P8106 Final - Models except for NN

Mingkuan Xu, Mengfan Luo, Yiqun Jin

5/6/2022

Data Preprocessing

```
df_salary = read_csv("NBA_season2122_player_salary.csv") %>%
  janitor::clean_names() %>%
  select(Player=x2,Team=x3,Salary=salary_4) %>%
  na.omit()

df_salary = df_salary[-1,]

df_stats = read_csv("NBA_season2122_player_stats.csv") %>%
  rename(Team=Tm) %>%
  select(-Rk)

df_players = inner_join(x=df_salary,y=df_stats,by=c("Player","Team")) %>%
  janitor::clean_names() %>%
  distinct()

df_players = df_players %>%
  arrange(player,desc(g)) %>%
  distinct(player,.keep_all = TRUE)

# Removed variables with missing data and resulted from division of other variables
df_players = df_players %>%
  select(-x3p_percent, -ft_percent, -fg_percent,-x2p_percent,-e_fg_percent)

# The final generated dataset for use: df_player.
```

```
# Convert salary from characters to numbers.
# Convert categorical variables to factors

df_players = df_players %>%
  separate(salary,into = c("symbol", "salary"),1) %>%
  select(-symbol)%>%
  mutate(salary = as.numeric(salary)/1000000,
         team = factor(team),
         pos = factor(pos)) %>%
  relocate(salary, .after = last_col())

colnames(df_players) = c("player", "team", "position", "age", "game","game_starting" ,"minute","field_g
```

```r
df_players = df_players %>%
  distinct(player,.keep_all = TRUE) %>%
  mutate(player = gsub("\\\\.*","",player)) %>%
  `row.names<-`(., NULL) %>%
  column_to_rownames('player')
```

```r
# Convert count data to rate by dividing variable `minute`

df_players = df_players %>%
  mutate(field_goal = field_goal/minute,
         fg_attempt = fg_attempt/minute,
         x3p = x3p/minute,
         x3p_attempt = x3p_attempt/minute,
         x2p = x2p/minute,
         x2p_attempt = x2p_attempt/minute,
         free_throw = free_throw/minute,
         ft_attempt = ft_attempt/minute,
         offensive_rb = offensive_rb/minute,
         defenssive_rb = defenssive_rb/minute,
         total_rb = total_rb/minute,
         assistance = assistance/minute,
         steal = steal/minute,
         block = block/minute,
         turnover = turnover/minute,
         personal_foul = personal_foul/minute,
         point = point/minute)
```

Models

```r
# Data partition
set.seed(8106)

indexTrain <- createDataPartition(y = df_players$salary, p = 0.8, list = FALSE, times = 1)
df_train <- df_players[indexTrain, ]
df_test <- df_players[-indexTrain, ]
df_train_2 = model.matrix(salary ~ ., df_train)[ ,-1]
df_test_2 = model.matrix(salary ~ ., df_test)[ ,-1]
x = df_train_2
y = df_train %>% pull(salary)

ctrl1 <- trainControl(method = "repeatedcv", number = 10, repeats = 5)
```

