Web scraping and social media scraping 2021

Author’s name

Błażej Popławski

Project description

I decided to scrap the basic match statistics including the match date, week of the league season, teams’ names, number of goals shot by both of teams (including half-time and full-time results), ball possession of each team, number of shots off and on target, corners, fouls committed, yellow and red cards given by the referee. I think it could be somewhat useful if someone would like to start betting matches result. This data-set could possibly be enhanced by some other statistics, such as exact times, at which the events occurred which could allow utilizing the sophisticated methods of survival analysis. It is however possible to conduct some advanced data analysis with the data already provided by the created web scraper. Knowing the numbers of each scraped event could be the beginning of conducting some statistical data analysis that could lead us to predicting the matches’ results. The scrapers used are BeautifulSoup, Scrapy and Selenium.

The data is scraped from the *footballcritic.com* website. It is a web portal which provides a very broad spectrum of information about football. We can find there some articles about everything that can be connected to football environment. The part of the website which is scraped in this project is the statistics part. It provides the data for many seasons in past, so we can scrap lot of data about already played matches.

The developed scraper can collect the data if it is provided with the proper link to the website presenting the history of matches played in the selected season. The user needs to provide this hyperlink to the scraper by pasting it in the proper place of the code. I have decided to scrap the data for season 2019/2020 of the English Premier League, which can be found under this link: https://www.footballcritic.com/premier-league/season-2019-2020/matches/2/21558

Scrapers’ mechanics and speed comparison

Each of the developed scrapers allows for gathering the data of one selected season of the selected country league. Firstly they collect all the hyperlinks to the played matches in the selected season. Web interface allows for printing match results for one game-week only. The data for the whole season is however available “under the hood” in the HTML code, and switching the view to the other game-week does not affect the HTML code in any significant manner. Match results presented in this page are clickable and provide the links to websites with the detailed results. The hyperlinks were collected by specifying a proper Xpath template for them.

In one season of Premier League there are 380 matches played. The data set collected for the demonstrative purposes was shrinked to the first 100 matches. BeautifulSoup and Scrapy scrapers collected all the hyperlinks at once and just afterwards the array of links was cut after 100 entries. For Selenium gathering the hyperlinks was conducted in a loop, so it needs to be terminated after executing 100 times.

There is a file script “run_scraper” which allows for calculating the execution time of the selected scraper. Selected lines of the file need to be uncommented in order to execute the chosen scraper. After finishing the scraping process the execution time is printed to the terminal. Gathering information about the first 100 matches played has taken BeautifulSoup scraper around 74 seconds. Scrapy scraper finished the task just after 13 seconds. Execution time of the Selenium scraper was much longer than the other two. It took 930 seconds for Selenium scraper to finish the process. Despite running it with the *headless* option, the HTML files loaded need to be interpreted by the browser what takes much time.

After gathering the hyperlinks to the matches all of them are opened and the data are scraped with the usage of Xpaths. In the case of Scrapy it is a two-step procedure: first one spider collects the hyperlinks and saves them in a csv file and then the second sprider collects the match statistics from each of the provided links. Each of the coded scrapers uses the same philosophy: gathers all the basic match statistics in the form of single variables and saves them to the instance of some more complex data type, such as pandas data frame. When the scraping process is finished, the collected data are saved to the csv file which could be further utilized for some statistical analyses, for example calculating some basic statistics in the spreadsheet.

Output file description

Scrapers using BeautifulSoup and Selenium uses the same type of the data structure for storing the gathered data: Pandas data frame. At the end of the process it is exported to the csv file. The structure of the data frame structure is as follows:

```
stats = pd.DataFrame({'MatchDate':[], 'Week':[], 'HomeTeam':[], 'AwayTeam':[], \
                      'HomeGoalsHT':[], 'AwayGoalsHT':[], \
                      'HomeGoalsFT':[], 'AwayGoalsFT':[], \
                      'HomeBallPos':[], 'AwayBallPos':[], \
                      'HomeShotsOffTarget':[], 'AwayShotsOffTarget':[], \
                      'HomeShotsOnTarget':[], 'AwayShotsOnTarget':[], \
'HomeCorners':[], 'AwayCorners':[], \
'HomeFouls':[], 'AwayFouls':[], \
'HomeYellowCards':[], 'AwayYellowCards':[], \
'HomeRedCards':[], 'AwayRedCards':[], \
})
```

The names of the fields are self-explanatory and don’t need to be further elaborated.

