Using Data Analytics to Predict NBA Player Salaries

By

Daniel Spangler

A Thesis Submitted to The W.A. Franke Honors College

In Partial Fulfillment of the bachelor's degree
With Honors in
Statistics and Data Science

Program the honors thesis was completed in
Statistics and Data Science

THE UNIVERSITY OF ARIZONA

MAY 2025

Approved by:

Hao Helen Zhang
Chair, Statistics and Data Science GIDP

Abstract

This project uses advanced data analytics to develop a model to predict NBA player salaries using a combination of in season statistics and off the court characteristics of each player. The model aims to help aid general managers by providing them with an unbiased, data driven approach to evaluating NBA player value, to be used in important decisions like trades, free agent negotiations, and contract extensions. It helps by flagging players as “Underpaid” or “Overpaid” based on their performance and experience, helping general managers avoid players with hefty contracts that don’t contribute to winning.

Outside of the NBA front office, this model has lots of broad applications including helping player agents in contract negotiation and providing insight for sports betting companies and bettors to identify differences in mispriced player lines. Additionally, the model can be leveraged by sports companies to provide insight to audiences as to why a certain NBA player was able to sign a lucrative contract, and if it was an overpay or underpay by the team. Ultimately, this model offers a comprehensive framework for understanding the intersection of player performance, salary, and team economics, making it a valuable resource for general managers in and around the NBA.

Table of Contents

Chapter 1: Data Collection

Section 1.1: Data Organization

To begin the regression, I saved csv files from https://www.basketball-reference.com/ for the past 4 NBA seasons (2020-2021, 2021-2022, 2022-2023, 2023-2024). I saved csv files of different statistics from each year, including Per Game, Advanced, Play-by-Play, Shooting, and Adjusted Shooting. Altogether, this was a total of 20 csv files that I uploaded into R and converted into dataframes. In R, I was able to combine each year's statistics by grouping by player name, leaving me with 4 dataframes with the entire year's statistics in each one.

To run the regression, we need salary data for each NBA player. Using the salary table from Basketball Reference, I was able to read the csv that included salary data for each player from the year 2020 until 2029. I was able to upload this csv into R, and match using the player name. Because we are performing analysis over multiple years, we must factor in inflation into our regression. In the NBA, the Salary Cap is the amount of money that a team can spend on players every year. With the NBA growing each and every year, the Salary Cap continues to increase. This means that a player getting paid $20 million 5 years ago, is not at all equivalent to a player being paid $20 million today. To account for this, I divided the salary of the player by the salary cap of the NBA for that certain year. This leaves us with not the salary of the player, but the percent of the salary cap that a certain player takes up, making it now consistent throughout all 4 years of our data. Now the data for each year contains the player name, 76 columns of statistics, and the percent of the salary cap the player takes up for a particular season. Resulting in a combined 1495 rows of data across the 4 years of statistics.

To improve the accessibility of my data, I created a Relational Database Service through Amazon Web Services. Using R and SQL code, I was able to upload the 4 dataframes into AWS, making them more accessible in the future because they can be accessed on any computer through the AWS service, and not just files saved on my own device. I also saved a combined dataframe with values from each of the 4 years, with an extra column being used to determine the year that this was present.

Section 1.2: Web Scraping

Because the bulk of the data analysis was performed during the end of 2024 and beginning of 2025, I chose not to include the data from the 2024-2025 NBA season, as it would have been ever-changing throughout my analysis, resulting in incomplete data and bias. Opposed to the training data from the past 4 seasons, the data for the 2024-2025 season will be used as test data. To access this data, because it is constantly changing everyday, I had to use web scraping instead of simply downloading a csv of the data. I was able to do this using R, and accessing the same tables as above from https://www.basketball-reference.com/, but for the new NBA season. Once again, I uploaded this into the AWS database (which I had to constantly reupload since the data changed everyday), so I can access it during my future data analysis.

Chapter 2: Data Preprocessing

Section 2.1: Data Cleaning

The first step to cleaning the data was looking into the missing (or NA) values. A lot of these came from players that had never attempted a certain field goal, leaving their field goal percentage as NA. To combat this, I assigned all NA values in the data frames to 0, instead of just removing the rows associated. Additionally, I wanted to combat some injuries and the number of games played. Because of this I removed all rows in which the player didn't play 41 games, which is half of the season. I did this because if I were to keep players who played only a couple games, the variance for the performance of those players would be super high, and the data analysis could possibly be unreliable. Lastly, if a player gets traded mid-season, the tables on Basketball Reference list it as three separate values, the data from the first team, second team, and combined. In R I was able to code to include only the total statistics from the season regardless of team. The data now contained 1215 rows, as opposed to the 1495 before. The 78 column names are listed below:

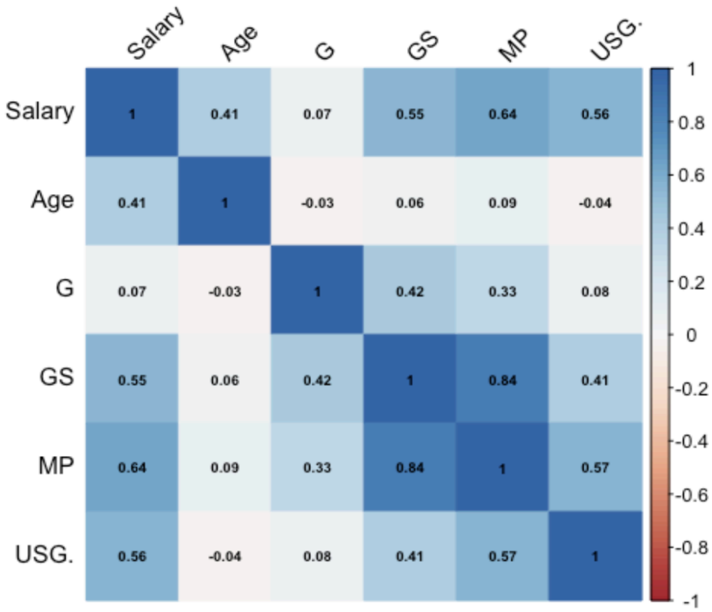
Position	Age	Team	Games	Games Started	Minutes Played per game	Field Goals per game	Field Goals Attempted per game
Field Goal Percentage	Three Point Shots per game	Three Point Shots Attempted per game	Three Point Percentage	Two Point Shots per game	Two Point Shots Attempted per game	Two Point Percentage	Effective Field Goal Percentage
Free Throws per game	Free Throws Attempted per game	Free Throw Percentage	Offensive Rebounds per game	Defensive Rebounds per game	Total Rebounds per game	Assists per game	Steals per game
Blocks per game	Turnovers per game	Personal Fouls per game	Player Efficiency Rating	True Shooting Percentage	Three Point Attempt Rate	Free Throw Attempt Rate	Offensive Rebound Percentage
Defensive Rebound Percentage	Total Rebound Percentage	Assist Percentage	Steal Percentage	Block Percentage	Turnover Percentage	Usage Percentage	Offensive Win Shares
Defensive Win Shares	Win Shares	Win Shares per 48 Minutes	Offensive Box Plus Minus	Defensive Box Plus Minus	Box Plus Minus	Value Over Replacement Player	Adjusted Field Goal Percentage
Adjusted Two Point Percentage	Adjusted Three Point Percentage	Adjusted Effective Field Goal Percentage	Adjusted Free Throw Percentage	AdjustedTrue Shooting Percentage	Adjusted Free Throw Rate	Adjusted Three Point Rate	League-Adjusted Field Goal Percentage
League-Adjusted Two Point Percentage	League-Adjusted Three Point Percentage	League-Adjusted Effective Field Goal Percentage	League-Adjusted Free Throw Percentage	League-Adjusted True Shooting Percentage	League-Adjusted Free Throw Rate	League-Adjusted Three Point Rate	Points Added by Field Goal Shooting
Points Added by True Shooting	On Court Plus Minus	On Off Court Plus Minus	Turnovers by Bad Pass	Lost Ball Turnovers	Shooting Fouls Committed	Offensive Fouls Committed	Shooting Fouls Drawn

Offensive Fouls Drawn	Points Generated by Assists	And 1 Field Goals	Field Goal Attempts Blocked	Points Per Game	Year		

Section 2.2: Exploratory Data Analysis

As you can probably imagine, there is bound to be some correlating data between the 78 columns of statistics. To visualize this correlation, I created a matrix correlation plot for each category of statistics, because graphing all 78 variables would be way too big to visualize.

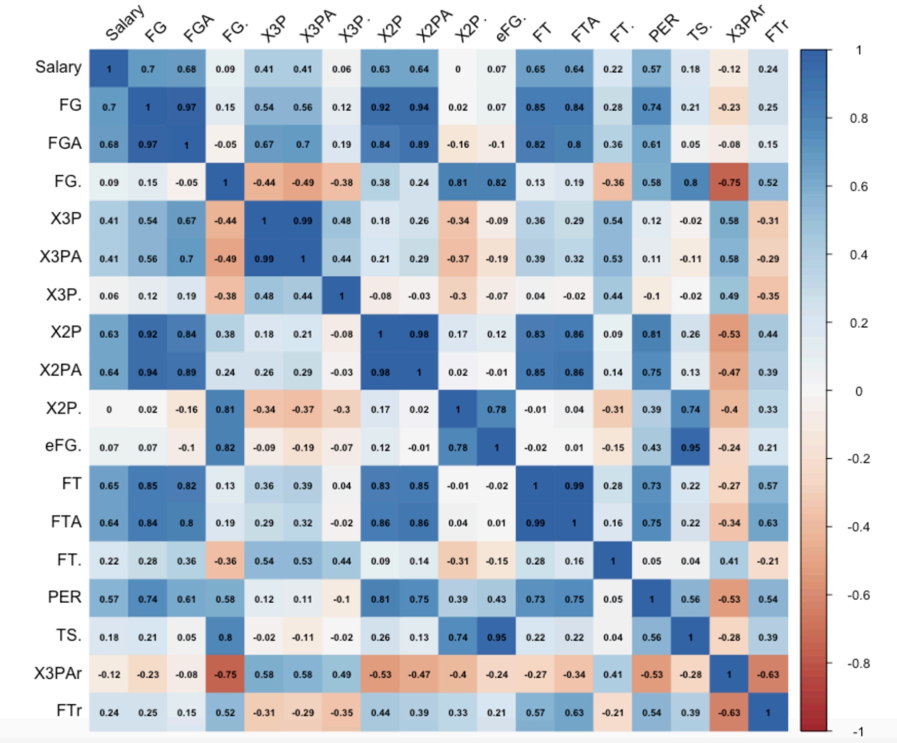
Firstly, I tested the general statistics for correlation between variables. This included Age, Games, Games Started, Minutes Played, and Usage Rate. The matrix can be seen below. In common practice, you want to remove variables that have a correlation variable of greater than 0.8. Looking at the graph, you can see that Minutes Per Game and Games Started are correlated. Because Minutes per game has a greater correlation with Salary overall, I decided to remove Games Started from the dataframe I will use for analysis. I will use this same process to remove all variables necessary.



The next grouping is the shooting statistics, including:

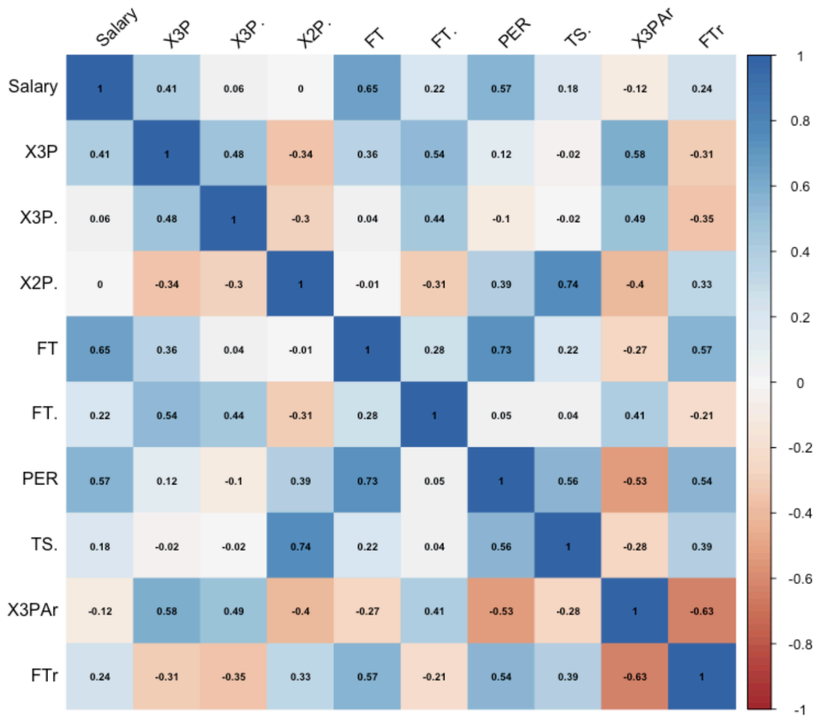
Field Goals	Field Goals Attempted	Field Goal Percentage	Three Point Field Goals	Three Pointers Attempted	Three Point Percentage
Two Point Field Goals	Two Pointers Attempted	Two Point Percentage	Effective Field Goal Percentage	Free Throws	Free Throws Attempted
Free Throw Percentage	Player Efficiency Rating	True Shooting Percentage	Three Point Attempt Rate	Free Throw Attempt Rate	

The correlation matrix can be seen below. Looking at the matrix, we can see that there are a couple of correlations in the data. After using the same process as before, I decided to remove variables Field Goals, Field Goals Attempted, Field Goal Percentage, Three Pointers Attempted, Two Point Field Goals, Two Point Field Goals Attempted, Effective Field Goal Percentage, and Free Throws Attempted.



	Salary	FG	FGA	FG.	X3P	X3PA	X3P.	X2P	X2PA	X2P.	eFG.	FT	FTA	FT.	PER	TS.	X3PAr	FTr
Salary	1	0.7	0.68	0.09	0.41	0.41	0.06	0.63	0.64	0	0.07	0.65	0.64	0.22	0.57	0.18	-0.12	0.24
FG	0.7	1	0.97	0.15	0.54	0.56	0.12	0.92	0.94	0.02	0.07	0.85	0.84	0.28	0.74	0.21	-0.23	0.25
FGA	0.68	0.97	1	-0.05	0.67	0.7	0.19	0.84	0.89	-0.16	-0.1	0.82	0.8	0.36	0.61	0.05	-0.08	0.15
FG.	0.09	0.15	-0.05	1	-0.44	-0.49	-0.38	0.38	0.24	0.81	0.82	0.13	0.19	-0.36	0.58	0.8	-0.75	0.52
X3P	0.41	0.54	0.67	-0.44	1	0.99	0.48	0.18	0.26	-0.34	-0.09	0.36	0.29	0.54	0.12	-0.02	0.58	-0.31
X3PA	0.41	0.56	0.7	-0.49	0.99	1	0.44	0.21	0.29	-0.37	-0.19	0.39	0.32	0.53	0.11	-0.11	0.58	-0.29
X3P.	0.06	0.12	0.19	-0.38	0.48	0.44	1	-0.08	-0.03	-0.3	-0.07	0.04	-0.02	0.44	-0.1	-0.02	0.49	-0.35
X2P	0.63	0.92	0.84	0.38	0.18	0.21	-0.08	1	0.98	0.17	0.12	0.83	0.86	0.09	0.81	0.26	-0.53	0.44
X2PA	0.64	0.94	0.89	0.24	0.26	0.29	-0.03	0.98	1	0.02	-0.01	0.85	0.86	0.14	0.75	0.13	-0.47	0.39
X2P.	0	0.02	-0.16	0.81	-0.34	-0.37	-0.3	0.17	0.02	1	0.78	-0.01	0.04	-0.31	0.39	0.74	-0.4	0.33
eFG.	0.07	0.07	-0.1	0.82	-0.09	-0.19	-0.07	0.12	-0.01	0.78	1	-0.02	0.01	-0.15	0.43	0.95	-0.24	0.21
FT	0.65	0.85	0.82	0.13	0.36	0.39	0.04	0.83	0.85	-0.01	-0.02	1	0.99	0.28	0.73	0.22	-0.27	0.57
FTA	0.64	0.84	0.8	0.19	0.29	0.32	-0.02	0.86	0.86	0.04	0.01	0.99	1	0.16	0.75	0.22	-0.34	0.63
FT.	0.22	0.28	0.36	-0.36	0.54	0.53	0.44	0.09	0.14	-0.31	-0.15	0.28	0.16	1	0.05	0.04	0.41	-0.21
PER	0.57	0.74	0.61	0.58	0.12	0.11	-0.1	0.81	0.75	0.39	0.43	0.73	0.75	0.05	1	0.56	-0.53	0.54
TS.	0.18	0.21	0.05	0.8	-0.02	-0.11	-0.02	0.26	0.13	0.74	0.95	0.22	0.22	0.04	0.56	1	-0.28	0.39
X3PAr	-0.12	-0.23	-0.08	-0.75	0.58	0.58	0.49	-0.53	-0.47	-0.4	-0.24	-0.27	-0.34	0.41	-0.53	-0.28	1	-0.63
FTr	0.24	0.25	0.15	0.52	-0.31	-0.29	-0.35	0.44	0.39	0.33	0.21	0.57	0.63	-0.21	0.54	0.39	-0.63	1

After removing the necessary variables, the correlation matrix looks as such with no correlations over the value of 0.8.

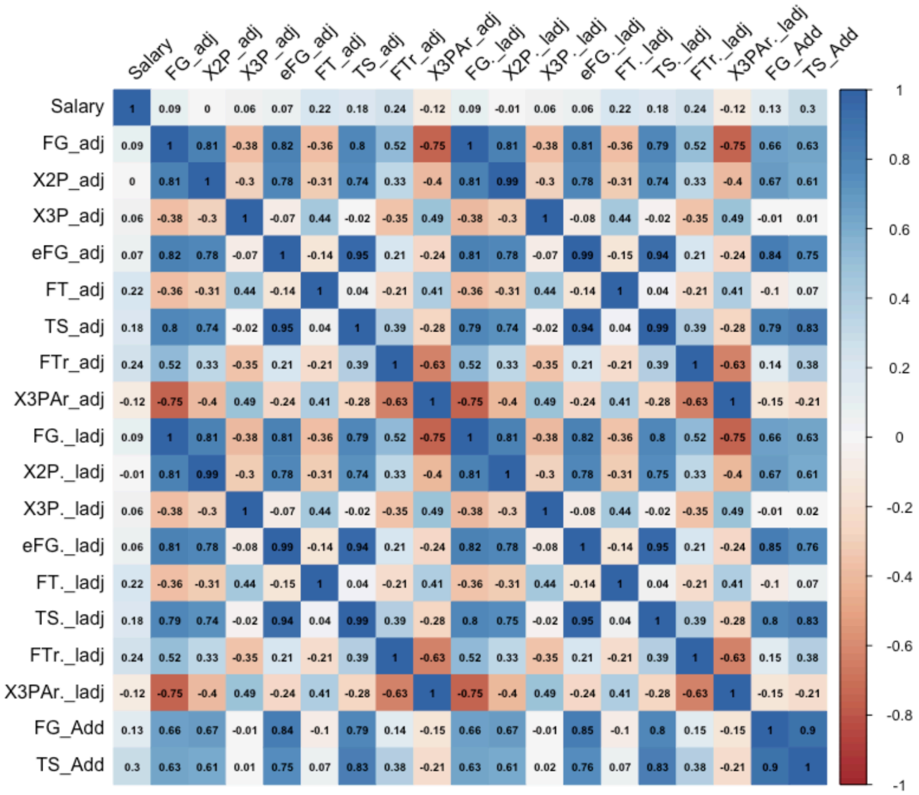


	Salary	X3P	X3P.	X2P.	FT	FT.	PER	TS.	X3PAr	FTr
Salary	1	0.41	0.06	0	0.65	0.22	0.57	0.18	-0.12	0.24
X3P	0.41	1	0.48	-0.34	0.36	0.54	0.12	-0.02	0.58	-0.31
X3P.	0.06	0.48	1	-0.3	0.04	0.44	-0.1	-0.02	0.49	-0.35
X2P.	0	-0.34	-0.3	1	-0.01	-0.31	0.39	0.74	-0.4	0.33
FT	0.65	0.36	0.04	-0.01	1	0.28	0.73	0.22	-0.27	0.57
FT.	0.22	0.54	0.44	-0.31	0.28	1	0.05	0.04	0.41	-0.21
PER	0.57	0.12	-0.1	0.39	0.73	0.05	1	0.56	-0.53	0.54
TS.	0.18	-0.02	-0.02	0.74	0.22	0.04	0.56	1	-0.28	0.39
X3PAr	-0.12	0.58	0.49	-0.4	-0.27	0.41	-0.53	-0.28	1	-0.63
FTr	0.24	-0.31	-0.35	0.33	0.57	-0.21	0.54	0.39	-0.63	1

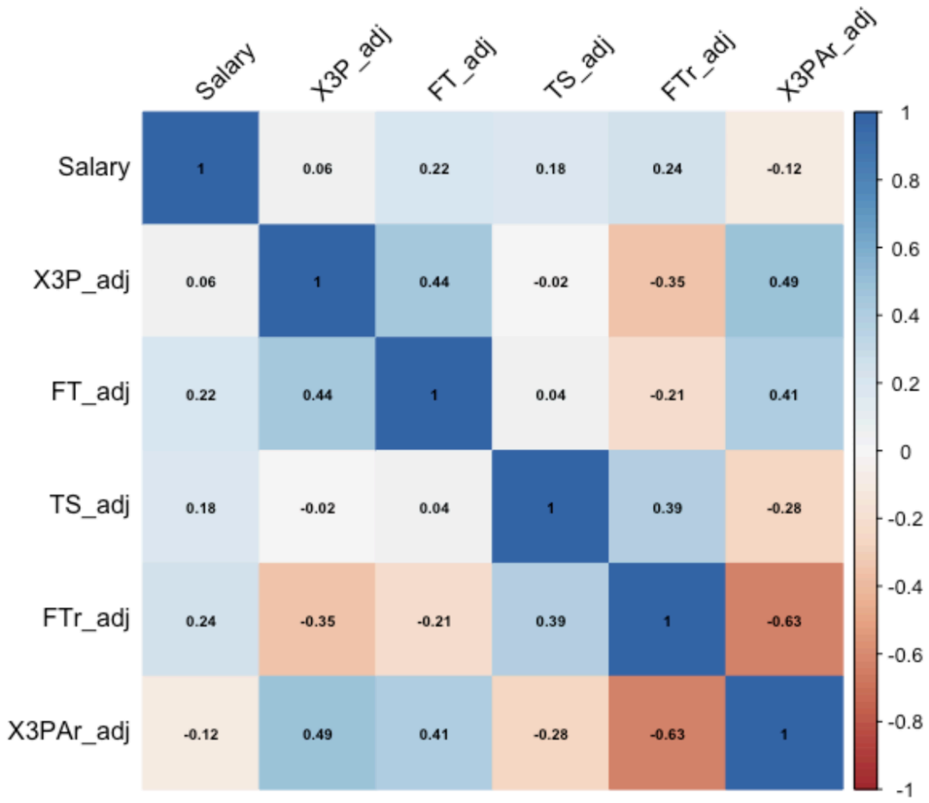
The next grouping is the adjusted shooting statistics, including:

Adjusted Field Goal Percentage	Adjusted Two Point Percentage	Adjusted Three Point Percentage	Adjusted Effective Field Goal Percentage	Adjusted Free Throw Percentage	Adjusted True Shooting
Adjusted Free Throw Rate	Adjusted Three Point Attempt Rate	League-Adjusted Field Goal Percentage	League-Adjusted Two Point Percentage	League-Adjusted Three Point Percentage	League-Adjusted Effective Field Goal Percentage
League-Adjusted Free Throw Percentage	League-Adjusted True Shooting	League-Adjusted Free Throw Rate	League-Adjusted Three Point Attempt Rate	Points Added by Field Goal Shooting	Points Added by True Shooting

The correlation matrix looks as follows:



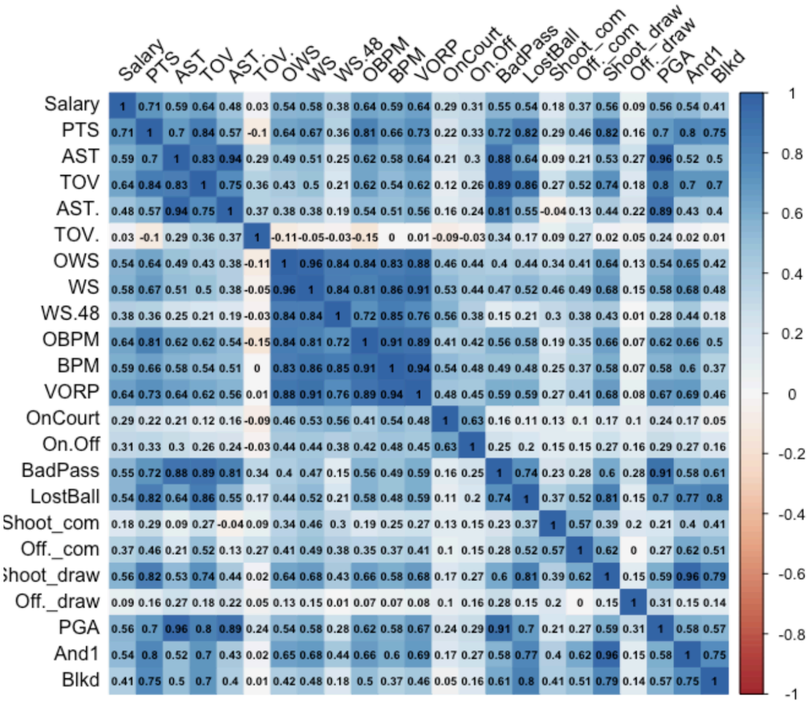
Looking at this correlation matrix, the data suggested we remove variables including FG_adj, X2P_adj, eFG_adj, FG_Add, TS_Add, and all variables that were league adjusted and . This left us with a correlation matrix with no correlations over 0.8.



