Model Comparison 보고서

2011100038 박병진
2011100074 김혁준
2012100005 강나루

Contents

B. 앙상블

D. 모형비교

A. 전처리

C. KNN/LDA

Bank term deposit

Bank Direct Marketing

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

기타 다양한 모델을 활용한 모델 적합도 비교

Transformation

√ 회귀분석의 자료와 동일하므로 description 생략,
변수변환과정 간단히 설명

√ Training set과 Test set Partition을 6:4로 설정

√ cutoff 0.11로 설정

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.	

Bagging

앙상블 기법(Bagging)을 위한 전처리

〉유의미한 변수 선택

– 로지스틱 회귀분석을 할 때 사용하였던 stepwise 기법을 통해서 AIC 기준으로 변수를 선택하였다.
– default, previous, age 제외

〉의사결정나무 모형설정

– Pruning을 하지 않은 최대 가능 모형으로 할 때 bagging의 적합도가 높아지기 때문에 cp=0으로 설정

〉Bagging sampling 횟수 설정

– Sampling 횟수를 예측 정확도에 따라 비교 –〉 100회로 결정

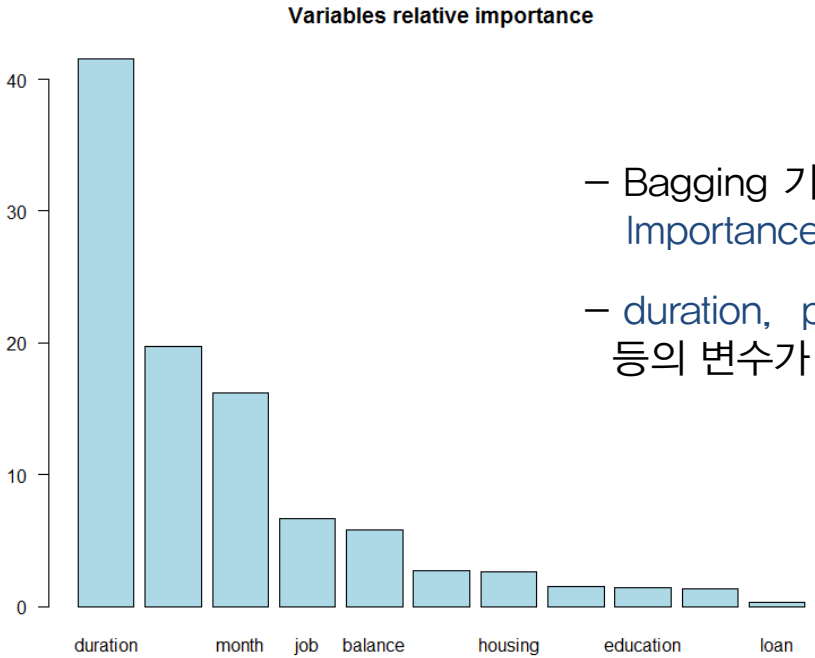
50회 :
```
> pred.acc = 1 - miss.err; pred.acc #Prediction Accuracy
[1] 0.9071203
```

100회 :
```
> pred.acc = 1 - miss.err; pred.acc #Prediction Accuracy
[1] 0.908735
```

Bagging

앙상블 기법(Bagging)

> 변수 중요도



- Bagging 기법을 통해서 적합시킨 후 Importance plot 도출
- duration, poutome, month, job 등의 변수가 중요한 변수인 것을 확인.

```
> print(fit$importance)
   balance   campaign    contact   duration  education    housing        job       loan    marital
 5.8600520  1.5606852  2.6861734 41.5307367  1.4162072  2.6629216  6.6387059  0.3321133  1.3725616
     month   poutcome
16.1941390 19.7457042
```

Bagging

앙상블 기법(Bagging)

> 적합 결과

```
> head(fit$prob,10)
       [,1] [,2]
 [1,] 0.88 0.12
 [2,] 0.92 0.08
 [3,] 0.92 0.08
 [4,] 0.93 0.07
 [5,] 0.89 0.11
 [6,] 0.93 0.07
 [7,] 0.89 0.11
 [8,] 0.92 0.08
 [9,] 0.96 0.04
[10,] 0.96 0.04
```

