**Big Data Code Drop Final**

Prachee Mhatre: N18242365 (phm265)

Shruti Goel: N12029775 (sg4224)

Sreerupa Nandi: N11396512 (sn1744)

Description:

In this research work we have measured various cities and towns in the United States based on livability conditions to find the most livable ones. We have analyzed data across over 300 cities to come up with absolute rankings in terms of the deciding factors. We have initially come up with some basic rankings without equal weight across all the factors. We have then further extended our studies by assigning appropriate weights to these factors and presented the twenty most livable cities across the nation.

Code:

----------------------------------------------------------------------------------------------------------------------------------

datacity = LOAD '/home/cloudera/Desktop/bigdataproj/population.txt' USING PigStorage('|') AS (City:chararray, State:chararray, Population:int, area:double, density:int);

filtercity = FILTER datacity by (City is not null) AND ($2 is not null) ;

filterspaces = FOREACH filtercity GENERATE REPLACE(City, ' ', '') as City, $1,$2,$3,$4;

citydata = ORDER filterspaces BY Population DESC;

top300PopulatedCities = LIMIT citydata 300;

// Calculating top300 cities based on population for filtering the datasets.

raw\_employ = LOAD '/home/cloudera/Desktop/bigdataproj/unemploy.txt' USING PigStorage(';') AS (city:chararray, state:chararray, Rate:float, Year:int);

raw = FILTER raw\_employ by ($0 is not null) AND ($2 is not null) AND ($3 is not null);

employ\_join = JOIN top300PopulatedCities by City, raw\_employ by city;

unemploycities = FOREACH employ\_join GENERATE top300PopulatedCities::City, raw\_employ::Rate , top300PopulatedCities::Population , raw\_employ::Year;

// Cleansing unemployment data

raw\_crime = LOAD '/home/cloudera/Desktop/bigdataproj/crime.txt' USING PigStorage(';') AS ( Year:int, City:chararray, Population:int, Violent\_crime:int , Murdermanslaughter:int, Forcible\_rape:int, Robbery:int, Aggravated\_assault:int, Property\_crime:int, Burglary:int, Larceny\_theft:int, Motor\_vehicle\_theft:int) ;

cal\_Total = FOREACH raw\_crime GENERATE Year,City,Population, ($3 + $8) As (TotalCrime:int);

crime\_join = JOIN cal\_Total by City, top300PopulatedCities by City;

crimecities = FOREACH crime\_join GENERATE top300PopulatedCities::City, cal\_Total::TotalCrime, top300PopulatedCities::Population, cal\_Total::Year;

// Cleansing crime data

raw\_air = LOAD '/home/cloudera/Desktop/bigdataproj/air.txt' USING PigStorage(';') AS (City:chararray,DaysAQI:int,goodDays:int,moderateDays:int,unhealthyDays:int, year:int);

raw = FILTER raw\_air by ($0 is not null) ;

air\_join = JOIN raw by city, top300PopulatedCities by $0;

aircities = FOREACH air\_join GENERATE raw::city as city, raw::goodDays as goodDays ,raw::unhealthyDays as unhealthyDays , top300PopulatedCities::Population as population , raw::year as year, raw::rank as int ;

//Cleansing air data

Unemploy2010 = Filter unemploycities BY Year == 2010;

Crime2010 = Filter crimecities BY Year == 2010;

crimeUnemploy\_join2010 = JOIN Unemploy2010 by City, Crime2010 by City;

topcitiesfor2010 = FOREACH crimeUnemploy\_join2010 GENERATE Unemploy2010::raw\_employ::Year , Unemploy2010::top300PopulatedCities::City , Unemploy2010::raw\_employ::Rate, Crime2010::cal\_Total::TotalCrime ;

sortcities2010 = order topcitiesfor2010 by TotalCrime asc , Rate asc;

STORE sortcities2010 INTO '/home/cloudera/Desktop/bigdataproj/output2010' USING PigStorage(',');

Unemploy2011 = Filter unemploycities BY Year == 2011;

Crime2011 = Filter crimecities BY Year == 2011;

crimeUnemploy\_join2011 = JOIN Unemploy2011 by City, Crime2011 by City;

topcitiesfor2011 = FOREACH crimeUnemploy\_join2011 GENERATE Unemploy2011::raw\_employ::Year , Unemploy2011::top300PopulatedCities::City , Unemploy2011::raw\_employ::Rate, Crime2011::cal\_Total::TotalCrime ;

sortcities2011 = order topcitiesfor2011 by TotalCrime asc , Rate asc;

