Classification Performance

Nate Wells

Math 243: Stat Learning

October 2nd, 2020

Outline

In today's class, we will. . .

- Implement KNN in R
- Analyze the performance of classification models
- Work in groups on a classification problem

Section 1

KNN in R

KNN

Recall: The KNN model estimates the conditional probability $P(Y = A_j \mid X)$ as

$$P(Y = A_j \mid X = x_0) \approx \frac{1}{K} \sum_{i \in N_0} I(y_i = A_j)$$

KNN

Recall: The KNN model estimates the conditional probability $P(Y = A_j \mid X)$ as

$$P(Y = A_j \mid X = x_0) \approx \frac{1}{K} \sum_{i \in N_0} I(y_i = A_j)$$

In R, we use the `knn` function in the `class` library.

KNN

Recall: The KNN model estimates the conditional probability $P(Y = A_j \mid X)$ as

$$P(Y = A_j \mid X = x_0) \approx \frac{1}{K} \sum_{i \in N_0} I(y_i = A_j)$$

In R, we use the `knn` function in the `class` library.

The `knn` function fits a model **and** makes predictions all in one command.

KNN

Recall: The KNN model estimates the conditional probability $P(Y = A_j \mid X)$ as

$$P(Y = A_j \mid X = x_0) \approx \frac{1}{K} \sum_{i \in N_0} I(y_i = A_j)$$

In R, we use the `knn` function in the `class` library.

The `knn` function fits a model **and** makes predictions all in one command.

- Unlike the `lm` and `glm`, which first fit a model and then make predictions using the `predict` function

Simulated Data

Suppose $X|Y = 1 \sim N(0, 1)$ and $X|Y = 0 \sim N(1, 1)$, and that each class is of the same size.

Simulated Data

Suppose $X|Y = 1 \sim N(0, 1)$ and $X|Y = 0 \sim N(1, 1)$, and that each class is of the same size.

- We'll also subset our data into test and training sets.

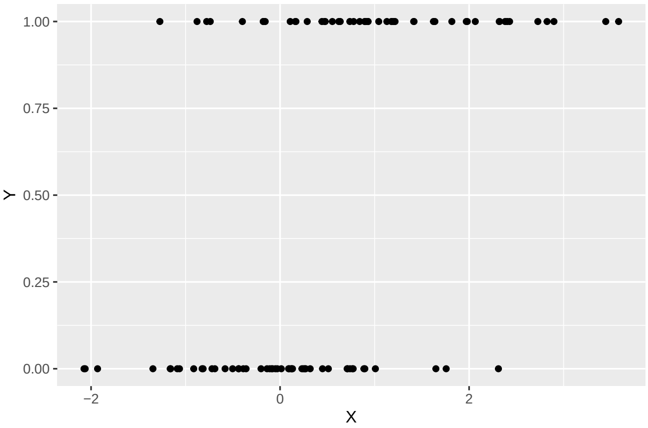
Simulated Data

Suppose $X|Y = 1 \sim N(0, 1)$ and $X|Y = 0 \sim N(1, 1)$, and that each class is of the same size.

- We'll also subset our data into test and training sets.

```
set.seed(100)
n<-100

Y<-rep(c(0,1), c(n/2, n/2))
X<-c(rnorm(n/2, 0, 1), rnorm(n/2, 1, 1) )

d<-data.frame(X,Y)

library(dplyr)
train_d<-d %>% sample_frac(.75)
test_d<-anti_join(d, train_d)
```

Scatterplot



The KNN Model

The `knn` function takes 4 arguments.

1. A data frame containing the predictors associated to the training data
2. A data frame containing the predictors associated to the test data
3. A vector containing the response associated to the training data
4. A value for K, the number of nearest neighbors.

