Logistic Regression

Let’s learn something!

- 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")

Reading Assignment

Sections 4-4.3 of

Introduction to Statistical Learning

By Gareth James, et al.

Python and Spark

- If you want to fully understand some of the concepts behind the evaluation methods and metrics behind classification, the reading is highly recommended!

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

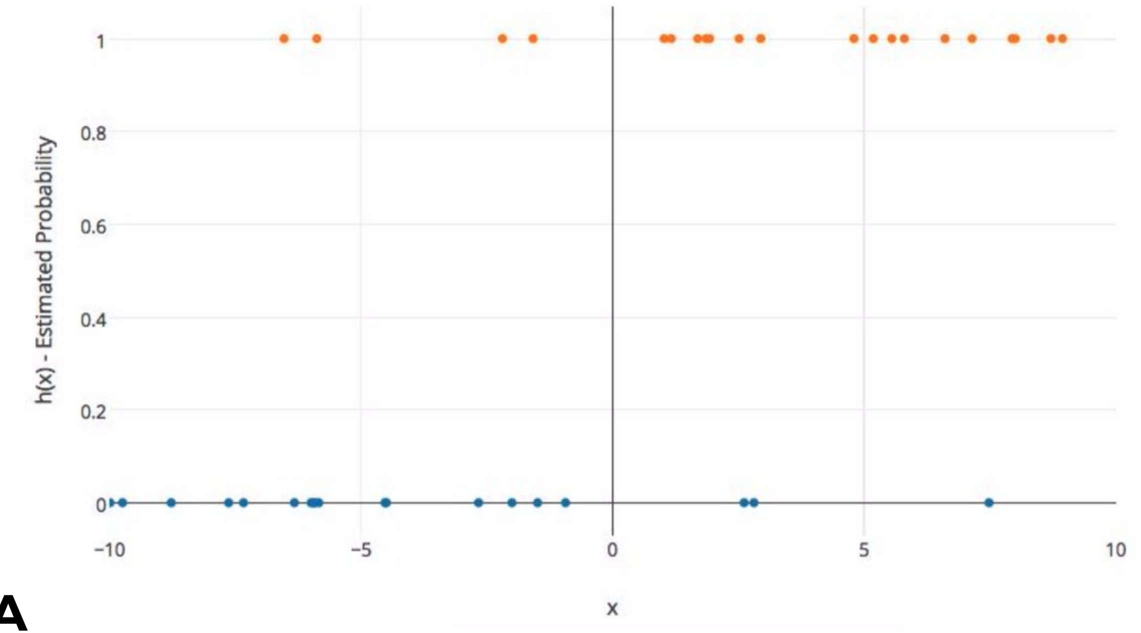
- So far we've only seen regression problems where we try to predict a continuous value.
- Although the name may be confusing at first, logistic regression allows us to solve **classification** problems, where we are trying to predict discrete categories.

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!

