Prabhmeet Gill

OKC Thunder Data Science Internship Questions

General Questions:

1.Give an example of a recent project you worked on using data and describe, generally, from start to finish how you approached, executed and completed the project.

I have been working on a project focusing on the most recent NBA playoffs. For the majority of this past year, I have been a part-time intern for a local start-up called DeepLearn, which uses a machine learning algorithm for daily fantasy NBA and MLB. As a way to gain more experience in statistical analysis and to provide input for their algorithm, I took the initiative and began a project focusing specifically on defensive rebounds. This season has seen a renaissance of guard rebounding lead by the likes of Russell Westbrook and James Harden. Additionally, with the evolution of more point-forwards like Giannis Antetokounmpo, there are more players capable of gaining a rebound and pushing the ball. Given the evolving nature of the game, I thought it would be interesting to see if teams in general would be more successful when their point guards record a defensive rebound. I am looking at two key statistics associated with defensive rebounds: the number of passes on each possession (a measure of the ball movement after the player records a rebound), and the amount of time the team takes on the possession after the rebound (in essence, a measure of the fast-break). I am still currently collecting data, however I have completed 4 playoff series from the first round. Based off preliminary data, I have begun to notice the trend that in fact teams seem to be more successful when the traditional big men rebound. There has been a general movement by certain teams in the NBA highlighting “chaos”. There is a time period between every change of possession that chaos ensues, where there is a higher probability for poor defense. Certain teams like the current Warriors or the past Steve Nash Suns teams were designed to take advantage of that time. But based on my current data, I would postulate that not every team should be aiming for this. Chaos is a double-edged sword. Chaotic possessions, which at times are successful, can also lead to a high number of turnovers; whereas a slower possession leads to more ball movement finding an open uncontested shot easier. In essence, old school fundamentals still have a place in the game.

2.Describe your experience with programming and software development.

I am a Computer Science major with an intended minor in Statistics. Through school, I am comfortable with Java, C, and Python. I also have some experience with Swift, Matlab, and the Unix platform. Additionally, my work experiences have provided me ample opportunity to gain programming and development experience outside of my school work. I interned over the summer of 2016 at JourneySales, a software company. During that time, I learned Apex and HTML, was introduced to JavaScript and Jquery, and worked with key concepts such as HTTP callouts, Oauth 2.0 and JavaScript remoting. More recently, I’ve been working part time with a start up in

Santa Cruz named DeepLearn, a machine learning company. Through that, I write test files in Ruby on Rails to test their machine learning algorithm, which makes optimal lineups for daily fantasy basketball and baseball. Furthermore, over the summer of 2017, I will be an Applications Developer intern at Kaiser Permanente, where I will likely be developing an IOS app.

Project Outline:

1.Describe, generally, from start to finish how you approached, executed and completed the project. Include relevant materials (ex: code) through a cloud storage provider (ex: Dropbox, Google Drive)

My approach to this project has been to take the example given to me and reverse engineer a model based on a statistical approach. I began by examining the example that was given to me - specifically this article (http://www.inpredictable.com/2015/02/updated-nba-win-probability-calculator.html) and the in-game win probability tool on the website. I noted that the win probability at a particular point, more than any other factor, needed to factor time and the current point difference. So, I started by organizing the data into wins and losses and sorting by the point difference at the start of the quarter vs. the point difference at the end. I plotted a histogram to help better visualize the data, and then using the sorted array, I calculated a raw win percentage based off the total number of games with a specific point difference and the total number of games won at that point difference. For instance, my data set included 149 games that began with a point-difference of 5 (home team winning), which resulted in a raw win percentage of 65%. Then using Python's built-in polyfit function, I performed linear regression with point difference as the dependent variable and win percentage as the predicated variable. However, this regression only took into account the raw win percentage at the start of the quarter. I needed to better account for various time intervals within my algorithm. I did not have enough data points at each minute interval for 12 different arrays to be accurate. My solution to this problem was to use Python's multipolyfit function and perform multi-variable linear regression, using a matrix of a point difference array and a time array as the dependent variables, and used the win percentage as the predicted value. This function yielded a graph and 3 constants which I plugged into the equation ax + by + c = win percentage, where x = time, and y = the point difference. This equation lead to my way of calculating the win percentage based off a time and point difference.

