Predicting and Preventing Injuries for NBA Players

Jay Messina

Oberlin College
jaymessina3@gmail.com

April 3, 2019

Abstract

Since the NBA began, in-game data such as points, assists, and rebounds of players have been available to coaches and fans. Now, with sensors in the arena the existing data has been expanded to include how fast and far a player ran in a game. With more data being made available on players throughout games, teams can see the usage and playing style of players. This sensor data can be utilized to learn why injuries occur and prevent them through acting on predictive analysis. Throughout this paper I will demonstrate how player's data can be aggregated, even in real time for the current season, and predict injuries in upcoming games using machine learning. The results of this study can be applied to optimize both player usage and experience, resulting in fewer injuries.

I. Introduction

Injuries have been a concern for the NBA and other sports organizations ever since their establishment. Some players sustain minor injuries like a muscle strain while others suffer from major injuries like a torn ACL that require surgery. However, the consistent factor among all athletes is that they continue to push themselves harder and harder until an injury occurs that takes them out of the game. Many people describe these injuries as "freak accidents" that nobody could have seen coming, but what if that wasn't the case? What if there were actually warning signs along the way?

At the 2016 MIT Sloan Sports Analytics Conference, Hisham Talukder and Thomas Vincent presented a paper where they claimed that "by resting the top 20% of high risk [NBA athletes] at any given day there is a potential to prevent 60% of all injuries" [4]. Their model aggregated in-game data from the 2013-2014 and 2014-2015 NBA seasons to make injury predictions throughout those seasons. They argued that by using such a model, teams could be informed if a player was at a high risk of injury for an upcoming game and rest them to prevent it. This model would enable players to be used more effectively and strategically throughout a season. Additionally, they can be warned if an injury is imminent and thus take steps such as being physically evaluated by an athletic trainer to minimize their risk. In an evaluation weaknesses and imbalances in the body can be noted and the player can be given exercises or treatment to correct those issues.

Talukder and Vincent also suggest that a more accurate model could be created if additional data was considered, such as the number of drives to the basket or number of post-ups [4]. They alluded that these could unveil further information on susceptibility to injury. In addition, they suggested that a real time model for the current NBA season would be even more beneficial for players and their team. While being able to predict injuries in past seasons helps our overall understanding of injury trends, it is more useful for coaches and sports organizations to have a model for the current season, which can in turn prevent injuries before they happen. Furthermore, there have been cases where players get injured when fouled, especially with flagrant fouls [5]. I hypothesize that by adding how many times a player is fouled in a game, the model will better predict a higher risk of injury for that player. There is also a high correlation of players who have been injured and have become injured again [6]. For example, when an athlete is recovering from a surgery, it puts them at a much higher risk of re-injury, which could skew the model [6]. Players who have had an ACL injury are 6% more likely to re-tear it, and 12% more likely to tear the other ACL given the effects of muscle compensation [8].

I believe that the accuracy for predicting injuries will increase when adding more variables from SportsVu data [3]. Furthermore, creating a real time model for the current NBA season would give teams the ability to strategically rest and prevent their players from missing significant game time.

II. Related Work

Data scientists have created a body of research revolving around predicting sports injuries. The primary work of which was presented at the MIT Sloan Sports Analytics Conference in 2016. Talukder and Vincent applied machine learning techniques to predict the probability of injury for NBA players. They analyzed SportsVu data [3] to predict whether a player will get injured in the upcoming week. This study found that the five most important features were (1) the average speed at which a player ran during games; (2) the total number of games played; (3) the average distance covered by a player; (4) the average number of minutes played; and (5) the average number of field goals attempted. In the model, they used a 14-day aggregation window of player in-game statistic data and a 7-day prediction window of injury data. With this data, organizations can both optimize the team performance and players health. The researchers also quantified the impact on the organizations revenue, which I will not be looking into.

The measure of performance this study used was Area Under the ROC Curve (AUC), which is measured from 0.0 as the worst to 1.0 as the best. An ROC curve (receiver operating characteristic curve) is a graph that shows how a classification model performs at different thresholds [13]. AUC determines how good the model is for distinguishing the given classes (injured or not injured), in terms of the predicted probability [13]. This study's machine learning algorithm had an overall AUC of 0.86 using the 2013-2014 and 2014-2015 player data. In their data they removed players who played less than 15 minutes on average and any games missed due to non-injury related injuries like sports hernia or illness.

