Logistic Regression

Let’s learn something!

Python and Spark

- Not all labels are continuous, sometimes you need to predict categories, this is known as classification.
- Logistic Regression is one of the basic ways to perform classification (don't be confused by the word "regression")

Background

- We want to learn about Logistic Regression as a method for **Classification.**
- Some examples of classification problems:
  - Spam versus “Ham” emails
  - Loan Default (yes/no)
  - Disease Diagnosis
- Above were all examples of Binary Classification

Background

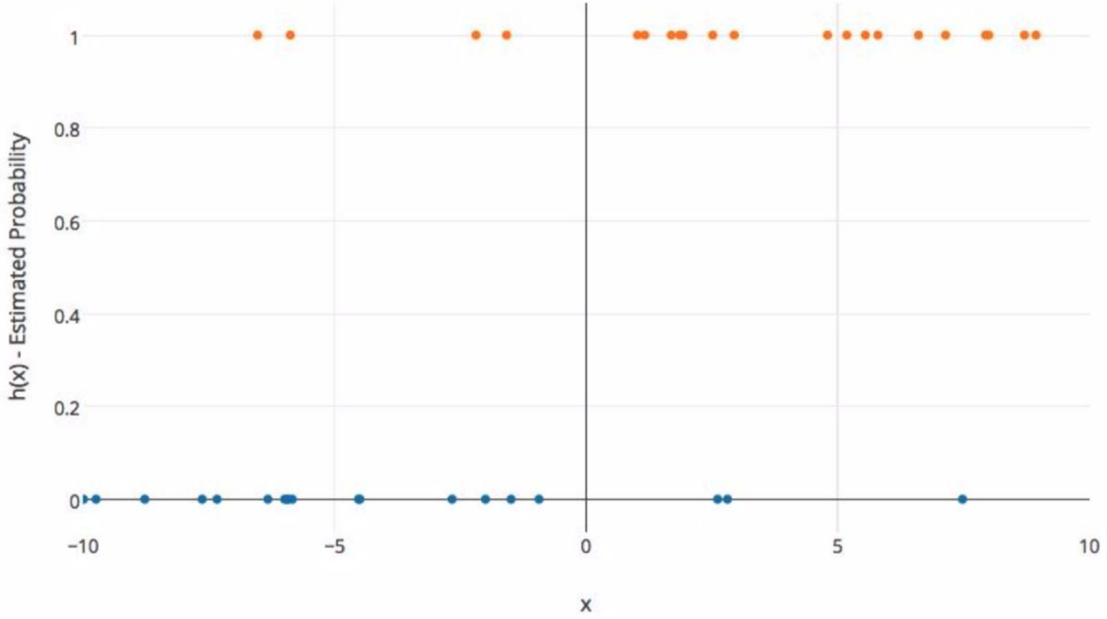
- So far we’ve only seen regression problems where we try to predict a continuous value.
- Although the name may be confusing at first, logistic regression allows us to solve **classification** problems, where we are trying to predict discrete categories.

Background

- The convention for binary classification is to have two classes 0 and 1.
- Let’s walk through the basic idea for logistic regression.
- We’ll also explain why it has the term regression in it, even though it’s used for classification!

