Introduction to Data Science CS61 June 12 - July 12, 2018

Dr. Ash Pahwa

Lesson 8: kNN Model Assessment

Lesson 8.2: Sensitivity, Specificity and ROC Curves

Outline

- Sensitivity & Specificity
- Computing Sensitivity & Specificity from Confusion Matrix
- Visualization of Sensitivity & Specificity
- ROC Curves
- Building ROC Curves in R
- Building ROC Curves in Python

Sensitivity & Specificity

Definitions
Conditional Probabilities

	CORRECT DECISION	ERROR
Person has a disease 	Sensitivity: Probability that the test is positive given you have the disease	False Negative Rate (FNR): Probability that the test is negative given you have the disease
Person DOES NOT has a disease (a healthy person) 	Specificity: Probability that the test is negative given you do not have the disease	False Positive Rate (FPR): Probability that the test is positive given you do not have the disease

Definitions
Conditional Probabilities

- Sensitivity
  - Probability that the test is positive given you have the disease
  - P(Test=Positive | Person has disease)
- Specificity
  - Probability that the test is negative given you do not have the disease
  - P(Test=Negative | Person does not has disease)

Definitions
Conditional Probabilities

- Specificity = 100%
  - Test Predicts that all healthy people are healthy
- Specificity = 0%
  - Test Predicts that all healthy people are sick

- Sensitivity = 100%
  - Test Predicts that all sick people are sick
- Sensitivity = 0%
  - Test Predicts that all sick people are healthy

If a Test is too **Lenient**
Specificity = 100%, Sensitivity = 0%

- **Specificity = 100%**
  - **Test Predicts that all healthy people are healthy**
- ~~Specificity = 0%~~
  - ~~Test Predicts that all healthy people are sick~~

- ~~Sensitivity = 100%~~
  - ~~Test Predicts that all sick people are sick~~
- **Sensitivity = 0%**
  - **Test Predicts that all sick people are healthy**

If a Test is too **Strict**
Specificity = 0%, Sensitivity =100%

- ~~Specificity = 100%~~
  - ~~Test Predicts that all healthy people are healthy~~
- **Specificity = 0%**
  - **Test Predicts that all healthy people are sick**

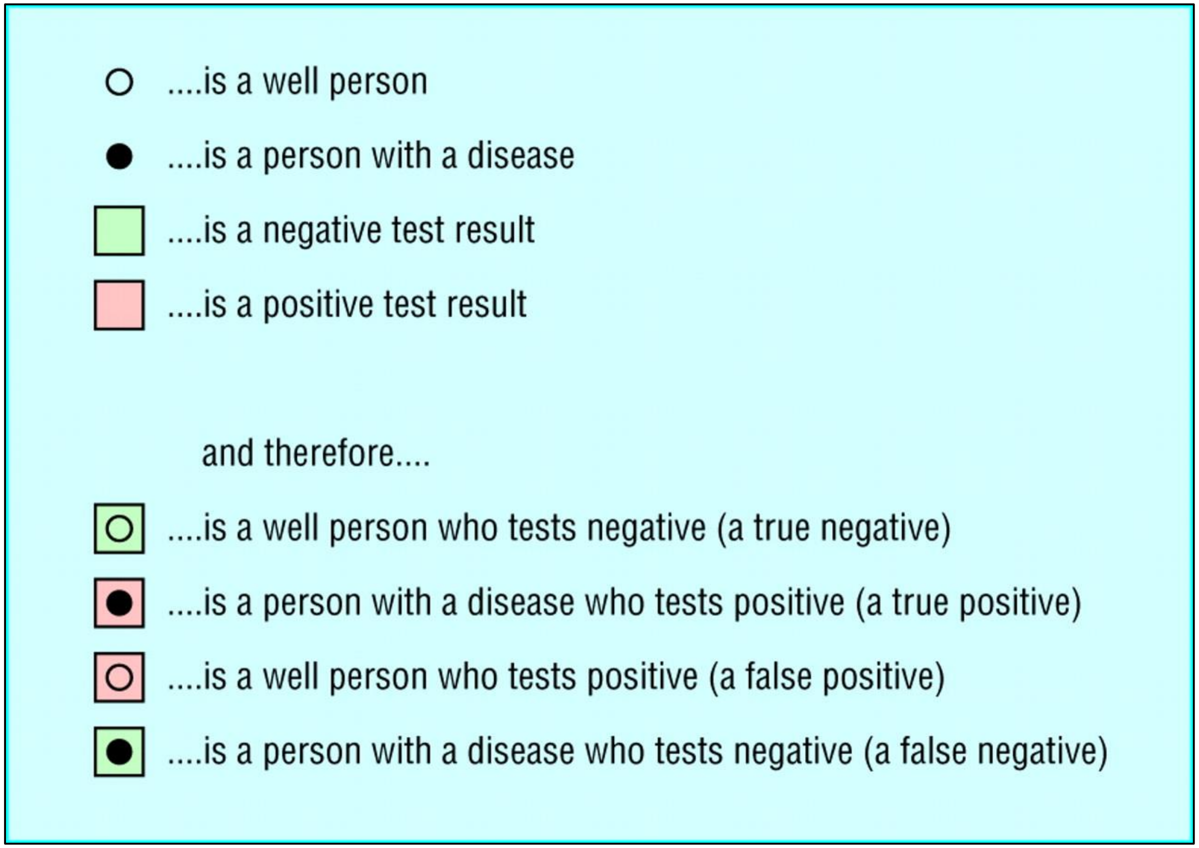
- **Sensitivity = 100%**
  - **Test Predicts that all sick people are sick**
- ~~Sensitivity = 0%~~
  - ~~Test Predicts that all sick people are healthy~~

Perfect Test

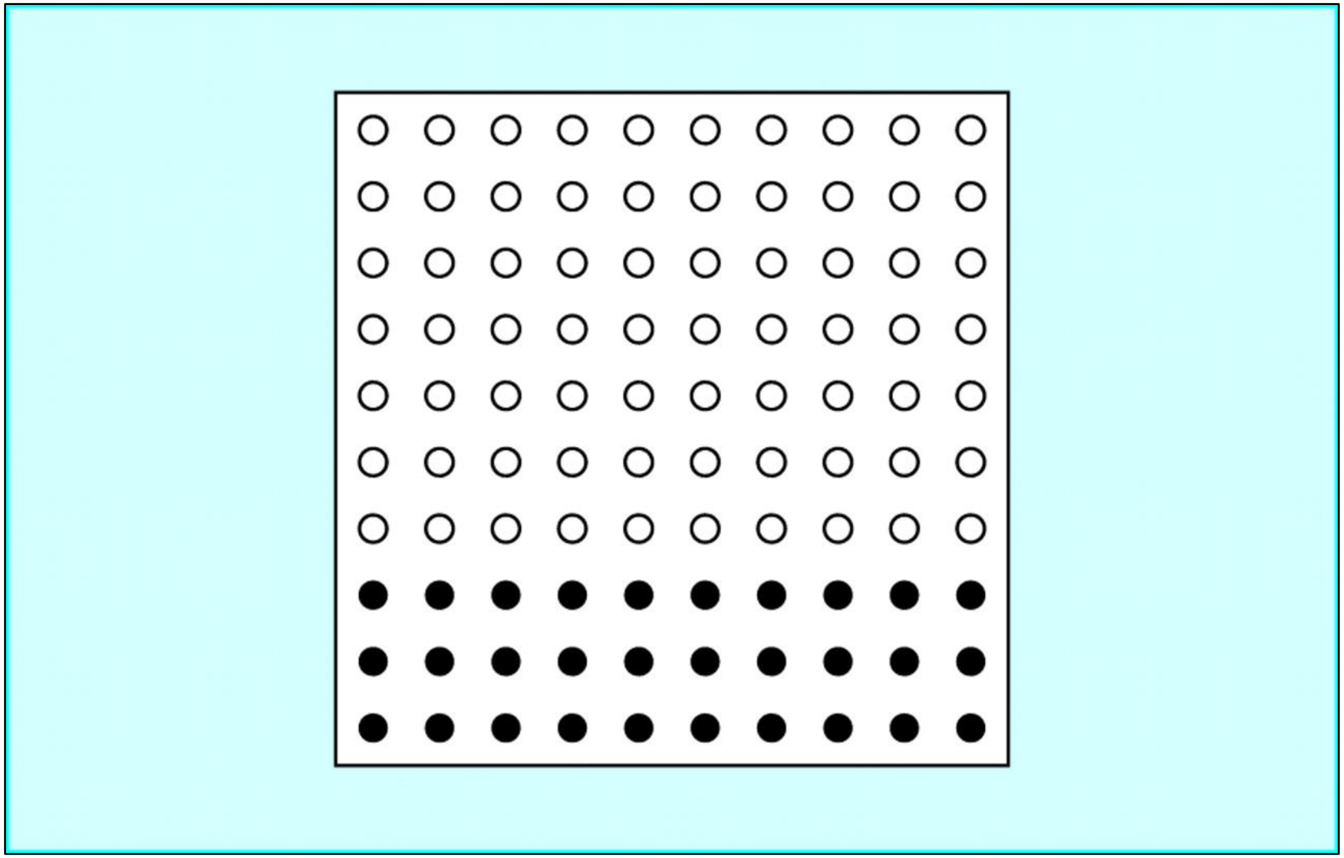
- Specificity = 100%
  - Test Predicts that all healthy people are healthy

