CS 471: Introduction to AI

Module 6 Part II: Machine Learning

Linear Regression

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

https://scikit-learn.org/stable/auto_examples/linear_model/plot_ols.html#sphx-glr-auto-examples-linear-model-plot-ols-py

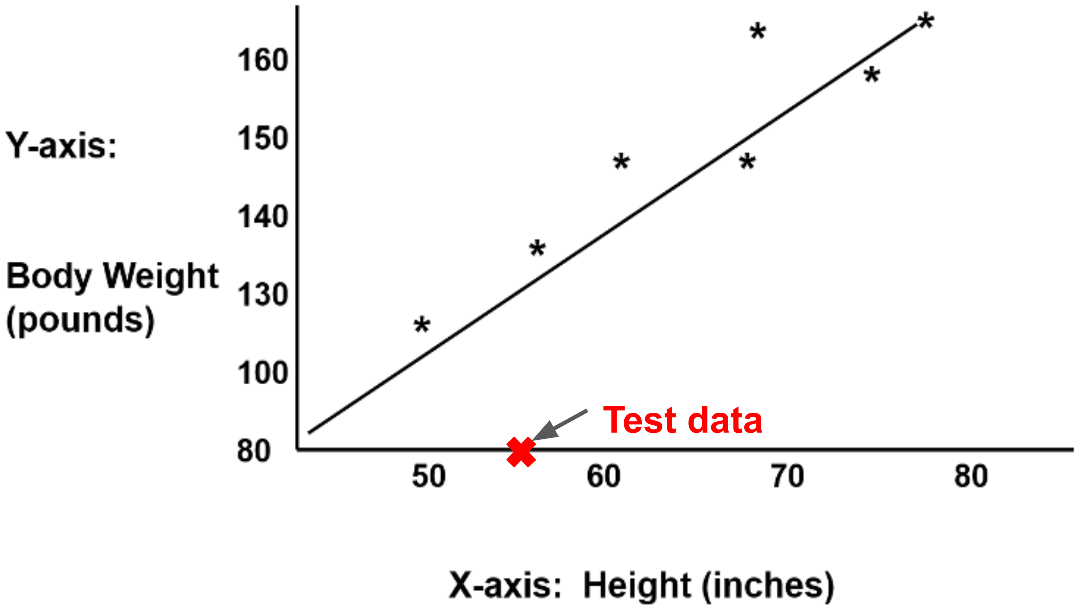
Linear Regression

What is regression?

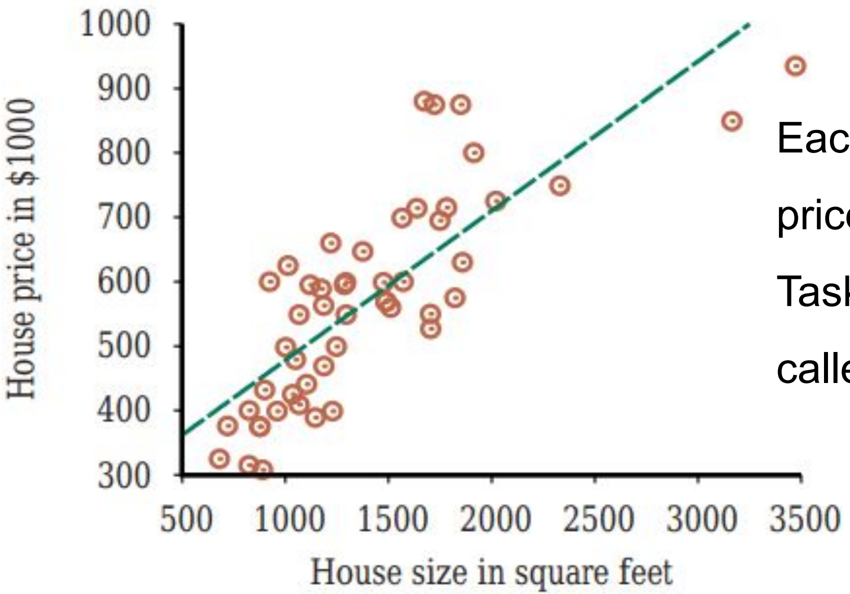
- A univariate linear function (a straight line) with input x and output y has the form

  $y = w_1x + w_0$, where w_0 and w_1 are real-valued coefficients to be learned.

- We use the letter w because we think of the coefficients as weights.



Univariate Linear Regression



Each point represents the size in square feet and the price of a house.

Task of finding the model h_w that best fits these data is called **linear regression**.

Data points of price versus floor space of houses, along with the linear function model that minimizes squared-error loss: y = 0.232x+ 246.

Multivariable Linear Regression

We can easily extend to multivariable linear regression problems, in which each example x_j is an n-element vector.

$$h_{\mathbf{w}}(\mathbf{x}_j) = w_0 + w_1 x_{j,1} + \cdots + w_n x_{j,n} = w_0 + \sum_i w_i x_{j,i} \,.$$

Features **Label**

Size ($feet^2$)	Number of bedrooms	Number of floors	age of home (years)	Price($1000)
2104	5	1	45	460
1416	3	2	40	232
1534	2	2	30	315
...	...	...	...	...

