NBA Data Analysis

Alex Fung, Patrick Osborne, Tony Lee, Viswesh Krishnamurthy

14/03/2020

ABSTRACT

The use of various predictive metrics in sports has been occurring as long as human beings have watched each other compete. Initially this started out as simple gut feeling or a subjective assessment of the competitors – the bets usually go to the bigger fighter! In more recent times, humanity has refined its predictive power with the advent of statistics and logical decision making (famously put to use in Major League Baseball, as shown in the film Moneyball). With the advent of the information age and the possibility for large-scale data analytics and machine learning, the National Basketball Association has decided to puruse this analysis to better understand player matchups (defender vs offender) and to assess & optimize shooter performance.

BUSINESS UNDERSTANDING

Detailed statistics are already available to the NBA as these have been tracked for many years, supporting classic statistical decision making. The goal is to run both unsupervised and supervised machine learning algorithms on the statistical data available. In technical terms, we aim to deliver predictive metrics for threat/benefit level at an individual player level, as well as an interactive application that identifies the likelihood of a shot landing from a specific offender shooting from a specific position on the court, against a specific defender located a certain distance away. This will allow coaches to run limited scenarios in the predictive model, to inform both their practice routines and to assist in making strategic decisions during live games.

As an example, consider the matchup of Lebron James on offence and Serge Ibaka on defence. Let's assume that Lebron typically tries to shoot from top of the key, and is being defended by Serge Ibaka, 5 feet away. The model takes these discrete inputs and outputs a real-world percentage success of 10.5% (example). If the average shooting success rate is 30%, we can identify this as a bad shot, and encourage Lebron to pass in these situations.

DATA UNDERSTANDING

We begin with understanding each feature available in the data. The available data set is data of all shots attempted at NBA games between 2014 and 2015. For each shot attempted, the most important outcome of that attempt, whether the shot was made or missed is available. This is seen in 2 columns SHOT_RESULTS and FGM. FGM stands for "Field Goal Made". In support of this outcome, there are a number of other data points to be seen, like the player who attempted the shot and who defended the sho, how far away was the defender, how far away from the basket was the shot attempted, was the match at home or away etc. With that data understanding, let's look at the "head" of the data.

GAME_ID	MATCHUP	LOCATION	W	FINAL_MARGIN	SHOT_NUMBER	PERIOD	GAME_CLOCK
21400899	MAR 04, 2015 - CHA @ BKN	A	W	24	1	1	1:09
21400899	MAR 04, 2015 - CHA @ BKN	A	W	24	2	1	0:14
21400899	MAR 04, 2015 - CHA @ BKN	A	W	24	3	1	0:00
21400899	MAR 04, 2015 - CHA @ BKN	A	W	24	4	2	11:47
21400899	MAR 04, 2015 - CHA @ BKN	A	W	24	5	2	10:34
21400899	MAR 04, 2015 - CHA @ BKN	A	W	24	6	2	8:15

