Homework 2: Classification Methods

Wen Li Teng

February 2 2020

```r
# Load packages
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------- tidyverse 1.2.1 --
## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
## -- Conflicts ---------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(rsample)
```

```
## Warning: package 'rsample' was built under R version 3.6.2
```

```r
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.6.2
## Loading required package: lattice
##
## Attaching package: 'caret'
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
library(ggplot2)
library(pROC)
```

```
## Warning: package 'pROC' was built under R version 3.6.2
## Type 'citation("pROC")' for a citation.
##
## Attaching package: 'pROC'
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```r
library(class)

# Set seed
set.seed(1234)

# 5(a) Split the data into a training and test set
data <- read.csv("mental_health.csv")
data <- na.omit(data)

split <- initial_split(data, prop = 0.7)
train <- training(split)
test <- testing(split)

# 5(b) Estimate models with vote96 as the response variable

# 5(b)(i) Logistic regression model
logit <- glm(vote96 ~ mhealth_sum + age + educ + black + female + married + inc10,
             data = data, family = binomial)

# 5(b)(ii) Linear discriminant model
ldm <- MASS::lda(vote96 ~ mhealth_sum + age + educ + black + female + married + inc10,
                 data = data, family = binomial)

# 5(b)(iii) Quadratic discriminant model
qdm <- MASS::qda(vote96 ~ mhealth_sum + age + educ + black + female + married + inc10,
                 data = data, family = binomial)

# 5(b)(iv) Naive Bayes

features <- (setdiff(names(train), "vote96"))
x <- (train[, features])
y <- as.factor(train$vote96)

train_control <- trainControl(
  method = "cv",
  sampling = "up",
  number = 10
)

naive_bayes <- train(
  x = x,
  y = y,
  method = "nb",
  trControl = train_control
)
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 3

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 67

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
```

```
## observation 16

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 54

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 5

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 10

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 14

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 64

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 74

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 20

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 71

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 17

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 49

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 29

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 30

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 59

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 80
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 10

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 25

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 47

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 64
```

```r
# 5(b)(v) K-nearest neighbors

mse_knn <- tibble(k = 1:10,
                  knn_train = map(k, ~ class::knn(select(train, -vote96),
                                                  test = select(train, -vote96),
                                                  cl = train$vote96, k = .)),
                  knn_test = map(k, ~ class::knn(select(train, -vote96),
                                                 test = select(test, -vote96),
                                                 cl = train$vote96, k = .)),
                  err_train = map_dbl(knn_train, ~ mean(test$vote96 != .)),
                  err_test = map_dbl(knn_test, ~ mean(test$vote96 != .)))
```

```
## Warning in `!=.default`(test$vote96, .): longer object length is not a
## multiple of shorter object length

## Warning in is.na(e1) | is.na(e2): longer object length is not a multiple of
## shorter object length

## Warning in `!=.default`(test$vote96, .): longer object length is not a
## multiple of shorter object length

## Warning in is.na(e1) | is.na(e2): longer object length is not a multiple of
## shorter object length

## Warning in `!=.default`(test$vote96, .): longer object length is not a
## multiple of shorter object length

## Warning in is.na(e1) | is.na(e2): longer object length is not a multiple of
## shorter object length

## Warning in `!=.default`(test$vote96, .): longer object length is not a
## multiple of shorter object length

## Warning in is.na(e1) | is.na(e2): longer object length is not a multiple of
## shorter object length

## Warning in `!=.default`(test$vote96, .): longer object length is not a
## multiple of shorter object length
```

```
## Warning in is.na(e1) | is.na(e2): longer object length is not a multiple of
## shorter object length

## Warning in `!=.default`(test$vote96, .): longer object length is not a
## multiple of shorter object length

## Warning in is.na(e1) | is.na(e2): longer object length is not a multiple of
## shorter object length

## Warning in `!=.default`(test$vote96, .): longer object length is not a
## multiple of shorter object length

## Warning in is.na(e1) | is.na(e2): longer object length is not a multiple of
## shorter object length

## Warning in `!=.default`(test$vote96, .): longer object length is not a
## multiple of shorter object length

## Warning in is.na(e1) | is.na(e2): longer object length is not a multiple of
## shorter object length

## Warning in `!=.default`(test$vote96, .): longer object length is not a
## multiple of shorter object length

## Warning in is.na(e1) | is.na(e2): longer object length is not a multiple of
## shorter object length

## Warning in `!=.default`(test$vote96, .): longer object length is not a
## multiple of shorter object length

## Warning in is.na(e1) | is.na(e2): longer object length is not a multiple of
## shorter object length
```

```r
# 5(c) Calculate performance metrics

# 5(c)(i) Error rate

# Logistic regression model

with(summary(logit), 1 - deviance/null.deviance)
```

```
## [1] 0.1543051
```

```r
# Linear discriminant model

test_predicted_lda <- predict(ldm, newdata = test)
lda_cm <- table(test$vote96, test_predicted_lda$class)
lda_error <- as.data.frame(test) %>%
  mutate(lda.pred = (test_predicted_lda$class)) %>%
  drop_na() %>%
```

```r
  summarize(lda.error = mean(vote96 != lda.pred))

# Quadratic discriminant model

test_predicted_qda <- predict(qdm, newdata = test)
qda_cm <- table(test$vote96, test_predicted_qda$class)
qda_error <- as.data.frame(test) %>%
  mutate(qda.pred = (test_predicted_qda$class)) %>%
  summarize(qda.error = mean(vote96 != qda.pred))

# Naive Bayes
confusionMatrix(naive_bayes)
```

