New York Times Data Analysis using Google Cloud Dataproc and Hive

Following are the steps implemented:

# 1. Login, connected to the console on Google cloud platform

# 2. Loaded all the csv, dictionary files on Google Storage(HDFS shell)

# 3. Connected to the Hive shell

# 4. Created all Articles-2017, 2018 and Comments-2017 and 2018 tables on Hive

# 5. Executed all desired queries

# 6. Download all the queries csv file to local

#===

#1. Create cloud compute engine and enable API on Google cloud platform

# Google Cloud Dataproc API and enable API-

Name - API key 1

AIzaSyAz7_rM2mkgBFbFSHfF1uaA52pad2TJPqk

# Create cluster on Google Dataproc API

create cluster > Go to big data > Dataproc > Create cluster

cluster name: nyt-market-cluster

#===

#2. Loaded all the csv, dictionary files on Google Storage(HDFS shell)

# Google Storage and save all csv and dictionary files --

create bucket > Go to storage > browser- create bucket

name - nyt-market-bucket

create data folder- upload files > Articles2017.csv, Articles2018.csv, CommentYear2017, CommentYear2018

create logs folder

create output folder

#===

#3. Connected to the Hive shell

```
#Enter the below command on HDFS shell

#Check all the folders and files presented in HDFS shell

   gsutil ls gs://nyt-market-bucket/


# Submit the job if required --

   Job submit -

   gcloud dataproc jobs submit hive --cluster=nyt-market-cluster \

     --file=gs://nyt-market-bucket/Queries/articlequery2017.txt


# Below is the command to connect with HIVE shell

Format - beeline -u jdbc:hive2://localhost:10000/default -n myusername@market-data-cluster-m -d
org.apache.hive.jdbc.HiveDriver

Actual command - beeline -u jdbc:hive2://localhost:10000/default -n test@nyt-market-cluster-m -d
org.apache.hive.jdbc.HiveDriver


#==================================================================================

#4. Created all Articles-2017, 2018 and Comments-2017 and 2018 tables on Hive


# Databases created:

   create database NYTimes;

# Tables created:


#Article Year 2017 -

create external table if not exists articleyear2017(Month_Name STRING,articleID STRING,abstract
STRING,byline STRING,documentType STRING,headline STRING,keywords STRING,multimedia
INT,newDesk STRING,printPage INT,pubDate TIMESTAMP,source STRING,typeOfMaterial
STRING,webURL STRING,articleWordCount BIGINT) ROW FORMAT DELIMITED FIELDS TERMINATED BY
```

```
'\t' STORED AS TEXTFILE location 'gs://nyt-market-bucket/Article2017/' TBLPROPERTIES
('skip.header.line.count'='1');
```

#Sequence of columns to show in tables-

```
  Month_Name        articleID       abstract        byline documentType headline
        keywords         multimedia      newDesk         printPage       pubDate         source
        typeOfMaterial   webURL          articleWordCount
```

#Article Year 2018 -

```
  create external table if not exists articleyear2018(Month_Name STRING, abstract STRING, articleID
STRING,byline STRING,documentType STRING,headline STRING,keywords STRING,multimedia INT,
newDesk STRING,printPage INT,pubDate TIMESTAMP,typeOfMaterial STRING,webURL STRING,
articleWordCount BIGINT) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' STORED AS TEXTFILE
location 'gs://nyt-market-bucket/Article2018/' TBLPROPERTIES ('skip.header.line.count'='1');
```

#Sequence of columns to show in tables-

```
Month_Name   abstractarticleID       byline   documentType  headline               keywords
        multimedia      newDesk        printPage       pubDate         typeOfMaterial webURL
        articleWordCount
```

