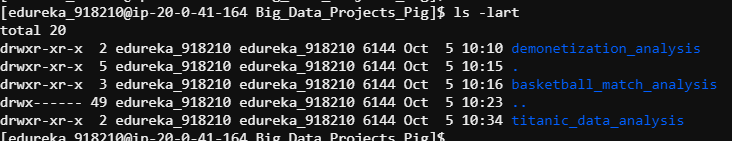
**TITANIC ANALYSIS**



titanic\_data = load '/mnt/home/edureka\_918210/Big\_Data\_Projects\_Pig/titanic\_data\_analysis/TitanicData.txt' using PigStorage(',') as (PassengerId:int,Survived:int,Pclass:int,Name:chararray,Sex:chararray,Age:int,SibSp:int,Parch:int,Ticket:chararray,Fare:int,Cabin:chararray,Embarked:chararray);

feature\_data = foreach titanic\_data generate Survived,Sex,Age;

SPLIT feature\_data INTO died\_Passanger IF(Survived==0),survived\_Passanger IF(Survived==1);

survived\_gender\_grp = GROUP survived\_Passanger by Sex;

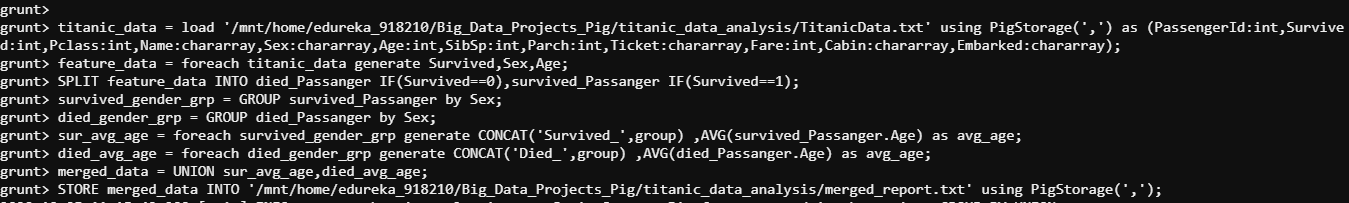
died\_gender\_grp = GROUP died\_Passanger by Sex;

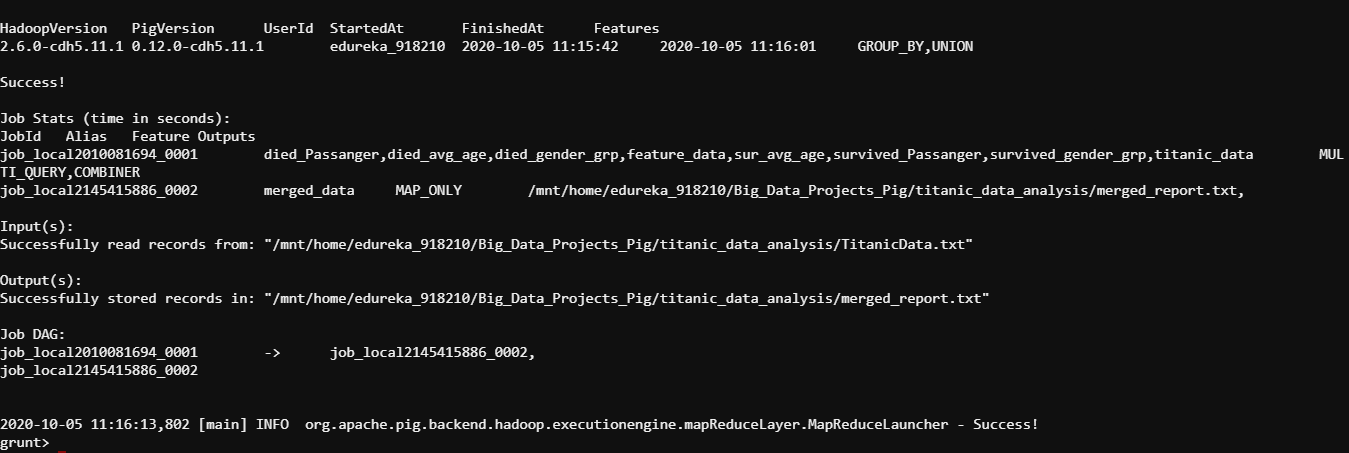
sur\_avg\_age = foreach survived\_gender\_grp generate CONCAT('Survived\_',group) ,AVG(survived\_Passanger.Age) as avg\_age;