The KNN Model

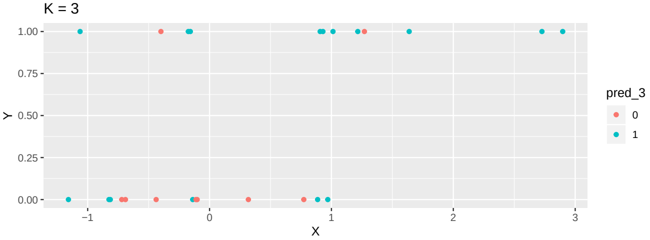
The `knn` function takes 4 arguments.

1. A data frame containing the predictors associated to the training data
2. A data frame containing the predictors associated to the test data
3. A vector containing the response associated to the training data
4. A value for K, the number of nearest neighbors.

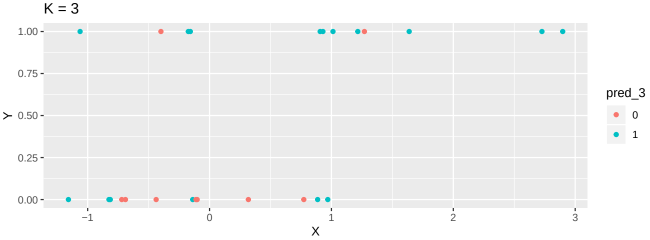
```
library(class)
set.seed(200)

pred_3<-knn(train_d %>% select(X),
    test_d %>% select(X),
    train_d$Y,
    3)

pred_3
```

```
##  [1] 1 0 1 0 0 1 0 1 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1 1 1
## Levels: 0 1
```

Results $K = 3$

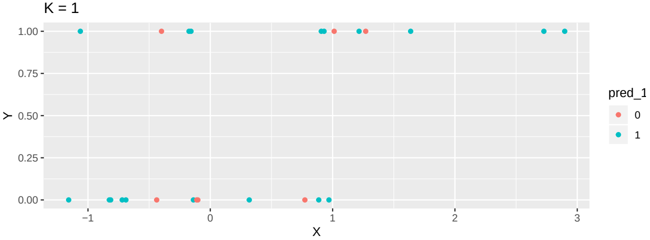


Results $K = 3$

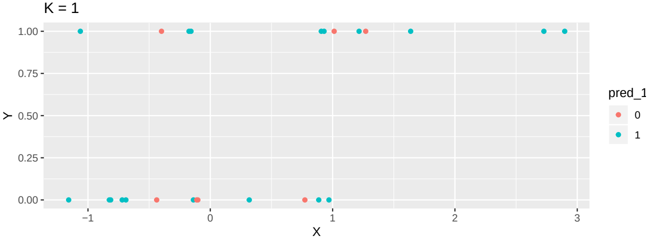



```
##
##      0  1
##   0  7  6
##   1  2 10

## [1] 0.68
```

Results $K = 1$



Results $K = 1$



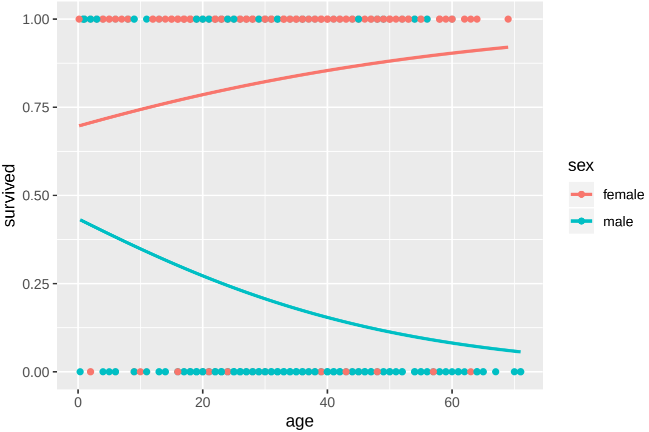

```
##
##      0 1
##    0 4 9
##    1 3 9

## [1] 0.52
```

