P8106 - Final Project - NBA Players Salary Prediction

Mingkuan Xu, Mengfan Luo, Yiqun Jin

Introduction

For teams 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 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, elastic net regression, principle component regression (PCR), generalized addictive model (GAM), multivariate adaptive regression spline (MARS) model, random forest and neural network on selected feature variables, and compared these models to determine a final predictive model.

Data Preprocessing

We will conduct data analysis and model construction based on two datasets on NBA players' contracted salary [1] and performance statistics per game [2] in 2021-2022. The following steps are included in our data preparation:

- Two original datasets were inner joined by players and teams
- Kept only one record with most number of games played for each of players, given a player may transfer to other teams during the session and have multiple records.
- Removed 5 variables with missing values caused by division of other existing variables.
- Divided count variables (`field_goal`, `free_throw`, etc.) by variable `minute` to convert them to efficiency

The final cleaned dataset has 442 records and 24 variables, including 2 categorical variables, 21 numerical variables and 1 numeric response variable `salary`.

Variable Name	Meaning	Varianble Type
position	Position of the player	categorical (5 classes)
age	Player's age on February 1 of the season	numeric
team	Team that the player belong to	categorical (30 classes)
game	Number of games played	numeric
game_starting	Number of games played as a starter	numeric
minute	Minutes played per game	numeric
field_goal	Field goals per minute	numeric
fg_attempt	Field goal attempts per minute	numeric
x3p	3-point field goals per minute	numeric
x3p_attempt	3-point field goal attempts per minute	numeric
x2p	2-point field goals per minute	numeric
x2p_attempt	2-point field goal attempts per minute	numeric
free_throw	Free throws per minute	numeric

Variable Name	Meaning	Varianble Type
ft_attempt	Free throw attempts per minute	numeric
offensive_rb	Offensive rebounds per minute	numeric
defenssive_rb	Defensive rebounds per minute	numeric
total_rb	Total rebounds per minute	numeric
assistance	Assists per minute	numeric
steal	Steals per minute	numeric
block	Blocks per minute	numeric
turnover	Turnovers per minute	numeric
personal_foul	Personal fouls per minute	numeric
point	Points per minute	numeric
salary	Salary of the player in million (Response)	numeric

Exploratory Analysis

Univariate Analysis

The following plots show distribution of each univariable. For categorical variables `team` and `position`, they are distributed quite evenly. There are 30 unique values in `team`, which may result in too many dummy variables in the model. Therefore, we may consider exclude `team` or cluster it into fewer classes in selected models.

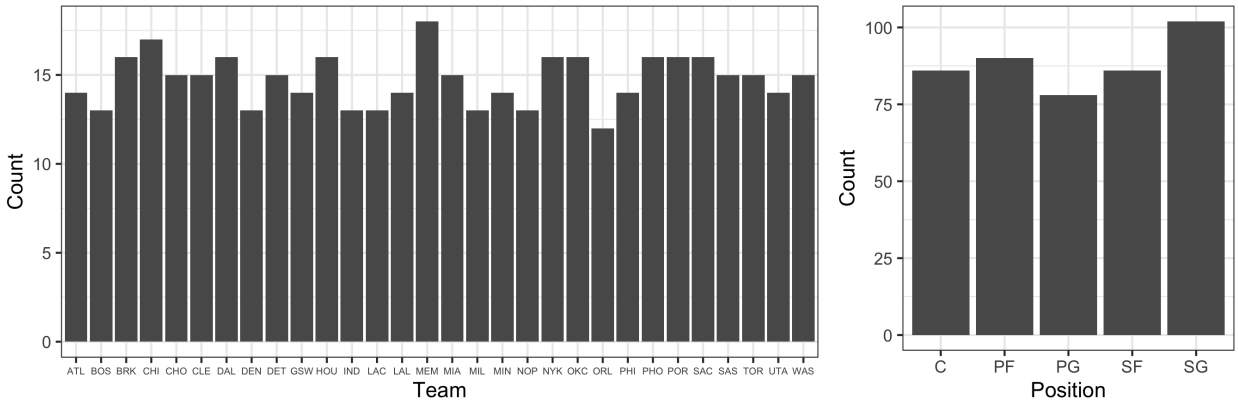


Figure 1: Histograms of categorical predictive variables

For numeric variables, some of them (`gs`, `ft`, `orb`,`blk`), including response `salary` are skewed, with some players have extremely high salary. Visualization for all variables are enclosed in Appendix A.

Correlation Analysis

From the correlation heat map, it is obvious that multicolinearity could be a problem, which we may consider using penalized models (ridge, lasso) or ensembled models (random forest, boosting, neural network) to fix. The feature maps demonstrated that some correlations are non-linear, which we may consider using GAM or MARS to address.

From categorical variable `position` and `team`, extremely high values and large variance in salary show in all positions and some teams.

