20152410 배형준 Data Mining HW2

[Main text 2](#_Toc40842540)

[1. Construct (a) a naive Bayes classifier, (b) a classification tree classifier, and (c) a logistic regression classifier. 2](#_Toc40842541)

[2. For each classifier, make an ROC curve, calculate the AUC, and compare the three classifiers. 3](#_Toc40842542)

[3. For each classifier, find the optimal cutoff value to maximize the accuracy. Compare the three classifiers. 4](#_Toc40842543)

[4. For each classifier, find the optimal cutoff value to maximize the F1 score. Compare the three classifiers. 5](#_Toc40842544)

[5. Write your conclusions and discussion. 6](#_Toc40842545)

[Appendix : R codes 7](#_Toc40842546)

[1. Construct (a) a naive Bayes classifier, (b) a classification tree classifier, and (c) a logistic regression classifier. 7](#_Toc40842547)

[(a) a naive Bayes classifier 8](#_Toc40842548)

[(b) a classification tree classifier 9](#_Toc40842549)

[(c) a logistic regression classifier 12](#_Toc40842550)

[2. For each classifier, make an ROC curve, calculate the AUC, and compare the three classifiers. 13](#_Toc40842551)

[3. For each classifier, find the optimal cutoff value to maximize the accuracy. Compare the three classifiers. 16](#_Toc40842552)

[4. For each classifier, find the optimal cutoff value to maximize the F1 score. Compare the three classifiers. 19](#_Toc40842553)

# Main text

## 1. Construct (a) a naive Bayes classifier, (b) a classification tree classifier, and (c) a logistic regression classifier.

### (a) a naive Bayes classifier

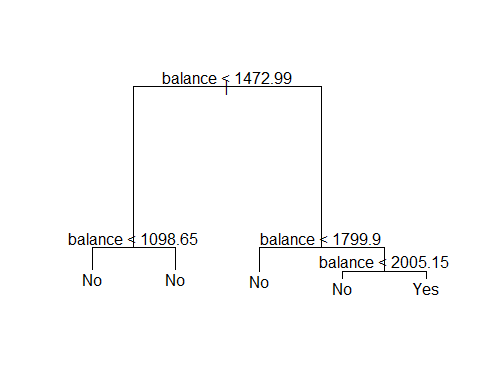
naiveBayes 함수를 이용하여 naïve Bayes classifier를 학습하였다. 모델의 기본 가정에 의해 student 변수는 ‘No’, ‘Yes’로 이뤄진 범주형 변수이므로 binomial dstn을 따른다고 가정했고 balance, income 변수는 연속형 변수이므로 각각 독립적인 normal dstn을 따른다고 가정했다. 코드 결과를 살펴보면 default의 값에 따라 student 변수의 binomial dstn의 추정된 분포를 확인할 수 있고, default 값에 따라 balance, income 변수의 normal dstn의 추정된 평균과 분산을 확인할 수 있다.

### (b) a classification tree classifier

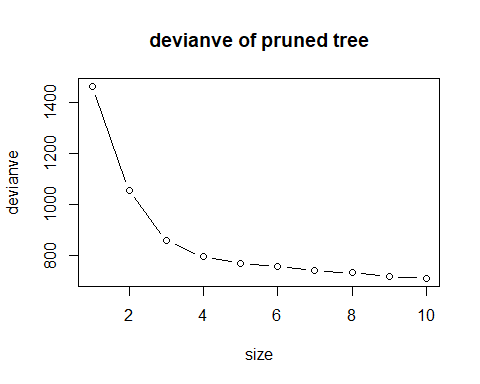
tree 함수를 사용하여 classification tree를 학습할 때 기준 통계량으로 deviance와 gini index으로 사용할 수 있어서 각각 학습해봤다.

