# Using Random Forests to Forecast NBA Careers

Mario Muñoz Serrano

01/15/24

# Introduction

The purpose of this project is to develop a robust and accurate method for predicting the career arcs of NBA players. Specifically, it aims to address the challenge of determining the optimal length of player contracts by forecasting how long a player can contribute their skills to an NBA team. This project explores various modeling techniques, with a focus on Random Forests as a nonparametric approach, to provide insights and solutions to the player career arc problem. Ultimately, the goal is to offer a practical tool that can assist General Managers and decision-makers in making informed decisions about player contracts in the ever-evolving landscape of professional basketball.

**Note:**

**Throughout this document, any season column represents the year each season started. For example, the 2019-20 season will be in the dataset as 2019.**

# Setup and Data

```
library(tidyverse)
library(dplyr)


awards <- read_csv("Collected data/awards_clean.csv")
player_data <- read_csv("Collected data/player_stats_clean.csv")
```

```
# RENAME COLUMNS

# Rename multiple columns using direct assignment
awards$Defensive_Player_Of_The_Year_rk <- awards$Defensive_Player_of_The_Year
awards$Most_Valuable_Player_rk <- awards$Most_Valuable_Player


# Remove old columns
awards <- awards[, !names(awards) %in% c("Defensive_Player_of_The_Year", "Most_Valuable_Player")]
```

```
# REMOVE PLAYERS THAT HAVE PLAYED 0 MINUTES IN A CERTAIN SEASON

player_data <- player_data[player_data$mins != 0, ]
```

We've seen in our datasets, there are 3 players that have been drafted after playing in the NBA. Since this fact does not make sense. We will pick those players and modify their draft year by the year of the lower season played by them in the NBA.

```
# Check if we have players drafted after playing in the NBA

drafted_after_NBA <- player_data %>%
  group_by(nbapersonid) %>%
  summarize(
    player = first(player),  # Get the first player name within each group
    draftyear = first(draftyear),  # Get the first draft year within each group
    draftpick = first(draftpick),  # Get the first draft pick within each group
    lower_season = min(season)  # Get the lowest season value within each group
  ) %>%
  ungroup()

drafted_after_NBA <- drafted_after_NBA %>%
  filter(draftyear > lower_season)
```

```
drafted_after_NBA
```

```
## # A tibble: 3 × 5
##   nbapersonid player          draftyear draftpick lower_season
##         <dbl> <chr>               <dbl>     <dbl>        <dbl>
## 1         141 Anthony Tucker       1996        NA         1994
## 2        1682 Reggie Hanson        1998        NA         1997
## 3        1872 Randell Jackson      1999        NA         1998
```

```
drafted_after_NBA <- drafted_after_NBA %>%
  select(nbapersonid, lower_season)
```

```
# Merge player_data and filtered_df
player_data <- merge(player_data, drafted_after_NBA, by = "nbapersonid", all.x = TRUE)

# Update draftyear
player_data$draftyear <- ifelse(!is.na(player_data$lower_season), player_data$lower_season, player_data$draftyear
)

# Remove lower_season column
player_data <- subset(player_data, select = -lower_season)
```

Set two global variables that indicates the scope of our dataset. In the case of the collected dataset, we have data for all the NBA players that have played in the NBA since the 1983 season.

```
first_NBA_season_In_data = min(as.integer(player_data$season))
last_NBA_season_In_data = max(as.integer(player_data$season))
```

```
rename_columns_with_underscore <- function(dataframe) {
  new_colnames <- gsub(" ", "_", colnames(dataframe))
  colnames(dataframe) <- new_colnames
  return(dataframe)
}

awards <- rename_columns_with_underscore(awards)
```

```
selection <- c("nbapersonid", "season", "player", "draftyear",
               "draftpick", "nbateamid", "team", "games", "games_start",
               "mins", "fgm", "fga", "fgp", "fgm3",
               "fga3", "fgp3", "fgm2", "fga2", "fgp2",
               "efg", "ftm", "fta", "ftp", "off_reb",
               "def_reb", "tot_reb", "ast", "steals", "blocks",
               "tov", "tot_fouls", "points")

player_data <- player_data[, selection]
```

# Part 1 – Cleaning data and compute labels

In this section we're going to work with data from player awards and player statistics. We'll clean and manipulate the data to be able to compute different levels of career success.