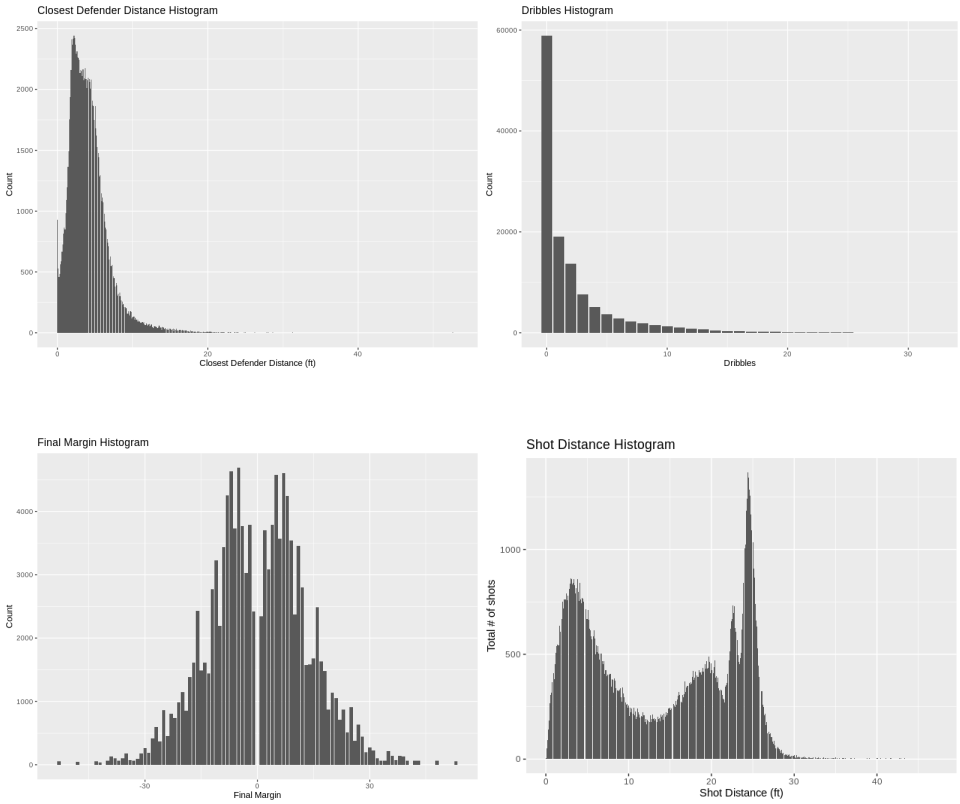
SHOT_CLOCK	DRIBBLES	TOUCH_TIME	SHOT_DIST	PTS_TYPE	SHOT_RESULT	CLOSEST_DEFENDER
10.80	2	1.90	7.70	2	made	Anderson, Alan
3.40	0	0.80	28.20	3	missed	Bogdanovic, Bojan
	3	2.70	10.10	2	missed	Bogdanovic, Bojan
10.30	2	1.90	17.20	2	missed	Brown, Markel
10.90	2	2.70	3.70	2	missed	Young, Thaddeus
9.10	2	4.40	18.40	2	missed	Williams, Deron

Table 1: Data Dictionary - NBA Data

Feature	Feature.Description
GAME_ID	A unique ID for each game
MATCHUP	Shows the date of the match and the teams the match is between
LOCATION	A - Away, H - Home. Shows the location of the match with respect to the first team in the match up column
W	Win or Loss. W means win and L means Loss, with respect to the first team in the match up column
FINAL_MARGIN	Points difference between the teams at the end of the game
SHOT_NUMBER	To be read in conjunction with the PERIOD column. Indicates the shot number in a given game period
PERIOD	Indicates the game period
GAME_CLOCK	Time elapsed since the period commenced. This datset shows the time at which the shot was attempted. Max 12 minutes per period
SHOT_CLOCK	The length of time for a given shot in seconds. Max - 24 seconds is a rule
DRIBBLES	The number of times the ball was dribbled before the shot was attempted
TOUCH_TIME	The length of time a player touched the ball
SHOT_DIST	The distance from which a shot was attempted. Distance in feet
PTS_TYPE	Points awarded if a shot was made. 2 pointer or 3 pointer shots
SHOT_RESULT	Indicates whether the shot was made or missed
CLOSEST_DEFENDER	Shows the name of the player that was the closest defender
CLOSEST_DEFENDER_PLAYER_ID	Unique ID of the closest defender
CLOSE_DEF_DIST	Distance of the closest defender in feet
FGM	An abbreviation for FIELD GOALS MADE. A proxy for SHOT_RESULT, 0 indicates missed shot and 1 indicates shot made
PTS	Points awarded for shots made
player_name	Name of the player who attempted the shot
player_id	Unique ID of the player who attempted the shot

Plots

We attempt to further understand the data using the following plots. A histogram of the "closest defender distance" shows that a majority of the shots were defended from within 5 feet of the player attempting the shot and it is safe to say that more than 90% of the shots were defended from within 10 feet. The "dribbles count" shows that most of the shots were attempted soon after getting the ball and that more than 80% of the shots were attempted within 3 dribbles. "Final Margin" histogram shows that most matches were won or lost within a 15 point margin. The "Shot distance" histogram shows that most of the shots were attempted from "top of the key" and followed by 2 to 4 feet range from the basket



DATA PREPARATION

SHOT_CLOCK

Looking at the data, some of the NA values need to be dealt with. The "SHOT_CLOCK" column has some NA values and the assumption is that the SHOT_CLOCK was equal to the GAME_CLOCK and therefore it may not be recorded. For such cases, the GAME_CLOCK is assumed to be equal to SHOT_CLOCK.

```
cleanData <- initialData
gameClock <- as.vector(second(fast_strptime(cleanData$GAME_CLOCK, "%M:%S"))) +
  as.vector(minute(fast_strptime(cleanData$GAME_CLOCK, "%M:%S"))) * 60
shotClock <- is.na(initialData$SHOT_CLOCK)
for(i in 1:length(gameClock)){
  if(shotClock[i] & gameClock[i] < 25){
    cleanData$SHOT_CLOCK[i] <- gameClock[i]
  }
}
```