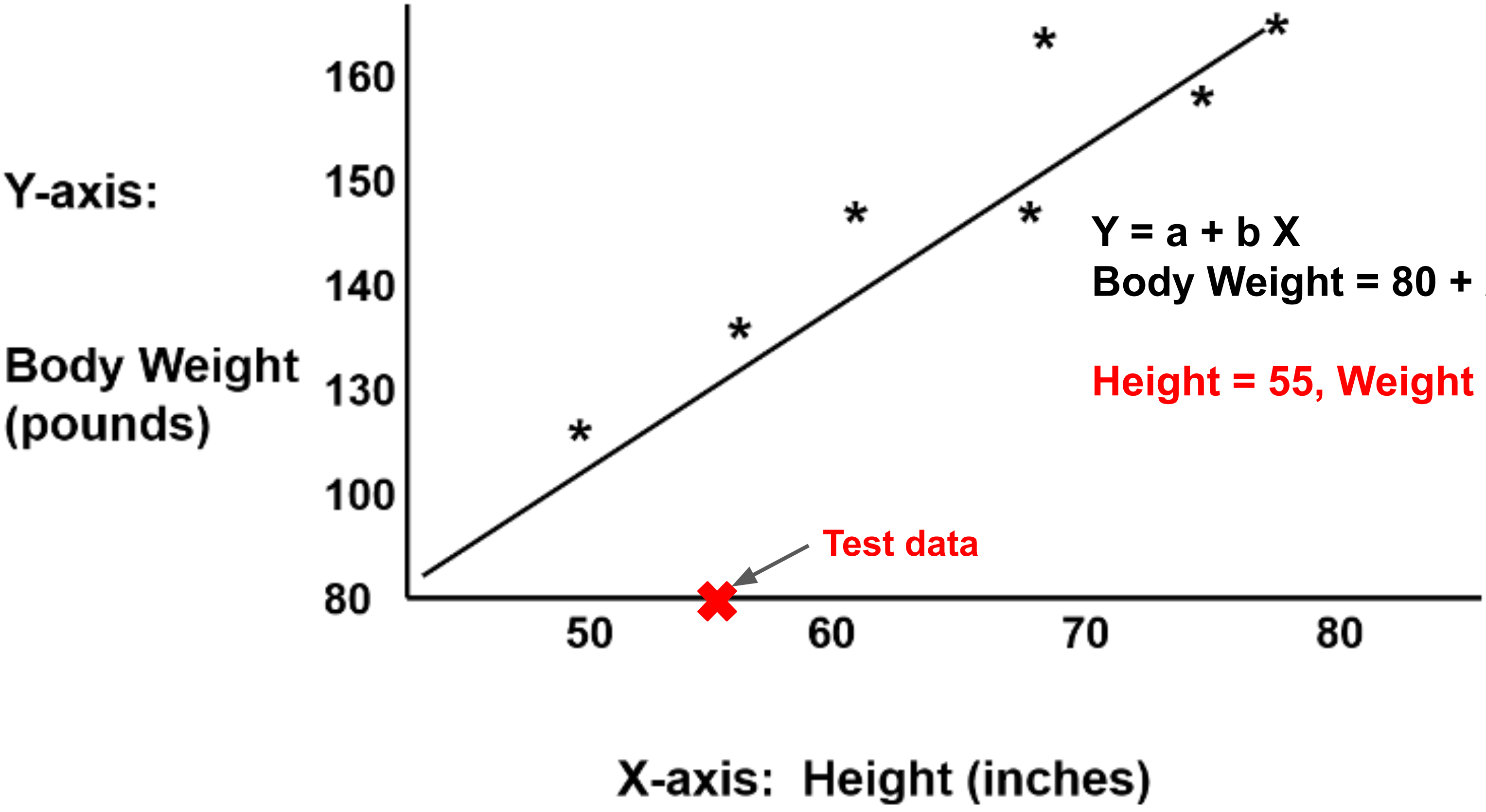
A data

What is the value of n?

Write the equation?