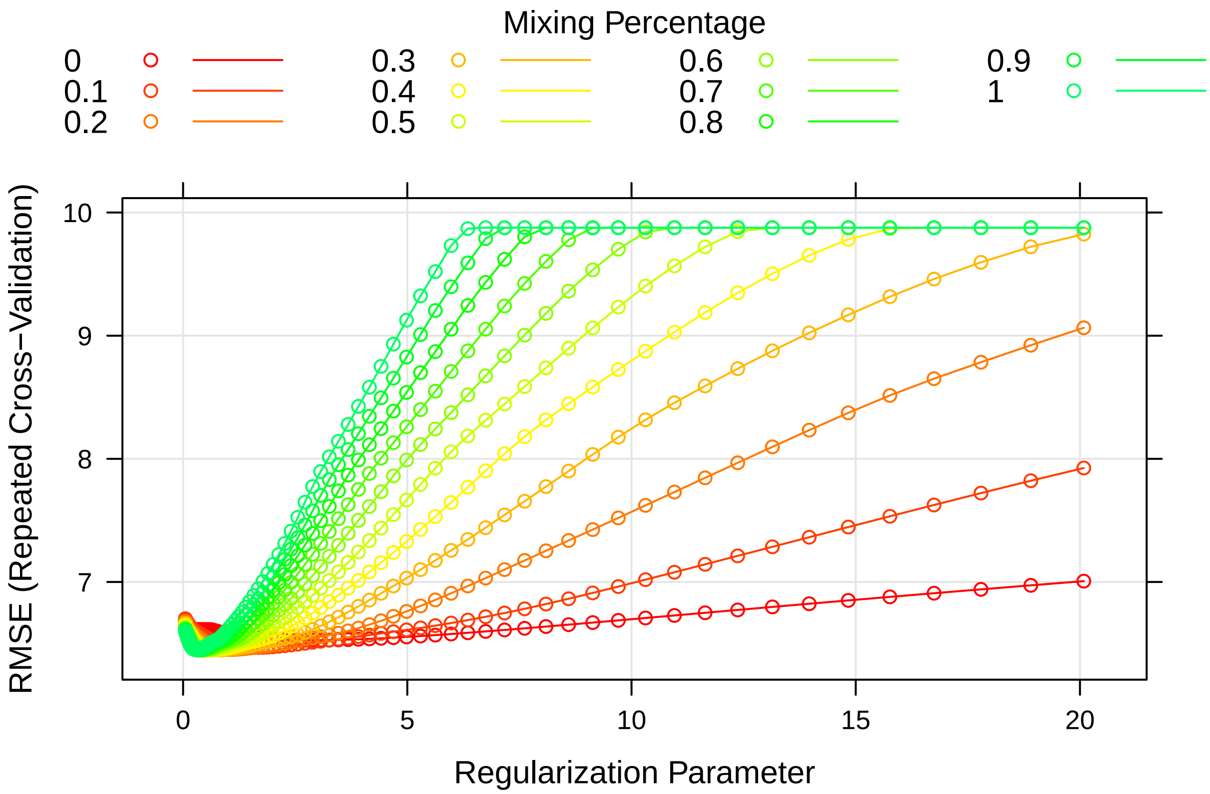
Part 1 Linear regression

(a) Standard Least-Squared

(b) Elastic Net (including lasso/ridge)

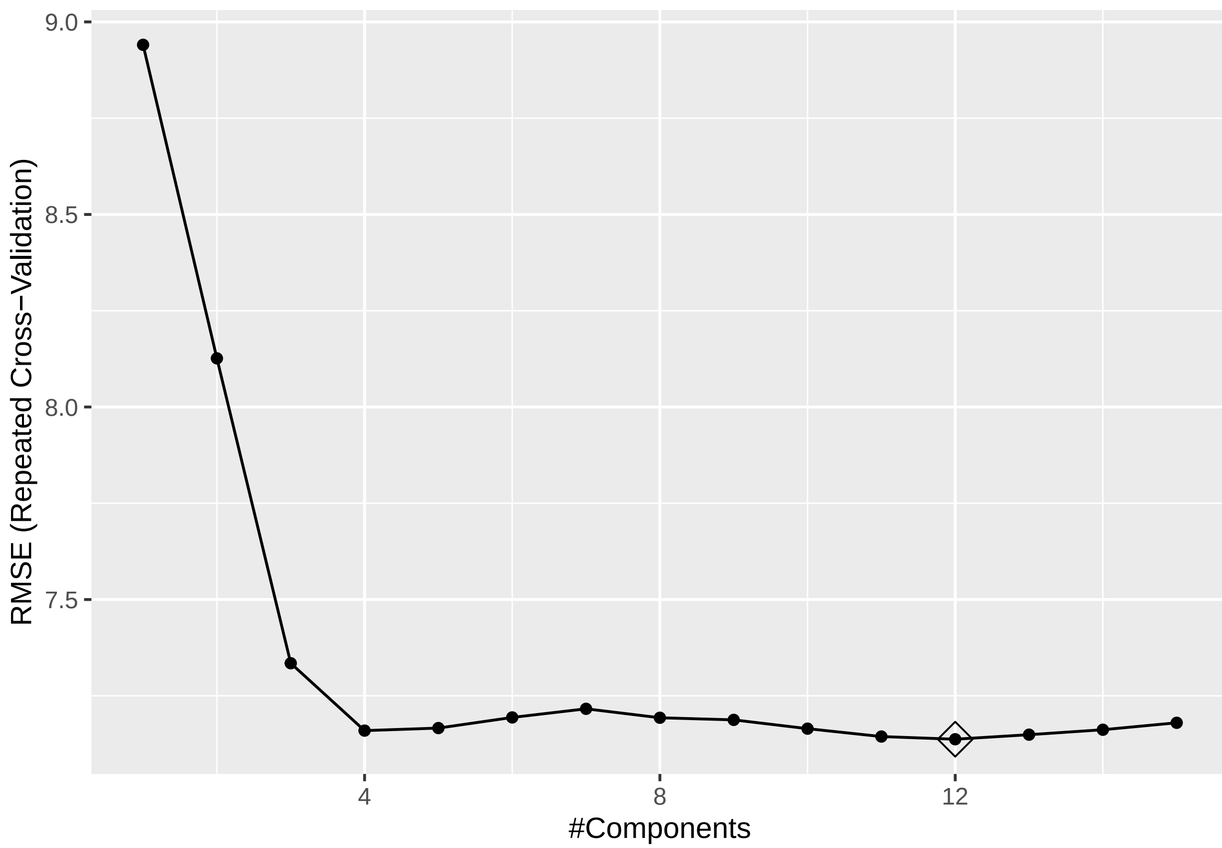
```
##      alpha    lambda
```

```
## 637   0.6 0.4412332
```



