Massive Data Processing - Big Data Project

Adrià Bonet Vidal
Sergio Salcedo Heredia

June 2019

Contents

1 Introduction

In this activity we needed to think about a project where we can gather data from an API or different resources and then process the data to have an uniform data and end up taking some conclusions or answering some questions.

In our case, we like a lot the game League of Legends of Riot Games company so we decided to make a project that involved this game.

League of legends is a MOBA game (Multiplayer online battle arena), where 2 teams of 5 play one versus the other one. Every game, all players have to choose a character to play, those characters commonly known as champions. In the game there are a lot of different functionalities and factors, but the main objective is to destroy the base of the other team, by killing 'monsters' and destroying the turrets of the other team.

We have searched for different datasets in order to think some questions that could be useful or interesting for us, the players. There's a lot of sites that provides you datasets, but at the end we extracted the data directly from Riot Games API, that gives us the data that can be trusted the most. Now, with the data our objective is to make some investigations and studies of the datasets and to answer general questions like 'What's the champion with a most win rate? And the one with less?', 'At what time a champion has more probability to win?, etc.

Also, this project is the first part of an even bigger project, that will be continued next year with machine learning. Our main objectives are to be able to predict predict what team will win given some circumstances like 2 teams with certain champion picks (we would be able to see their synergy in numbers and percentages), the relation with the hour of the day you are playing, the champion used, and if you won or lost, etc.

All the code used to make this first part can be found at the following Github repository:

Github Code

2 Project structure

In order to develop this project we made 3 different scripts with python in order to make the data gathering, the data cleanup and finally the data processing.

The first code is a simple python script that have the function of gathering the information. It can be found at */src/data/datagather.py* on the repository. Just typing "python datagather.py" is enough to execute it, but there are some restrictions. The requisites to execute this script are to have an account of the game and also on the Riot Games api website, the account of the game will have an ID associated and the api account will generate a token, both of them are necessary in order to execute the script to gather data. This token can be utilized by 24h before expires and you have to ask for another one. In order to take our ID it's necessary to call an specific method of the api using your game username.
The api has some restrictions, like the number of petitions per second, so it's because of this there's an sleep method for every petition to the api. Making more petitions than your limits, even if returns an specific http code, it may result in a temporal block.
The API does not provide a method that gives you a range of games, so what we did is to take the id of one of our last games and increase this id for every loop in order to take other posterior games. The limit of 1100 specified in the loop is to have different types of files (in size), so we can make different tests.

The api can provide us a lot of different data, like descriptions of the players, sales, etc, but the data we are getting is just the data of games. This data gives an immense quantity of details of every game. The most important information we are going to use is basically the date when the game started, the duration, the information of all the players (including the champion the chose and their statistics) and the teams.

For the data of the champions, the api does not provide any information but in the documentation its specified as static data, so we took that file with others that may be useful in a future. This file has not been updated for months, so in the notebook of the cleanup we filled all the remaining champions with its correspondent data.

After the gathering we had 2 different jupyter notebooks (the notebooks have been done in Google Collab, that provides the same functionality, due to the pirgi cluster that was not working properly with spark environment during the development of the project). Those notebooks can be found at */notebooks/*

The first notebook (the cleanup notebook), *LolNotebook* read the data previous extracted and cleaned it. This was done by using pandas. The data extracted has been so clear and it hasn't any type of mistake or problems since

the company checks and uses this data before publishing it. About the cleanup, we have filtered the data of the games by its type (traditional game mode) and also we removed the columns we didn't and we are not going to use. The filling of the remaining champions is also done here, the missing data is the basic one and can be found at the main page of the League of Legends game. At the end of the notebook there are also some queries, without spark ar hadoop, to check some statistics

The second notebook, *lolAnswers* it's the one in charge of processing this data. We decided to do it using pyspark. In this notebook we take all the data parsed in the cleanup notebook, we make some transformations in order to facilitate the following work and we make some statistics and plots, so we can play with our data and take some conclusions. Some of the statistics we have done, together with some plots are about the champion that is most played, the win rate of every champion, which team has more probability to win, etc. Some of them can be improved taking into account also the skill level of the player (comparing professional players with others of the same level or the same with the newer players). All the plots and results of the statistics made can be found in the notebook.

3 Test files

In order to test the notebooks there's some files in the data folder of the repository. To test the cleanup notebook (LolNotebook) it's necessary to use the raw data, that is placed on the raw folder, the necessary files are the data2 and the champions files.
To test the last notebook, if you executed the last notebook you will have already the parsed files *game* and *championsComplete*, if not, the files will be accessible in the processed folder, inside /data on the repository.

The script to gather data is not executable unless the requisites explained in the document are accomplished.

Finally, in the static folder, also inside the data path there's various files that may be useful in the future, but are not processed (there's also the raw champions.json, but in order to let others to test the notebooks easier we put it together also with the raw data)