STORE sortcities2011 INTO '/home/cloudera/Desktop/bigdataproj/output2011' USING PigStorage(',');

Unemploy2012 = Filter unemploycities BY Year == 2012;

Crime2012 = Filter crimecities BY Year == 2012;

crimeUnemploy\_join2012 = JOIN Unemploy2012 by City, Crime2012 by City;

topcitiesfor2012 = FOREACH crimeUnemploy\_join2012 GENERATE Unemploy2012::raw\_employ::Year , Unemploy2012::top300PopulatedCities::City , Unemploy2012::raw\_employ::Rate, Crime2012::cal\_Total::TotalCrime ;

sortcities2012 = order topcitiesfor2012 by TotalCrime asc , Rate asc;

STORE sortcities2012 INTO '/home/cloudera/Desktop/bigdataproj/output2012' USING PigStorage(',');

Unemploy2013 = Filter unemploycities BY Year == 2013;

Crime2013 = Filter crimecities BY Year == 2013;

crimeUnemploy\_join2013 = JOIN Unemploy2013 by City, Crime2013 by City;

topcitiesfor2013 = FOREACH crimeUnemploy\_join2013 GENERATE Unemploy2013::raw\_employ::Year , Unemploy2013::top300PopulatedCities::City , Unemploy2013::raw\_employ::Rate, Crime2013::cal\_Total::TotalCrime ;

sortcities2013 = order topcitiesfor2013 by TotalCrime asc , Rate asc;

STORE sortcities2013 INTO '/home/cloudera/Desktop/bigdataproj/output2013' USING PigStorage(',');

Unemploy2014 = Filter unemploycities BY Year == 2014;

Crime2014 = Filter crimecities BY Year == 2014;

crimeUnemploy\_join2014 = JOIN Unemploy2014 by City, Crime2014 by City;

topcitiesfor2014 = FOREACH crimeUnemploy\_join2014 GENERATE Unemploy2014::raw\_employ::Year , Unemploy2014::top300PopulatedCities::City , Unemploy2014::raw\_employ::Rate, Crime2014::cal\_Total::TotalCrime ;

sortcities2014 = order topcitiesfor2014 by TotalCrime asc , Rate asc;

STORE sortcities2014 INTO '/home/cloudera/Desktop/bigdataproj/output2014' USING PigStorage(',');

airdata = ORDER aircities BY unhealthyDays ASC;

groupyearair = GROUP airdata BY $4;

Top\_air = foreach groupyearair {

data2 = ORDER data BY goodDays DESC;

top = limit data2 200;

generate group, flatten(top);

};

STORE Top\_air INTO '/home/cloudera/Desktop/bigdataproj/air' USING PigStorage(',');

// filtering the data based on year

//store the data in hdfs

hdfs dfs -put /home/cloudera/Desktop/bigdataproj/output2010/part-r-00000 /user/cloudera/BDP/output2010/result2010

hdfs dfs -put /home/cloudera/Desktop/bigdataproj/output2011/part-r-00000 /user/cloudera/BDP/output2011/result2011

hdfs dfs -put /home/cloudera/Desktop/bigdataproj/output2012/part-r-00000 /user/cloudera/BDP/output2012/result2012

hdfs dfs -put /home/cloudera/Desktop/bigdataproj/output2013/part-r-00000 /user/cloudera/BDP/output2013/result2013

hdfs dfs -put /home/cloudera/Desktop/bigdataproj/output2014/part-r-00000 /user/cloudera/BDP/output2014/result2014

hdfs dfs -put /home/cloudera/Desktop/bigdataproj/air/part-r-00000 /user/cloudera/BDP/air

create external table topcitiesfor2010(Year int, City String , Rate float, TotalCrime int) row format delimited fields terminated by ',' location '/user/cloudera/BDP/output2010/';

create external table topcitiesfor2011(Year int, City String , Rate float, TotalCrime int) row format delimited fields terminated by ',' location '/user/cloudera/BDP/output2011/';

create external table topcitiesfor2012(Year int, City String , Rate float, TotalCrime int) row format delimited fields terminated by ',' location '/user/cloudera/BDP/output2012/';

create external table topcitiesfor2013(Year int, City String , Rate float, TotalCrime int) row format delimited fields terminated by ',' location '/user/cloudera/BDP/output2013/';

create external table topcitiesfor2014(Year int, City String , Rate float, TotalCrime int) row format delimited fields terminated by ',' location '/user/cloudera/BDP/output2014/';

hdfs dfs -put /home/cloudera/Desktop/bigdataproj/air.txt /user/cloudera/BDP/air/airdata

create external table air(Rank int, Year int, City String , status String) row format delimited fields terminated by ';' location '/user/cloudera/BDP/air/';

