AI Vs ML Vs DL Vs DS

1) Artificial Intelligence
2) Machine Learning
3) Deep Learning
4) Data Science

AI - To create an application which can perform its own task without any human intervention

Ex: Netflix Recommendation System

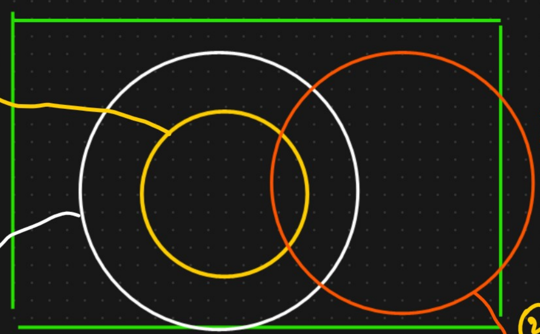
→ Action Movies → Action Movies Recommendation

② Self Driving Car

{Multi Layered NN}

Deep Learning

{ Mimic the Human Brain



ML

Data Science

It provides Stats tools to analyze, visualize, prediction, forecasting the data.

3 Types of Machine Learning.

Machine Learning Techniques

Supervised ML — Unsupervised ML — Reinforcement Learning

Supervised ML Technique

- Regression ✓
- Classification ✓

Predict House price

Dataset

Independent feature

Dependent or o/p feature

Size of House	# of Rooms	Price
5000	5	450K
6000	6	500K
—	—	—
—	—	—

Continuous

1. Dependent or o/p feature
2. Continuous → Regression
3. Categorical → Classification

Classification

Dependent | Feature

No. of study hours	No. of play hours	Pass/Fail
7	3	Pass
2	6	Fail

May Be

Binary classification

OR

Multiclass classification

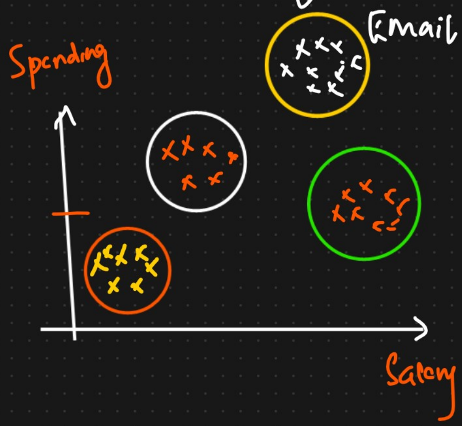
② Unsupervised ML : Eg: Customer Segmentation → Clusters

Ecommerce Company

⇓

Email with some discount to

Salary	Spending Score (1-10)
20000	9
45000	2
–	–
–	–
–	–



Supervised ML

1. Linear Regression
2. Ridge & Lasso
3. ElasticNet
4. Logistic Regression (Classification)
5. Decision Tree
6. Random Forest
7. AdaBoost
8. Xgboost

$\left.\begin{array}{l}\text{Decision Tree}\\ \text{Random Forest}\\ \text{AdaBoost}\\ \text{Xgboost}\end{array}\right\}$ Both Classification & Regression.

Unsupervised ML

1. K Means
2. Hierarichel Mean
3. DBScan clustering

Reinforcement Learning

Simple Linear Regression

Supervised ML $\longrightarrow$ Regression

TRAIN

$\downarrow$

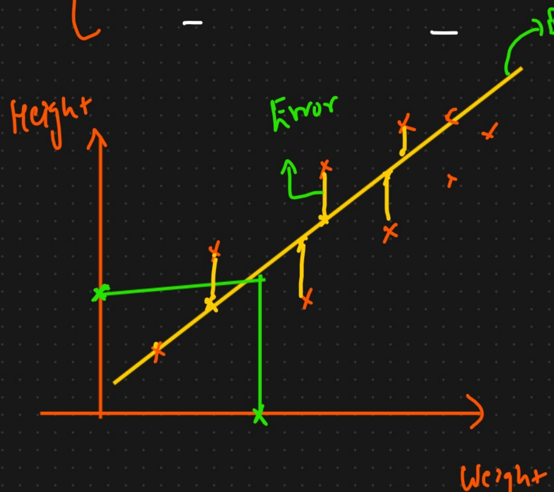
New weight $\longrightarrow$ $\boxed{\text{Model}}$ $\longrightarrow$ Height