Baseball has also been analyzed for sports injuries. A group of Harvard researchers set out to create a model that predicted MLB pitchers' injuries [7]. They investigated "whether or not a pitcher would get injured the following season based on traditional and advanced statistics from both the previous year and the entire career of each pitcher [7]." They only looked at injuries that relate to the pitcher's arm, back, shoulder, and side as other injuries would not come from pitching. They found that analyzing a pitcher's statistics is useful, but more qualitative variables such as eating habits or pitching mechanics would need to be analyzed for better injury predictions.

Soccer has not been analyzed for sports injuries, but there is a significant amount of data out there that could be used to make predictive models. A group of researchers from Hudl created a passing probability model using tracking data, which records a player's position, and event data, which records statistics like goals, assists, and passes completed during a match [2]. With this data and an injury report for the same year, an injury predictive model could be created.

The challenge with other sports besides basketball is that the necessary data is not readily available or able to be scraped off the internet easily. If I wanted to analyze another sport I would need to reach out to schools or other researchers to obtain their team data. Many developments are being worked on in football, but the data is all not publicly available.

III. Methodology

i. Learning

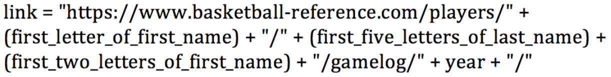
The first step to this project was gaining experience with getting player data from public databases, known as web scraping. For web scraping I utilized a json to csv converter [9] and a python library called Beautiful Soup. This data was then put into sheets by applying another python library called Panda. The next learning obstacle was machine learning. For the machine learning algorithms, accuracy predictions, and AUC I used the Sci-Kit learn python library. Once I finished sample programs working with these technologies I moved on to the NBA data.

ii. SportsVu Data

Step two was getting the SportsVu data. The data table on SportsVu.com is loaded from a json file, so I extracted it by inspecting the elements of the page and downloading the leaguedashptstats.json file. I then loaded the file into the json to csv converter [9]. There were three different pages I needed data from. The first page contained average minutes played, average distance in feet per game, and average speed per game. The second page had average drives to the basket. The last page held average post ups. I combined all these data points into one excel spreadsheet and then removed players that averaged less than 15 minutes of playing time per game, as Talukder and Vincent did in their study [4].

iii. Scraping Basketball Reference

The next step was to make excel sheets with each player's game by game data. I looped through the SportsVu data sheet and parsed the players name correctly to make a request from Basketball Reference's website. I extracted the player's bio for their weight. Next I scraped the table for each game and obtained the minutes, field goal attempts, three point field goal attempts, free throw attempts, offensive rebounds, defensive rebounds, assists, turnovers, fouls, and points.



```
link = "https://www.basketball-reference.com/players/" +
(first_letter_of_first_name) + "/" + (first_five_letters_of_last_name) +
(first_two_letters_of_first_name) + "/gamelog/" + year + "/"
```

Figure 1: URL Formation

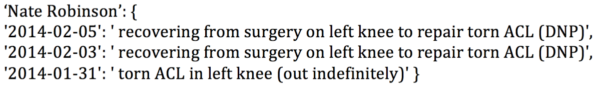
I selected these variables because Talukder and Vincent's study already evaluated and ranked each variable, and I chose the top 50% (Fig 3). They created a graph that displays variable importance ranked from highest to lowest impact. The highest were average game speed and total games played in a season, while the lowest were back-to-back games and total games played in last 14 days. Knowing that the frequency and number of games in the window are not significant contributors in predicting injuries further supports the idea that it is the player's playing style that causes injury.

I also added in the averages from the SportsVu data sheet to each row. Since SportsVu data only holds averages for the entire season, I divided how many minutes the player played in a game by their average minutes played and multiplied it by the SportsVu minutes average. For example, if they averaged 20 minutes of playing time a night and ran 8,000 feet per game, but played 5 minutes that night, their distance would be 2,000 feet. I hypothesize that if SportsVu releases game by game sensor stats instead of season averages, my algorithm would be more accurate.

iv. Adding Injury Data

Step four was to add a column indicating whether an injury occurred in that particular game or not. This injury data was scraped from a pro sports transactions website [10]. I created a dictionary data structure, which kept a list of date/injury pairs for each player (Fig 2). Injuries such as torn ACL, knee sprain, and broken finger were included in the dictionary. Injuries such as flu, illness, sports hernia, and skin infection were not added to the dictionary as they are not in-game related injuries. Each player's spreadsheet was looped through and injuries added accordingly in the injury column (Fig 5).