#Comment Year 2017 -

```
  create external table if not exists CommentYear2017(Month_Name STRING,approveDate
STRING,articleID STRING,articleWordCount BIGINT,commentBody STRING,commentID
STRING,commentSequence STRING,commentTitle STRING,commentType STRING,createDate
STRING,depth INT,editorsSelection INT,inReplyTo STRING,newDesk STRING,parentID
STRING,parentUserDisplayName STRING, permID STRING, picURL STRING, printPage INT,
recommendations INT, recommendedFlag INT, replyCount INT, reportAbuseFlag INT, sectionName
STRING, sharing INT, status STRING, timespeople INT, trusted INT, updateDate STRING,
userDisplayName STRING, userID STRING, userLocation STRING, userTitle STRING, userURL
STRING,typeofmaterial STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' STORED AS
TEXTFILE location "gs://nyt-market-bucket/CommentYear2017/" TBLPROPERTIES
('skip.header.line.count'='1');
```

#CommentYear2018 -

```
  create external table if not exists CommentYear2018(Month_Name STRING,approveDate
STRING,articleID STRING,articleWordCount BIGINT,commentBody STRING,commentID
STRING,commentSequence STRING,commentTitle STRING,commentType STRING,createDate
STRING,depth INT,editorsSelection INT,inReplyTo STRING,newDesk STRING,parentID
STRING,parentUserDisplayName STRING, permID STRING, picURL STRING, printPage INT,
recommendations INT, recommendedFlag INT, replyCount INT, reportAbuseFlag INT, sectionName
STRING, sharing INT, status STRING, timespeople INT, trusted INT, typeofmaterial STRING, updateDate
STRING, userDisplayName STRING, userID STRING, userLocation STRING, userTitle STRING, userURL
STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' STORED AS TEXTFILE location"gs://nyt-
market-bucket/CommentYear2018/" TBLPROPERTIES ('skip.header.line.count'='1');
```

#Sequence of columns to show in tables-

```
Month_name approveDate       articleID       articleWordCount        commentBody commentID
     commentSequence         commentTitle    commentType     createDate     depth
     editorsSelection        inReplyTo       newDesk         parentID
     parentUserDisplayName   permID picURL   printPage       recommendations recommendedFlag
                     replyCount      reportAbuseFlag         sectionName           sharing status
     timespeople     trusted typeOfMaterial  updateDate      userDisplayName       userID
     userLocation    userTitle       userURL
```

#Created Dictionary -

```
CREATE EXTERNAL TABLE if not exists dictionary (type string,length int,word string,pos string,
stemmed string, polarity string ) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' STORED AS
TEXTFILE LOCATION "gs://nyt-market-bucket/Dictonary1/"
```

#5. Executed all desired queries

#All Article queries --

#===

#Query-1, Show the count of document type by type of material for the year 2017 and 2018?

#For year 2017 -

#query1-2017.csv

```
insert overwrite directory 'gs://nyt-market-bucket/output/query1-2017' row format delimited fields
terminated by ',' SELECT documentType,count(typeOfMaterial) from articleyear2017 GROUP BY
documentType;
```

#For year 2018 -
#query1-2018.csv

```
insert overwrite directory 'gs://nyt-market-bucket/output/query1-2018' row format delimited fields
terminated by ',' SELECT documentType, count(typeOfMaterial) from articleyear2018 GROUP BY
documentType;
```

```
#Show the count of type of material with respect to Articles for the year 2017 and 2018?

#query1-2017-article.csv

insert overwrite directory 'gs://nyt-market-bucket/output/query1-2017-article' row format delimited
fields terminated by ',' SELECT typeOfMaterial, count(typeOfMaterial) from articleyear2017 where
documentType = "article" GROUP BY typeOfMaterial;


#query1-2017-blogpost.csv

insert overwrite directory 'gs://nyt-market-bucket/output/query1-2017-blogpost' row format
delimited fields terminated by ',' SELECT typeOfMaterial, count(typeOfMaterial) from articleyear2017
where documentType = "blogpost" GROUP BY typeOfMaterial;


#query1-2018-docType-Material.csv

insert overwrite directory 'gs://nyt-market-bucket/output/query1-2018-docType' row format
delimited fields terminated by ',' SELECT documentType, typeOfMaterial, count(typeOfMaterial) from
articleyear2018 GROUP BY documentType, typeOfMaterial;

#query1-2018-article.csv

insert overwrite directory 'gs://nyt-market-bucket/output/query1-2018-article' row format delimited
fields terminated by ',' SELECT documentType, typeOfMaterial, count(typeOfMaterial) from
articleyear2018 GROUP BY documentType, typeOfMaterial;


#query1-2018-blogpost.csv

insert overwrite directory 'gs://nyt-market-bucket/output/query1-2018-blogpost' row format
delimited fields terminated by ',' SELECT typeOfMaterial, count(typeOfMaterial) from articleyear2017
where documentType = "blogpost" GROUP BY typeOfMaterial;

#============================================================================================
```

