P8106 Midterm Report

Mingkuan Xu (mx2262)

Introduction

For any team in the National Basketball Association (NBA), a key strategy to win more games is to properly allocate their salary cap - an agreement that places a limit on the amount of money that a team can spend on players' salaries. How to evaluate the performance of each NBA player and give a suitable level of salary is a therefore complicated problem. In this project, we intend to predict the salary of NBA players in the 2021-2022 season based on their game statistics. We collected game statistics that are commonly used to evaluate players from the NBA official website, built both linear and non-linear models, including linear regression, ridge regression, lasso regression, GAM and MARS on selected feature variables, and compared these models to determine a final predictive model.

Data Preprocessing

We used two important data sets in this project:

- **NBA Player Salary (2021-2022)**: that contains the amount of salary each player received. (**Link**)
- **NBA Player Stats (2021-2022)** that contains the game statistics of each player. (**Link**)

After cleaning up and joining the two data sets based on player's name and team, we obtained a dataframe with 442 rows and the following columns:

- `Pos` – A categorical variable of the player's position (C, PF, SF, SG, PG)
- `Age` – Player's age on February 1 of the season
- `Team` – A categorical variable of the player's playig team
- `G` – Number of games played
- `GS` – Number of games played as a starter
- `MP` – Minutes played per game
- `FG` – Field goals per game
- `FGA` – Field goal attempts per game
- `FG%` – Field goal percentage
- `3P` – 3-point field goals per game
- `3PA` – 3-point field goal attempts per game
- `3P%` – 3-point field goal percentage
- `2P` – 2-point field goals per game

- `2PA` – 2-point field goal attempts per game
- `2P%` – 2-point field goal percentage
- `FT` – Free throws per game
- `FTA` – Free throw attempts per game
- `FT%` – Free throw percentage
- `ORB` – Offensive rebounds per game
- `DRB` – Defensive rebounds per game
- `TRB` – Total rebounds per game
- `AST` – Assists per game
- `STL` – Steals per game
- `BLK` – Blocks per game
- `TOV` – Turnovers per game
- `PF` – Personal fouls per game
- `PTS` – Points per game

Given that some players do not have any field goal, 2-point, 3-point, or free throw attempts, resulting in NAs in `FG%`, `2P%`, `3P%`, and `FT%`, we simply discarded these columns. Notice that dropping these columns will not result in any loss of information, as their values can be calculated using other columns (since `percentage` = `goals/attempts`).

Exploratory Analysis/Visualization

After splitting the dataset into training (80%) and test (20%) set, we started examining patterns of data on the training set. We did exploratory analysis, including the box plots showing the distribution of variables, the correlation heat map, and feature maps.

From the above correlation heatmap and the feature maps, we could identify positive correlations between some predictive variables and the response variable. However, from the correlation heat map, it is obvious that multicolinearity could be a problem, which we may consider using models such as ridge regression or lasso regression to fix; the feature maps also demonstrated that some correlations are non-linear, which we may consider using GAM or MARS to address.

Models

Models Trained

Based on the exploratory analysis, we built 7 different models in total: a simple Linear Regression model, a Ridge Regression model, a Lasso Regression model, an Elastic-Net model, a Principal Component Regression (PCR) model, a Generalized Addictive (GAM) model, and a Multivariate Adaptive Regression Splines (MARS) model (see codes.Rmd for details).