The next set of variables included those that have to do with offensive statistics

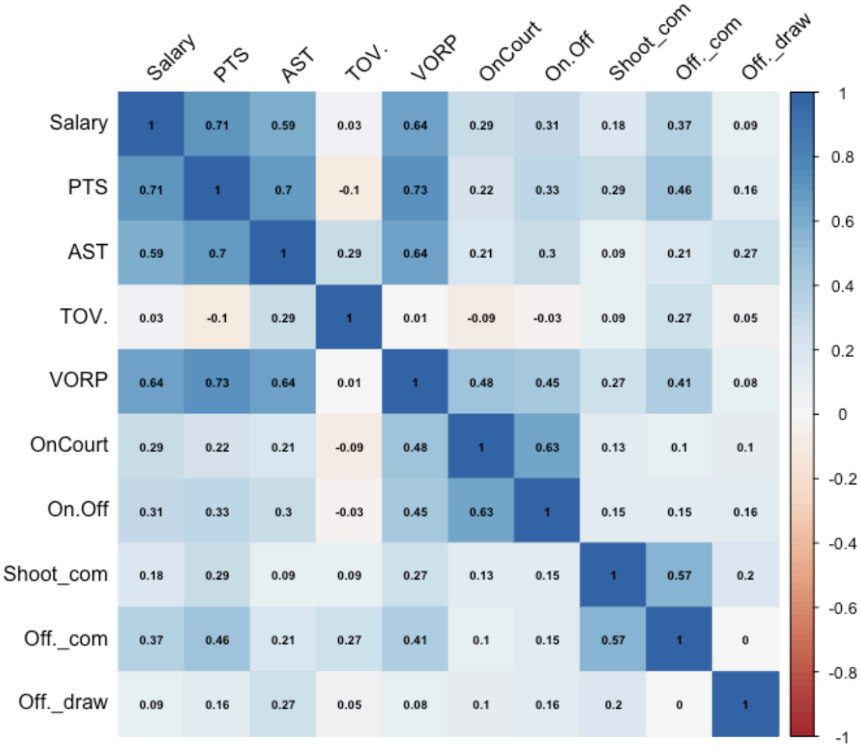
Assists per game	Turnovers per game	Assist Percentage	Turnover Percentage	Offensive Win Shares	Win Shares
Win Shares per 48 minutes	Offensive Box Plus Minus	Box Plus Minus	VORP	On Court	On-Off
Bad Passes	Lost Balls	Shooting Fouls Committed	Offensive Fouls Committed	Shooting fouls drawn	Offensive fouls drawn
PGA	And 1	Blocked Shots	Points		

These statistics had a correlation matrix as follows:



	Salary	PTS	AST	TOV	AST.	TOV.	OWS	WS	WS.48	OBPM	BPM	VORP	OnCourt	On.Off	BadPass	LostBall	Shoot_com	Off._com	Shoot_draw	Off._draw	PGA	And1	Blkd
Salary	1	0.71	0.59	0.64	0.48	0.03	0.54	0.58	0.38	0.64	0.59	0.64	0.29	0.31	0.55	0.54	0.18	0.37	0.56	0.09	0.56	0.54	0.41
PTS	0.71	1	0.7	0.84	0.57	-0.1	0.64	0.67	0.36	0.81	0.66	0.73	0.22	0.33	0.72	0.82	0.29	0.46	0.82	0.16	0.7	0.8	0.75
AST	0.59	0.7	1	0.83	0.94	0.29	0.49	0.51	0.25	0.62	0.58	0.64	0.21	0.3	0.88	0.64	0.09	0.21	0.53	0.27	0.96	0.52	0.5
TOV	0.64	0.84	0.83	1	0.75	0.36	0.43	0.5	0.21	0.62	0.54	0.62	0.12	0.26	0.89	0.86	0.27	0.52	0.74	0.18	0.8	0.7	0.7
AST.	0.48	0.57	0.94	0.75	1	0.37	0.38	0.38	0.19	0.54	0.51	0.56	0.16	0.24	0.81	0.55	-0.04	0.13	0.44	0.22	0.89	0.43	0.4
TOV.	0.03	-0.1	0.29	0.36	0.37	1	-0.11	-0.05	-0.03	-0.15	0	0.01	-0.09	-0.03	0.34	0.17	0.09	0.27	0.02	0.05	0.24	0.02	0.01
OWS	0.54	0.64	0.49	0.43	0.38	-0.11	1	0.96	0.84	0.84	0.83	0.88	0.46	0.44	0.4	0.44	0.34	0.41	0.64	0.13	0.54	0.65	0.42
WS	0.58	0.67	0.51	0.5	0.38	-0.05	0.96	1	0.84	0.81	0.86	0.91	0.53	0.44	0.47	0.52	0.46	0.49	0.68	0.15	0.58	0.68	0.48
WS.48	0.38	0.36	0.25	0.21	0.19	-0.03	0.84	0.84	1	0.72	0.85	0.76	0.56	0.38	0.15	0.21	0.3	0.38	0.43	0.01	0.28	0.44	0.18
OBPM	0.64	0.81	0.62	0.62	0.54	-0.15	0.84	0.81	0.72	1	0.91	0.89	0.41	0.42	0.56	0.58	0.19	0.35	0.66	0.07	0.62	0.66	0.5
BPM	0.59	0.66	0.58	0.54	0.51	0	0.83	0.86	0.85	0.91	1	0.94	0.54	0.48	0.49	0.48	0.25	0.37	0.58	0.07	0.58	0.6	0.37
VORP	0.64	0.73	0.64	0.62	0.56	0.01	0.88	0.91	0.76	0.89	0.94	1	0.48	0.45	0.59	0.59	0.27	0.41	0.68	0.08	0.67	0.69	0.46
OnCourt	0.29	0.22	0.21	0.12	0.16	-0.09	0.46	0.53	0.56	0.41	0.54	0.48	1	0.63	0.16	0.11	0.13	0.1	0.17	0.1	0.24	0.17	0.05
On.Off	0.31	0.33	0.3	0.26	0.24	-0.03	0.44	0.44	0.38	0.42	0.48	0.45	0.63	1	0.25	0.2	0.15	0.15	0.27	0.16	0.29	0.27	0.16
BadPass	0.55	0.72	0.88	0.89	0.81	0.34	0.4	0.47	0.15	0.56	0.49	0.59	0.16	0.25	1	0.74	0.23	0.28	0.6	0.28	0.91	0.58	0.61
LostBall	0.54	0.82	0.64	0.86	0.55	0.17	0.44	0.52	0.21	0.58	0.48	0.59	0.11	0.2	0.74	1	0.37	0.52	0.81	0.15	0.7	0.77	0.8
Shoot_com	0.18	0.29	0.09	0.27	-0.04	0.09	0.34	0.46	0.3	0.19	0.25	0.27	0.13	0.15	0.23	0.37	1	0.57	0.39	0.2	0.21	0.4	0.41
Off._com	0.37	0.46	0.21	0.52	0.13	0.27	0.41	0.49	0.38	0.35	0.37	0.41	0.1	0.15	0.28	0.52	0.57	1	0.62	0	0.27	0.62	0.51
Shoot_draw	0.56	0.82	0.53	0.74	0.44	0.02	0.64	0.68	0.43	0.66	0.58	0.68	0.17	0.27	0.6	0.81	0.39	0.62	1	0.15	0.59	0.96	0.79
Off._draw	0.09	0.16	0.27	0.18	0.22	0.05	0.13	0.15	0.01	0.07	0.07	0.08	0.1	0.16	0.28	0.15	0.2	0	0.15	1	0.31	0.15	0.14
PGA	0.56	0.7	0.96	0.8	0.89	0.24	0.54	0.58	0.28	0.62	0.58	0.67	0.24	0.29	0.91	0.7	0.21	0.27	0.59	0.31	1	0.58	0.57
And1	0.54	0.8	0.52	0.7	0.43	0.02	0.65	0.68	0.44	0.66	0.6	0.69	0.17	0.27	0.58	0.77	0.4	0.62	0.96	0.15	0.58	1	0.75
Blkd	0.41	0.75	0.5	0.7	0.4	0.01	0.42	0.48	0.18	0.5	0.37	0.46	0.05	0.16	0.61	0.8	0.41	0.51	0.79	0.14	0.57	0.75	1

To form a new correlation matrix, the variables removed included TOV, AST%, OWS, WS, Ws.48, OBPM, BPM, Bad Pass, Lost Ball, Shoot_draw, PGA, And1, and Blocked. To form a new matrix:

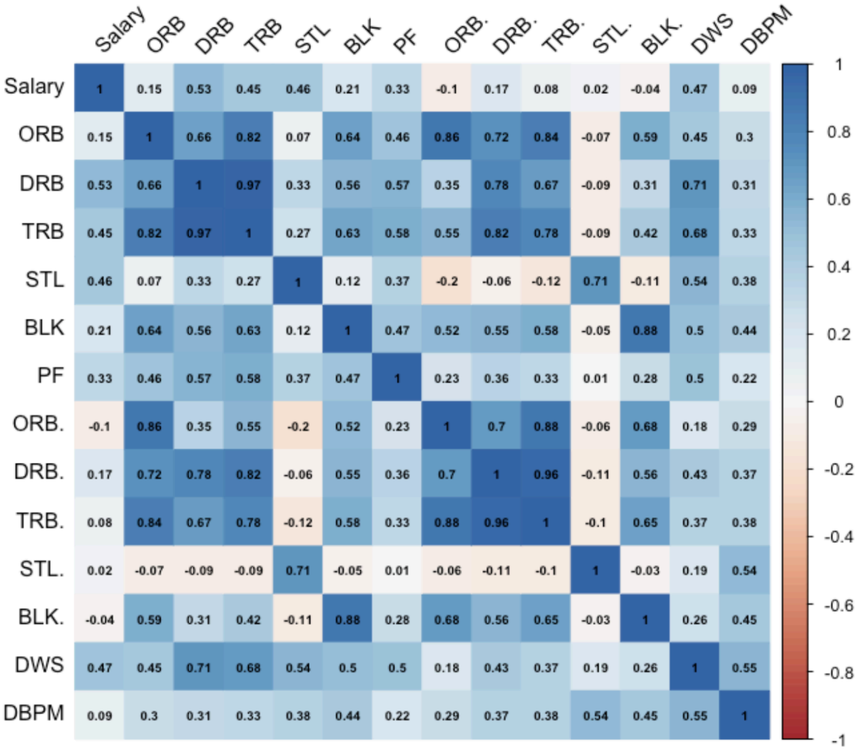


	Salary	PTS	AST	TOV.	VORP	OnCourt	On.Off	Shoot_com	Off._com	Off._draw
Salary	1	0.71	0.59	0.03	0.64	0.29	0.31	0.18	0.37	0.09
PTS	0.71	1	0.7	-0.1	0.73	0.22	0.33	0.29	0.46	0.16
AST	0.59	0.7	1	0.29	0.64	0.21	0.3	0.09	0.21	0.27
TOV.	0.03	-0.1	0.29	1	0.01	-0.09	-0.03	0.09	0.27	0.05
VORP	0.64	0.73	0.64	0.01	1	0.48	0.45	0.27	0.41	0.08
OnCourt	0.29	0.22	0.21	-0.09	0.48	1	0.63	0.13	0.1	0.1
On.Off	0.31	0.33	0.3	-0.03	0.45	0.63	1	0.15	0.15	0.16
Shoot_com	0.18	0.29	0.09	0.09	0.27	0.13	0.15	1	0.57	0.2
Off._com	0.37	0.46	0.21	0.27	0.41	0.1	0.15	0.57	1	0
Off._draw	0.09	0.16	0.27	0.05	0.08	0.1	0.16	0.2	0	1

Finally, the last category of variables were defensive statistics including:

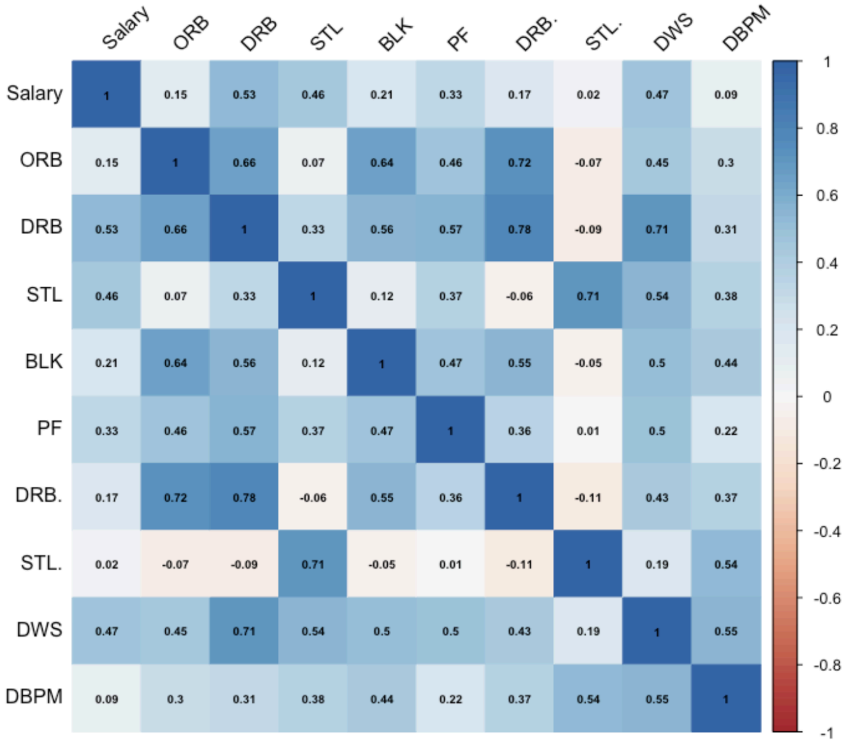
Offensive Rebounds per game	Defensive Rebounds per game	Total Rebounds per game	Steals per game	Blocks per game	Personal Fouls
Offensive Rebound percentage	Defensive Rebound percentage	Total Rebound percentage	Steal percentage	Block percentage	Defensive Win Shares
Defensive Box Plus Minus					

The correlation graph originally looked like this:



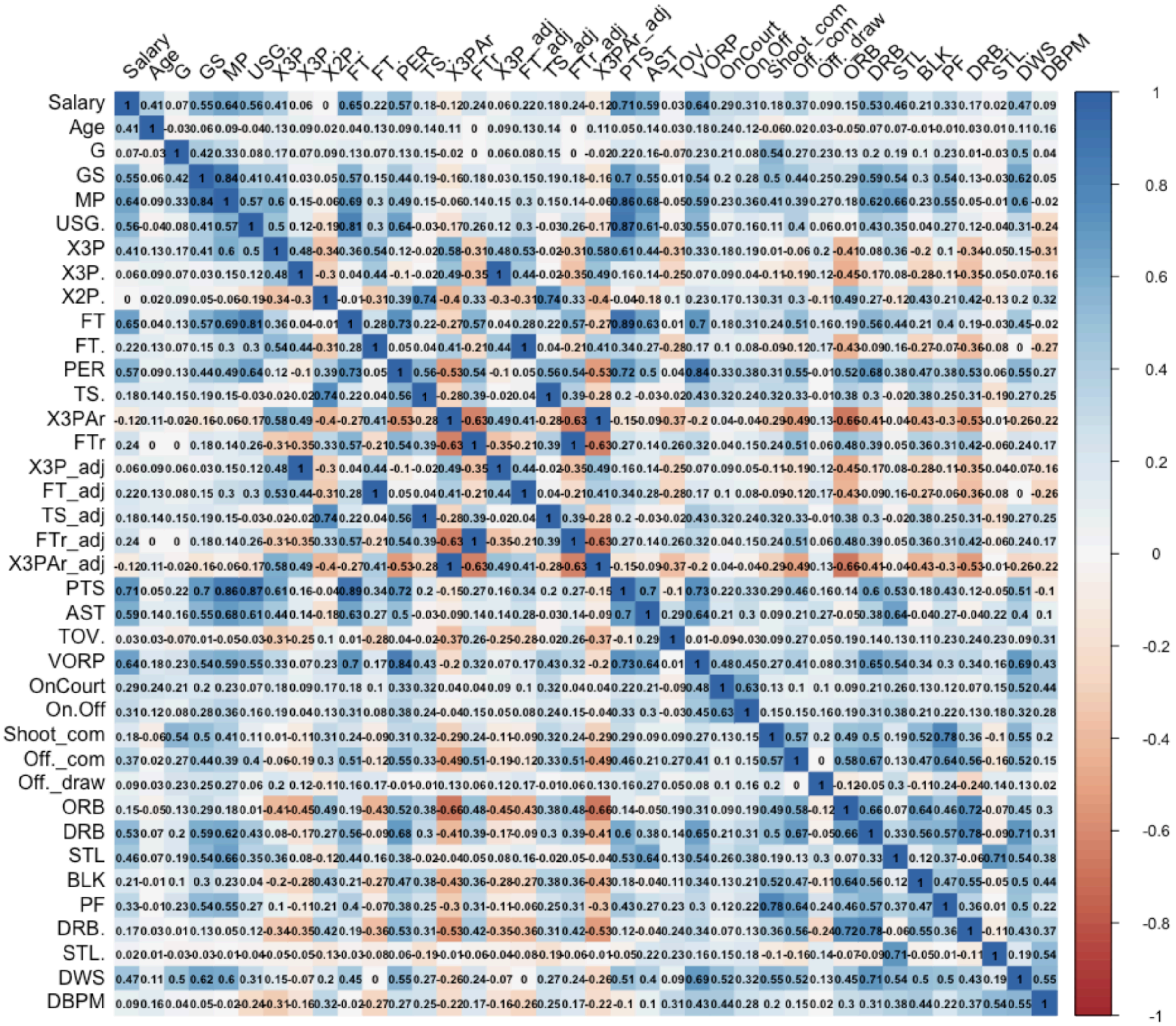
	Salary	ORB	DRB	TRB	STL	BLK	PF	ORB.	DRB.	TRB.	STL.	BLK.	DWS	DBPM
Salary	1	0.15	0.53	0.45	0.46	0.21	0.33	-0.1	0.17	0.08	0.02	-0.04	0.47	0.09
ORB	0.15	1	0.66	0.82	0.07	0.64	0.46	0.86	0.72	0.84	-0.07	0.59	0.45	0.3
DRB	0.53	0.66	1	0.97	0.33	0.56	0.57	0.35	0.78	0.67	-0.09	0.31	0.71	0.31
TRB	0.45	0.82	0.97	1	0.27	0.63	0.58	0.55	0.82	0.78	-0.09	0.42	0.68	0.33
STL	0.46	0.07	0.33	0.27	1	0.12	0.37	-0.2	-0.06	-0.12	0.71	-0.11	0.54	0.38
BLK	0.21	0.64	0.56	0.63	0.12	1	0.47	0.52	0.55	0.58	-0.05	0.88	0.5	0.44
PF	0.33	0.46	0.57	0.58	0.37	0.47	1	0.23	0.36	0.33	0.01	0.28	0.5	0.22
ORB.	-0.1	0.86	0.35	0.55	-0.2	0.52	0.23	1	0.7	0.88	-0.06	0.68	0.18	0.29
DRB.	0.17	0.72	0.78	0.82	-0.06	0.55	0.36	0.7	1	0.96	-0.11	0.56	0.43	0.37
TRB.	0.08	0.84	0.67	0.78	-0.12	0.58	0.33	0.88	0.96	1	-0.1	0.65	0.37	0.38
STL.	0.02	-0.07	-0.09	-0.09	0.71	-0.05	0.01	-0.06	-0.11	-0.1	1	-0.03	0.19	0.54
BLK.	-0.04	0.59	0.31	0.42	-0.11	0.88	0.28	0.68	0.56	0.65	-0.03	1	0.26	0.45
DWS	0.47	0.45	0.71	0.68	0.54	0.5	0.5	0.18	0.43	0.37	0.19	0.26	1	0.55
DBPM	0.09	0.3	0.31	0.33	0.38	0.44	0.22	0.29	0.37	0.38	0.54	0.45	0.55	1

After removing variables TRB, ORB%, TRB%, BLK%, and DWS, the correlation matrix is as follows:



	Salary	ORB	DRB	STL	BLK	PF	DRB.	STL.	DWS	DBPM
Salary	1	0.15	0.53	0.46	0.21	0.33	0.17	0.02	0.47	0.09
ORB	0.15	1	0.66	0.07	0.64	0.46	0.72	-0.07	0.45	0.3
DRB	0.53	0.66	1	0.33	0.56	0.57	0.78	-0.09	0.71	0.31
STL	0.46	0.07	0.33	1	0.12	0.37	-0.06	0.71	0.54	0.38
BLK	0.21	0.64	0.56	0.12	1	0.47	0.55	-0.05	0.5	0.44
PF	0.33	0.46	0.57	0.37	0.47	1	0.36	0.01	0.5	0.22
DRB.	0.17	0.72	0.78	-0.06	0.55	0.36	1	-0.11	0.43	0.37
STL.	0.02	-0.07	-0.09	0.71	-0.05	0.01	-0.11	1	0.19	0.54
DWS	0.47	0.45	0.71	0.54	0.5	0.5	0.43	0.19	1	0.55
DBPM	0.09	0.3	0.31	0.38	0.44	0.22	0.37	0.54	0.55	1

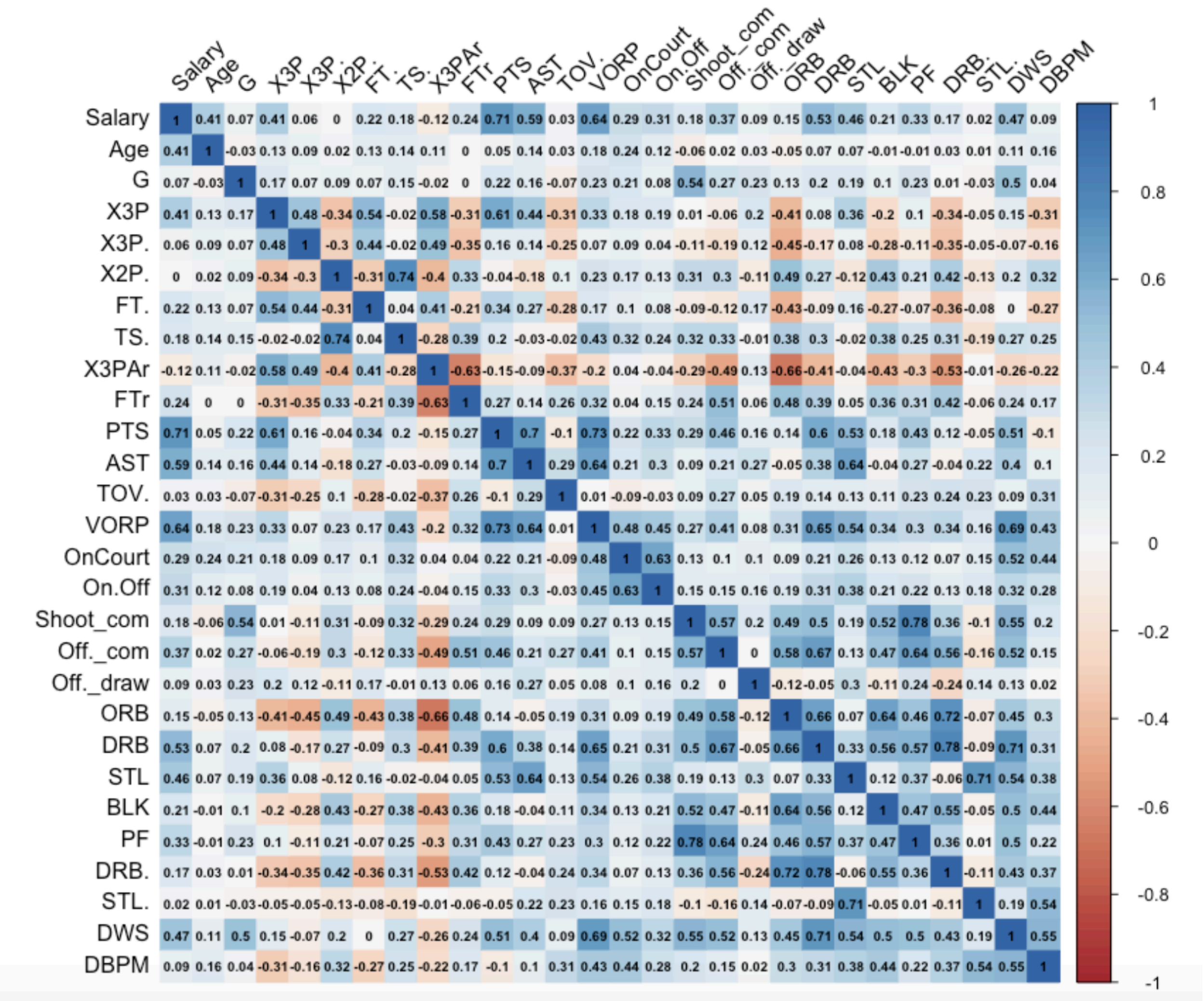
Lastly we want to compare the correlations between variables in different categories. To do this I will form a big correlation matrix comparing all variables that have been left in the model after the initial shedding. The combined correlation matrix is shown below:



There are still a few problem points with variables correlating across categories. Using the same process as before, I removed variables GS, USG%, all adjusted statistics, FT, MP, and PER. This leaves us with 27 variables including:

Age	Games	3 Pointers per game	3 Point percentage	2 Point percentage	Free Throw percentage
True Shooting percentage	3 Point Attempt Rate	Free Throw Rate	Points per game	Assists per game	Turnover percentage
VORP	On Court	On Off	Shooting Fouls Committed	Offensive Fouls Committed	Offensive Fouls Drawn
Offensive Rebounds per game	Defensive Rebounds per game	Steals per game	Blocks per game	Personal Fouls	Defensive Rebound percentage
Steal percentage	Defensive Win Shares	Defensive Box Plus Minus			

And a final correlation matrix of:



	Salary	Age	G	X3P	X3P.	X2P.	FT.	TS.	X3PAr	FTr	PTS	AST	TOV.	VORP	OnCourt	On.Off	Shoot_com	Off._com	Off._draw	ORB	DRB	STL	BLK	PF	DRB.	STL.	DWS	DBPM
Salary	1	0.41	0.07	0.41	0.06	0	0.22	0.18	-0.12	0.24	0.71	0.59	0.03	0.64	0.29	0.31	0.18	0.37	0.09	0.15	0.53	0.46	0.21	0.33	0.17	0.02	0.47	0.09
Age	0.41	1	-0.03	0.13	0.09	0.02	0.13	0.14	0.11	0	0.05	0.14	0.03	0.18	0.24	0.12	-0.06	0.02	0.03	-0.05	0.07	0.07	-0.01	-0.01	0.03	0.01	0.11	0.16
G	0.07	-0.03	1	0.17	0.07	0.09	0.07	0.15	-0.02	0	0.22	0.16	-0.07	0.23	0.21	0.08	0.54	0.27	0.23	0.13	0.2	0.19	0.1	0.23	0.01	-0.03	0.5	0.04
X3P	0.41	0.13	0.17	1	0.48	-0.34	0.54	-0.02	0.58	-0.31	0.61	0.44	-0.31	0.33	0.18	0.19	0.01	-0.06	0.2	-0.41	0.08	0.36	-0.2	0.1	-0.34	-0.05	0.15	-0.31
X3P.	0.06	0.09	0.07	0.48	1	-0.3	0.44	-0.02	0.49	-0.35	0.16	0.14	-0.25	0.07	0.09	0.04	-0.11	-0.19	0.12	-0.45	-0.17	0.08	-0.28	-0.11	-0.35	-0.05	-0.07	-0.16
X2P.	0	0.02	0.09	-0.34	-0.3	1	-0.31	0.74	-0.4	0.33	-0.04	-0.18	0.1	0.23	0.17	0.13	0.31	0.3	-0.11	0.49	0.27	-0.12	0.43	0.21	0.42	-0.13	0.2	0.32
FT.	0.22	0.13	0.07	0.54	0.44	-0.31	1	0.04	0.41	-0.21	0.34	0.27	-0.28	0.17	0.1	0.08	-0.09	-0.12	0.17	-0.43	-0.09	0.16	-0.27	-0.07	-0.36	-0.08	0	-0.27
TS.	0.18	0.14	0.15	-0.02	-0.02	0.74	0.04	1	-0.28	0.39	0.2	-0.03	-0.02	0.43	0.32	0.24	0.32	0.33	-0.01	0.38	0.3	-0.02	0.38	0.25	0.31	-0.19	0.27	0.25
X3PAr	-0.12	0.11	-0.02	0.58	0.49	-0.4	0.41	-0.28	1	-0.63	-0.15	-0.09	-0.37	-0.2	0.04	-0.04	-0.29	-0.49	0.13	-0.66	-0.41	-0.04	-0.43	-0.3	-0.53	-0.01	-0.26	-0.22
FTr	0.24	0	0	-0.31	-0.35	0.33	-0.21	0.39	-0.63	1	0.27	0.14	0.26	0.32	0.04	0.15	0.24	0.51	0.06	0.48	0.39	0.05	0.36	0.31	0.42	-0.06	0.24	0.17
PTS	0.71	0.05	0.22	0.61	0.16	-0.04	0.34	0.2	-0.15	0.27	1	0.7	-0.1	0.73	0.22	0.33	0.29	0.46	0.16	0.14	0.6	0.53	0.18	0.43	0.12	-0.05	0.51	-0.1
AST	0.59	0.14	0.16	0.44	0.14	-0.18	0.27	-0.03	-0.09	0.14	0.7	1	0.29	0.64	0.21	0.3	0.09	0.21	0.27	-0.05	0.38	0.64	-0.04	0.27	-0.04	0.22	0.4	0.1
TOV.	0.03	0.03	-0.07	-0.31	-0.25	0.1	-0.28	-0.02	-0.37	0.26	-0.1	0.29	1	0.01	-0.09	-0.03	0.09	0.27	0.05	0.19	0.14	0.13	0.11	0.23	0.24	0.23	0.09	0.31
VORP	0.64	0.18	0.23	0.33	0.07	0.23	0.17	0.43	-0.2	0.32	0.73	0.64	0.01	1	0.48	0.45	0.27	0.41	0.08	0.31	0.65	0.54	0.34	0.3	0.34	0.16	0.69	0.43
OnCourt	0.29	0.24	0.21	0.18	0.09	0.17	0.1	0.32	0.04	0.04	0.22	0.21	-0.09	0.48	1	0.63	0.13	0.1	0.1	0.09	0.21	0.26	0.13	0.12	0.07	0.15	0.52	0.44
On.Off	0.31	0.12	0.08	0.19	0.04	0.13	0.08	0.24	-0.04	0.15	0.33	0.3	-0.03	0.45	0.63	1	0.15	0.15	0.16	0.19	0.31	0.38	0.21	0.22	0.13	0.18	0.32	0.28
Shoot_com	0.18	-0.06	0.54	0.01	-0.11	0.31	-0.09	0.32	-0.29	0.24	0.29	0.09	0.09	0.27	0.13	0.15	1	0.57	0.2	0.49	0.5	0.19	0.52	0.78	0.36	-0.1	0.55	0.2
Off._com	0.37	0.02	0.27	-0.06	-0.19	0.3	-0.12	0.33	-0.49	0.51	0.46	0.21	0.27	0.41	0.1	0.15	0.57	1	0	0.58	0.67	0.13	0.47	0.64	0.56	-0.16	0.52	0.15
Off._draw	0.09	0.03	0.23	0.2	0.12	-0.11	0.17	-0.01	0.13	0.06	0.16	0.27	0.05	0.08	0.1	0.16	0.2	0	1	-0.12	-0.05	0.3	-0.11	0.24	-0.24	0.14	0.13	0.02
ORB	0.15	-0.05	0.13	-0.41	-0.45	0.49	-0.43	0.38	-0.66	0.48	0.14	-0.05	0.19	0.31	0.09	0.19	0.49	0.58	-0.12	1	0.66	0.07	0.64	0.46	0.72	-0.07	0.45	0.3
DRB	0.53	0.07	0.2	0.08	-0.17	0.27	-0.09	0.3	-0.41	0.39	0.6	0.38	0.14	0.65	0.21	0.31	0.5	0.67	-0.05	0.66	1	0.33	0.56	0.57	0.78	-0.09	0.71	0.31
STL	0.46	0.07	0.19	0.36	0.08	-0.12	0.16	-0.02	-0.04	0.05	0.53	0.64	0.13	0.54	0.26	0.38	0.19	0.13	0.3	0.07	0.33	1	0.12	0.37	-0.06	0.71	0.54	0.38
BLK	0.21	-0.01	0.1	-0.2	-0.28	0.43	-0.27	0.38	-0.43	0.36	0.18	-0.04	0.11	0.34	0.13	0.21	0.52	0.47	-0.11	0.64	0.56	0.12	1	0.47	0.55	-0.05	0.5	0.44
PF	0.33	-0.01	0.23	0.1	-0.11	0.21	-0.07	0.25	-0.3	0.31	0.43	0.27	0.23	0.3	0.12	0.22	0.78	0.64	0.24	0.46	0.57	0.37	0.47	1	0.36	0.01	0.5	0.22
DRB.	0.17	0.03	0.01	-0.34	-0.35	0.42	-0.36	0.31	-0.53	0.42	0.12	-0.04	0.24	0.34	0.07	0.13	0.36	0.56	-0.24	0.72	0.78	-0.06	0.55	0.36	1	-0.11	0.43	0.37
STL.	0.02	0.01	-0.03	-0.05	-0.05	-0.13	-0.08	-0.19	-0.01	-0.06	-0.05	0.22	0.23	0.16	0.15	0.18	-0.1	-0.16	0.14	-0.07	-0.09	0.71	-0.05	0.01	-0.11	1	0.19	0.54
DWS	0.47	0.11	0.5	0.15	-0.07	0.2	0	0.27	-0.26	0.24	0.51	0.4	0.09	0.69	0.52	0.32	0.55	0.52	0.13	0.45	0.71	0.54	0.5	0.5	0.43	0.19	1	0.55
DBPM	0.09	0.16	0.04	-0.31	-0.16	0.32	-0.27	0.25	-0.22	0.17	-0.1	0.1	0.31	0.43	0.44	0.28	0.2	0.15	0.02	0.3	0.31	0.38	0.44	0.22	0.37	0.54	0.55	1

Chapter 3: Data Analysis

Section 3.1: Simple Linear Model

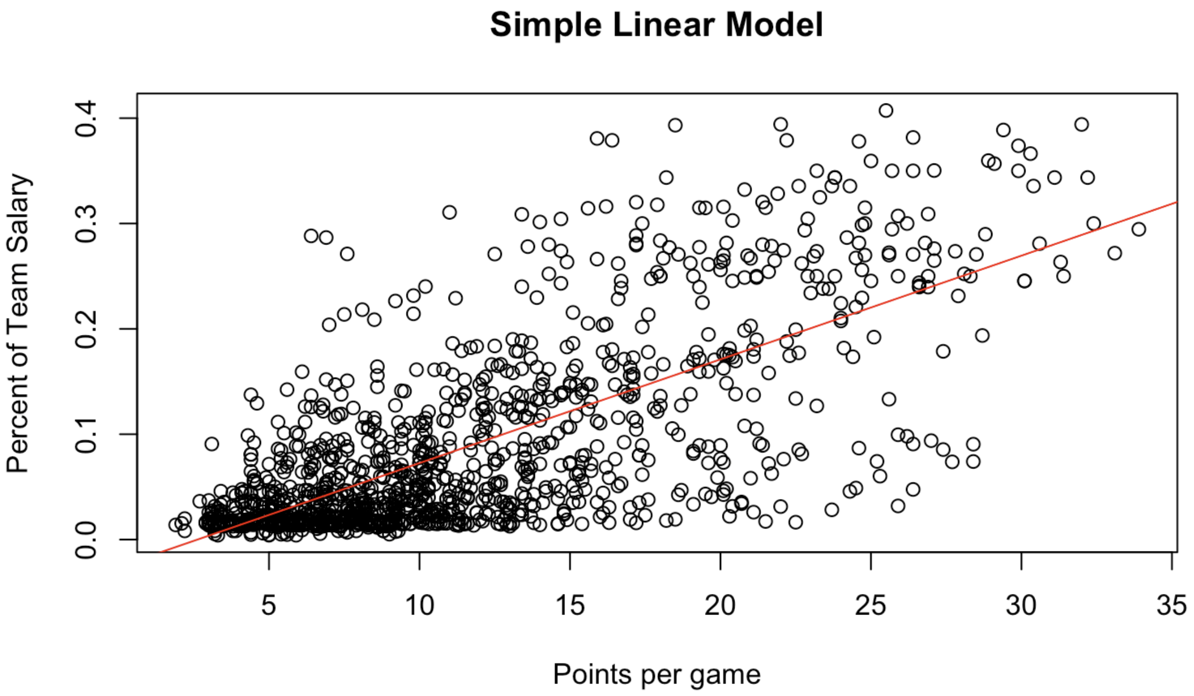
The first model that I wanted to test the performance of was a simple linear model, with the predicting variable being Points per game, as that had the highest overall correlation with Salary when looking at the correlation matrices from before.

After coding this model into R, the output was the equation as follows:

$$\hat{Salary} = -0.0258305 + 0.0098487(PTS)$$

The model was able to perform relatively well, and resulted in an R^2 value of 0.5042, and adjusted R^2 value of 0.5037 with residual standard error of 0.06404 on 1213 degrees of freedom.

The plot below shows the Salary on the y-axis and Points per game on the x-axis. The red line shown in the plot is the equation produced by the simple linear model.



It is essential to remember that the Salary is given in percent of the team's salary cap for that certain year, since this study was conducted over a 5 year span. For example, if we wanted to predict the Salary of a player averaging 20 points per game in the year 2025, we would plug 20 into the equation:

$$\widehat{Salary} \ = \ -\ 0.0258305 \ + \ 0.0098487(20) \ = 0.1711435$$

In this case, the simple linear model would predict the player to use about 17.11% of the team's salary. Given that this example player was playing in the year 2025, we would simply multiply this 0.17 times the salary cap for that year, which is 154.647 million.

$$Predicted\ Salary \ = \ 0.1711435(154.647) \ = \ 26.4668288445 \approx 26.47\ million\ per\ year$$

The model would then predict the player to be making 26.47 million dollars per year. To give context, some players today averaging around 20 points per game are Ja Morant (making 36 million per year), Pascal Siakam (42 million per year), Miles Bridges (27 million per year), Scottie Barnes (10 million per year), and Domantas Sabonis (40.5 million per year).

As you can see in these values, there is a lot of variability for players averaging 20 points per game, and only a few of them fall around the range of 26.47 million like the model predicted. Using this we can assume this model isn't the most effective, but to give a more certain answer we can perform a test to find the Training and Test Error. The Training Error is the error from predictions using the data sets from NBA Seasons 2021-2024, and the Test Error is using data from the 2025 NBA Season that was not used when training the model.

$$\begin{gathered} Training\ MSE\ = 0.00409395408548487 \\ Test\ MSE\ = \ 0.00421935399582417 \end{gathered}$$

Since we haven't performed any other models yet, we don't know how good or bad these numbers are, and therefore how the model has performed. But it will be in our best interest to keep them in mind when conducting further research and creating more complicated models.

Section 3.2: Multiple Linear Model

The next model I chose to test is a multiple linear model, with all 26 variables included in the correlation matrix shown before (except for Age). After running the code in R, we can determine that the formula used to determine the model is:

$$\begin{gathered} \hat{Salary} \ = \ 0.0687642 \ - \ 0.0011216(G) \ + \ 0.0029934(X3P) \ - \ 0.0497894(X3P\%) \\ - \ 0.0748918(X2P\%) \ - \ 0.0015254(FT\%) \ + \ 0.0708171(TS\%) \ + \ 0.0165238(X3PAr) \\ + \ 0.0306694(FTr) \ + \ 0.0034641(PTS) \ + \ 0.0058266(AST) \ + \ 0.0005486(TOV\%) \\ + \ 0.0105818(VORP) \ + \ 0.0022242(OnCourt) \ - \ 0.0011019(On.Off) \ - \ 0.0001524(Shoot_{com}) \\ + \ 0.0008222(Off._{com}) \ - \ 0.0001302(Off._{draw}) \ - \ 0.0063340(ORB) \ + \ 0.0128799(DRB) \\ + \ 0.0340839(STL) \ + \ 0.0160640(BLK) \ - \ 0.0019940(PF) \ - \ 0.0019272(DRB\%) \\ - \ 0.0120977(STL\%) \ + \ 0.0005835(DWS) \ - \ 0.0052148(DBPM) \end{gathered}$$

I know this may be a lot to handle, but overall this equation is very similar to the simple linear one we saw before. For every variable name (such as G or X3P) we want to input the value for the statistic of the certain season. For example, if we wanted to analyze Devin Booker from the 2023-2024 NBA Season. After inputting all the variables seen above into the equation, we get a result of:

$$\hat{Salary} \ = 0.22922880(136.021) \ = \ 31.18\, million\, per\, year$$

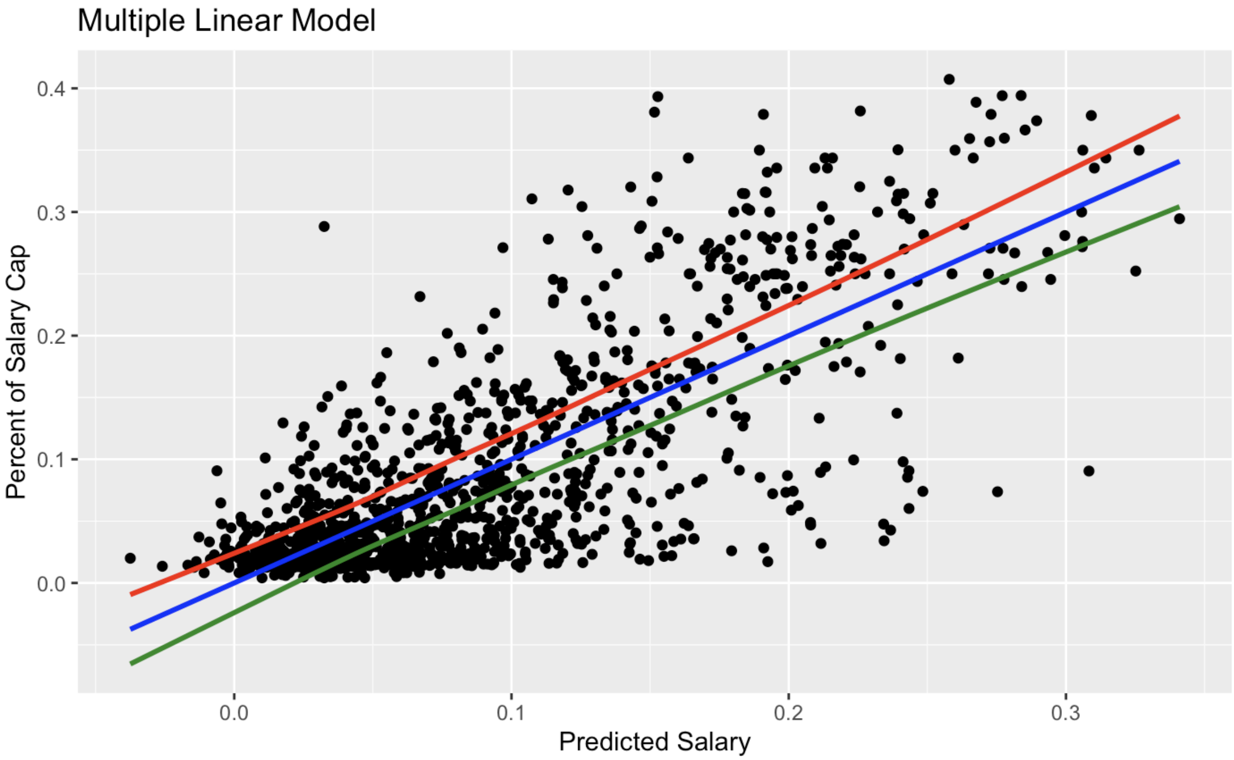
In reality, Devin Booker last year made 36.02 million. This model would suggest that Devin Booker is being overpaid based on his performative statistics. This is up for debate, as some say Devin Booker is being overshadowed by the Sun's addition of Kevin Durant this past season.

Looking at the Training MSE and Test MSE, we find that this model produces values of:

$$Training\ MSE\ = 0.00340870499371067$$
$$Test\ MSE\ =\ 0.00402765859969061$$

Comparing these values to the simple linear model, both the Training MSE (For Linear Model: 0.00409395408548487) and the Test MSE (For Linear Model: 0.00421935399582417) for the Multiple Linear Model are lower than that of the Simple Linear Model. This is a good sign in relation to accuracy of the model. This Multiple Linear Model also has an R^2 value of 0.5871 and adjusted R^2 value of 0.5781, both of these values are also better than that of the simple linear model (as a value closer to 1 means more accuracy). Therefore using these tests, we can conclude that the Multiple Linear Model is more suitable to predict Salary than the simple linear model, and will be used to compare against further models.

The plot below shows the Real Percent of Salary on the y-axis and Predicted Percent of Salary on the x-axis. The blue line shows a straight line of y=x, where most of the points should lie. More points surrounding this line would mean a greater accuracy of the prediction. The red and green lines shown are a 99% confidence upper and lower limits of the prediction model. All points that lie above the red line are players that are predicted to be “Overpaid” and all points below the green line are players that the model has predicted to be “Underpaid”.



Looking at the graph we can see that the points on the plot follow a somewhat pattern along the blue line, but there seem to be a lot of points above the line compared to below. This suggests that there may be a more accurate model that can be used that doesn't rely on linearity such as this model does.

Section 3.3: Log Multiple Linear Model

Looking at the Simple and Multiple Linear Models, they seemed to perform really well near the top and bottom of the Salaries, but as you proceeded to the lower performing players, I found that some of the Predicted values for Salary was negative. This is a problem point, as obviously a player cannot be paid a negative amount of money. To combat this, I wanted to transform our data by forming a model to predict the Log(Salary), because after transforming it back to the regular Salary, it will never result in a negative number.

The model can be written as seen below:

$$\begin{gathered} log(\hat{Salary}) \;=\; -\,3.709040 \;-\; 0.007009(G) \;+\; 0.095511(X3P) \;-\; 0.634431(X3P\%) \\ -\,1.297936(X2P\%) \;+\; 0.152092(FT\%) \;+\; 1.468159(TS\%) \;+\; 0.145458(X3PAr) \\ 0.023091(FTr) \;+\; 0.031909(PTS) + \; 0.084216(AST) \;-\; 0.004236(TOV\%) \;-\; 0.019603(VORP) \\ +\; 0.027854(OnCourt) \;-\; 0.016411(On.Off) \;-\; 0.001801(Shoot_{com}) \;+\; 0.009558(Off._{com}) \\ -\; 0.001384(Off._{draw}) \;+\; 0.041666(ORB) \;+\; 0.125794(DRB) \;+\; 0.810130(STL) \\ +\; 0.229719(BLK) \;+ 0.009655(PF) \;-\; 0.016862(DRB\%) \;-\; 0.324540(STL\%) \\ +\; 0.007587(DWS) \;-\; 0.021770(DBPM) \end{gathered}$$

If we were to look at Devin Booker again, this Log Salary model would predict that he would be making:

$$\hat{Salary} \;= (136.021)e^{-1.6489814} \;=\; 26.15\, million\; per\; year$$

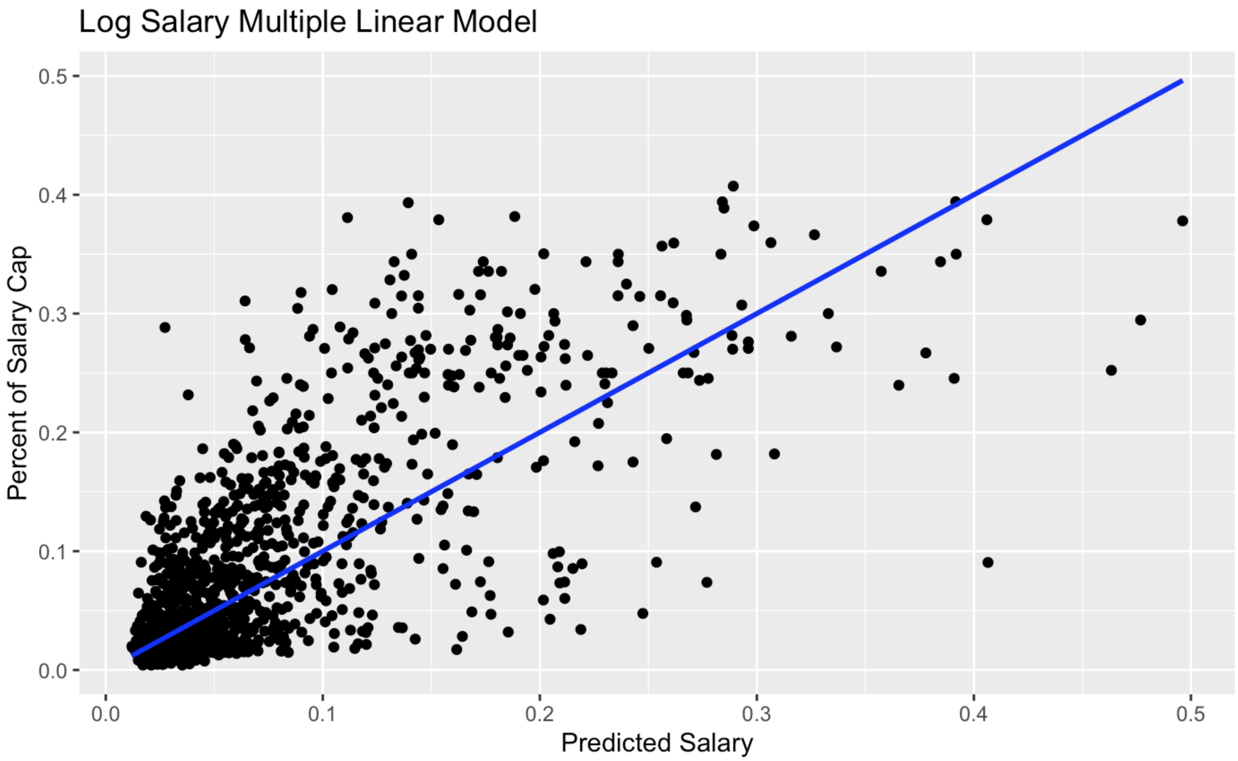
As we talked about before, Devin Booker made 36.02 million last year. According to this model Booker is severely overpaid, as the values are very different from each other. Since the difference is so great, we can predict there might be a problem with the prediction of this model, but there are more tests to be done.

