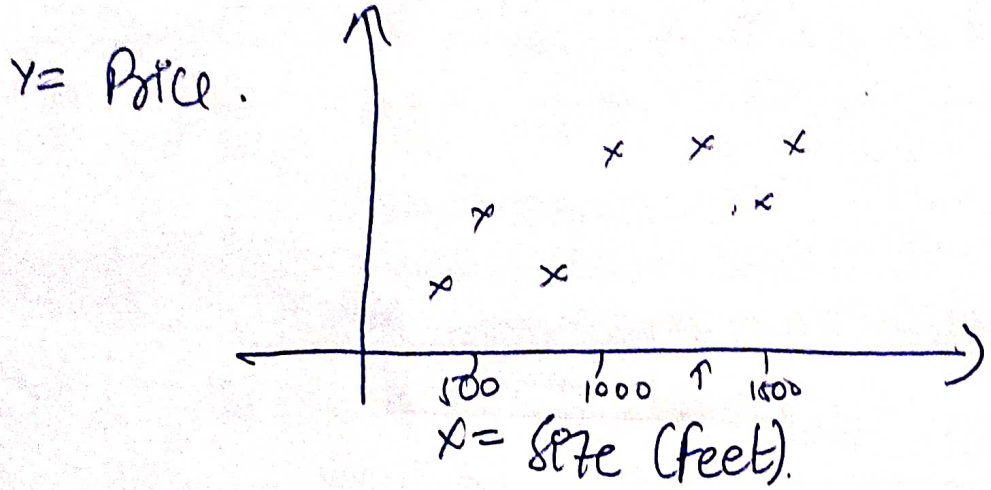
Machine learning → Field of study that gives computers the ability to learn without being explicitly programmed.

Supervised learning

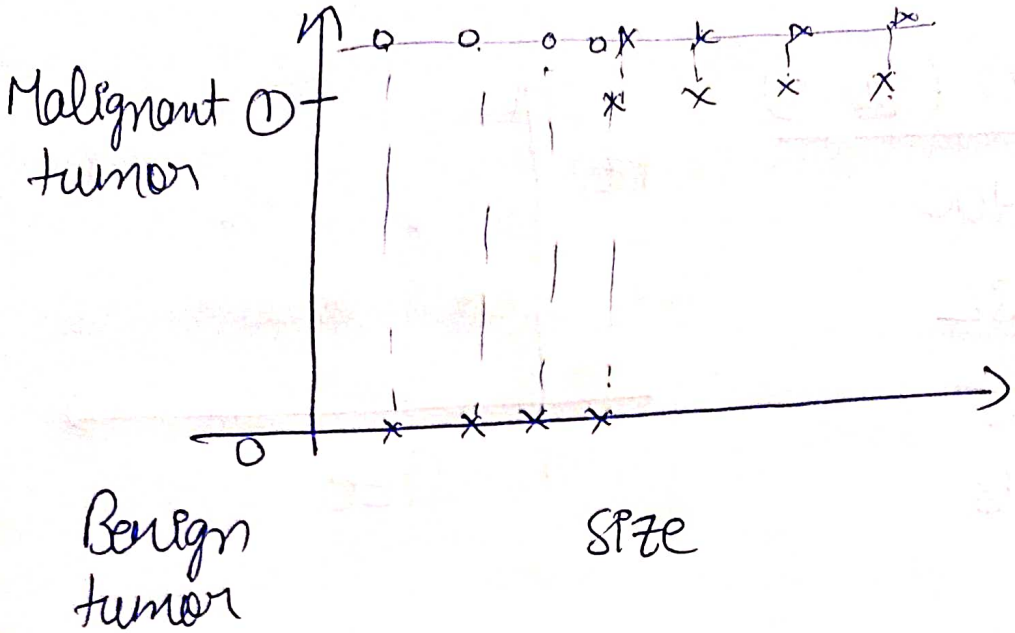


(X, Y), we need to learn mapping X → Y

This is example of Regression Problem.

