Drexel University

College of Computing & Informatics

Data Science Capstone Project

Final Report

Draft Smarter, Win Bigger: A Recommender System for Fantasy Football Drafts

Date: June 5, 2025

The General Managers - Group 7:

Caleb Miller - cm3962@drexel.edu

David Blankenship - dwb65@drexel.edu

Hashim Afzal - ha695@drexel.edu

Thomas Kiefer - tmk326@drexel.edu

Abstract

This project presents the **Fantasy Football Draft Assistant**, a comprehensive decision-support tool designed to help users make smarter, real-time draft picks in competitive fantasy football leagues. By combining seven seasons of historical NFL data with machine learning predictions and strategy-aware rule logic, our system generates context-sensitive player recommendations tailored to team needs, scoring format, and draft position. We developed and compared multiple regression models including XGBoost, MLP, and a stacked ensemble to forecast future fantasy performance under both PPR and Standard scoring systems. The best-performing model for each scoring type was integrated into a custom-built dashboard that simulates a live draft experience, complete with dynamic team tracking, positional constraints, and serpentine logic. Our assistant balances statistical projections with domain heuristics to produce draft suggestions that are both analytically grounded and strategically sound, empowering users to build more competitive, balanced rosters.

---

1 Introduction

Fantasy football has exploded in popularity over the last 25 years, with over 30 million people participating annually in the United States alone [1]. It consistently generates more than $10 billion each season and has become a major part of the larger sports gambling industry, which saw over $120 billion in wagers in 2024. Sports gambling is now legal in 39 states, and as the space has grown more lucrative, so has the level of competition [2]. A diehard community of amateur managers has emerged, with everyone looking for new ways to gain an edge through deeper insights, strategy, and data-driven decision-making.

At the competitive level, fantasy football is much more than a casual hobby. It involves real-time risk management, opponent strategy, and long-term team construction. Managers spend hours researching projections, analyzing matchups, and adjusting lineups based on the latest performance data. The annual draft is often the most crucial part of the season, and making smarter picks early is often the difference between a winning and losing record.

The goal of our project, the Fantasy Football Draft Assistant, is to support users during their draft by providing real-time player recommendations. This system is designed to help users make informed decisions by combining historical performance data, fantasy scoring formats, and modeling techniques that predict future outcomes. Unlike static rankings or one-size-fits-all draft guides, this assistant adjusts to user inputs such as scoring format, team composition, and live draft progress.

We built this system using a large dataset of NFL player statistics collected over 7 full seasons. After preprocessing our data and engineering features, we trained several models to predict fantasy performance in both standard and PPR scoring formats. These outputs are integrated into an interactive dashboard that allows users to make draft selections with more confidence and flexibility. Our focus throughout this project has been on creating a tool that is both accurate and usable, with the goal of helping fantasy football players draft stronger and more balanced teams to gain a competitive edge.

2 Methods

2.1 Data Acquisition & Preprocessing

To create a reliable draft assistant, we first had to gather and prepare a detailed dataset of NFL player performance and fantasy outcomes from 2018 to 2024. This involved scraping data from multiple public sources using custom python scripts

before merging and transforming it into a format suitable for modeling.

Data Sources and Scope

We pulled data from the following sources:

- **Pro Football Reference and NFL.com**: season-level player statistics across offensive and defensive positions
- **FantasyPros**: average draft position (ADP) data from mock and live drafts
- **ESPN.com**: fantasy point totals based on both standard and PPR scoring formats
- **StatMuse**: special teams and additional statistics used for feature enhancement

These sources provided a mix of raw NFL performance data and fantasy-specific metrics. Our dataset includes all NFL players who recorded statistics from 2018 to 2024, even if they played limited or no snaps. This was important for capturing variability and involvement during model training.

Scraping and Primary Cleaning

We used Python scripts to scrape data from each source. After collecting the data, we had to resolve issues such as inconsistent player name formatting, team abbreviation mismatches, and positional labeling differences. To merge everything appropriately, we standardized player names, aligned team codes, and created a consistent player-season identifier.