Names

To further handle player names in this exercise, all names are standardized to read as "First Name" followed by "Last Name". A custom function was written to achieve this result.

```
nameformatreverse <- function(s) {
  fname <- str_extract(s, "^\\w+")
  lname <- str_extract(s, "\\w+$")
  s <- paste(lname, fname, sep = ", ")
```

```
}
```

All Shooter & Defender names are then put through the function to standardize names

```
shooterName <- cleanNoNAData$player_name
shooterName <- toupper(shooterName)
shooterName <- nameformatreverse(shooterName)
```

```
cleanNoNAData$player_name <- shooterName
cleanNoNAData$CLOSEST_DEFENDER <- toupper(cleanNoNAData$CLOSEST_DEFENDER)
cleanNoNAData$CLOSEST_DEFENDER <- gsub("[.]", "", cleanNoNAData$CLOSEST_DEFENDER)
```

Game Clock

It makes best sense to have the GAME_CLOCK expressed in seconds.

```
cleanNoNASecondsClockData <- cleanNoNAData
cleanNoNASecondsClockData$GAME_CLOCK <-
  as.vector(second(fast_strptime(cleanNoNAData$GAME_CLOCK, "%M:%S"))) +
  as.vector(minute(fast_strptime(cleanNoNAData$GAME_CLOCK, "%M:%S"))) * 60
```

Touch time

Any row that has TOUCH_TIME less than 0.1 seconds is not right and hence are omitted

```
cleanNoNASecondsClockData <- cleanNoNASecondsClockData[cleanNoNASecondsClockData$TOUCH_TIME > 0, ]
```

MODELLING

K-Means Clustering

The Elbow method is a popular, non computation intensive process of determining the most optimal number of clusters for a dataset by looking at a dropoff of variance. The other methods, eg, Bayesian Inference which we ran is more computation intensive, and produced optimal clusters that didnt agree with the visual Elbow method.Therefore, after plotting and analyzing a few different features against each other and highlighting the clusters by colouring the datapoints, we find that 3 clusters is likely the best compromise for the important features, namely, shot distance and closest defender distance.

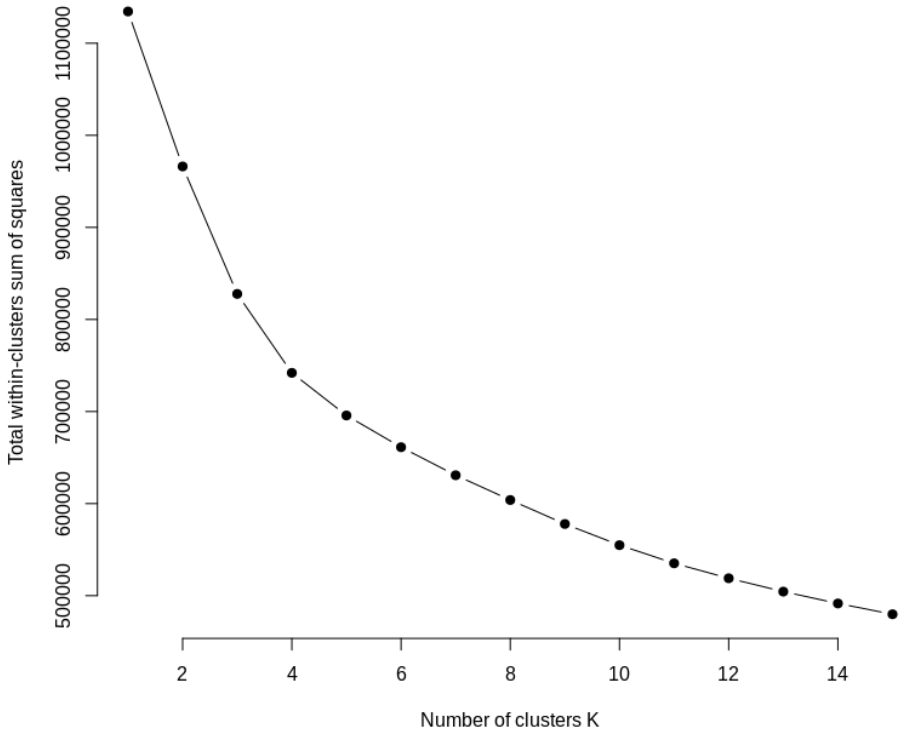
To perform clustering, only the numeric columns from the data are selected.

```
kdataunscaled <- cleanNoNASecondsClockData[, c("SHOT_NUMBER", "PERIOD",
                                               "GAME_CLOCK", "SHOT_CLOCK", "DRIBBLES",
                                               "TOUCH_TIME", "SHOT_DIST", "CLOSE_DEF_DIST")]
kdata <- scale(kdataunscaled)
```