#Query 2: What is the reply count for the document type month wise for Year 2017 & 2018?

#For year 2017 -

#query2-2017-article.csv

```
insert overwrite directory 'gs://nyt-market-bucket/output/query2-2017-article' row format delimited
fields terminated by ',' SELECT a.Month_Name,count(a.documentType) as doctype ,count(c.replycount)
as replycount FROM articleyear2017 a LEFT OUTER JOIN commentyear2017 c ON (a.articlewordcount =
c.articlewordcount) where a.documentType ="article" Group BY a.Month_Name;
```

#For year 2018 -

#query2-2018-article.csv

```
insert overwrite directory 'gs://nyt-market-bucket/output/query2-2018-article' row format delimited
fields terminated by ',' SELECT a.Month_Name,count(a.documentType) as doctype ,count(c.replycount)
as replycount FROM articleyear2018 a LEFT OUTER JOIN commentyear2018 c ON (a.articlewordcount =
c.articlewordcount) where a.documentType ="article" Group BY a.Month_Name;
```

#query2-2017-art-commt.csv

```
insert overwrite directory 'gs://nyt-market-bucket/output/query2-2017-art-commt' row format
delimited fields terminated by ',' SELECT a.Month_Name,count(a.documentType) as doctype
,count(c.replycount) as replycount FROM articleyear2017 a LEFT OUTER JOIN commentyear2017 c ON
(a.articlewordcount = c.articlewordcount) where a.documentType ="blogpost" Group BY
a.Month_Name;
```

#query2-2018-art-commt.csv

```
insert overwrite directory 'gs://nyt-market-bucket/output/query2-2018-art-commt' row format
delimited fields terminated by ',' SELECT a.Month_Name,count(a.documentType) as doctype
,count(c.replycount) as replycount FROM articleyear2018 a LEFT OUTER JOIN commentyear2018 c ON
(a.articlewordcount = c.articlewordcount) where a.documentType ="blogpost" Group BY
a.Month_Name;
```