$\uparrow$

Simple Linear Regression

<u>Dataset</u> I/P features

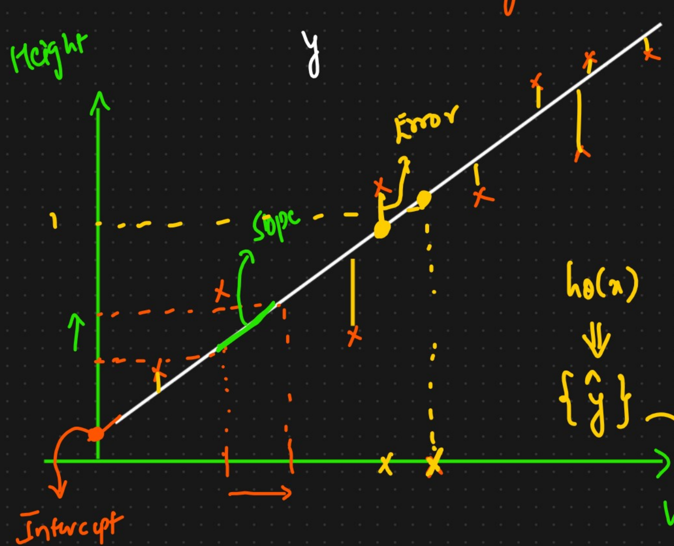
$\uparrow$ X — Weight	Y $\uparrow$ O/P or dependent feature — Height
74	170cm
80	180cm
75	175.5cm
–	–



Minimal

θ_0 & θ_1 Minimal Error

Dataset



$h_\theta(x)$

$\Downarrow$

$\{\hat{y}\}$ $\longrightarrow$ predicted point

$y = mx + c$

$y = \beta_0 + \beta_1 x$

$$\boxed{h_\theta(x) = \theta_0 + \theta_1 x}$$

$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \dots + \theta_n x_n$

<u>Error</u> $(y - \hat{y})$

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

if $x = 0$

$h_\theta(x) = \theta_0$

θ_0 = Intercept

θ_1 = Slope or Coefficient

Cost function

predicted ⇑   True O/P ⇑

$$J(\theta_0,\theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x)^{(i)} - y^{(i)}\right)^2 \Rightarrow \text{Mean Squared Error}$$

↳ Error

Final Aim what we need to solve

Minimize $\quad J(\theta_0,\theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x)^{i} - y^{(i)}\right)^2$ ↓↓↓

