Introduction to logistic regression

Stephanie J. Spielman, PhD

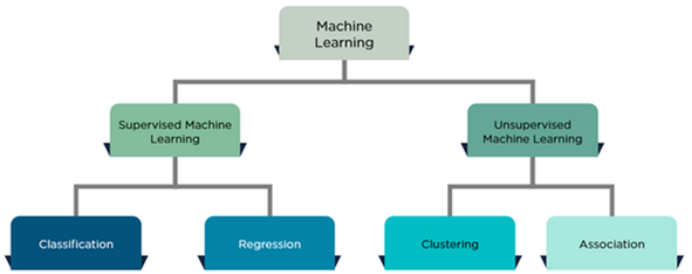
Data Science for Biologists, Fall 2020

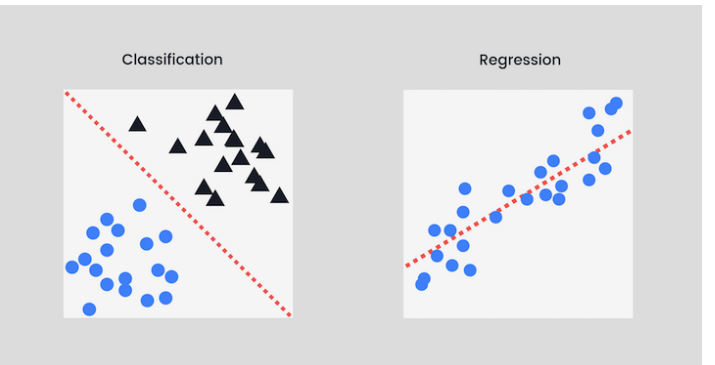
Linear regression vs. logistic regression

- Linear regression: How much do these (linearly-related) predictors explain variation in my *numeric* response variable?
- Logistic regression: How well do these predictors explain variation in my *categorical* ***binary*** response variable?
  - E.g. predicting Species in the iris dataset would be a categorical predictor, but NOT binary
  - Type of classifier

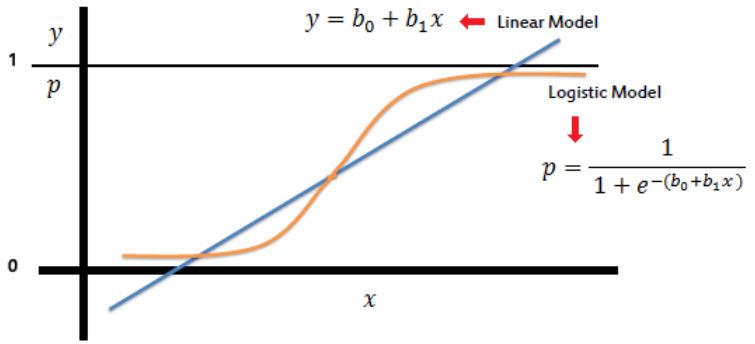
Where are we in the "machine learning" universe?

- Machine learning = the computer learns through experience
  - More data = more experience! *Training models on data IS machine learning*
  - Ignore the AI hype.



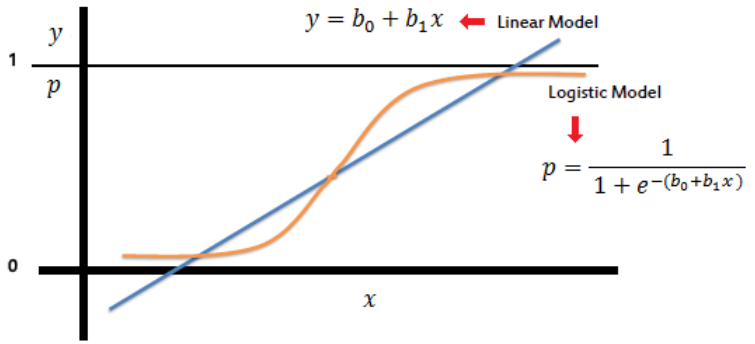


Logistic regression



- Linear regression: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 \ldots + \beta_N X_N + \epsilon$

Logistic regression



- Linear regression: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 \ldots + \beta_N X_N + \epsilon$
- Logistic regression *transforms the predictors*
  - $t = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 \ldots + \beta_N X_N + \epsilon$
  - $Y = \frac{1}{1+e^{-t}}$ (or, $p = \ldots$ in image)

```
# too large to fit on slide..
data_url <- paste0("https://raw.githubusercontent.com/sjspielman/",
                   "datascience_for_biologists/master/docs/",
                   "fall2020/slides/biopsy.csv")

biopsy <- read_csv(data_url)

dplyr::glimpse(biopsy)
## Rows: 683
## Columns: 10
## $ clump_thickness      <dbl> 5, 5, 3, 6, 4, 8, 1, 2, 2, 4, 1, 2, 5, 1, 8, 7, …
## $ uniform_cell_size    <dbl> 1, 4, 1, 8, 1, 10, 1, 1, 1, 2, 1, 1, 3, 1, 7, 4,…
## $ uniform_cell_shape   <dbl> 1, 4, 1, 8, 1, 10, 1, 2, 1, 1, 1, 1, 3, 1, 5, 6,…
## $ marg_adhesion        <dbl> 1, 5, 1, 1, 3, 8, 1, 1, 1, 1, 1, 1, 3, 1, 10, 4,…
## $ epithelial_cell_size <dbl> 2, 7, 2, 3, 2, 7, 2, 2, 2, 2, 1, 2, 2, 2, 7, 6, …
## $ bare_nuclei          <dbl> 1, 10, 2, 4, 1, 10, 10, 1, 1, 1, 1, 1, 3, 3, 9, …
## $ bland_chromatin      <dbl> 3, 3, 3, 3, 3, 9, 3, 3, 1, 2, 3, 2, 4, 3, 5, 4, …
## $ normal_nucleoli      <dbl> 1, 2, 1, 7, 1, 7, 1, 1, 1, 1, 1, 1, 4, 1, 5, 3, …
## $ mitoses              <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 5, 1, 1, 1, 1, 1, 4, 1, …
## $ outcome              <chr> "benign", "benign", "benign", "benign", "benign"…
```

