Towards Efficient Sports Player Prop Markets

Eric He Jonathan Lin

1 Introduction

The rapid growth of legalized sports betting across the United States has led to an explosion in popularity within different platforms such as DraftKings, Bovada, Kalshi, and FanDuel. This surge in recreational users, all with the intent of making "quick and easy" cash, leads to an oversaturated market that is not always driven by quantitative qualities. Instead, these betting lines are often influenced by public sentiment, media narratives, and the behavioral biases of recreational bettors. As a result, sportsbooks adjust their lines not solely to reflect true statistical expectations, but also with the intent to balance betting volume across both sides of a wager, in order to minimize their risk. This creates opportunities for mispricings in the market, where the posted odds deviate from the outcome's true likelihood. Such inefficiencies present an edge over the market for data-driven bettors, particularly quantitative models.

Before we dive in, let's define some of the necessary terms related to sports betting. The act of *sports betting* is placing a wager on the outcome of an event, where a bookmaker sets the corresponding odds to each event. The odds are usually either represented as American odds (ex. -120 means you bet \$120 to profit \$100 and +150 means you bet \$100 to profit \$150) or payout percentage (ex. 1.8x means profit of 80% the original bet). From the given odds, we can also calculate the *implied probability* of a bet, which represents the likelihood of an event as determined by the bookmaker. If the bettor guesses the correct outcome of the game, they win the profit amount, otherwise they lose their original amount bet. For our work, we will focus mainly on *player props*. In a player prop bet, a bettor must predict whether a certain player will go over or under a statistical line determined by the bookmaker. In most cases, the *prop line* is set with a half-point increment (ex. 4.5), ensuring that the outcome cannot result in a tie.

Needless to say, the goal of any bookmaker is to maximize profit, so the odds offered to bettors are not perfectly "fair". In practice, the implied probabilities derived from the posted odds sum to greater than 100%, a margin commonly known as the *vig* or *juice*. This built-in "fee" essentially means that the sportsbook almost always captures a profit over the long run. For example, a standard two-sided market may price each outcome at -110 odds, which corresponds to implied probabilities of roughly 52.4% per side, summing to $104.8\% > 100\%$. This means that if they receive an even amount of bets from both sides of m, then regardless of the outcome, the bookmaker will receive $\approx 4.8\%$ of the total betting volume of $2m$. Therefore, it is usually in the best interest of the bookmaker to dynamically adjust either the prop line or corresponding odds of an event, so the betting volume for the Over and Under is balanced. The excess percentage above 100% represents the bookmaker's edge, as with fair odds, any two-sided Over/Under player prop has a combined implied probability of 100% (either the player goes over or under the given prop line). For player props, the given odds are often identical at -110 or -120 at worst as bookmakers will generally shift the prop line itself instead. Thus, in order to have a profitable betting portfolio, we must predict Over/Under outcomes above the "break-even" threshold of -120 or 54.5%.

This paper presents an end-to-end sports betting pipeline designed to identify profitable opportunities within NBA and NFL player prop categories such as rebounds, passing yards, and more. The entire pipeline is fully automated and continuously maintained. It scrapes and calculates live data, updates and stores information in a centralized database, retrains models as new data becomes available, and generates real-time predictions.

The main contributions of this paper are as follows. First, it establishes a framework for identifying betting edges in player prop markets by formalizing how market mispricings can be detected and exploited. Second, it introduces a preprocessing pipeline capable of handling noisy and high-dimensional sports data, ensuring that features remain reliable across different contexts and seasons. Third, it out-lines general modeling strategies for predicting player props, withholding full architectural details to preserve the proprietary value of the approach. Finally, the paper proposes betting strategies designed to strike an optimal balance between expected return and variance. Unlike traditional work focused on moneyline or spread betting, this pipeline is centered around player props, which have only recently become very popular.

2 Edge

We first want to address the natural question: how can our sports betting pipeline be profitable? If prop lines were perfectly efficient, then no bettor could consistently outperform the lines set by bookmakers. In practice, however, inefficiencies in the prop line market represent a bettor's *edge*. More rigorously, edge is defined as the absolute difference between the true probability of an event and the implied probability of an event derived from the given odds. At the heart of this work is the assumption that we do not need to build an all-encompassing model that can directly outperform the bookmakers' models.