(c) Principle Component Regression

```
##    ncomp
## 12    12
```



Part 2 Generalized Linear Regression

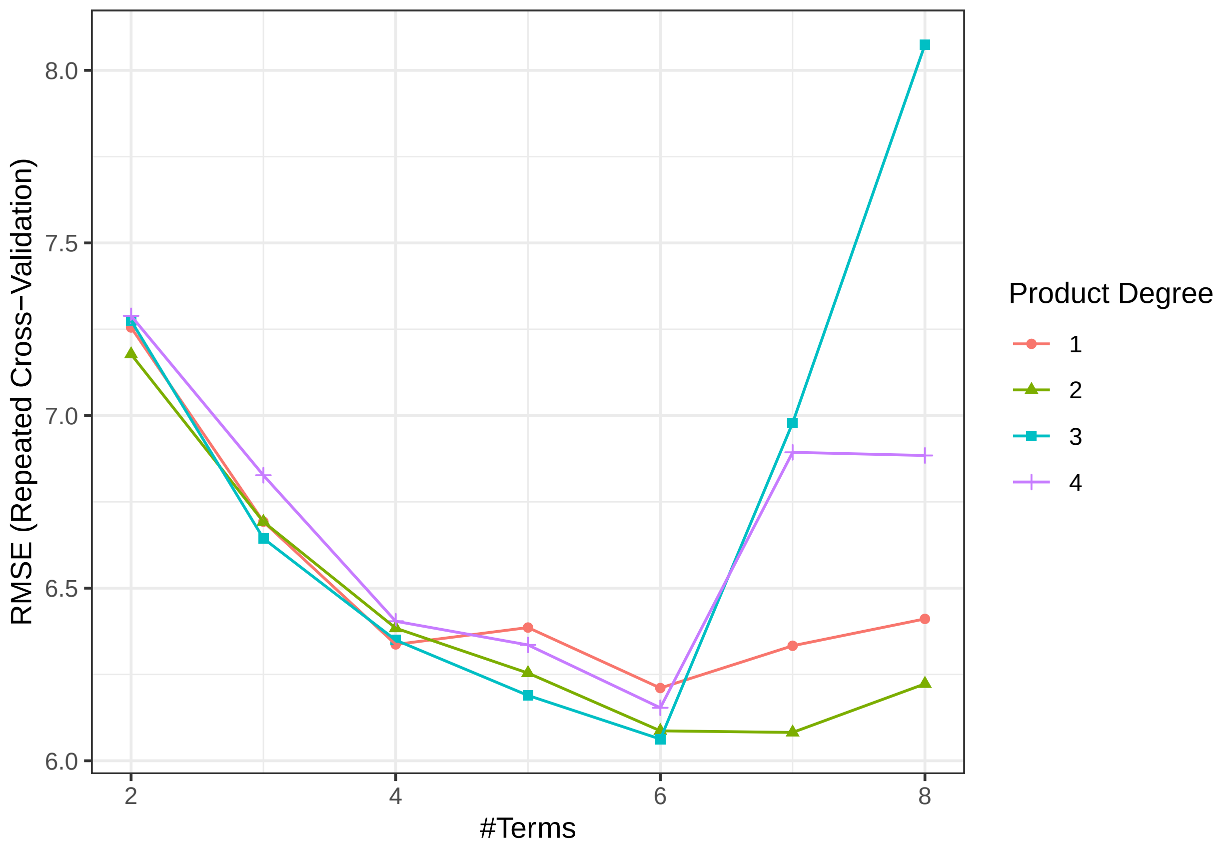
(a) GAM

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## salary ~ s(age) + s(game) + s(game_starting) + s(free_throw) +
##     s(ft_attempt) + s(defenssive_rb) + s(assistance) + s(block) +
##     s(personal_foul) + s(point)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.5293     0.2958   28.84   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                    edf Ref.df      F  p-value
## s(age)           4.414  5.455 16.961  < 2e-16 ***
## s(game)          1.695  2.101  4.623  0.00973 **
## s(game_starting) 1.482  1.805 25.494  < 2e-16 ***
## s(free_throw)    8.147  8.791  3.083  0.00538 **
```

```
