Betting Specifications Document

Project Name: Sports Betting Application

Project Link: https://comp-413-sports-betting.vercel.app/

Author: Alex Biyanov

Here we outline my own novel suggested approach for how our app will generate betting odds for NBA games. This is to be revised, improved, and implemented by the Betting Lead.

I. Calculate the Initial (Seed) Probabilities

Suppose the game to generate odds for is between team A and team B. A and B go into the game with certain records which impact our perspective on their performance. For each team, **we no longer** compute a win percentage from their record. We instead ask AI to scrape the web, and use the win percentage and other factors to come up with initial probabilities, which we name as:

$pA_{initial}$ and $pB_{initial}$.

II. Calculate the Implied (Market) Probabilities

Now users on the app bet on the game between A and B, and everytime a bet is made, we dynamically shift our market.

Let:

$|A| = total\ amount\ bet\ on\ A$

$|B| = total\ amount\ bet\ on\ A$

Then:

$pA_{market} = pA_m = |A| / (|A| + |B|)$ **if** $|A|$ **or** $|B|$ **else** pA_i

$pB_{market} = pB_m = |B| / (|A| + |B|)$ **if** $|A|$ **or** $|B|$ **else** pB_i

III. Combine the Initial and Implied Probabilities

We would like the odds to be generated from a combination of the initial and implied probabilities. To accomplish this, we use a weighting factor α.

For example:

$pA_{weighted} = pA_w = \alpha\,(pA_i) + (1 - \alpha)(pA_m)$

$$pB_{weighted} = pB_w = \alpha\,(pB_i) + (1 - \alpha)(pB_m)$$

Normalizing so that these can serve as final probabilities:

$$pA_{final} = pA_f = pA_w / (pA_w + pB_w)$$

$$pB_{final} = pB_f = pB_w / (pA_w + pB_w)$$

IV. From Probabilities to Odds

Express as decimal odds:

$$odds\ that\ A\ wins = 1/pA_f$$

$$odds\ that\ B\ wins = 1/pB_f$$

V. Smooth Updating

We do not want users to have to suffer sudden jarring changes as they bet. To avoid this, we might use a smoothing function to limit the update to a maximum amount (δ) per odds readjustment.

For example, suppose we have calculated a $pA_{final,\,new}$ to change to from a $pA_{final,\,old}$.

Then:

$$pA_f = pA_{final,\,old} \pm \delta \textbf{ if } |pA_{final,\,new} - pA_{final,\,old}| \,/ pA_{final,\,old} > \delta \textbf{ else } pA_{final,\,new}$$

VI. Updating Payout

We want to mimic stock trading with sports trading, so bets can go up and down in value.

We now have a market for games, which means we can image the bets as trades that function in said market. When the odds of a game go up and down, a bet that was made on the game changes its value, just like an investment that was made in the stock market.

Suppose a user bet P on team A in a game, expecting a payout of $E = P / pA_i$.

Then the underlying odds of the game change with more bets, from pA_i to pA_f.

The value V of the user's bet can be calculated at any given current odds point as:

$$V = P * (pA_f / pA_i)$$