Premier League Football & Betting Statistics

Matthias Besnier, Augustin Gervreau, Jules Merigot[1]

[1]Paris Dauphine University - Data Acquisition, Extraction, and Storage

January 2024

Introduction

This project focuses on the extraction of data from various sources related to the Premier League in order to build a comprehensive database containing important information and statistics about the matches played during the 2023-24 season, the players of all participating teams, and the betting odds of future matches. This is done with the goal of informing potential sports betters and football fanatics on the statistics of past games and the betting odds of future matches so that they may be better informed for these upcoming games.

Our final database will consist of datasets on all matches played in the 2023-24 season and their statistical results and information, the general information and performance statistics of all the players in the Premier League, various news articles related to the Premier League, and the betting odds of future games acquired from two well-known online betting websites.

The final code and full database can be found at the following GitHub repository: Premier League Data Project.

1 Acquiring and Extracting the Data

Matches Played

The statistics of the matches already played in the 2023-24 season have been scraped from the website onefootball.com which provides relevant information about the Premier league (PL) matches. One Football is a German platform-based football media company that features live-scores, statistics and news from 200 leagues in 12 different languages. Some examples of usual and information are the *teams involved*, the *date* or the *location* but also some useful statistics such as *possession*, *total shot*, *shots on target* and so on. Since the data structure was fixed and clear, we decided to store everything in csv files (see `Premier_league_Scores_Schedule.ipynb` in the associated GitHub repository) to build our database. A web crawler like `Scrapy` was not relevant here since we weren't guaranteed to get every played match by following links from page to page. Using the `BeautifulSoup` package, we opted in favor of a less elegant method. Finding the *ID* of the first 2023-2024 PL match and then iterating until no other PL played matches are found.

Player Statistics

In order to acquire and extract all the desired information and statistics on the players of the Premiere league teams, we decided to once again scrap the Premier League page of the One Football website. All the relevant code related to the acquisition and extraction of the players' information and statistics can be found in the `Premier_league_players.ipynb` file in the associated GitHub repository.

The python code blocks work to extract all important information and statistics about the football players of each team in the Premier League. This is done by first collecting the href links of all teams in the premier league, then collecting all players hrefs to collect their info, and finally collecting all basic information about the players and collecting all their statistics in two seperate Pandas dataframes, before merging the two into one final Pandas dataframe that serves as our final, cleaned dataset for the player statistics. An important note is that a minority of players have no associated statistics page on the One Football website due to lack of playing time this season, therefore the values for these players show up as "Nan" in the dataset. We decided to keep these players in the final dataset however, because a future and updated scrapping of the website could reveal new values if the players are given playing time.

News Articles

Premier League has always been a hot topic in the UK, dozens of articles are written every day about it. We have aggregated some PL-related articles (as URL) from Premier League page of One Football and Premier League page of BBC. Starting from these web pages we picked web links and applied a Breadth for Search (BFS) algorithm to explore the other pages looking for other articles (since there are always some related articles on those pages). Some limitations of this strategy are the inability to ensure a complete exploration of all articles related to PL on the website (it depends on the starting points) and that we only deal with PL-related articles. From those websites, we only extract *article titles*, *released date*, *URL*, and some *mentions* for possible content filtering applications. These operations (see) have been done with the BeautifulSoup package, we also tried to use Selenium to interact with the JavaScript elements (like the "Show more" button) but it failed.

Betting Odds

For this section, we focused on two French betting websites, Winamax and Betclic. As our approach focuses on the Premier League data, we acquired our data by scraping from the Premier League websites of Winamax and Betclic. Our code could easily be extended to start from the main page to scrape all available bets on both sites. The scraping was done with `Scrapy` and the code for this section is in the `tutorial` folder in the associated GitHub repository.

Scraping Winamax Winamax uses Javascript to build its pages, however, it accesses a JSON-like dictionary which is present on the scraped page. Since all the necessary information is present on that page, this allows us to scrape data by querying this data as a Python dictionary. All that is necessary is then combining the appropriate IDs used to get the data in the appropriate structure. We did not look at acquiring all possible information from this, but instead chose to focus on only the data on proposed odds, on the percentages of choices for betters on the platform, and some information on the matches such as the date and the time.

