NBA Prediction System

Executive Summary

Project Introduction

The NBA Prediction System aims to provide a data-driven solution that forecasts NBA game outcomes. This document outlines the complete project approach, from initial data collection to model development and final deployment. The system will provide predictions including win probabilities, score projections and player performance forecasts in a CSV format.

Business Context

In the sports analytics makret, accurate NBA predictions serves multiple needs:

- Sports media outlets require data-driven content for engagement
- Fantasy sports platforms rely on effective performance projections
- Betting operators rely on accurate proabilistic forecasting

Solution

This project addresses these needs by creating a flexible, consistent and manageable prediction system built on sound data science principles.

Project Scope

The focus of this project will concentrate around the following:

- Collection and processing of historical NBA team and player data
- Development of predictive models for game outcomes and player statistics
- Daily prediction generation system for upcoming games
- Complete documentation and codebase suitable for production implementation

Objectives & Success Criteria

Primary Objectives

1. Develop a reliable data pipeline for NBA stats collection and processing
2. Create predictive models for game outcomes with measurable accuracy
3. Implement player performance prediction capabilities
4. Establish a system/program for daily prediciton generation
5. Deliver a well doccumented, workable codebase

Key Performance Indicators

1. Model Accuracy Metrics:
    - Team win prediction: >65% accuracy, log-loss <0.55
    - Score prediction: Mean Absolute Error <5.5 points
    - Player scoring prediction: Mean Absolute Error <3.5 points
    - Player rebounding prediction: Mean Absolute Error <1.5 rebounds
2. System Performances:
    - Data refresh completed in <15 minutes
    - Prediction generation complete in <10 minutes
3. Code quality:
    - 90% test coverage for core functionality
    - All modules pass linting standards

Executive Summary

Project Timeline & Milestones

Phase	Milestone	Deliverables	Target Completion
1: Setup & Foundation	Environment & data collection established	Project structure, GitHub repository, initial data collection scripts	Week 1
2. Data Pipeline	Complete data processing system	Team & player stats processors, Feature engineering framework	Week 3
3. Initial Modelling	Basic prediction capability	Team win probability model, model evaluation framework	Week 5
4. Enhanced Modelling	Complete prediction capability	Score prediction model, player performance models, model integration	Week 8
5. Refinement	Production-ready system	Data refresh programmed, daily prediction generator, performance optimization	Week 10
6. Validation & Documentation	Project completion	Final validation report, complete documentation, project presentation	Week 12

NBA Prediction System

Executive Summary

Resources

Technical resources:

- Development environment: Python 3.9+, Visual Studio Code
- Version Control: Git, GitHub
- Core libraries: pandas, numpy, scikit-learn, matplotlib, nba_api
- Compute requirements: 16GB RAM recommended for data processing
- Storage: 5GB for historical data and model artifacts

Skills requirements:

- Python programming
- Data manipulation with pandas
- Statistic modelling and machine learning concepts
- Time series analysis
- Sports analytics domain knowledge

Methodology

1. **Data Collection Strategy:**
   - Use NBA API for official statistics
   - Implement incremental data updates
   - Establish data validations checks
2. **Model Development Method:**
   - Start with simple baseline models
   - Evaluate multiple algorithm approaches
   - Utilise cross validation for reliable performance estimation
   - Implement feature importance analysis
   - Iteratively improve prediction accuracy
3. **Technical implementation:**
   - Modular code structure with clear seperation of concerns
   - Test-driven development for core functionality
   - continuous integration via GitHub actions
   - Comprehensive logging and error handling

NBA Prediction System

Executive Summary

Risk Assessment

Risk	Impact	Probability	Mitigation Strategy
NBA API limitations or changes	High	Medium	Implement backup data sources, create API response caching
Insufficient prediction accuracy	High	Medium	Start with simpler subproblems, establish baseline metrics early
Data Quality Issues	Medium	High	Implement thorough data validations checks
Scope creep	Medium	Medium	Maintain clear project boundaries, implement iterative deliverables
Computing Resource Limitations	Low	Low	Optimize high-computation processes, consider cloud resources

Quality Assurance Plan

Testing Strategy

- Unit Tests for all data processing functions
- Integration tests for completed pipeline
- Backtesting framework for model evaluation
- Model validation against historical seasons

Documentation Requirements

- API documentation for all models & functions
- Data dictionary for all collected & generated features
- Model documentation including training methodology & evaluation
- User guide for running prediction generation
- Development journal capturing decisions & learning

NBA Prediction System

Executive Summary

Deliverables

Final Project Artifacts:

1. Source code repository with complete implementation
2. Comprehensive documentation set
3. Trained prediction models with validation results
4. Sample prediction outputs
5. Project presentation materials

Documentation Set:

1. Project README and setup instructions
2. Technical design documentation
3. Data dictionary and feature explanations
4. Model performance analysis
5. User guide for system operations

Communication Plan

Stakeholder	Information Needs	Frequency	Method
Project sponsor	Progress Updates, key decisions	Bi-weekly	Status Report
Technical team	Technical decisions, implementation details	Weekly	Code reviews, technical notes
End Users	System capabilities, interpretation guide	Key-milestones	User documentation
Portfolio reviewers	Project overview, technical approach, results	Final	GitHub README, presentation