NBA Free Agency Contract Prediction 2023

Sumitro Datta

2023-06-25

Introduction

This is going to be my fourth year attempting to predict free agent contracts. What happened last year?

- **James Harden & Bradley Beal** both opted out of their player options & re-signed with their teams. Harden signed a 2-year, $68.6M contract including a player option with the Sixers. Beal signed a five-year, $251M maximum contract (including a no-trade clause) to stay with the Wizards.
- Restricted free agents **Deandre Ayton & Anfernee Simons** cashed in, signing long-term deals with their current teams. Ayton signed a 4-year, $133M offer sheet with the Pacers that was subsequently matched by the Suns. Simons agreed to a 4-year, $100M contract with the Blazers.
- Unrestricted free agents **Zach LaVine & Jalen Brunson** were the only other players to secure deals totalling more than $100M. LaVine extended his tenure with the Bulls by agreeing to a 5-year, $215M maximum contract. Jalen Brunson was the only player in the top 6 and one of 2 in the top 13 to switch teams, moving from Dallas to New York on a 4-year, $104M contract.

Top players holding player options include **Khris Middleton, James Harden & Kristaps Porzingis.** Middleton was a key part of the Bucks championship in 2021, but struggled with health this year. He had offseason wrist surgery causing him to miss the first 20 games of the season and he encountered some knee soreness that cost him eighteen games in December & January. Harden signed a short-term contract at less than his maximum amount to give the Sixers some cap flexibility; this allowed the team to sign Harden's former teammate PJ Tucker. Harden was the sidekick to eventual MVP Joel Embiid, but the Sixers bowed out in the second round of the playoffs to their hated rivals in the Boston Celtics. Porzingis had his first full season in Washington after being traded there in 2022 from the Mavericks, where the Luka-Zinger pairing had seemingly run its course. Porzingis had his healthiest season in five years (since his All-Star year in 2018), setting career highs in points per game and shooting efficiency.

In terms of restricted free agents, the top players are **Cameron Johnson, PJ Washington & Austin Reaves**. In his sophomore year, Johnson was the Suns sixth man on their run to the Finals in 2021. He has improved his efficiency as his field goal attempts have increased. Johnson was part of the trade bringing Kevin Durant to the desert, landing in Brooklyn with Mikal Bridges. Washington has spent the first four years of his NBA career in Charlotte as a steady big in a team usually dominated by its guards (Terry Rozier has been on the team for the same time as Washington, Devonte' Graham led the team in scoring in Washington's first year, and LaMelo Ball was drafted at number 3 in 2021). Reaves went undrafted in 2021 and signed a two-way deal with the Lakers. He was converted to a regular deal a month before the season started, and showed some promise in a season in which the Lakers missed the playoffs due to stars LeBron James & Anthony Davis missing large chunks of the season. Reaves leveled up in 2023, averaging the fifth-most minutes per game & points per game on the team.

On the unrestricted free agent side, we've got **Kyrie Irving, Nikola Vucevic & Fred VanVleet** topping the lists. Kyrie had a $36.5M option last year that he picked up, surprising even his own team. He was named an All-Star for the eighth time in his career, but after talks stalled on an extension, Kyrie requested a trade in early February. His wish was granted two days later when he was sent to the Dallas Mavericks. Vucevic was acquired at the 2021 trade deadline by the Bulls in a trade with the Magic to give Zach LaVine the best big man he had played with. After signing Lonzo Ball and DeMar DeRozan in the 2021 offseason, the Bulls looked to make some noise. Unfortunately, the 2022 Bulls were the 6th-seed and lost in the first round to the eventual champion Bucks, while the 2023 Bulls lost in the play-in. With Ball having a seemingly career-ending injury, perhaps the Bulls let Vucevic walk rather than locking themselves into a lower-end playoff team at best? VanVleet was a free agent back in 2020 on the first iteration of this project. Predictions came in at 4-years and $93-97 million, and VanVleet ended up re-signing with the Raptors on a 4-year, $85M contract with a player option on the final year. VanVleet was an All-Star in 2022, but also never shot better than 40.3% from the field in the past three years despite taking the second-most shots on the team.

What I wanted to do was predict what contracts this year's free agent class might get based off previous offseasons. Stars generally get star-type money on account of there being a maximum contract; but in tiers below, contracts of comparable players usually come up in discussing contract value.

Methods/Analysis

Loading Packages

Let's start by loading required packages.

```
packages=c("tidyverse",
           "tidymodels",
           "janitor", # cleaning variable names
           "glmnet", # multinomial regression
           "ranger", # random forest algorithm wrapper
           "kknn", # nearest neighbors algorithm wrapper
           "rpart", # decision tree algorithm wrapper
           "rpart.plot", #plot decision tree outputs
           "kernlab", # svm kernel wrapper
           "vip", # variable importance
           "zoo", # rolling/window operations
           "matrixStats",
           "RColorBrewer",
           "readxl", #read excel files
           "ggdark", # dark background plots
           "gt") # table creator (replace kableExtra & formattable)
for (pkg in packages){
  if(!require(pkg,character.only = TRUE)){
    install.packages(pkg,repos = "http://cran.us.r-project.org")
  }
}
```

Importing the Data

For the statistical data, I've scraped total and advanced stats from Basketball-Reference and stored them in .csv files. This was actually part of a larger project to scrape complete statistics for teams, players and awards (the Kaggle dataset resides here). To my knowledge, my dataset is unique in that it includes BAA stats and ABA stats, which is not really of use here.

The advanced stats I kept were cumulative (offensive win shares, defensive win shares and value over replacement player). For players who played on multiple teams in one season, I kept their total stats and discarded the team-specific season portions. There was an initial desire to use totals to bake in availability/body fragility, but the shortened seasons would cause the model to declare all players to be fragile and underestimate their contract.

In the first iteration of this project, we scaled games played and games started to a normal distribution due to fluctuations in games played between seasons caused by the COVID-19 pandemic. We will convert the games started to a percentage of games played and we will change the games played to a percentage of maximum playable games. This maximum will differ for players who played for multiple teams in one season.

```r
#specify columns because otherwise birth year is read as logical
cols_for_stats=cols(
  .default = col_double(),
  player = col_character(),
  pos = col_character(),
  lg = col_character(),
  tm = col_character()
)

advanced<-read_csv("Data/Advanced.csv",col_types = cols_for_stats) %>%
  select(seas_id:mp,ows:ws,vorp) %>%
  #players with 0 mp (rounded down) would have div by zero error
  mutate(ws_per_48=if_else(mp==0,0,ws/mp*48),.before="vorp")
totals<-read_csv("Data/Player Totals.csv",col_types = cols_for_stats)
#max games per season for players on multiple teams
max_games_tots=totals %>% filter(tm=="TOT") %>% group_by(season,lg,tm) %>%
  summarize(max_games_tot=max(g,na.rm = TRUE)) %>% ungroup()
#max games per season for players on single team
max_games=totals %>% filter(tm !="TOT") %>% group_by(season,lg) %>%
  summarize(max_games_non_tot=max(g,na.rm = TRUE)) %>% ungroup()
#coalesce above two into one column in totals df
totals_enhanced=left_join(totals,max_games_tots) %>% left_join(.,max_games) %>%
  mutate(max_games_playable=coalesce(max_games_tot,max_games_non_tot)) %>%
  select(-c(max_games_non_tot,max_games_tot))
advanced_and_totals<-left_join(totals_enhanced,advanced) %>%
  #if player played for multiple teams in season, only take total row
  mutate(tm=ifelse(tm=="TOT","1TOT",tm)) %>%
  group_by(player_id,season) %>% arrange(tm) %>% slice(1) %>%
  mutate(tm=ifelse(tm=="1TOT","TOT",tm)) %>%
  arrange(season,player) %>%
  mutate(g_percent=g/max_games_playable,gs_percent=gs/g,.before=g) %>%
  select(-c(gs,max_games_playable)) %>%
  #filter since 1997 to match w/play-by-play + faster pre-processing
  filter(season > 1996) %>% ungroup()
```

A comment was raised on an earlier iteration regarding positional scarcity. Teams will overpay for the potential piece that puts them “over the hump”, whether that be into playoff contention or the more loftier goal of championship contention. Teams might also panic to acquire a player that is deemed to be the last one in a talent tier above the remaining free agents in the same position.

The play-by-play data (available since the 1997 season) keeps track of the percentage of minutes played a player has played in each traditional position (point guard, shooting guard, small forward, power forward & center). I converted the percentages to raw minutes played at each position, and summed across all teams a player played for in the same season. I felt this method was more accurate than using the "totals" row.

```r
play_by_play<-read_csv("Data/Player Play By Play.csv") %>%
  filter(tm!="TOT") %>%
  select(seas_id:player,mp:c_percent)

#replace NA's with zeroes
play_by_play[is.na(play_by_play)] <- 0

pbp_pos_mins=play_by_play %>%
  #convert percents to minutes played at position
  mutate(across(pg_percent:c_percent,~./100*mp)) %>%
  rename_with(.fn=~gsub(x = ., pattern = "_percent", replacement = "_mp"),
              .cols=pg_percent:c_percent) %>%
  #sum season minutes across different teams
  group_by(season,player_id) %>%
  mutate(across(mp:c_mp,sum,.names="{col}_summed")) %>%
  slice(1) %>% ungroup() %>% select(-c(mp:c_mp))
```

Another concern that I noticed is that top tier players have depressed contract predictions. Due to the smaller number of players on the court and smaller rosters in general, an individual player's importance is heightened compared to other team sports. As such, elite players get paid the most money. To identify elite players, I brought in end-of-season All-Defense and All-NBA Team voting.

There are two All-Defense teams made up of 5 players each, honoring the top defensive players in the league. They have been voted on by media since 2014. Prior to that, NBA head coaches voted on the All-Defensive team recipients, with the caveat that they could not vote for players on their own team. Players get 2 points for a first-team vote, and 1 point for a second-team vote. There are three All-NBA teams made of 5 players each, honoring the top overall players in the league. The All-NBA teams are voted on by media, and were expanded to the current 3-team setup in 1989. Players get 5 points for a first-team vote, 3 points for a second-team vote and 1 point for a third-team vote. To normalize differences in total points received, I calculated a player's vote share by dividing their points received by the maximum points available (essentially if every possible voter gave that player a first-team vote).

The All-NBA voting is also pulled from Basketball-Reference from each season's awards voting section (here's 2023 for an example). Basketball-Reference very recently added All-Defense voting, so I compiled it myself. Since 2014, the NBA has published all voter ballots and I've scraped those PDF files for a separate project. To get earlier results, I relied on Patricia Bender's site as well as newspapers.com user iknowball.

```
all_defense_voting_since_2014=read_csv("Data/all-def.csv") %>%
  pivot_longer(cols=first_fwd:second_g_2,names_to="points",values_to="player") %>%
  #2023 awards have no comma between last name & first name
  mutate(player=if_else(year==2023,str_replace(player," ",", "),player)) %>%
  mutate(player=gsub("\\(.*","",player)) %>%
  mutate(player=gsub("--.*","",player)) %>% mutate(player=str_trim(player)) %>%
  separate(player,into=c("last","first"),sep=", ",convert = TRUE) %>%
  unite("player",c(first,last),sep=" ",na.rm=TRUE) %>%
  mutate(points_given=case_when(
    (str_detect(points,"first"))~2,
    (str_detect(points,"second"))~1,
  )) %>%
  mutate(player=case_when(
    #3 players in all-def 2015 have one dash rather than two
    str_detect(player," - IND")~"George Hill",
    str_detect(player," - SA")~"Danny Green",
    str_detect(player," - Mil")~"Giannis Antetokounmpo",
    str_detect(player,"PJ Tucker")~"P.J. Tucker",
    str_detect(player,"TJ McConnell")~"T.J. McConnell",
    str_detect(player,"Ginobili")~"Manu Ginóbili",
    str_detect(player,"Porzingis")~"Kristaps Porziņģis",
    str_detect(player,"Jokic")~"Nikola Jokić",
    str_detect(player,"Doncic")~"Luka Dončić",
    str_detect(player,"Schroder")~"Dennis Schröder",
    str_detect(player,"Robert Williams III")~"Robert Williams",
    str_detect(player,"Michael Jr. Porter")~"Michael Porter Jr.",
    str_detect(player,"Jr. Jaren Jackson")~"Jaren Jackson Jr.",
    str_detect(player,"O.G. Anunoby")~"OG Anunoby",
    TRUE~player)) %>%
  #replace non-ascii dashes
  mutate(player=str_replace(player,"\u2010","-")) %>%
  mutate(vote_position=word(points,start=2,sep="_")) %>%
  select(-points) %>% rename(season=year)

all_d_vote_shares_since_2014=all_defense_voting_since_2014 %>%
  #points maximum is number of ballots * 2 points for first-team vote
```

```
    #number of ballots is number of choices divided by 10 (5 1st team, 5 2nd team)
    group_by(season) %>% mutate(pts_max=n()/10*2) %>%
    group_by(season,player,pts_max) %>%
    summarize(pts_won=sum(points_given),
              x1st_tm=sum(points_given==2),
              x2nd_tm=sum(points_given==1)) %>%
    ungroup() %>% mutate(share=pts_won/pts_max)

  all_d_vote_shares_until_2013=read_excel("Data/All-Defense Voting < 2013.xlsx")

  all_d_vote_shares_since_1991=bind_rows(all_d_vote_shares_since_2014,
                                         all_d_vote_shares_until_2013) %>%
    select(season,player,all_defense_share=share)

  all_nba_voting_shares=read_csv("Data/End of Season Teams (Voting).csv") %>%
    filter(type=="All-NBA") %>%
    select(season,player,seas_id,player_id,all_nba_share=share)
```

I decided to completely revamp the free agency training set. Rather than scrape the Basketball-Reference free agents tracker, I decided to scrape the Pro Sports Transaction Archive. The benefits to this decision were twofold:

- the Basketball-Reference free agents tracker only goes back to the 2016 offseason. The Pro Sports Transaction Archive allows me to add free agency periods before then.
- Basketball-Reference would mostly include players that were on team rosters on the final day of the regular season, excluding players who were waived during the season.

I started the new training set from the 2012 offseason, which was the first offseason of the new collective bargaining agreement. I kept the logic fairly similar from the previous training set. Players who signed from overseas and players who retired before the start of the next season were not included as free agents. The former would have no statistical data to pull, and the latter would artificially inflate the number of players who didn't sign a contract. Players whose salary & contract years were set to zero either went overseas, had explicitly non-guaranteed first years in their contracts (training camp deals, two ways, ten days, exhibit 10s), or didn't end up signing a contract before the season started. Option years & partially guaranteed years were included in the calculation of contract years; I looked at it as both player & team intending to see out the contract. The majority of year 1 salaries were gathered using Spotrac. Other minor sources include Basketball-Insiders, Basketball-Reference and Patricia Bender.