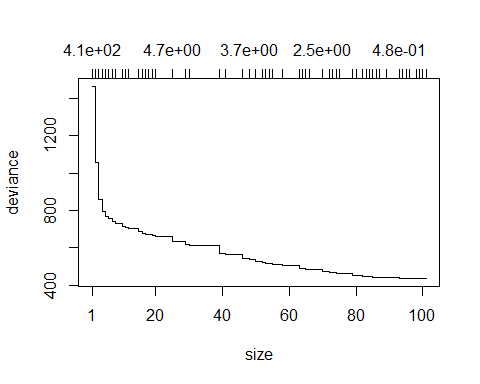
1. 기준 통계량으로 deviance 사용

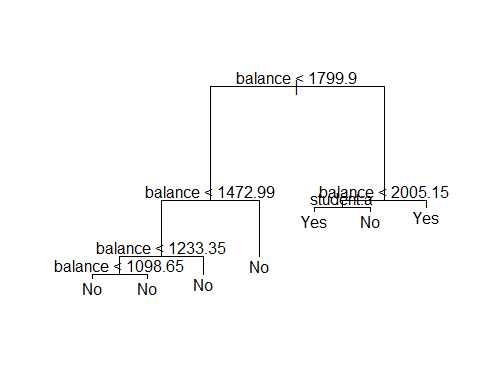
Deviance로 tree를 학습했을 때 tree의 규칙은 아래와 같다. 첫 번째 분할에선 balance 변수가 1472.99 미만 여부로 나눴고 아래에서도 balance 변수만 이용하여 tree를 학습하였다.



1. 기준 통계량으로 gini index 사용

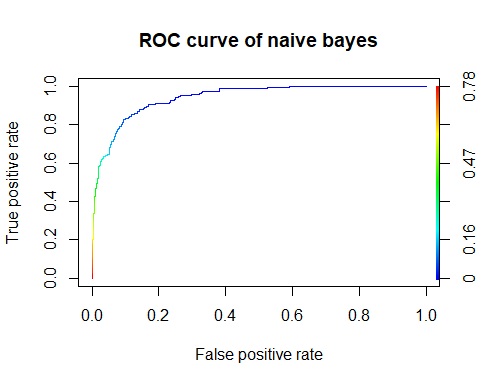
Gini index로 tree를 학습했을 때 노드의 개수가 101개까지 올라가서 과적합을 우려해 pruning을 해주었다. 아래 그래프는 tree의 노드 개수에 따른 deviance이다. Deviance가 급격하게 떨어진 이후엔 과적합이 의심되어 size가 커져도 deviance가 많이 감소하지 않는 size=4

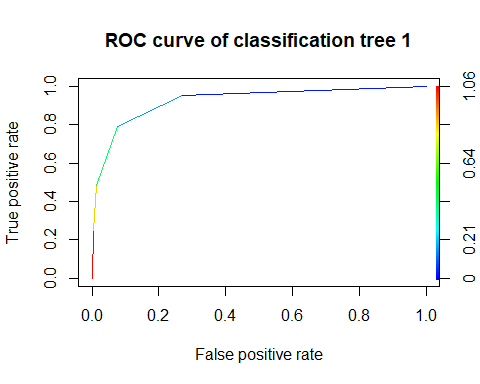


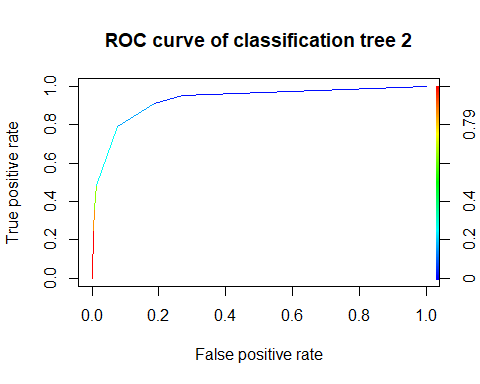


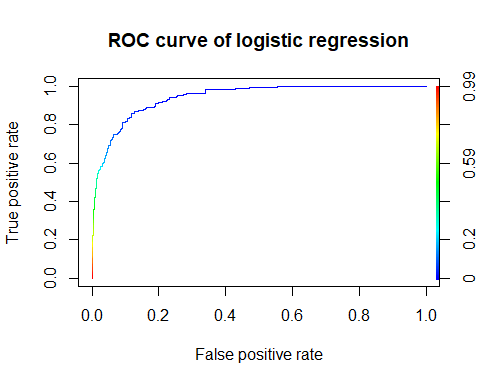
### (c) a logistic regression classifier

