**SESSION 10-11: PROJECT I**

-- Common commands for all problems

ratingsData = LOAD 'ratings.dat' AS (line:chararray);

Tratings = FOREACH ratingsData GENERATE

FLATTEN(REGEX\_EXTRACT\_ALL(line,'(.\*?)::(.\*?)::(.\*?)::(.\*?)')) AS (uid:chararray, mid:chararray, rating:chararray, timestamp:chararray);

moviesData = LOAD 'movies.dat' AS (line:chararray);

Tmovies = FOREACH moviesData GENERATE

FLATTEN(REGEX\_EXTRACT\_ALL(line,'(.\*?)::(.\*?)::(.\*?)')) AS (mid:chararray, title:chararray, genres:chararray);

usersData = LOAD ‘users.dat’ AS (line:chararray);

Tusers = FOREACH usersData GENERATE

FLATTEN(REGEX\_EXTRACT\_ALL(line,’(.\*?)::(.\*?)::(.\*?)::(.\*?)::(.\*?)’)) AS (uid:chararray, gender:chararray, age:chararray, ocp:chararray, zip:chararray);

1. **Top 10 most viewed movies with their movies name (Ascending or Descending order).**

ratings = FOREACH Tratings GENERATE (int)uid, (int)mid, (int)rating, (long)timestamp;

movies = FOREACH Tmovies GENERATE (int)mid, title, genres;

grouped\_mid = GROUP ratings BY mid;

watch\_count = FOREACH grouped\_mid GENERATE group, COUNT(ratings.uid) as count;

sorted\_watch\_count = ORDER watch\_count BY count DESC;

top\_viewed\_movies = LIMIT sorted\_watch\_count 10;

Tresult = JOIN top\_viewed\_movies BY group, movies BY mid;

result = FOREACH Tresult GENERATE movies::mid, movies::title;

1. **Top twenty rated movies (Condition: The movie should be rated/viewed by at least 40 users)**

ratings = FOREACH Tratings GENERATE (int)uid, (int)mid, (double)rating, (long)timestamp;

movies = FOREACH Tmovies GENERATE (int)mid, title, genres;

grouped\_mid = GROUP ratings BY mid;

cal\_data = FOREACH grouped\_mid GENERATE group, COUNT(ratings.uid) as count, AVG(ratings.rating) as avgrating;

filter\_data = FILTER cal\_data BY count>=40;

sorted\_filter\_data = ORDER filter\_data BY avgrating DESC;

top\_rated\_movies = LIMIT sorted\_filter\_data 20;

Tresult = JOIN top\_rated\_movies BY group, movies BY mid;

result = FOREACH Tresult GENERATE movies::mid, movies::title;

1. **Top twenty rated movies (which is calculated in the previous step) with the number of views in the following age group**
2. **Young (<20 years)**
3. **Young adults (20-40 years)**
4. **Adults (>40 years)**

-- To find no of views of all movies in given age groups

ratings = FOREACH Tratings GENERATE (int)uid, (int)mid;

users = FOREACH Tusers GENERATE (int)uid, (int)age;

tviewers = JOIN ratings BY uid, users BY uid;

viewers = FOREACH tviewers GENERATE ratings::uid as uid, ratings::mid as mid, users::age as

age;

SPLIT viewers INTO

yviewers IF age<20,

yaviewers IF age>=20 AND age<=40,

aviewers IF age>40;

group\_yviewers = GROUP yviewers BY mid;

young\_views = FOREACH group\_yviewers GENERATE group as mid, COUNT(yviewers.uid) as

views;

group\_yaviewers = GROUP yaviewers BY mid;

youngadult\_views = FOREACH group\_yaviewers GENERATE group as mid,

COUNT(yaviewers.uid) as views;

group\_aviewers = GROUP aviewers BY mid;

adult\_views = FOREACH group\_aviewers GENERATE group as mid, COUNT(aviewers.uid) as

views;

-- To find top 20 rated movies

dratings = FOREACH Tratings GENERATE (int)uid, (int)mid, (double)rating, (long)timestamp;

movies = FOREACH Tmovies GENERATE (int)mid, title, genres;

grouped\_ratings = GROUP dratings BY mid;

cal\_data = FOREACH grouped\_ratings GENERATE group, COUNT(dratings.uid) as count,

AVG(dratings.rating) as avgrating;

filter\_data = FILTER cal\_data BY count>=40;

sorted\_filter\_data = ORDER filter\_data BY avgrating DESC;

top\_rated\_movies = LIMIT sorted\_filter\_data 20;

Tresult = JOIN top\_rated\_movies BY group, movies BY mid;

result = FOREACH Tresult GENERATE movies::mid as mid, movies:title as title;

-- To find number of views of top 20 rated movies in the given age groups

top\_rated\_movies\_views = JOIN result BY mid, young\_views BY mid, youngadult\_views BY mid,

adult\_views BY mid;

FinalResult = FOREACH top\_rated\_movies\_views GENERATE result::mid as mid, result::title as

title, young\_views::views as youngViewCount, youngadult\_views::views as

youngadultViewCount, adult\_views::views as adultViewCount;;

1. **Top ten critics (users who have given very low ratings; Condition: The users should have at least rated 40 movies)**

ratings = FOREACH Tratings GENERATE (int)uid, (int)mid, (double)rating;

grouped\_ratings = GROUP ratings BY uid;

cal\_data = FOREACH grouped\_ratings GENERATE group as uid, COUNT(ratings.mid) as count,

AVG(ratings.rating) as avgrating;

filter\_data = FILTER cal\_data BY count>=40;

sorted\_filter\_data = ORDER filter\_data BY avgrating ASC;

top\_critics = LIMIT sorted\_filter\_data 10;

dump top\_critics;