died\_avg\_age = foreach died\_gender\_grp generate CONCAT('Died\_',group) ,AVG(died\_Passanger.Age) as avg\_age;

merged\_data = UNION sur\_avg\_age,died\_avg\_age;

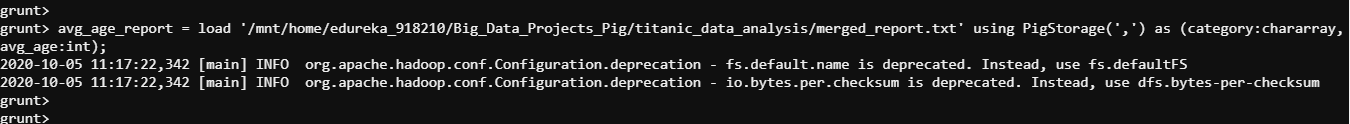
STORE merged\_data INTO '/mnt/home/edureka\_918210/Big\_Data\_Projects\_Pig/titanic\_data\_analysis/merged\_report.txt' using PigStorage(',');

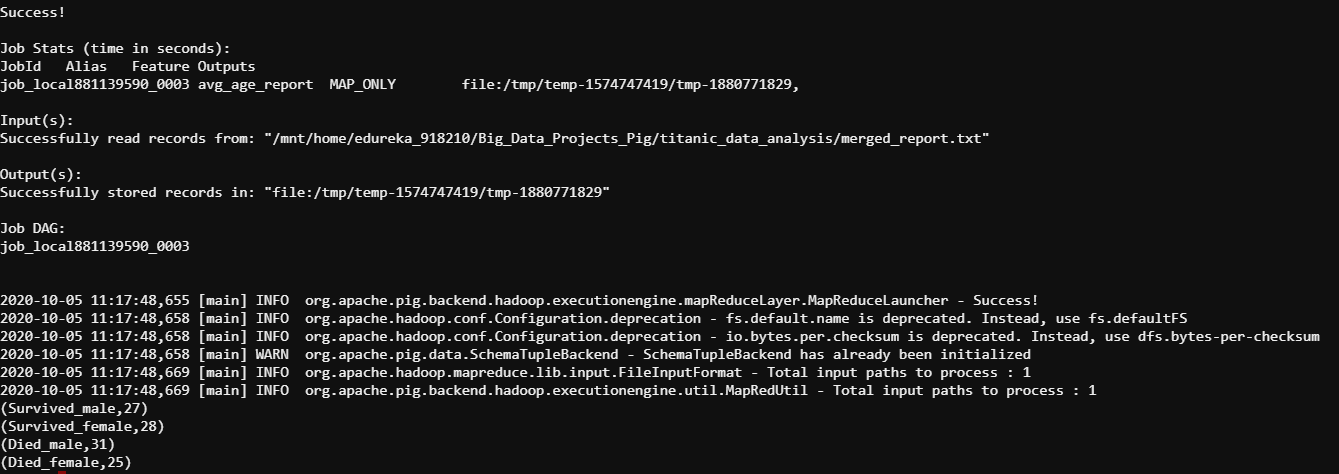




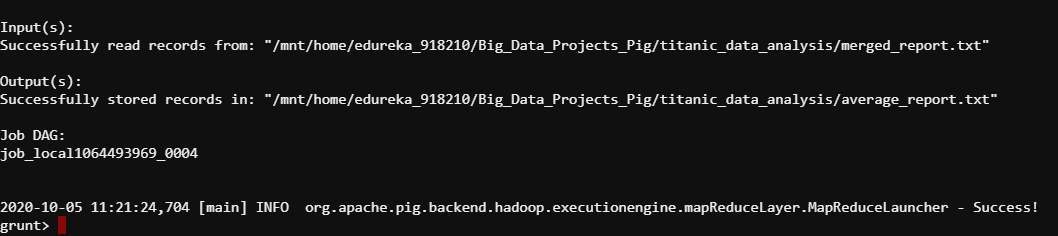
avg\_age\_report = load '/mnt/home/edureka\_918210/Big\_Data\_Projects\_Pig/titanic\_data\_analysis/merged\_report.txt' using PigStorage(',') as (category:chararray,avg\_age:int);