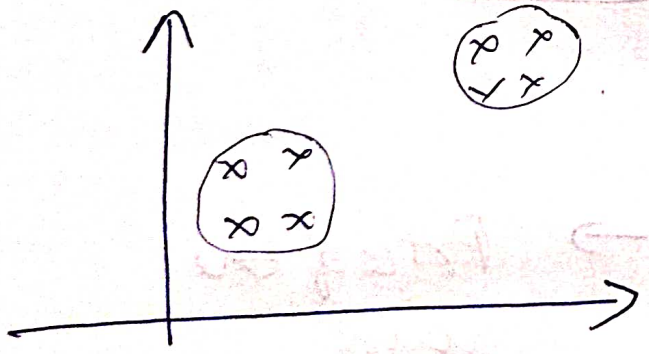
When there are discrete K number of outputs, it is known as classification problem

Example of classification ~~pro~~ problem:-



In real world we would be having $n \rightarrow$ number of inputs to predict the feature.

Unsupervised learning

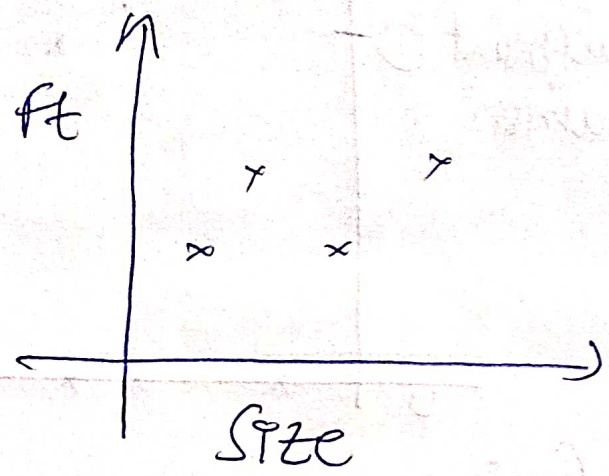


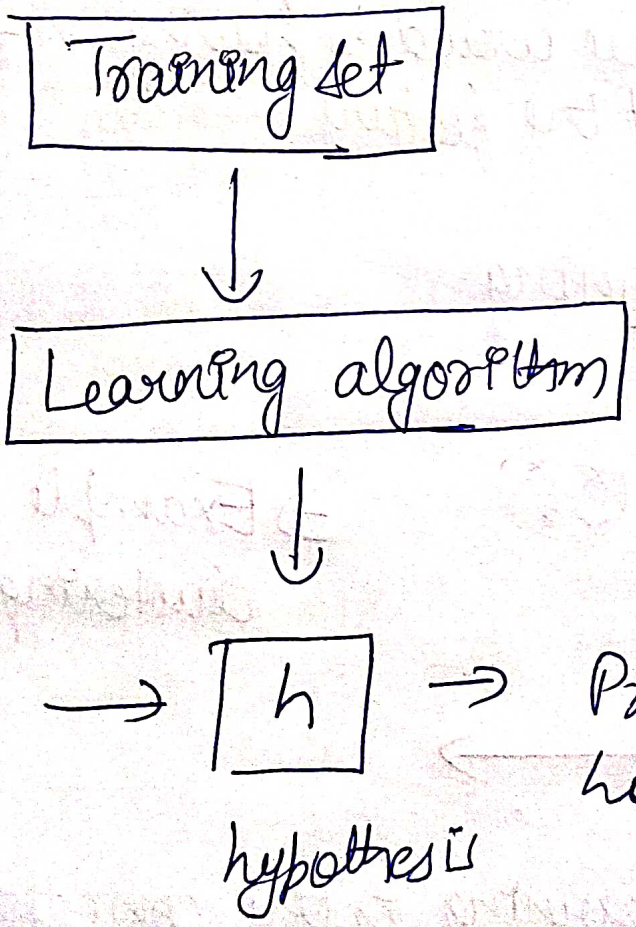
⇒ Example of clustering algorithm

Unsupervised learning is the concept of using unlabelled data

Linear Regression and Gradient descent

Size (ft)	Price ($1000)
2104	400
1416	232
1534	315
852	718





① How do we represent hypothesis?

$$h(x) = \theta_0 + \theta_1 x$$

If we have more information, then

~~$h(x) = \theta_0 + \theta_1 x$~~

x_1 = size

x_2 = no. of bedrooms

$$h(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

$$h(x) = \sum_{j=0}^{n} \theta_j x_j$$

$x_0 = 1$.

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix} \quad x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix}$$

$(x^{(i)}, y^{(i)})$ = i^{th} training example.

θ = "parameters"

m = training example or rows

x = Inputs / features

y = Output / target variable.

(x, y) = traing example

We need to choose θ such that $h(x) \approx y$ for training examples.

