Neural & Decision Tree
보고서

2011100038 박병진
2011100074 김혁준
2012100005 강나루

Contents

B. 신경망 분석

D. 모형비교

A. 전처리

C. 결정나무

Bank term deposit

Bank Direct Marketing

√ 해당 데이터는 고객들의 인적 사항 및 마케팅 전략 과정의 여러 정보, **정기예금상품 신청여부**에 대한 자료이다.

Neural Network / Decision Tree/ Regression Model 간의 모델 적합도 비교

Transformation

① 변수 변환

√ 회귀분석의 자료와 동일하므로 description 생략,
변수변환과정 간단히 설명

변수제거	Day	대체로 고른 분포와 month만으로 판단 가능하다 생각되어 **제거**
	Pdays	첫 고객이 전체 81%, previous와 유사하다 생각되어 **제거**
결측값 대체	age	**Median** 대체
	Marital	30세 ↑ : married /30세 ↓ : single
	기타 결측치	Unknown을 **하나의 범주**로 둠.
변수 변형	Balance/ Campaign	비대칭적 자료라 **로그변환**
반응 변수	중요한 정보라서 **결측치 제거**함.	

Prediction

② 예측 모델

√ 회귀모형의 Error rate(좌-7:3 우-6:4)

```
> miss.err
[1] 0.09885315
```

```
> miss.err
[1] 0.09772353
```

√ 신경 연결망의 Error rate(좌-7:3 우-6:4)

```
> miss.err
[1] 0.1069342519
```

```
> miss.err
[1] 0.1017062937
```

√ 결정나무모형의 Error rate (좌-7:3 우-6:4)

```
> miss.err
[1] 0.2421238158
```

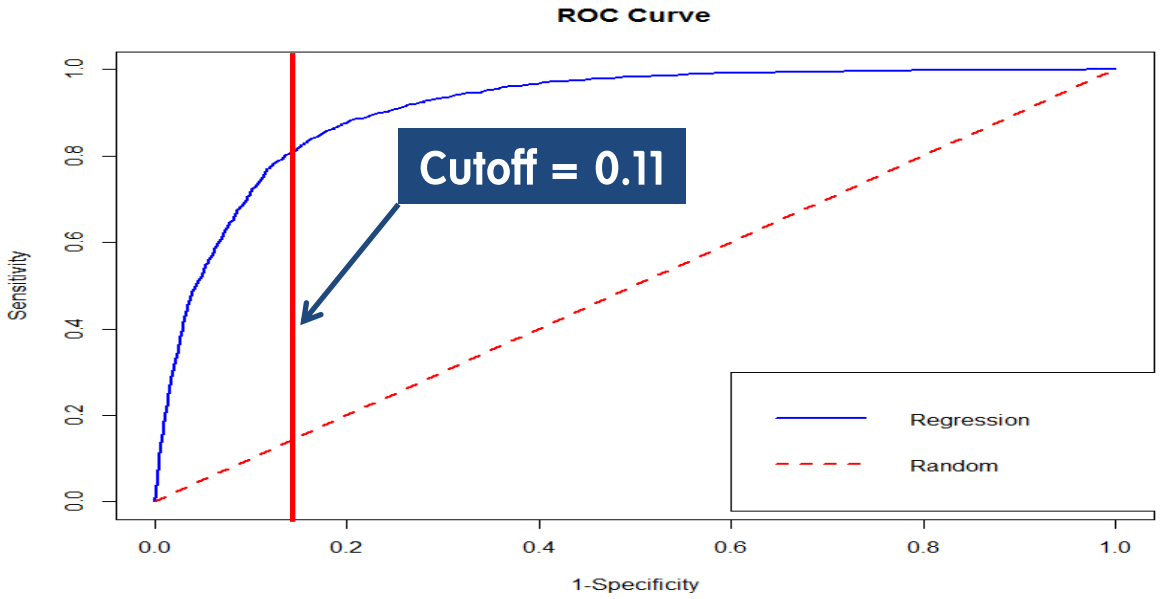
```
> miss.err
[1] 0.2395833333
```

Error rate을 통해서 Validation partition을 6:4로 결정.