Logistic Regression

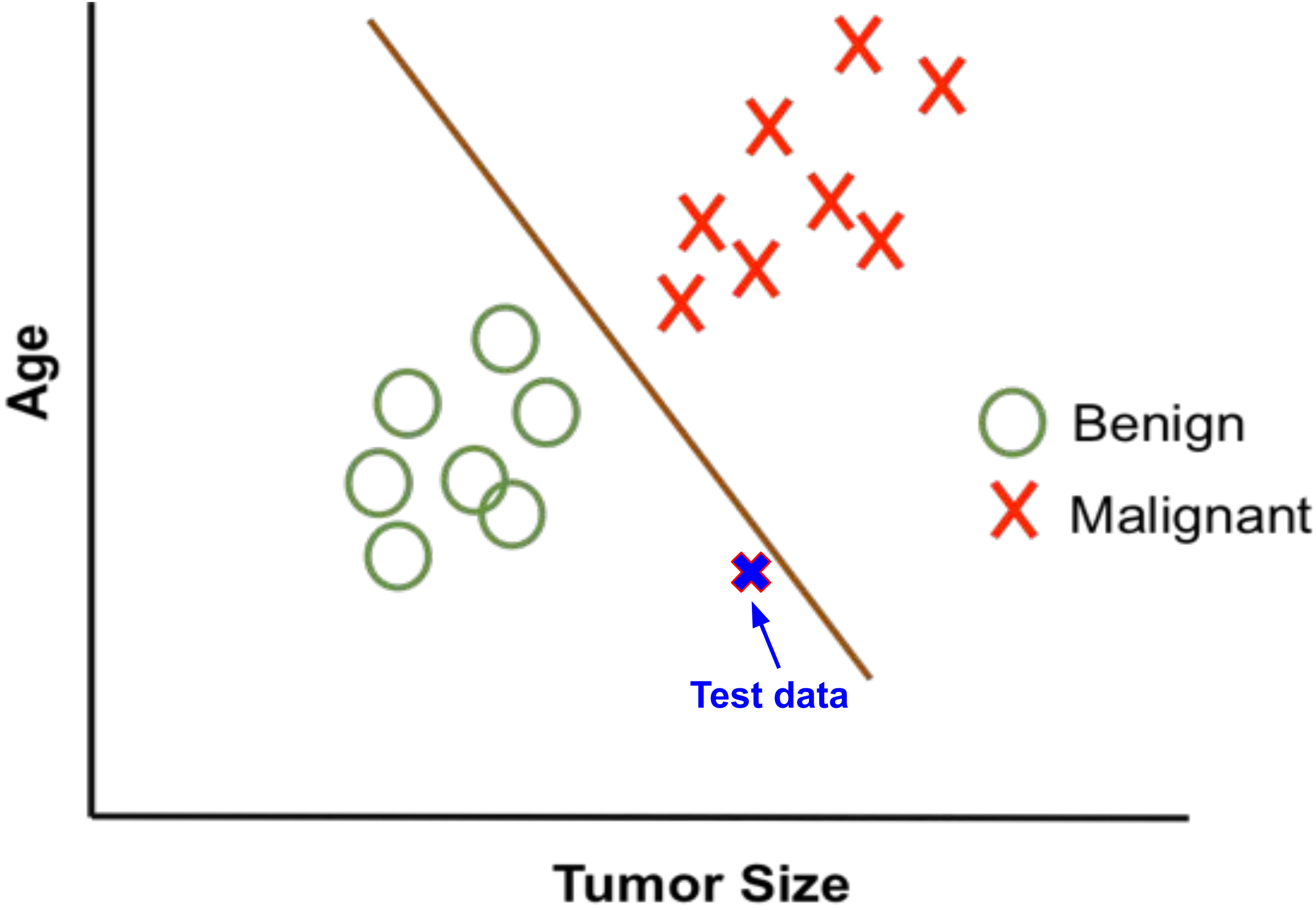
Linear Regression



Y = a + b X
Body Weight = 80 + 2 * Height

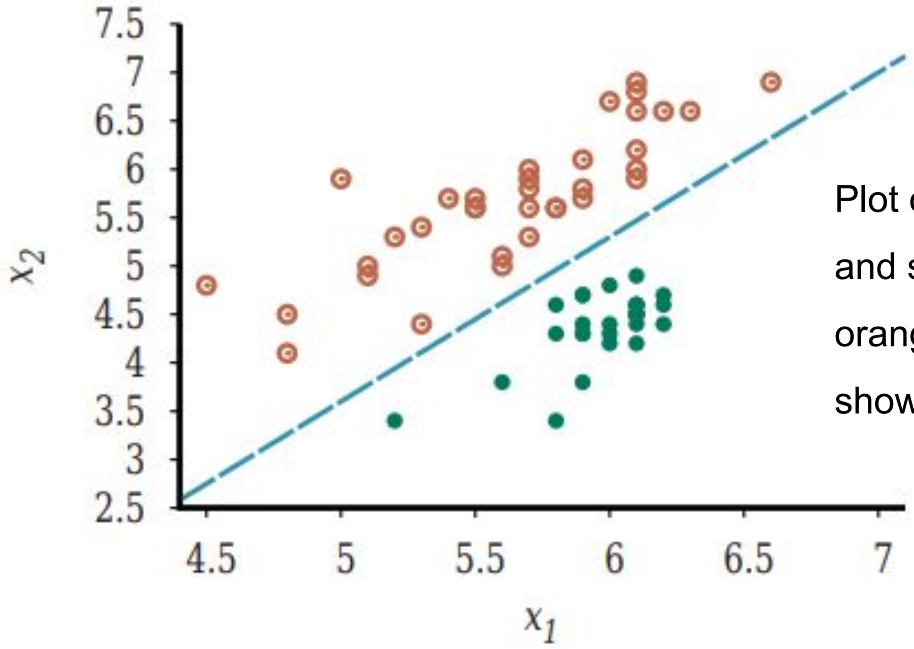
Height = 55, Weight = ?

