Zhenghui Chen
zhengh04@stanford.edu

Betting Prediction Using Bayesian Network on NBA Games

Introduction

With the rise of online sports betting, many people have been getting into apps like Prizepicks, Sleeper, and others. Some try to make a quick buck but this is a genuine career for others. However, no matter where you fall on this spectrum, the underlying question stays the same, will a player go over or under a certain stat-line? Some people may just go with their gut feeling ("Lebron's definitely getting 50 tonight, they're playing the Wizards") or they may go a little deeper ("Jokic's out, Nuggets got a nobody center on the bench, Anthony Davis is going to eat"). Not a lot of people make bets off pure statistics as this may take the fun away for certain people or others may just not know how to do it. What I'm trying to do with this project is find the probability that a player will go over or under a certain stat-line so people(me) can make more informed bets backed by statistics and potentially make more money/have fun.

Background

This question, the probability that a player will go over or under a certain stat-line is determined by factors from NBA games such as playing at home or away, how good the opposing team is, prior performance against a team, and etc. Therefore, we can model this main question as P(Stats = stats | Factors = factors) where each stat RV and factor RV are modeled as an indicator variable. For my project, I've decided to have the probability take the form P(Stats = stats | Home = home, Playoff = playoff, Playin = playin, Bad = bad) where Stats can be an indicator variable for different stats (specified later) with 1 indicating going over that threshold and 0 going under, and Home, Playoff, Playin, and Bad indicating the factors I'm considering with 1 indicating which conditions we're in. This is a general inference question that relies on many different RVs so we can use a Bayesian Network to model the relationship between all these nodes/RVs that I've described above. *(Figure 1)*

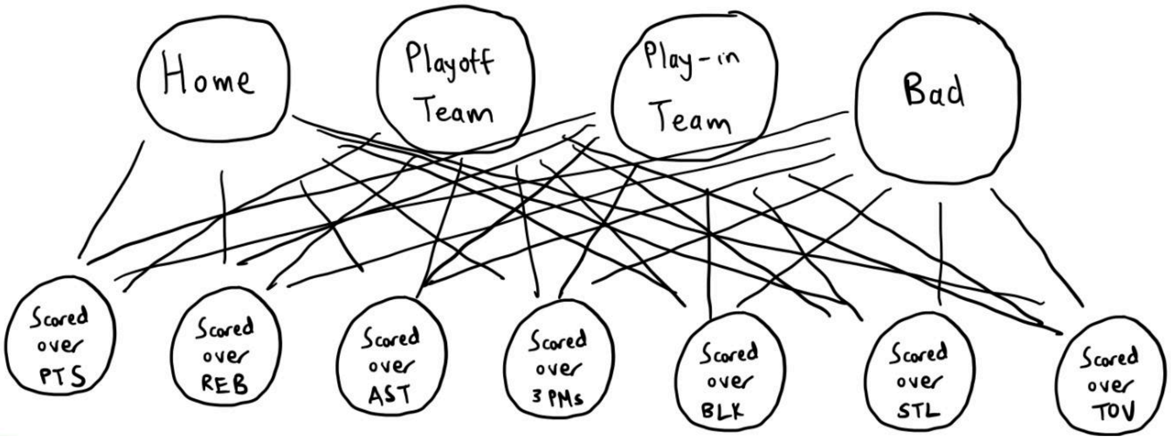


Figure 1: Proposed Bayesian Network

As our root nodes, we have nodes that are unaffected by anything else which is where you play and how good your opponent is. As our leaf nodes, we have stats that are commonly betted on (PTS, REB, AST, 3PM, BLK, STL, and TOV) which are influenced by all the nodes in the layer above them. Playing at home versus a bad team may give a player the best chance to do really good (score a lot of PTS, REB, or some other stat) while playing away against a playoff team will greatly reduce the chance of a player doing good. As with all Bayesian networks, we will need to find the probabilities of the root nodes as well as our leaf nodes to answer inference questions which will be determined through a counting approach. The set of probabilities that we will need to find is attached in the figure below. *(Figure 2)*

