

# Northeastern University

## Final Project Report



**Topic: Yelp Dataset Analysis**

Engineering of Big Data  
Academic Year: Spring 2021

Professor: Mr. Yusuf Ozbek  
TA: Deepak Gopalan

**Submitted by: Poojitha Muppalla**

**Summary:**

Yelp Dataset Analysis is the Dataset that I wanted to use for project. Dataset is downloaded from the Kaggle Website which can be found at <https://www.kaggle.com/ksjpswaroop/yelp-data-analysis/data>

The dataset has millions of rows. The dataset has 5 csv files namely yelp\_review.json, yelp\_business.json, yelp\_user.json, yelp\_tip.json and yelp\_checking.json. In which I will be working on 3 namely yelp\_review.json, yelp\_business.json and yelp\_user.json.

**About the Dataset:**

- **Yelp\_review**

**Attribute used:**

1. Review Id
2. User ID
3. Business ID
4. Review
5. Review Date
6. Review Comment

- **Yelp\_business**

**Attributes Used:**

1. Business\_id
2. Business Name
3. City
4. State
5. Postal
6. Rating

- **Yelp\_user**

**Attribute Used:**

1. User Id
2. Name
3. Year Joined

I have worked upon the below analysis for the project:

S.No	Analysis Description	Implementation Details
1	Number of business in a city	<b>MongoDB</b>
2	To find which user has posted reviews for businesses	<b>Apache Pig – Replicated Join</b>
3	To find details of Business that have user reviews less than 100.	<b>Apache Pig</b>
4	To find top 3 business based on rating for each city.	<b>Apache Hive</b>
5	To find list of businesses that have above average rating in each state	<b>Apache Hive</b>
6	Partitioning on the basis of year user joined the yelp	<b>MapReduce – Data Organization Partitioning</b>
7	Calculate of no. of users signed up on yelp each year	<b>MapReduce – Chaining Map Reduce Summarization</b>
8	Users who have yelp account but never provided any reviews yet	<b>MapReduce – Joins</b>
9	To find the top 25 most reviewed business by users	<b>MapReduce – Filtering Techniques Top n Filtering Pattern</b>
10	To find out how well a business is been rated by users per year	<b>MapReduce – Secondary Sort</b>
11	Mahout Recommendation	<b>Data Cleaning + Mahout Recommendation</b>
12	Output Visualization	<b>Tableau</b>

## Getting Started

### STARTING HADOOP

1. Navigate to Hadoop Directory:  

```
cd /usr/local/cellar/hadoop/3.3.0/sbin
```
2. Start Hadoop daemons  

```
./start-all.sh
```
3. Check Hadoop Daemons  

```
jps
```
4. Navigate to Hadoop UI localhost:9870/

```
● ● ● sbin — -zsh — 97x24
poojithamuppalla@Poojithas-MacBook-Pro sbin % ./start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as poojithamuppalla in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [Poojithas-MacBook-Pro.local]
2021-04-19 10:09:48,370 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Starting resourcemanager
Starting nodemanagers
poojithamuppalla@Poojithas-MacBook-Pro sbin % jps
26342 DataNode
26664 ResourceManager
26762 NodeManager
26826 Jps
26475 SecondaryNameNode
poojithamuppalla@Poojithas-MacBook-Pro sbin % hadoop fs -mkdir -p /FinalProject/Dataset/
2021-04-19 10:11:44,315 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
mkdir: Call From Poojithas-MacBook-Pro.local/127.0.0.1 to localhost:8020 failed on connection
eption: java.net.ConnectException: Connection refused; For more details see: http://wiki.apache.org/hadoop/ConnectionRefused
.....
```

#### COPYING FILE FROM LOCAL FILE SYSTEM TO HDFS

```
poojithamuppalla@Poojithas-MacBook-Pro sbin % hadoop fs -mkdir -p /FinalProject/Dataset/
2021-04-19 10:11:44,315 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
.....
```

## MONGODB ANALYSIS:

**Analysis # 01:** To find the count of number of business per city:

### 1. Create Mongo Database:

```
> use finalProject;
switched to db finalProject
```

### 2. Create Collection

```
> db.createCollection('CityRestaurants');
{ "ok" : 1 }
```

```
...nt/FinalResult1/ | ...sbin — mongod ... | ...bin — mongo | ~ — -bash | +
> show database;
[2019-08-12T19:52:25.197-0400 E QUERY      [js] Error: don't know how to show [data]
base] :
[shellHelper.show@src/mongo/shell/utils.js:1055:11
shellHelper@src/mongo/shell/utils.js:766:15
@(shellhelp2):1:1
> show databases;
Part4DB    0.002GB
admin      0.000GB
config     0.000GB
local      0.000GB
moviedb   0.000GB
movielens  0.036GB
mydb       0.694GB
part6     0.000GB
schooldb   0.000GB
> Use FinalProject;
2019-08-12T19:52:50.829-0400 E QUERY      [js] SyntaxError: missing ; before state
ment @(shell):1:4
> use finalProject;
switched to db finalProject
> db.createCollection('CityRestaurants');
{ "ok" : 1 }
```

### 3. Import csv file to MongoDB:

```
mongoimport --db finalProject --collection CityRestaurants --type csv --file
/Users/poojithamuppalla/Downloads/Project/Dataset/yelp_business.csv -headerline;
```

### 4 . Write Map & Reduce Functions to perform Analysis:

```
var map= function (){ var key =
this.city; var value = {count:1};
emit(key, value);
} var reduce= function (key, values){ var cnt =
{count:0}; values.forEach(function(val){
cnt.count += parseInt(val.count);
});
return cnt;
}
```

```
{
  "_id" : "Allegheny", "value" : { "count" : 1 } }
{ "_id" : "Allison Park", "value" : { "count" : 81 } }
{ "_id" : "Ambridge", "value" : { "count" : 36 } }
{ "_id" : "Amherst", "value" : { "count" : 91 } }
{ "_id" : "Anjou", "value" : { "count" : 14 } }
{ "_id" : "Ansnorveldt", "value" : { "count" : 1 } }
{ "_id" : "Anthem", "value" : { "count" : 87 } }

Type "it" for more
[> edit map
[> map
function () {
  var key = this.city;
  var value = {count:1};
  emit(key, value);
}
[> reduce
function (key, values) {
  var cnt = {count:0};
  values.forEach(function(val) {
    cnt.count += parseInt(val.count);
  });
  return cnt;
}
]> ]
```

## 5. Execute MapReduce Job:

```
db.CityRestaurants.mapReduce(map, reduce, {out:"restCity2"});
```

## 6. Get results:

```
db.restCity2.find().pretty()
```

...nt/FinalResult1/	...sbin — mongod ...	...bin — mongo	~ — -bash	+
<pre>} [&gt; db.restCity2.find().pretty() { "_id" : "", "value" : { "count" : 1 } } { "_id" : "110 Las Vegas", "value" : { "count" : 1 } } { "_id" : "AGINCOURT", "value" : { "count" : 1 } } { "_id" : "Aberdour", "value" : { "count" : 1 } } { "_id" : "Aberlady", "value" : { "count" : 2 } } { "_id" : "Ahwahtukee", "value" : { "count" : 1 } } { "_id" : "Ahwatukee", "value" : { "count" : 16 } } { "_id" : "Ahwatukee Foothills Village", "value" : { "count" : 1 } } { "_id" : "Aichwald", "value" : { "count" : 2 } } { "_id" : "Ajax", "value" : { "count" : 252 } } { "_id" : "Alburg", "value" : { "count" : 1 } } { "_id" : "Alburgh", "value" : { "count" : 1 } } { "_id" : "Alex", "value" : { "count" : 1 } } { "_id" : "Allegheny", "value" : { "count" : 1 } } { "_id" : "Allison Park", "value" : { "count" : 81 } } { "_id" : "Ambridge", "value" : { "count" : 36 } } { "_id" : "Amherst", "value" : { "count" : 91 } } { "_id" : "Anjou", "value" : { "count" : 14 } } { "_id" : "Ansnorveldt", "value" : { "count" : 1 } } { "_id" : "Anthem", "value" : { "count" : 87 } }  Type "it" for more [&gt; ]</pre>				

## APACHE PIG ANALYSIS

**Analysis # 02: To implement Replicated join to show which user has posted what reviews against the business**

Performing replicated join since the reviews dataset is very large, between reviews and users.

**1. LOAD the review dataset from HDFS Using CSVExcelStorage and skip the header**

```
reviews= LOAD
'hdbs://localhost:9000/FinalProject/Dataset/yelp_review.csv' USING
org.apache.pig.piggybank.storage.CSVExcelStorage(',', 'NO_MULTILINE',
'UNIX', 'SKIP_INPUT_HEADER');
```

**2. LOAD the User dataset from HDFS Using CSVExcelStorage and skip the header**

```
LOAD='hdbs://localhost:9000/FinalProject/Dataset/yel
p_user.csv' USING
org.apache.pig.piggybank.storage.CSVExcelStorage(',',
'NO_MULTILINE','UNIX',
'SKIP_INPUT_HEADER');
describe users;
```

**3. Filtering the reviews & users file**

```
fltrvw = FOREACH reviews GENERATE (chararray) $0 as reviewId, (chararray) $1 as userId,
(chararray) $2 as businessID, (chararray) $3 as rating, (chararray) $4 as date, (chararray) $5 as
userReview;
```

```
describe fltrvw;
```

```
fltusr = FOREACH users GENERATE (chararray) $0 as userId,(chararray) $1 as userName;
```

```
describe fltusr;
```

**4. Performing Join operations:**

```
joined = Join fltrvw by userId, fltusr by userId USING 'replicated';
```

```
lmtjnd = LIMIT joined 5;
```

**5. Storing the output file to HDFS**

```
Store lmtjnd into 'hdbs://localhost:9000/FinalProject/pig_join' Using PIGSTORAGE (',');
```

```

part-r-00000
IXv0zsEMYtiJI0CARmj770,bv2nCi5Qv5vroFiqKGopiw,ACFtxLv8pGrrxMm6EgjreA,4,2016-05-28,Love
coming here. Yes the place always needs the floor swept but when you give out peanuts in
the shell how won't it always be a bit dirty. ,bv2nCi5Qv5vroFiqKGopiw,Tim
L_9BTb55X0GDTThi6GlZ6w,bv2nCi5Qv5vroFiqKGopiw,s2I_Ni76bjJNK9yG60iD-Q,4,2016-05-28,Had
their chocolate almond croissant and it was amazing! So light and buttery and oh my how
chocolaty.,bv2nCi5Qv5vroFiqKGopiw,Tim
MV3CcKScW05u5LVfF6ok0g,bv2nCi5Qv5vroFiqKGopiw,CKC0-M0WMqoeWf6s-szl8g,5,2016-05-28,Lester's
is located in a beautiful neighborhood and has been there since 1951. They are known for
smoked meat which most deli's have but their brisket sandwich is what I come to montreal
for. They've got about 12 seats outside to go along with the
inside. ,bv2nCi5Qv5vroFiqKGopiw,Tim
n6QzIUObkYshz4dz2QRJTw,bv2nCi5Qv5vroFiqKGopiw,VR6GpWIda3SfvPC-lg9H3w,5,2016-05-28,Small
unassuming place that changes their menu every so often. Cool decor and vibe inside their
30 seat restaurant. Call for a reservation. ,bv2nCi5Qv5vroFiqKGopiw,Tim
vkVSCC7xljjrAI4UGfnKEQ,bv2nCi5Qv5vroFiqKGopiw,AEx2SYEUJmTxVVBlLlCwA,5,2016-05-28,Super
simple place but amazing nonetheless. It's been around since the 30's and they still serve
the same thing they started with: a bologna and salami sandwich with
mustard. ,bv2nCi5Qv5vroFiqKGopiw,Tim

```

### Analysis # 03: To find the business details that have been reviewed less than 100 times

#### 1. LOAD the review dataset from HDFS

```
reviews = LOAD 'hdfs://localhost:9000/FinalProject/Dataset/yelp_review.csv' USING
org.apache.pig.piggybank.storage.CSVExcelStorage(',,', 'NO_MULTILINE',
'SKIP_INPUT_HEADER');
```

#### 2. Filtering the review dataset

```
fltrvw = FOREACH reviews GENERATE (chararray) $0 as reviewId, (chararray) $1 as userId, (chararray)
$2 as businessID, (chararray) $3 as rating, (chararray) $4 as date, (chararray) $5 as userReview;
```

#### 2. LOAD the buiness dataset from HDFS

```
business = LOAD 'hdfs://localhost:9000/FinalProject/Dataset/yelp_business.csv' USING
org.apache.pig.piggybank.storage.CSVExcelStorage(',,', 'NO_MULTILINE',
'SKIP_INPUT_HEADER');
```

#### 4. Filtering the business dataset from HDFS

```
fltbus = FOREACH reviews GENERATE (chararray) $0 as businessId, (chararray) $1 as businessName,
(chararray) $4 as city;
```

