NBA 선수들의 기록에 따른 연봉모델

박장대소

201612069 박종혁

201612070 박태현

201612097 장유성

데이터마이닝 Team Project

지도교수 : 김용수 교수님

목차

NBA 선수들의 연봉협상 모델 제작

- 001 주제 선정 및 배경
- 002 변수설명
- 003 데이터 전처리
- 004 데이터 시각화
- 005 다중회귀분석
- 006 의사결정나무
- 007 인공신경망
- 008 모형 비교 및 결론
- 부록

001 – 주제설명

스토브리그란?

- 비시즌 기간동안 구단과 선수단이 회의를 통해 다음시즌 방향을 잡는 과정.
- 활동 : **연봉협상**, 선수영입, 드래프트 등

박장대소의 주제

1. 연봉이 높은 선수들의 **기록 특징** 파악.
2. 구단 입장에서 선수의 능력에 알맞은 **연봉 예측 모델** 생성

002 – 변수설명

변수	변수명	변수설명	형태
종속변수(Y)	Salary	선수 임금(달러)	수치형
독립변수(X1)	Age	나이	수치형
독립변수(X2)	G	경기출장횟수	수치형
독립변수(X3)	GS	선발출장횟수	수치형
독립변수(X4)	MP	경기출장시간(분)	수치형
독립변수(X5)	PER	선수효율지수	수치형
독립변수(X6)	TS%	슈팅효율성지수	수치형
독립변수(X7)	ORB	공격리바운드개수	수치형
독립변수(X8)	DRB	수비리바운드개수	수치형
독립변수(X9)	AST	도움개수	수치형

변수	변수명	변수설명	형태
독립변수(X20)	FT%	자유투 성공률	수치형
독립변수(X21)	PF	개인 파울	수치형
독립변수(X22)	PTS	득점	수치형

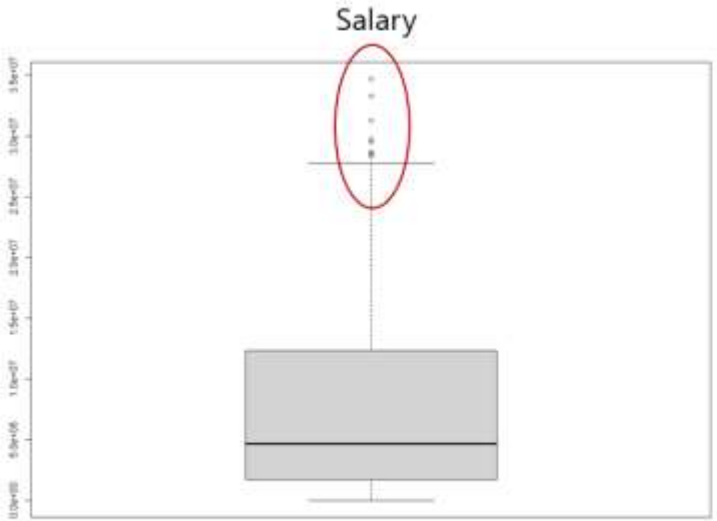
'NBA players' Salary Dataset (2017 - 2018)'을 바탕으로 종속변수와 독립변수로 나눔.

종속변수 : **1**개
독립변수 : **22**개

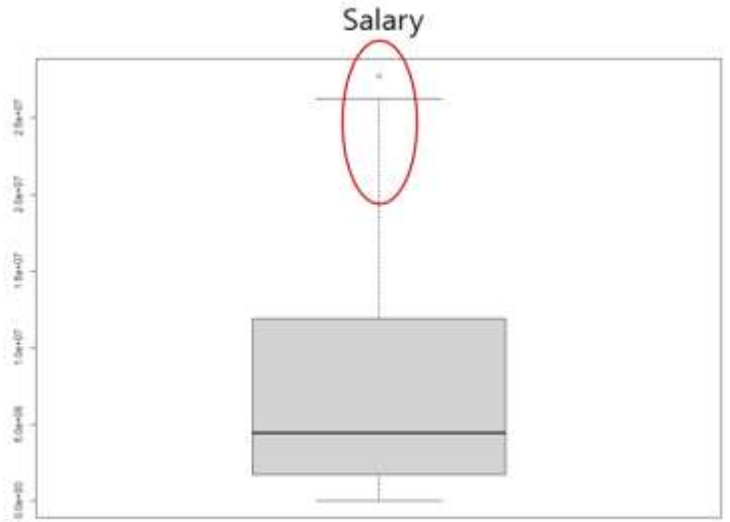
종속변수인 Salary과 독립변수들 모두 : **수치형**

003 데이터 전처리 – 이상치 제거 및 변화

- 변수들의 이상치 제거전과 평균대치법 적용하여 나타낸 boxplot 그림
- 이상치를 결측치(NA)로 바꿔서 하였으나, 관측된 자료를 무시하므로 생기는 효율성 상실과 통계적 추론의 문제가 생겼음.
- 따라서 자료의 평균값으로 결측값을 대치하는 **평균대치법** 활용



[이상치 존재]



