기계학습을 이용한
빅데이터 분석 강좌
3주차

- 모형의 복잡도
- 모형 평가
- 모형 선택
- R 실습

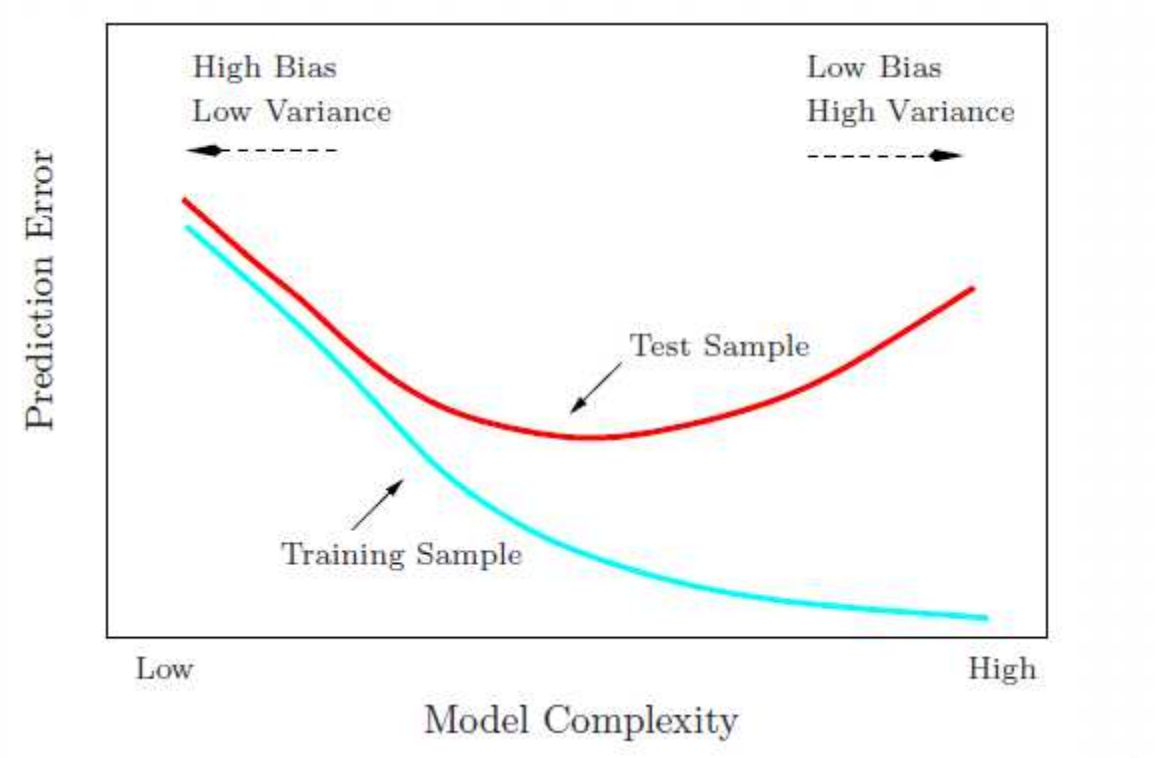
1. 모형의 복잡도

※ P(차원) -> ↑

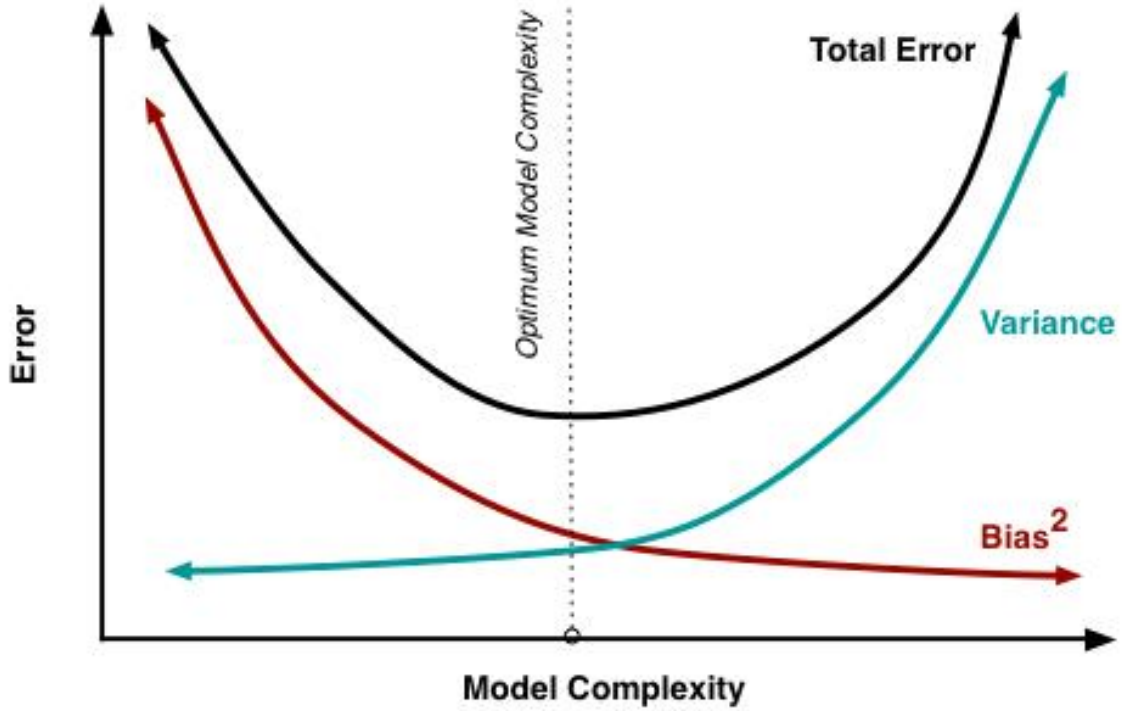
- 모형의 복잡도(model complexity) 증가
- 모형의 편향(bias) $E(\hat{f}(X)) - f(X)$ 줄어듬
- 모형의 분산(Variance) $Var(\hat{f}(x))$ 증가
- 모형 적합에 사용된 training error 줄어듬
- 모형적합에 사용되지 않은 데이터에서는 정확도 지표가 초반에는 개선되나, 나중에는 안좋아짐

  -> 과적합(overfitting)

Bias-Variance tradeoff



출처: https://stats.stackexchange.com/questions/152882/question-about-bias-variance-tradeoff



출처: https://stats.stackexchange.com/questions/152882/question-about-bias-variance-tradeoff

2. 모형 평가

※ 모형 평가방법 개념 정의

※ 좋은 모형의 조건(what are good models?)