Building the logistic regression: Prepare the data

```
## Ensure the column is a factor, OR it has 0/1 values
## Help yourself by coding success = 1, failure = 0. This way you don't need
alphabetical order
biopsy %>%
  mutate(outcome_01 = case_when(outcome == "malignant" ~ 1, # "success"
                                outcome == "benign" ~ 0)) %>%
  select(-outcome) %>%
  select(outcome_01, everything()) -> biopsy_outcome01

head(biopsy_outcome01)
## # A tibble: 6 x 10
##   outcome_01 clump_thickness uniform_cell_si… uniform_cell_sh… marg_adhesion
##        <dbl>           <dbl>            <dbl>            <dbl>         <dbl>
## 1          0               5                1                1             1
## 2          0               5                4                4             5
## 3          0               3                1                1             1
## 4          0               6                8                8             1
## 5          0               4                1                1             3
## 6          1               8               10               10             8
## # … with 5 more variables: epithelial_cell_size <dbl>, bare_nuclei <dbl>,
## #   bland_chromatin <dbl>, normal_nucleoli <dbl>, mitoses <dbl>
```

Building the logistic regression: Build the model

`glm(response ~ predictors, data = data, family = "binomial")`

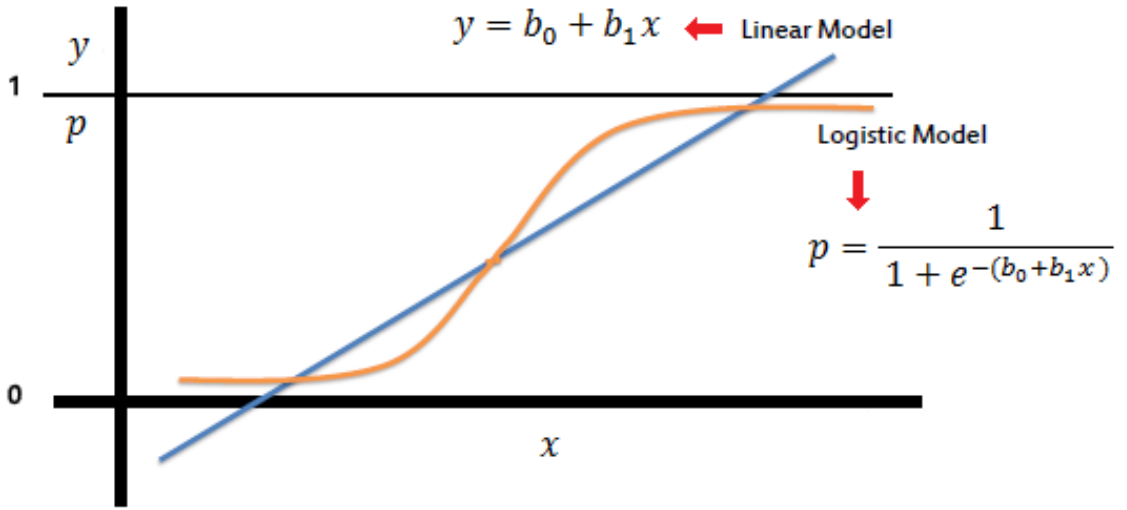
```
baseline_logit_fit <- glm(outcome_01 ~ ., data = biopsy_outcome01, family =
"binomial")

fit <- step(baseline_logit_fit, trace = F) # Read "Introduction to Model
Selection"!!
```

Interpreting the logistic regression coefficients

```
broom::tidy(fit)
## # A tibble: 8 x 5
##   term               estimate std.error statistic  p.value
##   <chr>                 <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)          -9.98     1.13       -8.86 7.66e-19
## 2 clump_thickness       0.534    0.141       3.79 1.49e- 4
## 3 uniform_cell_shape    0.345    0.172       2.01 4.43e- 2
## 4 marg_adhesion         0.342    0.119       2.87 4.07e- 3
## 5 bare_nuclei           0.388    0.0936      4.15 3.32e- 5
## 6 bland_chromatin       0.462    0.168       2.75 6.02e- 3
## 7 normal_nucleoli       0.226    0.111       2.04 4.16e- 2
## 8 mitoses               0.531    0.324       1.64 1.02e- 1
```