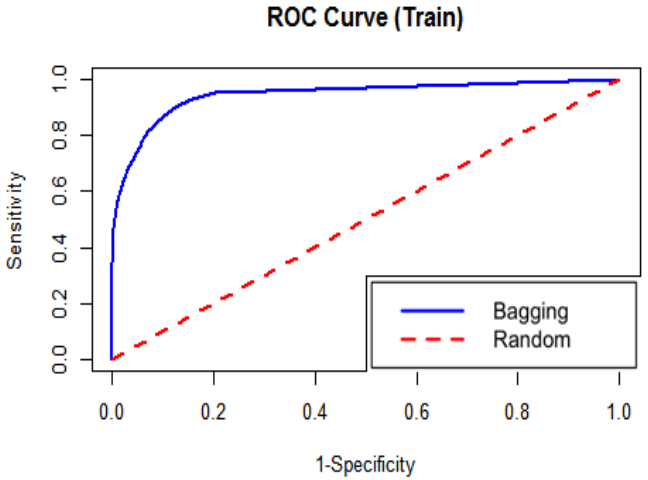
- 왼쪽은 Bagging 적합값에서 prob 변수를 뽑은 것으로 각 집단별 Y의 비율에 대한 10개의 자료이다.
- 아래는 cutoff 0.11로 계산한 오분류률, 정확도와 민감도와 특이도를 보여주고 있다.

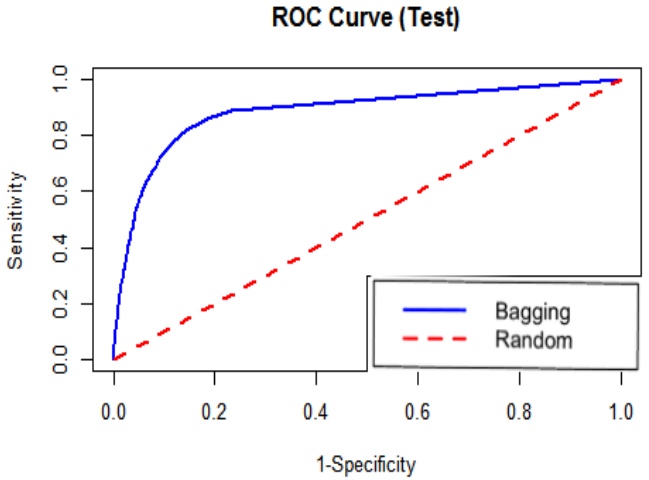
```
> ### Errors
>
> miss.err = 1-sum(diag(ctable))/sum(ctable); miss.err # Misclassification Rate
[1] 0.1203916
> pred.acc = 1 - miss.err; pred.acc #Prediction Accuracy
[1] 0.8796084
> diag(ctable)[2]/apply(ctable, 1, sum)[2] # Sensitivity
        1
0.7426929
> diag(ctable)[1]/apply(ctable, 1, sum)[1] # Specificity
        0
0.8977071
```

Bagging

앙상블 기법(Bagging)

› 적합 결과






```
> performance(pred, "auc")@y.values #AUC
[[1]]
[1] 0.9432583
```

