Using a Bayesian Hierarchical Model to predict future NBA game spreads and try to generate a profit on sports betting

Peter Chu

November 2024

1 Introduction

Sports betting is a fun "hobby" that millions of Americans take part in every year. From placing bets on favorite football teams, to trying to make the perfect March Madness bracket, billions of dollars are moved. However, in the case of sports betting, it is often like casino gambling. The majority of players lose all their money, while few take it all. Knowledge and good understanding of a sport and its players from prior games can help people make educated bets, but there is no guaranteed way to make correct predictions. This project aims to capitalize on said prior information to make more confident bets, specifically for NBA money line bets. Moneyline bets are defined as simply betting on the overall winner of the match and payout amounts are defined by odds.

This project will use the 2023-2024 NBA regular season as prior data for the Bayesian hierarchical model which aims to predict the spread of future 2024-2025 NBA season games. This is in turn to try and win moneyline bets.
Some limitations of this model is that it will only focus on teamwide statistics. It will not include any specific player effects nor take into account any roster shuffles.

This project follow these steps:

1. Use past research/academic papers to create a reasonable model to estimate total points scored by a team.
2. Use all 2023-2024 regular season NBA games as prior information. We use only regular games because playoffs involves multiple matches between the same teams which affects the model negatively. For example, the scoring intensities and adaptions to opponents can affect model assumptions which aims to predict regular season games, not playoff games.
3. Update the model with the already played 2024-2025 regular season NBA games and then use the posterior to predict future matches.

4. Use predictions, actual match results, and sports betting website odds to calculate money gains and losses. It is important to note that we are not believing/trying to make the model 100% accurate, but rather $> 50\%$ accurate. This is so that it is more accurate than simply tossing a coin and betting.

1.1 Moneyline betting explanation

Moneyline bets are defined as bets that simply choose the winner of a game. However, it is important to note that the team that is expected to win will often have a lower payout than the underdog team. Thus, even if our spread predictions are correct, that doesn't mean it is always the best use of money. Also, it a hit or miss bet, so you either win your money back plus a certain payout, or lose everything. To understand the notation of a money line bet, it is often written as: TEAM 1 : +/-XXX TEAM 2 : -/+ XXX, where the +XXX indicates the payout for correctly betting $100, while the -XXX indicates the amount required to bet to win $100 [1][2]. For example, one game that we will try to predict later is the Los Angeles Lakers vs Oklahoma City Thunder game played on November 29, 2024. The betting websites had the moneyline odds as LAL +125 OKC -145. This means that if you were to bet $100 on the Lakers, the underdogs, and the Lakers win, you would win back your initial $100 plus an additional $125. If you were to bet on OKC winning, you would need to correctly bet $145 to win $100. The general formula is as follows [2]:

$$\text{Profit for +XXX Odds} = \frac{\text{Odds}}{100} \times \text{Bet Amount}$$
$$\text{Profit for -XXX Odds} = \frac{100}{\|\text{Odds}\|} \times \text{Bet Amount}$$

2 Model creation

When creating this model, it was mainly influenced by the papers ”Bayesian Hierarchical Modelling of Basketball Team Performance: An NBA Regular Season Case Study” [3] by Paul Attard, David Suda, and Fiona Sammut, and ”Bayesian hierarchical model for the prediction of football results” [4] by Gianluca Baio and Marta A. Blangiardo. From these papers it was decided to set up the model as such:

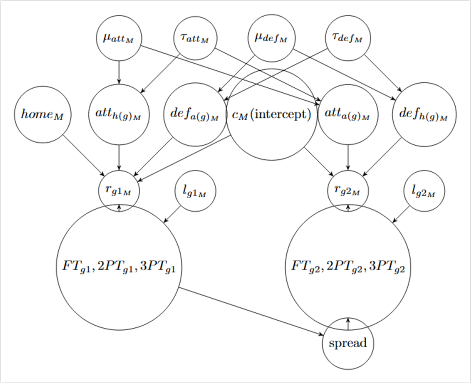


Figure 1: Graph of model

The distributions and meaning for each parameter is as follow:

$$FT_{gj} \mid l_{gj_{\text{FT}}}, r_{gj_{\text{FT}}} \sim \text{Negative Binomial}(\mu = l_{gj_{\text{FT}}}, \sigma = r_{gj_{\text{FT}}}) \quad (1)$$
$$2PT_{gj} \mid l_{gj_{\text{2PT}}}, r_{gj_{\text{2PT}}} \sim \text{Negative Binomial}(\mu = l_{gj_{\text{2PT}}}, \sigma = r_{gj_{\text{2PT}}}) \quad (2)$$
$$3PT_{gj} \mid l_{gj_{\text{3PT}}}, r_{gj_{\text{3PT}}} \sim \text{Negative Binomial}(\mu = l_{gj_{\text{3PT}}}, \sigma = r_{gj_{\text{3PT}}}) \quad (3)$$
$$TP_{gj} = FT_{gj} + 2 \times 2PT_{gj} + 3 \times 3PT_{gj} \quad (4)$$

Where g represents a single match, j represents if the game was a home or away game (1 for home, 2 for away), l represents the probability of a shot type being made and follows as Uniform(0,1) distribution. This is important as it must take values between 0 and 1. r represents the stopping parameters with respect to shot type. TP represents the total points scored in match g for j (home or away) team

r_{gj_M} was furthered modeled into components with att_M and def_M reprinting the attack (offense) and defense intensities, $h(g)$ and $a(g)$ representing the home and away team index, c_M representing the intercept, and $home_M$ representing the home advantage. M represents the type of shot being Free Throw (FT), 2 Point Shot (2PT), or 3 Point Shot (3PT).r_{gj_m} was modeled as such:

$$\log(r_{g0_M}) = att_{h(g)_M} + def_{a(g)_M} + c_M + home_M \quad (5)$$
$$\log(r_{g1_M}) = att_{a(g)_M} + def_{h(g)_M} + c_M \quad (6)$$

For each parameter, the prior distributions were chosen:

$$
\begin{aligned}
att_{t_M} &\sim \text{Normal}(\mu = \mu_{att}, \tau = \tau_{att}) && (7)\\
def_{t_M} &\sim \text{Normal}(\mu = \mu_{def}, \tau = \tau_{def}) && (8)\\
\mu_{att_M} &\sim \text{Normal}(\mu = 0, \tau = 0.0001) && (9)\\
\mu_{def_M} &\sim \text{Normal}(\mu = 0, \tau = 0.0001) && (10)\\
\tau_{att_M} &\sim \text{Gamma}(\mu = 0.1, \sigma = 0.1) && (11)\\
\tau_{def_M} &\sim \text{Gamma}(\mu = 0.1, \sigma = 0.1) && (12)\\
home_M &\sim \text{Normal}(\mu = 0, \tau = 0.001) && (13)\\
c_M &\sim \text{Normal}(\mu = 0, \tau = 0.001) && (14)
\end{aligned}
$$

Where t represents the team index.

A constraint added for identifiability purposes was the sum-to-zero constraint and was done by subtracting the mean from each att_t and def_t values [4].(There are 30 teams in the NBA) Thus:

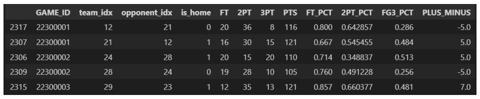
$$\Sigma_{t=0}^{30} att_{t_M} = 0 \text{ and } \Sigma_{t=0}^{30} def_{t_M} = 0$$

With the model setup, we can now use our 2023-2024 NBA regular season data in the model

3 2023-2024 NBA regular season data and model

The 2023-2024 NBA regular season data was obtained using the 'nba_api' package [5]. It contains detailed statistics for almost every NBA and G league game and also includes player statistics.

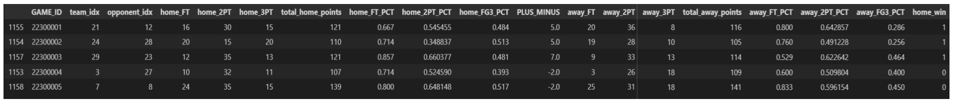
Using this package, the data was obtained and then transformed to only include necessary columns which were: the unique game ID, unique team ID, home team indicator, number of shots made for each shot type, total points, each shot type percentage made, and the overall game +/-.