- For every unit increase in the predictor, the **log odds of success** of the response increases by the coefficient
  - $Pr(success)$ = probability of *malignant* biopsy for a given set of observations (predictors)
  - $Pr(failure)$ = probability of *benign* biopsy for a given set of observations
  - **Log odds** = $ln\left(\frac{Pr(success)}{Pr(failure)}\right)$

Visualizing the logistic regression



```
## USING head() to make it fit on slides!!

## What would have been your Y-values if this were regression
## YOUR X-AXIS !!
head(fit$linear.predictors)
##         1         2         3         4         5         6
## -4.093622  2.032920 -4.773329  1.378604 -3.942642 10.636051

## The logit transformed - PROBABILITIES OF SUCCESS
## YOUR Y-AXIS !!
head(fit$fitted.values)
##           1           2           3           4           5           6
## 0.016405105 0.884210413 0.008381356 0.798766714 0.019027825 0.999975967
```

```
## USING head() to make it fit on slides!!

## What would have been your Y-values if this were regression
## YOUR X-AXIS !!
head(fit$linear.predictors)
##         1         2         3         4         5         6
## -4.093622  2.032920 -4.773329  1.378604 -3.942642 10.636051

## The logit transformed - PROBABILITIES OF SUCCESS
## YOUR Y-AXIS !!
head(fit$fitted.values)
##           1           2           3           4           5           6
## 0.016405105 0.884210413 0.008381356 0.798766714 0.019027825 0.999975967
```

- $t = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 \ldots + \beta_N X_N + \epsilon$
- $Y = \frac{1}{1+e^{-t}}$

```
## USING head() to make it fit on slides!!

## What would have been your Y-values if this were regression
## YOUR X-AXIS !!
head(fit$linear.predictors)
##         1         2         3         4         5         6
## -4.093622  2.032920 -4.773329  1.378604 -3.942642 10.636051

## The logit transformed - PROBABILITIES OF SUCCESS
## YOUR Y-AXIS !!
head(fit$fitted.values)
##           1           2           3           4           5           6
## 0.016405105 0.884210413 0.008381356 0.798766714 0.019027825 0.999975967
```

- $t = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 \ldots + \beta_N X_N + \epsilon$
- $Y = \frac{1}{1+e^{-t}}$

```
1/(1 + exp(-1 * fit$linear.predictors)) %>% head()
##           1           2           3           4           5           6
## 0.016405105 0.884210413 0.008381356 0.798766714 0.019027825 0.999975967
```

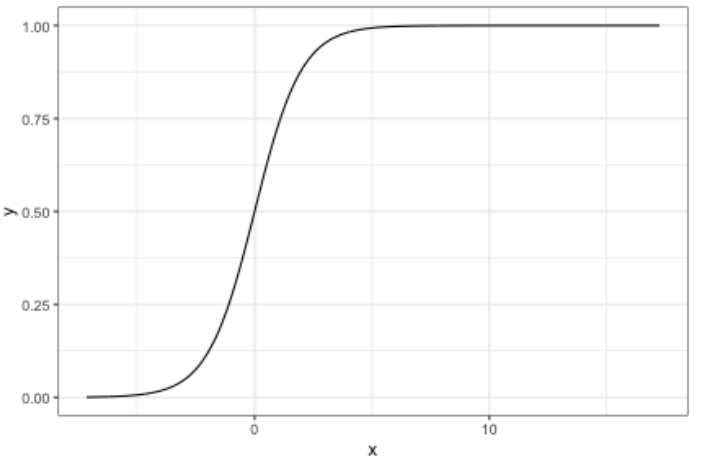
Visualizing the model: Prepare the data

```
tibble(x = fit$linear.predictors,
       y = fit$fitted.values,
       # Helps to use the ORIGINAL biopsy version so that outcome is
"malignant"/"benign"
       outcome = biopsy$outcome) -> fit_tibble

fit_tibble
## # A tibble: 683 x 3
##        x       y outcome
##    <dbl>   <dbl> <chr>
##  1 -4.09 0.0164  benign
##  2  2.03 0.884   benign
##  3 -4.77 0.00838 benign
##  4  1.38 0.799   benign
##  5 -3.94 0.0190  benign
##  6 10.6  1.00    malignant
##  7 -2.73 0.0609  benign
##  8 -5.35 0.00472 benign
##  9 -4.49 0.0110  benign
## 10 -5.09 0.00612 benign
## # … with 673 more rows
```

Visualizing the model

```
head(fit_tibble)
## # A tibble: 6 x 3
##       x       y outcome
##   <dbl>   <dbl> <chr>
## 1 -4.09 0.0164  benign
## 2  2.03 0.884   benign
## 3 -4.77 0.00838 benign
## 4  1.38 0.799   benign
## 5 -3.94 0.0190  benign
## 6 10.6  1.00    malignant
```