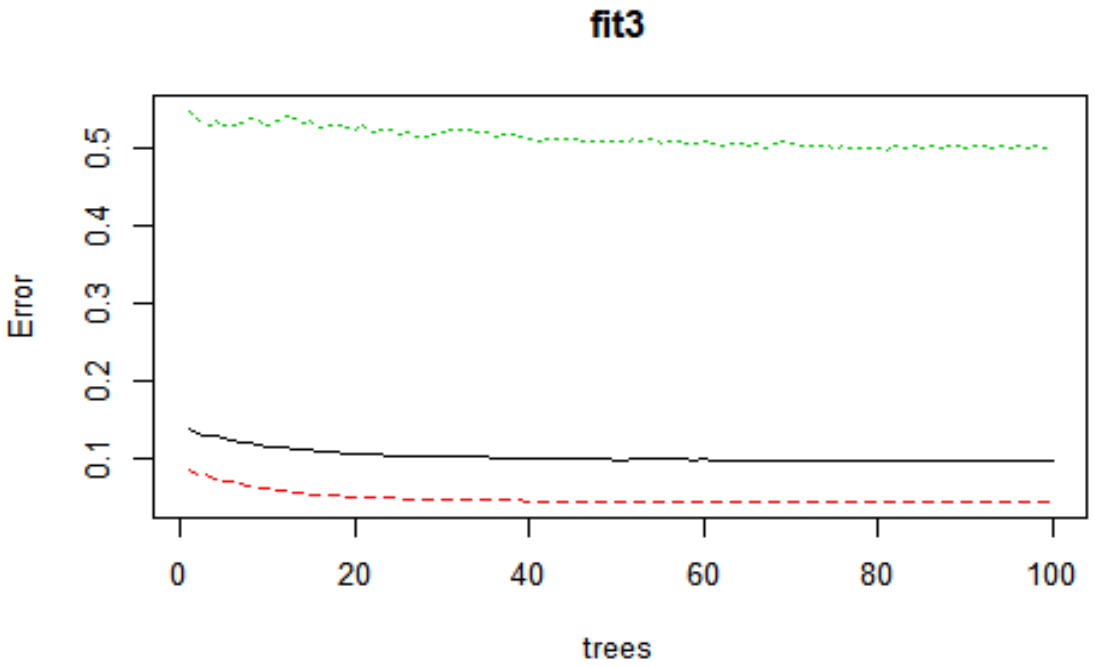
```
> performance(pred, "auc")@y.values #AUC
[[1]]
[1] 0.8848334
```

– Bagging 기법으로 적합한 결과 Training, Test set의 AUC가 각각 0.9432583과 0.8848334임을 확인할 수 있다.

Random Forest

앙상블 기법(Random Forest)

> RF 분류기 개수 결정



- 분류기 개수가 80개 이상일 때 오분류의 결과가 비교적 안정된 형태를 보임
- 비용과 시간을 고려하여 분류기의 개수를 100개로 정함

Random Forest

앙상블 기법(Random Forest)

› RF mtry 결정

- Mtry 옵션은 분류나무의 중간노드마다 임의로 선택되는 변수의 개수를 정하는 옵션.
- 일반적으로 임의로 선택되는 변수의 개수는 다음과 같다.

  분류나무의 경우 $-\sqrt{p}$

  회귀나무의 경우 $-\frac{p}{3}$

- 해당 데이터에서 사용된 변수는 11개이므로 임의 선택 변수의 개수를 **4개로 설정**하였다.

$$\sqrt{11} \fallingdotseq 4$$

Random Forest

앙상블 기법(Random Forest)

> 적합결과

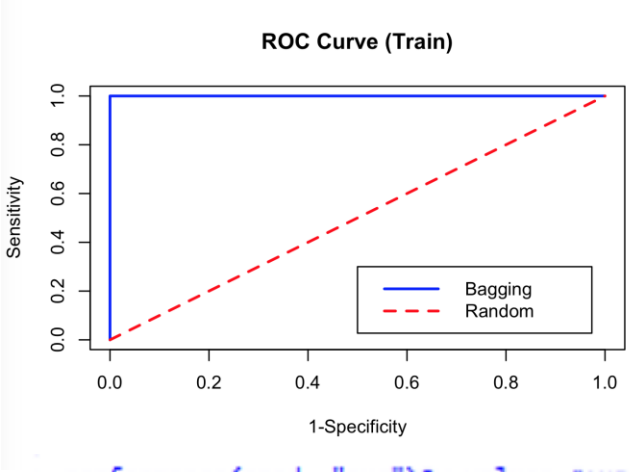
```
> fit3 = randomForest(y~job+marital+education+balance+housing+
+                       loan+contact+month+duration+campaign+poutcome,
+                     data=bank.train, ntree=100, mtry=4, importance=T)
> pred = predict(fit3, newdata=bank.test, type="prob")
> cutoff = 0.11
> yhat = ifelse(pred[,2] > cutoff, 1, 0)
> ctable = table(bank.test$y, yhat, dnn=c("Actual", "Predicted")); ctable #classification table
      Predicted
Actual     0     1
     0 12578  3210
     1   216  1871

> miss.err = 1-sum(diag(ctable))/sum(ctable); miss.err # Misclassification Rate
[1] 0.1916643
> pred.acc = 1 - miss.err; pred.acc #Prediction Accuracy
[1] 0.8083357
> diag(ctable)[2]/apply(ctable, 1, sum)[2] # Sensitivity
        1
0.8965022
> diag(ctable)[1]/apply(ctable, 1, sum)[1] # Specificity
        0
0.796681
```