The data was originally merged by season and converted to a wide format, with each player occupying one row and each season in separate columns. For modeling purposes, we changed our approach to reshaped the dataset into a long format, where each row represented a single player-season combination. This allowed us to train models using each season as a separate observation. By structuring the data in a long format, we were able to capture season-level variation in player performance and draft context, which is crucial for modeling trends, changes in values, and potential breakout patterns across different years. We also addressed players who played multiple positions by assigning them based on their primary fantasy role, as determined by total touches or points.

Handling Missing Values and Outliers

Some players had missing values due to injuries, bye weeks, or limited playing time. While some may say these nulls or zeros should be removed, it is vital for this project to include them. Similarly, we retained outliers in the dataset, as they reflect real and impactful player performances. These situations can represent breakout performances, seasonal busts, and the overall involvement of players that are important to be captured by our recommendation system. Players that are outliers based on their statistically significant performance are the exact kind of players we wanted the Fantasy Football Draft Assistant to identify and prioritize.

2.2 Exploratory Data Analysis

We explored our data in order to develop an understanding of the dataset, understand the necessary techniques for preprocessing, and catch potential issues with our data. Our dataset consists of 45 features, 2 targets (PPR and Standard Fantasy points), and 1 metadata column (Player Name). Since we have already documented our EDA extensively in a 40-page report and shared the full notebooks in our GitHub repository, this paper focuses on highlighting only the analyses most relevant to the decisions and directions we pursued. Our EDA report is cited in our appendix.

First we look at the histogram of positions in our dataset in figure 1. We see that there is a class imbalance in positions with Wide Receiver and Running Backs most represented and Defensive teams the least represented. Initially we had considered handling this imbalance through the development of separate models for different positions with the theoretical reasoning that different positions use different features. However

we ultimately found that a single model with all positions had the highest performance.

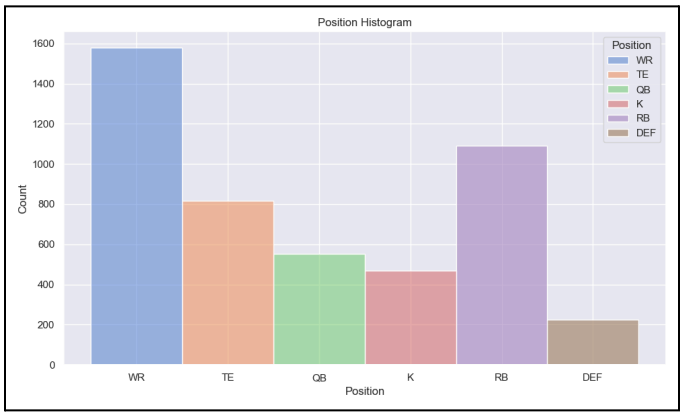


Figure 1: Distribution of player positions

A common distribution in our dataset was that features would display a power law curve with a tall head and a long tail. This typically displays itself as a right skew where most players have very small results while a select few are obtaining very high values in the feature. We show figure 2 displaying the rushing touchdowns of our players as a typical example of this trend. This led us to consider using transformations in our dataset to see if we could get our data in a more normal distribution. We attempted to use log, sqrt, Box-Cox, and Yeo-Johnson transformations but ultimately found they didn't provide significant improvement. As such we elected to go without the transformations.