A couple of new wrinkles while expanding the training set:

- some players would sign multiple contracts, either due to failing a physical and voiding a contract, or converting from a non-guaranteed contract to a guaranteed one. I decided to take the first contract signed, as future contracts would have been based on additional information.

- how to handle restricted free agents whose offers were rescinded or not even offered? I could take the status as of contract signing (which would decrease the amount of RFA data to train on) or I could take the status as of the start of free agency (which could be misleading, as the player might not have gotten the same contract as an RFA). I decided on the former.

```r
past_free_agents<-read_csv("Data/Past Free Agents.csv")
```

The next file I used was salary cap history, scraped from RealGM. To somewhat normalize comparisons across years, I converted the first year salary to a percentage of the salary cap.

```r
#subtract one from year to match up with offseason in which contract was signed
salary_cap_hist<-read_csv("Data/Salary Cap History.csv") %>% mutate(season=season-1)
current_salary_cap=salary_cap_hist %>% filter(season==2023) %>% pull()
#create variable of first year salary as percentage of cap
#easier to compare across years
past_free_agents<-past_free_agents %>% select(-c(terms,Source)) %>%
  left_join(.,salary_cap_hist) %>%
  mutate(first_year_percent_of_cap=yr_1_salary/cap) %>%
  select(-c(yr_1_salary,cap))
```

Quick question, who do you think has the highest recorded salary cap percentage in the dataset? If you chose one of LeBron James, Kevin Durant or Stephen Curry, you'd probably be surprised to find out that you're incorrect! The highest percentage belongs to Carmelo Anthony, who re-signed with the New York Knicks in 2014 for a first-year salary cap percentage of 35.61%. Due to an exception that "the maximum salary in the first season of a contract is never less than 105% of the salary in the last year of the player's previous contract", Carmelo was able to surpass the 35% league-wide maximum salary.

The last file loaded is our evaluation set: the 2023 free agent class, retrieved from Spotrac. I had to edit this dataset to match the Basketball-Reference names (mainly adding diacritics to European names). In addition, I filtered out players with options. Players who decline player options and players who have their team options declined with more than 3 years of experience become unrestricted free agents. Players with less than or equal to 3 years of experience and a declined team option become restricted free agents. I'll use this fact to see which players & teams might decline their option.

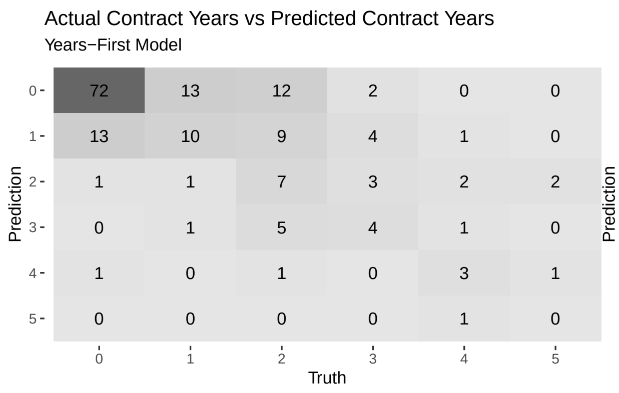
```r
current_fa<-read_csv("Data/Free Agents 2023.csv")
#separate out options to compare what players options get if declined
current_fa_options<-current_fa %>% filter(str_detect(type,"Player|Club")) %>%
  select(-c(experience,contract_yrs)) %>%
  rename(option_type=type,option_amt=first_year_percent_of_cap)
#make player options all declined (UFA's)
#make club options ufa or rfa depending on exp
current_fa<-current_fa %>%
  mutate(type=case_when((type=="Player"|(type=="Club" & experience >= 4))~"UFA",
                        (type=="Club" & experience < 4)~"RFA",
                        TRUE~type)) %>%
  group_by(player) %>% select(-experience) %>% slice(1) %>% ungroup() %>%
  mutate(first_year_percent_of_cap=NA)
```

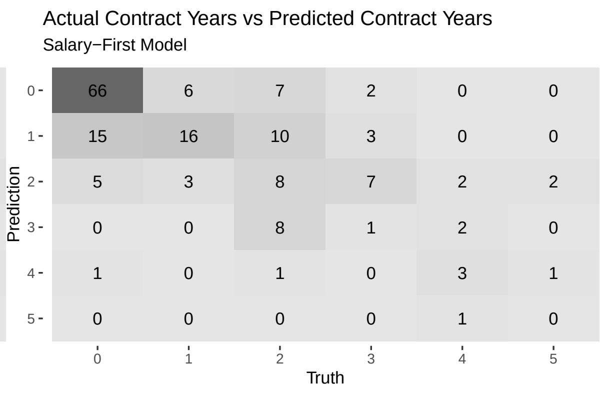
In the GitHub repository where this project is located, a file called `free agents.r` has more details on exactly how I scraped the train set, evaluation set and the salary cap history.

Retrospective on Last Year's Results

Before getting into pre-processing, we'll take a look at last year's results and see how the models performed.

The years accuracy of the years-first model was 56.47%, while the years accuracy of the salary-first model was 55.29%. The 2020 models were at 49-51% accuracy, but the 2021 models were at 58-60% accuracy. There's been a decline in accuracy after the initial improvement in 2021. Here's some confusion matrices on how each model handled the prediction of contract years.





The incorrect predictions were pessimistic, in that they skewed toward predicting less years than received as evidenced by the sum of the upper triangle being greater than the lower triangle (models forecasting no contract for players who received a one-year contract, one year for players who got two years, etc).

Here are the worst misses for both models.

player	yrs_y1s2	contract_yrs
Miles Bridges	4	0
Bradley Beal	2	5
Luguentz Dort	2	5
Collin Sexton	1	4
Vlatko Čančar	0	3
Sam Hauser	0	3

player	yrs_s1y2	contract_yrs
Miles Bridges	4	0
Bradley Beal	2	5
Luguentz Dort	2	5
Vlatko Čančar	0	3
Sam Hauser	0	3

On June 29, 2022 (one day before NBA free agency opened), Bridges was arrested in Los Angeles for felony domestic violence and was released on $130,000 bond. Beal was the main reason I decided to bring in end-of-season team voting shares, because I immediately knew he would be getting more than 2 years in a contract offer. Čančar and Hauser re-signed with the Nuggets and Celtics respectively.

Let's shift our focus to the salary predictions. First, the residual mean squared error of the years-first model was 0.0340088, while the salary-first model had an RMSE of 0.0322795. However, this includes Miles Bridges not receiving a contract when he was predicted for 23.55% of the salary cap under Y1S2 and 21.92% of the salary cap under S1Y2.

Removing Bridges, the new years-first RMSE is 0.0289011, while the new salary-first RMSE is 0.0276373. Last year's Y1 RMSE was 0.0245718 and S1 RMSE was 0.0248204. Even excluding the outlier of Bridges, the RMSEs are greater.

As we did with the years models, let's look at the most extreme salary misses.

player	y1s2_cap_percent	first_year_percent_of_cap
Bradley Beal	13.15%	35.00%
Anfernee Simons	7.69%	18.05%
Deandre Ayton	17.08%	25.00%
Victor Oladipo	0.00%	7.08%
Montrezl Harrell	8.99%	1.99%
Marvin Bagley III	3.21%	10.11%
Collin Sexton	6.51%	13.34%
Gary Harris	4.69%	10.51%
Zach LaVine	24.66%	30.00%
Jusuf Nurkić	7.36%	12.64%

player	s1y2_cap_percent	first_year_percent_of_cap
Bradley Beal	15.13%	35.00%
Anfernee Simons	9.13%	18.05%

Deandre Ayton	16.23%	25.00%
Montrezl Harrell	10.67%	1.99%
Carmelo Anthony	7.16%	0.00%
Dennis Schröder	9.11%	2.14%
LaMarcus Aldridge	6.50%	0.00%
Hassan Whiteside	6.15%	0.00%
Marvin Bagley III	4.04%	10.11%
Jalen Brunson	16.77%	22.43%

Common players missed on both models are Bradley Beal, Anfernee Simons, Deandre Ayton, Montrezl Harrell & Marvin Bagley III. Simons, Ayton & Bagley were all restricted free agents. Simons re-upped with the Trail Blazers after stepping into the scoring void left by CJ McCollum in his trade to the Pelicans in February 2022. Ayton signed an offer sheet with the Pacers that was subsequently matched by the Suns. Bagley was never able to live up to the billing of being the number 2 overall pick in 2019, especially with the 3 picks immediately after him being Luka Doncic (4x First-Team All-NBA, 4x All-Star), Jaren Jackson Jr. (2x First-Team All-Defense, 1x Defensive Player of the Year, 1x All-Star) and Trae Young (1x Third-Team All-NBA, 2x All-Star). He was traded from the Kings to the Pistons in January 2022, and re-signed with the Pistons in the offseason. Harrell signed with Philadelphia to back up eventual 2023 Most Valuable Player Joel Embiid and pursue a championship.

On a more positive note, here's some players on which the models were very close on. We'll restrict our view to contracts that had a first year salary that was greater than 5% of the salary cap, as it's easier to get close to minimum & near-minimum contract amounts.

player	y1s2_cap_percent	first_year_percent_of_cap
Bruce Brown	5.26%	5.24%
Malik Monk	7.77%	7.66%
Delon Wright	6.70%	6.31%
Isaiah Hartenstein	5.82%	6.31%
Chris Boucher	9.36%	10.26%
Jae'Sean Tate	6.70%	5.71%
Patty Mills	6.39%	5.24%
P.J. Tucker	7.16%	8.48%
Nic Claxton	5.39%	6.87%
Cody Martin	7.41%	5.66%

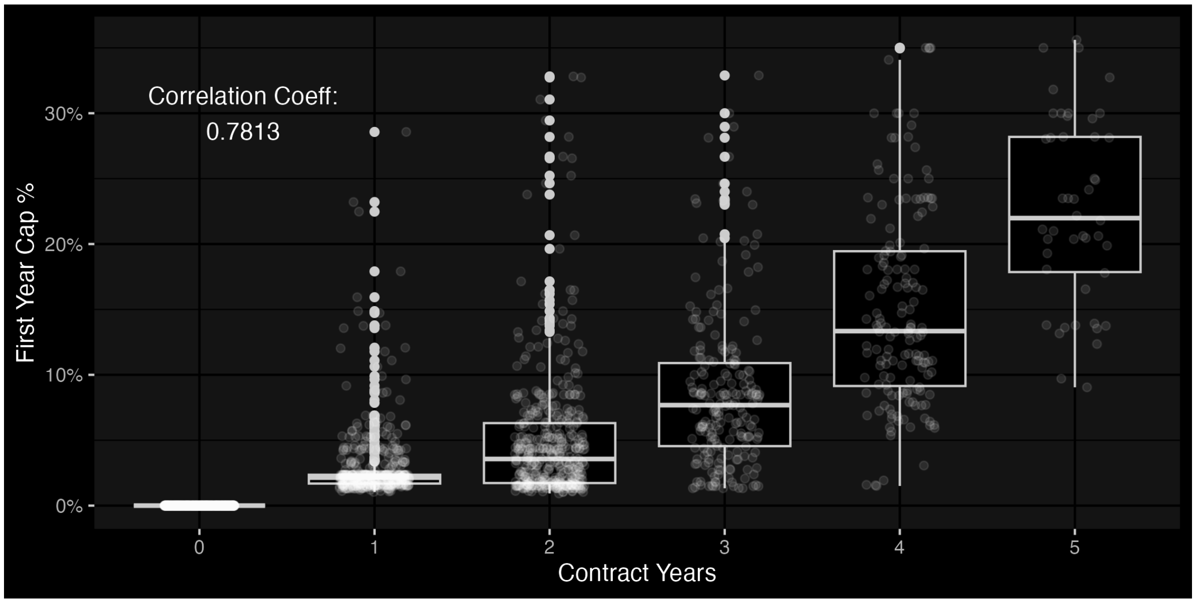
player	s1y2_cap_percent	first_year_percent_of_cap
Kyle Anderson	7.10%	7.10%
P.J. Tucker	8.44%	8.48%

Isaiah Hartenstein	6.59%	6.31%
Joe Ingles	5.97%	5.24%
Gary Payton II	5.98%	6.71%
Delon Wright	7.22%	6.31%
Jae'Sean Tate	6.64%	5.71%
Cody Martin	6.62%	5.66%
Mo Bamba	9.52%	8.33%
Nicolas Batum	7.45%	8.77%

Kyle Anderson is the first instance in the 3 years of predictions of an exact match with the actual salary percent. Players on both top 10s are P.J. Tucker, Delon Wright, Isaiah Hartenstein, Jae'Sean Tate and Cody Martin. Only one of the top 10 closest salary hits exceeds 10% as an actual first year salary percentage.

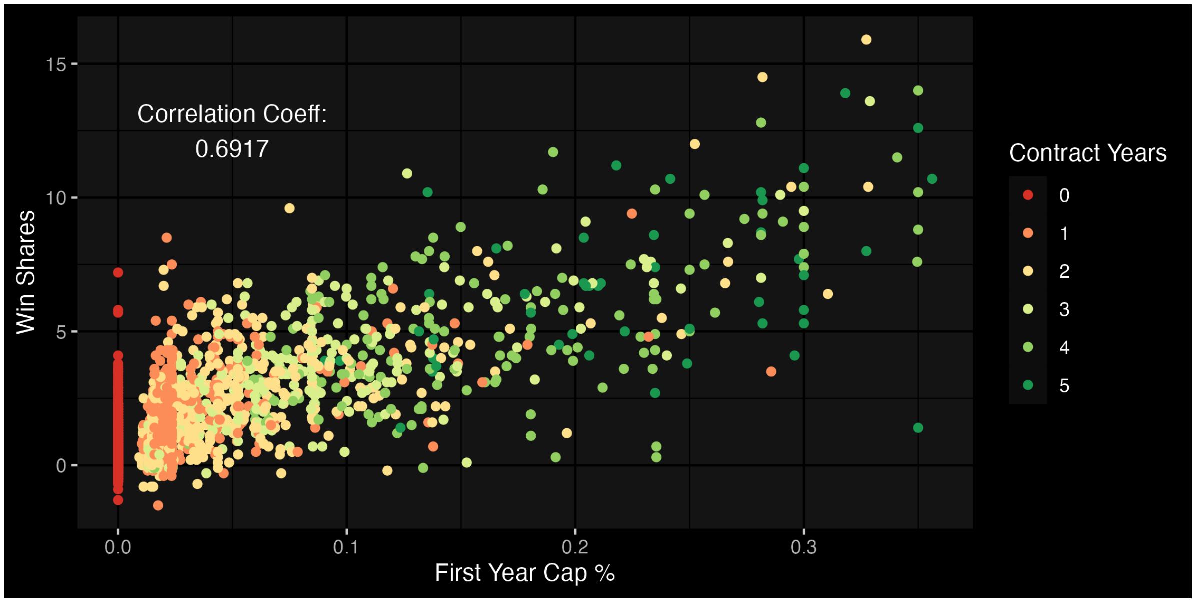
Data Exploration and Visualizations

Before we train models, let's see how some of the predictors and some of the targets interact. First, let's see how our targets correlate with each other. I've created a box and whisker plot, as well as added the points themselves in a transparent layer with some random variation to differentiate between points.