## s(ft_attempt)    1.000  1.000  0.155  0.69382
## s(defenssive_rb) 1.000  1.000  1.680  0.19591
## s(assistance)    1.000  1.000 18.244 2.58e-05 ***
## s(block)         1.000  1.000  2.758  0.09777 .
## s(personal_foul) 6.851  7.891  5.172 6.56e-06 ***
## s(point)         6.152  7.361  5.415 5.90e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =   0.69   Deviance explained = 71.8%
## GCV = 34.237  Scale est. = 30.974    n = 354
```

(b) MARS

```
##    nprune degree
## 19      6      3
```




```
## [1] 26.58079
```

Part 3 Tree-based models

Feature engineering for tree-based models

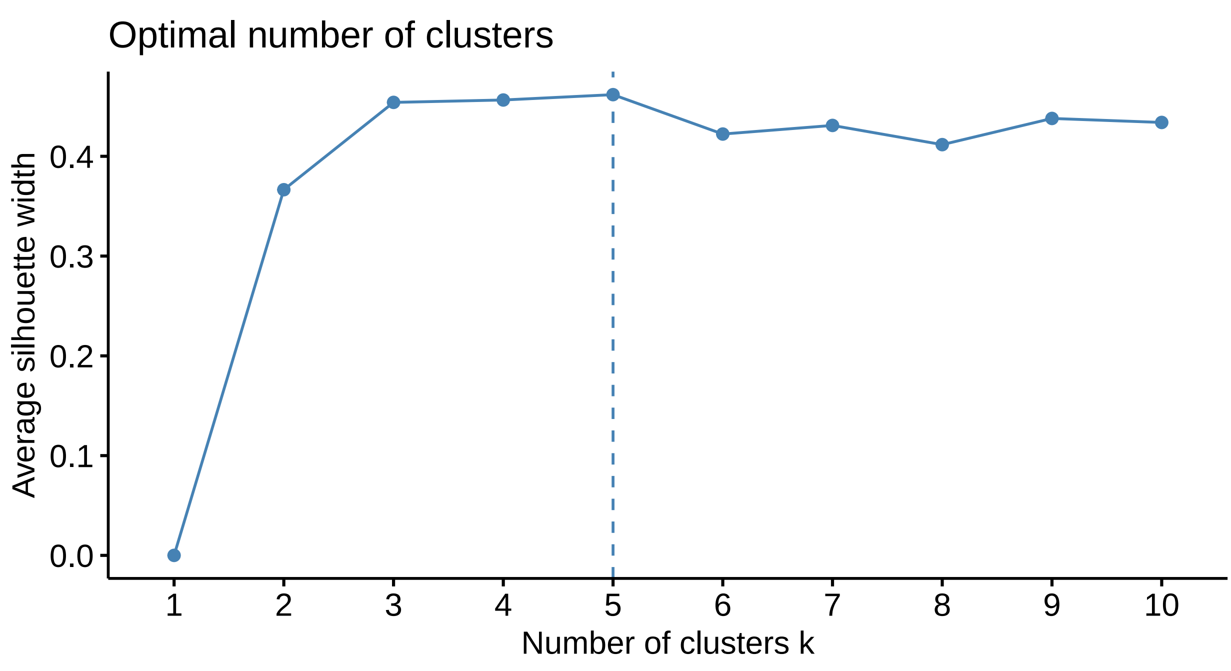
Categorical variable `team` have 30 classes, which will resulted in too much dummy variables in our models. Therefore, we consider clustering `team` into fewer class according to similar trends in the median and standard deviation of player's salary in each team.

