Python for Data Science

Introduction to Logistic Regression

Introduction to Logistic Regression

Reading Assignment

Sections 4-4.3 of
Introduction to Statistical Learning
By Gareth James, et al.

Background

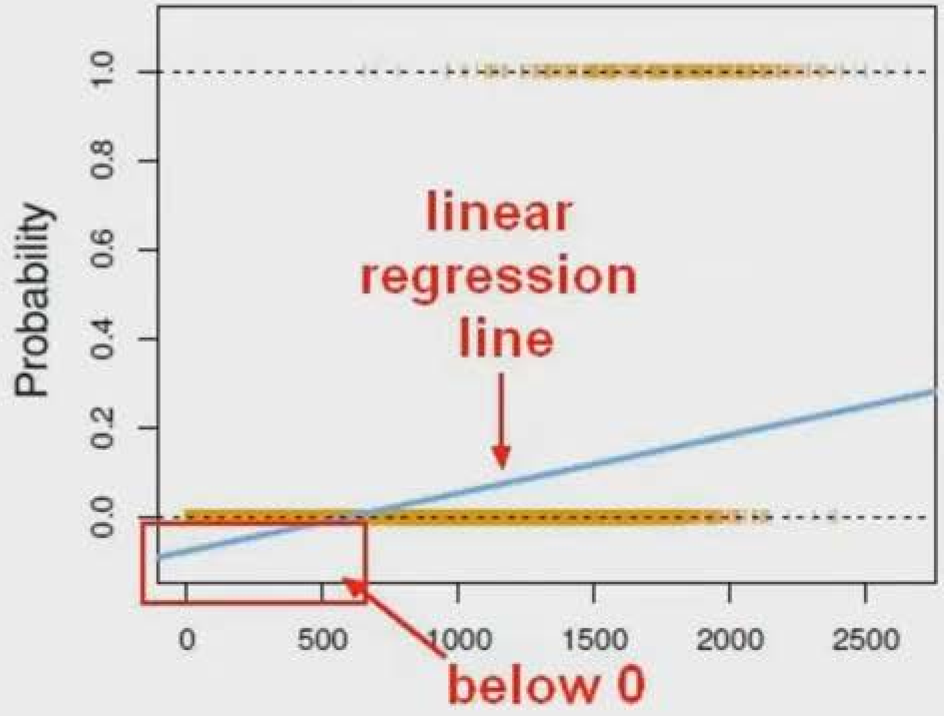
- We want to learn about Logistic Regression as a method for **Classification.**
- Some examples of classification problems:
  - Spam versus “Ham” emails
  - Loan Default (yes/no)
  - Disease Diagnosis
- Above were all examples of Binary Classification

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.
- The convention for binary classification is to have two classes 0 and 1.