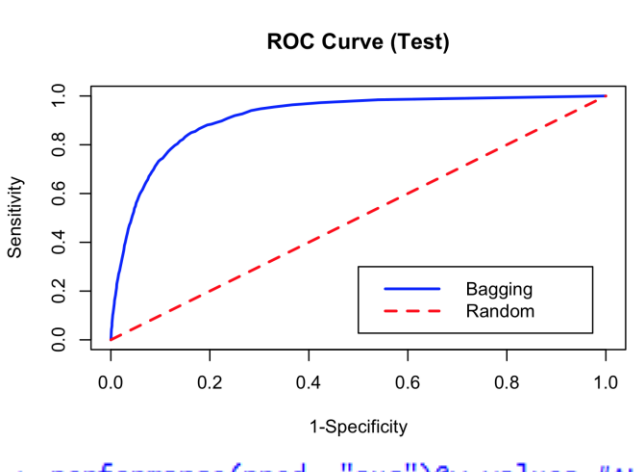
Random Forest

앙상블 기법(Random Forest)

› 적합 결과




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



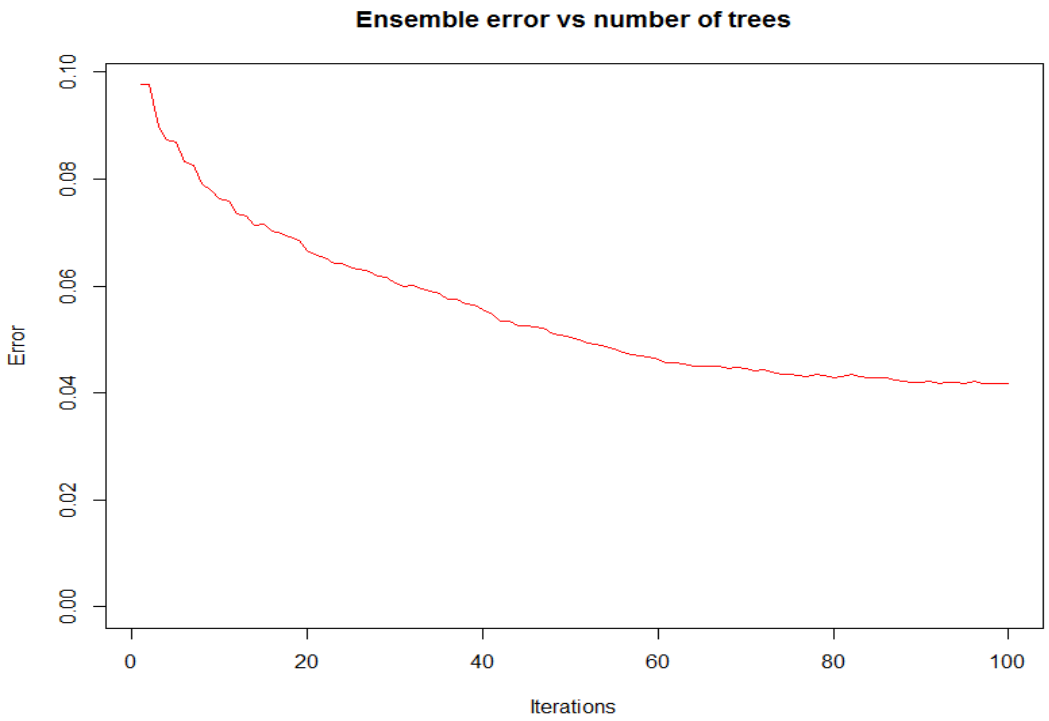

```
> performance(pred, "auc")@y.values #AUC
[[1]]
[1] 0.915038
```

