SI 618 Fall 2022 Lab 7 - SparkSQL

In this lab, we will practice SparkSQL.

We will be diving into a dataset of League of Legends matches and individual performance in the North American Region. This open dataset was created as part of Lee, C. S., & Ramler, I. (2017, August).

The datafile to be used in this lab can be found at

/scratch/siads618f22_class_root/siads618f22_class/shared_data/lab7_data/euw_ranked_team.csv

LoL is a multiplayer online battle arena game where two teams of 5 (or 3) players battle each other on a map called an arena and try to destroy each other's base (or Nexus) while protecting their own. Each player can choose from among over 100 heros (or champions) with different skills and play different roles (attack, support, top, bottom, jungle).

League of Legends Arena

Dataset description

This dataset includes player behavior data for a total of 130,988 ranked 5 vs. 5 matches in Western Europe during the period from December 4, 2014, to November 11, 2015, corresponding to 1,309,880 rows (no. of matches x 10 players per match) and 452,567 players. The dataset contains 80 columns providing a great deal of information about the behavior of players during each match (see the

ReadMe.txt file for description for all columns), but we will only be using a few of them in this lab.

Question 1: (20 points)

Load the csv file from
/scratch/siads618f22_class_root/siads618f22_class/shared_data/lab7_data/euw_ranked_team.csv
and register it as a table/view.

Note: If your sparksql instance reads SummonerId/predictedRole/kill/death/assist as a string, please cast them into integer values.

Cases of high player dominance (using a Dominance Factor)

In LoL, DF or the Dominance Factor is a popular measure of individual performance. A player scores a ***kill*** whenever they deal the final blow that defeats another player of the opposing team in combat and the opposing player suffers a ***death***. Since LoL is a team game, a third player of the same team as the first player may ***assist*** them in defeating the second player. Based on these actions, the dominance factor (DF) is a simple sum of "points", where kills count as 2, deaths count as -3, and assists are 1.
DF = 2*kills + assists - 3*deaths.
For this task, calculate the DF for each player for each match and display them along with the number of matches played by each player for players who have played at least 10 matches. The output should be sorted by descending order of the DF and ties resolved by the descending order of the number of matches played by the player. Further ties are resolved by the ascending orders of summonerId and matchId.

Eg.
As an example, if a player X had played 2 matches and had the following performance statistics,

Match 1 3 kills 1 death 1 assist
Match 2 1 kills 3 deaths 1 assists

Their match 1 DF would be 2*3 + 1 – 3*1 = 4 and match 2 KDA would be 2*1+ 1 – 3*3 = -6. The output for these two lines should be

In the input, the column **summonerId** is a unique identifier for a player. Your output should have the format
summonerId<tab>matchId<tab>DF<tab>number of matches of the player

Save it as UNIQUENAME_si618_lab7_output_1.csv. Your output should match si618_lab7_desired_output_1.csv.

Question 2: (30 points)

While DF is a reasonable indicator of player performance in a match, we can do better. A player's performance during a single match may be affected by the role they play, the champion they select, their experience, and the skill levels of the two teams.

Dominance levels for players (using a normalized Dominance Factor)

Let's try to improve the DF measure by taking the player's team into account and try to bin users into levels between 5 and -5 (approximately). Calculate a new adjusted DF for each player's performance during a match by normalizing their match DF by the absolute total match DF of their own team as follows: (player DF)/(absolute value of total team DF). In those cases where the denominator value is 0, replace them by 1 to avoid the division_by_zero error.

Now calculate the average normalized DF per match for each player who has played at least 10 matches across all matches. Round UP this average normalized DF to the closest integer.

Note that the ***winner*** column can be used to identify the team to which a player belonged in a match. The value 1 corresponds to the winning team and 0 corresponds to the losing team. Also, when you calculate the team DFs for a match, you should consider all players in a team irrespective of the total number of matches they have played.

Your output should be in the form of
summonerId<tab>average normalized DF<tab>number of matches of the player

The values are sorted in the descending order of the average normalized DF and then by the decreasing order of the number of matches played by the player. Further ties are resolved by ascending order of the summonerId.

Save it as UNIQUENAME si618_lab7_output_2.csv. Your output should match si618_lab7_desired_output_2.csv.

Question 3: (50 points)

In LoL, players select roles and champions based on community knowledge about which of them are more appropriate as choices to battle the opposing team (in ranked games teams pick champions in stages).

Roles in matches with greatest differences in Dominance Factor

Let's try to discover which role pairs in matches from the two opposing teams have the largest difference in the Dominance Factor.

For each match, get the pairs of opposing players who played in the same role but had the highest absolute difference in DF between them. If there is more than one role in a match that shares the highest absolute difference in the DF for the summoners who opposed each other in the same role, you must include those as well.

The column **predictedRole** specifies the role (0-top,1-jungle,2-middle,3-attack damage carry,4-support). For this question, you can write two separate SQL queries to get the output, but we encourage you to write a single query with multiple nested queries.

The output should contain rows of the form
matchID<tab>common_role<tab>Absolute difference of DF<tab>SummonerID1<tab> SummonerID2

The output is sorted in the decreasing order of the absolute difference of DF and then the further ties are resolved by the ascending order of role, matchId, summonerId1 and summonerId2 respectively.

Save it as si618_lab7_output_3_UNIQUENAME.csv. Your output should match si618_lab7_desired_output_3.csv.

You MUST use **SparkSQL** to do this lab. Other solutions will not get any credit.

What to submit:

Submit your Python source code file:uniquename_si618_lab6.py, batch job file: uniqname_batch-job.sh and the 3 csv files. Please submit these as separate files and not as a zipped folder (This helps the IAs to grade more efficiently)