- Sensitivity = 100%
  - Test Predicts that all sick people are sick

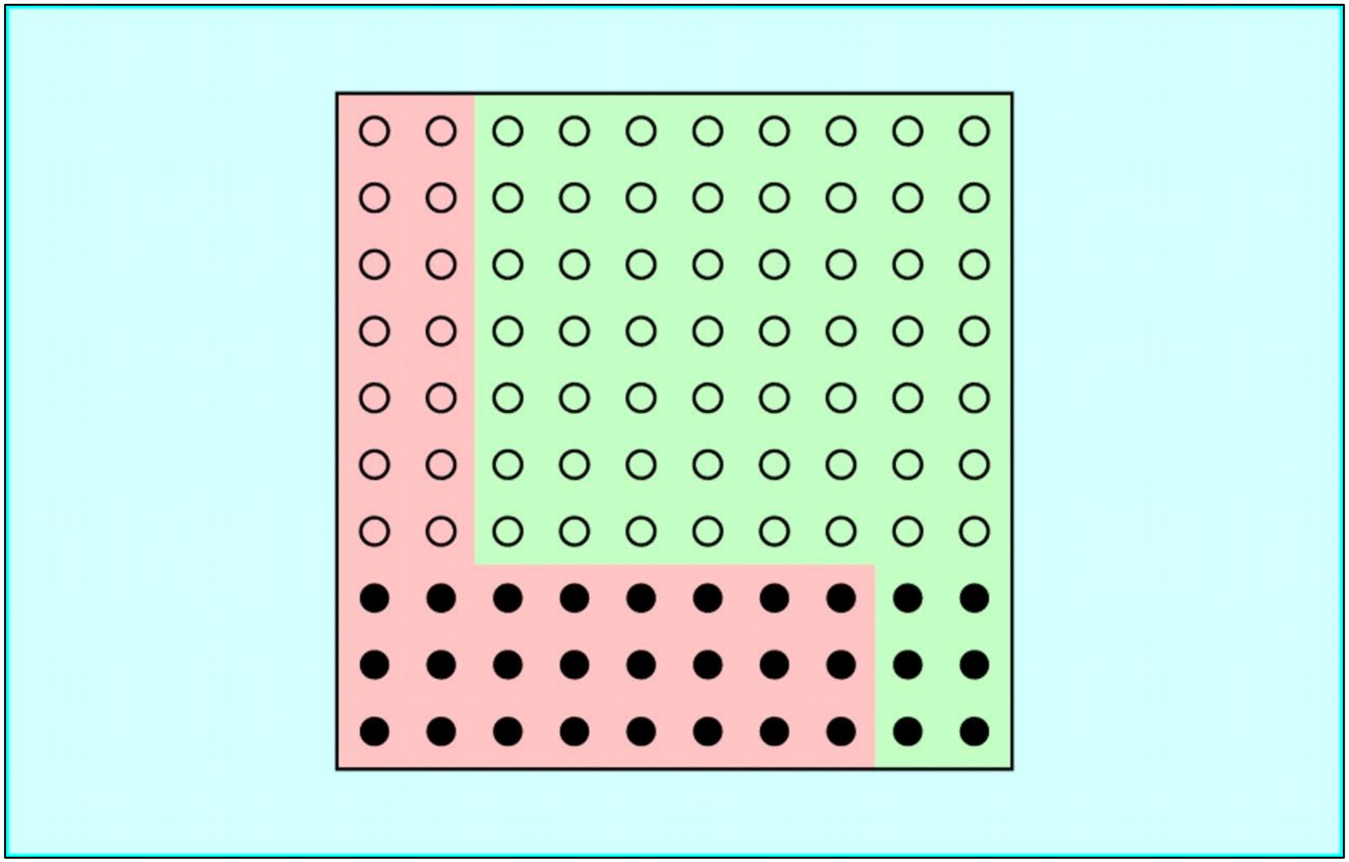
Visual Demonstration Example



Population = 100
70% are healthy ; 30% are sick

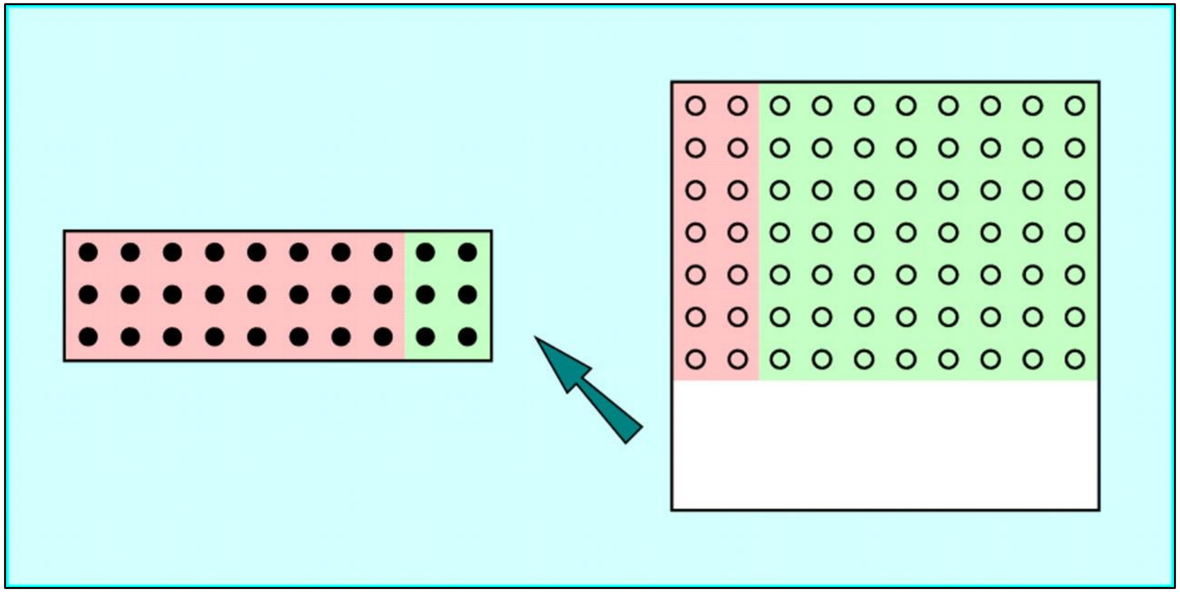


Result of a Test



Sensitivity

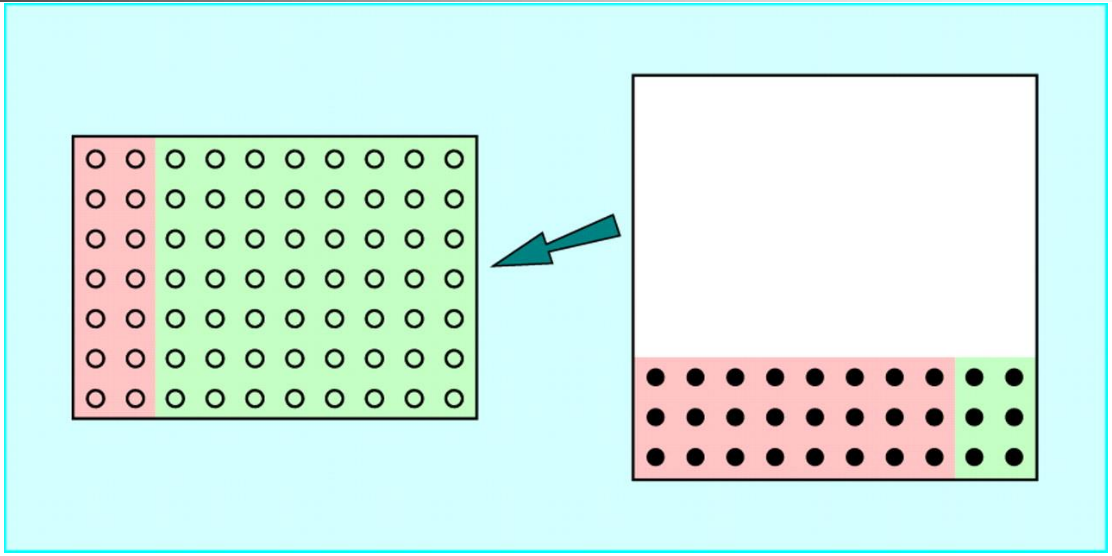
Person has a disease	Sensitivity: Probability that the test is positive given you have the disease	False Negative Rate (FNR): Probability that the test is negative given you have the disease



- Sensitivity = 24/30 = 80%
- False Negative Rate (FNR) = 6/30 = 20%
- Sensitivity + FNR = 100%

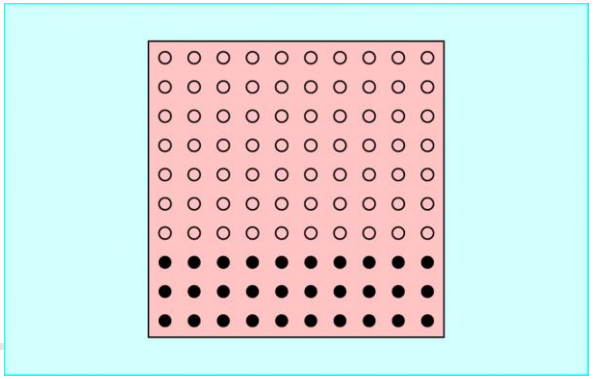
Specificity

Person DOES NOT has a disease (a healthy person)	Specificity: Probability that the test is negative given you do not have the disease	False Positive Rate (FPR): Probability that the test is positive given you do not have the disease



- Specificity = 56/70 = 80%
- False Positive Rate (FPR) = 14/70 = 20%
- Specificity + FPR = 100%

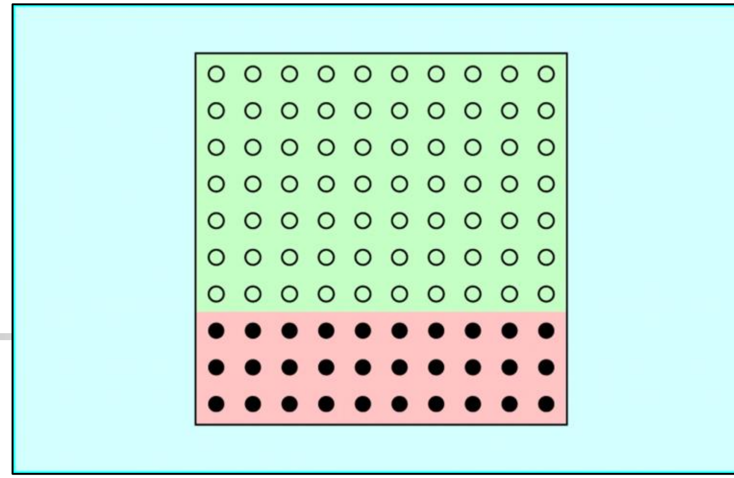
If a Test is too **Strict**
Specificity = 0%, Sensitivity =100%



- ~~Specificity = 100%~~
  - ~~Test Predicts that all healthy people are healthy~~
- **Specificity = 0%**
  - **Test Predicts that all healthy people are sick**

