University of Liège

Big Data Project

Review 2 - Data collection

Master 1 in Data Science & Engineering

Authors :
Tom Crasset
Maxime Lamborelle
Antoine Louis

Professors :
G. Louppe
P. Geurts
B. Cornelusse

Academic year 2018-2019

1 Collecting data

In order to fit our model, we need two types of data : historical data about previous tennis matches and statistical data about the players and their performances. The former is widely available online on different tennis websites such as `www.tennis-data.co.uk`, that provides historical data on matches in a structured form (CSV or Excel files). Concerning the amount of data needed, we decided to download all available data on this website, that is, all tennis matches of ATP tournaments from 2000 to 2018, as one year of matches only takes 520 Ko on average.

Statistical data about players is a bit more challenging to obtain. In fact, there doesn't exist a dataset that includes all player statistics, there rather are a multitude of websites that computed the statistics from the many matches. A solution to this problem is to scrape these websites using a small python script.

At first, this seemed straight forward, one just needs to extract the html from the site. However, a big hurdle arose : nearly everything on the site was dynamically generated using a javascript code and so the html returned from the first scraper script returned no useful data. The solution was to use a browser driver to emulate a browser that executes the javascript from the site and then extract that hmtl code with all the statistics it included. After that, it was only a matter of using the `BeautifoulSoup` package to extract the relevant information from the site. The site in question is the following : `http://www.ultimatetennisstatistics.com/playerProfile?playerId=XXXX`. It was just a matter of iterating from playerID 1 to 50000, extracting only the players that are currently active.

At the end, our data consists of one `xlsx` file of a bit more than 50000 lines containing all historical data about matches from 2000 to 2018, and more than 1000 `csv` files, each one containing statistical data about a particular active player.

2 Cleaning data

Once our data has been collected, some cleaning is necessary as our datasets are imperfect. Indeed, after concatenating the multiple `csv` files into one unique file containing all the data about the players, we notice that some feature values are missing for some lower ranked players, as data for this kind of players is more limited. Moreover, some values are written in a non suitable form and a lot of features scrapped from the website are just useless for our model and can be ignored. The cleaning of our data can easily be done with some python script using the `pandas` library.

2.1 Cleaning data about matches

The cleaning of the matches was quite simple. We decided to get rid of some useless features such as the city of the tournament and its ATP number. We also ignore all the odds of betting websites because too many values were missing. The "Date" feature was not in a suitable form (dd-mm-yy), so we split it in two new features "Year" and "Month" and scraped the day values. After renaming the different features with clearer words, we exported everything as a `csv` . The different features taken into account for the matches are the following :

Tournament	Winner	Games won by winner in set 1
Year	Loser	Games won by loser in set 1
Month	Winner ranking	Games won by winner in set 2
Country	Loser ranking	Games won by loser in set 2
Series	Nb sets max	Games won by winner in set 3
Court	Winner sets	Games won by loser in set 3
Surface	Loser sets	Games won by winner in set 4
Round	Completed or retired	Games won by loser in set 4
		Games won by winner in set 5
		Games won by loser in set 5

TABLE 1 – Features of the matches data

2.2 Cleaning data about players

The players data was a bit more laborious to clean because there were a lot more features to consider (initially 168 columns). The first step was to get rid of the useless features such as the Facebook page of the player, his nickname or the name of his coach. After making a first selection, we discarded approximately 50 features.

Then, we had to deal with the invalid percentage values. To be used, all values representing percentages must be real numbers between 0 and 1. The percentages in our datasets were represented in a string format ("xx%"), and so needed to be converted.

After that, from our 1000 players that are currently active, we decided to keep only the ones that are currently ranked under 400. That choice comes from the fact that only the best 104 players are selected for the tournament and only the 24 players left are ranked above the TOP 104. Taking the TOP 400 players is maybe even too big and we could actually only consider the TOP 200 as it is very rare that a player ranked outside the TOP 200 goes far in the tournament.

Next, we noticed that we had two features for the player's name in our dataset, "Last Name" and "First Name", while it was written in the format "LastName F." in the dataset concerning the matches. For a question of simplicity, we decided to stick with the second format and converted all names from the player's dataset in that format.

Finally, we had to deal with the missing values. By analysing the number of missing values in each column of our dataset, we decided to ignore the ones that had more than 60% of them. Moreover, we noticed that, for some reasons, a same group of approximately 60 low-ranked players had a lot of missing values for the different features. It would make no sense to keep these players in our dataset as almost no data concerning them are available, so we decided to get rid of these players. The other missing values only concern players with lower rankings that, for example, had no values in the feature "Final" because they simply never reached a final. For these types of missing values, we simply replace all NaN with zero values.