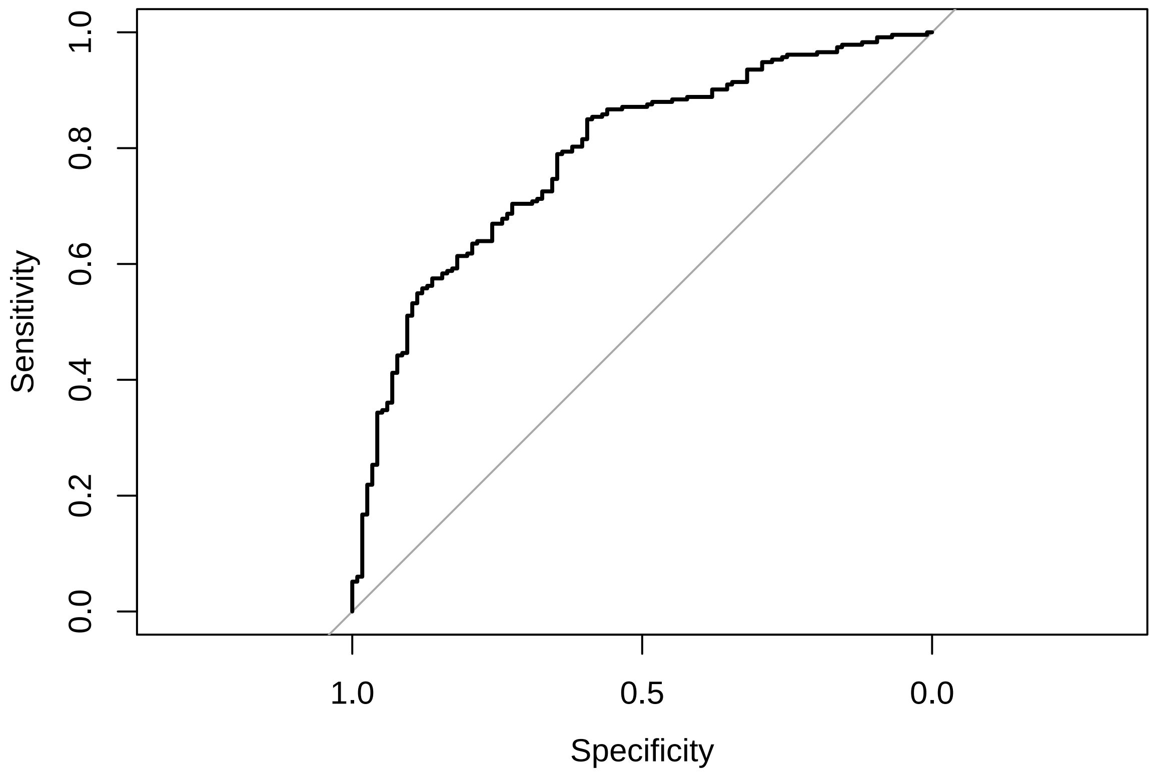
```
## Cross-Validated (10 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##           Reference
## Prediction    0    1
##          0 21.8 22.5
##          1 10.3 45.3
##
##  Accuracy (average) : 0.6716
```

```r
# K-nearest neighbors

knn_error <- select(mse_knn, k, err_train, err_test)

# 5(c)(ii) ROC curve(s)/Area under the curve (AUC)

# Logistic regression model
logit_pred <- predict(logit, newdata = test, type = "response")
test <- cbind(test, logit_pred)
logit_roc_obj <- roc(test$vote96, test$logit_pred)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```r
plot(logit_roc_obj)
```



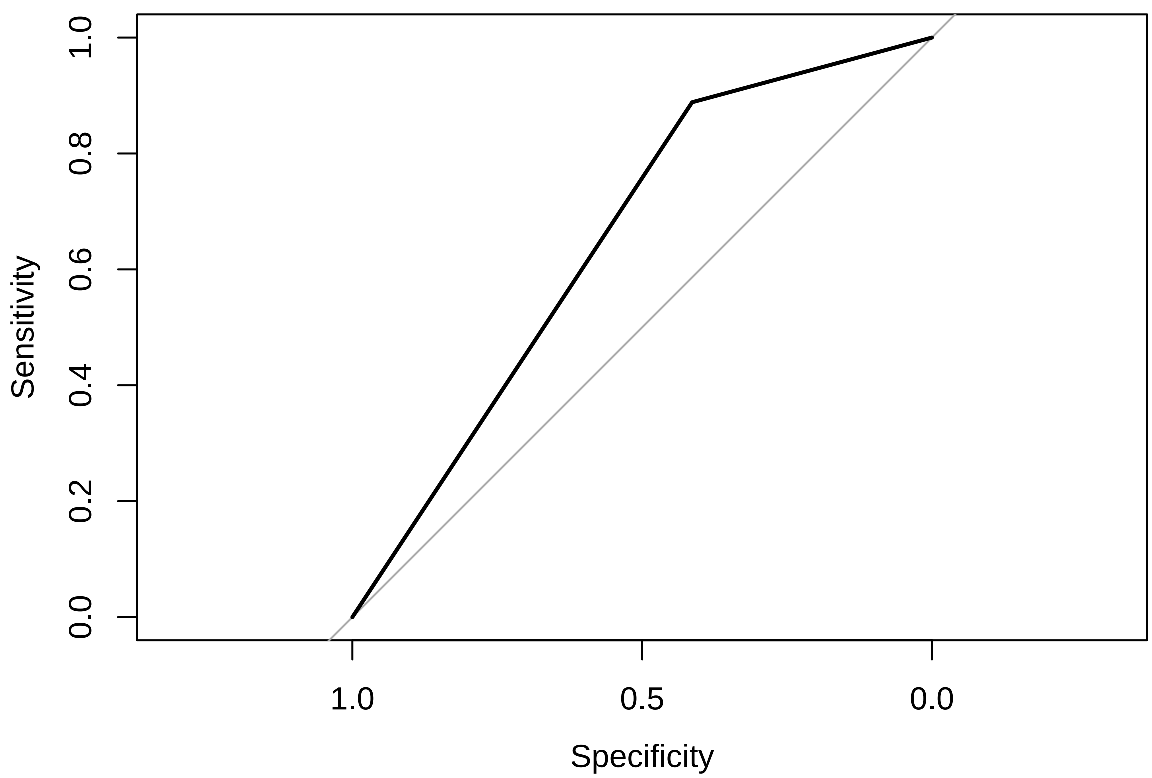

```
auc(logit_roc_obj)
```

```
## Area under the curve: 0.7867
```

```
# Linear discriminant model
ldm_pred <- predict(ldm, newdata = test)
test <- as.data.frame(test) %>%
  mutate(lda.pred = (ldm_pred$class))
ldm_roc_obj <- roc(test$vote96, as.numeric(test$lda.pred))
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
plot(ldm_roc_obj)
```