## 2. For each classifier, make an ROC curve, calculate the AUC, and compare the three classifiers.









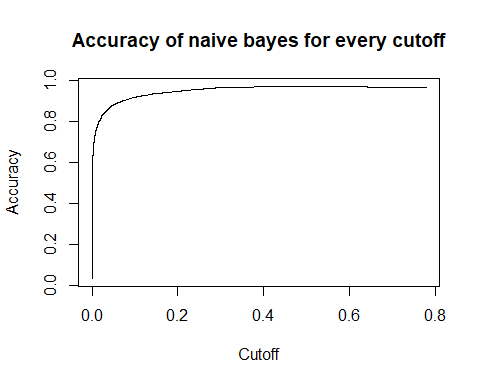
## AUC of naive bayes : 0.9409379

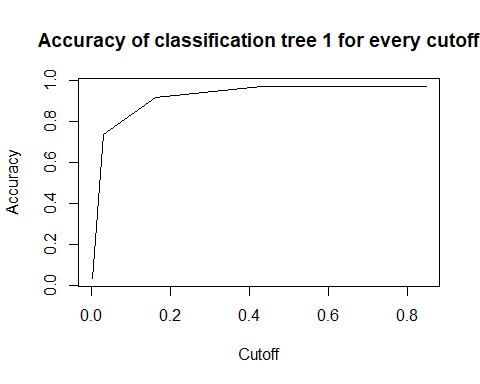
## AUC of classification tree 1 : 0.9248116

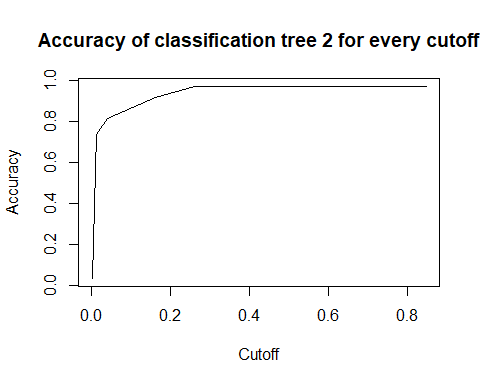
## AUC of classification tree 2 : 0.9277102

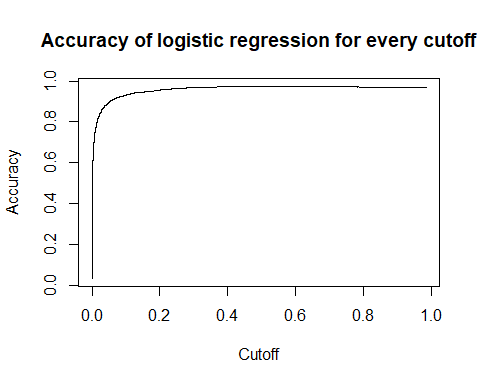
## AUC of logistic regression : 0.941824

## 3. For each classifier, find the optimal cutoff value to maximize the accuracy. Compare the three classifiers.









## Optimal cutoff of naive bayes : 0.4706127

## Maximum accuracy of naive bayes : 0.9734

## Optimal cutoff of classification tree 1 : 0.8478261

## Maximum accuracy of classification tree 1 : 0.9716

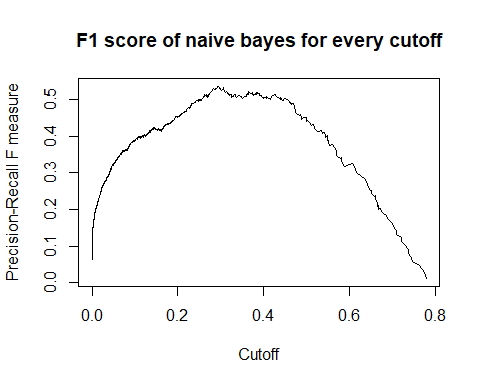
## Optimal cutoff of classification tree 2: 0.8478261

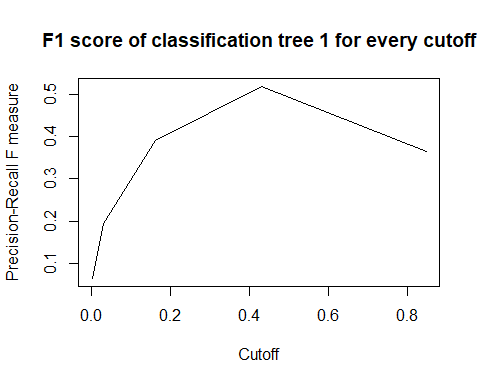
## Maximum accuracy of classification tree 2 : 0.9716

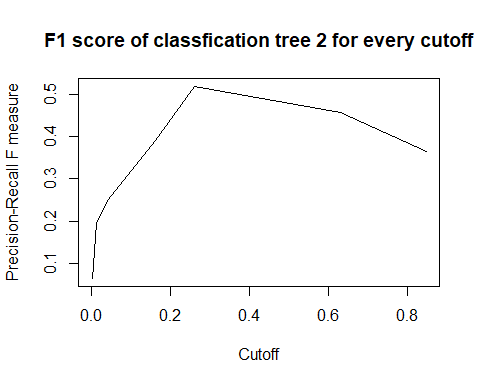
## Optimal cutoff of logistic regression : 0.5119097