Below is a portion of Nate Robinson's dictionary and spreadsheet loaded accordingly. In my spreadsheet both '2014-01-31' and '2014-01-29' are listed with a torn ACL, but only '2014-01-31' is in the dictionary I created. Sometimes, as in this case, injuries are not reported until the next day. So, when injuries occurred I loaded the same injury in the cell value above if they played in that game.



```
'Nate Robinson': {
'2014-02-05': ' recovering from surgery on left knee to repair torn ACL (DNP)',
'2014-02-03': ' recovering from surgery on left knee to repair torn ACL (DNP)',
'2014-01-31': ' torn ACL in left knee (out indefinitely)' }
```

Figure 2: Sample of Nate Robinson's Dictionary

v. Sliding Window and Machine Learning

Step five was to load the spreadsheets into training and test sets to make predictions using machine learning algorithms. I loaded the data with a game window instead of a date window, which the model in Talukder and Vincent's study used [4]. With my spreadsheets it was easier to program and get 8 games (roughly 14 days) at a time and predict 4 games (roughly 7 days) ahead (Fig 4). To improve accuracy I removed data windows where players did not play and were not injured because the cause of missing the game was non-injury related.

If the player played in all 8 games, I combined the 8 in-game data rows into an array Xsubset, then added Xsubset to an aggregation array called X. The 4 games of subsequent injury data was added into an array Ysubset, then added to an prediction array called Y. This process was done in every players' spreadsheet, resulting in a final X and Y that contained every players' set of aggregation and prediction windows. Once completed, the X and Y arrays were split into X_train, X_test, Y_train, and Y_test array, where 80% of the X and Y pairs were chosen with a random seed to be in X_train and Y_train, and the other 20% to be in X_test and Y_test. There were far more games where no injury occurred so I had a class imbalance problem. To fix this I reduced my array to have an equal number of injury and non-injury windows. As in the study, the random forest algorithm performed best. I used a B=100 regression trees as a classifier to predict the injury/non-injury status after the aggregation window. "A random forest first creates B bootstrap samples from training data, then fits separate classification trees for each bootstrap sample (finding optimal variable/split-points), to finally make a committee or majority vote on the injury/non-injury status and computing estimated injury probabilities i for each player i based on the average results across the different trees [11,12]."

vi. Real Time Model

The final step was creating an app for real time predictions in the current season. All these data resources update as new games occur, making real time predictions in the current season achievable through cron, a time-based job scheduler, to run a bash script. This script runs the python programs to update all the spreadsheets every night and then runs the machine learning program on those spreadsheets. To get the injury data every night the end date in the URL needs to be advanced to the current date, which can be retrieved and added at runtime by the cron job. The basketball reference website updates its page after every game so the program just needs to make a new request to the database. Once all the data is loaded, I can display any given players predicted probability of injury going into their next game. The transition from using 2013-2014/2014-2015 season data to using real time data was much easier given the web scraping programs only needed a few adjustments. Also, I could apply the 2013-2014/2014-2015 season data as extra training data for the current season while I loaded in new aggregation and predictions windows as they occurred. The code for creating the spreadsheets and model is on GitHub (`https://github.com/jaymessina3/Honors`).

IV. Results

i. Small Scale Investigation

My first results used only the Boston Celtics player's data. This smaller scale example was used to gain experience with the technology and have a proof of concept. I utilized each player's season statistics from SportVu data [3], then filled in a column stating true or false on whether they got injured that season. I was able to get an average accuracy of 80%. This example was only using players with 15 minutes or more playing time and using the top 5 statistics Talukder and Vincent's study used [4]. After I had learned how to use sklearn, panda, and how to scrape the web, I moved to a bigger data set. I obtained all NBA players averages over the course of the whole season and created a column to record if they got injured or not during that season. With the whole NBA, I got an average accuracy of 76%. These results did not use sliding windows as they were taking one data row and predicting one injury row.