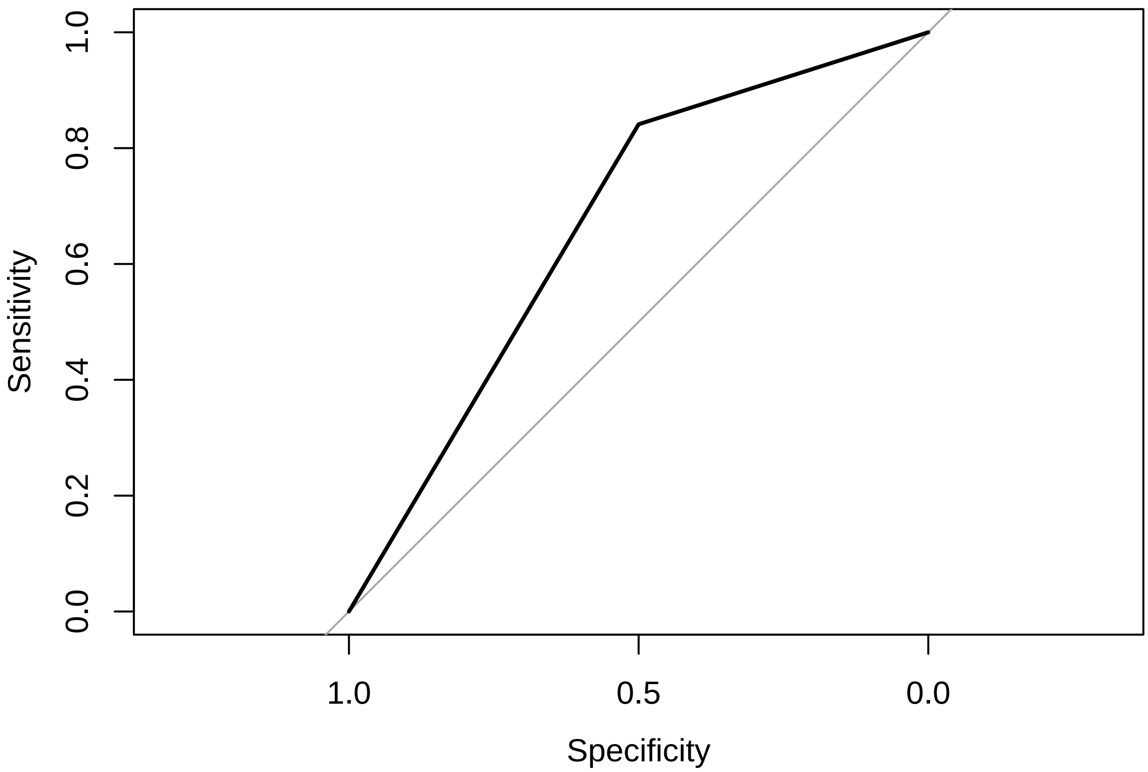

```
auc(ldm_roc_obj)
```

```
## Area under the curve: 0.6511
```

```
# Quadratic discriminant model
qdm_pred <- predict(qdm, newdata = test)
test <- as.data.frame(test) %>%
  mutate(qda.pred = (qdm_pred$class))
qdm_roc_obj <- roc(test$vote96, as.numeric(test$qda.pred))
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
plot(qdm_roc_obj)
```



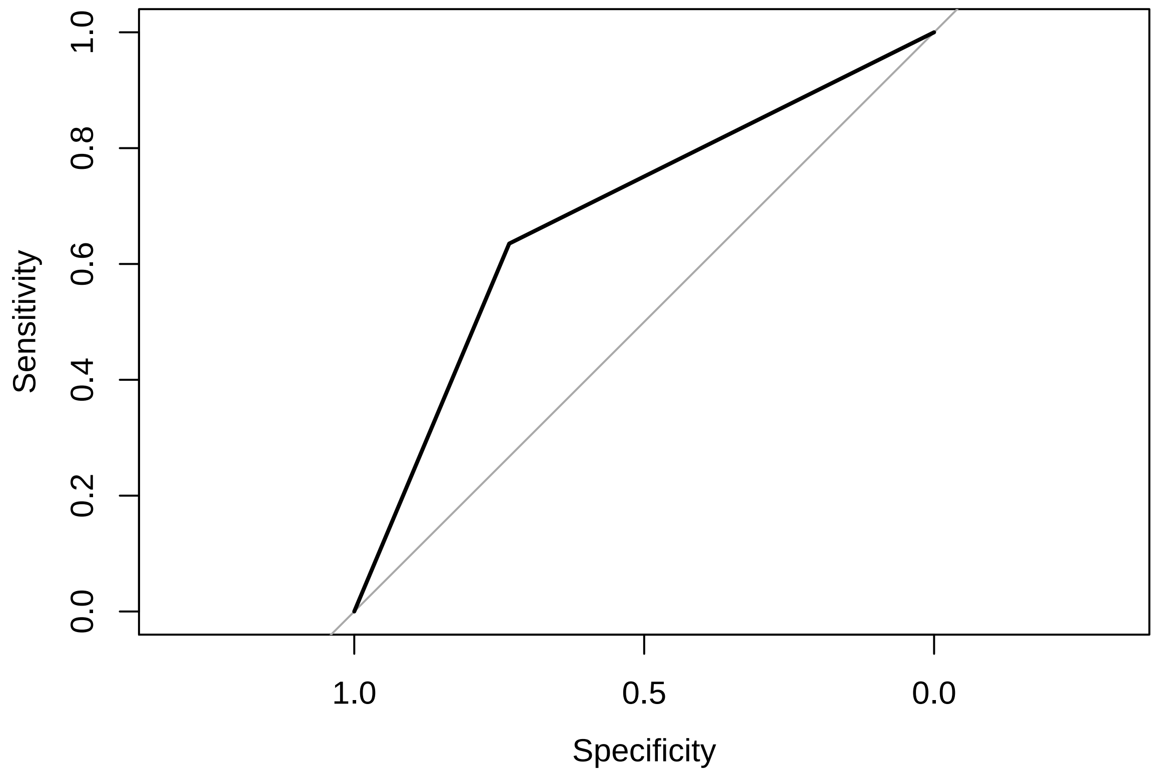

```r
auc(qdm_roc_obj)
```

```
## Area under the curve: 0.6706
```

```r
# Naive Bayes
nb_pred <- predict(naive_bayes, newdata = test)
test <- as.data.frame(test) %>%
  mutate(nb.pred = (nb_pred))
nb_roc_obj <- roc(test$vote96, as.numeric(test$nb.pred))
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```r
plot(nb_roc_obj)
```



