COMP9417: Machine Learning Project

z5113817

University of New South Wales — August 1, 2021

Motivation

In the 1984 NBA draft, Sam Bowie was drafted as the number 2 pick to the Trail Blazers. It might be shocking to hear that Bowie was drafted one place *above* the hall of fame superstar Michael Jordan. This was because the Trail Blazers needed a new superstar "big man" to replace the Center they lossed the season before. Bowie had an impressive 76 games with the Trail Blazers until a fracture in his left tibia put him out for the season. Even though Bowie followed the recommened recovery time, the rest of Bowie's career was undermined by the recurring injury. In 10 seasons with the NBA, Bowie only appeared in 511 games.

The question is, even though injuries in sport are seen as an unforeseeable tragedy, can a machine learning model be used to eliminate some of the unpredictability and quantify the likelihood that a player will suffer a major injury in the current season?

The Goal

Create a model that assigns a likelihood that a player will *suffer a major injury* in any given season. Suffering a major injury will be defined a physical injury that leaves a player on the injury list for more than 34 days.

The Data

Datasets

The datasets were scraped from various sources such as prosporttransactions and basketball-refrence. The scrapers were sourced from **elap733**'s repository found here.

The following datasets can be found in the **data/raw** directory:

- **player_stats**: Contains every NBA player's basic statistics for a given season.
- **injury_list**: Contains information on when players were acquired on and relinquished from NBA injury list.
- **missed_games**: Contains information on games which players missed (not necessarily due to injury).
- **all_games_schedule**: The schedule for every NBA team.

The data ranges from 2010 to today.

Cleaning

For the most part, the data contains pretty clean data with no missing entries or gibberish values. The datasets were last scraped in 2019 however, so I had to do some additional scraping to obtain the latest data. Luckily, the scrapers still worked just fine.

For cleaning, all that I had to do was append the latest scraped data with the existing datasets. The cleaned datasets can be viewed at **data/cleaned**. The code used to execute this is stored at **scripts/-clean_data.py**.

Processing Data

The Injury List

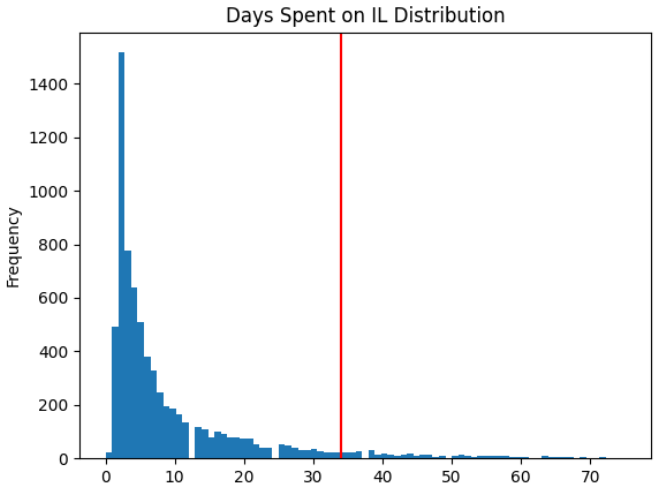
This section explains the reasoning and steps take to produce the **data/processed/physical_injuries_2010_2021.csv** file and what each of the columns mean. The code used to execute this is stored at **scipts/process_injury_list.py**.

The injury list in the NBA isn't quite as the name suggests. Rather than being a list of players who are currently suffering from physical injury, players who miss games for other reasons can be placed on this list. Other reasons include:

- **illness**: 413
- **surgery**: 253
- **COVID-19**: 11
- **personal reasons**: 11

In addition, the list includes players who are *relinquished* from their team (put on the injury list) and players who are *acquired* to their team (removed from the injury list). This means there are essentially 2 entries for each individual injury. We want to move the date of this second entry to its own column so that for each injury we know the dates that player was put on and removed from the injury list.

Earlier I defined a "major injury" as a physical injury that leaves a player on the injury list for more than 34 days. I defined it in this way by looking at the distribution of days spent on the injury list in our dataset.



The red line shows the 80th percentile of this distribution which is at **34** days spent on the injury list [1]. I selected this to be the boundary between a minor and major. Even though this was done somewhat arbitarily (I manually decided to use the 80th percentile), I believe this is a reasonable number since it represents the injuries that left a player on the injury list for just over a month and represents only 20% of all NBA injuries since 2010.

The processed injury list dataset is stored at **data/processed/physical_injuries_2010_2021.csv**. There are **8,552** entries in this dataset and **1,711** are major injuries.