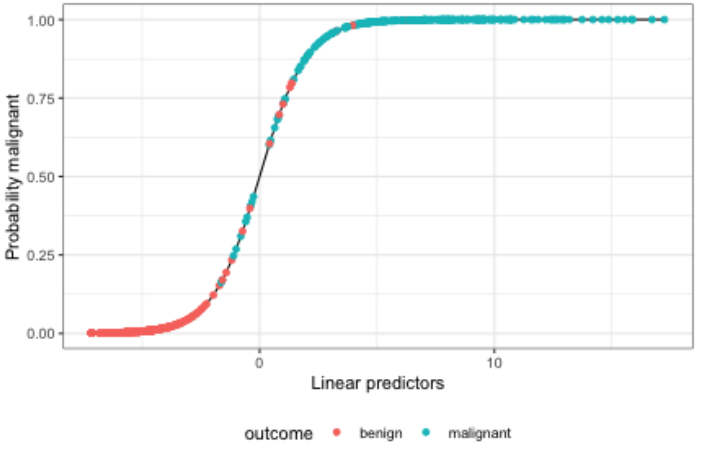
```
ggplot(fit_tibble, aes(x = x, y = y))
+
 geom_line() +
 theme(legend.position = "bottom")
```



Visualizing the model FULLY!!!

```
head(fit_tibble)
## # A tibble: 6 x 3
##       x       y outcome
##   <dbl>   <dbl> <chr>
## 1 -4.09 0.0164  benign
## 2  2.03 0.884   benign
## 3 -4.77 0.00838 benign
## 4  1.38 0.799   benign
## 5 -3.94 0.0190  benign
## 6 10.6  1.00    malignant
```

```
ggplot(fit_tibble, aes(x = x, y = y))
+
 geom_line() +
geom_point(aes(color = outcome)) +
 theme(legend.position = "bottom") +
  labs(x = "Linear predictors",
       y = "Probability malignant")
```



Confusion matrix time

	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

- **First ask:** is the result positive or negative? **Then ask:** should we have gotten that result though?
  - If yes, *TRUE*. If not, *FALSE*.

What is it?

A new arthritis drug does help pain clinical trials, even though it actually does reduce arthritis pain.

What is it?

A new arthritis drug does help pain clinical trials, even though it actually does reduce arthritis pain.

A person with HIV receives a positive test result for HIV.

What is it?

A new arthritis drug does help pain clinical trials, even though it actually does reduce arthritis pain.

A person with HIV receives a positive test result for HIV.

A person using illegal performing enhancing drugs passes a test clearing them of drug use.

What is it?

A new arthritis drug does help pain clinical trials, even though it actually does reduce arthritis pain.

A person with HIV receives a positive test result for HIV.

A person using illegal performing enhancing drugs passes a test clearing them of drug use.

A study found a significant relationship between neck strain and jogging, when reality there is no relationship.

What is it?

A new arthritis drug does help pain clinical trials, even though it actually does reduce arthritis pain.

A person with HIV receives a positive test result for HIV.

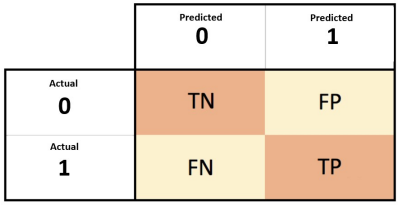
A person using illegal performing enhancing drugs passes a test clearing them of drug use.

A study found a significant relationship between neck strain and jogging, when reality there is no relationship.

A healthy individual gets a positive cancer biopsy result.

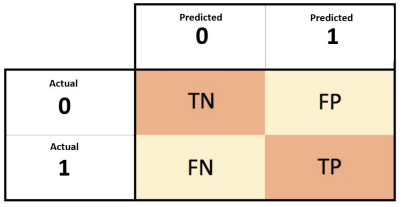
Classification metrics (an abbreviated set)

- True positive rate: $TPR = TP/P = \frac{TP}{TP+FN}$
  - AKA *sensitivity* AKA *recall*



Classification metrics (an abbreviated set)

- True positive rate: $TPR = TP/P = \frac{TP}{TP+FN}$
  - AKA *sensitivity* AKA *recall*
- True negative rate: $TNR = TN/N = \frac{TN}{FP+TN}$
  - AKA *specificity*



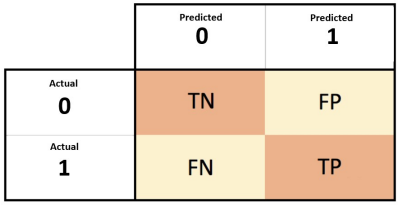
Classification metrics (an abbreviated set)

- True positive rate: $TPR = TP/P = \frac{TP}{TP+FN}$
  - AKA *sensitivity* AKA *recall*
- True negative rate: $TNR = TN/N = \frac{TN}{FP+TN}$
  - AKA *specificity*
- False positive rate: $FPR = FP/N = \frac{FP}{FP+TN}$
  - AKA *1 - specificity*

	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

Classification metrics (an abbreviated set)

- True positive rate: $TPR = TP/P = \frac{TP}{TP+FN}$
  - AKA *sensitivity* AKA *recall*
- True negative rate: $TNR = TN/N = \frac{TN}{FP+TN}$
  - AKA *specificity*
- False positive rate: $FPR = FP/N = \frac{FP}{FP+TN}$
  - AKA *1 - specificity*
- Precision: $PPV = \frac{TP}{TP+FP}$
  - AKA *positive predictive value*