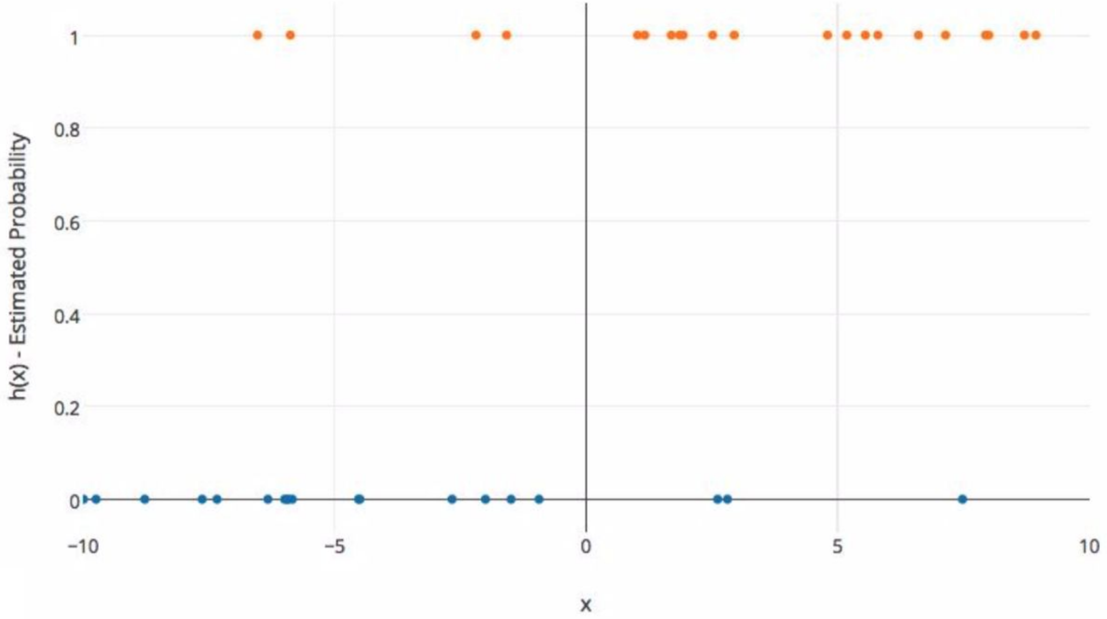
Background

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



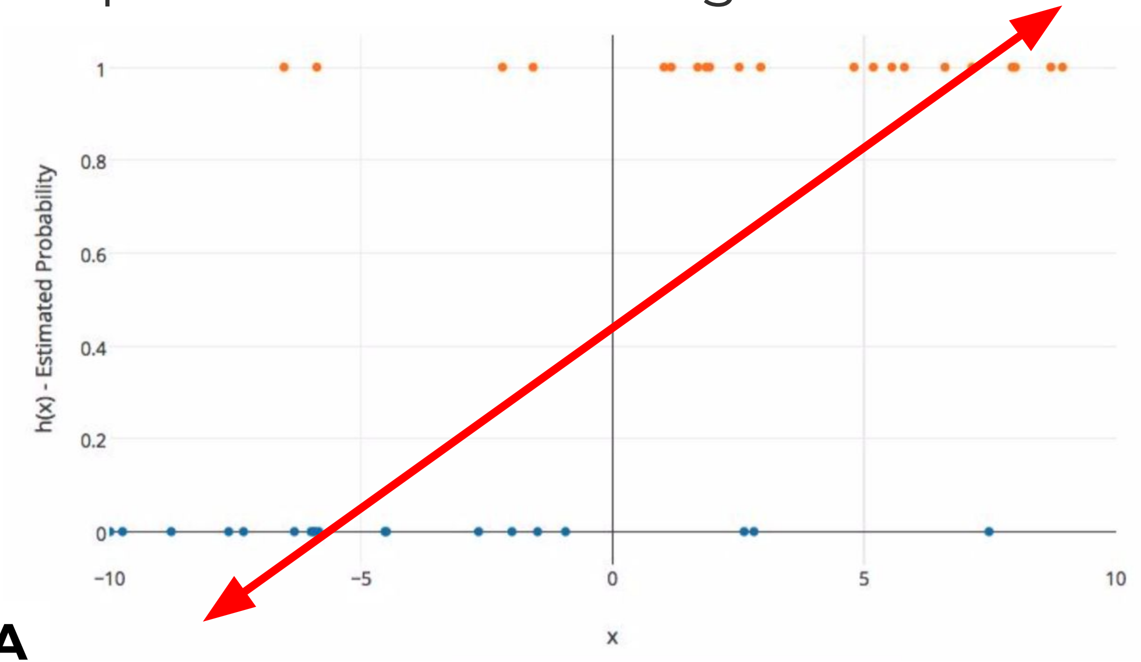
Background

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



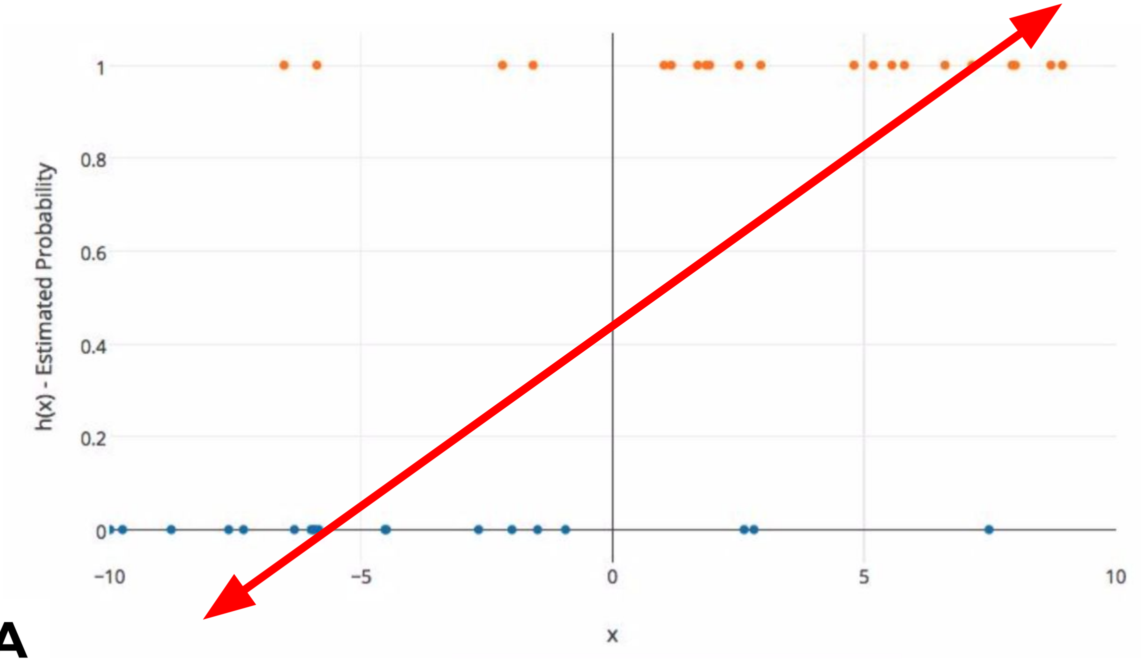
Background

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