#==

```
#Query-3: What is the reply count for each comment type?

#For year 2017 -

#query3-2017-comment.csv

insert overwrite directory 'gs://nyt-market-bucket/output/query3-2017-comment' row format
delimited fields terminated by ',' SELECT commentType, count (replyCount), rank () over (ORDER BY
count (replyCount) desc) AS rank from commentyear2017 GROUP BY commentType limit 3;

#For year 2018 -

#query3-2018-comment.csv

insert overwrite directory 'gs://nyt-market-bucket/output/query3-2018-comment' row format
delimited fields terminated by ',' SELECT commentType, count (replyCount), rank () over (ORDER BY
count (replyCount) desc) AS rank from commentyear2018 GROUP BY commentType limit 3;

#====================================================================================

#Query-4: What is the count of new desk month wise?

#For Year 2017 -

#query4-2017-article.csv

insert overwrite directory 'gs://nyt-market-bucket/output/query4-2017-article' row format delimited
fields terminated by ',' SELECT documenttype, count(newDesk),month_name FROM articleyear2017
where documenttype='article' GROUP BY month_name, documenttype;


#query4-2017-blogpost.csv

insert overwrite directory 'gs://nyt-market-bucket/output/query4-2017-blogpost' row format
delimited fields terminated by ',' SELECT documenttype, count(newDesk),month_name FROM
articleyear2017 where documenttype='blogpost' GROUP BY month_name, documenttype;

#For Year 2018 -

#query4-2018-article.csv

insert overwrite directory 'gs://nyt-market-bucket/output/query4-2018-article' row format delimited
fields terminated by ',' SELECT documenttype, count(newDesk),month_name FROM articleyear2018
where documenttype='article' GROUP BY month_name, documenttype;

#query4-2018-blogpost.csv

insert overwrite directory 'gs://nyt-market-bucket/output/query4-2018-blogpost' row format
delimited fields terminated by ',' SELECT documenttype, count(newDesk),month_name FROM
articleyear2018 where documenttype='blogpost' GROUP BY month_name, documenttype;

#====================================================================================
#Query-5; What is the count of new desk based on recommendations?

#For Year 2017 -

#query5-2017

insert overwrite directory 'gs://nyt-market-bucket/output/query5-2017' row format delimited fields
terminated by ',' SELECT newDesk,count(recommendations),rank() over (order by
count(recommendations)desc) AS rank from commentyear2017 where newDesk LIKE 'OpEd' OR
newDesk LIKE 'National' OR newDesk LIKE 'Business' OR newDesk LIKE 'Foreign' OR newDesk LIKE
'Editorial' OR newDesk LIKE 'Magazine' OR newDesk LIKE 'Learning' GROUP BY newDesk;


#For Year 2018 -

#query5-2018

insert overwrite directory 'gs://nyt-market-bucket/output/query5-2018' row format delimited fields
terminated by ',' SELECT newDesk,count(recommendations),rank() over (order by
count(recommendations)desc) AS rank from commentyear2018 where newDesk LIKE 'OpEd' OR
newDesk LIKE 'National' OR newDesk LIKE 'Business' OR newDesk LIKE 'Foreign' OR newDesk LIKE
'Editorial' OR newDesk LIKE 'Magazine' OR newDesk LIKE 'Learning' GROUP BY newDesk;


#query5-2017-recomnd -

insert overwrite directory 'gs://nyt-market-bucket/output/query5-2017-recomnd' row format
delimited fields terminated by ',' SELECT sectionname,count(recommendations),rank() over (order by
count(recommendations)desc) AS rank from commentyear2017 where sectionname LIKE 'Art & Design'
OR sectionname LIKE 'Economy' OR sectionname LIKE 'Music' OR sectionname LIKE 'Media' OR
sectionname LIKE 'Personal Tech' OR sectionname LIKE 'The Daily' OR sectionname LIKE 'Book Review'
GROUP BY sectionname;


#query5-2018-recomnd -

insert overwrite directory 'gs://nyt-market-bucket/output/query5-2018-recomnd' row format
delimited fields terminated by ',' SELECT sectionname,count(recommendations),rank() over (order by
count(recommendations)desc) AS rank from commentyear2018 where sectionname LIKE 'Art & Design'
OR sectionname LIKE 'Economy' OR sectionname LIKE 'Music' OR sectionname LIKE 'Media' OR
sectionname LIKE 'Personal Tech' OR sectionname LIKE 'The Daily' OR sectionname LIKE 'Book Review'
GROUP BY sectionname;

#=============================================================================
```

#Query 6: What is the degree of polarity by most positive headlines for the year 2017?