θ_0,θ_1

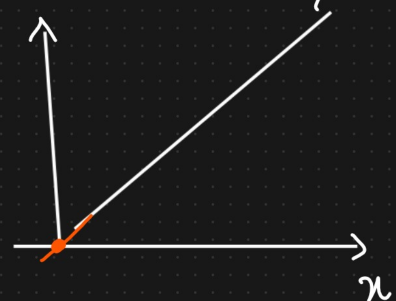
DATASET

x	y
1	1
2	2
3	3

① $h_\theta(x) = \theta_0 + \theta_1 x$

$\boxed{\theta_0 = 0}$

$\boxed{h_\theta(x) = \theta_1 x}$



$\theta_0 = 0$

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

let $\theta_1 = 1$ {slope}

$h_\theta(x) = 1 \quad x=1$

$h_\theta(x) = 2 \quad x=2$

$h_\theta(x) = 3 \quad x=3$

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

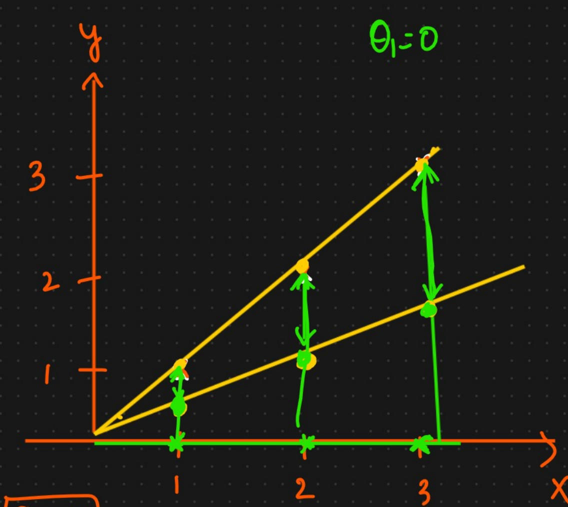
↓

let $\theta_1 = 0.5$

$h_\theta(x) = 0.5 \quad$ if $x=1$

$h_\theta(x) = 1 \quad$ if $x=2$

$h_\theta(x) = 1.5 \quad$ if $x=3$

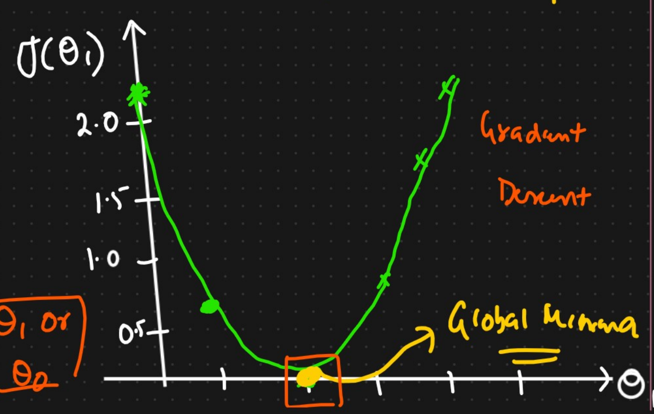


$\boxed{\theta_1 = 1}$

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

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

$\boxed{\theta_1 \text{ or } \theta_0}$



$J(\theta_1) = 0$ ← Error has been minimized

$\theta_1 = 0.5$

0.5 1 1.5 2.0 2.5

$$J(\theta_1) = \frac{1}{2\times3}\left[(0.5-1)^2 + (1-2)^2 + (1.5-3)^2\right]$$

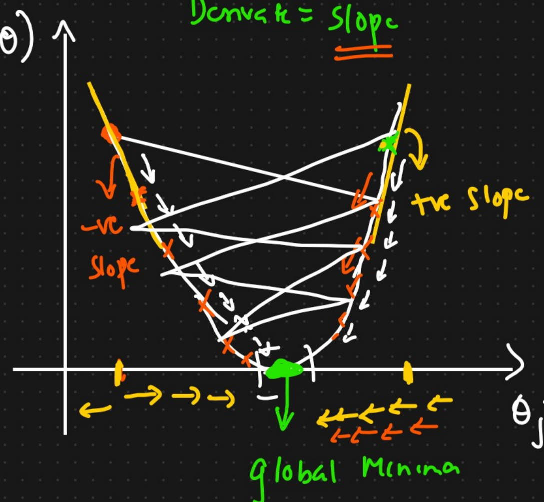
$$J(\theta_1) = \approx 0.58$$

if $\theta_1 = 0$

$$J(\theta_1) = \frac{1}{2\times3}\left[(0-1)^2 + (0-2)^2 + (0-3)^2\right]$$

$$J(\theta_1) \approx 2.3$$

Convergence Algorithm {Optimize the changes of θ_1 value}



Repeat until Convergence

{

$$\left\{ \theta_j := \theta_j - \alpha \boxed{\frac{\partial}{\partial \theta_j} J(\theta_j)} \right. \rightarrow \text{-ve}$$

}

θ_1 value much more efficiently

$$\theta_j := \theta_j - \alpha(+ve)$$

$$= \theta_j - (+ve)$$

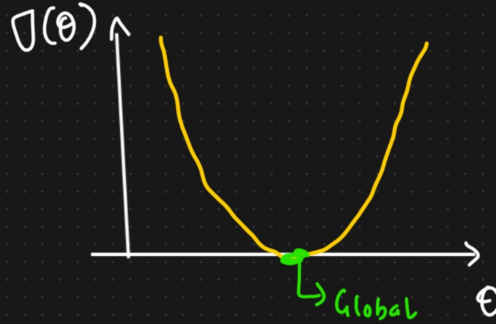
$$\left\{ \begin{aligned} \theta_j &= \theta_j - \boxed{\alpha}(-ve) \\ \theta_j &= \theta_j + (+ve) \end{aligned} \right\}$$

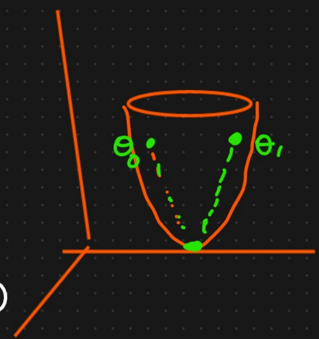
α : Learning Rate

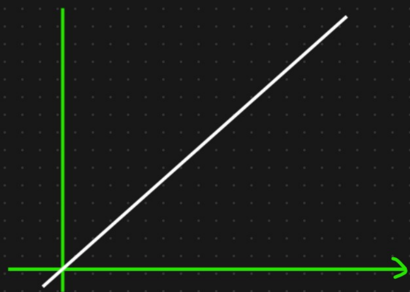
$$\boxed{\alpha = 0.001} \leftarrow$$

Final Conclusion

GRADIENT DESCENT







$\theta_0 = 0$

$\theta_1 =$ Changeable

Convergence Algorithm

repeat until convergence

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x)^{(i)} - y^{(i)}\right)^2$$

