Determining Successful Career Longevity for NBA Rookies

Spencer Yee

Table of Contents

# Abstract

Identifying NBA rookies that have high value depends on the ability to estimate a rookie’s career longevity and plays a huge role when it comes to NBA franchise management and future plans. This study examines a dataset of 1340 recent NBA rookies. Logistic regression models were fit to the binary response for successful or unsuccessful career longevity. A modified variable for total games played and a variable for points per game are found to be important determinants of such. The methods used and final prediction function will be useful in conducting administrative basketball decisions.

# Introduction

Administration and management for teams within The National Basketball Association (NBA) pay very close attention to the development of their rookies with the hope that each player will have a successful (and thus very lengthy) NBA career. There are numerous factors that go towards analyzing NBA rookies and their rookie-season performance statistics not limited to; points per game, field goal percentage and total games played. However, these single-season statistical variables can potentially be used to predict whether a rookie will have a career longevity of at least 5 years, or experience a shortened career of at most 4 years. This is of high interest for team management as it can provide insight that can ultimately help teams to gauge both the value and potential of their assets (players). In addition, gaining a sense of player value can help to make difficult decisions regarding roster composition, extension contracts and trades. To put this more simply, an NBA rookie who is likely capable of lasting 5+ years in the league holds much higher value than one who is likely incapable. This is because longer careers are indicative that a player is at least useful and talented enough to be worthy of a roster spot past their initial rookie contract. This NBA data can help to distinguish between the two and make predictions of rookie career longevity.

This study examines a statistics dataset of NBA rookies drafted into the league between the 1985 and 2013 seasons who played minimum 10 games in their first season. It contains the necessary features and content in order to conduct proper analysis and produce a structurally sound logistic regression model. The more prominent per game statistics such as points, assists and rebounds are typically associated with player skill level and are likely to have an effect in determining 5-year career longevity. Variables concerning player usage are also likely to have an effect as it is indicative of a rookie who is at least performing well enough to earn sufficient playing time in their first year. As this generally bodes well for future development and career longevity, we will ensure that a player usage variable is included in our potential models to make proper analysis. All other variables will also be investigated to determine if their effects are significant enough to be included in the model. The organization for the rest of the report is as follows. Section 2 will contain some data characteristics and exploration techniques for choosing significant variables. Section 3 will delve into a discussion of model selection, methods and analysis. Finally, concluding remarks and reccomendations will be found in section 4, along with a brief overview of the final prediction function.

# Data Characteristics and Exploration

NBA = read.csv("nba\_stats.csv")  
library(magrittr)  
library(dplyr)  
library(forcats)  
library(ggplot2)  
library(car)  
any(is.na(NBA))  
sum(is.na(NBA))  
nba = NBA[rowSums(is.na(NBA)) == 0,]  
any(is.na(nba))

The data are cross-sectional and was retrieved from the open online data community [*Data.World, Inc.*](#References). It is based on first-season performance statistics taken on yearly NBA rookies entering the league between 1985 to 2013 who played at minumum 10 games. It was found that 11 observations contained missing values. These observtions were removed from the data as it is important for our study that the data does not induce bias or reduce statistical power. The outcome of interest is the successful career indicator which is a binary categorical variable where value “1” represents a true condition for a career greater than 5 years and “0” as false for less than 5 years. We have 19 other variables describing total rookie season statistics per game. There is an additional categorical variable that identifies each data record by player name, however this will not be used. There are 1340 total observations. [*Table 1.1*](#Data%20Characteristics%20and%20Exploration) shows the variables available and their definitions.

**Table 1.1**

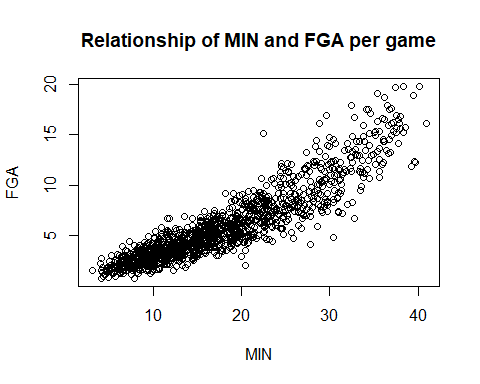
|  |  |  |
| --- | --- | --- |
| Item | Variable | Definition |
| 1 | Name | Player name |
| 2 | GP | Total games played |
| 3 | MIN | Minutes per game |
| 4 | PTS | Points per game |
| 5 | FGM | Field goals made per game |
| 6 | FGA | Field goals attempted per game |
| 7 | FG. | Field goal percentage |
| 8 | X3PM | 3-point field goals made per game |
| 9 | X3PA | 3-point field goals attempted per game |
| 10 | X3P. | 3-point field goal percentage |
| 11 | FTM | Free throws made per game |
| 12 | FTA | Free throws attempted per game |
| 13 | FT. | Free throw percentage |
| 14 | OREB | Offensive rebounds per game |
| 15 | DREB | Defensive rebounds per game |
| 16 | REB | Total rebounds per game |
| 17 | AST | Assists per game |
| 18 | STL | Steals per game |
| 19 | BLK | Blocks per game |
| 20 | TOV | Turnovers per game |
| 21 | Target\_5Yrs | Successful Career indicator (1 = 5+ years, 0 = <5 years) |

### Data Characteristics and Redundancy

All potential predictor variables are rather easy to comprehend and need no further explanation. However, we can make logical assumptions of which variables may be redundant and uneccessary to consider for further exploration for our model. Take for example the three variables pertaining to field goal production (FG., FGM, FGA). Mathematically, field goal percentage is simply the amount made divided by the amount attempted. However consider two players, one with 90% field goal percentage who made 9 out of 10 shots, and another with 40% who made 180 out of 450. The scale and weight to which these percentages are calculated are clearly not equal which can result in error and reduce the statistical power of our model. The significance of this variable is questionable as it is directly related to FGM and FGA. Furthermore, we can logically deduce that field goal attempts per game is also not very likely to be a significant predictor as it is not a measure of basketball skill competency, but merely a continuous count variable. While higher or lower values may be able to provide us an idea of player usage (more attempts means more minutes), we already have a minutes per game variable included. We can gain a visual of this in [*Plot 1.1*](###Data%20Characteristics%20and%20Redundancy) below:

**Plot 1.1**

plot(nba$MIN, nba$FGA, xlab = 'MIN', ylab = 'FGA', main = 'Relationship of MIN and FGA per game')



We can see that there is a very strong positive correlation between minutes per game and FGA per game, providing us stronger evidence to omit the latter. In order to eliminate the risk of multicollinearity, this variable will be removed from consideration. Field goals made per game is evidently the strongest potential predictor of these three similar variables. The continous variable is both an indicator of performance and is relevant in scale. We can apply this same reasoning to the other variable trios for 3-point field goals and free throws. This effectively eliminates a total of 6 variables, however we will consider and test the significance of interactions between percentage and made shots variables during model construction.

