NFL Fantasy Prediction Engine

Machine Learning Capability Status Report

Date: December 7, 2024

Version: 3.0.0 (Phase 3 Complete)

Classification: Technical Status Report

Cover Letter

Dear Stakeholder,

I am pleased to present this comprehensive status report for the NFL Fantasy Prediction Engine project. This document provides a PhD-level technical overview of the system architecture, data infrastructure, and machine learning readiness.

Executive Summary:

The NFL Fantasy Prediction Engine has successfully completed its foundational phases (1-3), establishing a robust data pipeline and feature engineering system capable of supporting advanced machine learning models. The system currently processes data for 5 NFL seasons (2021-2025) with 29,204 player-game feature records.

Key Achievements:

- Complete data ingestion pipeline from multiple NFL data sources
- 207-column feature matrix with anti-leakage architecture
- Operational baseline predictor achieving 0.735 correlation on target predictions
- Cross-platform player ID mapping covering 5,956 players

Project Readiness:

Milestone	Status
Data Infrastructure	Complete
Feature Engineering	Complete
Baseline Predictor	Complete
ML Model Training	Ready to Implement
Production Ensemble	Planned

The system is now ML-ready - the foundational data and feature infrastructure is complete, and Phase 4 (machine learning model training) can commence immediately.

Respectfully submitted,
NFL Prediction Engine Development Team

1. Project Overview

1.1 Mission Statement

Build a production-grade machine learning system for NFL fantasy football predictions that generates accurate weekly player projections, handles uncertainty and provides confidence intervals, combines multiple model architectures via ensemble learning, and operates reliably throughout the NFL season.

1.2 Architecture Philosophy

The system employs a phased, modular architecture where each phase builds upon the previous:

```
Phase 1 (Data Ingestion) -> Phase 2 (Feature Engineering) -> Phase 3 (Baseline Predictor)
-> Phase 4 (ML Models) -> Phase 5 (Production Ensemble)
```

1.3 Technology Stack

Component	Technology
Language	Python 3.9+
Database	SQLite 3
Data Processing	Pandas, NumPy
ML Framework	Scikit-learn, XGBoost (Phase 4)
Data Sources	Sleeper API, nflverse/nflfastR, nfl_data_py

2. Phase 1: Data Ingestion

2.1 Purpose

Phase 1 establishes the foundational data infrastructure by ingesting raw NFL data from multiple sources into a unified SQLite database.

2.2 Key Ingestion Scripts

Script	Purpose	Records
ingest_games.py	NFL game schedules	1,331
ingest_players.py	Player biographical info	2,462
ingest_stats_offense.py	Offensive player stats	29,204
ingest_stats_defense.py	Team defense stats	1,384
populate_defender_stats.py	Individual defender stats	52,306
ingest_nflverse.py	Advanced weekly stats	62,117
backfill_sleeper_stats.py	Historical backfill (2021-22)	8,898

3. Phase 2: Feature Engineering

3.1 Anti-Leakage Design

All features are calculated using ONLY data available BEFORE the game being predicted. This prevents data leakage which would artificially inflate model performance during training but fail in production.

3.2 Feature Categories

Category	Columns	Coverage	Description
usage_*	16	74%	Volume metrics (targets, carries, snaps)
eff_*	48	29%	Efficiency (yards/carry, catch rate)
oppdef_*	51	58%	Opponent defense metrics
ngs_*	42	9%	Next Gen Stats tracking data
sched_*	6	100%	Schedule context (rest, home/away)
ctx_*	6	0%	Vegas context (spread, total)
weather_*	13	0%	Weather conditions
label_*	14	100%	Target variables (predictions)

3.3 Rolling Window Architecture

Window	Description	Use Case
last1	Previous game only	Capture hot streaks
last3	3-game rolling average	Recent form
season_to_date	All games this season	Stable baseline

4. Phase 3: Baseline Predictor

4.1 Key Distinction: NOT Machine Learning

The baseline predictor uses FIXED mathematical formulas, not learned parameters. Weights are predetermined constants (60% long-term, 25% short-term, 15% opponent adjustment), making predictions deterministic and interpretable. This serves as a floor that ML models must exceed and a fallback when ML confidence is low.

4.2 Performance Metrics (2021-2024 Backtest)

Metric	Targets	Receptions	Rec Yards	Fantasy Pts (PPR)
MAE	1.13	0.89	9.45	4.89
RMSE	2.31	1.78	18.2	8.12
Correlation	0.735	0.712	0.654	0.503
Bias	+0.12	+0.08	-1.2	+0.34

5. Database Architecture

5.1 Database Statistics

Metric	Value
Database File	phase_1/database/nfl_data.db
Total Tables	37
Total Records	300,000+
Database Size	~150 MB
Seasons Covered	2021-2025

5.2 Key Populated Tables

Table	Rows	Description
games	1,331	NFL game schedules and results
players	2,462	Player biographical information
player_game_stats	29,204	Per-game player statistics
player_game_features	29,204	Engineered feature matrix
baseline_predictions	13,520	Phase 3 predictions
defender_game_stats	52,306	Individual defender stats
nflverse_weekly_stats	62,117	Advanced weekly statistics
player_id_mapping	5,956	Cross-platform ID mapping
snap_counts	71,187	Player snap count data
betting_lines	16,460	Vegas betting lines

6. Data Source Analysis

6.1 Primary Data Sources

Sleeper API (api.sleeper.app)

- Player metadata, weekly statistics, real-time roster updates
- Coverage: 2021-present
- Strengths: Real-time updates, fantasy-focused, clean API

nflverse / nflfastR (github.com/nflverse)

- Play-by-play data, EPA, WPA, advanced receiving/rushing metrics
- Coverage: 1999-present
- Strengths: Most comprehensive NFL data source, academic-quality metrics

NFL Next Gen Stats

- Tracking data: separation, cushion, time to throw, completion probability
- Coverage: 2016-present (limited)
- Note: Only ~9% coverage in our feature matrix

7. Empty Tables and Missing Data

7.1 Tables Awaiting Data

Table	Intended Purpose	Priority
game_injuries	Injury reports	High
predictions	ML model predictions	High (Phase 4)
coverage_events	Coverage play tracking	Medium
team_tendencies	Play tendency data	Medium
play_by_play	Full PBP data	Low (too large)
redzone_stats	Red zone analytics	Medium

7.2 Features with 0% Coverage

Vegas Context (ctx_*): Betting data exists in betting_lines table (16,460 rows) but not yet linked to features. Implementation in progress.

Weather (weather_*): Schema exists with 13 columns, only 15 rows populated. API integration needed. Open-Meteo or Visual Crossing recommended.

8. Future Development Plan

8.1 Phase 4: Machine Learning Models

Phase 4 will introduce actual ML training with gradient boosting and neural network models:

- XGBoost - Primary model, excellent for tabular data
- LightGBM - Fast training, good for iteration
- CatBoost - Handles categoricals natively
- Neural Network - Capture non-linear interactions (optional)

Training Strategy:

Dataset	Seasons	Samples
Training Set	2021-2022	8,898
Validation Set	2023	6,104
Test Set	2024	7,416

8.2 Data Improvements Needed

Improvement	Priority	Effort	Impact
Vegas context linking	High	Low	+5-10% accuracy
Injury data	High	High	+5-15% accuracy
Weather API integration	Medium	Medium	+2-5% accuracy
Real-time updates	Low	High	UX improvement

Document Control

Version	Date	Author	Changes
1.0	2024-12-07	Development Team	Initial release