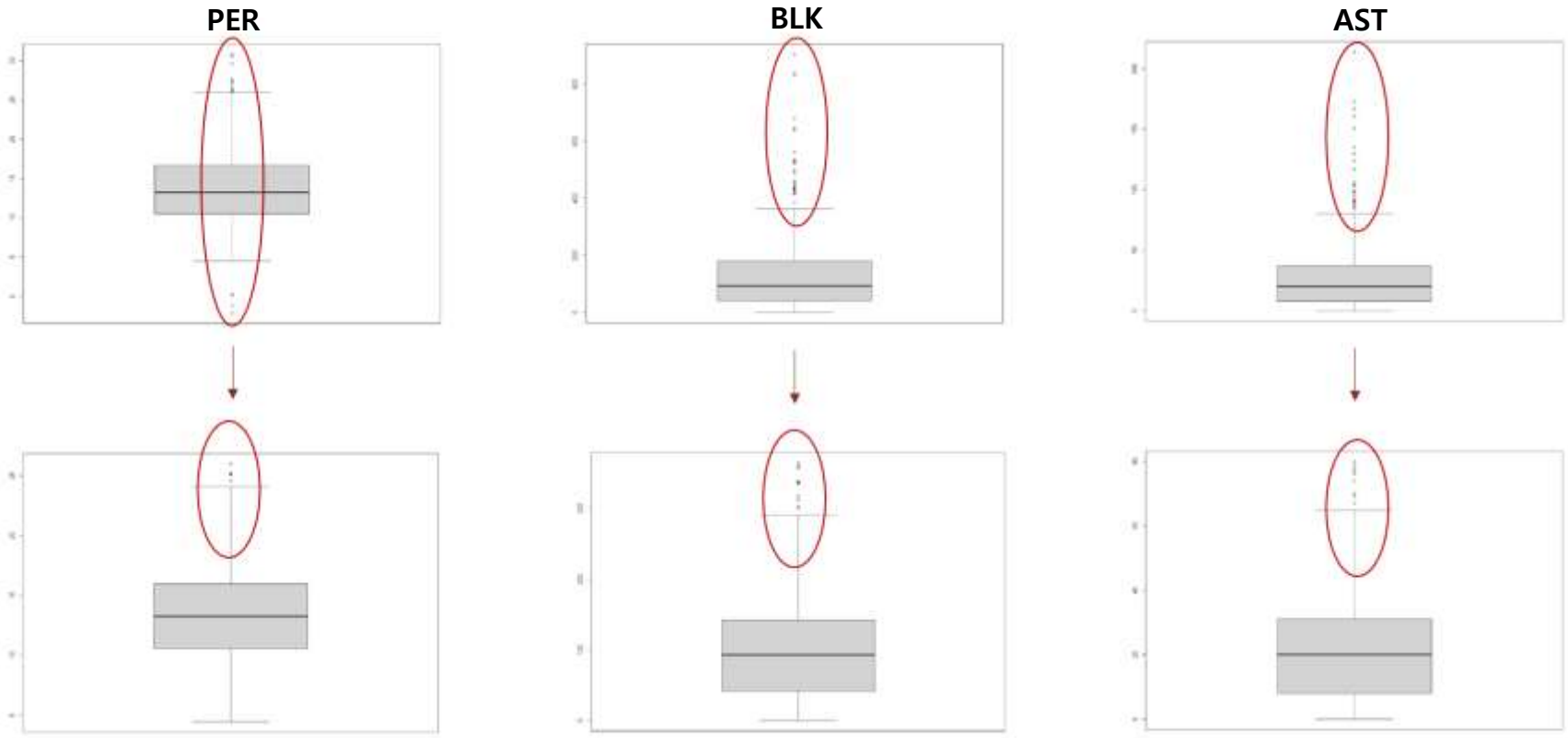
[평균대치법 활용]

*Salary : 연봉

003 데이터 전처리 – 이상치 제거 및 변화

- 나머지 독립변수들에도 평균대치법 적용하여 나타낸 boxplot 그림
- **결과 : 확연히 줄어듬을 알 수 있음.**

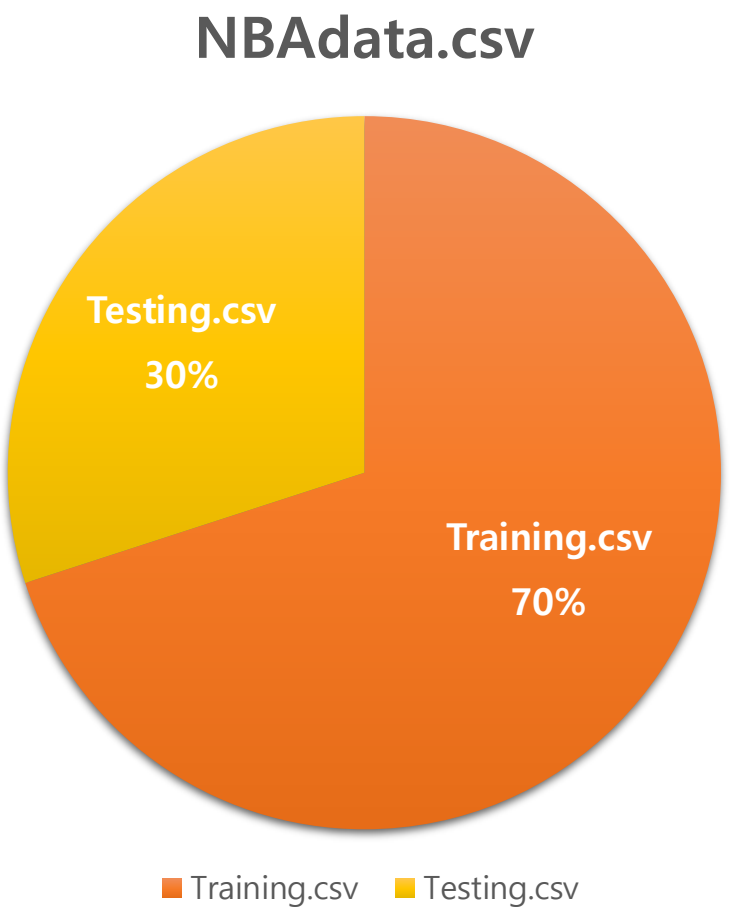
[대표적인 3가지 예]



***PER :** 선수효율지수 **BLK :** 블락(슈팅저지) 개수 **AST : 도움개수**

003 데이터 전처리 – 두 데이터로 분리

- Training data와 Testing data 분리



연구과정

1. 7:3의 비율로 Training 데이터와 Testing 데이터로 나눔

2. 70%의 Training 데이터로 다중회귀분석, 의사결정나무, 인경신공망 세 가지 예측모형을 training 함.