- Random Forests기법으로 적합한 결과 Training, Test set의 AUC가 각각 1과 0.915임을 확인할 수 있다.
- 특히 Training Set으로 적합한 결과의 AUC는 1로서 과적합의 가능성을 엿볼 수 있음

Boosting

앙상블 기법(Boosting)

› 분류기 개수 설정



- 분류기 개수가 80개 이상일 때 오분류의 결과가 비교적 안정된 형태를 보임
- 비용과 시간을 고려하여 분류기의 개수를 100개로 정함

Boosting

앙상블 기법(Boosting)

> 적합 결과

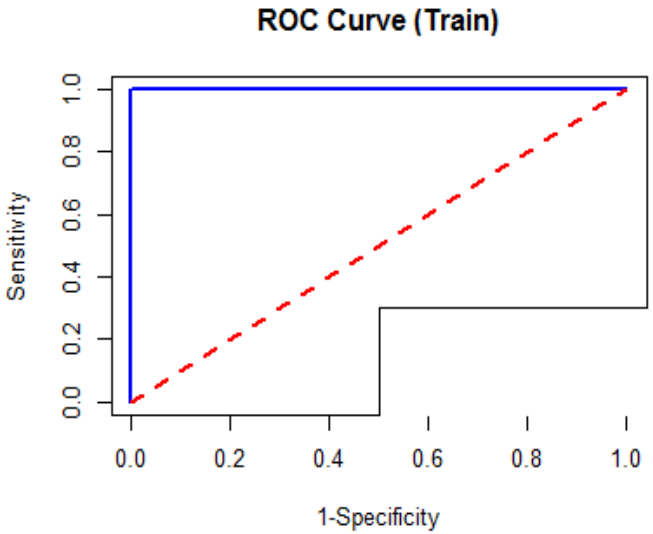
```
> pred = predict.boosting(fit2, newdata=bank.test)
> cutoff = 0.11
> yhat = ifelse(pred$prob[,2] > cutoff, 1, 0)
> ctable = table(bank.test$y, yhat, dnn=c("Actual", "Predicted")); ctable #classification table
      Predicted
Actual     0     1
   no    196 15592
   yes     0  2087
> miss.err = 1-sum(diag(ctable))/sum(ctable); miss.err # Misclassification Rate
[1] 0.8722797
> pred.acc = 1 - miss.err; pred.acc #Prediction Accuracy
[1] 0.1277203
> diag(ctable)[2]/apply(ctable, 1, sum)[2] # Sensitivity
 yes
   1
> diag(ctable)[1]/apply(ctable, 1, sum)[1] # Specificity
         no
0.01241449
```

- Rpart.control 옵션 중 maxdepth를 10으로 두고 계산한 결과라서 prediction Accuracy가 현저하게 떨어지는 것을 확인할 수 있다는 점에서 한계점으로 꼽을 수 있다.
  (원래는 maxdepth를 1로 두고 계산해야 하나 연산속도의 문제로 시행하지 못함)
- 따라서 최종 비교분석에서는 boosting 기법을 생략하기로 결정

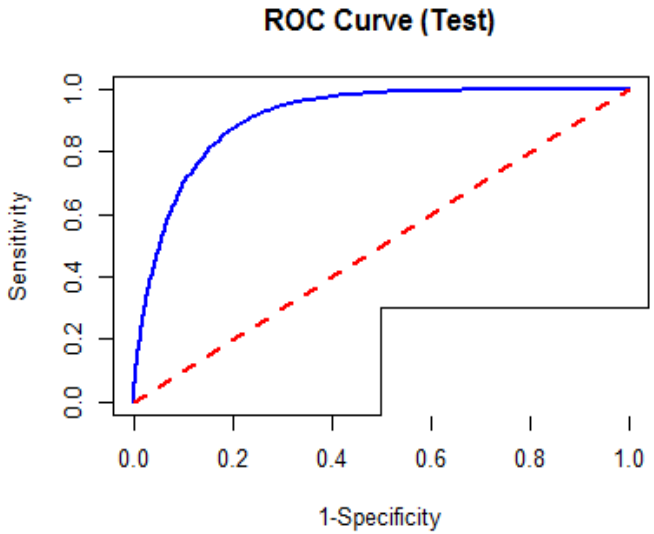
Boosting

앙상블 기법(Boosting)