There are many more precise bets which can be extracted for each individual match, on separate pages. To access these, we do not have links, but instead the observation of the fact that the link of each match can be found by adding the match id at the end of `"https://www.winamax.fr/paris-sportifs/match/"`, which gives us all the links we are looking for. Each page also has information in the shape of a JSON-like dictionary, so each page can be scraped the same way (which differs from the way the data scraping is done on the first page).

Scraping Betclic The data on the Betclic Website can be found directly embedded in the `HTML` script received when scraping the page using `Scrappy`. All the individual results we want to extract are found within `div` tags, which allows us to simply scrape the page linearly. We then have links with specific tags that send us to the individual match pages, which all have a similar basic structure and can therefore be scraped using the same parsing function.

Processing the data With all information extracted in JSON files in a semi-structured way, we noticed that some of the scraped data lies outside of the scope of the Premier League, this issue was dealt with when the data extracted from the betting websites was crossed with the data from our other sources.

Aggregating the data Since the betting data were extracted as JSON files, in order to make our storage solution more simple later on, we decided to aggregate the numerous acquired JSON files for the matchs' odds into two CSV files. One CSV file for the scrapped Winamax data, and one CSV file for the scrapped Betclic data. This is done in the `bets_json_to_csv.ipynb` file, and the betting data can be found in the `betclic_bets_PL_matches.csv`

and `winamax_bets_PL_matches.csv` CSV files. This allows us to have only 6 CSV files total to store within our database storage solution.

Discussion/Possible extension of this approach We noticed that the betting websites do not give access to the data on the entire football season, this could be expected since the odds are probably not yet made. However, this means that in order to scrape a large amount of information, we would need to launch our `Scrapy` spiders regularly, over the span of about one year. We would then be able to combine the information of consecutive scrapes by simply concatenating the data in the shape of Python dictionaries.

A second observation we made is the fact that the odds and percentages of bets made change over time, even within the same hour. This means that an extension of our work could be to scrape at several time intervals, and then build time series of the odds and betting percentages. Such an approach would mean that we would have to stick to semi-structured data as time series of arbitrary size could not be stored in a traditional Relational Database Management System (RDBMS). We did not implement this approach. but a simple way to do this would be to write a python function combining two JSON files/python dictionaries.

2 Storing the Data

Storage Solution In order to properly store our data, we had to choose the right storage solution based on our datasets. Due to our final datasets being in CSV files with structured formats, we decided that a Relational Database Management System (RDBMS) was the best choice. We chose to use an SQLite DBMS to store and manage our extracted data as it allows us to have a lighter, file-based database. This allows us, as well as other users, to use SQL in order to make complex queries which makes data retrieval efficient and straightforward. Additionally, RDBMSs adhere to ACID (Atomicity, Consistency, Isolation, Durability) properties, which ensures reliable transaction processing and data integrity. This makes our database easily accessable and reliable for all users, such as serious sports betters looking to better inform themselves on betting odds for the Premier League.

Creating the database We created the SQLite database using the `sqlite3` package for Python. This allows to easily edit the database via Python as well as provide clarity on how the database is created. The code can be found in the `create_Premier_league_db.ipynb` file on the repository.

Tables in the database In our SQLite DBMS we are storing six total tables from 6 different CSV files containing important data about the 2023-24 football season of the Premier League. These files are the data on: statistics of matches played, statistics of future matches, player performance statistics, related news articles, betting statistics from Winamax, and betting statistics from Betclic.

Database schema The database is made up of the six aforementioned datasets acquired as CSV files. In the SQLite database, the tables are named `articles, past_matches, future_matches, player_stats, articles, bets_Betclic, and bets_Winamax`. Below is a description of their table schemas.

Articles: All relevant articles related to the Premier League.

- title (TEXT): Primary key, title of the news article.
- released_date (DATE): Date the article was released.
- source (TEXT): News source of the article.
- url (TEXT): Url of the article.
- mentions (TEXT): Relevant teams mentioned in the article.