	GAME_ID	team_idx	opponent_idx	is_home	FT	2PT	3PT	PTS	FT_PCT	2PT_PCT	FG3_PCT	PLUS_MINUS
2317	22300001	12	21	0	20	36	8	116	0.800	0.642857	0.286	-5.0
2307	22300001	21	12	1	16	30	15	121	0.667	0.545455	0.484	5.0
2306	22300002	24	28	1	20	15	20	110	0.714	0.348837	0.513	5.0
2309	22300002	28	24	0	19	28	10	105	0.760	0.491228	0.256	-5.0
2315	22300003	29	23	1	12	35	13	121	0.857	0.660377	0.481	7.0

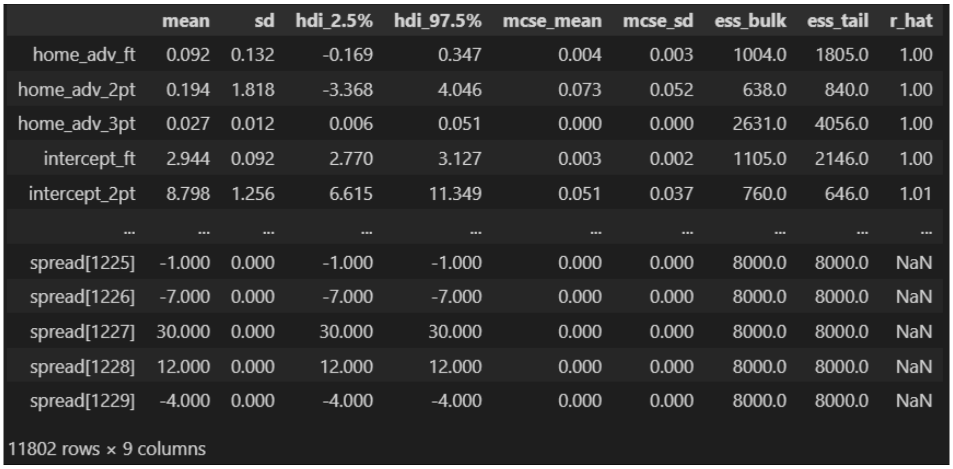
Figure 2: Sample rows from transformed data

However, this data clearly contains 2 rows per game. One for the home team statistics, and one for the away team statistics. Thus this data needs to be "unstacked" for our model purposes to avoid double counting statistics.
Now that the data is fully transformed, we can use it in our model. It was run using 2,000 samples with 1,000 burn ins.



	GAME_ID	team_idx	opponent_idx	home_FT	home_2PT	home_3PT	total_home_points	home_FT_PCT	home_2PT_PCT	home_FG3_PCT	PLUS_MINUS	away_FT	away_2PT	away_3PT	total_away_points	away_FT_PCT	away_2PT_PCT	away_FG3_PCT	home_win
1155	22300001	21	12	16	30	15	121	0.667	0.545455	0.484	5.0	20	36	8	116	0.800	0.642857	0.286	1
1154	22300002	24	28	20	15	20	110	0.714	0.348837	0.513	5.0	19	28	10	105	0.760	0.491228	0.256	1
1157	22300003	29	23	12	35	13	121	0.857	0.660377	0.481	7.0	9	33	13	114	0.529	0.622642	0.464	1
1153	22300004	3	27	10	32	11	107	0.714	0.524590	0.393	-2.0	3	26	18	109	0.600	0.509804	0.400	0
1158	22300005	7	8	24	35	15	139	0.800	0.648148	0.517	-2.0	25	31	18	141	0.833	0.596154	0.450	0