› 적합 결과




```
> performance(pred, "auc")@y.values #AUC
[[1]]
[1] 0.9999966
```




```
> performance(pred, "auc")@y.values #AUC
[[1]]
[1] 0.9086856
```

- Boosting 기법으로 적합한 결과 Training, Test set의 AUC가 각각 0.9999966과 0.9086856임을 확인할 수 있다.
- 특히 Training Set으로 적합한 결과의 AUC는 1에 매우 가까운 것으로 보아 RF와 동일하게 과적합의 가능성을 엿볼 수 있음

KNN

K-Nearest Neighbors(KNN)

› KNN 분석방법의 한계점

– KNN분석 방법의 경우 범주형 자료가 포함된 데이터에는 적용하는 데 한계를 가지고 있다.
– 해당 분석에서는 수치형 변수에 대해서만 분석을 시행

=〉**분석의 한계점**

```
####수치형변수 선택
bank2=bank[,c(1,6,11,12,13,15)]
```

› 표준화작업

– Scale 함수를 활용해 수치형 변수 표준화

```
####표준화작업
bank3=data.frame(scale(bank2[,-6]),bank2[,6])
```

KNN

K-Nearest Neighbors(KNN)

〉근방 개체수의 결정(K의 개수 설정)

- 근방 개체수를 결정하기 위해서 K=5일 때와 K=10일 때의 적합결과 비교
- 적합결과 K=5일 때 예측정확도가 0.9089783이고, K=10일 때 예측정확도가 0.8981398로 K가 5일 때 예측정확도가 더 높은 것을 확인할 수 있다.

=〉K=5로 설정

K=5:

```
> pred.acc = 1 - miss.err; pred.acc #Prediction Accuracy
[1] 0.9089783
```

K=10:

```
> pred.acc = 1 - miss.err; pred.acc #Prediction Accuracy
[1] 0.8981398
```

KNN

K-Nearest Neighbors(KNN)

› 적합 결과

– 아래는 KNN을 통해서 적합한 분할표와 오분류률, 예측정확도, 민감도, 특이도에 대한 결과값을 보여주고 있다.

```
> miss.err = 1-sum(diag(ctable))/sum(ctable); miss.err # Misclassification Rate
[1] 0.1220608
> pred.acc = 1 - miss.err; pred.acc #Prediction Accuracy
[1] 0.8779392
> diag(ctable)[2]/apply(ctable, 1, sum)[2] # Sensitivity
      yes
0.2698718
> diag(ctable)[1]/apply(ctable, 1, sum)[1] # Specificity
       no
0.9583687
```

Actual	Predicted no	yes
no	11303	491
yes	1139	421

LDA

Linear Discriminant Analysis(LDA)

› 유의미한 변수 선택

- 로지스틱 회귀분석을 할 때 사용하였던 stepwise 기법을 통해서 AIC 기준으로 변수를 선택하였다.
- default, previous, age 제외