Classification metrics (an abbreviated set)

- True positive rate: $TPR = TP/P = \frac{TP}{TP+FN}$
  - AKA *sensitivity* AKA *recall*
- True negative rate: $TNR = TN/N = \frac{TN}{FP+TN}$
  - AKA *specificity*
- False positive rate: $FPR = FP/N = \frac{FP}{FP+TN}$
  - AKA *1 - specificity*
- Precision: $PPV = \frac{TP}{TP+FP}$
  - AKA *positive predictive value*
- Accuracy: $\frac{TP+TN}{TP+TN+FP+FN}$

	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

Recall our model:

```
# Recall:
biopsy %>%
  mutate(outcome_01 = case_when(outcome == "malignant" ~ 1, # "success"
                                outcome == "benign" ~ 0)) %>%
  select(-outcome) %>%
  select(outcome_01, everything()) -> biopsy_outcome01
baseline_logit_fit <- glm(outcome_01 ~ ., data = biopsy_outcome01, family =
"binomial")
fit <- step(baseline_logit_fit, trace = F) # Read "Introduction to Model
Selection"!!

tibble(x = fit$linear.predictors,
       y = fit$fitted.values,
       outcome = biopsy$outcome) -> fit_tibble

head(fit_tibble)
## # A tibble: 6 x 3
##       x       y outcome
##   <dbl>   <dbl> <chr>
## 1 -4.09 0.0164  benign
## 2  2.03 0.884   benign
## 3 -4.77 0.00838 benign
## 4  1.38 0.799   benign
## 5 -3.94 0.0190  benign
## 6 10.6  1.00    malignant
```

Caculating performance measures

- Requires a *threshold* to call malignant/benign outcomes.
- For an example, let's say >=0.75 is malignant (success). <0.75 is benign (failure)
- Accuracy: $\frac{TP+TN}{TP+TN+FP+FN}$

```
threshold <- 0.75
fit_tibble %>%
  rename(truth = outcome) %>%
  mutate(pred = if_else(y >= threshold, "P", "N"))
## # A tibble: 683 x 4
##        x       y truth     pred
##    <dbl>   <dbl> <chr>     <chr>
##  1 -4.09 0.0164  benign    N
##  2  2.03 0.884   benign    P
##  3 -4.77 0.00838 benign    N
##  4  1.38 0.799   benign    P
##  5 -3.94 0.0190  benign    N
##  6 10.6  1.00    malignant P
##  7 -2.73 0.0609  benign    N
##  8 -5.35 0.00472 benign    N
##  9 -4.49 0.0110  benign    N
## 10 -5.09 0.00612 benign    N
## # … with 673 more rows
```

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

```
threshold <- 0.75
fit_tibble %>%
  rename(truth = outcome) %>%
  mutate(pred = if_else(y >= threshold, "P", "N")) %>%
  mutate(classif = case_when(truth == "malignant" & pred == "P" ~ "TP",
                             truth == "malignant" & pred == "N" ~ "FN",
                             truth == "benign"    & pred == "N" ~ "TN",
                             truth == "benign"    & pred == "P" ~ "FP")) ->
model_classif

model_classif
## # A tibble: 683 x 5
##        x       y truth     pred  classif
##    <dbl>   <dbl> <chr>     <chr> <chr>
##  1 -4.09 0.0164  benign    N     TN
##  2  2.03 0.884   benign    P     FP
##  3 -4.77 0.00838 benign    N     TN
##  4  1.38 0.799   benign    P     FP
##  5 -3.94 0.0190  benign    N     TN
##  6 10.6  1.00    malignant P     TP
##  7 -2.73 0.0609  benign    N     TN
##  8 -5.35 0.00472 benign    N     TN
##  9 -4.49 0.0110  benign    N     TN
## 10 -5.09 0.00612 benign    N     TN
## # … with 673 more rows
```

```
model_classif %>%
  # how many in each classif category?
  count(classif)
## # A tibble: 4 x 2
##   classif     n
##   <chr>   <int>
## 1 FN         20
## 2 FP          7
## 3 TN        437
## 4 TP        219
```

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

- Accuracy = (437 + 219) / (20 + 7 + 437 + 219) = **0.96**

```
model_classif %>%
  # how many in each classif category?
  count(classif)
## # A tibble: 4 x 2
##   classif     n
##   <chr>   <int>
## 1 FN         20
## 2 FP          7
## 3 TN        437
## 4 TP        219
```

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

- Accuracy = (437 + 219) / (20 + 7 + 437 + 219) = **0.96**

```
model_classif %>%
  count(classif) %>%
  pivot_wider(names_from = classif,
values_from = n)
## # A tibble: 1 x 4
##      FN    FP    TN    TP
##   <int> <int> <int> <int>
## 1    20     7   437   219
```

```
model_classif %>%
  count(classif) %>%
  pivot_wider(names_from = classif,
values_from = n) %>%
  mutate(accuracy = (TP + TN)/(TP +
TN + FP + FN))
## # A tibble: 1 x 5
##      FN    FP    TN    TP accuracy
##   <int> <int> <int> <int>    <dbl>
## 1    20     7   437   219    0.960
```

How good is the model?