ii. Window Size Evaluation

I did not experiment with different sized windows because Talukder and Vincent's study had already done that with different sizes of aggregation and prediction windows. As expected, the larger the windows, the more accurate the predictions. When the aggregation windows are longer, there is more data to make the decision and when the prediction windows are longer predictions can be made with a higher confidence. While longer windows would yield even higher accuracy, they would not be able to be used in practice because it is more valuable to know if a player is going to get injured in a given week than a given month. With a one-week window, a coach could take immediate action and get the player in the training room for a physical evaluation or rest them for an upcoming game. However, with a one-month window, a coach might might sit a player more often when in actuality they could be playing or the player could become paranoid of imminent injury. So the combination of an 8 game aggregation window and 4 game prediction window was the best option.

iii. Large Scale Model and Uncovering Biases

Once the window length was decided, I aggregated the game by game statistics and implemented the sliding windows to make injury predictions. I started off just applying the top 5 statistics from Talukder and Vincent's study as a baseline, but later on added more variables. I had been using accuracy as my measure, but realized that it is actually the AUC that I wanted. The curve is the receiver operating characteristic curve (ROC), which plots both true positive and false positive rate. This curve tells us how good the model is for distinguishing the given classes in terms of the predicted probability [13]. The AUC-ROC curve is one of the most commonly utilized metrics to evaluate the performance of machine learning algorithms, particularly in the cases where we have imbalanced datasets, like the one at hand [13]. The dataset had far more non-injury games than injury games. This data disparity created a class imbalance problem that made the non-injury prediction accuracy 100%, but the injury prediction at only around 10%. When I made the classes of injury and non-injury sets balanced I improved my

AUC up to 0.80.

iv. Improving Model to get Final Result

Once I had that baseline I started adding more variables and improving my injury web scraping. I had been taking out flu and illness, but not rest, infections and ingrown toenails injury reports for missing games. I added 10 more variables, all mentioned in my methods section. I also removed games where players did not play and no injury was noted. This improved my AUC to 0.90.

F	F	F	F	F	F
F	F	F	F	F	F
F	F	F	F	F	F
F	F	F	F	F	F
F	F	F	F	F	F
F	F	F	F	F	F
T	T	T	**T**	T	T
T	T	T	T	T	T
T	T	T	T	T	T

Table 1: Nate Robinson Injury Prediction Array

0.33	0.28	0.36	0.09	0.11	0.17
0.15	0.11	0.13	0.26	0.14	0.13
0.16	0.12	0.17	0.12	0.13	0.30
0.18	0.17	0.16	0.12	0.32	0.22
0.11	0.13	0.16	0.25	0.21	0.19
0.07	0.27	0.19	0.20	0.20	0.18,
0.72	0.99	1.00	**1.00**	1.00	1.00
1.00	1.00	1.00	1.00	1.00	1.00
1.00	1.00	1.00	1.00	1.00	1.00

Table 2: Nate Robinson Predicted Probabilities

Knowing overall AUC helped prove that my model could predict injuries accurately, but the real benefit was seeing individual player results. One player of interest was Nate Robinson in 2014. He played significant minutes in 44 consecutive games, and on the 44th tore his ACL and was out for the rest of the season. My model was able to predict his injury as shown below. The array's first value has game's 1-8 as the aggregation window, and games 9-12 as the prediction window. The second value has game's 2-9 as the aggregation window, and games 10-13 as the prediction window. These windows continue for all games in the season.

The predicted probability is anywhere from 0.09 to 0.36 throughout the season, but at the first occurrence of an injury in the prediction window the predicted probability spikes to 0.72 (Table 2). The T's in Table 1 represent that there is an injury in the players prediction window. For example, the array [healthy, healthy, healthy, injury] and [injury, injury, injury, injury] are represented by T, and the array [healthy, healthy, healthy, healthy] is represented by F. When the first T (bolded in Table 1) occurs, the prediction window array is [healthy, healthy, healthy, injury]. This is not when the injury occurs, but when the player is seen as a player at risk of injury. The injury actually occurs in the second bolded value in Table 1 where the prediction window array is [injury, injury, injury, injury], and the predicted probability is 1.00. Nate Robinson's name would pop up as a player at risk of injury when the first T is seen. This is very useful because it is a couple of days in advance of the actual injury, so the athletic training staff would be able to take precautionary steps to evaluate the player's health and potentially rest them for the next game where the possibility of injury is very high.