Figure 3: Sample rows from transformed and unstacked data

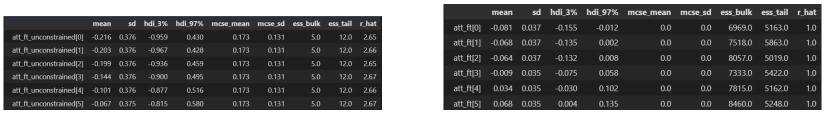


	mean	sd	hdi_2.5%	hdi_97.5%	mcse_mean	mcse_sd	ess_bulk	ess_tail	r_hat
home_adv_ft	0.092	0.132	-0.169	0.347	0.004	0.003	1004.0	1805.0	1.00
home_adv_2pt	0.194	1.818	-3.368	4.046	0.073	0.052	638.0	840.0	1.00
home_adv_3pt	0.027	0.012	0.006	0.051	0.000	0.000	2631.0	4056.0	1.00
intercept_ft	2.944	0.092	2.770	3.127	0.003	0.002	1105.0	2146.0	1.00
intercept_2pt	8.798	1.256	6.615	11.349	0.051	0.037	760.0	646.0	1.01
...	...	...	...	...	...	...	...	...	...
spread[1225]	-1.000	0.000	-1.000	-1.000	0.000	0.000	8000.0	8000.0	NaN
spread[1226]	-7.000	0.000	-7.000	-7.000	0.000	0.000	8000.0	8000.0	NaN
spread[1227]	30.000	0.000	30.000	30.000	0.000	0.000	8000.0	8000.0	NaN
spread[1228]	12.000	0.000	12.000	12.000	0.000	0.000	8000.0	8000.0	NaN
spread[1229]	-4.000	0.000	-4.000	-4.000	0.000	0.000	8000.0	8000.0	NaN

11802 rows × 9 columns

Figure 4: Sample rows from trace

The model ended up having 251 rows where $\hat{r} > 1.01$ and requires investigation. Due to the large amount of variables and number of samples for each variable, it is not possible to concisely show it all in figures. Thus, all the variable outputs are in separate excel files. Noticeably, most variables that had a $\hat{r} > 1.01$ were in the range of [1.01,1.06], and those that didn't were primarily the unconstrained $\mu_{att/def}$ Thus, we can proceed without worry that these will largely affect the model.



	mean	sd	hdi_3%	hdi_97%	mcse_mean	mcse_sd	ess_bulk	ess_tail	r_hat
att_ft_unconstrained[0]	-0.216	0.376	-0.959	0.430	0.173	0.131	5.0	12.0	2.65
att_ft_unconstrained[1]	-0.203	0.376	-0.967	0.428	0.173	0.131	5.0	12.0	2.66
att_ft_unconstrained[2]	-0.199	0.376	-0.936	0.459	0.173	0.131	5.0	12.0	2.65
att_ft_unconstrained[3]	-0.144	0.376	-0.900	0.495	0.173	0.131	5.0	12.0	2.67
att_ft_unconstrained[4]	-0.101	0.376	-0.877	0.516	0.173	0.131	5.0	12.0	2.66
att_ft_unconstrained[5]	-0.067	0.375	-0.815	0.580	0.173	0.131	5.0	12.0	2.67

(a) Unconstrained att_ft

	mean	sd	hdi_3%	hdi_97%	mcse_mean	mcse_sd	ess_bulk	ess_tail	r_hat
att_ft[0]	-0.081	0.037	-0.155	-0.012	0.0	0.0	6969.0	5163.0	1.0
att_ft[1]	-0.068	0.037	-0.135	0.002	0.0	0.0	7518.0	5863.0	1.0
att_ft[2]	-0.064	0.037	-0.132	0.008	0.0	0.0	8057.0	5019.0	1.0
att_ft[3]	-0.009	0.035	-0.075	0.058	0.0	0.0	7333.0	5422.0	1.0
att_ft[4]	0.034	0.035	-0.030	0.102	0.0	0.0	7815.0	5162.0	1.0
att_ft[5]	0.068	0.035	0.004	0.135	0.0	0.0	8460.0	5248.0	1.0

(b) Constrained att_ft

Figure 5: Unconstrained vs Constrained $\hat{r}$ comparison

Again to avoid taking up many pages with graphs, all the posterior graphs are in a separate file.

Some immediate issues that are noted have to due with l_{2pt} and l_{3pt}. They both have extremely high mean values and asymmetrical/jagged graphs as seen in Figure 4.