## 1.1 Add new columns useful for the future label creation.

First, convert `all_star_game` column from true false to 0,1. true = 1 false = 0. And add column `is_all_NBA_selected` to know if a player is in Any of the All NBA teams that season.

```
add_is_all_NBA_selected_binary <- function(awards_df) {

  # Convert all_star_game to binary
  awards_df$all_star_game <- as.integer(awards_df$all_star_game)

  # Get which players have been selected in the All NBA First, Second or Third Team
  is_all_NBA_selected = (awards_df$All_NBA_First_Team == 1 | awards_df$All_NBA_Second_Team == 1 | awards_df$All_N
BA_Third_Team == 1)
  awards_df$All_NBA_Selected <- as.numeric(is_all_NBA_selected)
  awards_df <- awards_df %>% relocate(All_NBA_Selected, .after = All_NBA_Third_Team)

  return(awards_df)
}
```

Second, from the ranking columns `Defensive_Player_Of_The_Year_rk` and `Most_Valuable_Player_rk` add two new columns `Defensive_Player_Of_The_Year` and `Most_Valuable_Player` to know which players have been DPOY and MVP in each season.

```
add_DPOY_and_MVP_binary <- function(awards_df){
  # Get which players have been awarded with the MVP and also which players with the DPOY
  # Converting rk to a binary columns
  awards_df <- awards_df %>%
    group_by(season) %>%
    mutate(Defensive_Player_Of_The_Year = ifelse(Defensive_Player_Of_The_Year_rk != 1, 0, Defensive_Player_Of_The
_Year_rk),
           Most_Valuable_Player = ifelse(Most_Valuable_Player_rk != 1, 0, Most_Valuable_Player_rk)) %>%
    ungroup() %>%
    relocate(Defensive_Player_Of_The_Year, .after = Defensive_Player_Of_The_Year_rk) %>%
    relocate(Most_Valuable_Player, .after = Most_Valuable_Player_rk)

  return(awards_df)
}
```

After that we create the function init_awards to perform the operations explained before

```
init_awards_df <- function(awards_df) {

  # Select required variables
  selection = c("season", "nbapersonid", "All_NBA_First_Team", "All_NBA_Second_Team", "All_NBA_Third_Team", "all_
star_game", "Defensive_Player_Of_The_Year_rk", "Most_Valuable_Player_rk")
  awards_df <- awards_df[, selection]

  # Add column to know which players have been all nba
  awards_df <- add_is_all_NBA_selected_binary(awards_df)
  # And another to know which ones DPOY and MVP
  awards_df <- add_DPOY_and_MVP_binary(awards_df)

  return(awards_df)
}
```

Finally, we can join the awards information with the player data using the function

```
init_players_data_and_awards <- function(players_df, awards_df){

  # Join awards info to player data
  selection = c('nbapersonid', 'player', 'draftyear', 'draftpick', 'season', 'nbateamid', 'team', 'games', 'games
_start', 'mins')
  players_df <- players_df[, selection]
  players_and_awd_df <- left_join(players_df, awards_df, by = c("nbapersonid", "season"), relationship = "many-to
-many")

  # Convert NA to 0 to avoid future problems

  columns_to_fix <- c("All_NBA_First_Team", "All_NBA_Second_Team", "All_NBA_Third_Team","All_NBA_Selected", "all_
star_game", "Defensive_Player_Of_The_Year", "Defensive_Player_Of_The_Year_rk", "Most_Valuable_Player_rk", "Most_V
aluable_Player")
  players_and_awd_df <- players_and_awd_df %>%
    mutate(across(all_of(columns_to_fix), ~ coalesce(., 0)))

  return(players_and_awd_df)
}
```

## 1.2 Adjust the statistics in seasons 1998, 2011, 2019 and 2020

Rescaling statistics in NBA seasons with fewer games played ensures fair comparisons. This adjustment accounts for variations in game duration, ensuring equitable evaluations of player performance. Specifically, the 1998, 2011 Lockout-Shortened, and 2019-2020 Covid-Shortened seasons had different game counts: 50, 66, and 72 respectively. Scaling factors (82/50), (82/66), and (82/72) were used for normalization, focusing on absolute statistics like minutes played. This process allows for consistent comparisons across seasons while maintaining fairness in player assessments.

```
adjust_stat <- function(player_data_df, n_games_played, n_games, stat_name, new_stat_name, seasons){

  player_data_df <- player_data_df %>%
    mutate(!!sym(new_stat_name) := ifelse(season %in% seasons, round(!!sym(stat_name) * (n_games_played/n_games))
, !!sym(stat_name)))

  return(player_data_df)

}
```

```
adjust_stats <- function(player_data_df, stats){

  #stats = c("games", "games_start",
  #          "mins", "fgm", "fga", "fgm3",
  #          "fga3", "fgm2", "fga2", "ftm", "fta", "off_reb",
  #          "def_reb", "tot_reb", "ast", "steals", "blocks",
  #          "tov", "tot_fouls", "points")

  for (stat in stats){
    s = stat
    s_adj = paste(stat, "_adjusted", sep = "")



    player_data_df <-adjust_stat(player_data_df, n_games_played = 82, n_games = 66, s, s_adj, c(2011))
    player_data_df <-adjust_stat(player_data_df, n_games_played = 82, n_games = 72, s_adj, s_adj, c(2019,2020))
    player_data_df <-adjust_stat(player_data_df, n_games_played = 82, n_games = 50, s_adj, s_adj, c(1998))

  }


  return(player_data_df)
}
```