V. Conclusion and future work

In this project I gained experience with both web scraping and machine learning. I saw how NBA player's in-game data and playing style could directly correlate to injuries. If organizations know when their players need to rest, they can optimize their playing time throughout the season so they can be healthy for playoff games. This study could be taken even further than the current data provided. There are sensors in the arena that can detect player's movement patterns and speed, but with wearable technology organizations could see players heart rate and hydration levels. With more data points like these I believe we could predict when to rest players throughout an entire game.

VI. Acknowledgements

Thank you to my advisor John Donaldson for meeting with me every week throughout the year and providing mentorship through this honors project. I would also like to thank Adam Eck for helping me improve my machine learning algorithm and find biases in my data.

References

[1] Bullard, K.: Predicting Pitcher Injuries (2016)

[2] Spearman, W.: Physics-Based Modeling of Pass Probabilities in Soccer (2017)

[3] SportsVU. http://stats.nba.com/players/speed-distance/

[4] Talukder, H., Vincent, T.: Preventing in-game injuries for NBA players (2016)

[5] Zillgit, J.: NBA referees to crack down on continuation calls, dangerous closeouts this season (2017) https://www.usatoday.com/story/sports/nba/2017/09/21/nba-referees-crack-down-continuation-calls-dangerous-closeouts-season/691158001/

[6] Boskovic, S.: Top 10 Most Injury-Prone Players in the NBA https://fadeawayworld.com/2017/12/04/top-10-most-injury-prone-players-in-the- nba/ (2017)

[7] Deitsch, J.: Injury Risk in Professional Basketball Players (2006)

[8] McManus, T.: Bradford's ACL: What Are the Odds? https://www.phillymag.com/birds247/2015/05/26/bradfords-acl-what-are-the- odds/ (2015)

[9] Iyiewuare, Peace.: NBAJSONCSV https://github.com/peaceiyi/NBAJSONCSV (2017)

[10] Pro Sports Transaction Archive. Pro Sports Transaction Archive. http://www.prosportstransactions.com

[11] Hastie, T., Tibshirani, R., and Hastie, T.: The Elements of Statistical Learning. Springer New York (2009).

[12] Efron, B. and Tibshirani, R. J.: An Introduction to the Bootstrap. Chapman & Hall/CRC, London (2006).

[13] Classification: ROC Curve and AUC https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc (2019)