The final cleaned dataset was exported in a `csv` format and contains, in part, the following features :

Name	Clay %	Ace %
Country	Outdoor %	Double Fault %
Current rank	Matches Won %	Match Time (minutes)
Age	Service Games Won %	Game Time (minutes)
Favorite surface	Return Games Won %	Points per Game
Grand Slam %	1st Serve Won %	Points per Service Game
Round of 128 %	1st Srv. Return Won %	Points per Return Game
Round of 64 %	2nd Serve Won %	After Losing 1st Set
Round of 32 %	2nd Srv. Return Won %	After Winning 1st Set
Round of 16 %	Break Points Won %	Opponent Rank
Quarter-Final %	Tie Breaks Won %	...
Semi-Final %	Double Fault %	...
Final %	Vs Top 100 %	...

TABLE 2 – Some features of the players data

3 Storing data

The data was cleaned and then stored as a `.csv` file and thus it was really easy to use the `mongoimport` command to import it into a MongoDB database. To have consistent storage space amongst all the group members, the choice fell on a Mongo Atlas cluster. From this point on, the manipulation of the data will pass through this database.

4 Visualising data

Because of the big number of features, analysing the data before visualising it is mandatory. It is not possible to explore all the data only with charts.

4.1 Data analysis

To analyse the data, the free software `R` was used. The aim of this analysis was to highlight the correlation between some of the variables. `R` allows to compute the correlation matrix at once. Then, this correlation matrix is filtered to keep only the correlated variables.

Over the 120 features, many are well-correlated but a huge part of these correlations are trivial. For example, the '`Ace % per set`' is well correlated with the '`Ace % per match`', which totally makes sense. In a general way, all the features representing data about sets are well-correlated to those representing data about matches. In the next section, only the non-trivial correlations between variables are presented.

4.2 Data visualisation

To visualise our data, making charts simply is mandatory. We use the free software `Tableau`, allowing us to create all different kind of charts by just dragging and dropping data. Some important correlations between variables are presented in the next part.

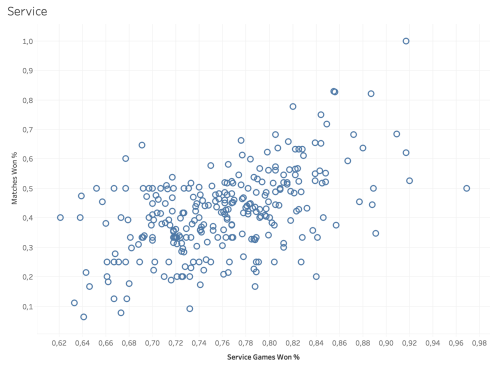
4.2.1 Principal correlated variables

In Figure 1 are presented the variables that are well correlated with the `Match Won %` or `Set Won %`, these two being obviously correlated. It concerns :

— The service : in professional tennis, winning its own game service is really important because the service has generally a great speed and so is an advantage for the server.

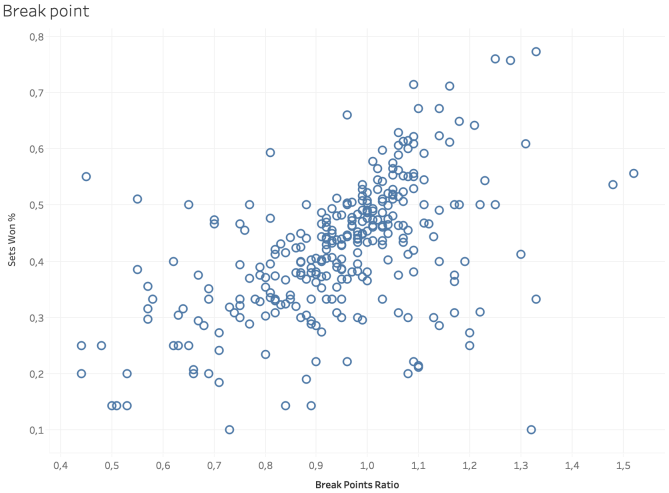
— The tie Breaks : tie breaks represent the last game when the set score is 6-6. They represent a huge amount of pressure because the winner of the tie break wins the set. So it is normal that players which can face this pressure wins more sets.

— The break point ratio (break points converted over total break points) : a break point is the occasion for the returner to win the game and then take advantage on the server in the set. So it is very important for players to take this opportunity.



