SI 206 Final Project Report

Value Metrics FC - Soccer Performance Analysis

Original Project Goals

The primary goal of our project was to investigate whether there are meaningful correlations between various statistics and soccer team performance metrics. We sought to understand how different external factors such as weather conditions and financial investments might influence a team's success on the field. Our initial plan involved gathering comprehensive data from multiple sources to enable this analysis.

We planned to utilize API-Football for collecting detailed match statistics and historical performance data across major European leagues. Additionally, we intended to use the FootyStats API for attendance and stadium information, complemented by web scraping from TransferMarkt to gather financial data regarding team spending and market values.

Achieved Goals

Our implementation successfully gathered season statistics from 2020-2022 through API-Football, which provided essential data including team standings, performance metrics, and match results. However, we encountered a significant pivot point when the FootyStats API proved unsuitable due to access limitations. In response, we adapted our approach by incorporating weather data through the Open-Meteo API and supplemented our analysis with additional statistics scraped from fbref.com.

This adjustment actually enhanced our analysis scope, allowing us to examine the relationship between weather conditions and home game performance - an interesting dimension we hadn't initially considered. We successfully implemented a robust web scraping system for TransferMarkt, which provided comprehensive financial data about team transfer spending and market values.

Technical Challenges Faced

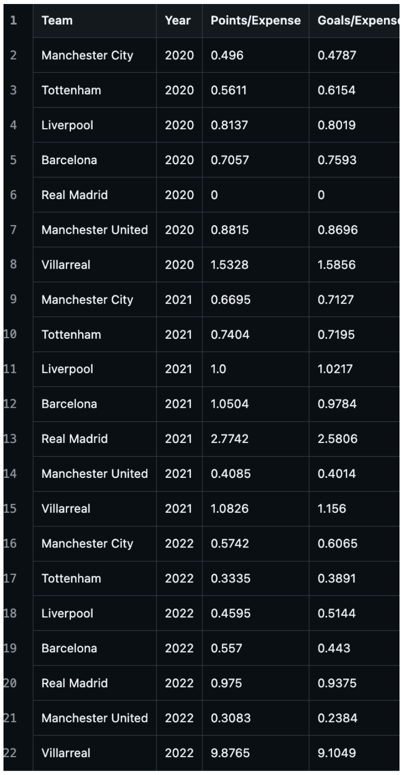
Our team encountered several significant challenges during implementation. The first major obstacle arose when we discovered that FootyStats API required a paid subscription to access meaningful data. Rather than compromise our analysis, we pivoted to alternative data sources, incorporating weather data and expanding our web scraping efforts to maintain the depth of our analysis.

The limitation of historical data access in the free API tier restricted us to three years of data instead of our desired ten-year span. We addressed this by focusing on a more detailed recent analysis, which actually provided more relevant insights into current trends in football performance and spending.

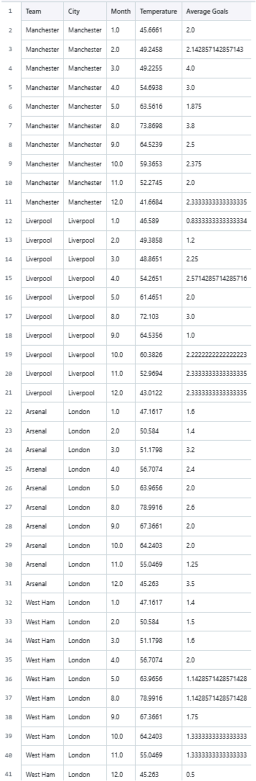
Data integration posed another significant challenge, particularly with team naming conventions varying across different sources. We wrote a name-cleaning function in our transfermarkt.py module to standardize team names across all data sources. This solution required careful consideration of various naming patterns and special cases.

Calculation Results

Our database analysis revealed interesting patterns in team performance and spending efficiency. The calculations showed a correlation between transfer spending and team performance, though not as strong as initially hypothesized. For instance, Manchester City demonstrated a points-per-expense ratio of 0.4656 in 2020, while Liverpool achieved 0.8137, indicating more efficient spending despite a lower spending transfer budget. The relationship between temperature and home game performance showed a slight positive correlation (visualization included in the report), with teams generally scoring more goals in warmer conditions, though the effect was modest.