3. 30%의 Testing 데이터를 통해 RMSE값은 구함.

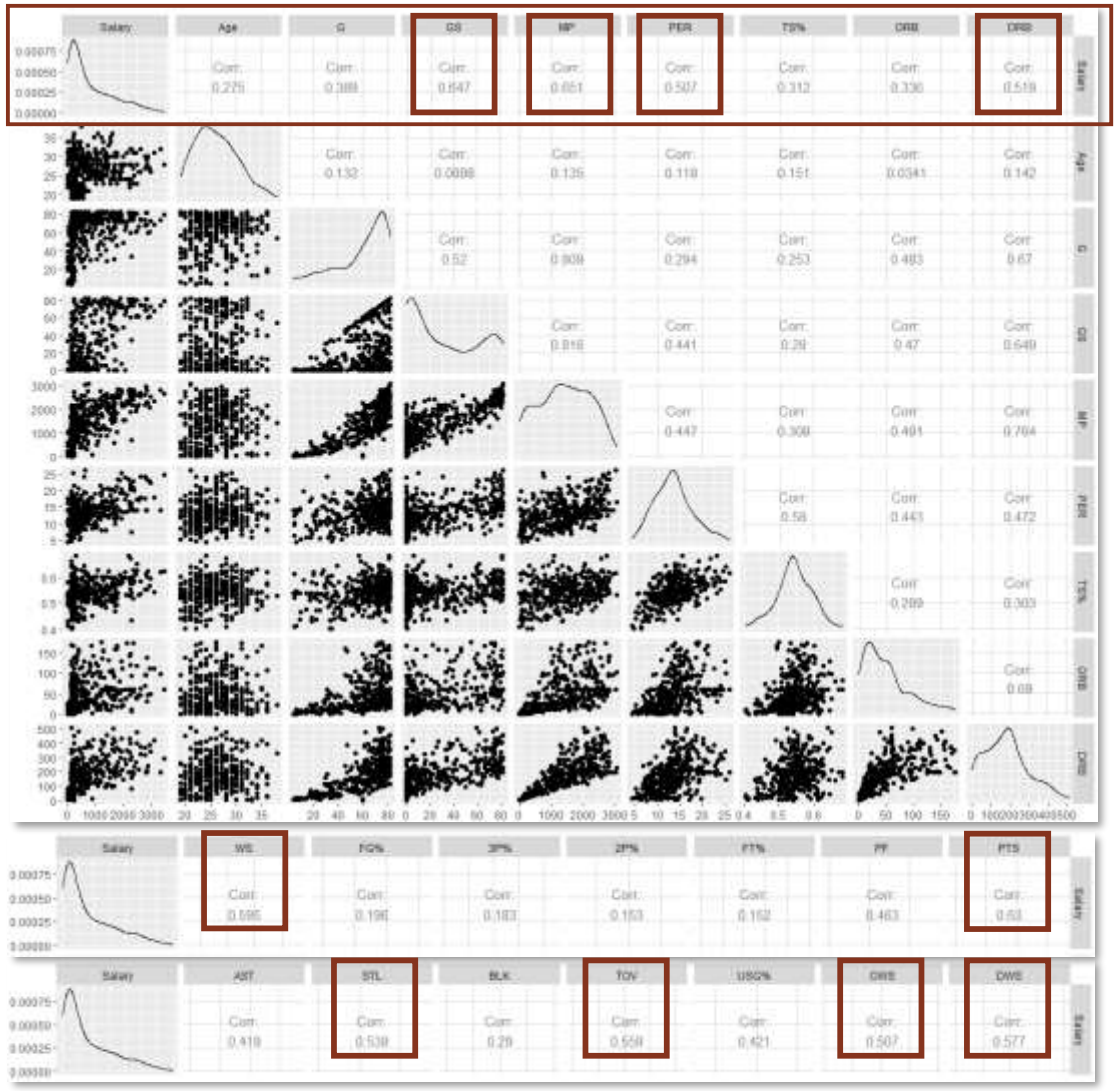
4. 세 가지 예측모형의 RMSE값을 비교함.

5. RMSE값이 가장 작은 예측모형을 적합하다고 판단함.

목적

과적합이 일어나지 않도록 하기 위해서
training data : testing data = 7:3의 비율로
랜덤하게 나눈 상태로 예측모델을 만듬.