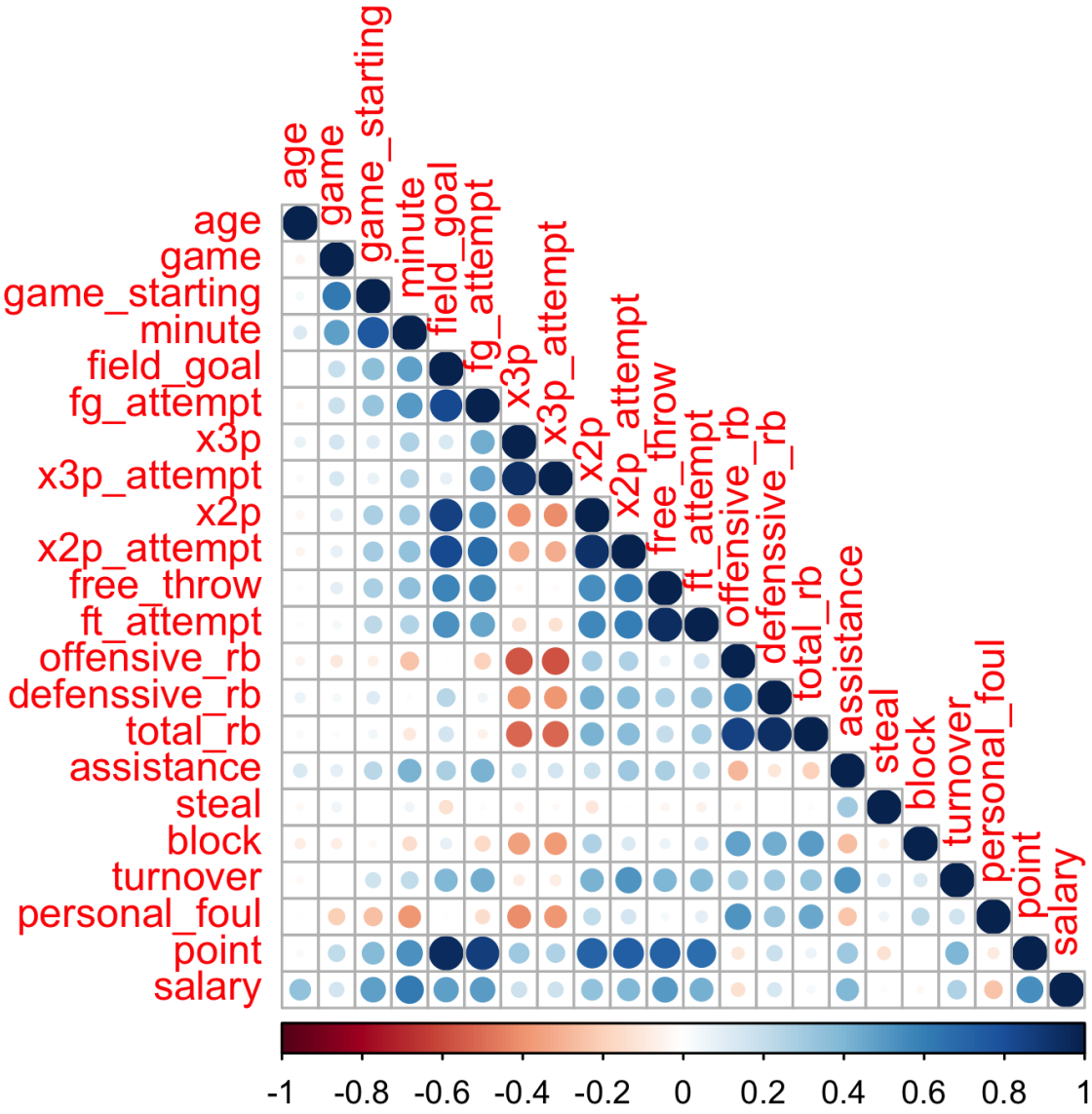


Figure 2: Correlation Heatmap

Analyzing trends in data

From numeric variables, we found that `age`,`game`, `game_starting`,`free_throw`, `personal_foul`, `point` seem to have some non-linear trends. Therefore, non-linear models, including GAM, MARS, random forest and neural network model, were used to predict the salary.

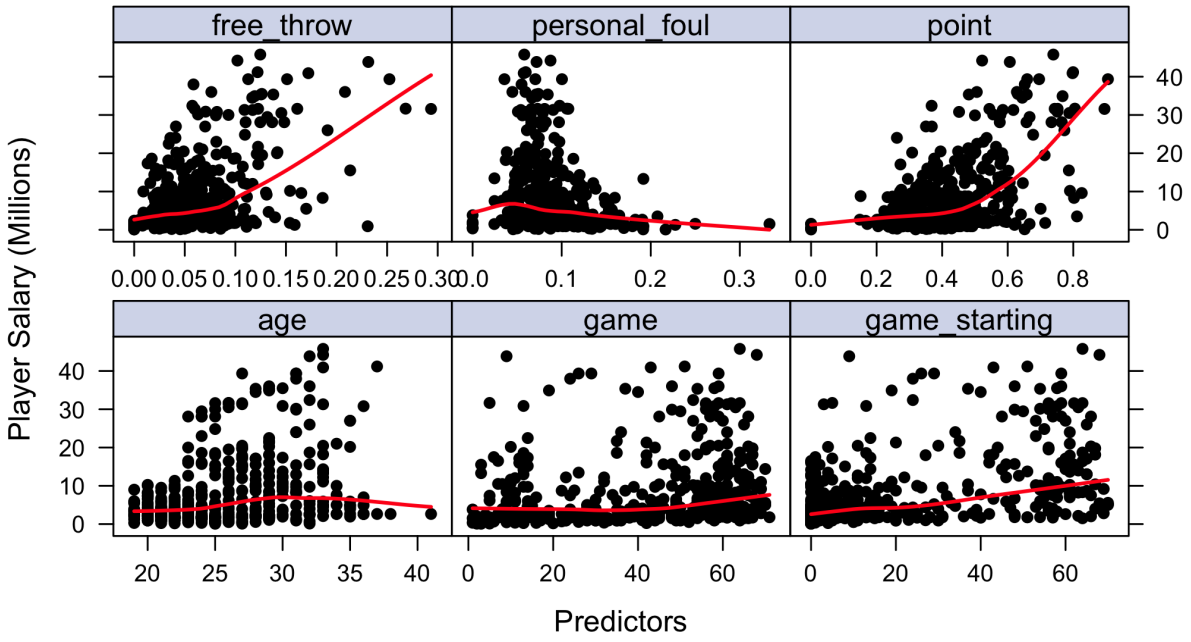


Figure 3: Featureplot of predictors with non-linear trends

From categorical variable `position`, extremely high values in salary show in all positions and some teams (see Appendix A figure5).

Feature Engineering

Categorical variable `team` have 30 classes, which will result in too much dummy variables in our models. Therefore, we consider clustering `team` into fewer class according to similar trends in the median and standard deviation of player's salary in each team. We choose number of clusters k = 3 based on average silhouette width and `team` are clustered into the following 3 clusters:

- Cluster 1: BRK, GSW, LAL, MIA, MIL, NOP, PHI, POR, UTA
- Cluster 2: ATL, CHI, CHO, CLE, DAL, DEN, DET, HOU, IND, MEM, MIN, NYK, OKC, ORL, PHO, SAC, SAS, TOR
- Cluster 3: BOS, LAC, WAS

The resulting new variable is named `team_cluster`.

In tree-based models, we replace variable `team` with the newly generated variable `team_cluster` to achieve higher prediction accuracy. For the rest of our models, we still use variable `team` for model construction.

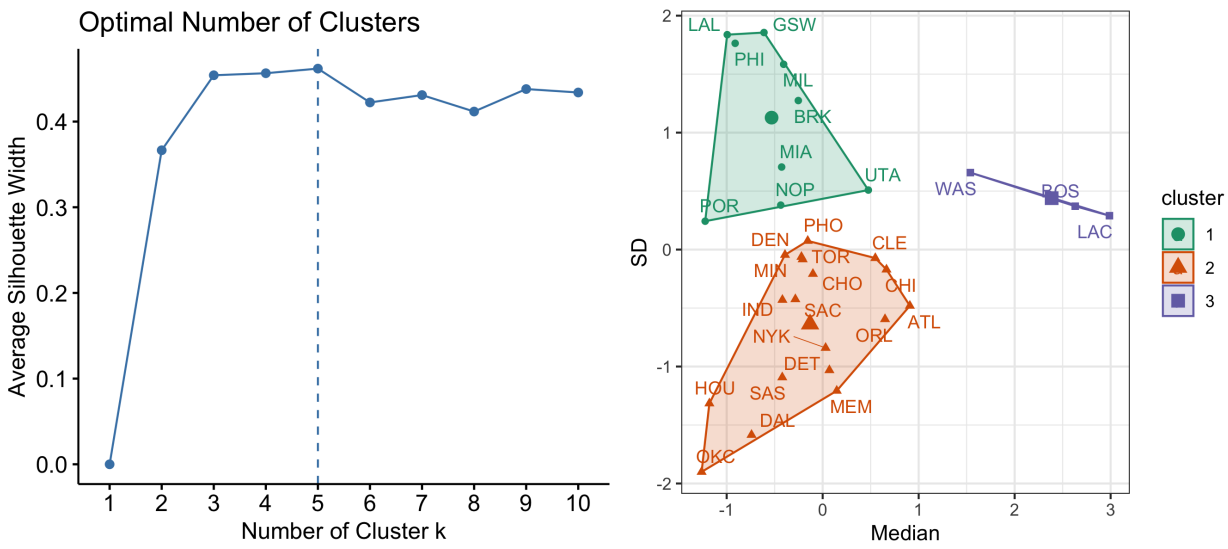


Figure 4: Clustering on variable team

Model Construction

After getting an overview of data from exploratory analysis, we splitted the dataset into training (80%) and testing (20%). We would use 10 fold repeated cross validation to compare each model using training data and then select a best model to predict on testing data. Based on the exploratory analysis, we would build 8 models in four category:

1. Linear Regression: (1) simple Linear Regression Model, (2) Elastic-net Model, (3) Principal Component Regression Model (PCR)
2. Generalized Linear Regression: (4) Generalized Addictive Model (GAM), (5) Multivariate Adaptive Regression Splines Model (MARS)
3. Tree based Models Models: (6) Random Forest, (7) Generalized Boosted Regression Modeling (GBM)
4. Blackbox Model (8) neural network

Part A - Linear Regression Models

(1) Standard Least-Squared There is no tuning parameter for standard least-squared model.

(2) Elastic Net The elastic-net model has two parameter, which are alpha (compromise between LASSO and ridge) and lambda (the penalty term limits the number or magnitude of predictor coefficients). The elastic-net model reached its best tune at $\alpha = 1$ and $\lambda = 0.44$ (see Appendix.B figure1).

(3) Principle Component Regression The tuning parameter of PCR is the number of predictors included in the final model. There are 12 components included in the model with minimum RMSE (see Appendix.B figure2).

Part B - Generalized Linear Regression Models

(4) GAM There is no tuning parameter for GAM. The GAM model can capture the non-linear trend in the model, but it may have a high variance. `age`, `game_starting`, `assistance`, `personal_foul`, and `point` are statistically significant predictors at 0.0001 significant level.

(5) MARS The tuning parameter for MARS is `nprune` and `degree`. When attempting to fit the MARS model, we noticed that the RMSE increased drastically when degree is over 3 and nprune is over 8. Therefore, we would choose the range of degrees as 1:4 and range of nprune as 2:8. When number of terms is 6 and product degree is 3, MARS model reached its best tune and RMSE is lowest. The MARS model selected 6 of 69 terms, and 6 of 54 predictors. And the top 3 important predictors are: `age`, `minute`, `game`. MARS model is highly adaptive comparing with previous models and has a higher prediction accuracy (see Appendix.B figure3).

Part 3: Tree-Based Models

Categorical variable `team` have 30 classes, which will result in too much dummy variables in our models. Therefore, we consider clustering `team` into fewer class according to similar trends in the median and standard deviation of player's salary in each team. We replace `team` with newly generated variable `team_cluster`, which contains values 1, 2, and 3 representing each clusters.

(6) Random Forest Tuning parameter for random forest regression in package `ranger` are `mtry`, number of variables to split at in each node; and `min.node.size`, minimal size of each node. Through 10-fold repeated cv, the optimal random forest model have parameters `mtry = 26` and `min.node.size = 1`. Random forest preserve the advantage of single decision trees that can handle correlation between variables and non-linearity. However, since here `mtry = 26` equals our total number of variables, this random forest estimator may not well decorrelate single trees, and thus may overfit the dataset.

(7) Generalized Boosted Regression Modeling (GBM) Tuning parameters for Generalized boosted regression modeling (GBM) are `n.trees`, the total number of trees to fit; `interaction.depth`: maximum depth of each tree; `shrinkage`, learning rate; and `n.minobsinnode`, the minimum number of observations in the terminal nodes of the trees. Through 10-fold repeated cv, the optimal random forest model have parameters `n.trees = 6000`, `interaction.depth = 5`, `shrinkage = 0.0008`, and `n.minobsinnode = 1`.

Part 4: Neural Network

Several 2-hidden layer neural networks were built to fit the data. Despite trying different number of nodes and applying regularization techniques (L2 and dropout), the resulting models still have a noticeable overfitting problem. Given the size of the dataset is very small ($n = 442$), the performance of neural network is not as good as some traditional statistical models. It is more useful when the size of dataset is much larger with more variables. The optimal model fitted after parameter tuning is a 2-hidden layer neural network with $n_1 = 10$ and $n_2 = 5$ nodes in the first and second layers, applying dropout regularization. The figure shows the resulting MSE in the training and validation sets after 250 epochs.

As shown in the figure, as the number of nodes in the first and second hidden layers increases, neural networks can provide very accurate fittings of the training data, with much lower MSEs compared to other methods. However, the predictions are not satisfying when the models are applying to the test data.

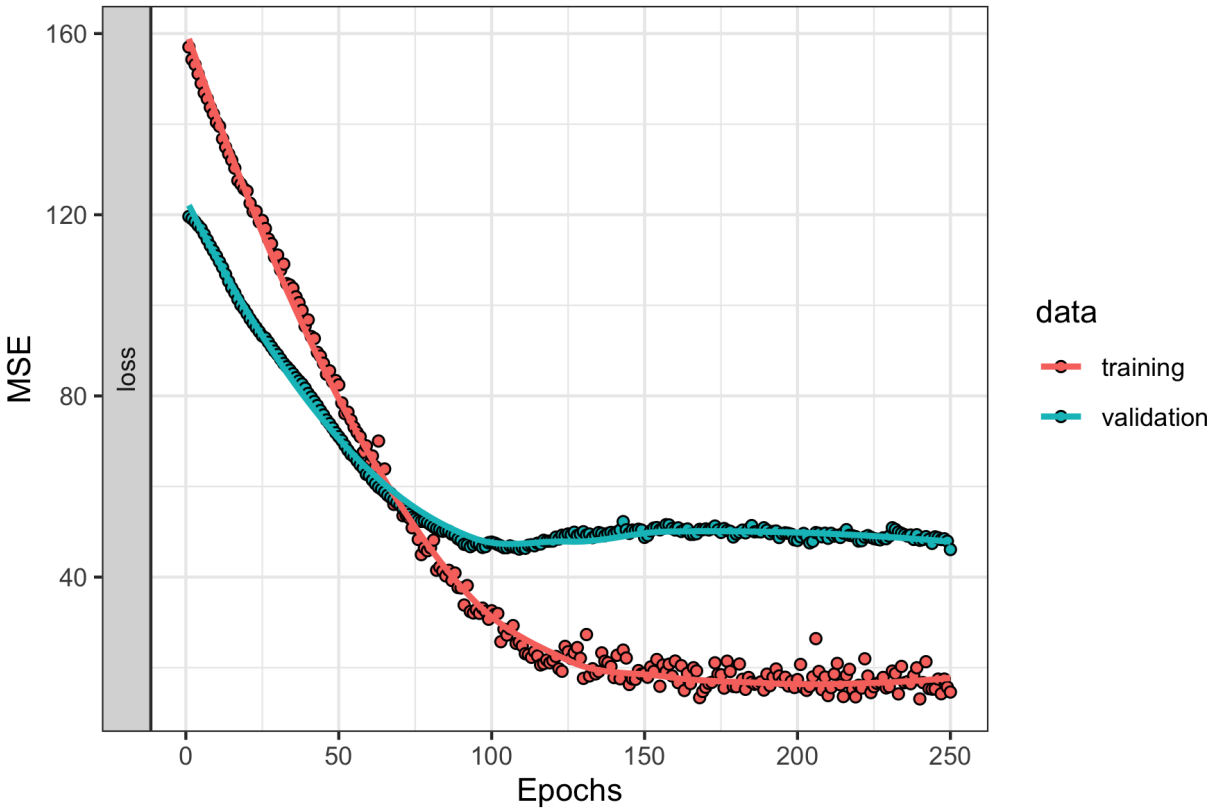
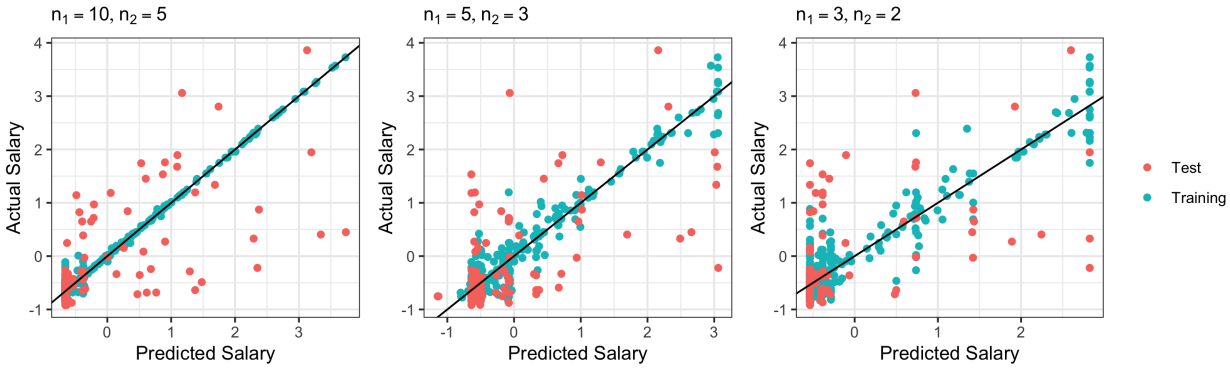