좋은 모형의 조건은 간단(simple)하고 타당(valid)하며, 강건(robust)함을 요소로 가지고 있어야 한다(Sayama, 2015)

모형의 단순함(simplicity)는 무엇을 모델링할 것인지의 핵심요소이다. 우리가 모형을 구축하려는 주된 이유는 현실을 짧고 단순하게 묘사하고 싶기 때문이다. Occam's razor 의 유명한 원칙에 의하면, 두 모형이 같은 예측력을 가지고 있을 때, 간단한 모형을 선택해야 한다는 것이다. 이것은 이론이나 논리적으로 입증된 사실이 아니라 과학에서 일반적으로 받아들여지는 관행이라고 할 수 있다. 간소화(law of parsimony)는 경제성과 통찰력 측면에서 좋다. 모형에서 손실(loss)이 없다면 모수(parameters), 변수(variables), 가정(assumption)을 하나라도 제거하는 것이 좋다.

모형의 타당성(validity)은 모형의 예측이 관측된 현실과 얼마나 가깝게 일치하는가이다. 모형의 타당성은 실용적인 관점에서 가장 중요하다. 만약 모형의 예측이 합리적으로 관측치와 일치하지 않는다면 모형은 현실을 나타내지 못하는 것이고, 쓸모없게 된다. 따라서 모형의 예측력뿐만 아니라 사용하는 가정에 대한 타당성을 확인하는 것이 중요하다. 즉, 기존의 지식과 상식을 고려하여 모형에서 사용된 각각의 가정이 의미가 있는지를 살펴보아야 한다.

때로는 모형의 단순성과 타당성 사이에 상충관계(trade off)가 존재한다. 모형의 복잡도를 증가시키면 모형은 관측된 데이터를 잘 설명하지만 단순성에서 멀어질 수 있으며 과적합(overfitting)의 문제도 생길 수 있다. 따라서 모형을 설계할 때 두 기준 사이에서 균형을 맞추는 것이 필요하다.

마지막으로, **모형의 강건성(robustness)**은 모형의 예측이 모형의 가정(assumptions)과 모수(parameters) 세팅의 사소한 변화에 얼마나 둔감한지를 나타낸다. 모형의 강건성은 현실세계로부터 가정을 추가하거나 모수값을 측정할 때 항상 오류(error)가 존재하기 때문에 중요한 부분이다. 만약 모형의 예측력이 미세한 변화에도 민감하다면, 그것으로부터 파생된 결론은 아마도 신뢰할 수 없을 것이다. 하지만 모형이 강건하다면, 결과는 가정과 모수의 작은 변화에도 유지될 것이고, 현실에 더 많이 적용될 수 있고 신뢰를 가질 수 있다.

위의 좋은 모형이 갖추어야 할 요소들을 고려해볼 때, 모형 평가는 하나의 자료를 분석하는 경우 여러 통계모형들을 비교하여 분석하는 것이 바람직하며, 이 중 자료를 최적으로 설명하는 모형을 선택하는 것이 좋다. 최적의 모형을 선택하기 위해서는 여러 모형을 비교 및 평가해야 하고, 선정된 최적의 모형이 다른 모형에 비하여 우수하다는 사실을 입증해야 한다.

모형을 평가하는 방법에서 고려해야 할 사항은 다음 4가지임

- 예측력: 얼마나 예측을 잘하는가?
- 해석력: 입력변수와 출력변수와의 관계를 잘 설명하는가?
- 효율성: 얼마나 적은 수의 입력변수로 모형을 구축했는가?
- 안정성: 모집단 내 다른 자료에 적용하는 경우 같은 결과를 주는가?

예측력, 해석력, 효율성, 안정성 모두 모형을 평가할 때 중요한 요소이지만 예측 문제에서 가장 중요한 고려 사항은 예측력이다. 아무리 안정적이고 효율적이며 해석이 쉬워도, 실제 문제에 적용했을 경우 빗나간 결과를 도출하는 경우, 모형을 적용하는 의미가 없다.

따라서 **모형의 평가란, 예측(prediction)을 위해 만든 모형이 임의의 모형(random model)보다 예측력이 우수한지, 고려된 다른 모형들 중 어느 모형이 가장 우수한 예측력을 보유하고 있는지를 비교, 분석하는 과정**이라고 할 수 있다.

3. 모형 선택

모형을 선택하는 기준은 회귀모형은 Mallow's Cp, Adjusted R^2 , 분류모형은 오분류율 (misclassification rate) 등으로 모형을 비교할 수 있다.

여러가지 모형을 이용하여 종속변수의 사후확률 $\Pr(y=1|x_1, ..., x_p)$을 계산한다. 즉, 사후확률은 독립변수 $(x_1, ..., x_p)$가 주어졌을 때, 목표변수 Y가 1이 될 확률이다. 사전확률은 독립변수를 고려하지 않은 경우의 확률 Pr(Y=1) (이 경우 자료의 분포와 모집단의 분포가 같아야 함)이고, 사후확률은 독립변수를 고려한 확률 $\Pr(y=1|x)$을 의미한다.

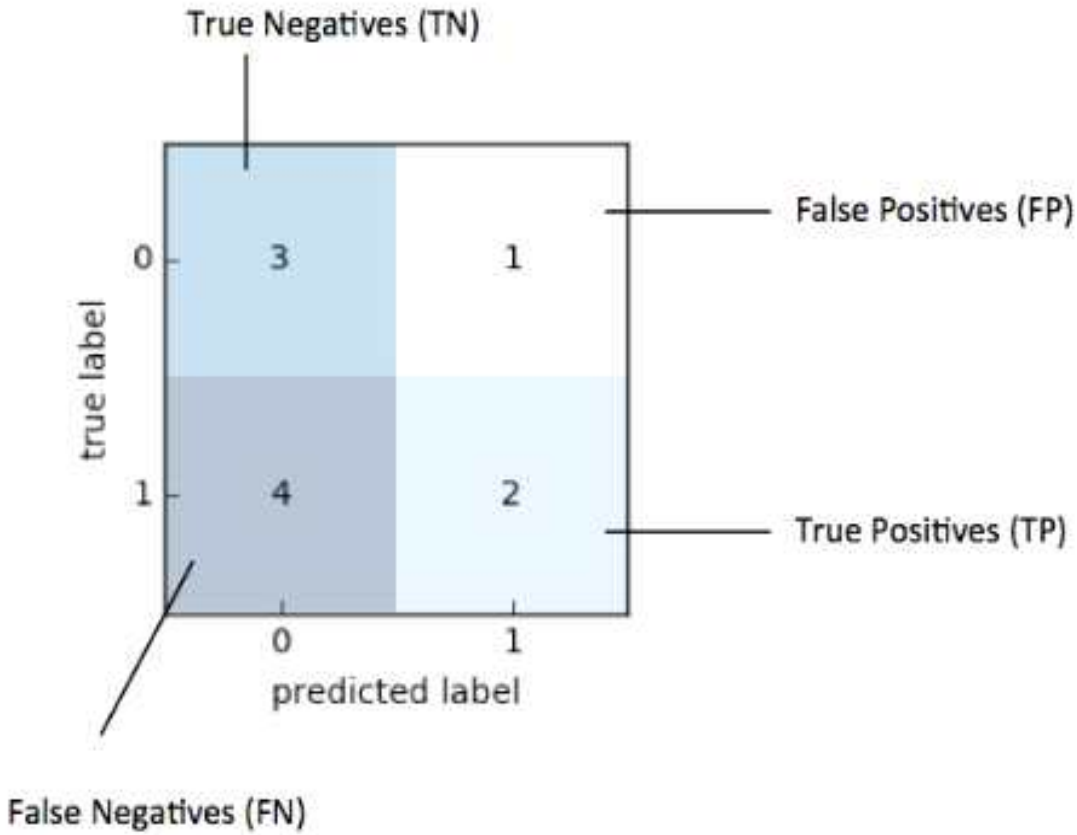
분류 기준값이란 사후확률을 구한 후, 이를 이용하여 자료를 분류(목표변수의 범주를 결정)하기 위하여 정하는 값이다. 예를 들어, 사후확률이 0.8 이상이면 자료를 1그룹에 할당하고 사후확률이 0.8 미만이면 자료를 0 그룹에 할당한다. 분류 기준값은 사전확률과 손실함수 등 여러가지를 고려하여 결정하도록 한다.

