# National University of Computer and Emerging Sciences



# Cricket Team Analysis and Prediction

## Project Deliverable 1

#### Group Members:

|  |  |  |
| --- | --- | --- |
| **Name** | **Roll Number** | **Section** |
| **Muhammad Sharjeel** | 22L-7526 | 5A |
| **Abdullah** | 22L-7500 | 5A |
| **Usman Ahmed** | 21L-6273 | 5A |
| **Ammar Ahmed** | 20L-0961 | 5A |

AI2002 Artificial Intelligence  
Spring 2023  
Department of Computer Science  
FAST-NU, Lahore, Pakistan

### Introduction

In this project we will predict the perfect and optimal cricket team of 11 players for T20 and ODI matches, as well as forecast the runs a player will score against a specific team in an upcoming match.

### Approach

The method used to scrape the data involves web scraping using the BeautifulSoup library in combination with curl\_cffi for handling HTTP requests. The process is broken down into the following steps:

1. **Importing Libraries**: We start by importing the necessary libraries, including BeautifulSoup for parsing HTML, requests from curl\_cffi for making HTTP requests, and pandas for data manipulation and storage.
2. **Handling Anti-Bot Detection**: Initially, we encountered a 403 Forbidden error when using the standard requests library due to anti-bot detection mechanisms on the target site. To bypass this, we switched to curl\_cffi, which allows us to impersonate a real browser (e.g., Chrome) by setting appropriate headers.
3. **Scraping Functions**: We defined two main functions:
   * scrape\_page(url): This function scrapes a single page of data. It sends an HTTP GET request to the specified URL, parses the HTML content using BeautifulSoup, and extracts the relevant table data into a pandas DataFrame.
   * scrape\_all\_pages(base\_url, max\_pages): This function handles pagination. It iterates through multiple pages of the target site, calling scrape\_page for each page, and concatenates the results into a single DataFrame.
4. **Data Extraction and Storage**: We defined a dictionary of URLs for different cricket formats and types (all-round, batting, bowling). For each URL, we called scrape\_all\_pages to extract the data and saved the resulting DataFrame to a CSV file.

### Sources

The data was scraped from the following URLs on ESPN Cricinfo:

1. Tests All-round: https://stats.espncricinfo.com/ci/engine/stats/index.html?class=1;spanmin1=24+Nov+2012;spanval1=span;team=7;template=results;type=allround
2. ODIs All-round: https://stats.espncricinfo.com/ci/engine/stats/index.html?class=2;spanmin1=24+Nov+2012;spanval1=span;team=7;template=results;type=allround
3. T20Is All-round: https://stats.espncricinfo.com/ci/engine/stats/index.html?class=3;spanmin1=24+Nov+2012;spanval1=span;team=7;template=results;type=allround
4. Tests Batting: https://stats.espncricinfo.com/ci/engine/stats/index.html?class=1;spanmin1=24+Nov+2012;spanval1=span;team=7;template=results;type=batting
5. ODIs Batting: https://stats.espncricinfo.com/ci/engine/stats/index.html?class=2;spanmin1=24+Nov+2012;spanval1=span;team=7;template=results;type=batting
6. T20Is Batting: https://stats.espncricinfo.com/ci/engine/stats/index.html?class=3;spanmin1=24+Nov+2012;spanval1=span;team=7;template=results;type=batting
7. Tests Bowling: https://stats.espncricinfo.com/ci/engine/stats/index.html?class=1;spanmin1=24+Nov+2012;spanval1=span;team=7;template=results;type=bowling
8. ODIs Bowling: https://stats.espncricinfo.com/ci/engine/stats/index.html?class=2;spanmin1=24+Nov+2012;spanval1=span;team=7;template=results;type=bowling
9. T20Is Bowling: https://stats.espncricinfo.com/ci/engine/stats/index.html?class=3;spanmin1=24+Nov+2012;spanval1=span;team=7;template=results;type=bowling
10. Best Baller Tests (2017 – 2023/24): https://stats.espncricinfo.com/ci/engine/player/681305.html?class=1;template=results;type=bowling;view=match
11. Best Baller ODIs (2016 – 2023/24): https://stats.espncricinfo.com/ci/engine/player/681305.html?class=2;template=results;type=bowling;view=match
12. Best Baller T20Is (2016 - 2024): https://stats.espncricinfo.com/ci/engine/player/681305.html?class=3;template=results;type=bowling;view=match
13. Best Batter Tests (2016\_17 - 2024): https://stats.espncricinfo.com/ci/engine/player/348144.html?class=1;template=results;type=batting;view=match
14. Best Batter ODIs (2015 – 2023/24): https://stats.espncricinfo.com/ci/engine/player/348144.html?class=2;template=results;type=batting;view=match
15. Best Batter T20Is (2016 - 2024): https://stats.espncricinfo.com/ci/engine/player/348144.html?class=3;template=results;type=batting;view=match

### Rationale

The choice of approach was influenced by several factors:

1. **Ease of Implementation**: Using BeautifulSoup and curl\_cffi is straightforward and well-documented. This combination allows for quick setup and efficient data extraction without the need for complex configurations.
2. **Data Accessibility**: The target site, ESPN Cricinfo, provides structured HTML tables that are relatively easy to parse with BeautifulSoup. This made it a suitable candidate for web scraping.
3. **Handling Anti-Bot Mechanisms**: The initial 403 Forbidden error indicated the presence of anti-bot detection. Switching to curl\_cffi allowed us to impersonate a real browser, effectively bypassing these restrictions and ensuring successful data retrieval.
4. **Scalability**: The approach is scalable, as the scrape\_all\_pages function can handle multiple pages of data. This is crucial for extracting comprehensive datasets that span several pages.
5. **Specific Site Requirements**: ESPN Cricinfo's structure and anti-bot mechanisms required a flexible and robust solution. curl\_cffi provided the necessary capabilities to handle these requirements, making it the optimal choice for this project.