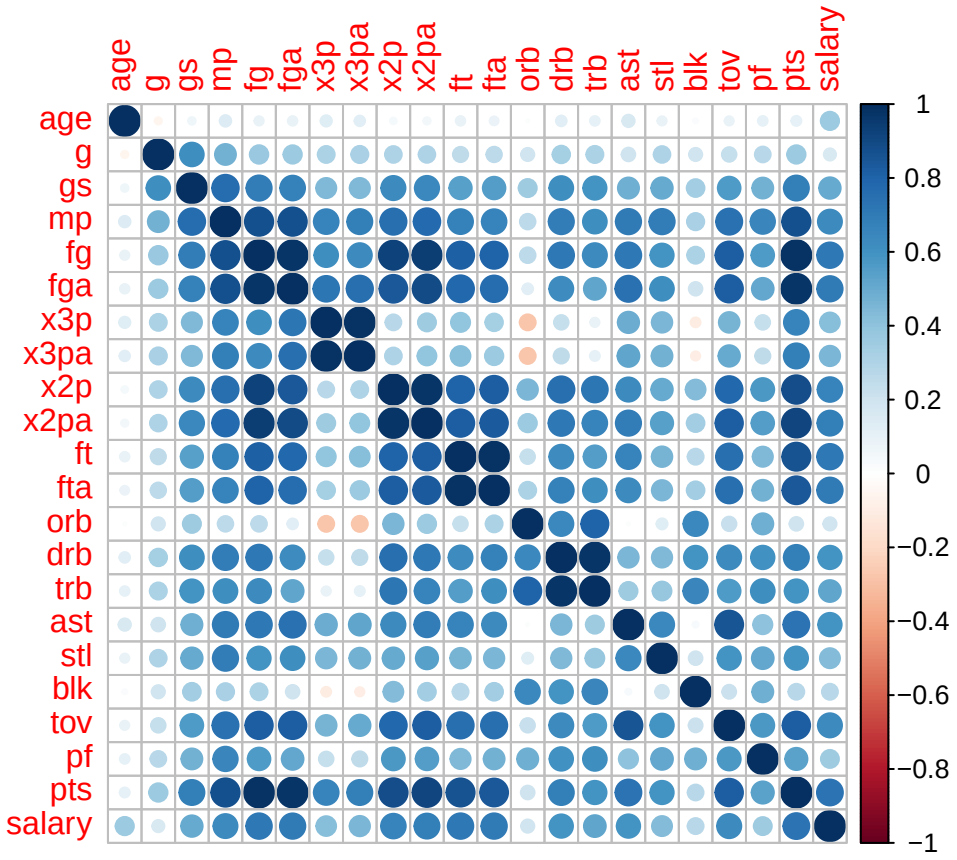

Figure 1: Correlation Heatmap of Player Salary and Game Statistics

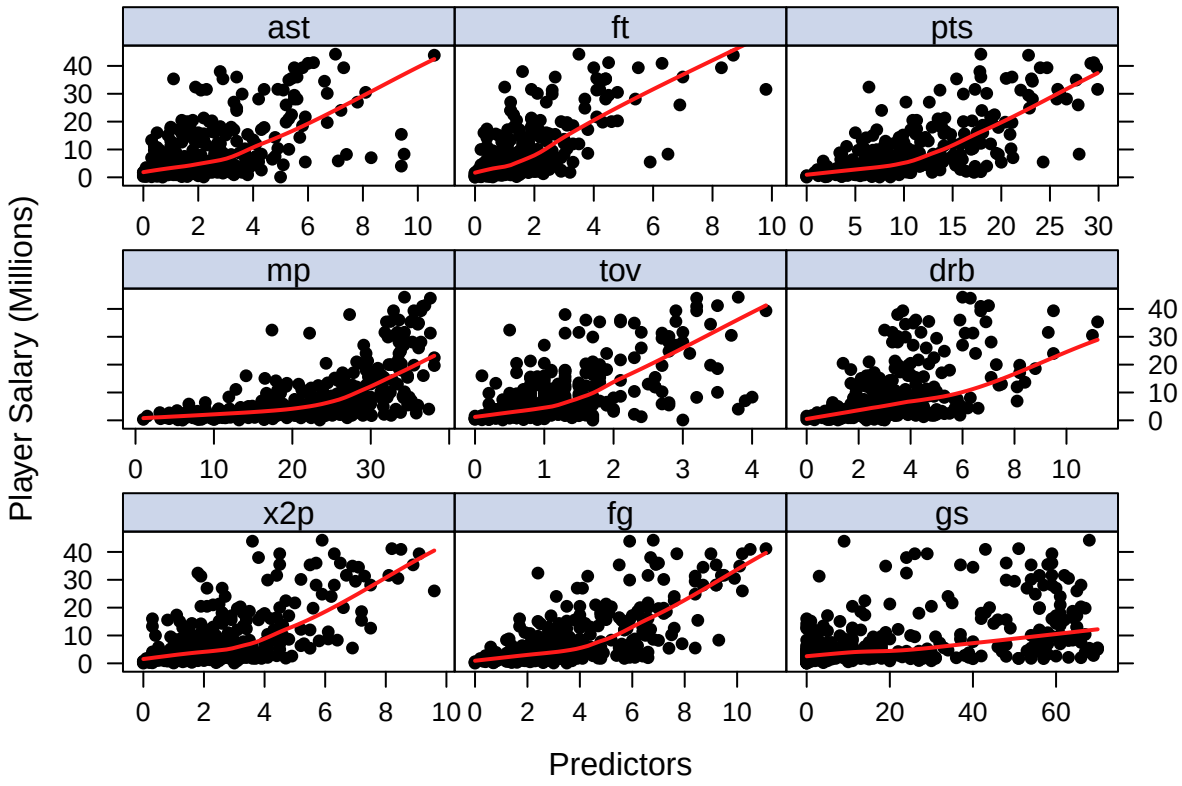

Figure 2: Feature Maps of Player Salary and Game Statistics

Parameter Tunings

In fitting the ridge/lasso/elastic-net models, I tried various ranges of lambda. The optimal lambda value for the ridge regression is 3.57, whereas the optimal lambda value for the lasso regression is 0.200. The elastic net model reached its best tune at $\alpha = 1$, i.e. a lasso model.

In attempting to fit the MARS model, I noticed that the RMSE increased drastically when `degree` is over 3 and `n_prune` is over 8. Therefore, I finally chose the range of degrees as 1:3, and range of `n_prune` as 2:8. I experienced difficulties fitting the MARS, however, that some times when I ran the code chunk, despite using the same script and the same random seed, the resulting model will be a slightly different with different RMSEs. This led to some potential inconsistencies in explaining the model coefficient at the end of the report. In the future, this problem can be fixed by saving the model object locally and reading from the file, instead of fitting the model again.