// create tables in hive from the stored output from pig

ALter table topcitiesfor2010 add columns (weight int);

ALter table topcitiesfor2011 add columns (weight int);

ALter table topcitiesfor2012 add columns (weight int);

ALter table topcitiesfor2013 add columns (weight int);

ALter table topcitiesfor2014 add columns (weight int);

update topcitiesfor2010 set weight= 1 where rownum <= 100 and rownum > 90;

update topcitiesfor2010 set weight= 2 where rownum <=90 and rownum > 80;

update topcitiesfor2010 set weight= 3 where rownum <=80 and rownum > 70;

update topcitiesfor2010 set weight= 4 where rownum <=70 and rownum > 60;

update topcitiesfor2010 set weight= 5 where rownum <=60 and rownum > 50;

update topcitiesfor2010 set weight= 6 where rownum <=50 and rownum > 40;

update topcitiesfor2010 set weight= 7 where rownum <=40 and rownum > 30;

update topcitiesfor2010 set weight= 8 where rownum <=30 and rownum > 20;

update topcitiesfor2010 set weight= 9 where rownum <=20 and rownum > 10;

update topcitiesfor2010 set weight= 10 where rownum <=10;

update topcitiesfor2011 set weight= 1 where rownum <= 100 and rownum > 90;

update topcitiesfor2011 set weight= 2 where rownum <=90 and rownum > 80;

update topcitiesfor2011 set weight= 3 where rownum <=80 and rownum > 70;

update topcitiesfor2011 set weight= 4 where rownum <=70 and rownum > 60;

update topcitiesfor2011 set weight= 5 where rownum <=60 and rownum > 50;

update topcitiesfor2011 set weight= 6 where rownum <=50 and rownum > 40;

update topcitiesfor2011 set weight= 7 where rownum <=40 and rownum > 30;

update topcitiesfor2011 set weight= 8 where rownum <=30 and rownum > 20;

update topcitiesfor2011 set weight= 9 where rownum <=20 and rownum > 10;

update topcitiesfor2011 set weight= 10 where rownum <=10;

update topcitiesfor2012 set weight= 1 where rownum <= 100 and rownum > 90;

update topcitiesfor2012 set weight= 2 where rownum <=90 and rownum > 80;

update topcitiesfor2012 set weight= 3 where rownum <=80 and rownum > 70;

update topcitiesfor2012 set weight= 4 where rownum <=70 and rownum > 60;

update topcitiesfor2012 set weight= 5 where rownum <=60 and rownum > 50;

update topcitiesfor2012 set weight= 6 where rownum <=50 and rownum > 40;

update topcitiesfor2012 set weight= 7 where rownum <=40 and rownum > 30;

update topcitiesfor2012 set weight= 8 where rownum <=30 and rownum > 20;

update topcitiesfor2012 set weight= 9 where rownum <=20 and rownum > 10;

update topcitiesfor2012 set weight= 10 where rownum <=10;

update topcitiesfor2013 set weight= 1 where rownum <= 100 and rownum > 90;

update topcitiesfor2013 set weight= 2 where rownum <=90 and rownum > 80;

update topcitiesfor2013 set weight= 3 where rownum <=80 and rownum > 70;

update topcitiesfor2013 set weight= 4 where rownum <=70 and rownum > 60;

update topcitiesfor2013 set weight= 5 where rownum <=60 and rownum > 50;

update topcitiesfor2013 set weight= 6 where rownum <=50 and rownum > 40;

update topcitiesfor2013 set weight= 7 where rownum <=40 and rownum > 30;

update topcitiesfor2013 set weight= 8 where rownum <=30 and rownum > 20;

update topcitiesfor2013 set weight= 9 where rownum <=20 and rownum > 10;

update topcitiesfor2013 set weight= 10 where rownum <=10;

update topcitiesfor2014 set weight= 1 where rownum <= 100 and rownum > 90;

update topcitiesfor2014 set weight= 2 where rownum <=90 and rownum > 80;

update topcitiesfor2014 set weight= 3 where rownum <=80 and rownum > 70;

update topcitiesfor2014 set weight= 4 where rownum <=70 and rownum > 60;

update topcitiesfor2014 set weight= 5 where rownum <=60 and rownum > 50;

update topcitiesfor2014 set weight= 6 where rownum <=50 and rownum > 40;

update topcitiesfor2014 set weight= 7 where rownum <=40 and rownum > 30;

update topcitiesfor2014 set weight= 8 where rownum <=30 and rownum > 20;

update topcitiesfor2014 set weight= 9 where rownum <=20 and rownum > 10;

update topcitiesfor2014 set weight= 10 where rownum <=10;

