Automated Scouting Report Generation for the Adidas Next Generation Tournament (ANGT)

Aleix Vindel

June 2025

1 Introduction

“ *”Information is a source of learning. But unless it is organized, processed and available to the right people in a format for decision making, it is a burden, not a benefit.”* — William Pollard ”

In the current landscape of data science applied to sports, this statement resonates strongly. The availability of data does not guarantee competitive advantage of the game. In contrast, the lack of automated structures to organize and analyze data often makes information a burden for coaches, analysts, and technical staff. This challenge is particularly evident in development tournaments such as the **Adidas Next Generation Tournament (ANGT)**, where data access is limited, fragmented, and lacks standardization.

In these contexts, manual scraping of statistics — though widely used by scouts and agents — is a repetitive, non-reusable, and time-consuming task. This presents a significant obstacle for those who want to produce scouting reports based on easily shareable data.

This project proposes the development of a **tool aimed at automating the extraction and structuring of statistical data from matches and players in the ANGT competition.** Through a technical approach, the objective is to make the data available so that teams can turn them into useful information for decision making and performance analysis.

2 Context

This project is not designed for the exclusive benefit of a single club, but rather expects to provide a useful tool for all organizations participating in the Adidas Next Generation Tournament (ANGT). The ANGT is an international U18 basketball competition organized by Euroleague Basketball, in which only selected teams are invited to compete. These are typically elite-level clubs or academies with highly regarded prospects. In recent years, teams such as Žalgiris Kaunas, Real Madrid, Mega Basket, and FC Barcelona have been among the winners.

The structure of the competition consists of four qualifying tournaments held in different cities throughout the season. Each tournament features eight invited teams, divided into two groups of four. Teams compete within their groups, and the top teams from each group face off in a final to determine the winner.

The winners of the four qualifiers, along with additional invited teams, advance to a final tournament with the same group-stage format. The winner of this final event is crowned the ANGT champion for the season.

3 Justification and State of the Art

One of the persistent challenges in the field of sports analytics is how to store and structure data so that it becomes usable and meaningful. While access to performance data has increased over recent years, most organizations still lack the technical knowledge, infrastructure, or financial resources required to build and maintain data systems that allow for scalable and actionable use.

Furthermore, many professionals in the sector — including statisticians and mathematicians — often lack experience in areas such as software architecture, web scraping, and automated data pipelines. This technical gap creates a significant barrier when trying to develop tools that go beyond isolated analyses and move toward fully integrated, long-term solutions.

With these limitations in mind, this project proposes the creation of a customizable and user-friendly infrastructure for extracting, storing, and structuring ANGT data. The goal is to offer a flexible and accessible system that empowers analysts and coaches to focus on what truly matters: interpreting data and making informed decisions.

State of the Art

There is a wide range of existing tools and studies related to data scraping and analysis in other basketball contexts. In Spain, one of the most prominent initiatives was developed by Adrià Arbués: *Bue Stats*, a platform that provided users with a graphical interface to download and explore basketball statistics [Arbués, 2020]. Building upon that foundation, Nil Crespo later launched *Bue Stats 2.0*, expanding the functionality through a web-based interface that improved accessibility and performance [Crespo, 2023].

In one of my previous works, I also explored this concept by developing a scouting platform for FEB leagues (Spanish Basketball Federation), with a focus on structuring player data for easier evaluation and comparison [Vindel, 2022].

At the global level, numerous examples exist, including open-source repositories containing play-by-play data for professional leagues like the ACB [Díaz, 2021] or NBA [Shufinskiy, 2020]. In addition, various academic and technical works have analyzed how to structure sports data using spreadsheets or lightweight databases, emphasizing reproducibility and accessibility [Broman and Woo, 2005].

However, despite this body of work, there appear to be no publicly available tools or studies focused on the Adidas Next Generation Tournament (ANGT).