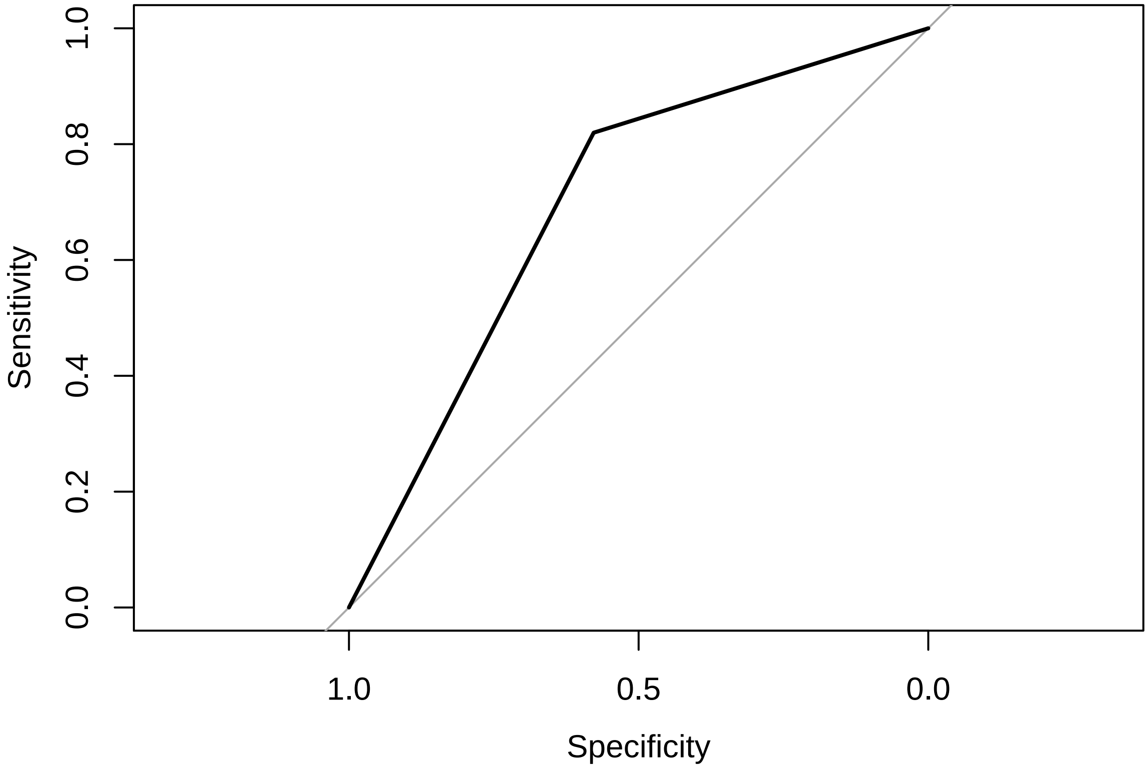

```
auc(nb_roc_obj)
```

```
## Area under the curve: 0.684
```

```
# K-nearest neighbors

test_new <- test[, 1:8]
knn_pred_1 <- knn(train = train, test = test_new, cl = train$vote96, k=1)
test <- as.data.frame(test) %>%
  mutate(knn.pred.1 = (knn_pred_1))
knn_1_roc_obj <- roc(test$vote96, as.numeric(test$knn.pred.1))
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
plot(knn_1_roc_obj)
```



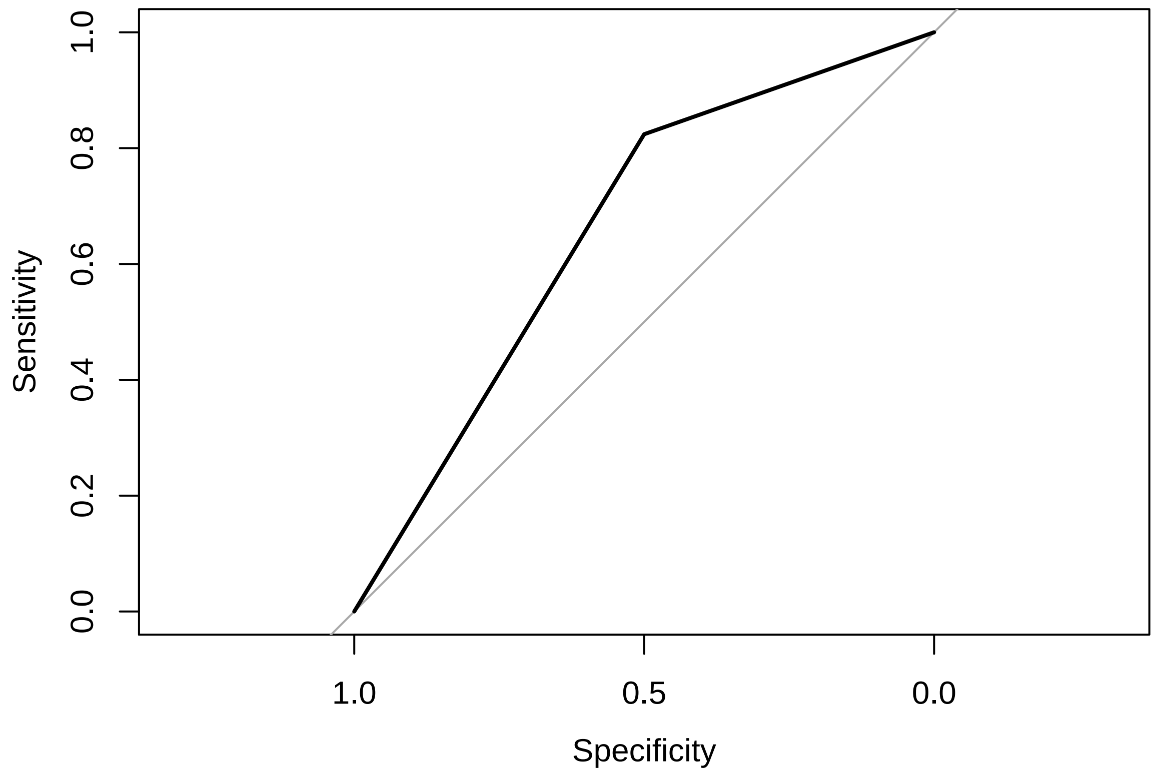

```
auc(knn_1_roc_obj)
```

```
## Area under the curve: 0.6987
```

```
test_new <- test[, 1:8]
knn_pred_2 <- knn(train = train, test = test_new, cl = train$vote96, k=2)
test <- as.data.frame(test) %>%
  mutate(knn.pred.2 = (knn_pred_2))
knn_2_roc_obj <- roc(test$vote96, as.numeric(test$knn.pred.2))
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
plot(knn_2_roc_obj)
```



