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 / answering some questions.

In our case, we like a lot the game League of Legends of Riot Games company so we decided to take 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 player choose a different character (a champion) every game.
We searched for different datasets in order to think some questions that could be useful or interesting for us, the players. Finally we end up extracting the data directly from Riot Games API, even though it does not provide optimal methods, and answering general questions like: 'What's the champion with a most winrate? And the one with less?'.

Also, this project had the need of being able to have machine learning for the next course, and we want to be able to predict what team will win, given some circumstances like 2 teams with certain champion picks, the hour of the day when the game is done, etc.
All the code can be found at:

Github Code

2 Project structure

In order to develop this project we made 3 different scripts with python. The first one is a simple python script that have the function of gathering the information, it can be found at /src/data/datagather.py. To execute it, just type "python datagather.py". The requisites to execute this script is to have an account of the game, that will have an ID associated and a token that is generated also by Riot Games. 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 already to call a method of the api using your 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.
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 games. The limit of 1100 in the loop is to have different types of files (in size), so we can make faster tests.

Then we created 2 different jupyter notebooks(but we worked with Google Collab, since we had problems with UdL tunnels and local installing environment). Those notebooks can be found at /notebooks/
The first notebook, LolNotebook readed the data previous extracted and cleaned it. This was done by using pandas. The data extracted was so clear and without any mistakes or problems since the company use this data before publishing it, it seems they do some cleanup, or just the extract step it's well done. Even though, we have filtered some of the data like the type of games and descarted some other, like the columns we didn't and we are not going to use. Also, for the informations of the champions (playable characters in the game), the API does not provide any information, but there's an static data file on the documentation, so we have used it. 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.

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 other notebook, we make some transformations and we make some statistics and plots, so we can play with our data.