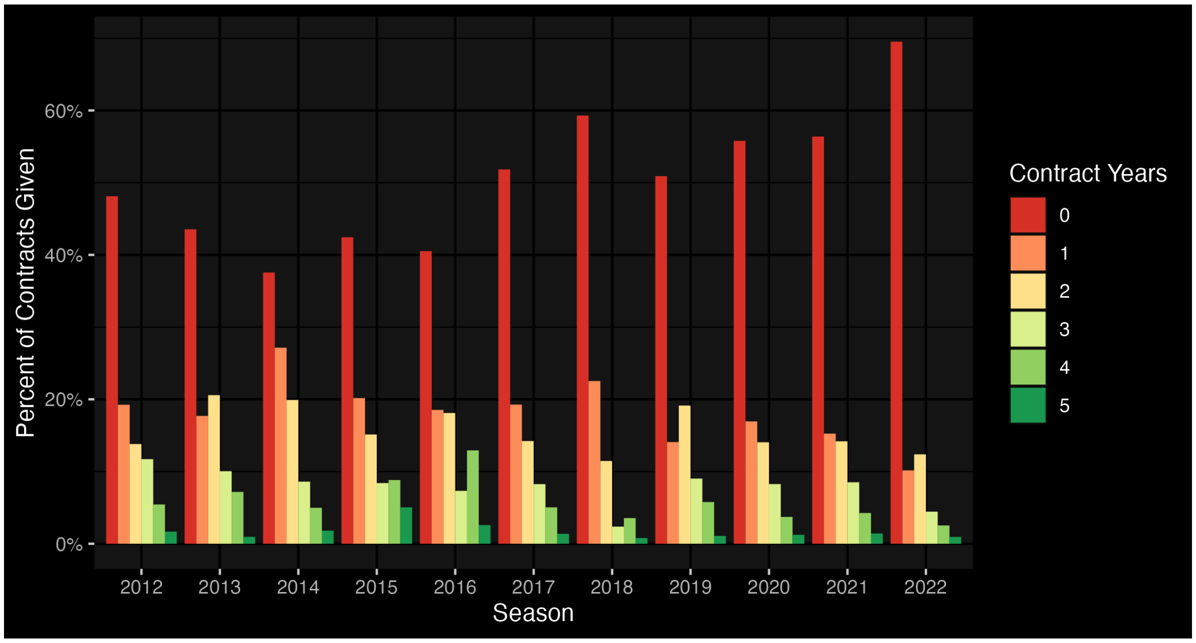
The correlation coefficient is 78.13%, which shows that the two targets are strongly and positively correlated. The median value of the first year cap % (middle line in each box) increases with an increase in contract length. The highest increase in first year cap % is between a 4-year and 5-year contract.

Next, let's see if an all-encompassing advanced statistic has a relationship with first year cap percentage. Win Shares represent how much a player has contributed to his team's wins by comparing his output to a marginal player. Higher win shares generally indicate a better player.



With a correlation coefficient of 69.17%, win shares are highly correlated with first year cap percentage. This shouldn't be too groundbreaking: better players get paid more. The player with the highest Win Shares to not get a contract is Miles Bridges in 2022 at 7.2, who we covered earlier. The players with the lowest win shares to get a five-year contract are Bradley Beal & Luguentz Dort in 2022 with 1.4 Win Shares (Beal got a maximum contract at 35% of the salary cap, while Dort received 12.4%).

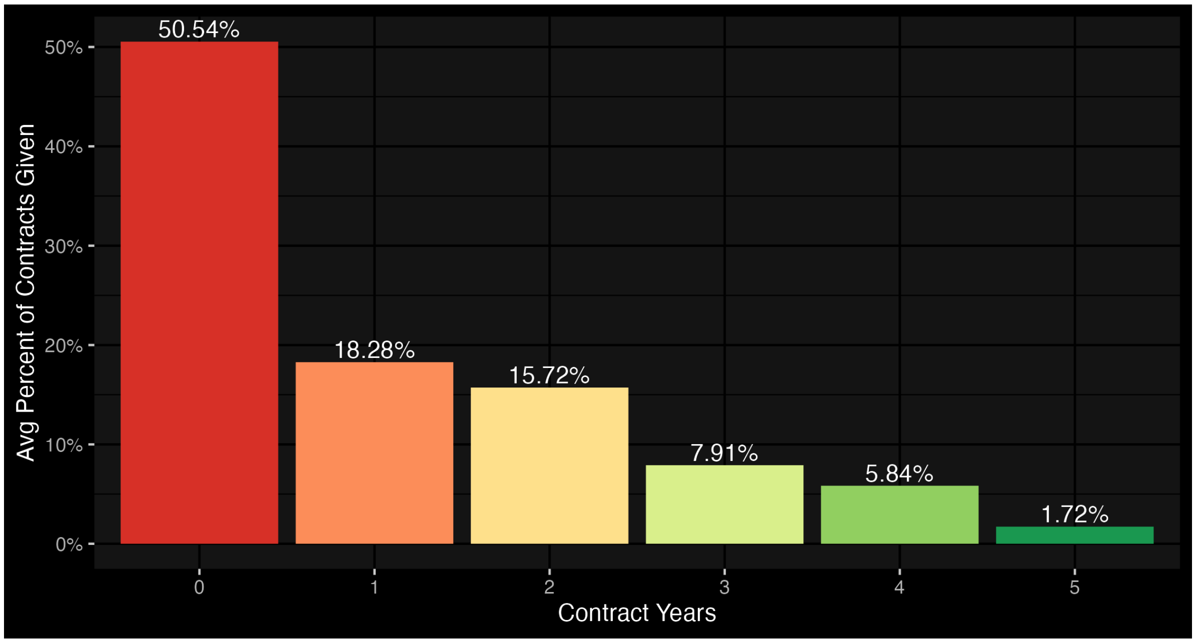
Finally, let's see how many contracts of each length were given in each offseason.



It comes as no surprise that as contract length increases, the percent of contracts of that length given out decreases. 2012, 2014, 2017, 2020 & 2021 all follow this descending pattern. There are some outliers:

- 4-year contracts almost doubling 3-year contracts in the reckless spending offseason of 2016 (the year of the cap spike, when the salary cap jumped from \$70 million to \$94 million)
- 2-year contracts outnumbering 1-year contracts in 2013, 2019 and 2022

Compiling all the seasonal data in the previous graph, we see an average offseason has 50% of free agents not signing a contract. The biggest drop is from 2-year contracts to 3-year contracts.



Pre-Processing

Pre-Processing Stats & Vote Shares

I used regular season stats, although I do understand that some players get paid on the strength of playoff performance. I started off with contract year stats, because there's anecdotal evidence that players exert more effort in their contract year (*cough cough Hassan Whiteside*). All stats except for the advanced stats (OWS, DWS and VORP) were converted to per game. Percentages were left alone.

In addition to using contract year stats and vote shares, I summed the past two years and the contract year.

Why I settled on 3 years:

- Players do get paid on past performance, so just using contract year stats was out of the question
- 2 years opens up the possibility of a fluke year
  - Kawhi would have his nine game 2018 season bring down his averages significantly from his 2019 season with the Raptors: adding another year somewhat lessens this effect
- On the other hand, it's quite unlikely that teams factor in stats from more than 4 years ago, a lot would have changed

    - the Celtics didn't pay Blake Griffin to recapture his form of his 2019 All-Star year in Detroit (I would hope)

- Another reason I settled on 3 years is that I can keep the same model for restricted free agents
    - my thought is that the rookie year is a bonus: great if you did well, but doesn't matter in the grand scheme of things if you did poorly
    - rookie extension is more based on how you improved over the course of that initial contract
    - For example, if Ja Morant had a worse rookie year but had the same level of play that he has achieved in his second and third year (as well as next year), I highly doubt that Memphis would have offered him a significantly less amount of money due to that substandard rookie year

I performed the same processing on the three-year totals, using the three-year game total as the denominator for converting to per game. I had to calculate the three-year percentages, and also re-engineered the win shares per 48 minutes metric.

I removed categories that were linear combinations of one another. For example, total rebounds can be found by simply adding up offensive and defensive rebounds, and points are just the result of 2*(number of 2 point field goals made) & 3*(number of 3 point field goals made). I kept age and experience as predictor variables, but removed position because I felt it would ultimately reflect in the stats themselves.

I divided the three-year versions of games played percentage, games started percentage, All-NBA voting share and All-Defense voting share by 3 to rescale the columns back to a 0-1 range. For example, James Harden received a 100% vote share in 3 consecutive seasons from 2017-2019. Rather than showing all_nba_share_last_3_yrs as 1+1+1=3 in his 2019 row, dividing by 3 gives us the more easily understood 100% of possible points received in the last 3 years.

The final step was to replace missing values in the shooting percentages with zeroes. These NA's were originally due to lack of attempts.

```r
stats_and_shares=advanced_and_totals %>%
  left_join(.,all_nba_voting_shares) %>%
  left_join(.,all_d_vote_shares_since_1991) %>%
  replace_na(list(all_nba_share=0,all_defense_share=0))
three_year_rolling_stats_and_shares=stats_and_shares %>% group_by(player_id) %>%
  #three year sum
  mutate(across(-c(1:9,fg_percent,x3p_percent,
                  x2p_percent:e_fg_percent,ft_percent),
                list(three_yrs=~rollapplyr(.,3,sum,partial=TRUE)),
                .names="{col}_last_3_yrs")) %>%
  mutate(ws_per_48_last_3_yrs=ws_last_3_yrs/mp_last_3_yrs*48) %>%
  mutate(fg_percent=ifelse(fga==0,0,fg/fga),
         x3p_percent=ifelse(x3pa==0,0,x3p/x3pa),
         x2p_percent=ifelse(x2pa==0,0,x2p/x2pa),
         e_fg_percent=ifelse(fga==0,0,(fg+0.5*x3p)/fga),
         ft_percent=ifelse(fta==0,0,ft/fta)) %>%
  mutate(fg_percent_last_3_yrs=
           ifelse(fga_last_3_yrs==0,0,fg_last_3_yrs/fga_last_3_yrs),
         x3p_percent_last_3_yrs=
           ifelse(x3pa_last_3_yrs==0,0,x3p_last_3_yrs/x3pa_last_3_yrs),
         x2p_percent_last_3_yrs=
           ifelse(x2pa_last_3_yrs==0,0,x2p_last_3_yrs/x2pa_last_3_yrs),
         e_fg_percent_last_3_yrs=
           ifelse(fga_last_3_yrs==0,0,
                  (fg_last_3_yrs+0.5*x3p_last_3_yrs)/fga_last_3_yrs),
         ft_percent_last_3_yrs=
           ifelse(fta_last_3_yrs==0,0,ft_last_3_yrs/fta_last_3_yrs)) %>%
  #remove categories that aren't predictive vars or linear combo of others
  select(-c(lg,pos,birth_year,tm,
            trb,trb_last_3_yrs,
            fg,fga,fg_last_3_yrs,fga_last_3_yrs,
            pts,pts_last_3_yrs)) %>%
  #convert contract year and last 3 year stats to per game (except games)
  mutate(across(c(mp,x3p:x3pa,x2p:x2pa,ft:fta,orb:pf),list(per_game=~./g)),
         .after="gs_percent") %>%
  select(-c(g,mp,x3p:x3pa,x2p:x2pa,ft:fta,orb:pf,ws)) %>%
  mutate(across(mp_last_3_yrs:pf_last_3_yrs,list(per_game=~./g_last_3_yrs)),
         .after="gs_percent_last_3_yrs") %>%
  select(-c(g_last_3_yrs,mp_last_3_yrs:pf_last_3_yrs,ws_last_3_yrs)) %>%
  ungroup() %>%
  #rescale games percentages & vote shares over 3 years back to 0-1
  mutate(across(c(g_percent_last_3_yrs:gs_percent_last_3_yrs,
                  all_nba_share_last_3_yrs,
                  all_defense_share_last_3_yrs),~./3)) %>%
  replace_na(list(fg_percent=0,x3p_percent=0,x2p_percent=0,
                  e_fg_percent=0,ft_percent=0))
```

Pre-Processing Positions

I took a three-year rolling sum of minutes played for consistency with the previous stats pre-processing, and converted the totals back to percents.

With 3-year positional percentages in hand, it was time to assign the actual positions. Some players play one position almost exclusively, while other players are more flexible and alternate between positions. I set the baseline for a player to be considered at a "pure position" at 75%: if the player played at any position more than 75% of the time, they were deemed to be that position.