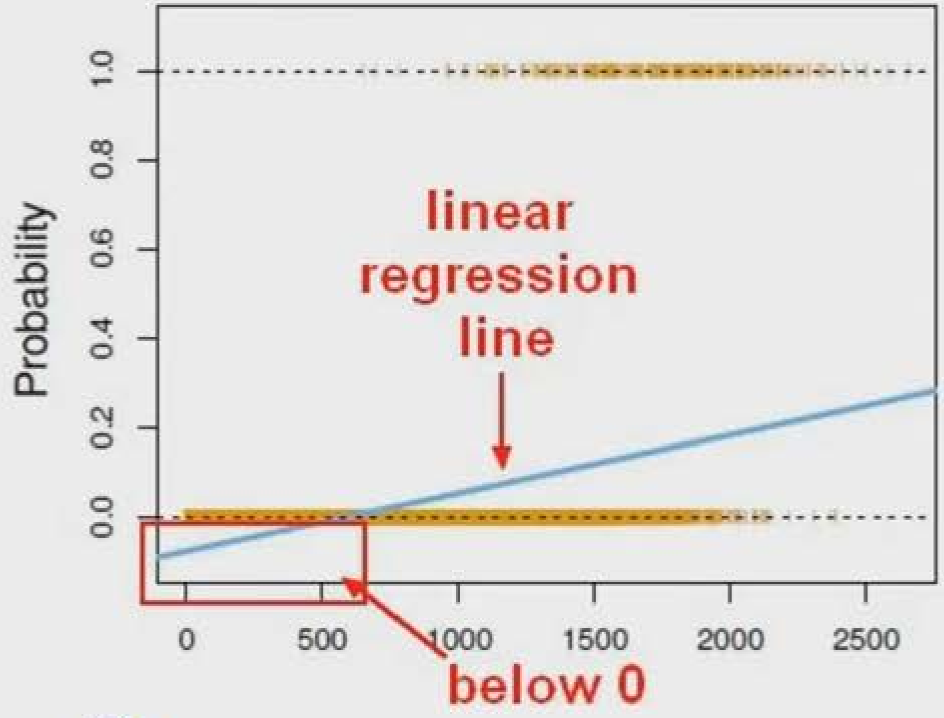
Background

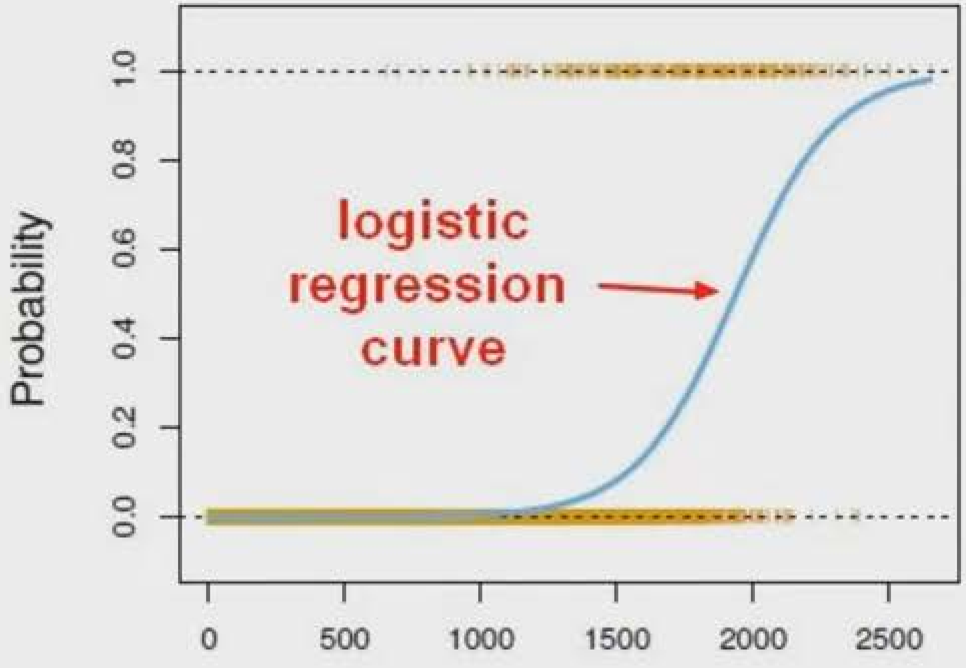
- We can't use a normal linear regression model on binary groups. It won't lead to a good fit:



