Software Implementation and Testing Document

For

Group 3

Version 1.0

Authors:

Kevin R

Luke M

Daniel T

Christopher T

1. Programming Languages (5 points)

Python backend and Javascript frontend, React for the framework.

2. Platforms, APIs, Databases, and other technologies used (5 points)

- pandas
- matplotlib
- numpy
- scikit-learn
- cartopy
- keras
- keras-tuner
- Tensorflow
- Juypter Notebook

3. Execution-based Functional Testing (10 points)

For this increment, functional testing focused on verifying the correctness of our data aggregation pipeline. Since our primary data source is the Basketball Reference Web Scraper, we performed several tests to ensure that the extracted data matched the expected statistical formats for key player metrics such as points, rebounds, and assists. Specifically, we:

- Validated that the scraper retrieved complete and accurate data for multiple seasons.
- Checked that the extracted data conformed to the expected schema (column names, data types, and formats).
- Cross-referenced sample data points against Basketball Reference to confirm accuracy.

4. Execution-based Non-Functional Testing (10 points)

Non-functional testing at this stage primarily involved:

- Scalability Testing: By running the scraper on different historical periods (e.g., single season vs. multiple seasons), we assessed how well our system handles increasing data volumes.
- Collaboration & Development Environment Testing: Using VS Code Live Share, we ensured that all team members could collaborate in real-time without experiencing connectivity or synchronization issues.
- Data Integrity Checks: We reviewed our extracted datasets to ensure completeness and consistency, verifying that all relevant player statistics were properly structured and formatted before use in model training.

While UI performance testing is not applicable yet, we plan to conduct response time and load testing in later increments as the frontend develops.

5. Non-Execution-based Testing (10 points)

Since development is still in the early stages, our primary non-execution-based testing involved code reviews, inspections, and structured walkthroughs:

- Code Reviews: Before finalizing the data aggregation scripts, team members reviewed each other's contributions to ensure clarity, efficiency, and adherence to best coding practices.
- Walkthroughs & Team Discussions: During team meetings, we walked through the logic of our data extraction approach, identifying and addressing potential edge cases (e.g., missing player stats, handling different data formats).
- Issue Tracking & Documentation: We documented challenges and proposed solutions in our issue tracker, ensuring structured debugging and accountability.

These activities allowed us to refine our approach, improving both data accuracy and system efficiency before moving forward with model training.