## 1.3 Compute season outcome

Now, once we have all the stats with respect the same number of games. And all the information about those All NBA, DPOY and MVP awarded players. We can compute the labels that we'll be used in our model:

- Elite: A player that won any All NBA award (1st, 2nd, or 3rd team), MVP, or DPOY in that season.
- All-Star: A player selected to be an All-Star that season.
- Starter: A player that started in at least 41 games in the season OR played at least 2000 minutes in the season.
- Rotation: A player that played at least 1000 minutes in the season.
- Roster: A player that played at least 1 minute for an NBA team but did not meet any of the above criteria.
- Out of the League: A player that is not in the NBA in that season.

First, only compute the first categories looking to the conditions defined using the compute_season_outcome_column method. Taking into account possible team changes.

```
# Compute season outcome taking into account possible team changes
compute_season_outcome_column <- function(data) {
  result <- data


  result <- result %>% group_by(nbapersonid, season) %>% mutate(isRoster = ifelse(sum(mins_adjusted) >= 1, "Roste
r", NA)) %>% ungroup()

  result <- result %>% group_by(nbapersonid, season) %>% mutate(isRotation = ifelse(sum(mins_adjusted) >= 1000, "
Rotation" , NA)) %>% ungroup()

  result <- result %>% group_by(nbapersonid, season) %>% mutate(isStarter = ifelse(sum(games_start_adjusted) >= 4
1 || sum(mins_adjusted) >= 2000, "Starter" , NA)) %>% ungroup()

  result <- result %>% group_by(nbapersonid, season) %>% mutate(isAllStar = ifelse(any(all_star_game == 1), "All-
Star" , NA)) %>% ungroup()

  result <- result %>% group_by(nbapersonid, season) %>% mutate(isElite = ifelse(any(All_NBA_Selected == 1) || an
y(Most_Valuable_Player == 1) || any(Defensive_Player_Of_The_Year == 1), "Elite" , NA)) %>% ungroup()

  result <- result %>% rowwise() %>% mutate(season_outcome_list = paste(na.omit(c_across(isRoster:isElite)), coll
apse = ", "))




  ########

  result <- result %>% group_by(nbapersonid, season) %>% mutate(season_outcome = ifelse(sum(mins_adjusted) >= 1,
"Roster", season_outcome)) %>% ungroup()

  result <- result %>% group_by(nbapersonid, season) %>% mutate(season_outcome = ifelse(sum(mins_adjusted) >= 100
0, "Rotation" , season_outcome)) %>% ungroup()

  result <- result %>% group_by(nbapersonid, season) %>% mutate(season_outcome = ifelse(sum(games_start_adjusted)
>= 41 || sum(mins_adjusted) >= 2000, "Starter" , season_outcome)) %>% ungroup()

  result <- result %>% group_by(nbapersonid, season) %>% mutate(season_outcome = ifelse(any(all_star_game == 1),
"All-Star" , season_outcome)) %>% ungroup()

  result <- result %>% group_by(nbapersonid, season) %>% mutate(season_outcome = ifelse(any(All_NBA_Selected == 1
) || any(Most_Valuable_Player == 1) || any(Defensive_Player_Of_The_Year == 1), "Elite" , season_outcome)) %>% ung
roup()

  return(result)
}
```