Background

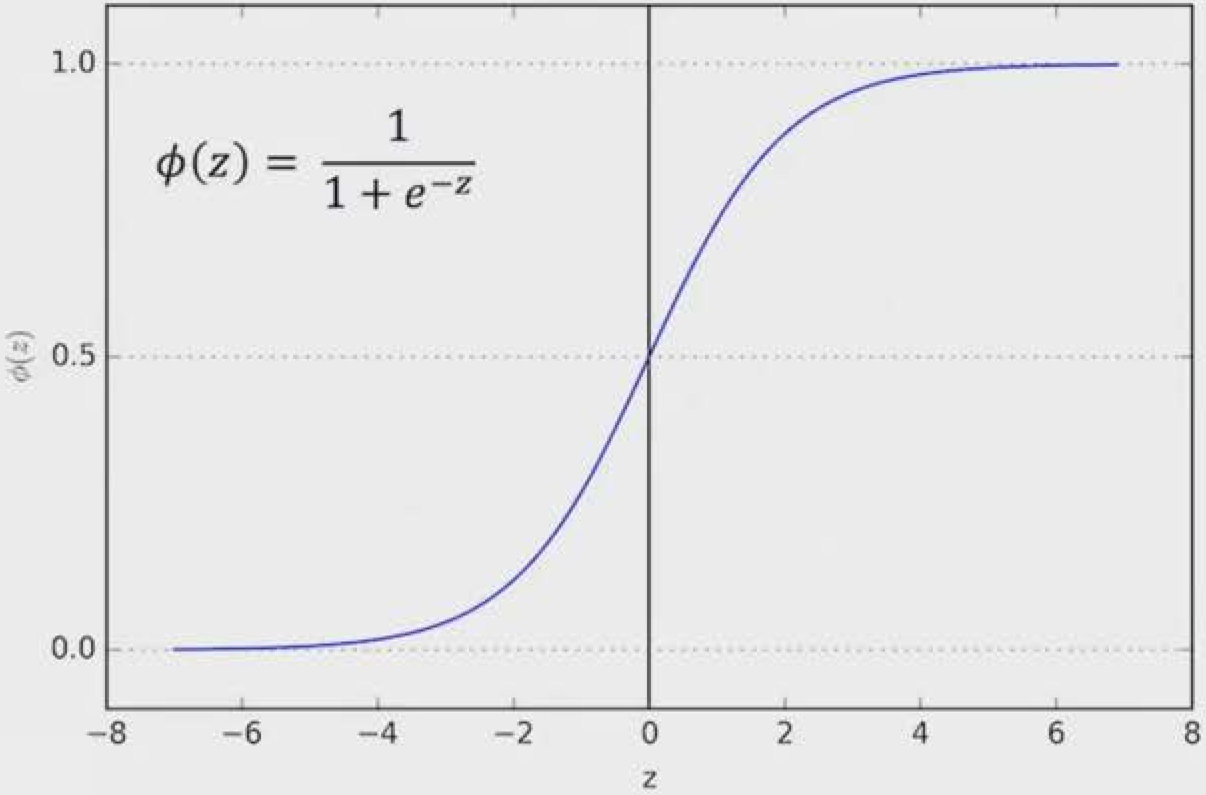
- Instead we can transform our linear regression to a logistic regression curve.