This model results in an R^2 value of 0.5323 and an adjusted R^2 value of 0.522. Both of these values are lower than that of the Multiple Linear Model, so this signifies that the Multiple Linear Model may be more accurate. To make sure the Multiple Linear Model is more accurate, we will once again look at the Training and Test MLE.

$$\begin{gathered} Training\; MSE \;=\; 0.00393383801433403 \\ Test\; MSE \;=\; 0.00462388376241877 \end{gathered}$$

Compared to the Multiple Linear Model, the Training and Test MSE values for the Log Salary Model are higher. This means that there is significant evidence to conclude that the Multiple Linear Model is more accurate than this model.

Similar to the last model, the plot below shows the Real Percent of Salary on the y-axis and Predicted Percent of Salary on the x-axis with the blue line being the fit of the model, and under that line representing players being "Underpaid" while on top of the blue line is "Overpaid".

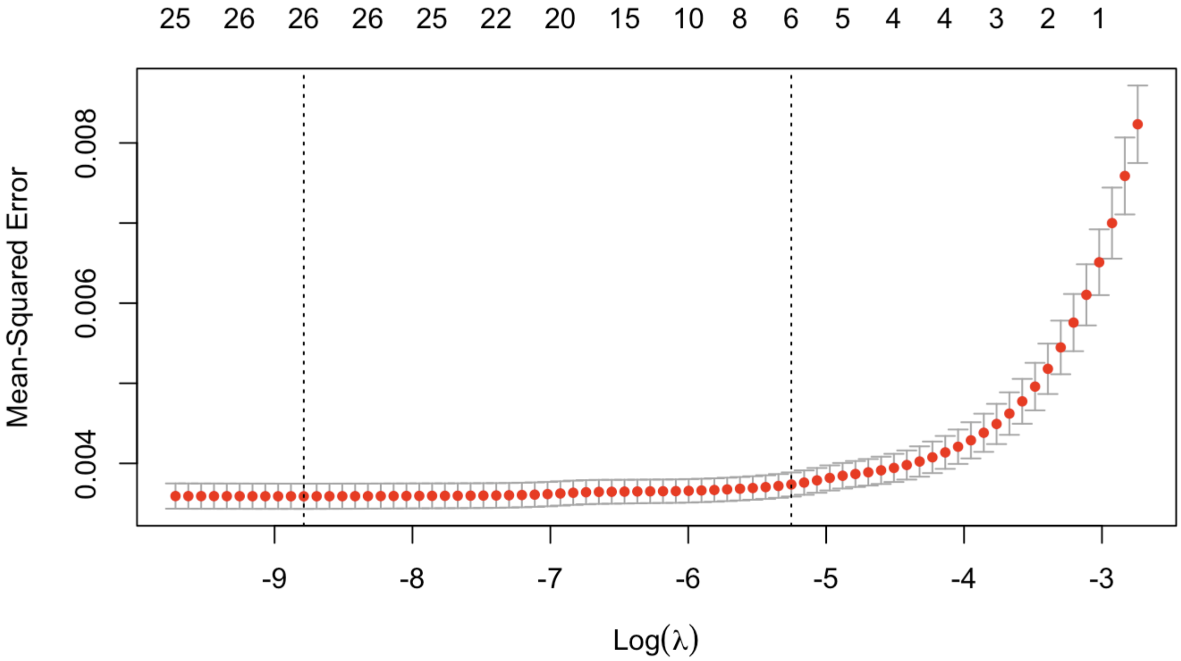


As we can see with this model, the points on the plot do not seem to be hugging the blue line, and therefore this would suggest it is not very accurate, and other models that we have seen before (such as the Multiple Linear Model) are more fit for prediction. This is backed up by the values we saw in the Test and Training MLE, as they are the highest we have seen so far.

Section 3.4: LASSO Model

The next model that I attempted to try was a LASSO model. A LASSO model is useful in data analysis and regression because it is a variable selector. This means that after performing analysis, it automatically eliminates some variables that it finds insignificant.

The odd thing about this analysis is that during the LASSO analysis, the model determined that no variables should be removed, but simply adjusted the coefficients for variables to be put into a multiple linear equation. The lambda that was used to determine these coefficients was 0.0001525505, and was chosen using the graph below:



The equation of the regression line determined through LASSO regression is:

$$\hat{Salary} = 0.065083558 - 0.001069752(G) + 0.00384556(X3P) - 0.045224441(X3P\%) - 0.064673352(X2P\%) + 0.000975103(FT\%) + 0.055511508(TS\%) + 0.01406392(X3PAr) + 0.02836791(FTr) + 0.003753058(PTS) + 0.006204814(AST) + 0.000359313(TOV\%) + 0.009705538(VORP) + 0.002067622(OnCourt) - 0.000971658(On.Off) - 0.000163337(Shoot_{com}) + 0.0007598022(Off._{com}) - 0.0001168903(Off._{draw}) - 0.0045743127(ORB) + 0.0115864368(DRB) + 0.0289501449(STL) + 0.0143012903(BLK) - 0.0007829850(PF) - 0.0016764475(DRB\%) - 0.0110385965(STL\%) + 0.0006411497(DWS) - 0.0036037539(DBPM)$$

Similar to the previous examples, I think it is best to look at Devin Booker as an example in this case:

$$\hat{Salary} = 0.2204085021(136.021) = 29.98\ million\ per\ year$$

Like we said before, Devin Booker made 36.02 million last year. Therefore according to this model he is slightly overpaid. This is similar to the number that was given by the Multiple Linear Regression model, which makes sense because like in that model, all variables were included in the LASSO regression model.

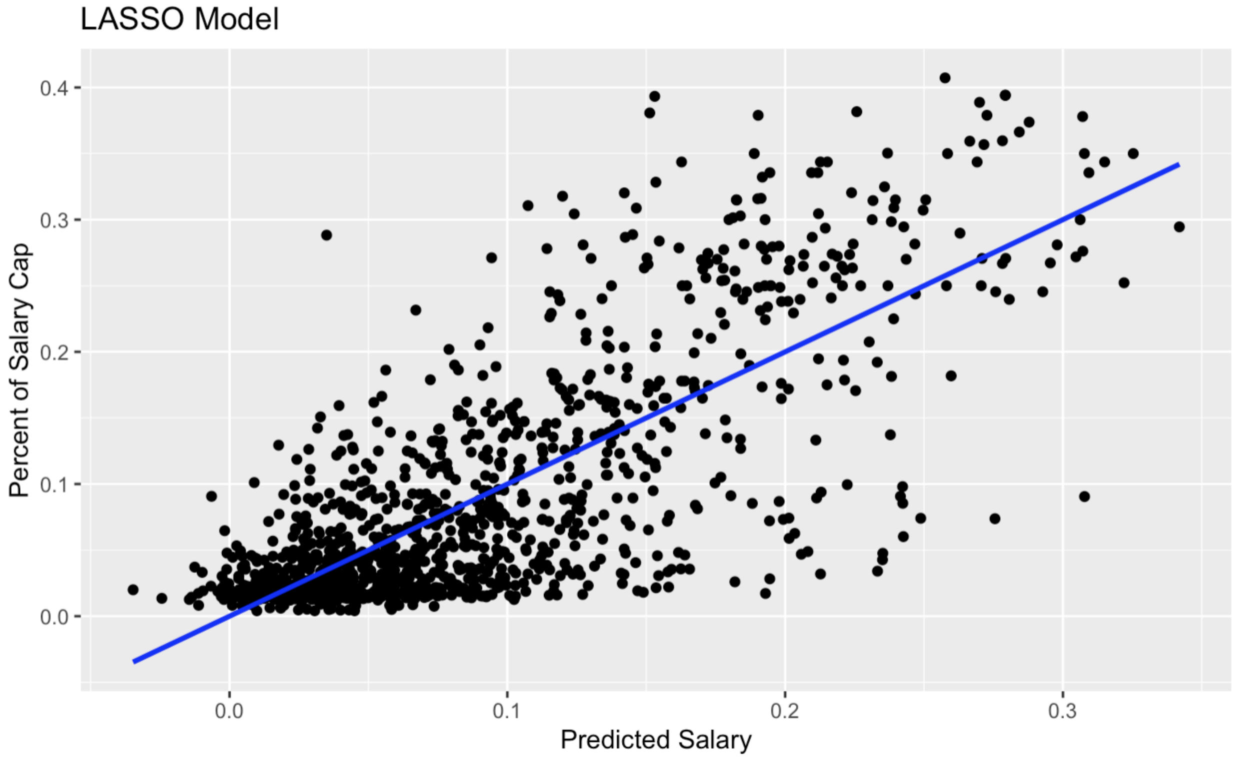
Looking at the Training MSE and Test MSE, we find that this model produces values of:

$$Training\ MSE = 0.00341092081087323$$

$$Test\ MSE\ =\ 0.00399414738247675$$

Comparing these values to the multiple linear model, the Training MSE is higher, but the Test MSE is slightly lower. Because Test MSE is more important in determining the reliability of a model, we will conclude that the LASSO regression model is the most accurate so far, and will use it as the basis for testing further models.

Finally, the graph below shows the relationship between the Predicted Percent of Salary (x-axis) and the Actual Percent of Salary (y-axis). If this model is somewhat accurate, the points on the graph should relatively hug the line.

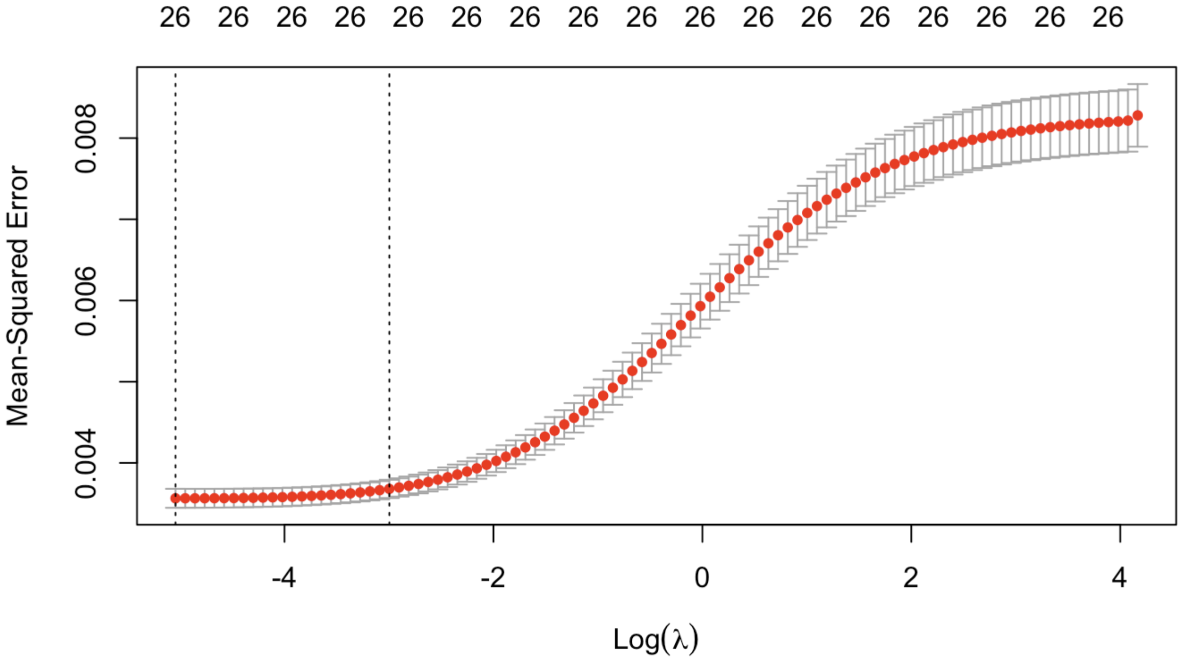


We can see that the points follow a somewhat linear path, but there seems to be a high variance and we will continue to test further models.

Section 3.5: Ridge Regression Model

The next model that I attempted to try was a Ridge Regression model. As opposed to the LASSO model, the Ridge model will change the coefficients of the multiple linear regression line, all the while including all of the variables. But because LASSO included all variables as well, I was curious to see if the models would be exactly the same. It turns out that they were different, because of the different alpha being used to calculate the coefficients. In the Ridge model, the alpha used was 0, whereas in LASSO it was 1.

The lambda that was used to determine these coefficients was 0.006451733, and was chosen using the graph below:



The equation of the regression line determined through LASSO regression is:

$$\hat{Salary} = 0.0500423456 - 0.0010217446(G) + 0.0081236317(X3P) - 0.0385367233(X3P\%) \\ - 0.0528551147(X2P\%) + 0.0138536582(FT\%) + 0.0412532324(TS\%) + 0.0049574631(X3PAr) \\ + 0.0355338967(FTr) + 0.0031364712(PTS) + 0.0069192242(AST) + 0.0001066338(TOV\%) \\ + 0.009440592(VORP) + 0.001577143(OnCourt) - 0.000574273(On.Off) - 0.00019133(Shoot_{com}) \\ + 0.0007529222(Off._{com}) - 0.0001535961(Off._{draw}) - 0.0032628422(ORB) + 0.0079647760(DRB) \\ + 0.0266176842(STL) + 0.0125767569(BLK) + 0.0011662508(PF) - 0.0009412442(DRB\%) \\ - 0.0108427406(STL\%) + 0.0050032978(DWS) - 0.0035919519(DBPM)$$

Once again we will look at Devin Booker’s Salary to put this into context:

$$\hat{Salary} = 0.2184956735(136.021) = 29.72\ million\ per\ year$$

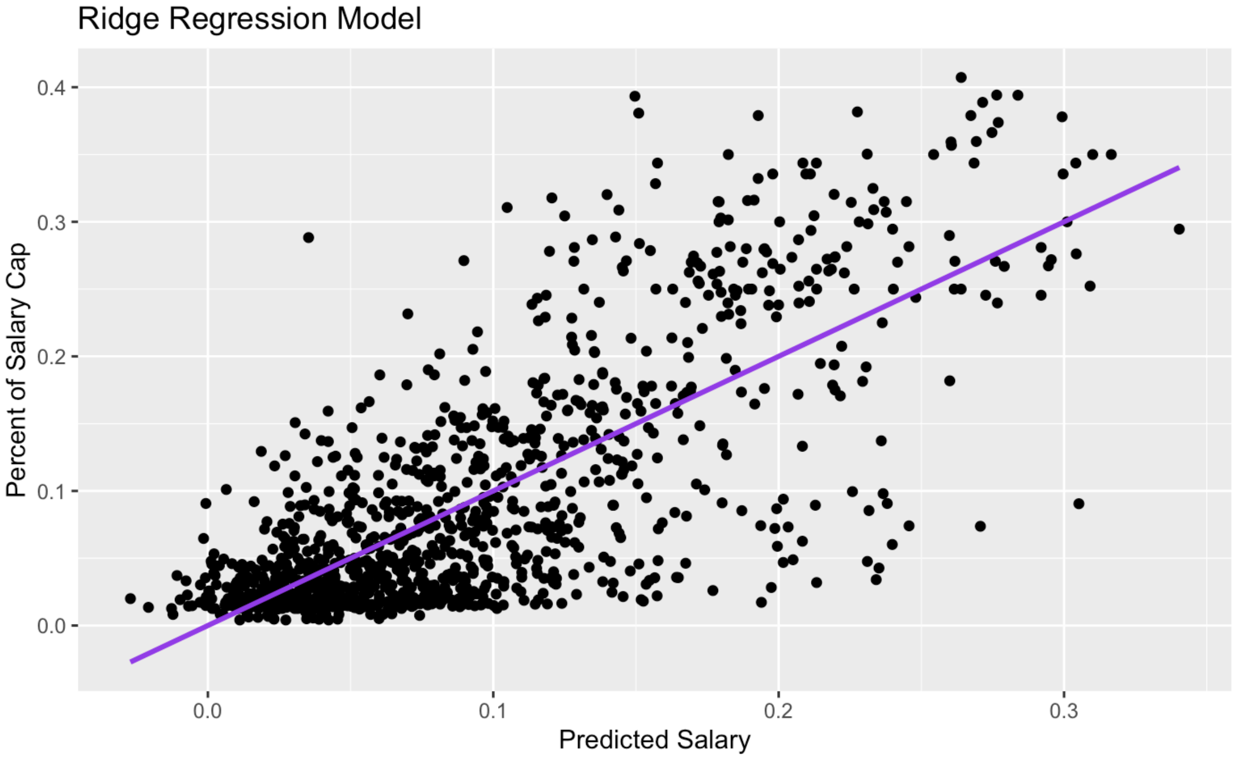
This model has Booker making slightly less than the LASSO model, which gave us a value of 29.98 million per year. This shows how similar the LASSO and Ridge regression models are.

Looking at the Training MSE and Test MSE, we find that this model produces values of:

$$Training\ MSE\ = 0.00342764507536179$$
$$Test\ MSE\ =\ 0.00397753757962287$$

Once again we will compare these values to our current leader, which is the LASSO model. The LASSO model had a Training MSE of 0.00341092081087323 and Test MSE of 0.00399414738247675. Therefore we can see that although the Training MSE is higher for the Ridge model, the Test MSE is lower than that of LASSO. Therefore we can conclude that the Ridge Regression model is the most accurate so far.

Finally, the graph below shows the relationship between the Predicted Percent of Salary (x-axis) and the Actual Percent of Salary (y-axis). If this model is somewhat accurate, the points on the graph should relatively hug the purple line.



We can see that the points follow a somewhat linear path, but there seems to once again be a high variance so we will continue to test further models.

Section 3.6: Polynomial Regression Model

The next model that I wanted to try is a polynomial regression model. This model makes the assumption that not all predictors have a linear relationship with a player's Salary. This model might give us an insight into if any variables have a quadratic or cubic relationship with the Salary.

After performing analysis on which variables are significant in the second and third degrees, I came to the conclusion that the only variables that need to be polynomial in the model are PTS and On.Off. For PTS, all three degrees are significant, whereas for On.Off the first and second degree are significant. At this point, I also removed Games as a predictor, because it was not significant, and after further thought the amount of games only regulates how much a player gets paid in rare occurrences (i.e. a player that is very injury-prone). Knowing this, we can rewrite the model as:

$$\overset{\wedge}{Salary} = -0.0236035 + 0.0066971(X3P) - 0.0458489(X3P\%)$$
$$- 0.0937810(X2P\%) + 0.0152754(FT\%) + 0.1616754(TS\%) + 0.0076089(X3PAr)$$
$$+ 0.0282633(FTr) + 0.6060956(PTS, 1) + 0.3748323(PTS,\ 2) - 0.2226729(PTS, 3)$$
$$+ 0.0082093(AST) - 0.0006238(TOV\%) + 0.0018127(VORP) + 0.0022833(OnCourt)$$
$$- 0.2132399(On.Off, 1) + 0.1289594(On.Off, 2) - 0.0004768(Shoot_{com}) + 0.0005186(Off._{com})$$
$$- 0.0001125(Off._{draw}) - 0.0039741(ORB) + 0.0165117(DRB) + 0.0442942(STL)$$
$$+ 0.0248388(BLK) + 0.0099439(PF) - 0.0017670(DRB.) - 0.0131720(STL.)$$
$$- 0.0038667(DWS) - 0.0046333(DBPM)$$

Keeping everything consistent, now let’s have a look at Devin Booker’s predicted salary:

$$\overset{\wedge}{Salary} = 0.23587081627(136.021) = 32.08\, million\, per\, year$$

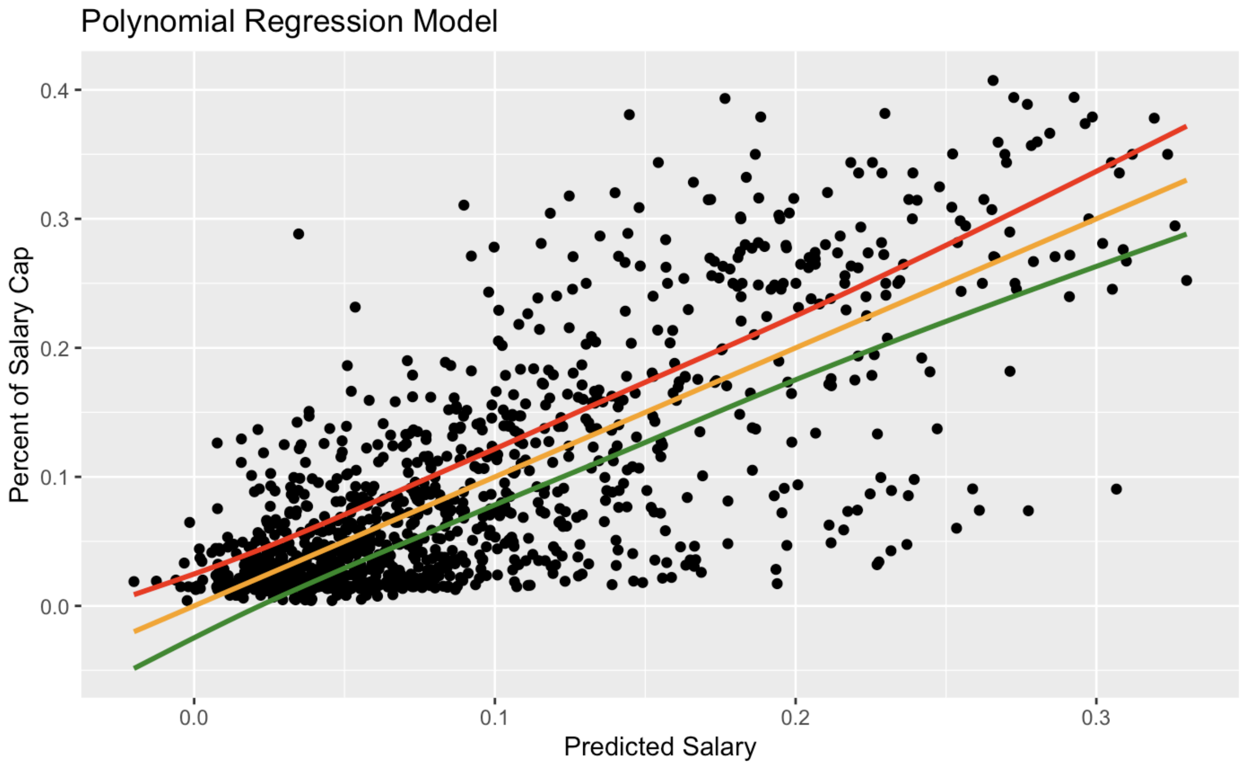
Given that Booker made around 36 million dollars last season, this is the most accurate model to date. The prediction of 32 states that Booker is being slightly overpaid, but overall is valued very close to what his performance states.