Improving the story

Prior to his career in the NBA, Bowie had developed a stress fracture in his left tibia. Even though he rested and took the recommended amount of time off for this minor injury, this would be the same place that Bowie suffered his first major injury when he joined the NBA.

Now that we have 1,711 examples where a player were acquired on the injury list for longer than 34 days, we want to tell a bigger story with these entries by attaching more data to these injuries. The case of Bowie can inspire us to ask questions such as:

- How was the player performing before this injury?
- What was their average gametime?
- How intense did the player play?
- How intense was their team's schedule?
- Have they suffered an injury prior?

The script **scripts/derive_new_fields.py** attempts to answer some of these questions.

The dataset **data/cleaned/player_stats_2010_2021.csv** contains a general overview of player statistics for a given season. We can perform a *left join* on this dataset with the processed physical injuries dataset so that each injury is now linked to the player's performance during the season that they were injuried.

[1] Some injuries were marked as "player out for season" which means they don't have another entry in the dataset showing their return. For this case, the player was marked down as being on the IL for 100 days. These players are not included in the histogram however do play a role in this percentile calculation

Player	Year	Season	Position	Age	Injury Date	Duration	Notes
Malik Allen	2010	regular	PF	32	2010-10-28	9	placed on IL
Malik Allen	2010	regular	PF	32	2010-11-10	5	placed on IL
Malik Allen	2010	regular	PF	32	2010-11-20	4	placed on IL
Malik Allen	2010	regular	PF	32	2010-11-30	1	placed on IL
Malik Allen	2010	regular	PF	32	2010-12-23	43	placed on IL with sprained left ankle

Table 1: Dataset showing Malik Allen suffering multiple injuries in the 2010 regular season

But what about players who were injured multiple times during the same season? In table 1, we can see that Allen suffered a few injuries in the 2010 season. Each of these rows also contain data about Allen's performance for the 2010 season, which are all identical in their values, since it's all for the 2010 season. Recall that the aim of this model is to predict the likelihood that a player will suffer a major injury in a given season. Feeding Allen's data into a model would confuse the model since it has 4 examples where a player didn't suffer a major injury and 1 example where they did, all with the *exact same* stats for the season.

We want to group these examples together so that the model doesn't have contradicting data, but we also want to keep the information that Allen suffered *minor* injuries prior to his major injury.

Let's create a new field **Recent Minor Injury Count** which for each season for a given player, counts the number of minor injuries they received in that season. A *minor injury* will be defined as an injury that wasn't a major injury.

Let's also create another field **Previous Major Injury Count** which counts the number of *major injuries* that a player has suffered strictly prior to that season. The new dataset is created at **data/processed/aggregated.csv**.

Tokenising

Fields such as the player's position and whether the season is post season or regular are string values. We want to encode these to a numerical value and store their encoding somewhere.

The script **tokenize_data.py** performs these operations and stores the new dataset at **data/processed/-tokenized.csv**.

Normalising and Standardising

This section explains the steps in **normalise.py** which is used to create the dataset at **date/processed/nomarlised.csv**.

Some players play more minutes per game than others. Because of this, some of fields such as *FGA* (Field Goal Attempts per game) will be greatly skewed by the average amount of minutes a player spends in a game. As such, the first step I have taken to normalise is make all of "per game" stats a ratio with the average minutes played [2].

Since the data has a lot of continuous values, it is a good idea to nomralise these between 0 and 1. I don't want fields such as the age of a player to be weighted significantly higher than a player's average offensive rebounds per game, just because the value of a player's age is always much larger.

Also, I will assess various models and I do not know the distribution of the dataset across all of its values. As such, standardising is a good idea.

[2] Note that all normalising and standardising after this point is done *after* training and testing data has been established

Data Selection

I now have a dataset with a lot of meaningful values - almost too many. In the script, **select_data.py** I clean out datapoints that I think will mislead the model (for example, the player's name) or are correlated and such, provide no further information (for example, Field Goals per game with Field Goal Attempts per game). This results in a final processed dataset at **data/processed/final.csv** which contains the data that I will be using to train various models on.

An explanation on all of the columns contained in this dataset and what they each mean can be found in appendix .

Exploratory Analysis

I've been doing a bit of exploring of the data previously to make decisions about deriving new fields for processing. This section will focus on using the final "processed" dataset and looking at what we have.

Model Selection

Model Evaluation

Conclusion

Nice to haves

Would've been nice to incorperate the intensity of the schedule some more. Do some NLP analysis on the Notes of the injury.

Appendix

Final Dataset

List of all columns:

- **Season**: whether the row is from regular (1) or post (0) season