The bookmaker's task for creating player prop lines is essentially a regression problem, they assign a continuous statistical line (e.g., 7.5 rebounds, 245.5 passing yards) that reflects a player's expected performance. Our model's task, however, can be reframed as a much simpler classification problem of predicting a binary outcome of whether each player will go Over or Under the given line. From a machine learning perspective, this is often a more tractable problem, as binary classification generally requires fewer data features since it only needs to create an accurate decision boundary not model the whole output distribution.

Beyond statistical considerations, there are also psychological and behavioral factors at play. As mentioned in the introduction, bookmakers will often shift prop lines based on live betting patterns by the public. This allows them to hedge any risk they take on and guarantee profit through their *vig* fee. However, these adjustments may create market inefficiencies that deviate from the true expected probability of an event. Furthermore, public betting tendencies are often biased toward Overs and star player performances, which also skew prop lines away from their true expectation.

3 Problem Definition

Let's say each game $g \in \{1, \dots, n\}$ offers a set S_g of unique player prop bets. A player prop bet corresponds to a binary outcome: the chosen player either goes *Over* or *Under* the statistical line L set by the bookmaker. For example, a rebound prop line of $L = 7.5$ implies the two possible outcomes are "Over 7.5 rebounds" or "Under 7.5 rebounds."

Let $S = \bigcup_{g=1}^{n} S_g$ denote the collection of all available player prop outcomes across a betting round. The bookmaker assigns odds $o_j \in \mathbb{R}, o_j > 1$ in terms of payout percentage to each outcome $j \in S$. Let B denote our current bankroll. For each outcome $j \in S$, we allocate an amount $b_j \in [0, 1]$, where b_j represents the fraction of the bankroll wagered on that specific outcome. Since we can select only one direction (*Over* or *Under*) per player prop, at most one $b_j > 0$ for each prop. Moreover, we impose the budget constraint

$$\sum_{j \in S} b_j \leq 1,$$

so that the bettor may stake only a portion of the bankroll in each round. This formulation accommodates strategies such as the Kelly criterion, where the total wagered amount varies depending on the perceived edge and the bankroll state.

If outcome j occurs, the bettor receives a payoff of $o_j b_j B$; otherwise, the payoff is zero. Let p_j denote the true probability of outcome j. The profit from a single bet is therefore

$$P_j = \begin{cases} o_j b_j - b_j & \text{with probability } p_j, \\ -b_j & \text{with probability } 1 - p_j, \end{cases}$$

so that the expected profit is

$$\mathbb{E}[P_j] = (p_j o_j - 1) b_j.$$

The total profit across all bets in a round is

$$P = \sum_{j \in S} P_j,$$

with expected value

$$\mathbb{E}[P] = \sum_{j \in S} (p_j o_j - 1) b_j.$$

The central challenge is that the true probabilities $p_1, p_2, \ldots, p_j$ are unknown. Instead, we estimate these probabilities as $\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_j$ produced by our predictive model:

$$\hat{p} : D \to [0, 1],$$

where D denotes the input feature space containing historical game data, player statistics, team context, and other factors.

The problem therefore reduces to two main tasks:

1. Constructing accurate estimators $\hat{p}_i$ of the true outcome probabilities.
2. Designing a betting strategy that uses these estimates, along with bookmaker odds o_i, to allocate stakes b_i in a way that maximizes long-term profitability, i.e., by exploring the trade-off between expected value and risk.

4 Pipeline

In this section, we describe the end-to-end pipeline developed to identify profitable opportunities in NBA and NFL player prop betting. As mentioned previously, the pipeline is designed to be fully automated, continuously ingesting new data, updating models, and outputting actionable betting recommendations.

4.1 Data Collection and Database Construction

The first and arguably most important stage of the pipeline involves creating a structured database of historical and live player statistics, game results, and betting lines that is continuously being updated. Since no such free public API exists, we collect data from publicly available sportsbooks and sports data providers such as ESPN. Using Selenium, we can scrape raw data off different websites and store them together in our own database structure. Each raw statistic from an individual performance and team performance is stored under the season-week-team hierarchy. From past literature and our own intuition, current seasons are the most relevant time window for quantifying player/team performance. Thus, we compute seasonal aggregates and advanced statistics using the scraped data. Most seasonal aggregates are represented as averages across numbers of games, while some are represented as averages "per minute" (mainly from the NBA). The database is updated after every week for NFL games and after every day for NBA games. This design enables efficient querying of player and team level features, and ensures that new information can be seamlessly integrated in the future.