All other players were bucketed into combo positions based on the maximum of the following sums of two traditional positions:

- combo guard (point guard/shooting guard)
- small wing (shooting guard/small forward)
- big wing (small forward/power forward)
- big man (power forward/center)

Some players had multiple combo positions listed due to a small amount of minutes played and two players had no positions listed due to playing less than a full minute, so those players were added manually.

```r
pbp_last_three_years=pbp_pos_mins %>% group_by(player_id) %>%
  #rolling 3 year sum of minutes played total & position
  mutate(across(mp_summed:c_mp_summed,
                list(three_yrs=~rollapplyr(.,3,sum,partial=TRUE)),
                .names="{col}_last_3_yrs")) %>% ungroup() %>%
  select(-c(mp_summed:c_mp_summed)) %>%
  #convert totals back to percents
  mutate(across(pg_mp_summed_last_3_yrs:c_mp_summed_last_3_yrs,
                ~./mp_summed_last_3_yrs)) %>%
  rename_with(.fn=~gsub(x = ., pattern = "_mp", replacement = "_percent"),
              .cols=pg_mp_summed_last_3_yrs:c_mp_summed_last_3_yrs) %>%
  select(-mp_summed_last_3_yrs)

#assign pure positions to players with 75% of time at one position
pure_position=pbp_last_three_years %>%
  pivot_longer(.,cols=c(pg_percent_summed_last_3_yrs:c_percent_summed_last_3_yrs),
               names_to = "pos",values_to = "percent_at_pos") %>%
  group_by(seas_id) %>% slice_max(percent_at_pos) %>% ungroup() %>%
  filter(percent_at_pos>0.75) %>% mutate(pos=word(pos,1,sep="_"))

combo_position=
  #remove players with pure positions
  anti_join(pbp_last_three_years,pure_position %>% select(1:4)) %>%
  #create buckets of in-between positions
  mutate(combo_guard=pg_percent_summed_last_3_yrs+sg_percent_summed_last_3_yrs,
         small_wing=sg_percent_summed_last_3_yrs+sf_percent_summed_last_3_yrs,
         big_wing=sf_percent_summed_last_3_yrs+pf_percent_summed_last_3_yrs,
         big_man=pf_percent_summed_last_3_yrs+c_percent_summed_last_3_yrs) %>%
  select(-c(pg_percent_summed_last_3_yrs:c_percent_summed_last_3_yrs)) %>%
  pivot_longer(.,cols=c(combo_guard:big_man),
               names_to = "pos",values_to = "percent_at_pos") %>%
  group_by(seas_id) %>% slice_max(percent_at_pos) %>% ungroup() %>%
  #4 players had 2 separate combo positions listed
  filter(!(player=="Terrance Roberson" & season==2001 & pos=="big_wing") &
          !(player=="Ty Jerome" & season==2020 & pos=="small_wing") &
          !(player=="Anthony Gill" & season==2021 & pos=="big_man") &
          !(player=="Chris Duarte" & season==2022 & pos=="combo_guard") &
          !(player=="Jordan Hall" & season==2023 & pos=="combo_guard"))

all_player_pos=bind_rows(pure_position,combo_position) %>%
  #2 players had 0 MP, so they were lost in both combo & pure
  add_row(seas_id=19914,season=2006,player_id=3589,
          player="Alex Scales",pos="sg",percent_at_pos=1) %>%
  add_row(seas_id=22403,season=2010,player_id=3882,
          player="JamesOn Curry",pos="pg",percent_at_pos=1) %>%
  filter(season>2009) %>% select(-c(seas_id,percent_at_pos)) %>%
  mutate(pos_group=case_when(pos %in% c("pg","sg","combo_guard")~"guard",
                             pos %in% c("sf","small_wing","big_wing")~"wing",
                             pos %in% c("pf","c","big_man")~"big"))
```

Combining Pre-Processing Dataframes

I joined the positional data and grouped by season and positional group to get the percentage of VORP a player has contributed to their position as a proxy for positional scarcity. While a player may not have had a raw high VORP compared to all offseasons, they could have a significant proportion of their positional group's VORP in that specific offseason and possibly induce a bidding war between teams due to unappetizing other options.

I changed contract years from a numeric column to a factor/category column. This changes its prediction from a regression problem to a classification problem. A 2.5-year contract doesn't make much sense, so it is in our best interest to discretize the years and store them as factors rather than round a regression result.

```r
train_set=inner_join(three_year_rolling_stats_and_shares,past_free_agents) %>%
  #multiple free agents with same name in same season
  filter(!(player=="Tony Mitchell" & seas_id==25056)) %>%
  filter(!(player=="Chris Johnson" & seas_id==23991)) %>%
  left_join(.,all_player_pos) %>% group_by(season,pos_group) %>%
  mutate(position_vorp=sum(vorp_last_3_yrs)) %>% ungroup() %>%
  mutate(percent_of_pos_vorp=vorp_last_3_yrs/position_vorp) %>%
  mutate(contract_yrs=factor(contract_yrs,levels = 0:5)) %>%
  select(-c(pos,pos_group,position_vorp)) %>%
  mutate(across(-c(seas_id:experience,type:contract_yrs),~round(.,digits=4)))
write_csv(train_set,"Data/Train Set.csv")
```

Training Models

There is no need for a subset of the training set to be withheld as a test set before running the models on the evaluation set, because there is built-in cross validation. In the first iteration of this project, I utilized leave-one-out cross validation since the dataset is relatively small (<1000 observations). How this works is that the model is run excluding one observation. Then, the model attempts to predict the result of that excluded observation. This is repeated for every observation. However, on subsequent runs as the dataset has grown, I've decided to use k-fold cross validation. I've achieved even better results while cutting down significantly on training time.

As we saw in data visualization, the two target variables (contract years and first year salary as a percentage of the salary cap) are fairly well correlated, as they have a Pearson correlation coefficient of 0.7813. The way I chose to handle this is:

- predict one target first without the other as a predictor
- choose the best model (be that a single model or an ensemble of multiple models)
- use the first target's predictions as an input to predict the second target

One potential problem is compounding errors. If there's an incorrect year prediction, it might lead to an incorrect salary prediction. Initially, to alleviate this problem, I thought it would be sufficient to utilize tidymodels' ability to run multivariate models with more than one outcome. However, I realized that tuning wasn't yet implemented with multi-output regression, and I'm uneasy about choosing arbitrary parameters. An additional (minor) drawback is the outcomes must be of the same type, so we would undo the contract years conversion done in the last section.

The Models

I used a total of six models.

- linear regression model as a baseline for salary, and multinomial regression as a baseline for years
  - the separation is due to the classification/regression split
- k-nearest neighbors model: take the distance between the statistics of two players (the absolute value of the difference) and then take the average of the outcome variable of the k nearest neighbours
  - the intuition being that similar players get similar contracts
- decision tree model (`rpart`): maybe as a player passes certain statistical thresholds, their contract increases
  - only using for predicting the contract years; since there are so many different salary percentages, a solitary decision tree would either be useless or far too complicated
- random forest model (`ranger`): reduces instability by averaging multiple trees
  - costs interpretability as there is no tree diagram that is representative of the decisions made
- support vector machine model: attempt to separate classes with a hyperplane
  - support vectors are the points closest to the hyperplane, named as such because the hyperplane would change if those points were removed
  - I believe the following image from Wikipedia succinctly explains an SVM

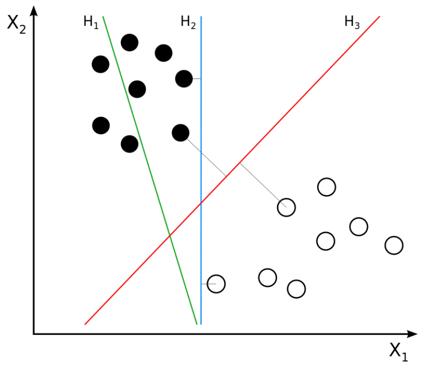


Figure 1: H1 does not separate the classes. H2 does, but only with a small margin. H3 separates them with the maximal margin. By User:ZackWeinberg, based on PNG version by User:Cyc - This file was derived from: Svm separating hyperplanes.png, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=22877598

Predicting Years First, then Salary

We'll start by predicting years first and then salary with years as an input.

With caret, everything could be thrown into the train function: cross-validation, tuning, data, etc. Tidymodels is more explicit in its steps. The standard 10-fold cross validation is how the data is going to be resampled. In addition, we'll stratify selection, so there is sufficient data to predict all contract year lengths. A hypothetical (albeit unlikely) scenario that could occur without stratification is a fold could include all instances of restricted free agents (which are more rare) in the test portion, and none in the train portion. The chosen algorithm would be confused as to how to predict something for which it has no training.

Recipes is tidymodels' pre-processing package. The first step to change the roles of some variables. The season_id, season, player_id & player variables are of no use as predictors, but are useful to identify observations, so they are given the "id" role. Since we are predicting years first, the first_year_percent_of_cap variable will be removed. Finally, we will convert the free agent type column to a numeric column.

```
set.seed(100)
cv=vfold_cv(train_set,v=10,strata=type,repeats=5)
y1_recipe=recipe(contract_yrs~.,data=train_set) %>%
  update_role(seas_id:player,new_role="id") %>%
  step_rm(first_year_percent_of_cap) %>% step_dummy(type)
```

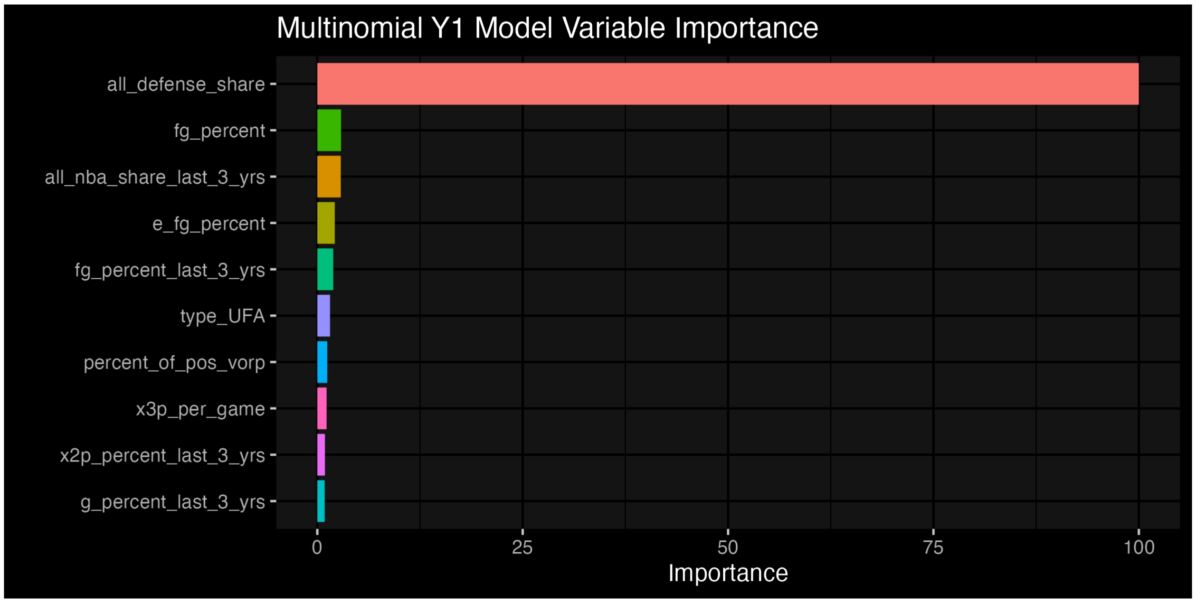
Initially, accuracy was chosen as the metric to determine the best submodel by cross-validation. However, with the inherent imbalance of the outcome classes, the F1 score is a better metric. As an extreme example, if there were only two classes with a 90:10 split, a classifier could achieve 90% accuracy by simply predicting the more populous class for every case. On the other hand, the F1 score attempts to minimize both false positives & false negatives. However, the default averaging for F1 is macro-averaging, which gives equal weights to all classes and is exactly the problem we were trying to distance ourselves from by not choosing accuracy. We need to change the averaging function to be macro-weighted. In order to include it within yardstick's metric_set, we have to create a new function wrapping the metric, set the options within the wrapper & formalize it as a new metric.

```
macro_weight_f1 <- function(data,truth,estimate,beta = 1,
                            estimator="macro_weighted",na_rm = TRUE,
                            event_level = yardstick_event_level(),...){
  f_meas(data=data,
         truth = !! rlang::enquo(truth),
         estimate = !! rlang::enquo(estimate),
         estimator = "macro_weighted",
         beta=beta,
         na_rm=na_rm,
         event_level=event_level,
         ...
         )
}
macro_weight_f1 <- new_class_metric(macro_weight_f1,"maximize")
```

First up is the multinomial regression model.

```
wf=workflow() %>% add_recipe(y1_recipe) %>%
  add_model(multinom_reg(penalty=0) %>%
               set_engine("glmnet") %>%
               set_mode("classification"))
tune_multinom=wf %>% tune_grid(resamples=cv,
                               metrics=metric_set(macro_weight_f1))
wf=wf %>% finalize_workflow(tune_multinom %>% select_best())
fit_y1_multinom=fit(wf,train_set)
```

Let's get the most important variables of the multinomial model.



The contract-year all-defense share dominates the variable importance, with the next best variable being less than 3% as important.

Next up is the k-nearest neighbors model.

```r
wf=workflow() %>% add_recipe(y1_recipe) %>%
  add_model(nearest_neighbor(neighbors = tune()) %>%
  set_engine("kknn") %>% set_mode("classification"))
tune_knn=wf %>% tune_grid(resamples=cv,
                          grid=expand.grid(neighbors=5:50),
                          metrics=metric_set(macro_weight_f1))
wf=wf %>% finalize_workflow(tune_knn %>% select_best())
fit_y1_knn=fit(wf,train_set)
tune_knn %>% select_best() %>% gt()
```

neighbors	.config
45	Preprocessor1_Model41

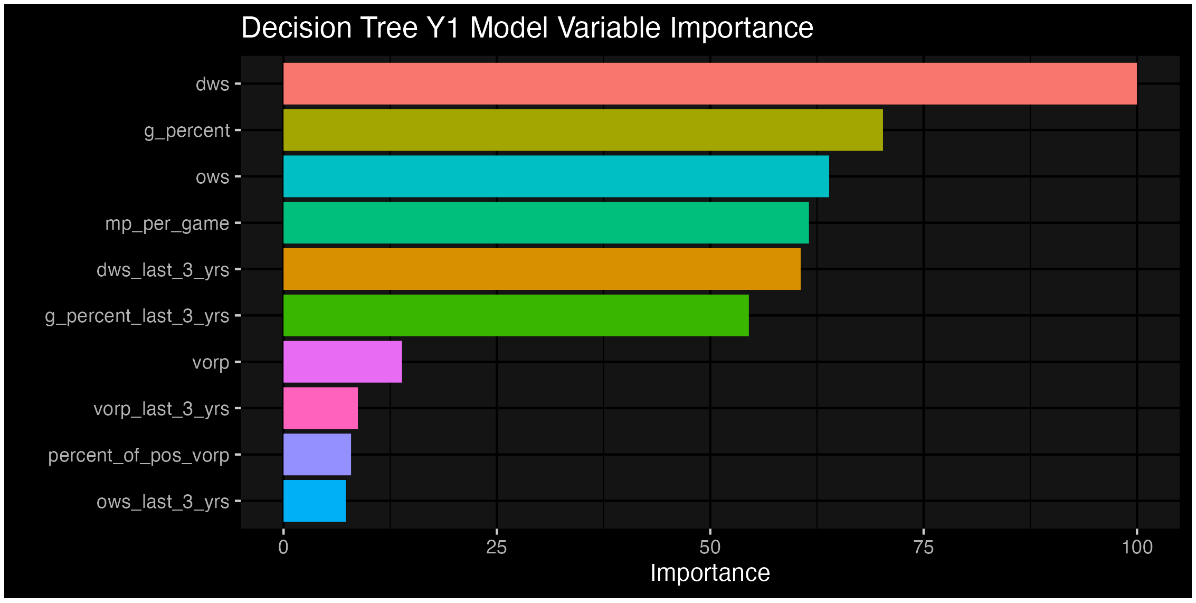
The best knn model is near the upper limit of the tuning grid, taking into account 45 of the closest comparable players.

