NBA Shot Logs Analysis

Multivariate Analysis

Sergio Llana Pérez and Pau Madrero Pardo

1-. Introduction

Information is a very valuable asset and it often contains hidden knowledge which is not always exploited. After this idea, a lot of companies emerged between the 80’s and the 90’s applying a process called KDD (Knowledge Discovery in Databases) on a wide variety of fields such as business and health.

Sports is a field of obvious application due to the huge amount of data that is generated in every game. The first two real cases of application were developed by IBM and A.C. Milan. The former was a system created in 1996 in order to detect statistical patterns and odd events in NBA games. On the other hand, the latter was a decision support system developed to predict football player injuries in 2002.

1.1-. Data Understanding

In order to perform a practical analysis, we will use a public dataset called "NBA shot logs” obtained from Kaggle. It contains information about over 125 thousand NBA shots, which occurred during 2014-2015 season. The dataset has been obtained by scrapping NBAs official API and it is composed by the following 21 features:

- GAME ID: numeric identifier of the game.
- MATCHUP: variable containing the date of the match and the involved teams.
- LOCATION: whether the game is played home (H) or away (A) regarding the shooter.
- W: whether the game of the shot was eventually won by the shooters team.
- FINAL MARGIN: difference in the score of both teams at the end of the game (from shooters point of view).
- SHOT NUMBER: helps to keep the order of the shots in the same game.
- PERIOD: period of the game when the shot occurred. Note that periods 5 to 7 are extra-time periods because of a draw.
- GAME CLOCK: remaining time in the period when the shot was done.
- SHOT CLOCK: remaining possession time when the shot was done (out of 24 seconds).
- DRIBBLES: count of dribbles that the shooter did before executing the shot.
- TOUCH TIME: count of seconds that the shooter held the ball before shooting.
- SHOT DIST: distance (in feet) from shooters position regarding to the basket.
- PTS TYPE: whether the shot was a 2-pointer or a 3-pointer.
- SHOT RESULT: whether the shot was made or missed.
- CLOSEST DEFENDER: closest defenders name.
- CLOSEST DEDENDER PLAYER ID: closest defender’s numeric identifier.

- CLOSE DEF DIST: distance (in feet) from shooters position to the closest defender.
- FGM: whether the show was successful or not (stands for ”Field Goals Made”).
- PTS: amount of points added to shooter teams score after the shot.
- player name: shooters name.
- player id: shooters numeric identifier.

2-. Data Pre-Processing

Once the raw data is read from the CSV file, we started by splitting the “MATCHUP” column, which contains the date of the game and the competing teams in the format “MMM DD, YYYY - TEAM1 vs. TEAM2”. Although in some observations, the separator between teams is “@” instead of “vs.”.

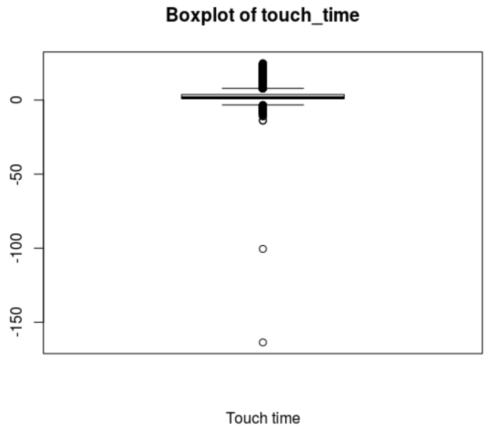
Secondly, in order to have the “GAME_CLOCK” in an understandable format for machine learning methods, we have implemented a function that turns a “MM:SS” format into the number of remaining seconds when the shot was done. In addition, the distances were transformed from feet to meters.

Finally, we adjust the type of some columns such as logical variables (into numeric) and ordered factors. The last transformation is to create a new data frame which filters out those unnecessary columns (e.g. “FGM” or “PTS”) and renames the existing features to have homogeneous column names.

2.1-. Data Cleansing

In this second phase, we have explored the data frame in order to look for possible wrong values, missing values and outliers.