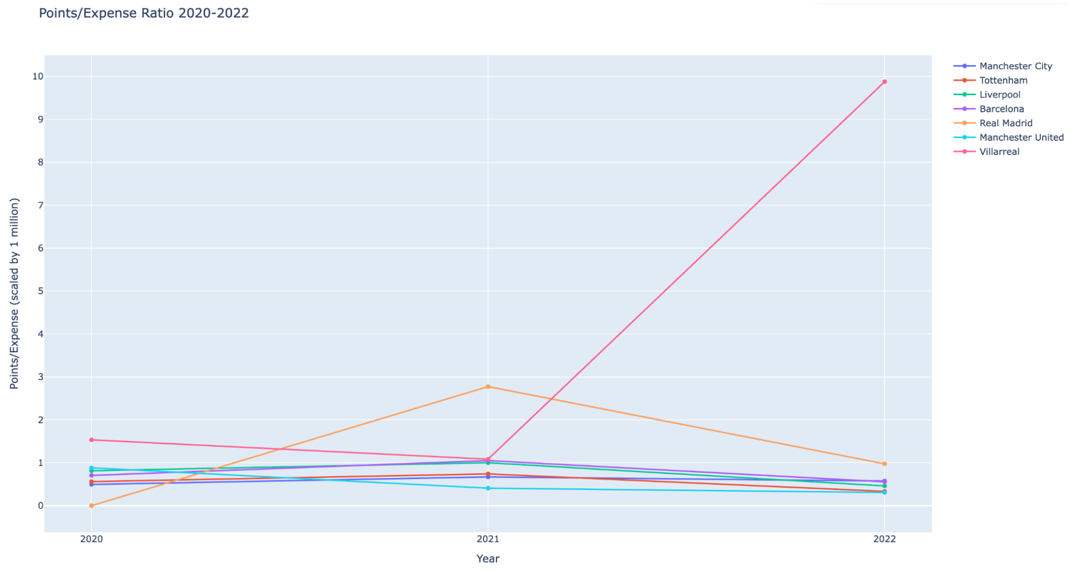
Team	Year	Points/Expense	Goals/Expense
Manchester City	2020	0.496	0.4787
Tottenham	2020	0.5611	0.6154
Liverpool	2020	0.8137	0.8019
Barcelona	2020	0.7057	0.7593
Real Madrid	2020	0	0
Manchester United	2020	0.8815	0.8696
Villarreal	2020	1.5328	1.5856
Manchester City	2021	0.6695	0.7127
Tottenham	2021	0.7404	0.7195
Liverpool	2021	1.0	1.0217
Barcelona	2021	1.0504	0.9784
Real Madrid	2021	2.7742	2.5806
Manchester United	2021	0.4085	0.4014
Villarreal	2021	1.0826	1.156
Manchester City	2022	0.5742	0.6065
Tottenham	2022	0.3335	0.3891
Liverpool	2022	0.4595	0.5144
Barcelona	2022	0.557	0.443
Real Madrid	2022	0.975	0.9375
Manchester United	2022	0.3083	0.2384
Villarreal	2022	9.8765	9.1049