The next model is the decision tree model. We can tune the cost complexity parameter (cp) in this model. CP is the minimum for how much the residual sum of squares must improve for another partition to be added. A CP that is too high will have too few branches, while a CP that is too low will be difficult to follow since there are many branches. We lean towards the lower end of the spectrum. We will also change the min_n, which is the minimum number of data points in a node that are required for the node to be split further. The default is 20, but the 5-year portion of the dataset is small, so we will decrease min_n to 1. Since there exists an element of randomness in choosing samples to model a decision tree, we need to set a seed to keep the work reproducible.

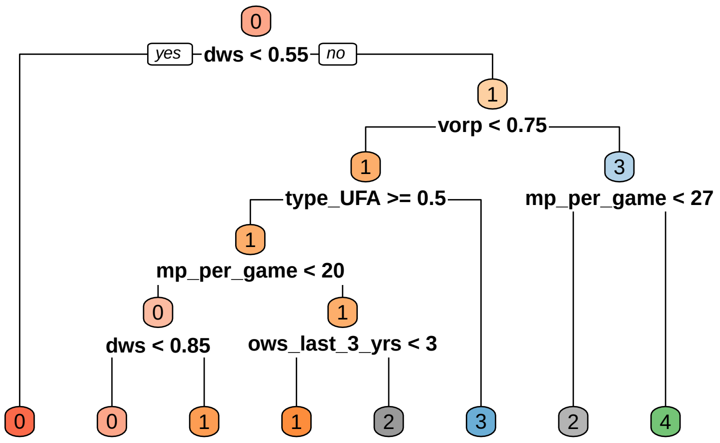
```r
set.seed(100,sample.kind = "Rounding")
wf=workflow() %>% add_recipe(y1_recipe) %>%
  add_model(decision_tree(cost_complexity = tune(),min_n=tune()) %>%
  set_engine("rpart") %>% set_mode("classification"))
tune_tree=wf %>% tune_grid(resamples=cv,
                          grid=expand.grid(cost_complexity=0.1^seq(2,5),
                                           min_n=seq(1:10)),
                          metrics=metric_set(macro_weight_f1))
wf=wf %>% finalize_workflow(tune_tree %>% select_best())
fit_y1_tree=fit(wf,train_set)
tune_tree %>% select_best() %>% gt()
```

cost_complexity	min_n	.config
0.01	1	Preprocessor1_Model01

Let's get the most important variables of the decision_tree model.



The individual contract-year components of the holistic advanced stat of win shares rank within the top-4 most important variables. Durability and availabilty are also highly desirable characteristics, as evidenced by the high values of contract-year minutes per game, contract-year games played percentage and last-3-years games played percentage.



The decision tree does not predict any 5-year contracts. The decision tree maximizes its prediction at 4 contract years when a player does all of the following:

- has defensive win shares above 0.55 in the contract year
- plays more than 27 minutes per game in the contract year
- has a value over replacement player above 0.75

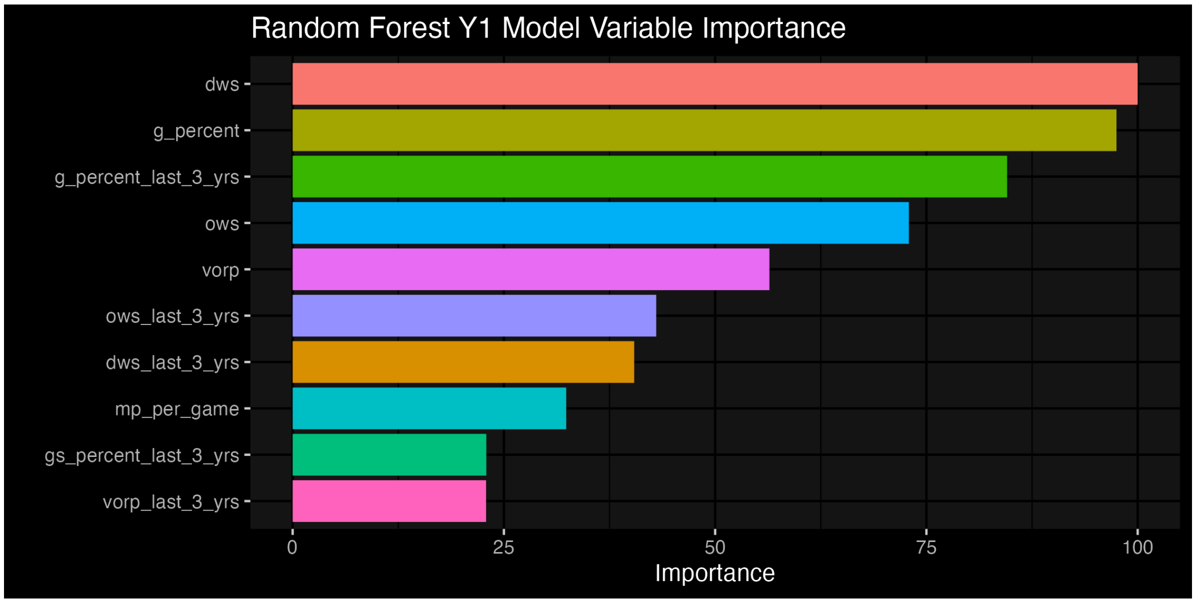
The next model is a random forest. Since `ranger` is a *random* forest algorithm, we need to set a seed to keep the work reproducible.

Random forest algorithms require an explicit call for variable importance, so we'll ask for permutation importance. A simplified explanation for permutation importance is shuffling a predictor's values and seeing how much the error increases. As a predictor's importance increases, it is difficult for the rest of the model to compute accurate predictions without it. There are 3 tuning parameters:

- trees is the number of decision trees to create
  - the default is 500, which we'll keep
- mtry is the number of variables to split at each decision node
  - the default is the rounded square root of the number of variables, which in this case would be `round(sqrt(55))`
  - we will try all integers between 5 & 10
- min_n is the same as the decision tree
  - the default is 1 for a classification problem like this, which we'll keep

```r
set.seed(100,sample.kind = "Rounding")
wf=workflow() %>% add_recipe(y1_recipe) %>%
  add_model(rand_forest(mtry = tune()) %>%
  set_engine("ranger",importance="permutation") %>% set_mode("classification"))
tune_forest=wf %>% tune_grid(resamples=cv,
                             grid=expand.grid(mtry=5:10),
                             metrics=metric_set(macro_weight_f1))
wf=wf %>% finalize_workflow(tune_forest %>% select_best())
fit_y1_forest=fit(wf,train_set)
tune_forest %>% select_best() %>% gt()
```

mtry	.config
10	Preprocessor1_Model6



Random forest variable importance has a more gradual decline than either decision tree or multinomial. The contract year versions of the holistic advanced stats are in the top 5 most important variables in the random forest model, with the other 2 spots taken by both versions of the games played percentage.

Finally, we train the support vector machine on the model. There are two tuning parameters:

- rbf_sigma: determines how well to fit the training data (higher=fit closer to training)
    - with a high sigma, every workaday big with middling stats might be predicted to get close to 2016 Mozgov money
- cost: tradeoff between smoothness of boundaries and correct classification
    - with a high cost, leads to too wiggly of a boundary, and might not generalize to test sets
    - tests using C=0.25, C=0.5 and C=1

```r
set.seed(100,sample.kind = "Rounding")
wf=workflow() %>% add_recipe(y1_recipe) %>%
  add_model(svm_rbf(rbf_sigma=tune(),cost=tune()) %>%
  set_engine("kernlab") %>% set_mode("classification"))
tune_svm=wf %>% tune_grid(resamples=cv,
                        grid=expand.grid(rbf_sigma=0.1^seq(2:5),
                                         cost=c(0.25,0.5,1)),
                        metrics=metric_set(macro_weight_f1))
```

```
wf=wf %>% finalize_workflow(tune_svm %>% select_best())
fit_y1_svm=fit(wf,train_set)
tune_svm %>% select_best() %>% gt()
```

cost	rbf_sigma	.config
0.25	1e-04	Preprocessor1_Model04

Unfortunately, there is no concept of variable importance for an SVM model.

Let's look at some performance metrics, and see which models we want to take along as inputs for predicting salary.

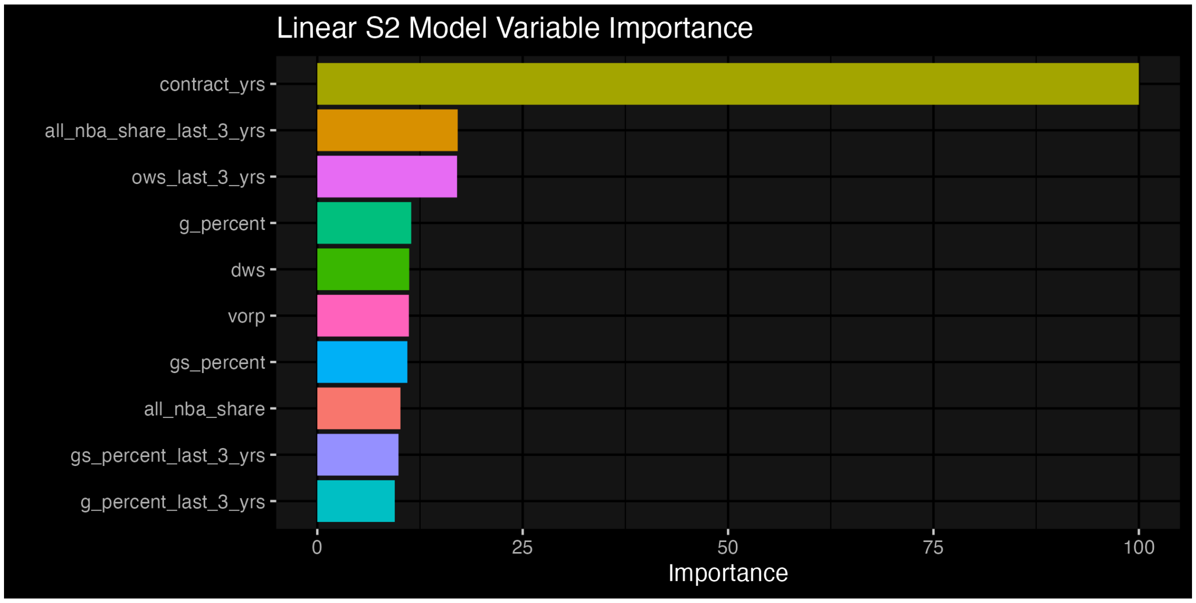
Method	Correct Predict %	Off by >1 Yr	Max Year Predicts	F1 Score
Multinomial	65.77%	12.58%	34	0.5869749
KNN	70.21%	11.63%	13	0.5683985
Decision Tree	61.56%	13.02%	0	0.5652467
Random Forest	97.32%	1.65%	46	0.5746079
SVM	51.39%	30.70%	0	0.6783399

The SVM has the highest F1 score, but the lowest accuracy. The random forest stands far above the rest with ~97% accuracy! In 2020's project, the random forests had the best performance as well, but had difficulty distinguishing 5-year contracts. Shifting to a classification problem has solved that. The multinomial model had significantly more max-year predictions than KNN as well as the highest F1-score barring the SVM, whereas KNN had ~4% better accuracy. While anticlimactic, I think the best course of action is to simply use the random forest model to make contract-year predictions. We'll convert the prediction column to use it as a numeric predictor.

```
train_set_after_y1=train_set %>% select(-contract_yrs) %>%
  bind_cols(contract_yrs=
              as.numeric(
                as.character(
                  predict(fit_y1_forest,new_data=train_set) %>% pull()
                  )
                )
            )
cv=vfold_cv(train_set_after_y1,v=10,strata=type,repeats=5)
s2_recipe=recipe(first_year_percent_of_cap~.,data=train_set_after_y1) %>%
  update_role(seas_id:player,new_role="id") %>% step_dummy(type)
```

Now I can run through the models for salary using the predicted years as an input. The models I won't reuse are the multinomial model and the decision tree. The multinomial model is for classification only, while we are attempting to predict a continuous outcome in the first year salary as a percentage of the salary cap. We'll sub in a linear regression model as our new baseline. Since there's so many different salary percentages, a decision tree model would be useless or far too complicated.

```r
wf=workflow() %>% add_recipe(s2_recipe) %>%
  add_model(linear_reg() %>%
              set_engine("lm") %>%
              set_mode("regression"))
tune_lin=wf %>% tune_grid(resamples=cv,metrics=metric_set(rmse,mae))
wf=wf %>% finalize_workflow(tune_lin %>% select_best(metric="rmse"))
fit_s2_lin=fit(wf,train_set_after_y1)
```



Contract years dwarf all other variables in terms of importance, with both versions of All-NBA share showing up in the top 10 (last-3-years showing up 2nd with contract-year farther down at number 8).