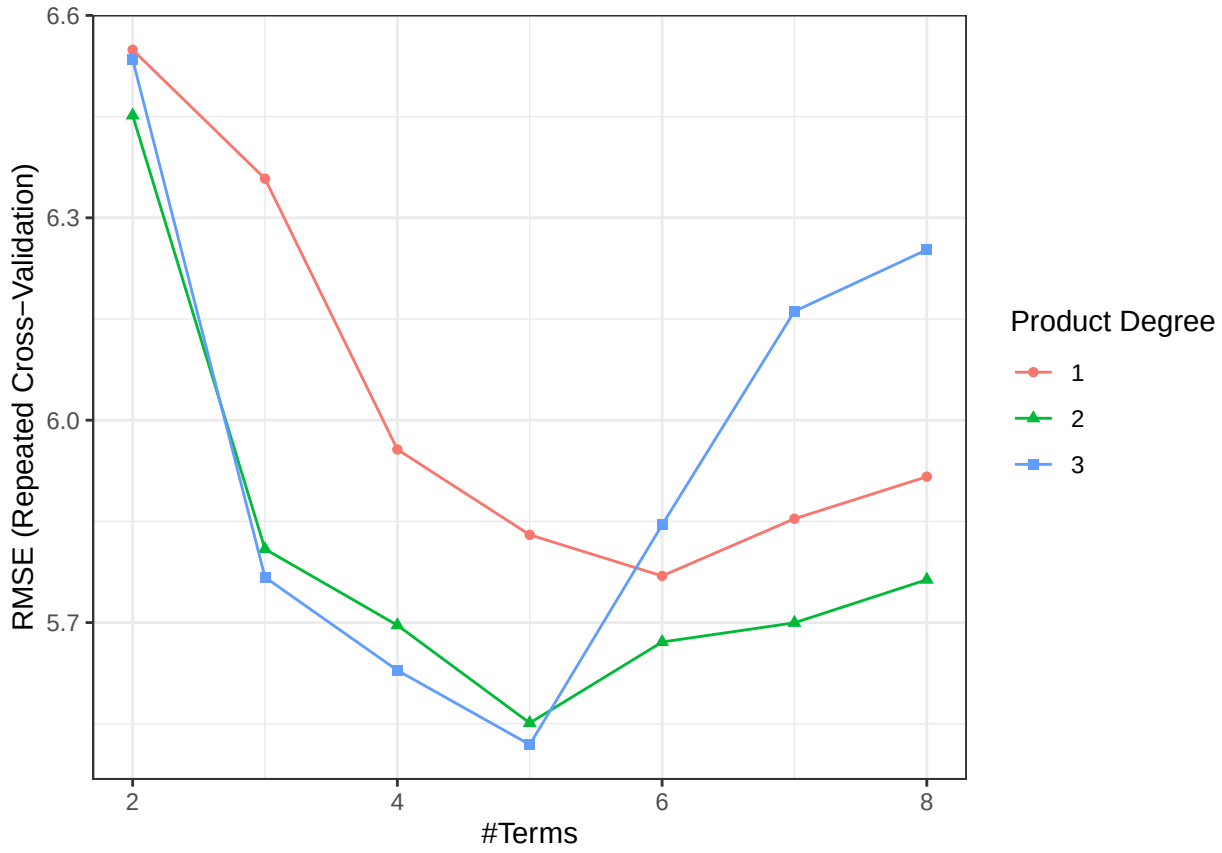

Figure 3: Parameter Tuning of MARS model

Results & Discussion

To compare the performance of these models, we listed their MAE, RMSE, R-square, and also RMSE on the test sets:

Table 1: Table 1: RMSE of Different Linear Models

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
LeastSquare	4.11	5.55	6.23	6.15	6.89	7.77	0
Ridge	4.11	5.29	5.90	5.91	6.54	7.43	0
Lasso	4.00	5.10	5.87	5.81	6.30	8.08	0
ElasticNet	4.26	5.29	5.58	5.76	6.32	7.66	0
PCR	3.92	5.00	5.82	5.70	6.37	7.43	0

Table 2: Table 2: RMSE of Different Models on Test Set

	Linear	Lasso	Rigde	ElasticNet	PCR	GAM	MARS
RMSE	6.85	6.62	6.61	6.61	6.27	6.96	5.91

In table 1, we compared the 5 models: standard linear regression, ridge regression, and lasso regression, elastic net, and principle component regression. Compared to the basic linear model, while all other models showed some improvements, it can be concluded that the PCR model provided a better fitting of the data in terms of RMSE. This can be explained by the fact that the PCA technique used by PCR regression is well-suited for dataset showing high levels of multicollinearity in this case.

In table 2. we compared the performance of the 7 models in predicting new data in the test set. The MARS model achieved a much better performance than all other 6 models, because it better captured the non-linear association between the predictive variables and the response variable.

Finally, we took a closer look of the coefficients of the MARS model:

Table 3: Table 3: Coeeficients of the MARS Model

	x
(Intercept)	16.121
h(pts-9.5)	1.686
h(29-age) * h(pts-9.5)	-0.138
h(33-age)	-1.204
h(age-23) * h(6.3-drb)	-0.233

From the coefficients, we observed that age and points earned are two important factors, which is what we expected. While most players entered the league at similar ages, they will have their player contract renewed after 4 years - that is why some talented rookies usually get a skyrocketed salary at the age of 24~25; on the other hand, older players with less score also have less potential compared to their younger counterparts, which explains why after a certain age, there is a negative correlation between age and salary.

Conclusion

Above all, after utilizing different methods introduced in this course to fit predictive models, we came up with a optimal MARS model, which best captures the underlying patterns of the player's data and gives reasonable predictions on a player's salary based on game statistics.