dump avg\_age\_report;

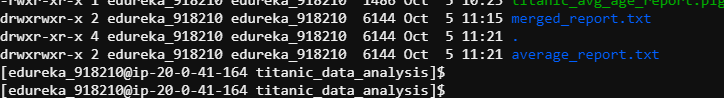




STORE avg\_age\_report INTO '/mnt/home/edureka\_918210/Big\_Data\_Projects\_Pig/titanic\_data\_analysis/average\_report.txt' using PigStorage(',');



**RESULTS**



**DEMONITIZATION ANALYSIS**

raw\_data = load '/mnt/home/edureka\_918210/Big\_Data\_Projects\_Pig/demonetization\_analysis/demonetization-tweets.csv' using PigStorage(',');

get\_details = foreach raw\_data generate $0 as id,$1 as text;

tokens = foreach get\_details generate id,text,FLATTEN(TOKENIZE(text)) as words;

dictionary = load '/mnt/home/edureka\_918210/Big\_Data\_Projects\_Pig/demonetization\_analysis/AFINN.txt' using PigStorage('\t') as (word:chararray,rating:int);

word\_ratings = join tokens by words left outer, dictionary by word using 'replicated';

describe word\_ratings;

ratings = foreach word\_ratings generate tokens::id as id,tokens::text as text,dictionary::rating as rating;

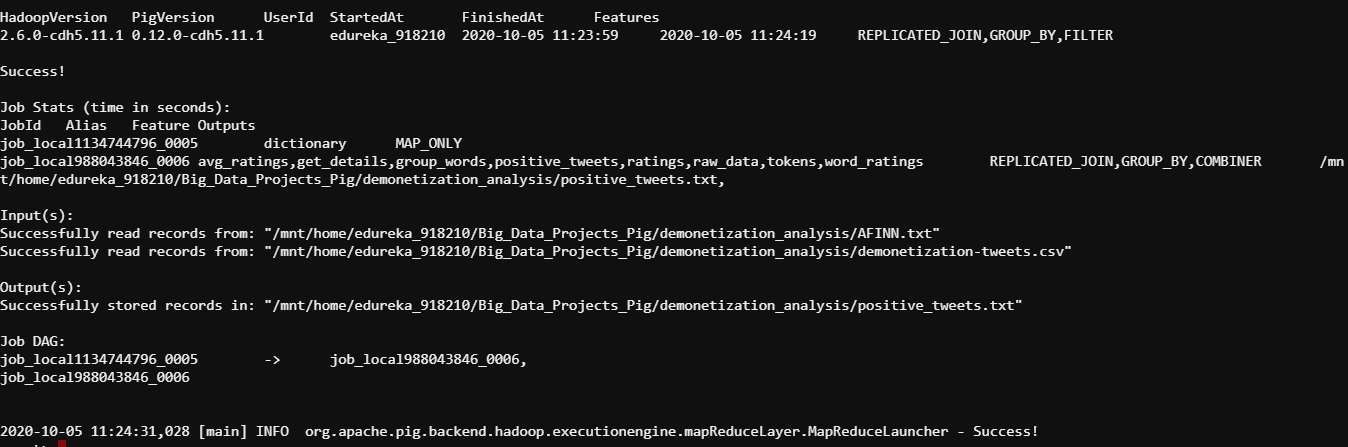
group\_words = group ratings by (id,text);

avg\_ratings = foreach group\_words generate group,AVG(ratings.rating) as tweet\_rating;