- 회귀모형에서는 7:3이 error rate가 더 적으나 차이가 크지 않다고 판단.
- 신경 연결망과 결정나무모형의 경우 6:4가 error rate가 더 낮다.
- 최종적으로 6:4가 적합하다고 판단

Cutoff

③ Cutoff 설정



√ **회귀모형의 ROC를 참고하여 Cutoff설정.**

① 원 자료에서 y값이 NO가 많은 점을 고려하여 민감도를 높이는 방향으로 Cutoff설정
(민감도 : 0.8532051, 특이도 : 0.8245718)

② Promotion인 점을 감안해서 Cutoff를 낮게 설정.

=> 0.11로 Cutoff 결정

Preparation

신경망 분석을 위한 사전 작업

〉유의미한 변수 선택

– 모든 변수를 사용하기에는 연산속도가 문제가 됨
– 유의미한 변수 설정을 위해서 회귀분석에서 사용되었던 AIC를 참고하여 변수설정.

=〉 변수 default, previous, age를 제외

```
> fit2$anova
        Step Df    Deviance Resid. Df Resid. Dev      AIC
1         NA NA          NA     31814   15199.53 15281.53
2  - default  1 0.04871219     31815   15199.57 15279.57
3 - previous  1 0.75218979     31816   15200.33 15278.33
4      - age  1 0.90794475     31817   15201.23 15277.23
```

〉가변수 생성

– 범주형 자료에 대한 가변수 생성
– 각 가변수 가장 마지막 가변수를 기준으로 둠

```
#Data handling
dvar = c(1,2,3,5,6,7,8,11) #find categorical variables
# 15,18,22,24,26,29,41,45
bank2 = dummy(x=bank[,dvar])
bank2 = bank2[,-c(16,19,23,25,27,30,42,46)] # delete redundant dummy variables
bank2 = cbind(bank[,-dvar], bank2) # combine them
```