```
df_team = df_players[indexTrain,] %>%
  group_by(team) %>%
  summarize(median = median(salary),
            sd = sd(salary)) %>%
  mutate(team = as.character(team))

df_team1 = data.frame(median = df_team$median, sd = df_team$sd)
rownames(df_team1) = df_team$team
df_team1  = scale(df_team1)
```

We use k-mean clustering to cluster variable `team` in the training data with class number k = 3. Variable `team` are clustered into the following 3 clusters:

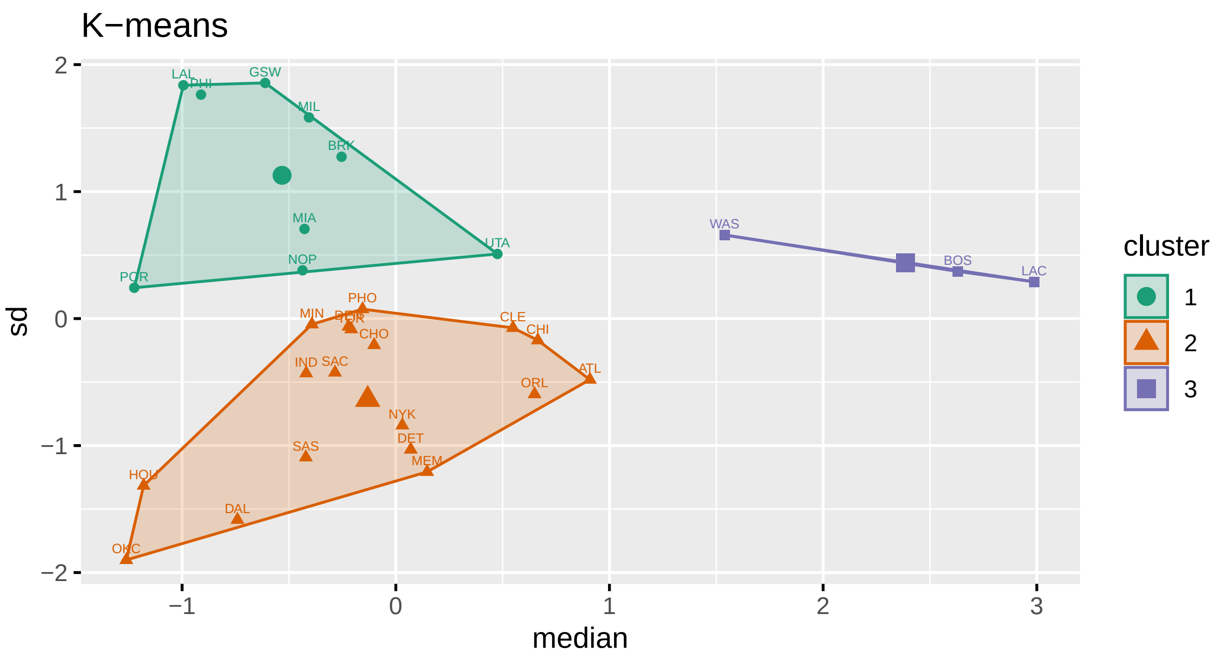
- Cluster 1: BRK, GSW, LAL, MIA, MIL, NOP, PHI, POR, UTA
- Cluster 2: ATL, CHI, CHO, CLE, DAL, DEN, DET, HOU, IND, MEM, MIN, NYK, OKC, ORL, PHO, SAC, SAS, TOR
- Cluster 3: BOS, LAC, WAS

```
set.seed(8106)
fviz_nbclust(df_team1,
             FUNcluster = kmeans,
             method = "silhouette")
```




```
km <- kmeans(df_team1, centers = 3, nstart = 30)

km_vis <- fviz_cluster(list(data = df_team1, cluster = km$cluster),
                       ellipse.type = "convex",
                       geom = c("point","text"),
```

```
                                labelsize = 5,
                                palette = "Dark2") + labs(title = "K-means")

km_vis
```




```
team_dict = data.frame(
  team = df_team$team,
  team_cluster = factor(unname(km$cluster))
)
```

We add class labels for the newly generated clusters of `team` as `team_cluster`, with values 1, 2, and 3 representing each clusters.