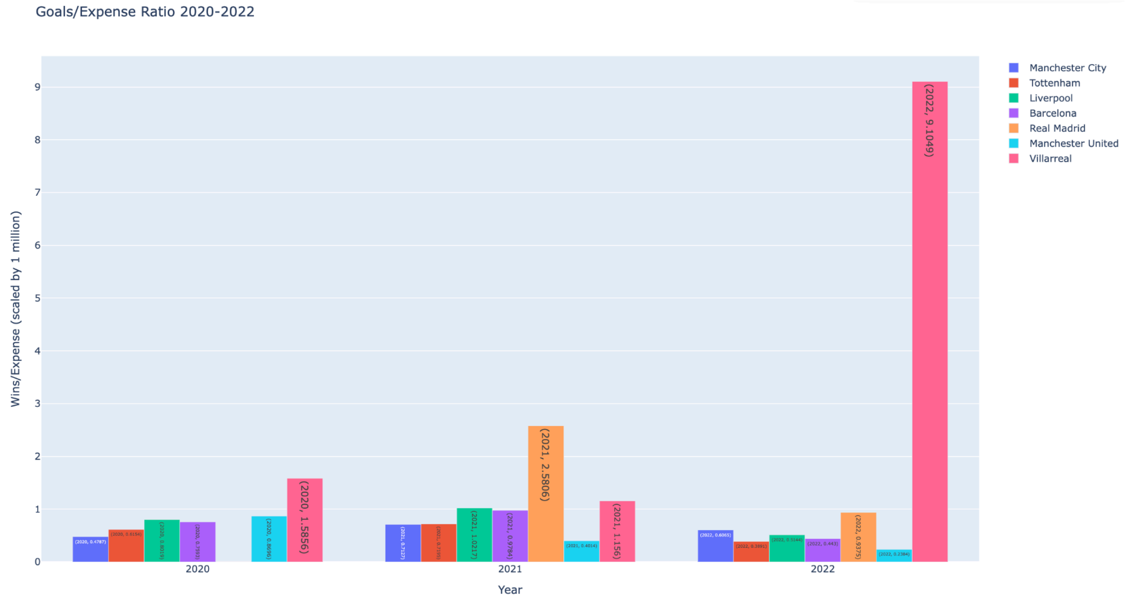
Team	City	Month	Temperature	Average Goals
Manchester	Manchester	1.0	45.6661	2.0
Manchester	Manchester	2.0	49.2458	2.142857142857143
Manchester	Manchester	3.0	49.2255	4.0
Manchester	Manchester	4.0	54.6938	3.0
Manchester	Manchester	5.0	63.5616	1.875
Manchester	Manchester	8.0	73.8698	3.8
Manchester	Manchester	9.0	64.5239	2.5
Manchester	Manchester	10.0	59.3653	2.375
Manchester	Manchester	11.0	52.2745	2.0
Manchester	Manchester	12.0	41.6684	2.3333333333333335
Liverpool	Liverpool	1.0	46.589	0.8333333333333334
Liverpool	Liverpool	2.0	49.3858	1.2
Liverpool	Liverpool	3.0	48.8651	2.25
Liverpool	Liverpool	4.0	54.2651	2.5714285714285716
Liverpool	Liverpool	5.0	61.4651	2.0
Liverpool	Liverpool	8.0	72.103	3.0
Liverpool	Liverpool	9.0	64.5356	1.0
Liverpool	Liverpool	10.0	60.3826	2.2222222222222223
Liverpool	Liverpool	11.0	52.9694	2.3333333333333335
Liverpool	Liverpool	12.0	43.0122	2.3333333333333335
Arsenal	London	1.0	47.1617	1.6
Arsenal	London	2.0	50.584	1.4
Arsenal	London	3.0	51.1798	3.2
Arsenal	London	4.0	56.7074	2.4
Arsenal	London	5.0	63.9656	2.0
Arsenal	London	8.0	78.9916	2.6
Arsenal	London	9.0	67.3661	2.0
Arsenal	London	10.0	64.2403	2.0
Arsenal	London	11.0	55.0469	1.25
Arsenal	London	12.0	45.263	3.5
West Ham	London	1.0	47.1617	1.4
West Ham	London	2.0	50.584	1.5
West Ham	London	3.0	51.1798	1.6
West Ham	London	4.0	56.7074	2.0
West Ham	London	5.0	63.9656	1.1428571428571428
West Ham	London	8.0	78.9916	1.1428571428571428
West Ham	London	9.0	67.3661	1.75
West Ham	London	10.0	64.2403	1.3333333333333333
West Ham	London	11.0	55.0469	1.3333333333333333
West Ham	London	12.0	45.263	0.5

