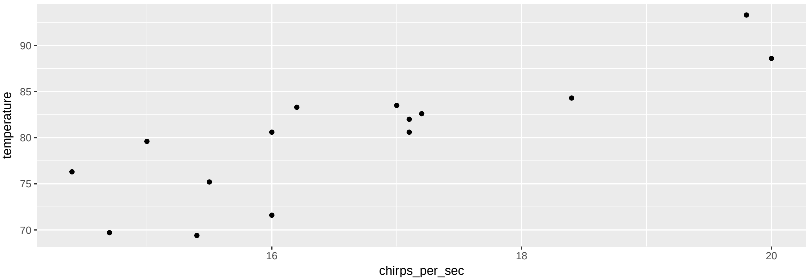
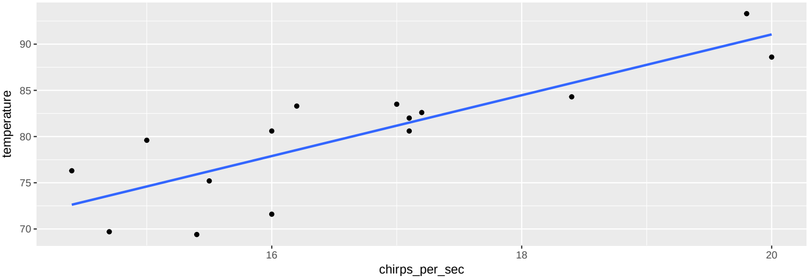
Logistic Regression

UTB

Predicciones Binarias

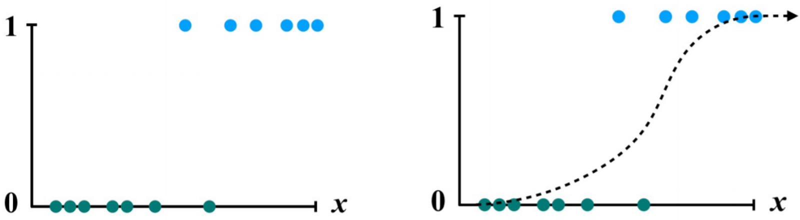
Remember Linear Regression





How it works?

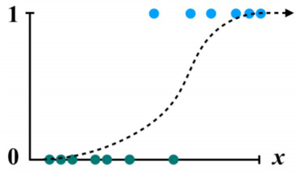
image:



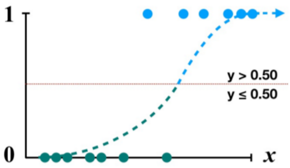
Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

Making predictios using R

image:




```
m <- glm(y ~ x1 + x2 + x3,
         data = my_dataset,
         family = "binomial")
```




```
prob <- predict(m, test_dataset,
        type = "response")
```

```
pred <- ifelse(prob > 0.50, 1, 0)
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

Example

Examine the dataset to identify potential independent variables

```
donors<- read.csv('donors.csv')
str(donors)
```

```
## 'data.frame':    93462 obs. of  13 variables:
##  $ donated           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ veteran           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ bad_address       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ age               : int  60 46 NA 70 78 NA 38 NA NA 65
##  $ has_children      : int  0 1 0 0 1 0 1 0 0 0 ...
##  $ wealth_rating     : int  0 3 1 2 1 0 2 3 1 0 ...
##  $ interest_veterans: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ interest_religion: int  0 0 0 0 1 0 0 0 0 0 ...
##  $ pet_owner         : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ catalog_shopper   : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ recency           : Factor w/ 2 levels "CURRENT","LAPS
##  $ frequency         : Factor w/ 2 levels "FREQUENT","INF
##  $ money             : Factor w/ 2 levels "HIGH","MEDIUM
```

Build the donation model

```
donation_model <-
  glm(donated ~ bad_address + interest_religion
                      + interest_veterans,
                      data = donors, family = "binomial")

# Summarize the model results
donation_model$coefficients
```

```
##       (Intercept)       bad_address interest_religion in
##       -2.95138685       -0.30779707        0.06723943
```

Summarize the model results

```r
# Summarize the model results
summary(donation_model)
```

```
##
## Call:
## glm(formula = donated ~ bad_address + interest_religion
##     family = "binomial", data = donors)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.3480  -0.3192  -0.3192  -0.3192   2.5678
##
## Coefficients:
##                    Estimate Std. Error  z value Pr(>|z|)
## (Intercept)        -2.95139    0.01652 -178.664   <2e-16
## bad_address        -0.30780    0.14348   -2.145   0.0319
## interest_religion   0.06724    0.05069    1.327   0.1847
## interest_veterans   0.11009    0.04676    2.354   0.0186
```

Response Variable