Shifting focus to the trio of variables regarding rebounds, it is reasonably obvious that total rebounds (REB) is calculated from the combined values of offensive (OREB) and defensive (DREB) rebounds. Hence, it is logical to solely explore total rebounds and ignore the two distinguishing variables. While it may be argued that offensive rebounds are more celebrated and thus indicative of greater skill, we can conclude that this somewhat sophisticated measure of skill is not significantly relevant to the outcome variable of this study. We may risk multicollinearity in attempting such and in not doing so, we also reduce the pool of potential predictors.

The total games played (GP) variable is related to player usage and has a limited value range of 10 to 82 games (remember this dataset consists of rookies playing minimum 10 games and that the maximum games in a single regular season is 82). Given this limit, it will be ideal to transform these numerics into a factor variable with multiple sublevels, each containing a smaller reference range of games played values. This will allow us to draw more coherant conclusions and to better gauge the effect of this predictor on our response variable during model construction. This process is called ‘binning’. Our original games played variable will no longer be considered as it will sensibly correlate too closely with the newly created variable which we will name ‘GP.lev’. The structure of this transformation can be found in Appendix [*A2.*](#Appendix)

nba <- nba %>%  
 mutate(GP.lev = case\_when(  
 GP <= 20 ~ "(0,20]", #20 or less games  
 GP <= 30 ~ "(20,30]", #20-30 games  
 GP <= 40 ~ "(30,40]", #30-40 games  
 GP <= 50 ~ "(40,50]", #40-50 games  
 GP <= 60 ~ "(50,60]", #50-60 games  
 GP <= 70 ~ "(60,70]", #60-70 games  
 GP <= 82 ~ "(70,82]")) #70-82 games  
nba$GP.lev <- factor(nba$GP.lev,  
 levels = c("(0,20]","(20,30]",  
 "(30,40]","(40,50]","(50,60]",  
 "(60,70]","(70,82]"))

Excluding player name, we are left with 6 other variables: minutes per game (MIN), points per game (PTS), assists per game (AST), steals per game (STL), blocks per game (BLK) and turnovers per game (TOV). For logistic regression, normal distribution is not required but we must pay close attention multicollinearity and independence. As these are game statistics, it is likely these values will increase with player usage, especially correlating with the per game usage variable measuring minutes. This suggests that games played (GP) would be a more desireable usage predictor as it logically would not correlate as closely with the others. We find evidence for this in [*Table 1.2*](###Data%20Characteristics%20and%20Redundancy) which displays high correlation coefficients between minutes per game and other variables, indicating high correlation. In [*Table 1.3*](###Data%20Characteristics%20and%20Redundancy) variable correlation between games played is less severe.

**Table 1.2**

writeLines("Minutes Correlation")

## Minutes Correlation

rbind("MIN\_PTS" = cor(nba$MIN, nba$PTS),   
 "MIN\_FGM" = cor(nba$MIN, nba$FGM),  
 "MIN\_X3PM" = cor(nba$MIN, nba$X3PM),  
 "MIN\_FTM" = cor(nba$MIN, nba$FTM),  
 "MIN\_REB" = cor(nba$MIN, nba$REB),  
 "MIN\_AST" = cor(nba$MIN, nba$AST),  
 "MIN\_STL" = cor(nba$MIN, nba$STL),  
 "MIN\_BLK" = cor(nba$MIN, nba$BLK),  
 "MIN\_TOV" = cor(nba$MIN, nba$TOV))

## [,1]  
## MIN\_PTS 0.9117464  
## MIN\_FGM 0.9028637  
## MIN\_X3PM 0.3894737  
## MIN\_FTM 0.7910888  
## MIN\_REB 0.7108532  
## MIN\_AST 0.6291470  
## MIN\_STL 0.7570502  
## MIN\_BLK 0.4010105  
## MIN\_TOV 0.8264309

**Table 1.3**

writeLines("Games Correlation")

## Games Correlation

rbind("GP\_PTS" = cor(nba$GP, nba$PTS),  
 "GP\_FGM" = cor(nba$GP, nba$FGM),  
 "GP\_X3PM" = cor(nba$GP, nba$X3PM),  
 "GP\_FTM" = cor(nba$GP, nba$FTM),  
 "GP\_REB" = cor(nba$GP, nba$REB),  
 "GP\_AST" = cor(nba$GP, nba$AST),  
 "GP\_STL" = cor(nba$GP, nba$STL),  
 "GP\_BLK" = cor(nba$GP, nba$BLK),  
 "GP\_TOV" = cor(nba$GP, nba$TOV))

## [,1]  
## GP\_PTS 0.5390694  
## GP\_FGM 0.5430001  
## GP\_X3PM 0.1081947  
## GP\_FTM 0.4831846  
## GP\_REB 0.4603902  
## GP\_AST 0.3743112  
## GP\_STL 0.4527257  
## GP\_BLK 0.2766998  
## GP\_TOV 0.5186933

Including the minutes per game variable in any model would induce the risk of multicollinearity should we also include other variables. As this can provide limitations during model construction, we will disregard minutes as a potential predictor and test the significance of games played as the sole variable accounting for player usage.

### Bootstrap Estimate for True Population Mean

As our target variable is binary, the rate of successful players (5+ years) for this sample can easily be calculated from the total counts for each value. The total success (“1”) and failure (“0”) values are 826 and 503 respectively, which we can use to calculate that the mean rate for this sample is 0.621. While this can be helpful, it only tells us of this sample specifically. We can instead utilize the bootstrap method to generate a non-parametric estimate of the true mean career success rate for the entire population of NBA rookies. This technique essentially takes the average of all mean success rates across a chosen number of random samples. We chose to take 10000 samples with replacement. The result is found in [*Table 1.4*](###Data%20Characteristics%20and%20Redundancy).