To further improve the equation, I tackled variance and importance of the final minutes of the game. For variance, I divided my equation by the standard deviation of the data to standardize the equation. The final minutes of the game are the most important, and in close games, due to the number of timeouts and game management, the flow and gameplan for the game can considerably change. Based on my equation, I did not believe my equation was able to reflect the impact of

pressure from time and change in game flow as well as it should have. So, I created a simple decision tree for when there is less than a minute left in the game. This decision tree is essentially based off key statistics like effective field goal percentage, 2 point field goal percentage, and 3 point field percentage (I used the averages for the NBA this season on basketball reference). For example, if you are down 2 with thirty seconds left you will probably have one possession, so your win percentage/percentage-to-force-overtime on average will be 45.7%. For a four point game it will 12.8% (35.8% * 35.8%), since you need to make two threes in order to tie, assuming both free throws are made by the opposing team. Following similar logic, I calculated win percentages for under a minute, for point differences of 1 through 6.

For the play-by-play changes in win-percentage, I again needed to stress the importance of time and point difference. I accounted for a made shot, missed shot, rebound, free throw, or turnover. For this equation I postulated win percentage change = ((time * stat) / point diff)*10. The stat variable represents how likely an event will occur based off percentages from basketball reference (http://www.basketball-reference.com/leagues/NBA_stats.html) on the 2016-2017 season. For example, the effective field goal percentage for the entire league is 51.4%, so the stat variable would equal .514. I divide by the point difference because if the game is close then the shift should be larger, whereas if the game is a blowout the shift should be smaller. The same goes for time, the longer into the game you are the more the shift should change, whereas the earlier you are the shift should be smaller. I multiply by 10 because I am working with a percent, and it gives a value typically less than 1 if I do not multiply by 10.

2.Include and describe a visualization from the project. The visualization should highlight a feature or insight from your predictive model. Explain the decisions you made in constructing the visualization.

My visualization for the project is similar to the example that was given. I printed a graph of the win percentages throughout the entire game, while highlighting and printing each data point. My other files display a histogram of the win percentage data and show a graph of my linear regression formula. I also added a win probability calculator that mimics the functionality of this link (https://stats.inpredictable.com/nba/wpCalc.php). I ask for the user input of the time in game, the point difference, and the quarter, based off those parameters it spits out a win percentage.

3.Imagine that a coach comes to you and asks you for help with in-game strategy. Using takeaways from your model, describe how you would advise the coach in the following scenarios:

a. We're down 106-105 with 29 seconds to go. They get a defensive rebound. Should we foul in that situation?

Based off my model, we would have about a 46% chance to win the game after the rebound occurs. Furthermore, if the team were to make a shot we would have about a 35% chance to win the game, or at least force OT. Based on average numbers from basketball reference for the league, the opposing team has a 45.7% to make a 2, and a 35.8% to make a three, whereas the average free throw percentage in the NBA is 77.2%. If the team were to miss the shot, we would have about a 49% to win the game. However, if a poor free throw shooter gets the original defensive rebound, then it would be advantageous to foul. By my calculations if the player shoots less than 71.7%, then the foul is in our favor.

The opposing team has a 51.4% chance to score based on the league average effective field goal percentage. If fouled, an average player has 59.5% (77.2%*77.2%) chance to make 2 free throws. For it to be advantageous to foul, a player's chance to score 2 points should be less than both the 59.5% and 51.4%. Taking the lower number, 51.4% and taking the square root of it to reflect the two free throws that must be made, the free throw percentage comes to 71.7%. So essentially it comes down to whether or not the player who gets the rebound shoots less than 71.7% from the free throw line or not. If he does shoot less, then foul; if not then, we should trust our defense.

b. We want to rest key players whenever possible to increase their longevity over the season and limit injury risk. Under what circumstances should the coach consider resting high-minute players while trying to minimize the chance it could affect the outcome of the game?

I think there are two ways to obtain rest for particular players: either reduce minutes in blowouts or to rest them for an entire game. I think when wanting to rest players for an entire game it is strictly matchup dependent. Let's say you want to rest Russell Westbrook, naturally your offensive and even defensive efficiencies will decrease, so in order to compensate for that you want to rest him against teams that have comparable ratings when he is off the court. So, a team like Brooklyn or Phoenix might be a great team to rest him against, since they are so poor defensively, especially against point guards. In these situations, reserve players like Semaj Christon or Norris Cole can be expected to perform at better than normal levels. However, if you play a team like Utah, or Dallas who are great defensively against point guards you will probably see Christon or Cole struggle, whereas someone like Russell Westbrook tends to be matchup proof.

Alternatively, I think reducing the minutes of players in a blowout is also a viable strategy. The coach could set a win percentage threshold, such as 85%. If that point is hit then he can sub out his high minute players, until the end of the game or until the game suddenly gets close. This will gradually reduce the minutes, whereas reducing them all at once. Furthermore, if the game was to end up being close you can still sub in the high usage player back into the game, which you cannot do when he is resting for the entire game.