Logistic Regression



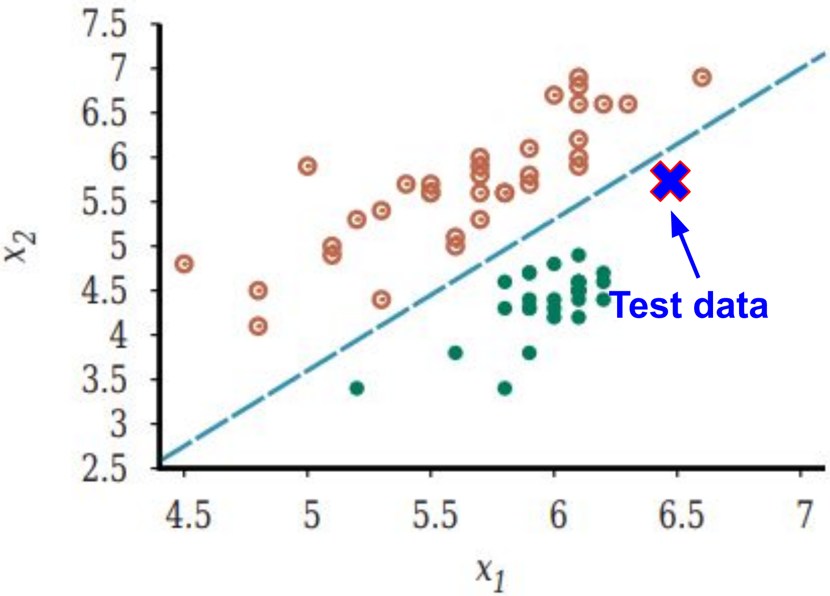
Linear Classifiers with a Hard Threshold

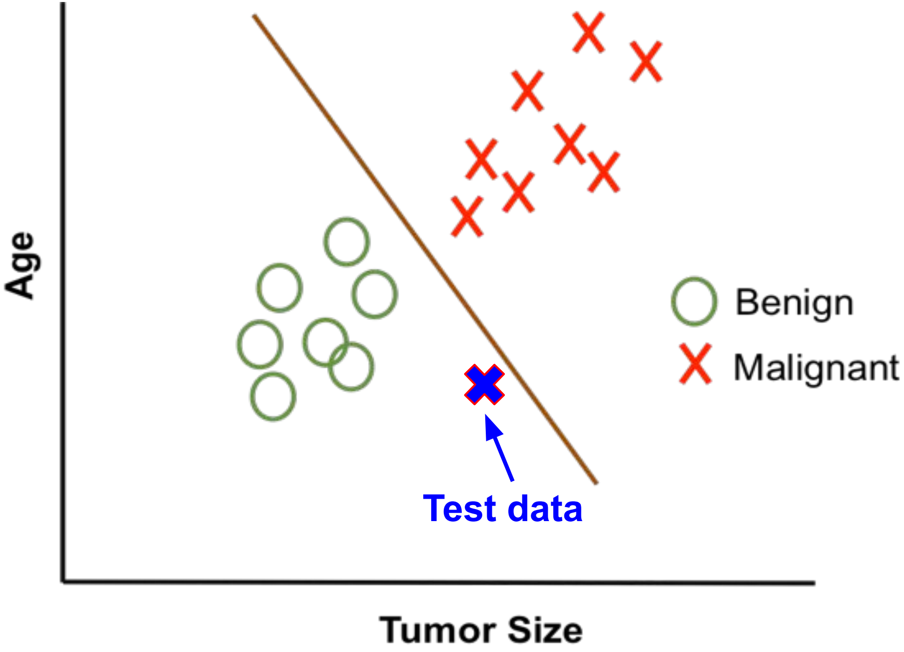
- Data points of two classes: **earthquakes (which are of interest to seismologists)** and **nuclear explosions (which are of interest to arms control experts)**.
- x_1 and x_2 refers to body and surface wave magnitudes computed from the seismic signal.



Plot of two seismic data parameters, body wave magnitude x_1 and surface wave magnitude x_2, for earthquakes (open orange circles) and nuclear explosions (green circles). Also shown is a decision boundary between the classes.

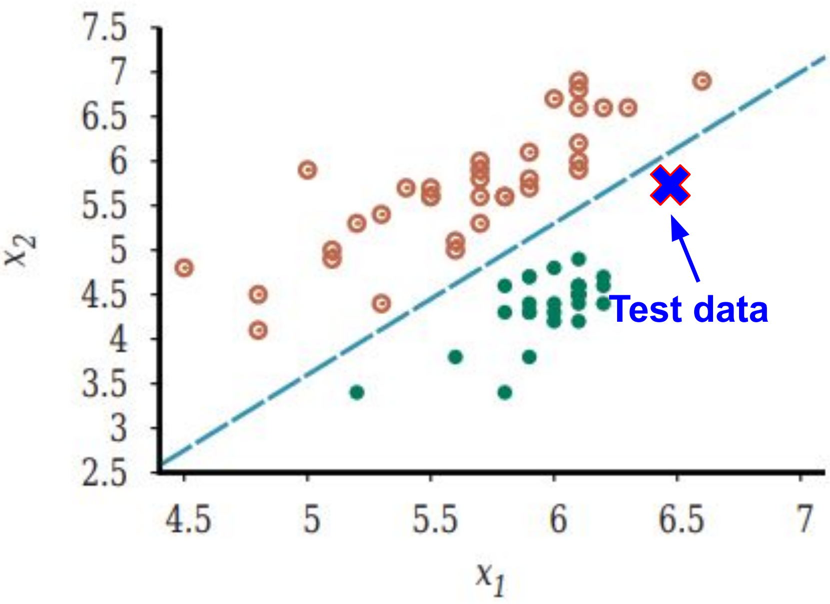
Linear Classifiers with a Hard Threshold