Figure 5: MSE of the resulting 2-hidden layer neural network in training and validation sets



Model Comparasion and Final Model Interpretation

Model Comparison

The 10-fold CV RMSE (validation set rmse for neural network) and test RMSE for all our candidate models are shown in the following table. Generalized Boosted Regression Model have the lowest CV RMSE, thus we will select GBM as our final model.

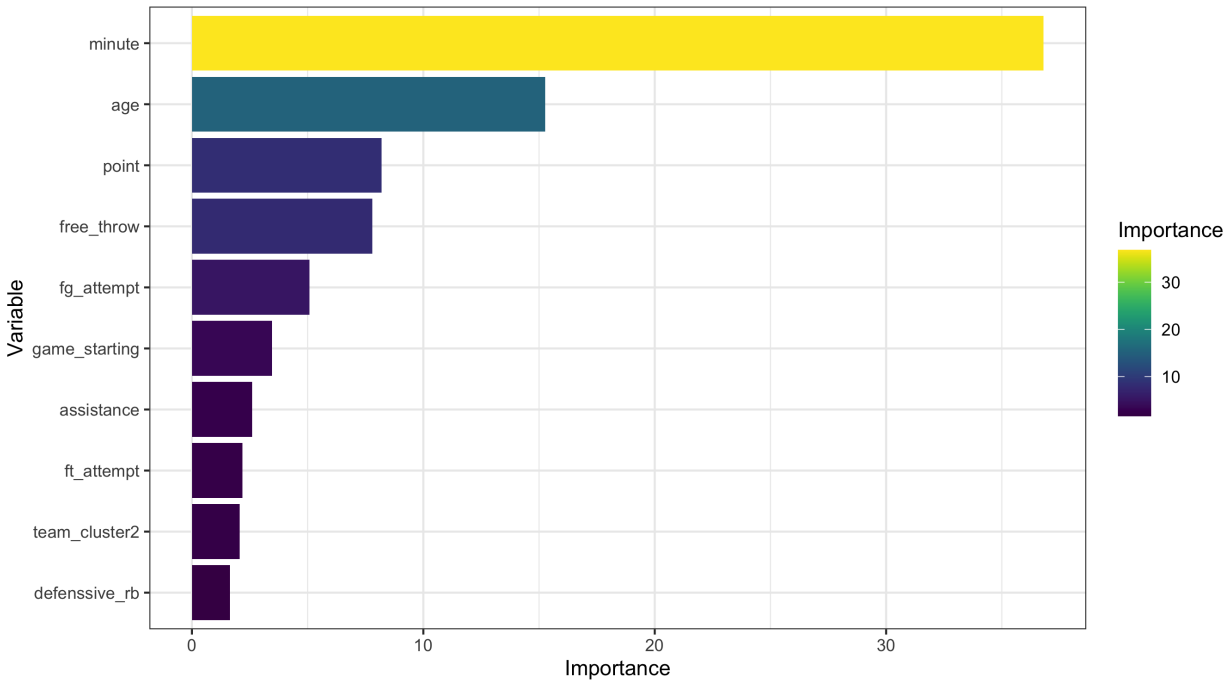
	Linear	ElasticNet	PCR	GAM	MARS	RandomForest	GBM	NeuralNetwork
Training RMSE	6.79	6.45	7.16	6.84	6.06	5.42	5.41	6.40
Test RMSE	6.66	6.04	5.39	6.84	5.16	4.83	4.75	6.64

Final Model Interpretation

Our best model is Generalized Boosted Regression Modeling (GBM) with tuning parameters:

- `n.trees = 6000`
- `interaction.depth = 5`
- `shrinkage = 0.0008`:
- `n.minobsinnode = 1`

10 most important variables (computed from permuting OOB data) are `minute`, `age`, `point`, `free_throw`, `fg_attempt`, `game_starting`, `assistance`, `ft_attempt`, `team_cluster`, and `defensive_rb`.