신경망 분석을 위한 사전 작업

› 표준화 작업

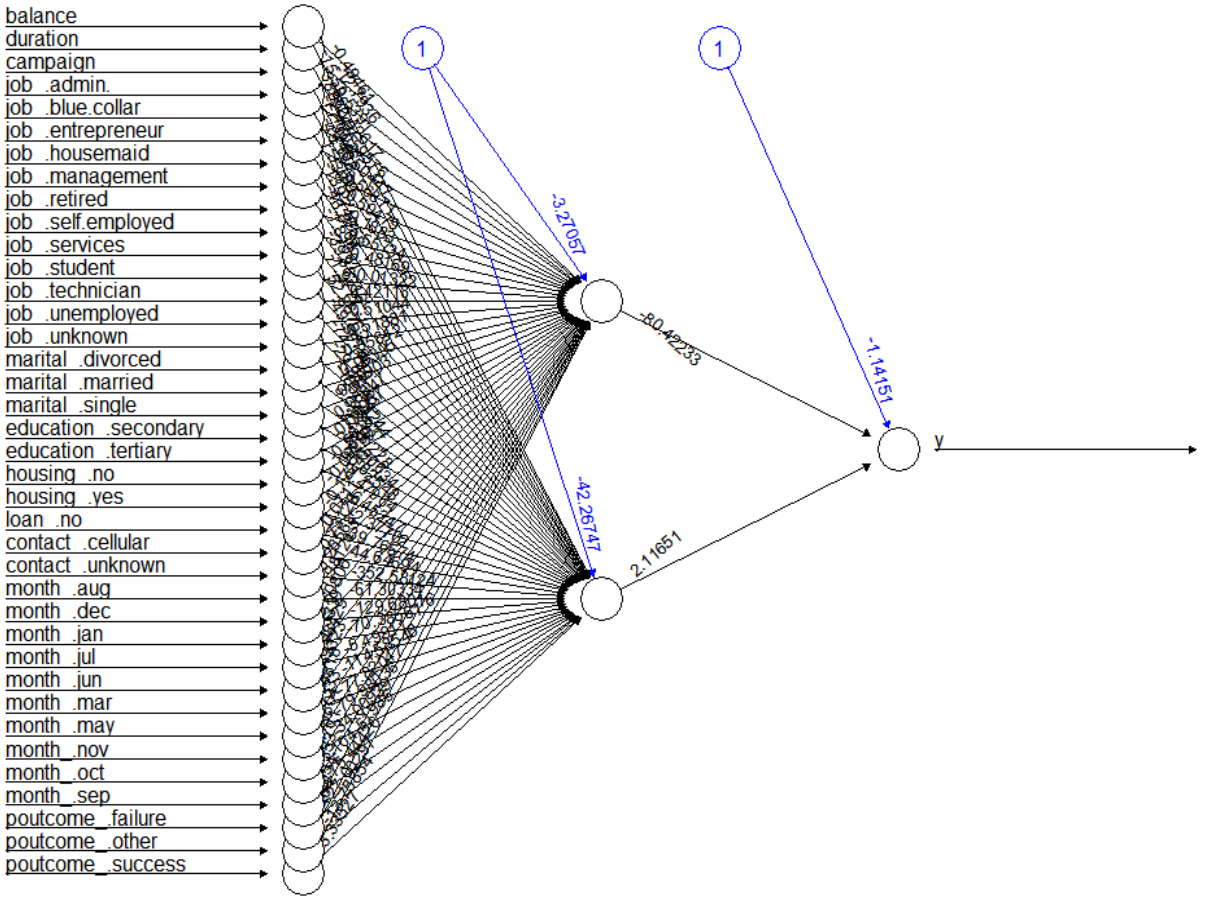
- 연속형 자료에 대해서
  입력 변수값이 0과 1사이에 존재하도록 조정
- $$\text{조정변수값} = \frac{\text{실제값} - \text{최솟값}}{\text{최댓값} - \text{최솟값}}$$

```
for(i in 1: ncol(bank)) if(!is.numeric(bank[,i])) bank[,i] = as.numeric(bank[,i])
max1 = apply(bank, 2, max)
min1 = apply(bank, 2, min)
gdat = scale(bank, center = min1, scale = max1 - min1) #Standaization
gdat = as.data.frame(gdat)
```