**Table 1.4**

set.seed(754)  
N <- 10000  
targ.five.rate <- numeric(N)  
i <- 1  
for(i in 1:N) {  
 l.sample <- sample(nba$TARGET\_5Yrs, nrow(nba), replace = TRUE)  
 targ.five.rate[i] <- mean(l.sample)  
 i <- i + 1  
}  
m <- mean(targ.five.rate)  
q <- quantile(targ.five.rate, probs = c(0.025, 0.975))  
round(c(q[1], "Mean" = m, q[2]), 3)

## 2.5% Mean 97.5%   
## 0.595 0.622 0.648

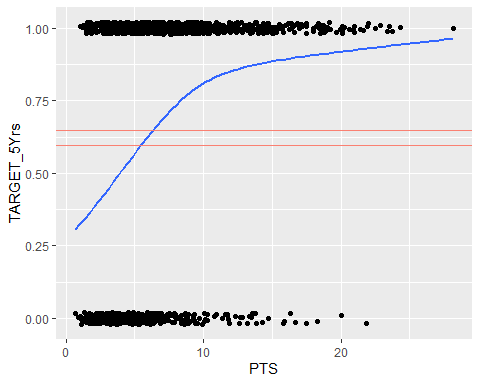
These numbers indicate that we can be 95% confident that the true population career success rate for all NBA rookies will fall between 0.595 and 0.648. We can use this when estimating the skill and fit of a model. In addition, we can use it to determine whether a variable is useful for our model by visualizing any deviance from this interval when comparing to the response variable

### Data Exploration

After concluding earlier in this section that 10 variables are irrelevant, we can conduct exploration for the remaining potential predictors in order to determine whether any display association to the response variable. We will do so by producing individual plots of each against the target variable and including a line of fit (blue). The bootstrap interval (red) will also be displayed in the plot in order to efficiently gauge variation in mean career longevity caused by the independent variable. We would expect a viable significant predictor to show variation whereas an insignificant predictor will not. By this idea, only variables found to be significant will be utilized during model construction and testing. [*Plot 1.2*](###Data%20Exploration) shows the explained plot for points per game against the reponse.

**Plot 1.2**

ggplot(nba[-nba$PTS,]) +  
 aes(x = PTS, y = TARGET\_5Yrs) +  
 geom\_jitter(height = 0.02) +  
 geom\_smooth(se = FALSE) +   
 geom\_hline(yintercept = q, col = "salmon")



We can see from [*Plot 1.2*](###Data%20Exploration) that the line of fit is indicative of variation. A flatter line would be representative of no variation. This particular plot shows an increase in likelihood of having a 5+ year career as points per game also increases. Thus, this variable is deemed significant. This method was conducted for all variables and it was determined that all were indeed significant. These are points per game (PTS), field goals made per game (FGM), three point field goals made per game (X3PM), free throws made per game (FTM), rebounds per game (REB), assists per game (AST), steals per game (STL), blocks per game (BLK) and turnovers per game (TOV). While this technique can be applied to such numerical statistics, it will not work for our transformed factor variable of games played (GP.lev). Instead, we will produce [*Table 1.5*](###Data%20Exploration) to show variation between the sublevels.

**Table 1.5**

tbl <- rbind(tapply(nba$TARGET\_5Yrs, nba$GP.lev, sum),  
 tapply(1-nba$TARGET\_5Yrs, nba$GP.lev, sum),  
 tapply(nba$TARGET\_5Yrs, nba$GP.lev, mean))  
dimnames(tbl)[[1]] <- c("5+", "<5", "%5+")  
tbl

## (0,20] (20,30] (30,40] (40,50] (50,60] (60,70]  
## 5+ 3.0000000 13.0000000 52.0000000 96.00 109.00000 136.0000000  
## <5 15.0000000 35.0000000 93.0000000 104.00 98.00000 70.0000000  
## %5+ 0.1666667 0.2708333 0.3586207 0.48 0.52657 0.6601942  
## (70,82]  
## 5+ 417.0000000  
## <5 88.0000000  
## %5+ 0.8257426

writeLines("The 5+ career success rates are not consistent between the levels. This variable is also significant.")

## The 5+ career success rates are not consistent between the levels. This variable is also significant.

The next section will discuss the utilization of these variables during model building, and ultimately a single model selection.

# Model Selection & Interpretation

Section 2 determined that there is relevant association between career longevity and rookie season statistics variables. It also described how some of these variables could be transformed to provide better predictors for a logistic regression model. This model is intended to be included in a function that can be useful in predicting whether we should expect an NBA rookie to have a successful career of 5+ years or not. This section summarizes the methods used in selecting the best model and will interpret the ???nal prediction function and its results in context. Finally, this section will touch base on the utilization and bene???ts of this function from an analytic basketball management standpoint.

### Model Construction

Preliminarily, there were fourteen potential models that were constructed, each consisting of different potentially significant components, but each fitted with maximum likelihood. This was mainly based on performing the analysis of variance (anova) technique on the first potential model which contained all feasible variables determined in section 2 as predictors. The output can be found in [*Figure 1.1*](#####A5.%20Preliminary%20Model%20Compositions%20-%20R%20Input). From analysis of variance we are specifically seeking out the variables that have larger deviance. It is evident that points per game (PTS), free throws made per game (FTM), rebounds per game (REB), blocks per game (BLK) and turnovers per game (TOV) induce largest variation. It is important to note that this is not suggestive that this specific combination of predictors will produce the best model, however 1 preliminary model does contain such composition. The remaining 12 were constructed keeping this analysis in mind as well as favoring simplicity over complexity. The composition for these models (labeled m1-m14) are found in Appendix [*Figure 1.2*](#####A5.%20Preliminary%20Model%20Compositions%20-%20R%20Input).

m1 <- glm(TARGET\_5Yrs ~ GP.lev + PTS + FGM + FTM  
 + X3PM + REB + AST + STL + BLK + TOV,  
 data = nba,  
 family = binomial(link = "logit"))  
  
m2 <- glm(TARGET\_5Yrs ~ GP.lev + PTS,  
 data = nba,  
 family = binomial(link = "logit"))  
  
m3 <- glm(TARGET\_5Yrs ~ GP.lev + PTS + AST + REB,  
 data = nba,  
 family = binomial(link = "logit"))  
  
m4 <- glm(TARGET\_5Yrs ~ GP.lev + PTS + REB,  
 data = nba,  
 family = binomial(link = "logit"))  
  
m5 <- glm(TARGET\_5Yrs ~ PTS + REB,  
 data = nba,  
 family = binomial(link = "logit"))  
  
m6 <- glm(TARGET\_5Yrs ~ GP.lev + FGM,  
 data = nba,  
 family = binomial(link = "logit"))  
  
m7 <- glm(TARGET\_5Yrs ~ GP.lev + FTM,  
 data = nba,  
 family = binomial(link = "logit"))  
  
m8 <- glm(TARGET\_5Yrs ~ GP.lev + STL,  
 data = nba,  
 family = binomial(link = "logit"))  
  
m9 <- glm(TARGET\_5Yrs ~ GP.lev + BLK,  
 data = nba,  
 family = binomial(link = "logit"))  
  
m10 <- glm(TARGET\_5Yrs ~ GP.lev + FGM + FTM + X3PM,  
 data = nba,  
 family = binomial(link = "logit"))  
  
m11 <- glm(TARGET\_5Yrs ~ GP.lev + FGM + FTM + X3PM + TOV,  
 data = nba,  
 family = binomial(link = "logit"))  
  
m12 <- glm(TARGET\_5Yrs ~ GP.lev + PTS + FTM + REB + BLK + TOV  
 + REB + AST + STL + BLK + TOV,  
 data = nba,  
 family = binomial(link = "logit"))  
  
m13 <- glm(TARGET\_5Yrs ~ GP.lev + FTM + REB + BLK + TOV  
 + REB + AST + BLK + TOV,  
 data = nba,  
 family = binomial(link = "logit"))  
  
m14 <- glm(TARGET\_5Yrs ~ GP.lev + PTS + FTM + TOV,  
 data = nba,  
 family = binomial(link = "logit"))