Now regarding the Out of the League. If a player between the first season in our dataset and the final season in the dataset. Has some season that does not appear in the dataset we add them as Out of the League season, with the function add_out_of_the_league_seasons.

```
add_out_of_the_league_seasons <- function(players_and_awd_df){
  # For each player id add all sample seasons
  all_combinations_df <- expand.grid(
    nbapersonid = unique(players_and_awd_df$nbapersonid),
    season = first_NBA_season_In_data:last_NBA_season_In_data
  )

  # Obtain the info (name, draft year ...) of each player
  player_info_df <- players_and_awd_df %>% distinct(nbapersonid, .keep_all = TRUE)

  # Join the info with each id
  all_combinations_df <- inner_join(all_combinations_df, player_info_df, by = "nbapersonid",suffix = c("", ".orig
"), relationship = "many-to-many")


  selection = c("nbapersonid", "player", "draftyear", "draftpick", "season")
  all_combinations_df <- all_combinations_df[, selection]

  # If we have data for a player in certain season we add them
  players_and_awd_df <- left_join(all_combinations_df, players_and_awd_df, by = c("nbapersonid", "season"), suffi
x = c("", ".orig"), relationship = "many-to-many")

  players_and_awd_df <- players_and_awd_df %>% select(-ends_with(".orig"))

  # If a player has no team in a season he is out of the league
  players_and_awd_df <- players_and_awd_df %>% mutate(season_outcome = ifelse(is.na(nbateamid), "Out of the Leagu
e", season_outcome))
  players_and_awd_df <- players_and_awd_df %>% mutate(season_outcome_list = ifelse(is.na(nbateamid), "Out of the
League", season_outcome_list))
  return(players_and_awd_df)

}
```

Now once we have all the methods we can create a function compute_season_outcome that performs each step explained before to have a df that has a label season_outcome for each player and season.

```
compute_season_outcome <- function(awards_df, players_df) {

  st = c("mins","games_start")

  awards_df <- init_awards_df(awards_df)

  players_and_awd_df <- init_players_data_and_awards(players_df, awards_df)

  players_and_awd_df <- adjust_stats(players_and_awd_df, st)

  players_and_awd_df <- compute_season_outcome_column(players_and_awd_df)

  players_and_awd_df <- add_out_of_the_league_seasons(players_and_awd_df)


  players_and_awd_df <- players_and_awd_df %>% group_by(nbapersonid, season) %>% distinct(nbapersonid, season, .k
eep_all = TRUE) %>% ungroup()

  selection = c("nbapersonid", "player", "draftyear", "draftpick", "season", "season_outcome_list", "season_outco
me")


  return(players_and_awd_df[, selection])
}
```

```
season_outcome_data <- compute_season_outcome(awards, player_data)
```

```
manu_example <- subset(season_outcome_data, nbapersonid == 1938)
manu_example <- manu_example[order(manu_example$nbapersonid, manu_example$season), ]
```

As we can see in the Manu Ginobili example below, we have the personal information of the player including their nbapersonid, player name, draft year and draft pick. And also for each season we have two more variables `season_outcome_list` , that has all of the categories for which the player is qualified that season and another variable `season_outcome` which takes the highest category for which the player is qualified that season.

```
knitr::kable(manu_example, "pipe")
```

| nbapersonid | player | draftyear | draftpick | season | season_outcome_list | season_outcome |
|---|---|---|---|---|---|---|
| 1938 | Manu Ginobili | 1999 | 57 | 1983 | Out of the League | Out of the League |
| 1938 | Manu Ginobili | 1999 | 57 | 1984 | Out of the League | Out of the League |
| 1938 | Manu Ginobili | 1999 | 57 | 1985 | Out of the League | Out of the League |
| 1938 | Manu Ginobili | 1999 | 57 | 1986 | Out of the League | Out of the League |
| 1938 | Manu Ginobili | 1999 | 57 | 1987 | Out of the League | Out of the League |
| 1938 | Manu Ginobili | 1999 | 57 | 1988 | Out of the League | Out of the League |
| 1938 | Manu Ginobili | 1999 | 57 | 1989 | Out of the League | Out of the League |
| 1938 | Manu Ginobili | 1999 | 57 | 1990 | Out of the League | Out of the League |
| 1938 | Manu Ginobili | 1999 | 57 | 1991 | Out of the League | Out of the League |
| 1938 | Manu Ginobili | 1999 | 57 | 1992 | Out of the League | Out of the League |
| 1938 | Manu Ginobili | 1999 | 57 | 1993 | Out of the League | Out of the League |
| 1938 | Manu Ginobili | 1999 | 57 | 1994 | Out of the League | Out of the League |
| 1938 | Manu Ginobili | 1999 | 57 | 1995 | Out of the League | Out of the League |
| 1938 | Manu Ginobili | 1999 | 57 | 1996 | Out of the League | Out of the League |
| 1938 | Manu Ginobili | 1999 | 57 | 1997 | Out of the League | Out of the League |
| 1938 | Manu Ginobili | 1999 | 57 | 1998 | Out of the League | Out of the League |
| 1938 | Manu Ginobili | 1999 | 57 | 1999 | Out of the League | Out of the League |
| 1938 | Manu Ginobili | 1999 | 57 | 2000 | Out of the League | Out of the League |
| 1938 | Manu Ginobili | 1999 | 57 | 2001 | Out of the League | Out of the League |
| 1938 | Manu Ginobili | 1999 | 57 | 2002 | Roster, Rotation | Rotation |
| 1938 | Manu Ginobili | 1999 | 57 | 2003 | Roster, Rotation, Starter | Starter |
| 1938 | Manu Ginobili | 1999 | 57 | 2004 | Roster, Rotation, Starter, All-Star | All-Star |
| 1938 | Manu Ginobili | 1999 | 57 | 2005 | Roster, Rotation, Starter | Starter |
| 1938 | Manu Ginobili | 1999 | 57 | 2006 | Roster, Rotation, Starter | Starter |
| 1938 | Manu Ginobili | 1999 | 57 | 2007 | Roster, Rotation, Starter, Elite | Elite |
| 1938 | Manu Ginobili | 1999 | 57 | 2008 | Roster, Rotation | Rotation |
| 1938 | Manu Ginobili | 1999 | 57 | 2009 | Roster, Rotation, Starter | Starter |
| 1938 | Manu Ginobili | 1999 | 57 | 2010 | Roster, Rotation, Starter, All-Star, Elite | Elite |
| 1938 | Manu Ginobili | 1999 | 57 | 2011 | Roster | Roster |
| 1938 | Manu Ginobili | 1999 | 57 | 2012 | Roster, Rotation | Rotation |