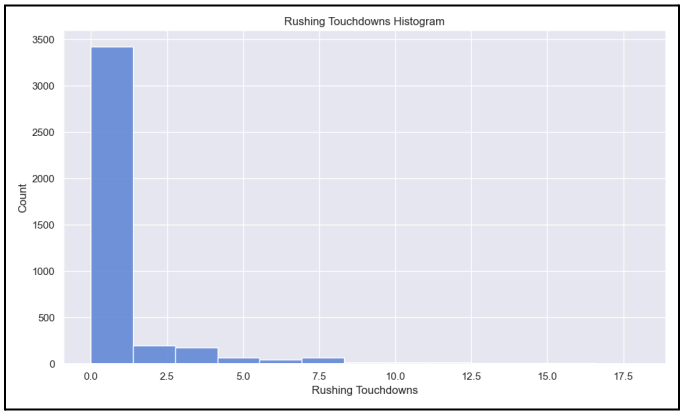
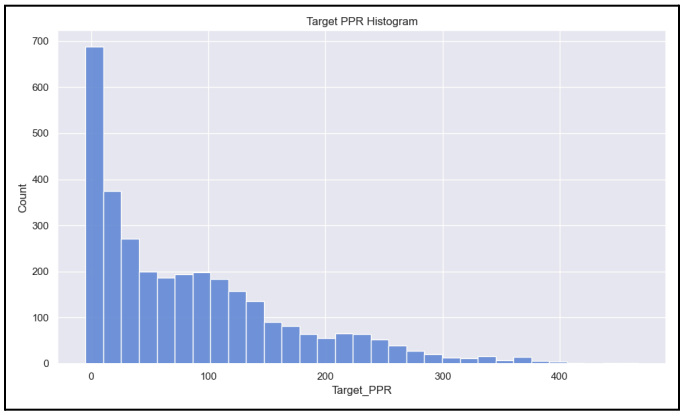
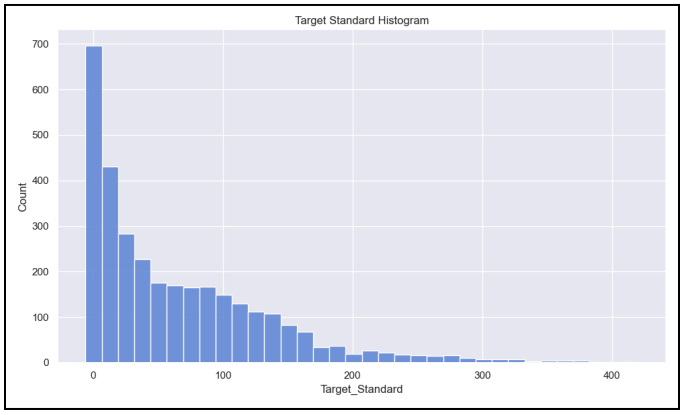


Figure 2: Rushing touchdowns for all players

The long tail distribution is very relevant for our understanding of outliers. In many ML problems developing a strategy to handle outliers is of critical importance. However we have a very unique scenario in our problem in that we are actually looking for the outliers to select them. In our context outliers are the high performing players that consistently achieve the top results for their position. We can see that in figures 3 and 4. For both of our targets we see a right skew distribution where the majority of players have very low fantasy scores while a select few have the much higher scores. The latter are the outliers that we're looking to provide as recommendations.





Figures 3 and 4: PPR and Standard Fantasy points

Thanks to our use of "golden standard" websites in football data that we scrapped from we were able to create a high quality dataset with no traditional missing values. This is due to the completeness of the sources that we used. Instead what we found was that we had many null values that correspond to features irrelevant to the players position. For example, a wide receiver will not have kicking stats.

2.3 Feature Engineering & Selection

Effective fantasy football prediction requires transforming raw game statistics into informative, predictive features that generalize across seasons. To achieve this, we implemented several layers of feature engineering, emphasizing fantasy-specific insights, trend analysis, and multicollinearity reduction.

Feature Construction

We engineered a rich set of features derived from both raw performance statistics and fantasy-relevant context:

- **Per-play and per-opportunity metrics** such as Yards per Completion, Catch Percentage, and Yards per Touchdown were created to standardize production across player roles and usage levels.
- **Rolling trend features** were introduced to capture momentum and consistency over time. For instance, we calculated Rolling 3-Year PPR Points to smooth out year-to-year volatility.
- **Year-over-year delta features** captured the directional changes in key performance indicators—such as touchdowns, yardage totals, or usage rates—between seasons. These were particularly useful for signaling potential breakouts or regressions based on a player's recent trajectory.
- **Fantasy-specific attributes** included Positional Average Draft Position (ADP), which served as a proxy for preseason expectations and relative player value within each position group.
- Additional **contextual features** like Position, Team, Age, and Years of Experience were retained and appropriately encoded to provide interpretive and categorical grounding.

Target Creation

We defined our prediction targets as each player's fantasy points in the following season, calculated separately for PPR and Standard formats. All input features for a given target season were limited strictly to data from prior seasons to ensure temporal integrity and prevent leakage.

Players from the 2024 season were intentionally left without targets, as their outcomes (i.e., 2025 fantasy performance) have not yet occurred. This mirrors our real-world use case: generating predictions for upcoming drafts where true labels are unknown. As a result, 2024 players were used solely for inference, not for model training or validation.