### Model Testing

The Hosmer-Lemeshow Test of Goodness Fit was selected as a viable method of selecting the best possible model. This test will break the data into a chosen number of groups (10) and separately apply each model to sort the data in order of ascending prediction probabilities. It will then compare the expected values of our target variable to the actual values for each group, and records the average of all mean differences for each group to produce the test statistic. The test determines the number of successful players (5+ years or y = 1) by averaging each model’s prediction probabilities for each group, and multiplying it to the total observations in the respective group. The number of expected unsuccessful players (<5 years or y = 0) is simply the difference. Generally a low test stat and high p-value is desired.

In addition to the HL Test, we will also produce classification metrics to further interpret model performance. Such metrics include precision, accuracy, sensitivity and specificity. It should be noted that such classification metrics are producible only if a probability threshold is set in order for the model to classify/predict each observation as having 5+ year longevity or not. A player that has prediction probability above the threshold will be predicted to have value “1” and “0” otherwise in relation to our response variable. A threshold of 0.55 was ultimately chosen as it maximized the overall performance accuracy of all preliminary models. [*Table 1.6*](###Hosmer-Lemeshow%20Test%20of%20Goodness%20Fit) displays the results of the HL Test and [*Table 1.7*](###Hosmer-Lemeshow%20Test%20of%20Goodness%20Fit) shows the classification metrics for each of the 14 models below.

**Table 1.6**

HL <- function(a, e, g = 10) {  
 y <- a  
 yhat <- e  
 qq <- quantile(yhat, probs = seq(0, 1, 1/g))  
 cutyhat <- cut(yhat, breaks = qq, include.lowest = TRUE)  
 observed <- xtabs(cbind(y0 = 1 - y, y1 = y) ~ cutyhat)  
 expected <- xtabs(cbind(yhat0 = 1 - yhat, yhat1 = yhat) ~ cutyhat)  
 C.hat <- sum((observed - expected)^2/expected)  
 p.val <- 1 - pchisq(C.hat, g - 2)  
 ans <- c("HL Stat." = C.hat,  
 "P-Value" = p.val)  
 return(ans)  
}  
  
rbind("m1" = HL(nba$TARGET\_5Yrs, predict(m1, type = "response")),  
 "m2" = HL(nba$TARGET\_5Yrs, predict(m2, type = "response")),  
 "m3" = HL(nba$TARGET\_5Yrs, predict(m3, type = "response")),  
 "m4" = HL(nba$TARGET\_5Yrs, predict(m4, type = "response")),  
 "m5" = HL(nba$TARGET\_5Yrs, predict(m5, type = "response")),  
 "m6" = HL(nba$TARGET\_5Yrs, predict(m6, type = "response")),  
 "m7" = HL(nba$TARGET\_5Yrs, predict(m7, type = "response")),  
 "m8" = HL(nba$TARGET\_5Yrs, predict(m8, type = "response")),  
 "m9" = HL(nba$TARGET\_5Yrs, predict(m9, type = "response")),  
 "m10" = HL(nba$TARGET\_5Yrs, predict(m10, type = "response")),  
 "m11" = HL(nba$TARGET\_5Yrs, predict(m11, type = "response")),  
 "m12" = HL(nba$TARGET\_5Yrs, predict(m12, type = "response")),  
 "m13" = HL(nba$TARGET\_5Yrs, predict(m13, type = "response")),  
 "m14" = HL(nba$TARGET\_5Yrs, predict(m14, type = "response")))

## HL Stat. P-Value  
## m1 6.2550741 0.6186835  
## m2 0.5264908 0.9998378  
## m3 8.6101400 0.3762430  
## m4 9.5868953 0.2952251  
## m5 6.4479865 0.5971826  
## m6 3.4426231 0.9035968  
## m7 6.6761507 0.5719420  
## m8 4.3940033 0.8199417  
## m9 5.2263439 0.7331329  
## m10 9.7773505 0.2810021  
## m11 7.9525962 0.4381144  
## m12 5.3508321 0.7195045  
## m13 4.7638215 0.7824952  
## m14 9.6775318 0.2883922

**Table 1.7**

cm.metrics <- function(cm) {  
 acc <- sum(diag(cm))/sum(cm)  
 pre <- cm[1,1]/sum(cm[1,])  
 sen <- cm[1,1]/sum(cm[,1])  
 spe <- cm[2,2]/sum(cm[,2])  
   
 ans <- c("Accuracy" = acc,  
 "Precision" = pre,  
 "Sensitivity" = sen,  
 "Specificity" = spe)  
 return(ans)  
}  
  
M <- matrix(NA, nrow = 14, ncol = 4)  
l <- list(m1, m2, m3, m4, m5, m6, m7, m8, m9, m10, m11, m12, m13, m14)  
i <- 1  
for(f in l){  
 p <- predict(f, type = "response")  
 PC <- ifelse(p > 0.55, 1, 0)  
 TC <- nba$TARGET\_5Yrs  
 cm <- table(factor(PC, levels = 1:0),  
 factor(TC, levels = 1:0))  
 M[i,] <- cm.metrics(cm)  
 i <- i + 1  
}  
dimnames(M) <- list(paste("m", 1:14, sep = ""),  
 c("Accuracy", "Precision", "Sensitivity", "Specificity"))  
round(M \* 100, 2)

## Accuracy Precision Sensitivity Specificity  
## m1 70.73 77.76 74.09 65.21  
## m2 68.77 76.25 72.28 63.02  
## m3 70.35 77.41 73.85 64.61  
## m4 70.43 77.44 73.97 64.61  
## m5 67.72 74.84 72.40 60.04  
## m6 69.38 76.76 72.76 63.82  
## m7 68.92 76.44 72.28 63.42  
## m8 68.02 76.14 70.70 63.62  
## m9 69.07 76.57 72.40 63.62  
## m10 69.53 77.02 72.64 64.41  
## m11 68.85 76.48 72.03 63.62  
## m12 70.50 77.47 74.09 64.61  
## m13 70.43 77.44 73.97 64.61  
## m14 68.55 76.02 72.15 62.62