{

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

}

$j = 0$ and 1

$$\frac{\partial}{\partial x} (x)^2 = 2x$$

$$\frac{\partial}{\partial x} x^n = n x^{n-1}$$

$$\rightarrow \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) = \frac{\partial}{\partial \theta_j} \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x)^{(i)} - y^{(i)}\right)^2$$

if

$$h_\theta(x) = \theta_0 + \boxed{\theta_1 x} \longrightarrow 0$$

$$j = 0 \Rightarrow \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{\partial}{\partial \theta_0} \frac{1}{2m} \left[\sum_{i=1}^{m} \left(h_\theta(x)^{(i)} - y^{(i)}\right)^2 \right]$$

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

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

$$= \frac{1}{m} \sum_{i=1}^{m}\left((\theta_0+\theta_1 x) - y^{(i)}\right)(x)$$

$$\boxed{\frac{\partial}{\partial \theta_1}\left[\theta_0 + \theta_1 x\right] \Rightarrow \underline{\underline{x}}}$$

Repeat until Convergence

$$\{$$

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

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

$$\}$$

Multiple Linear Regression

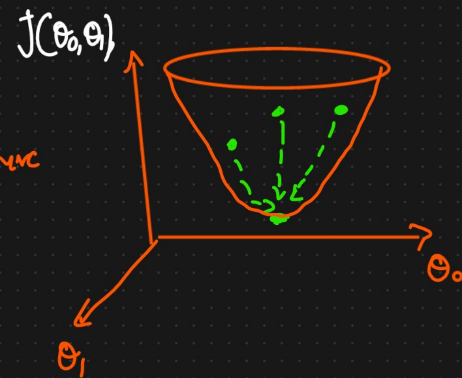
Dataset

I/p	O/p feature
Weight	Height
—	—
—	—

$h_\theta(x) = \theta_0 + \theta_1 x$ ← I/p or Independent

θ_0 = Intercept

θ_1 = Slope



House Pricing Dataset

$$\underbrace{\text{No.of Rooms} \quad \text{Size of House} \quad \text{Location}}_{\text{Independent features}} \quad \overbrace{\text{Price}}^{\text{O/p feature}}$$

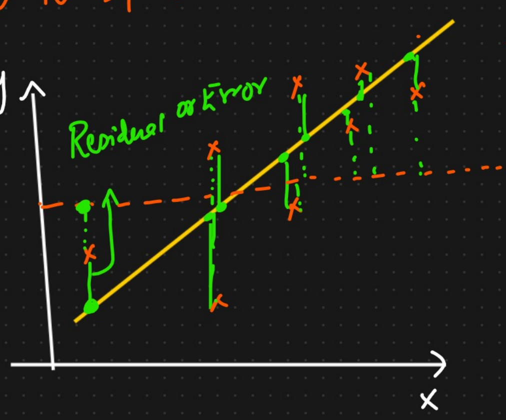
$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$$ {Multiple Linear Regression}

$\theta_1, \theta_2, \theta_3$ = coefficient

θ_0 = Intercept

Performance Metrics Used In Linear Regression

① R squared ② Adjusted R squared



$$R\text{squared} = 1 - \frac{SS_{Res}}{SS_{TOTAL}} \qquad \hat{y}_i$$

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

$$= 1 - \frac{\text{Small number}}{\text{Bigger number}}$$

$= 1 -$ Smaller number $\rightarrow \boxed{1}$

≈ 1 0.70 ⇒ 70%

↓

Accurate My Model Is?