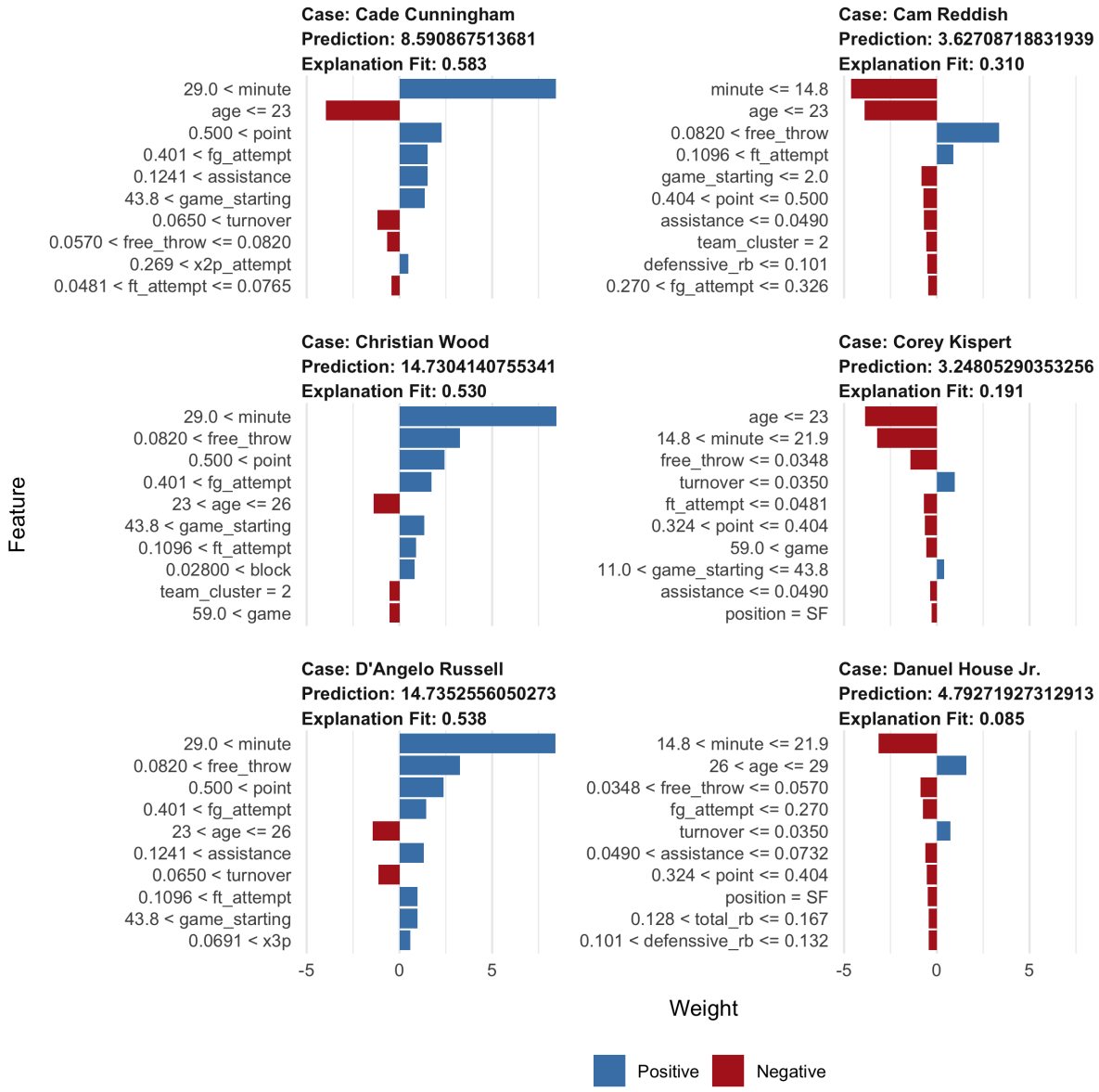


With our fitted GBM model, we can make prediction on new observations. The RMSE on our test data is 4.745948.

Given that GBM is a black-box model, we refer to `lime` package to achieve explanations of the result of the model on new observations, by fitting a simpler model to the permuted data with the above 15 most important features. We randomly selected 6 observations of the test data. The players' name, true salary (in million), and predicted salary from GBM are:

player	true_salary	predicted_salary
Cade Cunningham	10.050120	8.590867
Cam Reddish	4.670160	3.627087
Christian Wood	13.666667	14.730414
Corey Kispert	3.383640	3.248053
D'Angelo Russell	30.013500	14.735256
Danuel House Jr.	2.045094	4.792719

The explanation of the GBM model from lime are shown in the following figure. Inside the plot, the x-axis shows the relative strength of each variables, and positive values (blue) show that the the variable increase the value of the prediction, while the negative values (red) decrease the prediction value.



Take the first case of player Cade Cunningham as an example. Cade's true salary is 10.050120 million. His predicted salary from GBM is 8.590868 million, which are quite similar to each other. Among the 10 most important variables, factors `mintue > 90`, `point > 0.5`, `game_starting > 43.8`, `assistance > 0.1241`, `fg_attempt > 0.401` and `x2p_attempt > 0.269` increases Cade's salary, while factors `age <= 23`, `turnover > 0.065`, `0.057 < free_throw <= 0.082` and `team_cluster = 2` decreases his salary.