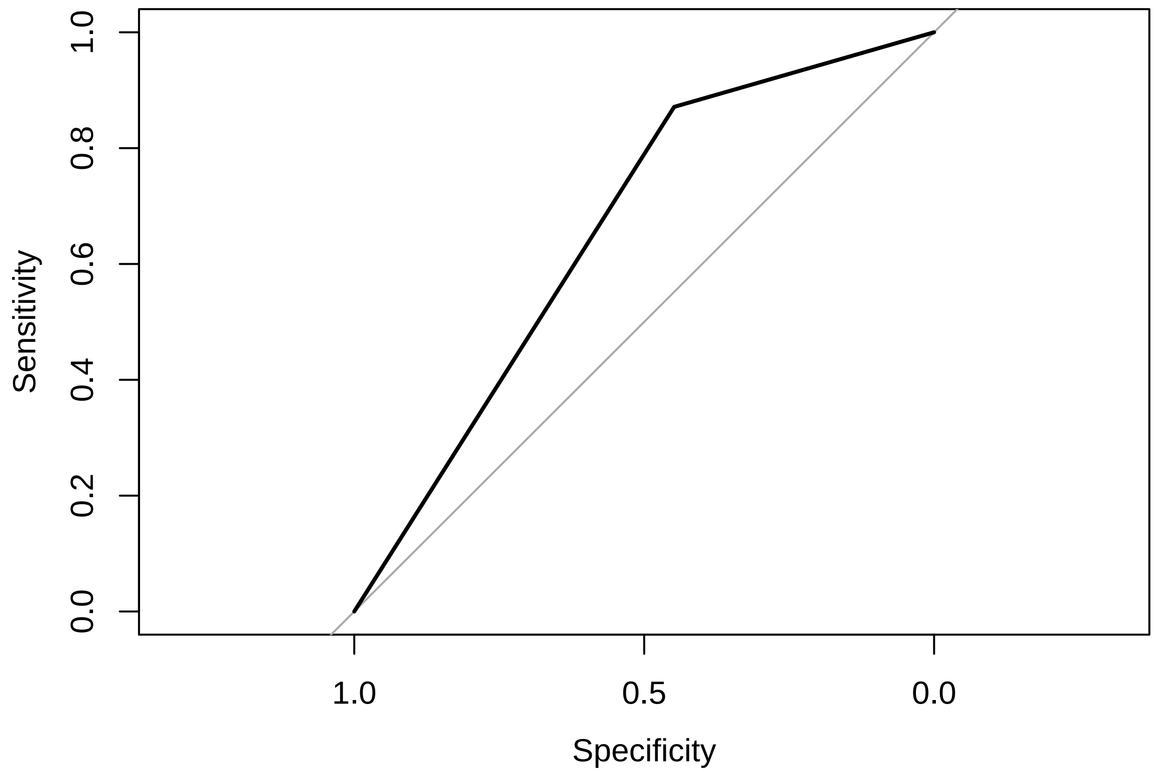

```r
auc(knn_2_roc_obj)
```

```
## Area under the curve: 0.662
```

```r
test_new <- test[, 1:8]
knn_pred_3 <- knn(train = train, test = test_new, cl = train$vote96, k=3)
test <- as.data.frame(test) %>%
  mutate(knn.pred.3 = (knn_pred_3))
knn_3_roc_obj <- roc(test$vote96, as.numeric(test$knn.pred.3))
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```r
plot(knn_3_roc_obj)
```



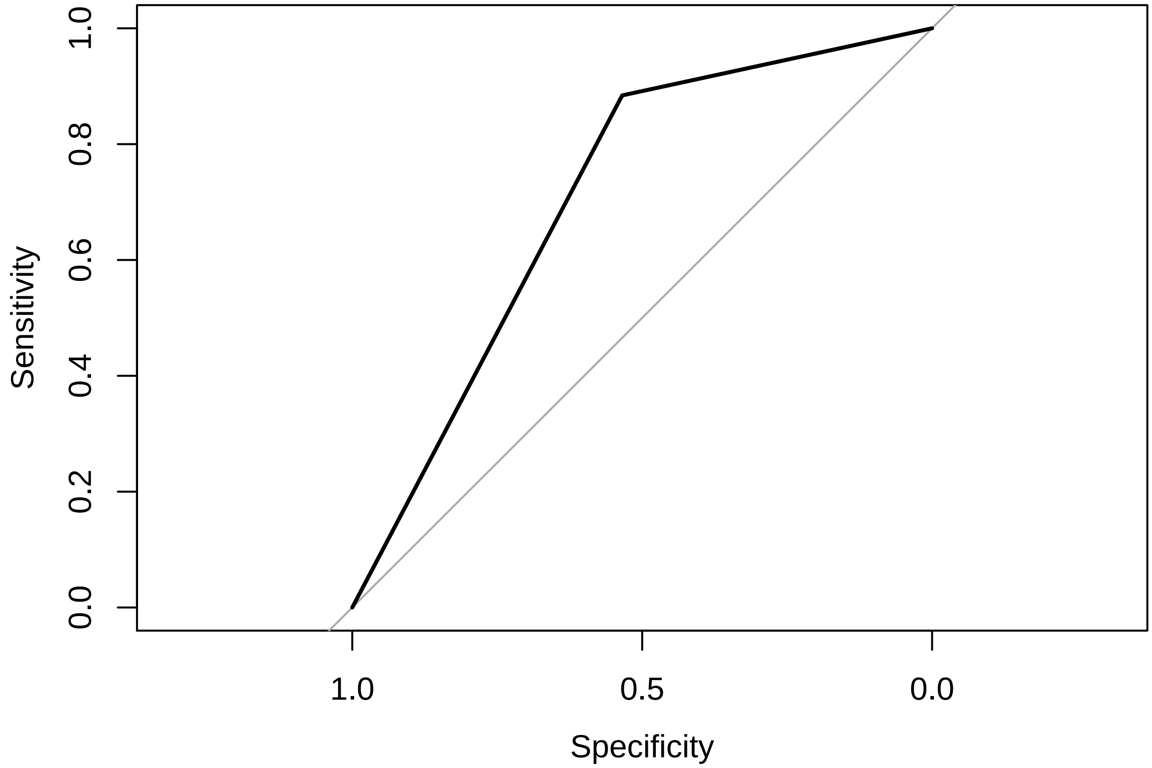

```r
auc(knn_3_roc_obj)
```

```
## Area under the curve: 0.6598
```

```r
test_new <- test[, 1:8]
knn_pred_4 <- knn(train = train, test = test_new, cl = train$vote96, k=4)
test <- as.data.frame(test) %>%
  mutate(knn.pred.4 = (knn_pred_4))
knn_4_roc_obj <- roc(test$vote96, as.numeric(test$knn.pred.4))
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```r
plot(knn_4_roc_obj)
```