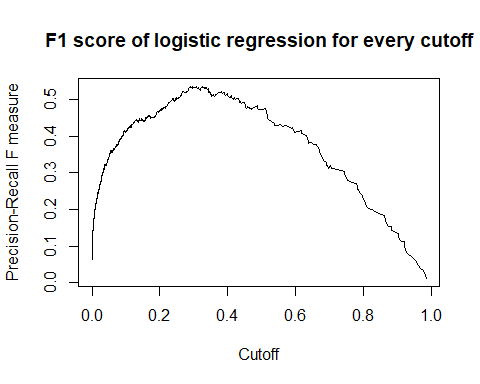
## Maximum accuracy of logistic regression : 0.9736

## 4. For each classifier, find the optimal cutoff value to maximize the F1 score. Compare the three classifiers.









## Optimal cutoff of naive bayes : 0.2946947

## Maximum F1 score of naive bayes : 0.5359116

## Optimal cutoff of classfication tree 1 : 0.43

## Maximum F1 score of classfication tree 1 : 0.5194805

## Optimal cutoff of classification tree 2 : 0.2592593

## Maximum F1 score of classfication tree 2 : 0.5194805

## Optimal cutoff of logistic regression : 0.2898158

## Maximum F1 score of logistic regression : 0.5360231

## 5. Write your conclusions and discussion.

# Appendix : R codes

## 1. Construct (a) a naive Bayes classifier, (b) a classification tree classifier, and (c) a logistic regression classifier.

library(ISLR)  
  
data = ISLR::Default  
  
n = dim(data)[1]  
train\_size = 0.5  
student = 20152410  
  
set.seed(student)  
train\_index = sample(1:n, n\*train\_size, replace=FALSE)  
train = data[train\_index, ]  
test = data[-train\_index, ]

### (a) a naive Bayes classifier

library(e1071)

model\_nb = naiveBayes(default ~ ., data=train)  
pred\_nb = predict(model\_nb, newdata=test, type='raw')  
model\_nb

##   
## Naive Bayes Classifier for Discrete Predictors  
##   
## Call:  
## naiveBayes.default(x = X, y = Y, laplace = laplace)  
##   
## A-priori probabilities:  
## Y  
## No Yes   
## 0.9666 0.0334   
##   
## Conditional probabilities:  
## student  
## Y No Yes  
## No 0.7020484 0.2979516  
## Yes 0.6706587 0.3293413  
##   
## balance  
## Y [,1] [,2]  
## No 804.1743 451.4752  
## Yes 1744.2139 309.7295  
##   
## income  
## Y [,1] [,2]  
## No 33336.39 13336.79  
## Yes 32960.67 13787.46

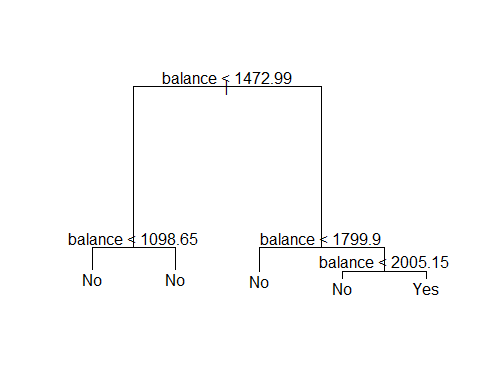
### (b) a classification tree classifier

library(tree)

# model\_tree1 : devianve를 기준으로 tree를 만듬  
model\_tree1 = tree(default ~ ., data=train, split='deviance')  
pred\_tree1 = predict(model\_tree1, newdata=test)  
summary(model\_tree1)

##   
## Classification tree:  
## tree(formula = default ~ ., data = train, split = "deviance")  
## Variables actually used in tree construction:  
## [1] "balance"  
## Number of terminal nodes: 5   
## Residual mean deviance: 0.1525 = 761.7 / 4995   
## Misclassification error rate: 0.027 = 135 / 5000