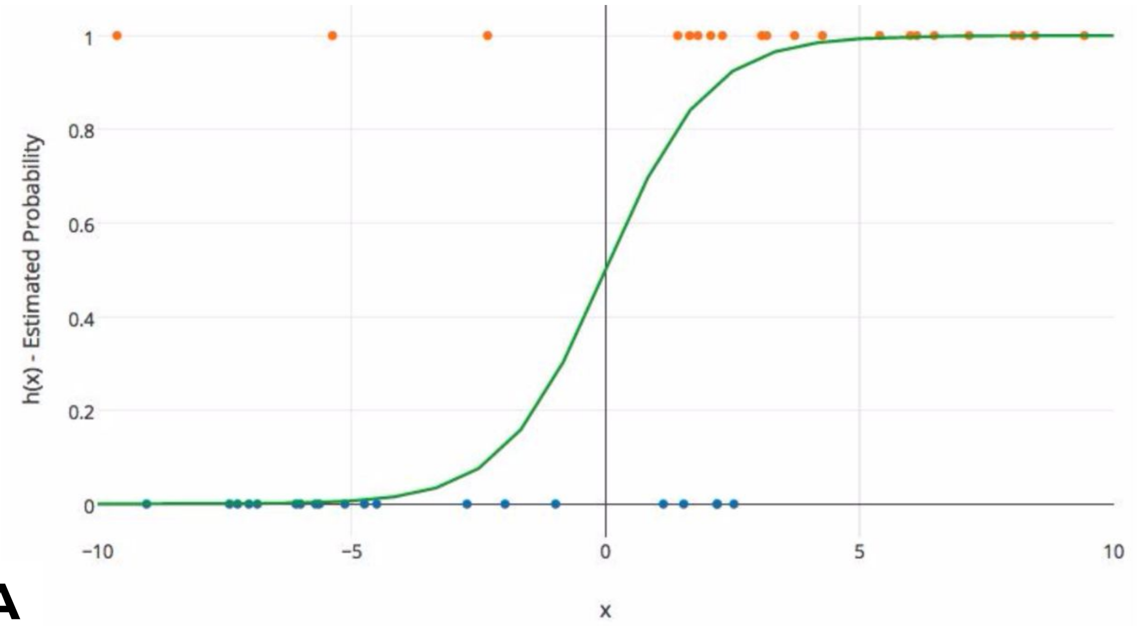
Background

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



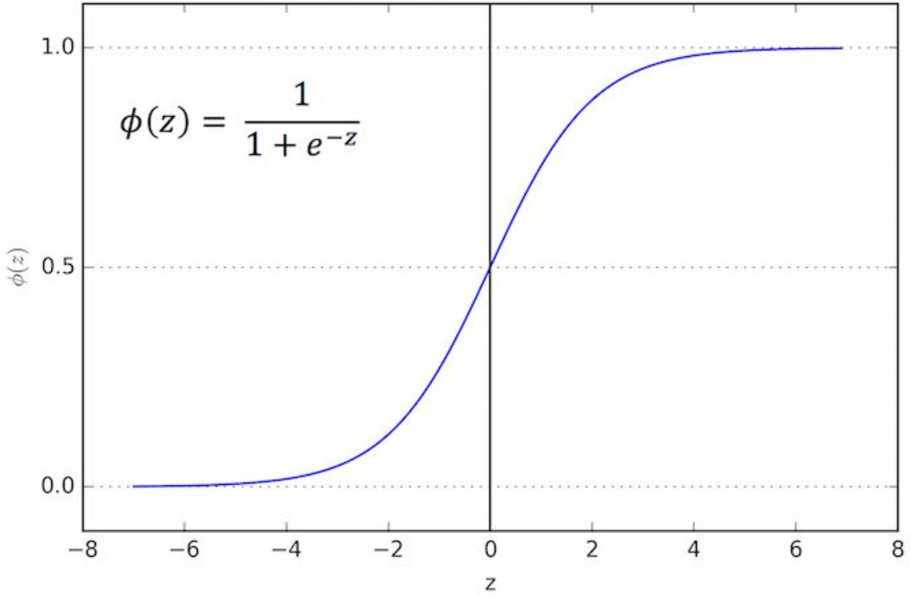
Background

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



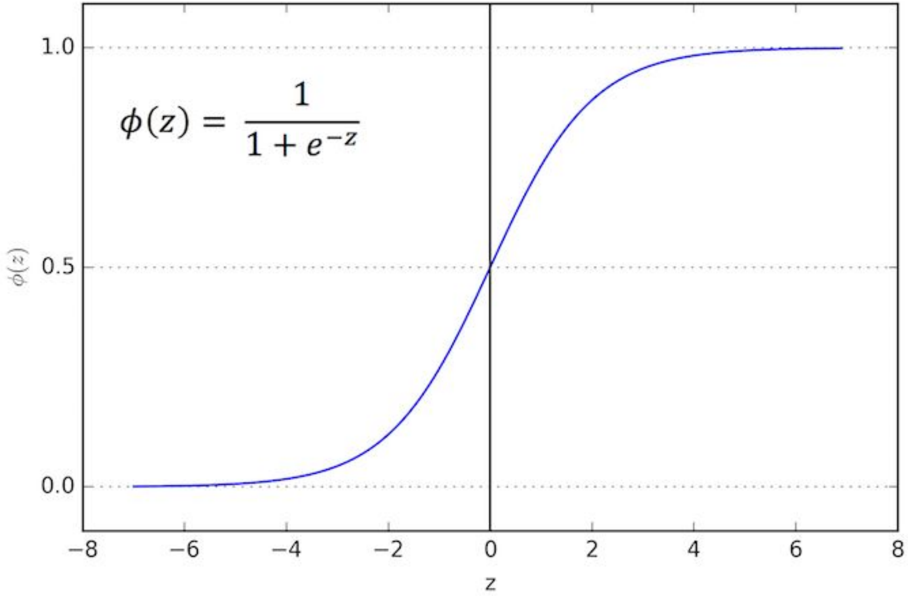
Sigmoid Function