- In linear regression, we often uses R^2 values to compare different viable models. Higher R^2 often (but not always!) means, "more predictive model"

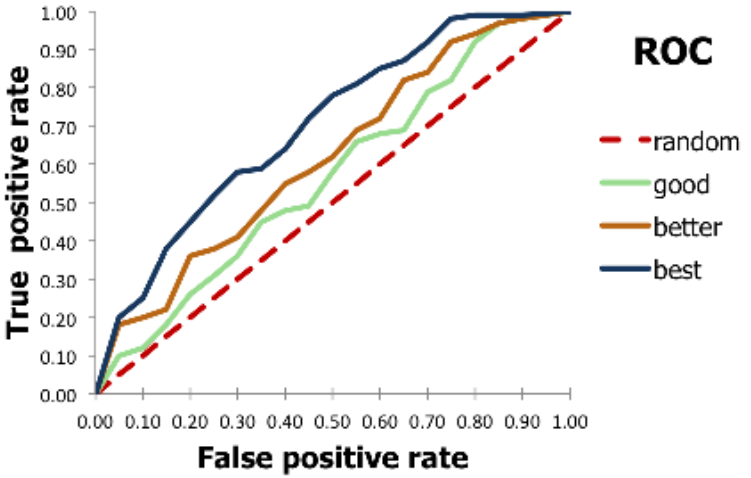
- In logistic regression, performance **depends** on your chosen threshold! So, how do we choose a threshold?
  - Usually, find the threshold that makes the false positive rate <5%>
- We also use **AUC** (area under the curve... what curve?)

Evaluating logistic regressions

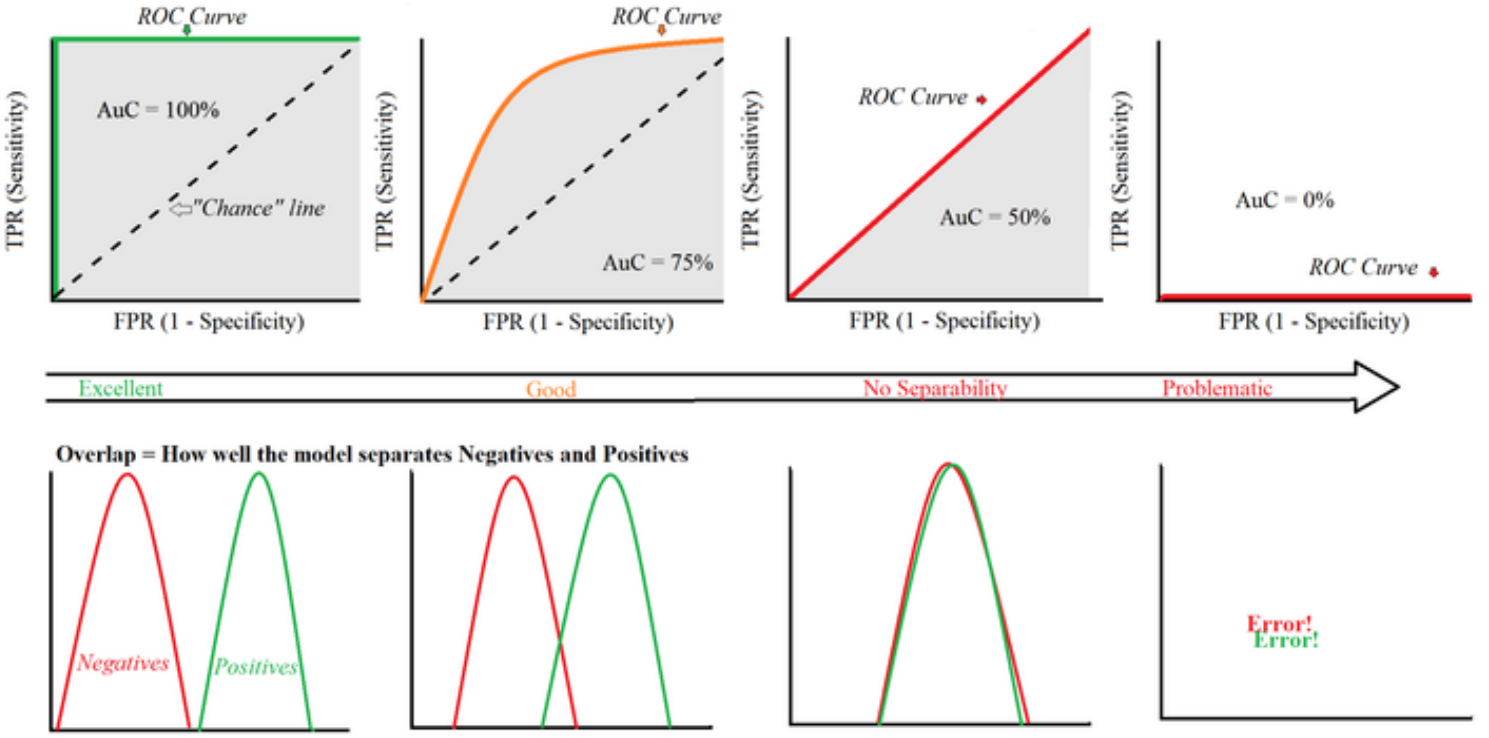
Receiver Operating Characteristic Curve

- TPR on Y-axis
- FPR (1 - specificity) on X-axis
- The AUC (**a**rea **u**nder the **c**urve) is an overall assessment of performance *at any threshold*