0.85 ⇒ 85%

{Overfitting, Underfitting} 0.90 ⇒ 90%

R squared ↑↑↑

Size of the house ↑ Price ↑

tve Correlaton

② Adjusted R squared

Dataset → (Price)

No. of bedroom ↑ Price ↑

tve correlation

Gender	Size of the house	No. of bedrooms	Location	Price

R squared = 75% ⇒ 0.75

This is the problem of R Squared

R squared $\Rightarrow$ 80% $\Rightarrow$ 0.80

R squared $\Rightarrow$ 85% $\Rightarrow$ 0.85

R squared $\Rightarrow$ 87% $\Rightarrow$ 0.87

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

$P=2$ $R^2 = 90\%$ R^2 adjusted $= 86\%$

$P=3$ $R^2 = 92\%$ R^2 adjusted $= 82\%$

N = No. of data points

P = No. of Independent features

Overfitting And Underfitting (Bias And Variance)

1. Training dataset
2. Test dataset
3. Validation dataset

70%-30%

Dataset

1000 datapoints

→ TRAINING DATASET (700) → Model To TRAIN

→ TEST DATASET (300) → Testing Model

Size of House	No. of bedrooms	Price
—	—	—
—	—	—
—	—	—
—	—	—

TRAINING DATASET

TRAIN	VALIDATION
↓	↓
Train the model	Hyperparameter Tunning your Model

TRAIN	Very Good Accuracy (90%) [Low Bias]	Very Good Accuracy (90%) [Low Bias]
TEST	Very Good Accuracy (85%) [Low Variance]	Bad Accuracy (50%) [High Variance]

⇑ → Generalized Model

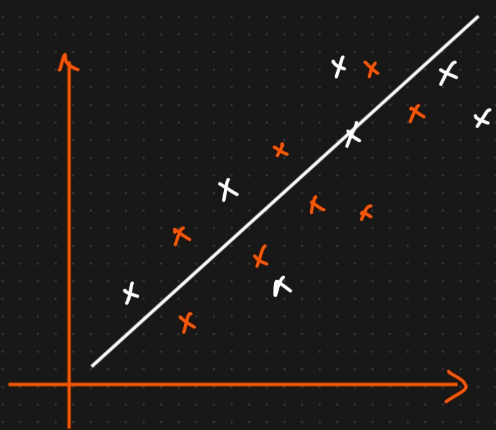
⇓ Model is Overfitting

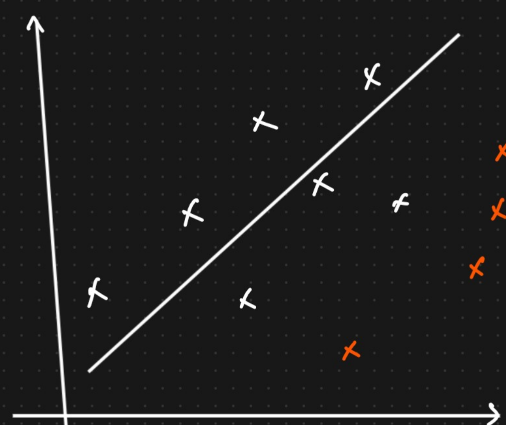
TRAIN Model Accuracy is Low [High Bias]

TEST Model Accuracy is Low [High Variance]

⇓

Model is underfitting





Generalized Model

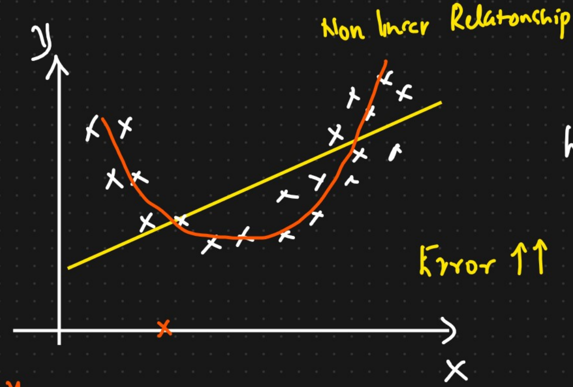
⇓

Low Bias, Low Variance

Overfitting

Low Bias, High Variance

Polynomial Regression



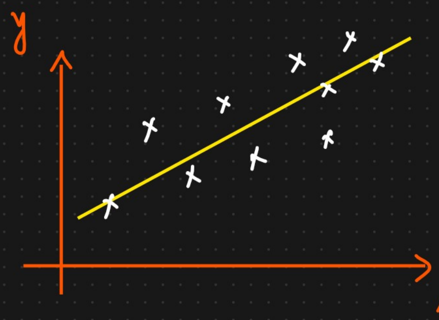
Error ↑↑

$h_\theta(x) = \beta_0 + \beta_1 x$ — Simple Linear Regression

$h_\theta(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$ — Multiple Linear Regression