4 Objectives and Methodology

The project is structured around two main components: (1) the creation of scripts to extract and store play-by-play and box score data; and (2) the use of this data to generate detailed scouting reports.

This division is essential to ensure that the project is both customizable and reusable. Once the data collection is complete and the information is stored in structured files, these datasets become adaptable resources that teams can use according to their specific needs and priorities.

The first phase is primarily technical and will be most useful for professionals who may not have expertise in web scraping, but who do have a background in statistics or performance analysis. This part of the system will provide pre-processed and summarized data files, allowing users to bypass the complexity of raw data extraction.

The second phase is more flexible and oriented toward customization. It focuses on applying the previously collected data to produce tailored scouting reports. In this case, a demonstration will be developed to illustrate the analytical potential of combining play-by-play and box score data. This demo report will showcase how the data can be used to analyze a specific team's performance in depth, offering a tangible example of the project's practical utility.

General Objective

To develop an automated system for extracting, structuring, and analyzing data from the ANGT tournament to support scouting and performance analysis.

Specific Objectives

- To implement scraping scripts capable of retrieving box score and play-by-play data from ANGT sources.
- To store the extracted data in structured and reusable formats.
- To generate scouting reports using pre-processed datasets.
- To demonstrate the utility of the system through a case study on a specific team.

Methodology

The technical development of the project is divided into two main stages: data extraction and data analysis.

For the data extraction phase, the project will use **Python** as the primary programming language, alongside libraries such as **BeautifulSoup** and **Selenium**. These tools will be employed to scrape both box score and play-by-play statistics from publicly available sources related to the ANGT tournament. Selenium will allow interaction with dynamically loaded web elements, while BeautifulSoup will be used for parsing static HTML content.

The extracted data will be stored in structured **CSV files**, chosen for their simplicity, portability, and ease of use in a wide range of data analysis environments. Each game's box score and event log will be saved in separate files, with standardized formats to enable reuse across different analysis pipelines.

In the second phase, the project will use **Python** libraries such as **pandas**, **matplotlib**, and potentially **plotly** or **seaborn** to analyze and visualize the data. This will include both aggregated metrics and game-specific event flows. The goal is to translate raw statistical data into actionable insights through clear and interpretable visualizations and summaries.

The report generation will be based on this processed data, although the final visualization interface (e.g., web app, dashboard, or notebook) is still under consideration.

Timeline and Resource Planning

Timeline. The project is expected to be completed over a period of approximately 4 weeks, as the duration of the course, divided into three main phases:

- **Week 1:** Identification of relevant data sources, and definition of scraping targets and file structures.
- **Week 2:** Development and testing of the scraping scripts using Python, Selenium, and BeautifulSoup. The goal is to extract both box score and play-by-play data and store it in standardized CSV files.
- **Weeks 3–4:** Data cleaning, analysis, and visualization. During this phase, processed datasets will be used to create scouting summaries and performance reports. A case study based on a specific ANGT team will be developed as a demonstration of the system's utility.

Technical Resources. The project will be developed using a personal computer equipped with:

- Python 3.10+
- Libraries: Selenium, BeautifulSoup, pandas, matplotlib, seaborn, plotly
- Development tools: VSCode

Data will be stored locally in CSV format. Git may be used for version control.

Human Resources. The project will be developed individually by the author.

5 Variable Selection and Data Characterization

In this project, the selection of variables and the characterization of the data will depend largely on the needs and context of the team or analyst using the

tool. The general objective is to provide the maximum amount of structured information possible from various data sources, allowing users to define specific variables and indicators based on their scouting goals.

Although the project's core design aims to produce reusable data files through automated extraction, the actual choice of variables and their interpretation will always be tied to the specific purpose of the analysis — typically a scouting report on an opponent or the team itself.

The demonstration case presented in this report will follow that logic: using the available dataset to generate a scouting report on a selected team. This will include aggregated statistics and contextual insights derived from box score and play-by-play data.