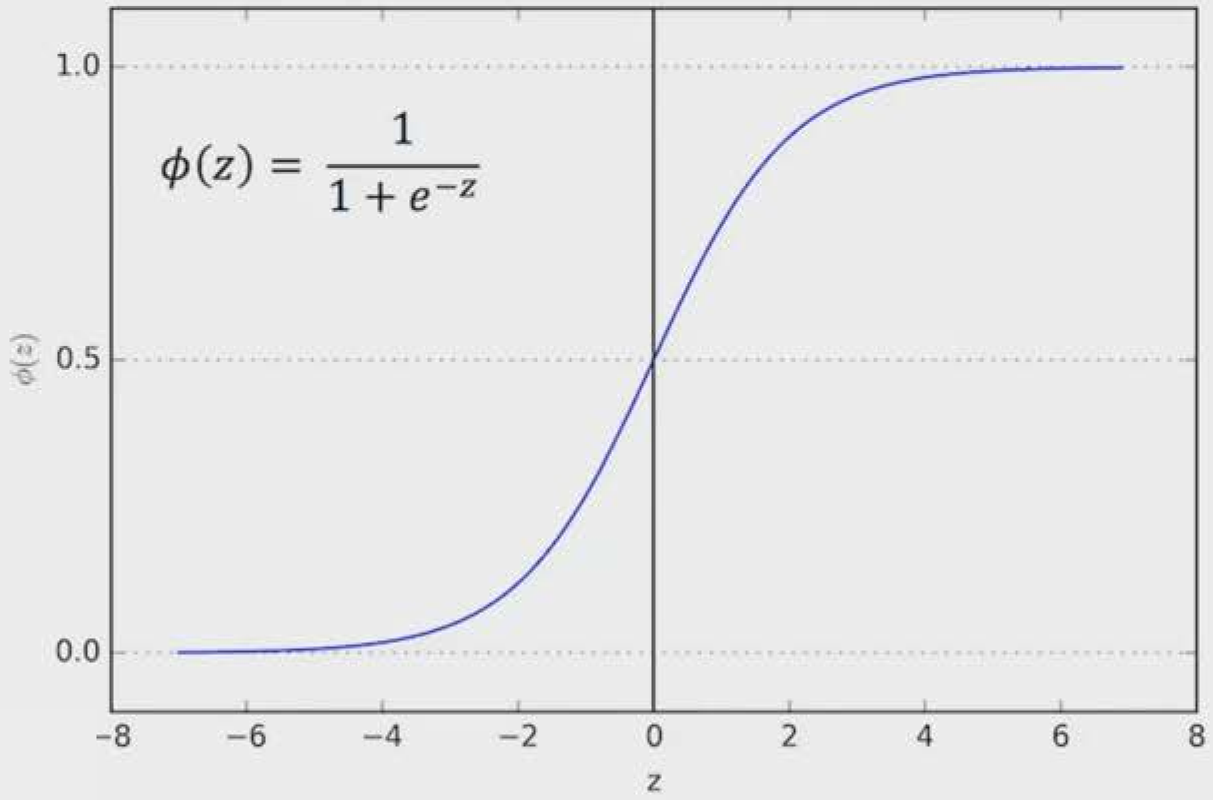
Sigmoid Function

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



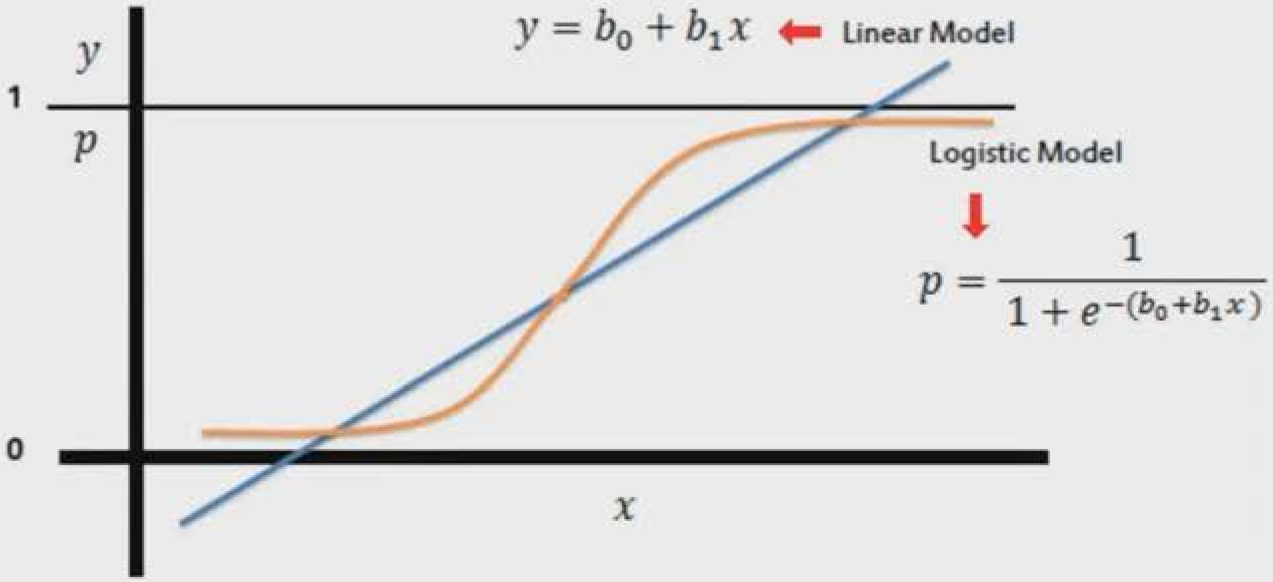
Sigmoid Function

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



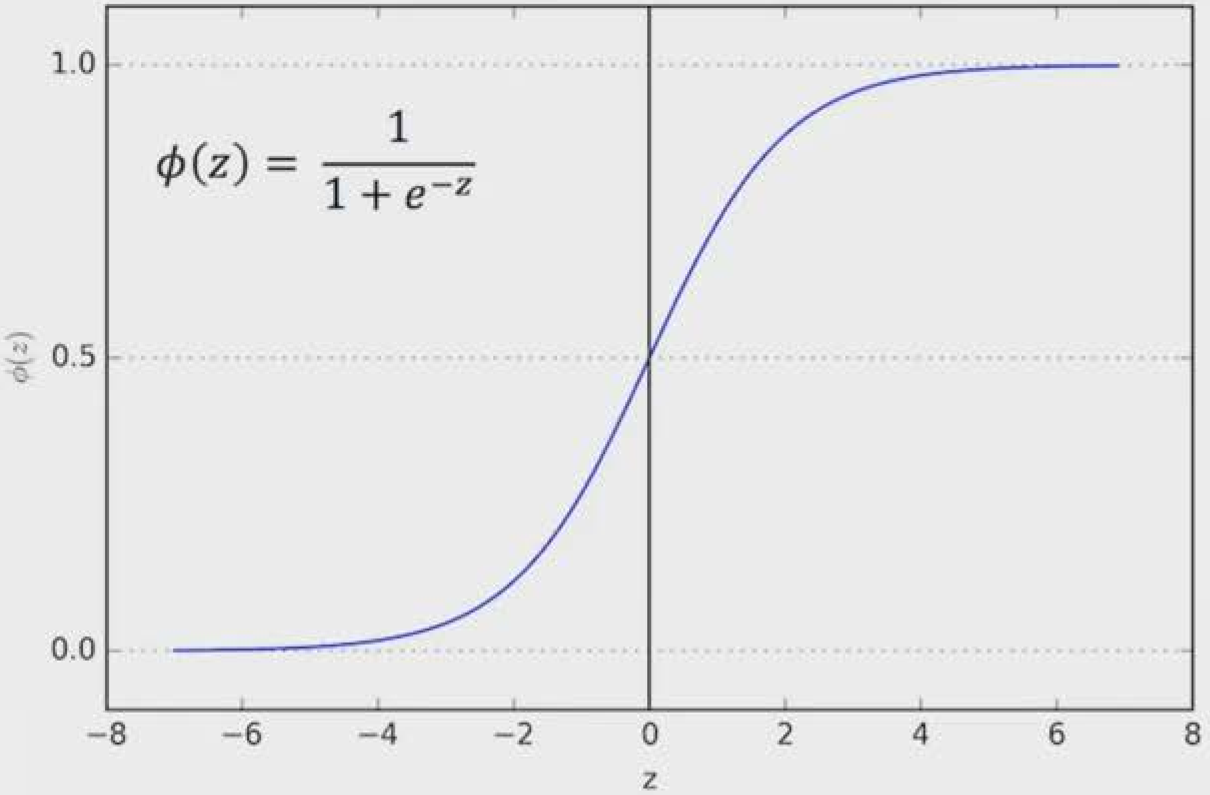
Sigmoid Function

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