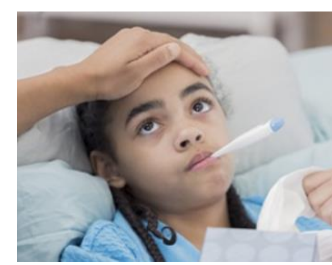
- **Sensitivity = 100%**
  - **Test Predicts that all sick people are sick**
- ~~Sensitivity = 0%~~
  - ~~Test Predicts that all sick people are healthy~~

Perfect Test



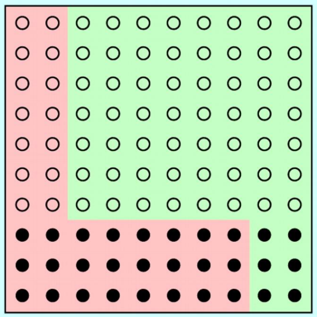
- Specificity = 100%
  - Test Predicts that all healthy people are healthy

- Sensitivity = 100%
  - Test Predicts that all sick people are sick



Computing Sensitivity and Specificity from Confusion Matrix

Computing Sensitivity + Specificity from Confusion Matrix



	CORRECT DECISION	**ERROR**
Person has a disease	Sensitivity: Probability that the test is positive given you have the disease	False Negative Rate (FNR): Probability that the test is negative given you have the disease
Person DOES NOT has a disease (a healthy person)	Specificity: Probability that the test is negative given you do not have the disease	False Positive Rate (FPR): Probability that the test is positive given you do not have the disease

		Reality	Reality	
		No Condition	Condition	Total
Test	No Condition	56	6	29
Test	Condition	14	24	23
		70	30	100

- Sensitivity = 24/30 = 80.0%; False Negative Rate = 6/30 = 20.0%
- Specificity = 56/70 = 80.0%; False Positive Rate = 14/70 = 20.0%