Analysis

Neural Network

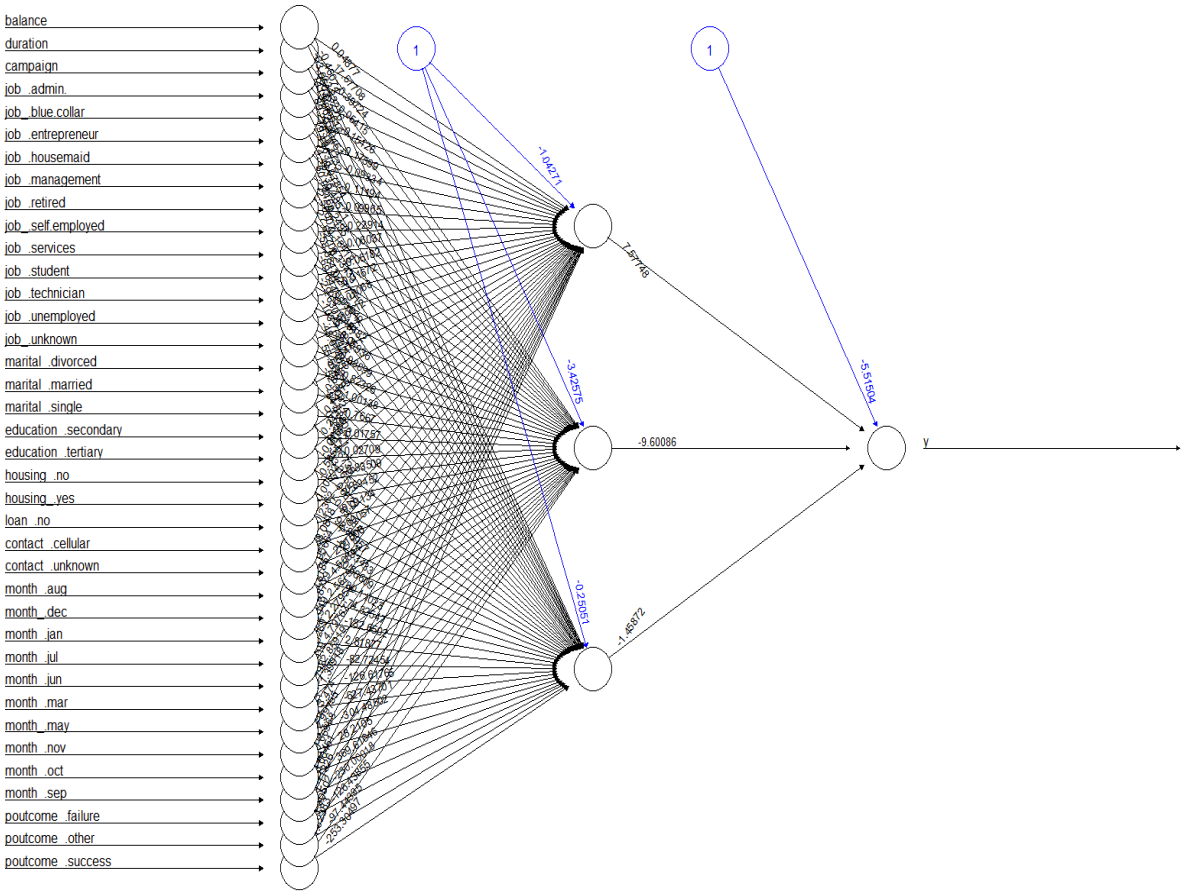
√ Hidden layer 1개, Hidden node 2개로 설정



Analysis

Neural Network

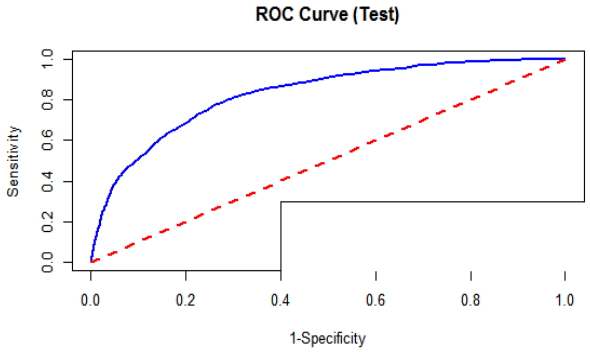
√ Hidden layer 1개, Hidden node 3개로 설정



Analysis

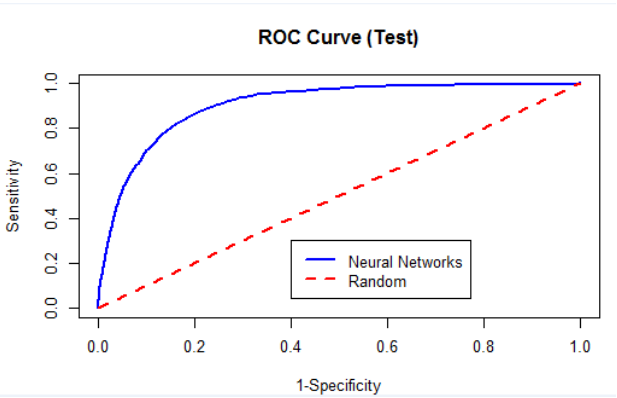
Neural Network

√ 모형 Selection



> Node 2개

```
> performance(pred, "auc")@y.values #AUC
[[1]]
[1] 0.8269614012
```



> Node 3개

```
> performance(pred, "auc")@y.values #AUC
[[1]]
[1] 0.9048488544
```

AUC를 참고해서 노드3개로 결정

- Layer 2개는 연산시간 문제로 제외하고 Layer 1개에서 고려
- 복잡성을 줄이면서 설명력을 높이는 정도가 Layer 1개이고 노드 3개일 때로 판단

Analysis

Neural Network

√ 모델 예측(노드 layer=1, 노드=3 / Cutoff=0.11)

```
> print(ctable) # classification table
         Predicted
Actual       0     1
     0   12794  2994
     1     308  1779
```