```
df_players2 = inner_join(x = df_players,y = team_dict,by = "team") %>%
  relocate(team_cluster, .before = team) %>%
  select(-team)
```

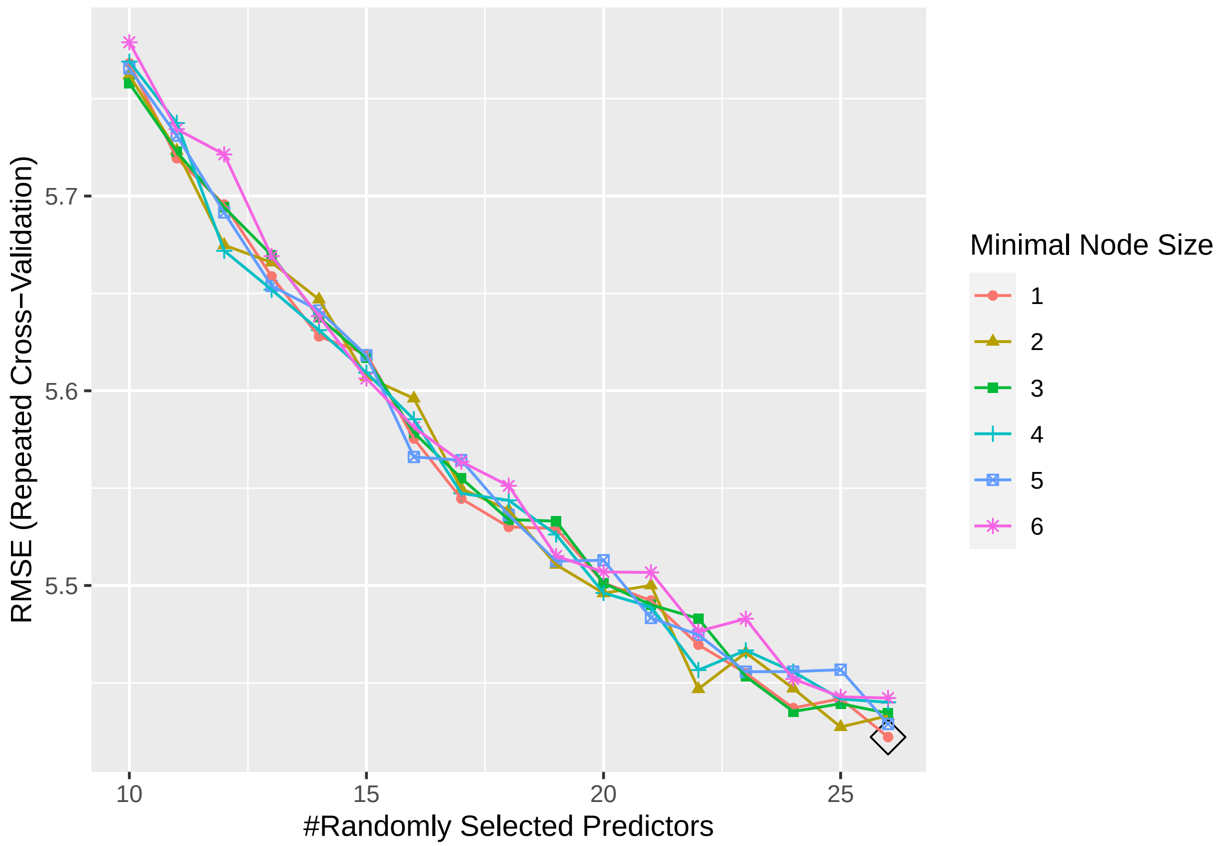
(a) Random forest

```
rf.grid3 <- expand.grid(
  mtry = 10:26,
  splitrule = "variance",
  min.node.size = 1:6)

set.seed(8106)
rf.fit3 <- train(salary ~ . ,
                df_players2[indexTrain,][1:24],
                method = "ranger",
                tuneGrid = rf.grid3,
```

