Big Data Analytics

Mini Project 2

Mohamed Abdelwahed 201801978

Ibrahim Abdalaal 201800224

Data Analysis

The data is available for public access through (https://developer.riotgames.com/apis), so for receiving data, we used python requests library to access the api.

We have to get through multiple steps to collect matches data.

- First, we prepare the data by choosing the region from [‘EUW1’], then we choose all leaguers with different leagues ['challengerleagues', 'grandmasterleagues', 'masterleagues', 'DIAMOND', 'PLATINUM', 'GOLD', 'SILVER', 'BRONZE', 'IRON'] with different distribution based on (https://www.leagueofgraphs.com/rankings/rank-distribution), that will be the summoner file we built.
- Second, we will get the global ID which we can use which is PUUIDs by querying the summoners API. After that, we will get the match ids using PUUIDs which get us nearly 76 thousand. We choose to take the unique values which yield us nearly 75 thousand.
- Third, we use the match ids to get the matches themselves. This step takes a long time to receive data and ensure the api limit is not exceeded.

For using the data efficiently and cleanly, we used information from datadragon to help us identify some things. This is mainly used in getting the champion name by using the champion id, getting the item name by using item id, and getting the champion class using the champion name. Item name and classes were not included in the API.

Challenges and optimization:

- The biggest challenge was getting the data from the API was a very slow process because of the rate limit. For solving this problem, we choose to run the data analysis on a segment from the dataset before we use the whole dataset in the final.
- Some item id were invalid because they were zero or some other ids that were not present in the datadragon, so we used filters to remove them to better utilize the mapping without any problems. Additionally, some champion id were negative, so we used filters to remove them.

Data analysis

The analysis uses Apache spark on python (PySpark) and google colab.

1- requirement 1: champion win, pick and ban rates.

We start by using matches. After that, we count the number of each champion name for picks and for calculating the rate will be as

pick rate= number of champions picked/ total number of all champions

For wins, we count the number of champions when the win value is true and calculate the win rate as

Win rate= number of champion wins/ number of champions picked

For bans, we count the number of bans for every champion id, which is present in the bans list, and calculate the rate as below.

Ban rate= number of bans/ total number of games

2- Requirement 2: champion synergy

The synergy is calculated based on if there are two champions in the same team who have a higher winning. So, we calculate the number of champion pairs being in the winning team over the total number of the champions pair being in any team(equation). Grouping the pair is done by first selecting only the winning participants, and only the winning participants with the same match id, do we provide different unique combinations for. Our calculation is done by grouping and calculating the number of every champion name pair from the same match id if win is true and the same pairs with filtering by using the boolean variable win.

We provide synergy value based on the equation above and multiply it by 100%.

3-Requirement 3: item win and pick rates:

From the participant dictionary, we group by each item and then we calculate the item total and we also calculate the same value when the variable win equals true. And then we calculate the win rate=total for every item win/total pick for every item

Pick rate=total for every item pick/total items.

Then we map the item id from the extracted data from the datadragon we mapping the item id to item name.

4- Requirement 4: item synergy

For calculating the item-champion synergy, For every champion when the win value is true, we get the items and we aggregate its counts, and we also calculate the item count when the win value is removed. So, for calculating the synergy, we get the value of total items when win is true/total count for every item.

For getting item-class synergy, we do the exact same as above, and then we map the champion name to its class from dragondata.

5- Requirement 5: item suggestion:

For item suggestion, we provided the items for 2 champions from different classes based on their item synergy from the previous task.

6- Extra requirement: lane and champion synergy(best lane for every champion)

For calculating the lane-champion synergy, For every champion when the win value is true, we get the lanes and we aggregate its counts, and we also calculate the lane count when the win value is removed.

7-extra requirement: regular used maps

For calculating the regular used maps, we count the maps used in every game to determine the regular used maps by the players.