- **Missclassification Rate :** 0.1847272727
- **Prediction accuracy :** 0.8152727273
- **Sensitivity :** 0.8524197413
- **Specificity :** 0.810362300

Algorithm

Algorithm Selection

- CHAID는 범주형 자료에 적합한데 해당 데이터는 범주형 자료와 연속형 자료 혼합이라 제외
- CTREE 알고리즘의 경우 permutation test로 변수를 선택하지만 해석이 복잡해서 제외
- CART 알고리즘은 연속형, 범주형 자료 모두 사용 가능하고 해석도 용이해서 CART 알고리즘을 채택
- 패키지 중 rpart가 tree보다 선호됨

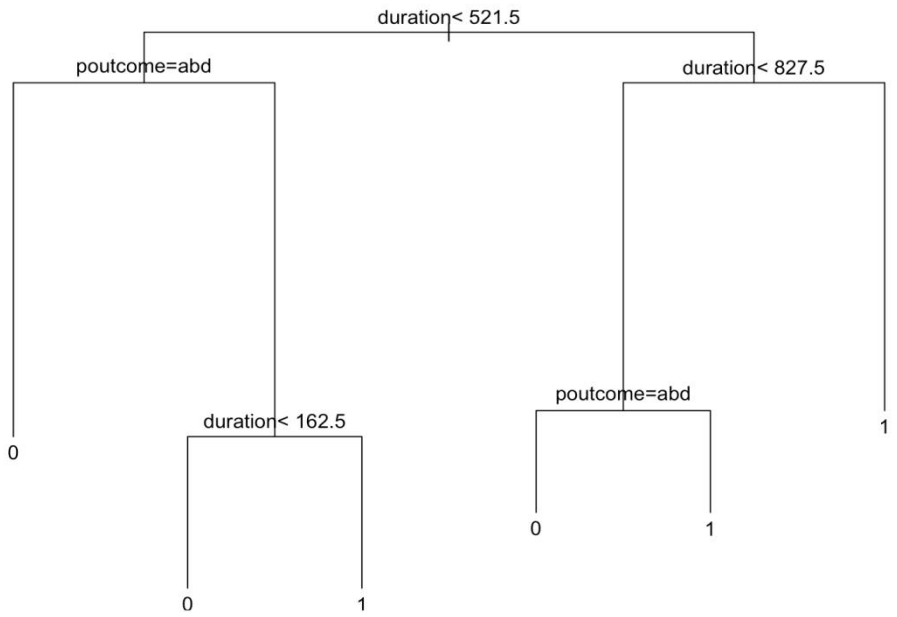
=>**CART 알고리즘**을 채택, **rpart 패키지** 사용

Analysis

Decision Tree

√ Default값으로 실행 결과

```
set.seed(1)
fit = rpart(y ~ job+marital+education+balance+housing+loan+contact+month+duration+
              campaign+poutcome, data=bank, method="class") #cp=0.01 (default),
#NOTE: parms=list(split='gini') is the default, parms=list(split='information') is optional
```