6 Data Collection and Processing Strategy

One of the main challenges of this project is the automation of data collection to allow post-analysis. To address this, the project follows a structured workflow:

```
scraping → processing → CSV generation → usage → visualization
```

The first two steps — data extraction and processing — are crucial to ensure that the information is clean, structured, and reusable.

By analyzing the data available on the ANGT website, several types of information can be identified:

- Player statistics aggregated by tournament (total box scores)
- Team and individual box scores for each game
- Shot charts for each player and game
- Play-by-play logs of every match

Among this wide range of available information, this project will focus specifically on:

- **Team box scores** for every game
- **Play-by-play logs** from all tournament matches

To obtain these two types of information, an analysis of the URL structure and navigation hierarchy of the official ANGT website was carried out.

The full implementation of the data extraction functions can be found in the author's public repository [Vindel, 2025].

Team Box Scores

Team box scores follow a predictable URL pattern that facilitates automation. A typical example is:

```
https://www.euroleaguebasketball.net/es/ngt/teams/u18-c
rvena-zvezda-belgrade/statistics/jre/?season=2024-2025%
20Belgrade&phase=All%20phases#accumulated
```

This pattern can be generalized as:

```
https://DOMAIN/es/ngt/teams/TEAM_IDENTIFIER/statistics/XXX/
?season=SEASON+VENUE&phase=PHASE#accumulated
```

Several key elements are worth noting:

- The **TEAM_IDENTIFIER** is unique for each team and remains consistent across all subpages (e.g., `/statistics`, `/roster`, etc.).
- The **SEASON and VENUE** parameters (e.g., `2024-2025 Belgrade`) are known in advance, as they correspond to the specific ANGT tournament under study.
- The **PHASE** parameter is always set to `All phases`, since the project want to collect statistics from every game played by each team, regardless of whether they are group-stage or final matches.
- The final fragment `#ACCUMULATED` selects the accumulated statistics tab, as opposed to average per-game stats, which aligns with the scouting use case.

With these concepts in mind, a simple code structure can be defined to handle the extraction of team box scores. The process is divided into two main functions:

```
teams_urls = scrap.scrap_urls_teams(tournament_id)
box_scores = scrap.scrap_box_scores(teams_urls)
```

Without going into excessive detail, the first line of code retrieves all team identifiers and dynamically constructs the URLs that contain each team's box score statistics:

```
teams_urls = scrap.scrap_urls_teams(tournament_id)
```

The second line visits each of those URLs, scrapes the statistical table, and stores each player's box score information into a Python list:

```
box_scores = scrap.scrap_box_scores(teams_urls)
```

From a technical standpoint, both scraping processes rely on the `Selenium` library. This is necessary because the ANGT website does not render statistical values directly in the static HTML DOM. Instead, the content is loaded dynamically via JavaScript after the initial DOM is built. For this reason, the scraper must "simulate" a user loading the page and waiting for the data to appear — a task well suited for Selenium.

The general workflow is consistent across both steps: the ChromeDriver is executed, the script waits for the page and target elements to fully load, and then extracts the required information using CSS selectors.

In the first function, the script waits for the tournament selector to become active, selects the appropriate tournament, waits for the updated list of teams to render, and then retrieves the URLs and names of all participating teams.

In the second function, the same logic is applied to visit each team's statistics page, wait for the data table to load, and extract player-level statistics. As a specific design choice, team-level aggregate rows (i.e., total team stats) are excluded from the final dataset.

Finally, the extracted data is stored locally as a CSV file using the following line:

```
export.save_csv_total_box_score(box_scores)
```

Play-by-play (PbP)

The scraping process for play-by-play (PbP) data is slightly more complex than that of the box scores, although the general idea is similar: retrieve all game URLs, scrape their content, clean the data, and then save it in structured CSV files.

