

Athlete Load Management & Performance Optimization Platform

Team Member 1 (Health Data Science): Harsha Prakash

Database Development:

- Design and implement medical/recovery database schema (injuries, recovery_metrics, risk_predictions tables)
- Define appropriate data types, constraints, and indexes for medical tables
- Create foreign key relationships linking to athlete master table
- Document medical schema with comments explaining each field's purpose

Data Generation:

- Develop Python script to generate synthetic medical/injury data (50 athletes, 6 months)
- Create realistic injury patterns with appropriate severity distributions
- Generate recovery progression data with logical timelines
- Ensure injury recurrence patterns follow sports medicine literature

SQL Reports:

- Injury distribution analysis by muscle group and severity levels
- Recovery progress tracking with timeline analysis and milestones
- Predicted vs actual injury outcomes comparison with accuracy metrics
- Risk factor analysis ranked by athlete demographics and history

Machine Learning:

- Build injury risk prediction model (Random Forest/XGBoost)
- Engineer medical features (injury history, recovery rates, recurrence patterns)
- Implement model evaluation pipeline (Precision, Recall, F1, AUC-ROC)
- Create feature importance analysis for medical risk factors
- Generate confusion matrix and ROC curves for model validation

Documentation:

- Document medical data generation methodology and parameters
- Write detailed comments for injury prediction model
- Create data dictionary for all medical/recovery fields
- Explain medical domain assumptions and constraints

Team Member 2 (Sports Data Science): Samuel Greeman

Database Development:

- Design and implement performance/training database schema (training_sessions, performance_metrics, load_calculations tables)
- Define appropriate data types for GPS, heart rate, and training data
- Create foreign key relationships linking to athlete master table
- Document performance schema with sports science context

Data Generation:

- Develop Python script to generate synthetic performance data (50 athletes, 6 months)
- Create realistic training load patterns (periodization, tapering)
- Generate GPS metrics (distance, speed, acceleration data)
- Ensure training patterns align with position-specific requirements

SQL Reports:

- Workload spike analysis and correlation with injury probability
- Training load averages segmented by position and time period
- Weekly performance load summaries for individual athletes
- Team-wide load intensity patterns and distribution analysis

Algorithm Development:

- Implement ACWR (Acute:Chronic Workload Ratio) calculations
- Develop training monotony and strain algorithms
- Create load aggregation functions for different time windows
- Build performance trend analysis functions
- Generate training load visualizations and patterns

Documentation:

- Document performance data generation methodology
- Write detailed explanations of load calculation formulas
- Create data dictionary for all performance/training fields
- Explain sports science principles behind metrics