#### 5. Grouping the filtered review by businessID

```
groupedByBussId = group fltrvw by businessID;
```

```
counted = FOREACH groupedByBussId GENERATE group as businessID,
```

COUNT(fltrvw) as cnt;

res = FOREACH counted GENERATE businessID, cnt;

## 6. Filtering the filtered review data

result = FILTER res BY cnt <100;

## 7. Joining the filtered review and filtered business

joined = JOIN result by \$0, fltbus by businessId;

## 8. Storing the output file to HDFS

STORE joined INTO '/Users/poojithamuppalla/Desktop/output.txt' Using PigStorage(',');

```

part-r-00000
-zEpEmDffQL-ph0N3BDlXA,22,-zEpEmDffQL-ph0N3BDlXA,DSisShlBk30MErTfFywpA,2014-01-17
3BCsAgo_1i4xMuTyLkMLRQ,20,3BCsAgo_1i4xMuTyLkMLRQ,tudvRKj-g59Hb9okRpLn4w,2013-01-12
3jKUbhGSIjTV5jZ0wnW0xA,39,3jKUbhGSIjTV5jZ0wnW0xA,fgd07pIGKyZ9G0RaWLmpq,2010-12-16
679eSYC15Sc17TN9Dj8sg,28,679eSYC15Sc17TN9Dj8sg,t2idZ0ntAxHdkrLw80hdCg,2013-03-06
6wc5DPXGudcNi2XK70f0ta,29,6wc5DPXGudcNi2XK70f0ta,smlzHF2g1hnyJ0K_QA9_Q,2009-02-25
A4NuAm61e1aY0tF0c,rb00,16,A4NuAm61e1aY0tF0c,rb00,14o2zXvNCzbH1_gsNKVTNA,2012-10-06
ASq9gG8x4IaaYnePCBa2Kg,16,ASq9gG8x4IaaYnePCBa2Kg,N40Yc-GnE0RWbXqKna_tug,2010-10-11
EFPrQuqF0qMit_iCpVa-6A,20,EFPrQuqF0qMit_iCpVa-6A,xXTdJXPGoP-NUYctc0wBg,2010-07-15
Ec9CBmL3285XkeHaNp-BSQ,107,Ec9CBmL3285XkeHaNp-BSQ,6MTc132D_jVVZGUSoDctUw,2014-03-02
FhgAHo-8--equMw85UZ410,36,FhgAHo-8--equMw85UZ410,qgbRvkZ8tUiAtNAM34YBzg,2010-10-26
Nm0Bomyz63TP9NN2q4Z1rg,41,Nm0Bomyz63TP9NN2q4Z1rg,i24a0vY6bcOC-pJEvepkaw,2010-09-08
QTSCFDpcuROEBUCvGS8F1w,15,QTSCFDpcuROEBUCvGS8F1w,6kyGowEk1AvUUwHlkJYNg,2010-11-19
TrGBHqHVzJM5tk2C1D1ct0,31,TrGBHqHVzJM5tk2C1D1ct0,oog9E5RS1XR93MxpizVhsQ,2014-07-28
UYh1N1x0hOh-a7nX92xFzQ,105,UYh1N1x0hOh-a7nX92xFzQ,QrPSD3WAXun5zHyms5BzYw,2010-10-16
Xd6vHhvFYghzth0ou2FtkQ,16,Xd6vHhvFYghzth0ou2FtkQ,-mZckm02bVLeNcantp3Rg,2014-09-01
c11hu4ntD1UV5Gewz_esng,18,c11hu4ntD1UV5Gewz_esng,TMhtsBPE-L57Hoghsq9Mw,2011-04-09
fZM_o3KKZ9mR-1pvBeow8A,26,fZM_o3KKZ9mR-1pvBeow8A,AecE6t97JJRjuRVqFJbkw,2010-10-13
gMUAn6xcuE-TbY1seFw_Ww,25,gMUAn6xcuE-TbY1seFw_Ww,Iv161_sMHxd3AqnvC88uqa,2014-07-31
hPclwunHi36YY8cPgHWJBg,37,hPclwunHi36YY8cPgHWJBg,ue4-00owOUua1gY61dui3A,2010-09-15
K1ddGAcNIpE_aB5sd1hNxw,45,K1ddGAcNIpE_aB5sd1hNxw,d5cPdp291f0snJkkONGwmQ,2011-03-07
liGn7sP0vUuy3kcwbhRkbA,18,liGn7sP0vUuy3kcwbhRkbA,fgd07pIGKyZ9G0RaWLmpq,2010-10-19
o2Qh45iGYJ7BK4hP7dfkrw,19,o2Qh45iGYJ7BK4hP7dfkrw,tI3hHapsH3bz67fo_UPow,2014-07-11
w6frk94BwGwxenre_zeXUg,15,w6frk94BwGwxenre_zeXUg,XocHLif4ubin1pkzSgkGrw,2011-03-22
xyyzmAnZp2snpBklfcr3Sw,27,xyyzmAnZp2snpBklfcr3Sw,r4giBBijFw7MuXp0w0iH3g,2014-10-22

```

## APACHE HIVE ANALYSIS

**Analysis # 04 :** To find Top 3 business based on rating for each city.

### 1. Create Table 'business\_all'

Create table business\_all(business\_id String, name String, neighborhood String, address String, city String, state String, postal\_code String, latitude String, longitude String, stars Float, review\_count int, is\_open BOOLEAN, categories String);

### 2. Load csv Data

Load Data inpath

'hdfs://localhost:9820/FinalProject/Dataset/yelp\_business.csv' into table business\_all;

### 3. Create business Table

Create Table business as Select business\_id, name, city, state, Cast(stars as Double) As Ratings from business\_all;

#### 4 . Top 3 business based on rating for each city:

Select \* from( Select business\_id, name, city, state, ratings, rank() over (partition by city order by ratings desc) as rank from business) b where rank <3 limit 10;

#### 5 . Inserting the Hive Output to HDFS:

Insert Overwrite Directory 'hdfs://localhost:9000/FinalProject/Hive/Analysis3/' Select \* from( Select business\_id, name, city, state, ratings, rank() over (partition by city order by ratings desc) as rank from business) b where rank <3;

#### 1. Create Table 'business\_all'

```
ellar/hive/3.1.1/libexec/lib/hive-cli-3.1.1.jar org.apache.hadoop.hive.cli.CliDriver /usr/local/bin/hadoop-3/hadoop-3.1.2/sbin -- bash | /usr/local/bin/hadoop-3/hadoop-3.1.2/sbin -- bash ... +  
hive> Create table business_all(business_id String, name String, neighborhood String, address String, city String, state String, postal_code String, latitude String, longitude String, stars String, review_count String, is_open String, categories String) Row Format Delimited Fields Terminated by "\t";  
OK  
Time taken: 0.109 seconds  
hive> Load Data Inpath 'hdfs://localhost:9000/FinalProject/Dataset/yelp_business_tsv.tsv' into table business_all;  
Loading data to table default.business_all  
OK  
Time taken: 0.169 seconds  
hive> describe formatted business_all;  
OK  
# col_name          data_type            comment  
business_id          string  
name                string  
neighborhood        string  
address              string  
city                string  
state               string  
postal_code         string  
latitude             string  
longitude            string  
stars               string  
review_count         string  
is_open              string  
categories           string  
  
# Detailed Table Information  
Database:          default  
Owner:              user  
Owner:              saarthakgopal  
CreateTime:         Tue Aug 13 14:04:82 EDT 2019  
LastUpdate:         Unknown  
Retention:          0  
Location:          hdfs://localhost:9000/user/hive/warehouse/business_all  
Table Type:         MANAGED_TABLE  
Table Parameters:  
  bucketing_version    2  
  numFiles             2  
  numRows              0  
  rawDataSize         0  
  totalSize            5988748  
  transient_lastDdlTime 15665719462  
  
# Storage Information  
SerDe Library:     org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe  
InputFormat:        org.apache.hadoop.mapred.TextInputFormat  
OutputFormat:       org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat  
Compressed:         No  
Num Buckets:       -1  
Bucket Columns:    []  
Sort Columns:       []  
Storage Desc Params:  
  field.delim          \t  
  serialization.format \t  
Time taken: 0.065 seconds, Fetched: 43 row(s)  
hive>   

```

#### 2. Load tsv Data into business\_all table that you just created

```
sbin - java -Dproc_jar -Java.net.preferIPv4Stack=true -Dproc_hivecli -Dlog4j.configurationFile=hive-log4j2.properties -Djava.util.logging.config.file=/usr/local/Cellar/hive/3.1.1/libexec/conf/parquet-logging.properties -D...  
.../hive/3.1.1/libexec/lib/hive-cli-3.1.1.jar org.apache.hadoop.hive.cli.CliDriver /usr/local/bin/hadoop-3/hadoop-3.1.2/sbin -- bash ... | /usr/local/bin/hadoop-3/hadoop-3.1.2/sbin -- bash ... +  
SerDe Library:     org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe  
InputFormat:        org.apache.hadoop.mapred.TextInputFormat  
OutputFormat:       org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat  
Compressed:         No  
Num Buckets:       -1  
Bucket Columns:    []  
Sort Columns:       []  
Storage Desc Params:  
  field.delim          \t  
  serialization.format \t  
Time taken: 0.065 seconds, Fetched: 43 row(s)  
hive> Select * from business_all limit 5;  
OK  
business_id      name      neighborhood   address   city      state      postal_code   latitude   longitude   stars   review_count   is_open   categories  
FYWN1meve18bWNgQjZ0Ng   **"Dental by Design"**   ***4855 E Warner Rd, Ste B9***   Ahwatukee   AZ      85044   33.336902   -111.978592   4      22      1      Dentists;General Dentist  
stry;Health & Medical;Oral Surgeons;Cosmetic Dentists;Orthodontists  
He-G7wVjzVUys1KrnFnbPUQ   **"Stephen Szabo Salon"**   ***3101 Washington Rd***   McMurray   PA      15317   40.2916853   -80.1048999   3      11      1      Hair Stylists;  
Hair Salons;Men's Hair Salons;Blow Dry/Out Services;Hair Extensions;Beauty & Spas  
KGPPW8lFFfly5B72MxISZ0A   **"Western Motor Vehicle"**   ***6025 N 27th Ave, Ste 1***   Phoenix AZ    85017   33.5249025   -112.1153098   1.5     18      1      Departments of Motor Vehicles;Public Services & Government  
8DSHNS-LufQpEWip0hXijA   **"Sports Authority"**   ***5000 Arizona Mills Cr, Ste 435***   Tempe   AZ      85282   33.3831468   -111.9647254   3      9      0      Sporting Goods;Shopping  
g  
Time taken: 0.093 seconds, Fetched: 5 row(s)  
hive>   

```

### 3. Create business Table

```
ellar/hive/3.1.1/libexec/lib/hive-cll-3.1.1.jar org.apache.hadoop.hive.cli.CliDriver /usr/local/bin/hadoop-3/hadoop-3.1.2/sbin -- bash ... /usr/local/bin/hadoop-3/hadoop-3.1.2/sbin -- bash ... +
```

Table Parameters:

- bucketing\_version 2
- numFiles 1
- numPartitions 0
- rowDataSize 0
- totalSize 5988740
- transient\_lastDdlTime 1565719462

# Storage Information

SerDe Library: org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe

InputFormat: org.apache.hadoop.mapred.TextInputFormat

OutputFormat: org.apache.hadoop.hive.io.HiveIgnoreKeyTextOutputFormat

Compressed: No

Num Buckets: 1

Bucket Columns: []

Sort Columns: []

Storage Desc Params:

- field.delim \t
- serialization.format \t

Time taken: 0.865 seconds, Fetched: 43 row(s)

hive> Select \* from business\_all limit 5;

OK

business_id	name	neighborhood	address	city	state	postal_code	latitude	longitude	stars	review_count	is_open	categories
FVNWnwv18BWhNgQd0g	**"Design by Design"	"North Olmsted,"	"#4855 E Warner Rd, Ste B9"	Ahwatukee	AZ	85094	33.3306992	-111.9785992	4	22	1	Dentists;General Dentistry;H
He-07VwVjzVuviKrTNPUQ	**"Stephen Szabo Salon"		"3101 Washington Rd"	McMurray	PA	15317	40.2916853	-80.1048999	3	11	1	Hair Stylists;Hair S
alonsMen's Hair Salons;Blow Dry/Beauty Services;Hair Extensions;Beauty & Spas			"6025 N 27th Ave, Ste 1"	Phoenix AZ	85017	33.5249825	-112.1153098	1.5	18	1	Departments of Motor Vehicle	
KOPWlfifjyS8T2MxJzCQ	**"Western Motor Vehicle"		"5000 Arizona Mills Cr, Ste 435"	Tempe AZ	85282	33.3831468	-111.9647254	3	9	0	Sporting Goods;Shopping	
BDSHMS-LufppEWInpHkjA	**"Sports Authority"		"5000 Arizona Mills Cr, Ste 435"									

Time taken: 0.093 seconds, Fetched: 5 rows(s)

hive> Create Table business as Select business\_id, name, city, state, Cast(stars as int) As Ratings from business\_all;

Query ID: 2019-08-13\_14-06-39\_799

Time taken: 0.093 seconds, Fetched: 5 rows(s)

hive> Total jobs = 3

Launching Job 1 out of 3

Number of reduce tasks is set to 0 since there's no reduce operator

Starting Job = job\_1565704711733\_0025, Tracking URL = http://sarthsaks-mbp.ent.core.medtronic.com:8088/proxy/application\_1565704711733\_0025/Kill Command = /usr/local/bin/hadoop-3/hadoop-3.1.2/bin/mapred job -kill job\_1565704711733\_0024

Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0

2019-08-13 14:06:39,799 Stage-1 map = 0%, reduce = 0%

2019-08-13 14:06:40,799 Stage-1 map = 100%, reduce = 0%

Ended Job = job\_1565704711733\_0024

Stage-4 is selected by condition resolver.