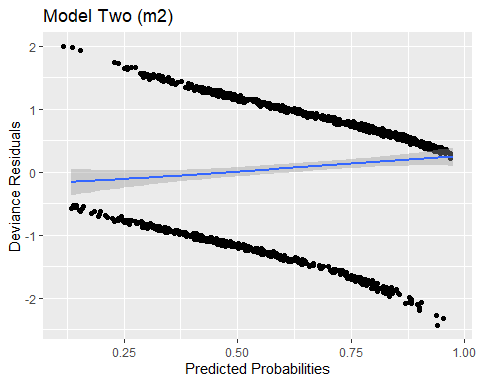
### Model Selection and Interpretation

After evaluating these results, we selected model two (m2) as the best possible model to move forward with. This is because it produces a noticeably better HL test statistic and although it doesn’t perform the best in terms of performance accuracy, we are willing to look past these minimal differences in favor of the test results. We can attempt to visually justify our choice by fitting [*Plot 1.3*](###Model%20Selection%20and%20Interpretation) of the model’s predicted probabilities against its deviance residuals. Again, we are simply hoping to observe a line of fit that is as horizontal as possible. Some noise (shaded gray region) will be added to see results more clearly.

**Plot 1.3**

p <- predict(m2, type = "response")  
r <- resid(m2, type = "deviance")  
ggplot(data = data.frame(x = p, y = r)) +  
 aes(x = x, y = y) +  
 geom\_jitter(height = 0.02, width = 0.02) +  
 geom\_smooth() +  
 labs(title = "Model Two (m2)", x = "Predicted Probabilities", y = "Deviance Residuals")

## `geom\_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'



writeLines("The line is quite flat indicating favorable performance.")

## The line is quite flat indicating favorable performance.

With our selection justified, we can move on to interpretation of our model. To reiterate, the systematic components include the binned games played (GP.lev) and points per game (PPG) variables. Model’s Two’s summary statistics and exponentiated coefficients can be viewed in Appendix [*Figure 1.3*](#Appendix) and [*Figure 1.4*](#Appendix) respectively.

To brie???y describe the functionality of this model, the respective coe???cients for points per game and each level for games played represent the e???ect of those variables on the odds ratio for our dependent variable. However, if we exponentiate each coefficient we can gain a clearer interpretation of effect. For example, the new coefficient for points per game is 1.113 which tells us that a one unit increase in points per game results in a 11.3% increase in the estimated odds of having a successful career of 5+ years. By binning the games played variable, we can also see the immense scale to which games played affects career longevity. This model supports the idea of high player usage being indicative of a successful career. The ???nalized odds ratio produced by this model ultimately determines the probability of a rookie lasting 5 or more years in the league. However as we are more interested in this model ability to generate a de???nitive output, we can once again set a probability threshold which will essentially de???ne players with odds ratio above the threshold as inevitably successful (Target\_5Yrs = 1) and those below it as not (Target\_5Yrs = 0). We will choose the same threshold used for our metrics testing of 0.55 (55%).

### Model Performance

The K-fold cross validation method will be utilized to further test this model’s performance against random training samples. This will allow us to determine if we have over???t our data and to assess the applicability of our model to similar independent sample datasets. We will select k = 10 to test the model against 10 random training sets (folds) formed from the original data, and compare the average of each fold’s mean classification metrics to the model’s metrics for the entire dataset. This will be our last and final test justifying the integrity and strength of our model. Test results can be viewed in [*Table 1.8*](###Model%20Performance)

**Table 1.8**

set.seed(754)  
nba$fold <- sample(c(rep(0, 133), rep(1, 133), rep(2, 133),  
 rep(3, 133), rep(4, 133), rep(5, 133),  
 rep(6, 133), rep(7, 133), rep(8, 133), rep(9, 132)),  
 nrow(nba),  
 replace = FALSE)  
  
set.seed(754)  
F <- matrix(NA, nrow = 10, ncol = 5) # for storing our metrics for each fold  
dimnames(F)[[2]] <- c("fold", "accuracy", "precision", "sensitivity", "specificity")  
i <- 1  
for(fld in 0:9){  
 fit <- glm(TARGET\_5Yrs ~ GP.lev + PTS,  
 data = nba,  
 subset = fold != fld, # do not use one fold  
 family = binomial(link = "logit"))  
   
 p <- predict(fit,  
 newdata = subset(nba, subset = fold == fld), # on the fold not used  
 type = "response")  
 PC <- ifelse(p > 0.55, 1, 0)  
 TC <- nba$TARGET\_5Yrs[nba$fold == fld] # True condition on the fold predicted  
 cm <- table(factor(PC, levels = 1:0),  
 factor(TC, levels = 1:0))  
 F[i,] <- c(fld, cm.metrics(cm))  
 i <- i + 1  
}  
(fld.means <- apply(F, 2, mean)[2:5]) # calculate means for each column

## accuracy precision sensitivity specificity   
## 0.6885224 0.7654019 0.7206100 0.6389107

whole.sample <- M[2,]  
tbl <- rbind("Whole Sample" = whole.sample,  
 "Cross-Validated" = fld.means,  
 "Difference" = whole.sample - fld.means)  
dimnames(tbl)[[2]] <- c("Accuracy", "Precision", "Sensitivity", "Specificity")  
round(tbl,4)

## Accuracy Precision Sensitivity Specificity  
## Whole Sample 0.6877 0.7625 0.7228 0.6302  
## Cross-Validated 0.6885 0.7654 0.7206 0.6389  
## Difference -0.0008 -0.0029 0.0022 -0.0087

From [*Table 1.8*](###Model%20Performance) we can see that the model performs very well and we should feel very comfortable in our model of choice. When performing such cross validation, we would ideally like the difference between our entire sample and cross-validated data to be as close to zero as possible. The results are representative of such, and it should be noted that the differences will never be exactly zero as our model learned from the original dataset while the cross-validated data is essentially a new sample. This method was chosen over test/training paritioning because it granted the ability to simultaneously allow the validation of model parameters while maintaining the flexibility of using the entirety of the sample for training. Whereas with a 20/80 split of the data, it would prevent us from training our model with 20% of the data. This can make a difference and utilizing k-fold validation further maximized our model’s strength.

### Prediction Function

We can now create a predictive function called “longevity” which will grant the ability to read in any NBA rookie dataset of sufficient size and variable similarity, and predict whether rookies will have a successful career longevity of 5 or more years. We have crafted a test sample dataset of 40 NBA rookies from the 2018 season and applied it to concretely test functionality. The results can be found in Appendix \*[A9.](#Appendix) along with the function’s configuration.