Computing Sensitivity + Specificity from Confusion Matrix Example 2

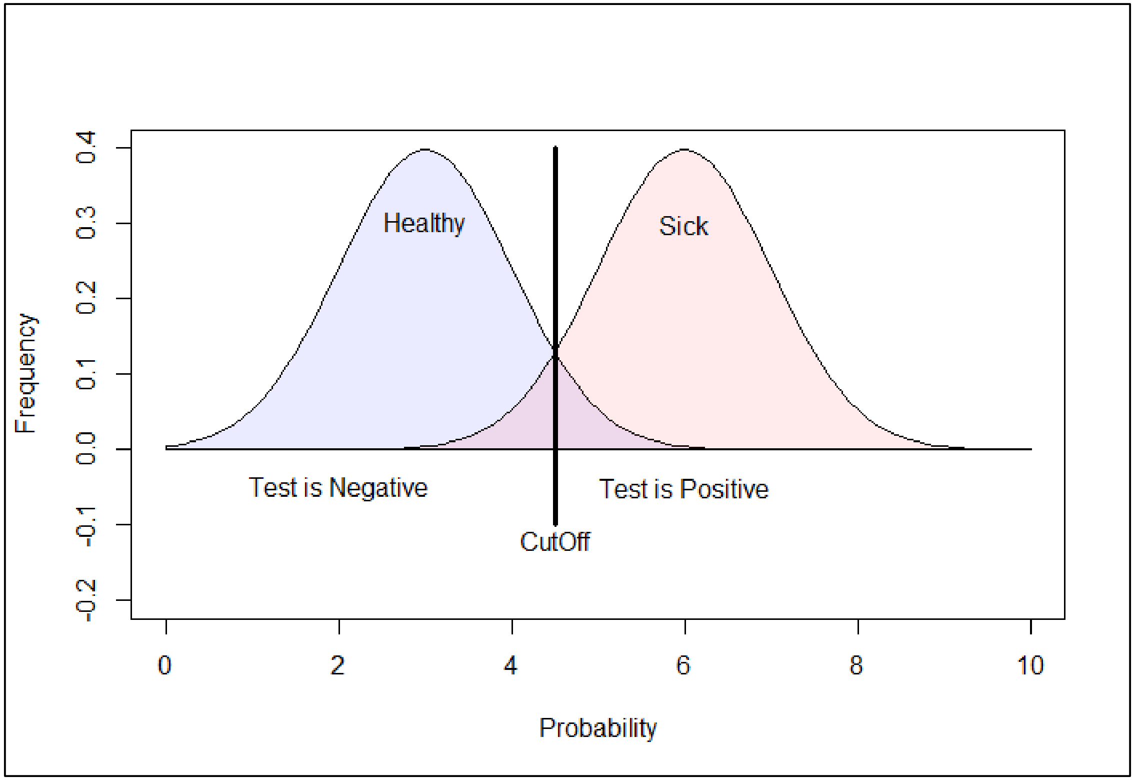
	CORRECT DECISION	ERROR
Person has a disease	Sensitivity: Probability that the test is positive given you have the disease	False Negative Rate (FNR): Probability that the test is negative given you have the disease
Person DOES NOT has a disease (a healthy person)	Specificity: Probability that the test is negative given you do not have the disease	False Positive Rate (FPR): Probability that the test is positive given you do not have the disease

		Reality	Reality	
		No Condition	Condition	Total
Test	No Condition	27	2	29
Test	Condition	10	13	23
		37	15	52

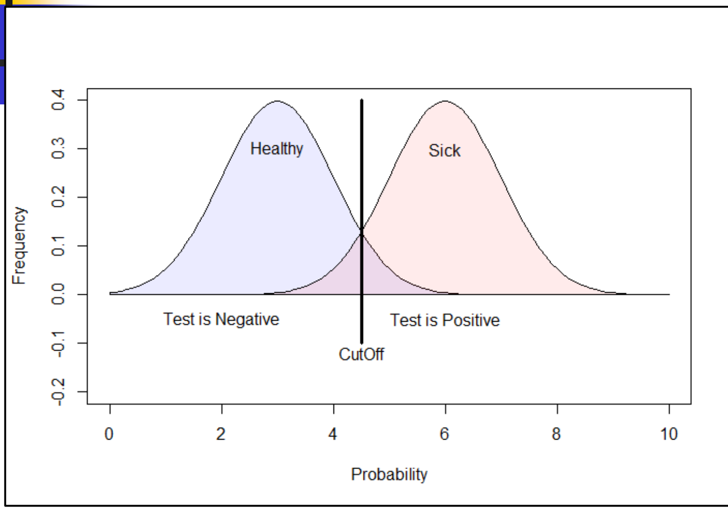
- Sensitivity = 13/15 = 86.6%; False Negative Rate = 2/15 = 13.3%
- Specificity = 27/37 = 72.9%; False Positive Rate = 10/37 = 27.0%

Visualization of Sensitivity & Specificity Using Data Distribution

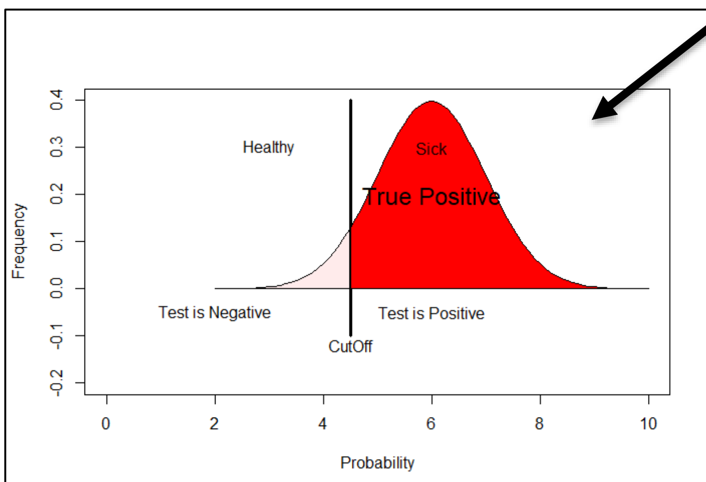
Distribution of Healthy and Sick People

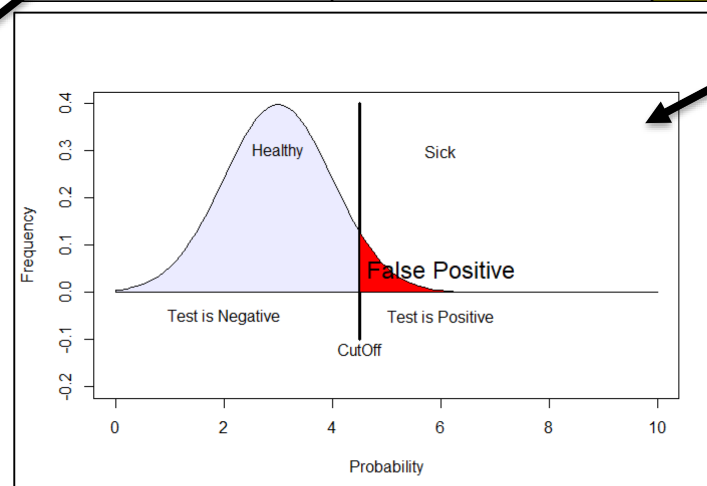


True Positive
False Positive

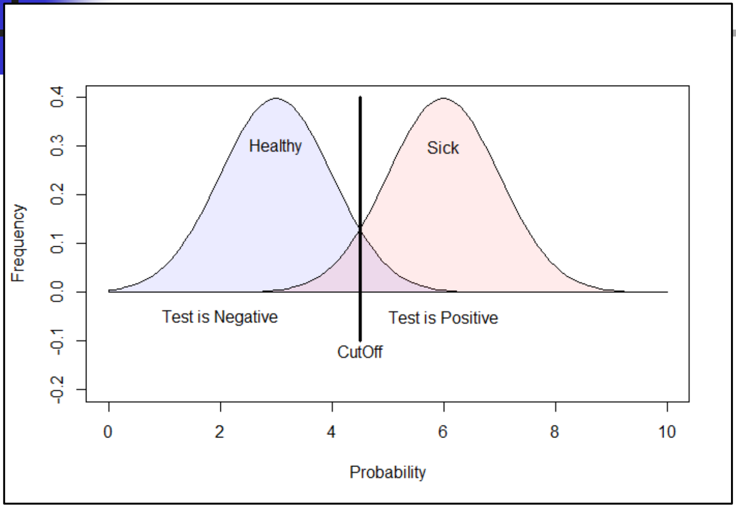


	CORRECT DECISION	ERROR
Person has a disease	Sensitivity: Probability that the test is positive given you have the disease	False Negative Rate (FNR): Probability that the test is negative given you have the disease
Person DOES NOT has a disease (a healthy person)	Specificity: Probability that the test is negative given you do not have the disease	False Positive Rate (FPR): Probability that the test is positive given you do not have the disease