Multicollinearity Handling

To reduce feature redundancy and improve model interpretability, we implemented a multicollinearity filtering step:

- **Correlation matrices** identified strongly related feature pairs, particularly among yardage and scoring efficiency stats.
- **Variance Inflation Factor (VIF) analysis** further quantified feature redundancy.
- Based on these diagnostics, we dropped or consolidated overlapping features (e.g., choosing between total yards vs. yards per game) to streamline the feature space.

Feature Filtering

Our final feature set was curated using a blend of:

- **Domain expertise**, selecting predictors known to influence fantasy production (e.g., pass-catching volume for RBs).
- **Predictive utility**, based on feature importance assessments during model training.
- **Data availability and completeness**, ensuring consistency across players and seasons.

Linear	XGBoost	LightGBM	MLP	DNN
fit_intercept	n_estimators	n_estimators	activation	activation
positive	learning_rate	learning_rate	alpha	alpha
	max_depth	max_depth	batch_size	batch_size
	subsample	subsample	early_stopping	early_stopping
	reg_alpha	reg_alpha	hidden_layer_sizes	hidden_layer_sizes
	reg_lambda	reg_lambda	learning_rate_init	learning_rate_init
	colsample_bytree	min_child_samples	max_iter	max_iter
	min_child_weight	num_leaves	n_iter_no_change	n_iter_no_change
	gamma	colsample_bytree	validation_fraction	validation_fraction
			solver	learning_rate

Table 1: Hyperparameters by model

We explicitly removed any leakage-prone features - such as actual fantasy points scored in the target season - to preserve the generalizability and realism of model predictions.

2.4 Machine Learning Models

We developed several predictive regression models using a variety of ML methods to predict future fantasy football performance with PPR and Standard scoring as the model target. The idea behind developing these models was to obtain accurate player projections to drive recommendations in the Draft Assistant. The methods used to develop these models can be broken into three parts; the shared modeling pipeline, the 6 ML techniques we used, and hyperparameter tuning.

For the modeling pipeline we created a modular and reusable pipeline for ease of use for all models we developed. For null values we inserted zeroes as they represented actual zeroes, not missing values in our dataset. We handled categorical columns using a one-hot encoder. We also scale our numerical data with a Standard scaler.

We used what we call a train-test-”predict” split. For our train and test split we use the previous years data to predict the fantasy score for the next season. For example if we have 2018 data we predict the fantasy score performance for 2019. We also use an 80:20 split for this. Our “predict” split is for predicting the upcoming season’s performance and includes the 2024 performance data to predict the upcoming fantasy scores for the 2025 season.

We elected to explore 6 different regression model types: Linear, XGBoost, LightGBM, a multilayer perceptron (MLP), a Deep Neural Network (DNN), and a stacked model combining MLP and XGBoost. We elected to use a linear model to provide a simple baseline comparison for our more complex models. XGBoost is a high-performance tree-based model that uses gradient boosting to capture

nonlinear relationships. LightGBM is also a tree-based model that emphasizes speed and performance with high out-of-the-box performance. The MLP model is a basic neural network that uses a tanh activation to capture nonlinearity. The Keras-based DNN model includes multiple hidden layers to capture potential higher complexity relationships. Finally, the stacked model combines the MLP and XGBoost methods, potentially capturing higher complexity through the use of multiple models.

To obtain the highest possible performance we also elected to perform hyperparameter tuning. We used scikit-learn's GridSearchCV function which combines cross validation with hyperparameter tuning. We used 5-fold validation and tuned hyperparameters relevant to the specific model used. We list the relevant hyperparameters in table 1 above. Note that we simply used the highest performing parameters for each component of the stacked model. To score the highest performing parameters we used R^2.

2.5 Recommender System

Our recommender system is designed to simulate realistic, strategy-aligned fantasy football drafts by combining machine learning predictions with dynamic, rule-based logic. The system evaluates a user's current draft state, predicts future player performance, applies positional value adjustments, and ranks eligible players based on both performance and strategic fit.

Objective

The objective of the recommender system is to provide real-time, context-aware draft suggestions that reflect both data-driven performance projections and the nuanced strategy of successful fantasy managers. Rather than naively recommending the player with the highest projected score, the system takes into account team needs, draft round, positional scarcity, and widely accepted heuristics like delaying kicker/defense picks or avoiding early quarterbacks unless they are elite.