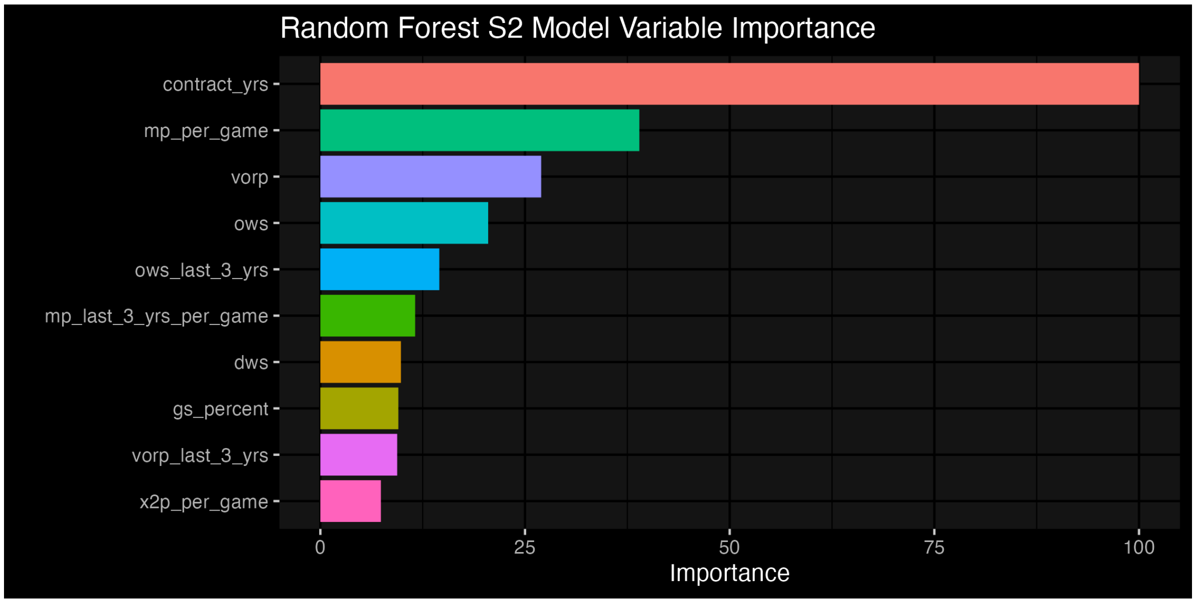
```r
wf=workflow() %>% add_recipe(s2_recipe) %>%
  add_model(nearest_neighbor(neighbors=tune()) %>%
              set_engine("kknn") %>%
              set_mode("regression"))
tune_knn=wf %>% tune_grid(resamples=cv,
                            grid=expand.grid(neighbors=5:50),
                            metrics=metric_set(rmse,mae))
wf=wf %>% finalize_workflow(tune_knn %>% select_best(metric="rmse"))
fit_s2_knn=fit(wf,train_set_after_y1)
tune_knn %>% select_best(metric="rmse") %>% gt()
```

neighbors	.config
24	Preprocessor1_Model20

The best knn salary model is around the middle of the tune grid, using 24 comparables.

```
set.seed(100,sample.kind = "Rounding")
wf=workflow() %>% add_recipe(s2_recipe) %>%
  add_model(rand_forest(mtry = tune()) %>%
  set_engine("ranger",importance="permutation") %>% set_mode("regression"))
tune_forest=wf %>% tune_grid(resamples=cv,
                            grid=expand.grid(mtry=5:10),
                            metrics=metric_set(rmse,mae))
wf=wf %>% finalize_workflow(tune_forest %>% select_best(metric="rmse"))
fit_s2_forest=fit(wf,train_set_after_y1)
tune_forest %>% select_best(metric="rmse") %>% gt()
```

mtry	.config
10	Preprocessor1_Model6



Like the linear model, contract years are by far the most important variable in the random forest model, although not to the same extent.

```
set.seed(100,sample.kind = "Rounding")
wf=workflow() %>% add_recipe(s2_recipe) %>%
  add_model(svm_rbf(rbf_sigma=tune(),cost=tune()) %>%
  set_engine("kernlab") %>% set_mode("regression"))
tune_svm=wf %>% tune_grid(resamples=cv,
                        grid=expand.grid(rbf_sigma=0.1^seq(2:5),
                                         cost=c(0.25,0.5,1)),
                        metrics=metric_set(rmse,mae))
wf=wf %>% finalize_workflow(tune_svm %>% select_best(metric="rmse"))
fit_s2_svm=fit(wf,train_set_after_y1)
tune_svm %>% select_best(metric="rmse") %>% gt()
```

cost	rbf_sigma	.config
1	0.001	Preprocessor1_Model11

Here are the performance metrics for the salary portion of the Y1S2 model.

Method	Off By >5%	Within 2%	MAE	RMSE
Linear	5.14%	76.67%	0.01539038	0.02424253
KNN	4.99%	83.31%	0.01432022	0.02589373

Random Forest	0.37%	94.83%	0.01198780	0.02274163
SVM	4.99%	82.65%	0.01262624	0.02348583

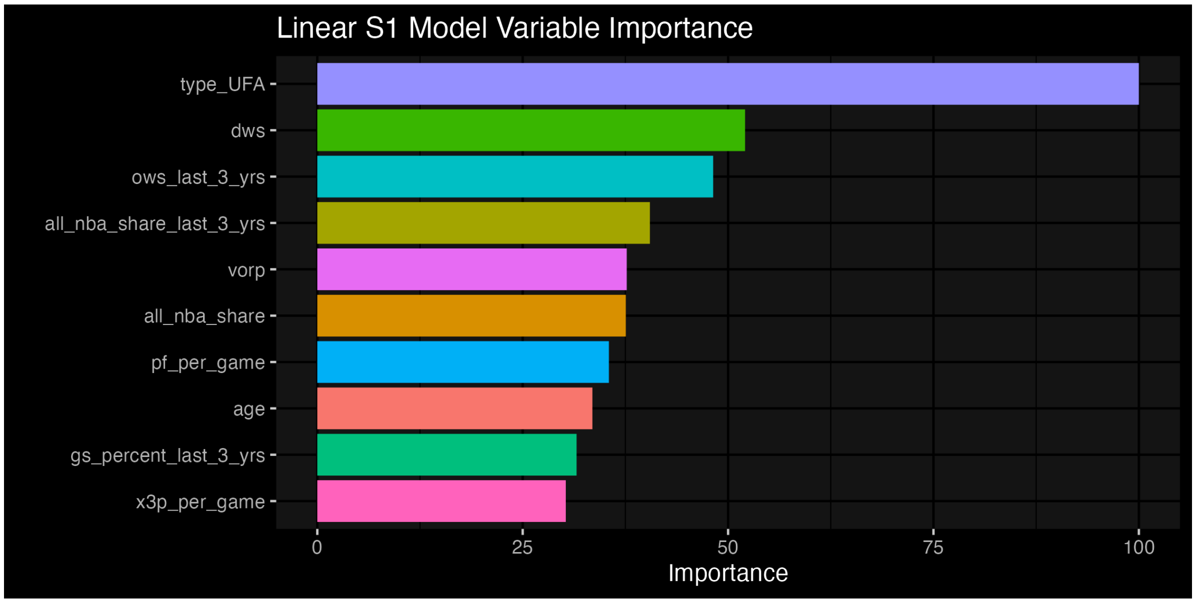
Mean absolute error is the measure of the average difference between forecasts, while the residual mean squared error penalizes large errors. The random forest has the best RMSE, and also tops the leaderboard of maximizing predictions within 2% of actual value, and minimizing predictions that are more than 5% away. The KNN model has the worst RMSE, but still performs better on predictions than the linear model and the SVM model. With the SVM & random forest being so close in traditional metrics as well as the SVM being no slouch on dataset-specific metrics, we will take the mean of the SVM & random forest as our salary prediction in the Y1S2 model.

Predicting Salary First, then Years

Now I'll switch up the order, predicting salary first and then years.

```r
set.seed(100)
cv=vfold_cv(train_set,v=10,strata=type,repeats=5)
s1_recipe=recipe(first_year_percent_of_cap~.,data=train_set) %>%
  update_role(seas_id:player,new_role="id") %>%
  step_rm(contract_yrs) %>% step_dummy(type)
```

```r
wf=workflow() %>% add_recipe(s1_recipe) %>%
  add_model(linear_reg() %>%
              set_engine("lm") %>%
              set_mode("regression"))
tune_lin=wf %>% tune_grid(resamples=cv,metrics=metric_set(rmse,mae))
wf=wf %>% finalize_workflow(tune_lin %>% select_best(metric="rmse"))
fit_s1_lin=fit(wf,train_set)
```



The most important factor is the type of free agent a player is. Once again, both versions of All-NBA voting share show up in the top 10.

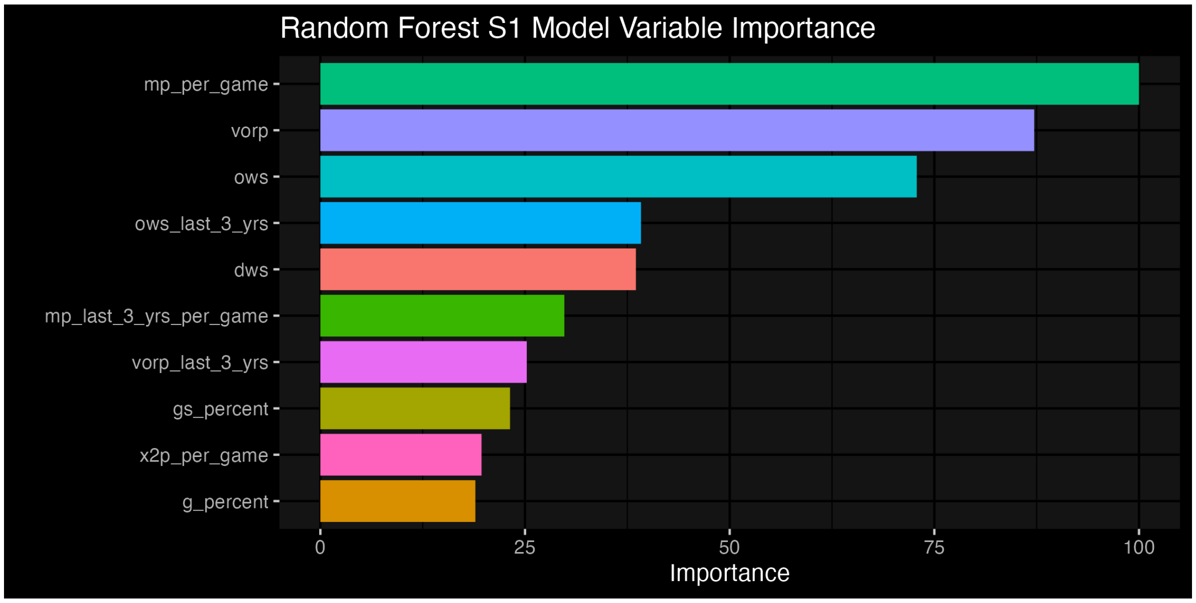
```
wf=workflow() %>% add_recipe(s1_recipe) %>%
  add_model(nearest_neighbor(neighbors=tune()) %>%
              set_engine("kknn") %>%
              set_mode("regression"))
tune_knn=wf %>% tune_grid(resamples=cv,
                          grid=expand.grid(neighbors=5:50),
                          metrics=metric_set(rmse,mae))
wf=wf %>% finalize_workflow(tune_knn %>% select_best(metric="rmse"))
fit_s1_knn=fit(wf,train_set)
tune_knn %>% select_best(metric="rmse") %>% gt()
```

neighbors	.config
29	Preprocessor1_Model25

The best KNN salary-first model is much like the KNN salary-second model in that the tuned neighbor parameter is in the middle of the tune grid, taking the 29 closest players in similarity to construct a player's salary prediction.

```r
set.seed(100,sample.kind = "Rounding")
wf=workflow() %>% add_recipe(s1_recipe) %>%
  add_model(rand_forest(mtry = tune()) %>%
  set_engine("ranger",importance="permutation") %>% set_mode("regression"))
tune_forest=wf %>% tune_grid(resamples=cv,
                             grid=expand.grid(mtry=5:10),
                             metrics=metric_set(rmse,mae))
wf=wf %>% finalize_workflow(tune_forest %>% select_best(metric="rmse"))
fit_s1_forest=fit(wf,train_set)
tune_forest %>% select_best(metric="rmse") %>% gt()
```

mtry	.config
10	Preprocessor1_Model6



The salary-first forest is more widely clustered at the top than the salary-second forest, having 3 variables >50% importance vs 1 respectively. Excluding contract years in the salary-second forest, the top 4 variables of importance follow the exact same order in both forests.

```r
set.seed(100,sample.kind = "Rounding")
wf=workflow() %>% add_recipe(s1_recipe) %>%
  add_model(svm_rbf(rbf_sigma=tune(),cost=tune()) %>%
  set_engine("kernlab") %>% set_mode("regression"))
tune_svm=wf %>% tune_grid(resamples=cv,
                          grid=expand.grid(rbf_sigma=0.1^seq(2:5),
                                           cost=c(0.25,0.5,1)),
                          metrics=metric_set(rmse,mae))
wf=wf %>% finalize_workflow(tune_svm %>% select_best(metric="rmse"))
fit_s1_svm=fit(wf,train_set)
tune_svm %>% select_best(metric="rmse") %>% gt()
```

cost	rbf_sigma	.config
1	0.001	Preprocessor1_Model11

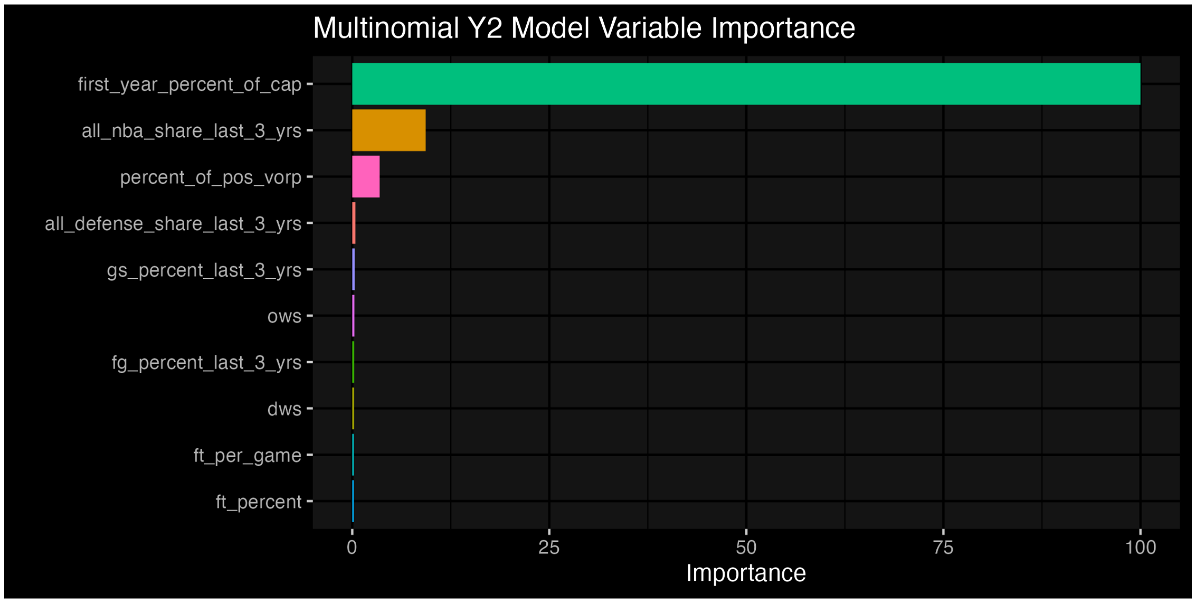
Below are the performance metrics for the salary-first models.

Method	Off By >5%	Within 2%	MAE	RMSE
Linear	7.26%	70.47%	0.01861413	0.02880876
KNN	6.27%	79.90%	0.01643719	0.02837975
Random Forest	0.55%	92.74%	0.01614994	0.02713783
SVM	7.37%	77.04%	0.01626831	0.02791222

Once again, the random forest is the top performer in both traditional and dataset-specific metrics. All 4 models performed worse when predicting salary-first vs salary-second. With the SVM dipping below 80%, we will use the random forest as our salary prediction.

```
train_set_after_s1=train_set %>% select(-first_year_percent_of_cap) %>%
  bind_cols(first_year_percent_of_cap=predict(fit_s1_forest,new_data=train_set) %>%
              pull())
cv=vfold_cv(train_set_after_s1,v=10,strata=type,repeats=5)
y2_recipe=recipe(contract_yrs~.,data=train_set_after_s1) %>%
  update_role(seas_id:player,new_role="id") %>% step_dummy(type)
```

```
wf=workflow() %>% add_recipe(y2_recipe) %>%
  add_model(multinom_reg(penalty=0) %>%
              set_engine("glmnet") %>%
              set_mode("classification"))
tune_multinom=wf %>% tune_grid(resamples=cv,metrics=metric_set(macro_weight_f1))
wf=wf %>% finalize_workflow(tune_multinom %>% select_best())
fit_y2_multinom=fit(wf,train_set_after_s1)
```



The multinomial years-second model essentially discards all other variables in favor of the predicted first-year-percent-of-cap variable. The two variables that are not less than 1% variable importance are all_nba_share_last_3_yrs (added this year) and percent_of_pos_vorp (added last year).