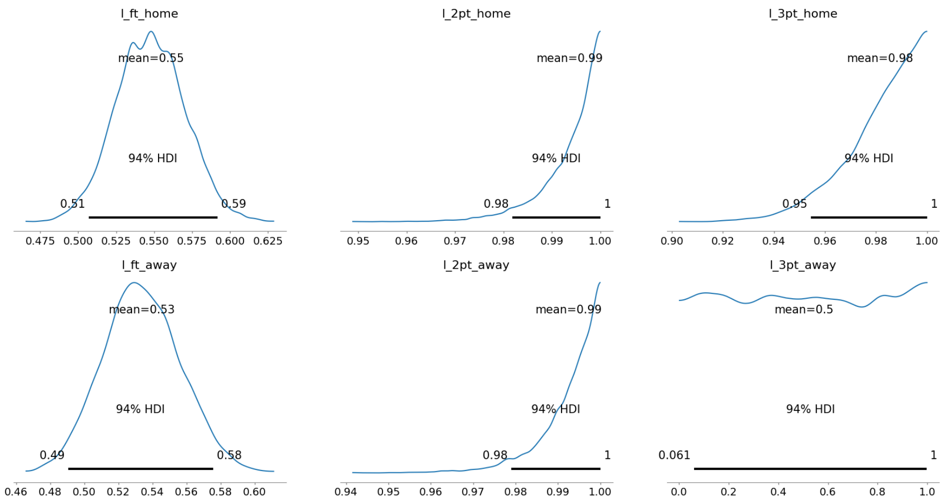


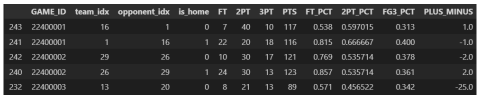
Figure 6: 2023-2024 l graphs

The free throw graph for both home and away is fine makes sense, but the graphs for 2PT and 3PT does not. Since we are following the model from papers [1] and [2], we will not change it, but for the 2024-2025 model we will change it to a different distribution that still only produced values in between 0 and 1.

Now that we have parameter estimates for each variables, we can use our 2023-2024 posterior estimates as our prior estimates for our 2024-2025 model and then use it to generate predictions for future games.

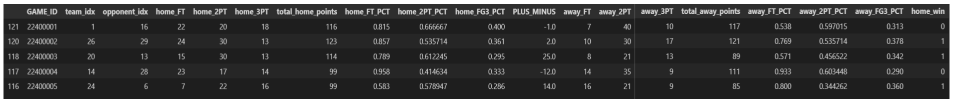
4 2024-2025 data and updating 2023-2024 model

The 2024-2025 NBA regular season data was obtained the same way as the 2023-2024 NBA regular season data using the 'nba_api' package again and performing the same transformations.



	GAME_ID	team_idx	opponent_idx	is_home	FT	2PT	3PT	PTS	FT_PCT	2PT_PCT	FG3_PCT	PLUS_MINUS
243	22400001	16	1	0	7	40	10	117	0.538	0.597015	0.313	1.0
241	22400001	1	16	1	22	20	18	116	0.815	0.666667	0.400	-1.0
242	22400002	29	26	0	10	30	17	121	0.769	0.535714	0.378	-2.0
240	22400002	26	29	1	24	30	13	123	0.857	0.535714	0.361	2.0
232	22400003	13	20	0	8	21	13	89	0.571	0.456522	0.342	-25.0

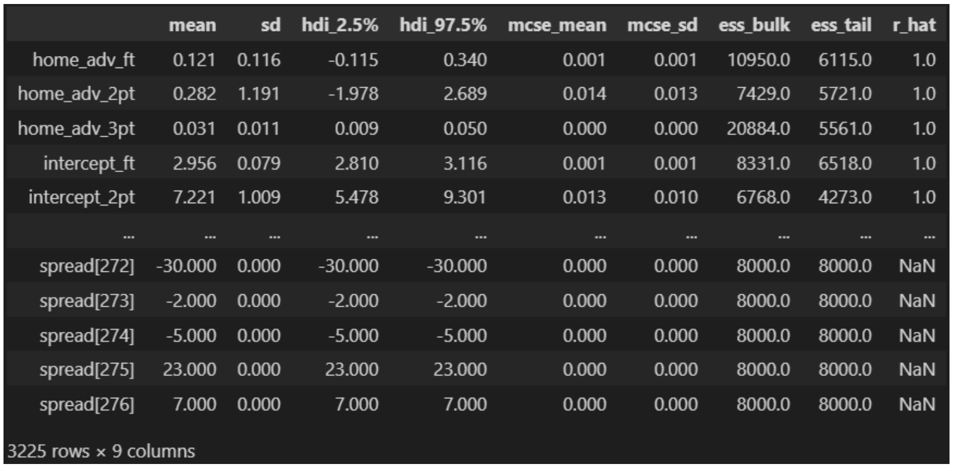
Figure 7: Sample rows from transformed data