The goal is to guide users through their draft in a way that balances raw predictive value with strategic realism, helping them construct a competitive and well-rounded team. The system is intended to be used iteratively during live drafts, updating recommendations each round based on prior picks and evolving team composition.

System Inputs

The system takes the following inputs:

- **2024 player dataset** with contextual features
- **Trained model pipeline** for predicting fantasy points (e.g., a stacked regressor)
- **Team draft state**, including filled positions, drafted players, and current round
- **Customizable roster configuration**, indicating positional limits
- **Scoring format**, either PPR or Standard

Recommendation Flow

Step 1: Predict Fantasy Points

The recommender first applies the trained model pipeline to the eligible player pool (2024 season only). This pipeline includes all necessary preprocessing steps and outputs a Predicted_Points value for each player based on their input features. These values serve as the foundation for subsequent ranking and filtering.

Step 2: Calculate Value Over Replacement (VOR)

To account for positional scarcity and opportunity cost, we calculate each player's Value Over Replacement (VOR):

- **Replacement levels** are dynamically defined per position based on the expected number of drafted players in a 12-team league (e.g., RB24, WR24, QB16).
- **VOR** is calculated as the predicted points minus the position's replacement-level point threshold.

- **Position-specific multipliers** are applied (e.g., QB = 1.2, K = 0.3) to adjust for positional impact.
- A bonus is added for elite quarterbacks scoring above a set threshold.
- Both VOR and Predicted Points are normalized to a 0–1 scale.
- A **composite score** (Overall_Score) is then calculated using a tunable weight:

$$\text{Overall\_Score} = \text{Normalized\_VOR} \times \text{vor\_weight} + \text{Normalized\_Points} \times (1 - \text{vor\_weight})$$

Step 3: Apply Draft Strategy Rules

The system enforces realistic draft logic through a set of configurable rules:

- **Avoid early QB/TE picks unless elite**, based on top 5% predicted points at their position
- **Delay kicker and defense picks** until the final 2–3 rounds
- **Respect roster limits**, preventing overfilling any position
- **Prioritize unmet positional needs** based on current team composition
- **Defer backup QBs** until the final rounds if a starter has already been selected
- **Force-fill K/DEF** in the final rounds if still empty

These rules are applied using the team's draft state and round context to mimic experienced drafter behavior.

Step 4: Filter and Rank Eligible Players

All players are passed through the rules filter, and the eligible subset is sorted by their overall score. The top N players are returned as draft recommendations, each annotated with their predicted performance and strategic value.

System Features

- **Live draft integration** via dynamic team state and round input
- **VOR-based scoring**, weighted to prioritize positional scarcity
- **Elite player detection** using quantiles from predicted point distributions
- **Flexible rule system** for tuning drafting behavior
- **Support for both PPR and Standard formats**
- **User-defined roster configurations**, supporting varied league structures

Summary

The recommender system strikes a deliberate balance between statistical accuracy and strategic realism. Rather than suggesting purely high-scoring players, it guides users through a draft that mirrors expert tactics, roster needs, and positional value tradeoffs. This hybrid design delivers smarter, more adaptable draft decisions that enhance competitiveness and usability across diverse fantasy league formats.

3 Dashboard App Overview

Our recommender system is deployed through a web-based dashboard built with Plotly Dash, designed to be both intuitive and powerful for fantasy football users. The application enables users to simulate an entire draft experience by integrating real-time recommendations, draft tracking, and customizable team configurations.

Goals

- **Usability**: Prioritize a clean, digestible presentation of complex data to accommodate users of all experience levels.
- **Speed**: Provide real-time recommendations after each pick, with minimal load time.
- **Customizability**: Allow users to configure scoring formats, roster constraints, and dynamically respond to draft progression.

Core Features

- **League Setup Panel**: Users select between PPR and Standard scoring formats and define their custom roster configuration (e.g., max 4 RBs, 2 TEs).
- **Live Draft Board**: Displays the current draft state, including selected players, round number, and team composition.
- **Dynamic Recommendation Engine**: Returns player suggestions in real time based on draft position, team state, and scoring format. Candidates are filtered by rule logic and ranked using a blended score (Predicted Points + Value Over Replacement).
- **Team Roster Display**: Tracks positions filled in real time to help users maintain a balanced roster and avoid exceeding positional limits.
- **Manual Override**: Allows users to manually draft any player, supporting mixed usage of AI recommendations and personal draft strategies.