```r
table(donors$donated)
```

```
##
##     0     1
## 88751  4711
```

Estimate the donation probability

```
donors$donation_prob <-
  predict(donation_model, type = "response")
```

Find the donation probability of the average prospect

```
mean(donors$donated)
```

```
## [1] 0.05040551
```

Predict a donation if probability of donation is greater than average (0.0504)

```
donors$donation_pred <- ifelse(donors$donation_prob > 0.050

## Calculate the model's accuracy
mean(donors$donated == donors$donation_pred)
```

```
## [1] 0.794815
```

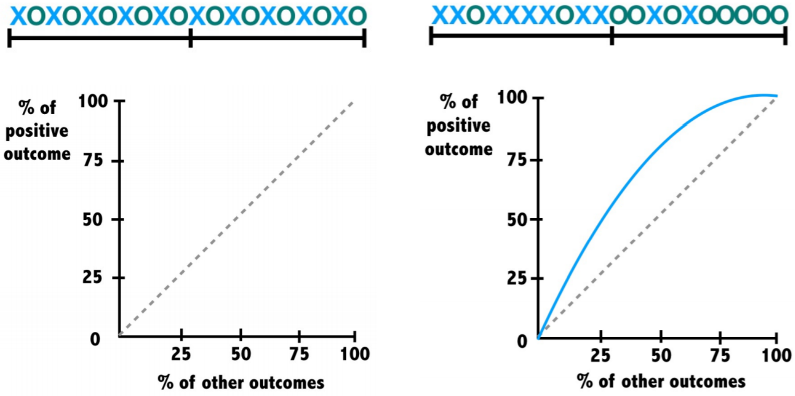
The limitations of accuracy

In the previous exercise, you found that the logistic regression model made a correct prediction nearly 80% of the time. Despite this relatively high accuracy, the result is misleading due to the rarity of outcome being predicted.

The donors dataset is available in your workspace. What would the accuracy have been if a model had simply predicted "no donation" for each person?

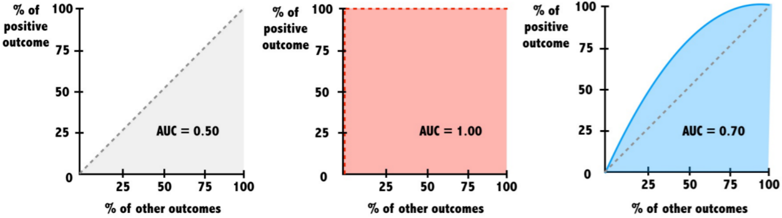
Making predictios using R

image:



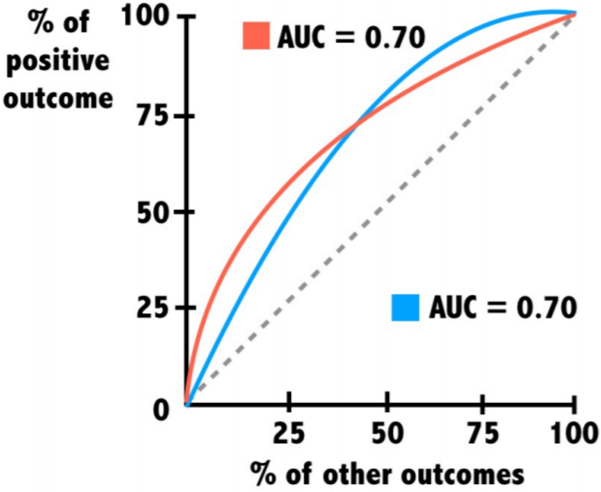
Area under the curve ROC

image:



Comparing ROC Curves

image:



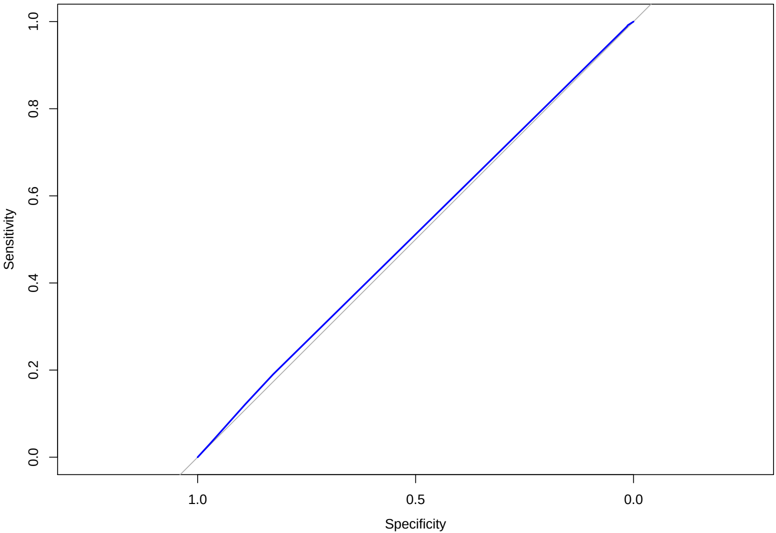
Implementing ROC Curve