Error ↓↓ ⟶ Polynomial Regression

Polynomial Degrees



→ Hyperplane

degree = 0

Simple Polynomial Regression {1 I/p and 1 o/p feature}

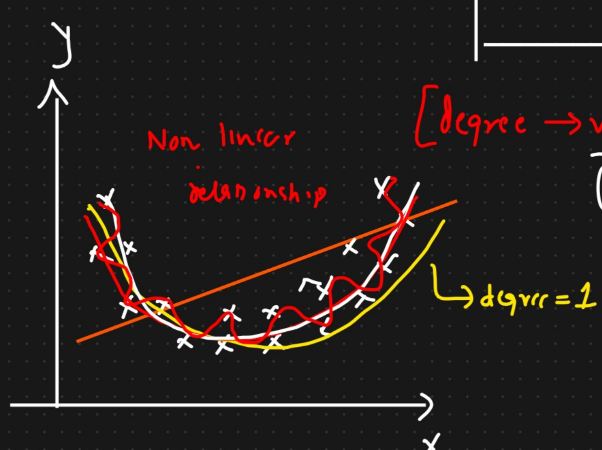
deg = 0

polynomial degree = 0 $\quad h_\theta(x) = \beta_0 \times x^0$ ⇒ Constant value

polynomial degree = 1 $\quad \boxed{h_\theta(x) = \beta_0 * x^0 + \beta_1 * x^{(1)}}$ → Simple Linear Regression

polynomial degree = 2 $\quad h_\theta(x) = \beta_0 * x^0 + \beta_1 * x^{(1)} + \beta_2 * x^{(2)}$

polynomial degree = n $\quad \boxed{h_\theta(x) = \beta_0 * x^0 + \beta_1 * x^1 + \beta_2 * x^{(3)} + \beta_3 * x^{(3)} + \dots \beta_n * x^{(n)}}$



[degree → values]

degree = 15

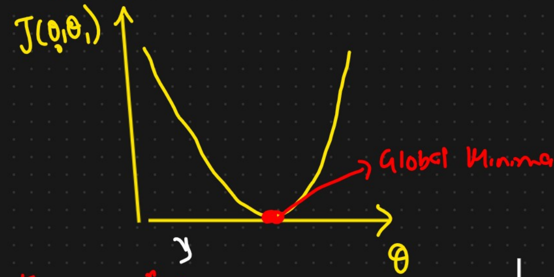
{2 independent feature}

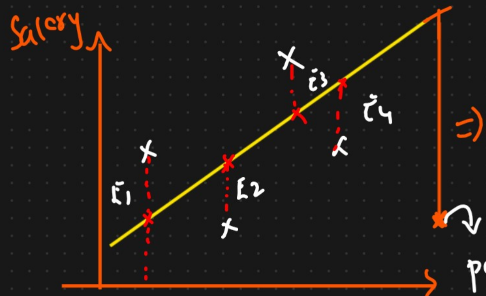
degree = 1 $\quad h_\theta(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$

$\boxed{\text{degree} = 2}$ $\quad h_\theta(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2$

MSE, MAE, RMSE [Cost function] → Performance Metrics

R^2 and Adjusted R^2

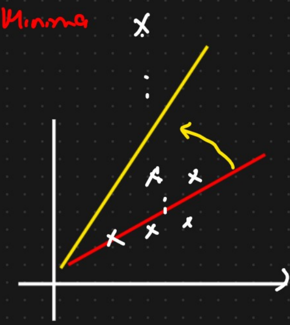




=) Squaring

penalizing the outliers

Exp	Salary
–	–
–	–
–	–

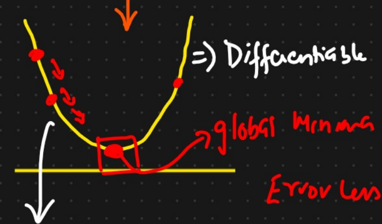


$\boxed{MSE \uparrow\uparrow}$

MSE

$$MSE = \frac{\sum_{i=1}^{n} (y-\hat{y})^2}{n}$$ =) Cost function ↓↓

→ Quadratic Equation



Advantage

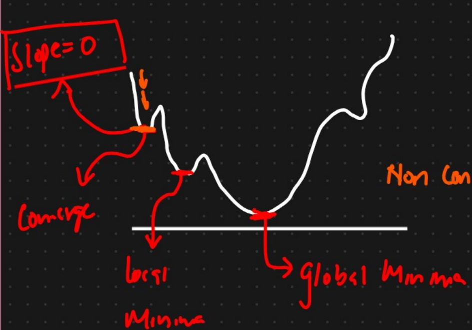
1. Differentiable ✓
2. It has one local and one global Minima
3. Converges faster