오분류표는 목표변수의 실제 범주와 모형에 의해 예측된 범주 사이의 관계를 나타내는 표이다. 오분류율=1-정분류율이며, 정분류율을 정확도(accuracy), 오분류율을 오차율(error rate)이라고 표현하기도 한다. 또한, 오분류표를 오분류 행렬(confusion matrix)이라고도 한다. 오분류율에 대한 다양한 추정치가 있을 수 있는데, 범주가 2개인 경우의 오분류표의 구성은 다음과 같다.

오분류표

구분		예측 변수		
		0	1	
목표변수	0	실제0, 예측0	실제0, 예측1	실제 0
	1	실제1, 예측0	실제1, 예측1	실제 1
		예측 0	예측 1	

오분류표



자료: R aschka, S. (2017). Lift Score. https://rasbt.github.io/mlxtend/user_guide/ evaluate/lift_score/에서 2017. 11. 29. 인출.

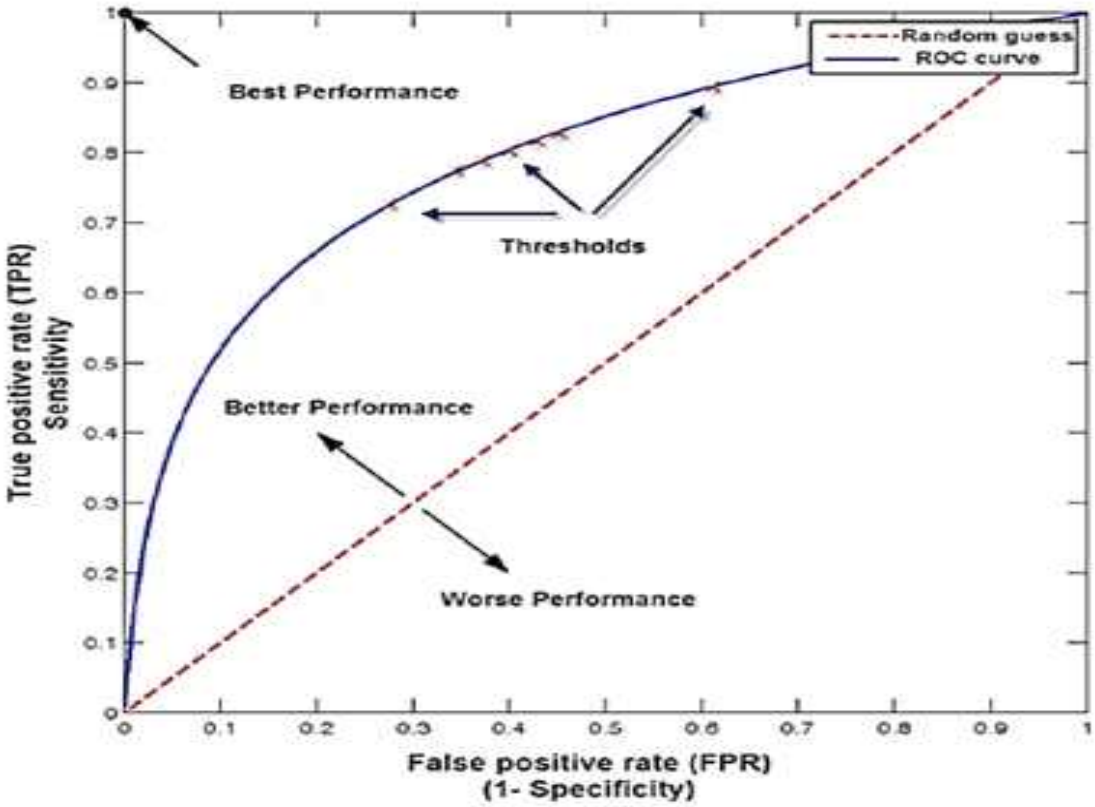
오분류율에 대한 다양한 측정치는 다음과 같다.

- 정분류율=(실제0,예측0)빈도+(실제1,예측1)의 빈도/전체빈도
- 오분류율=(실제0,예측1)빈도+(실제1,예측0)의 빈도/전체빈도
- 민감도(sensitivity)=(실제1,예측1)의 빈도/실제 1의 빈도
- 특이도(specificity)=(실제0,예측0)의 빈도/실제 0의 빈도

민감도는 범주 1에서의 정분류율이고, 특이도는 범주 0에서의 정분류율을 의미한다.

ROC(receiver operating characteristic) 곡선은 구축한 모형의 성능을 민감도와 특이도에 의해 판단할 수 있도록 시각화한 그림으로 신호감지이론(Signal Decision Theory)으로부터 출발하였고 신호와 잡음의 분리를 목적으로 한다.