(a) Service

(b) Tie breaks

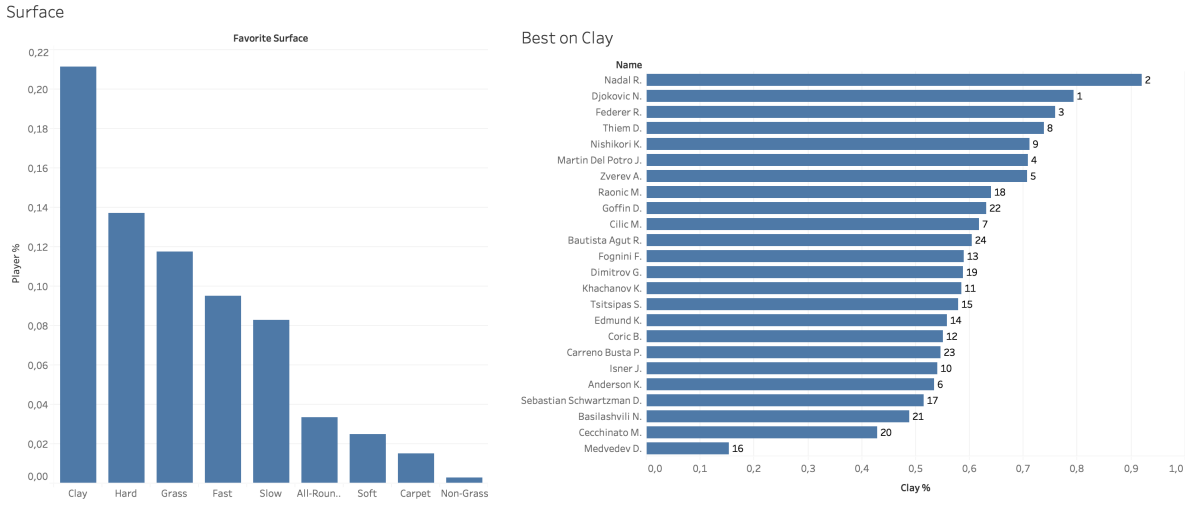


(c) Break points

FIGURE 1 – Correlated variables

4.2.2 Surface of the French Open

The French Open is known for being the only Grand Slam on clay. As can be seen in the FIGURE 2(a), the clay is the favourite surface of the majority of the players. Then in the FIGURE 2(b), the best players on clay are shown. The number after each bar represents the current ranking of the players. We can see that some players with a low ranking are however some of the best players on clay (as the Belgian guy, David Goffin).



(a) Favourite surface of players

(b) Best players on clay

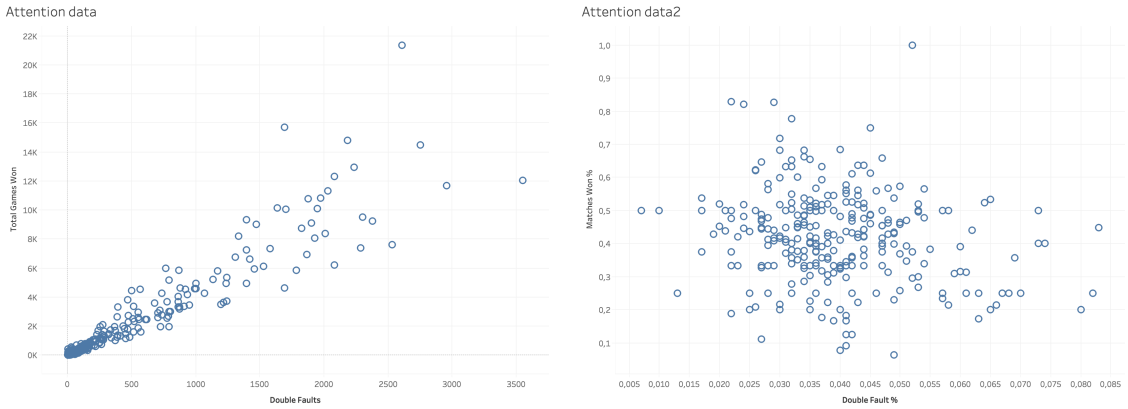
FIGURE 2 – Data on surface

4.2.3 Uncorrelated variables

When analysing data, we need to be careful not to conclude too quickly about some correlation. To illustrate that, here is an example.

In the correlation matrix, we found that the `total number of double faults` (miss of two consecutive services) is very well correlated with the `total number of matches won` as can be seen in FIGURE 3 (a). This is quite interesting because if the server makes a double fault, he looses one point. To ensure that, we then look at the correlation of the `Double fault %` and the `Match won %`. The result can be seen in the FIGURE 3 (b). In this scatter plot, there is actually no correlation between the two variables.

This paradigm is due to the fact that the first charts are not in percentage and that, effectively, if we wins more games, we have a greater probability to make double faults.



(a) Correlation between total match won and total double faults

(b) Correlation between % match won and % double fault

FIGURE 3 – Data proportion