	GAME_ID	team_idx	opponent_idx	home_FT	home_2PT	home_3PT	total_home_points	home_FT_PCT	home_2PT_PCT	home_FG3_PCT	PLUS_MINUS	away_FT	away_2PT	away_3PT	total_away_points	away_FT_PCT	away_2PT_PCT	away_FG3_PCT	home_win
121	22400001	1	16	22	20	18	116	0.815	0.666667	0.400	-1.0	7	40	10	117	0.538	0.597015	0.313	0
120	22400002	26	29	24	30	13	123	0.857	0.535714	0.361	2.0	10	30	17	121	0.769	0.535714	0.378	1
118	22400003	20	13	15	30	13	114	0.789	0.612245	0.295	25.0	8	21	13	89	0.571	0.456522	0.342	1
117	22400004	14	28	23	17	14	99	0.958	0.414634	0.333	-12.0	14	35	9	111	0.933	0.603448	0.290	0
116	22400005	24	6	7	22	16	99	0.583	0.578947	0.286	14.0	16	21	9	85	0.800	0.344262	0.360	1

Figure 8: Sample rows from transformed and unstacked data

Now with our new data set as observed, we can run the model again drawing 2,000 samples with 1,000 burn ins. This time we will change l to follow a Beta($\alpha = 1, \beta = 1$) distribution. This still only has values between 0 and 1 and has a mean of 0.5. This will hopefully align our l values to be more realistic.



	mean	sd	hdi_2.5%	hdi_97.5%	mcse_mean	mcse_sd	ess_bulk	ess_tail	r_hat
home_adv_ft	0.121	0.116	-0.115	0.340	0.001	0.001	10950.0	6115.0	1.0
home_adv_2pt	0.282	1.191	-1.978	2.689	0.014	0.013	7429.0	5721.0	1.0
home_adv_3pt	0.031	0.011	0.009	0.050	0.000	0.000	20884.0	5561.0	1.0
intercept_ft	2.956	0.079	2.810	3.116	0.001	0.001	8331.0	6518.0	1.0
intercept_2pt	7.221	1.009	5.478	9.301	0.013	0.010	6768.0	4273.0	1.0
...	...	...	...	...	...	...	...	...	...
spread[272]	-30.000	0.000	-30.000	-30.000	0.000	0.000	8000.0	8000.0	NaN
spread[273]	-2.000	0.000	-2.000	-2.000	0.000	0.000	8000.0	8000.0	NaN
spread[274]	-5.000	0.000	-5.000	-5.000	0.000	0.000	8000.0	8000.0	NaN
spread[275]	23.000	0.000	23.000	23.000	0.000	0.000	8000.0	8000.0	NaN
spread[276]	7.000	0.000	7.000	7.000	0.000	0.000	8000.0	8000.0	NaN

3225 rows × 9 columns

Figure 9: Sample rows from 2024-2025 trace

As we can see, the only $\hat{r}$ values greater than 1.01 are still the unconstrained att and def variables. Thus our model has converged well, but is still not perfect. The model l values and graphs are still unrealistic. Thus, for future models, this should be looked into more thoroughly to resolve. For now, we will continue to use this model for predictions.