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



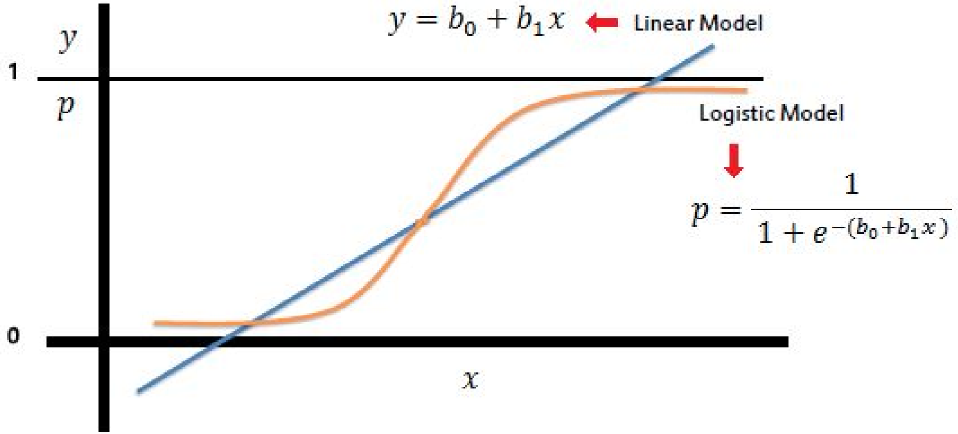
Sigmoid Function

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



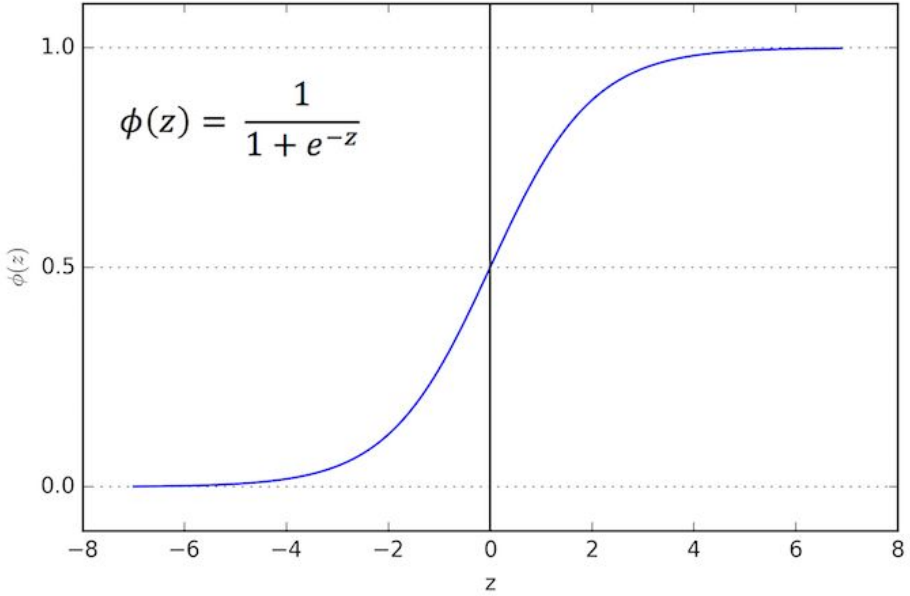
Sigmoid Function

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



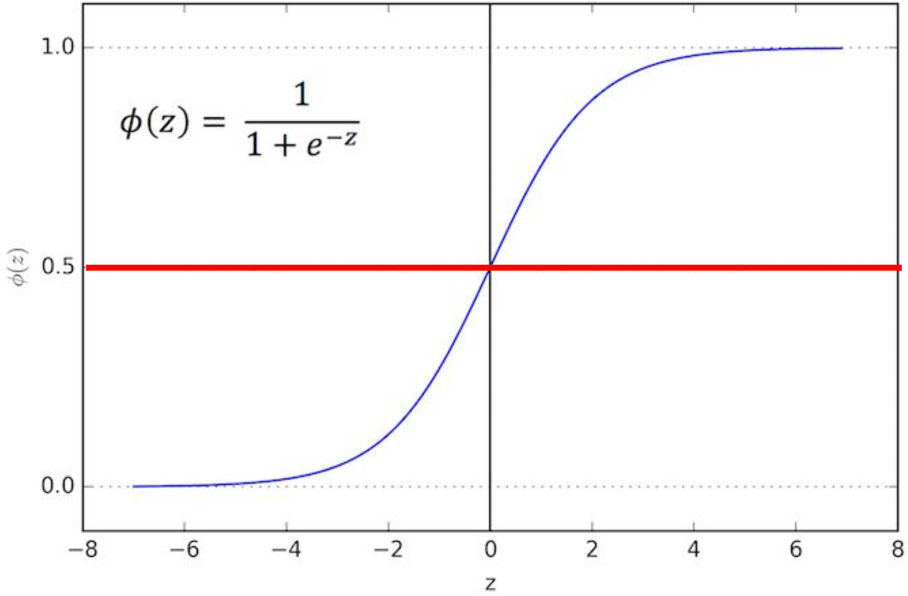