This function can be of great use for both the ownership and management of NBA teams in a number of ways. It can provide statistical insight on the true value of their players. It can help to make cardinal decisions in relation to team roster and salary designation. It can help management to account for unknowns when setting goals and subsequently help franchises gain a clearer picture of their future. The benefits of statistical prediction is limitless, and while it is slowly gaining popularity in the modern world, it can absolutely be useful in NBA basketball.

# Summary and Concluding Remarks

It is obvious that all NBA rookies are unique and perform differently over the course of their rookie seasons. However it is certainly possible to establish significant determinants of which rookies will experience successful careers beyond 4 years. The best-fitted logistic model concluded that successful career longevity can be predicted by the points per game variable, and also the total games played variable after being split into a specific range of values. The categorically transformed games played variable allowed for the model to consider player usage. It also led to the conclusion that higher volume in a rookie’s playing time (player usage) during their first season is highly indicative of continued development and an eventual successful career. At the same time, the continuous points per game variable allowed for the model to account for rookie talent, as points per game is a universal measure of basketball competency. This particular model construction was also motivated by statistical evidence that these variables better fit the variation in response variable data and that a less complex model would provide more accurate predictions.

This study was based on 1340 NBA rookies who played minimum 10 games in their first season. It contained sufficient variables relating to first-season rookie statistics that may prove to influence career longevity, and allowed us to build and test numerous models in order to determine the best for predicting a successful NBA career.

The findings of our analysis is based purely on NBA rookies between 1985 and 2013. The prediction function created in this report may or may not be directly applicable to new incoming NBA rookies. Regardless, the methods and techniques explored in this report should be applicable to a majority of NBA rookies.

# References

**Data.World, Inc.** (2016, December 21). Binary Classification Dataset. Retrieved from <https://data.world/community/open-community/>

# Appendix

#### A1. Missing Value - R Input

any(is.na(NBA))

## [1] TRUE

sum(is.na(NBA))

## [1] 11

nba = NBA[rowSums(is.na(NBA)) == 0,]  
any(is.na(nba))

## [1] FALSE

#### A2. GP Variable Transformation - R Input

nba <- nba %>%  
 mutate(GP.lev = case\_when(  
 GP <= 20 ~ "(0,20]", #20 or less games  
 GP <= 30 ~ "(20,30]", #20-30 games  
 GP <= 40 ~ "(30,40]", #30-40 games  
 GP <= 50 ~ "(40,50]", #40-50 games  
 GP <= 60 ~ "(50,60]", #50-60 games  
 GP <= 70 ~ "(60,70]", #60-70 games  
 GP <= 82 ~ "(70,82]")) #70-82 games  
nba$GP.lev <- factor(nba$GP.lev,  
 levels = c("(0,20]","(20,30]",  
 "(30,40]","(40,50]","(50,60]",  
 "(60,70]","(70,82]"))

#### A3. Bootstrap Estimate - R Input

set.seed(754)  
N <- 10000  
targ.five.rate <- numeric(N)  
i <- 1  
for(i in 1:N) {  
 l.sample <- sample(nba$TARGET\_5Yrs, nrow(nba), replace = TRUE)  
 targ.five.rate[i] <- mean(l.sample)  
 i <- i + 1  
}  
m <- mean(targ.five.rate)  
q <- quantile(targ.five.rate, probs = c(0.025, 0.975))  
round(c(q[1], "Mean" = m, q[2]), 3)

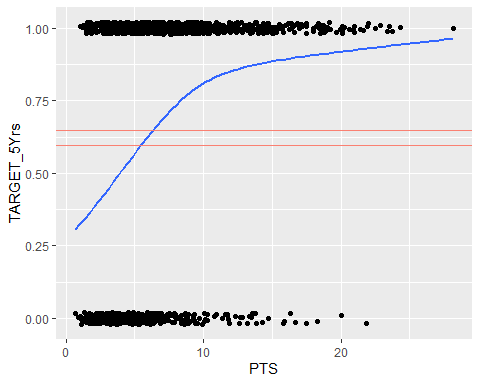
## 2.5% Mean 97.5%   
## 0.595 0.622 0.648

#### A4. Data Exploration Methods - R Input

Simply change variable name to view desired plots or tables.

###### Mean Variance Plot

ggplot(nba[-nba$PTS,]) +  
 aes(x = PTS, y = TARGET\_5Yrs) +  
 geom\_jitter(height = 0.02) +  
 geom\_smooth(se = FALSE) +   
 geom\_hline(yintercept = q, col = "salmon")



###### Mean Variance Table

tbl <- rbind(tapply(nba$TARGET\_5Yrs, nba$GP.lev, sum),  
 tapply(1-nba$TARGET\_5Yrs, nba$GP.lev, sum),  
 tapply(nba$TARGET\_5Yrs, nba$GP.lev, mean))  
dimnames(tbl)[[1]] <- c("5+", "<5", "%5+")  
tbl

## (0,20] (20,30] (30,40] (40,50] (50,60] (60,70]  
## 5+ 3.0000000 13.0000000 52.0000000 96.00 109.00000 136.0000000  
## <5 15.0000000 35.0000000 93.0000000 104.00 98.00000 70.0000000  
## %5+ 0.1666667 0.2708333 0.3586207 0.48 0.52657 0.6601942  
## (70,82]  
## 5+ 417.0000000  
## <5 88.0000000  
## %5+ 0.8257426

#### A5. Preliminary Model Compositions - R Input

**Figure 1.1**

m1 <- glm(TARGET\_5Yrs ~ GP.lev + PTS + FGM + FTM  
 + X3PM + REB + AST + STL + BLK + TOV,  
 data = nba,  
 family = binomial(link = "logit"))  
anova(m1)

## Analysis of Deviance Table  
##   
## Model: binomial, link: logit  
##   
## Response: TARGET\_5Yrs  
##   
## Terms added sequentially (first to last)  
##   
##   
## Df Deviance Resid. Df Resid. Dev  
## NULL 1328 1763.1  
## GP.lev 6 206.970 1322 1556.1  
## PTS 1 31.276 1321 1524.8  
## FGM 1 0.302 1320 1524.5  
## FTM 1 8.246 1319 1516.3  
## X3PM 1 0.000 1318 1516.3  
## REB 1 7.925 1317 1508.4  
## AST 1 0.090 1316 1508.3  
## STL 1 0.522 1315 1507.8  
## BLK 1 2.795 1314 1505.0  
## TOV 1 3.679 1313 1501.3