Disadvantage

1. Not Robust to outliers
2. It is not in same unit.

$ax+by+c$

$\boxed{(a-b)^2 = a^2 - 2ab + b^2}$



Non Convex functions

$$\boxed{MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}$$

Salary (lakhs)

x y $(y_i-\hat{y}_i)^2$ $(\text{lakh})^2$ (MSE)

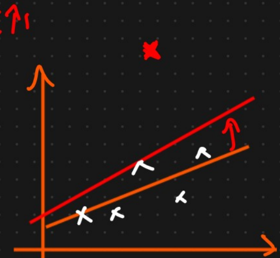
Error c 2.5 =) $(\text{lakh})^2$ ⇇

② Mean Absolute Error (MAE)

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y-\hat{y}|$$

lakhs

MAE↑↑

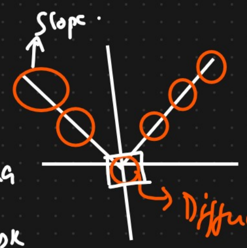


Advantage

1. Robust to outliers ✓
2. It will be in the same unit

Disadvantage

1. Convergence usually take more time. Optimization is a complex task
2. Time consuming



Subgradient

Cannot Differentiate at 0

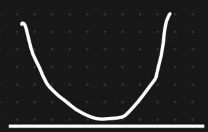
3. RMSE {Root Mean Squared Error}

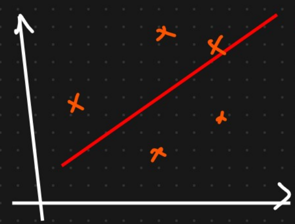
$\boxed{\text{MAE, MSE, RMSE}}$

Performance metric

$$RMSE = \sqrt{MSE}$$

$$= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$





→ MSE ✓
→ MAE ✓
→ RMSE ✓
↳ R^2 ✓
↳ Adjusted R^2 ✓

Advantage

1. Same unit
2. Differentiable

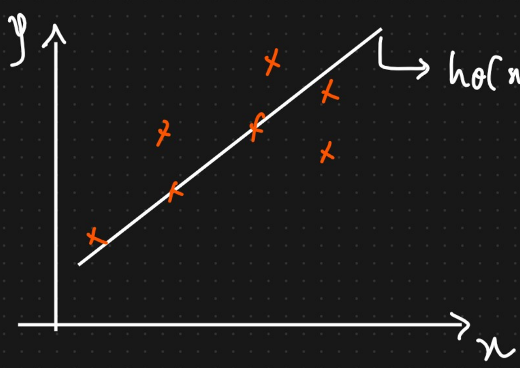
Disadvantage

1. Not Robust to outliers

MSE vs MAE vs RMSE
R^2 vs Adjusted R^2

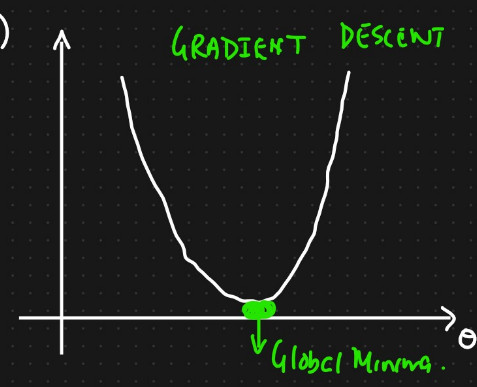
Ridge Regression, Lasso Regression, Elasticnet Regression

Linear Regression



Independent features

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



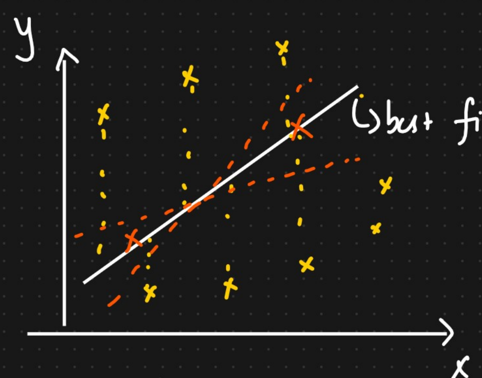
$$\text{Cost fn} = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x)^{(i)} - y^{(i)}\right)^2$$