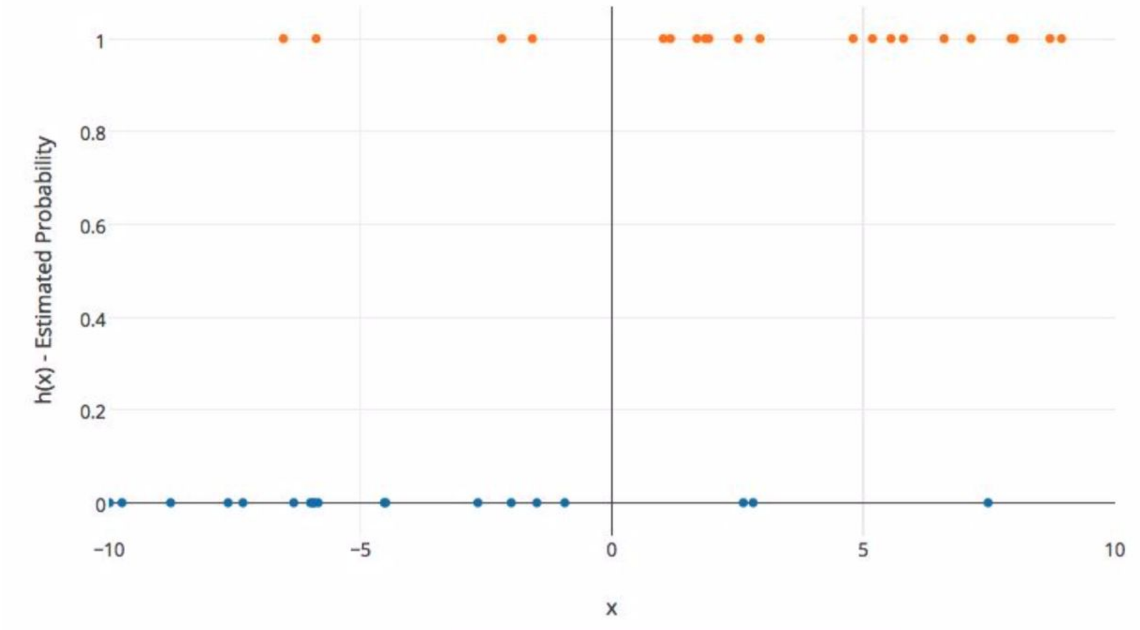
Background

- Imagine we plotted out some categorical data against one feature.



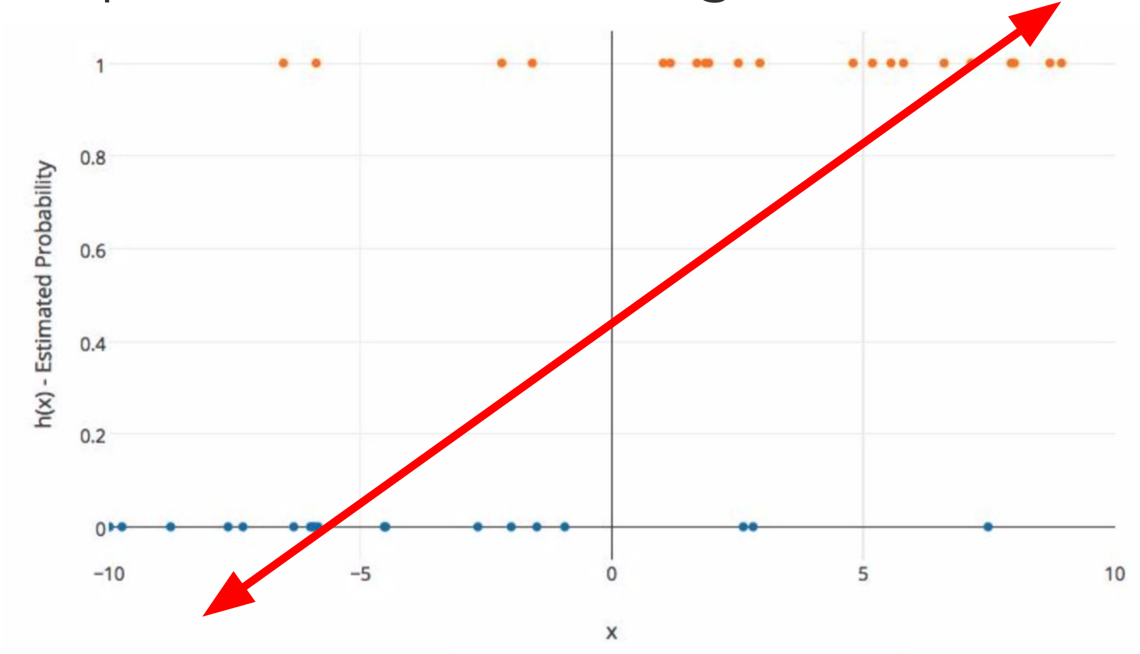
Background

- The X axis represents a feature value and the Y axis represents the probability of belonging to class 1.