We use the 'Elbow' method to determine the ideal number of clusters

```
set.seed(123)
# Compute and plot wss for k = 1 to k = 15
k.max <- 10
wss <- sapply(1:k.max, function(k){kmeans(kdata, k, nstart=50,iter.max = 15 )$tot.withinss})
wss
plot(1:k.max, wss,
     type="b", pch = 19, frame = FALSE,
     xlab="Number of clusters K",
     ylab="Total within-clusters sum of squares")
```

The "number of clusters" vs "sum of squares" plot helps us identify the 'Elbow' and decide on the right number of clusters. From the plot, we will try creating clusters with k = 2,3 & 4

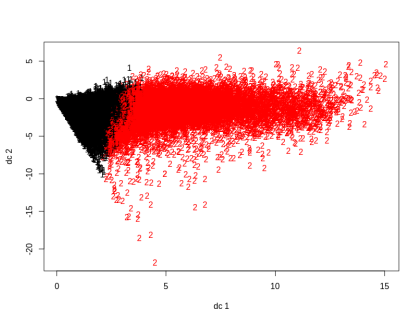


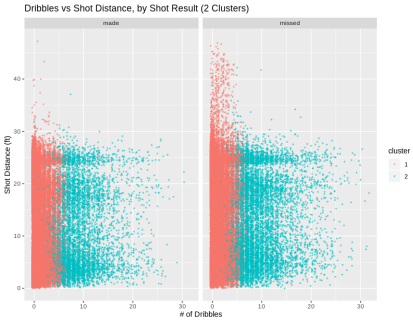
Bayesian Inference Criterion for k means to validate choice from Elbow Method

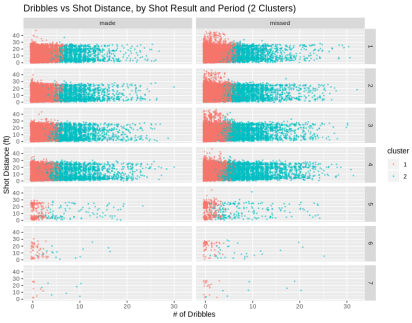
```
d_clust <- Mclust(as.matrix(kdata), G=1:10,
                  modelNames = mclust.options("emModelNames"))
d_clust$BIC
plot(d_clust)

# Let us apply kmeans for k=2 clusters
kmm.2 <- kmeans(kdata, 2, nstart = 50, iter.max = 15)
# Let us apply kmeans for k=3 clusters
kmm.3 <- kmeans(kdata, 3, nstart = 50, iter.max = 15)
# Let us apply kmeans for k=3 clusters
kmm.4 <- kmeans(kdata, 4, nstart = 50, iter.max = 15)
# We keep number of iter.max=15 to ensure the algorithm converges and nstart=50 to
# Ensure that atleat 50 random sets are choosen
kmm.2
kmm.3
kmm.4

# Plot the clusters
clusplot(kdataunscaled, kmm.3$cluster, color=TRUE, shade=TRUE, labels=2, lines=0)

# Centroid Plot against 1st 2 discriminant functions
plotcluster(kdataunscaled, kmm.2$cluster)
```

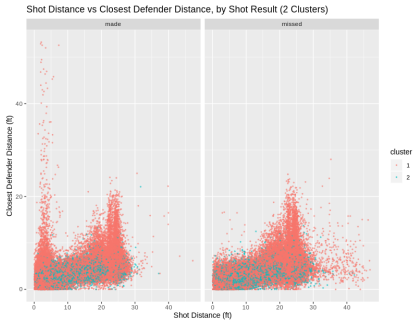
```
plotcluster(kdataunscaled, kmm.3$cluster)
plotcluster(kdataunscaled, kmm.4$cluster)
```