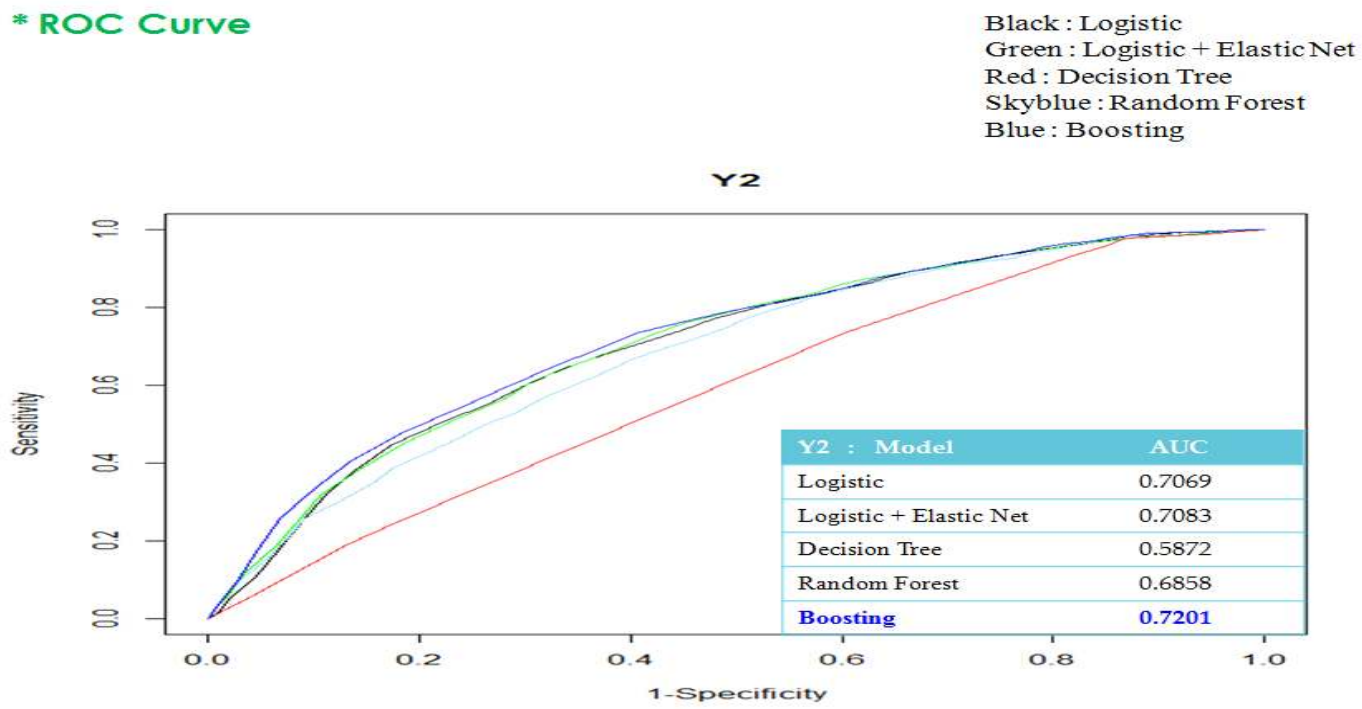
ROC



자료 : Hassouna, M., Tarhini, A., Elyas, T. (2015). Customer Churn in Mobile Markets: A Comparison of Techniques. International Business Research, Vol 8(6), pp. 224-237.

분류 기준값이 바뀌면서 특이도와 민감도의 값이 바뀌며, 여러 개의 분류 기준값에서 구하여진 민감도 특이도 값의 쌍들을 x 축에는 1-특이도, y 축에는 민감도를 지정하고 그린 곡선이다. 민감도와 특이도는 반비례하고, 따라서 ROC 곡선은 증가하는 형태를 띤다.

AUC(Area Under the Curve)는 classifier의 합리성을 나타내는 지표로 ROC 곡선의 아래 면적을 의미하며, AUC가 높을수록 좋은 모형이다.



이익도표(Lift Chart) 개념 역시 모형 평가 기준 중 하나이다. 이익도표 작성 과정은 다음과 같다.

우선, 모형적합을 통한 사후확률을 계산하고 사후확률의 순서에 의한 자료를 정리한다. 정렬된 자료를 균일하게 N등분하고 N등분의 각 등급에서 목표변수의 특정 범주에 대한 빈도를 구한다. N등분의 각 등급에서 %Captured Response, %Response 및 Lift 통계량을 구한다. 이익도표의 plot 은 수평축에는 N등분의 등급을, 수직축에는 위의 3개의 통계량 중 하나를 이용해 그래프를 그린다.

이익도표에 사용되는 통계량 %Captured response, %Response 수식은 다음과 같다.

%Captured response=해당 등급에서 목표변수의 특정 범주의 빈도*100/(전체에서 목표변수의 특정 범주의 빈도)

%Response=해당 등급에서 목표변수의 특정 범주의 빈도*100/(해당 등급에서 전체 빈도)

%Response 통계량은 예측확률을 순서대로 정렬하였을 때, 예측확률 상위 x% 데이터에서의 정확도를 의미하며, 일반적으로 %Response는 등급이 낮아지면서 감소한다.

Base Line Lift를 전체에서 목표변수의 특정범주의 빈도*100/(전체 빈도)로 나타낼 때, Lift는

- Lift= 해당등급의 %Response/ Base Line Lift

로 계산된다.

앞의 오분류율 표를 적용해보면 Lift는 다음과 같다.

$$Lift = \frac{(TP/(TP+FN)}{(TP+FP)/(TP+TN+FP+FN)}$$

모형평가기준 요약

	Domain	Plot	Explanation
Lift chart	Marketing	TP Subset size	TP (TP+FP)/(TP+FP+TN+FN)
ROC curve	Communications	TP rate FP rate	TP/(TP+FN) FP/(FP+TN)
Recall-precision curve	Information retrieval	Recall Precision	TP/(TP+FN) TP/(TP+FP)

자료: Ian H, W. & Eibe, F. (2016). Evaluation—Lift and Costs.

A summary of models and some of their characteristics

Model	Allow n < p	Pre-processing	Interpretable	Automatic feature selection	# Tuning parameters	Robust to predictor noise	Computation time
Linear regression[1)]	×	CS, NZV, Corr	✓	×	0	×	✓
Elastic net/lasso	✓	CS, NZV	✓	✓	1-2	×	✓
Support vector machines	✓	CS	×	×	1-3	×	×
Single trees	✓		○	✓	1	✓	✓
Model trees/rules[1)]	✓		○	✓	1-2	✓	✓
Random forest	✓		×	○	0-1	✓	×
Boosted trees	✓		×	✓	3	✓	×
Logistic regression[2)]	×	CS, NZV, Corr	✓	×	0	×	✓

주: 1) regression only.
2) classification only.
Symbols represent affirmative (✓), negative (×), and somewhere in between (○).
CS = centering and scaling.
NZV = remove near-zero predictors.
Corr = remove highly correlated predictors.

자료: Kuhn, M. & Johnson, K. (2013). Applied Predictive Modeling. USA: Springer Science & Business Media.