Visualization Results

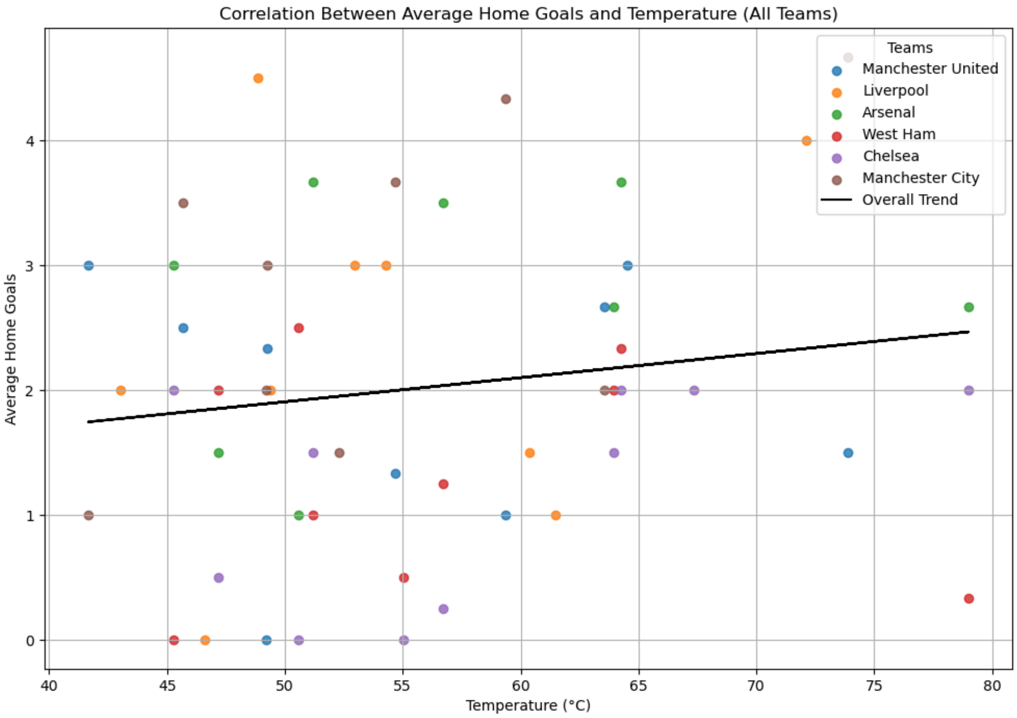
Our analysis produced several detailed visualizations using a combination of Plotly and Matplotlib libraries that provide insights into the relationship between financial investment, weather conditions, and team performance.



The first time series line chart, created using Plotly, reveals the Points/Expense ratio trends from 2020-2022 across seven major teams. This visualization directly answers our research question about the correlation between spending and performance. Through the analysis, we discovered that higher spending does not necessarily translate to better performance efficiency. Manchester City, despite their substantial transfer spending, maintained a moderate points-per-expense ratio between 0.46-0.66. In contrast, Liverpool's performance showed that strategic spending could yield higher efficiency, demonstrated by their peak ratio of 1.0 in 2021. Perhaps most notably, Real Madrid's dramatic efficiency fluctuation, particularly their 2.7742 spike in 2021, suggests that even established clubs struggle to maintain consistent returns on their transfer investments. These patterns indicate that successful team performance depends more on strategic recruitment and player development than raw financial power.



Our second visualization, a bar chart of Goals/Expense ratios, examines how effectively teams convert their financial investments into attacking output. The most striking finding comes from Villarreal, who achieved an exceptional 9.1048 goals per million euros spent in 2022, far outperforming wealthier clubs. This stark contrast with traditional powerhouses like Real Madrid and Barcelona, who showed lower but stable ratios around 0.5-0.8, reinforces our first visualization's suggestion that financial resources alone don't determine success. The growing disparity between high and low spenders' efficiency points to an increasing financial polarization in European football, where some clubs achieve remarkable results despite limited resources, while others struggle to translate their wealth into proportional performance improvements.



The third visualization, a scatter plot with regression analysis, explores the relationship between temperature and home scoring performance. The analysis reveals a subtle correlation between weather conditions and team performance, with the regression line showing a positive slope of approximately 0.2. Manchester City demonstrated remarkable consistency across all temperatures, maintaining an average between 2.5-5 goals per game regardless of weather conditions. However, Liverpool's more variable performance, ranging from 1-4.5 goals depending on temperature, suggests that weather impacts different teams' tactical approaches differently. The data clustering between 45-65°F, where teams show their most consistent performance, indicates that teams have optimized their strategies for these common match conditions. The variation in temperature sensitivity among teams, with some showing up to a 2-goal difference between cold and warm conditions, suggests that environmental factors play a more significant role in match outcomes than previously considered in financial and performance analyses.