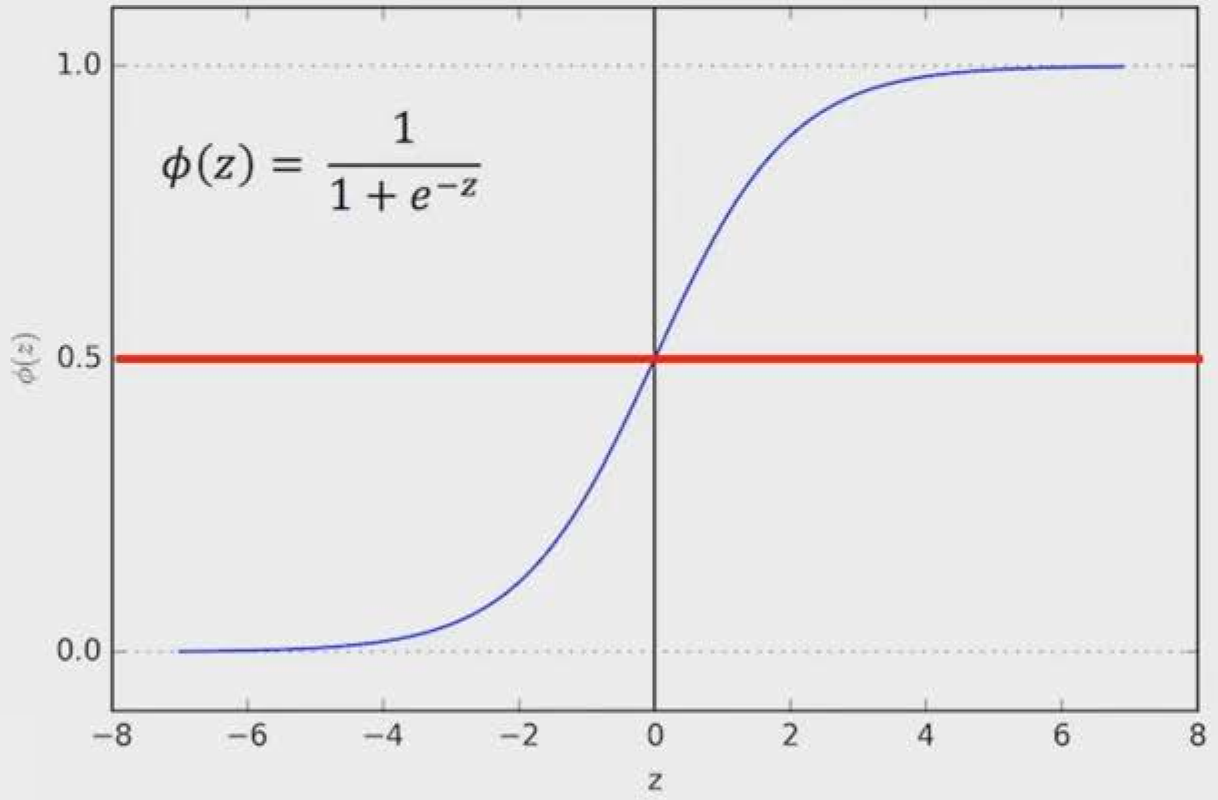
Sigmoid Function

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



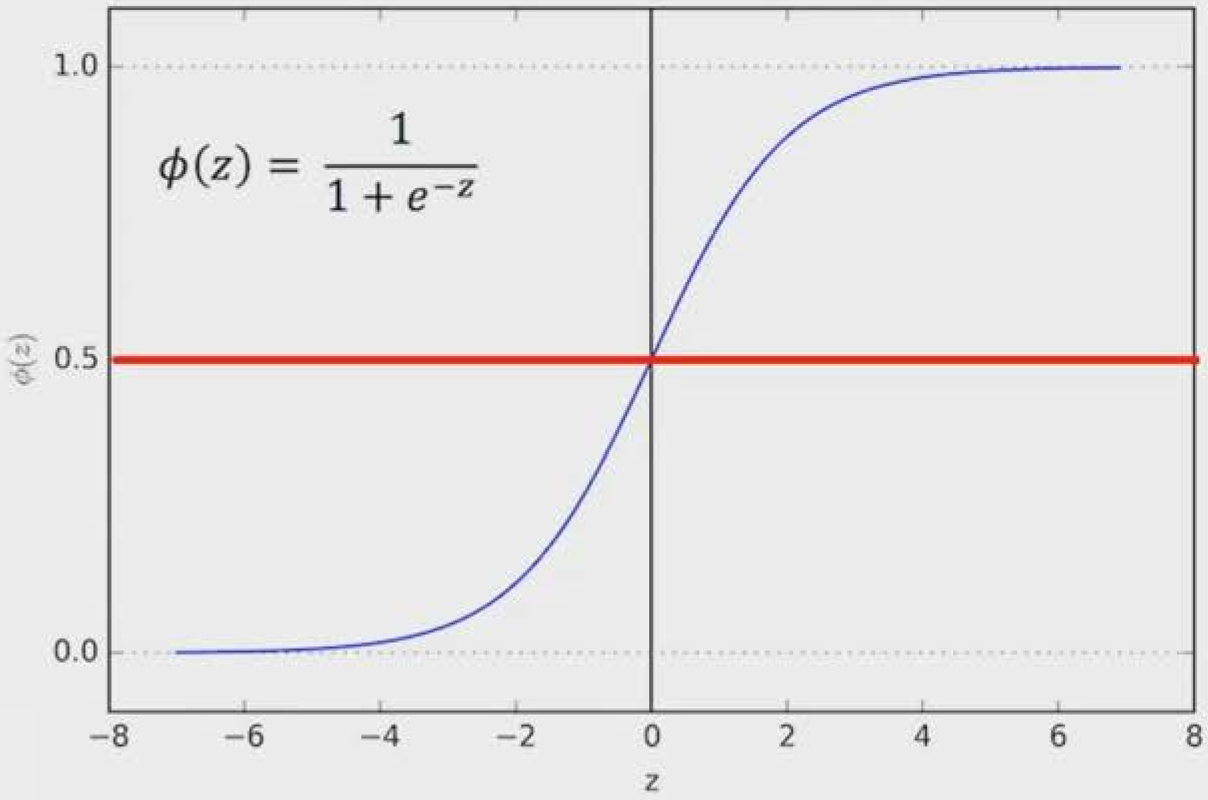
Sigmoid Function

- 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.

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

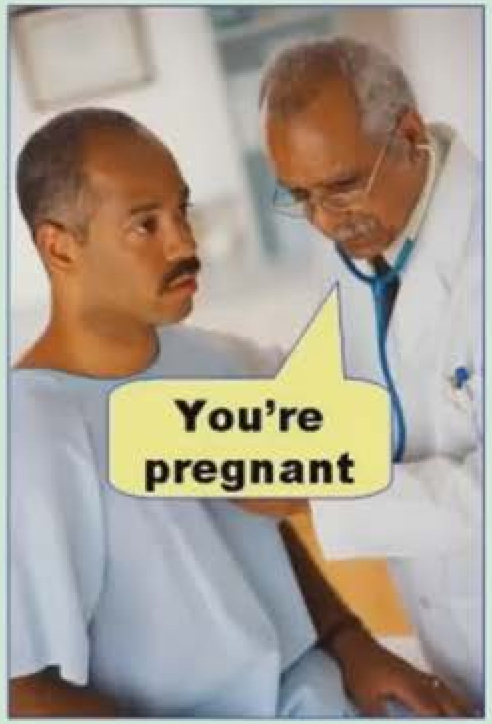