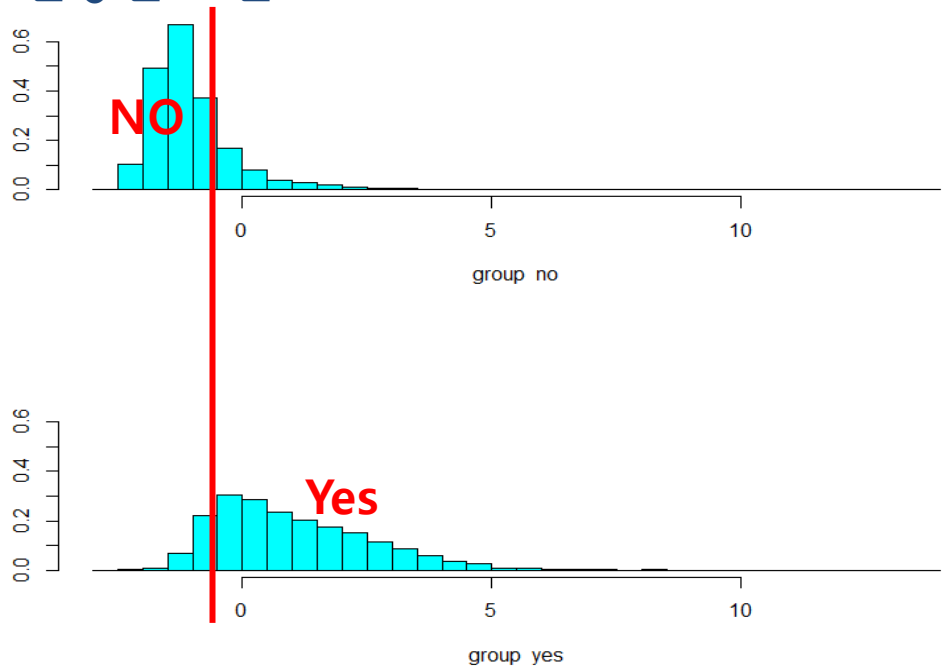
› 적합결과(cutoff=0.11)

```
> cutoff = 0.11
> pred = predict(fit, newdata=bank.test)$posterior
> yhat = ifelse(pred[,2] > cutoff, 1, 0)
> ctable = table(bank.test$y, yhat, dnn=c("Actual", "Predicted")); ctable #classification table
      Predicted
Actual     0     1
   no  10530  1264
   yes   436  1124
> ### Errors
> miss.err = 1-sum(diag(ctable))/sum(ctable); miss.err # Misclassification Rate
[1] 0.1273027
> pred.acc = 1 - miss.err; pred.acc #Prediction Accuracy
[1] 0.8726973
> diag(ctable)[2]/apply(ctable, 1, sum)[2] # Sensitivity
      yes
0.7205128
> diag(ctable)[1]/apply(ctable, 1, sum)[1] # Specificity
       no
0.8928269
```

LDA

Linear Discriminant Analysis(LDA)

> 반응변수 분류

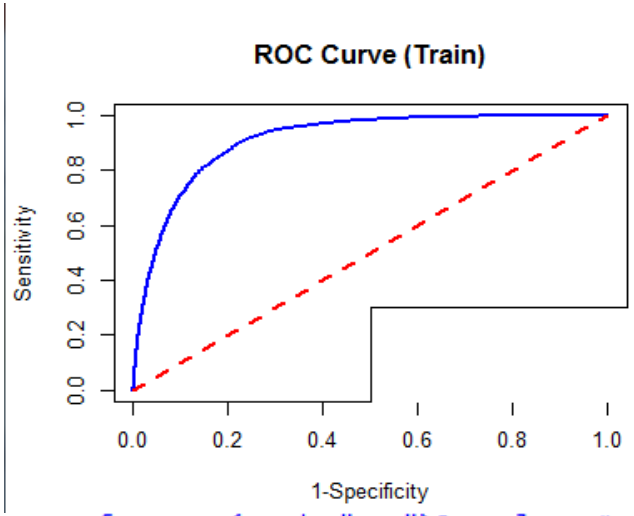


- 표준화된 자료에 대해서 확률값에 대한 플롯을 그려보면 다음과 같은 분포를 보인다.
- 위의 기준선 왼쪽에 있는 경우 y를 no로 예측, 오른쪽에 있는 경우 yes로 예측