Stage-3 is filtered out by condition resolver.

Stage-2 is filtered out by condition resolver.

Moving data to directory hdfs://localhost:9000/user/hive/warehouse/hive\_staging\_hive\_2019-08-13\_14-06-33\_235\_167652496237927768-1/-ext-10002

Moving data to directory hdfs://localhost:9000/user/hive/warehouse/business

MapReduce Jobs Launched:

- Stage-Stage-1: Map: 1

Stage-Stage-1: Map: 1 HDFS Read: 5915140 HDFS Write: 2061669 SUCCESS

Total MapReduce CPU Time Spent: 0 msec

OK

Time taken: 14.066 seconds

hive> show tables;

OK

business	business_all	invites	pokes	students
x	x	x	x	x

Time taken: 0.045 seconds, Fetched: 6 row(s)

hive>

### 4. Top 3 business based on rating for each city using rank() and Partition by clause

```
ellar/hive/3.1.1/libexec/lib/hive-cll-3.1.1.jar org.apache.hadoop.hive.cli.CliDriver /usr/local/bin/hadoop-3/hadoop-3.1.2/sbin -- bash ... /usr/local/bin/hadoop-3/hadoop-3.1.2/sbin -- bash ... +
```

Total jobs = 3

Launching Job 1 out of 3

Number of reduce tasks is set to 0 since there's no reduce operator

Starting Job = job\_1565704711733\_0025, Tracking URL = http://sarthsaks-mbp.ent.core.medtronic.com:8088/proxy/application\_1565704711733\_0025/Kill Command = /usr/local/bin/hadoop-3/hadoop-3.1.2/bin/mapred job -kill job\_1565704711733\_0024

Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0

2019-08-13 14:09:18,663 Stage-1 map = 0%, reduce = 0%

2019-08-13 14:09:24,832 Stage-1 map = 100%, reduce = 0%

Ended Job = job\_1565704711733\_0024

Stage-3 is selected by condition resolver.

Stage-3 is filtered out by condition resolver.

Stage-2 is filtered out by condition resolver.

Moving data to directory hdfs://localhost:9000/user/hive/warehouse/hive\_staging\_hive\_2019-08-13\_14-09-10\_692\_7418816964744245002-1/-ext-10002

Moving data to directory hdfs://localhost:9000/user/hive/warehouse/business

MapReduce Jobs Launched:

- Stage-Stage-1: Map: 1

Stage-Stage-1: Map: 1 HDFS Read: 5915157 HDFS Write: 2127333 SUCCESS

Total MapReduce CPU Time Spent: 0 msec

OK

Time taken: 15.348 seconds

hive> show tables;

OK

business	business_all	invites	pokes	students
x	x	x	x	x

Time taken: 0.040 seconds, Fetched: 6 row(s)

hive> Select \* from (Select business\_id, name, city, state, ratings, rank() over (partition by city order by ratings desc) as rank from business) b where rank <3 limit 10;

Query ID: 2019-08-13\_14-10-20\_682

Total jobs = 1

Launching Job 1 out of 1.

Number of reduce tasks not specified. Estimated from input data size: 1

In order to change the average load for a reducer (in bytes):

- set hive.exec.reducers.bytes.per.reducer<number>
- In order to set the 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\_1565704711733\_0026, Tracking URL = http://sarthsaks-mbp.ent.core.medtronic.com:8088/proxy/application\_1565704711733\_0026/Kill Command = /usr/local/bin/hadoop-3/hadoop-3.1.2/bin/mapred job -kill job\_1565704711733\_0026

Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1

2019-08-13 14:10:07,424 Stage-1 map = 0%, reduce = 0%

2019-08-13 14:10:15,424 Stage-1 map = 100%, reduce = 0%

2019-08-13 14:10:20,682 Stage-1 map = 100%, reduce = 100%

Ended Job = job\_1565704711733\_0026

MapReduce Jobs Launched:

- Stage-Stage-1: Map: 1 Reduce: 1 HDFS Read: 2412153 HDFS Write: 862 SUCCESS

Total MapReduce CPU Time Spent: 0 msec

OK

name	city	state	ratings	rank()
"Cookies by Design"	"North Olmsted,"	OH	5.0	1
"McDonald's"	"AGINCOURT"	ON	2.5	
"Kathy's Alterations"	"Ahwatukee"	AZ	5.0	1
"Dental by Design"	"Ahwatukee"	AZ	4.0	2
"AHA! Tailor & Alterations"	"Ahwatukee"	AZ	4.0	2
"Desert Foothills Trailhead"	"Ahwatukee Foothills Village"	AZ	5.0	1
"Country Cheese"	Ajax	ON	5.0	1
"Vape Assassins"	Ajax	ON	5.0	1
"Ransom Bar Inn B & B"	Aisbury	VT	5.0	1

Time taken: 20.066 seconds, Fetched: 10 row(s)

hive>

## 5. Inserting the Hive Output to HDFS using Insert Overwrite Directory command:

```

...apache.hadoop.hive.cli.CliDriver | ...3/hadoop-3.1.2/sbin -- bash ... | ...adoop-3.1.2/sbin -- bash ... | +
hive> Insert Overwrite Directory 'hdfs://localhost:9000/FinalProject/Hive/Analysis3/' Select *
from ( Select business_id, name, city, state, ratings, rank() over (partition by city order by
ratings desc) as rank from business ) b where rank <3;
Query ID = sarthakgoel_20190813161717_3c1fb46f-697a-4cdc-96db-8a91e3d11117
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
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>

[roxy/application_1565726254196_003]
Kill Command = /usr/local/bin/hadoop-3/hadoop-3.1.2/bin/mapred job -kill job_1565726254196_0003
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2019-08-13 16:17:26,505 Stage-1 map = 0%, reduce = 0%
2019-08-13 16:17:32,714 Stage-1 map = 100%, reduce = 0%
2019-08-13 16:17:39,903 Stage-1 map = 100%, reduce = 100%
Ended Job = job_1565726254196_0003
Moving data to directory hdfs://localhost:9000/FinalProject/Hive/Analysis3
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 HDFS Read: 2140582 HDFS Write: 403328 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
Time taken: 23.543 seconds
hive> ■

```

```

000000_0
bAqt_TW-f42GWCg2h0_80""Cookies by Design""North Olmsted,"OH05.01
LixFCMKGdpIw8WrsjAl5c0""McDonald's""AGINCOURT2.51
qdCwzhJ5Yo_Sdm_bYDf00""Kathy's Alterations""AhwatukeeAZ5.01
FYN1wnev18bWNq01J2GNq""Dewy by Design""AhwatukeeAZ4.02
#NAME?""My Wine Cellar""AhwatukeeAZ4.02
qRUEfIk4qzmb8TfU6PuN_w""Desert Foothills Headache""Ahwatukee Foothills VillageAZ5.01
NZBWywxk1Ptqg3G8gtvReEa""Country Cheese""AjaxON5.01
1RItXbaPII-4XOLJh0XSMO""Anh's Tailor & Alterations""AjaxON5.01
0qmfv3-yi48cEWjX-y1h4o""Vape Awesome""AjaxON5.01
tJRDl15vgpZwehenzE2cSo""Ransom Bay Inn B & B""AlburgVT5.01
_WbkQha9xnfT2nYmHAAqo""Island Tree Service""AlburghVT3.51
UIRSllusSmvTzpulunCEgqv""West End Overlook""AlleghenyPA5.01
rmcIZnP31ztqjikKWL.Gpo""Yeager Ed Auto Body""Allison ParkPA5.01
CrDHPkNGvHuGLAMvNKFBZA""Go Ape""Allison ParkPA5.01
ftx72aqAdrmYm01MytdtCo""Maid On The Spot""AmbridgePA5.01
E0mRHPyvR8Ba5R21zu08c0""Cafe 501""AmbridgePA4.52
729grSelmsnhiv7D5u0Xg""Pizza House""AmbridgePA4.52
GIWWKf0Y9129.CHNw0120""The Stuffed Pepper""AmbridgePA4.52
ASmJVV8M-T9Av4R6tV95KA""AutoNation Ford Amherst""AmherstOH5.01
ps2Gr56f03Dh5XKyZtXPo""Sundays Plus Delicatessen""AmherstOH5.01
vn11Te7vlomDV1ZimvFcT0""Zibo Anjou""AnjouQC3.51
1RaqcUnY19KK4jeznx1860""Baton Rouge""AnjouQC3.51
mMTJXj-9E2f448559995A""Costco Anjou""AnjouQC3.51
ffNehzZmf-yu70F130At7A""Holland Marsh Suptest""Ansngryeldton4.51
TbdVadxs5S4PfZq74Y8IA""Andrew Z""AnthemAZ5.01
Ke_ydwvg9LTMB1jNjMfd""Douglas and Tina Kellock - Certified iCracked Technician""AnthemAZ5.01
J9kvstleo3IVGr73yZekqA""Deer Valley Credit Union""AnthemAZ5.01
fULZVaEt3akFZ1Mw-cc0k0""Rockman Pool Service""AnthemAZ5.01
mxp1ZTFnnnT0V3D9qclKw""Auntie Anne's""AnthemAZ5.01
Ca5Llzxv0Xgiu8MEhXJA""Bodytech Health & Fitness""AspinwallPA4.01
u5Wuzp8uC79iuC9xpq""Casa Del Sole Pizza""AspinwallPA3.52
6FCohdH30SX5_M8m0VJye1la""House O' Hockey""AspinwallPA3.52
1s3U0Z74sNch_ijzr9CUP0""Homegoods""AuroraOH5.01
2KbMYZcbBb206NyQFAXVm0""5th Avenue Beauty Studio""AuroraON5.01
D0arSE104c84m0lnBnirtA""That Stereo Shop""AvalonPA5.01
ZYU-IZBvXn9MZWGn91AoIA""Golf Zone""AvonOH5.01
cwkX37aiw4B7KTUJZ5FjIA""Tan With Care""AvonOH5.01
v15V11R0D5w5CaqFWM61_A""Sylvester Truck & Tire Service""AvonOH5.01
xxopKQq7nuUKAk022_d0A""Avon Animal & Bird Hospital""AvonOH5.01
a2Liav11Zzv8vPhB0M-1lo""Discount Tire Store - Avon, OH""AvonOH5.01
Y2av1Zfw7Ms0vtY17iz_bg""CLE Dog Training""AvonOH5.01
Cs5iUqF_dWKf3pCY40tQo""Prayers from Maria - Field of Hope""AvonOH5.01
gsBmndWvuG1K30Tm4uIJIA""Pet People""AvonOH5.01
WCfMZKbzEADT32157TPoWe""Humble Pie""Avon LakeOH5.01
uEBv968sBjFTkdqWcDsko""Avon lake Eye Care""Avon LakeOH5.01
14315RKkjysven200UmOVCo""Floor Coverings International of Cleveland West""Avon LakeOH5.01
j1dUpfIA4VOBMy_LaaSZY""You Buy and We Fly""AvondaleAZ5.01
d1sT0XPYzYs2v10WStDNAA""9th World Vapor""AvondaleAZ5.01
swLkX/nBnsjs5ESVfT3KV-W""G & H Air Conditioning and Heating""AvondaleAZ5.01
H8tTQz1f5459719P9Kvr-o""Cien Motor Werks""AvondaleAZ5.01
jIA0sxd3mLV-cPbnMp2t0""Avondale Recycling Center""AvondaleAZ5.01
NCNGUdWwqTRp4iV2cbjNow""Modern Revival Events""AvondaleAZ5.01

```

**Analysis # 05: To find list of businesses that have above average rating in each state.**

## 1. Implementing inner join in HQL :

Select a.business\_id, a.name, a.city, a.state, a.ratings from business a inner join (Select avg(ratings) as Average, state from business group by state) b where a.state = b.state AND a.ratings > b.Average;

## 2. Inserting the Hive Output to HDFS :

```
Insert Overwrite Directory 'hdfs://localhost:9000/FinalProject/Hive/Analysis4/' Select  
a.business_id, a.name, a.city, a.state, a.ratings from business a inner join (Select avg(ratings) as Average,  
state from business group by state) b where a.state = b.state AND a.ratings > b.Average;
```

### **3. Check the file in HDFS Directory :**

```
hadoop fs -lsr /FinalProject/Hive/Analysis4/
```

```
poojithas-mbp:sbin poojithamuppalla$ hadoop fs -cat /FinalProject/Hive/Analysis4/000000_0
```

**1. Implementing inner joins in HQL on business and average rating values of business grouped by each state**

## **2. Inserting the Hive Output to HDFS using the Insert Overwrite:**

```
hive> Insert Overwrite Directory 'hdfs://localhost:9000/FinalProject/Hive/Analysis4/' S
elect a.business_id, a.name, a.city, a.state, a.ratings from business a inner join (S
elect avg(ratings) as Average, state from business group by state) b where a.state = b.
state AND a.ratings > b.Average;
```

## MAPREDUCE ANALYSIS:

**Analysis # 06:** To implement the partitioning on the basis of year the user joined the yelp.

#### Partitioning Pattern of Data Organization Technique Used

Partitioning is done on the basis of year. Year is extracted from the Date column in user dataset.