**Figure 1.2**

m1 <- glm(TARGET\_5Yrs ~ GP.lev + PTS + FGM + FTM  
 + X3PM + REB + AST + STL + BLK + TOV,  
 data = nba,  
 family = binomial(link = "logit"))  
  
m2 <- glm(TARGET\_5Yrs ~ GP.lev + PTS,  
 data = nba,  
 family = binomial(link = "logit"))  
  
m3 <- glm(TARGET\_5Yrs ~ GP.lev + PTS + AST + REB,  
 data = nba,  
 family = binomial(link = "logit"))  
  
m4 <- glm(TARGET\_5Yrs ~ GP.lev + PTS + REB,  
 data = nba,  
 family = binomial(link = "logit"))  
  
m5 <- glm(TARGET\_5Yrs ~ PTS + REB,  
 data = nba,  
 family = binomial(link = "logit"))  
  
m6 <- glm(TARGET\_5Yrs ~ GP.lev + FGM,  
 data = nba,  
 family = binomial(link = "logit"))  
  
m7 <- glm(TARGET\_5Yrs ~ GP.lev + FTM,  
 data = nba,  
 family = binomial(link = "logit"))  
  
m8 <- glm(TARGET\_5Yrs ~ GP.lev + STL,  
 data = nba,  
 family = binomial(link = "logit"))  
  
m9 <- glm(TARGET\_5Yrs ~ GP.lev + BLK,  
 data = nba,  
 family = binomial(link = "logit"))  
  
m10 <- glm(TARGET\_5Yrs ~ GP.lev + FGM + FTM + X3PM,  
 data = nba,  
 family = binomial(link = "logit"))  
  
m11 <- glm(TARGET\_5Yrs ~ GP.lev + FGM + FTM + X3PM + TOV,  
 data = nba,  
 family = binomial(link = "logit"))  
  
m12 <- glm(TARGET\_5Yrs ~ GP.lev + PTS + FTM + REB + BLK + TOV  
 + REB + AST + STL + BLK + TOV,  
 data = nba,  
 family = binomial(link = "logit"))  
  
m13 <- glm(TARGET\_5Yrs ~ GP.lev + FTM + REB + BLK + TOV  
 + REB + AST + BLK + TOV,  
 data = nba,  
 family = binomial(link = "logit"))  
  
m14 <- glm(TARGET\_5Yrs ~ GP.lev + PTS + FTM + TOV,  
 data = nba,  
 family = binomial(link = "logit"))

#### A6. Model Selection Methods - R Input

###### Hosmer-Lemeshow

HL <- function(a, e, g = 10) {  
 y <- a  
 yhat <- e  
 qq <- quantile(yhat, probs = seq(0, 1, 1/g))  
 cutyhat <- cut(yhat, breaks = qq, include.lowest = TRUE)  
 observed <- xtabs(cbind(y0 = 1 - y, y1 = y) ~ cutyhat)  
 expected <- xtabs(cbind(yhat0 = 1 - yhat, yhat1 = yhat) ~ cutyhat)  
 C.hat <- sum((observed - expected)^2/expected)  
 p.val <- 1 - pchisq(C.hat, g - 2)  
 ans <- c("HL Stat." = C.hat,  
 "P-Value" = p.val)  
 return(ans)  
}  
  
rbind("m1" = HL(nba$TARGET\_5Yrs, predict(m1, type = "response")),  
 "m2" = HL(nba$TARGET\_5Yrs, predict(m2, type = "response")),  
 "m3" = HL(nba$TARGET\_5Yrs, predict(m3, type = "response")),  
 "m4" = HL(nba$TARGET\_5Yrs, predict(m4, type = "response")),  
 "m5" = HL(nba$TARGET\_5Yrs, predict(m5, type = "response")),  
 "m6" = HL(nba$TARGET\_5Yrs, predict(m6, type = "response")),  
 "m7" = HL(nba$TARGET\_5Yrs, predict(m7, type = "response")),  
 "m8" = HL(nba$TARGET\_5Yrs, predict(m8, type = "response")),  
 "m9" = HL(nba$TARGET\_5Yrs, predict(m9, type = "response")),  
 "m10" = HL(nba$TARGET\_5Yrs, predict(m10, type = "response")),  
 "m11" = HL(nba$TARGET\_5Yrs, predict(m11, type = "response")),  
 "m12" = HL(nba$TARGET\_5Yrs, predict(m12, type = "response")),  
 "m13" = HL(nba$TARGET\_5Yrs, predict(m13, type = "response")),  
 "m14" = HL(nba$TARGET\_5Yrs, predict(m14, type = "response")))

## HL Stat. P-Value  
## m1 6.2550741 0.6186835  
## m2 0.5264908 0.9998378  
## m3 8.6101400 0.3762430  
## m4 9.5868953 0.2952251  
## m5 6.4479865 0.5971826  
## m6 3.4426231 0.9035968  
## m7 6.6761507 0.5719420  
## m8 4.3940033 0.8199417  
## m9 5.2263439 0.7331329  
## m10 9.7773505 0.2810021  
## m11 7.9525962 0.4381144  
## m12 5.3508321 0.7195045  
## m13 4.7638215 0.7824952  
## m14 9.6775318 0.2883922

###### Classification Metrics

cm.metrics <- function(cm) {  
 acc <- sum(diag(cm))/sum(cm)  
 pre <- cm[1,1]/sum(cm[1,])  
 sen <- cm[1,1]/sum(cm[,1])  
 spe <- cm[2,2]/sum(cm[,2])  
   
 ans <- c("Accuracy" = acc,  
 "Precision" = pre,  
 "Sensitivity" = sen,  
 "Specificity" = spe)  
 return(ans)  
}  
  
M <- matrix(NA, nrow = 14, ncol = 4)  
l <- list(m1, m2, m3, m4, m5, m6, m7, m8, m9, m10, m11, m12, m13, m14)  
i <- 1  
for(f in l){  
 p <- predict(f, type = "response")  
 PC <- ifelse(p > 0.55, 1, 0)  
 TC <- nba$TARGET\_5Yrs  
 cm <- table(factor(PC, levels = 1:0),  
 factor(TC, levels = 1:0))  
 M[i,] <- cm.metrics(cm)  
 i <- i + 1  
}  
dimnames(M) <- list(paste("m", 1:14, sep = ""),  
 c("Accuracy", "Precision", "Sensitivity", "Specificity"))  
round(M \* 100, 2)

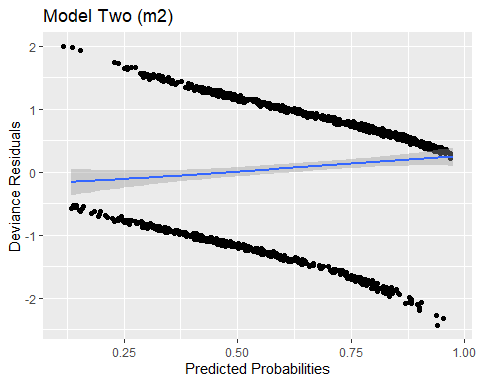
## Accuracy Precision Sensitivity Specificity  
## m1 70.73 77.76 74.09 65.21  
## m2 68.77 76.25 72.28 63.02  
## m3 70.35 77.41 73.85 64.61  
## m4 70.43 77.44 73.97 64.61  
## m5 67.72 74.84 72.40 60.04  
## m6 69.38 76.76 72.76 63.82  
## m7 68.92 76.44 72.28 63.42  
## m8 68.02 76.14 70.70 63.62  
## m9 69.07 76.57 72.40 63.62  
## m10 69.53 77.02 72.64 64.41  
## m11 68.85 76.48 72.03 63.62  
## m12 70.50 77.47 74.09 64.61  
## m13 70.43 77.44 73.97 64.61  
## m14 68.55 76.02 72.15 62.62