Sigmoid Function

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



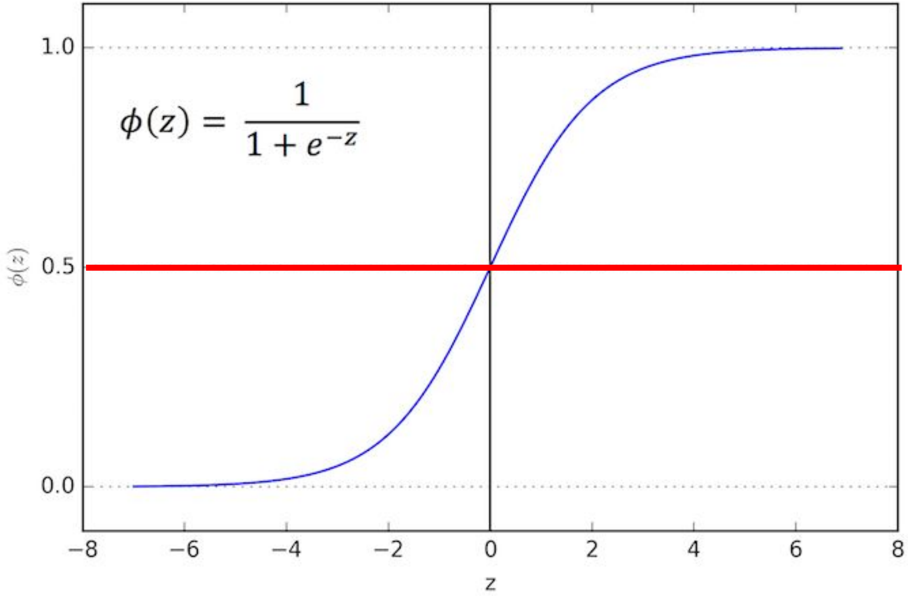
Sigmoid Function

- 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

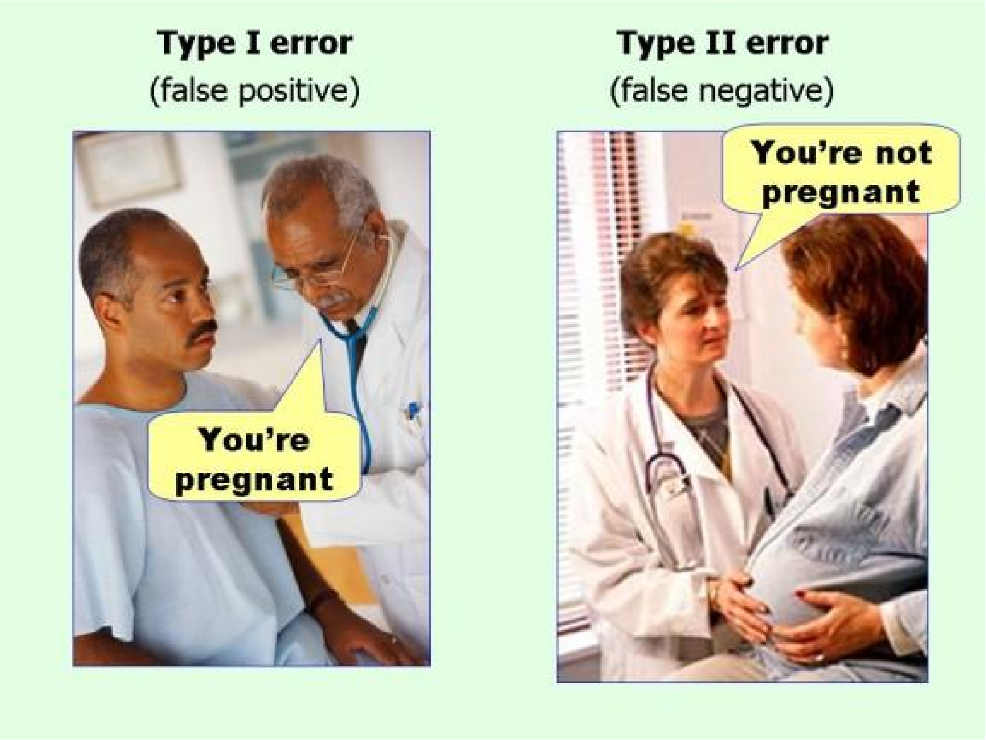
Confusion Matrix

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