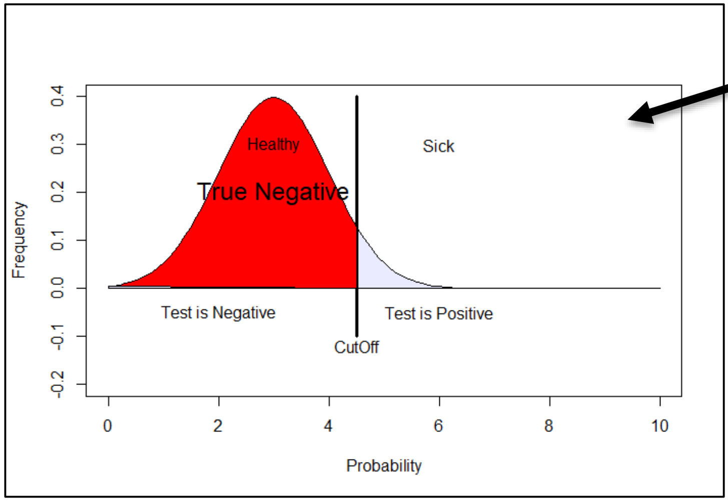


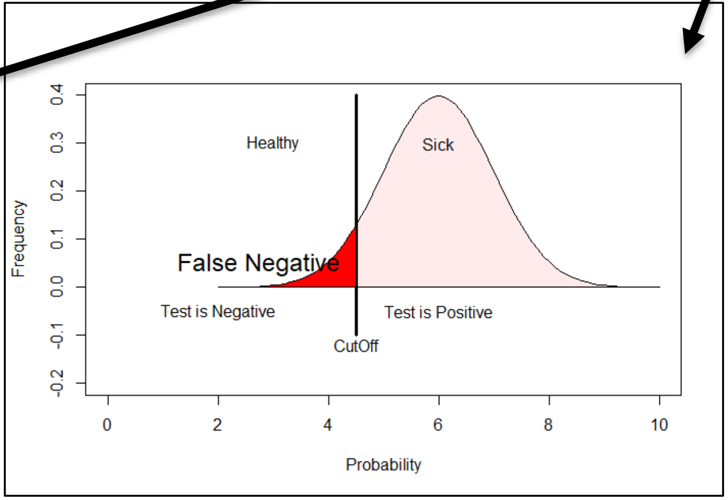


True Negative
False Negative



	CORRECT DECISION	ERROR
Person has a disease	Sensitivity: Probability that the test is positive given you have the disease	False Negative Rate (FNR): Probability that the test is negative given you have the disease
Person DOES NOT has a disease (a healthy person)	Specificity: Probability that the test is negative given you do not have the disease	False Positive Rate (FPR): Probability that the test is positive given you do not have the disease

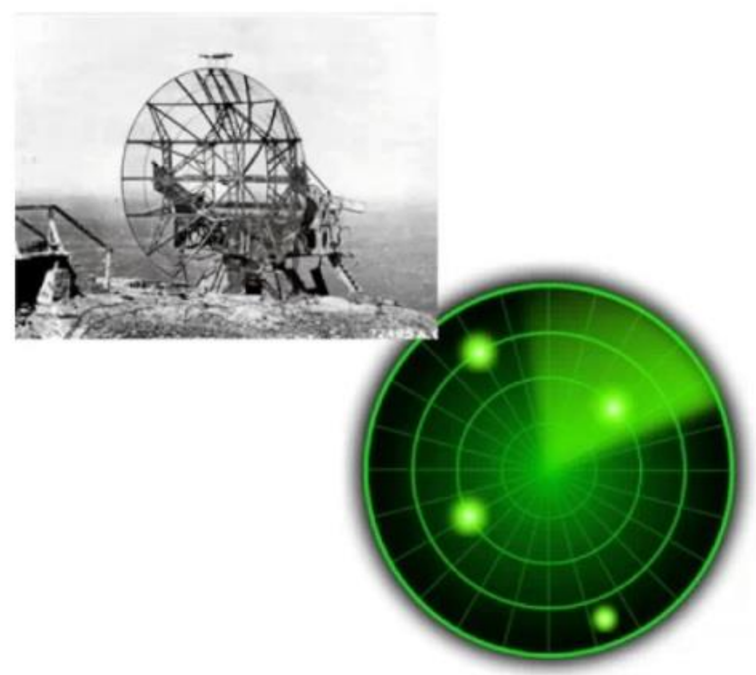




ROC Curve

ROC: Receiver Operating Characteristics

- The name ROC comes from communication theory
- The ROC curve was first developed by electrical engineers and radar engineers during World War II for detecting enemy objects in battlefields
- ROC analysis since then has been used in
  - medicine
  - radiology
  - biometrics

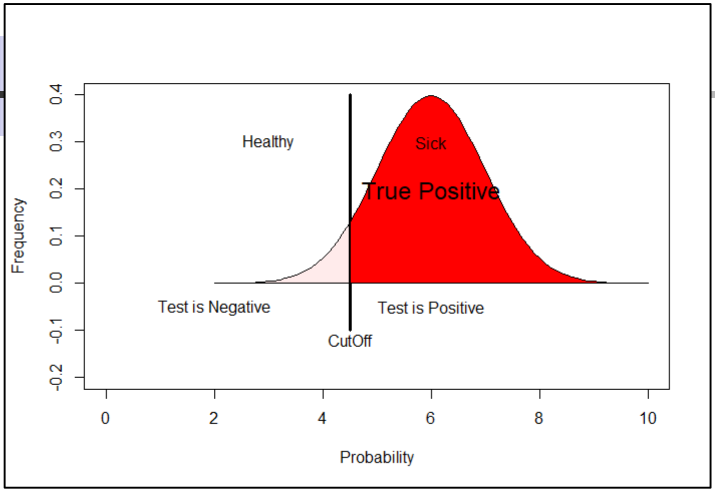


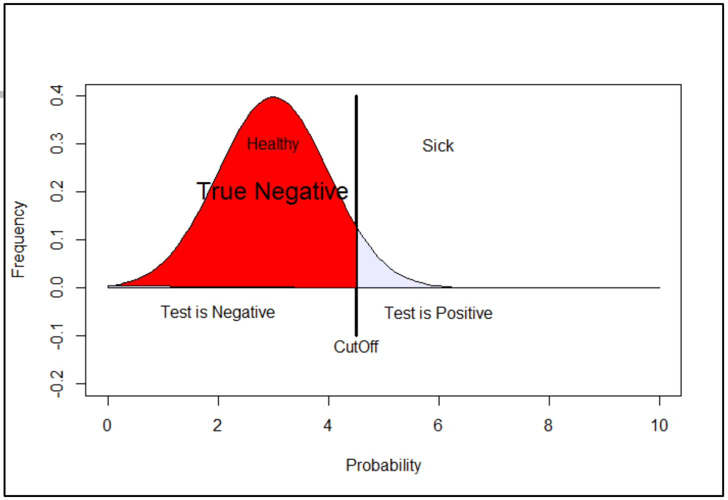
ROC Curves

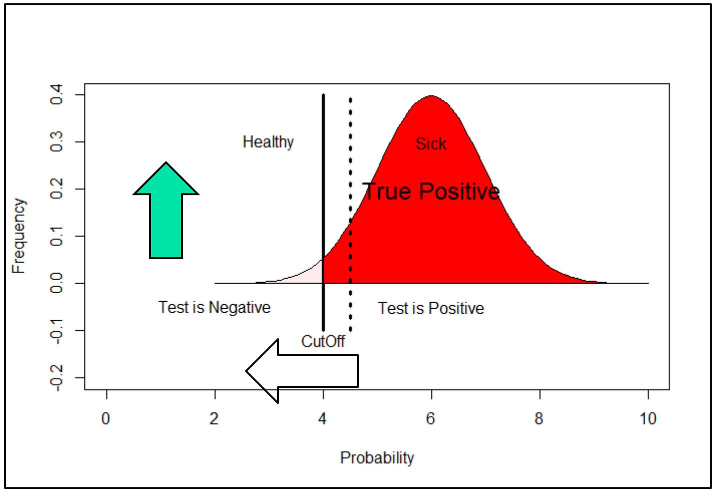
- In Machine Learning, a receiver operating characteristic (ROC), or ROC curve, is a graphical plot that illustrates the performance of a binary classifier system as its discrimination threshold is varied.
- The curve is created by plotting the
  - Sensitivity: True positive rate (TPR) against the
  - (1-Specificity): False positive rate (FPR) at various threshold settings.