Past Matches: Statistics about all matches already played during the 2023-24 season.
(This table has many columns so only the first few most important ones will be outlined here, the rest may be seen in the dataset directly in the repository.)

- team_home (TEXT): Primary key, home team hosting the match.
- team_away (TEXT): Away team being hosted for the match.

- datetime (DATE): Date of the match.
- location (TEXT): Stadium hosting the match.
- score_home (INTEGER): Score of the home team.
- score_away (INTEGER): Score of the away team.
- home_win_proba (INTEGER): Probability of the home team winning the match.

Future Matches: Important information about the future matches of the 2023-24 season.

- team_home (TEXT): Primary key, home team hosting the match.
- team_away (TEXT): Away team being hosted for the match.
- datetime (DATE): Date of the match.
- location (TEXT): Stadium hosting the match.

Player Statistics: Important statistics about the performance of players in the Premier League. Separated in 5 main categories, 'key stats', 'defence', 'distribution', 'offense', and 'discipline'.
(This table has many columns so only the first few most important ones will be outlined here, the rest may be seen in the dataset directly in the repository.)

- name (TEXT): Primary key, first and last name of the player.
- team (TEXT): Team that the player plays for.
- position (TEXT): Playing position of the player.
- height_cm (INTEGER): Height in centimeters of the player.
- weight_kg (INTEGER): Weight in kilograms of the player.
- key_stats_goals (INTEGER): Number of goals scored by the player.
- key_stats_assists (INTEGER): Number of assists produced by the player.

Betclic Bets: The betting odds and statistics for all possible bets for each match scraped from the Betclic website.

- team_home (TEXT): Primary key, home team hosting the match.
- team_away (TEXT): Away team being hosted for the match.
- bet_type (TEXT): The type of bet (e.g.: total goals scored for Arsenal).
- outcome (TEXT): The outcome of the bet type.
- odds (REAL): Odds of the bet.

Winamax Bets: The betting odds and statistics for all possible bets for each match scraped from the Winamax website.

- team_home (TEXT): Primary key, home team hosting the match.
- team_away (TEXT): Away team being hosted for the match.
- bet_type (TEXT): The type of bet (e.g.: total goals scored for Arsenal).
- outcome (TEXT): The outcome of the bet type.
- odds (REAL): Odds of the bet.
- percentage (INTEGER): Percentage odds for that bet.

3 Assessing the Data

Unwanted data Some of the scraped data should not appear in the dataset, like teams from the wrong league or even from the wrong sport. We eliminated this by crossing the sources, we made sure that the remaining data contains exclusively what we are interested in: Premier League data.

Missing Data Our dataset only contains the data from a few matches. This is due to the availability of the data as described in the section on Betting Odds, and could only be resolved by scraping the data over the whole year (each week/day).

Incorrect data We also encountered some situations where data was not present on the website (for instance on betting percentages, where the sum over outcomes does not add up to 100%). This was seen by manual sampling on the dataset. The cause we found for this issue is that the missing data was replaced by 0 during scraping. Thus on the betting websites in some instances, the data is not directly present.

A second problem is the fact that the format of the data does not always come in the same shape on all the websites. There are different ways of formatting the names of the teams or the players for instance. We made the following observation: since the Premier League is a championship, each team meets twice (first leg, second leg) and (*team home*, *team away*) can be used as a key because it only happens once in the entire season. To normalize the teams' names from each table, we apply a method (*has subwords()*) to check if a team is part of the Premier League (with a subword detection) but it also returns the correct full name of the team based on a reference list from onefootball.com.

Conclusion

In conclusion, this project has successfully created a comprehensive database for the Premier League 2023-24 season, effectively blending detailed match statistics, player performances, relevant news, and betting odds. This rich compilation of data serves as a valuable asset for sports bettors and football enthusiasts, offering insightful perspectives on past and future games. It could also be useful for an analysis on betting habits, as we could extract meaningful information such as expected loss of betting or get insights on biases in betters' habits. Our work demonstrates the significant impact of data-driven analysis in sports, enhancing fans' and bettors' engagement with the game. Looking ahead, the potential for further development and refinement of this database is vast, promising continued support and enrichment of the football community's experience.