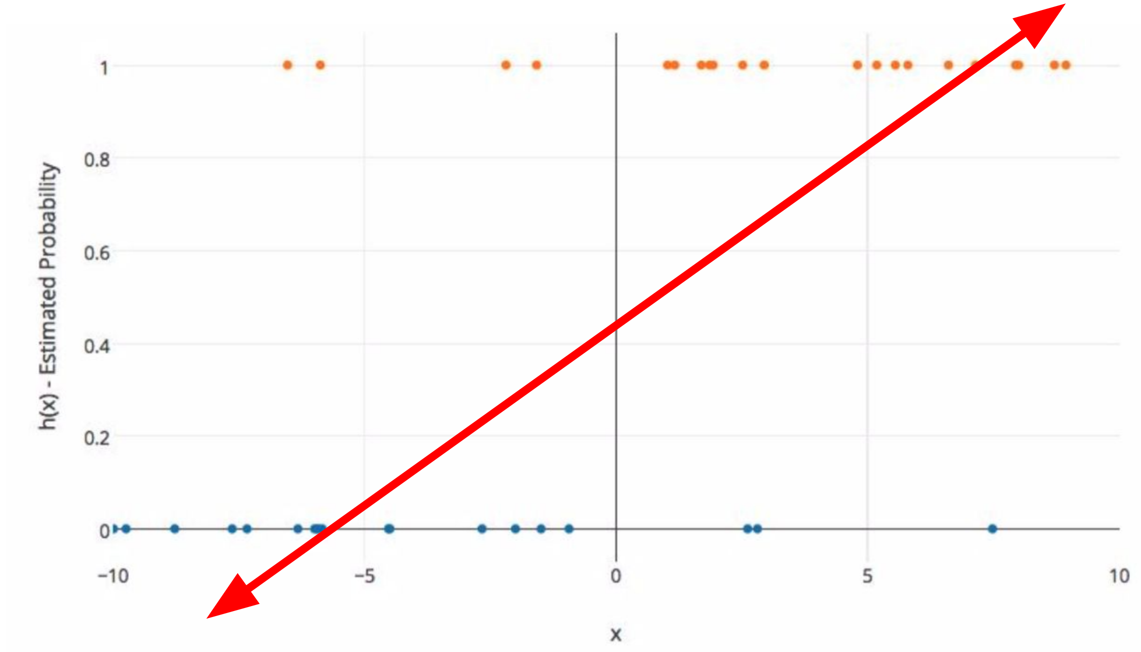
Background

- We can’t use a normal linear regression model on binary groups. It won’t lead to a good fit:



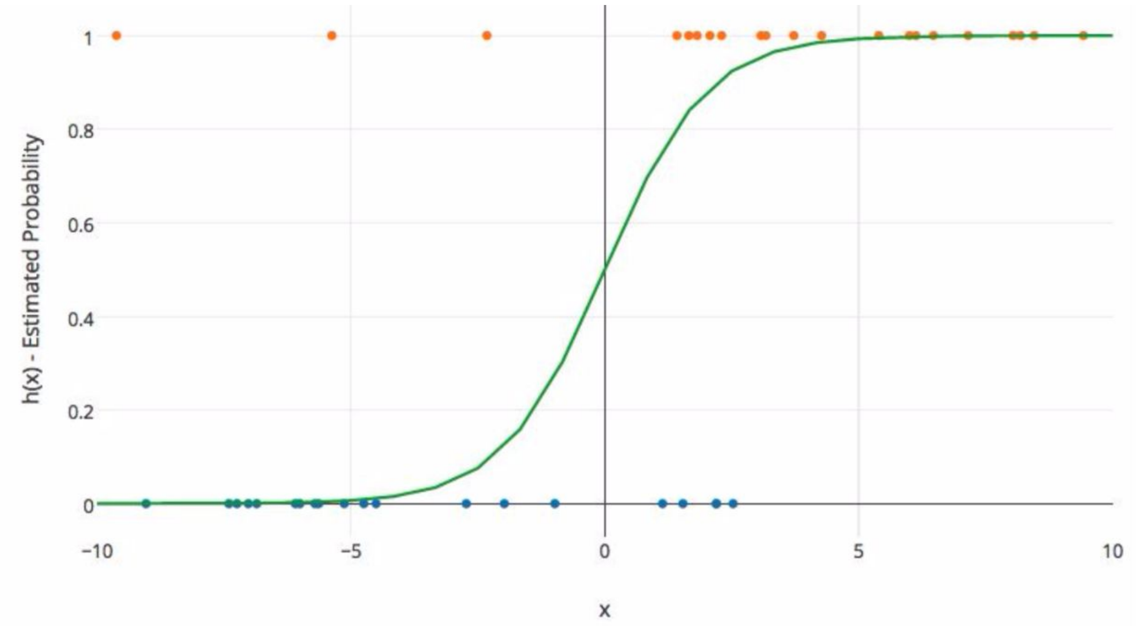
Background

- We need a function that will fit binary categorical data!



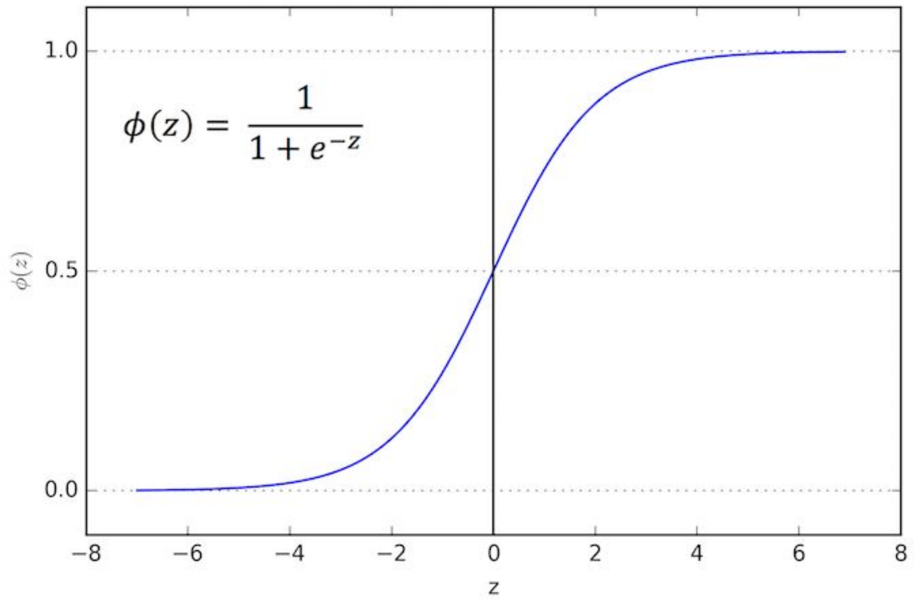
Background

- It would be great if we could find a function with this sort of behavior:



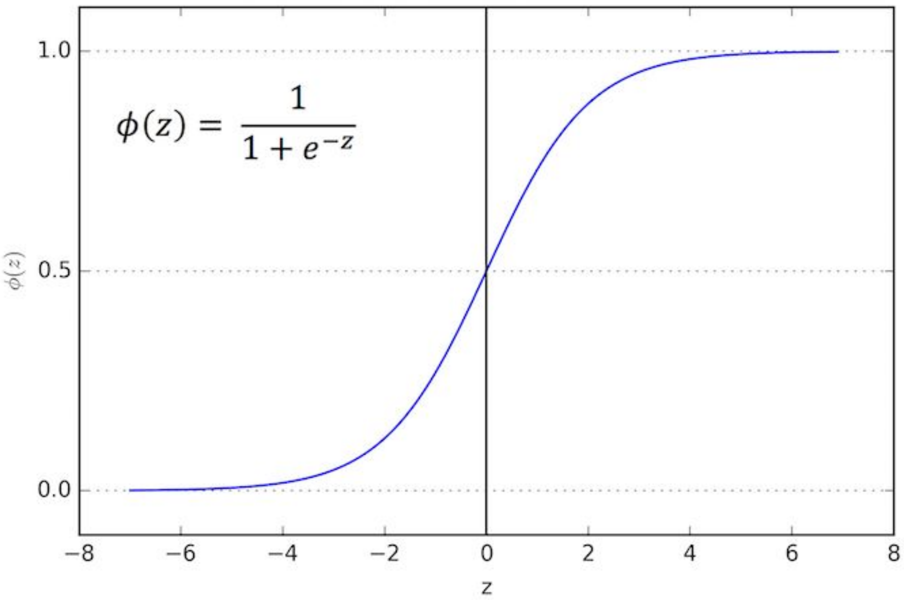
Sigmoid Function

- The Sigmoid (aka Logistic) Function takes in any value and outputs it to be between 0 and 1.



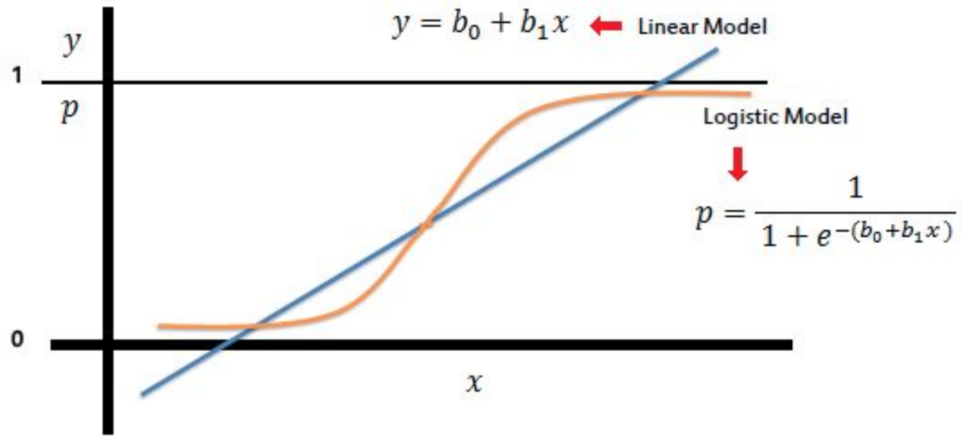
Sigmoid Function

- This means we can take our Linear Regression Solution and place it into the Sigmoid Function.



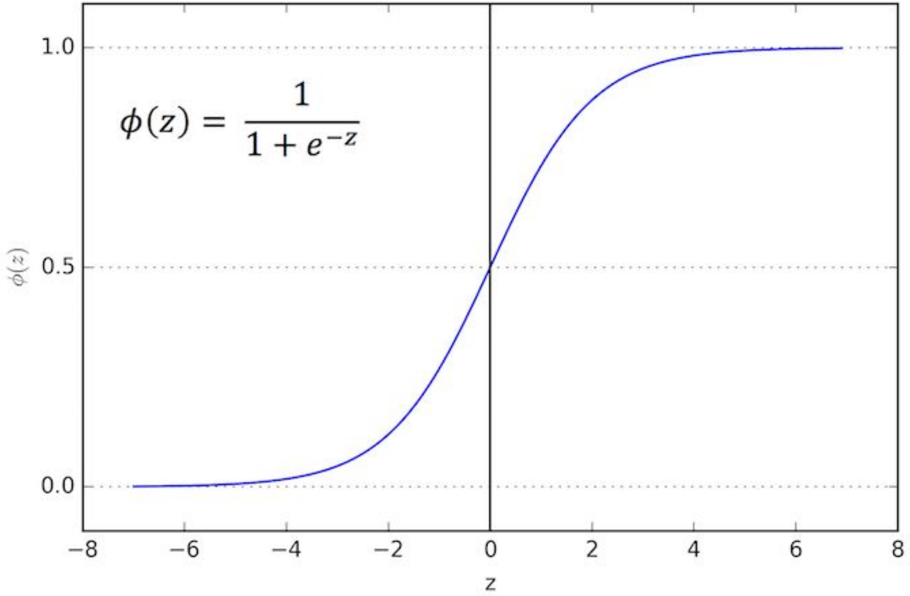
Sigmoid Function

- This means we can take our Linear Regression Solution and place it into the Sigmoid Function.