| | | | | | | |
|---|---|---|---|---|---|---|
| 1938 | Manu Ginobili | 1999 | 57 | 2013 | Roster, Rotation | Rotation |
| 1938 | Manu Ginobili | 1999 | 57 | 2014 | Roster, Rotation | Rotation |
| 1938 | Manu Ginobili | 1999 | 57 | 2015 | Roster, Rotation | Rotation |
| 1938 | Manu Ginobili | 1999 | 57 | 2016 | Roster, Rotation | Rotation |
| 1938 | Manu Ginobili | 1999 | 57 | 2017 | Roster, Rotation | Rotation |
| 1938 | Manu Ginobili | 1999 | 57 | 2018 | Out of the League | Out of the League |
| 1938 | Manu Ginobili | 1999 | 57 | 2019 | Out of the League | Out of the League |
| 1938 | Manu Ginobili | 1999 | 57 | 2020 | Out of the League | Out of the League |
| 1938 | Manu Ginobili | 1999 | 57 | 2021 | Out of the League | Out of the League |
| 1938 | Manu Ginobili | 1999 | 57 | 2022 | Out of the League | Out of the League |

## 1.3 Compute career outcome

Now we defined the **career outcome** for each player, representing the highest level of success that the player achieved for at least two seasons after his first four seasons in the league.

So, first we need to know which is the first season the player played in the league. That will be the season 1. Then we don't look for the season outcome in the season 1, 2, 3 and 4. And from the season 5 onwards we'll be checking the season_outcome_list to see which is the highest level of succes repeated in two different seasons.

But let's start from the beginning. With the functions find_first_non_out and add_first_nba_season_in_dataset. We find first, for a player which is the first season played in the NBA that we have in our dataset. And we add this year to another column for each player.

```
# Function to find the first non- "Out of the League" column name in a row
find_first_non_out <- function(...) {
  years <- first_NBA_season_In_data:last_NBA_season_In_data
  columns_to_check <- paste0("s_outc_", years)

  first_non_out <- which(c(...) != "Out of the League")[1]
  if (!is.na(first_non_out)) {
    col_name <- columns_to_check[first_non_out]
    numeric_part <- as.numeric(gsub("\\D", "", col_name))  # Extract numeric part
    as.character(numeric_part)

  } else {
    NA
  }
}
```

```
add_first_nba_season_in_dataset <- function(career_outcome_df){
  years <- first_NBA_season_In_data:last_NBA_season_In_data
  columns_to_check <- paste0("s_outc_", years)
  career_outcome_df <- career_outcome_df %>% mutate(first_nba_season_in_dataset = pmap_chr(select(., all_of(colum
ns_to_check)), find_first_non_out))

  career_outcome_df$first_nba_season_in_dataset <- as.numeric(career_outcome_df$first_nba_season_in_dataset)
  return(career_outcome_df)

}
```