In partitioning technique, the only thing is we need to know how many partitions will be there.

## **Mapper Class:**

```
public class Mapper1 extends Mapper<Object, Text, IntWritable, Text> {
```

```
private final static SimpleDateFormat frmt = new  
SimpleDateFormat("yyyy-MM-DD");
```

```

private IntWritable outkey = new IntWritable();

@Override
    protected void map(Object key, Text value, Context context) throws
IOException, InterruptedException {

    String input[] = value.toString().split(",", -1);
        if(input[3].length() == 10) {
        String strDate = input[3];
        Calendar cal = Calendar.getInstance();

        try {
            cal.setTime(frmt.parse(strDate));
        } catch (ParseException e) {

        }

        outkey.set(cal.get(Calendar.YEAR));      context.write(outkey, value);
    }
}
}

```

**Custom Practitioner Class:**

```

public class Partitioner1 extends Partitioner<IntWritable, Text> implements Configurable {

    private static final String MIN_LAST_ACCESS_DATE_YEAR =
    "min.last.access.date.year";

    private Configuration conf = null;
    private int minLastAccessDateYear = 0;

    public int getPartition(IntWritable key, Text value, int numPartitions) {
        return key.get() - minLastAccessDateYear;
    }

    public Configuration getConf() {
        return
    conf;
    }

    public void setConf(Configuration conf) {
        this.conf = conf;
        minLastAccessDateYear = conf.getInt(MIN_LAST_ACCESS_DATE_YEAR,
0);
    }

    public static void setMinLastAccessDate(Job job, int minLastAccessDateYear) {
        job.getConfiguration().setInt(MIN_LAST_ACCESS_DATE_YEAR,
minLastAccessDateYear);
    }
}

```

```

        }
    }
}
```

### Reducer Class

```

public class Reducer1 extends Reducer<IntWritable, Text, Text,
NullWritable> {

    protected void reduce(IntWritable key, Iterable<Text> values, Context context) throws
IOException, InterruptedException {
        for (Text t : values) {
            context.write(t, NullWritable.get());
        }
    }
}
```

### Driver Class

```

public static void main(String[] args) throws IOException,
ClassNotFoundException, InterruptedException {

    Configuration conf = new Configuration();           Job job =
new Job(conf, "DataOrg");
    job.setJarByClass(DriverClass.class);

    job.setPartitionerClass(Partitioner1.class);
Partitioner1.setMinLastAccessDate(job, 2004);

    job.setNumReduceTasks(14);

    FileInputFormat.addInputPath(job, new Path(args[0]));
FileOutputFormat.setOutputPath(job, new Path(args[1]));

    job.setMapOutputKeyClass(IntWritable.class);      job.setMapOutputValueClass(Text.class);

    job.setMapperClass(Mapper1.class);      job.setReducerClass(Reducer1.class);

    job.setOutputKeyClass(Text.class);      job.setOutputValueClass(NullWritable.class);

    System.exit(job.waitForCompletion(true)? 0:1);
}
}
```

```

bin -- bash -- 94x29
...3/hadoop-3.1.2/sbin -- bash ... | ...se/1.3.4/libexec/bin -- bash | ...adoop-3.1.2/sbin -- bash ...
Generic options supported are:
-conf <configuration file> specify an application configuration file
-D <property=value> define a value for a given property
-fs <file://|hdfs://>namenode:port> specify default filesystem URL to use, overrides 'fs.defaultFS' property from configurations.
-jt <local|resourcemanager:port> specify a ResourceManager
-files <file1,...> specify a comma-separated list of files to be copied to the map reduce cluster
-libjars <jar1,...> specify a comma-separated list of jar files to be included in the classpath
-archives <archive1,...> specify a comma-separated list of archives to be unarchived on the compute machines

The general command line syntax is:
command [genericOptions] [commandOptions]

Usage: hadoop fs [generic options] -cat [-ignoreCrc] <src> ...
sarthaks-mbp:bin s ... /FinalProject/Top25Restaurant/FinalResult47/part-r-00000
2019-08-14 17:36:55,527 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
5IX9FOesKrLRyz618xc6tw,Helen,_993,2004-10-12,691,750,615,126,"2005, 2008, 2012, 2011, 2007, 2009, 2006, 2013, 2010",3,99,1 7,19,14,87,5,92,142,248,248,27,62
ngqmB6pYczql84-4y4RYcw,Dave,32,2004-12-15,6,0,0,1,None,3.4,0,0,1,0,0,2,0,0,0,0,0,0
XoMETjjFuH0AhgY131GDIA,Chris,81,2004-10-14,14,2,0,2,None,4.05,0,2,0,0,0,2,1,1,1,0
23J4vG9_xxxdmni8CBX7Ng,Joan,1659,2004-10-12,49143,13232,35318,1222,"2016, 2017, 2011, 2009, 2014, 2015, 2013, 2010, 2007, 2008, 2006, 2012, 2005",4.36,3259,282,269,334,98,1263,4661,4373,4373,1629,2381

```

**Analysis # 07:** MapReduce Chaining to calculate the no. of users signed up on yelp each year

Chaining Operation Performed

MapReduce Job 01: Partition the files on the basis of years – Partitioning Technique

MapReduce Job 02: Count the no of files in a given partition

### Driver Class

```

public class DriverClass {
    public static void main(String[] args) throws IOException,
    ClassNotFoundException, InterruptedException {
        Configuration conf = new Configuration();
        Job job = new Job(conf, "MRchain");
        job.setJarByClass(DriverClass.class);

        job.setPartitionerClass(Partitioner1.class);
        Partitioner1.setMinLastAccessDate(job, 2004);

        job.setNumReduceTasks(14);

        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
    }
}

```

```
job.setMapOutputKeyClass(IntWritable.class);           job.setMapOutputValueClass(Text.class);

job.setMapperClass(Mapper1.class);      job.setReducerClass(Reducer1.class);

job.setOutputKeyClass(Text.class);      job.setOutputValueClass(NullWritable.class);

boolean result = job.waitForCompletion(true);          if(result) {
    Job job1 = Job.getInstance(conf, "chain");
    job1.setJarByClass(DriverClass.class);

    job1.setMapperClass(Mapper2.class);          job1.setCombinerClass(Reducer2.class);
    job1.setReducerClass(Reducer2.class);

    job1.setOutputKeyClass(Text.class);
    job1.setOutputValueClass(IntWritable.class);

    FileInputFormat.addInputPath(job1, new Path(args[1]));
    FileOutputFormat.setOutputPath(job1, new Path(args[2]));

    result = job1.waitForCompletion(true);
}
```

## Mapper01

```
public class Mapper1 extends Mapper<Object, Text, IntWritable, Text> {  
  
    private final static SimpleDateFormat frmt = new  
    SimpleDateFormat("yyyy-MM-DD");  
  
    private IntWritable outkey = new IntWritable();  
  
    @Override  
    protected void map(Object key, Text value, Context context) throws  
    IOException, InterruptedException {  
  
        String input[] = value.toString().split(",", -1);  
        if(input[3].length() == 10) {  
            String strDate = input[3];  
            Calendar cal = Calendar.getInstance();  
            try {  
                cal.setTime(frmt.parse(strDate));  
            } catch (ParseException e) {  
            }  
            outkey.set(cal.get(Calendar.YEAR));  
            context.write(outkey, value);  
        }  
    }  
}
```

```

        }
    }
}
```

**Custom practitioner class**

```

public class Partitioner1 extends Partitioner<IntWritable, Text> implements Configurable {

    private static final String MIN_LAST_ACCESS_DATE_YEAR =
    "min.last.access.date.year";

    private Configuration conf = null;           private          int
    minLastAccessDateYear = 0;

    public int getPartition(IntWritable key, Text value, int numPartitions) {
        return key.get() - minLastAccessDateYear;
    }

    public Configuration getConf() {             return
    conf;
    }

    public void setConf(Configuration conf) {     this.conf = conf;
        minLastAccessDateYear = conf.getInt(MIN_LAST_ACCESS_DATE_YEAR,
    0);
    }

    public static void setMinLastAccessDate(Job job, int minLastAccessDateYear) {
        job.getConfiguration().setInt(MIN_LAST_ACCESS_DATE_YEAR,
    minLastAccessDateYear);
    }
}
```

**Reducer01**

```

public class Reducer1 extends Reducer<IntWritable, Text, Text,
NullWritable> {
```

```

    protected void reduce(IntWritable key, Iterable<Text> values, Context context) throws
    IOException, InterruptedException {
        for (Text t : values) {
            context.write(t, NullWritable.get());
        }
    }
}
```

**Mapper2**

```

public class Mapper2 extends Mapper<Object, Text, Text, IntWritable> {
```

```
    private final static SimpleDateFormat frmt = new SimpleDateFormat("yyyy-MM-DD");
```

```

    private IntWritable outVal = new IntWritable();
    private Text outKey = new Text();

    @Override
        protected void map(Object key, Text value, Context context) throws
IOException, InterruptedException {

        String input[] = value.toString().split(",");
        if(input[3].length() == 10) {
            String strDate = input[3];
            Calendar cal = Calendar.getInstance();

            try {
                cal.setTime(frmr.parse(strDate));
            } catch (ParseException e) {

            }
            outKey.set(String.valueOf(cal.get(Calendar.YEAR)));
            outVal.set(1);

            context.write(outKey, outVal);
        }
    }
}

```

**Reducer2**

```
private IntWritable result = new IntWritable();
```

```

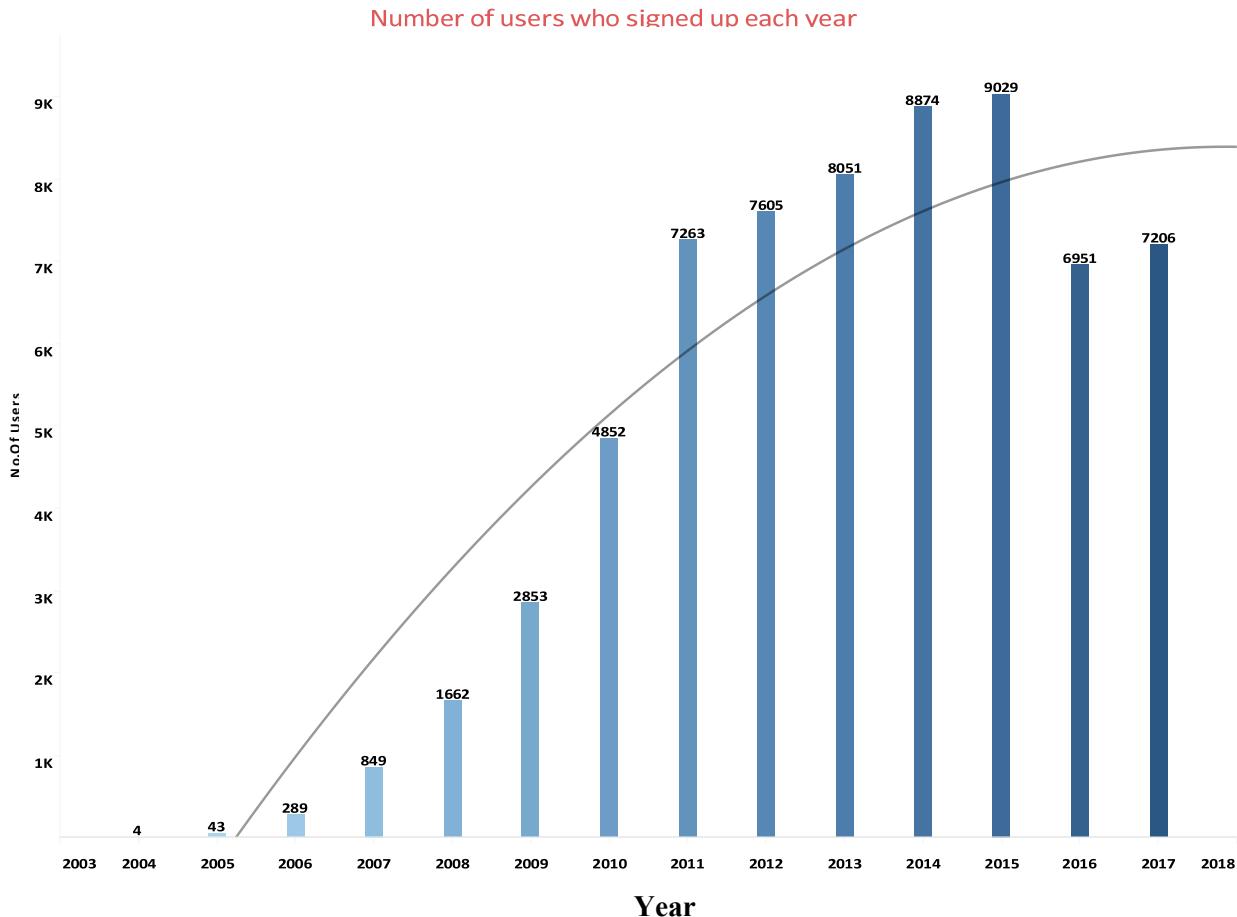
    public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException,
InterruptedException
    {
        int sum = 0;
        for (IntWritable val : values)
        {
            sum += val.get();
        }
        result.set(sum);
    }
}