Design Decisions

- **Pick-Based Draft Control:** Instead of asking users to track the round manually, the app prompts for the user's current overall pick number (e.g., Pick 5 in a 12-team league). The system then automatically calculates the round and turn using standard snake/serpentine logic. This maintains draft realism while remaining format-agnostic and lightweight.
- **Transparency Over Automation:** Recommendations include supporting metrics (Predicted Points, VOR, Overall Score), promoting trust and user agency in decision-making.
- **Simplicity and Portability:** The dashboard runs locally in the browser with no need for login credentials, server deployment, or internet connectivity during use.

Demo Access

App Walkthrough:
https://drive.google.com/file/d/1DamL9t8kiPj6vd7bk9KdQm18qez_1hz7/view?usp=sharing

Sample Draft Simulation:
https://drive.google.com/file/d/1CHSYgv2sO0_bDoalzzup14KPzZn5v-zL/view?usp=sharing

4 Results

To assess model performance, we used Root Mean Squared Error (RMSE) and R^2 as our primary evaluation metrics. RMSE provides an interpretable measure of average prediction error in the same units as the target variable, offering insight into how far off predictions tend to be from actual fantasy point outcomes. R^2 quantifies the proportion of variance in fantasy scores explained by the model, indicating overall goodness of fit. These metrics were chosen for their ability to capture both accuracy and explanatory power, making them well-suited for evaluating regression performance in the context of fantasy point prediction.

Model	R-Squared	RMSE
Stacked Model (XGBoost + MLP)	0.637	41.6 Points
XGBoost	0.632	41.9 Points
LightGBM	0.629	42.1 Points
Linear Regression	0.605	43.4 Points
Deep Neural Network	0.587	44.4 Points
Multi-Layer Perceptron	0.587	44.4 Points

Table 2: Performance comparison of standard scoring models using RMSE and R^2

Model	R-Squared	RMSE
LightGBM	0.619	51.0 Points
XGBoost	0.612	51.4 Points
Multi-Layer Perceptron	0.603	52.0 Points
Stacked Model (XGBoost + MLP)	0.600	52.2 Points
Linear Regression	0.583	53.3 Points
Deep Neural Network	0.581	53.4 Points

Table 3: Performance comparison of PPR scoring models using RMSE and R^2

We further visualized model accuracy through actual vs. predicted scatter plots, a sample of which can be seen in figures 5 and 6, enabling assessment of prediction quality and outlier behavior. All results were logged systematically to support reproducibility and facilitate across model iterations. Based on performance evaluation metrics, we selected a Stacked Regressor as the final model for standard scoring formats due to its superior accuracy and ability to capture variance among different players. For Points Per Reception (PPR) formats, we deployed a LightGBM model, which offered superior performance in capturing the impact of receptions on scoring.

Model outputs were utilized to generate player recommendations based on projected fantasy points, VOR and overall score as mentioned above. During testing, we observed that the Stacked model and LightGBM produced noticeably different rankings, particularly for positions heavily influenced by receptions (e.g., running backs, wide-receivers, and tight ends). These differences highlighted the need for separate models tailored to scoring formats, as a single model would risk underrepresenting positional value depending on the league type. By maintaining two different pipelines, the system ensured recommendations aligned with the scoring of each format, delivering context-aware draft guidance.

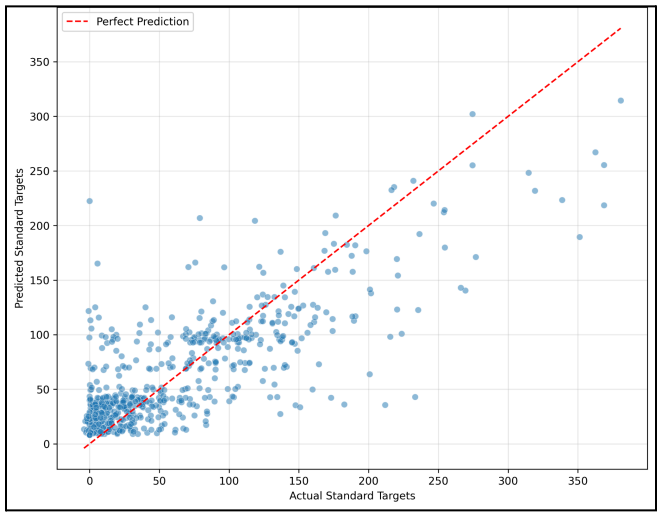


Figure 5: Predicted vs Actual Points comparison for Stacked Model in Standard Scoring

