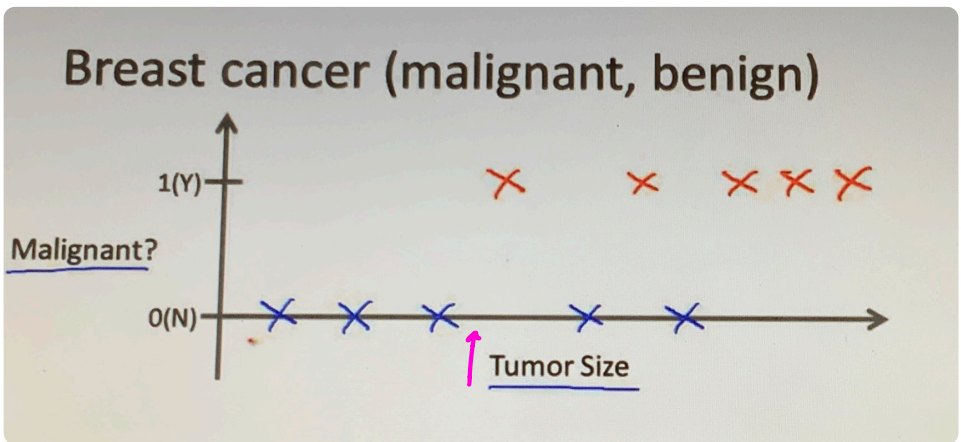
Stanford ML

Week 1:

Supervised Learning

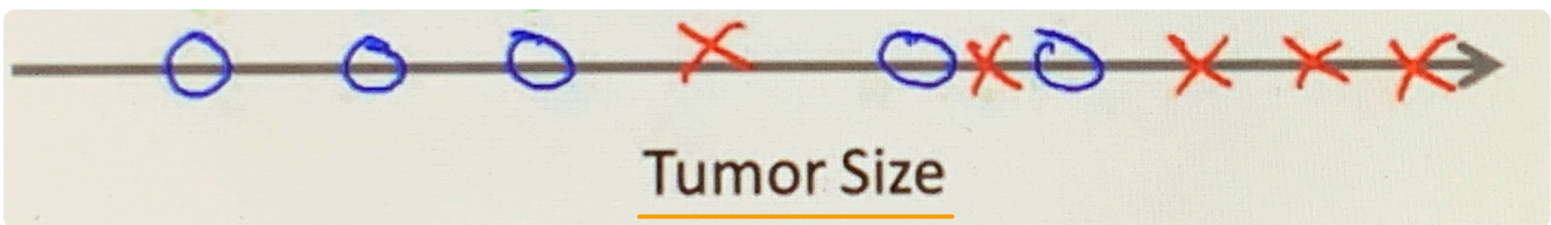
- 'Correct' answers are known
  i.e every house (in feet^2) had a corresponding price which was known
- REGRESSION:
  Prediction ——→ CONTINUOUS
  i.e. value output (price)

—→ : indicates new entry, what is the probability of this tumor being malignant or benign?



- Discrete value output (0 or 1)
- May wish to change DATA VISUALISATION

Classification Problem

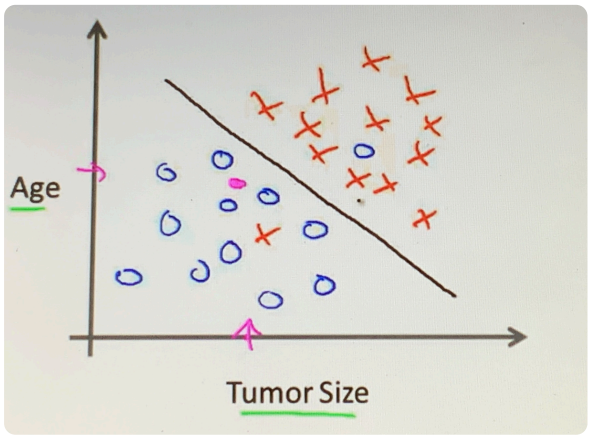


↳ Tumor size is used as the

sole FEATURE / ATTRIBUTE to determine whether a tumour is malignant or benign.

NEW ATTRIBUTE : Age

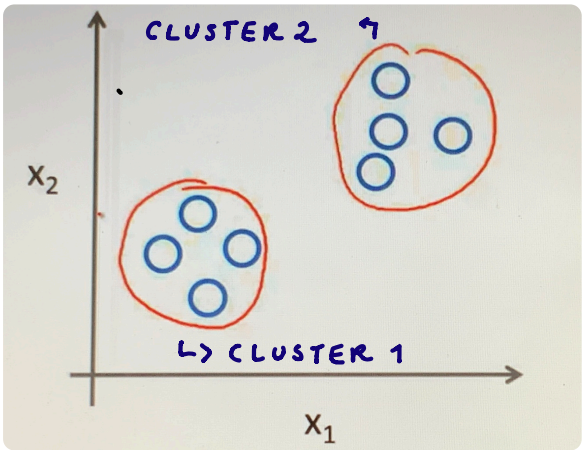
- Now including 2nd attribute
- Algorithm may choose to separate the data as shown.
- Some algorithms deal with an ∞ no. of 'features' ⟶ SVMs.