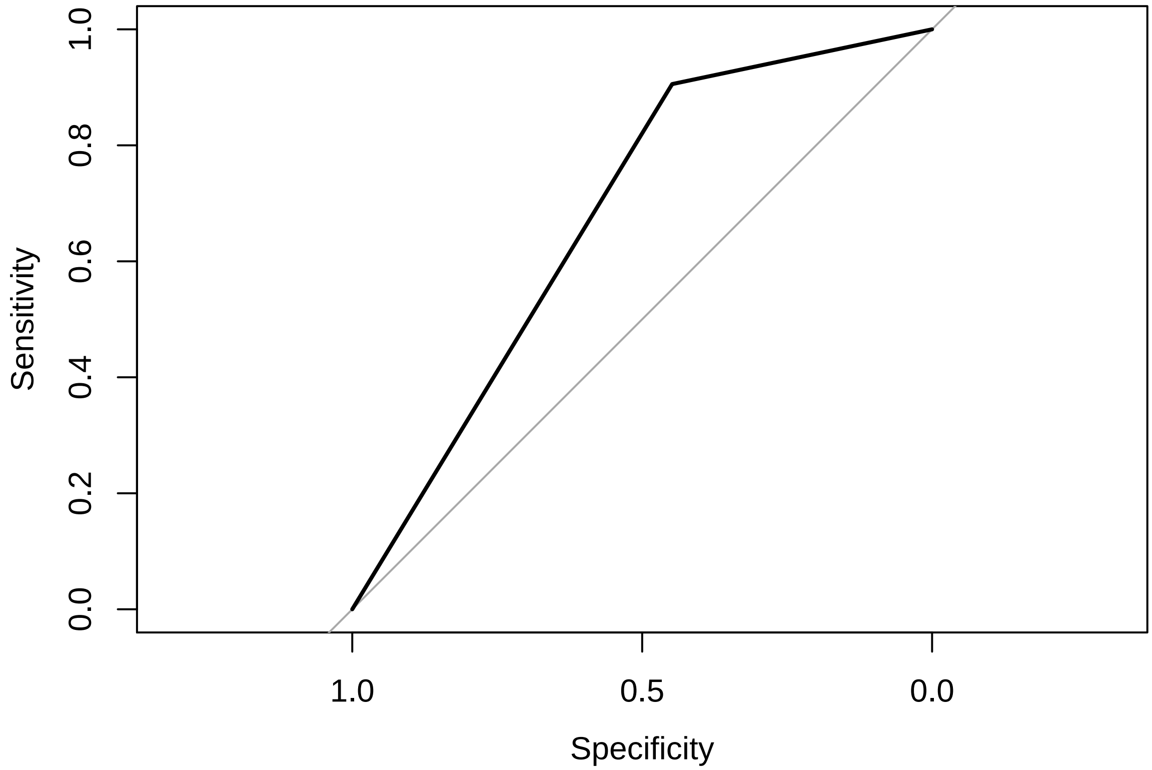

```r
auc(knn_4_roc_obj)
```

```
## Area under the curve: 0.7093
```

```r
test_new <- test[, 1:8]
knn_pred_5 <- knn(train = train, test = test_new, cl = train$vote96, k=5)
test <- as.data.frame(test) %>%
  mutate(knn.pred.5 = (knn_pred_5))
knn_5_roc_obj <- roc(test$vote96, as.numeric(test$knn.pred.5))
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```r
plot(knn_5_roc_obj)
```



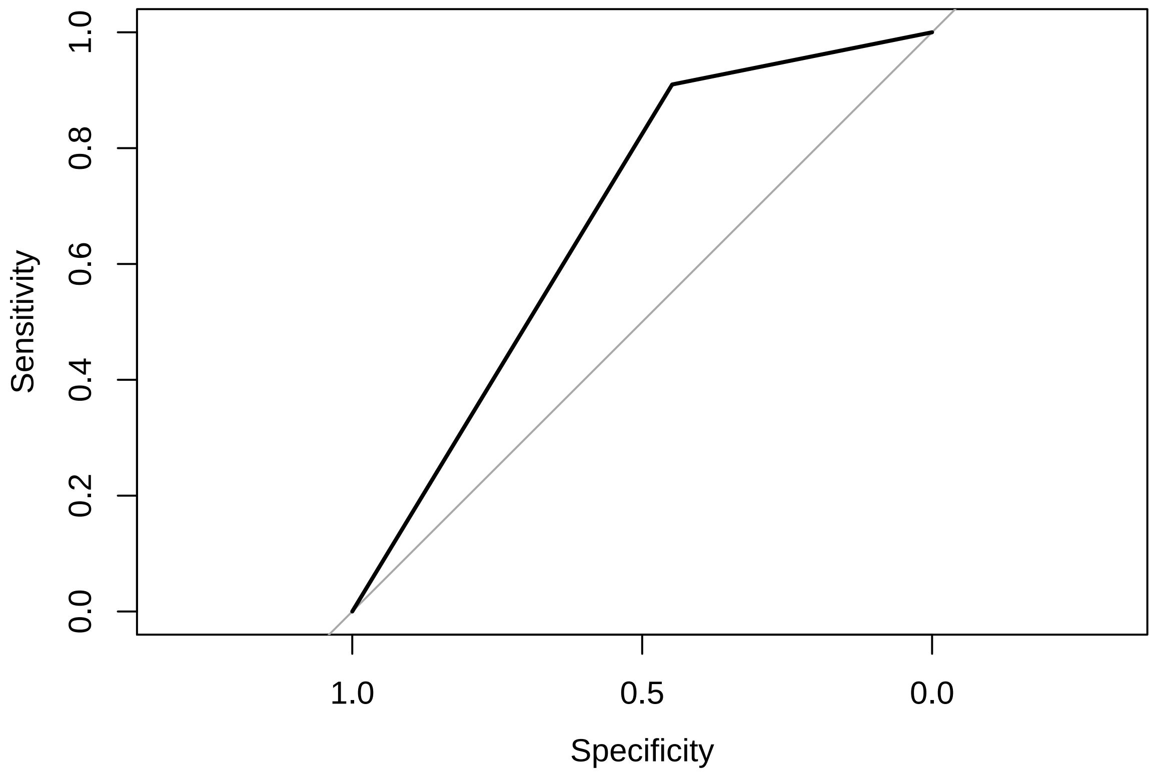

```
auc(knn_5_roc_obj)
```

```
## Area under the curve: 0.6769
```

```
test_new <- test[, 1:8]
knn_pred_6 <- knn(train = train, test = test_new, cl = train$vote96, k=6)
test <- as.data.frame(test) %>%
  mutate(knn.pred.6 = (knn_pred_6))
knn_6_roc_obj <- roc(test$vote96, as.numeric(test$knn.pred.6))
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
plot(knn_6_roc_obj)
```