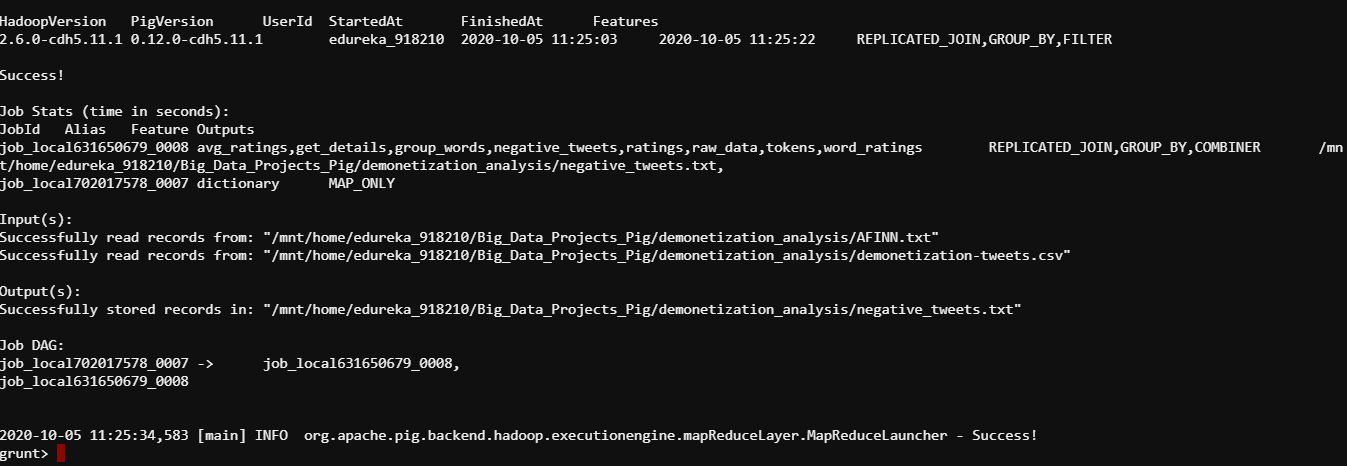
positive\_tweets = filter avg\_ratings by tweet\_rating > 0;

negative\_tweets = filter avg\_ratings by tweet\_rating < 0;

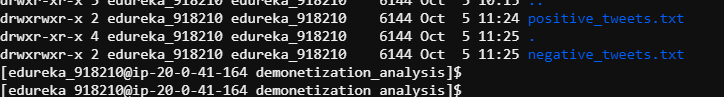
store positive\_tweets INTO '/mnt/home/edureka\_918210/Big\_Data\_Projects\_Pig/demonetization\_analysis/positive\_tweets.txt' using PigStorage(',');



store negative\_tweets INTO '/mnt/home/edureka\_918210/Big\_Data\_Projects\_Pig/demonetization\_analysis/negative\_tweets.txt' using PigStorage(',');



**RESULTS**



**BASKETBALL ANALYSIS**

**id\_to\_name**

raw\_data = LOAD '/mnt/home/edureka\_918210/Big\_Data\_Projects\_Pig/basketball\_match\_analysis/gamedata/\*.EV{A,N}' using PigStorage(',') as (type:chararray,id:chararray,name:chararray);

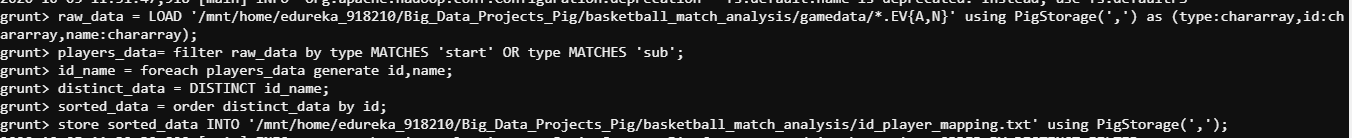
players\_data= filter raw\_data by type MATCHES 'start' OR type MATCHES 'sub';