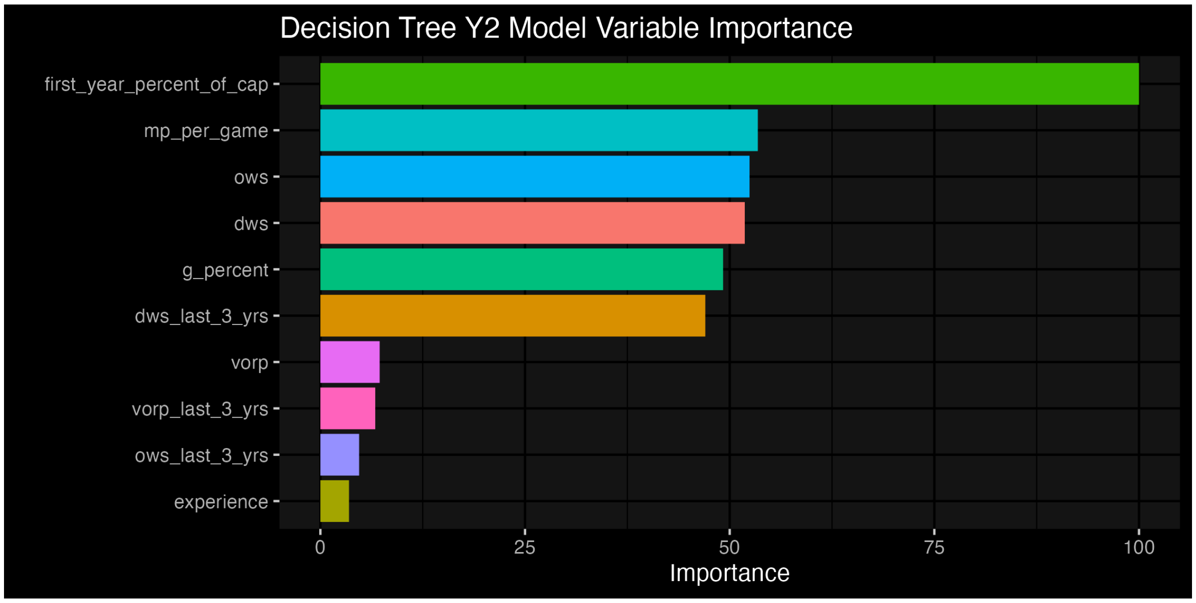
```
wf=workflow() %>% add_recipe(y2_recipe) %>%
  add_model(nearest_neighbor(neighbors = tune()) %>%
  set_engine("kknn") %>% set_mode("classification"))
tune_knn=wf %>% tune_grid(resamples=cv,
                        grid=expand.grid(neighbors=5:50),
                        metrics=metric_set(macro_weight_f1))
wf=wf %>% finalize_workflow(tune_knn %>% select_best())
fit_y2_knn=fit(wf,train_set_after_s1)
tune_knn %>% select_best() %>% gt()
```

neighbors	.config
47	Preprocessor1_Model43

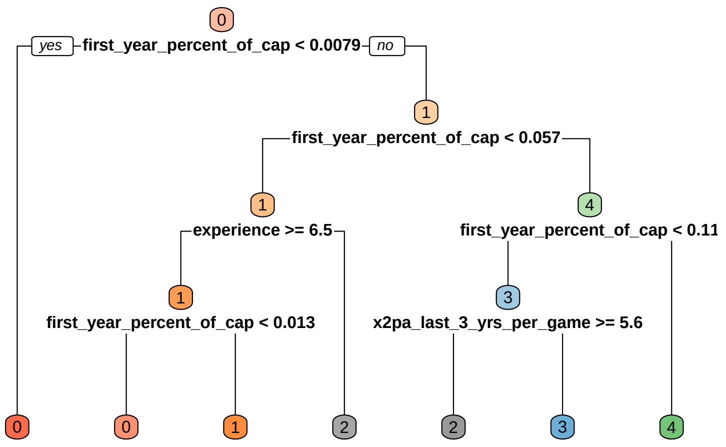
47 neighbors are needed in order for the KNN years-second model to classify a player's contract years.

```
set.seed(100,sample.kind = "Rounding")
wf=workflow() %>% add_recipe(y2_recipe) %>%
  add_model(decision_tree(cost_complexity = tune(),min_n=tune()) %>%
  set_engine("rpart") %>% set_mode("classification"))
tune_tree=wf %>% tune_grid(resamples=cv,
                           grid=expand.grid(cost_complexity=0.1^seq(2,5),
                                            min_n=seq(1:10)),
                           metrics=metric_set(macro_weight_f1))
wf=wf %>% finalize_workflow(tune_tree %>% select_best())
fit_y2_tree=fit(wf,train_set_after_s1)
tune_tree %>% select_best() %>% gt()
```

cost_complexity	min_n	.config
0.01	1	Preprocessor1_Model01



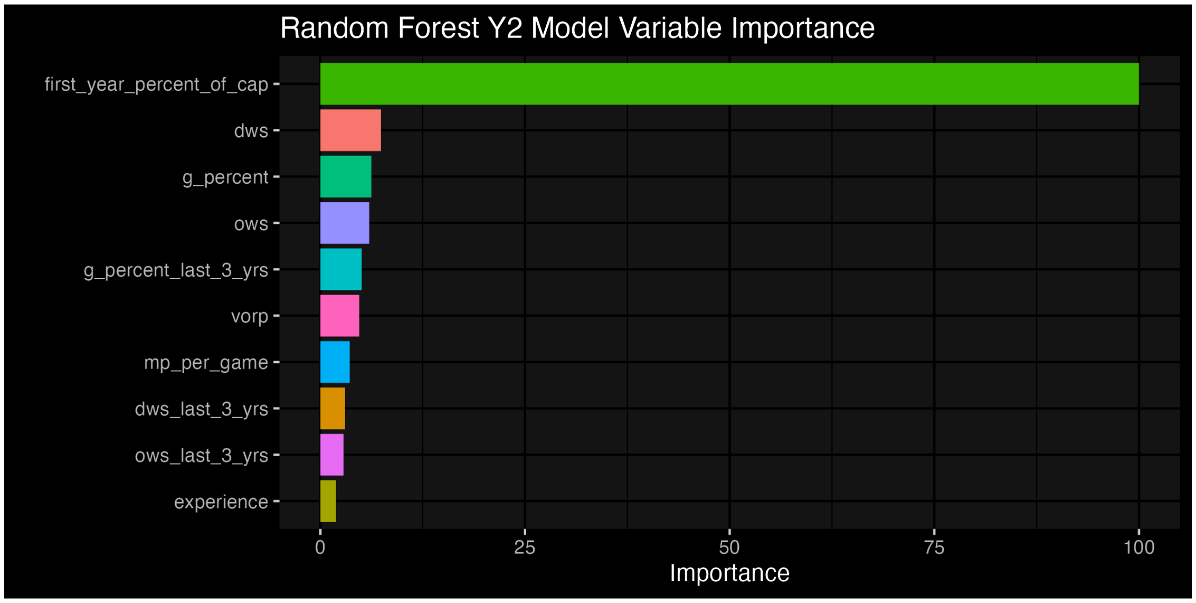
First-year-percent-of-cap shoots up to 1st in importance in the years-second decision tree, much like we saw in the years-second multinomial model. There's a second cluster of variables with importance between 45% and 55%.


0
yes
first_year_percent_of_cap < 0.0079
no
1
first_year_percent_of_cap < 0.057
1
experience >= 6.5
4
first_year_percent_of_cap < 0.11
1
first_year_percent_of_cap < 0.013
3
x2pa_last_3_yrs_per_game >= 5.6
0
0
1
2
2
3
4

The years-second decision tree still can't seem to figure out the division between 4- and 5-year contracts. It maximizes its prediction at 4 contract years if the first-year-percent-of-cap is greater than 11%.

```
set.seed(100,sample.kind = "Rounding")
wf=workflow() %>% add_recipe(y2_recipe) %>%
  add_model(rand_forest(mtry = tune()) %>%
  set_engine("ranger",importance="permutation") %>% set_mode("classification"))
tune_forest=wf %>% tune_grid(resamples=cv,
                             grid=expand.grid(mtry=5:10),
                             metrics=metric_set(macro_weight_f1))
wf=wf %>% finalize_workflow(tune_forest %>% select_best())
fit_y2_forest=fit(wf,train_set_after_s1)
tune_forest %>% select_best() %>% gt()
```

mtry	.config
10	Preprocessor1_Model6



The variable importance graph looks like the multinomial model with a slightly less severe division between the first-year-percent-of-cap variable importance and the rest of the variable importances.

```
set.seed(100,sample.kind = "Rounding")
wf=workflow() %>% add_recipe(y2_recipe) %>%
  add_model(svm_rbf(rbf_sigma=tune(),cost=tune()) %>%
  set_engine("kernlab") %>% set_mode("classification"))
tune_svm=wf %>% tune_grid(resamples=cv,
                          grid=expand.grid(rbf_sigma=0.1^seq(2:5),
                                           cost=c(0.25,0.5,1)),
                          metrics=metric_set(macro_weight_f1))
wf=wf %>% finalize_workflow(tune_svm %>% select_best())
fit_y2_svm=fit(wf,train_set_after_s1)
tune_svm %>% select_best() %>% gt()
```

cost	rbf_sigma	.config
0.25	1e-04	Preprocessor1_Model04

Last but not least, here are the years performance metrics of the S1Y2 model.

Method	Correct Predict %	Off by >1 Yr	Max Year Predicts	F1 Score
Multinomial	80.15%	4.29%	38	0.7494829

KNN	70.65%	10.86%	14	0.5760852
Decision Tree	73.48%	5.47%	0	0.7337852
Random Forest	99.93%	0.07%	46	0.7269443
SVM	51.39%	30.67%	0	0.6785064

The random forest is third in F1 score, but also has less than one percent of predictions that are off by more than one year. The multinomial model has the highest F1 score, and the second-best performer (yet still distant) in all other metrics after the random forest. We will use the random forest as the sole model to generate Y2 predictions.

Results

Now it's time to run the models on the 2023 data.

Since the models depend on contract year stats, these players were removed from the evaluation set:

player	type
Collin Gillespie	RFA
D.J. Augustin	UFA
Danilo Gallinari	UFA
Ibou Badji	RFA
Justin Lewis	RFA
Tristan Thompson	UFA
Willie Cauley-Stein	UFA

Years First, Salary Second

As a reminder, we're using a random forest as our years prediction, and using the mean of a random forest and SVM to predict salary.

```r
eval_set=inner_join(three_year_rolling_stats_and_shares,current_fa) %>%
  left_join(.,all_player_pos) %>% group_by(season,pos_group) %>%
  mutate(position_vorp=sum(vorp_last_3_yrs)) %>% ungroup() %>%
  mutate(percent_of_pos_vorp=vorp_last_3_yrs/position_vorp) %>%
  mutate(contract_yrs=factor(contract_yrs,levels = 0:5)) %>%
  select(-c(pos,pos_group,position_vorp)) %>%
  mutate(across(-c(seas_id:experience,type:contract_yrs),~round(.,digits=4)))
write_csv(eval_set,"Data/Eval Set.csv")
```

```
eval_y1_predict=as.numeric(as.character(
  predict(fit_y1_forest,new_data=eval_set) %>% pull()))
eval_s2_predict_forest=predict(fit_s2_forest,
                               new_data=eval_set %>% select(-contract_yrs) %>%
                                 bind_cols(contract_yrs=eval_y1_predict))
eval_s2_predict_svm=predict(fit_s2_svm,
                               new_data=eval_set %>% select(-contract_yrs) %>%
                                 bind_cols(contract_yrs=eval_y1_predict))
eval_s2_predict=tibble(forest=eval_s2_predict_forest %>% pull(),
                       svm=eval_s2_predict_svm %>% pull()) %>%
  mutate(m=(forest+svm)/2) %>% pull(m)
```

Salary First, Years Second

For the S1Y2 model, we are using random forests as our salary prediction, and using the mean of a random forest and multinomial to predict years. One problem could happen in differing predictions between the two: a half-year average prediction (if one model predicts 3 years and another predicts 4 years, the average is 3.5 years, which does not make sense). We will tend to the optimistic side of prediction by rounding up any half-year predictions. Additionally, due to the rarity of 5-year contract predictions, if either model predicts a 5-year contract for a player, that will be the final prediction regardless of the other model's value.

```
eval_s1_predict=predict(fit_s1_forest,new_data=eval_set) %>% pull()
eval_y2_predict=as.numeric(as.character(
  predict(fit_y2_forest,new_data=eval_set %>% select(-first_year_percent_of_cap) %>%
            bind_cols(first_year_percent_of_cap=eval_s1_predict)) %>%
  pull()))
```

Player Option Decisions and Selected Free Agents

Before I look at which players might be accepting or declining their option and then some selected high-profile free agents, let's look at the distribution of contract years.

On the first iteration of this project, both the Y1S2 & the S1Y2 had only 4-5 players being out of the league. This didn't make sense, as the annual rookie draft brings in at least 30 new players. The problem was that there were players who were predicted to receive contracts but receiving salaries that were less than the minimum. My solution is to convert any salary predictions less than the minimum to zero, and any years predictions to zero where the salary prediction is zero.

yrs_Y1S2	n
0	101
1	29
2	38
3	8
4	16
5	1

yrs_S1Y2	n
0	62
1	42
2	63
3	9
4	16
5	1

For the 2023-24 season, the salary cap is $134,000,000. The minimum contract from RealGM is $1,774,999. Therefore the minimum first year percent of cap in 2023-2024 is 1.32% .

```
first_yr_min=1774999/current_salary_cap
results_mins_conversion=results %>%
  mutate(yrs_Y1S2=ifelse(yr1_cap_percent_Y1S2<=first_yr_min,0,yrs_Y1S2)) %>%
  mutate(yr1_cap_percent_Y1S2=ifelse(yrs_Y1S2==0,0,yr1_cap_percent_Y1S2)) %>%
  mutate(yrs_S1Y2=ifelse(yr1_cap_percent_S1Y2<=first_yr_min,0,yrs_S1Y2)) %>%
  mutate(yr1_cap_percent_S1Y2=ifelse(yrs_S1Y2==0,0,yr1_cap_percent_S1Y2))
```

yrs_Y1S2	n
0	101
1	29
2	38
3	8
4	16
5	1

yrs_S1Y2	n
0	84
1	37
2	46
3	9
4	16
5	1

The Y1S2 predictions remain unchanged, while some 1-year and a fair amount of 2-year contracts in the S1Y2 predictions get converted to 0-year due to the salary being below the minimum threshold.