Once again looking at Training and Test MSE, the values are as follows:

$$Training\ MSE = 0.00337073918387944$$
$$Test\ MSE = 0.00390224269682603$$

These values are both lower compared to the Ridge Regression model, meaning that this model is more accurate and viable for prediction in the future. This model also has an R^2 value of 0.5917 and adjusted R^2 of 0.5821. These values are the highest we have seen so far.

The plot below shows the Real Salary on the y-axis and Predicted Salary on the x-axis. The orange line shows a straight line of y=x, where most of the points should lie. More points surrounding this line would mean a greater accuracy of the prediction. The red and green lines shown are a 99% confidence upper and lower limits of the prediction model. All points that lie above the red line are players that are predicted to be “Overpaid” and all points below the green line are players that the model has predicted to be “Underpaid”.

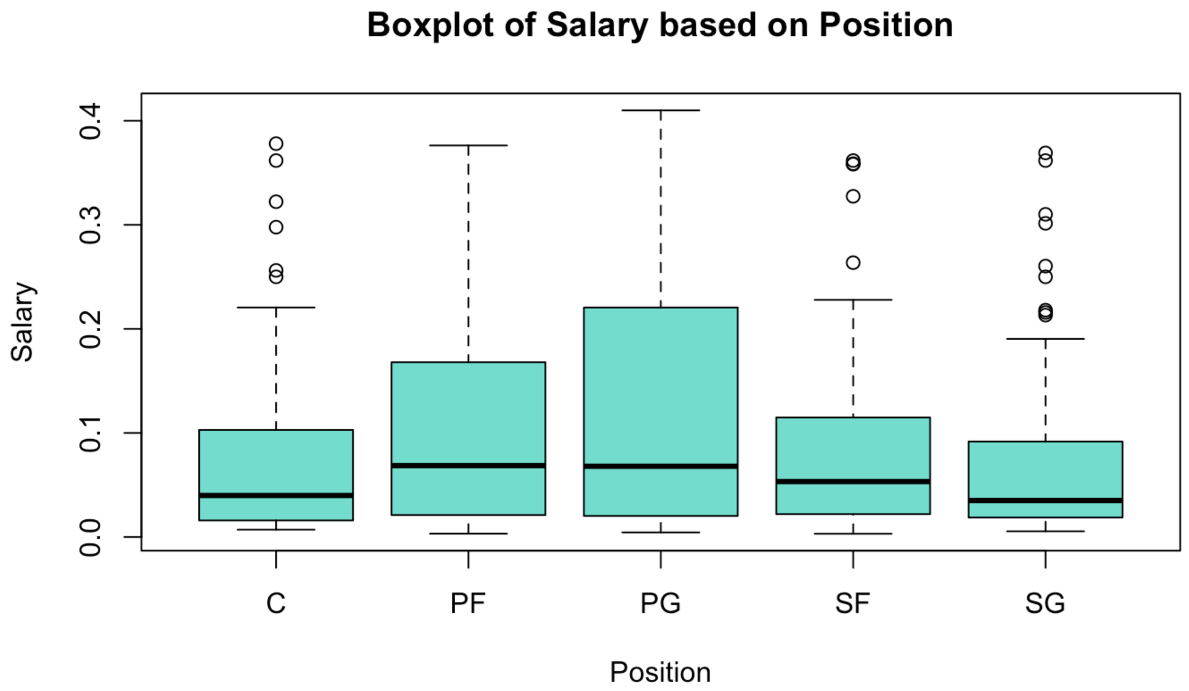


Looking at the graph once again, the points seem to hug the line, with a good amount of players being both below and above the red and green lines respectively. Although there seem to be more players above the red line, meaning that the majority of players are being overpaid, which I would want to get more evened out in the future. Therefore we can continue to look for a better model fit with the Polynomial Model as our new baseline.

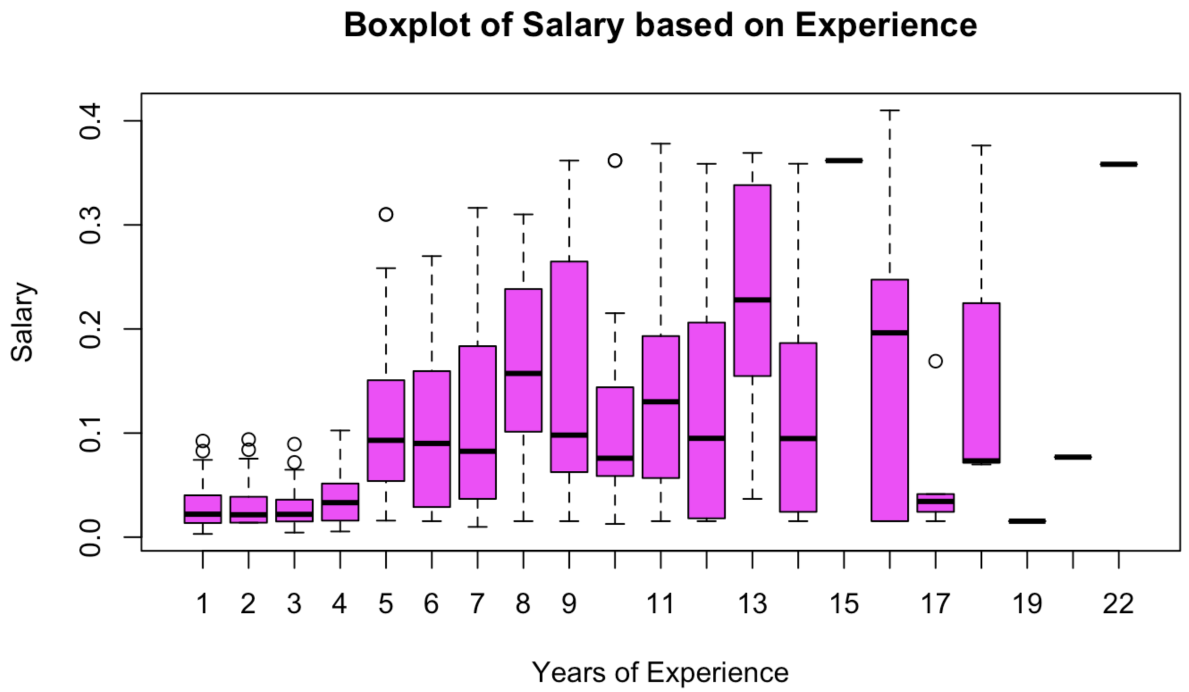
Section 3.7: LOESS Model

Up to now, we have simply been quantifying a way to track a player's performance. The higher the "Predicted Percent of Salary" means how well that player had performed on the court for that certain season. Now taking things beyond the court, there are a lot of factors that go into how much a player is making on the season. Factors like age, position, draft position are all things that are necessary to consider if we want a good estimate of salary.

These external factors may be more of a predictor of salary than you might think. It makes sense thinking about it from a GM's perspective. If a player is really young or really old, maybe offering a big contract isn't the smartest decision. In respect to position, according to my analysis, the current NBA is paying point guards significantly more than other positions (as seen in the box plot below), which makes sense because they are commonly the ones running the offense, and their performance is a good predictor of team success.



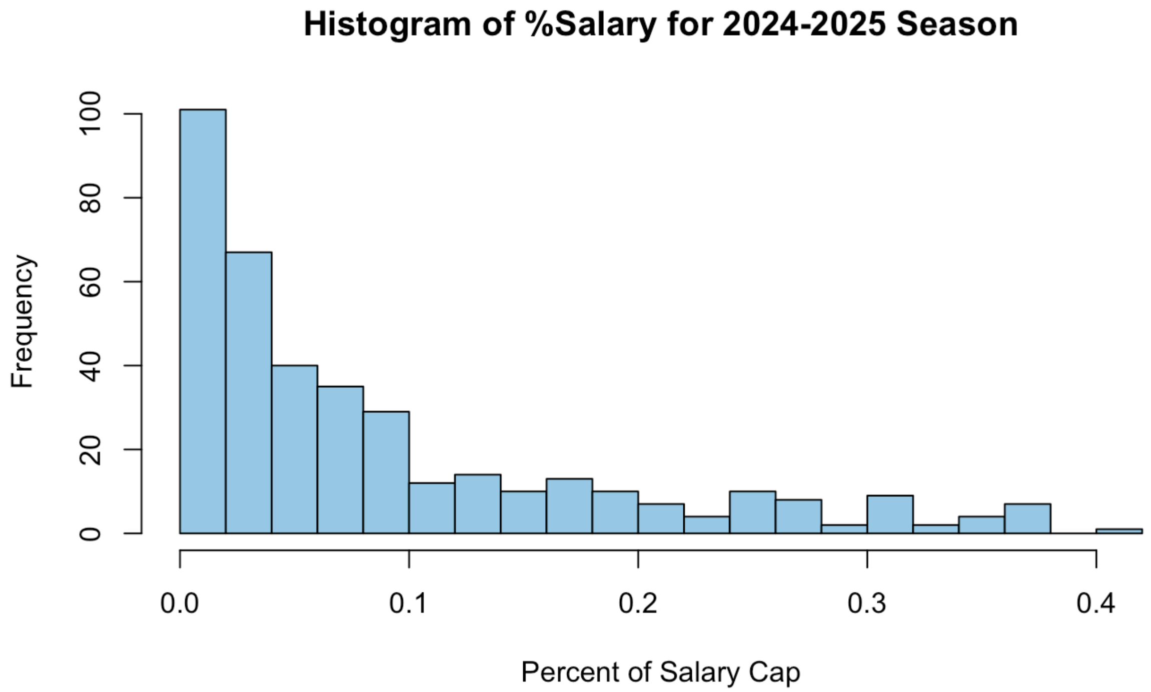
Similarly, this is the case for many other variables for NBA players, including Experience. The common rookie contract for an NBA player is 4 years, with the first two years being guaranteed. After these four years, the certain player has a chance to resign with the team, or test their market in free agency if the team that drafted them does not want to continue with them. Therefore, we see a big jump in salaries for players entering their 5th year in the league, which can be seen in the boxplot below:



Therefore, in our model, it would be essential to have this “Years Experience” variable in our data, because the model can be trained to increase the pay for 5th year players, and keep pay lower for those during their first four years in the league.

Although talent and performance might be normally distributed throughout the league, we cannot assume that this is the same for pay. To account for this, I decided to use a combination of the player performance model I just created using the Polynomial Model, and Local Regression (LOESS model) to gain a better insight on how salary is distributed in the NBA.

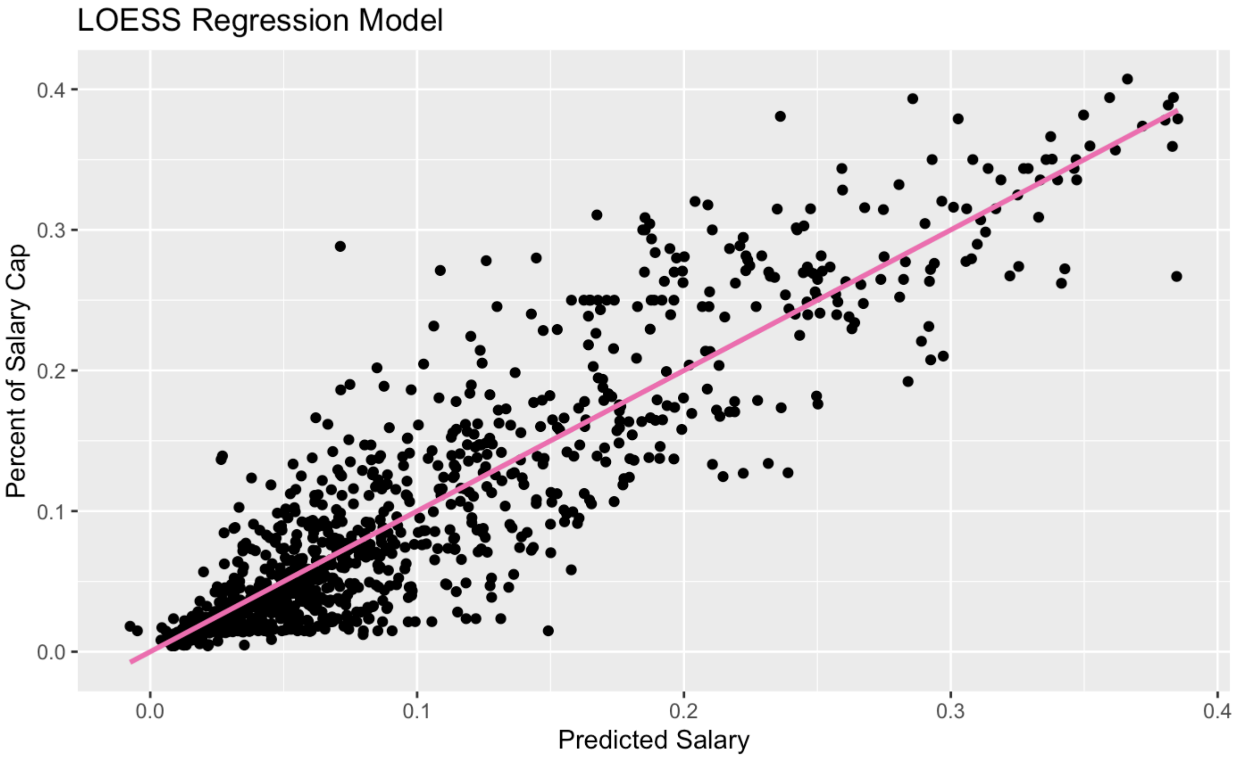
A LOESS model is a non-parametric regression method that fits a smooth curve to the data by analyzing the neighbors of a certain datapoint, and fitting multiple little models into different subsets of the data. This is essential for our model because as we discussed before, the distribution of salaries in the NBA is not normal as seen here:



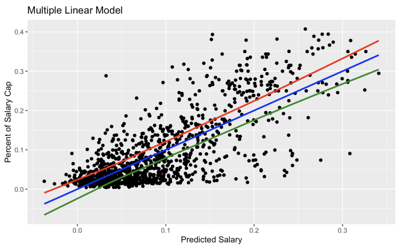
As you can see, a lot more players are getting paid a very small percentage of a team's salary cap, opposed to very few players getting paid over 30%. This should be reflected in our model, which is why we may benefit from using LOESS.

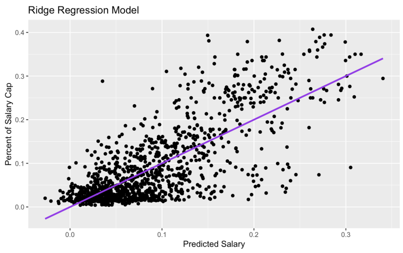
Using the prediction of the polynomial model from the last section as a "Performance" variable, I now added the variables of Pick Number (in their respective NBA Draft), Experience, and Position into a LOESS model. This model now is being used to predict salary, using Performance, Pick Number, Experience, and Position.

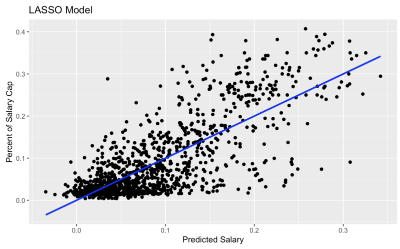
For a LOESS model, you cannot simply write out the equation like we have been doing in the past, instead I will just jump straight into the graph, errors, and R^2 values. The graph looks like this:



Compared to some other graphs we've seen, which I'll show below, this graph visibly hugs the line a lot tighter with fewer overall outliers.







The Test and Training Errors for the LOESS model are as follows:

$$Training\ MSE\ = 0.00138046562193216$$
$$Test\ MSE\ = 0.00202372217663988$$

These values are significantly lower than from the models we have seen before as the polynomial model had a Training MSE of 0.00337073918387944 and a Test MSE of 0.00390224269682603. The LOESS model produced an R^2 value of 0.8334 and Adjusted R^2 of 0.8333. This is also significantly higher than our polynomial model, which had respective values of 0.5917 and 0.5821.

Finally before we move on, we will come back to Devin Booker's predicted salary. After performing the analysis, the LOESS model predicted a salary of:

$$\hat{Salary} = 0.282293033(136.021) = 38.40\, million\, per\, year$$

This is quite the change from the last model, which stated he was being overpaid. As a reminder Booker made around $36 million last year, so this model looks pretty accurate. His predicted "Performance" that was retrieved from the polynomial model had him valued at $32 million per year, but given his age, draft pick number, and position, this number was able to increase around $6 million.

Section 3.8: Polynomial Regression Model Extended

The next model to test is a simple polynomial model including the external factors we just talked about, being Experience, Position, and Draft Pick. The model takes in these variables and the Performance metric we calculated with the Polynomial Regression model using on-court statistics, to predict the salary of the certain player.

The equation of the line is as follows:

$$\begin{gathered}\hat{Salary} = -0.09628 + 0.2.103(Performance, 1) + 0.1906(Performance, 2) \\ -0.0004695(Pk) + 1.077(Experience, 1) - 0.2358(Experience, 2) + 0.1542(Experience, 3) \\ + 0.004549(C - PF) + 0.007812(PF) - 0.04694(PF - SF) + 0.00248(PG) \\ - 0.02271(PG - SG) + 0.01514(SF) + 0.002786(SF - SG) + 0.002274(SG) \\ + 0.01292(SG - PG) - 0.006526(SG - SF)\end{gathered}$$

Analyzing Devin Booker once again, his predicted salary can be written as:

$$\hat{Salary} = 0.2466918(136.021) = 33.56\, million\, per\, year$$

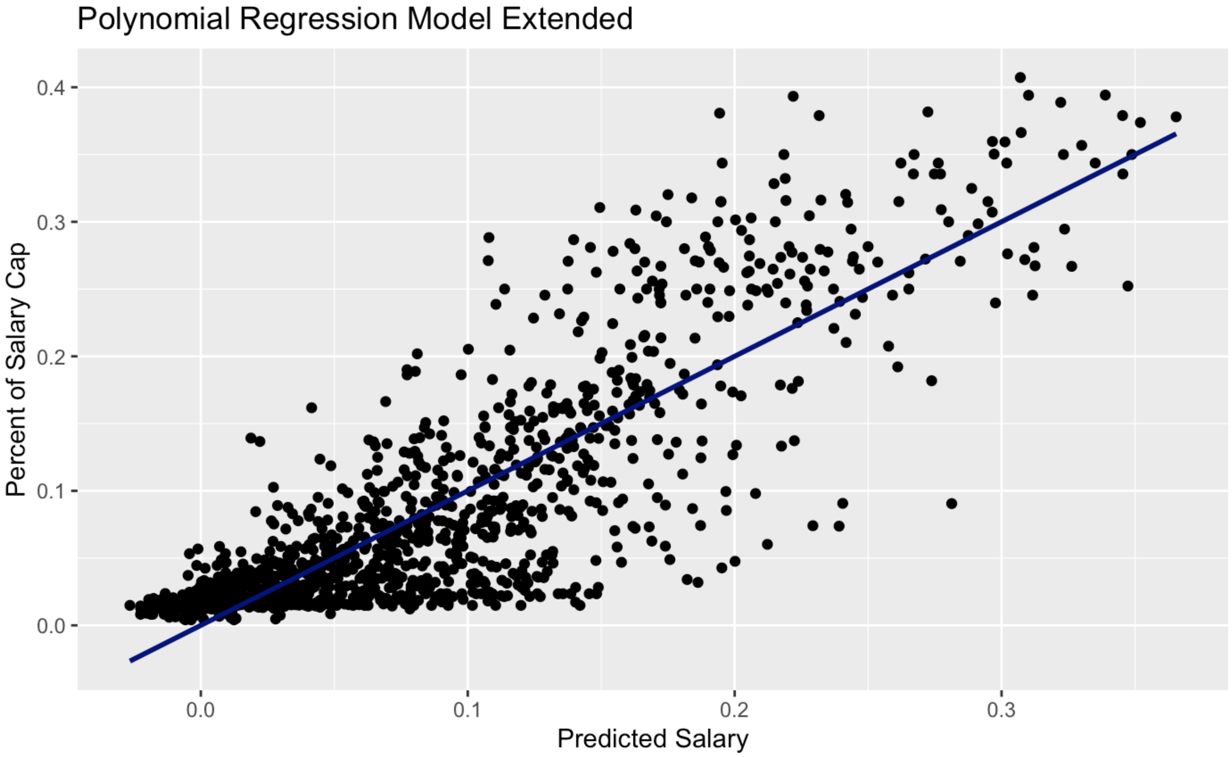
This is a relatively accurate value, being very close to Booker's real salary of $36 million per year. This is about equally as far as the LOESS model, which predicted Booker's salary to be around $38 million last year.

For Test and Training Error, the values are:

$$\begin{gathered}Training\ MSE = 0.0023585754841071 \\ Test\ MSE = 0.00305915948600013\end{gathered}$$

Looking at these values compared to the respective values from the LOESS model, these values are significantly higher in both aspects by around 0.001. Therefore we can conclude that although a pretty good model, with an R^2 value of 0.7143, it doesn't compare to that of the LOESS model, which we will continue to use as our top model.

If you are interested, here is the graph for the Extended Polynomial Regression Model:

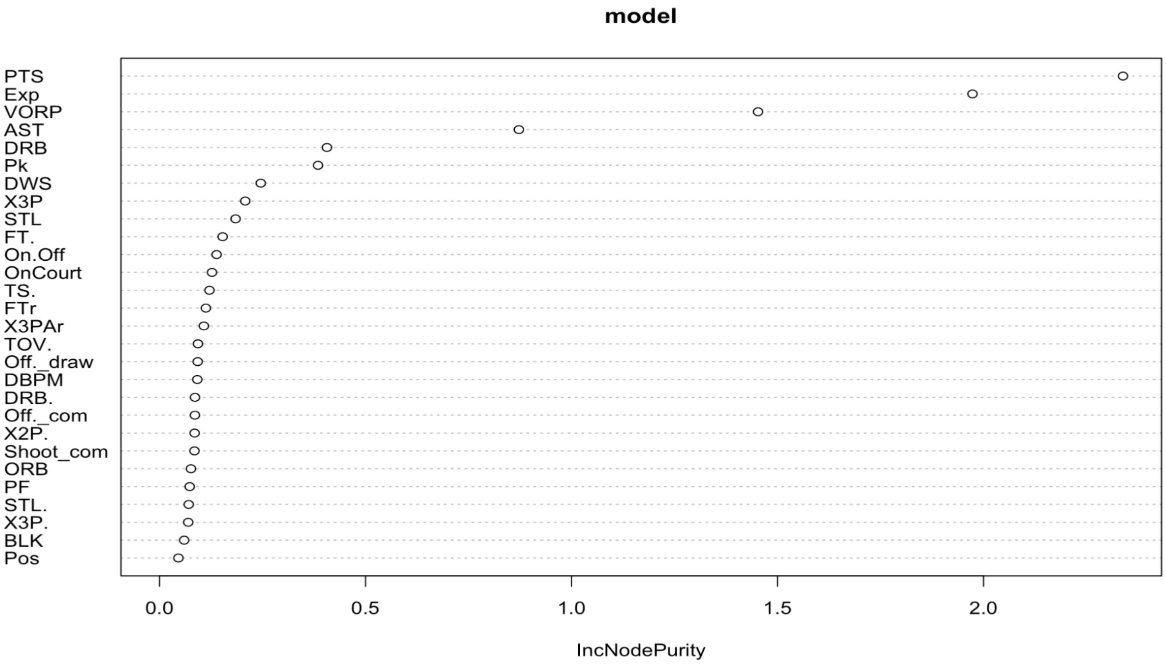


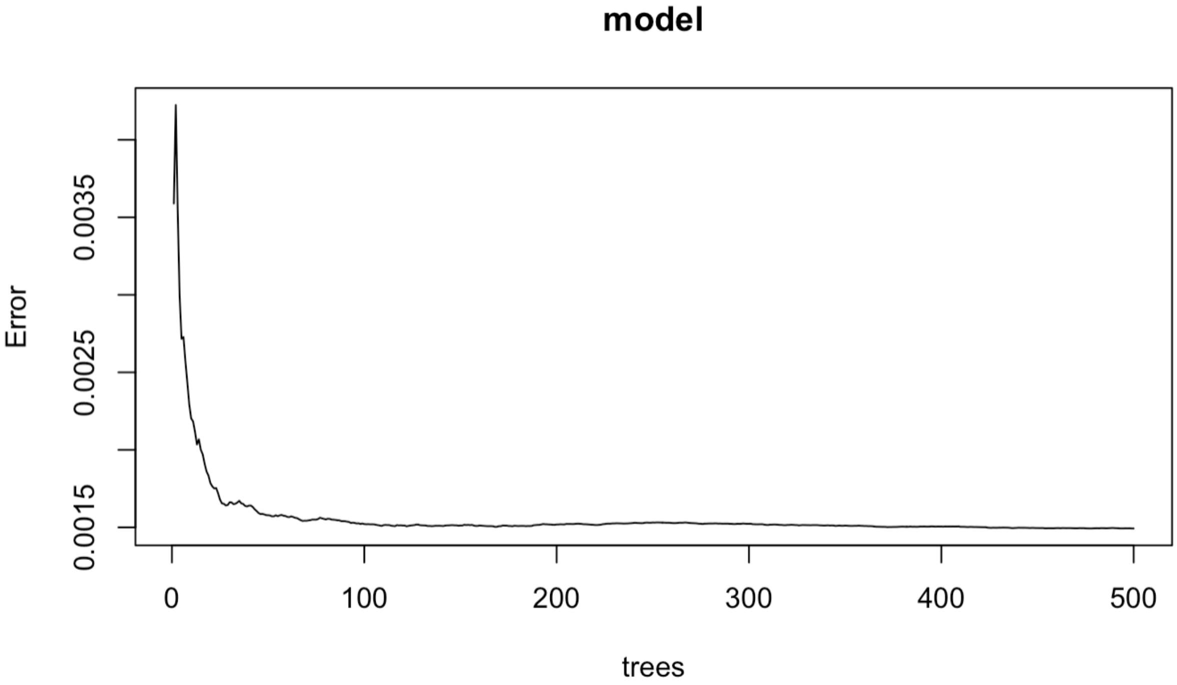
Section 3.9: Random Forest Model

Shifting gears from least squares regression models, I also wanted to try a Random Forest Model. A Random Forest Model is a model that uses decision trees to learn about complex relationships and patterns in the data that least squares regression methods might have missed. A Random Forest is a compilation and average of these decision trees, to help find an accurate prediction model for the data.

