Alvin Jang
DS210
Final ProjectWriteup

Soccer Data Graph Analysis Project

Objective:

The primary objective of my project is to analyze relationships and patterns in soccer player data using graph theory concepts. Using the FIFA 2017 dataset, my project constructs a graph-based representation of soccer players and their clubs to extract meaningful insights. Each player is represented as a node in the graph, and edges connect players who belong to the same club. The project includes features such as player comparisons, shortest path calculations (e.g., "Six Degrees of Separation"), clustering analysis, and subgraph density evaluation. These functionalities provide a deeper understanding of player relationships and team dynamics while demonstrating the practical application of graph theory principles.

My Dataset:

I used the "FIFA17_official_data (1).csv", which contains detailed statistics about soccer players, including attributes like name, age, nationality, club, overall rating, potential, position, and various skill metrics. The data is parsed to create nodes (players) and edges (connections) in the graph. Key attributes such as player overall ratings and performance metrics are used to compare players and evaluate their roles within clubs and teams. The dataset serves as the foundation for all analysis and graph-based operations performed in the project.

Overview of Key Features:

The project is modular and consists of five main Rust modules: "graph.rs", "analysis.rs", "clustering.rs", "main.rs", and "test.rs". Each module contributes specific functionality to meet the project's goals. The "graph.rs" module is responsible for building the graph structure and implementing basic graph operations. It defines the "Graph" struct, which contains an adjacency list to represent connections between nodes. Functions like "add_edge" and "shortest_path" are implemented here. The "add_edge" function connects two nodes by adding an edge between them in the adjacency list. For instance, calling "add_edge("A", "B")" establishes a connection between players A and B, representing their shared club membership. The "shortest_path" function implements the Breadth-First Search (BFS) algorithm to find the shortest path between two players in the graph. This can be used to analyze how players are indirectly connected through shared teammates.

The "analysis.rs" module provides tools to explore the graph's properties. One key function is "degree_distribution", which calculates the distribution of node degrees (the number of connections each player has). For example, this function might return a result showing that 1,500 players have one connection, 700 players have two connections, and so on. Another function, "average_degree", computes the average number of connections across all nodes in the graph, providing a summary metric of graph connectivity. The "densest_subgraph" function identifies subgraphs with the highest edge-to-node density, which can be useful for finding

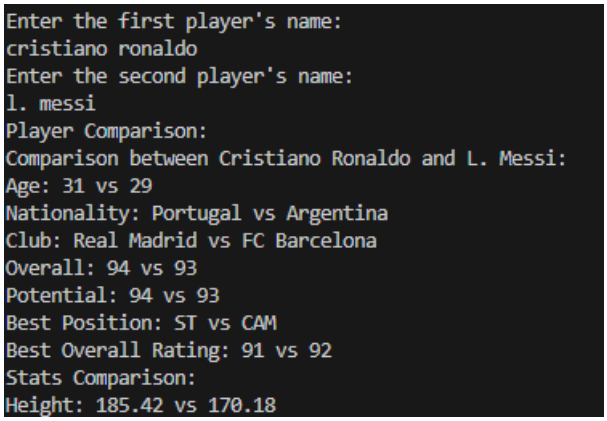
highly interconnected groups of players or clubs. This function iterates over nodes and evaluates the density of each potential subgraph, returning the densest one along with its calculated density.

The "clustering.rs" module focuses on analyzing player-club cohesion using clustering coefficients. The "clustering_coefficient" function computes the local clustering coefficient for a specific player, measuring the likelihood that their teammates are also teammates with each other. For example, if a player has three teammates, and two of them are also connected, the clustering coefficient would be "0.67". The "global_clustering_coefficient" function calculates the overall clustering coefficient for the entire graph, averaging the local coefficients. Additionally, the "average_clustering_coefficient" function provides a broader measure of graph cohesion by averaging the clustering coefficients of all nodes. These metrics help analyze the tightness of player-club relationships and the overall interconnectedness of the graph.

The "main.rs" file serves as the entry point for running the project and demonstrating its functionalities. It includes a player comparison feature, which allows users to compare two players based on attributes such as age, nationality, overall rating, and specific skills like dribbling and stamina. The "compare_players" function outputs a detailed comparison report, highlighting similarities and differences between the selected players. Another key feature is the "Six Degrees of Separation" implementation, which uses the "shortest_path" function from "graph.rs" to find the shortest connection path between two players. For instance, it can determine how Lionel Messi and Cristiano Ronaldo are connected through shared teammates or clubs.

The project also includes a comprehensive testing framework in "test.rs". This module contains unit tests for all major functionalities, ensuring the correctness and robustness of the code. For example, the "test_degree_distribution" function verifies that the degree distribution is calculated accurately, while the "test_shortest_path" function ensures the BFS algorithm correctly identifies paths between nodes. The tests are designed to cover edge cases and validate the expected behavior of the functions. Running "cargo test" confirms that all features work as intended and that the project meets the quality standards outlined in the rubric.

The output of the project includes detailed player comparisons, graph metrics, and insights into player relationships. For example, when I compare two of the most famous soccer players in the world, Cristiano Ronaldo and Lionel Messi, this is the output that I get:



```
Height: 185.42 vs 170.18
Weight: 176 vs 159
Crossing: 84 vs 77
Finishing: 93 vs 95
HeadingAccuracy: 85 vs 71
ShortPassing: 83 vs 88
Volleys: 88 vs 85
Dribbling: 92 vs 97
Curve: 81 vs 89
LongPassing: 76 vs 90
BallControl: 77 vs 87
Acceleration: 93 vs 95
SprintSpeed: 91 vs 92
Agility: 92 vs 87
Reactions: 90 vs 90
```

```
Reactions: 90 vs 90
Balance: 96 vs 95
ShotPower: 63 vs 95
Jumping: 92 vs 85
Stamina: 95 vs 68
Strength: 92 vs 74
LongShots: 80 vs 59
Interceptions: 90 vs 88
Positioning: 63 vs 48
Vision: 29 vs 22
```

The "Six Degrees of Separation" feature provides a connection path like this:

```
Finding connection path...
Connection path found:
1. cristiano ronaldo
2. danilo
3. felipe
4. joão teixeira
5. j. ward
6. m. phillips
7. c. jones
8. t. walker
9. n. smith
10. s. harrison
11. j. martin
12. bruno
13. l. suárez
14. l. messi
```

In conclusion, this project successfully applies graph theory to analyze soccer player data, providing valuable insights into player relationships and team dynamics. The modular structure, comprehensive analysis tools, and robust testing ensure the project adheres to the rubric and delivers meaningful results. By combining graph construction, clustering analysis, and player comparisons, the project demonstrates the power of graph-based analysis in sports data analytics.