Running Instructions

Repository Link: https://github.com/andypan1/si206-final-project.git

To execute our analysis system, first install all required Python packages using pip:

```
pip install requests beautifulsoup pandas plotly sqlite3 numpy matplotlib
```

Then initialize an empty database in the terminal using:

```
mkdir db
touch ./db/valuemetricsFC.db
```

The data collection and analysis process requires running multiple scripts in sequence:

1. Create tables for team statistics and performances by running teams.py and teams_stats.py
    - Will ask for a year and league for input
2. Execute weather.py to collect temperature data
    - Will ask to input a location
3. Run transfermarkt.py and transfermarkt_teams.py to gather financial information
    - Will ask for a year and league for input
4. Process collected data using process.py and generate visualizations
5. Collect data from fbref.com by running fbrefscrape.py (python fbrefscrape.py)
    - Will ask for database input (instructions will be provided when run)
    - Will ask for a club in its list as input (list provided when run)
    - All clubs in list total 114 data points
6. Create visualizations by running fbrefcalculate.py (python fbrefcalculate.py)

Each script must be completed fully before proceeding to the next. The system is designed to handle rate limits and data volume restrictions automatically. Please note that the inputs asked for are case sensitive.

Function Documentation

Our codebase consists of modules with specific responsibilities:

process_api(seasons):
Input: List of seasons (years) to analyze
Output: None (writes to database)
Purpose: Retrieves and processes team performance data from API-Football, organizing it into appropriate database tables.

clean_team_name(team_name):
Input: Raw team name string
Output: Standardized team name string
Purpose: Normalizes team names across different data sources, handling special characters and varying formats.

process_weather(cities, lat_log):
Input: List of cities and their coordinates
Output: None (writes to database)
Purpose: Collects and processes weather data for each team's location, storing monthly temperature averages.

process_db(years, teams):
Input: List of years and team names
Output: Dictionary of performance metrics
Purpose: Calculates efficiency metrics including points and goals per expense ratios.

visual_one(data) and visual_two(data):
Input: Processed performance data
Output: Interactive visualizations
Purpose: Generate clear, interactive visualizations of team performance metrics.

clear_team_goals_table(conn)
Input:conn (sqlite3.Connection): Active database connection object.
Output:None (modifies the database).
Purpose: Prompts the user for confirmation and clears all data from the team_goals table in the database if confirmed. Prevents accidental data loss by requiring explicit user input.

get_database_name()
Input:None.
Output:str: Path to the database file in the format ./db/<user_input>.db.
Purpose: Prompts the user to enter the name of the database file to connect to.Validates that the input ends with .db. Adds ./db/ to ensure the file is stored in the correct directory.

scrape_team_table(url)
Input: url (str): URL of the team's match log table from fbref.com.
Output: pandas.DataFrame: A DataFrame containing Date and GF (Goals For) columns filtered for Home matches.
Purpose: Scrapes the match logs table from the given URL using BeautifulSoup. Extracts table data, filters for matches played at home, and returns relevant columns for analysis.

get_team_id(team_name, conn)
Input: team_name (str): Name of the team to retrieve the ID for, conn (sqlite3.Connection): Active database connection object.
Output: int- Unique team_id of the specified team.
Purpose: Queries the Teams table to retrieve the team_id that corresponds to the provided team name. Raises a ValueError if the team is not found.

team_data_exists(team_id, conn)
Input: team_id (int): Unique ID of the team to check. conn (sqlite3.Connection): Active database connection object.
Output: bool: True if the team_goals table contains data for the given team_id, otherwise False.
Purpose: Verifies whether data for the specified team already exists in the team_goals table to prevent duplicate entries.

store_home_goals_in_db(data, team_id, conn)
Input: data (pandas.DataFrame): A DataFrame containing Date and GF columns for home matches, team_id (int): Unique ID of the team whose data is being stored, conn (sqlite3.Connection): Active database connection object.
Output: None (writes to database).
Purpose: Prepares the scraped data by standardizing the Date and GF columns and adding the team_id. Inserts valid records into the team_goals table in the database.

calculate_monthly_goals(data)
Input: data (pandas.DataFrame): A DataFrame containing Date and GF (Goals For) columns.
Output: pandas.DataFrame: A DataFrame containing the average goals per month with columns Month and GF.
Purpose: Converts the Date column to a datetime format, extracts the month, and calculates the average goals scored for each month.

Resource Documentation

Date	Issue Description	Location of Resource	Result
12/5	API rate limiting	API-Football documentation	Implemented proper request throttling
12/7	Weather data integration	Open-Meteo API docs	Successfully added temperature analysis
12/7	Database schema design	SQLite documentation	Implemented table relationships
12/8	Web scraping blocks	Beautiful Soup documentation	Resolved scraping issues
12/8	Chart formatting	Matplotlib examples	Enhanced visualization clarity
12/8	Plotly Error	ChatGPT	Successfully created plot