While a Random Forest Model is very accurate in most cases, and will most likely result in a high R^2 value. This can be double checked in our case by using the test data to retrieve the resulting Test Error of the model, and comparing it against the LOESS Model that we have created before.

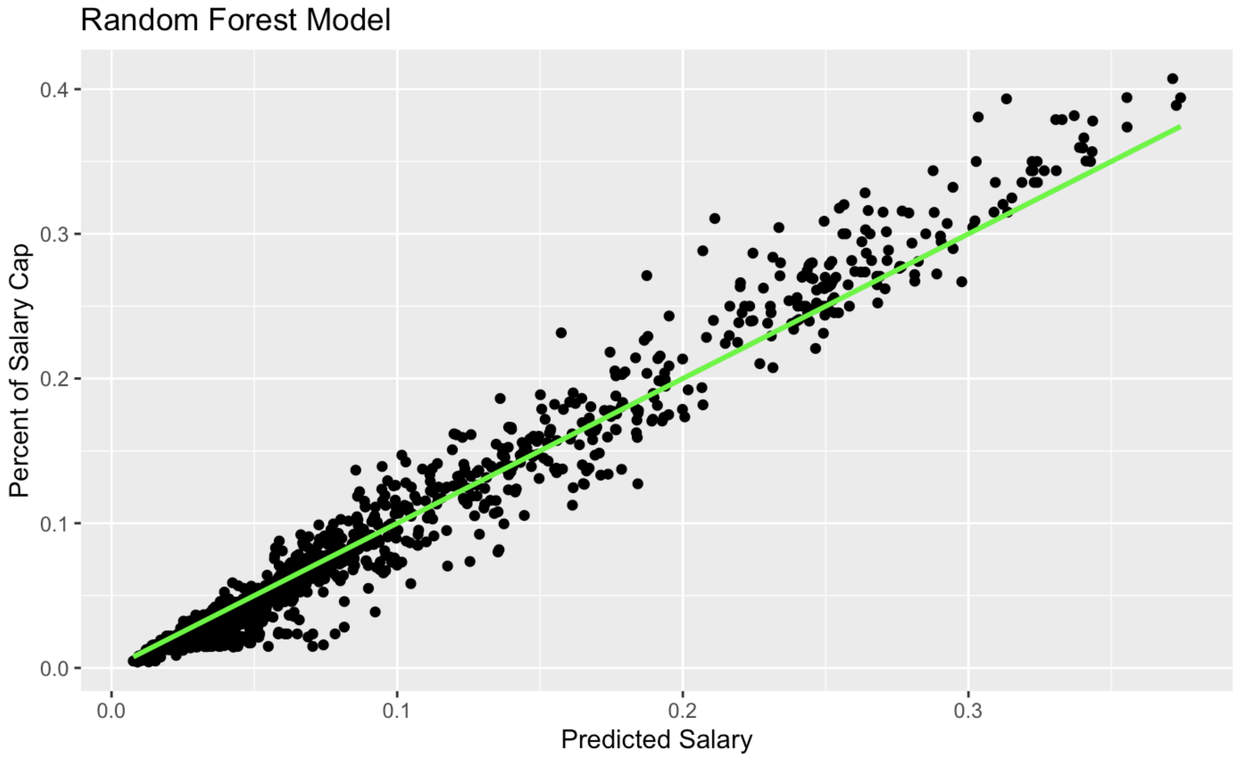
Similar to a LOESS Model, a Random Forest Model lacks interpretability, as you are not able to display a simple equation that represents the pattern in the data. Although we can visualize the most valuable predictors and optimized number of trees given the error, as we can see in both of these graphs:





We can see that the most valuable predictor when it comes to an NBA player's salary is points per game, followed by experience, VORP, assists per game, and defensive rebounds per game. This makes sense given our data, as points, rebounds, and assists are the three main statistics that are taken during an NBA game. As a reminder, VORP stands for Value Over Replacement Player, and is also a significant predictor of talent and production in the NBA. Looking at the second graph, we can see the error continues to decrease with increased number of trees. But as we can see, the error seems to level off at about 100 trees, and from then on the error is about the same for the increased number of trees. To help with computation cost and efficiency, we will run the model with 100 trees.

After running the model and finding the predicted values for NBA players since 2020, the graph of the Random Forest Model looks as follows:



The Random Forest model seems to perform incredibly, with data points hugging the line at every value along the axis. This model resulted in an R^2 value of 0.968, which means very accurate predictions. Like we talked about, Random Forest Models are commonly prone to overfitting, so the real test to see if this model is more accurate than our LOESS Model is the Test Error, which is shown below:

$$Training\ MSE\ = 0.000267980143817912$$
$$Test\ MSE\ = 0.00192524919189945$$

As a reminder, the Test MSE for the LOESS Model was 0.00202372217663988. This means that the Random Forest Model has a better Test MSE by a mere 0.0000985. This is very close to equivalent, and further testing will be used to determine the best model between the two.

Looking at Booker's salary again, the Random Forest Model is very accurate, giving us a value of:

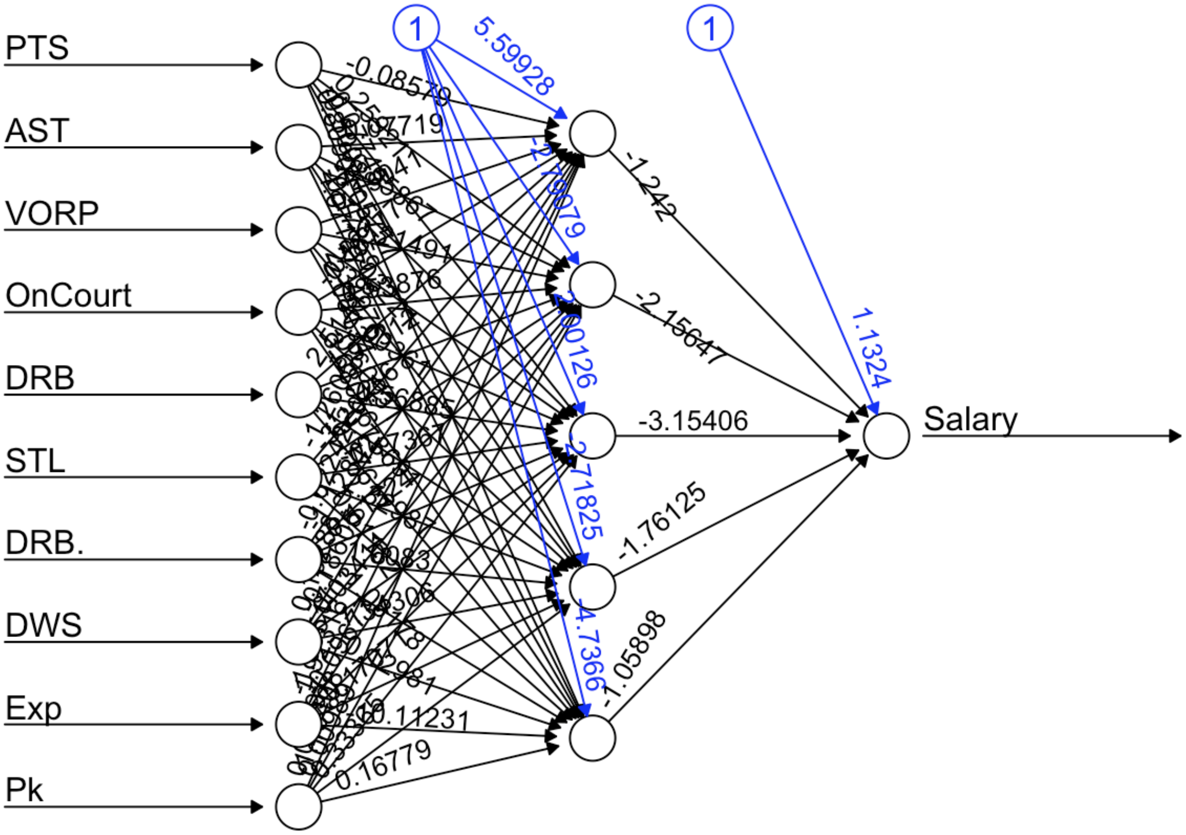
$$\hat{Salary} = 0.26821183(136.021) = 36.48\,million\,per\,year$$

This is very close to Booker’s real salary of $36.02 million in 2024, but as we talked about before, the Random Forest Model is prone to overfitting, so given that Booker was in the training data, this doesn’t tell us a lot about which model is better for prediction.

Section 3.10: Neural Network Model

The next model we will try is a neural network model. This model is beneficial to use when the data is complicated, extensive, and doesn’t have an exact pattern. A neural network model works by associating weights with different predictors, and determining how those predictors lead to the output. The model works by initially predicting the player salary given their stats, and then determines if that was a good “guess” by looking at the associated salary of the player. If the model thinks there is room for improvement, it adjusts the initial weights and does the same process again. It keeps doing this until it comes to a conclusion when the error is below a certain threshold.

Looking at the error associated with a certain amount of predictors, I determined that the model is better at predicting future values and has a lower error when a couple of predictors were removed. The model was left with 10 predictors, including PTS, VORP, AST, OnCourt, DRB, STL, DRB., DWS, Experience, and Pick Number. Once these values were input into the model, the resulting neural network visualization could be shown:



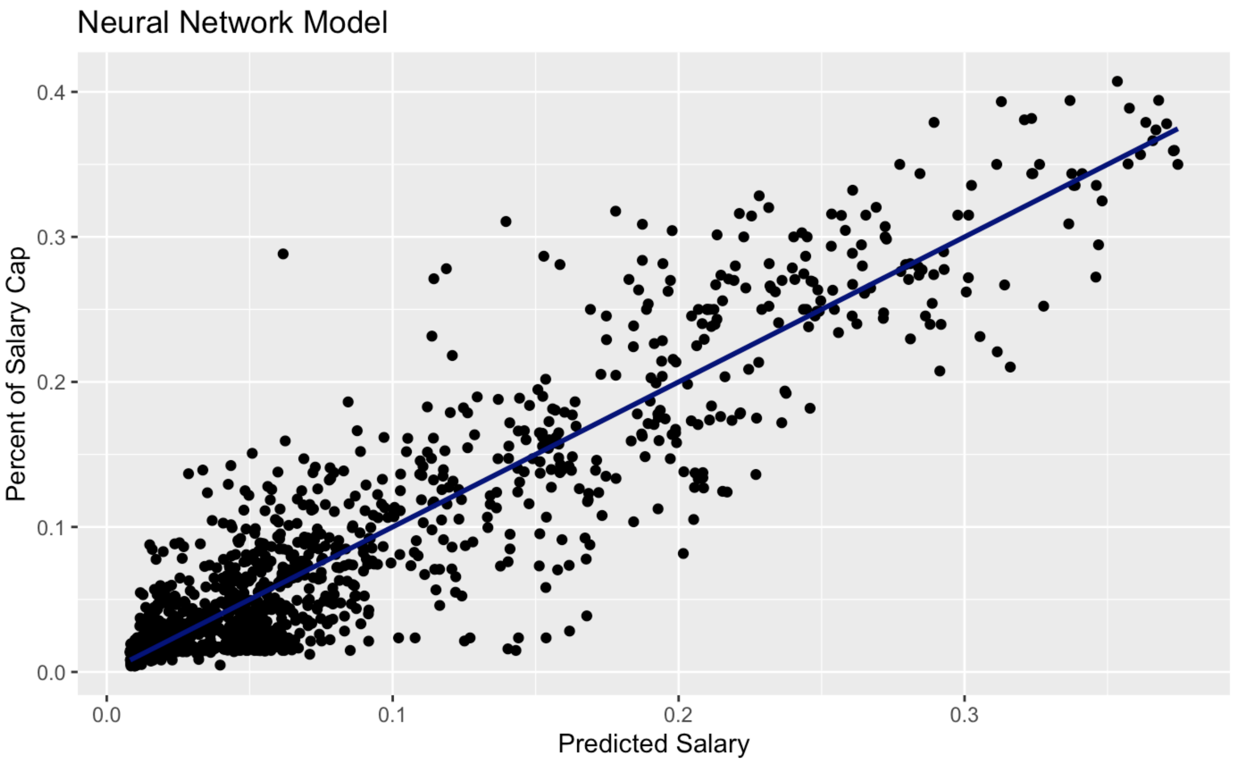
The values that are seen between the first layer of dots are the weights associated with each variable, and because there are so many, the values are a little difficult to see. This is the problem with neural network models, and a lot of tree models as well. The interpretability of these models is very small, and the data is being thrown into a “black box” with the predicted salary just spewing out. There is very little room for the user to understand what is happening, and the model acts as a black box where you don’t exactly know what is happening on the inside.

Regardless, the model was able to produce error values of:

$$\begin{aligned} Training\ MSE &= 0.00130611249506993 \\ Test\ MSE &= 0.00207053589943387 \end{aligned}$$

These error values are higher than that of the LOESS and Random Forest Model, and therefore neural networks will not be used to determine the final model in this case.

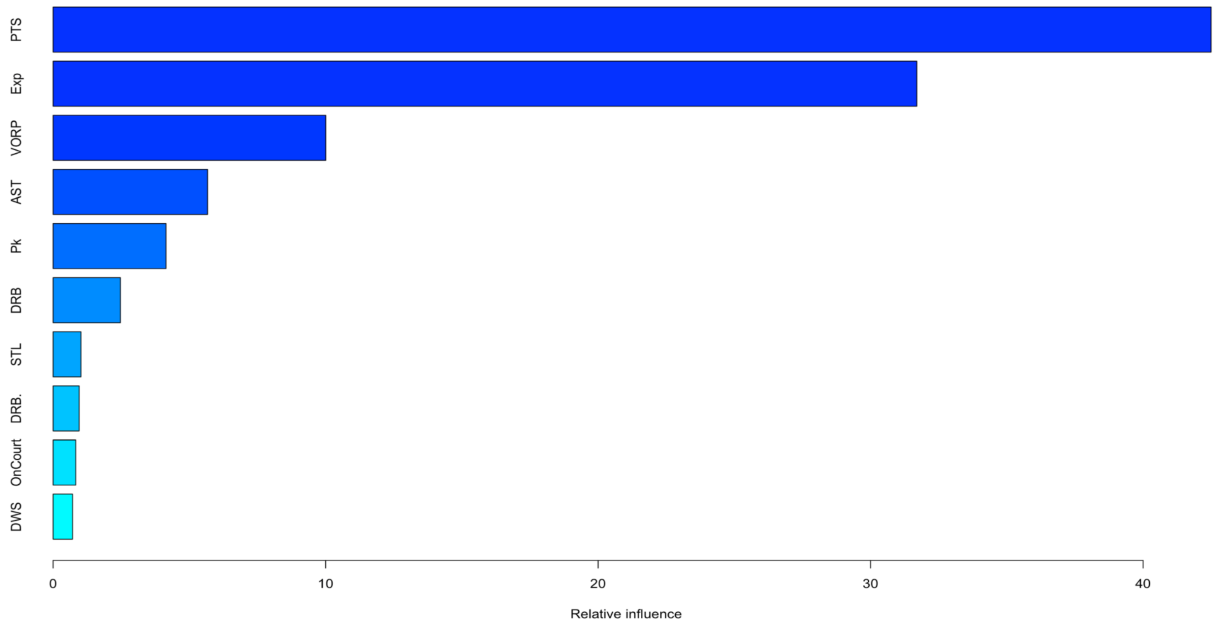
The graph of Predicted Salary versus Actual Salary is shown below:



Section 3.11: Gradient Boosting Model

As opposed to a Random Forest model, where many different trees are created, and the final conclusion of the model is the average of all those independent trees, a boosting model works by improving the tree model throughout each iteration. It “learns” from its previous mistakes, and keeps improving the model until it comes to a conclusion when the error is small enough. This makes a boosting model more accurate than that of a Random Forest, although equally less interpretable.

In this boosting model, similar to that of a Neural Network, the model did not see all predictors as significant variables in the model. It was eliminated until there were just 10 left, which were PTS, VORP, AST, OnCourt, DRB, STL, DRB., DWS, Experience, and Pick Number. After inputting all of these predictors into the model, we are shown a graph of the relative influence of each predictor on the salary. The graph is shown below:



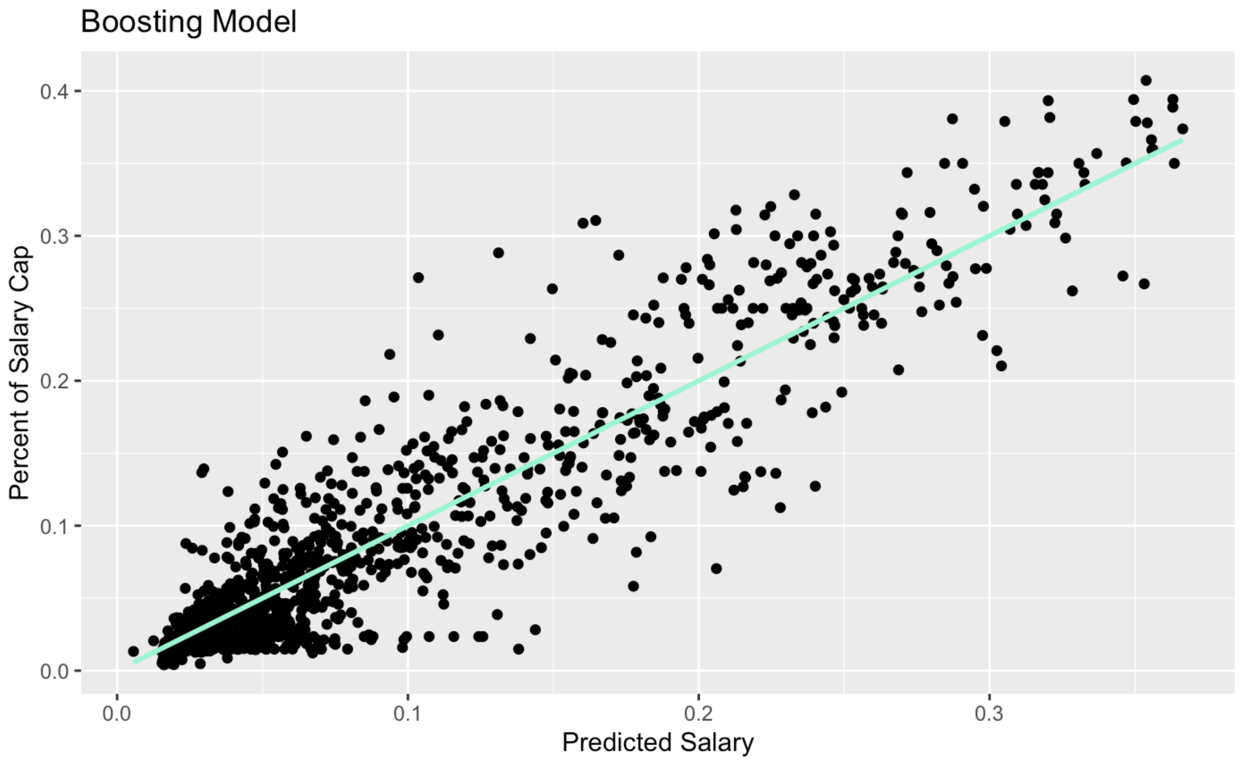
As seen on the graph, the predictors with the biggest influence on salary are PTS, Experience, VORP, and Assists. This is very similar to the influence of variables on the Random Forest model, with PTS, Experience, and VORP also being the top 3 most influential.

The errors for the Boosting Model are:

$$Training\ MSE\ = 0.00112768430581282$$
$$Test\ MSE\ =\ 0.00186166869232583$$

Compared to that of Random Forest that has the lowest Test MSE at the moment, the Boosting model has a higher Training MSE by a significant amount, but also a slightly lower Test MSE by about 0.00006. This is such a small difference, but through this analysis we can conclude that the Boosting Model is one of the top 3 models in predicting player salary along with Random Forest and LOESS. We will use further analysis to determine which one will be used in the final iteration.

The graph has a lower R^2 value than the Random Forest Model, with a value of 0.8634, but higher than the 0.8333 we found using the LOESS Model. The graph is shown below:



Chapter 4: Final Model Choice

Section 4.1: Comparison Table

Below is a comparison table looking at all the models, and the significant factors to determine which models are the most applicable for the project.

Model Name	**Training Error**	**Test Error**	**R squared**	**Computation Time (sec)**
Simple Linear Model	4.09E-03	4.22E-03	0.50	0.03
Multiple Linear Model	3.41E-03	4.03E-03	0.59	0.02
Log Multiple Linear Model	3.93E-03	4.62E-03	0.53	0.02
LASSO Model	3.41E-03	3.99E-03	0.59	0.02
Ridge Regression Model	3.43E-03	3.98E-03	0.57	0.09
Polynomial Regression Model	3.37E-03	3.90E-03	0.59	0.02
LOESS Model	1.38E-03	2.02E-03	0.83	0.34
Polynomial Regression	2.36E-03	3.06E-03	0.71	0.02

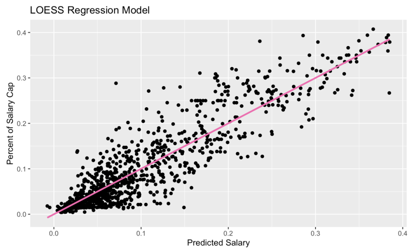
Model 2				
Random Forest Model	2.68E-04	1.93E-03	0.97	0.71
Neural Network Model	1.31E-03	2.07E-03	0.81	2.33
Gradient Boosting Model	1.13E-03	1.86E-03	0.86	5.11

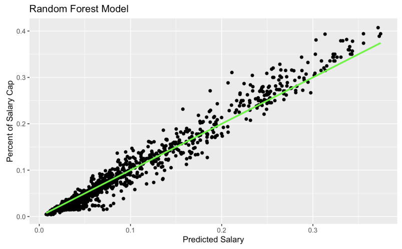
Section 4.2: The Top 3 Models

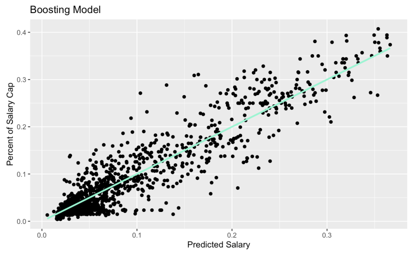
Because the Test Error is the smallest for the LOESS, Random Forest, and Boosting Models, we can conclude that those are the top 3 models created. But because the Test Error is all very similar, we will have to perform some more analysis to make a decision which model we will use in our final product.

As a reminder, the Random Forest Model was able to very accurately predict the training data, having an R^2 value 0.968, compared to the LOESS Model having a value of 0.8334 and Boosting with a value of 0.8634. This is deceiving, as a high R^2 value like the Random Forest Model can mean very high variance, making our model less accurate for future predictions. This is common in Random Forest, as on any given day, the model can choose a different tree path at the start, leading to a totally different end value, something that you don't have to worry about with the LOESS model. The Boosting Model is somewhere in the middle of the other two, while still using decision trees but learning from its mistakes and reducing error in the process unlike Random Forest.