```

**Command :** hadoop jar /Users/poojithamuppalla/eclipse-workspace/MRchain/target/MRchain-0.0.1-SNAPSHOT.jar com.neu.edu.DriverClass/FinalProject/Dataset/yelp\_user.csv /FinalProject/Top25Restaurant/FinalResult54/ /FinalProject/Top25Restaurant/FinalResult55

```
2019-08-14 21:53:09,750 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
2019-08-14 21:53:10,263 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
2019-08-14 21:53:10,706 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2019-08-14 21:53:10,722 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/sarthakgoel/.staging/job_1565831423867_0007
2019-08-14 21:53:10,889 INFO input.FileInputFormat: Total input files to process : 1
2019-08-14 21:53:10,928 INFO mapreduce.JobSubmitter: number of splits:1
2019-08-14 21:53:11,033 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1565831423867_0007
2019-08-14 21:53:11,035 INFO mapreduce.JobSubmitter: Executing with tokens: []
2019-08-14 21:53:11,168 INFO conf.Configuration: resource-types.xml not found
2019-08-14 21:53:11,168 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2019-08-14 21:53:11,224 INFO impl.YarnClientImpl: Submitted application application_1565831423867_0007
2019-08-14 21:53:11,258 INFO mapreduce.Job: The url to track the job: http://Sarthaks-MacBook-Pro.local:8088/proxy/application_1565831423867_0007/
2019-08-14 21:53:11,259 INFO mapreduce.Job: Running job: job_1565831423867_0007
2019-08-14 21:53:17,361 INFO mapreduce.Job: Job job_1565831423867_0007 running in uber mode : false
2019-08-14 21:53:17,362 INFO mapreduce.Job: map 0% reduce 0%
2019-08-14 21:53:22,427 INFO mapreduce.Job: map 100% reduce 0%
2019-08-14 21:53:27,464 INFO mapreduce.Job: map 100% reduce 7%
2019-08-14 21:53:30,486 INFO mapreduce.Job: map 100% reduce 14%
```

```
ellar/hive/0.11/libexec/lib/hive-clis-3.1.1jar.org.apache.hadoop.hive.cli.ClDriver ... /usr/local/bin/hadoop-3/hadoop-3.2.0/bin --bash /usr/local/bin/hadoop-3/hadoop-3.2.0/bin --bash
2019-08-13 15:09:57.191 WARN Util.NativeCodeLoader: Unable to load native-hadoop library for your platform.. using builtin-Java classes where applicable
019-08-13 15:09:57.191 INFO Util.NativeCodeLoader: Native library is not loaded
2019-08-13 15:09:57.191 INFO Util.NativeCodeLoader: Unable to load native-hadoop library for your platform.. using builtin-Java classes where applicable
YNNMa10ZHaXAr8oZtKngQo**" Dental by Design***Athenaeum Dentistry ***" "ScottsdaleAZ.0
YRSMta0ZHaXAr8oZtKngQo**" Royim Thai Cuisine ***MesaAZ.0
JasMy10ZHaXAr8oZtKngQo**" Koray by JTMChandlerAZ.0
JLUA-0ZHaXAr8oZtKngQo**" Phoenix Glass ***MesaAZ.0
fHnNuUcucbyEccjgpo**" Zeroco Phoenix ***PhoenixAZ.0
JLUA-0ZHaXAr8oZtKngQo**" Phoenix Glass ***MesaAZ.0
YD649WmL0Qq4QdXGTCs8Eig**" Ashlatisu-Thai Massage by Sarah ***PhoenixAZ.0
z7QpLw1QQ4QdXGTCs8Eig**" Salad and Guts ***TollesonAZ.4
z4PexFp1QdXGTCs8Eig**" The Green Room ***MesaAZ.4
z4PexFp1QdXGTCs8Eig**" True REST Fleet Spa ***ScottsdaleAZ.4
z4PexFp1QdXGTCs8Eig**" East Valley Chiropractic ***ScottsdaleAZ.4
z4PexFp1QdXGTCs8Eig**" Acupuncture and Oriental Medicine ***ScottsdaleAZ.0
F9c5cOv1Sv1Hs0kV0jCk6Q**" May's Family Haircutters ***MesaAZ.0
z4PexFp1QdXGTCs8Eig**" El Kiosco ***PhoenixAZ.5
z4PexFp1QdXGTCs8Eig**" West Valley Periodontics ***AvondaleAZ.0
KjyjWnCkx0Mf7QDwB1J9Q**" Fountain Park ***Fountain HillsAZ.5
z4PexFp1QdXGTCs8Eig**" My House ***ChandlerAZ.0
z4PexFp1QdXGTCs8Eig**" Steven Jones - JR Jones Really ***ScottsdaleAZ.0
z4PexFp1QdXGTCs8Eig**" Akishana Sushi & Grill ***PhoenixAZ.0
z4PexFp1QdXGTCs8Eig**" Revive Stainless Steel ***ScottsdaleAZ.0
z4PexFp1QdXGTCs8Eig**" Arcadia Endodontics ***PhoenixAZ.0
z4PexFp1QdXGTCs8Eig**" Shelly's Autobody ***MesaAZ.0
z4PexFp1QdXGTCs8Eig**" Revive Stainless Steel ***ScottsdaleAZ.0
z4PexFp1QdXGTCs8Eig**" Teddy Shonka - North & Co ***PhoenixAZ.0
z4PexFp1QdXGTCs8Eig**" European Master ***PhoenixAZ.0
z4PexFp1QdXGTCs8Eig**" Eco King ***PhoenixAZ.0
Ku7z5Z2v1ZQdy0zNANgQ3Q**" Bindi Chiropractic ***ChandlerAZ.0
z4PexFp1QdXGTCs8Eig**" Bindi Chiropractic ***PhoenixAZ.0
z4PexFp1QdXGTCs8Eig**" Golden Spoke Cyclery ***GlendaleAZ.0
z4PexFp1QdXGTCs8Eig**" Golden Spoke Cyclery ***PhoenixAZ.0
z4PexFp1QdXGTCs8Eig**" Phoenix Christian ***PhoenixAZ.0
z4PexFp1QdXGTCs8Eig**" Phoenix Christian Sun City ***Sun CityAZ.0
z4PexFp1QdXGTCs8Eig**" Phoenix Christian Sun City ***Sun CityAZ.0
z4PexFp1QdXGTCs8Eig**" American Family ***Dale Herzer Agency ***TempeAZ.0
z4PexFp1QdXGTCs8Eig**" American Family ***Dale Herzer Agency ***TempeAZ.0
z4PexFp1QdXGTCs8Eig**" True Home Maintenance Air Conditioning & Heating ***MesaAZ.0
z4PexFp1QdXGTCs8Eig**" True Home Maintenance Air Conditioning & Heating ***MesaAZ.0
z4PexFp1QdXGTCs8Eig**" Pro440RENG - V159pva ***Kirkland ***"OliverAZ.4
z4PexFp1QdXGTCs8Eig**" NAME*** See Cee Cee The Corner ***GlendaleAZ.0
z4PexFp1QdXGTCs8Eig**" NAME*** See Cee Cee The Corner ***GlendaleAZ.0
z4PexFp1QdXGTCs8Eig**" Eco Mexican Food ***ChandlerAZ.5
z4PexFp1QdXGTCs8Eig**" Eco Mexican Food ***ChandlerAZ.5
z4PexFp1QdXGTCs8Eig**" Golden Spoke Cyclery ***GlendaleAZ.0
z4PexFp1QdXGTCs8Eig**" Golden Spoke Cyclery ***PhoenixAZ.0
z4PexFp1QdXGTCs8Eig**" Silvia Grill ***ChandlerAZ.9
z4PexFp1QdXGTCs8Eig**" Silvia Grill ***PhoenixAZ.9
z4PexFp1QdXGTCs8Eig**" Metier Pharmacy ***PhoenixAZ.5
z4PexFp1QdXGTCs8Eig**" Day Galaxi - DDS ***PhoenixAZ.0
z4PexFp1QdXGTCs8Eig**" Day Galaxi - DDS ***PhoenixAZ.0
z4PexFp1QdXGTCs8Eig**" Phoenix Pet Hospital ***ScottsdaleAZ.0
z4PexFp1QdXGTCs8Eig**" Phoenix Pet Hospital ***ScottsdaleAZ.0
z4PexFp1QdXGTCs8Eig**" Pine North Animal Hospital ***ScottsdaleAZ.4
z4PexFp1QdXGTCs8Eig**" Pine North Animal Hospital ***ScottsdaleAZ.4
z4PexFp1QdXGTCs8Eig**" Nighthider Jewelry ***ScottsdaleAZ.0
z4PexFp1QdXGTCs8Eig**" Nighthider Jewelry ***ScottsdaleAZ.0
```



**Analysis # 08:** Number of users who have yelp account but never provided any reviews yet:

Performing Minus Operation to find the data entries present in dataset that are not present in another dataset. There are users on yelp who haven't reviewed/rated any business yet, this MapReduce program will help find the details of such users.

#### Mapper#01

```
public class Mapper1 extends Mapper<Object, Text, Text, Text>{
```

```
    public Text outKey = new Text();
    public Text outVal = new Text();

    @Override
    protected void map(Object key, Text value, Mapper<Object, Text, Text, Text>.Context context)
        throws IOException, InterruptedException {

        String[] input = value.toString().split(",", -1);
        String userId = input[0];

        if(userId!= null || !userId.isEmpty()) {
            outKey.set(userId);
            outVal.set("U" + value.toString());
        }
    }
}
```

```

        context.write(outKey, outVal);
    }
}
}

Mapper#02
public class Mapper2 extends Mapper<Object, Text, Text, Text>{
    public Text outKey
    = new Text();  public Text outVal = new Text();

    @Override
    protected void map(Object key, Text value, Mapper<Object, Text, Text, Text>.Context context)
        throws IOException, InterruptedException {

        String[] input = value.toString().split(",",-1);
        String userId = input[1];

        if(userId!= null || !userId.isEmpty()) {
            outKey.set(userId);
            outVal.set("R" + value.toString());
            context.write(outKey, outVal);
        }
    }
}

```

**Reducer#01**

```

public class Reducer1 extends Reducer<Text, Text, Text, Text>{

private static final Text EMPTY_TEXT = new Text(""); private ArrayList<Text> listUsers
= new ArrayList<Text>(); private ArrayList<Text> listReviews = new ArrayList<Text>();
private Text tmpVal = new Text();

private String joinType = null;

    public void setup(Reducer.Context context) {
        joinType = context.getConfiguration().get("join.type");
    }

    public void reduce(Text key, Iterable<Text> values, Context context) throws IOException,
InterruptedException {

        listUsers.clear();
        listReviews.clear();

        while (values.iterator().hasNext()) {
tmpVal = values.iterator().next();

            if (Character.toString((char)
tmpVal.charAt(0)).equals("U")) {
                listUsers.add(new
Text(tmpVal.toString().substring(1)));
            }
        }
    }
}