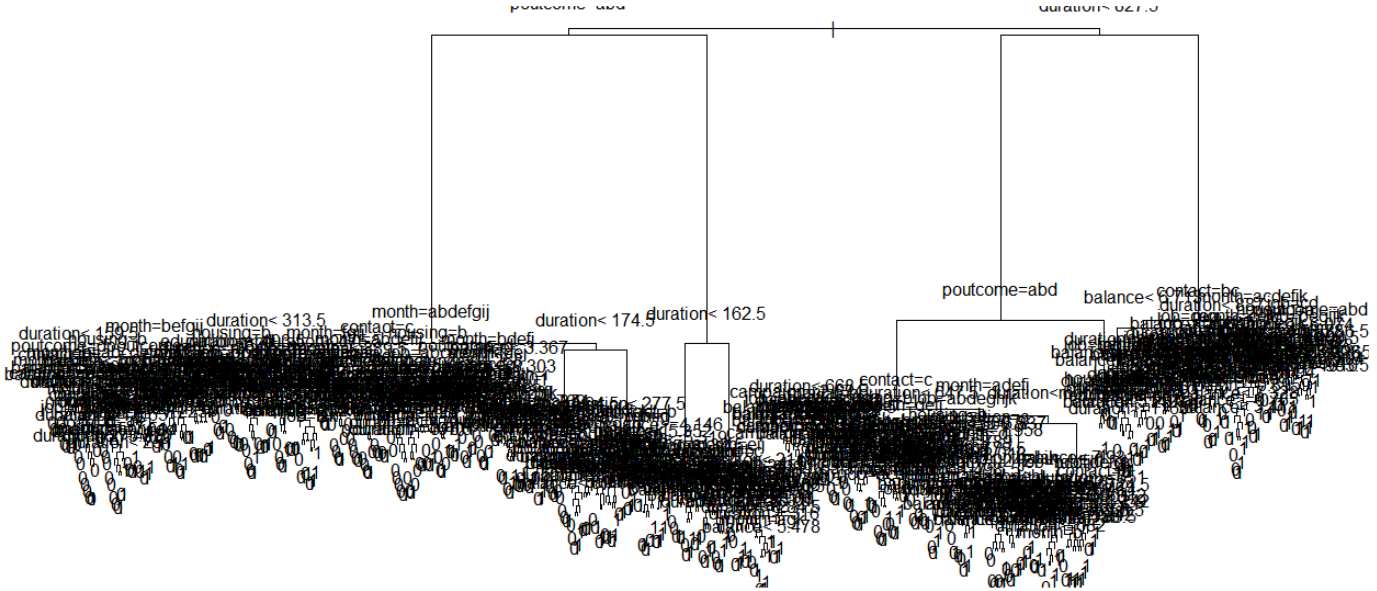


Analysis

Pruning

√ **Maximal tree**

– 가지치기를 하기 위해서 cp값을 0으로 주어 Maximal tree 생성



Analysis

Pruning

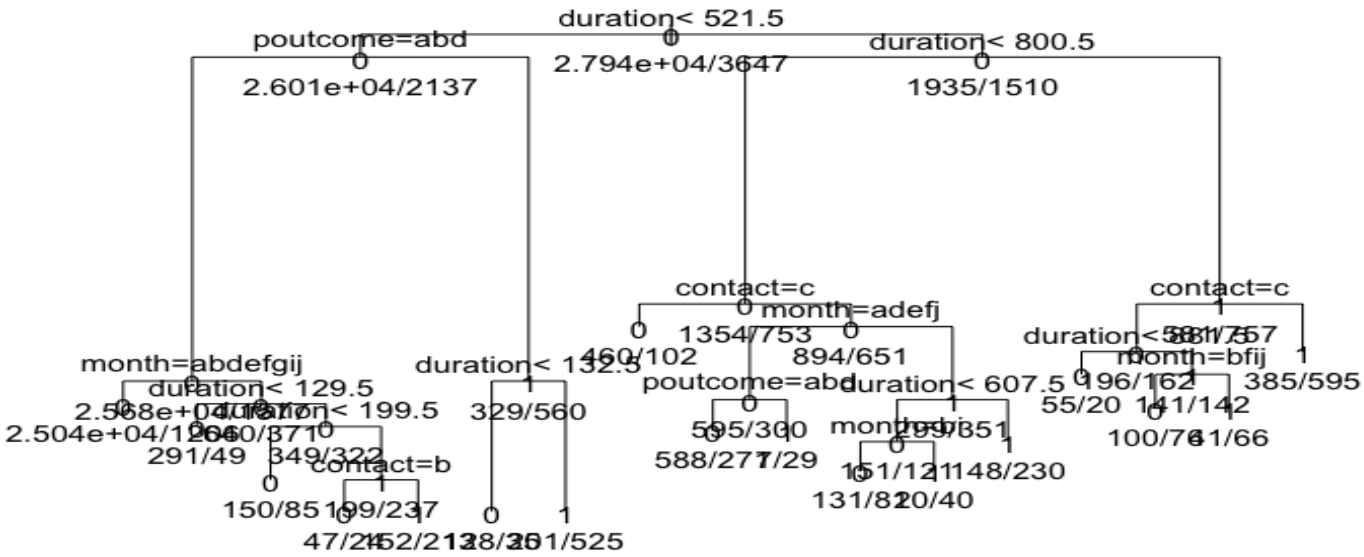
√ **Pruned tree**

– Cost-complexity를 최소화하는 tree를 찾음

– Cp=0.001418037인 것을 확인

```
### Prune the tree

tmp = printcp(fit2)
cp.tmp = tmp[which.min(tmp[,"xerror"]),"CP"]
fit.pruned = prune(fit2, cp=cp.tmp)
plot(fit.pruned, margin = 0.1);text(fit.pruned, use.n=TRUE)
```



Analysis

Interpretation

√ 유의미한 변수는 duration, poutcome, month, contact, job, balance인 것을 확인

```
Variable importance
duration poutcome    month  contact      job  balance
      52       29       14        2        1        1
```

√ node2와 node3에 대한 summary

```
Node number 2: 40236 observations,    complexity param=0.0380034
  predicted class=0  expected loss=0.07719455  P(node) =0.8899998
    class counts: 37130  3106
   probabilities: 0.923 0.077
  left son=4 (38939 obs) right son=5 (1297 obs)
  Primary splits:
      poutcome splits as  LLRL, improve=791.6783, (0 missing)
      month    splits as  LLRLLLLRLLRR, improve=509.8706, (0 missing)
      duration < 205.5    to the left,  improve=228.0395, (0 missing)