// assign weights to the cities

step 6:

create table topcities as

select topcitiesfor2014.city , topcitiesfor2010.weight + topcitiesfor2011.weight + topcitiesfor2012.weight + topcitiesfor2013.weight + topcitiesfor2014.weight + rank as [TotalWeight]

from topcitiesfor2010 join topcitiesfor2011 on topcitiesfor2010.city = topcitiesfor2011.city

join topcitiesfor2012 on topcitiesfor2010.city = topcitiesfor2012.city

join topcitiesfor2013 on topcitiesfor2010.city = topcitiesfor2013.city

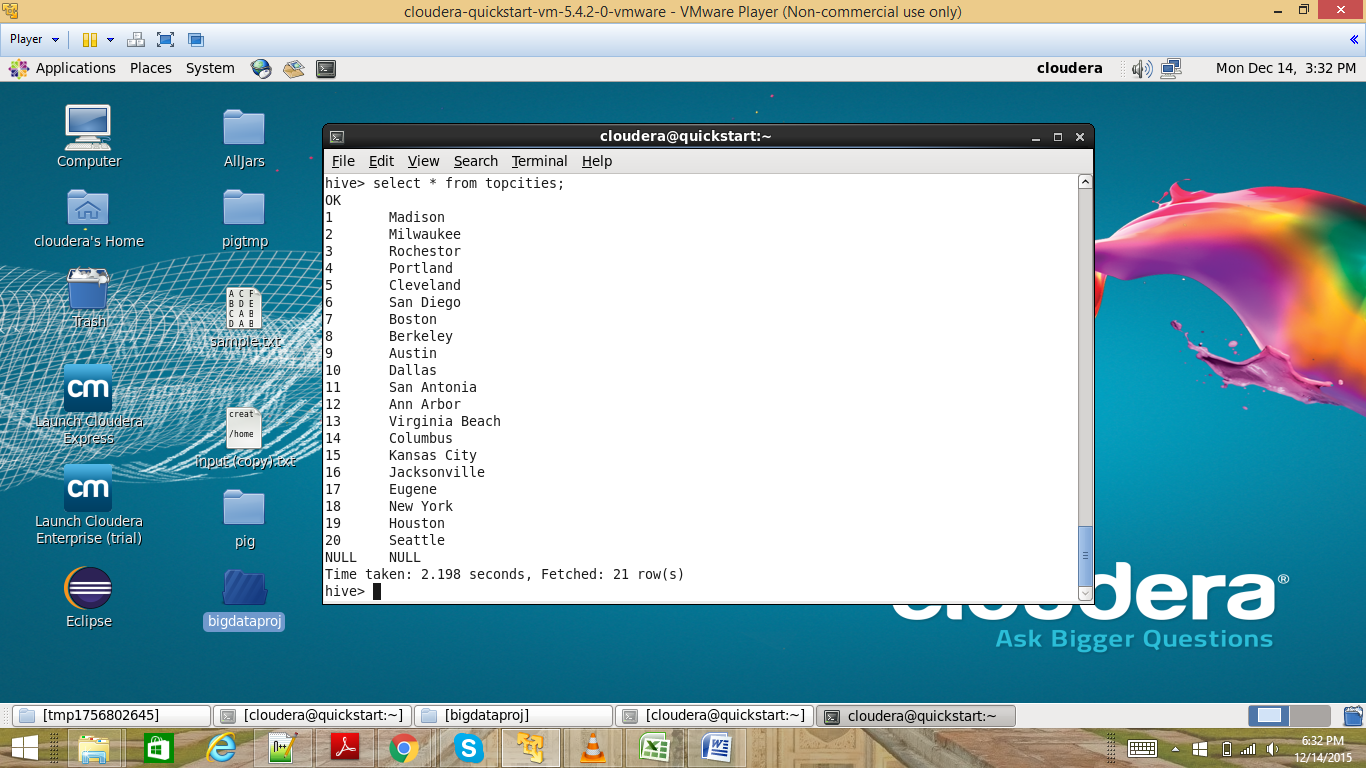
join topcitiesfor2014 on topcitiesfor2010.city = topcitiesfor2014.city

join air on topcitiesfor2010.city = air.city

WHERE air.status == "clean"

ORDER BY TotalWeight DESC;

**Screenshot:**

****