```
                  trControl = ctrl1)

rf.fit3$bestTune
```

```
##    mtry splitrule min.node.size
## 97   26  variance             1
```

```
ggplot(rf.fit3, highlight = TRUE)
```




```
y_test = df_players[-indexTrain,]$salary
y_pred <- predict(rf.fit3, newdata = df_players2[-indexTrain,])
rf.mse = mean((y_pred - y_test)^2)
```

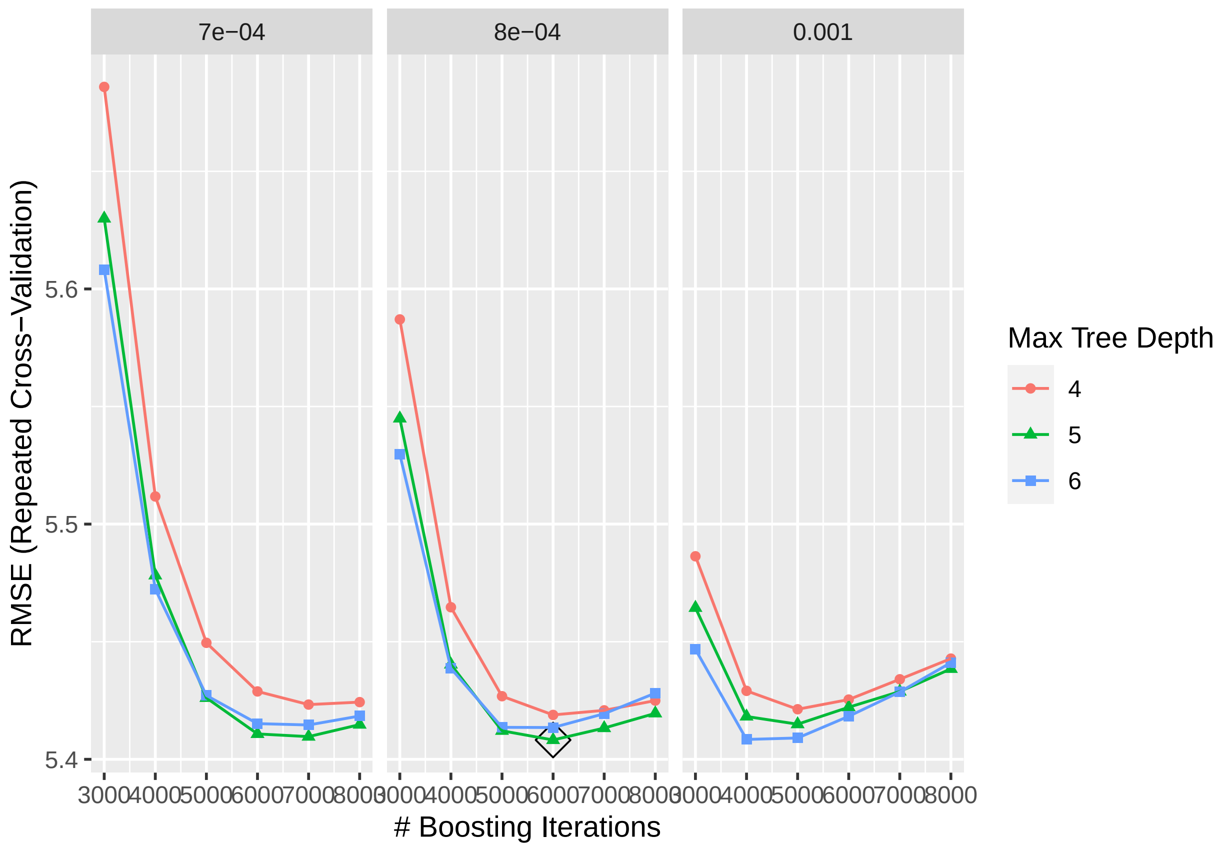
(b) Generalized Boosted Regression Modeling (GBM)

```
gbm.grid3 <- expand.grid(n.trees = c(3000,4000,5000,6000,7000,8000),
                        interaction.depth = 4:6,
                        shrinkage = c(0.0007,0.0008,0.001),
                        n.minobsinnode = 1)

set.seed(8106)
gbm.fit3 <- train(salary ~ . ,
```

```
                  df_players2[indexTrain,][1:24],
                  method = "gbm",
                  tuneGrid = gbm.grid3,
                  trControl = ctrl1,
                  verbose = FALSE)
gbm.fit3$bestTune
```

