Kobe Bryant Make His Next Shot: Linear Discriminant Analysis and Logistic Regression

Paul Adams
Reannan McDaniel
Jeff Nguyen
Southern Methodist University

21 November 2019

Abstract

This project investigates the correlation between multiple potential explanatory variables and Kobe Bryant's ability to make a shot while playing for the NBA team Los Angeles Lakers using data gathered from `1996-2015`.

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

In preparation for analysis, derrived data was neeeded to simplify overly complicated variables for the model. The over-complication is with respect to the availability of other predictor variables.

```
df[which(df$action_type == "Alley Oop Dunk Shot"),"action_type"] = "short"
df[which(df$action_type == "Cutting Layup Shot"),"action_type"] = "short"
df[which(df$action_type == "Driving Dunk Shot"),"action_type"] = "short"
df[which(df$action_type == "Driving Finger Roll Layup Shot"),"action_type"] = "short"
df[which(df$action_type == "Driving Layup Shot"),"action_type"] = "short"
df[which(df$action_type == "Driving Reverse Layup Shot"),"action_type"] = "short"
df[which(df$action_type == "Driving Slam Dunk Shot"),"action_type"] = "short"
df[which(df$action_type == "Dunk Shot"),"action_type"] = "short"
df[which(df$action_type == "Finger Roll Layup Shot"),"action_type"] = "short"
df[which(df$action_type == "Follow Up Dunk Shot"),"action_type"] = "short"
df[which(df$action_type == "Layup Shot"),"action_type"] = "short"
df[which(df$action_type == "Putback Dunk Shot"),"action_type"] = "short"
df[which(df$action_type == "Putback Layup Shot"),"action_type"] = "short"
df[which(df$action_type == "Putback Slam Dunk Shot"),"action_type"] = "short"
df[which(df$action_type == "Reverse Dunk Shot"),"action_type"] = "short"
df[which(df$action_type == "Reverse Layup Shot"),"action_type"] = "short"
df[which(df$action_type == "Reverse Slam Dunk Shot"),"action_type"] = "short"
df[which(df$action_type == "Running Dunk Shot"),"action_type"] = "short"
df[which(df$action_type == "Running Finger Roll Layup Shot"),"action_type"] = "short"
df[which(df$action_type == "Running Layup Shot"),"action_type"] = "short"
df[which(df$action_type == "Running Reverse Layup Shot"),"action_type"] = "short"
df[which(df$action_type == "Running Slam Dunk Shot"),"action_type"] = "short"
df[which(df$action_type == "Running Tip Shot"),"action_type"] = "short"
df[which(df$action_type == "Slam Dunk Shot"),"action_type"] = "short"
df[which(df$action_type == "Tip Layup Shot"),"action_type"] = "short"
df[which(df$action_type == "Tip Shot"),"action_type"] = "short"

df$action_type <- ifelse(df$action_type=="short", "short", "long")

df[which(df$combined_shot_type == "Jump Shot"),"combined_shot_type"] = "short"
```

```r
df[which(df$combined_shot_type == "Dunk"),"combined_shot_type"] = "short"
df[which(df$combined_shot_type == "Layup"),"combined_shot_type"] = "short"
df[which(df$combined_shot_type == "Tip Shot"),"combined_shot_type"] = "short"
df[which(df$combined_shot_type == "Hook Shot"),"combined_shot_type"] = "short"
df[which(df$combined_shot_type == "Bank Shot"),"combined_shot_type"] = "short"

df$combined_shot_type <- ifelse(df$combined_shot_type=="short", "short", "long")
```

Remove one-level factors. These will never change so are not useful to the model; including can cause issues with model sensitivity since linear trajectories will be down-weighted. Therefore, their significance will be lessened by the constant state of the additional parameters. While this is may not be significant, it is not condusive to model quality.

```r
#
badNews <- "Sorry, but your math is off. Please try again..."

tryCatch(
    {
    df <- df %>% subset(select=-c("team_id", "team_name"))
    },
    error = function(e)
    {
      badNews
    }
)
```

```
## [1] "Sorry, but your math is off. Please try again..."
```

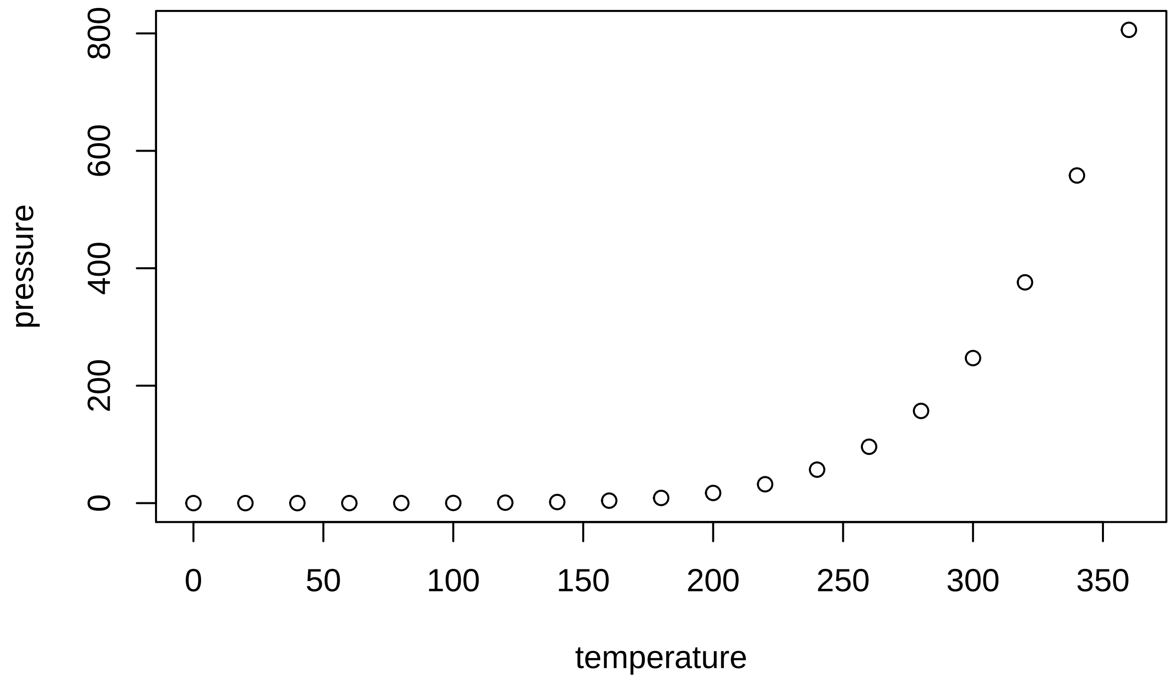
```r
# create numeric dataframe for correlation plot
df.numeric <- df %>% subset(select=-c(team_id, team_name, season, shot_zone_area, shot_zone_basic, shot_

# Convert all remaining integers to numeric and characters to factors with levels:
df <- df %>% mutate_if(is.integer, as.numeric) %>% mutate_if(is.character, as.factor) %>% data.frame()
```

```r
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.