$h_\theta(x) = h(x)$

$h_\theta(x) \Rightarrow$ hypothesis depends on both parameters and input feature x

~~In this~~ In linear regression we need to

$$\underset{\theta}{\text{minimise}} \frac{1}{2} \sum_{i=1}^{m} (h_\theta(x) - y)^2$$

$$\underset{J(\theta)}{\text{minimise}} \Rightarrow J(\theta) = \frac{1}{2} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

Gradient Decent

Start with θ (say $\theta = \vec{0}$).

Keep changing θ to reduce $J(\theta)$.

$\alpha \rightarrow$ Learning rate

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \quad (j = 0, 1, 2 \ldots n).$$

$$\boxed{\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \quad (j = 0, 1, 2 \ldots, n)}$$

~~$\frac{\partial}{\partial \theta_j} J(\theta)$~~

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{\partial}{\partial \theta_j} \frac{1}{2} (h_\theta(x) - y(x))^2$$

$$= 2 \frac{1}{2} (h_\theta(x) - y(x)) \cdot \frac{\partial}{\partial \theta_j} (h_\theta(x) - y)$$

$$= (h_\theta(x) - y) \cdot \frac{\partial}{\partial \theta_j} (\theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 \ldots - y)$$

~~$= [h_\theta(x) - y] \times x$~~

$$\boxed{\frac{\partial}{\partial \theta_j} J(\theta) = (h_\theta(x) - y) \cdot (x_j)}$$

$$\theta_j := \theta_j - \alpha\,(h_\theta(x) - y)\cdot x_j$$

$$\theta_j := \theta_j - \alpha \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_j^{(i)}$$

Repeat until convergence

$J(\theta)$ does not have local optima, the only local optima is the global optimum.

In batch gradient descent.

Repeat {

For i = 1 to m {

$\theta_j := \theta_j - \alpha (h_\theta(x^{(i)}) - y^{(i)}) \cdot x^{(i)}$

}

}

$$\nabla_\theta J(\theta) = \begin{bmatrix} \frac{\partial J}{\partial \theta_0} \\ \frac{\partial J}{\partial \theta_1} \\ \frac{\partial J}{\partial \theta_2} \end{bmatrix}$$

↳ Derivative

$$A \in \mathbb{R}^{2\times 2} \qquad A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

$$f: \mathbb{R}^{2\times 2} \to \mathbb{R}$$

$$f(A) = A_{11} + A_{12}^2$$

$$f\left(\begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}\right) = 5 + 6^2$$

$$\nabla_A f(A) = \begin{bmatrix} \frac{\partial f}{\partial A_{11}} & \frac{\partial f}{\partial A_{12}} \\ \frac{\partial f}{\partial A_{21}} & \frac{\partial f}{\partial A_{22}} \end{bmatrix} = \begin{bmatrix} 1 & 2A_{12} \\ 0 & 0 \end{bmatrix}$$

if A is a square matrix

$tr(A)$ = sum of diagonal entries

↙

trace

① $trA = trA^T$

$f(A) = trAB \Rightarrow \nabla_A f(A) = B^T$

② $trAB = trBA$

③ $trABC = trCAB$

④ $\nabla_A trAA^TC = CA + C^TA$

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{m} (h(x^{(i)}) - y^{(i)})^2$$

$$X\theta = \begin{bmatrix} -(x^{(1)})^T - \\ -(x^{(2)})^T - \\ \vdots \\ -(x^{(m)})^T - \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_n \end{bmatrix}$$

$$= \begin{bmatrix} x^{1T}\theta \\ x^{2T}\theta \\ \vdots \\ x^{mT}\theta \end{bmatrix}$$

$$\vec{y} = \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^m \end{bmatrix}$$

$$J(\theta) = \frac{1}{2}(X\theta - y)^T (X\theta - y)$$

$$\nabla_\theta J(\theta) = \nabla_\theta \frac{1}{2}(X\theta - y)^T (X\theta - y)$$

$$= \frac{1}{2} \nabla_\theta (\theta^T X^T - y^T)(X\theta - y)$$

$$= \frac{1}{2} \nabla_\theta [= \frac{1}{2}[X^T X\theta + X^T X\theta - X^T y - X^T y]$$

$$= X^T X\theta - X^T y \quad \text{setting } \nabla = 0$$

$$X^T X\theta = X^T y$$

$$\theta = (X^T X)^{-1} X^T y$$

Multiple Linear Regression

$$\hat{y} = b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_n x_n$$

Assumptions of Linear Regression

1. Linearity (Linear relationship b/w Y and each X)
2. Homoscedasticity (Equal Variance)
3. Multivariate Normality (Normality of error dist)
4. Independence (of observation. Includes no auto correlation)
5. Lack of Multicollinearity :- $X_1 \sim X_2$
6. Outlier check.