# Dictionary

https://s3.amazonaws.com/hipicdatasets/dictionary.tsv

```
create view IF NOT EXISTS l1 as select articleid,words from commentyear2017 lateral view
explode(sentences(lower(commentbody))) dummy as words;
```

```
create view IF NOT EXISTS l2 as select articleid, word from l1 lateral view explode(words) dummy as
word;
```

```
create view IF NOT EXISTS l3 as select articleid, l2.word, case d.polarity when 'negative' then -1 when
'positive' then 1 else 0 end as polarity from l2 left outer join dictionary d on l2.word = d.word;
```

```
create table IF NOT EXISTS sentiment_aggregate stored as orc as select articleid,sum( polarity )
sentiment from l3 group by articleid;
```

#query6-2017.csv

```
insert overwrite directory 'gs://nyt-market-bucket/output/query6-2017' row format delimited fields
terminated by ',' select sentiment_aggregate.sentiment,articleyear2017.headline from articleyear2017
inner join sentiment_aggregate on sentiment_aggregate.articleid=articleyear2017.articleid order by
sentiment asc limit 10;
```

#==

#Query-7; What is the degree of polarity by most negative headlines?

#query7-2017-negative-senti.csv

```
insert overwrite directory 'gs://nyt-market-bucket/output/query7-2017-negative-senti' row format
delimited fields terminated by ',' select sentiment_aggregate.sentiment,articleyear2017.headline from
articleyear2017 inner join sentiment_aggregate on
sentiment_aggregate.articleid=articleyear2017.articleid order by sentiment desc limit 10;
```

#==

#Query 9: Where is most active user’s from based on replycount for the year 2017?

#For Year 2017 -

```
insert overwrite directory 'gs://nyt-market-bucket/output/query9-2017' row format delimited fields
terminated by ',' select userLocation, count(replycount), rank() over (order by count(replycount)desc) AS
rank from commentyear2017 group by userLocation limit 100;
```

#==

#Query 10: Where is most active user’s from based on replycount for the year 2018?

#query10-2018.csv

```
insert overwrite directory 'gs://nyt-market-bucket/output/query10-2018' row format delimited fields
terminated by ',' select userLocation, count(replycount), rank() over (order by count(replycount)desc) AS
rank from commentyear2018 group by userLocation limit 100;
```

#===

#Query 12: Most Popular Author(byline) with respect to recommendations of public for the year 2017?

```
create table if not exists author1_byline2 as select sum(recommendations) as
recommendations,articleid from commentyear2017 group by articleid;
```

```
create table final_byline as select author1_byline2.recommendations
recommendations_count,author1_byline2.articleid articleid,articleyear2017.byline author from
author1_byline2 inner join articleyear2017 on author1_byline2.articleid = articleyear2017.articleid;
```

```
select * from final_byline order by recommendations_count desc limit 10;
```

#query12-2018.csv

```
insert overwrite directory 'gs://nyt-market-bucket/output/query12-2018' row format delimited fields
terminated by ',' select * from final_byline order by recommendations_count desc limit 10;
```

#===

#Query 13: Most Popular Author(byline) with respect to recommendations of public for the year 2018?

```
create table if not exists author1_byline2_2018 as select sum(recommendations) as
recommendations,articleid from commentyear2018 group by articleid;
```

```
create table final_byline_2018 as select author1_byline2_2018.recommendations
recommendations_count,author1_byline2_2018.articleid articleid,articleyear2018.byline author from
author1_byline2_2018 inner join articleyear2018 on author1_byline2_2018.articleid =
articleyear2018.articleid;
```

```
select * from final_byline_2018 order by recommendations_count desc limit 10;
```

#query13-2018.csv

```
insert overwrite directory 'gs://nyt-market-bucket/output/query13-2018' row format delimited fields
terminated by ',' select * from final_byline_2018 order by recommendations_count desc limit 10;
```

==

# Set up Hadoop on google cloud Dataproc

http://holowczak.com/getting-started-with-hive-on-google-cloud-dataproc/

https://medium.com/@cuongdo.uconn/how-to-hive-on-google-cloud-platform-dataproc-and-storage-d141536644cd

# Ways to upload multiple files to google cloud storage

https://www.youtube.com/watch?v=ji1DWCTI05A

# From Hive data to CSV file -

https://community.cloudera.com/t5/Support-Questions/how-to-download-hive-data-into-csv- format/td-p/59591

```
wget -O dictionary.tsv https://s3.amazonaws.com/hipicdatasets/dictionary.tsv
```