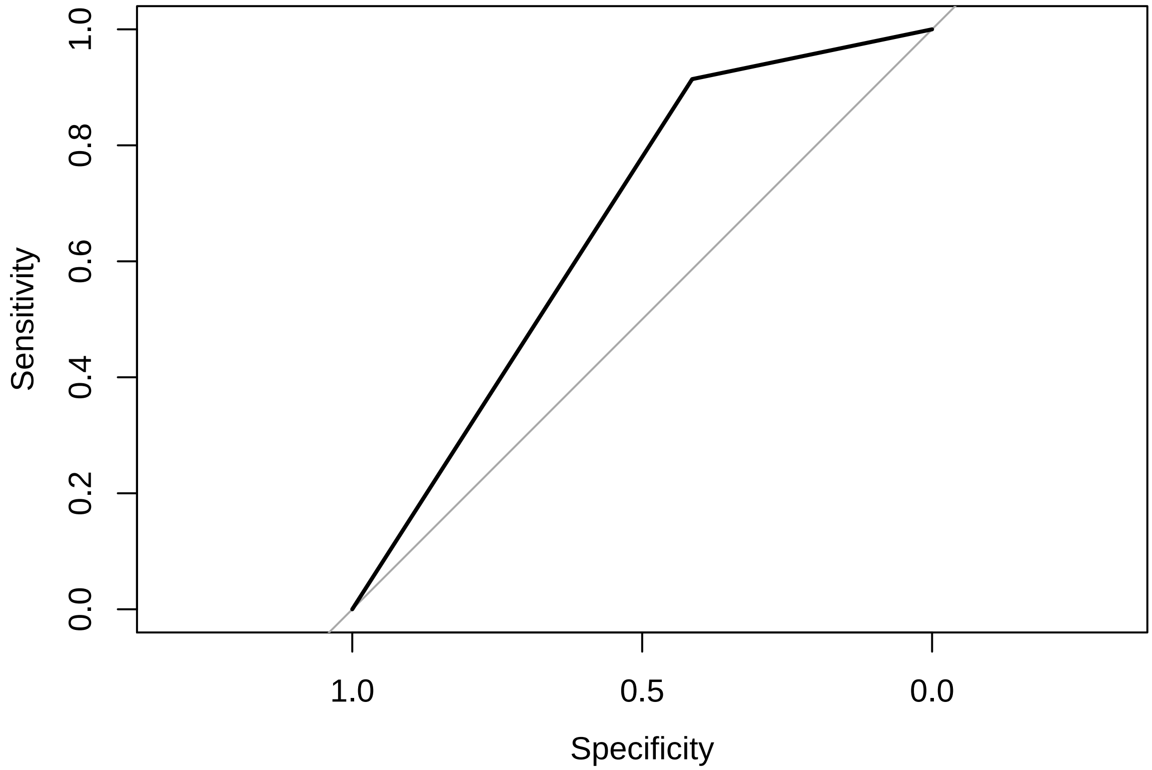

```r
auc(knn_6_roc_obj)
```

```
## Area under the curve: 0.6791
```

```r
test_new <- test[, 1:8]
knn_pred_7 <- knn(train = train, test = test_new, cl = train$vote96, k=7)
test <- as.data.frame(test) %>%
  mutate(knn.pred.7 = (knn_pred_7))
knn_7_roc_obj <- roc(test$vote96, as.numeric(test$knn.pred.7))
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```r
plot(knn_7_roc_obj)
```



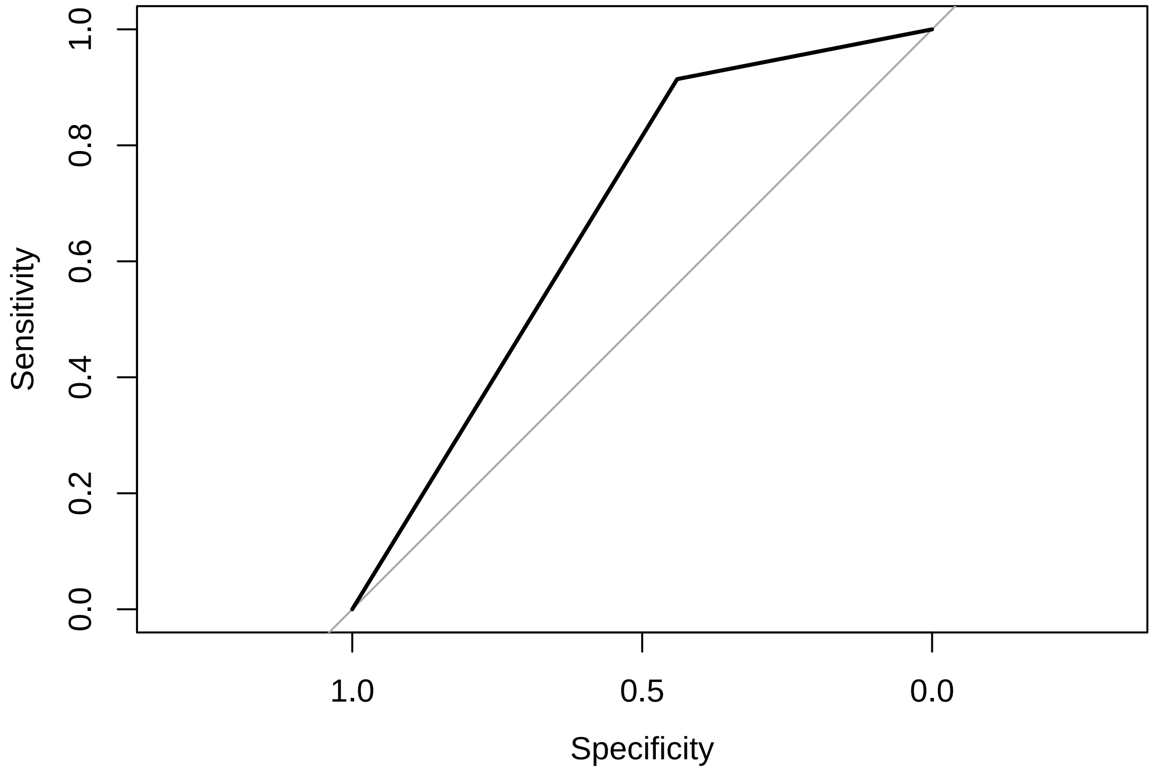

```
auc(knn_7_roc_obj)
```

```
## Area under the curve: 0.664
```

```
test_new <- test[, 1:8]
knn_pred_8 <- knn(train = train, test = test_new, cl = train$vote96, k=8)
test <- as.data.frame(test) %>%
  mutate(knn.pred.8 = (knn_pred_8))
knn_8_roc_obj <- roc(test$vote96, as.numeric(test$knn.pred.8))
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
plot(knn_8_roc_obj)
```