```

## DriverClass

```
public class DriverClass {  
  
    public static void main(String[] args) throws IOException,  
InterruptedException, ClassNotFoundException {  
    Configuration conf = new Configuration();  
    Job job =  
Job.getInstance(conf, "AntiJoin");  
    job.setJarByClass(DriverClass.class);  
  
    MultipleInputs.addInputPath(job, new Path(args[0]),  
TextInputFormat.class, Mapper1.class);  
    MultipleInputs.addInputPath(job, new Path(args[1]), TextInputFormat.class, Mapper2.class);  
    job.getConfiguration().set("join.type", "minus");  
    job.setReducerClass(Reducer1.class);  
  
    job.setOutputFormatClass(TextOutputFormat.class);  
    TextOutputFormat.setOutputPath(job, new  
Path(args[2]));  
    job.setOutputKeyClass(Text.class);  
    job.setOutputValueClass(Text.class);  
  
    System.exit(job.waitForCompletion(true)? 0: 1);  
}  
}
```

## Output

```
[...antiJoin-0 [...].jar com.neu.edu.DriverClass /FinalProject/Dataset/yelp_user.csv
/FinalProject/Top25Restaurant/Dataset/yelp_business.cs/ /FinalProject/Top25Restaurant/FinalRes
ult56/
2019-08-14 23:37:03,999 WARN util.NativeCodeLoader: Unable to load native-hadoop library for y
our platform... using builtin-java classes where applicable
2019-08-14 23:37:04,505 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
2019-08-14 23:37:05,020 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing
not performed. Implement the Tool interface and execute your application with ToolRunner to r
emedy this.
2019-08-14 23:37:05,034 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path:
/tmp/hadoop-yarn/staging/sarthakgoel/.staging/job_1565831423867_0010
2019-08-14 23:37:05,206 INFO input.FileInputFormat: Total input files to process : 1
2019-08-14 23:37:05,229 INFO mapreduce.JobSubmitter: Cleaning up the staging area /tmp/hadoop-
yarn/staging/sarthakgoel/.staging/job_1565831423867_0010
Exception in thread "main" org.apache.hadoop.mapreduce.lib.input.InvalidInputException: Input
path does not exist: hdfs://localhost:9000/FinalProject/Top25Restaurant/Dataset/yelp_business.
cs
        at org.apache.hadoop.mapreduce.lib.input.FileInputFormat.singleThreadedListStatus(File
InputFormat.java:332)
        at org.apache.hadoop.mapreduce.lib.input.FileInputFormat.listStatus(FileInputFormat.ja
va:274)
        at org.apache.hadoop.mapreduce.lib.input.FileInputFormat.getSplits(FileInputFormat.ja
v:396)
        at org.apache.hadoop.mapreduce.lib.input.DelegatingInputFormat.getSplits(DelegatingInp
utFormat.java:115)
        at org.apache.hadoop.mapreduce.JobSubmitter.writeNewSplits(JobSubmitter.java:310)
        at org.apache.hadoop.mapreduce.JobSubmitter.writeSplits(JobSubmitter.java:327)
        at org.apache.hadoop.mapreduce.JobSubmitter.submitJobInternal(JobSubmitter.java:200)
```

```
r-00000
2019-08-14 23:49:06,829 WARN util.NativeCodeLoader: Unable to load native-hadoop library for
our platform... using builtin-java classes where applicable
--0WZ5gk1OfbUIodJuKfaQ,Scott,8,2013-02-19,0,0,0,0,None,4.2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
--1mPJZdSY9KluaBYAGboQ,Bryan,5,2011-07-04,0,0,0,0,None,5.0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
--318wysfp49Z2TLnyT0vg,Benjamin,111,2013-12-14,97,57,32,2,2016,3.43,0,0,0,0,0,0,1,2,2,3,0
--4uW4yJiRT2oXMYkCPq1Q,Ariella,50,2016-10-28,3,2,1,2,2017,4.06,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
--9kVkrIDkSP6lqK2PDTDw,Michelle,1,2015-04-10,0,0,0,0,None,5.0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
--ERQVpqAAoi262TTbLVzQ,Regina,1,2011-04-02,0,0,0,0,None,5.0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
--ET3paBtrThD95dk72Cqg,Lynda,8,2008-05-25,39,2,5,0,None,4.56,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
--FtMNxFtTE_KixNgPD1AQ,Ranjana,1,2017-11-06,0,0,0,0,None,5.0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
--GwB-sktmoAOPBsAaiow,Nina,2,2014-08-27,0,0,0,0,None,5.0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
--I4PlFQXAQcpk_eUJEW6w,Branko,2,2010-04-22,0,0,0,0,None,2.0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
--JSMB52zXJr_LBlklikyA,Diana,1,2017-06-14,0,0,0,0,None,1.0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
--K1aJ5K8ZLIfDd_NjySbA,Lisa,1,2016-03-13,0,0,0,0,None,5.0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
--NIc98RMssgy0mSZL3vpA,Nick,65,2013-08-15,2,0,0,1,None,3.91,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
--vrLQPsqckRv9TVqzg5wA,Aameemanu,17,2010-10-04,50,14,10,10,None,3.78,8,2,1,0,0,6,9,11,11,0,0
--xdSqqUJmcvJot-30Iq0g,Michelle,494,2008-08-02,14,8,1,16,"2014, 2011, 2015, 2010, 2016, 2012
2013",3.53,5,4,1,0,1,0,4,8,8,6,0
-01Cd1PEJYHhZYabHv8mIw,parag,5,2011-08-20,0,0,0,0,None,1.8,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
-01CztZ8o5UbarmEiQLDsQ,Kelly,2,2015-07-05,0,0,0,0,None,5.0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
-04zuZ0tQoGpgG49PbY-2Q,miafri,2,2012-11-26,0,0,0,0,None,5.0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
-0AyZxS5C-WySnbW_Q8yQ,Lars,507,2010-07-26,4,6,2,11,"2017, 2016, 2013, 2012, 2015, 2014",3.6
2,4,0,0,2,8,5,5,6,0
-0HHK0ux-abVe1mPQjyh7g,Lauren,3,2012-06-24,0,0,0,0,None,3.67,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
-0HHSb5xE3rNzXdqYIY3rQ,Emil,1,2013-05-20,0,0,0,0,None,2.0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
-0Hbf-cgvSsu8749nt1uyg,Adina,2,2015-10-31,0,1,0,0,None,3.0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
-0tbhfz1TAETE7kso-BtPw,Max,4,2017-11-01,1,1,0,0,None,3.67,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

**Analysis # 09:** Secondary Sort to find out how well a business is been rated by users per year:

Composite Key – BusinessID + Year

Extract Year from the given date and sort on the basis of ratings given by users to a business.

Following classes are created to perform secondary sort on the data.

1. CompositeKeyClass
2. MapperClass
3. Custom Partitioner Class
4. GroupComapartor Class
5. Secondary Sort Class
6. ReducerClas

### CompositeKeyClass:

```
public class CompositeKeyClass implements Writable,
WritableComparable<CompositeKeyClass> {

    private String businessYear;
    private Float rating;

    public String getBusinessYear() {
        return businessYear;
    }

    public void setBusinessYear(String businessYear) {
        this.businessYear = businessYear;
    }

    public Float getRating() {
        return rating;
    }

    public void setRating(Float rating) {
        this.rating = rating;
    }

    public void readFields(DataInput in) throws IOException {
        in.readUTF();
        rating = in.readFloat();
    }

    public void write(DataOutput out) throws IOException {
        out.writeUTF(businessYear);
        out.writeFloat(rating);
    }

    public int compareTo(CompositeKeyClass o) {
        int result = this.businessYear.compareTo(o.getBusinessYear());
        if (result == 0) {

```

```

        return this.rating.compareTo(o.getRating());
    }
    return result;
}

@Override
public String toString() {
    return businessYear;
}
}

```

**Mapper**

```

public class MapperClass extends Mapper<Object, Text,
CompositeKeyClass, FloatWritable> {

    private final static SimpleDateFormat frmt = new SimpleDateFormat("MM/DD/yy");
    private FloatWritable outVal = new FloatWritable();

    @Override
    protected void map(Object key, Text value, Mapper<Object, Text, CompositeKeyClass,
FloatWritable>.Context context)
        throws IOException, InterruptedException {

        try {
            String input[] = value.toString().split("\t");
            if(input.length >
5) {

                Calendar cal = Calendar.getInstance();
                String strDate = input[4];
                String businessId = input[2];
                String ratings = input[3];
                try {
                    cal.setTime(frmt.parse(strDate));
                } catch (ParseException e) {

                }

                String businessYear = businessId + " - " +
String.valueOf(cal.get(Calendar.YEAR));
                CompositeKeyClass compKey = new
CompositeKeyClass();
                compKey.setBusinessYear(businessYear);
                compKey.setRating(Float.parseFloat(ratings));
                outVal.set(Float.parseFloat(ratings));

                context.write(compKey, outVal);
            }
        }
    }
}

```

```
        }  
    } catch(Exception e) {  
  
    }  
}
```

## **Partition Class:**

```
public class PartitionerClass extends Partitioner<CompositeKeyClass,  
FloatWritable> {  
  
    @Override  
    public int getPartition(CompositeKeyClass key, FloatWritable text, int numberOfPartitions) {  
        // TODO Auto-generated method stub  
        return Math.abs(key.getBusinessYear().hashCode() % numberOfPartitions);  
    }  
}
```

### **SecondarySortComparator:**

```
public class SecondarySortComparator extends WritableComparator {  
  
    protected SecondarySortComparator() {  
        super(CompositeKeyClass.class, true);  
    }  
  
    @Override  
    public int compare(WritableComparable a, WritableComparable b) {  
  
        CompositeKeyClass ckw1 = (CompositeKeyClass) a;  
        CompositeKeyClass ckw2 = (CompositeKeyClass) b;  
  
        int result =  
ckw1.getBusinessYear().compareTo(ckw2.getBusinessYear());  
  
        if (result == 0) {  
            return -ckw1.getRating().compareTo(ckw2.getRating());  
        }  
  
        return result;  
    }  
}
```

**Group Comparator:**

```

public class GroupComparator extends WritableComparator{

    protected GroupComparator() {
        super(CompositeKeyClass.class, true);
    }

    @Override
    public int compare(Object a, Object b) {

        CompositeKeyClass ckw1 = (CompositeKeyClass)a;
        CompositeKeyClass ckw2 = (CompositeKeyClass)b;

        return
ckw1.getBusinessYear().compareTo(ckw2.getBusiness Year());
    }
}

```

**Reducer:**

```

public class ReducerClass extends Reducer<CompositeKeyClass,
FloatWritable, CompositeKeyClass, Text> {

    Text outVal = new Text();

    @Override
    protected void reduce(CompositeKeyClass key,
Iterable<FloatWritable> values,
Reducer<CompositeKeyClass, FloatWritable,
CompositeKeyClass, Text>.Context context)
        throws IOException, InterruptedException {

        StringBuilder sortedRating = new StringBuilder();
        for(FloatWritable val : values){           sortedRating.append(val.get());
            sortedRating.append(",");
        }
        outVal.set(sortedRating.toString());
        context.write(key,outVal);
    }
}

```