The pipeline can be represented as follows:

```
games_urls = scrap.scrap_urls_games(tournament_id)

for game_name, game_url in games_urls:
    play_by_plays = scrap.scrap_play_by_plays(game_url)
    export.save_csv_play_by_plays_raw(game_name, play_by_plays)

    cleaned_play_by_plays = scrap.clean_play_by_plays(play_by_plays)
    export.save_csv_play_by_plays_clean(game_name, cleaned_play_by_plays)
```

As shown, the script first retrieves all available game URLs, then iterates over them to extract and clean the play-by-play logs.

Unlike the team box score URLs, which are embedded in each team profile, the play-by-play URLs are structured under a single base address. The scraper builds these dynamically using round-based query parameters:

```
https://DOMAIN/es/ngt/teams/TEAM_IDENTIFIER/statistics/X
XX/?season=SEASON+VENUE&phase=PHASE#accumulated
```

Here, **SEASON** refers to the tournament identifier (typically encoding both the season and host city), and **ROUND** refers to the game day or match round.

Each tournament typically consists of 4 rounds (three for the group phase and one final). However, to make the script reusable across different tournament formats, the scraper iteratively increases the round number until it detects a redirection to the final available round. This mechanism is used as a stopping condition to ensure all matches have been captured without hardcoding the maximum round count.

Once the raw data is scraped, it undergoes a cleaning process to normalize the format, discard irrelevant rows (e.g., unknown actions or team events), and ensure chronological order. Cleaned versions are stored separately to facilitate further analysis.

For each round of the tournament, the scraper searches for all match URLs by targeting the CSS selector used for linking to game pages. These URLs are collected along with the name of the match, which will later serve as an identifier for storing the extracted data.

The scraping logic for play-by-play data has additional complexity due to the dynamic structure of the website. Within each game's page, five buttons are displayed — one for each quarter and an additional one for overtime, which is disabled if not applicable.

The script iterates over each enabled button, corresponding to the actual periods played. Each period contains multiple PbP entries, which can be of two types:

- **Scoring entries**, which affect the scoreboard and include the updated score.
- **Non-scoring entries**, which describe other actions (missed shots, fouls, turnovers) without reflecting score changes.

In addition, the PbP entry must be mapped to the correct team side: *home* or *away*. This is determined through CSS class names that differentiate the alignment of each entry in the DOM (left or right side).

Each entry is then parsed based on whether it is a scoring or non-scoring event. The following information is extracted:

- Home team name
- Away team name
- Period (quarter)
- Game time
- Player name
- Action description
- Team responsible for the action

- Score (if applicable)

The final structured row will differ slightly depending on whether the event affected the score. For example:

- **Non-scoring entry:**

  `U18 EA7 Emporio Armani Milan, U18 Crvena Zvezda Belgrade, 1st Quarter, 00:53, "GARAVAGLIA, DIEGO", Missed Two Pointer (1/3 - 6 pt), U18 EA7 Emporio Armani Milan, ,`

- **Scoring entry:**

  `U18 EA7 Emporio Armani Milan, U18 Crvena Zvezda Belgrade, 1st Quarter, 01:11, "MARJANOVIC, STEFAN", Free Throw In (4/5 - 8 pt), U18 Crvena Zvezda Belgrade, 21, 22`

This cleaned structure facilitates detailed analysis of game dynamics and individual player performance, while maintaining compatibility with further automated processing steps.

Once all play-by-play entries have been extracted, the data is stored.

Play-by-play Data Cleaning

The data cleaning phase pursues three main goals: (1) to categorize the action performed in each play, (2) to reconstruct the score progression throughout the game, and (3) to identify the 10 players on the court at any given moment.

