Jonathan Greenberg
Final Report

Predicting NBA Injuries

Motivation

I started by considering 4 different topics of research. Questions like: Is there such a thing as having a shooting hot streak in basketball? Can I predict the success of a player's next shot using Markov chains? I considered how physical Attributes relate to talent in the NBA. Does your height affect your shot skill, or does your hand size affect the amount of steals you get? I considered how basketball players' college success translates to the NBA. Would I be able to predict NBA success solely based on college statistics?

After much consideration and factoring in different variables like quality of data I can retrieve, wanting an ambitious but realistic goal, and something that interests me the most, I decided to settle on the topic of predicting injuries in the NBA while also incorporating players' physical attributes. Questions that motivated my research were: Do injuries correlate with minutes played? Can I predict when a player is at risk for an injury, and what type of injury is most likely to occur? How does load management play a factor in injury prevention? Should coaches rest players more frequently or is that move unnecessary?

Iterative Process of Defining Goals

What was I looking to predict, and how would I do it? This was a question that was constantly changing from the beginning of the semester until just a few weeks prior to writing this paper. At first my goals were to see if I can predict when a player is most at risk of sustaining an injury. I was interested in seeing on a per-game basis when coaches should purposely rest their players. One topic I peered into for a bit was load management. Can we determine whether load management has any truth to the method? Once I realized getting schedule data would be too difficult, I pivoted from load management to If I am unable to get the schedule/rest day data.

At this stage I wanted to use a classification model: Injured or not? Can we determine if a player will get injured in the following game. To do this I would need data from the previous 5 games prior to the injury, and have half the rows of the data be healthy rows, players who played five games and did not get injured. My plan was to gather that player's data from the previous 5 games and get a meaningful descriptive statistic that would be representative of a deviation from some mean in their performance to indicate the likelihood of an injury in their next game. However, I found that retrieving the data for each game was taking very long and was too granular of a level to predict at…So I pivoted my prediction goal to the season level. Meaning, using a player's features: their age, position, team, weight, height, wingspan, and a bunch of game statistics from the previous season: points, games played, minutes played, personal fouls, etc.), can I predict the type of injury a player is most likely to sustain in the following season? These predictions would help nba players better target care and physical therapy for those parts of their players bodies. I was also interested in seeing what correlates to what kind of injuries/body parts getting injured.

Cleaning and Feature Engineering of Data

The original injury data from Kaggle only had the player, the date, and a description of what the injury was in colloquial english in each row. If the player's injury was in the 2017-18 season for example, then I added in NBA stats from the previous season using their 'player identifier' key from BasketballReference.com using the scraper I mention below under "Data Sources". I removed rows that contained players that played less than 15 minutes a game, because they don't have too many stats for them to be relevant in predicting injuries.

I added columns containing physical attributes like height, weight, wingspan and created an age column as well by subtracting their birth date from the current date. I tokenized the injury description column and parsed it into three separate columns: "severity" which contained words like DNP, DTD, and out indefinitely. The next column was "Body" which contained words like ankle, neck, and finger. The third column "Injury" contained words like sprained, soreness, and broken. I decided to only keep the top 10 categories in both "body" and "Injury" to limit the amount of target classes that I have to predict for.

After all the cleaning have 3473 rows with these features: ['Year', 'Age', 'Minutes','USG%', 'Points', "Height (in cm)", "Weight (in lb)", "Steals", "Defensive_Reb", "Turnovers"] with two of the features as my target categories: "body" and "injury". I was finally ready to begin my modeling.

Modeling

It didn't make sense to include the "injury" in "body part" prediction or vice versa because that is all info we receive once the injury actually occurs. I looked to predict two main variables, the "Body" column and the "Injury" column. I found quickly that the "severity" column was unpredictable, especially because DNP and DTD were treated the same semantically. The attributes I ended up using in predicting were the following: ['Year', 'Age', 'Minutes','USG%', 'Points', "Height (in cm)", "Weight (in lb)", "Steals", "Defensive_Reb", "Turnovers"]

I used the following models in my predictions: Dummy, Naive Bayes, Logistic regression, KNN, decision tree, and Random Forest. I used a dummy model as my first model because I needed a benchmark for what to compare the other models' f1 scores to. The dummy model predicted the values randomly but according to the distribution of the training data's target variables. The hyper-parameter was "(strategy='stratified')". I plotted the weighted (micro) average of the f1 score for each model I used. F1 score takes into account the imbalanced target classes as well as precision and recall. In the Final Report Folder you can find plots of all the models f1 score for predicting body part, and another one for predicting the injury type. Both of these are benchmarked by the dummy model, which has the lowest f1 score.

All the metrics like precision, recall, f1 score, support, and confusion matrices are listed for every type of model in the ipynb titled Analysis_Predictions.

Conclusions

1. There is no correlation between height, weight, or position with injury type or part of body that gets injured, which is an interesting finding. We see this in the correlation matrix.

2. I found that it was nearly impossible to predict the "severity" column. The class "out indefinitely" was highly underrepresented and DNP and DTD are basically the same thing, the NBA apparently switched terms in describing the same severity of an injury in around 2013. In this dataset DNP and DTD are semantically identical.

3. The best model, random forest, had an f1 score of 62%, which compared to the dummy model's score of 17% is pretty good. This can give better advice to NBA teams what type of injuries their players are more likely to sustain and prevent it by paying more attention to those parts of their body.

4. Using RandomForestClassifier feature importance I found that the top predictors of injury type are age, minutes, and turnovers, and the top predictors of body part are FT, minutes, and age. This makes sense because age, how many minutes a game players play, and the amount of free throws you take (which correlates with how aggressive you are and how often you get fouled by the opposing team) all relate to sustaining an injury.

Capstone Folder on Github

https://github.com/Yeshiva-University-CS/Jonathan-Greenberg/tree/master/DS%20Capstone
What can be found in this folder:
Data folder - Contains the final.cvs file is what I perform my predictions on. The rest is just the progression of my csv throughout the cleaning process.
Weekly reports - Contains all 9 of my weekly reports throughout the semester
Final Report - Contains this report and my F1 score model plots
Python Notebooks - Contains 3 ipynb notebooks

Data Sources

I had three main data sources:

1. Injury Data:https://www.kaggle.com/ghopkins/nba-injuries-2010-2018

2. Physical attributes: https://www.kaggle.com/whitefero/nba-players-measurements-19472017?
3. NBA player season Data scraper: https://pypi.org/project/basketball-reference-web-scraper/

Important terms key

1. DNP - did not play
2. DTD - day to day
3. FT - Free Throws
4. 3P - Three Pointers
5. USG% - How much a team uses a player (how many shots he takes, minutes he plays, etc)

F1 scores for both target variables

Injury Type:
Dummy model: 0.19577735124760076
Naive Bayes: 0.2629558541266795
Logistic Regression: 0.30422264875239924
KNN: 0.4980806142034549
Decision Tree: 0.54510556621881
Random Forest: 0.5604606525911708]

Body Part injured:
Dummy model: 0.1727447216890595
Naive Bayes: 0.24088291746641075
Logistic Regression: 0.30230326295585414
KNN: 0.5287907869481766
Decision Tree: 0.5950095969289827
Random Forest: 0.6209213051823417]