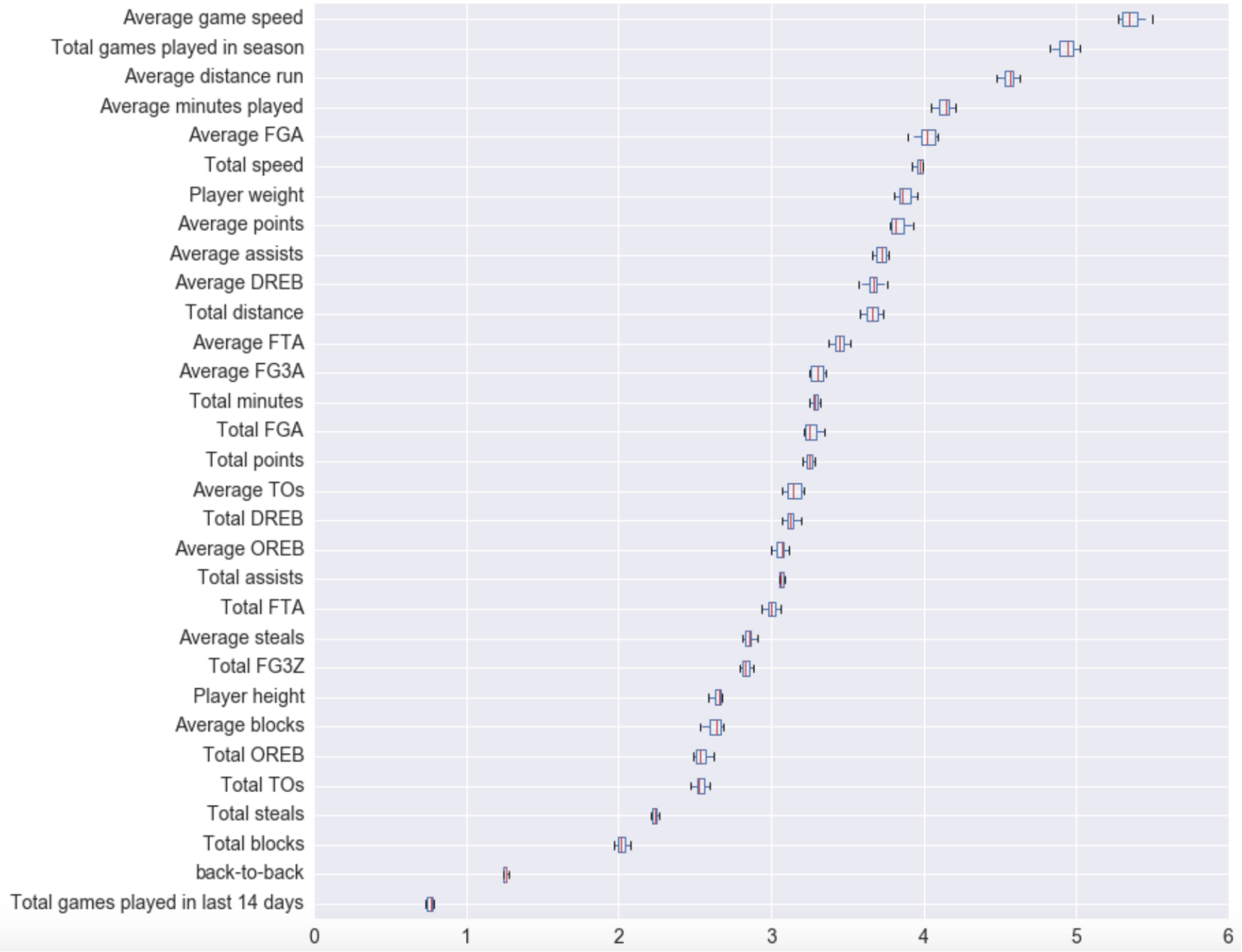


Figure 3: Variable Importance [4]