004 데이터 시각화

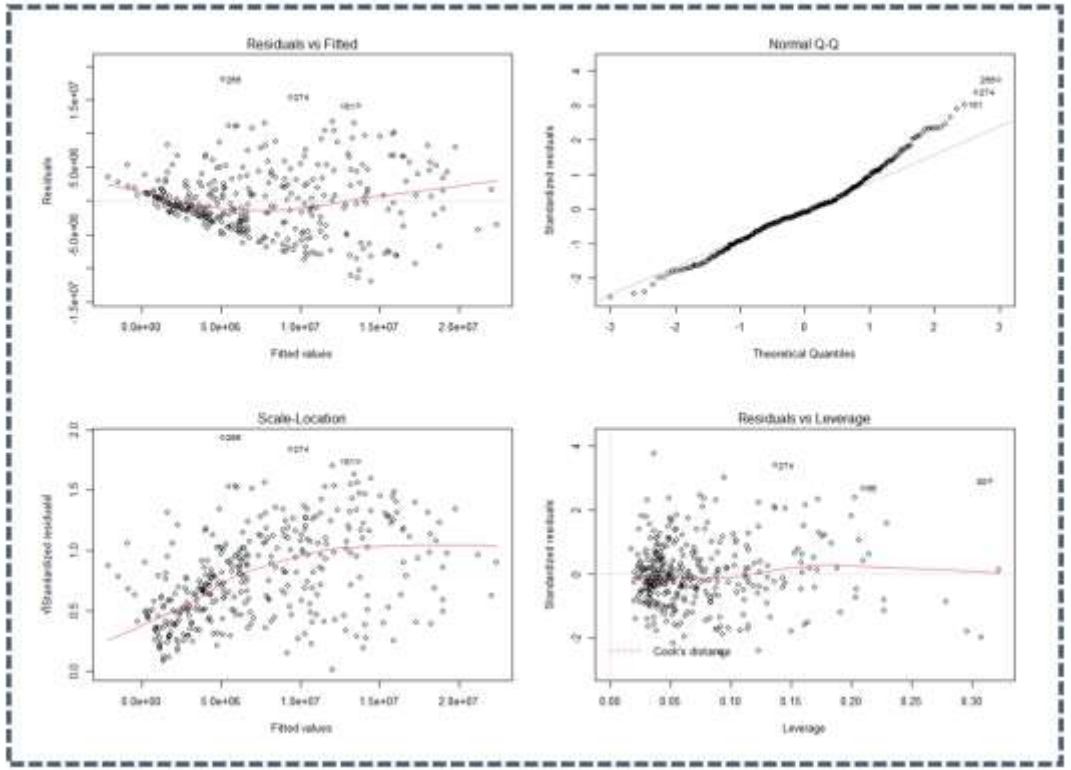


*주제 'NBA 선수들의 기록에 따른 연봉 예측'에 맞게 데이터를 산점도와 상관계수를 그래프로 나타낸 결과

< Salary와 높은 상관계수 추출한 결과 >

변수명	변수설명	수치
GS	선발출장횟수	0.647
MP	경기출장시간(분)	0.651
PER	선수효율지수	0.507
DRB	수비리바운드개수	0.519
PTS	득점	0.630
WS	팀 승리기여도	0.505
STL	스틸 개수	0.538
TOV	턴오버 개수	0.559
OWS	공격부분 팀 승리기여도	0.507
DWS	수비부분 팀 승리 기여도	0.577

< 회귀모형 적절성 평가를 위한 plot 작성 >



결과 설명

1. 잔차의 등분산성 검정- 특정패턴 없음
2. 잔차의 정규성 검정- 직선을 잘 따름
3. 잔차의 독립성 검정- 특정패턴 없음
4. 이상치 검출- 이상치가 검출됨

- Stepwise Regressions를 이용한 회귀변수 선택

분석내용

변수선택방법	Forward		Backward		Stepwise
선택된 설명변수	MP+PER+GS+PTS	=	MP+PER+GS+PTS	=	MP+PER+GS+PTS
AIC	3221.01		3221.01		3221.01

결과

Forward, Backward, Stepwise 세가지 방법의 **AIC** 값이 같고 추출해낸 **설명변수** 또한 같다.

따라서 **MP, PER, GS, PTS** 이 네 가지의 변수를 회귀변수로 채택

* **MP** : 경기출장시간(분) **PER** : 선수효율지수 **GS** : 선발출장횟수 **PTS** : 득점

- **선택된 회귀변수의 순위**

Parameter	AIC
MP	3223.0
PTS	3225.0 (+2)
GS	3230.6 (+7.6)
PER	3235.6 (+12.6)

AIC값이 작을수록 좋은 회귀변수이므로

다음과 같이

(1) MP
(2) PTS
(3) GS
(4) PER

순으로 좋은 회귀변수이다.

- **다중공선성**을 판단하기 위한 **분산팽창인자** 구하기

Parameter	vif	다중공선성 판단 (vif>10)
MP	5.783535	FALSE
PER	1.507711	FALSE
GS	3.355212	FALSE
PTS	4.022564	FALSE

Parameter	Estimate	Error	t value	Pr(>|t|)	
Intercept	-444.84857	121.02332	-3.675	0.00029	***
MP	0.18895	0.09459	1.998	0.04683	*
PER	38.87645	9.49708	4.094	0.0000573	***
GS	6.73976	1.97439	3.414	0.000747	***
PTS	0.40754	0.16687	2.442	0.015285	*

Adj R-squared : 0.556 **RMSE :** 613.3786

* **MP :** 경기출장시간(분) **PER :** 선수효율지수 **GS :** 선발출장횟수 **PTS :** 득점

회귀식 Y = -444.81857 + 0.08895X1 + 38.87645X2 + 6.73976X3 + 0.40754X4

결과 4가지 설명변수 모두 vif <10 이므로
다중공선성이 없음

006 의사결정나무

1. CP값 선정

순서	CP	nsplit	xerror
1	0.47028	0	1.01471
2	0.050437	1	0.63943
3	0.049602	2	0.60016
4	0.032171	3	0.56786
5	0.023901	4	0.54455
6	0.023608	5	0.54115
7	0.011048	6	0.53991
8	0.0099475	7	0.52449
9	0.0066424	8	0.53163
10	0.0063171	9	0.53709
11	0.0055759	10	0.54350
12	0.0054908	11	0.54561
13	0.0030422	12	0.54426
14	0.0028691	13	0.55818
15	0.0012719	14	0.55920