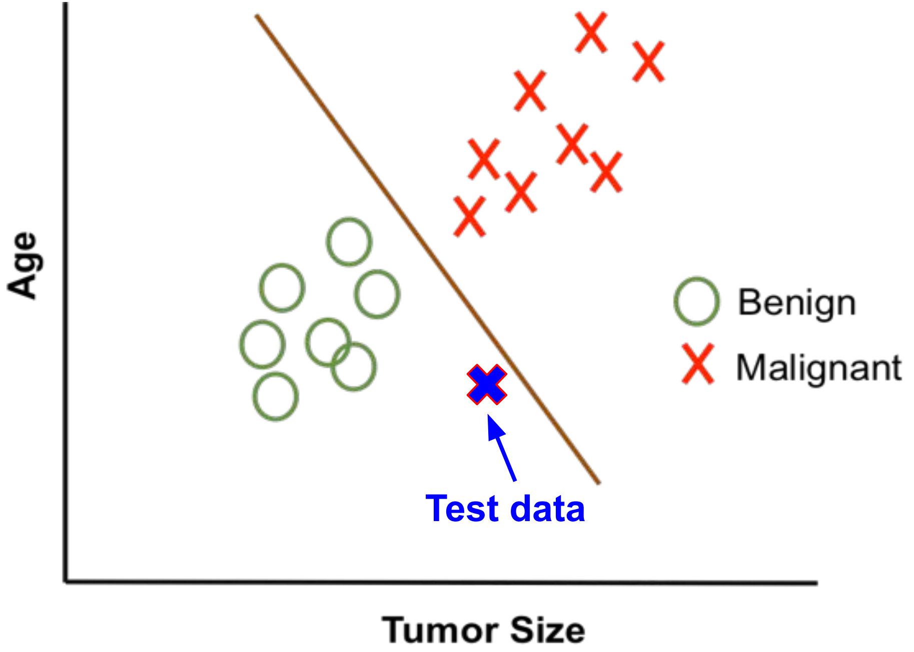




Given a new data point, how do you find the class/output/label?

Linear Classifiers with a Hard Threshold



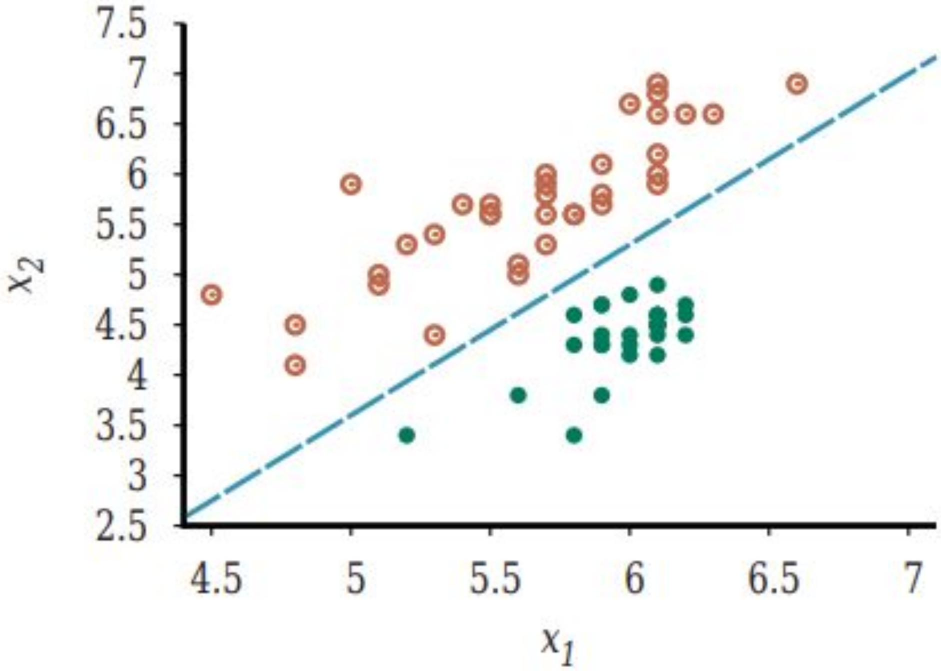


Given a new data point, how do you find the class/output/label?