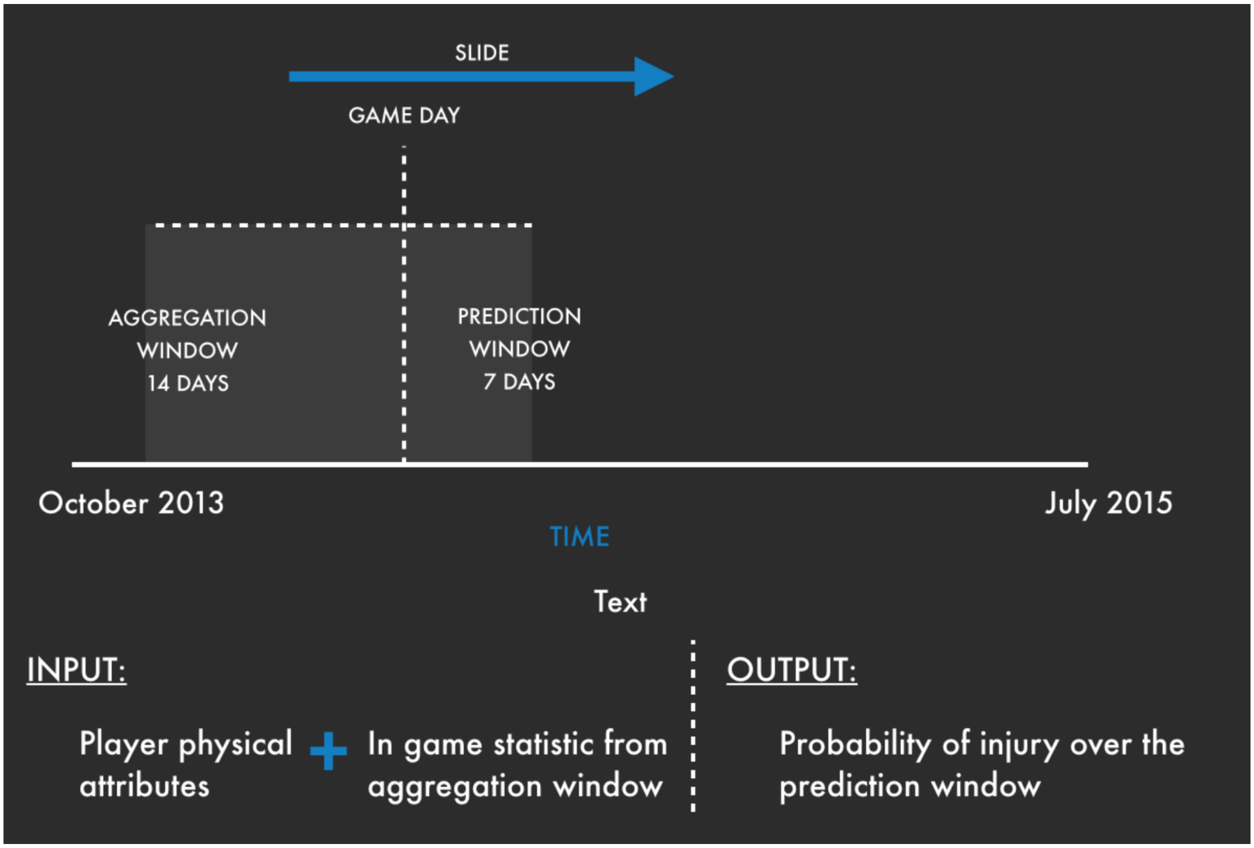
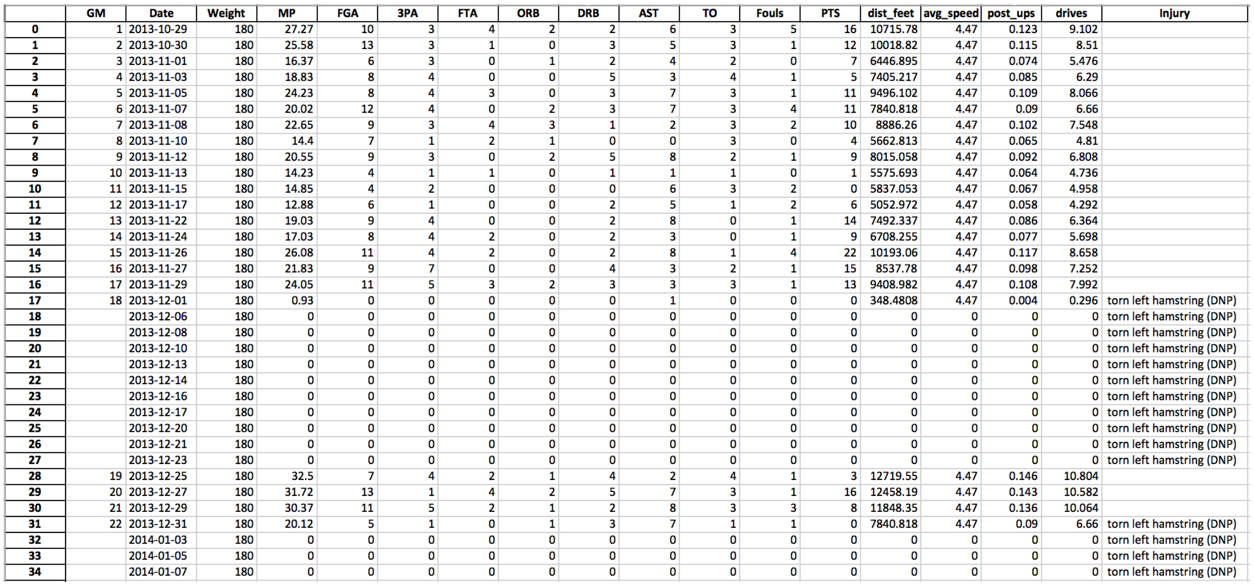

SLIDE
GAME DAY
AGGREGATION WINDOW 14 DAYS
PREDICTION WINDOW 7 DAYS
October 2013
July 2015
TIME
Text
INPUT:
Player physical attributes
+
In game statistic from aggregation window
OUTPUT:
Probability of injury over the prediction window

Figure 4: Sliding Window Illustration [4]