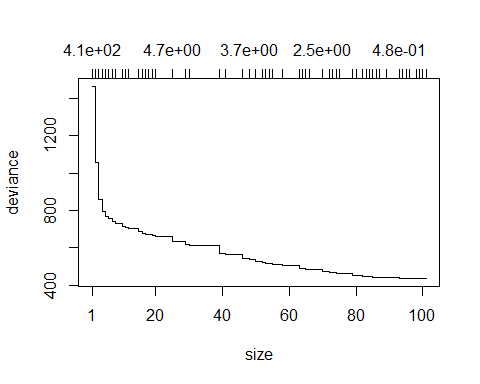
plot(model\_tree1)  
text(model\_tree1)



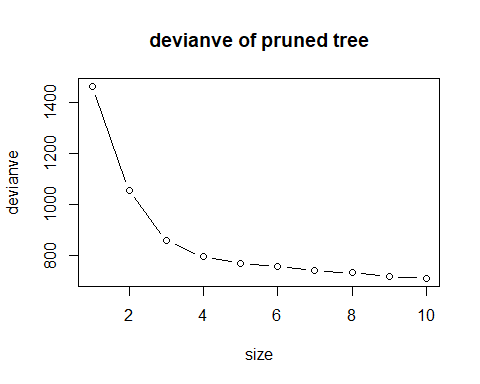
# model\_tree2 : gini index를 기준으로 tree를 만듬  
model\_tree2 = tree(default ~ ., data=train, split='gini')  
summary(model\_tree2)

##   
## Classification tree:  
## tree(formula = default ~ ., data = train, split = "gini")  
## Number of terminal nodes: 101   
## Residual mean deviance: 0.08913 = 436.7 / 4899   
## Misclassification error rate: 0.022 = 110 / 5000

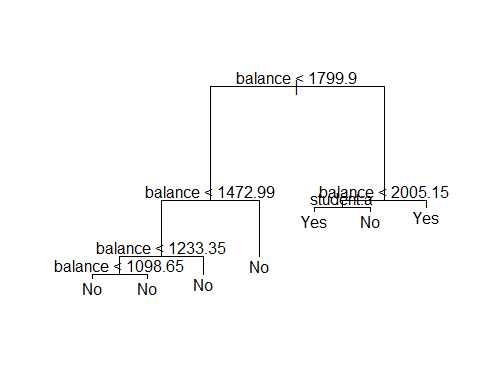
model\_ptree = prune.tree(model\_tree2, method='deviance')  
plot(model\_ptree)



len = length(model\_ptree$dev)  
len\_ = len - 9  
plot(10:1, model\_ptree$dev[len\_:len], xlab='size', ylab='devianve', type='b',  
 main='devianve of pruned tree')



# size가 4 이상부터는 devianve가 큰 변화를 가지지 않으므로 depth=4인 tree를 gini를 기준으로 만들도록 하겠다.  
model\_ptree\_ = prune.tree(model\_tree2, method='deviance', best=7)  
pred\_tree2 = predict(model\_ptree\_, newdata=test)  
plot(model\_ptree\_)  
text(model\_ptree\_)



### (c) a logistic regression classifier

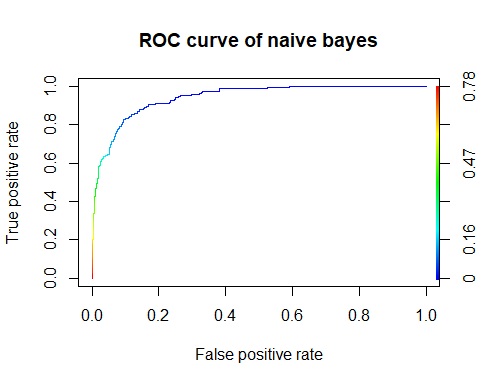
model\_logit = glm(default ~ ., data=train, family='binomial')  
pred\_logit = predict(model\_logit, newdata=test, type='response')  
summary(model\_logit)

##   
## Call:  
## glm(formula = default ~ ., family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.2136 -0.1307 -0.0465 -0.0166 3.8036   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.153e+01 7.333e-01 -15.726 < 2e-16 \*\*\*  
## studentYes -9.303e-01 3.333e-01 -2.791 0.00525 \*\*   
## balance 6.188e-03 3.555e-04 17.407 < 2e-16 \*\*\*  
## income 5.695e-06 1.141e-05 0.499 0.61767   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1463.7 on 4999 degrees of freedom  
## Residual deviance: 752.3 on 4996 degrees of freedom  
## AIC: 760.3  
##   
## Number of Fisher Scoring iterations: 8