분석내용

상관분석을 통해 추출해낸 설명변수들을 사용하여

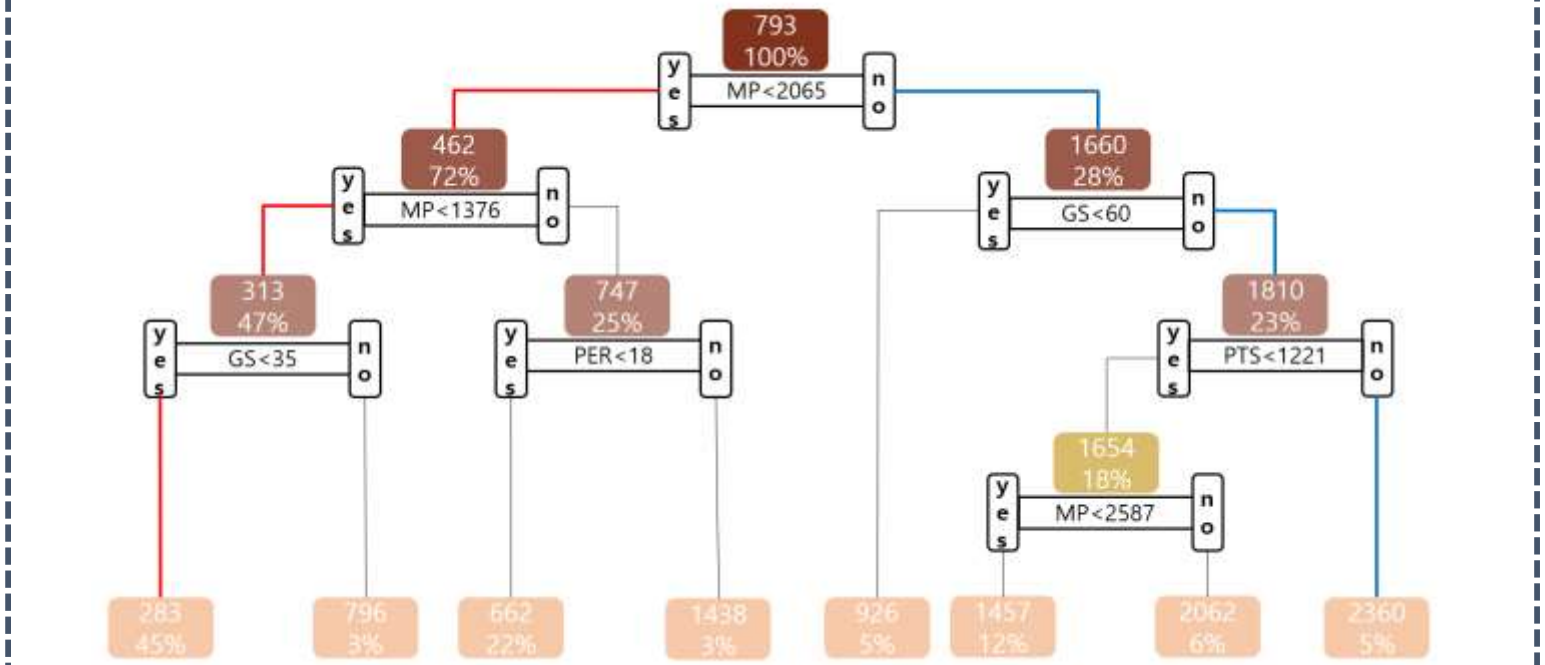
회귀나무 생성 xerror값이 가장 작은 지점의 cp값 설정

xerror값이 **가장 작은 지점인 nsplit=7**인

지점의 **Cp값 0.0099475 에서 가지치기**

2. 가지치기한 결과

- 전체 선수의 평균연봉 = 793 (만$) / MP, PER, GS, PTS에 따라 구분



결과

- 빨간색라인 최소값
MP<2065, MP<1376, GS<35
이와 같은 조건일 경우 **평균연봉이 283(만$)로 감소**

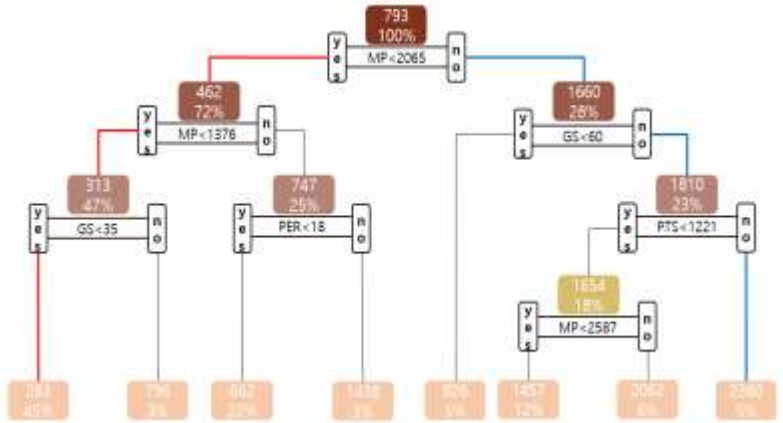
- 파란색라인 최대값
MP>=2065, GS>=60, PTS>=1221
이와 같은 조건일 경우 **평균연봉이 2360(만$)로 증가**

*** MP :** 경기출장시간(분) **PER :** 선수효율지수 **GS :** 선발출장횟수 **PTS :** 득점

3. 의사결정나무 예측모형 성능평가

+ Variable importance가 높을수록 더 의미있는 변수

변수명	Variable importance
MP	85,394,374
GS	65,674,081
PTS	49,441,270
PER	18,596,663



* **MP :** 경기출장시간(분) **PER :** 선수효율지수 **GS :** 선발출장횟수 **PTS :** 득점

사전정의

예측모형 설계시 의사결정나무 분석기법을 사용하여
전체 데이터의 70%인 Training데이터로 만든 예측모형과
나머지 30% Testing데이터를 예측해 보았을 때
실제값과 예측값에 대한 정확도를 RMSE를 통해 판단