#### A7. Final Fitted Logistic Regression Model - R Input/Output

###### Residual Plot for Model Two

p <- predict(m2, type = "response")  
r <- resid(m2, type = "deviance")  
ggplot(data = data.frame(x = p, y = r)) +  
 aes(x = x, y = y) +  
 geom\_jitter(height = 0.02, width = 0.02) +  
 geom\_smooth() +  
 labs(title = "Model Two (m2)", x = "Predicted Probabilities", y = "Deviance Residuals")

## `geom\_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'



writeLines("The line is quite flat indicating favorable performance.")

## The line is quite flat indicating favorable performance.

###### Model Two Summary Statistics

**Figure 1.3**

summary(m2)

##   
## Call:  
## glm(formula = TARGET\_5Yrs ~ GP.lev + PTS, family = binomial(link = "logit"),   
## data = nba)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.4489 -1.0657 0.5790 0.9089 2.0014   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.96524 0.63896 -3.076 0.00210 \*\*   
## GP.lev(20,30] 0.56420 0.71402 0.790 0.42943   
## GP.lev(30,40] 0.96665 0.65851 1.468 0.14212   
## GP.lev(40,50] 1.37430 0.65147 2.110 0.03490 \*   
## GP.lev(50,60] 1.54408 0.65103 2.372 0.01770 \*   
## GP.lev(60,70] 1.91585 0.65501 2.925 0.00345 \*\*   
## GP.lev(70,82] 2.55927 0.65372 3.915 9.04e-05 \*\*\*  
## PTS 0.10746 0.02043 5.261 1.44e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1763.1 on 1328 degrees of freedom  
## Residual deviance: 1524.8 on 1321 degrees of freedom  
## AIC: 1540.8  
##   
## Number of Fisher Scoring iterations: 4

writeLines("We can see that the p-values are significant for PTS and GP levels.")

## We can see that the p-values are significant for PTS and GP levels.

**Figure 1.4**

writeLines("Model Two's Exponentiated Coefficients")

## Model Two's Exponentiated Coefficients

exp(coef(m2))

## (Intercept) GP.lev(20,30] GP.lev(30,40] GP.lev(40,50] GP.lev(50,60]   
## 0.1401223 1.7580395 2.6291187 3.9522924 4.6836679   
## GP.lev(60,70] GP.lev(70,82] PTS   
## 6.7927240 12.9263632 1.1134481

#### A8. K-fold Cross Validation - R Input

**Figure 1.8**

set.seed(754)  
nba$fold <- sample(c(rep(0, 133), rep(1, 133), rep(2, 133),  
 rep(3, 133), rep(4, 133), rep(5, 133),  
 rep(6, 133), rep(7, 133), rep(8, 133), rep(9, 132)),  
 nrow(nba),  
 replace = FALSE)  
  
set.seed(754)  
F <- matrix(NA, nrow = 10, ncol = 5) # for storing our metrics for each fold  
dimnames(F)[[2]] <- c("fold", "accuracy", "precision", "sensitivity", "specificity")  
i <- 1  
for(fld in 0:9){  
 fit <- glm(TARGET\_5Yrs ~ GP.lev + PTS,  
 data = nba,  
 subset = fold != fld, # do not use one fold  
 family = binomial(link = "logit"))  
   
 p <- predict(fit,  
 newdata = subset(nba, subset = fold == fld), # on the fold not used  
 type = "response")  
 PC <- ifelse(p > 0.55, 1, 0)  
 TC <- nba$TARGET\_5Yrs[nba$fold == fld] # True condition on the fold predicted  
 cm <- table(factor(PC, levels = 1:0),  
 factor(TC, levels = 1:0))  
 F[i,] <- c(fld, cm.metrics(cm))  
 i <- i + 1  
}  
(fld.means <- apply(F, 2, mean)[2:5]) # calculate means for each column

## accuracy precision sensitivity specificity   
## 0.6885224 0.7654019 0.7206100 0.6389107

whole.sample <- M[2,]  
tbl <- rbind("Whole Sample" = whole.sample,  
 "Cross-Validated" = fld.means,  
 "Difference" = whole.sample - fld.means)  
dimnames(tbl)[[2]] <- c("Accuracy", "Precision", "Sensitivity", "Specificity")  
round(tbl,4)

## Accuracy Precision Sensitivity Specificity  
## Whole Sample 0.6877 0.7625 0.7228 0.6302  
## Cross-Validated 0.6885 0.7654 0.7206 0.6389  
## Difference -0.0008 -0.0029 0.0022 -0.0087

#### A9. Final Prediction Function and Application - R Input

###### Function

longevity <- function(newdata){  
 data <- newdata  
   
 data <- data %>%  
 mutate(GP.lev = case\_when(  
 GP <= 20 ~ "(0,20]",  
 GP <= 30 ~ "(20,30]",  
 GP <= 40 ~ "(30,40]",  
 GP <= 50 ~ "(40,50]",  
 GP <= 60 ~ "(50,60]",  
 GP <= 70 ~ "(60,70]",  
 GP <= 82 ~ "(70,82]"))  
 data$GP.lev <- factor(data$GP.lev,  
 levels = c("(0,20]","(20,30]",  
 "(30,40]","(40,50]","(50,60]",  
 "(60,70]","(70,82]"))  
   
 m2 <- glm(TARGET\_5Yrs ~ GP.lev + PTS,  
 data = data,  
 family = binomial(link = "logit"))  
   
 p <- predict(m2, newdata = data, type = "response")  
 ans <- ifelse(p > 0.55, 1, 0)  
 return(ans)  
}

###### Application

rooks = read.csv("2018\_Rooks.csv")  
writeLines("Predictions")

## Predictions

longevity(rooks)

## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25   
## 1 1 1 0 0 1 0 0 0 0 0 0 1 0 1 1 1 1 1 1 1 1 1 0 0   
## 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40   
## 0 1 1 0 0 1 1 0 1 0 0 1 1 0 0

writeLines("Sample Mean Longevity Rate")

## Sample Mean Longevity Rate

summary(longevity(rooks))

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.000 0.000 1.000 0.525 1.000 1.000