4. R 실습 : 모형 비교

Data : Adult
(https://archive.ics.uci.edu/ml/datasets/Adult)
(https://goo.gl/yV0qq)

목적: 설명변수에 근거해서 연소득(wage)이 $50K가 넘는지 예측

```
##############################################
# 기계학습을 이용한 빅데이터 분석강좌(3주차)                #
##############################################


install.packages(c("dplyr", "ggplot2", "ISLR", "MASS","glmnet",
                   "randomForest", "gbm", "rpart","boot"))

install.packages("ROCR")
```

```
library(dplyr)
library(ggplot2)
library(ISLR)
library(MASS)
library(glmnet)
library(randomForest)
library(gbm)
library(rpart)
library(boot)

library(ROCR)

adult <- read.csv("기계학습을이용한빅데이터분석강좌(2019)/adult.txt",header=FALSE,
strip.white=TRUE)
names(adult) <- c('age','workclass','fnlwgt',
                  'education','education.num','marital.status',
```

```
                    'occupation','relationship','race',
                    'sex','capital.gain','capital.loss','hours.per.week','native.country','wage')
head(adult)

#####################################
#   binomial_deviance               #
#####################################
```

$$D = 2\sum_{i=1}^{n}[y_i \log(y_i/\widehat{\mu_i}) + (1-y_i)\log((1-y_i)/(1-\widehat{\mu_i}))]$$ # 작을수록 좋은 모형

```
 binomial_deviance <- function(y_obs, yhat){
     epsilon = 0.0001
      yhat = ifelse(yhat < epsilon, epsilon, yhat)
      yhat = ifelse(yhat > 1-epsilon, 1-epsilon, yhat)
      a = ifelse(y_obs==0, 0, y_obs * log(y_obs/yhat))
      b = ifelse(y_obs==1, 0, (1-y_obs) * log((1-y_obs)/(1-yhat)))
      return(2*sum(a + b))
 }
```

```
####################################
#   훈련,검증, 테스트셋 구분                #
####################################

set.seed(1)
n=nrow(adult)
idx <- 1:n

training.idx <- sample(idx,n*.6)

idx <- setdiff(idx,training.idx)  # 나머지
validate.idx <- sample(idx,n*.2)
test.idx <- setdiff(idx,validate.idx)

training <- adult[training.idx,]
validation <- adult[validate.idx,]
test <- adult[test.idx,]
```

```
#################
##   Logistic
#################
ad_glm_full <- glm(wage~., data=training, family=binomial)

y_obs <- ifelse(validation$wage==">50K",1,0)

yhat_lm <- predict(ad_glm_full, newdata=validation, type="response")

### performance

binomial_deviance(y_obs, yhat_lm)

pred_lm <- prediction(yhat_lm, y_obs)

slotNames(pred_lm)
```

```
pred_lm@n.pos
pred_lm@cutoffs
pred_lm@fp

dim(validation)
length(pred_lm@cutoffs[[1]])



perf_lm <- performance(pred_lm, measure = "tpr", x.measure = "fpr")

plot(perf_lm, col='black', main="ROC Curve for GLM")
abline(0,1)
performance(pred_lm, "auc")@y.values[[1]]
```

```
#############
# lasso     #
#############

xx <- model.matrix(wage~.-1, adult)
x<- xx[training.idx,]
y <- ifelse(training$wage == ">50K",1,0)

ad_cvfit <- cv.glmnet(x,y,family="binomial")
plot(ad_cvfit)

yhat_glmnet <- predict(ad_cvfit, s="lambda.1se", newx=xx[validate.idx,], type='response')
yhat_glmnet <- yhat_glmnet[,1]

### performance

binomial_deviance(y_obs, yhat_glmnet)
```

```
pred_glmnet <- prediction(yhat_glmnet, y_obs)

perf_glmnet <- performance(pred_glmnet, measure = "tpr", x.measure = "fpr")

plot(perf_lm, col='black', main="ROC Curve ")
plot(perf_glmnet, col='blue', add=TRUE)
abline(0,1)
legend('bottomright', inset=.1,
       legend=c("GLM","glmnet"),
       col=c('black','blue'),lty=1,lwd=2)

performance(pred_glmnet, "auc")@y.values[[1]]
```

```
#########################
#    Decision Tree      #
#########################

cvr_tr <- rpart(wage~. , data=training)

yhat_tr <- predict(cvr_tr, validation)
head(yhat_tr)
yhat_tr <- yhat_tr[,">50K"]

### performance

binomial_deviance(y_obs, yhat_tr)

pred_tr <- prediction(yhat_tr, y_obs)

perf_tr <- performance(pred_tr, measure = "tpr", x.measure = "fpr")
```

```
plot(perf_lm, col='black', main="ROC Curve ")
plot(perf_glmnet, col='blue', add=TRUE)
plot(perf_tr, col='pink', add=TRUE)
abline(0,1)
legend('bottomright', inset=.1,
       legend=c("GLM","glmnet","Decision Tree"),
       col=c('black','blue','pink'),lty=1,lwd=2)

performance(pred_tr, "auc")@y.values[[1]]
```