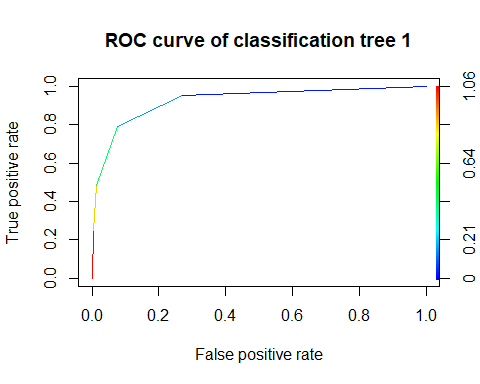
## 2. For each classifier, make an ROC curve, calculate the AUC, and compare the three classifiers.

library(ROCR)

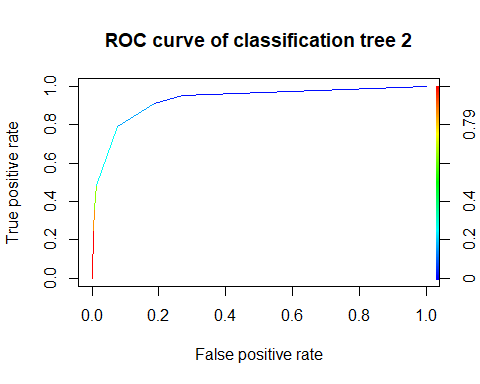
# ROC curve of 4 classifiers  
prediction\_nb = prediction(pred\_nb[, 'Yes'], test$default)  
performance\_nb = performance(prediction\_nb, 'tpr', 'fpr')  
plot(performance\_nb, main='ROC curve of naive bayes', colorize=TRUE)



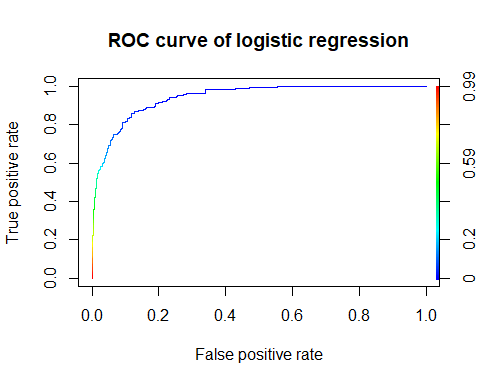
prediction\_tree1 = prediction(pred\_tree1[, 'Yes'], test$default)  
performance\_tree1 = performance(prediction\_tree1, 'tpr', 'fpr')  
plot(performance\_tree1, main='ROC curve of classification tree 1', colorize=TRUE)



prediction\_tree2 = prediction(pred\_tree2[, 'Yes'], test$default)  
performance\_tree2 = performance(prediction\_tree2, 'tpr', 'fpr')  
plot(performance\_tree2, main='ROC curve of classification tree 2', colorize=TRUE)



prediction\_logit = prediction(pred\_logit, test$default)  
performance\_logit = performance(prediction\_logit, 'tpr', 'fpr')  
plot(performance\_logit, main='ROC curve of logistic regression', colorize=TRUE)



# AUC of 4 classifiers  
auc\_nb = performance(prediction\_nb, 'auc')  
auc\_tree1 = performance(prediction\_tree1, 'auc')  
auc\_tree2 = performance(prediction\_tree2, 'auc')  
auc\_logit = performance(prediction\_logit, 'auc')  
  
cat('AUC of naive bayes : ', auc\_nb@y.values[[1]])

cat('AUC of classification tree 1 : ', auc\_tree1@y.values[[1]])

cat('AUC of classification tree 2 : ', auc\_tree2@y.values[[1]])

cat('AUC of logistic regression : ', auc\_logit@y.values[[1]])

## AUC of naive bayes : 0.9409379

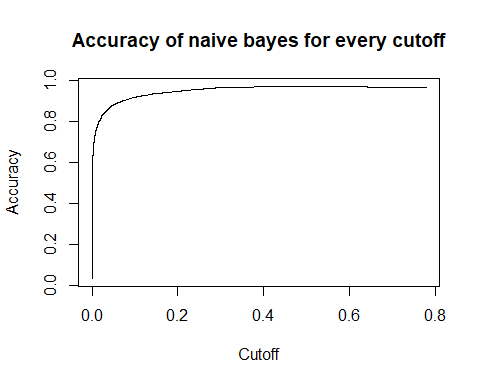
## AUC of classification tree 1 : 0.9248116