```
##    n.trees interaction.depth shrinkage n.minobsinnode
## 28    6000                 5     8e-04              1
```

```
ggplot(gbm.fit3, highlight = TRUE)
```




```
gbm.fit3$finalModel
```

```
## A gradient boosted model with gaussian loss function.
## 6000 iterations were performed.
## There were 27 predictors of which 27 had non-zero influence.
```

```
y_test = df_players[-indexTrain,]$salary
y_pred <- predict(gbm.fit3, newdata = df_players2[-indexTrain,])
gbm.mse = mean((y_pred - y_test)^2)
```

Table 1: RMSE of Different Models

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
LeastSquare	4.41	6.12	6.85	6.79	7.46	8.75	0
ElasticNet	4.57	5.95	6.37	6.45	7.06	8.55	0
PCR	5.17	6.24	7.17	7.14	7.87	9.34	0
MARS	4.05	5.26	6.04	6.06	6.74	8.74	0
RF	3.24	4.66	5.48	5.42	5.99	7.52	0
GBM	3.52	4.79	5.48	5.41	6.11	7.38	0

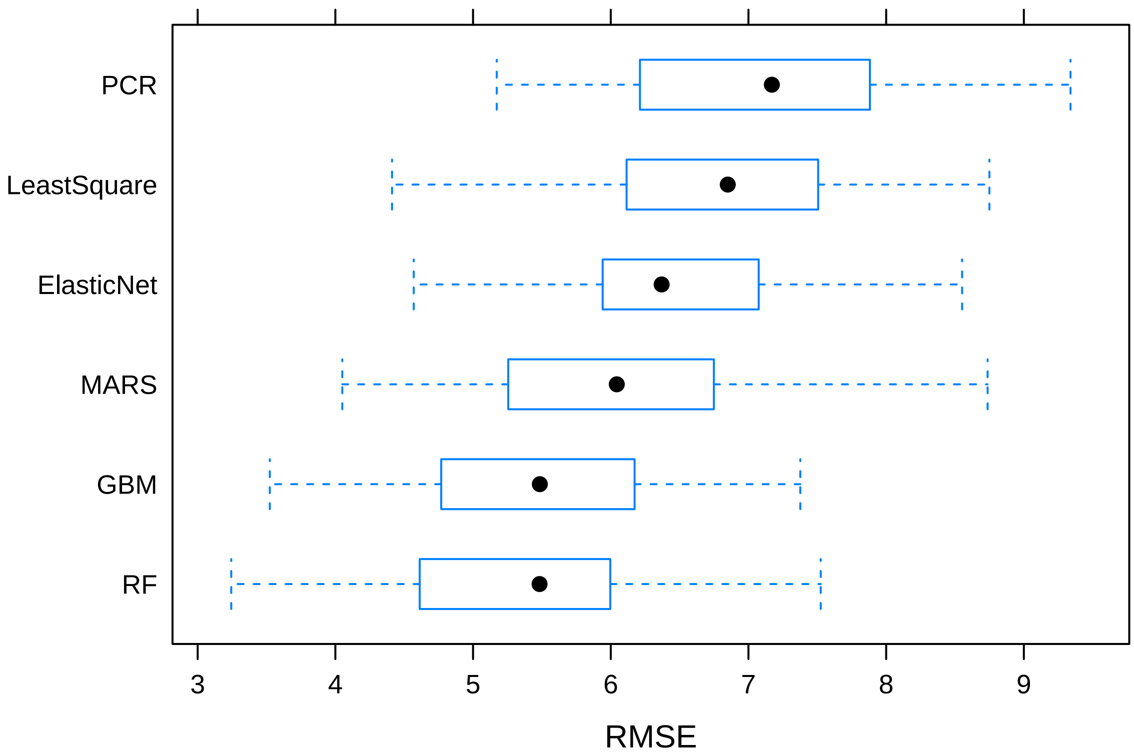


Table 2: RMSE of Different Models on Test Set

	Linear	ElasticNet	PCR	GAM	MARS	RandomForest	GBM
RMSE	6.66	6.04	5.46	6.84	5.16	4.83	4.75