Player Prop Machine Learning Analytics Platform

Project Proposal/Feasibility Report

Seid Cubro

2/4/2026

1. **Summary**

This project proposes the design, development and deployment of a cloud-native sports analytics platform focused initially on NFL player prop evaluation. The system will ingest comprehensive player, team, game and contextual data and store it in a scalable and auditable database architecture. The system will engineer features and train custom machine learning models to generate probabilistic projections for common NFL prop markets.

The platform will expose analytics through a public web dashboard and REST API, emphasizing transparency, explainability and reproducibility. The system is designed to scale horizontally to additional sports leagues such as the English Premier League, NBA, UFC, UEFA Champions League and NHL by reusing shared ingestion, modeling and serving infrastructure.

The project demonstrates full-stack software engineering, data engineering, MLOps, cloud infrastructure and production readiness.

2. **Problem Statement**

Sports prop markets are driven by complex interactions between player usage, opponent tendencies, team context and game environment. While raw statistics are widely available, they are fragmented and lack historical context. The statistics are rarely presented with probabilistic interpretation or transparent assumptions.

Existing tools often focus on recommendations without exposing methodology, making them unsuitable for academic, engineering or analytical evaluation.

This project addresses that gap by building a system that:

- Aggregates and normalizes detailed NFL data at scale
- Applies statistical and ML-based modeling to estimate realistic player outcomes
- Presents results as probabilistic distributions rather than deterministic claims
- Emphasizes explainability, auditability, and system design over prediction hype

3. **Project Objectives**

Primary Objectives:

- Design a scalable data ingestion and storage architecture capable of supporting all NFL players and games
- Engineer a feature pipeline that captures player usage, matchup context and game environment
- Train and deploy a custom ML model for probabilistic player prop projection
- Serve projections through a low-latency API and interactive web UI
- Demonstrate production-grade practices: CI/CD, observability, security and cost control

Secondary Objectives:

- Support historical backtesting and model evaluation
- Enable future expansion to additional leagues with minimal architectural change
- Produce professional documentation suitable for technical review and portfolio use

4. **Scope Definition**

In-Scope (V1):

- Sport: NFL only
- Initial Prop Markets:
    - Receiving yards
    - Receptions
    - Rushing attempts
    - Rushing yards
    - Passing yards
    - Passing touchdowns
    - Anytime touchdown scorer
- System Capabilities
    - Historical data backfill (multi-season)
    - Incremental daily ingestion
    - Feature engineering pipeline

- ML model training, versioning and inference
- Public dashboard and REST API
- Kubernetes-based deployment
- Observability, logging and metrics

Out of Scope (V1):

- Paid features and monetization
- Automated betting or wagering
- Guaranteed outcomes or recommendations
- Real-time in-game predictions
- User-specific betting behavior modeling

5. **Functional Requirements**

FR-1: Ingest and persist complete NFL player, team and game datasets
FR-2: Ingest and persist prop market and odds snapshot data (or simulated feed for V1)
FR-3: Maintain historical records with timestamped versioning
FR-4: Engineer model-ready features from raw and contextual data
FR-5: Train a custom ML model to estimate player stat distributions
FR-6: Generate probabilities for over/under outcomes relative to prop lines
FR-7: Version, store and manage trained models
FR-8: Expose projections, metadata and confidence intervals via REST API
FR-9: Provide an interactive web UI for analysis and visualization
FR-10: Support administrative operations (retraining, backfill, health checks)

6. **Non-Functional Requirements**

NFR-1: API p95 latency < 300 ms for common queries
NFR-2: ML inference p95 latency < 200 ms
NFR-3: Idempotent ingestion
NFR-4: Retriable jobs with failure alerts
NFR-5: Horizontal scaling via containers

NFR-6: League-agnostic data model

NFR-7: HTTPS

NFR-8: Secrets management

NFR-9: Least-privilege access

NFR-10: Rate limiting on public endpoints

NFR-11: Centralized logging

NFR-12: Metrics dashboards

NFR-13: Ingestion and inference health monitoring

NFR-14: Cloud budget enforcement

NFR-15: Lifecycle policies for raw data

NFR-16: Autoscaling with upper bounds

NFR-17: Versioned data, features, and models

NFR-18: Deterministic training pipelines

7. **System Architecture Overview**

High-Level Components:

- Data Ingestion Service
    - Scheduled jobs pull raw NFL data and odds feeds
    - Raw payloads stored in object storage
    - Normalized records written to relational database
- Data Warehouse Layers
    - Raw Layer (immutable)
    - Clean Layer (validated, normalized)
    - Feature Layer (model-ready)
    - Serving Layer (UI/API optimized)
- Feature Engineering Pipeline
    - Batch jobs compute rolling stats, usage metrics, matchup indicators
    - Outputs persisted with version metadata
- ML Training Service
    - Offline batch training
    - Baseline vs ML model comparison
    - Evaluation metrics logged and stored

- Model Registry
    - Stores model artifacts, hyperparameters, metrics, training data hash
- Inference Service
    - Loads active model
    - Generates projections on demand or batch
    - Writes results to serving tables
- API Service
    - Public read-only endpoints
    - Admin endpoints for operations
- Web UI
    - Player pages
    - Prop analysis dashboards
    - Distribution visualizations
    - Model explainability views

8. **Data & Feature Strategy**

Core Data Categories:

- Player Usage
    - Snap percentage
    - Targets, carries, routes run
    - Red zone usage
    - Goal-line attempts
- Team Context
    - Run/pass tendencies
    - Team efficiency metrics
    - Offensive pace
- Opponent matchup
    - Defensive performance by position
    - Pressure and coverage tendencies
    - Allowed yardage distributions

- Game Environment
    - Home/away
    - Weather
    - Stadium type
    - Rest days
    - Travel distance
- Historical performance
    - Rolling averages
    - Variance and consistency metrics
    - Usage trends

9. **Machine Learning Approach**

- Model Type (V1)
    - Gradient Boosted Regression (primary)
    - Rolling average baseline (control)
- Outputs
    - Expected stat value (mean)
    - Variance/dispersion
    - Full probability distribution
    - Over/under probabilities
- Evaluation Metrics
    - RMSE/MAE
    - Calibration error
    - Baseline comparison
    - Stability across retains
- MLOps Practices
    - Reproducible training
    - Model versioning
    - Backtesting on historical weeks
    - Model cards documenting purpose and limitations

10. **User Interface & Analytics**

Core UI Views:

- Player profile
- Market-specific analysis
- Odds and line movement history
- Matchup breakdown
- Distribution and confidence intervals
- Feature importance/explainability panel

UX Principles:

- Transparency over persuasion
- Clear uncertainty visualization
- No “pick” language

11. **Deployment & Operations**

- Containerized services (Docker)
- Kubernetes deployment
- CI/CD pipeline
- Infrastructure as Code
- Monitoring dashboards
- Operational readiness checklist

12. **Feasibility Analysis**

This project is highly feasible technically speaking. All components rely on established technologies and standard architecture. Complexity is mitigated by limiting V1 to a single league, and a small set of prop markets.

The scheduled scope for project completion is feasible within a semester-length timeline. By prioritizing core ingestion and modeling, deferring advanced markets and leagues and reusing shared infrastructure across components, a three month time span is realistic for completion.

All resources for the project are open-source and free to use. Cloud free tiers and student credits will be used. The project is single-developer viable with disciplined scope control.

Risk Assessment:

Risk	Mitigation
Data source limitations	Simulated feeds + modular ingestion
Scope creep	Locked V1 scope
Model overfitting	Baseline comparison + backtesting
Performance bottlenecks	Caching + async processing

13. **Success Criteria**

- End-to-end pipeline runs without manual intervention
- ML model outperforms baseline
- Deployed public demo with live data
- Clear documentation and reproducibility
- Positive technical evaluation by reviewers

14. **Future Expansion**

- Additional NFL prop markets
- English Premier League ingestion
- NBA and UFC modeling
- Injury NLP ingestion
- User accounts and alerts
- Advanced Bayesian models

15. **Conclusion**

The scheduled scope for project completion is feasible within a semester-length timeline. By prioritizing core ingestion and modeling, deferring advanced markets and leagues and reusing shared infrastructure across components, a three month time span is realistic for completion.