4.2 Preprocessing and Feature Engineering

Once raw data has been collected and stored into our database/API, the goal is to design quantitative data labels that can accurately capture a player's past performance and represent the underlying factors that influence their future outcomes. In order for our selected features to serve as reliable predictors, we should restrict our domain to focus on players with a sufficient amount of historical data. With some deeper analysis, we noticed players with small sample sizes, such as recent call-ups or bench players with limited minutes, tend to exhibit extremely high variance in their statistical outputs. To reduce noise that could undermine model stability, such players are excluded from the training set. Furthermore, we want to remove cases where player availability is uncertain, since the model is designed to predict performance conditional on a player receiving their normal role and playing time, without disruptions from injuries or irregular rotations. For example, a backup quarterback like Jake Browning might post unusually high passing yardage in a small stretch of games, because of contextual factors such as an injury to the starter or extended playing time during a blowout. These are qualitative events outside the scope of what historical statistical features can capture in our model that we would like to avoid in our training labels.

To pre-process out these small sample size players and outlier player performances, we devised a few successful strategies. The most obvious and effective solution was to filter out any player results where no historical market data o_j, betting line, could be found prior to the game for a specific statistic. If a player does not have an Over/Under line in a statistic before a game, it's likely either the player is injured or their historical sample size is too small in that field or the player does not receive reliable minutes. Another strategy we employed was to add a minutes played filter, where for each player label in our training data, they had to average above m minutes across a certain past game threshold. Lastly, one other possible outlier we wanted to account for were mid-game injuries, early benching, or extreme overtime results. Using a z-score threshold filter, we were able to filter out extreme minutes played changes from a player's historical game average to prevent these data outliers from being used in our training data.

With noisy and unreliable data filtered out, the next step is to identify which features are most informative for predicting player performance in specific statistical categories. Here, we investigate an important distinction between developing models to predict player prop bets compared to moneyline bets. In player prop bets, the bookmaker posts both the statistical line L and odds o_j for each bet. However, in moneyline betting, only o_j is given and directly incorporating bookmaker odds into a model's training input risks introducing a high degree of correlation with the bookmaker's own pricing, since the model could essentially be learning to replicate the bookmaker's probabilities. In contrast, for player props the situation is different as providing the statistical threshold L reduces the bettor's task to predicting whether the player's performance will fall *Over* or *Under* that line. Incorporating the prop line as a feature actually provides contextual framing of expected performance (e.g., naturally distinguishing player caliber between an 8.5 and a 14.5 rebound L). Thus, while the model should not rely completely on input L, using the bookmaker's statistical line in classification can enhance predictive accuracy without simply mirroring the sportsbook's implied probabilities.

For the remaining feature set, we adopted an iterative process of hypothesis and evaluation, beginning with intuitive performance indicators (e.g., average passing yards per game when predicting passing yard props) and progressively incorporating additional contextual variables. Our final feature set was selected based on its ability to deliver both the highest predictive accuracy and the most consistent performance across validation tests and modeling approaches. The specific feature composition is therefore tightly coupled with the type of predictive framework itself, which we will briefly detail in the subsequent section.

4.3 Predictive Modeling

With our curated dataset and possible feature sets in place, the next stage of the pipeline focuses on constructing predictive models for the binary *Over/Under* outcomes of player props. The objective of these models is to estimate probabilities $\hat{p}_i$ that each outcome occurs, providing a basis for evaluating the expected value of a prop line bet. In practice, we experimented with a wide range of modeling approaches, including linear regression variants, partial least squares regression (PLSR), neural networks, random forests, gradient boosting (XGBoost), and more. The goal is to use our pre-processed training labels to determine which combination of predictive feature labels and model will lead to the most accurate and consistent prediction accuracy of player prop results. We define predictive accuracy to be the model's estimated probability $\hat{p}_i$ converted into a binary decision (predicting *Over* if $\hat{p}_i \geq 0.5$ and *Under* otherwise) compared to the realized result of the prop bet.