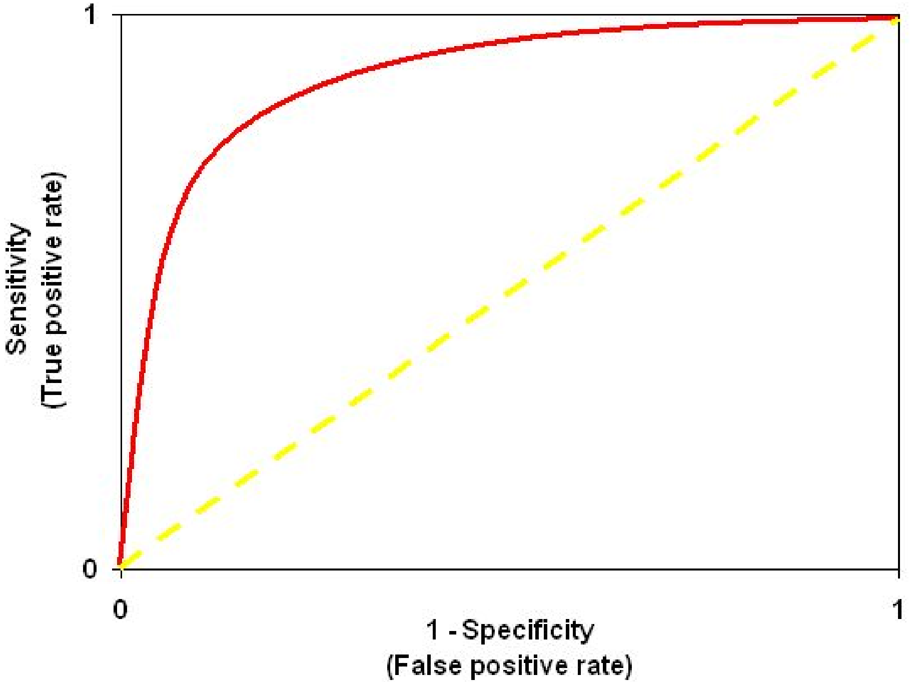
- Still confused on the confusion matrix?
- No problem! Check out the Wikipedia page for it, it has a really good diagram with all the formulas for all the metrics.
- Throughout the course, we'll usually just print out metrics (e.g. accuracy).