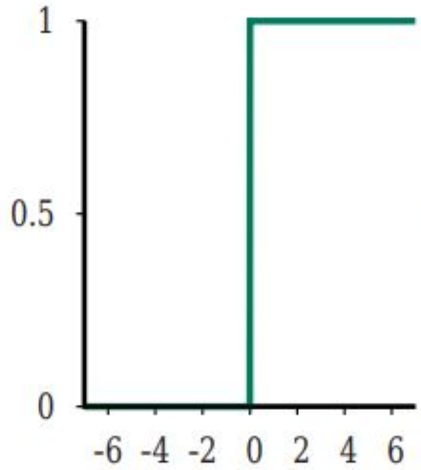
Passing the output of a linear function through the threshold function creates a linear classifier

Linear Classifiers with a Hard Threshold

- Given these training data, the task of classification is to learn a hypothesis h that will take new (x_1, x_2) points and return either 0 for earthquakes or 1 for explosions.
- Goal is to find the decision boundary that separates the two classes.

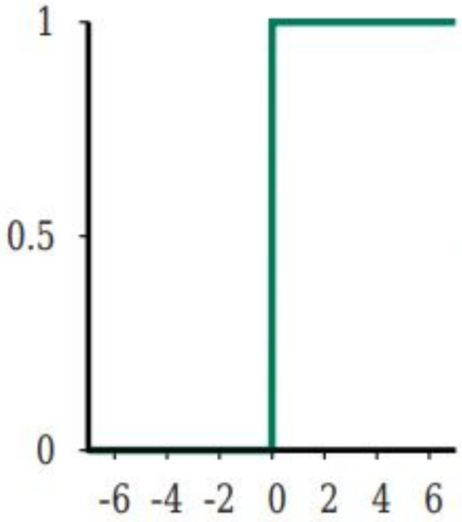


$$h_{\mathbf{w}}(\mathbf{x}) = 1 \text{ if } \mathbf{w} \cdot \mathbf{x} \geq 0 \text{ and } 0 \text{ otherwise.}$$



Problems with a Hard Threshold

- Here we cannot do either of those things because the gradient is zero almost everywhere in weight space, and at z = 0 the gradient is undefined.



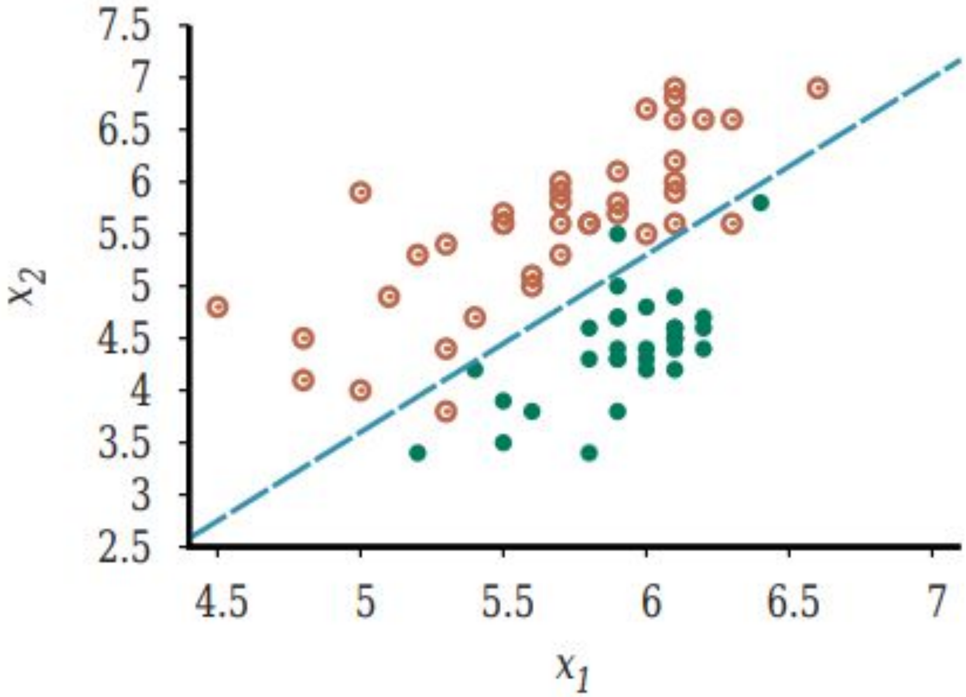
The hard threshold function with 0/1 output.

The function is non differentiable at z=0.

Problems with a Hard Threshold

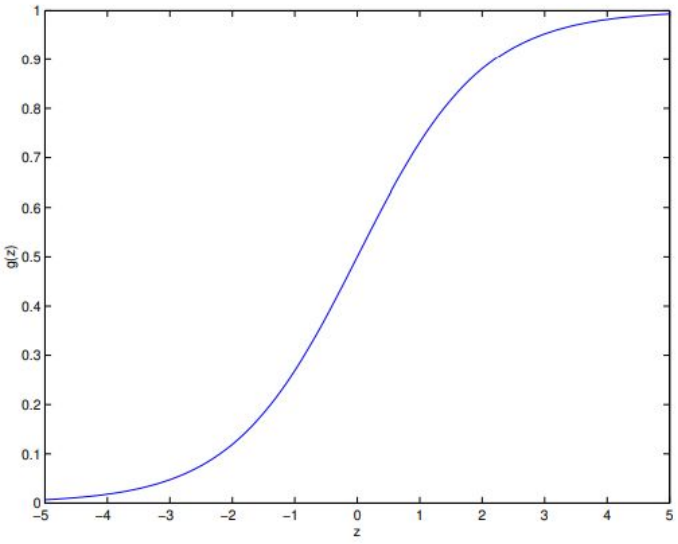
- The linear classifier gives a confident prediction of 1 or 0, even for examples that are very close to the boundary;

  it would be better if it could classify some examples as a clear 0 or 1, and others as unclear borderline cases.