n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Misclassification Rate (Error Rate):

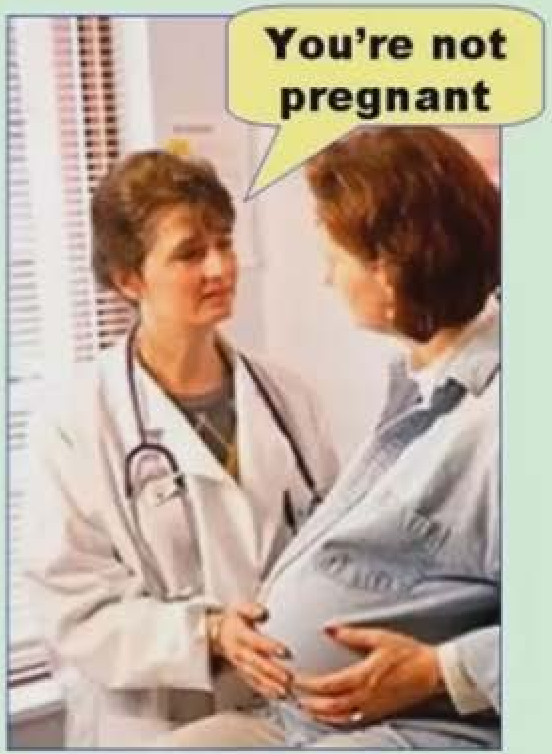
- Overall, how often is it **wrong**?
- (FP + FN) / total = 15/165 = 0.09

Confusion Matrix

Type I error
(false positive)



Type II error
(false negative)



Example with Python

Let's go ahead and begin to explore an example of Logistic Regression using the famous Titanic data set to attempt to predict whether or not a passenger survived based off of their features.

Then you'll have a portfolio project with some Advertising data trying to predict whether or not a customer clicked on an ad!

Thanks!

Any questions ?