Model Evaluation

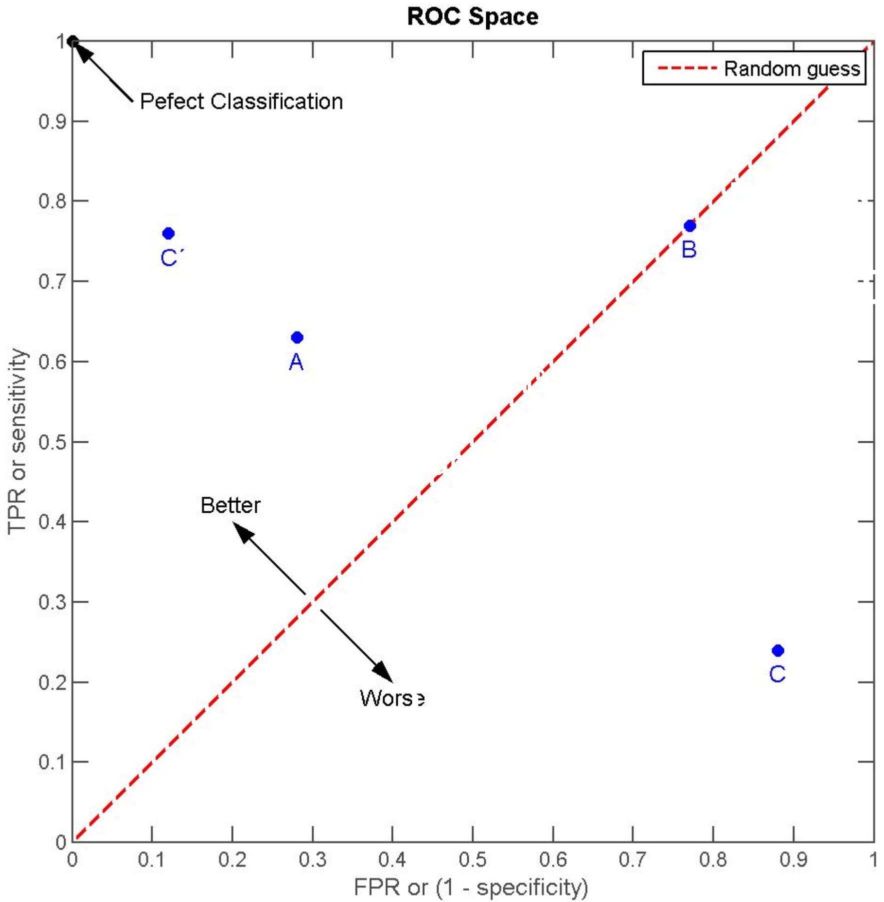
- Binary classification has some of its own special classification metrics.
- These include visualizations of metrics from the confusion matrix.
- The Receiver Operator Curve (ROC) curve was developed during World War II to help analyze radar data.