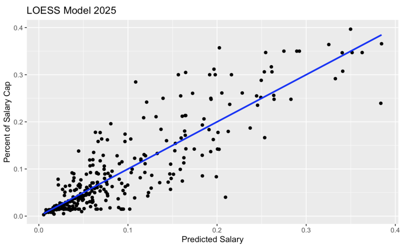
As a reminder the graphs of the three models are below:

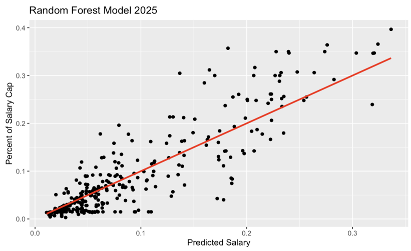


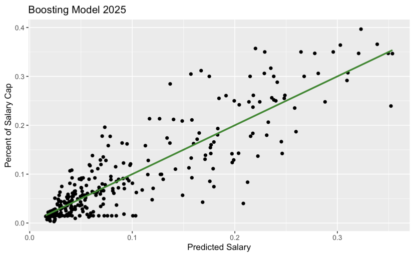




To compare the models accurately, we need to look at the graphs for the test data, or data from the 2024-2025 NBA Season. The graphs are given below, using only data from the current NBA season to act as a test for future prediction.

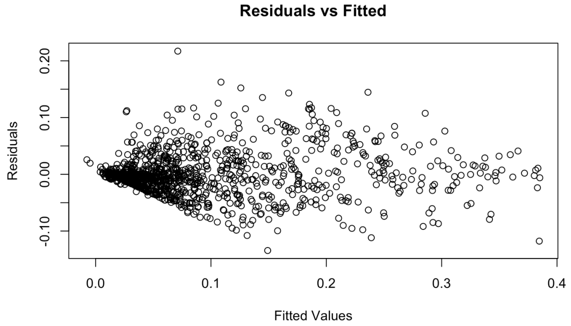


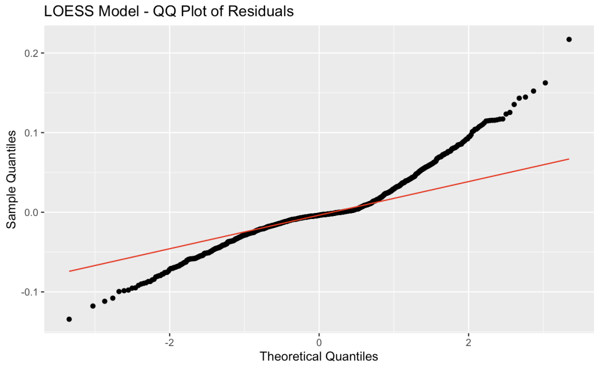




To determine which is the best model, we will look at the diagnostics of all the different models, including Residuals vs. Fitted plots and QQ plots.

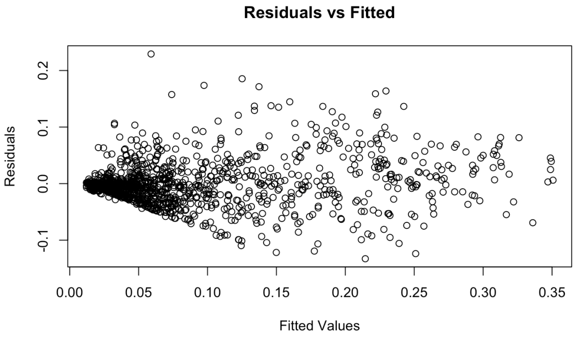
The Residuals vs. Fitted graph and QQ plot of the LOESS Model are shown below:

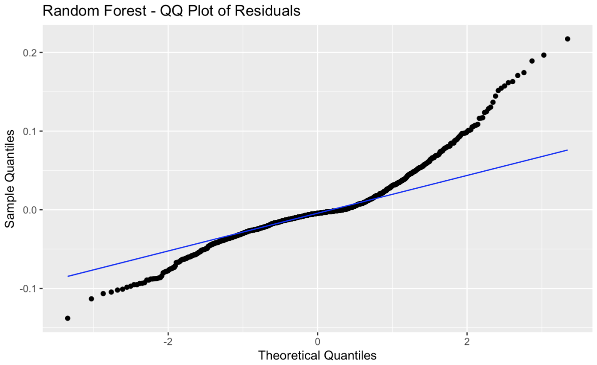




For the Residuals vs Fitted graph, you want the points to be somewhat randomly distributed around a horizontal line of y=0. Looking at the graph you can see that there is a much higher density of points on the left side of the graph, and the points follow a linear line downwards. This may mean the model is not suitable for prediction. For a QQ plot, you want the points to hug the dotted line throughout the x-axis. You also want the dotted line to be right at about 45 degrees. As you can see from the graph, the line is not exactly 45 degrees, but is relatively close. Near the end and beginning of the line, the points begin to stray a little bit. This might mean the model is not the best fit for our data, but looking at the other graphs will also help us come to this conclusion.

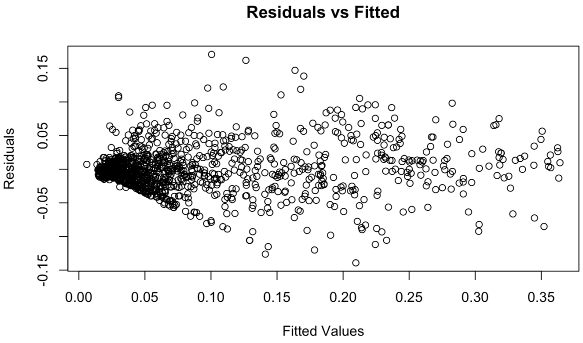
The graphs for the Random Forest model are below:

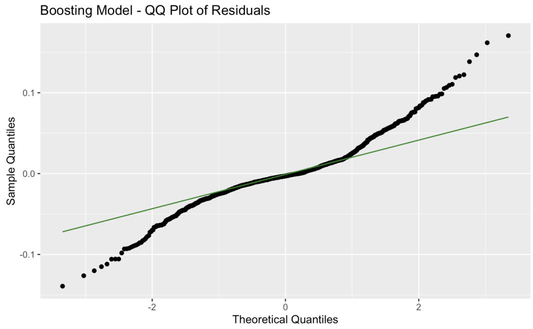




The Residuals vs Fitted graphs for the LOESS and Random Forest Model are very similar, with the points on the left side being much more dense than those on the right. But this could also be a "problem" with our data, as something as complicated as player salary is very hard to determine, and less players get paid large amounts of money. The QQ Plot also looks similar to the LOESS model, but you can argue that the points follow the line a little bit better than that of LOESS. But it is not perfect, as the points near the end and beginning of the line skew away a decent amount. Looking at all of this data I would conclude that the Random Forest is a better fit than the LOESS model so far.

Finally, we will look at the graphs for the Boosting Model:



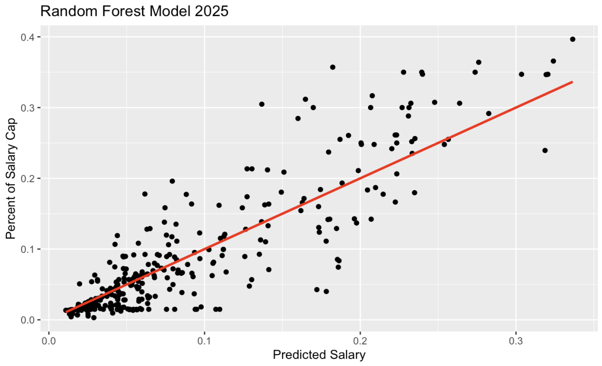


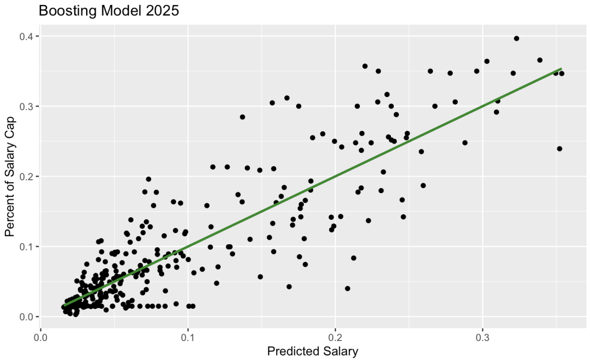
These plots look very similar to the Random Forest Model, which makes sense because a similar process of tree diagrams was used to determine the salaries for players. If anything, you could argue that the QQ plot for the Boosting Model looks a little bit better than the Random Forest, as it hugs the line for a little bit longer before straying off, but overall they are very similar.

To validate the testing error values for each of the models, I decided to perform an analysis using 500 different iterations of the dataset from the 2024-2025 NBA season. This will test more accurately which model has the lowest Test MSE values. After performing the analysis, the average Test MSE for the random forest model over the 500 datasets was 0.04325. Doing the same for the Boosting Model, the average Test MSE was 0.04268. This is a miniscule difference between the two, so further analysis is necessary.

As the last determining factor to choose between the Boosting and Random Forest Models, we will use the Test predictions from the 2024-2025 season to see if there is an equal amount of players being reported as Underpaid vs Overpaid. By saying this I am testing if the players who make more than their predicted values are about equal to players making less than their predicted values. A more equal relationship means a stronger model, and one that is more suitable for predictions during future seasons.

As a reminder the graphs for the 2025 NBA season and the predicted salaries is below:





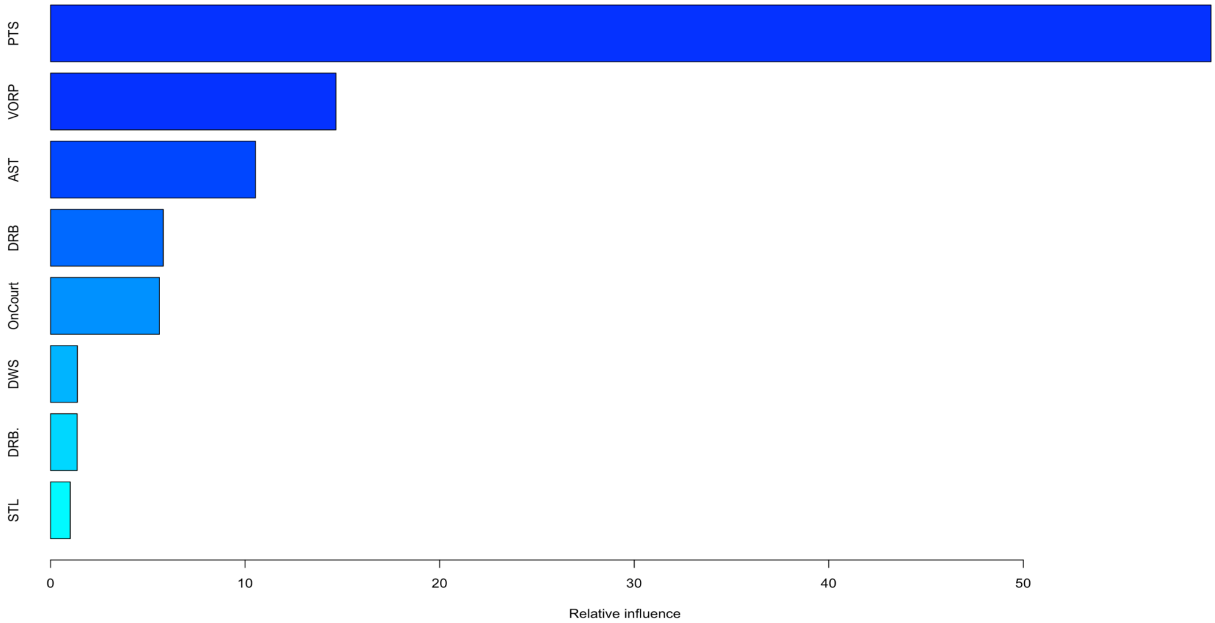
Out of 324 available players who have played at least 41 games (half of the full 82 games) this season, the results are as follows:

	Random Forest Model	**Boosting Model**
Overpaid Players	126	144
Underpaid Players	198	180
Ratio	0.636	0.80

From this table we can see that the Boosting Model is more suitable for prediction, as the number of Overpaid players is similar to Underpaid players, and has a ratio of about 0.8. Therefore the Boosting Model is our final model to be used to determine a player's salary from their performance, using variables such as PTS, VORP, AST, OnCourt, DRB, STL, DRB., DWS, Experience, and Pick Number.

Section 4.3: Boosting Model Continued

While the Boosting Model has been proven to be the most accurate model and is good for predicting the salaries of NBA players, it is not reliable to predict performance because of the external variables like Experience and Draft Pick Number. To account for this, I created a new model without those two variables, only including PTS, VORP, AST, OnCourt, DRB, STL, DRB., and DWS which are the most important statistics for predicting the performance of an NBA player. The chart below shows the relative influence of each variable on our new "Performance" variable.

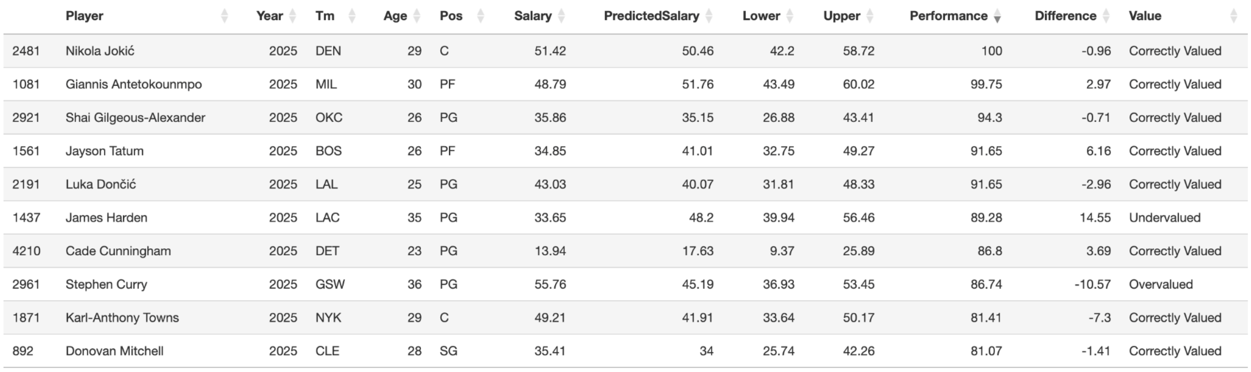


As you can see, points per game has a massive influence on the performance of a player, which makes sense given its importance in the game of basketball. Points per game is followed by assists, Value Over Replacement Player, and Assists.

I chose to use this model to predict Performance because it had a significantly less amount of variables compared to models we have looked at before, including LASSO, Ridge Regression, and Polynomial Regression. In these models we had over 20 variables, whereas for the Boosting Performance Model, we only have 8 important variables, and we are still able to predict Performance with good accuracy.

The R^2 value for this model is 0.62, although not near as accurate as the actual predictor model, this is a higher value than the Polynomial Regression Performance Model with a value of 0.5917.

Looking at the top Performers for the 2024-2025 NBA Season, we can see that the model has predicted the 3 consensus MVP candidates as the top performers, those being Nikola Jokic, Giannis Antetokounmpo, and Shai Gilgeous-Alexander. The top 10 performers from the 2024-2025 NBA Season are shown below:



	Player	Year	Tm	Age	Pos	Salary	PredictedSalary	Lower	Upper	Performance	Difference	Value
2481	Nikola Jokić	2025	DEN	29	C	51.42	50.46	42.2	58.72	100	-0.96	Correctly Valued
1081	Giannis Antetokounmpo	2025	MIL	30	PF	48.79	51.76	43.49	60.02	99.75	2.97	Correctly Valued
2921	Shai Gilgeous-Alexander	2025	OKC	26	PG	35.86	35.15	26.88	43.41	94.3	-0.71	Correctly Valued
1561	Jayson Tatum	2025	BOS	26	PF	34.85	41.01	32.75	49.27	91.65	6.16	Correctly Valued
2191	Luka Dončić	2025	LAL	25	PG	43.03	40.07	31.81	48.33	91.65	-2.96	Correctly Valued
1437	James Harden	2025	LAC	35	PG	33.65	48.2	39.94	56.46	89.28	14.55	Undervalued
4210	Cade Cunningham	2025	DET	23	PG	13.94	17.63	9.37	25.89	86.8	3.69	Correctly Valued
2961	Stephen Curry	2025	GSW	36	PG	55.76	45.19	36.93	53.45	86.74	-10.57	Overvalued
1871	Karl-Anthony Towns	2025	NYK	29	C	49.21	41.91	33.64	50.17	81.41	-7.3	Correctly Valued
892	Donovan Mitchell	2025	CLE	28	SG	35.41	34	25.74	42.26	81.07	-1.41	Correctly Valued

The Performance variable is scaled from 0-100, 100 being the best performer from the past season. Instead of this variable spanning all seasons from 2021-2025, this certain variable is calculated on a year by year basis. With the top performers from every season being named to have a Performance of 100, even if they aren't as good as the top player from the season before or after.

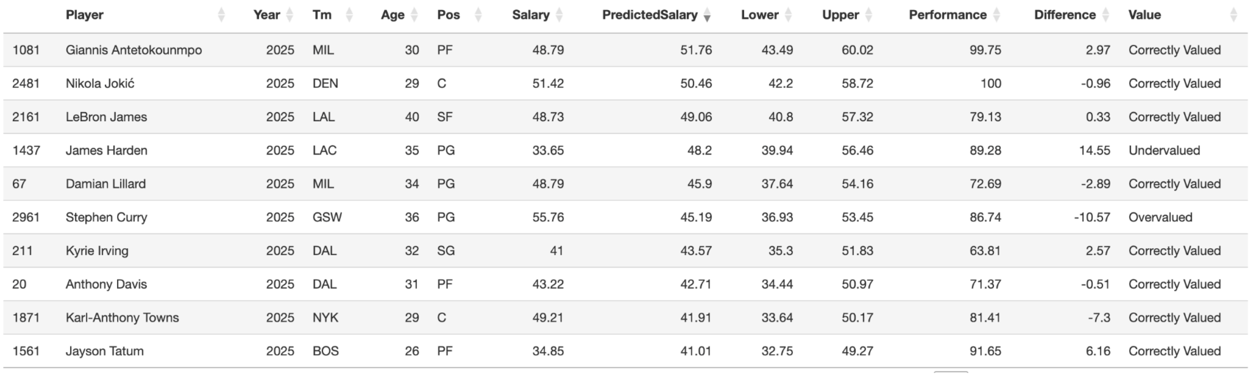
After perfecting the model, we are able to both predict the salary of a player, and quantify their performance in a certain season scaled from 0-100.

Chapter 5: Conclusions

Section 5.1: Model Conclusions 2025

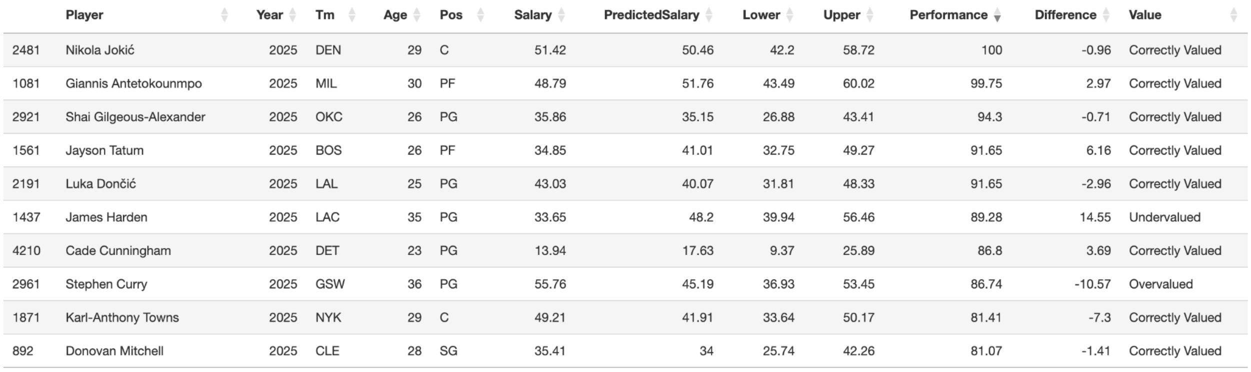
In this section we will take an in-depth look at what the model has predicted for the 2024-2025 NBA regular season, and who the model predicts to be the top players in the league, and the most under and overpaid players relative to their performance.

Starting with the most recent NBA season, where the playoffs are still in progress, the top 10 players for predicted salary are as follows.



	Player	Year	Tm	Age	Pos	Salary	PredictedSalary	Lower	Upper	Performance	Difference	Value
1081	Giannis Antetokounmpo	2025	MIL	30	PF	48.79	51.76	43.49	60.02	99.75	2.97	Correctly Valued
2481	Nikola Jokić	2025	DEN	29	C	51.42	50.46	42.2	58.72	100	-0.96	Correctly Valued
2161	LeBron James	2025	LAL	40	SF	48.73	49.06	40.8	57.32	79.13	0.33	Correctly Valued
1437	James Harden	2025	LAC	35	PG	33.65	48.2	39.94	56.46	89.28	14.55	Undervalued
67	Damian Lillard	2025	MIL	34	PG	48.79	45.9	37.64	54.16	72.69	-2.89	Correctly Valued
2961	Stephen Curry	2025	GSW	36	PG	55.76	45.19	36.93	53.45	86.74	-10.57	Overvalued
211	Kyrie Irving	2025	DAL	32	SG	41	43.57	35.3	51.83	63.81	2.57	Correctly Valued
20	Anthony Davis	2025	DAL	31	PF	43.22	42.71	34.44	50.97	71.37	-0.51	Correctly Valued
1871	Karl-Anthony Towns	2025	NYK	29	C	49.21	41.91	33.64	50.17	81.41	-7.3	Correctly Valued
1561	Jayson Tatum	2025	BOS	26	PF	34.85	41.01	32.75	49.27	91.65	6.16	Correctly Valued

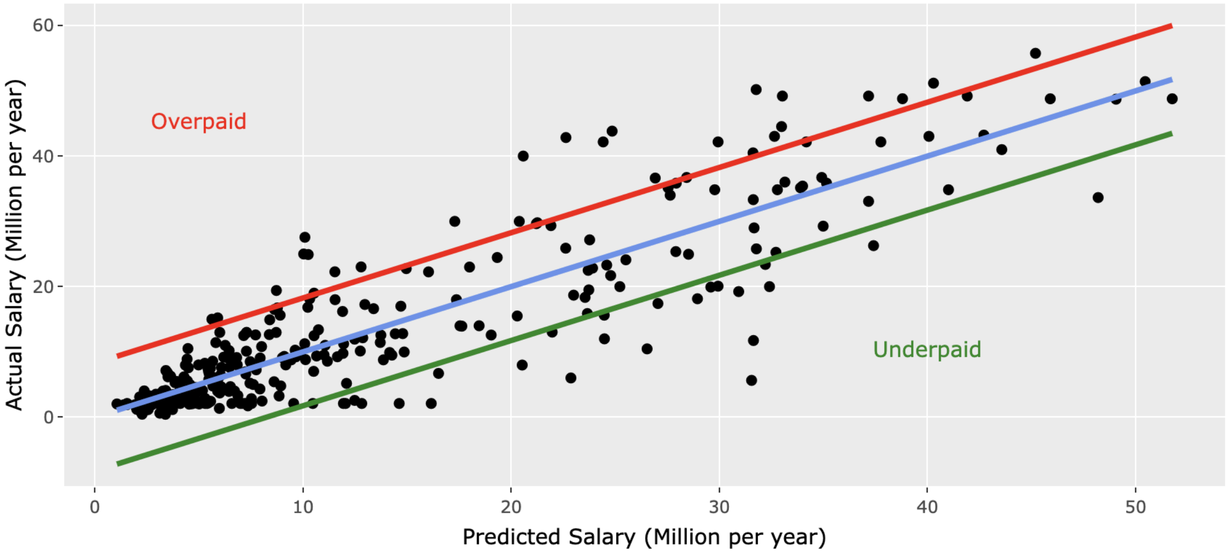
The top 10 highest predicted salaries in the NBA according to data from the 2024-2025 season are Giannis Antetokounmpo, Nikola Jokic, and LeBron James followed by more current All-Stars. Since our model included things like experience and draft pick, we can see a lot of the players in our top 10 are older players who have had multiple contracts in the league, and are therefore getting paid more than younger players. Jayson Tatum is the youngest player in the top 10, being 26 years old. Although the highest predicted salaries are commonly the older players, the highest performing players don't rely on experience at all, and only take statistics from the current season. The table showing the highest performing players is below.