```r
# Load the pROC package
library(pROC)

# Create a ROC curve
ROC <- roc(donors$donated, donors$donation_prob)
```

Plot the ROC curve

```
plot(ROC, col = "blue")
```



Interpretate ROC Curve

Calculate the area under the curve (AUC)

```r
auc(ROC)
```

```
## Area under the curve: 0.5102
```

Based on this visualization, the model isn't doing much better than baseline— a model doing nothing but making predictions at random.

A more sophisticated model

One of the best predictors of future giving is a history of recent, frequent, and large gifts. In marketing terms, this is known as R/F/M:

-Recency -Frequency -Money

Donors that haven given both recently and frequently may be especially likely to give again; in other words, the combined impact of recency and frequency may be greater than the sum of the separate effects.

Because these predictors together have a greater impact on the dependent variable, their joint effect must be modeled as an interaction.

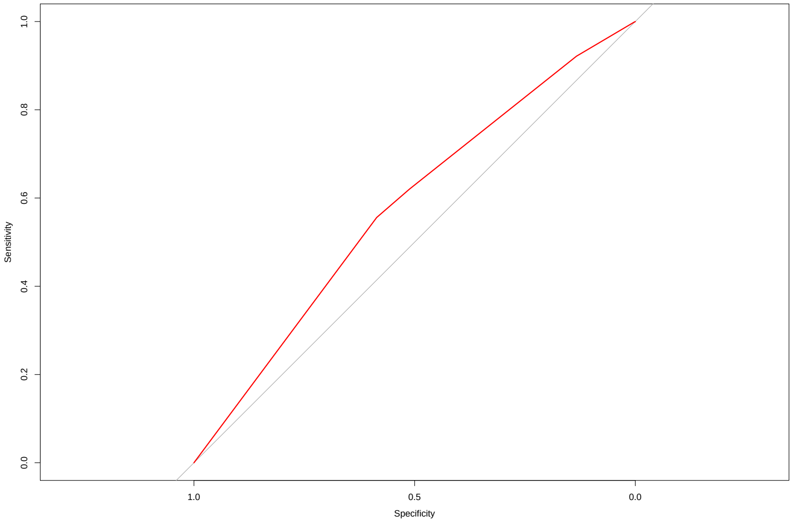
RFM MODEL

```r
# Build a recency, frequency, and money (RFM) model
rfm_model <-  glm(donated ~ money + recency*frequency, donc

# Compute predicted probabilities for the RFM model
rfm_prob <- predict(rfm_model, type = "response")
```

Plot the ROC curve and find AUC for the new model

```
ROC <- roc(donors$donated , rfm_prob)
plot(ROC, col = "red")
```



AUC Value

```r
auc(ROC)
```

```
## Area under the curve: 0.5785
```

Building a stepwise regression model

In the absence of subject-matter expertise, stepwise regression can assist with the search for the most important predictors of the outcome of interest.

In this exercise, you will use a forward stepwise approach to add predictors to the model one-by-one until no additional benefit is seen.

Stepwise Model

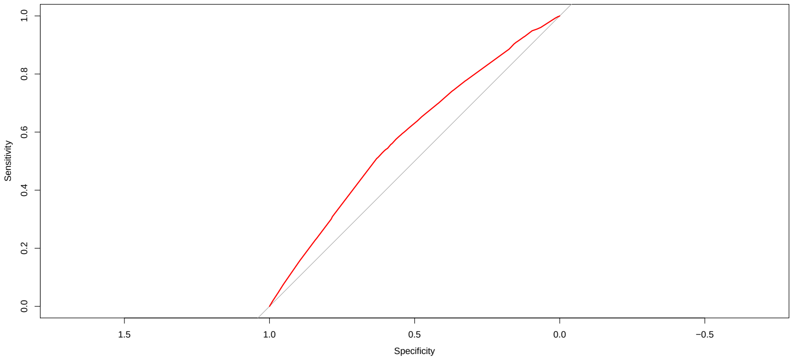
```
# Specify a null model with no predictors
null_model <- glm(donated ~ 1, data = donors, family = "bi

# Specify the full model using all of the potential predic
full_model <- glm(donated ~ . , data = donors, family = "bi

# Use a forward stepwise algorithm to build a parsimonious
step_model <- step(null_model, scope = list(lower = null_mo
```

Plot the ROC of the stepwise model

```
step_prob <- predict(step_model, type = "response")
ROC <- roc(donors$donated, step_prob)
plot(ROC, col = "red")
```



AUC VALUE

```r
auc(ROC)
```

```
## Area under the curve: 0.5855
```