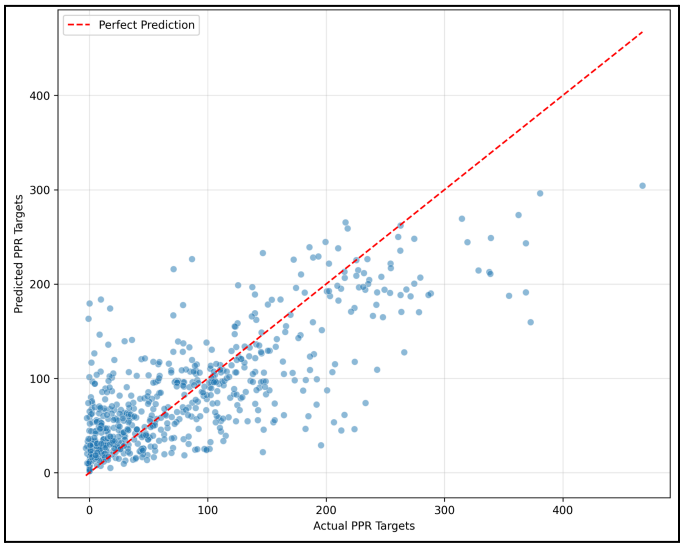


Figure 6: Predicted vs Actual Points comparison for LightGBM model in PPR scoring

To evaluate the quality of player recommendations beyond our metrics, we conducted simulation testing and mock draft exercises using outputs from both the Stacked model (standard scoring) and LightGBM (PPR) models. The goal was to assess whether the recommendations aligned with intuitive draft logic, such as positional scarcity and expected workload. Throughout testing, the models consistently produced draft strategies that "made sense" by prioritizing top-tier running backs and wide receivers early, and accounting for positional

depth in later rounds. These observations gave us confidence that the models and recommendation logic were not only statistically sound but also strategically aligned with how humans typically approach draft day decision making. To further validate performance, we compared the top-10 player projections from the PPR LightGBM model to ESPN's official 2025 fantasy football rankings. The model correctly identified 7 of the top 10 players in the draft, reinforcing its ability to capture the same high-value targets.

5 Conclusion & Future Work

This project aimed to build and deploy an intuitive Fantasy Football Draft Assistant, integrating predictive modeling for the 2025 NFL season with custom recommendation logic to support an interactive, user-facing dashboard. By leveraging ensemble methods such as the Stacked model and LightGBM, we generated context-aware player projections and tailored draft recommendations for both standard and PPR league formats. While our system produced strong results and demonstrated meaningful alignment with expert analyses, we recognize that several enhancements could further improve its predictive accuracy and practical utility.

One major limitation lies in the system's handling of rookie players. The absence of reliable data on incoming players, such as position drafted or detailed college performance statistics, limited the model's ability to forecast their fantasy value. Since rookies often have volatile but high-upside roles, incorporating some sort of pre-draft evaluation or scouting metrics could significantly enhance future predictions.

Another key limitation is the absence of injury data in the final model. While injury indicators were initially tested as features, their inclusion led to a drop in model performance, likely due to noise or inconsistencies in the data, and these were ultimately removed from the final model. Nonetheless, injuries are an integral part of football, and developing a method to encode injury data and severity could provide major upside in future iterations.

Additionally, the current models do not factor in bye weeks, which are a critical component of roster management and draft planning. Incorporating this aspect into the recommendation logic would allow users to better avoid lineup conflicts and strengthen weekly consistency in team construction.

Beyond improvements in model inputs and features, several engineering and user experience enhancements could also be planned for future iterations. One recommendation is to develop real-time data pipelines that allow the dashboard to update projections continuously based on live player stats, news, and depth chart changes. Hosting the dashboard on a public-facing website, would make it more accessible to users allowing for broader evaluation through public release. Looking forward, we believe that addressing these limitations could significantly raise the ceiling on performance as well as usability and user experience.