Section 2

Model Performance

The Unsinkable Example

The `Titanic` data set contains information on passengers of the *Titanic*



A better confusion matrix

The confusionMatrix function in the caret package provides a confusion matrix along withe the relevant statistics:

```
library(caret)
confusionMatrix(data =   factor(preds) , reference =  factor(Titanic1$survived)  )
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 308  82
##          1  44 199
##
##                Accuracy : 0.8009
##                  95% CI : (0.7677, 0.8314)
##     No Information Rate : 0.5561
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5912
##
##  Mcnemar's Test P-Value : 0.0009799
##
##             Sensitivity : 0.8750
##             Specificity : 0.7082
##          Pos Pred Value : 0.7897
##          Neg Pred Value : 0.8189
##              Prevalence : 0.5561
##          Detection Rate : 0.4866
##    Detection Prevalence : 0.6161
##       Balanced Accuracy : 0.7916
##
##        'Positive' Class : 0
##
```

Sensitivity and Specificity

Sensitivity: Rate of correct positive identification

- Type II Error rate: $1 - \text{Sensitivity}$

Specificity: Rate of correct negative identification

- Type I Error rate: $1 - \text{Specificity}$

Sensitivity and Specificity

Sensitivity: Rate of correct positive identification

- Type II Error rate: $1 - \text{Sensitivity}$

Specificity: Rate of correct negative identification

- Type I Error rate: $1 - \text{Specificity}$

By changing our classification cutoff, we can increase sensitivity to the detriment of specificity (or vice versa)

- But the tradeoff is non-linear

Sensitivity and Specificity

Sensitivity: Rate of correct positive identification

- Type II Error rate: $1 - \text{Sensitivity}$

Specificity: Rate of correct negative identification

- Type I Error rate: $1 - \text{Specificity}$

By changing our classification cutoff, we can increase sensitivity to the detriment of specificity (or vice versa)

- But the tradeoff is non-linear
  - Increasing specificity by .1 may decrease sensitivity by .15 when specificity is .8
  - But the same increase in specificity may decrease sensitivity by .25 when specificity is .9.

Sensitivity and Specificity

Sensitivity: Rate of correct positive identification

- Type II Error rate: $1 - \text{Sensitivity}$

Specificity: Rate of correct negative identification

- Type I Error rate: $1 - \text{Specificity}$

By changing our classification cutoff, we can increase sensitivity to the detriment of specificity (or vice versa)

- But the tradeoff is non-linear
  - Increasing specificity by .1 may decrease sensitivity by .15 when specificity is .8
  - But the same increase in specificity may decrease sensitivity by .25 when specificity is .9.

When might we want high specificity? High sensitivity?

Sensitivity and Specificity

Sensitivity: Rate of correct positive identification

- Type II Error rate: $1 - \text{Sensitivity}$

Specificity: Rate of correct negative identification

- Type I Error rate: $1 - \text{Specificity}$

By changing our classification cutoff, we can increase sensitivity to the detriment of specificity (or vice versa)

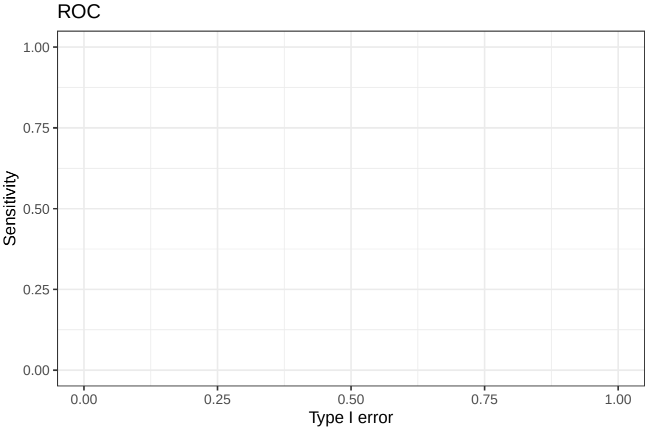
- But the tradeoff is non-linear
  - Increasing specificity by .1 may decrease sensitivity by .15 when specificity is .8
  - But the same increase in specificity may decrease sensitivity by .25 when specificity is .9.

When might we want high specificity? High sensitivity?

What are the ramifications of changing the classification cutoff?