To begin with, each raw action is mapped to a standard action code using the following dictionary:

```
"Missed Three Pointer" → 3PA
"Three Pointer"        → 3PM
"Missed Two Pointer"   → 2PA
"Two Pointer"          → 2PM
"Missed Free Throw"    → FTA
"Free Throw"           → FTM
"Foul Drawn"           → PFRec
"Foul"                 → PF
"Turnover"             → TOV
"Assist"               → AST
"Off Rebound"          → OREB
"Def Rebound"          → DREB
"Rebound"              → REB
"Shot Rejected"        → BLKRec
"Block"                → BLK
"Steal"                → STL
"TV Time Out"          → TO
"Time Out"             → TO
"In"                   → IN
"Out"                  → OUT
```

An additional level of logic is applied to distinguish whether a foul received (`PFRec`) is a shooting foul. If the next play corresponds to a free throw attempt by the same player — or if the team is already in the bonus — the foul is classified as a shooting foul.

To reconstruct the game state, especially the score and lineups, the play-by-play is processed in reverse chronological order (from earliest to latest), starting from an initial score of 0–0 and empty lineups. As plays are parsed, players involved in the first actions are assumed to be on court retroactively until a substitution is registered. Once both teams have five players assigned, substitutions are handled by explicitly recording "IN" and "OUT" events, ensuring lineup continuity.

In parallel, the evolving score is computed and assigned to each play, allowing for consistent tracking of the game's state at any moment.

For events that are not associated with individual players the performer is labeled as `TEAM`, distinguishing them from player-level actions.

After cleaning, each row includes the following enriched structure:

```
Period, Time, Player, ActionCode, Team, Score_Home, Score_Away,
Lineup_Home (5 names), Lineup_Away (5 names)
```

Example rows after cleaning:

```
1,08:55,"BIKIC, JOVAN",3PA,U18 Crvena Zvezda Belgrade,5,1,
"CECCATO, MATTIA",
"CORTELLINO, GIOVANNI",
"GARAVAGLIA, DIEGO",
"LONATI, ACHILLE",
"SUIGO, LUIGI",
"BIKIC, JOVAN",
"COPIC, MILOS",
"NEDELJKOVIC, ALEKSEJ",
"PAVLOVIC, NOVAK",
"STOJKOVIC, LAZAR"
```

Each cleaned dataset is saved as a separate CSV file per game.

Ethical and Legal Considerations in Data Management

From a legal standpoint, it is essential to acknowledge that this project deals with data — and we are never the owners of that data. The information collected via web scraping must not be used for personal or commercial purposes under any circumstances. The responsibility lies in respecting both the origin and the purpose of the data.

This concern becomes even more critical given that the Adidas Next Generation Tournament (ANGT) involves underage athletes. According to basic data

protection principles, particularly within the framework of Spanish and European legislation, the collection and use of any data related to minors must be conducted lawfully and with explicit consent. Moreover, any use must remain aligned with the original purpose for which the data was obtained.

Referring specifically to Article 84 of the Spanish Organic Law on Data Protection (LOPD), the dissemination of personal data related to minors on the internet can be considered an unlawful intrusion into their fundamental rights.

From an ethical perspective, it is important to contextualize all the data collected and analyzed, especially in high-exposure events like the ANGT. Exceptional statistical performances during the tournament may carry significant personal consequences for young athletes, who are often still in key developmental stages. In an age where social media is saturated with highlight reels designed to entertain rather than inform, analysts must take care not to contribute to unrealistic expectations or undue pressure. The maturity level and long-term development of the athlete should always be considered when interpreting or publishing scouting insights.

Selection and Justification of the Data Analysis Methodology

The methodology applied for data analysis is strongly tailored to a specific use case: generating a scouting report for one team in the ANGT. However, one of the project's core goals is to produce datasets that are flexible and reusable across different scouting needs, allowing each analyst or coaching staff to define their own analytical focus.

Context and Limitations. ANGT tournaments feature a small number of games, typically three to four per team. This results in a limited sample size, increasing the likelihood of statistical anomalies. For more robust evaluations, these data should ideally be complemented with additional sources such as domestic league stats (scraped from federation websites using similar tools) and video analysis.