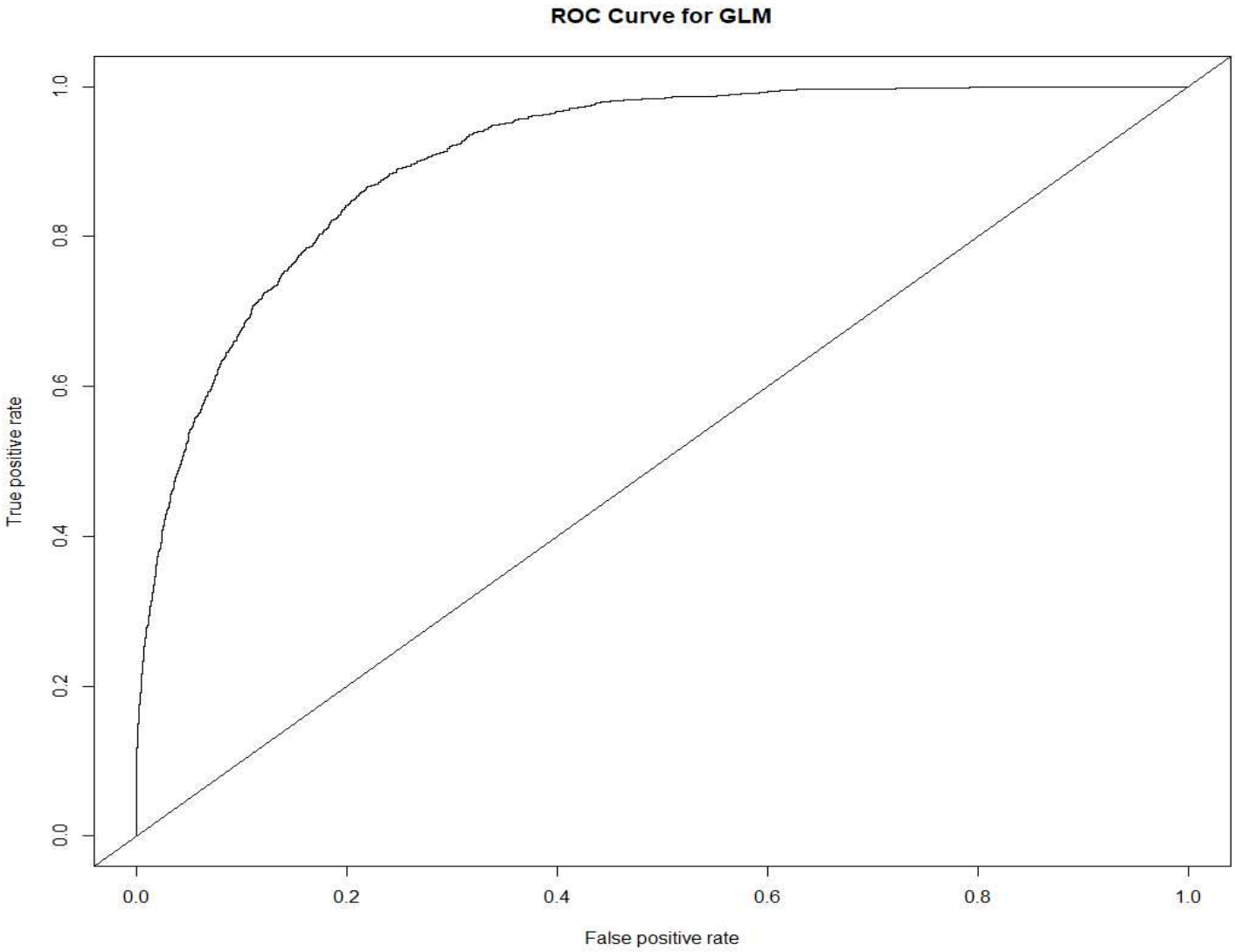
```
#########################
#    Random Forest      #
#########################

set.seed(1)
ad_rf <- randomForest(wage~., training)
ad_rf

yhat_rf  <- predict(ad_rf, validation, type='prob')[,'>50K']

### performance

binomial_deviance(y_obs, yhat_rf)

pred_rf <- prediction(yhat_rf, y_obs)

perf_rf <- performance(pred_rf, measure = "tpr", x.measure = "fpr")
```

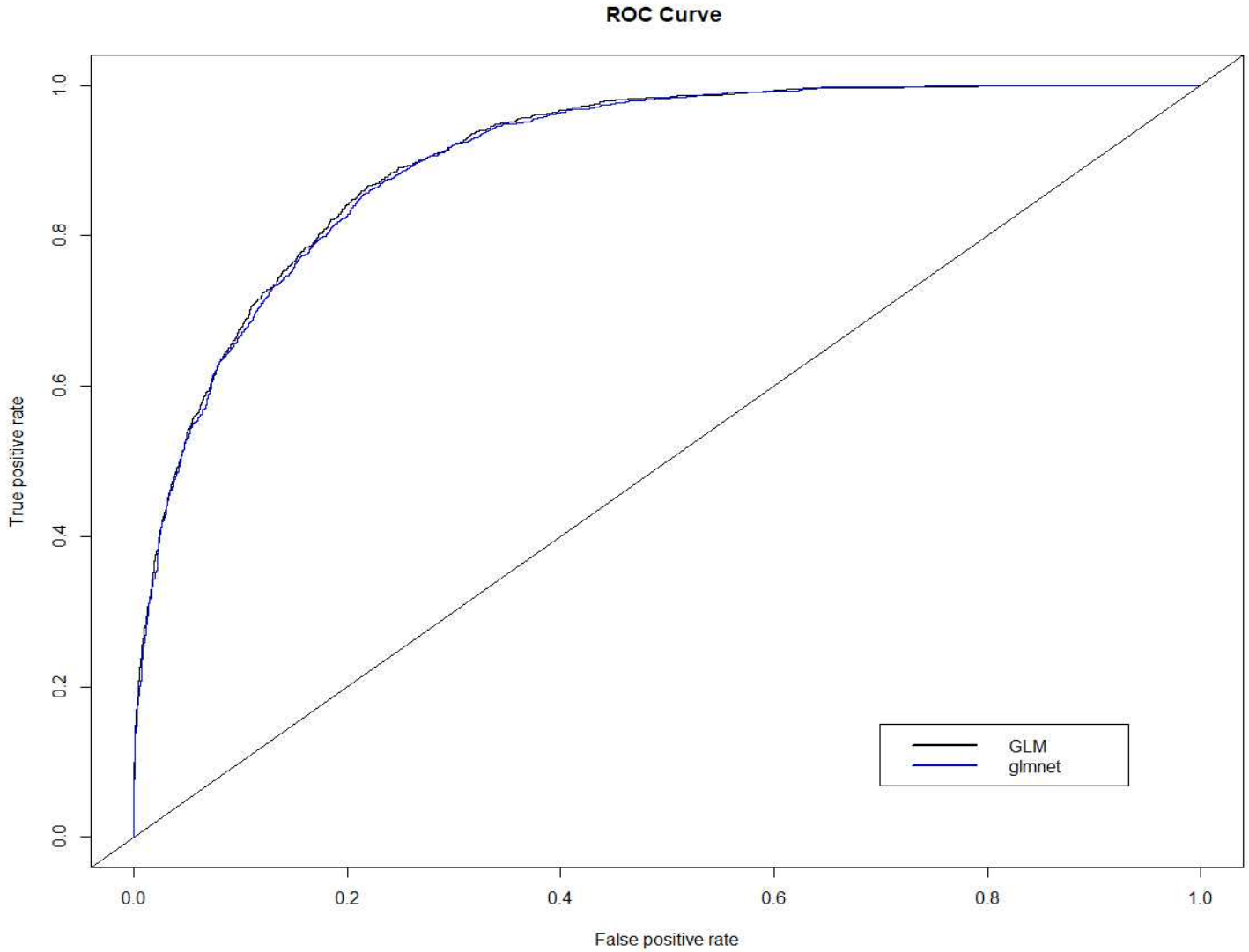
```
plot(perf_lm, col='black', main="ROC Curve ")
plot(perf_glmnet, col='blue', add=TRUE)
plot(perf_tr, col='pink', add=TRUE)
plot(perf_rf, col='red', add=TRUE)
abline(0,1)
legend('bottomright', inset=.1,
       legend=c("GLM","glmnet","Decision Tree","Random Forest"),
       col=c('black','blue','pink','red'),lty=1,lwd=2)

performance(pred_rf, "auc")@y.values[[1]]
```