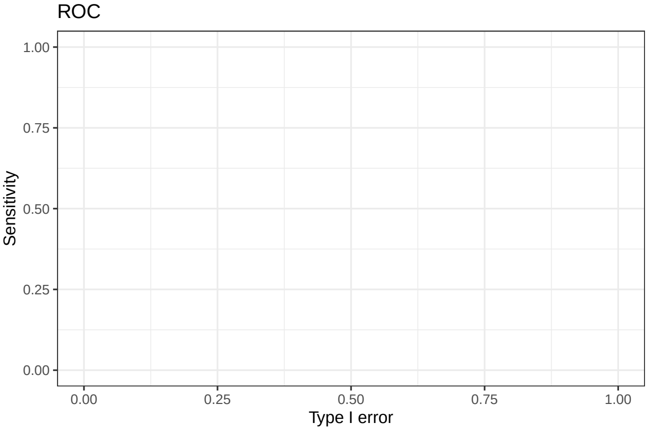
ROC Curves

A Receiver Operating Characteristic (ROC) curve is a plot of sensitivity vs. type I error rate, based on classification probabilities.



ROC Curves

A Receiver Operating Characteristic (ROC) curve is a plot of sensitivity vs. type I error rate, based on classification probabilities.

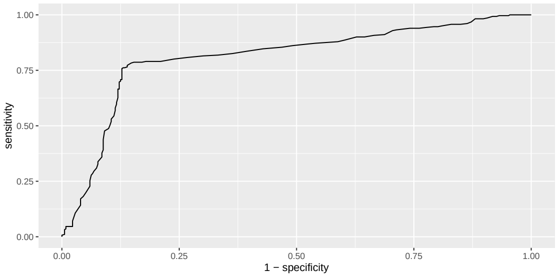


Poll: For a perfectly accurate model, what is the expected area under the ROC curve?

ROC Curves in R

The `roc` function in the `pROC` package can create ROC curves.

```
library(pROC)
roc_curve <- roc(response = Titanic1$survived, predictor = probs)
ggroc(roc_curve, legacy.axes=TRUE)
```

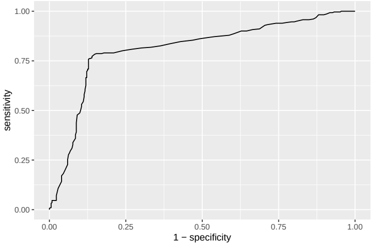



```
auc(roc_curve)
```

```
## Area under the curve: 0.8095
```

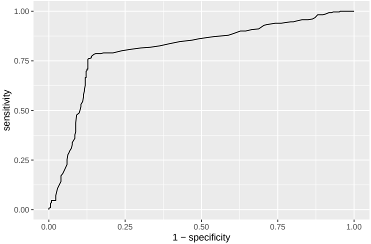
ROC Curves in R

What threshold corresponds to the "kink" in the ROC curve?



ROC Curves in R

What threshold corresponds to the "kink" in the ROC curve?




```
coords(roc_curve, "best", ret = "threshold")
```

```
##   threshold
## 1 0.2533806
```

```
coords(roc_curve, .253)
```

```
##   threshold specificity sensitivity
## 1     0.253   0.8522727   0.7829181
```

Section 3

Additional Practice

Mushroom Hunting

The `mushrooms` data set on the schedule page of the course website contains information on several species of mushrooms, including edibility.

Mushroom Hunting

The `mushrooms` data set on the schedule page of the course website contains information on several species of mushrooms, including edibility.

Can we predict whether a mushroom is edible?

Mushroom Hunting

The `mushrooms` data set on the schedule page of the course website contains information on several species of mushrooms, including edibility.

Can we predict whether a mushroom is edible?

- Create a Logistic Regression model using your choice of a small subset of predictors
  - You will need to recode your response `class` to take values 0 or 1.
- Then create an ROC curve and select a threshold that seems appropriate for this situation.
- Time permitting, create a KNN model for various values of K and compare to the logistic regression model.