

NBA Tactical Recommendations System

Final Project - Detailed Work Plan

Project Information

Project Name: NBA Tactical Recommendations System - Optimal Timeout Prediction

Track: Academic

Academic Year: 2025-2026

1. Introduction and Problem Definition

This project focuses on developing a Machine Learning-based tactical recommendations system for NBA coaches, centered on a crucial question: **When is the optimal time to call a Timeout?** This decision is one of the most important tactical choices during a game, requiring deep understanding of the concept of **Momentum** in basketball.

1.1 Why? (Motivation)

NBA coaches face complex tactical decisions in real-time. One critical decision is when to call a Timeout to stop an opponent's run, reorganize the team, or capitalize on a game situation. Currently, these decisions are based primarily on intuition and experience, without algorithmic computational support. This project aims to provide a science-based tool to help maximize the effectiveness of Timeout decisions.

1.2 What? (Objective)

Primary Goal: Develop a Machine Learning model that predicts the optimal time to call a Timeout, using existing research definitions of Momentum as a key explanatory variable. **Research Question:** Does the timing of a Timeout (relative to Momentum state) affect its effectiveness?

2. Algorithmic Challenges and Complexity

2.1 The Central Challenge: Defining Momentum

Momentum is a complex, multi-dimensional concept with no single agreed-upon definition in sports science. The main algorithmic challenge of this project is:

- **Quantitative Definition:** How to convert an abstract concept ("momentum") into a measurable, computable metric?
- **Multiple Variables:** Momentum is affected by many parameters - scoring runs, defense, player fatigue, fouls, crowd, etc.
- **Time Dependency:** Momentum changes dynamically during the game - what was "positive momentum" 2 minutes ago can turn negative
- **Hidden Variables:** Some factors cannot be directly measured (team morale, psychological pressure)
- **Non-linearity:** The relationship between variables and Momentum is non-linear - e.g., a 6-0 run in the last minute has different impact than 6-0 at game start

2.2 Our Approach to Complexity

The project addresses this complexity through **combining existing research with original algorithmic work**:

- **Using Research-Based Momentum Definitions:** Analyzing academic papers and existing Momentum definitions as starting point
- **Advanced Statistical Modeling:** Using Machine Learning (Random Forest, Gradient Boosting) to identify complex patterns
- **Feature Engineering:** Creating sophisticated derived variables (e.g., "time-weighted run", "pace change")
- **Retrospective Prediction Approach:** Learning from historical data - examining effectiveness of Timeouts already taken

3. The Machine Learning Component

Machine Learning is the algorithmic core of this project. This is not just using an existing library, but a complete research process including:

3.1 The Learning Process

Data Collection & Preprocessing:

- Extracting Play-by-Play data from NBA API
- Identifying Timeout events and their temporal context
- Cleaning missing data and handling outliers

Feature Engineering:

- Calculating Score Margin over time
- Identifying scoring "Runs" (consecutive scoring sequences)
- Computing Momentum Score based on research
- Creating derived variables: game pace, cumulative fatigue, fouls

Model Selection & Optimization:

- Testing multiple algorithms (Logistic Regression, Random Forest, XGBoost)
- Hyperparameter Tuning via Grid Search / Random Search
- Cross-validation to prevent Overfitting

Performance Evaluation:

- Precision, Recall, F1-Score metrics
- ROC Curve and AUC
- Error Analysis - when does the model fail and why?

3.2 Algorithmic Innovation

The research contribution of this project is **examining the relationship between Momentum and Timeout Timing**. This is research that has not been systematically done in sports science:

- Does a Timeout at the "right moment" (when Momentum is against us) actually stop the opponent's run?
- Is there an "optimal time window" for calling a Timeout?
- Is a Timeout that's too early or too late less effective?
- What's the difference between different types of Momentum (scoring, defense, crowd) in the context of Timeouts?

4. Real-Time Component: Chatbot for Coaches

As part of the project requirement for a Real-Time element, we will develop an **interactive chatbot** that allows coaches (or assistant coaches) to receive immediate recommendations during games.

4.1 Architecture

The system is built in two stages:

- **Stage 1 - Training (Offline):** The model is trained on historical data and learns optimal patterns
- **Stage 2 - Prediction (Real-Time):** Assistant coach inputs current parameters via chatbot, model returns recommendation within seconds

4.2 User Interface

The chatbot will include **pre-defined prompts** where the user only fills in values:

- Current Quarter and Time Remaining
- Score Margin
- Opponent Run: X-X in Y minutes
- Our Team Run
- Key Players Fouls
- Timeouts Remaining
- Additional Notes (free text - optional)

Advantage of this approach: This is genuine Real-Time Inference - the model runs predictions in real-time on current data. The fact that input comes from the user (rather than automatic API) doesn't compromise legitimacy - it's a practical, user-friendly approach for real-world use.