결과

첫번째로 의미있는 변수
MP

두번째로 의미있는 변수
GS

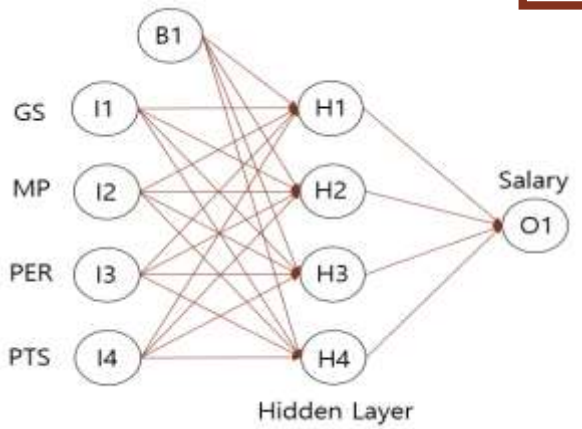
실제값과 예측값에 대한 평가
RMSE=611.9264

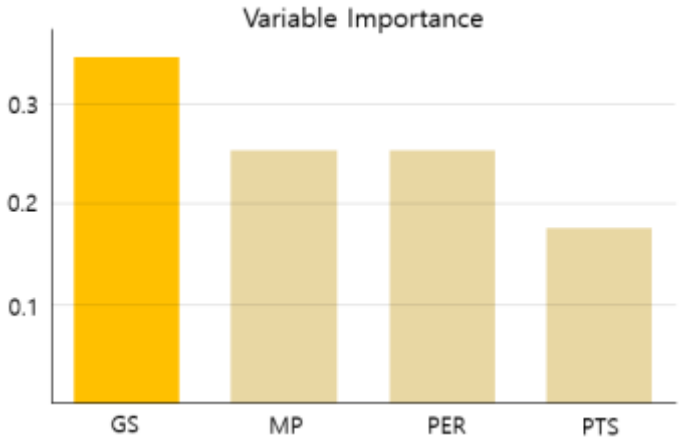
007 인공신경망

단층 퍼셉트론

- 인공신경망 단층 퍼셉트론 모형을 통한 예측
- ■ Hidden Node의 개수는 변수의 크기 4에 대해 [n/2~2n]으로 2~8로 설정하였다.
- ■ Training Set 70%, Test Set 30%로 설정한 후 노드 별, 반복수 별 추출된 RMSE값을 표를 만들었다.

반복수	2	3	4	5	6	7	8
1	0.1434	0.1462	0.1435	0.1467	0.1474	0.1439	0.1469
2	0.1452	0.1432	0.1415	0.1427	0.1420	0.1467	0.1455
3	0.1447	0.1458	0.1457	0.1428	0.1427	0.1413	0.1422
평균	0.1444	0.1450	**0.1435**	0.1440	0.1440	0.1439	0.1448





Hidden Node의 수를 4개로 했을 때의 단층 퍼셉트론 모형이다.
Variable Importance가 가장 큰 변수는 GS이다.

다층 퍼셉트론

■ Training Set 70%, Test Set 30%로 설정한 후 노드 별, 은닉층 개수별, 반복수 별로추출된 RMSE 값을 표로 만들었다.
■ 은닉층 개수에 따른 표이며,
[은닉층 수 4개, 각 Hidden Node의 개수가 7개] 일때, **RMSE값이 0.1170으로 가장 작음**.

은닉층1개

반복수	2	3	4	5	6	7	8
1	0.1434	0.1462	0.1435	0.1467	0.1474	0.1439	0.1469
2	0.1452	0.1432	0.1415	0.1427	0.1420	0.1467	0.1455
3	0.1447	0.1458	0.1457	0.1428	0.1427	0.1413	0.1422
평균	0.1444	0.1450	0.1435	0.1440	0.1440	0.1439	0.1448

은닉층2개

반복수	2*2	3*3	4*4	5*5	6*6	7*7	8*8
1	0.1454	0.1479	0.1454	0.1448	0.1257	0.1282	0.1336
2	0.1425	0.1413	0.1441	0.1130	0.1420	0.1376	0.1518
3	0.1455	0.1448	0.1449	0.1426	0.1411	0.1498	0.1711
평균	0.1444	0.1446	0.1448	0.1334	0.1362	0.1385	0.1521

