Big Data Analysis on Yelp Dataset

Data

Review {'type': 'review', 'business_id': (encrypted business id), 'user_id': (encrypted user id), 'stars': (star rating, rounded to half-stars), 'text': (review text), 'date': (date, formatted like '2012-03-14'), 'votes': {(vote type): (count)},}

User {'type': 'user', 'user_id': (encrypted user id), 'name': (first name), 'review_count': (review count), 'average_stars': (floating point average, like 4.31), 'votes': {(vote type): (count)}, 'friends': [(friend user_ids)], 'elite': [(years_elite)], 'yelping_since': (date, formatted like '2012-03'), 'compliments': { (compliment_type): (num_compliments_of_this_type), ... }, 'fans': (num_fans),}

Checkin {'type': 'checkin', 'business_id': (encrypted business id), 'checkin_info': { '0-0': (number of checkins from 00:00 to 01:00 on all Sundays), '1-0': (number of checkins from 01:00 to 02:00 on all Sundays), ... '14-4': (number of checkins from

14:00 to 15:00 on all Thursdays), ... '23-6': (number of checkins from 23:00 to 00:00 on all Saturdays) }, # if there was no checkin for a hour-day block it will not be in the dict}

Loading data into hdfs :

1)

```
./hdfs dfs -copyFromLocal '/Users/nimeshrajal/Desktop/Sharna_BigData/yelp_academic_dataset_user.json' '/FinalProject/rawdata'
```

2)

```
./hdfs dfs -copyFromLocal '/Users/nimeshrajal/Desktop/Sharna_BigData/yelp_academic_dataset_review.json' '/FinalProject/rawdata'
```

3)

```
./hdfs dfs -copyFromLocal '/Users/nimeshrajal/Desktop/Sharna_BigData/yelp_academic_dataset_checkin.json' '/FinalProject/rawdata'
```

4)

```
./hdfs dfs -copyFromLocal '/Users/nimeshrajal/Desktop/Sharna_BigData/yelp_academic_dataset_tip.json' '/FinalProject/rawdata'
```

5)

```
./hdfs dfs -copyFromLocal '/Users/nimeshrajal/Desktop/Sharna_BigData/yelp_academic_dataset_business.json' '/FinalProject/rawdata'
```

Using Pig

Registering the jars to use the json files

1)

```
Register /Users/nimeshrajal/Desktop/Sharna_BigData/elephant-bird-core-4.5.jar
```

2)

```
Register /Users/nimeshrajal/Desktop/Sharna_BigData/elephant-bird-pig-4.5.jar
```

3)

```
Register /Users/nimeshrajal/Desktop/Sharna_BigData/elephant-bird-hadoop-compat-4.5.jar
```

4)

```
Register /Users/nimeshrajal/Downloads/simple-json-1.1.jar
```

Analysis 1: Count top 25 tips by user

```
tipJson = load ‘/FinalProject/rawdata/yelp_academic_dataset_tip.json' using com.twitter.elephantbird.pig.load.JsonLoader('-nestedLoad');

red = foreach tipJson generate $0#'text' as text, $0#'user_id' as userid ,$0#'business_id' as businessid;

group_by_user = group red by (userid,businessid);

tipcount = foreach group_by_user generate group as tcount, COUNT(red) as tcounts;
```

```
orderbyre = order tipcount by tcounts desc;

top25 = limit orderbyre 25;
```

Analysis 2 : Top 15 likes by a user

```
genuser = foreach tipJson generate $0#'user_id' as userid, $0#'likes' as likes;

groupbyuser = group genuser by userid;

countlikes = foreach groupbyuser generate group as likesct, COUNT(genuser.likes) as likes1;

orderdesc = order countlikes by likes1 desc;

top15 = limit orderdesc 15;
```