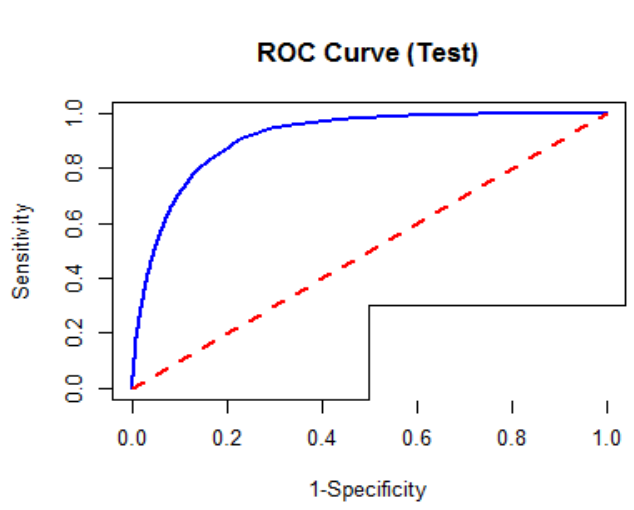
LDA

Linear Discriminant Analysis(LDA)

> ROC Curve




```
> performance(pred, "auc")@y.values #AUC
[[1]]
[1] 0.9093687
```




```
> performance(pred, "auc")@y.values #AUC
[[1]]
[1] 0.9105549
```

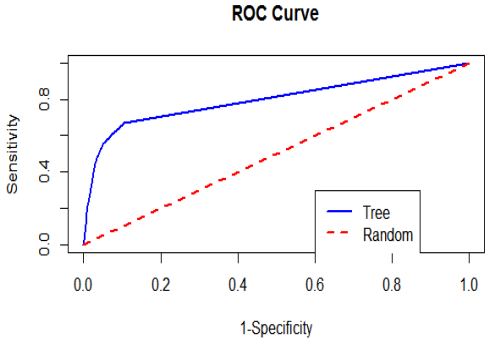
- LDA기법으로 적합한 결과 Training, Test set의 AUC가 각각 0.9093687과 0.9105549임을 확인할 수 있다.
- 두 적합결과가 비슷한 AUC를 보이지만 Test set의 적합결과가 오히려 약간 더 높음

Comparison

Model Comparison

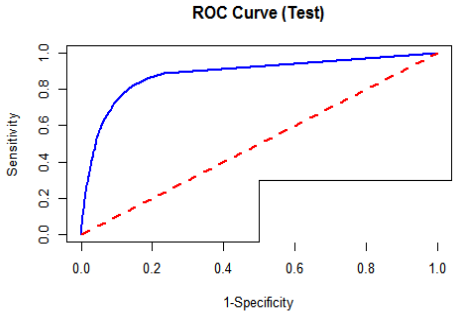
Decision Tree

AUC = 0.796



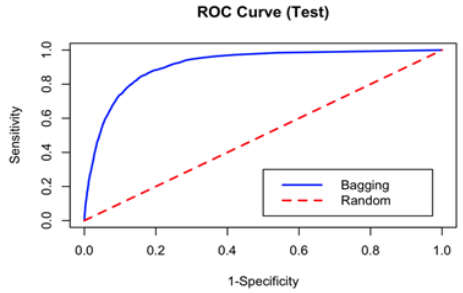
Bagging

AUC = 0.884



Random Forest

AUC = 0.915



- Ensembles 기법을 활용하여 분석을 했을 시 단순 Decision Tree에 비해 AUC가 증가하는 것을 볼 수 있다.

  =〉**예측력의 증가를 보여줌**

- 하지만, Ensembles 기법을 활용하면 해석이 힘들다는 한계점을 가지고 있다.

Comparison

Model Comparison

	Logistic model	Neural Network	Decision Tree	KNN	LDA
AUC	0.906	0.905	0.796	–	0.910
Prediction Accuracy (cutoff=0.11)	0.902	0.815	0.868	0.877 (Cutoff x)	0.872

- AUC만으로 확인했을 때는 LDA가 가장 높게 나오나 KNN기법과 LDA의 경우 범주형 자료에 대한 분석이 어렵다는 단점이 있어서 해당 데이터에서는 적합한 방법이라고 보기 힘들다.
- LDA의 경우는 y값이 binary가 아니라 3개 이상일 때 로지스틱 회귀분석에 비해 강점을 보일 것으로 생각된다.
- 최종적으로,
  해당 데이터에서는 적절한 해석이 가능하고 예측력도 적정수준을 가지고 있는 로지스틱 회귀분석이 적절한 방법이라고 생각된다.