Logistic Regression

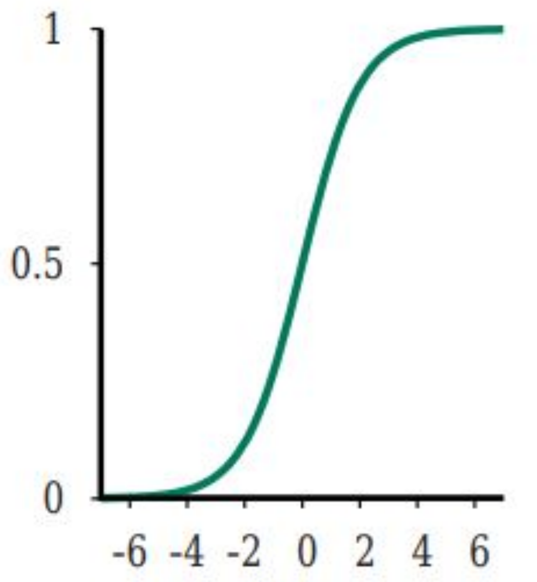
- These issues can be resolved by softening the threshold function, approximating the hard threshold with a continuous, differentiable function.
- Logistic (also called sigmoid) function: $$g(z) = \frac{1}{1 + e^{-z}}$$



Logistic Regression

The output, being a number between 0 and 1, can be interpreted as a probability of belonging to the class labeled 1.

Hypothesis forms a soft boundary in the input space and gives a probability of 0.5 for any input at the center of the boundary region, and approaches 0 or 1 as we move away from the boundary.



Logistic Regression

Linear classifier with a hard threshold = Passing the output of a linear function through the threshold function

Logistic Regression = Passing the output of a linear function through the sigmoid or logistic function

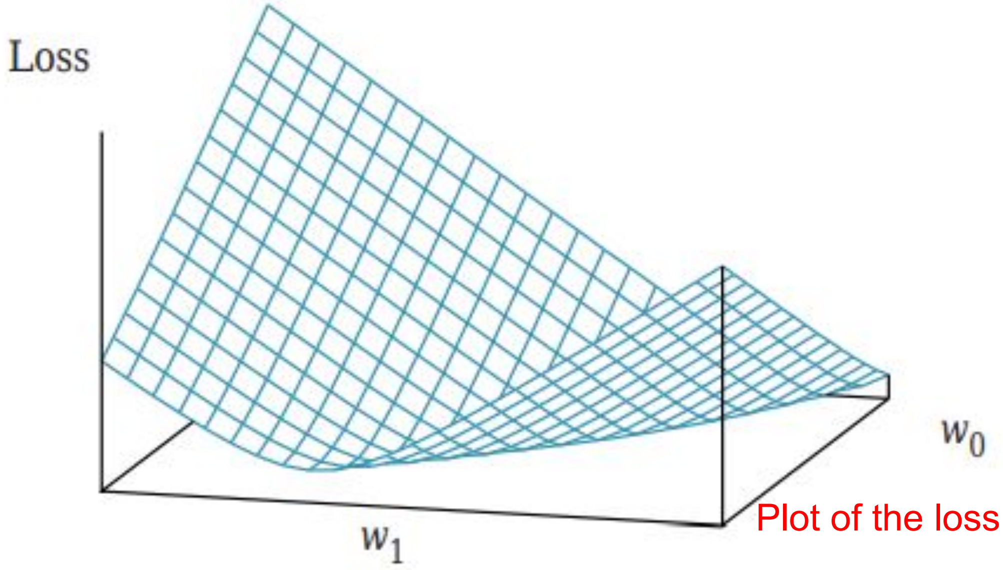
THANK YOU!

Univariate Linear Regression

- To fit a line, we have to find the values of the weights <w_0,w_1> that minimize the loss.
- Common to use the squared-error loss function, L_2, summed over all the training examples:

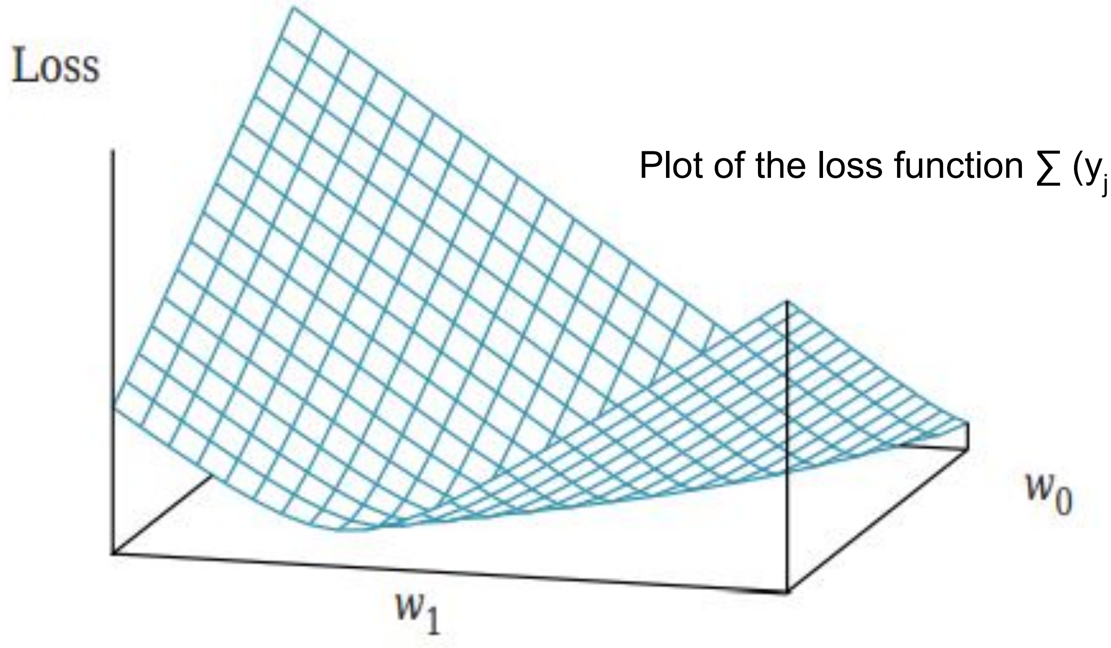
$$Loss(h_{\mathbf{w}}) = \sum_{j=1}^{N} L_2(y_j, h_{\mathbf{w}}(x_j)) = \sum_{j=1}^{N} (y_j - h_{\mathbf{w}}(x_j))^2 = \sum_{j=1}^{N} (y_j - \underline{(w_1 x_j + w_0)})^2 .$$

Actual (ground truth) Predicted



Plot of the loss function $\sum (y_j - w_1 x_j - w_0)^2$ for various values of w_0,w_1.

Univariate Linear Regression



Plot of the loss function $\sum (y_j - w_1 x_j + w_0)^2$ for various values of w_0, w_1.

How to find w_1 and w_0 to minimize loss function (actual - predicted)^2?

Univariate Linear Regression

Option 1:

Take the partial derivative of the loss function with respect to each weight and equate them to zero.

$$\frac{\partial}{\partial w_0} \sum_{j=1}^{N} (y_j - (w_1 x_j + w_0))^2 = 0 \text{ and } \frac{\partial}{\partial w_1} \sum_{j=1}^{N} (y_j - (w_1 x_j + w_0))^2 = 0.$$

These equations have a unique solution:

$$w_1 = \frac{N(\sum x_j y_j) - (\sum x_j)(\sum y_j)}{N(\sum x_j^2) - (\sum x_j)^2}; \quad w_0 = (\sum y_j - w_1(\sum x_j))/N.$$

Gradient Descent

- In many cases, we may not solve the equation partial derivatives = 0.

Option 2

- Search through a continuous weight space by incrementally modifying the parameters: gradient descent
- Choose any starting point in weight space: compute an estimate of the gradient and move a small amount in the steepest downhill direction, repeating until we converge on a point in weight space with (local) minimum loss.

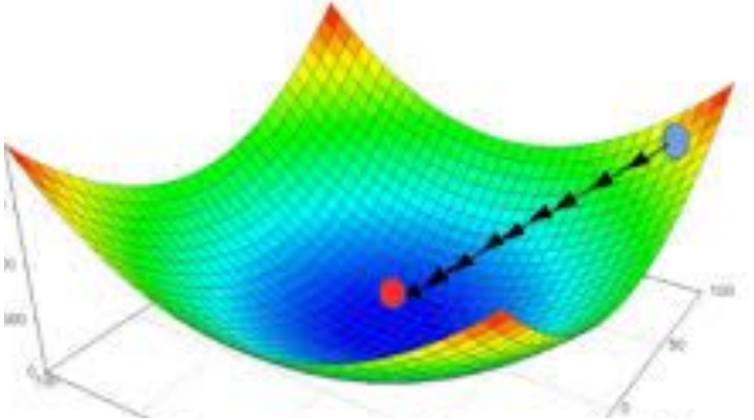
$\mathbf{w} \leftarrow$ any point in the parameter space
while not converged **do**
  for each w_i **in** $\mathbf{w}$ **do**

$$\boxed{w_i \leftarrow w_i - \alpha \frac{\partial}{\partial w_i} Loss(\mathbf{w})}$$

Parameter α, is called the step size, also called the learning rate.

Linear Classifiers with a Hard Threshold

- Solutions we have seen for linear regression
  - Setting the gradient to zero to compute the weights
  - Gradient descent in the weight space



- Can we apply the same techniques for a classification problem?