Unsupervised Learning

- Data points are not said to be "labelled" and we seek to find some kind of structure within the data-set.
- Ex: Clustering News on the same current affair
  - Organise computer clusters
  - Market Segmentation
  - Social Network Analysis

CLUSTERING



Linear Regression (1 VAR)

- Existence of a 'Training Set'

Training set of housing prices in Portland, OR

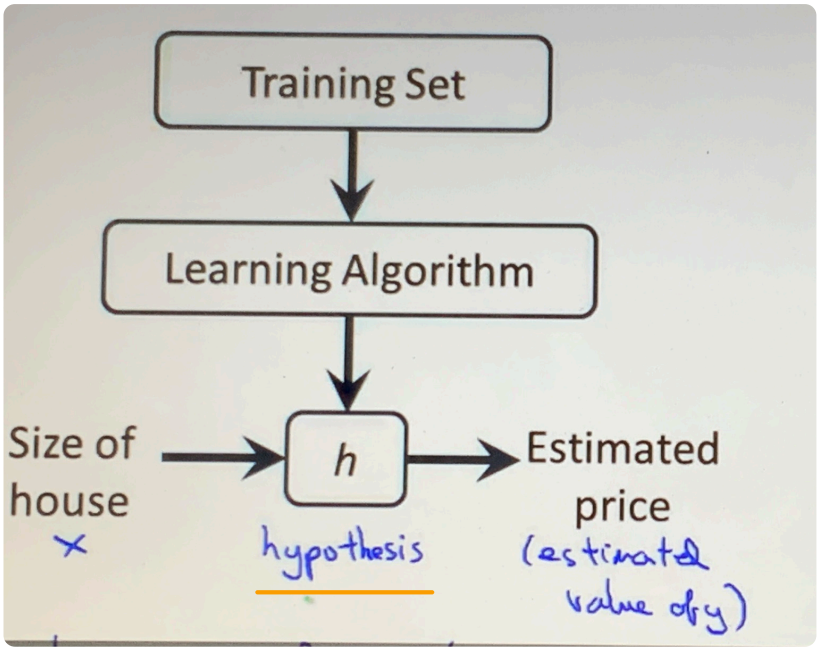
Size in Feet^2 (x)	Price (y)
2104	460
1416	232
1534	315
852	178
...	...

Notation:

- m = Number of training instances
- x = INPUT ; y = OUTPUT

FEATURE — TARGET

- (x, y) - one training example
- $(x^{(i)}, y^{(i)})$ - i^{th} training example



- Learning algorithm produces hypothesis function, h, which maps $x \longrightarrow y$

How do we represent h?

- For linear regression,

$$h_\theta(x) = \theta_0 + \theta_1 x$$

θ_i - PARAMETER VALUES → (1 VAR)

Cost Function $J(\theta_0, \theta_1)$

- Allows us to fit the best possible line to our data
- Need to choose parameters so that $h_\theta(x)$ is close to y for our TRAINING EXAMPLES

$$\underset{\theta_0\ \theta_1}{\text{MINIMIZE}} \; \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x) - y \right)^2$$

COST FUNC.

SQ. ERR.

2 just makes the numbers easier; minimizing 1/2 of something will give you the same θ_i

- Sq. error cost function is most common in linear regression type problems.

Summary

Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

Parameters: θ_0, θ_1

Cost Function: $J(\{\theta_i\}) = \frac{1}{2m} \sum_{i=1}^{m} \left[h_\theta(x^{(i)}) - y^{(i)} \right]^2$