5. Data Sources and Technical Infrastructure

5.1 Primary Sources

NBA API (nba_api Python package):

- Detailed Play-by-Play data for every game
- Timeout event identification (EVENTMSGTYPE)
- Score, Score Margin, game time
- Historical data spanning decades

Basketball-Reference (supplementary):

- Advanced player statistics
- Advanced Metrics (PER, Win Shares, etc.)

5.2 Development Environment

- **Python 3.7+** - Primary development language
- **nba_api** - NBA data access
- **pandas, numpy** - Data processing and analysis
- **scikit-learn** - ML algorithms
- **matplotlib, seaborn** - Visualization
- **Jupyter Notebooks** - Interactive research and development
- **Flask/Streamlit** - Chatbot development

6. Detailed Work Plan

Phase 1: Setup and Environment (Week 1)

- Install required Python packages
- Test NBA API connection and download sample data
- Set up organized project structure
- Initial reading of Momentum research papers

Phase 2: Research and Problem Definition (Weeks 2-3)

- Find and read 5-10 relevant academic papers
- Define Momentum - which definition will we use?
- Define "successful Timeout" - what are success criteria?
- Identify gaps in existing research
- Write Literature Review

Phase 3: Data Collection and Processing (Weeks 4-6)

- Download Play-by-Play data from multiple seasons
- Identify all Timeout events in the data
- Calculate Score Margins over time
- Identify "Runs" - algorithm for detecting scoring sequences
- Clean data and handle missing values
- Create unified, clean dataset

Phase 4: Feature Engineering (Weeks 7-8)

- Calculate Momentum Score based on research
- Create derived variables: game pace, fatigue, weighted momentum
- Analyze correlations between variables
- Feature Selection - choose important variables
- Create visualizations to understand the data

Phase 5: Building ML Models (Weeks 9-11)

- Split data into Train/Validation/Test sets
- Build simple Baseline Model (Logistic Regression)
- Build advanced models (Random Forest, XGBoost)

- Hyperparameter Tuning
- Cross-validation and performance evaluation
- Error Analysis - when does the model fail?

Phase 6: Chatbot Development (Weeks 12-14)

- Design user interface
- Build pre-defined prompts
- Integrate with trained model
- Test and optimize response times
- Add additional conversation capabilities (explanations, history)

Phase 7: Testing and Validation (Weeks 15-16)

- Unit Testing
- Integration Testing
- Validate results against known games
- Usability Testing
- Bug fixes and improvements

Phase 8: Documentation and Completion (Weeks 17-18)

- Write final report
- Prepare execution presentation
- Prepare poster for public day
- Prepare demonstrations
- Create explanatory video (optional)

7. Success Metrics and Quality Control

7.1 Quantitative Success Metrics

- **Precision:** Target - 75%+ (of "call timeout" recommendations, how many were correct?)
- **Recall:** Target - 70%+ (of all cases needing timeout, how many did we identify?)
- **F1-Score:** Target - 0.7+ (harmonic mean of Precision and Recall)
- **AUC-ROC:** Target - 0.75+ (overall model quality measure)
- **Chatbot Response Time:** Target - under 3 seconds

7.2 Qualitative Success Metrics

- Model can explain its decisions (Interpretability)
- Chatbot is user-friendly and easy to use
- Results are consistent with existing sports research knowledge
- System provides real value to coaches (based on feedback)

8. Summary and Course Requirements

This project combines all components required for an excellent final project:

- **Problem Definition:** Deep understanding of optimal Timeout question in Momentum context
- **Algorithms:** Original solution using ML, with complex handling of Momentum definition
- **Design:** Broad system design including data collection, ML model, and user interface
- **Programming:** Complex multi-layer software (Data Pipeline, ML Model, Chat Bot)
- **Results Testing:** Thorough evaluation with quantitative and qualitative metrics
- **Self-Learning:** Deep dive into data science, ML, NBA API, and sports research
- **Systematic Work:** Detailed work plan with quality controls at every stage

Two Central Open Questions the Project Will Address:

- **How to define Momentum quantitatively and precisely?** - This is the main algorithmic challenge
- **What defines a "successful Timeout"?** - This criterion will determine how we train and evaluate the model

Note: This work plan is a living document that will be updated during the project according to progress, research findings, and advisor guidance.