

# Betting Specifications Document

**Project Name:** Sports Betting Application

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

**Author:** Alex Biyanov

Here I 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, compute a win percentage from their record:

$$pA_0 = (\text{games won by A}) / (\text{games played by A})$$

$$pB_0 = (\text{games won by B}) / (\text{games played by B})$$

Normalize the win percentages so they can serve as the baseline initial probabilities:

$$pA_{initial} = pA_i = pA_0 / (pA_0 + pB_0)$$

$$pB_{initial} = pB_i = pB_0 / (pA_0 + pB_0)$$

## 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:

$$|A| = \text{total amount bet on A}$$

$$|B| = \text{total amount bet on A}$$

Then:

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

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

## 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  $\alpha$ .

For example:

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

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

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 ( $\delta$ ) per odds readjustment.

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

Then:

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

## VI. Additional Considerations

Time permitting, it would be highly valuable to also consider:

- A. Team Form and Recent Performance: this can affect  $pA_i, pB_i$
- B. Home-Court Advantage: this can affect  $pA_i, pB_i$
- C. Head-to-Head Records: limit scope to a certain period, maybe season so far?
- D. Difficult Recent Schedule: hard but not impossible to quantify, this would include long travel, back-to-back games, player fatigue and resting players