Live - Machine Learning

Purpose : Clear the interview.

AI application : Lots of applications.

Machine & Deep learning
- Supervised ML
  - Regression
  - Classification
- Unsupervised ML
  - Clustering
  - Dimensionality Reduction

Supervised ML

Age	Weight
24	62
25	63
21	72
27	62

O/P

Age → Hypothesis → O/P Weight

Age ↓ Independent features

Weight ↓ Dependent features.

① Regression Problem :—
whenever we are having o/p and it is continuous then it will become regression problem.

② Classification Problem :—

No. of hours	No. of play hours	No. of sleep	O/P
			P/F

O/P → Fixed no. of category then it will be classification problem.

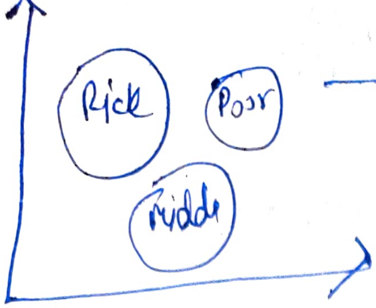
Unsupervised ML

Unsupervised ML → Clustering
→ Dimensionality Reduction

{ No dependent variable }

Salary	Age
—	—
—	—

Clustering → Customer segmentation



→ So here we segment our product to particular group based on their richness. → Clustering

② Dimensionality Reduction

1000 → Lower dimension

↳ 100

PCA, LDA.

what covered :—

~~Supervised~~

Supervised

① Linear Regression
② Ridge & Lasso
③ Logistic Reg.
④ Decision Tree
⑤ Adaboost
⑥ Random Forest
⑦ Gradient ~~Descent~~ Boosting
⑧ Xgboost
⑨ Naive Bayes.
⑩ SVM
⑪ KNN

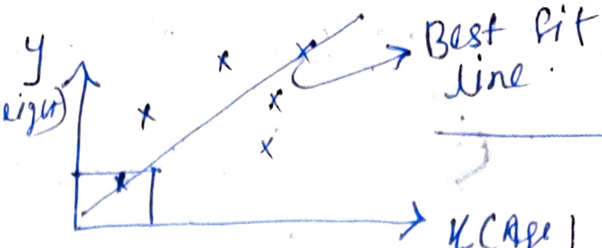
~~Un-supervised~~

Unsupervised

① K-Means.
② DB Scan
③ Hierarchical clustering
④ K-Nearest Neighbour Clustering
⑤ PCA
⑥ LDA.

Supervised

① Linear



$y = mx + c$

$y = \beta_0 + \beta_1 x$

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

Train Dataset

↓

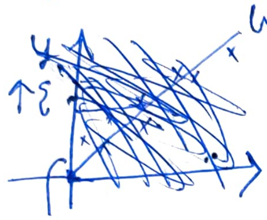
Model

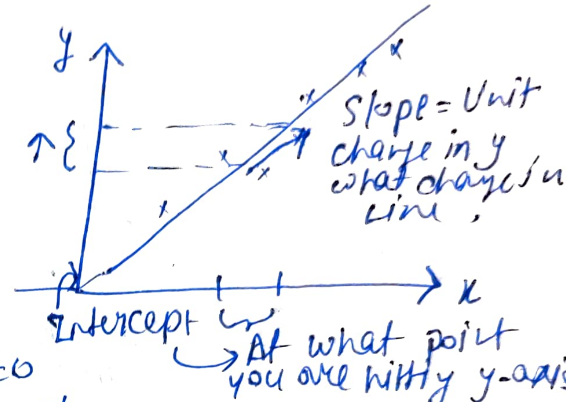
↓

New Age → Hypothesis → o/p weight.