      housing  splits as  RL, improve=169.0047, (0 missing)
      contact  splits as  RRL, improve=150.4805, (0 missing)
```

```
Node number 3: 4973 observations,    complexity param=0.0380034
  predicted class=0  expected loss=0.4389704  P(node) =0.1100002
    class counts:  2790  2183
   probabilities: 0.561 0.439
  left son=6 (3191 obs) right son=7 (1782 obs)
  Primary splits:
      duration < 827.5    to the left,  improve=112.62690, (0 missing)
      poutcome splits as  LLRL, improve= 61.04304, (0 missing)
      contact  splits as  RRL, improve= 58.36740, (0 missing)
      month    splits as  LRRLLLLRLLRR, improve= 32.23210, (0 missing)
      marital  splits as  RLR, improve= 26.05363, (0 missing)
  Surrogate splits:
      balance  < -7.760171 to the right, agree=0.642, adj=0.001, (0 split)
      campaign < 3.156774  to the left,  agree=0.642, adj=0.001, (0 split)
```

Analysis

Decision Tree(cutoff=0.11)

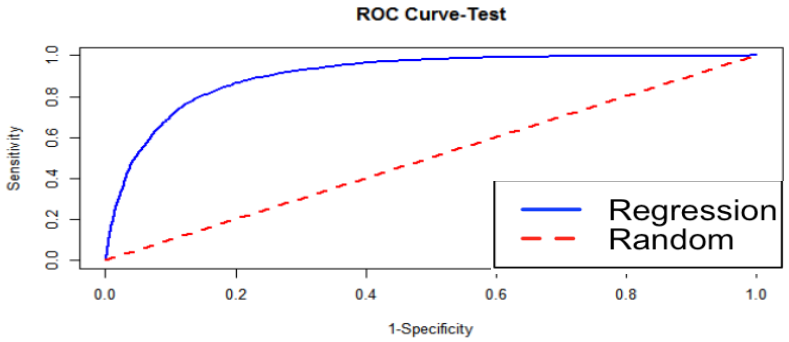
n table

Actual	Predicted 0	1
0	14324	1660
1	714	1446

- **Missclassification Rate :** 0.1311243337
- **Prediction accuracy :** 0.8688756663
- **Sensitivity :** 0.6696918132
- **Specificity :** 0.8952655311