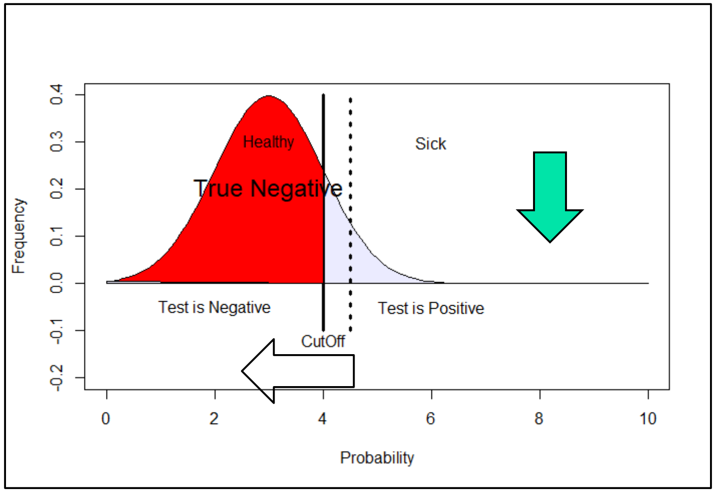
Move the Cutoff Line Left

Sensitivity Increases; Specificity Decreases

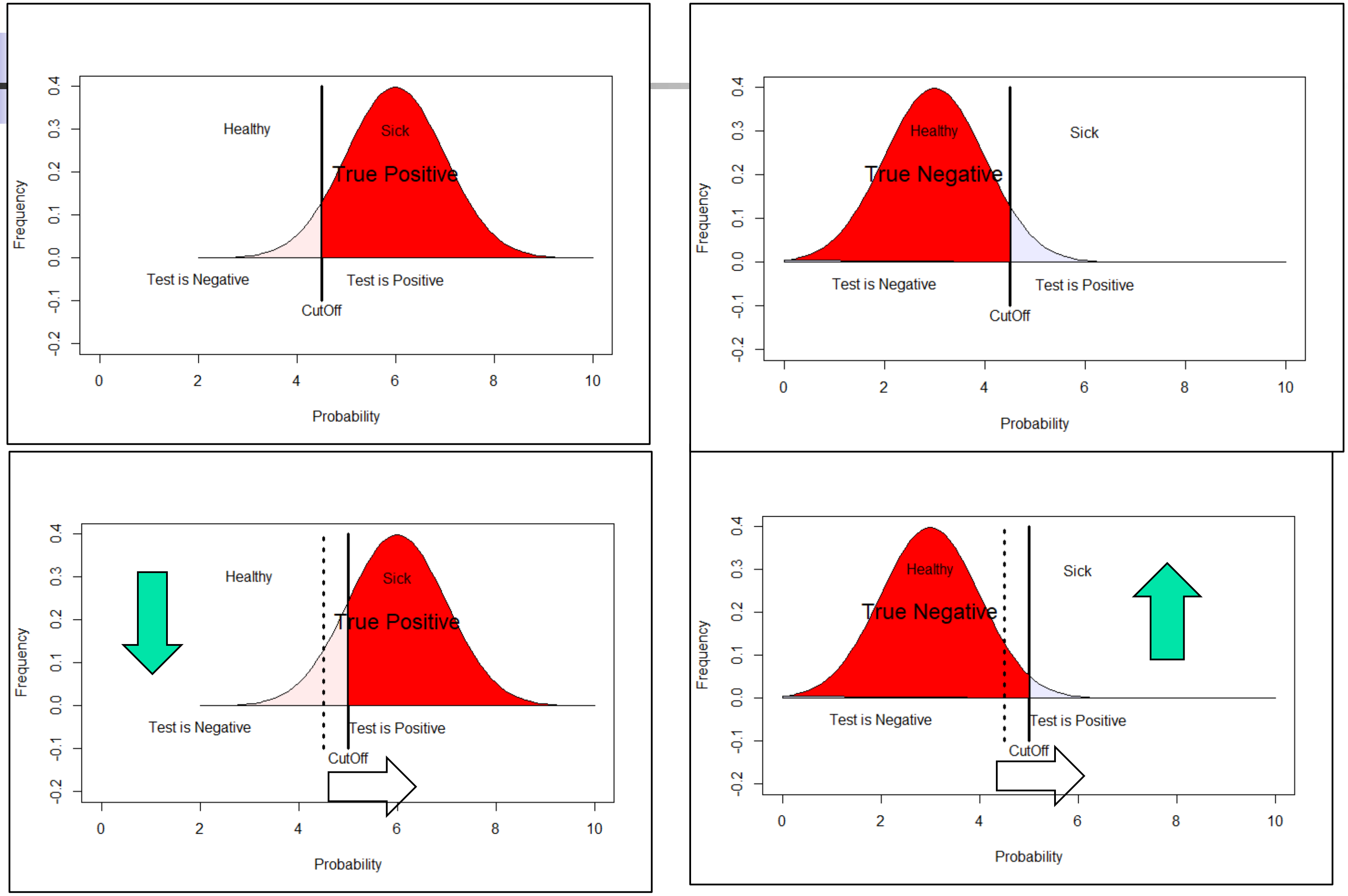






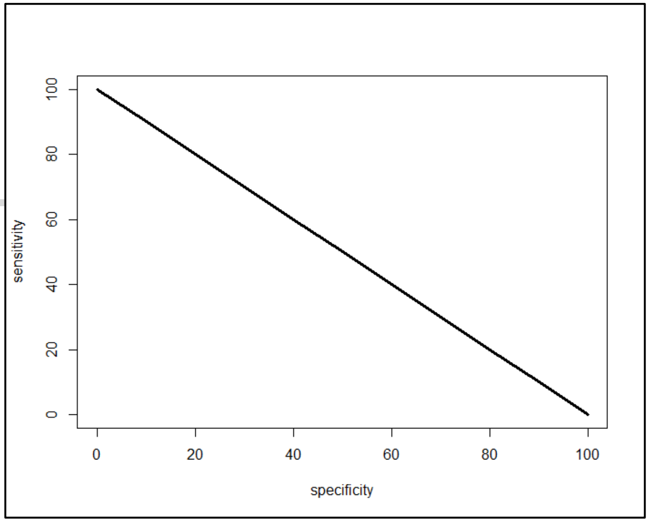


Move the Cutoff Line Right
Sensitivity Decreases; Specificity Increases

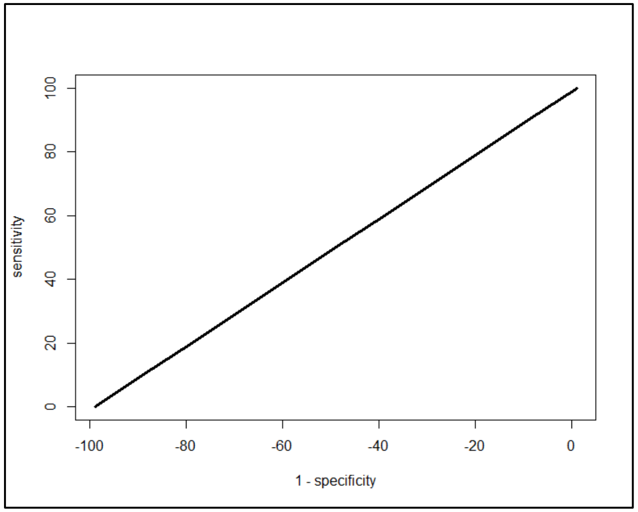


Sensitivity and Specificity are Inversely Proportional as the Cutoff is moved

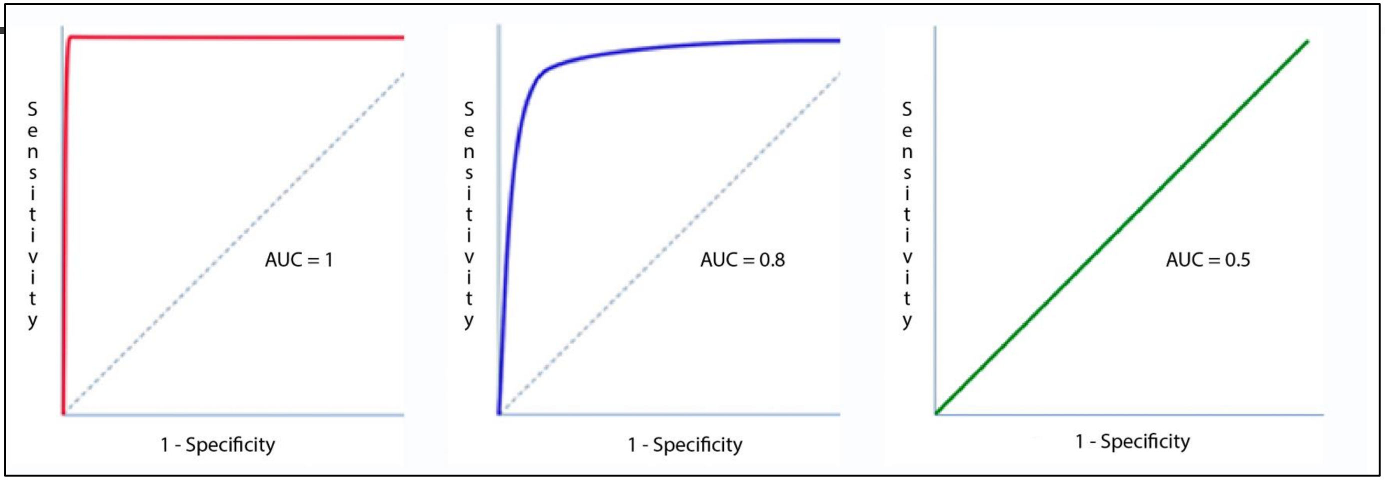
- When Sensitivity = 100%
  - Specificity = 0%
- When Specificity = 100%
  - Sensitivity = 0%

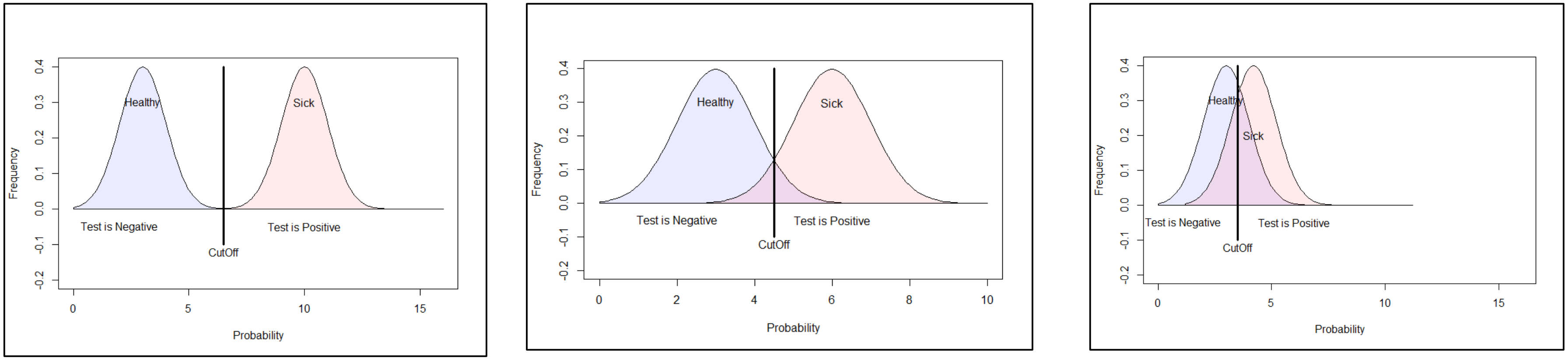