Eqn of a straight line

$h_\theta(x) = \theta_0 + \theta_1 x \rightarrow$ Intercept when $x = 0$





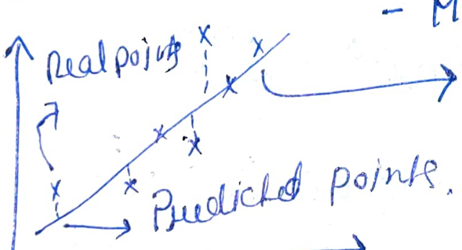
When $x = 0$

θ_0 = Intercept

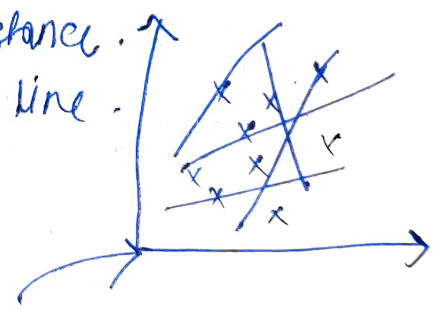
θ_1 = Slope or coefficent.

x_i = Data points.

Linear Regression



- Minimal distance.
  Best fit line.



- We can't do lots of iteration like this so we perform cost function for that problem.

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

Cost function : ~~$\frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x) - y)^2$~~ $\frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x) - y)^2$

Predicted points

To remove -ve no.

Avg. of all value

Distance b/w all points

$\frac{1}{2m} \rightarrow$ For derivation purpose.

$$\boxed{J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x) - y)^2} \rightarrow \text{Cost function}$$

↳ Squared error function.

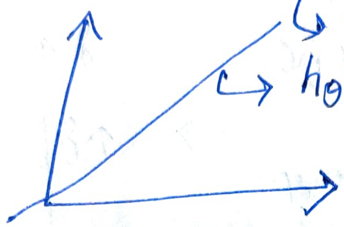
What we need to solve ?

↳ Minimise cost function by adjusting θ_0, θ_1.

So, $h_\theta(x) = \theta_0 + \theta_1 x$ If $\theta_0 = 0$.

↳ If

1



↳ $h_\theta(x) = \theta_1 x$

$\theta_1 = 1 \rightarrow$ (1, 1)
(2, 2)
(3, 3)

So, $= \frac{1}{2m}\sum_{i=1}^{3}(h_\theta(x) - y)^2$

$= \frac{1}{2m}\left[(1-1)^2 + (2-2)^2 + (3-3)^2\right]$

$\frac{1}{2m}(0) = 0$

when $\theta_1 = 1$ then $J(\theta_1) = 0$.

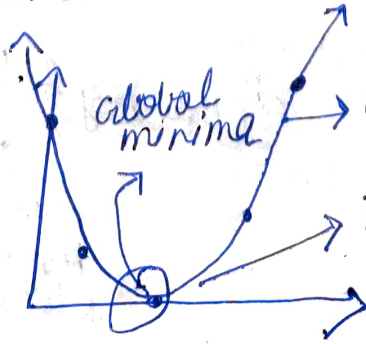
2 $\theta_1 = 0.5$

$J(\theta_1) = \frac{1}{2m}\sum_{i=1}^{3}(h_\theta(x^{(i)}) - y^{(i)})^2$

$= \frac{1}{2 \times 3}\left[(0.5-1)^2 + (1-2)^2 + (1.5-3)^2\right]$

$= \frac{1}{6}\left[0.25 + 1 + 2.25\right] = 0.58$.

3 $\theta_1 = 0$ → Some points
$\theta_1 = 2$



→ Gradient descendent

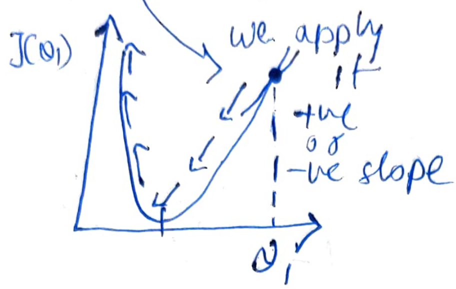
→ This is the main point for linear regression line

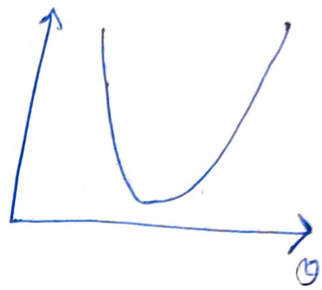
Convergence Algorithm :-

Repeat until convergence

$$\boxed{\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)}$$

↳ Derivative (slope)