```
/target/SecondarySorting-0.0.1-SNAPSHOT.jar com.neu.edu.DriverClass /FinalProject/Dataset/yelp_review_tsv.tsv /FinalProject/Top25Restaurant/FinalResult59
2019-08-15 18:39:56,403 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
2019-08-15 18:39:57,016 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0:8032
2019-08-15 18:39:57,360 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2019-08-15 18:39:57,377 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/sarthakgoel/.staging/job_1565908449557_0002
2019-08-15 18:39:57,541 INFO input.FileInputFormat: Total input files to process : 1
2019-08-15 18:39:57,586 INFO mapreduce.JobSubmitter: number of splits:1
2019-08-15 18:39:57,692 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1565908449557_0002
2019-08-15 18:39:57,694 INFO mapreduce.JobSubmitter: Executing with tokens: []
2019-08-15 18:39:57,831 INFO conf.Configuration: resource-types.xml not found
2019-08-15 18:39:57,832 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2019-08-15 18:39:57,882 INFO impl.YarnClientImpl: Submitted application application_1565908449557_0002
2019-08-15 18:39:57,920 INFO mapreduce.Job: The url to track the job: http://Sarthaks-MacBook-Pro.local:8088/proxy/application_1565908449557_0002/
2019-08-15 18:39:57,920 INFO mapreduce.Job: Running job: job_1565908449557_0002
2019-08-15 18:40:02,995 INFO mapreduce.Job: Job job_1565908449557_0002 running in uber mode : false
2019-08-15 18:40:02,997 INFO mapreduce.Job: map 0% reduce 0%
2019-08-15 18:40:09,060 INFO mapreduce.Job: map 100% reduce 0%
2019-08-15 18:40:14,105 INFO mapreduce.Job: map 100% reduce 100%
2019-08-15 18:40:14,113 INFO mapreduce.Job: Job job_1565908449557_0002 completed successfully
```

hadoop fs -cat /FinalProject/Top25Restaurant/FinalResult59/part-r-00000

Review ID	Rating
Fi-2ruy5x600SX4avnrfuA	2008
Fi-2ruy5x600SX4avnrfuA	2010
Fi-2ruy5x600SX4avnrfuA	2010
Fi-2ruy5x600SX4avnrfuA	2010
Fi-2ruy5x600SX4avnrfuA	2011
Fi-2ruy5x600SX4avnrfuA	2012
Fi-2ruy5x600SX4avnrfuA	2013
Fi-2ruy5x600SX4avnrfuA	2014
Fi-2ruy5x600SX4avnrfuA	2014
Fi-2ruy5x600SX4avnrfuA	2014
Fi-2ruy5x600SX4avnrfuA	2015
Fi-2ruy5x600SX4avnrfuA	2016
Fi-2ruy5x600SX4avnrfuA	2016
Fi-2ruy5x600SX4avnrfuA	2016
Fi-2ruy5x600SX4avnrfuA	2017
FiEckfGo-rbYyhokHw_0MA	2011
FiEckfGo-rbYyhokHw_0MA	2013
FiEckfGo-rbYyhokHw_0MA	2013
Fig8PzWKRyehtPPcPtOStw	2014
Fig8PzWKRyehtPPcPtOStw	2014
Fig8PzWKRyehtPPcPtOStw	2016
FigMvTi7o6A0K1lZxI_g	2014
Fineh_tnwbOfM_r142o_DQ	2017
FiOpUj3j7Mca6QnDq9wMxw	2017
FiP7mGgs-qelOSmapmbt1A	2017
Fixu0gt0r0nHC1ncWErQSA	2013

### Analysis # 10: To Find the top 25 most rated business:

#### MapReduce Chaining

**MapReduce Job 1:** To find the count of no. of times each business is rated

**MapReduce Job 2:** Apply Top N Filtering Technique to find the top 25 rated businesses

#### Mapper # 01

```
public class Mapper1 extends Mapper<Object, Text, Text, IntWritable>{

    Text outKey = new Text();
    IntWritable outVal = new IntWritable();
    @Override
    protected void map(Object key, Text value, Mapper<Object, Text, Text, IntWritable>.Context context)
            throws IOException, InterruptedException {

        try {
            String input[] = value.toString().split("\t");
            outKey.set(input[2]);
            outVal.set(1);
            context.write(outKey, outVal);
        } catch(Exception e) {

        }
    }
}
```

#### Reducer #01

```
public class Reducer1 extends Reducer<Text, IntWritable, Text,
IntWritable> {
```

```
    IntWritable result = new IntWritable();

    @Override
    protected void reduce(Text key, Iterable<IntWritable> values,
                         Reducer<Text, IntWritable, Text, IntWritable>.Context context)
            throws IOException, InterruptedException {

        int sum = 0;
        for(IntWritable val : values) {
            sum += val.get();
        }
        result.set(sum);
        context.write(key, result);
    }
}
```

Step 02 to implement top n filtering pattern

### **Top25Mapper**

```
public class Top25Mapper extends Mapper<Object, Text, NullWritable,
Text>{

    private TreeMap<Integer,Text> tm = new TreeMap<Integer, Text>(); @Override
    protected void map(Object key, Text value, Mapper<Object, Text, NullWritable, Text>.Context
context)
        throws IOException, InterruptedException {

        String input[] = value.toString().split("\t");
        try {
            String business = input[0];
            int noOfRating = Integer.parseInt(input[1]);
            tm.put(noOfRating, new Text(value));

            if(tm.size()>25) {
                tm.remove(tm.firstKey());
            }
        }catch(Exception e) {

        }
    }

    @Override
    protected void cleanup(Mapper<Object, Text, NullWritable, Text>.Context context)
        throws IOException, InterruptedException {

        for(Text t: tm.values()) {
            context.write(NullWritable.get(), t);
        }
    }
}
```

### **Top25Reducer**

```
public class Top25Reducer extends Reducer<NullWritable, Text,
NullWritable, Text>{

    private TreeMap<Integer, Text> tm = new TreeMap<Integer, Text>();
```

```

@Override
protected void reduce(NullWritable key, Iterable<Text> values, Reducer<NullWritable, Text,
NullWritable, Text>.Context context) throws IOException, InterruptedException {
    for(Text value : values) {
        String input[] = value.toString().split("\t");
        int count = Integer.parseInt(input[1]);
        tm.put(count, new Text(value));
        if(tm.size() > 25) {
            tm.remove(tm.firstKey());
        }
    }
}

@Override
protected void cleanup(Reducer<NullWritable, Text, NullWritable, Text>.Context context)
    throws IOException, InterruptedException {
    for(Text t : tm.descendingMap().values()) {
        context.write(NullWritable.get(), t);
    }
}
}

```

**Driver Class**

```

public class DriverClass {

    public static void main(String[] args) throws IOException,
    ClassNotFoundException, InterruptedException {
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "Top25Business");
        job.setJarByClass(DriverClass.class);

        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));

        job.setInputFormatClass(TextInputFormat.class);
        job.setOutputFormatClass(TextOutputFormat.class);
        job.setMapOutputKeyClass(Text.class);
        job.setMapOutputValueClass(IntWritable.class);

        job.setMapperClass(Mapper1.class);
        job.setReducerClass(Reducer1.class);

        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);

        boolean result = job.waitForCompletion(true);

        if(result) {
            Job job1 = Job.getInstance(conf, "Top25Business");
            job1.setJarByClass(DriverClass.class);
        }
    }
}

```

```

        job1.setInputFormatClass(TextInputFormat.class);
                job1.setOutputFormatClass(TextOutputFormat.class);
        job1.setMapperClass(Top25Mapper.class);
        job1.setReducerClass(Top25Reducer.class);

        job1.setOutputKeyClass(NullWritable.class);
        job1.setOutputValueClass(Text.class);

        FileInputFormat.addInputPath(job1, new Path(args[1]));
        FileOutputFormat.setOutputPath(job1, new Path(args[2]));

        result = job1.waitForCompletion(true);
    }
}
}
}

```

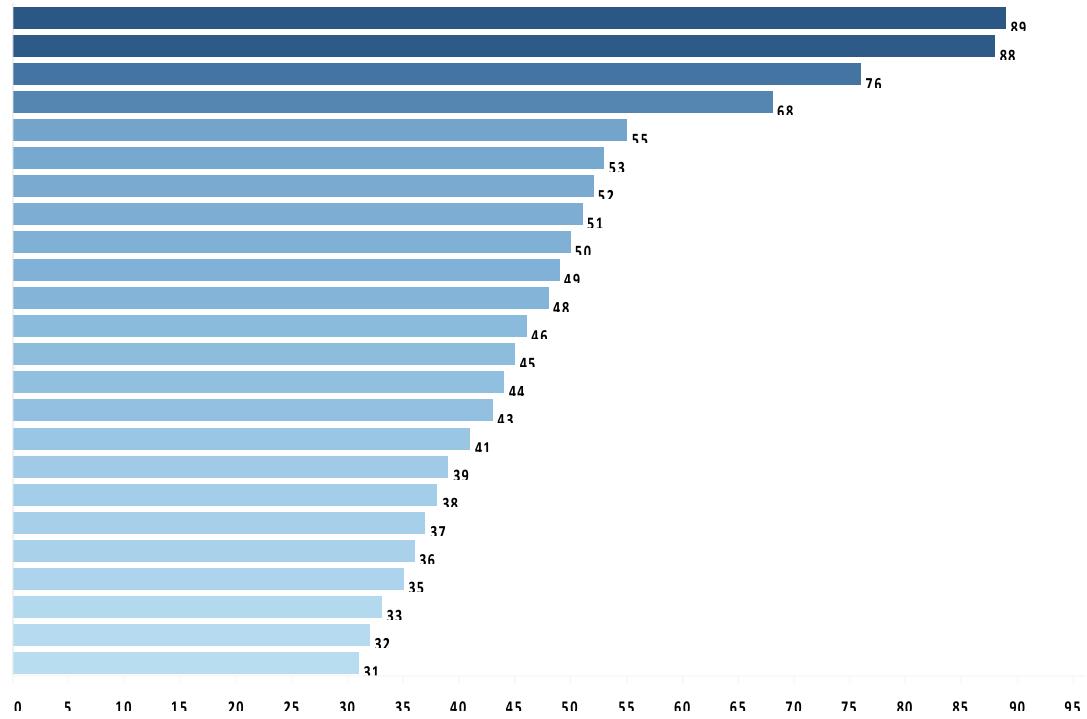
```

rget/NoOfRestaurants-0.0.1-SNAPSHOT.jar com.neu.edu.DriverClass /FinalProject/yelp_business.ts
v.tsv /FinalProject/Top25Restaurant/FinalResult80/ /FinalProject/Top25Restaurant/FinalResult81
/
2019-08-16 00:21:30,394 WARN util.NativeCodeLoader: Unable to load native-hadoop library for y
our platform... using builtin-java classes where applicable
2019-08-16 00:21:30,941 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
2019-08-16 00:21:31,577 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing
not performed. Implement the Tool interface and execute your application with ToolRunner to r
emedy this.
2019-08-16 00:21:31,592 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path:
/tmp/hadoop-yarn/staging/sarthakgoel/.staging/job_1565925804834_0016
2019-08-16 00:21:31,782 INFO input.FileInputFormat: Total input files to process : 1
2019-08-16 00:21:31,823 INFO mapreduce.JobSubmitter: number of splits:1
2019-08-16 00:21:31,942 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1565925804
834_0016
2019-08-16 00:21:31,944 INFO mapreduce.JobSubmitter: Executing with tokens: []
2019-08-16 00:21:32,009 INFO conf.Configuration: resource-types.xml not found
2019-08-16 00:21:32,009 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2019-08-16 00:21:32,143 INFO impl.YarnClientImpl: Submitted application application_1565925804
834_0016
2019-08-16 00:21:32,187 INFO mapreduce.Job: The url to track the job: http://Sarthaks-MacBook-
Pro.local:8088/proxy/application_1565925804834_0016/
2019-08-16 00:21:32,187 INFO mapreduce.Job: Running job: job_1565925804834_0016
2019-08-16 00:21:38,266 INFO mapreduce.Job: Job job_1565925804834_0016 running in uber mode :
false
2019-08-16 00:21:38,268 INFO mapreduce.Job: map 0% reduce 0%
2019-08-16 00:21:42,318 INFO mapreduce.Job: map 100% reduce 0%
2019-08-16 00:21:47,361 INFO mapreduce.Job: map 100% reduce 100%

```

4JNXUYY8wbaaDmk3BPzlw	89
RESDUcs7fIiihp38-d6_6g	88
K71wdNUhCbcnEvi0NhGewg	76
cYwJA2A6I12KNkm2rtXd5g	68
iCQpiavjjPzJ5_3gPD5Ebg	55
eoHdUeQDNgQ6WYEnP2aiRw	53
ujHiaprwCQ5ewziu0Vi9rw	52
f4x1YBxkLtzGz652xt2KR5g	51
5LNZ67Yw9RD6nf4_UhXOjw	50
E14Fc8jcawUVgw_0EIcbaQ	49
hihud--QRriCYZwlzZvW4g	48
KskYqH1Bi7Z_61pH60m8pg	46
SMPbvZLSMMb7KU76YNyMGg	45
rcaPajgKOJC2vo_13xa42A	44
FahADZARwnY4yvlvpnsfGA	43
yfxDa8RF0vJPQh0rNtakHA	41
eAc9Vd6lo0gRQo1MXQt6FA	39
uanCi40Gc1mHLG1_AT4JhQ	38
ii8sAGBexBOJoYRFafF9XQ	37
MpmFFw0GE_2iRFPdsRpJbA	36
XXW_OFaYQkkGOGniujZFhg	35
QJatAcxYgK1Zp9BRZMax7g	33
pH0BLkL4cbxKzu471VZnuA	32
na4Th5DrNauOv-c43QQFvA	31

### Most Reviewed Business By Users



### Analysis # 11: Recommendation Using Mahout

#### Step 1 Pig

```
A =LOAD 'hdfs://localhost:9000/FinalProject/Dataset/yelp_review_tsv.tsv' USING
org.apache.pig.piggybank.storage.CSVExcelStorage('\t', 'NO_MULTILINE',
'SKIP_INPUT_HEADER');
```

```
fltrvw = FOREACH A GENERATE (chararray) $1 as userId, (chararray) $2
as businessID, (chararray) $3 as rating;
```

```
STORE fltrvw INTO
'hdfs://localhost:9000/FinalProject/Hive/MahoutInputUsers/' Using
PigStorage('\t');
```

#### Step 02 Copy the file to local