Levels of Analysis. This study distinguishes two main levels of analysis:

1. **Team-level performance**, to understand overall playing style and effectiveness.
2. **Player-level metrics**, to identify individual contributions, efficiency, and player profiles.

Team Analysis. For team-level analysis, we apply descriptive and comparative metrics such as:

- *Offensive Efficiency Rating (OER)*, *Defensive Efficiency Rating (DER)*, and *Pace*.

- *Four Factors*: Effective Field Goal Percentage (eFG%), Turnover Rate (TOV%), Offensive Rebound Rate (OREB%), and Free Throw Rate (FT Rate).

These indicators provide insights into how a team executes both offensively and defensively. Comparative visualizations — such as OER/DER evolution over games, 3-point usage vs. effectiveness (%3PA vs. %3P), or scatter plots of performance across multiple teams — will be included to contextualize results.

In addition, we explore lineup-based performance using play-by-play data. By reconstructing the on-court quintets, we can assess the efficiency of each lineup and their usage patterns, including total minutes played and substitution behavior.

Player Analysis. Each player is evaluated based on volume (Plays), usage in offensive actions (OER), and their individual Four Factors. These metrics are compared internally (among teammates) and externally (against players in the same tournament).

Additional layers of insight may be generated through clustering algorithms, grouping players into profiles based on statistical similarity. These clusters provide a meaningful way to assign player "tags" or roles, useful in scouting unfamiliar athletes.

Key Metrics and Formulas

The following metrics will be used to analyze both team and player performance. All metrics are calculated using values extracted and cleaned from the play-by-play and box-score datasets.

Possessions (Estimated)

$$Poss = FGA + 0.44 \times FTA - OREB + TOV \tag{1}$$

Pace (Game Tempo)

$$Pace = \frac{40 \times (Possessions_{Team} + Possessions_{Opponent})}{2 \times MinutesPlayed} \tag{2}$$

Offensive Efficiency Rating (OER)

$$OER = \frac{PointsScored}{Possessions} \times 100 \tag{3}$$

Defensive Efficiency Rating (DER)

$$DER = \frac{PointsAllowed}{PossessionsAllowed} \times 100 \tag{4}$$

Four factors: Effective Field Goal Percentage (eFG%)

$$eFG\% = \frac{FGM + 0.5 \times 3PM}{FGA} \tag{5}$$

Four factors: Turnover Percentage (TOV%)

$$TOV\% = \frac{TOV}{FGA + 0.44 \times FTA + TOV} \tag{6}$$

Four factors: Offensive Rebound Percentage (OREB%)

$$OREB\% = \frac{OREB}{OREB + OpponentDREB} \tag{7}$$

Four factors: Free Throw Rate (FT Rate)

$$FTRate = \frac{FTM}{FGA} \tag{8}$$

7 References

References

[Arbués, 2020] Arbués, A. (2020). Bue stats: A gui-based tool for advanced basketball statistics retrieval. `https://www.upf.edu/web/adria-arbues/buestats`.

[Broman and Woo, 2005] Broman, K. W. and Woo, K. M. (2005). Data organization in spreadsheets. *The American Statistician*, 59(1):1–6.

[Crespo, 2023] Crespo, N. (2023). Bue stats 2.0: A web application for basketball data analysis. `https://buestats2.com`.

[Díaz, 2021] Díaz, P. (2021). Acb database project. `https://github.com/PabloDMC/ACB_DB/tree/main`. GitHub repository.

[Shufinskiy, 2020] Shufinskiy, D. (2020). Nba data scraping and analytics tools. `https://github.com/shufinskiy/nba_data`. GitHub repository.

[Vindel, 2022] Vindel, A. (2022). Scouting platform for spanish feb basketball (feb). `https://upcommons.upc.edu/handle/2117/379944?show=full`).

[Vindel, 2025] Vindel, A. (2025). Angt scouting automation tool. `https://github.com/vindeel98/microcredential`. Accessed June 2025.