- Plot#1
  - Y: Sensitivity
  - X: Specificity
- Plot#2
  - Y: Sensitivity (True Positive Rate)
  - X: (1- Specificity) or False Positive Rate



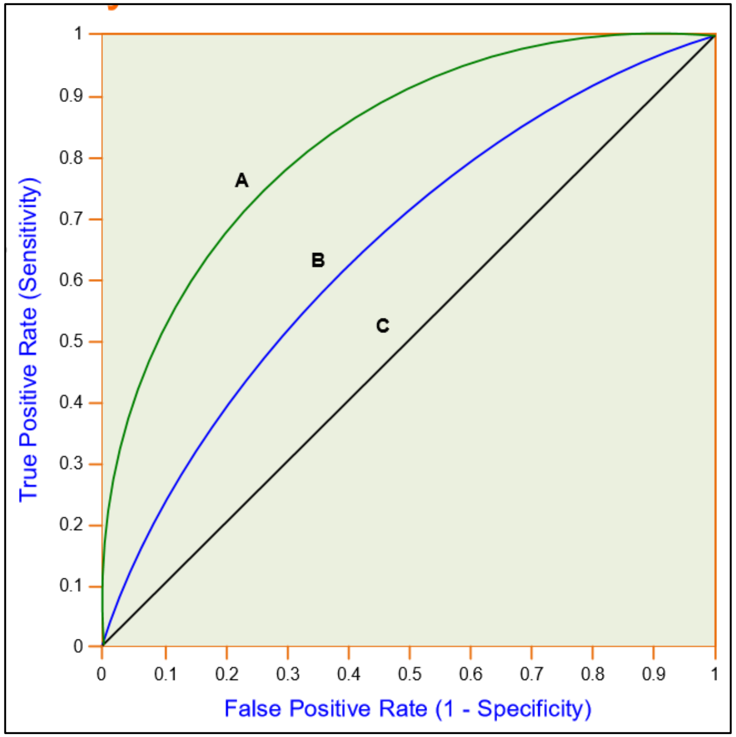
Plot of Sensitivity and (1 – Specificity) is called ROC Curve
AUC: Area Under the Curve



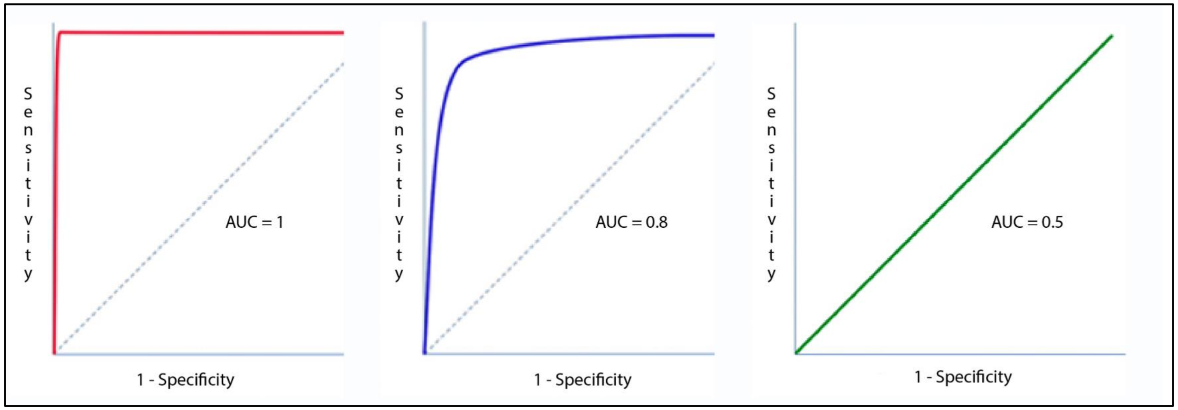


Receiver Operating Characteristic (ROC) Curve

- Sensitivity vs. (1 – Specificity)
- A popular metric for binary classification
- Curve closer to upper-left corner is better
  - A is better than B
  - C is the baseline
    - Denotes 50% (random chance)



