
```

# Increment 2 - Code



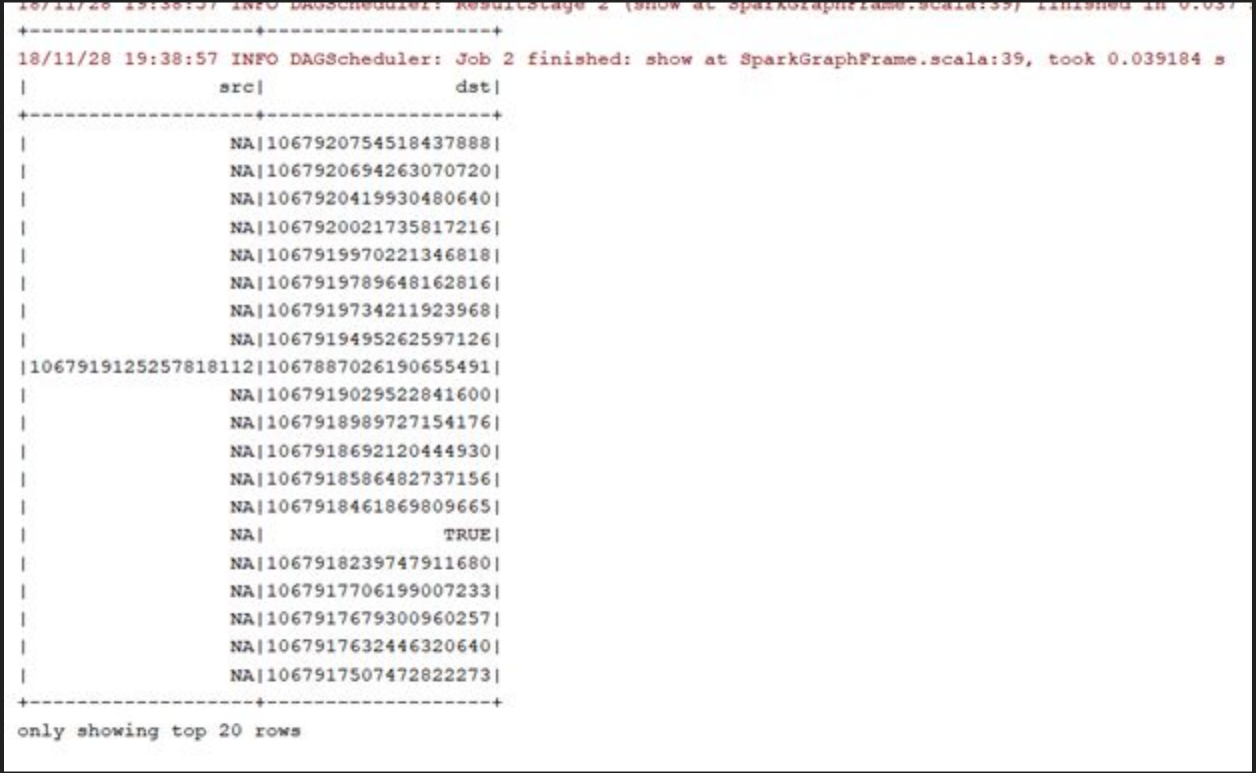
```
val tweets = spark.read.format( source = "csv").option("header", "true").load( path = "C:\\Users\\calcalocalo\\Documents\\food
tweets.createOrReplaceTempView( viewName = "tweet")
val v = spark.sql( sqlText = "select id, text, retweetCount from tweet where replyToSID <> 'NA'")
val e = spark.sql( sqlText = "select id as src, replyToSID as dst from tweet where replyToSID <> 'NA'")

val g = GraphFrame(v, e)
g.vertices.show()
g.edges.show()
```

## Increment 2 - Edges



```
+------------------+--------------------+------------+
|                id|                text|retweetCount|
+------------------+--------------------+------------+
|                NA|               FALSE|        TRUE|
|                NA|               FALSE|       FALSE|
|                NA|               FALSE|       FALSE|
|                NA|               FALSE|       FALSE|
|                NA|               FALSE|        TRUE|
|                NA|               FALSE|       FALSE|
|                NA|               FALSE|        TRUE|
|                NA|               FALSE|       FALSE|
|1067919125257818112|@nova_meat @Moeda...|           0|
|                NA|               FALSE|        TRUE|
|                NA|               FALSE|       FALSE|
|                NA|               FALSE|       FALSE|
|                NA|               FALSE|       FALSE|
|                NA|               FALSE|        TRUE|
|                NA|"""But #Daniel #r...|firstseekHim|
|                NA|               FALSE|       FALSE|
|                NA|               FALSE|        TRUE|
|                NA|               FALSE|       FALSE|
|                NA|               FALSE|        TRUE|
|                NA|               FALSE|        TRUE|
+------------------+--------------------+------------+
only showing top 20 rows
```

# Increment 2 - Vertices



```
18/11/28 19:38:57 INFO DAGScheduler: ResultStage 2 (show at SparkGraphFrame.scala:39) finished in 0.037
+-------------------+-------------------+
18/11/28 19:38:57 INFO DAGScheduler: Job 2 finished: show at SparkGraphFrame.scala:39, took 0.039184 s
|                src|                dst|
+-------------------+-------------------+
|                 NA|1067920754518437888|
|                 NA|1067920694263070720|
|                 NA|1067920419930480640|
|                 NA|1067920021735817216|
|                 NA|1067919970221346818|
|                 NA|1067919789648162816|
|                 NA|1067919734211923968|
|                 NA|1067919495262597126|
|1067919125257818112|1067887026190655491|
|                 NA|1067919029522841600|
|                 NA|1067918989727154176|
|                 NA|1067918692120444930|
|                 NA|1067918586482737156|
|                 NA|1067918461869809665|
|                 NA|               TRUE|
|                 NA|1067918239747911680|
|                 NA|1067917706199007233|
|                 NA|1067917679300960257|
|                 NA|1067917632446320640|
|                 NA|1067917507472822273|
+-------------------+-------------------+
only showing top 20 rows
```