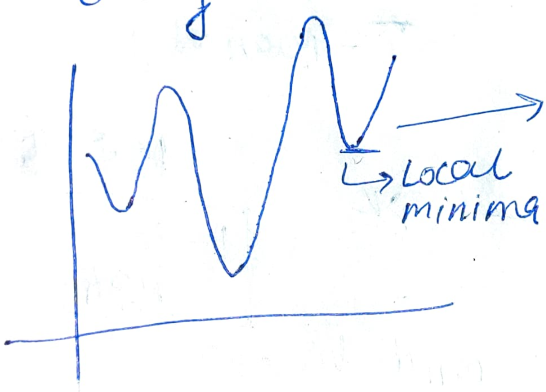


- So we need to reduce θ_1 to reach to global minima.

So, $\theta_1 = \theta_1 \cancel{+} \alpha(-ve)$

Learning rate = steps — It neither need to be so small nor so large.

Now,



(slope)
$\theta_1 = \theta_1 - \alpha(0)$
↳ Slope at local minima $= 0$.

Gradient Descent Algorithm :-

Repeat until convergence

$$\{ \; \theta_j = \theta_j - \alpha \boxed{\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)} \; \}$$

$j = 0 \Rightarrow \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)$

$j = 1 \Rightarrow \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) x^{(i)}$

Repeat until convergence;

$$\left\{ \begin{aligned} \theta_0 &:= \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right) \\ \theta_1 &:= \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right) x^{(i)} \end{aligned} \right\}$$

α - Learning rate.

<u>Performance Matrix</u> :-

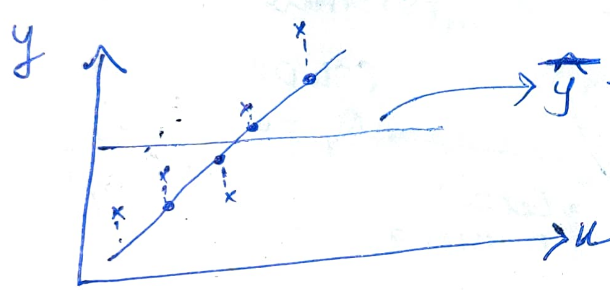
$h_\theta(x)$

1

R^2 & Adjusted R^2

$$R^2 = 1 - \frac{SS_{Res}}{SS_{Total}} = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}$$

Difference b/w real & predicted points.



$\bar{y}$ - mean of y.

$\boxed{1 - \frac{\text{Low}}{\text{High}}}$ = Big No.

- In ideal situation R^2 can't be -ve.

~~Gender~~ <u>Bedrooms</u> <u>Price</u> <u>Locations</u>

$R^2 = 85$

When we change gender then R^2 changes but ideally it should not. So then we use adjusted R^2.

<u>Adjusted R^2</u>

$$R^2 \text{ adjusted} = 1 - \frac{(1-R^2)(N-1)}{N-P-1}$$

$p = 2 = R^2 = 90\%$
R^2 adjusted $= 86\%$.

P = Features or predictors.

$P = 3 = R^2 = 91\%$
R^2 adjusted $= 82\%$

Why R^2 adjusted decreased ?

N = No. of data points.
P = No. of predictors.

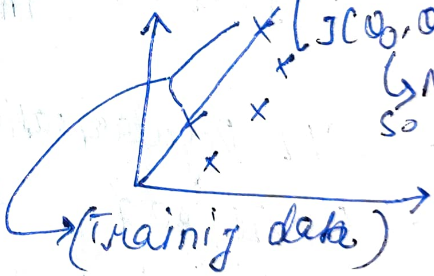
Because of entire eqn as $p \uparrow$ $N-P-1 \uparrow\uparrow$
which cause R^2 adjusted $\downarrow$
So, as $p >>>$, R^2 adjusted will be less then previous one.

Ridge & Lasso Regression

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

$J(\theta_0, \theta_1)$
give gradient descendent

$\left\{ \begin{array}{l} \theta_0 = 0. \\ J(\theta_0, \theta_1) = 0 \end{array} \right\}$
No diff so 0.



Overfitting: (My model performs well with training data) but [fails/to perform well with test data]?

Condition - Low Bias
High variance.