After that, we made check_possible_career_outcome and check_all_possible_career_outcomes that we'll be used to check in the season 5 onwards which is the highest season outcome repeated at least two times.

```
check_possible_career_outcome <- function(df_row, string){

  at_least_two_columns <- sum(apply(df_row, 2, function(col) grepl(string, col))) >= 2

  return(at_least_two_columns)
}
```

```
check_all_possible_career_outcomes <- function(df_row){
  result <- case_when(
    check_possible_career_outcome(df_row, "Elite") ~ "Elite",
    check_possible_career_outcome(df_row, "All-Star") ~ "All-Star",
    check_possible_career_outcome(df_row, "Starter") ~ "Starter",
    check_possible_career_outcome(df_row, "Rotation") ~ "Rotation",
    check_possible_career_outcome(df_row, "Roster") ~ "Roster",
    TRUE ~ "Out of the League"
  )

  return(result)
}
```

And we use add_career_outcome to iterate over each player calling the function check check_all_possible_career_outcomes to check the highest level of success repeated at least two times from the season 5 onwards.

```
add_career_outcome <- function(career_outcome_df){

  career_outcome_df$career_outcome <- NA


  for (i in 1:nrow(career_outcome_df)) {



    first_season_study <- (career_outcome_df[i, "first_nba_season_in_dataset"] + 4)

    if(first_season_study >= last_NBA_season_In_data){ career_outcome_df[i, "career_outcome"] = "Out of the Leagu
e"
    next
    }

    years <- first_season_study:last_NBA_season_In_data
    cols <- paste0("s_outc_list_", years)

    # Extract the relevant columns between start_column and end_column
    columns_to_check <- career_outcome_df[i, cols]



    career_outcome_df[i, "career_outcome"] <- check_all_possible_career_outcomes(columns_to_check)


  }

  return(career_outcome_df)
}
```

And finally we created the method compute_career_outcome to perform all the steps explained before.

```
compute_career_outcome <- function(season_outcome_df){

  selection <- setdiff(names(season_outcome_data), "season_outcome")
  pivot_s_ouctc_list <- season_outcome_data[, selection] %>% group_by(nbapersonid) %>% pivot_wider(names_from = s
eason, values_from = season_outcome_list, names_prefix = "s_outc_list_") %>% ungroup()

  selection <- c("nbapersonid", "season", "season_outcome")
  pivot_s_ouctc <- season_outcome_data[, selection] %>% group_by(nbapersonid) %>% pivot_wider(names_from = season
, values_from = season_outcome, names_prefix = "s_outc_") %>% ungroup()

  career_outcome_df <- merge(pivot_s_ouctc_list, pivot_s_ouctc, by = "nbapersonid")

  career_outcome_df <- add_first_nba_season_in_dataset(career_outcome_df)


  career_outcome_df <- add_career_outcome(career_outcome_df)

  selection = c("nbapersonid", "player", "draftyear", "draftpick", "career_outcome")

  return(career_outcome_df[, selection])

}
```

```
career_outcome_data <- compute_career_outcome(season_outcome_data)
```

```
career_outcome_model <- subset(career_outcome_data, draftyear >= first_NBA_season_In_data)
cat(paste("### ", 'Career Outcome Example', "\n"))
```

```
## ###  Career Outcome Example
```

```
knitr::kable(head(career_outcome_model,15), "pipe")
```

| | nbapersonid | player | draftyear | draftpick | career_outcome |
|---|---|---|---|---|---|
| 1 | 2 | Byron Scott | 1983 | 4 | Starter |
| 2 | 3 | Grant Long | 1988 | 33 | Starter |
| 4 | 9 | Sedale Threatt | 1983 | 139 | Starter |
| 5 | 12 | Chris King | 1992 | 45 | Out of the League |
| 6 | 15 | Eric Piatkowski | 1994 | 15 | Starter |
| 7 | 17 | Clyde Drexler | 1983 | 14 | Elite |
| 8 | 21 | Greg Anthony | 1991 | 12 | Starter |
| 9 | 22 | Rik Smits | 1988 | 2 | Starter |
| 10 | 23 | Dennis Rodman | 1986 | 27 | Elite |
| 11 | 24 | Keith Jennings | 1990 | NA | Out of the League |
| 12 | 26 | Luc Longley | 1991 | 7 | Starter |
| 13 | 28 | Doug West | 1989 | 38 | Starter |
| 14 | 29 | Jim McIlvaine | 1994 | 32 | Roster |

| | | | | | |
|---|---|---|---|---|---|
| 15 | 30 | Richard Dumas | 1991 | 46 | Out of the League |
| 16 | 31 | Lorenzo Williams | 1991 | NA | Roster |

```
# Count the number of NA values in each column
null_counts <- colSums(is.na(career_outcome_model))

# Print the number of nulls per column
print(null_counts)
```

```
##    nbapersonid         player      draftyear      draftpick career_outcome
##              0              0              0            889              0
```

```
category_counts <- table(career_outcome_model$career_outcome)

# Print the number of appearances of each category
print(category_counts)
```

```
##
##          All-Star             Elite Out of the League            Roster
##                47               72             1809               318
##          Rotation           Starter
##               243               487
```