Conclusion

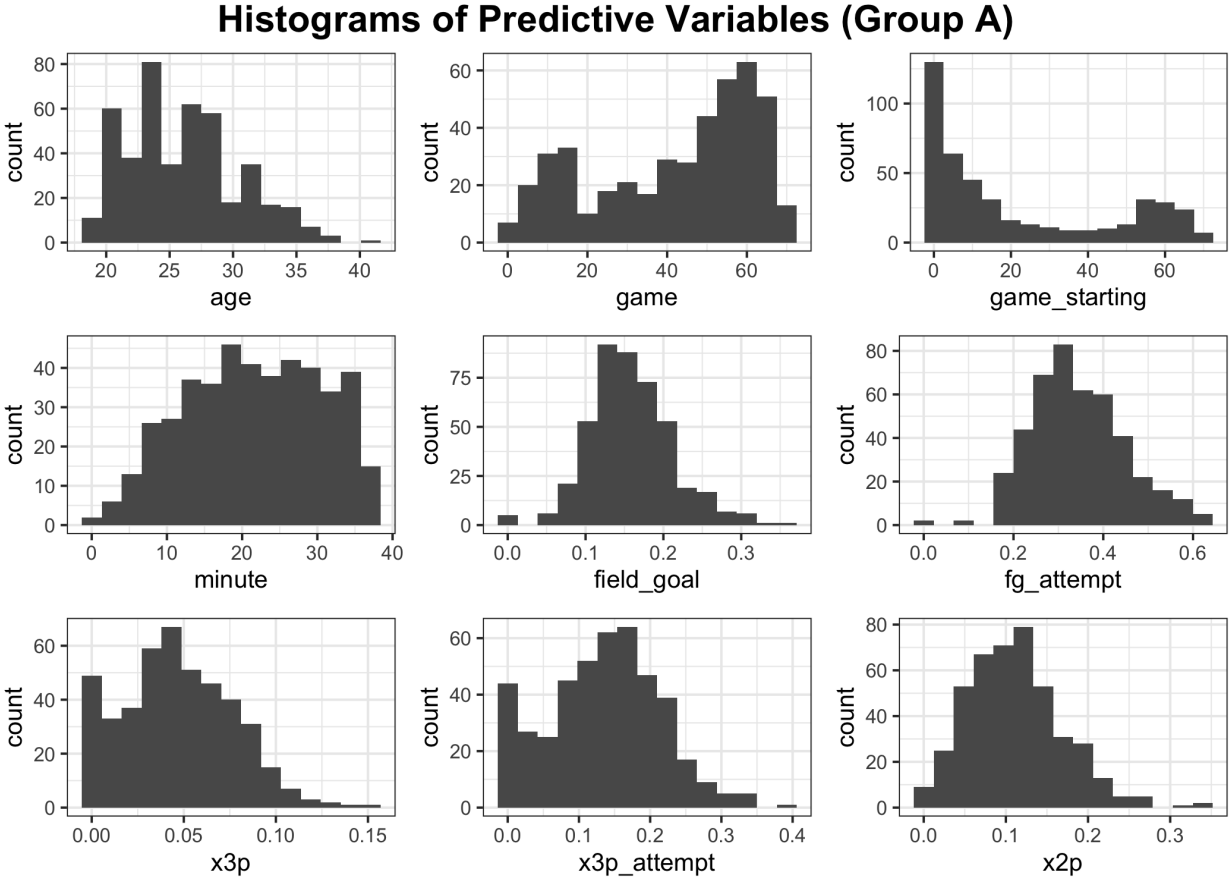
References

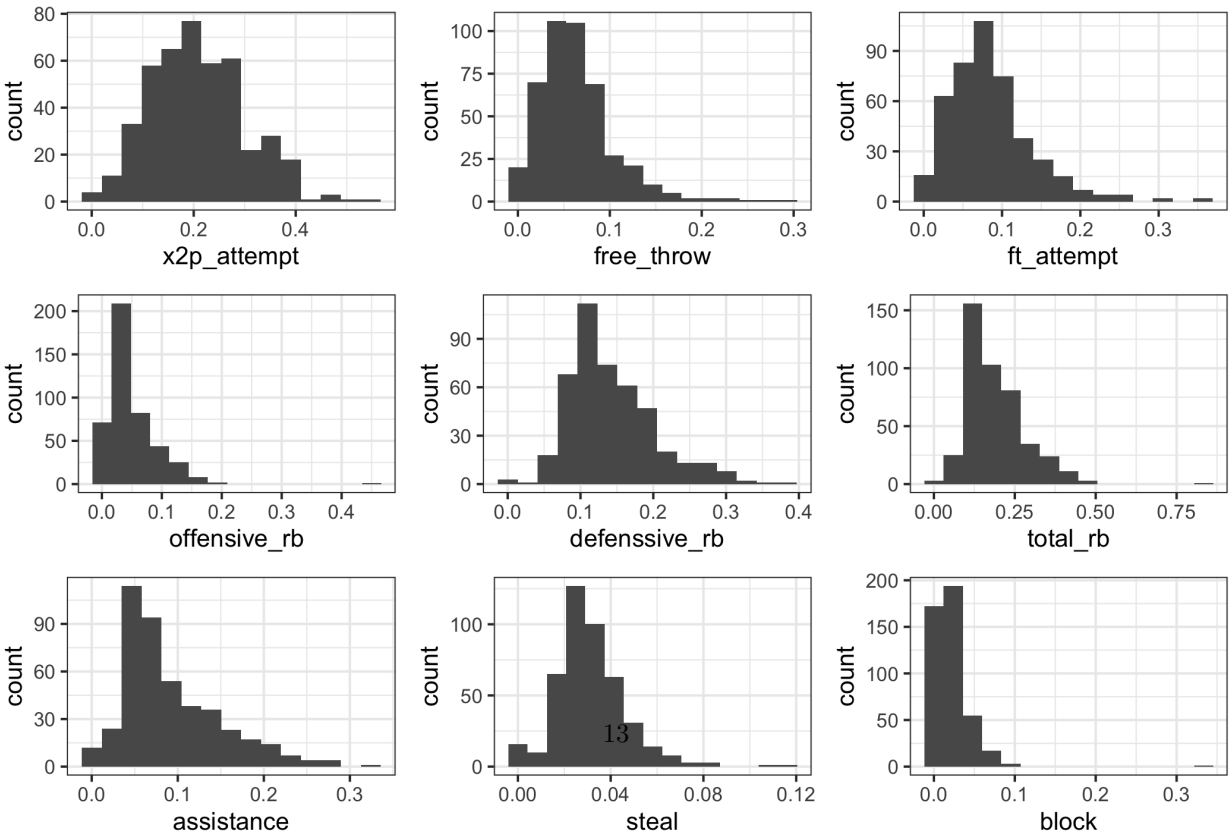
[1]https://www.basketball-reference.com/contracts/players.html

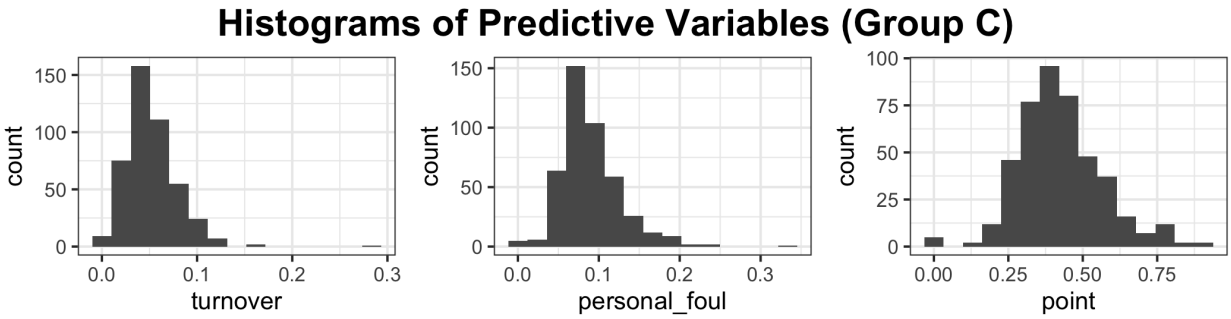
[2]https://www.basketball-reference.com/leagues/NBA_2022_per_game.html