5 methods of building models

1. All-in
2. Backward elimination.
3. Forward selection.
4. Bidirectional elimination.
5. Score Comparison.

(2), (3), (4) → Stepwise Regression.

Backward elimination

1. Select significance level
2. Fit full model with all possible predictors
3. If P > S.L go to 4 else FIN
4. Remove predictor.
5. Fit to model without this variables

(5) → (3)

Forward selection

① Select significance level.

② Fit all simple regression $y \sim x_n$. Select one with lowest p-value.

③ Keep this variable and fit all possible models with one extra predictor added to one already have

④ Consider the predictor with lowest P-value
if P<SL go to Step 3 else FIN
↓
Keep the previous model.

Birectional Elimination

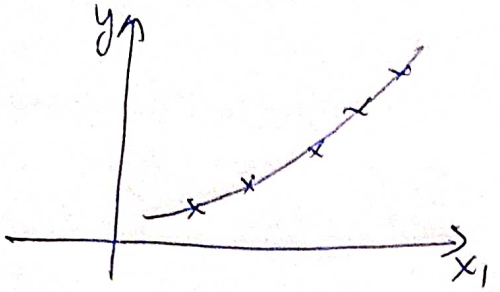
① Select a significance level to enter and to stay in the model. eg. SLEnter = 0.05 and
SL Stay = 0.05

② Perform next Step of fwd selection (P<SLEnter).

③ Perform all Steps of Backward elimination (old values must have P<SLStay)

④ No variable can enter or exit.

Polynomial Regression



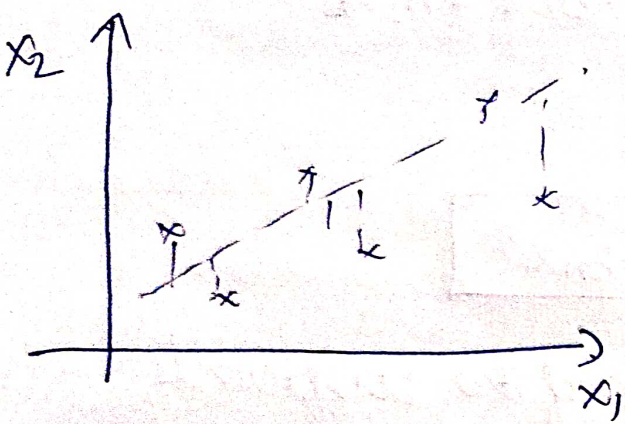
$$y = b_0 + b_1 x_1 + b_2 x_1^2$$

$$y = b_0 + b_1 x_1 + b_2 x_1^2 + \ldots + b_n x_1^n$$

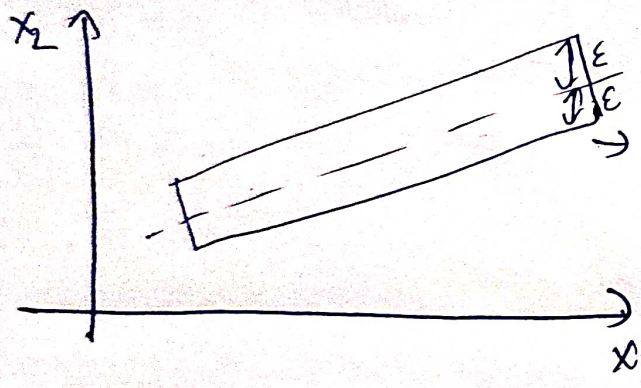
It is still called linear becoz of the coefficients are linear.

~~SVR Intro~~

SVR → Support Vector Regression



Ordinary Least squares $\sum (y - \hat{y})^2 \rightarrow \min$.



ε → Insensitive tube

→ Margin of error which is allowed.

If error is above insensitive tube then its ξ_i and if it is below insensitive tube then its ξ_i^*

$$\frac{1}{2}\|w\|^2 + C \sum_{i=1}^{m} (\xi_i + \xi_i^*) \rightarrow \min$$