- $TPR = TP/P = \frac{TP}{TP+FN}$ (*sensitivity* AKA *recall*)
- $TNR = TN/N = \frac{TN}{FP+TN}$ (*specificity*)
- $FPR = FP/N = \frac{FP}{FP+TN}$ (*1 - specificity*)

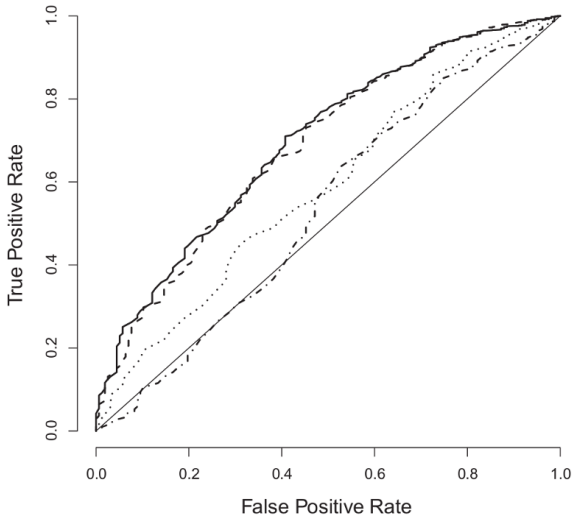


Getting a "feel" for ROC curves

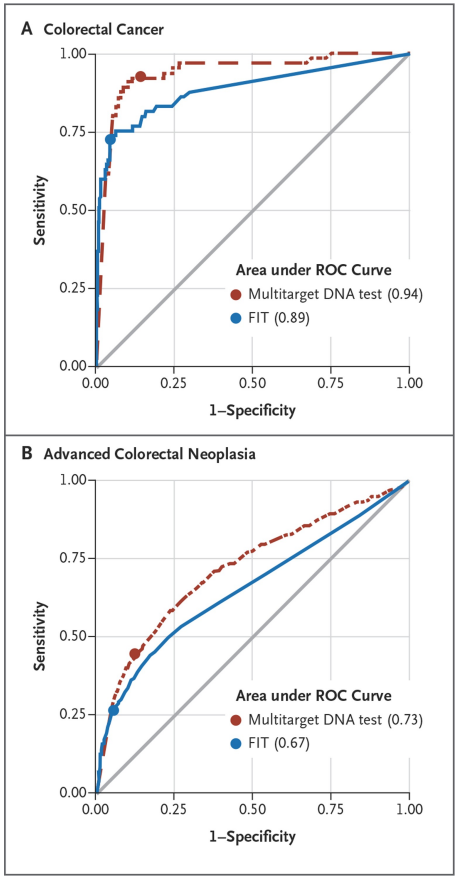


Examples of ROC curves in the literature

Keller et al. Genome Biol Evol 2012; 4:80-88



Imperiale et al. N Engl J Med 2014; 370:1287-1297



ROC vs PR

- ROC curves are suitable when data is *balanced*
  - Similar amounts of positives, negatives in the dataset
  - FPR (1 - specificity) on X-axis, TPR on Y-axis
- **Precision-Recall** curves are more suitable for *unbalanced* data
  - Precision (PPV) on Y-axis, recall (TPR) on X-axis

- $TPR = TP/P = \frac{TP}{TP+FN}$ (*recall*)
- $FPR = FP/N = \frac{FP}{FP+TN}$
- $PPV = \frac{TP}{TP+FP}$

Is the biopsy data balanced?

```
biopsy %>%
  count(outcome)
## # A tibble: 2 x 2
##   outcome       n
##   <chr>     <int>
## 1 benign      444
## 2 malignant   239
```

- About 2:1::benign:malignant
- Not very balanced, but it's reasonable. ROC is ok to use!
- *Problematically imbalanced* would be 4000 benign and 5 malignant (or vice versa).

Making ROC curves

- Recall:
  - Our model fit is saved in **`fit`**
  - Our model was built with **`biopsy_outcome01`** dataset

```
## Use the pROC library to help you
#install.packages("pROC")
library(pROC)
## Type 'citation("pROC")' for a citation.
##
## Attaching package: 'pROC'
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
# Use the function roc(), and data with the 0/1 coded outcome!!
model_roc <- roc(biopsy_outcome01$outcome_01, fit$linear.predictors)
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

# This also works the same:
model_roc <- roc(biopsy_outcome01$outcome_01, fit$fitted.values)
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

Getting information out