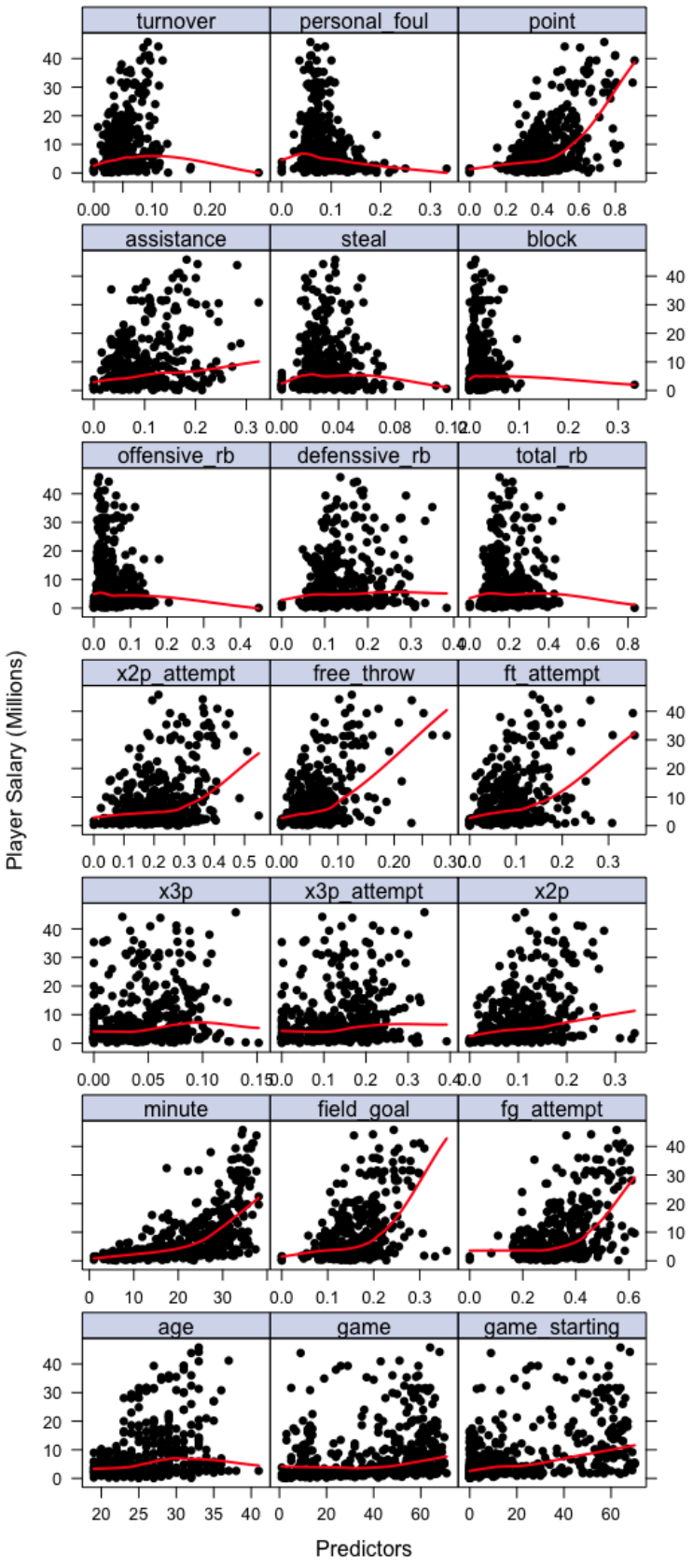
Appendices

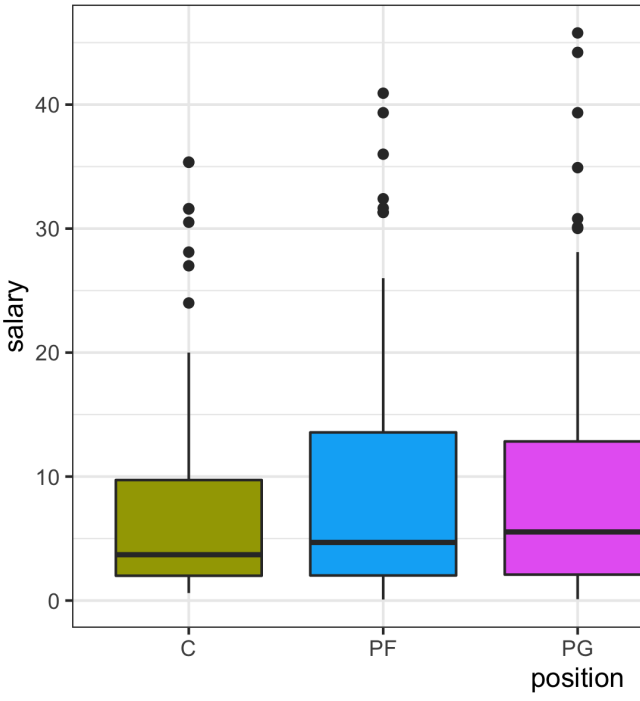
Appendix A - Numeric Variable Distribution