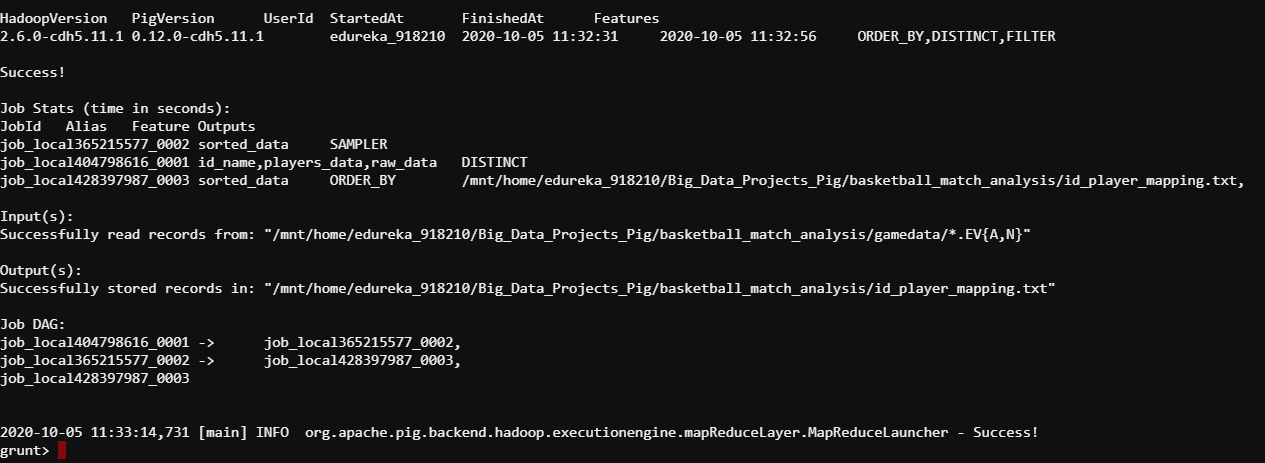
id\_name = foreach players\_data generate id,name;

distinct\_data = DISTINCT id\_name;

sorted\_data = order distinct\_data by id;

store sorted\_data INTO '/mnt/home/edureka\_918210/Big\_Data\_Projects\_Pig/basketball\_match\_analysis/id\_player\_mapping.txt' using PigStorage(',');





**total\_counts**

all\_data = load '/mnt/home/edureka\_918210/Big\_Data\_Projects\_Pig/basketball\_match\_analysis/gamedata/\*EV{A,N}' using PigStorage(',') as (teams:chararray);

al\_data = load '/mnt/home/edureka\_918210/Big\_Data\_Projects\_Pig/basketball\_match\_analysis/gamedata/\*EVA' using PigStorage(',') as (team:chararray);

nl\_data = load '/mnt/home/edureka\_918210/Big\_Data\_Projects\_Pig/basketball\_match\_analysis/gamedata/\*EVN' using PigStorage(',') as (team:chararray);

all\_data\_ids = filter all\_data by teams MATCHES 'id';

al\_data\_ids = filter al\_data by team MATCHES 'id';

nl\_data\_ids = filter nl\_data by team MATCHES 'id';

all\_id\_grp = group all\_data\_ids ALL;

al\_id\_grp = group al\_data\_ids ALL;