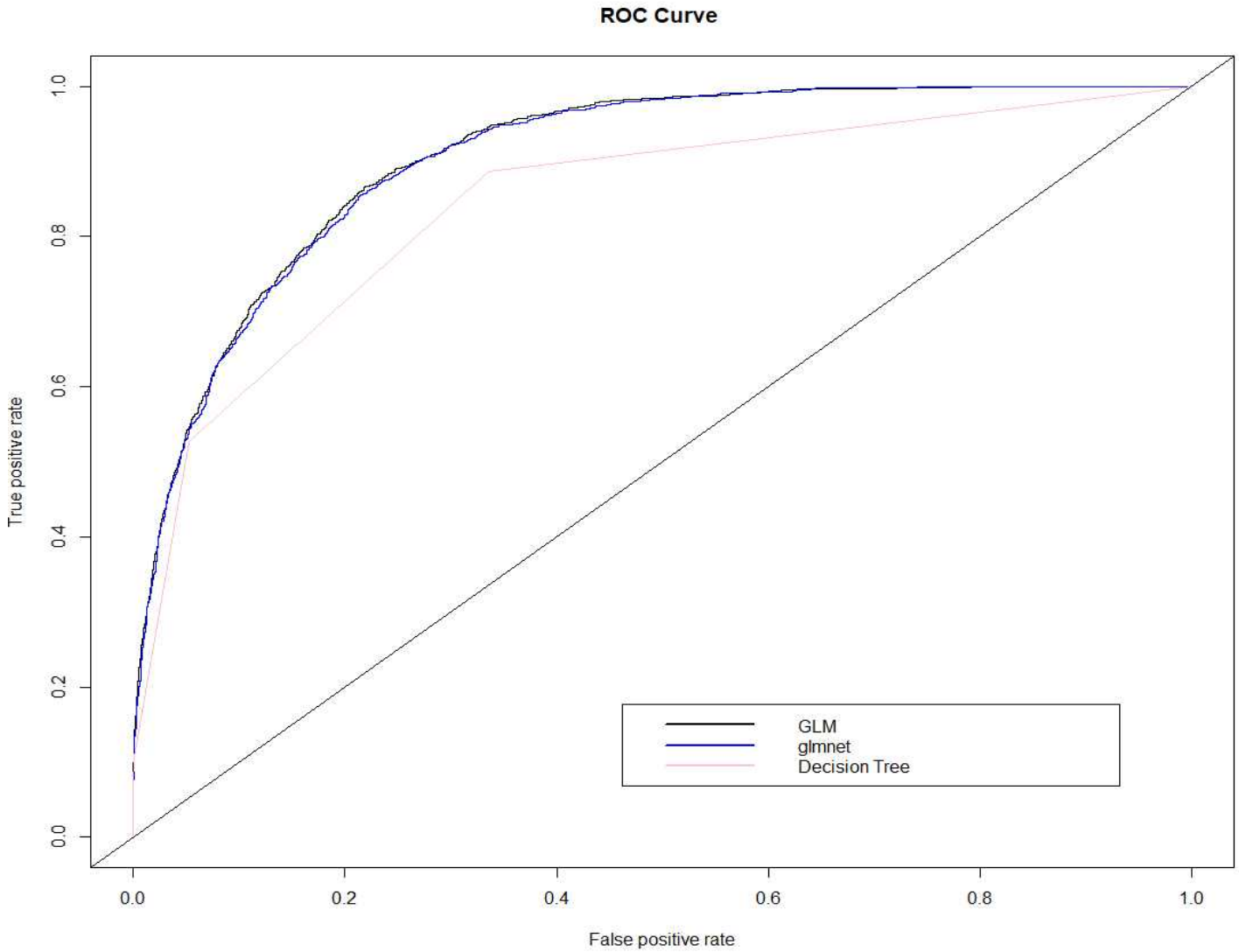
```
#########################
#    Boosting           #
#########################
set.seed(1)
adult_gbm <- training %>% mutate(wage=ifelse(wage==">50K",1,0))

ad_gbm <- gbm(wage~., data=adult_gbm,
              distribution="bernoulli",
              n.trees=20000, cv.folds=3, verbose=TRUE)
best_iter <- gbm.perf(ad_gbm, method="cv")

best_iter # 20000

yhat_gbm  <- predict(ad_gbm,n.trees=best_iter, newdata= validation, type='response')

### performance
```

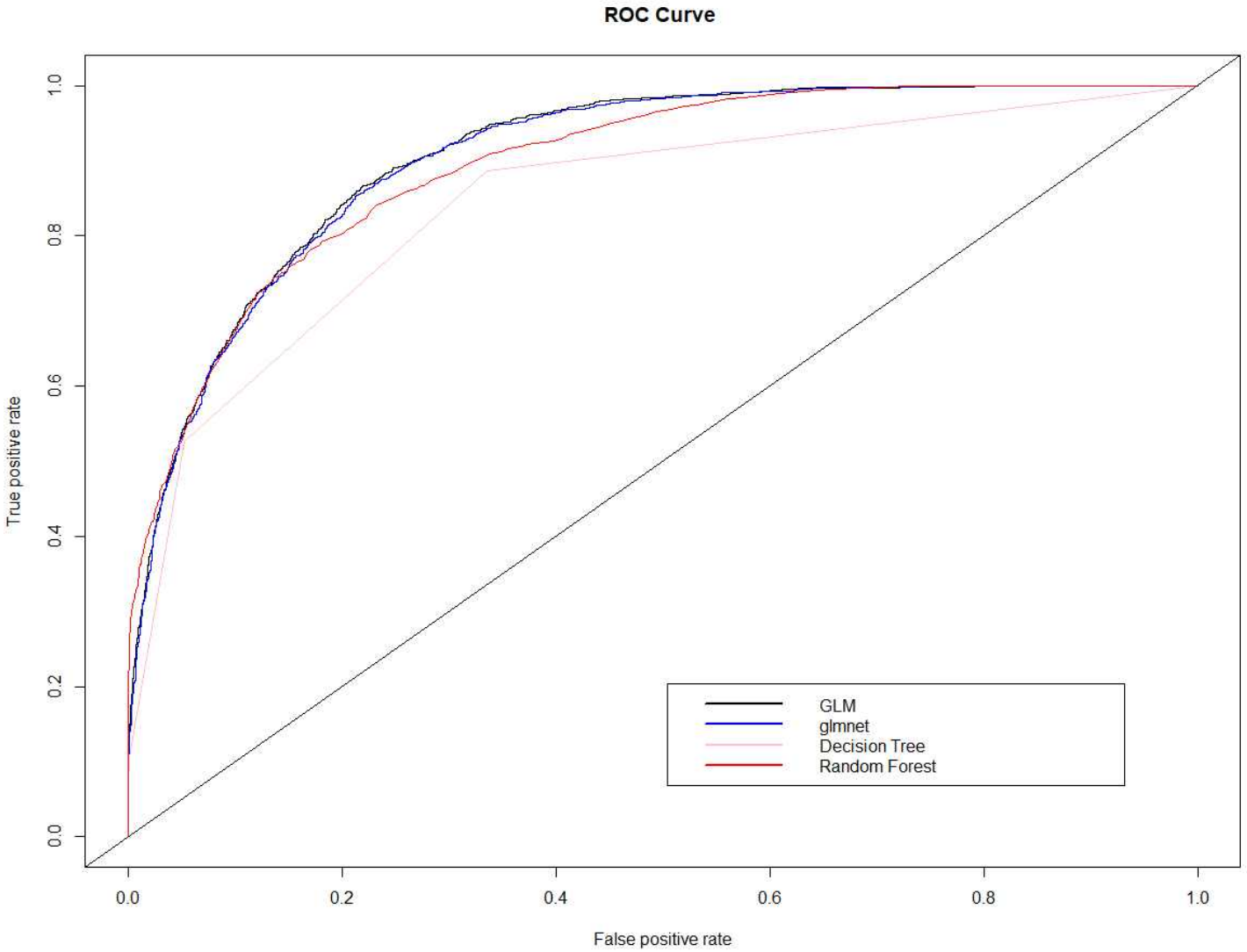
```
binomial_deviance(y_obs, yhat_gbm)

pred_gbm <- prediction(yhat_gbm, y_obs)

perf_gbm <- performance(pred_gbm, measure = "tpr", x.measure = "fpr")

plot(perf_lm, col='black', main="ROC Curve ")
plot(perf_glmnet, col='blue', add=TRUE)
plot(perf_tr, col='pink', add=TRUE)
plot(perf_rf, col='red', add=TRUE)
plot(perf_gbm, col='light blue', add=TRUE)
abline(0,1)
legend('bottomright', inset=.1,
       legend=c("GLM","glmnet","Decision Tree","Random Forest", "Boosting"),
       col=c('black','blue','pink','red','light blue'),lty=1,lwd=4)

performance(pred_gbm, "auc")@y.values[[1]]
```