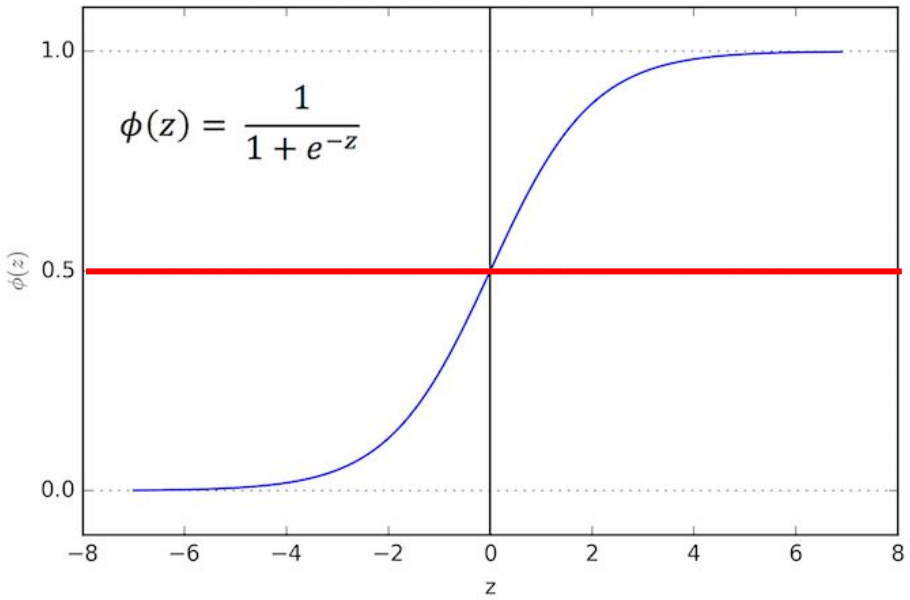
Sigmoid Function

- This results in a probability from 0 to 1 of belonging in the 1 class.



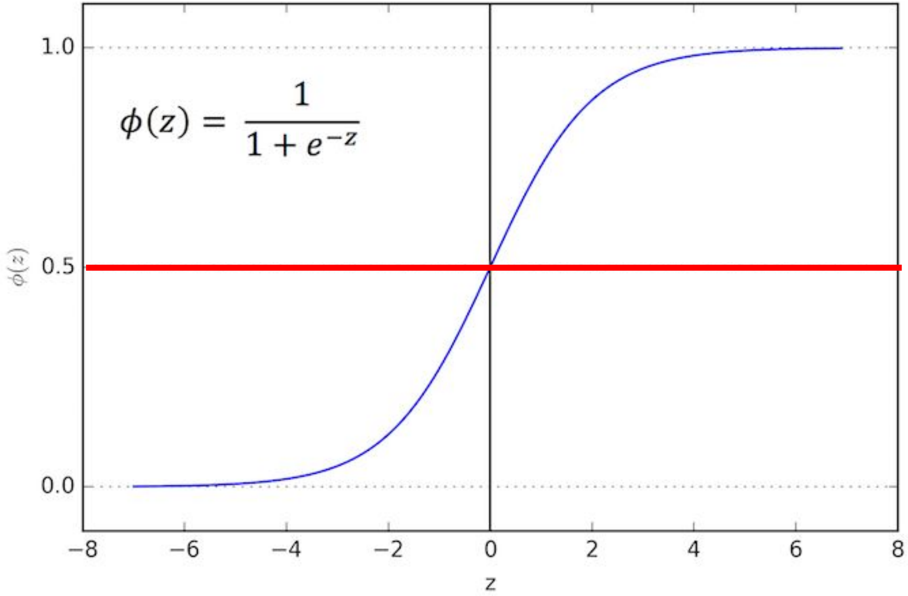
Sigmoid Function

- We can set a cutoff point at 0.5, anything below it results in class 0, anything above is class 1.



Review

- We use the logistic function to output a value ranging from 0 to 1. Based off of this probability we assign a class



Model Evaluation

- After you train a logistic regression model on some training data, you will evaluate your model's performance on some test data.
- You can use a confusion matrix to evaluate classification models.