Root Nodes

$$P(\text{Home} = 1) = \frac{\#\text{ games played at home}}{\#\text{ games played}}$$

$$P(\text{Playoff} = 1) = \frac{\#\text{ games played against playoff team}}{\#\text{ games played}}$$

$$P(\text{Playin} = 1) = \frac{\#\text{ games played against playin team}}{\#\text{ games played}}$$

$$P(\text{Bad} = 1) = \frac{\#\text{ games played against bad team}}{\#\text{ games played}}$$

Leaf Nodes

$$\left.\begin{array}{l}
P(\text{PTS} = 1 \mid \text{Home} = 1, \text{Playoff} = 1, \text{Playin} = 0, \text{Bad} = 0) \\
P(\text{PTS} = 1 \mid \text{Home} = 1, \text{Playoff} = 0, \text{Playin} = 1, \text{Bad} = 0) \\
P(\text{PTS} = 1 \mid \text{Home} = 1, \text{Playoff} = 0, \text{Playin} = 0, \text{Bad} = 1) \\
P(\text{PTS} = 1 \mid \text{Home} = 0, \text{Playoff} = 1, \text{Playin} = 0, \text{Bad} = 0) \\
P(\text{PTS} = 1 \mid \text{Home} = 0, \text{Playoff} = 0, \text{Playin} = 1, \text{Bad} = 0) \\
P(\text{PTS} = 1 \mid \text{Home} = 0, \text{Playoff} = 0, \text{Playin} = 0, \text{Bad} = 1)
\end{array}\right\} \text{Same for the other stats}$$

Figure 2: Probabilities we need to find and how to find them (calculation for lead node later in the paper)

Once we have determined all our probabilities, including conditional probabilities, for our Bayesian network, we will utilize the rejection sampling algorithm to return to us the final probability of our bet hitting or not. Utilizing the rejection sampling algorithm also allows us to ignore invalid probabilities which would be probabilities, in this case, where more than one opponent's strength is 1 since it is impossible to play a team that is a playoff and play in team at the same time. As a reminder of rejection sampling, our final probability will be the number of event samples (Stat = stat, Factors = factor) over the number of our observations happening (Factors = factor).

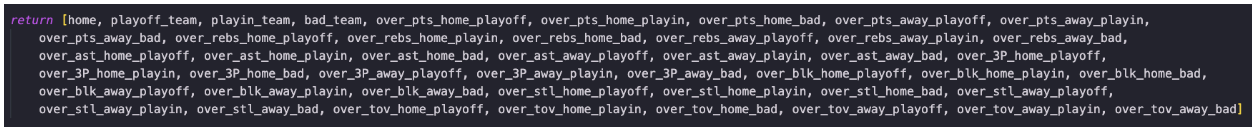
Approach

The coding portion of this problem was done on VSCode using Jupyter Notebook using the Pandas and Scipy libraries. I obtained the stats from an online basketball stat tracker websites (https://www.basketball-reference.com/ and https://www.statmuse.com/) and inputted the values into an Excel sheet which is then modified by code from Pandas to leave us with the data needed to make the necessary calculations. Additionally, the playoff teams, play in teams, and bad teams are stored in lists which are determined by their seeding. Next, we need to find and count the number of games that match our certain conditions over the total number of games played under those conditions. *(Figure 3)*

$$P(\text{PTS} = 1 \mid \text{Home} = 1, \text{Playoff} = 1, \text{Playin} = 0, \text{Bad} = 0) = \frac{\#\text{ games scoring over PTS threshold at home against playoff team}}{\#\text{ games played at home against playoff team}}$$

Figure 3: Calculating the probability for a leaf node. The same idea applies generally

