Project Scope and Plan: LifeLens - Sleep Score Analysis - Final Iteration

Muhaiminul Ashraf

November 17, 2024

1 Project Scope

The aim of this project is to design and implement a health recommendation system that analyzes user sleep data to generate actionable insights and personalized recommendations for improving overall health. By leverage graph-based modeling, the system provides a deatiled understanding of sleep patterns, deviations from optimal benchmarks, and actionable recommendations

The project's main objectives are:

- Model user sleep cycles using a graph-based approach to analyze temporal patterns and relationships between consecutive sleep cycles.
- Calculate and compare sleep performance metrics (light, deep, and REM sleep) against scientifically established benchmarks.
- Generated detailed visualizations to compare actual and optimal sleep durations, highlighting areas of improvement.
- Lay the groundwork for personalized health recommendations based on analyzed sleep patterns and overall sleep performance.

The expected outcomes include:

- A functional backend system for sleep performance analysis and visualization
- Detailed visualizations showcasing deviations from optimal sleep benchmarks.
- Comprehensive documentation and a final project report summarizing findings and insights.

2 Project Plan

2.1 Timeline

The overall timeline for the project is divided into phases:

- **Week 1 (October 7 - October 13)**: Define project scope, establish team roles, and outline skills/tools.
- **Week 2 (October 14 - October 20)**: Begin development, set up the project repository, and start coding basic system functionalities.
- **Week 3 (October 21 - October 27)**: Continue coding, work on backend integration, and start writing technical documentation.
- **Week 4 (October 28 - November 3)**: Complete the backend and integrate frontend elements. Begin PowerPoint presentation.
- **Week 5 (November 4 - November 10)**: Finalize the system, conduct testing, and continue with the report.

- **Week 6 (November 11 - November 17)**: Revise and finalize the technical report and the PowerPoint presentation.
- **Week 7 (November 18 - November 28)**: Final presentation, report submission, and project closure.

2.2 Milestones

Key milestones include:

- Project Scope and Plan (October 7).
- GitHub Repository Setup and Initial Development (October 9).
- Backend Completion (November 3).
- Graph Integration and Visualization (November 10).
- Final System Testing and Report Draft (November 17).
- Final Presentation and Report Submission (November 28).

2.3 Roles

Muhaiminul: Responsible for frontend design of the health recommendation system, graph system design to connect various nodes, and proper algorithm design to rank the importance of each health recommendation.

3 Progress Review (November 17)

3.1 Progress Update

During this iteration, significant progress was made in the backend implementation. The key achievements include:

- Development of a graph-based system in `SleepGraph.py`, where nodes represent individual sleep cycles, storing metrics such as light sleep, deep sleep, REM sleep, total sleep, and sleep performance ratings.
- Introduction of a function to compute overall sleep performance based on the ratio of total sleep to sleep need.
- Implementation of functions to compare actual sleep durations for light, deep, and REM phases against optimal benchmarks and calculate differences and performances.
- Creation of the `sleepPlotRender.py` script to generate detailed visualizations for sleep metrics over time, including:
  - Line plots for actual vs. optimal sleep durations.
  - Bar charts for differences between actual and optimal sleep durations.

3.2 Key Backend Features

- Graph-Based Data Representation: Sleep cycles are modeled as a graph with nodes storing detailed attributes and edges connecting consecutive cycles.
- Performance Metrics: Functions to evaluate sleep performance and compare actual vs. optimal values for light, deep, and REM sleep.
- Visualization: Visual tools for understanding trends and deviations in sleep performance across multiple cycles.

3.3 Challenges and Solutions

- Handling Missing Data: Ensured that missing values in the dataset are handled gracefully, avoiding disruptions in calculations or visualizations.
- Graph Integration: Designed a flexible graph structure to support both node-level and graph-level analyses, enabling temporal insights into user sleep patterns.

4 Updated Plan

4.1 Justification for Updates

Due to the incorporation of graph-based modeling and visualization, additional time was allocated for integrating these features and testing their accuracy. The project plan has been adjusted to include:

- Enhanced visualization capabilities for final reporting and presentation.
- More rigorous testing of graph-based algorithms to ensure accuracy in health recommendations.

5 Visualizations and Insights

5.1 Sample Outputs

Below are the visualizations generated from `sleepPlotRender.py`:

- **Light Sleep Analysis**:
  - Actual vs. Optimal Light Sleep durations.
  - Differences displayed as bar charts.
- **Deep Sleep Analysis**:
  - Actual vs. Optimal Deep Sleep durations.
  - Differences displayed as bar charts.
- **REM Sleep Analysis**:
  - Actual vs. Optimal REM Sleep durations.
  - Differences displayed as bar charts.