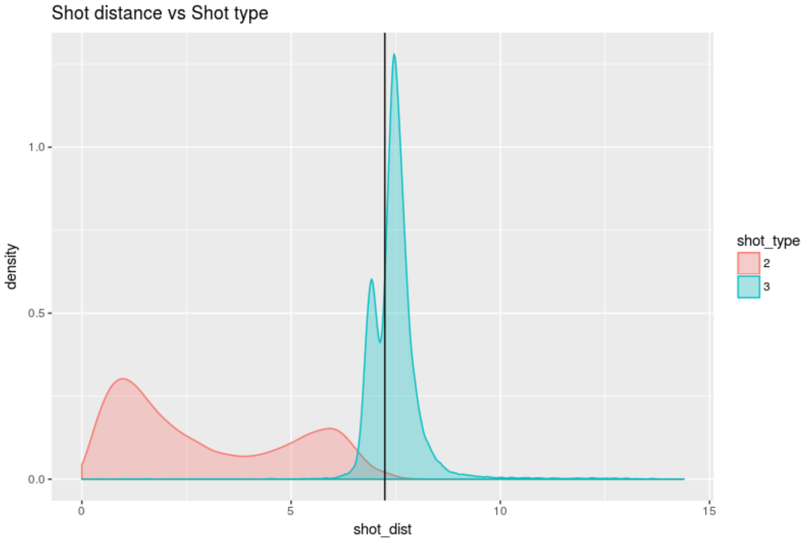
Regarding the “touch_time”, we know that their values should be between 0 and 24, as a possession in basketball cannot exceed that value. However, we can see in the following boxplot that there are values outside the range.



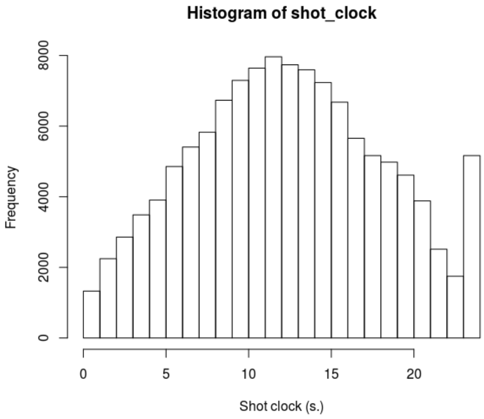
For handling them, there are two different case:

- As the values that exceed the 24 seconds limit are pretty close to it (e.g. 24.3 seconds), we have decided to round them to 24.
- On the other hand, those negative values have been marked as NAs, which will be handled later.

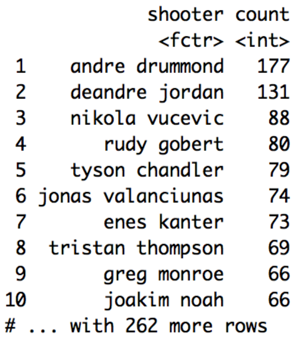
Then, we have taken a look to the interaction between the shooter’s distance and the type of shot (whether it is marked as a three or a two pointer). The following plot, created using the package “ggplot2”, shows the density of both 2 and 3-pointer shots in different colours. We can appreciate how they are not separated, but we do not know the exact meaning of this distance in the data source, so we cannot make a decision.



Concerning the “shot_clock” column, we have detected a spike in the histogram for values close to 24:

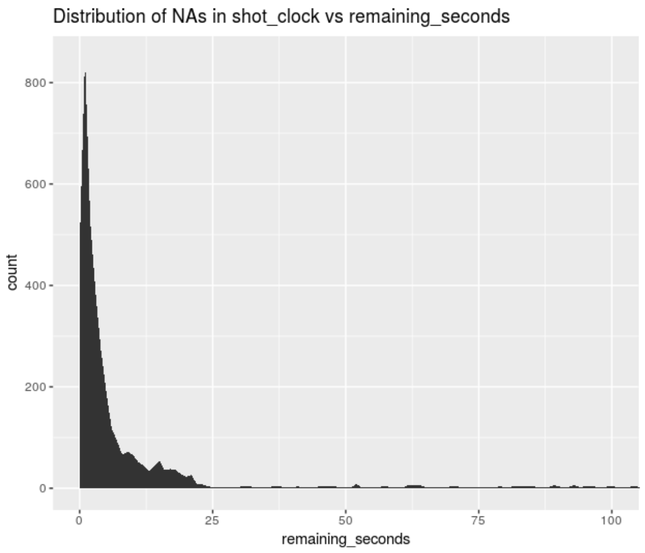


After doing some research online about the dataset, a basketball fan would realize these are all big men playing close to the hoop. Domain knowledge can lead to the conclusion that these are tip-ins or put-backs where a player near the hoop collects an offensive rebound after a teammate missed a shot.



```
                 shooter  count
                  <fctr>  <int>
1    andre drummond    177
2    deandre jordan    131
3     nikola vucevic     88
4         rudy gobert     80
5     tyson chandler     79
6 jonas valanciunas     74
7         enes kanter     73
8   tristan thompson     69
9         greg monroe     66
10        joakim noah     66
# ... with 262 more rows
```

Regarding the 5567 NAs in "shot_clock", most of them occur when "remaining_seconds" are lower than "shot_clock" and therefore the "shot_clock" is turned off, because it has no importance anymore. The following plot shows their density when compared to the "remaining_seconds".



2.2-. Feature Extraction

Finally, in order to enrich the dataset, we will create three new variables based on the existing ones. They could be very useful for future data visualization analysis or as supplementary variables in a Principal Components Analysis.