	GM	Date	Weight	MP	FGA	3PA	FTA	ORB	DRB	AST	TO	Fouls	PTS	dist_feet	avg_speed	post_ups	drives	Injury
0	1	2013-10-29	180	27.27	10	3	4	2	2	6	3	5	16	10715.78	4.47	0.123	9.102	
1	2	2013-10-30	180	25.58	13	3	1	0	3	5	3	1	12	10018.82	4.47	0.115	8.51	
2	3	2013-11-01	180	16.37	6	3	0	1	2	4	2	0	7	6446.895	4.47	0.074	5.476	
3	4	2013-11-03	180	18.83	8	4	0	0	5	3	4	1	5	7405.217	4.47	0.085	6.29	
4	5	2013-11-05	180	24.23	8	4	3	0	3	7	3	1	11	9496.102	4.47	0.109	8.066	
5	6	2013-11-07	180	20.02	12	4	0	2	3	7	3	4	11	7840.818	4.47	0.09	6.66	
6	7	2013-11-08	180	22.65	9	3	4	3	1	2	3	2	10	8886.26	4.47	0.102	7.548	
7	8	2013-11-10	180	14.4	7	1	2	1	0	0	3	0	4	5662.813	4.47	0.065	4.81	
8	9	2013-11-12	180	20.55	9	3	0	2	5	8	2	1	9	8015.058	4.47	0.092	6.808	
9	10	2013-11-13	180	14.23	4	1	1	0	1	1	1	0	1	5575.693	4.47	0.064	4.736	
10	11	2013-11-15	180	14.85	4	2	0	0	0	6	3	2	0	5837.053	4.47	0.067	4.958	
11	12	2013-11-17	180	12.88	6	1	0	0	2	5	1	2	6	5052.972	4.47	0.058	4.292	
12	13	2013-11-22	180	19.03	9	4	0	0	2	8	0	1	14	7492.337	4.47	0.086	6.364	
13	14	2013-11-24	180	17.03	8	4	2	0	2	3	0	1	9	6708.255	4.47	0.077	5.698	
14	15	2013-11-26	180	26.08	11	4	2	0	2	8	1	4	22	10193.06	4.47	0.117	8.658	
15	16	2013-11-27	180	21.83	9	7	0	0	4	3	2	1	15	8537.78	4.47	0.098	7.252	
16	17	2013-11-29	180	24.05	11	5	3	2	3	3	3	1	13	9408.982	4.47	0.108	7.992	
17	18	2013-12-01	180	0.93	0	0	0	0	0	1	0	0	0	348.4808	4.47	0.004	0.296	torn left hamstring (DNP)
18		2013-12-06	180	0	0	0	0	0	0	0	0	0	0	0	0	0	0	torn left hamstring (DNP)
19		2013-12-08	180	0	0	0	0	0	0	0	0	0	0	0	0	0	0	torn left hamstring (DNP)
20		2013-12-10	180	0	0	0	0	0	0	0	0	0	0	0	0	0	0	torn left hamstring (DNP)
21		2013-12-13	180	0	0	0	0	0	0	0	0	0	0	0	0	0	0	torn left hamstring (DNP)
22		2013-12-14	180	0	0	0	0	0	0	0	0	0	0	0	0	0	0	torn left hamstring (DNP)
23		2013-12-16	180	0	0	0	0	0	0	0	0	0	0	0	0	0	0	torn left hamstring (DNP)
24		2013-12-17	180	0	0	0	0	0	0	0	0	0	0	0	0	0	0	torn left hamstring (DNP)
25		2013-12-20	180	0	0	0	0	0	0	0	0	0	0	0	0	0	0	torn left hamstring (DNP)
26		2013-12-21	180	0	0	0	0	0	0	0	0	0	0	0	0	0	0	torn left hamstring (DNP)
27		2013-12-23	180	0	0	0	0	0	0	0	0	0	0	0	0	0	0	torn left hamstring (DNP)
28	19	2013-12-25	180	32.5	7	4	2	1	4	2	4	1	3	12719.55	4.47	0.146	10.804	
29	20	2013-12-27	180	31.72	13	1	4	2	5	7	3	1	16	12458.19	4.47	0.143	10.582	
30	21	2013-12-29	180	30.37	11	5	2	1	2	8	3	3	8	11848.35	4.47	0.136	10.064	
31	22	2013-12-31	180	20.12	5	1	0	1	3	7	1	1	0	7840.818	4.47	0.09	6.66	torn left hamstring (DNP)
32		2014-01-03	180	0	0	0	0	0	0	0	0	0	0	0	0	0	0	torn left hamstring (DNP)
33		2014-01-05	180	0	0	0	0	0	0	0	0	0	0	0	0	0	0	torn left hamstring (DNP)
34		2014-01-07	180	0	0	0	0	0	0	0	0	0	0	0	0	0	0	torn left hamstring (DNP)

Figure 5: Spreadsheet Example