```
model_roc$auc
## Area under the curve: 0.9963
```

- Models are usually *not this good*. This dataset comes from a package that teaches modeling - it was chosen for a reason...

Getting information out

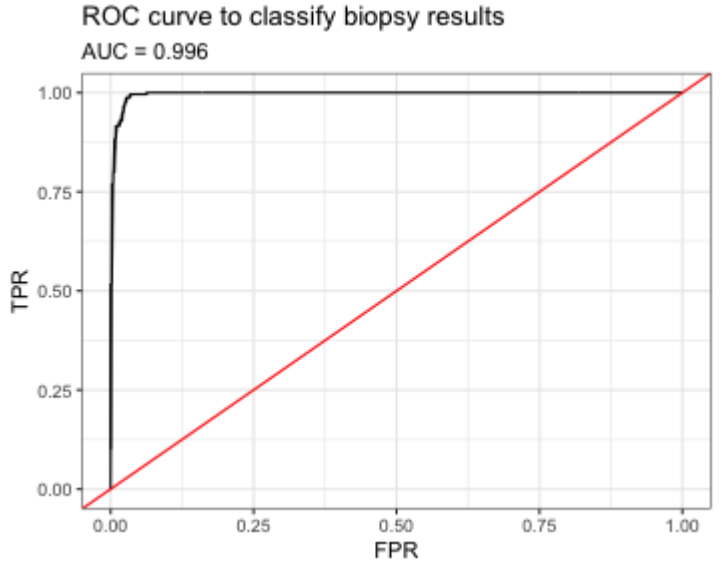
```
model_roc$auc
## Area under the curve: 0.9963
```

- Models are usually *not this good*. This dataset comes from a package that teaches modeling - it was chosen for a reason...

```
## Piped into head() to fit on the slide

## True positive rates
model_roc$sensitivities %>% head()
## [1] 1 1 1 1 1 1

## True negative rates
model_roc$specificities %>% head()
## [1] 0.00000000 0.07432432 0.07657658 0.08108108 0.08333333 0.08558559

## False positives rates
1 - model_roc$specificities %>% head()
## [1] 1.0000000 0.9256757 0.9234234 0.9189189 0.9166667 0.9144144
```

Make an ROC curve

```
tibble(TPR = model_roc$sensitivities,
       FPR = 1 - model_roc$specificities) %>%
  ggplot(aes(x = FPR, y = TPR)) +
  geom_line() +
  labs(title = "ROC curve to classify biopsy results",
       subtitle = paste("AUC =",round(model_roc$auc, 3)) ) +
  ## this is the y=x line to GUIDE US and help us interpret the curve
  geom_abline(col = "red")
```



Using ROC to determine the reliable model

- Build your candidate models
- Determine their AUC value using the pROC package
- The highest AUC is the most reliable model
- ...and of course, make a visualization!

Build your candidate models

```
names(biopsy_outcome01)
##  [1] "outcome_01"           "clump_thickness"      "uniform_cell_size"
##  [4] "uniform_cell_shape"   "marg_adhesion"        "epithelial_cell_size"
##  [7] "bare_nuclei"          "bland_chromatin"      "normal_nucleoli"
## [10] "mitoses"

# fit1: Predict outcome with mitoses and clump_thickness, for example
fit1 <- glm(outcome_01 ~ mitoses + clump_thickness, data = biopsy_outcome01,
family = "binomial")

# fit2: Predict outcome with mitoses and normal_nucleoli
fit2 <- glm(outcome_01 ~ mitoses + normal_nucleoli, data = biopsy_outcome01,
family = "binomial")

# fit3: Predict outcome with mitoses, normal_nucleoli, and clump_thickness
fit3 <- glm(outcome_01 ~ mitoses + normal_nucleoli + clump_thickness, data =
biopsy_outcome01, family = "binomial")
```

Determine their AUC values

```
fit1_roc <- roc(biopsy_outcome01$outcome_01, fit1$linear.predictors)
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
fit1_roc$auc
## Area under the curve: 0.927

fit2_roc <- roc(biopsy_outcome01$outcome_01, fit2$linear.predictors)
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
fit2_roc$auc
## Area under the curve: 0.9094

fit3_roc <- roc(biopsy_outcome01$outcome_01, fit3$linear.predictors)
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
fit3_roc$auc
## Area under the curve: 0.9717
```

Visualize: Create the data for plotting

- Need to combine all values into ONE tibble

```
tibble(TPR = fit1_roc$sensitivities,
       FPR = 1 - fit1_roc$specificities,
       model = "fit1") -> fit1_tibble

tibble(TPR = fit2_roc$sensitivities,
       FPR = 1 - fit2_roc$specificities,
       model = "fit2") -> fit2_tibble

tibble(TPR = fit3_roc$sensitivities,
       FPR = 1 - fit3_roc$specificities,
       model = "fit3") -> fit3_tibble

bind_rows(fit1_tibble, fit2_tibble) %>%
  bind_rows(fit3_tibble) -> final_tibble
```

Visualize: Plot away!

```
ggplot(final_tibble, aes(x = FPR, y = TPR, color = model, group = model)) +
  geom_line() +
  labs(title = "ROC curves for candidate models",
       x = "FPR", y = "TPR") +
  scale_color_brewer(palette = "Dark2") +
  ## this is the y=x line to GUIDE US and help us interpret the curve
  geom_abline()
```