Another task is to make the information more digestible. Contracts are usually reported in terms of total value, and players usually get five percent raises per year (not compounded, but rather a percent of the first year salary). I'll add a column for each method for total contract value assuming the projected cap value of $134,000,000.

```
calculate_total_value<-function(cap_percent,years){
  #first year salary is base for all years
  salary_base=cap_percent*current_salary_cap*years
  #amount over base in raises
  amount_in_raises=(0.5*years^2-0.5*years)*0.05*cap_percent*current_salary_cap
  return(salary_base+amount_in_raises)
}

final_results=results_mins_conversion %>%
  mutate(total_Y1S2=
           round(calculate_total_value(yr1_cap_percent_Y1S2,yrs_Y1S2),digits=-4),
         total_S1Y2=
           round(
           round(calculate_total_value(yr1_cap_percent_S1Y2,yrs_S1Y2),digits=-4),
               digits=-4))
```

Player Option Decisions

	Y1S2			S1Y2			
player	Cap %	Yrs	Total	Cap %	Yrs	Total	option_amt
Khris Middleton	6.50%	1	$8.71M	7.12%	1	$9.54M	$40.40M
Kristaps Porziņģis	23.38%	5	$172.32M	23.94%	5	$176.42M	$36.02M
James Harden	23.58%	2	$64.77M	26.80%	2	$73.63M	$35.64M
Draymond Green	11.97%	2	$32.89M	11.85%	3	$50.04M	$27.59M
Fred VanVleet	19.91%	4	$114.71M	19.45%	4	$112.06M	$22.82M
Gary Trent Jr.	13.62%	4	$78.49M	13.32%	4	$76.75M	$18.56M
Jordan Clarkson	9.57%	2	$26.28M	10.77%	2	$29.59M	$14.26M
Kyle Kuzma	8.53%	1	$11.43M	11.35%	2	$31.18M	$13.00M
Josh Hart	13.23%	4	$76.23M	13.17%	4	$75.90M	$12.96M
Talen Horton-Tucker	2.87%	1	$3.84M	2.69%	1	$3.61M	$11.02M
Victor Oladipo	2.92%	1	$3.91M	3.17%	1	$4.25M	$9.45M
Bruce Brown	10.79%	4	$62.20M	9.87%	4	$56.85M	$6.80M
Rudy Gay	0.00%	0	$0.00	0.00%	0	$0.00	$6.48M
Donte DiVincenzo	8.12%	3	$34.29M	8.05%	3	$33.97M	$4.72M
Danuel House Jr.	0.00%	0	$0.00	0.00%	0	$0.00	$4.31M
Andre Drummond	3.05%	1	$4.09M	3.60%	1	$4.82M	$3.36M
Derrick Jones Jr.	3.12%	2	$8.56M	2.47%	2	$6.78M	$3.36M
Montrezl Harrell	0.00%	0	$0.00	2.82%	1	$3.78M	$2.76M
Damian Jones	0.00%	0	$0.00	0.00%	0	$0.00	$2.59M
Jevon Carter	4.74%	2	$13.02M	5.38%	2	$14.77M	$2.24M

Khris Middleton is a massive outlier, even more so than Bradley Beal last year. A player of his caliber will surely receive a contract offer for longer than one year. Middleton only played 33 games in his contract year after having offseason wrist surgery and dealing with right knee soreness. In his prior two seasons, Middleton averaged 20 points on 46% from the field and 39% from three-point range, but those numbers dipped to 15 points, 43% from the field and 31% from three-point range in the contract year. Kristaps Porzingis is the only five year prediction in the entire dataset. His previous five-year contract isn't even in the training set due to Porzingis missing the entire 2019 season recovering from a torn ACL. Fred VanVleet, Draymond Green & Josh Hart have already declined their player options. Bruce Brown is in line for a massive pay bump after signing a 1+1 last year with the Nuggets and becoming a key part of their championship team.

	Y1S2			S1Y2			
player	Cap %	Yrs	Total	Cap %	Yrs	Total	option_amt
Malik Beasley	5.20%	2	$14.27M	4.41%	2	$12.10M	$16.52M
Derrick Rose	0.00%	0	$0.00	0.00%	0	$0.00	$15.60M
Alec Burks	7.94%	3	$33.50M	6.88%	3	$29.05M	$10.49M
Mike Muscala	2.35%	1	$3.15M	2.70%	1	$3.62M	$3.50M
Michael Carter-Williams	0.00%	0	$0.00	0.00%	0	$0.00	$3.05M
Kevin Knox	0.00%	0	$0.00	0.00%	0	$0.00	$3.00M
Willy Hernangómez	0.00%	0	$0.00	1.58%	1	$2.11M	$2.56M
Goga Bitadze	0.00%	0	$0.00	0.00%	0	$0.00	$2.07M
Admiral Schofield	0.00%	0	$0.00	0.00%	0	$0.00	$2.00M
Nathan Knight	0.00%	0	$0.00	0.00%	0	$0.00	$2.00M
Kenyon Martin Jr.	8.29%	2	$22.76M	10.74%	3	$45.35M	$1.93M
Lamar Stevens	3.51%	2	$9.63M	3.22%	2	$8.84M	$1.93M
Naji Marshall	3.50%	1	$4.69M	4.42%	2	$12.14M	$1.93M
Xavier Tillman Sr.	6.83%	3	$28.81M	6.33%	3	$26.70M	$1.93M
Eugene Omoruyi	0.00%	0	$0.00	0.00%	0	$0.00	$1.93M
Ish Wainright	2.86%	2	$7.85M	2.10%	2	$5.77M	$1.93M
Kessler Edwards	0.00%	0	$0.00	0.00%	0	$0.00	$1.93M
Lindy Waters III	2.65%	2	$7.27M	1.76%	2	$4.83M	$1.93M
Herbert Jones	11.39%	4	$65.63M	10.06%	4	$57.97M	$1.84M
Isaiah Livers	0.00%	0	$0.00	1.72%	2	$4.72M	$1.84M
Miles McBride	0.00%	0	$0.00	0.00%	0	$0.00	$1.84M

Kenyon Martin Jr and Xavier Tillman Sr are in the same boat, as 2020 second round picks who signed 3-year deals (coincidentally, both were picked by the Sacramento Kings and traded to the Rockets & Grizzlies respectively). Their teams could pick up the option, but would make the players unrestricted free agents in 2024. The Rockets are intimately familiar with the process, as they declined Jae'Sean Tate's option last year to re-sign him to a 3-year, $22.1M contract (I had him pegged for a 2-year, $16.7M contract). The Pelicans have a somewhat easier decision with Herb Jones, as he'd still be a restricted free agent in 2024 if the team picks up his option.

Selected Free Agents

Now let's see what the models have predicted for this year's other notable free agents. I'll present them five at a time and comment on some projections. We'll look at restricted free agents first.

	Y1S2			S1Y2		
player	Cap %	Yrs	Total	Cap %	Yrs	Total
Cameron Johnson	12.53%	4	$72.17M$	12.23%	4	$70.48M$
P.J. Washington	12.74%	4	$73.42M$	11.61%	4	$66.89M$
Austin Reaves	9.92%	3	$41.85M$	11.10%	4	$63.96M$
Tre Jones	8.91%	3	$37.59M$	9.32%	3	$39.32M$
Grant Williams	8.06%	3	$34.00M$	8.43%	3	$35.60M$

The RFA class is not as top heavy as last year. Cam Johnson, part of the Nets midseason trade return for Kevin Durant, would rank third last year in average projection behind Miles Bridges & former Suns teammate Deandre Ayton. Picked one pick after Johnson at number 12 in 2020, Washington has toiled away in Charlotte as a solid role player. He's played the second most games for the Hornets since being drafted, and set a career high in points per game last year at 15.7. Reaves is the second fan-favorite white Lakers guard to hit free agency in three seasons, but that's where the comparisons with Alex Caruso end. Caruso is a defensive menace who made his first All-Defensive team this year; Reaves is more gifted on the offensive side of the ball, particularly adept at drawing fouls (he led all NBA guards in free throw attempts per field goal attempt at 0.541).

	Y1S2			S1Y2		
player	Cap %	Yrs	Total	Cap %	Yrs	Total
Coby White	6.25%	2	$17.18M$	6.92%	2	$19.02M$
Ayo Dosunmu	6.84%	3	$28.89M$	6.15%	3	$25.96M$
Matisse Thybulle	5.31%	2	$14.60M$	4.50%	2	$12.37M$
Rui Hachimura	3.99%	1	$5.35M$	5.47%	2	$15.04M$
Jock Landale	5.02%	3	$21.19M$	4.35%	2	$11.94M$

The young Chicago Bulls guards (White & Dosunmu) have similar cap percentage projections, with Dosunmu getting more under Y1S2 and White getting more under S1Y2. Thybulle made the All-Defensive Second Team in 2021 & 2022, but his offensive limitations led the Sixers to trade him to the Trail Blazers for a 2029 2nd-round pick. Hachimura's another example of someone who'll get more than projected; at the very least, his qualifying offer is for $8.5 million. Hachimura performed relatively well next to LeBron James in the bright lights of LA as the Lakers made a run to the Western Conference Finals (reminder that the models do not take into account playoff stats).

On to the unrestricted free agents.

	Y1S2			S1Y2		
player	Cap %	Yrs	Total	Cap %	Yrs	Total
Kyrie Irving	24.75%	4	$142.63M	24.83%	3	$104.79M
Nikola Vučević	20.06%	4	$115.59M	18.87%	4	$108.71M
Jerami Grant	17.16%	4	$98.90M	18.18%	4	$104.75M
Brook Lopez	17.50%	4	$100.83M	15.50%	4	$89.33M
D'Angelo Russell	16.00%	4	$92.18M	16.91%	4	$97.41M

Kyrie somewhat surprisingly picked up his $36.5M option last year. He had another eventful year, being suspended in November for posting a link to an antisemitic film on Twitter & failing to unequivocally say he has no antisemitic beliefs, and then traded in February to the Dallas Mavericks. Jerami Grant was acquired by the Trail Blazers last offseason to flank loyal superstar Damian Lillard. Grant was able to merge the efficiency he had from 2018-2020 with the volume scoring he showed as the first option for Detroit in 2021 & 2022. The Blazers ended up tanking after the All-Star break, and jumped up to the number 3 pick. Do the Blazers trade Lillard, let Grant walk and begin a rebuild or do they dangle the number 3 pick for a star that'll hopefully push them over the hump? Brook Lopez actually cannot get his projected contract years due to the Over-38 CBA rule. Due to Lopez turning 38 in 2026, the complications of adding a fourth year to any contract offer aren't worth it.

	Y1S2			S1Y2		
player	Cap %	Yrs	Total	Cap %	Yrs	Total
Harrison Barnes	15.34%	4	$88.39M	16.10%	4	$92.78M
Jakob Poeltl	14.02%	4	$80.77M	13.28%	4	$76.54M
Mason Plumlee	13.11%	4	$75.53M	11.37%	4	$65.49M
Caris LeVert	12.19%	4	$70.26M	10.31%	4	$59.42M
Russell Westbrook	9.03%	1	$12.10M	11.60%	1	$15.55M

I feel very confident in saying that Mason Plumlee will not getting his projected contract. Poeltl follows the same archetype as a non-shooting, paint-bound center with some playmaking chops, with the added bonus that he is six years younger than Plumlee. The ever-polarizing Russell Westbrook has the highest salary projection for players only projected for 1 contract year.

	Y1S2			S1Y2		
player	Cap %	Yrs	Total	Cap %	Yrs	Total
Christian Wood	9.47%	2	$26.01M	10.42%	2	$28.63M

Kelly Oubre Jr.	8.58%	2	$23.57M	9.79%	2	$26.90M
Dillon Brooks	7.46%	2	$20.51M	7.41%	2	$20.36M
Max Strus	5.80%	2	$15.94M	6.51%	2	$17.88M
Naz Reid	5.86%	2	$16.11M	6.43%	2	$17.67M

Strus & Gabe Vincent (average projection of 2 years and $10.6M) were integral parts of the supporting cast flanking Jimmy Butler & Bam Adebayo during Miami's surprising run as an eight seed to the NBA Finals. They'll get more than projections after performing under the brightest of lights. While Brooks might take too many shots for someone as efficient as he is and talked his way out of Memphis (seriously, he will not be brought back under any circumstances), he is a ferocious and versatile defender. He made second-team All-Defense this year with a key part of that candidacy being that "he has matched up against All-Stars more than any other NBA player, and he has defended them better than anybody else."

Conclusion

Using a combination of contract year & last-three-years stats and end-of-season all-league voting share, we set out to predict NBA free agent contracts for the 2023 offseason. Since we had correlated targets in contract length and first year percent of salary cap, we predicted one target first and used its predictions as an input for the second target. The contract years were predicted as a classification problem, while the salary was predicted as a regression model. We used six models: linear (for salary) & multinomial (for contract length) as baselines, k-nearest-neighbors, decision tree (only for predicting contract length), a random forest and a support vector machine. Kristaps Porzingis, Fred VanVleet, Kyrie Irving, Nikola Vucevic, Jerami Grant and Brook Lopez are all projected to get above $100 million in total contract value in at least one method.

The models suffer from being unable to quantify intangibles like playing reputation, team fit, within-season role changes, or willingness to take a reduced salary to be on a championship contender. The models also can't determine which team will sign which player. That highly depends on a sequence of events: if Team A signs this player, they don't have enough money to resign Player B, who then goes to Team C for less money, etc.

Perhaps I should have implemented a time factor or weighted recent years more heavily, as team decision makers may have gotten smarter. I didn't remove highly correlated predictors by design, as I wanted the models themselves to perform feature selection and determine what the most important variables were.

Future work could include trying more models, like boosting (in which models are added sequentially, with later models in the sequence attempting to correct the errors of earlier models). Another example might be adding a draft pedigree variable, as teams might be more willing to:

A. take a chance on a player with mediocre surface-level stats, as the team could believe the player was misused and the inherent talent is still there
B. pay up for a player with good stats, as the high pick of the player could signal to the player's higher potential