K2 plots

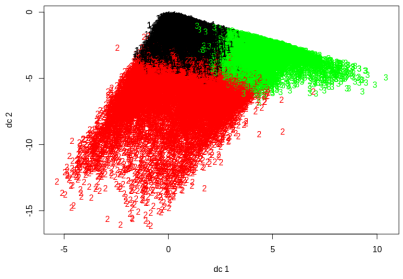


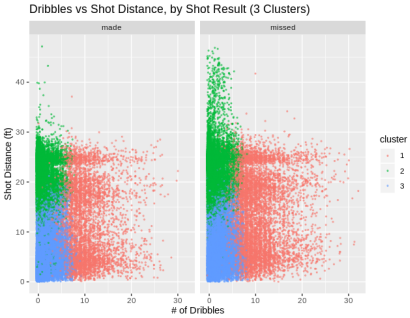


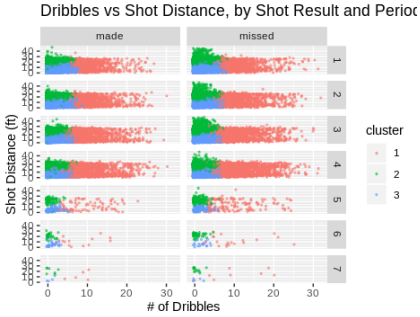


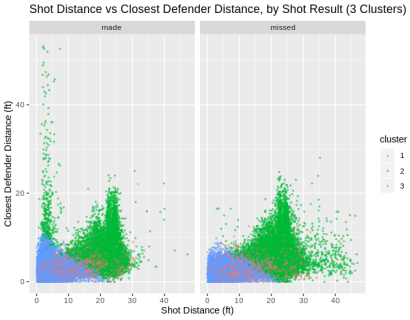


k3 plots


dc 1
dc 2

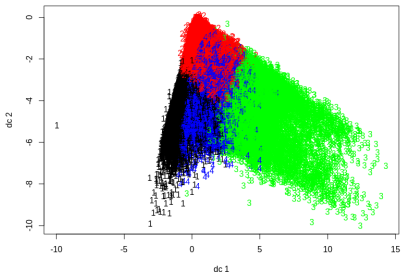

Dribbles vs Shot Distance, by Shot Result (3 Clusters)
made
missed
Shot Distance (ft)
# of Dribbles
cluster
1
2
3

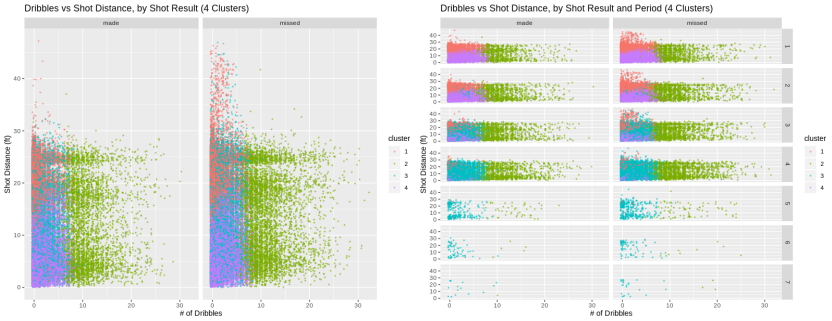

Dribbles vs Shot Distance, by Shot Result and Period
made
missed
Shot Distance (ft)
# of Dribbles
cluster
1
2
3

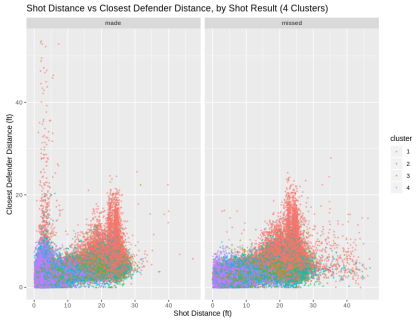

Shot Distance vs Closest Defender Distance, by Shot Result (3 Clusters)
made
missed
Closest Defender Distance (ft)
Shot Distance (ft)
cluster
1
2
3


Shot Distance vs Closest Defender Distance, by Shot Result and Period (3
made
missed
Closest Defender Distance
Shot Distance (ft)
cluster
1
2
3

k4 plots







Decision Trees