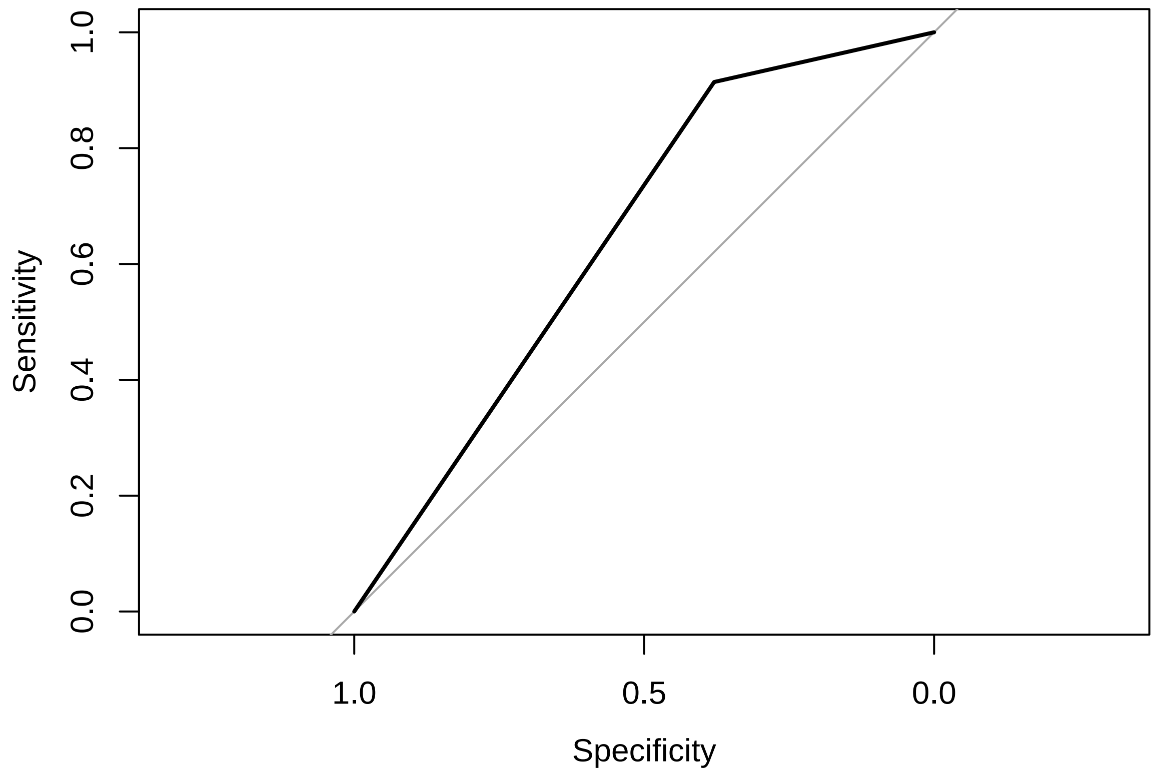

```r
auc(knn_8_roc_obj)
```

```
## Area under the curve: 0.6769
```

```r
test_new <- test[, 1:8]
knn_pred_9 <- knn(train = train, test = test_new, cl = train$vote96, k=9)
test <- as.data.frame(test) %>%
  mutate(knn.pred.9 = (knn_pred_9))
knn_9_roc_obj <- roc(test$vote96, as.numeric(test$knn.pred.9))
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```r
plot(knn_9_roc_obj)
```



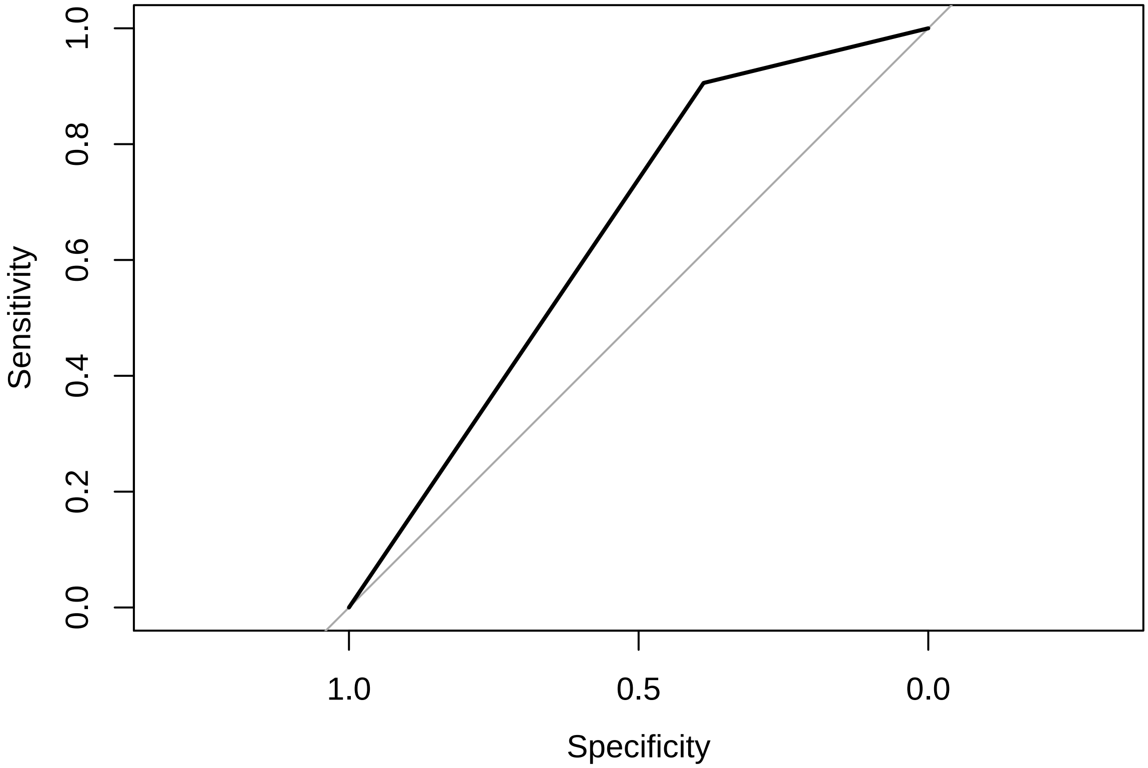

```
auc(knn_9_roc_obj)
```

```
## Area under the curve: 0.6467
```

```
test_new <- test[, 1:8]
knn_pred_10 <- knn(train = train, test = test_new, cl = train$vote96, k=10)
test <- as.data.frame(test) %>%
  mutate(knn.pred.10 = (knn_pred_10))
knn_10_roc_obj <- roc(test$vote96, as.numeric(test$knn.pred.10))
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
plot(knn_10_roc_obj)
```




```r
auc(knn_10_roc_obj)
```

```
## Area under the curve: 0.6468
```

```r
# 5(d) Best-performing model
```

Assessing the model performance using their error rates and areas under the curve, it appears that the logit model is the best.