To accomplish this, we implemented a rigorous validation framework to evaluate the effectiveness of each model-feature set combination. Since our dataset is inherently temporal, we cannot simply use conventional approaches such as k-fold validation, as randomly partitioning the data would allow the model to indirectly ”see the future” and thus cause data leakage. Instead, we will make train-test splits while slowly injecting data in chronological order and keeping a running average of predictive accuracy to maintain good validation measure. Instead, we adopt a rolling-origin evaluation procedure, where training and test sets are split in chronological order. At each step, new observations are incrementally introduced into the training set, while subsequent periods are used for testing. By maintaining a running average of predictive accuracy across these sequential splits, we obtain an accurate measure of out-of-sample performance.

For each type of statistic, we picked the combination of model type and feature set that resulted in a mix of high accuracy and consistency across predictive accuracies of $\hat{p}_i$ across all experiments. One consistent observation across these experiments was that models with excessive feature sets tended to overfit the training data, leading to inflated in-sample performance and poor out-of-sample performance. Simpler models or models trained on smaller feature sets performed more reliably, highlighting the importance of parsimony in this domain.

4.4 Betting Strategy

With our probability estimates $\hat{p}_i$ from the predictive model, the final component of the pipeline is the betting strategy. The central task is to compare these probabilities to the bookmaker's implied probabilities and identify wagers where a positive edge exists. To determine our stake sizes b_i, we use the Kelly criterion, which provides optimal fractional wagering amount to maximize long-term bankroll growth accounting for risk. Beyond Kelly, we also consider portfolio-style optimizations guided by Pareto efficiency, where allocations are balanced across competing objectives of maximizing expected return and minimizing variance. This approach allows for the construction of bet portfolios that lie on the efficient frontier of risk–reward trade-offs. While these ideas frame our methodology, we withhold further details on the exact implementation of our staking and optimization algorithms to preserve the proprietary nature of our trading.

5 Results

To evaluate the overall performance of our pipeline, we analyze the prediction accuracy of our final models across major player prop categories. Figure 1 illustrates the out-of-sample predictive accuracy achieved by our final models on a selection of prominent NBA and NFL prop lines. As defined above, prediction accuracy is the percentage of cases where the predicted class, determined by rounding $\hat{p}_i$ to the nearest binary outcome, matched the actual Over/Under result.

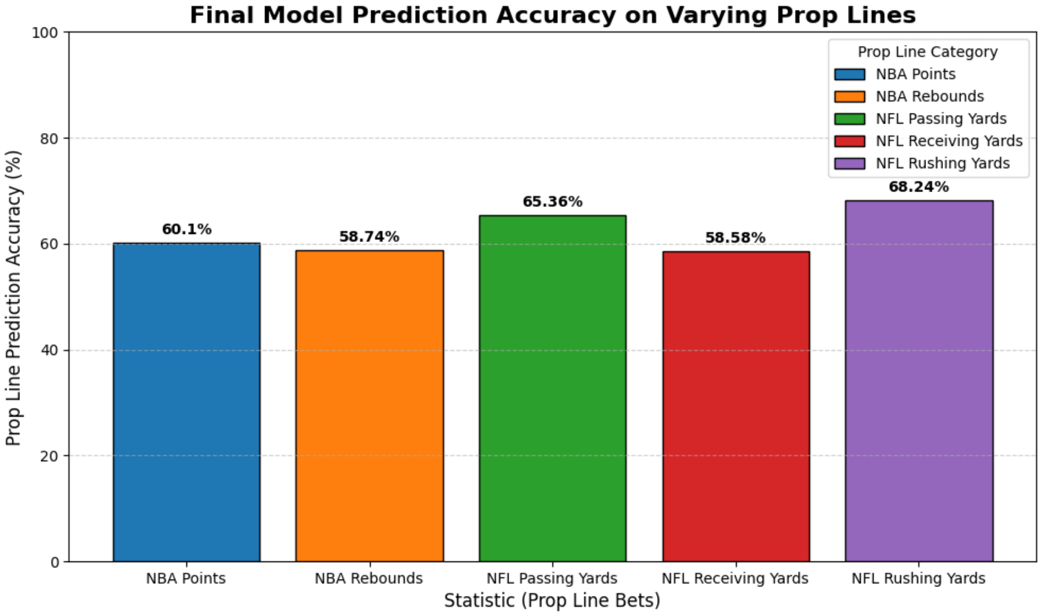

Figure 1: Final model prediction accuracy of selected NBA and NFL player prop categories across 2023 and 2024 seasons