# Part 1 – Random Forest Model

```
init_model_data <- function(season_outcome_df, career_outcome_df, player_data_df){
  season_outcome_df$season_num <- season_outcome_df$season - season_outcome_df$draftyear

  season_outcome_df <- season_outcome_df[, c("nbapersonid", "season", "season_num",   "season_outcome")]

  career_outcome_df <- career_outcome_df[,c("nbapersonid", "career_outcome")]

  data <- merge(season_outcome_df, career_outcome_df, by = "nbapersonid")


  data <- merge(player_data_df, data, by = c("nbapersonid", "season"))

  data <- data %>% relocate(season_num, .after = season)


  return(data)
}
```

```
process_data <- function(data) {
  # Step 1: Remove specified columns
  data <- data %>%
    select(-c(
      games, games_start, mins, fgm, fga, fgm3, fga3, fgm2, fga2,
      ftm, fta, off_reb, def_reb, tot_reb, ast, steals, blocks,
      tov, tot_fouls, points, fgp, fgp3, fgp2, ftp, efg
    ))

  # Step 2: Rename columns ending with "_adjusted"
  data <- data %>%
    rename_all(~sub("_adjusted$", "", .))

  # Step 3: Reorder columns
  data <- data %>%
    select(
      nbapersonid, season, season_num, player, draftyear, draftpick,
      nbateamid, team,
      games, games_start, mins, fgm, fga, fgm3, fga3, fgm2, fga2,
      ftm, fta, off_reb, def_reb, tot_reb, ast, steals, blocks,
      tov, tot_fouls, points,
      season_outcome, career_outcome
    )

  return(data)
}
```

```
summarize_multi_team_players <- function(data) {
  result <- data %>%
    group_by(nbapersonid, season) %>%
    mutate(
      num_teams = n_distinct(nbateamid),
      team_names = paste(unique(team), collapse = ", ")
    ) %>%
    ungroup() %>%
    group_by(nbapersonid, season) %>%
    summarise(
      season_num = first(season_num),
      player = first(player),
      draftyear = first(draftyear),
      draftpick = first(draftpick),
      games = sum(games),
      games_start = sum(games_start),
      mins = sum(mins),
      fgm = sum(fgm),
      fga = sum(fga),
      fgm3 = sum(fgm3),
      fga3 = sum(fga3),
      fgm2 = sum(fgm2),
      fga2 = sum(fga2),
      ftm = sum(ftm),
      fta = sum(fta),
      off_reb = sum(off_reb),
      def_reb = sum(def_reb),
      tot_reb = sum(tot_reb),
      ast = sum(ast),
      steals = sum(steals),
      blocks = sum(blocks),
      tov = sum(tov),
      tot_fouls = sum(tot_fouls),
      points = sum(points),
      season_outcome = first(season_outcome),
      career_outcome = first(career_outcome),
      num_teams = first(num_teams),
      team_names = first(team_names)
    )

  result <- result %>%
  select(nbapersonid, player, draftyear, draftpick, everything())

  result <- result %>%
  select(-season_outcome, -career_outcome, everything(), season_outcome, career_outcome)

  return(result)
}
```

```
compute_and_round_shooting_percentages <- function(data) {
  data <- data %>%
    mutate(
      # Field Goal Percentage (FGP)
      fgp = ifelse(fga == 0, 0, round(fgm / fga, 3)),

      # Field Goal Percentage for Two-Pointers (FGP2)
      fgp2 = ifelse(fga2 == 0, 0, round(fgm2 / fga2, 3)),

      # Field Goal Percentage for Three-Pointers (FGP3)
      fgp3 = ifelse(fga3 == 0, 0, round(fgm3 / fga3, 3)),

      # Effective Field Goal Percentage (eFG)
      efg = round((fgm + 0.5 * fgm3) / fga, 3),

      # Free Throw Percentage (FTP)
      ftp = ifelse(fta == 0, 0, round(ftm / fta, 3))
    )

  data <- data %>%
  select(
    nbapersonid, player, draftyear, draftpick, season, season_num, games, games_start,
    mins, fgm, fga, fgp, efg, fgm3, fga3, fgp3, fgm2, fga2, fgp2,
    ftm, fta, ftp, off_reb, def_reb, tot_reb, ast, steals, blocks, tov,
    tot_fouls, points, num_teams, team_names, season_outcome, career_outcome
  )

  return(data)
}
```

```
perform_pivot_wider <- function(data){

  result <- data %>%
  select(-season) %>%
  pivot_wider(
    id_cols = c(nbapersonid, player, draftyear, draftpick, career_outcome),
    names_from = season_num,
    values_from = c(games, games_start, mins, fgm, fga, fgp, efg, fgm3, fga3, fgp3, fgm2, fga2, fgp2, ftm, fta, f
tp, off_reb, def_reb, tot_reb, ast, steals, blocks, tov, tot_fouls, points, num_teams, team_names, season_outcome
),
    names_sep = "_Year_",
    names_sort = TRUE
  )

  return(result)
}
```

```
prepare_model_data <- function(season_outcome_df, career_outcome_df, player_data_df){

  st = c("games", "games_start",
           "mins", "fgm", "fga", "fgm3",
           "fga3", "fgm2", "fga2", "ftm", "fta", "off_reb",
           "def_reb", "tot_reb", "ast", "steals", "blocks",
           "tov", "tot_fouls", "points")

  model_data_df <- init_model_data(season_outcome_df, career_outcome_df, player_data_df)

  model_data_df <- adjust_stats(model_data_df, st)

  model_data_df <- process_data(model_data_df)

  model_data_df <- summarize_multi_team_players(model_data_df)

  model_data_df <- compute_and_round_shooting_percentages(model_data_df)

  model_data_df <- perform_pivot_wider(model_data_df)


  return(model_data_df)
}
```

```
## PREPARE INPUT TO BUILD MODEL DATA

# To have all the seasons that a player have played in the nba we filter by draft year equals to the first season
in our dataset

career_outcome_model <- subset(career_outcome_data, draftyear >= first_NBA_season_In_data)

season_outcome_model <- subset(season_outcome_data, draftyear >= first_NBA_season_In_data)
player_data_model <- subset(player_data, draftyear >= first_NBA_season_In_data)

write.csv(career_outcome_model, file = "career_outcome_model.csv", row.names = FALSE)
write.csv(season_outcome_model, file = "season_outcome_model.csv", row.names = FALSE)
write.csv(player_data_model, file = "player_data_model.csv", row.names = FALSE)
```

```
model_data <- prepare_model_data(season_outcome_model, career_outcome_model, player_data_model)
```

```
## `summarise()` has grouped output by 'nbapersonid'. You can override using the
## `.groups` argument.
```

```
write.csv(model_data, file = "data.csv", row.names = FALSE)
print(nrow(model_data))
```

```
## [1] 2976
```

```
cat('<div style="overflow-x:auto;">')
```

```
knitr::kable(head(model_data, 15), format = "html", table.attr = 'class="table table-bordered table-hover"')
```