Firstly, we will start by creating a categorical feature based on the defender's distance to the shooter. It will contain four levels:

defender_dist <= 2 ft.	→	Tightly Contested
2 ft. < defender_dist <= 3.5 ft.	→	Contested
3.5 ft. < defender_dist <= 6 ft.	→	Open
defender_dist > 6 ft.	→	Wide Open

Secondly, based on the type of shots that NBA defines in https://stats.nba.com, we will create another categorical feature which depends on the number of “dribbles” and the distance of the shot.

No dribbles AND shot_dist > 4 ft.	→	Catch & Shoot
No dribbles AND shot_dist <= 4ft.	→	Cut
dribbles AND shot_dist < 4 ft.	→	Drive
dribbles > 4	→	ISO / Post up
dribbles > 20	→	Long ISO
dribbles <= 1 AND shot_type = 3	→	Spot Up Three
…	→	Other

And finally, we will create a logical variable for clutch situations. We say that a shot was done in a clutch situation when they are executed during the last minute of the potential last period (i.e. 4th period or extra time) for those games that ended up with a margin of points lower than 5 points.

2.3-. Data Enrichment

As last part of the data pre-processing phase, we have found data provided by ESPN of the same season that includes both the salary and position of every player. We thought that it would be a good idea to use the salary as a measure of the player’s quality, so it could be a key feature for predicting the outcome of future shots.

As this dataset was only available in http://www.espn.com/nba/salaries/_/year/2015/, we had to use the package “rvest” in order to scrap the information from the website.

Then, when we tried to merge player’s names between both datasets we realised that names didn’t match correctly because of typing mistakes, abbreviations etc. We solve this challenge in two steps:

- First, we used “dplyr” to do a full join to figure out which players were not correctly mapped to the salaries dataset.
- Then, for those without a salary assigned, we used the “stringdist” package to calculate the most similar name in the other dataset using the OSA algorithm. In addition, we set a threshold in order to avoid matching wrong names.

All players were correctly fixed, except for two, which were changed manually.

2.4-. Handling Missing Values

shot_clock column

From the 5567 NAs in this column, we have proved before that most of them (3153) occur when the column “remaining_seconds” is lower than 24 seconds, which is the duration of a possession. The value of these observations will be imputed by a random number from a uniform distribution in the range of [0, remaining_seconds].

The remaining 1647 observations, which are a less than a 1.5% of the total observations, will be removed.

touch_time column

As we do not know the reason of the 285 values that this column contains, we will use the function “catdes” contained in “FactoMineR” with the purpose of revealing if certain values of the rest of columns are related with these missing values.

First of all, we need to create a boolean variable based on whether the “touch_time” is negative or not and then, we apply the function:

- Regarding the quantitative variables, we see in the results that shots with missing “touch_time” have a significantly lower and strangely exact ratio of success (0.3) compared to the overall mean (0.45), but most importantly, the mean of the number of dribbles before the shot is astonishingly lower than the overall mean (0.007 vs 2.05).
- On the other hand, for categorical variables, we see that the amount of shots of the categories "Cut" and "Catch&Shoot" is much higher than the average, as well as the positions "PF" and "C" and the difficulty "Tightly Contested". However, the shot categories "Drive", "ISO/Post up" and "Other" appear fewer times than average as well as the position "PG" and the difficulty "Open"

After these insights, one possible interpretation could be that shots with missing values in “touch_time” correspond mostly to quick plays where probably this time wasn't measured correctly due to the celerity (swiftness) of the game.

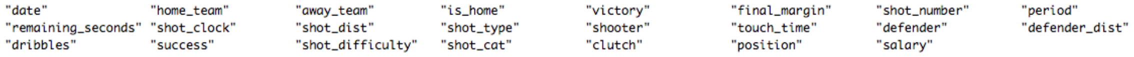
A naïve approach would be to impute the NAs by low random variables following the previous conclusions. However, we have decided to impute them using KNN (with K = 1) and supervise that the new values are generally lower than the average.

The mean is lower in the imputed values, so our hypothesis holds:

Mean of the imputed values: 2.1253
Mean of the original values: 2.8114

2.5-. Final dataset

Once all the previous transformations were applied, we ended up with a dataset of more than 110263 rows and the following 23 features:



```
"date"              "home_team"   "away_team"        "is_home"    "victory"  "final_margin" "shot_number" "period"
"remaining_seconds" "shot_clock"  "shot_dist"        "shot_type"  "shooter"  "touch_time"   "defender"    "defender_dist"
"dribbles"          "success"     "shot_difficulty"  "shot_cat"   "clutch"   "position"     "salary"
```

3-. Principal Components Analysis

4-. Validation Protocol

5-. Prediction Model

6-. Conclusions