Confusion Matrix

		predicted condition	
	total population	prediction positive	prediction negative
true condition	condition positive	**True Positive (TP)**	**False Negative (FN)** (type II error)
	condition negative	**False Positive (FP)** (Type I error)	**True Negative (TN)**

Confusion Matrix

		predicted condition		
	total population	prediction positive	prediction negative	Prevalence $= \frac{\Sigma \text{ condition positive}}{\Sigma \text{ total population}}$
true condition	condition positive	**True Positive (TP)**	**False Negative (FN)** (type II error)	True Positive Rate (TPR), Sensitivity, Recall, Probability of Detection $= \frac{\Sigma \text{ TP}}{\Sigma \text{ condition positive}}$
	condition negative	**False Positive (FP)** (Type I error)	**True Negative (TN)**	False Positive Rate (FPR), Fall-out, Probability of False Alarm $= \frac{\Sigma \text{ FP}}{\Sigma \text{ condition negative}}$
	Accuracy $= \frac{\Sigma \text{ TP} + \Sigma \text{ TN}}{\Sigma \text{ total population}}$	Positive Predictive Value (PPV), Precision $= \frac{\Sigma \text{ TP}}{\Sigma \text{ prediction positive}}$	False Omission Rate (FOR) $= \frac{\Sigma \text{ FN}}{\Sigma \text{ prediction negative}}$	Positive Likelihood Ratio (LR+) $= \frac{\text{TPR}}{\text{FPR}}$
		False Discovery Rate (FDR) $= \frac{\Sigma \text{ FP}}{\Sigma \text{ prediction positive}}$	Negative Predictive Value (NPV) $= \frac{\Sigma \text{ TN}}{\Sigma \text{ prediction negative}}$	Negative Likelihood Ratio (LR–) $= \frac{\text{FNR}}{\text{TNR}}$

Model Evaluation

- The main point to remember with the confusion matrix and the various calculated metrics is that they are all fundamentally ways of comparing the predicted values versus the true values.
- What constitutes “good” metrics, will really depend on the specific situation!

Model Evaluation

- We can use a confusion matrix to evaluate our model.
- For example, imagine testing for disease.

n=165	Predicted: NO	Predicted: YES
Actual: NO	50	10
Actual: YES	5	100

Example: Test for presence of disease
NO = negative test = False = 0
YES = positive test = True = 1

Confusion Matrix

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Basic Terminology:

- True Positives (TP)
- True Negatives (TN)
- False Positives (FP)
- False Negatives (FN)

Confusion Matrix

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Accuracy:

- Overall, how often is it **correct**?
- (TP + TN) / total = 150/165 = 0.91

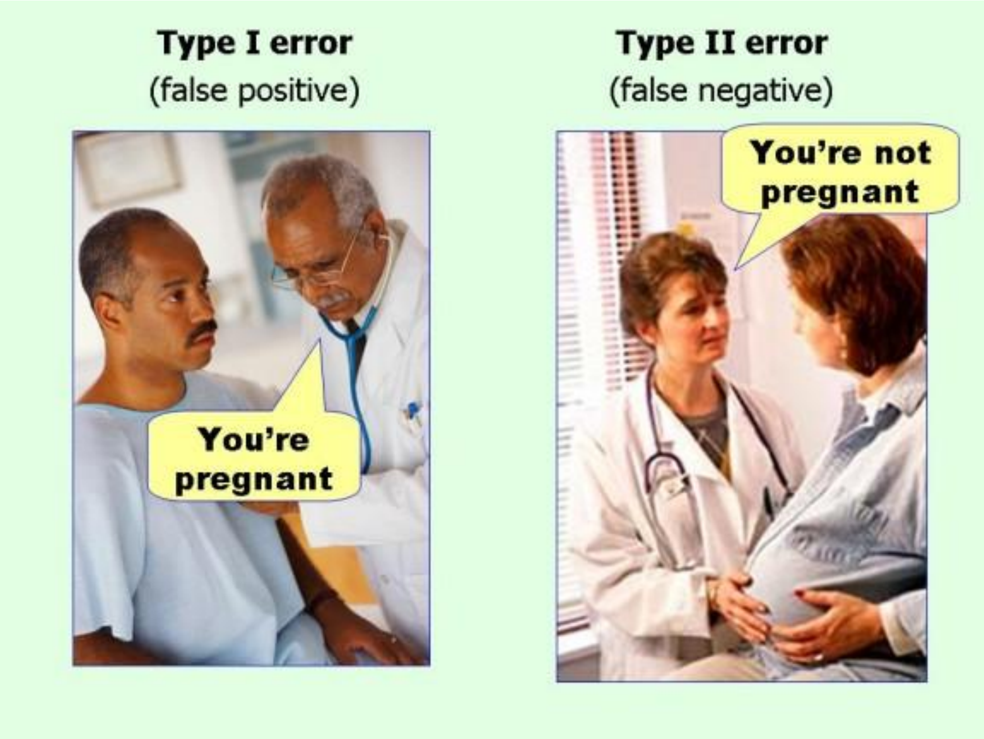
Confusion Matrix

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Misclassification Rate (Error Rate):

- Overall, how often is it **wrong**?
- (FP + FN) / total = 15/165 = 0.09

Confusion Matrix



Model Evaluation

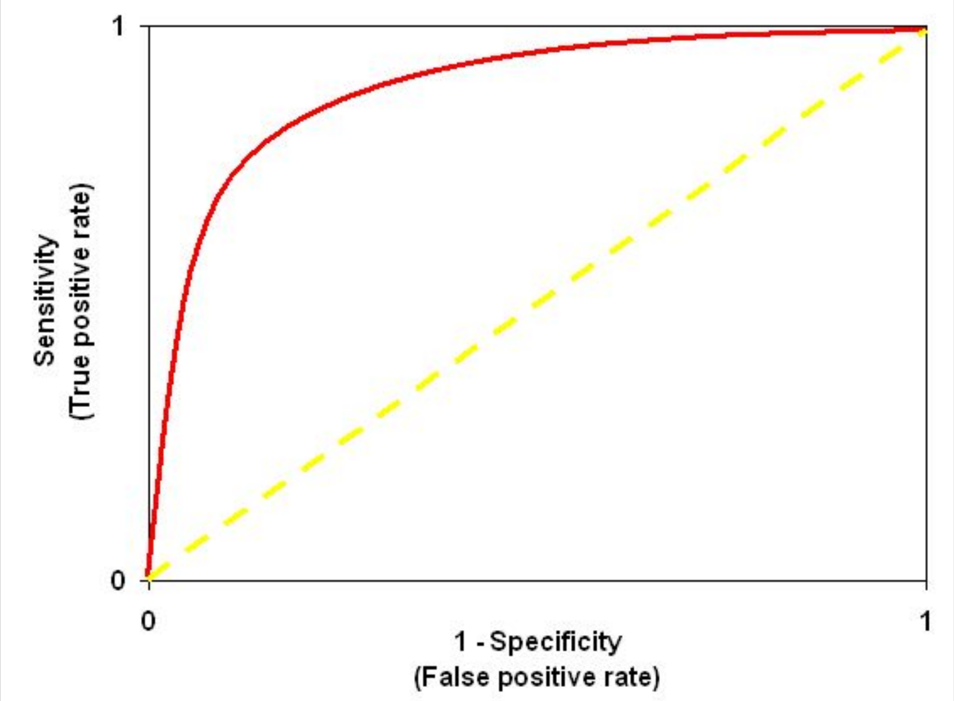
- Still confused on the confusion matrix?
- No problem! Check out the Wikipedia page for it, it has a really good diagram with all the formulas for all the metrics.
- Throughout the course, we'll usually just print out metrics (e.g. accuracy).

Model Evaluation

- Binary classification has some of its own special classification metrics.
- These include visualizations of metrics from the confusion matrix.
- The Receiver Operator Curve (ROC) curve was developed during World War II to help analyze radar data.

Model Evaluation

- The ROC curve:



Model Evaluation