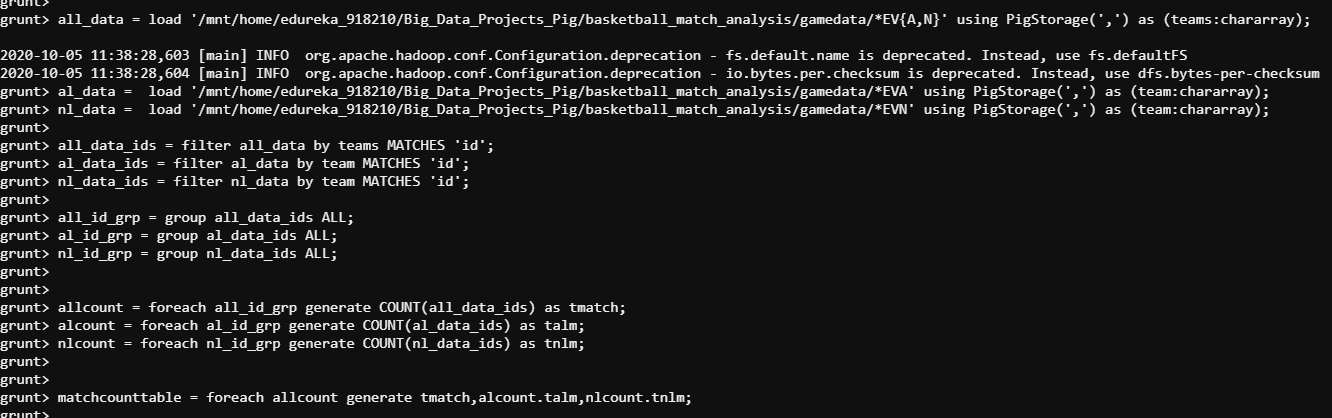
nl\_id\_grp = group nl\_data\_ids ALL;

allcount = foreach all\_id\_grp generate COUNT(all\_data\_ids) as tmatch;

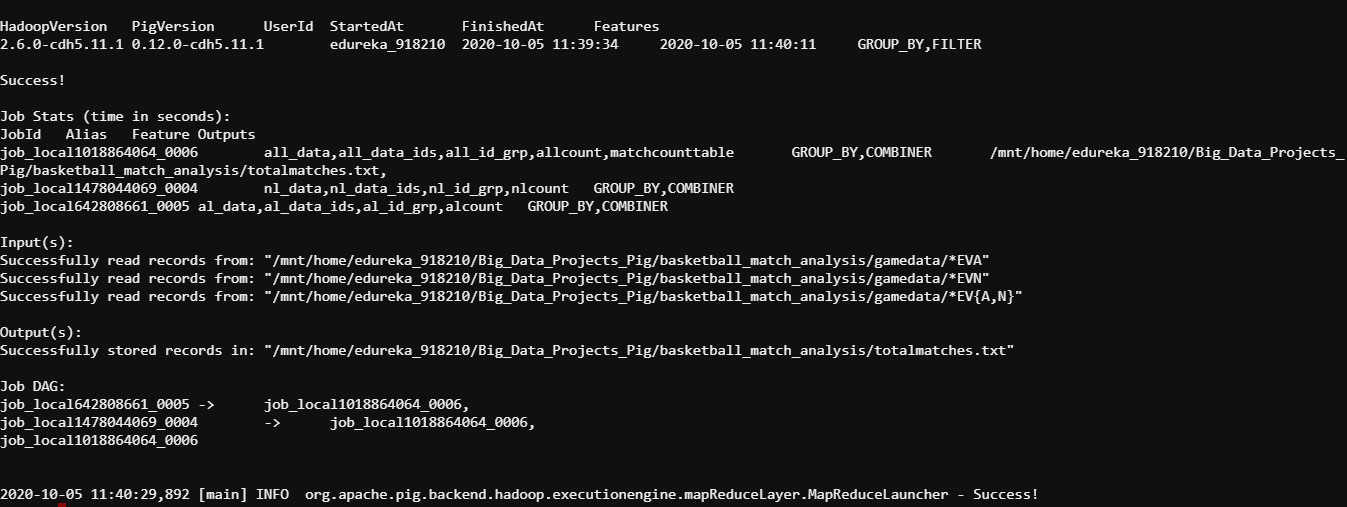
alcount = foreach al\_id\_grp generate COUNT(al\_data\_ids) as talm;

nlcount = foreach nl\_id\_grp generate COUNT(nl\_data\_ids) as tnlm;

matchcounttable = foreach allcount generate tmatch,alcount.talm,nlcount.tnlm;



store matchcounttable INTO '/mnt/home/edureka\_918210/Big\_Data\_Projects\_Pig/basketball\_match\_analysis/totalmatches.txt' using PigStorage(',');



dump matchcounttable;