| nbapersonid | player | draftyear | draftpick | career_outcome | games_Year_0 | games_Year_1 | games_Year_2 | games_Year_3 |
|---|---|---|---|---|---|---|---|---|
| 2 | Byron Scott | 1983 | 4 | Starter | 74 | 81 | 76 | 82 |
| 3 | Grant Long | 1988 | 33 | Starter | 82 | 81 | 80 | 82 |
| 9 | Sedale Threatt | 1983 | 139 | Starter | 45 | 82 | 70 | 68 |
| 12 | Chris King | 1992 | 45 | Out of the League | NA | 15 | NA | 80 |
| 15 | Eric Piatkowski | 1994 | 15 | Starter | 81 | 65 | 65 | 67 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 17 | Clyde Drexler | 1983 | 14 | Elite | 82 | 80 | 75 | 82 |
| 21 | Greg Anthony | 1991 | 12 | Starter | 82 | 70 | 80 | 61 |
| 22 | Rik Smits | 1988 | 2 | Starter | 82 | 82 | 76 | 74 |
| 23 | Dennis Rodman | 1986 | 27 | Elite | 77 | 82 | 82 | 82 |
| 24 | Keith Jennings | 1990 | NA | Out of the League | NA | NA | 8 | 76 |
| 26 | Luc Longley | 1991 | 7 | Starter | 66 | 55 | 76 | 55 |
| 28 | Doug West | 1989 | 38 | Starter | 52 | 75 | 80 | 80 |
| 29 | Jim McIlvaine | 1994 | 32 | Roster | 55 | 80 | 82 | 78 |
| 30 | Richard Dumas | 1991 | 46 | Out of the League | NA | 48 | NA | 15 |
| 31 | Lorenzo Williams | 1991 | NA | Roster | NA | 27 | 38 | 82 |

```
cat('</div>')
```