Goal: minimize θ_0, θ_1 wrt $J(\{\theta_i\})$

Gradient Descent

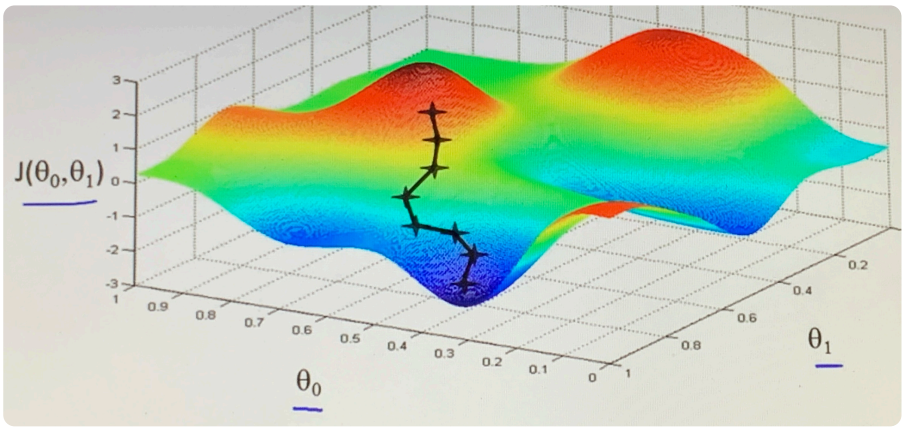
- Algorithm utilised to minimize the cost function but can be utilised more generally to minimize other function ML related.

Outline:

① Start with some set of parameters $\{\theta_i\}$

② Keep changing $\{\theta_i\}$ to reduce $J(\{\theta_i\})$ and hopefully end up at the minimum.

- Algorithm will converge at nearest local minimum from the initial starting point ∴ SENSITIVE TO INITIAL CONDITIONS.



ALG:

repeat until convergence {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\{\theta_j\}) \quad \forall\ j = 1, \ldots, n$$

ASSIGNMENT OPERATOR

}

Correct: SIMULTANEOUS UPDATE ∴ NEED TEMPORARY VARS

$$\text{temp } 0 := \theta_0 - \alpha\ \partial/\partial\theta_0\ J(\{\theta_j\})$$
$$\text{temp } 1 := \theta_1 - \alpha\ \partial/\partial\theta_1\ J(\{\theta_j\}) \quad \forall\ j = 1, \ldots, n$$
$$\theta_0 := \text{temp } 0 \qquad \theta_1 := \text{temp } 1$$

- α - LEARNING RATE: determines the STEP SIZE

in the gradient descent algorithm

$$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$
$$\theta_0 := \text{temp0}$$
$$\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$
$$\theta_1 := \text{temp1}$$

INCORRECT!
- As calculation of θ_1 will use updated θ_0 value!

NOTE: As we approach the local minimum, gradient descent will automatically decrease step size due to the smaller derivs.
∴ No need to adaptively change α.

Example:

GRADIENT DESCENT

- repeat until convergence

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\{\theta_j\})$$

for $\forall j = 1, \dots, n$

LINEAR REGRESSION

- $h_\theta(x) = \theta_0 + \theta_1 x$
- $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} [h(\theta^{(i)}) - y^{(i)}]^2$

apply G.D. to $J(\theta_0, \theta_1)$

$$\Rightarrow \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

$$\Rightarrow \frac{1}{2m} \sum_{i=1}^{m} \frac{\partial}{\partial \theta_j} [\theta_0 + \theta_1 x^{(i)} - y^{(i)}]^2$$

$$* \quad \frac{1}{m} \sum_{i=1}^{m} \{h_\theta(x^{(i)}) - y^{(i)}\} \quad \text{for } j = 0$$

$$* \quad \frac{1}{m} \sum_{i=1}^{m} \{h_\theta(x^{(i)}) - y^{(i)}\} \, x^{(i)} \quad \text{for } j = 1$$

"BATCH"

Gradient descent algorithm

repeat until convergence {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right) \cdot x^{(i)}$$

}

update θ_0 and θ_1 simultaneously

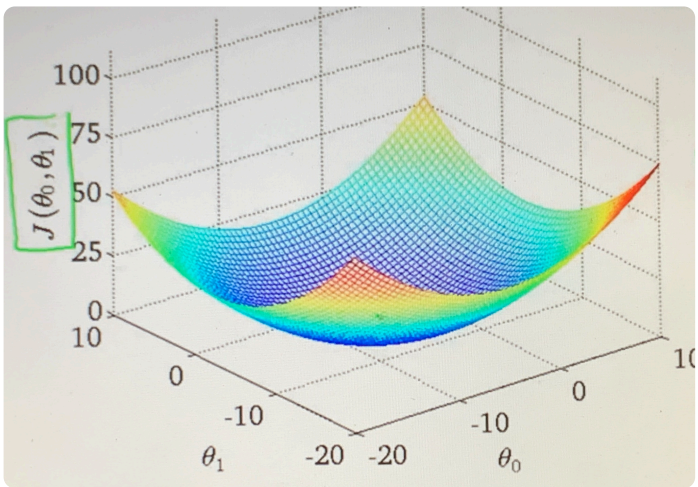
$\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$

$\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$

- For linear regression problems with a sq. err cost function. The cost function always ends up as a convex shape:

NO LOCAL MINIMA

∴ gradient descent will always converge to the GLOBAL min (assuming α not too large)



NOTE: 'Batch' refers to the fact we are looking at all the examples within the training set i.e. $\sum_{i=1}^{m}$