은닉층3개

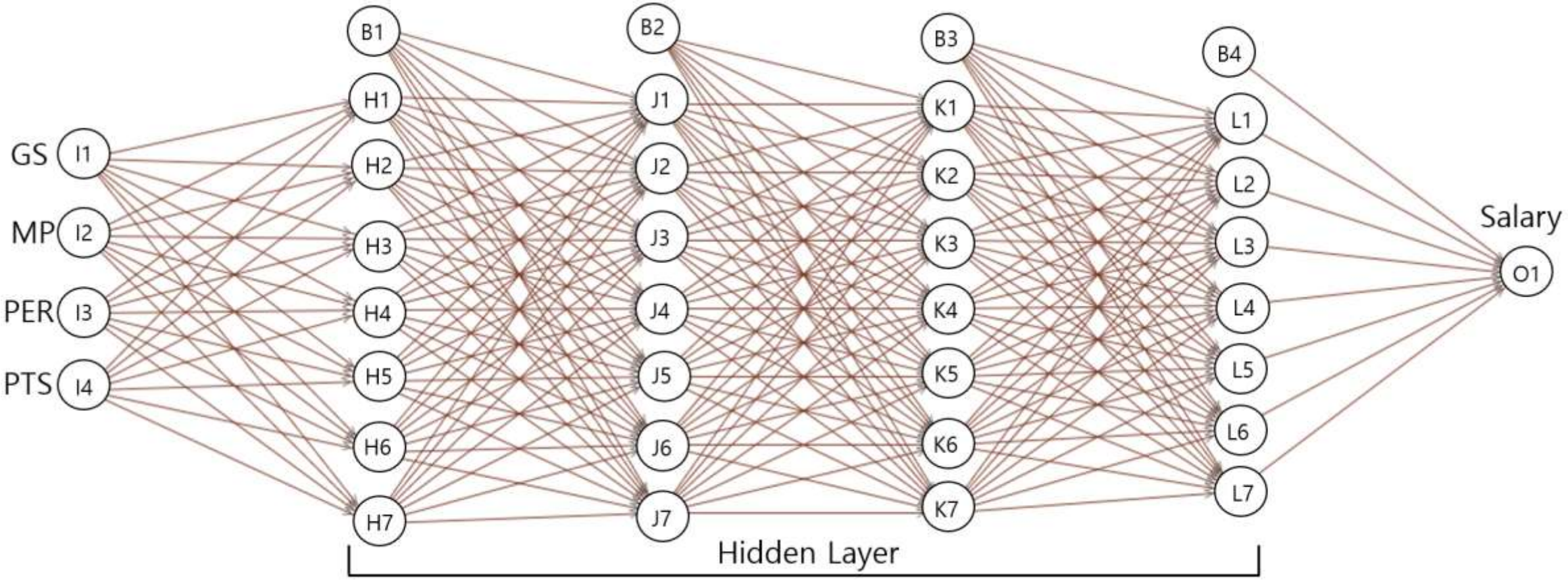
반복수	2*2*2	3*3*3	4*4*4	5*5*5	6*6*6	7*7*7	8*8*8
1	0.1439	0.1405	0.1434	0.1295	0.1374	0.1171	0.1438
2	0.1430	0.1456	0.1413	0.1457	0.1443	0.1199	0.1103
3	0.1454	0.1427	0.1438	0.1457	0.1204	0.1447	0.1719
평균	0.1441	0.1429	0.1428	0.1403	0.1340	0.1272	0.1420

은닉층4개

반복수	2*2*2*2	3*3*3*3	4*4*4*4	5*5*5*5	6*6*6*6	7*7*7*7	8*8*8*8
1	0.1448	0.12679	0.1370	0.1448	0.1451	0.1254	0.1462
2	0.1441	0.1422	0.1430	0.1363	0.1378	0.1187	0.1618
3	0.1449	0.1430	0.1451	0.1346	0.1437	0.1071	0.1475
평균	0.1446	0.1373	0.1417	0.1385	0.1422	**0.1170**	0.1518

007 인공신경망

[은닉층 수 4개, 각 Hidden Node 개수 7개] 다층 퍼셉트론 모형



* **MP :** 경기출장시간(분) **PER :** 선수효율지수 **GS :** 선발출장횟수 **PTS :** 득점

008 모형 비교 및 결론

사전정의

- 전처리된 총 3610개의 데이터를 통한 예측모델을 만들었다.
- 예측모델 3개의 RMSE값을 비교하여 가장 예측력이 좋은 모델을 선별한다.
- 데이터를 표준화하여 구한 인공신경망의 RMSE값은 0.1170 이다.
- 인공신경망의 예측모형과 실제값을 비교하여 구한 RMSE값은 371.7531 이다.

분석모델	회귀분석	의사결정나무	인공신경망
RMSE	611.9264	613.3786	**371.7531**

결론

- 비교결과 다층 인공신경망의 RMSE가 가장 작으므로 예측력이 가장 높다.