```
hadoop fs -copyToLocal /FinalProject/Hive/MahoutInputUsers/part-m-
00000 /Users/poojithamuppalla/Desktop/Project/Analysis/Mahout/Input/
```

### Step 03 Program to create a Mahout Input

```

static HashMap<String, Integer> um = new HashMap<String, Integer>();
static HashMap<String, Integer> bm = new HashMap<String, Integer>();
private static int getUser(String user)
{
if (!um.containsKey(user))
um.put(user, um.size() + 1);
return um.get(user);
}
private static int getBusiness(String business)
{
if (!bm.containsKey(business))
bm.put(business, bm.size() + 1);
return bm.get(business);
}
private static void saveMapping(HashMap<String, Integer> mapping,
String fileName) throws IOException
{
int limitPrint = mapping.size();
FileWriter fstream;
BufferedWriter out;
fstream = new
FileWriter("/Users/poojithamuppalla/Desktop/Project/Analysis/Mahout/Input/" + fileName);
out = new BufferedWriter(fstream);
for (Map.Entry<String, Integer> entry : mapping.entrySet()) {
//System.out.println("Key = " + entry.getKey() + ", Value= " + entry.getValue());
out.write(entry.getKey() + "\t");
out.write(entry.getValue() + "\n");
}
out.close();
}
public static void mahoutInput() throws IOException {
try {
File file = new
File("/Users/poojithamuppalla/Desktop/Project/Analysis/Mahout/Input/InputFile.csv");
FileWriter fstream = new
FileWriter("/Users/poojithamuppalla/Desktop/Project/Analysis/Mahout/Input/MahoutInput.csv");
BufferedReader br = new BufferedReader(new FileReader(file));
BufferedWriter out = new BufferedWriter(fstream);
String st;
while ((st = br.readLine()) != null) {
if(!st.isEmpty()) {
String input[] = st.split(",");
int user = getUser(input[0]);
int business = getBusiness(input[1]);
String ratings = input[2];
if(!String.valueOf(user).isEmpty() &&
!String.valueOf(business).isEmpty() && !ratings.isEmpty())
out.write(String.valueOf(user));
out.write(",");
}
}
}
}

```

```

out.write(String.valueOf(business));
out.write(",");
out.write(ratings);
out.write("\n");
}
br.close();
out.close();
} catch(Exception e){
}
//writer.Close();
saveMapping(um, "usersMap.csv");
saveMapping(bm, "songMapping.csv");
}
public static void main(String[] args) throws IOException {mahoutInput()};
}

```

#### **Step 04 Mahout Recommendation**

```

public class MahoutRecommender {
public static void main(String[] args) throws IOException,
TasteException{
File userPreferencesFile = new
File("/Users/poojithamuppalla/Desktop/Project/Analysis/Mahout/Input/MahoutInput.csv");
DataModel dataModel = new
FileDataModel(userPreferencesFile);

```

#### **Step01 Pig Commands to get the userId, businessId, Ratings**

The screenshot shows a file browser interface. At the top, there are three buttons: 'Download', 'Head the file (first 32K)', and 'Tail the file (last 32K)'. Below these, a green header bar displays 'Block information -- Block 0'. The main content area is divided into two sections: 'Block information' and 'File contents'.

**Block information:**

- Block ID: 1073745668
- Block Pool ID: BP-1416040093-172.20.20.20-1559801065801
- Generation Stamp: 4844
- Size: 3120588
- Availability:

  - 172.20.20.20

**File contents:**

```

zRRurs5wB7er42zvPkty9w zwNLJ2VgffEvGu7DDZjU4g 3
zRRurs5wB7er42zvPkty9w ropJyZdmv6NeDU2Vz1L_Hg 4
zRRurs5wB7er42zvPkty9w 0e1cb0Q 4
zRRurs5wB7er42zvPkty9w Mz2ABTdavsg1Hkj-CwvJBw 3
zRRurs5wB7er42zvPkty9w E14FC08jcauvUVgw_0E1cb0Q 4
zRRurs5wB7er42zvPkty9w 2o3YsY977f3GwZpPTNAvg 4
zRRurs5wB7er42zvPkty9w 072a959IU38uVeYuLLMA 3
zRRurs5wB7er42zvPkty9w DfgZINgKwBvCpA_0alumXw 4

```

A 'Close' button is located at the bottom right of the file contents panel.

**Step02 to create a mapping file, since the output file is unstructured and Mahout needs a proper structured file to perform recommendation so we are creating a hashmap for businessID and userID and save their mapping Business Mapping**

The screenshot shows a spreadsheet application interface with a dark theme. The top menu bar includes 'View', 'Zoom', 'Add Category', 'Insert', 'Table', 'Chart', 'Text', 'Shape', 'Media', 'Comment', 'Collaborate', 'Format', and 'Organize'. A status bar at the bottom right indicates 'Table data was imported.' and 'Adjust Settings'.

**Sheet 1**

**songMapping**

UidEFF1WpnU4duvev4fPIQ	5572
mD7zqv7Y3kvsa_p_MtTayg	1140
FzxodbuXZKALR2yNRN8PUA	4252
WO2nNar_wlQ3flAf3MM1Q	7627
J3H6VStgUTIACk_b_HPFaBaw	15
x_yLsQMQtIE0rdxnaf6J5w	648
FxEJLJZjUdC0CCcgEufeA	8514
7Pjsu4x63VoCXPmbaIVA	4679
p6M16PUsAviZiXIT2rJCGQ	5334
TvDsoVbGEGXKBCaVcICKVA	7770
AHTgvY2I6vK1FLCmndRbwQ	6468
c6f8wBjPLDzyubEBqggMnw	5783
ZOJTOFras-kMwJLMdn9JQQ	2142
N95IG-T9bbR2INL9_anISO	5036
85NJYsnJt076Nd0Kwslibg	4199
OJdufU3hVabgvIBHksYw	7047
p0vTYCIfgMm-wAXPW1ChnQ	5919
eQel7bUz75j0AVKnfslig	3644
Zwg2TWwdm7y50RK3yN6nDw	928
6qKj6eJ0zLb9ka-QbTuDa	2017
LulDt1xElhcvyha00wCwZA	4067
PNziJfTJAD7u41Gw/R98-w	263
HaRN97QISnUJHk21QAIKw	3451
psNjgpMDRhnBj6xzhHPVQ	2839
YSAl_I2wLGIGEcLgxR09Ug	7312
Oc7BNz-sOyo16LYElAza	4905
epvLkGNL6MOVk3sUJiTnTA	8034
hD77eYBzXoYQfpAU MY8dEw	604
DogtzIX8QqcSgixqnQ9WQ	4600
CPgjfcsyPegLhatajaSRoA	3450
TA1Q7RWVzMigI5X3FAmctw	177

## User Mapping

The screenshot shows a spreadsheet application interface with a dark theme. The top menu bar includes 'View', 'Zoom', 'Add Category', 'Insert', 'Table', 'Chart', 'Text', 'Shape', 'Media', 'Comment', 'Collaborate', 'Format', and 'Organize'. A status bar at the bottom right indicates 'Table data was imported.' and 'Adjust Settings'.

**Sheet 1**

**usersMap**

hxXN8h9_XXt14vqWVD6VJNg	87
m7UE-3UEVHoXzjZN9j2rQ	850
CVIF-KfhXlh5GOOAUDMTQ	1822
v1FXMGvlva3wF78f6u-aw	979
1BKX1svSaVpNaXMGxOyjUA	1579
ISvhvGDAL8T_bDneKaVgNA	1242
9b7zJ0opAxhqdqMKTAmpOA	1526
irkVoCgh393QZTXVbgdwA	568
07x1nr2ha5ODjaLLEPlw	1324
Nv0SGeP-EEtstyjQzJAFEA	23
8Q15MkHpK9szYg_8WAaZZSw	150
rxZmf0Soyju_l5Tj0s16dqw	408
iew9IGWB-kGRDMSMii_QTvw	644
OJYWsIBQInrEjjJhiULIA	275
UKGvWlzb9j2mJILBmLhrg	1903
Xz-ab3qbgIwgl2xFN12DQ	370
8VuoEfhtbcIJls-IFg6fCWw	1676
xyVht4ogkb3xNt8aQ5rA	654
Seh1qz6yijlbmB58Cf53A	1344
gu_-VnXhpEVGY4nnDu9IA	638
qTs_GibdRA1dasy6D3rbQ	795
haYMiQAtte_ZaiMDZZ3TQ	1562
OxeGhwFcDBVL6-TwTfIbg	606
RWXAs61RpvcNw8kYKK0BEw	1458
4sCNqow8kLbh2N1CgYQ	2414
lfFjEnf5vN6m_Fhxwy6aZA	584
AGWHoXr_g9x0pBtIJQDDQ	1338
mmqmknMhieEwfEOqx7h8Q0	2234
212rwgkduT9JwP8R_Cvvu	2351
nk0cnv1-asW0USIKdfLlQ	1771

## Mahout Input Passed to Mahout Recommendation Engine

	MahoutInput
1	1 5
2	2 5
3	3 5
4	4 4
5	5 4
6	6 5
7	7 4
8	8 4
9	9 3
10	10 5
11	11 2
12	12 2
13	13 5
14	14 5
15	15 1
16	16 3
17	17 4
18	18 1
19	19 4
20	20 5
21	21 4
22	22 4
23	23 2
24	24 4
25	25 4
26	26 5
27	27 3
28	28 1

## Step 04 Mahout Recommendation

```

1 package com.neu.edu;
2
3 import java.io.BufferedReader;
4
5 public class MahoutRecommendation {
6     public static void main(String[] args) throws IOException, TasteException{
7         File userPreferencesFile = new File("/Users/sarthakgoel/Desktop/Project/Analysis/Mahout/Input/MahoutInput");
8         DataModel dataModel = new FileDataModel(userPreferencesFile);
9
10        UserSimilarity similarity = new PearsonCorrelationSimilarity(dataModel);
11        UserNeighborhood neighborhood = new ThresholdUserNeighborhood(2.0, similarity, dataModel);
12
13        Recommender recommender = new GenericUserBasedRecommender(dataModel, neighborhood, similarity);
14
15        System.out.println("Recommendations for user " + user);
16        List<RecommendedItem> recommendations = recommender.recommend(user, 10);
17
18        for (RecommendedItem recommendation : recommendations) {
19            System.out.println("Recommended Item Id " + recommendation.getItemID() + ". Strength of the preference: " + recommendation.getStrength());
20        }
21    }
22}

```

Output from the Console tab:

```

User Id: 2252
No recommendations for this user.
User Id: 2253
No recommendations for this user.
User Id: 2254
Recommended Item Id 779. Strength of the preference: 5.000000
Recommended Item Id 648. Strength of the preference: 4.655193
Recommended Item Id 52. Strength of the preference: 4.267346
User Id: 2255
No recommendations for this user.
User Id: 2256
Recommended Item Id 1383. Strength of the preference: 5.000000
No recommendations for this user.
User Id: 2257
No recommendations for this user.
User Id: 2258
No recommendations for this user.
User Id: 2259
No recommendations for this user.
User Id: 2260
No recommendations for this user.
User Id: 2261
No recommendations for this user.
User Id: 2262
No recommendations for this user.
User Id: 2263
No recommendations for this user.
User Id: 2264
No recommendations for this user.
User Id: 2265
No recommendations for this user.
User Id: 2266
No recommendations for this user.
User Id: 2267
No recommendations for this user.
User Id: 2268
No recommendations for this user.
User Id: 2269
No recommendations for this user.

```

**Conclusion:**

I performed multiple analysis on this dataset by running Hadoop on single machine.

Apache Hive provides a SQL dialect that enables us to perform analysis on underlying data. These underlying queries are converted to MapReduce jobs and further you could store the output on HDFS.

On similar line, I used Apache Pig Latin and accessed the Pig using CLI and wrote Data Flow Queries to perform MapReduce operations on yelp dataset.

Apache Mahout is a powerful machine learning library for recommendation system, and I was able run the recommendations program over a large dataset very quickly.

**Citations:**

1. <http://pig.apache.org/docs/latest/api/org/apache/pig/builtin/PigStorage.html>
2. <https://pig.apache.org/docs/r0.17.0/api/org/apache/pig/piggybank/storage/package-summary.html>
3. <https://pig.apache.org/docs/r0.16.0/udf.pdf>
4. <https://hive.apache.org/>
5. [https://medium.com/@sudarshan\\_sreenivasan/what-is-secondary-sort-in-hadoop-and-how-does-it-work-fe35609b5319](https://medium.com/@sudarshan_sreenivasan/what-is-secondary-sort-in-hadoop-and-how-does-it-work-fe35609b5319)
6. <http://blog.ditullio.fr/2015/12/28/hadoop-basics-secondary-sort-in-mapreduce/>
7. <https://github.com/goelsarthak>
8. <https://github.com/AmiGandhi>