This work demonstrates how data science can meaningfully enhance the fantasy football experience by providing structure, insight, and strategic guidance. With continued investment in both technical and user-facing improvements, this assistant could become a key resource in the advancement of the fantasy football analytical products. By combining predictive modeling with recommendation logic and delivering it through an interactive dashboard, we created a system that not only mirrors expert insight but also provides a scalable framework for real-time draft support. As fantasy sports continue to evolve and grow in popularity, tools like this one can serve as a bridge between intuition and analytics, empowering players of all experience levels to make more informed, strategic decisions on draft day.

References

1. Oehy, A. (2024, August 30). *Fantasy football - NFL's fan engagement engine*. Medium. https://medium.com/the-ao/fantasy-football-nfls-fan-engagement-engine-c713699859e6

2. Hardy, K. (2024, December 5). *Growth of sports betting may be linked to financial woes, new studies find*. Stateline. https://stateline.org/2024/12/05/growth-of-sports-betting-may-be-linked-to-financial-woes-new-studies-find/

Data Sources

Average draft position (ADP). FantasyPros. (n.d.). https://www.fantasypros.com/nfl/adp/half-point-ppr-overall.php

ESPN Internet Ventures. (2005, June 19). *Rules - standard scoring system*. ESPN. https://www.espn.com/fantasy/football/ffl/story?page=fflrulesstandardscoring

NFL Enterprises LLC. (n.d.). *NFL Player Stats.* NFL.com. https://www.nfl.com/stats/player-stats/

NFL stats. StatMuse. (n.d.). https://www.statmuse.com/nfl

Sports Reference LLC. (n.d.). Pro Football Stats, history, scores, standings, playoffs, schedule & records. Pro Football Reference. https://www.pro-football-reference.com/

Tools and Packages Used

Category	Libraries/Tools
Data Manipulation	pandas, numpy, scipy, datetime, math, StringIO, defaultdict
Visualization	matplotlib, seaborn
Machine Learning	scikit-learn, xgboost, lightgbm, statsmodels, joblib
Web App Development	dash, dash_bootstrap_components, urlib.parse
Utilities	os, sys, time, logging, re, regex, base64, requests, BeautifulSoup4, utils

Appendix

Github Repository:

https://github.com/caleb10miller/FantasyFootballHelper

Reports & Presentations

Launch Report:
https://docs.google.com/document/d/1CQxeFVEh1mQrDLrERtKnSajse7rqxQco/edit?usp=sharing&ouid=111509805476584987479&rtpof=true&sd=true

Pitch Presentation:
https://drive.google.com/file/d/1E_uNNZJjbtu5UlaMU2E7Af9b6-4ReAhv/view?usp=sharing

Acquisition and Preprocessing Report:
https://docs.google.com/document/d/1TgcuJZKM9aFWG5al0km6b21ZIpxOtxQm/edit?usp=sharing&ouid=111509805476584987479&rtpof=true&sd=true

Exploratory Data Analysis Report:
https://docs.google.com/document/d/1wfEeCtZBkcaGsFDg3FPfBU1x9lZeYKLW/edit?usp=sharing&ouid=111509805476584987479&rtpof=true&sd=true

Status Presentation:
https://drive.google.com/file/d/1sydyFS7jj5hmBk8uRm4yC96Nk3kmWbUi/view?usp=sharing

Re-Launch Report (Status Update):
https://docs.google.com/document/d/17UAM0l6XQt

TCAvJdRqCO_9HCWWnL3-iA/edit?usp=sharing&ouid=111509805476584987479&rtpof=true&sd=true

Re-Launch Presentation (Status Update):
https://drive.google.com/file/d/10J3G2pH9ba27ePK6hLuzWdY5DMWxSGxC/view?usp=sharing
Re-Pitch Presentation (Status Update):
https://drive.google.com/file/d/1UggIRRsmcQKjLGuNMRlo-OhjuiEg6jPn/view?usp=sharing

Machine Learning Presentation:
https://drive.google.com/file/d/1dGfQNtN3hG8pgsHHRAvkaiDdq9QymP_E/view?usp=sharing