Overall, we observe accuracies in the 58–68% range, which exceeds the bookmaker's implied baseline of 54.5% for prop line wagers and suggests the model has a notable edge in certain predictions. We see that NFL passing and rushing yard props exhibit the strongest performance, likely due to the relatively stable distribution of quarterback and running back usage. In contrast, NBA rebound props and NFL receiving yard props show more volatility, which intuitively reflect the higher contextual variance in those statistics.

In addition to static backtesting of our models, we also conducted live-betting to evaluate real-world profitability under our staking strategies. While more detailed financial outcomes are currently witheld, we note that results were consistent with validation accuracy and demonstrated promising bankroll growth. A more thorough evaluation of live performance and robustness under real market conditions is left for future work. The following results are taken from a period of live betting from March-April 2025 consisting of originally $2.50 bets to $6.00 bets given an initial bankroll of $100.

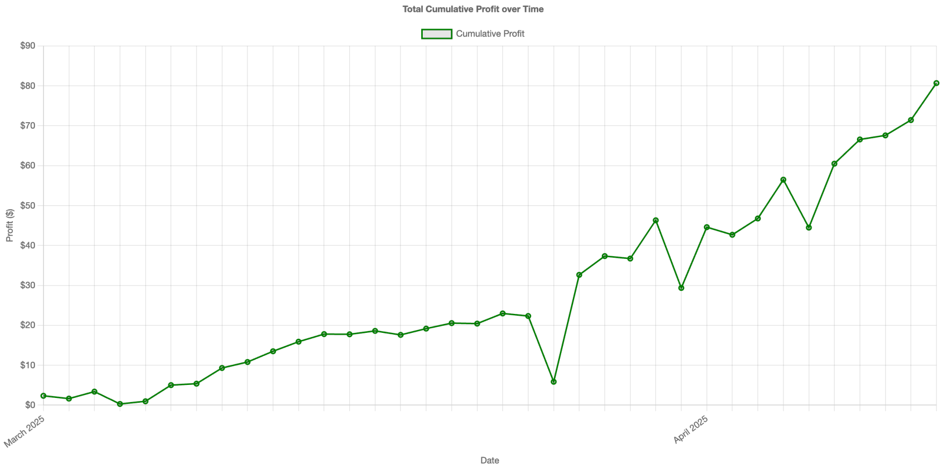

Figure 2: Total Cumulative Profit over Prop Line Bets

6 Future Directions

While our current pipeline demonstrates meaningful predictive performance and profitable betting strategies, there are other couple directions that could improve the pipeline's performance. A key area of interest is the integration of large language models (LLMs) into the workflow. Currently, we rely on only quantitative data to model a player's performance. However, qualitative data such as injuries, game pressure, and psychological factors (”revenge” games) may play a significant factor in player performance. Thus, we could integrate our final model outputs into an LLM that could also learn from:

- **News and Injury Reports:** Extract live signals from player injury announcements, coaching interviews, and pre-game reports.
- **Rotation and Usage Insights:** Summarize game previews or post-game commentary to estimate expected changes in playing time, lineup adjustments, or matchup-specific strategies.
- **Market Dynamics:** Analyze sportsbook line movement commentary and social media sentiment to capture shifts in the market.

Another direction is that continued exploration of portfolio optimization techniques beyond Pareto frontiers remains an important challenge. Incorporating multi-objective optimization frameworks that account for factors beyond variance and expected value could lead to more profitable betting strategies. Furthermore, our current betting approach is highly directional, correlating directly to the model's predictions. Future work could incorporate hedging strategies designed to mitigate risk and reduce variance, particularly given the inherently volatile nature of player prop markets.

7 Fun Fact

Out of all NBA data points for the 2024 season, there was only one discrepancy between the official NBA site and market data. On December 21st, ESPN and the official NBA site had Bam Adebayo logged for 24 points against the Orlando Magic, but market data had Bam Adebayo logged for only 23 points that day.