Operation Guide

Alberto Arienti - Sofia Turroni

Tennis Project

1 Data Acquisition

The site https://www.flashscore.com/ provides the results of the SLAM tournaments for the current year and also for the past years. The libraries needed in python 3 to perform the codes are selenium, webdrive and its extensions to navigate through dinaimic we pages, as well as json to create the documents in JSON format. So through the selenium library in python we performed a web scraping by getting data for the four SLAM tournaments (AUS Open, Rolland Garros, Wimbledon, US Open) from the year 2014 to the current year 2014. This by creating a JSON document for each of the torunament, having the years as the primary key, and each year is associated with a lists of disctionaries representing the matches, from the 64 to the final. Each match contains the following information:

- Match type (fial, semi-final, quarter-final. . .)
- Players: identified as player1 and player2
- Sets: for every player a list containing the games won in each set
- Winner: a string containing he winner name

Then, we have used information retreived by the web pages of https://www.atptour.com/. We performed different tasks of scraping, firstly to get the ranking of the top 100 players at the beginning of the month in which each tournament has been played for every year, resulting so in 4 different JSON documents, which of one contains the name of the player and the position in the ranking. Finally, to get the overall information about each player it's needed to perform two different scarping processes: the first one aim to get all the specific link of atptour.com associated to each player, then using a for cycle, scrape the following information from each specific page:

- Name
- Year of birth
- Height (in cm)

- Weight (in kg)
- Year in which the player turned pro
- Hand (right-handed or left-handed)
- Backhand (two-handed or one-handed)

2 Conflict Resolution

As deriving from two different sources, there are some conflicts in how the names of the players are registered: in the data deriving from flashscore.com the names are collected as 'Surname N.' (Sinner J.), while in data retrieved from atptour.com the names are stores as 'Name Surname'(Jannik Sinner). This conflict has been solved by introducing a resoluting function (can be find in the python file called Function conflict resolution), named *converti*, which takes as input the collection of dictionaries that need to be converted and the collection which have the full names. This function perform a comaprison between the surnames, taking into account taht they could be composed by one, two or three different names, and add a new value inside of each document having the full name, named *Name Formatted*, which will contain the name to match with the other documents.

3 Schema Transformation and Enrichment

As we scraped the data, there is the need to merge them into a single database, and we had the aim of creating four different collections, one for each tournament, where each document referred to a single match, and that document contained both information of the match and the players involved. We so needed firstly insert inside of MongoDB the documents of the match, by eliminating the primarly key we had in the row data of year and inserting them as a value inside of each document, and the same we will do regarding the Rank documents. After that, we performed firstly the integration between the rankings and the players overview, adding to the ranking documents the overview of players, and then we inputed each document from ranking into the value of *player1* or *player2*, using as matches the formatted name and the year in which the match is been played.
For the steps of modifying and creating documents and collection inside of MongoDB we used the functions offered in the library *pymongo*, and in particular *MongoClient* that connect us with our local MongoDB.

4 Data Querying

For all the steps performed during queriyng data it's always needed to use the *pymongo* and *MongoClient* librares, so that we can connect with our database, and using CRUD operations to develop queries and retrieve data we need to make visualizations. In add, we will need the library *matplotlib.pyplot* to develop the graphs we made, as when retreiving data, for example by calculating how many matches a player played, we will temporarly store those information in dictionaries, and this library is a tool to extract data and plot values of dictionaries.