AUC: Area Under the Curve



- Best case: Area = 1.0
- Worst Case: Area = 0.5
  - Equivalent to flipping a coin
- Higher the AUC, better the test

Building ROC Curves in R

Generate Random Numbers

```
> ##################################
> # Generate training and testing data similar to IRIS

 set.seed(100)
> numbers1 = rnorm(400)
> rows1 = 40
> columns1 = 10
➢ training = matrix(numbers1, rows1, columns1)


> ##############
> #set.seed(100)
> set.seed(0)
> numbers2 = rnorm(400)
> rows2 = 40
> columns2 = 10
> testing = matrix(numbers2, rows2, columns2)
>
```

Raw Data: Training + Testing

```
> ##############################
> head(training)
           [,1]       [,2]        [,3]       [,4]         [,5]        [,6]
[,7]      [,8]        [,9]      [,10]
[1,] -0.50219235 -0.1016292  0.89682227 -0.7737134 -0.242269499  0.02817177 -
0.7106219 1.1365325 -1.22228428 -1.4006790
[2,]  0.13153117  1.4032035 -0.04999577  0.4240024  0.059031382 -0.35670341
2.6133190 0.4217728  0.89119404  3.3041511
[3,] -0.07891709 -1.7767756 -1.34534931 -0.5839470 -0.177271868  0.85262638 -
1.6266474 1.3500826  0.25392284  0.8567775
[4,]  0.88678481  0.6228674 -1.93121153  0.4150357  0.794680268  0.51336525 -
1.6073063 1.1037569 -0.06581643  1.1610164
[5,]  0.11697127 -0.5222834  0.70958158 -1.5452617  0.006737787  1.01820300
0.3403174 0.6470461  0.20146603  0.2789369
[6,]  0.31863009  1.3222310 -0.15790503 -0.5187495 -0.629790293 -1.02147908
2.7278877 0.1756358  2.47770051 -0.0135485
> head(testing)
          [,1]       [,2]       [,3]       [,4]       [,5]       [,6]         [,7]
[,8]       [,9]      [,10]
[1,]  1.2629543  1.7579031 -0.7970895 -0.1187920  0.3178857 -1.0457177  0.501321828
0.9514985  0.4345367  0.7294513
[2,] -0.3262334  0.5607461  1.2540831  0.1976843 -0.4888056 -0.8962113 -1.013539670
-1.1131230 -0.5195367  0.2626652
[3,]  1.3297993 -0.4527840  0.7721422 -1.0686927  2.6586580  1.2693872  1.614752235
0.6169665 -0.8345590  0.5436579
[4,]  1.2724293 -0.8320433 -0.2195156 -0.8032132  1.6802782  0.5938409  0.005641985
0.5134937 -0.7566476  1.0410603
[5,]  0.4146414 -1.1665705 -0.4248103 -1.1137651  0.7795840  0.7756343 -2.904899060
0.3694591  1.0895035  0.1975062
[6,] -1.5399500 -1.0655906 -0.4189801  1.5800917  0.7132405  1.5573704 -1.107164819
1.7238941  1.5724329 -1.6295783
```

Generate Response Variable Data: Training + Testing

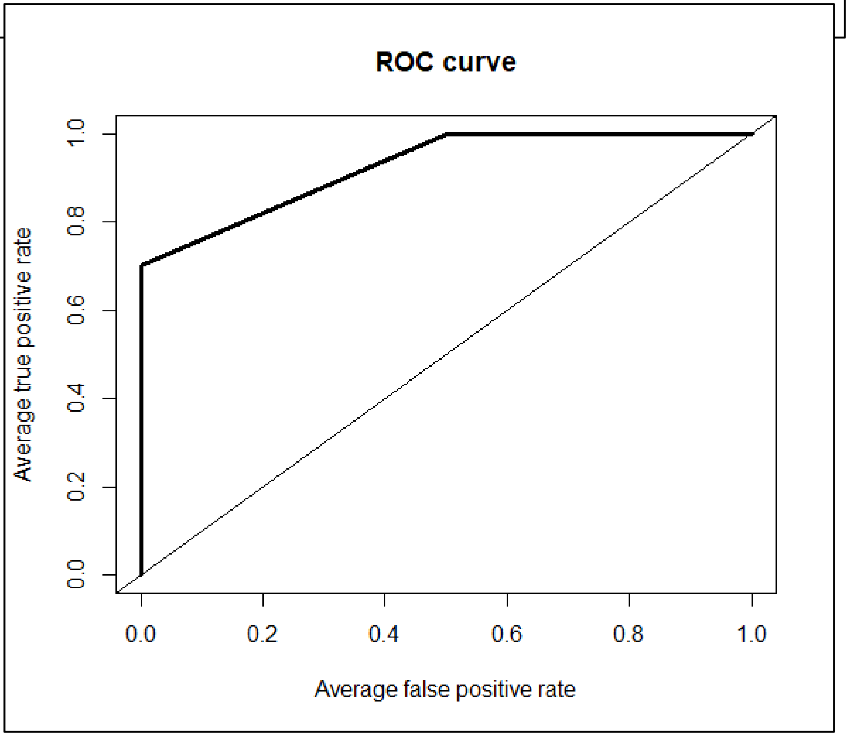
```
> ###############################################
> # Binary Response Variable for ROC Curve
> #
> # Generate class labels training data
> (cl_training <- rep(c(-1, 1), each=20))
 [1] -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  1  1  1  1  1  1
1  1  1  1  1  1  1  1  1  1  1  1  1  1
> length(cl_training)
[1] 40


> # Generate class labels testing data
> (cl_testing <- rep(c(-1,1),each=20))
 [1] -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  1  1  1  1  1  1
1  1  1  1  1  1  1  1  1  1  1  1  1  1
> length(cl_testing)
[1] 40
>
```

Build the kNN model Compute the Probabilities

```
> ########################################
> # Apply KNN Modeling method
> #
> m1 <- class::knn(training, testing, cl_training, k=2, prob=TRUE)
> m1
 [1] -1 -1 -1 -1 1  -1 -1 -1 -1 -1 1  -1 1  -1 1  -1 -1 1  -1 -1 -1 -1 1  1  1  1  1
1  -1 1  1  1  1  1  1  1  1  1  1  -1
attr(,"prob")
 [1] 1.0 1.0 1.0 1.0 0.5 1.0 1.0 0.5 0.5 0.5 0.5 1.0 0.5 1.0 0.5 0.5 1.0 0.5 1.0 0.5
0.5 0.5 1.0 1.0 1.0 1.0 1.0 1.0 0.5 1.0 1.0 0.5 0.5 1.0 1.0 1.0 1.0
[38] 1.0 1.0 0.5
Levels: -1 1
> # Compute the probabilities
> #
> (prob1 <- attr(m1, "prob"))
 [1] 1.0 1.0 1.0 1.0 0.5 1.0 1.0 0.5 0.5 0.5 0.5 1.0 0.5 1.0 0.5 0.5 1.0 0.5 1.0 0.5
0.5 0.5 1.0 1.0 1.0 1.0 1.0 1.0 0.5 1.0 1.0 0.5 0.5 1.0 1.0 1.0 1.0
[38] 1.0 1.0 0.5
> (prob2 <- 2*ifelse(m1 == "-1", 1-prob1, prob1) - 1)
 [1] -1 -1 -1 -1  0 -1 -1  0  0  0  0 -1  0 -1  0  0 -1  0 -1  0  0  0  1  1  1  1
1  1  0  1  1  0  0  1  1  1  1  1  1  0
```

Build the ROC Curve

```
 ########################################
> library(ROCR)
> pred_knn <- prediction(prob2, cl_testing)
> pred_knn <- performance(pred_knn, "tpr", "fpr")
> plot(pred_knn, avg= "threshold", lwd=3, main="ROC curve")
> abline(a=0,b=1)
```



Summary

- Sensitivity & Specificity
- Computing Sensitivity & Specificity from Confusion Matrix
- Visualization of Sensitivity & Specificity
- ROC Curves
- Building ROC Curves in R