Analysis 3 : Business based on stars

```
business = load '/home/ron/Documents/yelpdata/yelp_academic_dataset_business.json' using com.twitter.elephantbird.pig.load.JsonLoader('-nestedLoad');
```

```
joinBusiness = foreach business generate
$0#'business_id' as businessid;



tipJson = load
'/home/ron/Documents/yelpdata/yelp_academic_d
ataset_tip.json' using
com.twitter.elephantbird.pig.load.JsonLoader('-
nestedLoad')’;

genUser1 = foreach tipJson generate
$0#'business_id' as businessid,$0#'user_id' as userid,
$0#'likes' as likes ;

joinforlikes = join joinBusiness by businessid,
genUser1 by businessid;
```

Storing the output in HDFS

1)

```
STORE top25 INTO
‘/FinalProject/Output/PigOutput/Analysis1’ using
PigStorage(‘,’);
```

2)

```
STORE top15 INTO
‘/FinalProject/Output/PigOutput/Analysis2’ using
PigStorage(‘,’);
```

Storing PigOutPut from HDFS into Local

1)

```
./hdfs dfs –copyToLocal
‘/FinalProject/Output/PigOutput/Analysis1’
‘/Users/nimeshrajal/Desktop/Sharna_BigData/Pigou
tput/Analysis1’;
```

2)

```
./hdfs dfs –copyToLocal
‘/FinalProject/Output/PigOutput/Analysis2’
‘/Users/nimeshrajal/Desktop/Sharna_BigData/Pigou
tput/Analysis2’;
```

Using Hive

Add JAR files ‘json-serde-1.1.9.2-Hive13-jar-with-dependencies.jar’, ‘json-serde-1.1.9.2-Hive13.jar’

Creating External Tables :

1)

```
CREATE EXTERNAL TABLE business_yelp(business_id
string, name string,full_address string, city string,
state string, categories array<string>, latitude
double, longitude double) ROW FORMAT SERDE
'org.openx.data.jsonserde.JsonSerDe';
```

2)

```
CREATE EXTERNAL TABLE checkin_yelp (business_id
string, checkin_infor map<string,int>) ROW FORMAT
SERDE 'org.openx.data.jsonserde.JsonSerDe';
```

3)

```
CREATE EXTERNAL TABLE user_yelp(user_id string,
name string, review_count int, votes
map<string,int>, friends array<string>,fans int,
yelping_since string, elite array<string>,
complements map<string,int>, average_stars float)
ROW FORMAT SERDE
'org.openx.data.jsonserde.JsonSerDe';
```

4)

```
CREATE EXTERNAL TABLE  review_yelp(business_id
string, user_id string, stars float, date string, votes
map<string,int>) ROW FORMAT SERDE
'org.openx.data.jsonserde.JsonSerDe';
ALTER TABLE business_yelp SET
SERDEPROPERTIES("ignore.malformed.json"="true");
```

Loading data in the tables :

1)

```
load data inpath
'/FinalProject/rawdata/yelp_academic_dataset_revi
ew.json' into table review_yelp;
```

2)

```
load data inpath
'/FinalProject/rawdata/yelp_academic_dataset_user
.json' into table user_yelp;
```

3)

```
load data INPATH '/FinalProject/rawdata/yelp_academic_dataset_business.json' INTO business_yelp;
```

4)

```
load data inpath '/FinalProject/rawdata/yelp_academic_dataset_checkin.json' into table checkin_yelp;
```

5)

```
load data inpath '/FinalProject/rawdata/yelp_academic_dataset_tip.json' into table tip_yelp;
```

Query 1 :

```
select business_id,SUM(votes['useful']) as vote from review_yelp group by business_id limit 5;
```

Query 2:

```
select business_id,SUM(votes['useful']) as voteuseful,SUM(votes['cool']) as votecool, SUM(votes['funny']) as votesfunny from review_yelp group by business_id ;
```

Query 3:

```
select COUNT(*),yelping_since from user_yelp group
by yelping_since;
```

Query 4:

```
select user_id,name,elite from user_yelp order by
elite desc;
```

Query 5:

```
select d.count, d.city, d.name, d.longitude, d.latitude
from(select distinct B.name, B.city,
C.checkin_info['9-1'] as count,
        B.longitude, B.latitude, B.business_id from
business_yelp B FULL OUTER JOIN checkin_yelp C
on(B.business_id = C.business_id) ) d
        where name = 'Starbucks' and city='Phoenix'
and count < 5;
```