Model Evaluation

- The ROC curve:



Model Evaluation



PIERIAN DATA

Model Evaluation

- A full discussion of the ROC curve is beyond the scope of this course, but the reading assignment goes into much more detail.
- For now, you just need to know that the area under the curve is a metric for how well a model fit the data.

Model Evaluation

- Let's continue on exploring these concepts with a walk through of the documentation example
- We'll also add some more stuff on evaluation to the example!

Logistic Regression Documentation Example

Python and Spark

- Let's quickly walk through the documentation Logistic Regression example.
- After viewing this, you should begin to pick up on Spark's pattern for machine learning syntax.

- We will also introduce the concept of “Evaluators”.
- Evaluators behave similar to Machine Learning Algorithm objects, but are designed to take in evaluation DataFrames ~ **model.evaluate(test_data)**

Python and Spark

- Evaluators are technically still "experimental" according to the documentation, so use caution when using them for production code.
- But they have been a part of Spark since version 1.4 , so they have some stability.

Python and Spark

- The files for this lecture are
  - Logistic_Regression_Example.ipynb
  - sample_libsvm_data.txt
- Look under the Logistic Regression folder under the Machine Learning folder.
- All links we show can be found in the

ok

Python and Spark

- Let’s get started in a new notebook!

Logistic Regression
Code Along

Python and Spark

- Let’s work through a “classic” classification example!
- The titanic dataset is a common exercise for classification in machine learning, there are lots of examples of it online for other machine learning libraries!

Python and Spark

- We'll use it to attempt to predict what passengers survived the titanic crash based solely on passenger's features (age, cabin, children, etc...)
- We will also explore a few more things!

Python and Spark

* We'll see some better ways to deal with categorical data through a two-step process.
* We will also show how to use pipelines to set stages and build models that can be easily used again!

Python and Spark

- Our data will also have a lot of missing information, so we will need to deal with that as well.
- Let’s get started!

Logistic Regression Consulting Project

Python and Spark

- You did such a great job on the previous consulting project that word is starting to spread about your abilities!
- You’ve been contacted by a top marketing agency to help them out with customer churn!

Python and Spark

- You just landed in New York City!

Python and Spark

- You need to help out a marketing agency predict customer churn!

Python and Spark

- A marketing agency has many customers that use their service to produce ads for the client/customer websites.
- They've noticed that they have quite a bit of churn in clients.

- They currently randomly assign account managers , but want you to create a machine learning model that will help predict which customers will churn (stop buying their service) so that they can correctly assign the customers most at risk to churn an account manager.

Python and Spark

- Luckily they have some historical data, can you help them out?
- Create a classification algorithm that will help classify whether or not a customer churned.

Python and Spark

- Then the company can test this against incoming data for future customers to predict which customers will churn and assign them an account manager.

Python and Spark

- The data is under customer_churn.csv
- Let’s quickly go over the data and what your main task is.

Python and Spark

Name : Name of the latest contact at Company
Age: Customer Age
Total_Purchase: Total Ads Purchased
Account_Manager: Binary 0=No manager, 1= Account manager assigned
Years: Total Years as a customer
Num_sites: Number of websites that use the service.
Onboard_date: Date that the name of the latest contact was onboarded
Location: Client HQ Address
Company: Name of Client Company

Churn: 0 or 1 indicating whether customer has churned.

- Your goal is to create a model that can predict whether a customer will churn (0 or 1) based off the features.
- Remember that the account manager is currently randomly assigned!

Python and Spark

- As always, treat this consulting project as a loosely guided exercise, or skip ahead and treat it as a code along project!
- Best of luck!

Logistic Regression Consulting Project Solutions