## AUC of classification tree 2 : 0.9277102

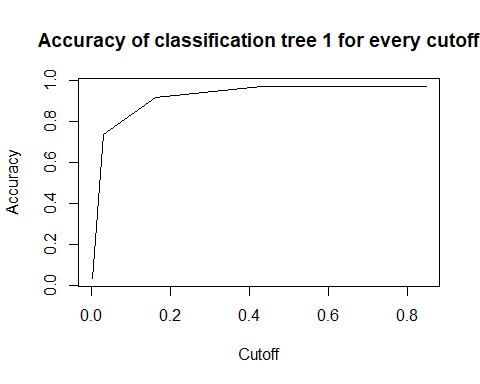
## AUC of logistic regression : 0.941824

## 3. For each classifier, find the optimal cutoff value to maximize the accuracy. Compare the three classifiers.

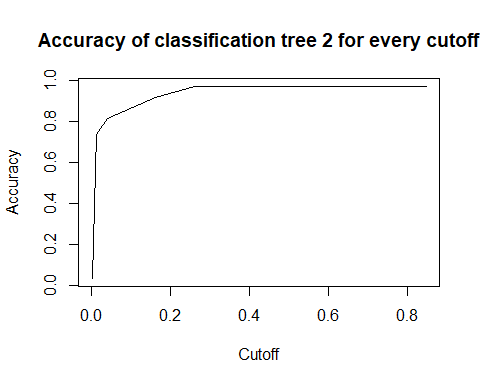
acc\_nb = performance(prediction\_nb, 'acc', 'cutoff')  
plot(acc\_nb, main='Accuracy of naive bayes for every cutoff')



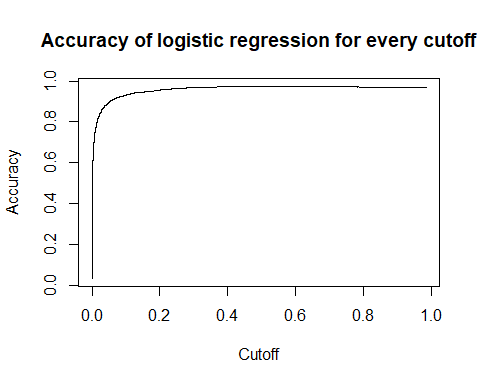
acc\_tree1 = performance(prediction\_tree1, 'acc', 'cutoff')  
plot(acc\_tree1, main='Accuracy of classification tree 1 for every cutoff')



acc\_tree2 = performance(prediction\_tree2, 'acc', 'cutoff')  
plot(acc\_tree2, main='Accuracy of classification tree 2 for every cutoff')



acc\_logit = performance(prediction\_logit, 'acc', 'cutoff')  
plot(acc\_logit, main='Accuracy of logistic regression for every cutoff')



cat('Optimal cutoff of naive bayes : ', acc\_nb@x.values[[1]][which.max(acc\_nb@y.values[[1]])])

cat('Maximum accuracy of naive bayes : ', max(acc\_nb@y.values[[1]]))

cat('Optimal cutoff of classification tree 1 : ', acc\_tree1@x.values[[1]][which.max(acc\_tree1@y.values[[1]])])

cat('Maximum accuracy of classification tree 1 : ', max(acc\_tree1@y.values[[1]]))

cat('Optimal cutoff of classification tree 2: ', acc\_tree2@x.values[[1]][which.max(acc\_tree2@y.values[[1]])])

cat('Maximum accuracy of classification tree 2 : ', max(acc\_tree2@y.values[[1]]))

cat('Optimal cutoff of logistic regression : ', acc\_logit@x.values[[1]][which.max(acc\_logit@y.values[[1]])])

cat('Maximum accuracy of logistic regression : ', max(acc\_logit@y.values[[1]]))

## Optimal cutoff of naive bayes : 0.4706127

## Maximum accuracy of naive bayes : 0.9734

## Optimal cutoff of classification tree 1 : 0.8478261

## Maximum accuracy of classification tree 1 : 0.9716

## Optimal cutoff of classification tree 2: 0.8478261

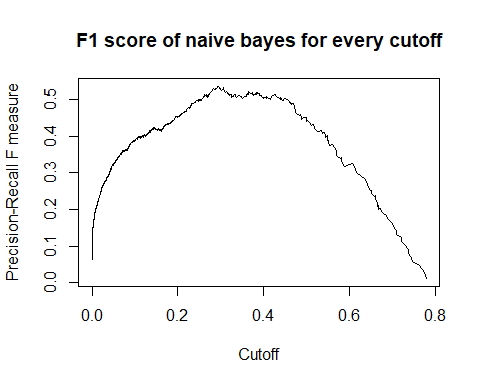
## Maximum accuracy of classification tree 2 : 0.9716