The scraper using Scrapy stores the same data but in the form of a Python class inheriting from the “scrapy.Item”.

The first line of the exported csv file is the heading of the stored data-set. It contains the names of each of the columns. Next lines are the data for each of the investigated match. A short sample of the collected data looks as follows:

```
0,"Aug 9, 2019",1,Liverpool,Norwich,4,0,4,1,58%,42%,5,6,7,5,11,2,9,9,0,2,0,0
1,"Aug 10, 2019",1,West Ham,Man City,0,1,0,5,43%,57%,2,3,3,9,1,1,6,13,2,2,0,0
2,"Aug 10, 2019",1,Burnley,Southampton,0,0,3,0,46%,54%,4,5,4,3,2,7,6,12,0,0,0,0
3,"Aug 10, 2019",1,Crystal Palace,Everton,0,0,0,0,35%,65%,0,3,2,3,6,2,16,14,2,2,0,0
4,"Aug 10, 2019",1,Watford,Brighton,0,1,0,3,48%,52%,4,2,3,3,5,2,15,11,0,1,0,0
5,"Aug 10, 2019",1,Bournemouth,Sheffield Utd,0,0,1,1,53%,47%,7,4,3,3,3,4,10,19,2,1,0,0
6,"Aug 10, 2019",1,Tottenham,Aston Villa,0,1,3,1,70%,30%,14,2,7,4,14,0,13,9,1,0,0,0
7,"Aug 11, 2019",1,Leicester City,Wolves,0,0,0,0,70%,30%,9,4,1,2,12,3,3,13,0,2,0,0
8,"Aug 11, 2019",1,Newcastle,Arsenal,0,0,0,1,38%,62%,4,4,2,2,5,3,12,7,1,3,0,0
9,"Aug 11, 2019",1,Man Utd,Chelsea,1,0,4,0,46%,54%,2,7,5,7,3,5,15,13,3,4,0,0
```

Each next value in the line (except the first one, which is simply the index of the created data structure) corresponds to the name provided in the code listing presenting the data frame structure.

Preliminary data analysis

The collected data can be used for some sophisticated statistical data analysis. Here some preliminary, and very simple, statistics for the is presented for demonstrative purposes.

The maximum and average values for each collected data are presented below:

	Max	**Sum**	**Average**
HomeGoalsHT	5	70	0.7
AwayGoalsHT	5	66	0.66
HomeGoalsFT	8	155	1.55
AwayGoalsFT	9	131	1.31
HomeBallPos	80		50.73
AwayBallPos	74		49.27
HomeShotsOffTarget	15	568	5.68
AwayShotsOffTarget	9	409	4.09
HomeShotsOnTarget	13	464	4.64
AwayShotsOnTarget	15	424	4.24
HomeCorners	14	626	6.26
AwayCorners	16	472	4.72

HomeFouls	19	1010	10.1
AwayFouls	19	1081	10.81
HomeYellowCards	6	163	1.63
AwayYellowCards	5	197	1.97
HomeRedCards	1	4	0.04
AwayRedCards	1	1	0.01

We see that for the first 100 matches played in the season 2019/2020 of English Premier League 286 (155+131) goals were scored, what gives 2.86 goals for the event. The average ball possession was almost equal between home and away teams. The highest ball possession shares were 80% for the home team and 74% for the away team. In the 100 matches investigated there were only 1 red card given by the referee. Home teams take more attempts to score a goal (shots off and on target)

The next table presents the frequencies of numbers of goals scored in the analysed matches.

Value	**HomeGoalsHT**	**AwayGoalsHT**	**HomeGoalsFT**	**AwayGoalsFT**
0	52	54	19	32
1	33	31	37	33
2	12	12	25	21
3	0	2	13	8
4	2	0	4	2
5	1	1	1	3
6	0	0	0	0
7	0	0	0	0
8	0	0	1	0
9	0	0	0	1
10	0	0	0	0

We see that the most frequent number of goals scored by both home and away teams is 1 – 37 and 33 times respectively. For half-time results the most frequent number of goals is 0 for both sides.