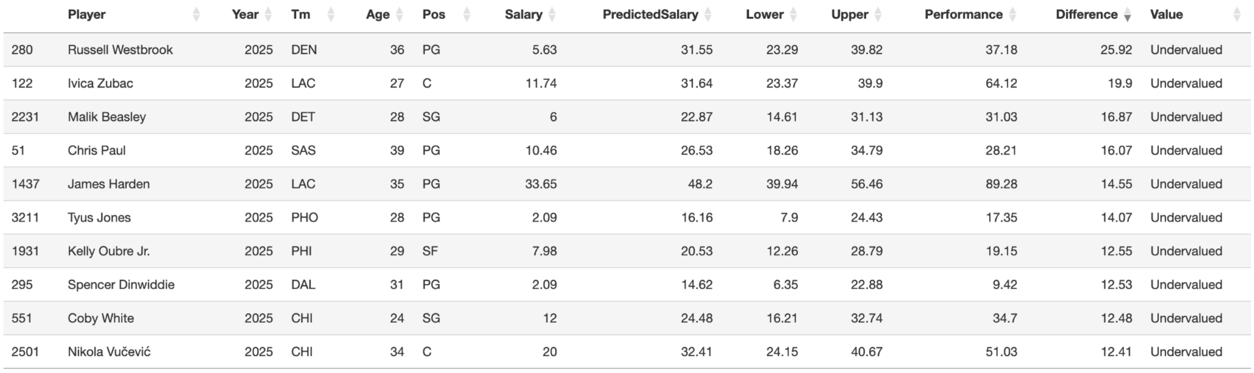
	Player	Year	Tm	Age	Pos	Salary	PredictedSalary	Lower	Upper	Performance	Difference	Value
2481	Nikola Jokić	2025	DEN	29	C	51.42	50.46	42.2	58.72	100	-0.96	Correctly Valued
1081	Giannis Antetokounmpo	2025	MIL	30	PF	48.79	51.76	43.49	60.02	99.75	2.97	Correctly Valued
2921	Shai Gilgeous-Alexander	2025	OKC	26	PG	35.86	35.15	26.88	43.41	94.3	-0.71	Correctly Valued
1561	Jayson Tatum	2025	BOS	26	PF	34.85	41.01	32.75	49.27	91.65	6.16	Correctly Valued
2191	Luka Dončić	2025	LAL	25	PG	43.03	40.07	31.81	48.33	91.65	-2.96	Correctly Valued
1437	James Harden	2025	LAC	35	PG	33.65	48.2	39.94	56.46	89.28	14.55	Undervalued
4210	Cade Cunningham	2025	DET	23	PG	13.94	17.63	9.37	25.89	86.8	3.69	Correctly Valued
2961	Stephen Curry	2025	GSW	36	PG	55.76	45.19	36.93	53.45	86.74	-10.57	Overvalued
1871	Karl-Anthony Towns	2025	NYK	29	C	49.21	41.91	33.64	50.17	81.41	-7.3	Correctly Valued
892	Donovan Mitchell	2025	CLE	28	SG	35.41	34	25.74	42.26	81.07	-1.41	Correctly Valued

As discussed before, this top 10 list includes much younger players, and NBA legends like LeBron James, Stephen Curry, Kyrie Irving and Anthony Davis don’t rank near as high. In this new list, the younger generation of players like Shai Gilgeous-Alexander, Jayson Tatum, Luka Doncic, and Cade Cunningham rank much higher than before. These youngsters are performing just as well or even better than the legends we talked about before, but because they are only on their second, and some even first contract, they aren’t getting paid near as much. As talked about before, this Performance list is able to predict the top 3 MVP candidates (Nikola Jokic, Giannis Antetokounmpo, and Shai Gilgeous-Alexander), showing the accuracy of the model.

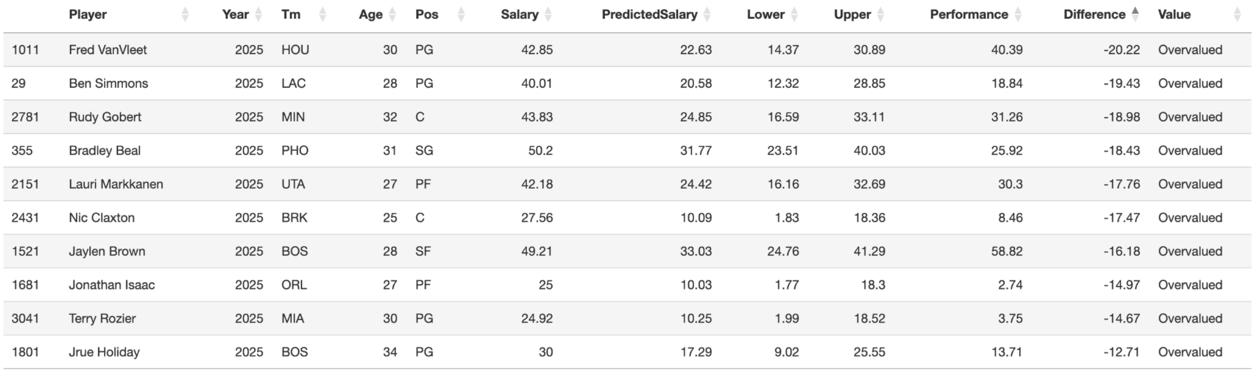
The chart below shows the Predicted Salary on the x-axis, and the actual Salary on the y-axis. The blue line is a y=x line straight down the middle of the data. The red and green lines are a 95% confidence interval for predictions. If the player datapoint lies below the green line, meaning they are outside the confidence interval for prediction, the player is quantified as “Underpaid”, whereas any player above the red line is given a value of “Overpaid”. Any player that lies between the two lines, close to the blue line in the middle has a value of “Correctly Valued”.



Now we will match the data points in the graph to actual player names. The first table is sorted by a new variable called Difference, that is simply the Predicted Salary subtracted by the Actual Salary. For the Difference variable, a higher number means they are underpaid by that many millions of dollars, whereas a negative number means they are being overpaid. The first table shows the top 10 most underpaid, and the second table shows the most overpaid.



	Player	Year	Tm	Age	Pos	Salary	PredictedSalary	Lower	Upper	Performance	Difference	Value
280	Russell Westbrook	2025	DEN	36	PG	5.63	31.55	23.29	39.82	37.18	25.92	Undervalued
122	Ivica Zubac	2025	LAC	27	C	11.74	31.64	23.37	39.9	64.12	19.9	Undervalued
2231	Malik Beasley	2025	DET	28	SG	6	22.87	14.61	31.13	31.03	16.87	Undervalued
51	Chris Paul	2025	SAS	39	PG	10.46	26.53	18.26	34.79	28.21	16.07	Undervalued
1437	James Harden	2025	LAC	35	PG	33.65	48.2	39.94	56.46	89.28	14.55	Undervalued
3211	Tyus Jones	2025	PHO	28	PG	2.09	16.16	7.9	24.43	17.35	14.07	Undervalued
1931	Kelly Oubre Jr.	2025	PHI	29	SF	7.98	20.53	12.26	28.79	19.15	12.55	Undervalued
295	Spencer Dinwiddie	2025	DAL	31	PG	2.09	14.62	6.35	22.88	9.42	12.53	Undervalued
551	Coby White	2025	CHI	24	SG	12	24.48	16.21	32.74	34.7	12.48	Undervalued
2501	Nikola Vučević	2025	CHI	34	C	20	32.41	24.15	40.67	51.03	12.41	Undervalued



	Player	Year	Tm	Age	Pos	Salary	PredictedSalary	Lower	Upper	Performance	Difference	Value
1011	Fred VanVleet	2025	HOU	30	PG	42.85	22.63	14.37	30.89	40.39	-20.22	Overvalued
29	Ben Simmons	2025	LAC	28	PG	40.01	20.58	12.32	28.85	18.84	-19.43	Overvalued
2781	Rudy Gobert	2025	MIN	32	C	43.83	24.85	16.59	33.11	31.26	-18.98	Overvalued
355	Bradley Beal	2025	PHO	31	SG	50.2	31.77	23.51	40.03	25.92	-18.43	Overvalued
2151	Lauri Markkanen	2025	UTA	27	PF	42.18	24.42	16.16	32.69	30.3	-17.76	Overvalued
2431	Nic Claxton	2025	BRK	25	C	27.56	10.09	1.83	18.36	8.46	-17.47	Overvalued
1521	Jaylen Brown	2025	BOS	28	SF	49.21	33.03	24.76	41.29	58.82	-16.18	Overvalued
1681	Jonathan Isaac	2025	ORL	27	PF	25	10.03	1.77	18.3	2.74	-14.97	Overvalued
3041	Terry Rozier	2025	MIA	30	PG	24.92	10.25	1.99	18.52	3.75	-14.67	Overvalued
1801	Jrue Holiday	2025	BOS	34	PG	30	17.29	9.02	25.55	13.71	-12.71	Overvalued

Russell Westbrook being the most underpaid player in the league this season does not come without controversy. Westbrook has had a few rough years recently, since signing with the Lakers in 2021, he has been viewed as a negative asset on the court. This was similar last year with the Clippers. But this year with the Nuggets, he has had a bit of a resurgence, although not back to his MVP form he has been able to effectively produce for the team, and has helped the Nuggets to gain a 4 seed in the West in this year's playoffs. Westbrook was predicted a 31 million salary by the model because of his experience and position in this league. Throughout the years, it is common that an older guard with lots of experience in the league gets paid a good amount, because they are very important to the success of a team. But because Westbrook had so many bad years recently, his stock drastically dropped leading to him getting a mere 5.63 million dollar contract. Westbrook is followed up by some great names on this list, including Ivica Zubac, Malik Beasley, Chris Paul, and James Harden.

Zubac started the year as a quality starting center, but throughout the year has proven himself to be one of the best centers in the league. His ability to finish alley-oops and anchor the team on the defensive side has not gone without notice. This performance has led him to be top 3 in Most Improved Player voting, which only validates that our model would predict him as underpaid because his performance this year has been much better than years before. Along with Zubac, Malik Beasley has been a great performer this year. Since signing with the Pistons this last offseason, Malik Beasley has helped turn that team around by hitting three pointers and being a great teammate. This has led him to be top 3 in 6th Man of the Year voting for award season.

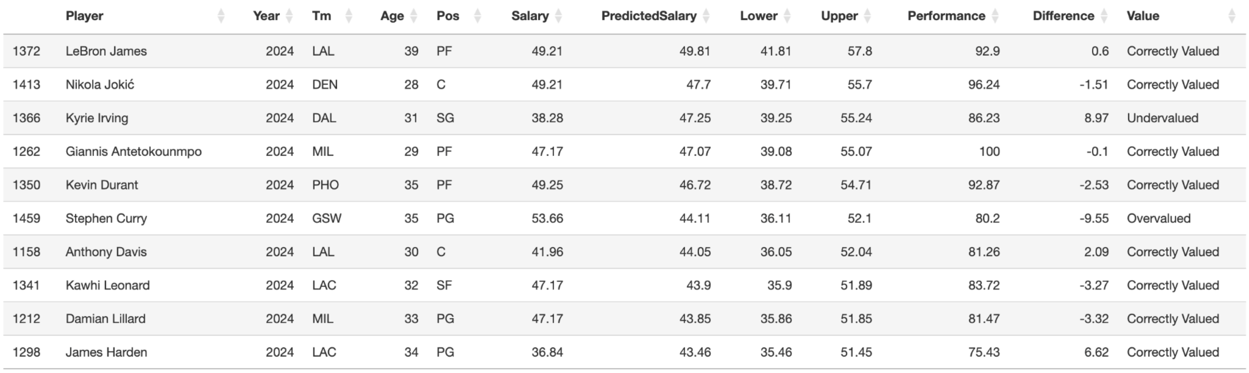
Turning to the Overpaid table, the top player on this list is Fred VanVleet, followed by Ben Simmons, Rudy Gobert, and Bradley Beal. These players are all making about $20 million more than they are projected, and have not had great seasons this year. Starting off with Fred VanVleet, he has been an average performer, but his large contract has led people to believe the Rockets signed him for too much money. After averaging 17 points per game and 8 assists for the Rockets last season, VanVleet is down to 14 points and 5.6 assists per game this season on 37% from the field and 34% from three. Although these aren't terrible numbers, it does not mean that he is worth the $42 million he is getting paid.

Ben Simmons has been one of the most overpaid players in the league for the last couple years, as he has been going through a lot of lingering injuries since signing his massive contract a few years ago. This season in February, Simmons and the Nets were able to come to a contract buyout agreement, where the Nets paid the remaining amount of his contract and he was able to sign with another team for the rest of the season. After this, Simmons signed with the Clippers for $1.08 million. Since signing with the Clippers Simmons has been a reliable player, so because this was a mid-season move, we will have to wait another year and see if Ben Simmons' performance means he is underpaid or overpaid next year in his next contract. Rudy Gobert and Bradley Beal have a similar story to VanVleet, although good NBA players, they simply aren't worth the contract that they are getting paid.

Section 5.2: Model Conclusions 2021-2024

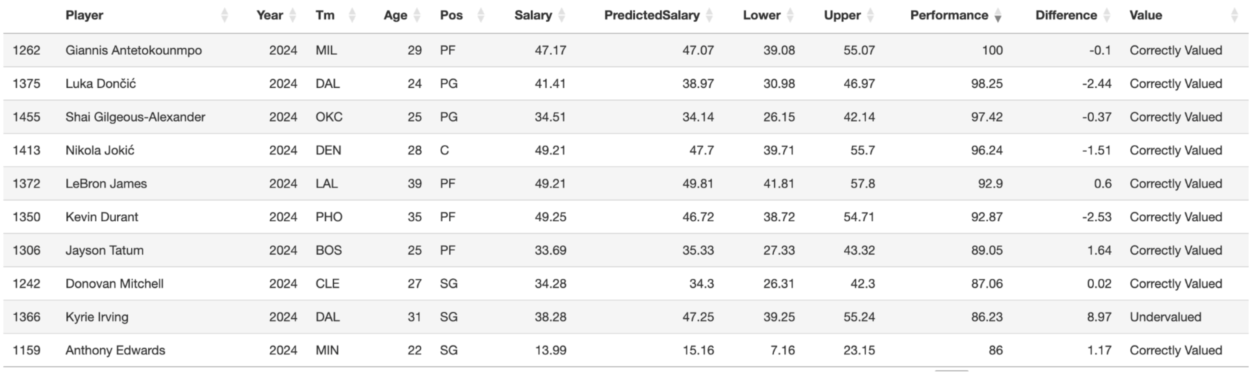
While the model was trained using data from 2021-2024, it was also able to come up with predictions in salary and performance for the past 4 years of the NBA seasons. I will quickly go over some of the top paid and top performers from the past few seasons.

Highest Predicted Salary 2023-2024 Season:



	Player	Year	Tm	Age	Pos	Salary	PredictedSalary	Lower	Upper	Performance	Difference	Value
1372	LeBron James	2024	LAL	39	PF	49.21	49.81	41.81	57.8	92.9	0.6	Correctly Valued
1413	Nikola Jokić	2024	DEN	28	C	49.21	47.7	39.71	55.7	96.24	-1.51	Correctly Valued
1366	Kyrie Irving	2024	DAL	31	SG	38.28	47.25	39.25	55.24	86.23	8.97	Undervalued
1262	Giannis Antetokounmpo	2024	MIL	29	PF	47.17	47.07	39.08	55.07	100	-0.1	Correctly Valued
1350	Kevin Durant	2024	PHO	35	PF	49.25	46.72	38.72	54.71	92.87	-2.53	Correctly Valued
1459	Stephen Curry	2024	GSW	35	PG	53.66	44.11	36.11	52.1	80.2	-9.55	Overvalued
1158	Anthony Davis	2024	LAL	30	C	41.96	44.05	36.05	52.04	81.26	2.09	Correctly Valued
1341	Kawhi Leonard	2024	LAC	32	SF	47.17	43.9	35.9	51.89	83.72	-3.27	Correctly Valued
1212	Damian Lillard	2024	MIL	33	PG	47.17	43.85	35.86	51.85	81.47	-3.32	Correctly Valued
1298	James Harden	2024	LAC	34	PG	36.84	43.46	35.46	51.45	75.43	6.62	Correctly Valued

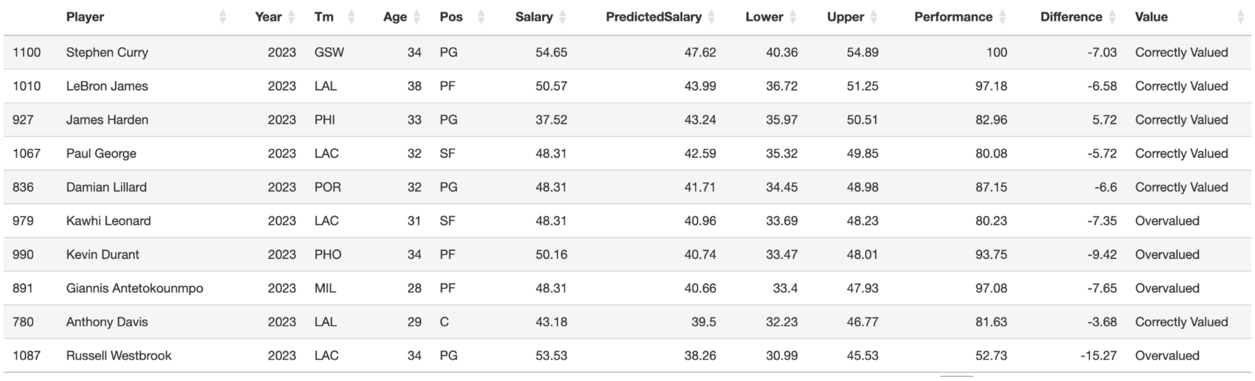
Highest Performance 2023-2024 Season:



	Player	Year	Tm	Age	Pos	Salary	PredictedSalary	Lower	Upper	Performance	Difference	Value
1262	Giannis Antetokounmpo	2024	MIL	29	PF	47.17	47.07	39.08	55.07	100	-0.1	Correctly Valued
1375	Luka Dončić	2024	DAL	24	PG	41.41	38.97	30.98	46.97	98.25	-2.44	Correctly Valued
1455	Shai Gilgeous-Alexander	2024	OKC	25	PG	34.51	34.14	26.15	42.14	97.42	-0.37	Correctly Valued
1413	Nikola Jokić	2024	DEN	28	C	49.21	47.7	39.71	55.7	96.24	-1.51	Correctly Valued
1372	LeBron James	2024	LAL	39	PF	49.21	49.81	41.81	57.8	92.9	0.6	Correctly Valued
1350	Kevin Durant	2024	PHO	35	PF	49.25	46.72	38.72	54.71	92.87	-2.53	Correctly Valued
1306	Jayson Tatum	2024	BOS	25	PF	33.69	35.33	27.33	43.32	89.05	1.64	Correctly Valued
1242	Donovan Mitchell	2024	CLE	27	SG	34.28	34.3	26.31	42.3	87.06	0.02	Correctly Valued
1366	Kyrie Irving	2024	DAL	31	SG	38.28	47.25	39.25	55.24	86.23	8.97	Undervalued
1159	Anthony Edwards	2024	MIN	22	SG	13.99	15.16	7.16	23.15	86	1.17	Correctly Valued

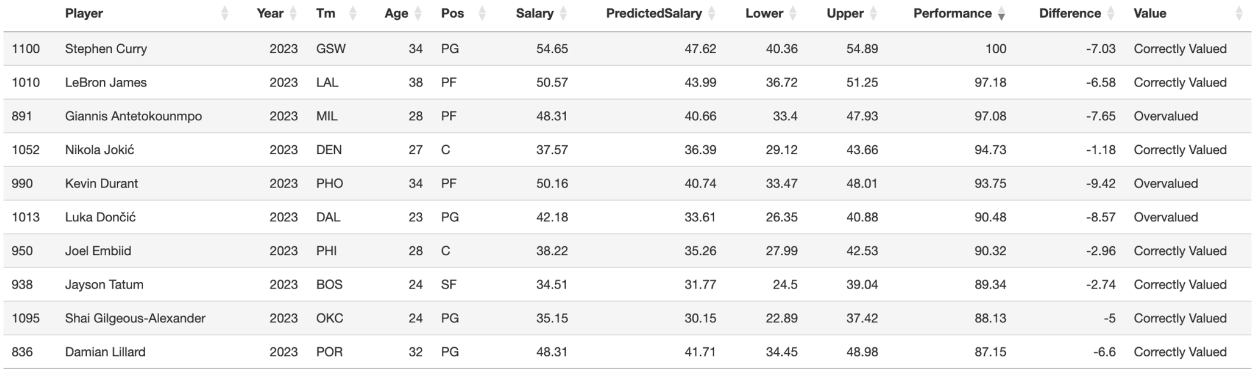
During this season, Nikola Jokic was named the MVP, followed by Shai Gilgeous-Alexander, Luka Doncic, and Giannis Antetokounmpo. Although those are the 4 players ranked in the top 4 according to the model, the model thinks that Giannis had the best year, and should have been named MVP although a close race. The difference between the two tables has the same story, with older more experienced players being paid more, but not exactly performing at the same level as the young and up-coming players.

Highest Predicted Salary 2022-2023 Season:



	Player	Year	Tm	Age	Pos	Salary	PredictedSalary	Lower	Upper	Performance	Difference	Value
1100	Stephen Curry	2023	GSW	34	PG	54.65	47.62	40.36	54.89	100	-7.03	Correctly Valued
1010	LeBron James	2023	LAL	38	PF	50.57	43.99	36.72	51.25	97.18	-6.58	Correctly Valued
927	James Harden	2023	PHI	33	PG	37.52	43.24	35.97	50.51	82.96	5.72	Correctly Valued
1067	Paul George	2023	LAC	32	SF	48.31	42.59	35.32	49.85	80.08	-5.72	Correctly Valued
836	Damian Lillard	2023	POR	32	PG	48.31	41.71	34.45	48.98	87.15	-6.6	Correctly Valued
979	Kawhi Leonard	2023	LAC	31	SF	48.31	40.96	33.69	48.23	80.23	-7.35	Overvalued
990	Kevin Durant	2023	PHO	34	PF	50.16	40.74	33.47	48.01	93.75	-9.42	Overvalued
891	Giannis Antetokounmpo	2023	MIL	28	PF	48.31	40.66	33.4	47.93	97.08	-7.65	Overvalued
780	Anthony Davis	2023	LAL	29	C	43.18	39.5	32.23	46.77	81.63	-3.68	Correctly Valued
1087	Russell Westbrook	2023	LAC	34	PG	53.53	38.26	30.99	45.53	52.73	-15.27	Overvalued

Highest Performance 2022-2023 Season:



	Player	Year	Tm	Age	Pos	Salary	PredictedSalary	Lower	Upper	Performance	Difference	Value
1100	Stephen Curry	2023	GSW	34	PG	54.65	47.62	40.36	54.89	100	-7.03	Correctly Valued
1010	LeBron James	2023	LAL	38	PF	50.57	43.99	36.72	51.25	97.18	-6.58	Correctly Valued
891	Giannis Antetokounmpo	2023	MIL	28	PF	48.31	40.66	33.4	47.93	97.08	-7.65	Overvalued
1052	Nikola Jokić	2023	DEN	27	C	37.57	36.39	29.12	43.66	94.73	-1.18	Correctly Valued
990	Kevin Durant	2023	PHO	34	PF	50.16	40.74	33.47	48.01	93.75	-9.42	Overvalued
1013	Luka Dončić	2023	DAL	23	PG	42.18	33.61	26.35	40.88	90.48	-8.57	Overvalued
950	Joel Embiid	2023	PHI	28	C	38.22	35.26	27.99	42.53	90.32	-2.96	Correctly Valued
938	Jayson Tatum	2023	BOS	24	SF	34.51	31.77	24.5	39.04	89.34	-2.74	Correctly Valued
1095	Shai Gilgeous-Alexander	2023	OKC	24	PG	35.15	30.15	22.89	37.42	88.13	-5	Correctly Valued
836	Damian Lillard	2023	POR	32	PG	48.31	41.71	34.45	48.98	87.15	-6.6	Correctly Valued

During this season, Joel Embiid was surprisingly voted the NBA MVP, beating out players like Nikola Jokic and Giannis Antetokounmpo who rounded out the top 3. Stephen Curry finished 9th in MVP voting, so it is interesting to see him rank so high according to the model, although he had a great season averaging almost 30 points per game shooting 49% from the field, 42% from three and 91% from the free throw line.

To save time, the Highest Predicted Salaries and top Performers from further seasons can be found by accessing my website that I created for this project at:

https://nbasalarypredictor.com/

Section 5.3: Conclusions and Applications

This salary prediction model offers NBA general managers a data-driven approach to evaluating player value, going beyond raw production to assess how on-court performance, experience, and draft pedigree translate into market compensation. By accurately estimating what a player should be earning based on current statistics, GMs can make smarter decisions around trades, contract extensions, and free agency offers. It allows front offices to identify undervalued assets ideal for building a competitive roster under the new NBA salary cap constraints. Conversely, it helps flag overpaid veterans or declining contributors, supporting tough calls on waivers, buyouts, or salary-dump trades.

Though its primary utility is in the front office, the model also supports other applications. It can guide agents in negotiations, help media analysts contextualize contracts, and even give sports bettors a statistical edge when assessing player prop lines. Ultimately, by quantifying salary through performance, the model becomes a powerful lens for navigating the modern NBA's complex ecosystem of talent, economics, and strategy.

The results are compiled in a website that I created specifically for this project, where users can go and explore the data themselves.

Link: https://nbasalarypredictor.com/