## Optimal cutoff of logistic regression : 0.5119097

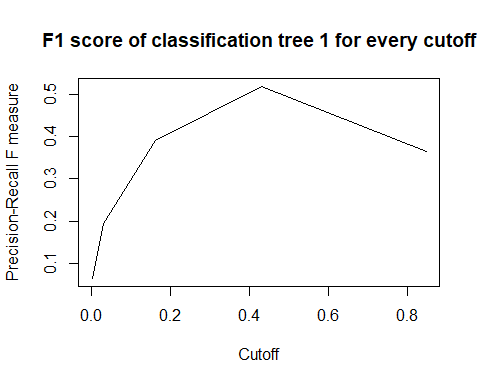
## Maximum accuracy of logistic regression : 0.9736

## 4. For each classifier, find the optimal cutoff value to maximize the F1 score. Compare the three classifiers.

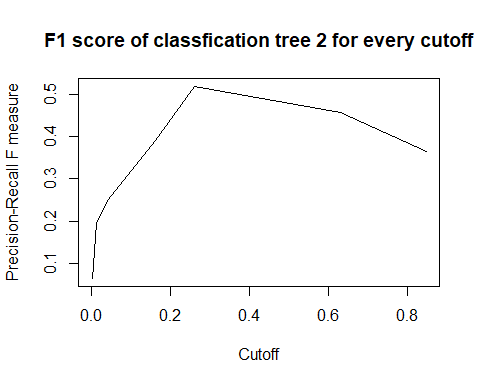
f1\_nb = performance(prediction\_nb, 'f', 'cutoff')  
plot(f1\_nb, main='F1 score of naive bayes for every cutoff')



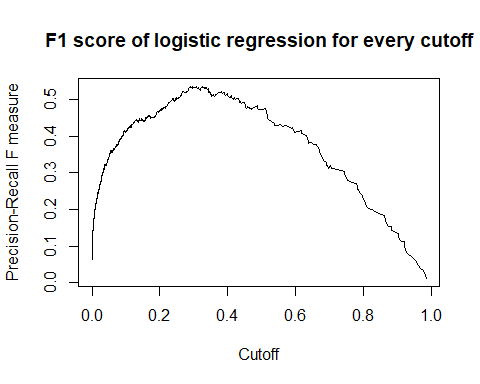
f1\_tree1 = performance(prediction\_tree1, 'f', 'cutoff')  
plot(f1\_tree1, main='F1 score of classification tree 1 for every cutoff')



f1\_tree2 = performance(prediction\_tree2, 'f', 'cutoff')  
plot(f1\_tree2, main='F1 score of classfication tree 2 for every cutoff')



f1\_logit = performance(prediction\_logit, 'f', 'cutoff')  
plot(f1\_logit, main='F1 score of logistic regression for every cutoff')



cat('Optimal cutoff of naive bayes : ', f1\_nb@x.values[[1]][which.max(f1\_nb@y.values[[1]])])

cat('Maximum F1 score of naive bayes : ', max(f1\_nb@y.values[[1]], na.rm=TRUE))

cat('Optimal cutoff of classfication tree 1 : ', f1\_tree1@x.values[[1]][which.max(f1\_tree1@y.values[[1]])])

cat('Maximum F1 score of classfication tree 1 : ', max(f1\_tree1@y.values[[1]], na.rm=TRUE))

cat('Optimal cutoff of classification tree 2 : ', f1\_tree2@x.values[[1]][which.max(f1\_tree2@y.values[[1]])])

cat('Maximum F1 score of classfication tree 2 : ', max(f1\_tree2@y.values[[1]], na.rm=TRUE))

cat('Optimal cutoff of logistic regression : ', f1\_logit@x.values[[1]][which.max(f1\_logit@y.values[[1]])])

cat('Maximum F1 score of logistic regression : ', max(f1\_logit@y.values[[1]], na.rm=TRUE))

## Optimal cutoff of naive bayes : 0.2946947

## Maximum F1 score of naive bayes : 0.5359116

## Optimal cutoff of classfication tree 1 : 0.43

## Maximum F1 score of classfication tree 1 : 0.5194805

## Optimal cutoff of classification tree 2 : 0.2592593

## Maximum F1 score of classfication tree 2 : 0.5194805

## Optimal cutoff of logistic regression : 0.2898158

## Maximum F1 score of logistic regression : 0.5360231