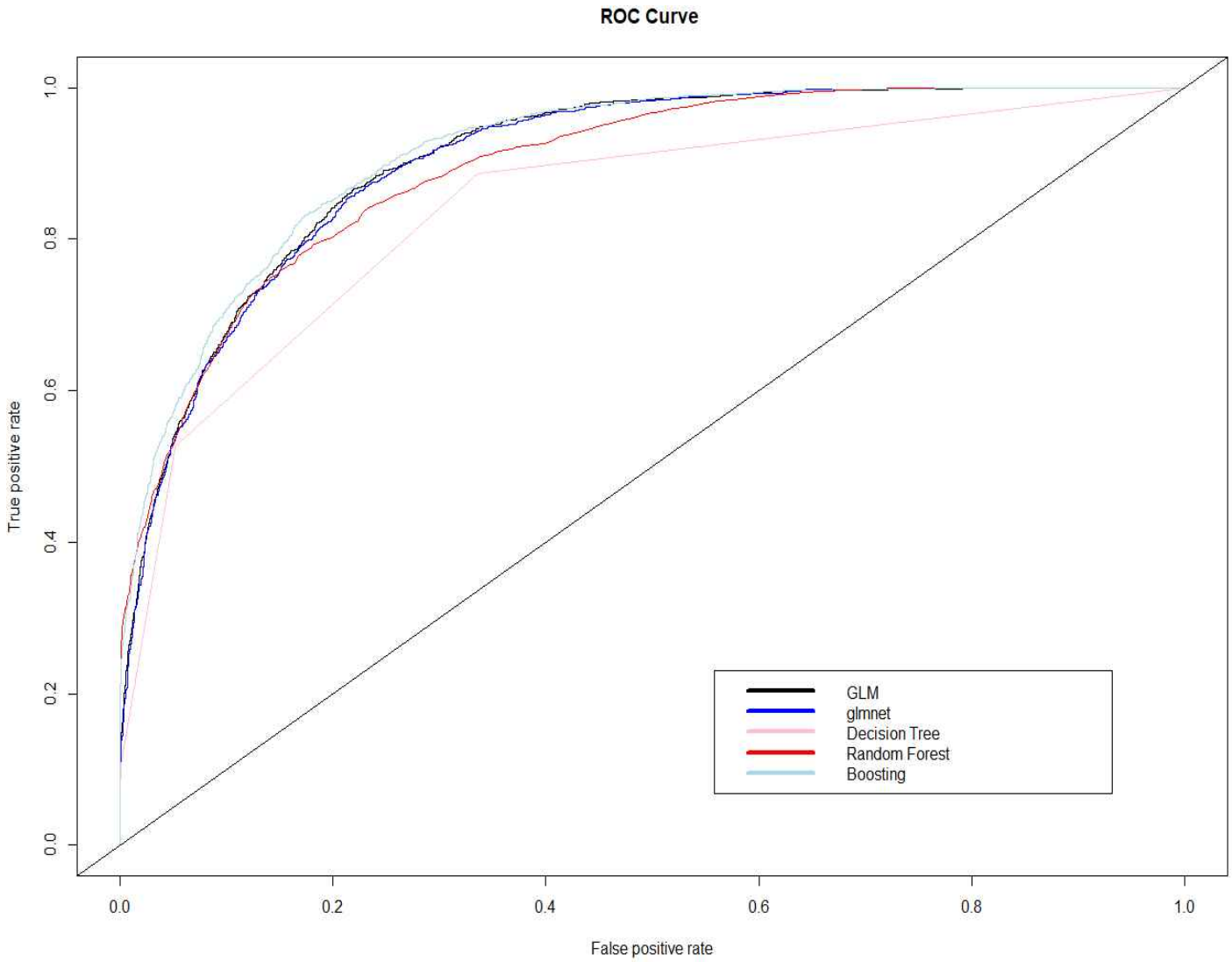
```
##############################################
##############################################

binomial_deviance(y_obs, yhat_lm)
binomial_deviance(y_obs, yhat_glmnet)
binomial_deviance(y_obs, yhat_tr)
binomial_deviance(y_obs, yhat_rf)
binomial_deviance(y_obs, yhat_gbm)

performance(pred_lm, "auc")@y.values[[1]]
performance(pred_glmnet, "auc")@y.values[[1]]
performance(pred_tr, "auc")@y.values[[1]]
performance(pred_rf, "auc")@y.values[[1]]
performance(pred_gbm, "auc")@y.values[[1]]
```











References

* 기계학습 기반 사회보장 빅데이터 분석 및 예측모형 연구(2017) (한국보건사회연구원 오미애)

* Big Data통계기법 수업 강의노트 (충남대학교 정보통계학과 이상인)

* Data Mining 수업 강의노트 (서울대 통계학과 김용대)

*실리콘밸리 데이터 과학자가 알려주는 따라하며 배우는 데이터 과학 (권재명)