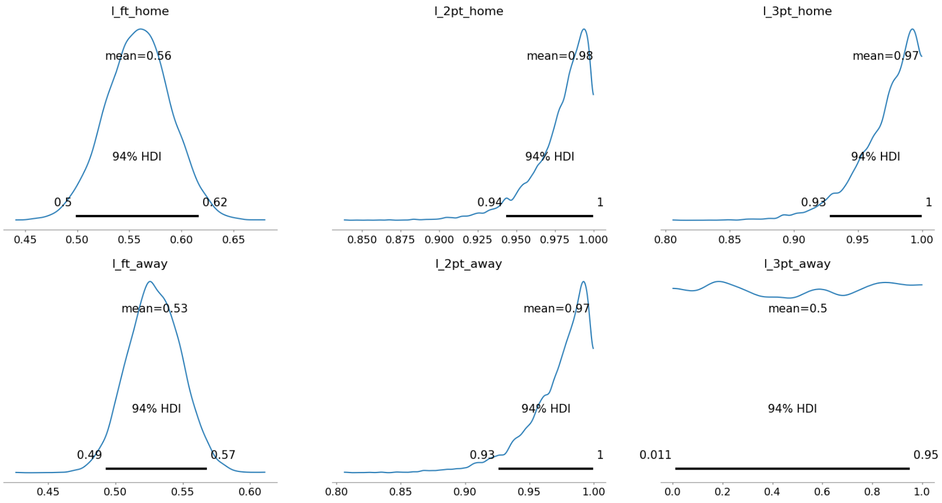


Figure 10: 2024-2025 model l graphs

5 Predictions for selected matches and results

Using our model we can look at future NBA games/matchups and get the predicted total points for each team. It is important to remember that we did not model any sort of player effects and only did a team wide model. Thus, this model does not account for any roster changes made prior to the start of the season. To see how our model performs, we will get the predicted spread of 3 games, and compare them with actual results. I chose these 3 games played on November 29, 2024:

1. Los Angeles Lakers (LAL) vs Oklahoma City Thunder (OKC) with LAL being the home team
2. Miami Heat (MIA) vs Toronto Raptors (TOR) with with MIA being the home team
3. Chicago Bulls (CHI) vs Boston Celtics (BOS) with CHI being the home team

I will also include the moneyline odds proposed by a online sports betting information website [6][7][8] to see if our model can generate profit with the assumption of $100 bets. The odds were set as:

1. LAL +125 OKC -145
2. MIA -340 TOR +270
3. CHI +510 BOS -740

Getting the mean of our posterior samples for the proposed matches we have that:

1. For the LAL vs OKC game our model predicted a +9.28 point lead LAL win, with the actual results being 93-101 OKC win
2. For the MIA vs TOR game our model predicted a +14.9 MIA win, with the actual result being 121-111 MIA win
3. For the CHI vs BOS game our model predicted a +4.2 BOS win, with the actual result being 129-138 BOS win

Thus our model correctly predicted 2 out of 3 games correctly with an expected profit of: -$100 for the LAL-OKC game, $29.4 for the MIA-TOR game, and $13.51 for the CHI-BOS game, totaling a total net profit of -$57.07.

6 conclusion

Overall we can see that our model had issues, but was still able to generate predictions. However, this model is not very refined, nor complex enough to capture all of the randomness in sports nor generate any completely meaningful predictions. Even though the model correctly predicted 2 out of 3 games, it did not take into account every factor, and only made the predictions off limited prior data. Additionally, assuming we had bet $100 on each game, we would be at a net negative profit. This goes to show that people can not just blindly bet with even $> 50\%$ accurate models as the losses exponentially outpace the gains.

I believe that this model had a good setup and was able to generate partially meaningful predictions and its utilization of a Bayesian hierarchical model led results that were better than just simply flipping a coin.

I believe that a far, far more complex model could more confidently make predictions, but not necessarily increase the number of accurate predictions. Sports is noisy, upsets happen, and there is a huge amount of randomness. Thus, I believe the model created here serves a good base to continue building on top of.

7 References

[1] Sohail, Shehryar. "Sports Betting Odds: How They Work and How to Read Them." Investopedia, Investopedia, www.investopedia.com/articles/investing/042115/betting-basics-fractional-decimal-american-moneyline-odds.asp. Accessed 1 Dec. 2024.

[2] How to Calculate Potential Payouts in Betting, www.legalsportsreport.com/how-to-bet/payouts/#: :text=The math behind calculating payouts on sports bets&text=When the odds are negative,Odds/100 * Stake = Profit. Accessed 2 Dec. 2024.

[3] Attard, Paul, et al. "Bayesian hierarchical modelling of Basketball Team Performance: An NBA regular season case study." Proceedings of the 11th International Conference on Sport Sciences Research and Technology Support, 2023, pp. 101–111, https://doi.org/10.5220/0012159100003587.

[4] Baio, Gianluca, and Marta Blangiardo. "Bayesian hierarchical model for the prediction of football results." Journal of Applied Statistics, vol. 37, no. 2, 21 Jan. 2010, pp. 253–264, https://doi.org/10.1080/02664760802684177.

[5] Swar. "Swar/NBA_API: An API Client Package to Access the Apis for Nba.Com." GitHub, github.com/swar/nba_api. Accessed 1 Dec. 2024.

[6] https://sportsdata.usatoday.com/basketball/nba/odds/2692553

[7] https://sportsdata.usatoday.com/basketball/nba/odds/2692551

[8] https://sportsdata.usatoday.com/basketball/nba/odds/2692552