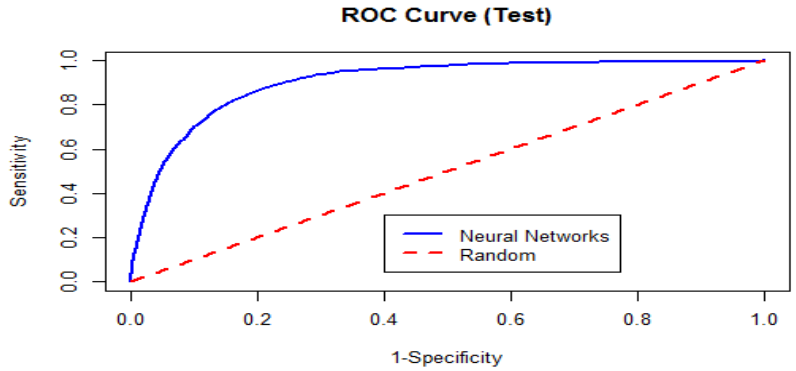
Comparison

Model Comparison



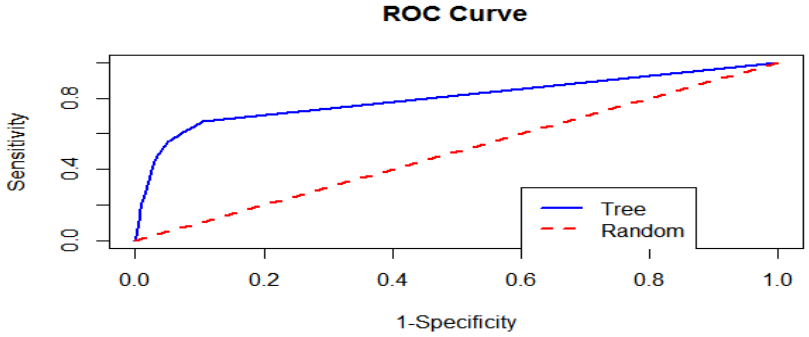
Logistic Regression

AUC = 0.906



Neural Network

AUC = 0.905



Decision Tree(rpart)

AUC = 0.796

Model Comparison

√ **Prediction**

– 앞의 ROC Curve와 AUC를 비교하면 regression model과 neural network가 비슷하고 decision tree가 낮게 나오는 것을 확인할 수 있음

– 이는 regression model과 neural network가 prediction을 잘 하고 있다는 것을 의미

– neural network의 node layer를 추가할 시 neural network의 설명력이 더 좋아질 것으로 예상되어 예측력은

Neural > Regression model > Decision Tree

Model Comparison

√ **Interpretation**

– Regression model의 경우 로지스틱 회귀를 사용하여 coefficient로 설명이 가능하지만 로짓으로 인해 해석의 불편함

– Neural network의 경우 hidden layer와 여러 함수들의 결합으로 인해 해석이 거의 불가능

– Decision tree의 경우 변수의 cutoff에 따른 군집을 나눠줌으로 인해서 어떤 변수가 중요하게 작용하는지 등 직관적인 해석이 용이함

– 따라서, 세 모델의 해석가능성은

Decision Tree > Regression model > Neural

Comparison

Model Comparison

특징	<-Interpretable		Predictable->
	Logistic Regression	Decision Tree	Neural Network
장점	-binary가 많은 실제 데이터 분석에 유용 -x는 정규성 가정을 필요x -타변수 영향을 고정한 상태에서 특정 변수의 유의여부 파악 용이	-이해 및 해석이 쉬움 -상호작용 발견, 결측치 처리 용이 -새로운 데이터에 대한 분류 및 예측이 쉽고 빠름	-비선형 형태의 복잡한 구조의 분석에 유리 -자료의 잡음에 크게 영향을 받지 않음 -정성적 변수의 효과적이고 신속한 처리
단점	-다중공선성 가능성 -특정 항목의 빈도가 작을 경우에는 분류정확도 편향 가능성 존재	-단순성과 분리점의 경직성으로 예측력이 떨어짐 -데이터 의존도가 커 구조 불안정	-과정의 복잡성으로 설명력 떨어짐 -잘못된 입력정보에 둔감