부록

```
#데이터 불러오기
setwd("C:/Users/82109/Desktop/datamining/NBA")
library(readxl)
sample <- read_excel("sample.xlsx")
View(sample)
#데이터 전처리 과정(이상치를 평균값으로 대체)
sample$Salary<-ifelse(sample$Salary > 28198470, 7958621, sample$Salary)
sample$Age<-ifelse(sample$Age>38,26.1,sample$Age)
sample$Age<-ifelse(sample$Age<14,26.1,sample$Age)
sample$G<-ifelse(sample$G>119.5,59.5,sample$G)
sample$G<-ifelse(sample$G<3.5,59.5,sample$G)
sample$GS<-ifelse(sample$GS>149.5,30.6,sample$GS)
sample$MP<-ifelse(sample$MP>4021.5,1417,sample$MP)
sample$PER<-ifelse(sample$PER>26, 13.91,sample$PER)
sample$PER<-ifelse(sample$PER<1.2,13.91,sample$PER)
sample$`TS%`<-ifelse(sample$`TS%`>0.684,0.5402,sample$`TS%`)
sample$`TS%`<-ifelse(sample$`TS%`<0.404,0.5402,sample$`TS%`)
sample$ORB<-ifelse(sample$ORB>176.5, 59.62,sample$ORB)
sample$DRB<-ifelse(sample$DRB>511.5,196.6,sample$DRB)
sample$AST<-ifelse(sample$AST>384.5, 134.2,sample$AST)
sample$STL<-ifelse(sample$STL>132.5,45.48,sample$STL)
sample$BLK<-ifelse(sample$BLK>80.5,28.04,sample$BLK)
sample$TOV<-ifelse(sample$TOV>215,78.49,sample$TOV)
sample$`USG%`<-ifelse(sample$`USG%`>32.8,18.97,sample$`USG%`)
sample$`USG%`<-ifelse(sample$`USG%`<4,18.97,sample$`USG%`)
sample$OWS<-ifelse(sample$OWS>6.1,1.654,sample$OWS)
sample$DWS<-ifelse(sample$DWS>4.35,1.455,sample$DWS)
sample$WS<-ifelse(sample$WS>9.15,3.11,sample$WS)
sample$`FG%`<-ifelse(sample$`FG%`>0.6265,0.4526,sample$`FG%`)
sample$`FG%`<-ifelse(sample$`FG%`<0.2705,0.4526,sample$`FG%`)
sample$`3P%`<-ifelse(sample$`3P%`>0.5385, 0.2897,sample$`3P%`)
sample$`3P%`<-ifelse(sample$`3P%`<0.1025, 0.2897,sample$`3P%`)
sample$`2P%`<-ifelse(sample$`2P%`>0.663, 0.4912,sample$`2P%`)
sample$`2P%`<-ifelse(sample$`2P%`<0.327, 0.4912,sample$`2P%`)
sample$`FT%`<-ifelse(sample$`FT%`>1.0345,0.7438,sample$`FT%`)
sample$`FT%`<-ifelse(sample$`FT%`<0.4945,0.7438,sample$`FT%`)
sample$PF<-ifelse(sample$PF>284,115.7,sample$PF)
sample$PTS<-ifelse(sample$PTS>1784.5, 628.9,sample$PTS)

#데이터 시각화
install.packages("GGally")
library(GGally)
ggpairs(sample[,1:9])
sample1<-sample[c(1,10,11,12,13,14,15,16)]
ggpairs(sample1[,1:8])
sample2<-sample[c(1,17,18,19,20,21,22,23)]
ggpairs(sample2[,1:8])
sample4<-sample[c(1,4,5,6,9,11,13,15,16,17,23)]
sample4<-as.data.frame(sample4)
```

```
#학습용데이터, 테스트용 데이터 나누기
install.packages("caret")
library(caret)
intrain<-createDataPartition(y=sample4$Salary,p=0.7,list=F)
training<-sample4[intrain,]
testing<-sample4[-intrain,]
#stepwise
model2<-lm(Salary~., data = training)
par(mfrow=c(2,2))
plot(model2)
model2_con<-lm(Salary~1,data=training)
model2_forward<-step(model2_con,scope=list(lower=model2_con,upper=model2),
direction="forward")
model2_forward
model2_backward<-step(model2,scope=list(lower=model2_con,upper=model2),
direction="backward")
model2_backward
model2_both<-step(model2_con,scope=list(lower=model2_con,upper=model2),
direction="both")
model2_both
summary(model2_backward)
summary(model2_forward)
summary(model2_both)
model2_both<-lm(Salary~MP+PER+GS+PTS,data=training)
model2_forward<-lm(Salary~PER+GS+DWS,data=training)
model2_backward<-lm(Salary~GS+PER+DWS+PTS,data=training)
summary(model2_backward)
summary(model2_forward)
summary(model2_both)
(car::vif(model2_forward))>10
(car::vif(model2_backward))>10
(car::vif(model2_both))>10
car::vif(model2_both)
model2_both<-lm(Salary~GS+PER+MP+DRB+DWS,data=training)
pre_salary_model2_both<-predict(model2_both,newdata=testing)
pre_salary_model2_both<-as.data.frame(pre_salary_model2_both)
pre_salary_model2_both
pre_salary_model2_both<-predict(model2_both,newdata=testing,interval = "confidence")
pre_salary_model2_both<-as.data.frame(pre_salary_model2_both)
pre_salary_model2_both
model4<-cbind(pre_salary_model2_both,testing$Salary)
model4
install.packages("Metrics")
library(Metrics)
rmse(testing$Salary,pre_salary_model2_both$fit)
```

```
#인공신경망
install.packages("neuralnet")
library(neuralnet)
sample5<-sample[c(1,4,5,6,23)]
normalize<-function(x){return((x-min(x))/(max(x)-min(x)))}
sample6<-as.data.frame(lapply(sample5,normalize))
intrain<-createDataPartition(y=sample6$Salary,p=0.7,list=F)
train<-sample6[intrain,]
test<-sample6[-intrain,]
Salary_model<-neuralnet(Salary~GS+MP+PER+PTS,data=train,rep=3,hidden=c(7,7,7,7))
Salary_result<-compute(Salary_model,test[,2:5])
predicted_Salary<-Salary_result$net.result
rmse(test$Salary,Salary_result$net.result)
plot(Salary_model)
Salary_result$net.result
install.packages("NeuralNetTools")
library(NeuralNetTools)
garson(Salary_model)

#의사결정나무
install.packages("rpart.plot")
library(rpart)
library(rpart.plot)
my_control<-rpart.control(xval=10,cp=0,minisplit=nrow(training)*0.05)
fit_salary<-rpart(Salary~MP+PER+GS+PTS,data=training,method='anova',control=my_control)
rpart.plot(fit_salary)
printcp(fit_salary)
plotcp(fit_salary)
which.min(fit_salary$cptable[,'xerror'])
fit_prune_salary<-prune(fit_salary,cp=fit_salary$cptable[which.min(fit_salary$cptable[,'xerror']),
"CP"])
fit_prune_salary$variable.importance

    rpart.plot(fit_prune_salary)
    pre_prune_salary<-predict(fit_prune_salary,newdata=testing,type="matrix")
    pre_prune_salary
    pre_prune_salary<-as.data.frame(pre_prune_salary)
    pre_prune_salary
    as.data.frame(testing$Salary)
    model5<-cbind(pre_prune_salary,testing$Salary)
    model5
    install.packages("Metrics")
    library(Metrics)
    rmse(testing$Salary,pre_prune_salary$pre_prune_salary)
```

모두들 고생하셨습니다.

박장대소

장유성

박태현

박종혁