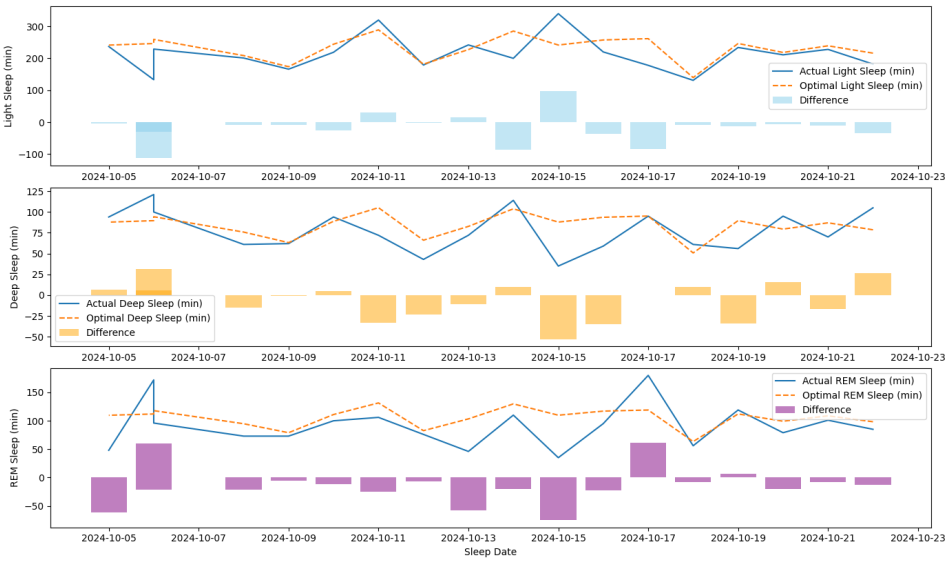

Figure 1: Sleep Performance Metrics as measured by the difference between optimal sleep and measured sleep.

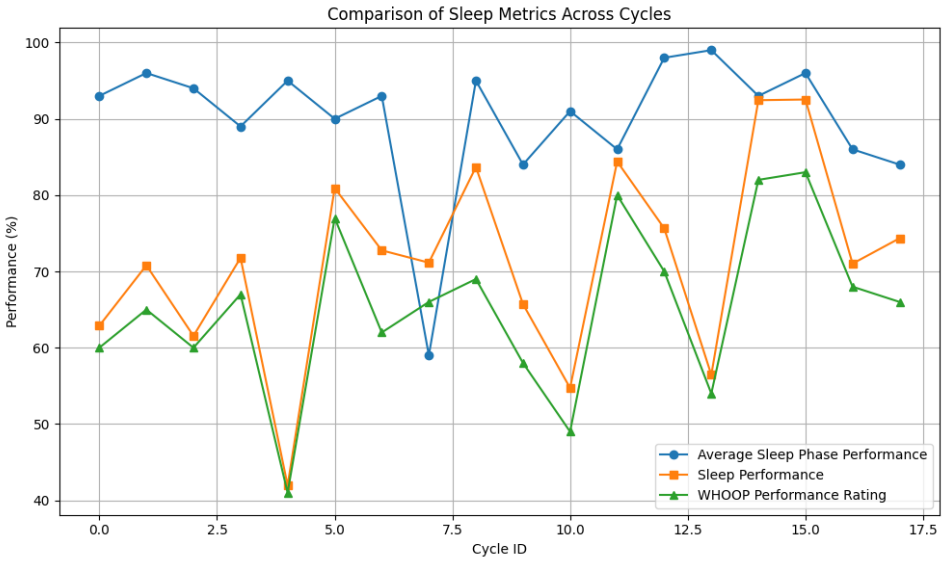

Figure 2: Sleep Performance Metrics as measured by the difference between Whoop's performance rating and the average sleep performance and calculated sleep performance.

6 Future Work

- Final integration of graph and visualization components into the main system.
- Testing with simulated data to verify the accuracy of recommendations.
- Refinement of the user interface for seamless data interpretation.

7 Appendix

7.1 Code Listings

- `SleepGraph.py`: The script that implements the graph-based system for analyzing sleep metrics.
- `sleepPlotRender.py`: The visualization script for generating plots of sleep performance metrics.