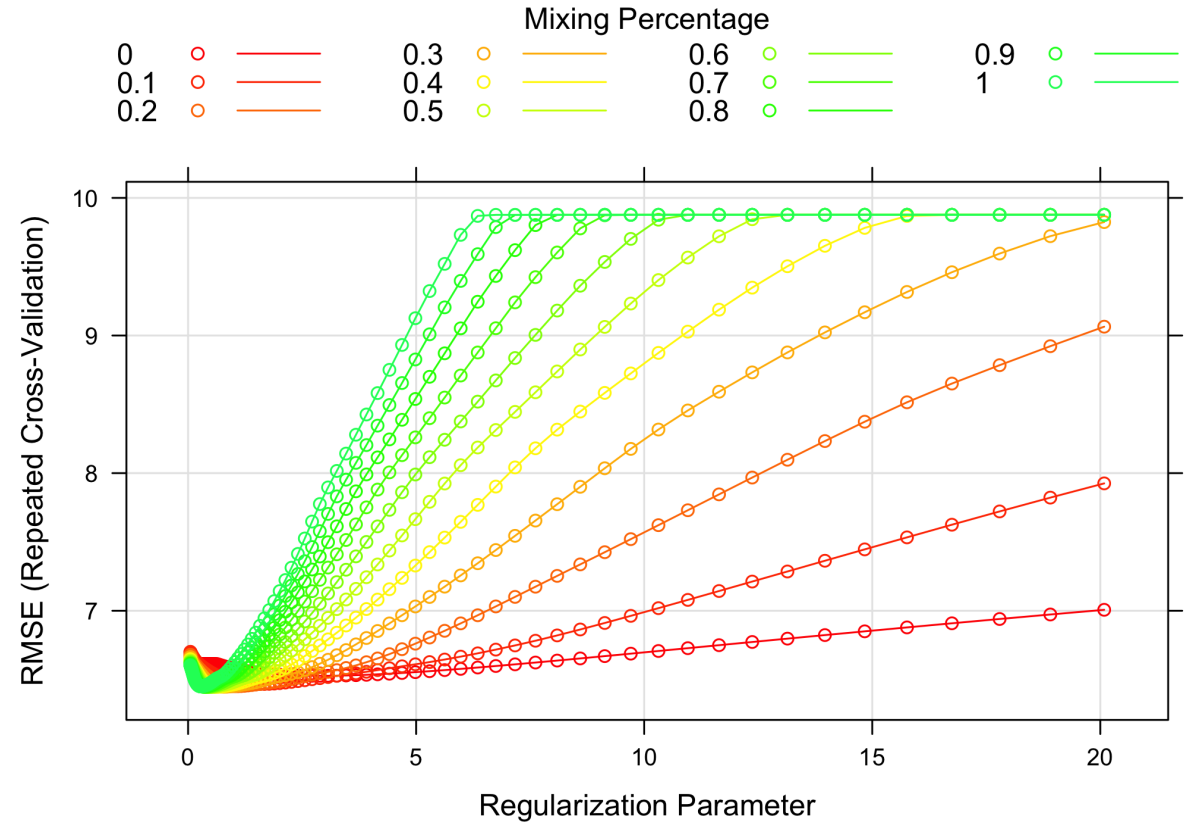


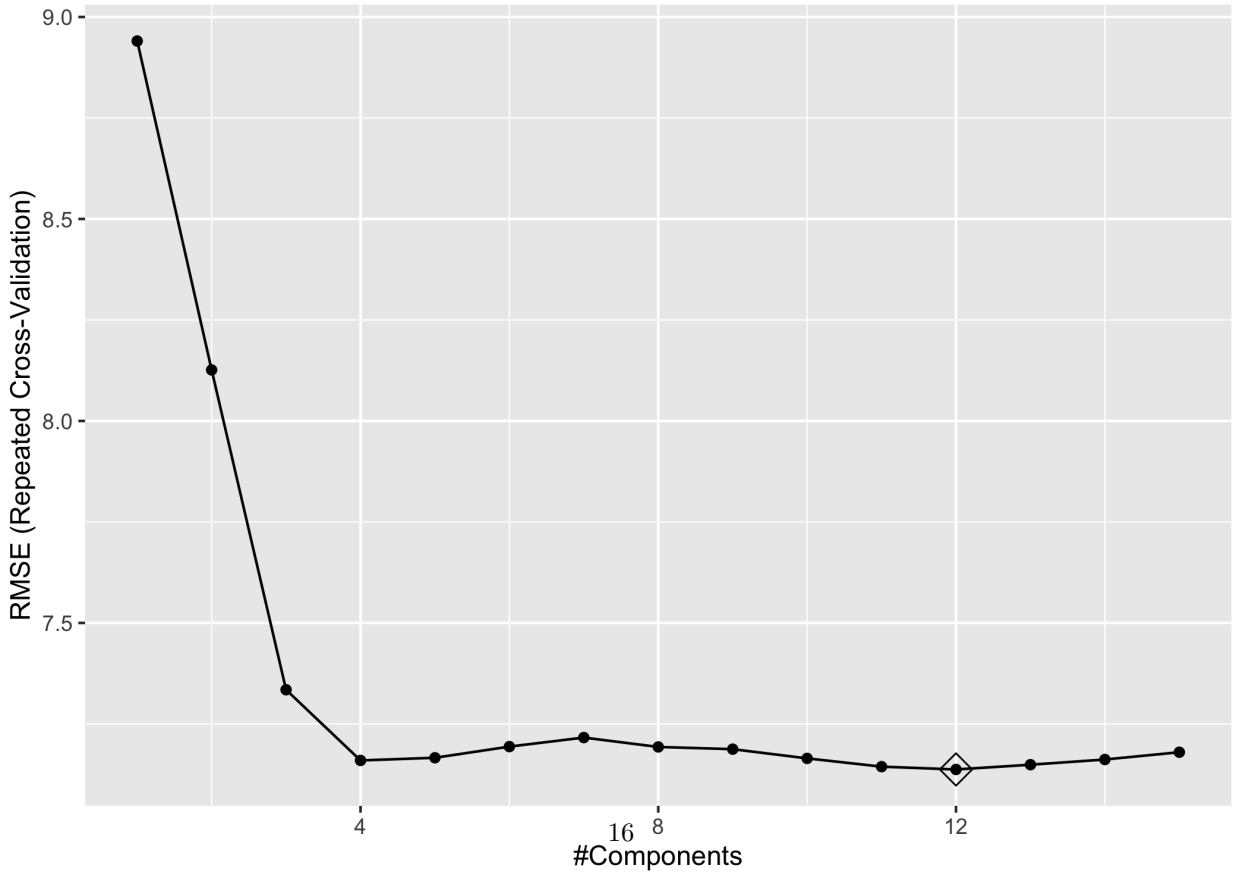

Histograms of Predictive Variables (Group C)
count
turnover
personal_foul
point
0
50
100
150
25
75
0.0
0.1
0.2
0.3
0.00
0.25
0.50
0.75

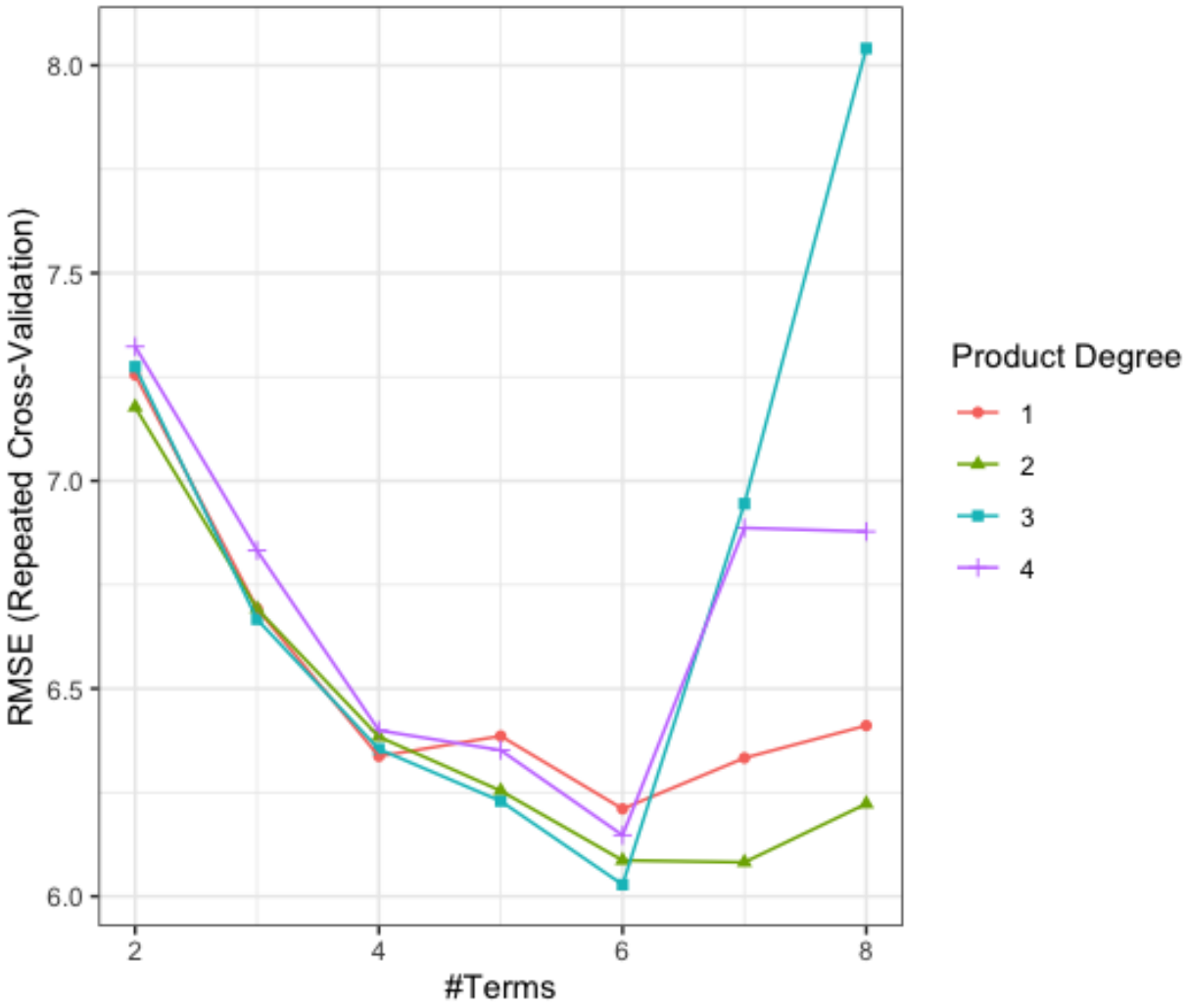

turnover
personal_foul
point
assistance
steal
block
offensive_rb
defenssive_rb
total_rb
x2p_attempt
free_throw
ft_attempt
x3p
x3p_attempt
x2p
minute
field_goal
fg_attempt
age
game
game_starting
Player Salary (Millions)
Predictors


salary
C
PF
PG
position

Appendix B - Linear Regerssion







Appendix C - Neural Network Model