Once all the probabilities are determined, we can utilize the rejection sampling algorithm learned from the lecture to output our final prediction. To do that, however, we will need to know how to translate a bet from the app into an observation list. For example, if there is a bet for Lebron over 25.5 points against the Clippers at home, our observation list would be {'Home': 1, 'Playoff': 1, 'Playin': 0, 'Bad': 0} since he’s playing at home against the Clippers, a playoff team. Next, to count the number of event samples, the numerator of our probability, we need to find the corresponding index in our generated sample list and count the number of times that that appears when we see our observation. *(Figure 4)*



```
return [home, playoff_team, playin_team, bad_team, over_pts_home_playoff, over_pts_home_playin, over_pts_home_bad, over_pts_away_playoff, over_pts_away_playin,
    over_pts_away_bad, over_rebs_home_playoff, over_rebs_home_playin, over_rebs_home_bad, over_rebs_away_playoff, over_rebs_away_playin, over_rebs_away_bad,
    over_ast_home_playoff, over_ast_home_playin, over_ast_home_bad, over_ast_away_playoff, over_ast_away_playin, over_ast_away_bad, over_3P_home_playoff,
    over_3P_home_playin, over_3P_home_bad, over_3P_away_playoff, over_3P_away_playin, over_3P_away_bad, over_blk_home_playoff, over_blk_home_playin, over_blk_home_bad,
    over_blk_away_playoff, over_blk_away_playin, over_blk_away_bad, over_stl_home_playoff, over_stl_home_playin, over_stl_home_bad, over_stl_away_playoff,
    over_stl_away_playin, over_stl_away_bad, over_tov_home_playoff, over_tov_home_playin, over_tov_home_bad, over_tov_away_playoff, over_tov_away_playin, over_tov_away_bad]
```

Figure 4: Sample list from a generated sample for the rejection sampling algorithm

Therefore, Lebron over 25.5 points against the Clippers at home is encapsulated in the variable “over_pts_home_playoff” which has the index 4. After determining our final probability, we will need a threshold to accept or reject the bet which has been put at 0.5 currently. Therefore, if the final probability is 0.5, we can bet the over, and if it’s under 0.5, we can bet the under. As you can see from Figure 4, there are a lot of similar variable names that were tedious to manually change so I used GPT-4 to automate the changing of variable names saving time and ensuring human mistakes were avoided.

Results

I decided to test this out on 6 bets which I showed in the video presentation. I bet Nikola Jokic @ SAS over 8.5 AST, Immanuel Quickley vs. ORL under 35.5 P + R + A, Kawhi Leonard @ NOP under 2.5 3PM, Terry Rozier @ DET under 0.5 BLK, Kevin Durant @ CHA over 26.5 PTS, and Collin Sexton vs. ATL over 22.5 PTS. (@ is an away game, vs. is a home game, abbreviations are opponents). These bets were randomly chosen and have a wide spread of stats we’re betting on, away/home games, skill level of players, over/under bets, and opponent strength. The probabilities for each bet were 75.62%, 2.49%, 44.69%, 13.23%, 62.10%, and 57.34% respectively (calculations shown in the video). Our result ended up being 2/6 bets hitting with two missing being really close to hitting and the last two not being close. We hit on Immanuel Quickley and Kawhi Leonard, missed on Collin Sexton and Terry Rozier by 2 points and 1 block respectively, and missed on Nikola Jokic and Kevin Durant pretty badly. Overall, these results show great promise as the numbers produced are somewhat accurate and there is still much finetuning we can do to improve our percentage of bets hitting discussed below.

Future Work

With these promising results, I will definitely spend more time in the future adding more layers to the Bayesian Network like FG%, 3P%, Field Goals Attempted, Free Throws Attempted, Personal Fouls, and Minutes Played which are influenced by the team you play which ultimately influences your stats making this a middle layer. I could also add more stats to bet on such as Double-Doubles, Triple-Doubles, Technical Fouls, and more in the leaf layer. Additionally, I could also gather more data from past seasons which is always a good thing when making predictions. These are just some directions that I plan to take the project in but there are definitely others. All in all, this was a really interesting project that I embarked on which allowed me to learn more about probability and coding and I look forward to taking this to the next level!

References

https://web.stanford.edu/class/archive/cs/cs109/cs109.1244/lectures/15_general_inference_annotated.pdf