⇓

Mean Squared Error

① Ridge Regression (L2 Regularization) ⟶ Reduce Overfitting

Overfitting



Traindata ⟶ Acc↑ ⟶ Low Bias

Test data ⟶ Acc↓ ⟶ High Variance

$\lambda = 0$

$\lambda = 1$

$\lambda = 10$

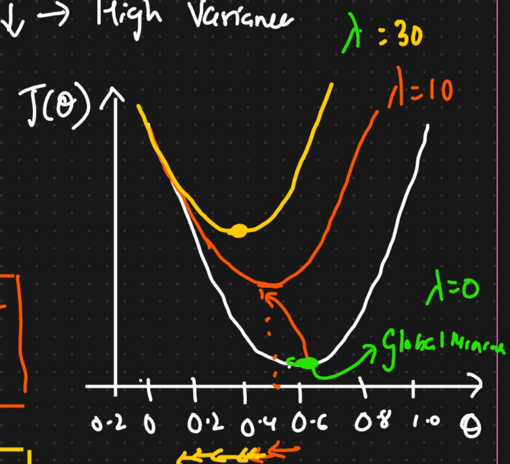
Hyperparameter

$h_\theta(x) = \theta_0 + \theta_1 x_1$

$$\text{Cost fn} = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x)^{(i)} - y^{(i)}\right)^2 + \boxed{\lambda \sum_{i=1}^{n} (\text{slope})^2}$$

$$= \quad 0 \quad + \quad (1)\left[(\theta_1)^2\right]$$

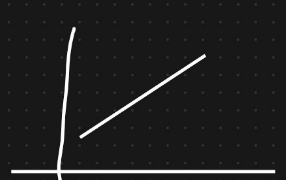
$\boxed{\lambda\uparrow \quad \text{Slope}\downarrow}$



λ : Hyperparameter

$\boxed{\lambda = 1}$

> 0



$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$

0.24 ⇓

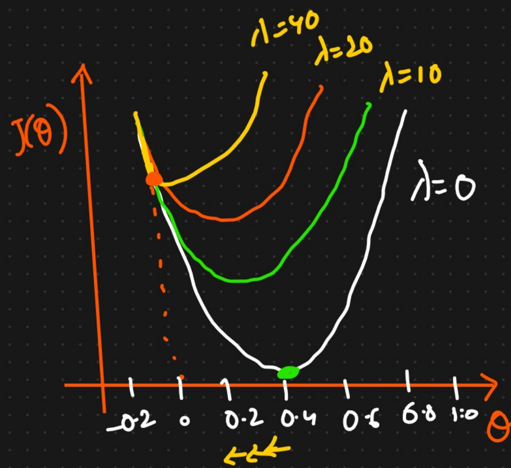
$= 0.34 + \underline{0.52}x_1 + \underline{0.48}x_2 + 0.24x_3$

⇓ ⇓ ⇓⇓

$$= 0.34 + 0.40x_1 + 0.38x_2 + \boxed{0.14x_3}$$

(2) Lasso Regression (L1 Regularization) → Feature Selection

$$\text{Cost fn} : \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x)^{(i)} - y^{(i)}\right) + \boxed{\lambda \sum_{i=1}^{n} |\text{slope}|}$$



$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4$$

$$h_\theta(x) = 0.52 + 0.65x_1 + 0.72x_2 + 0.34x_3 + \boxed{0.12}\,x_4$$

⇓

Lasso Regression

$$= 0.52 + 0.51x_1 + 0.60x_2 + 0.14x_3 + \boxed{0 \times \cancel{x_4}} \nearrow 0$$

(3) ElasticNet Regression

→ (1) Reduce Overfitting

→ (2) Feature Selection

$$\text{Cost fn} = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x)^{(i)} - y^{(i)}\right)^2 + \underbrace{\boxed{\lambda_1 \sum_{i=1}^{m} (\text{slope})^2}}_{\text{Reduce Overfitting}} + \underbrace{\boxed{\lambda_2 \sum_{i=1}^{m} |\text{slope}|}}_{\text{Feature Selection}}$$

{ Hyperparameter Tuning the Linear Regression }

Types of Cross Validation

1000 records

DATASET

70% → TRAINING

30% → TEST ← To check the performance of Model

① TRAINING OUR MODEL

② Hyperparameter Tuning

TRAINING → TRAIN, VALIDATION

CROSS VALIDATION

VALIDATION ← ⇒ Random_state

85% 92% 78%

① Leave One Out CV (LOOCV)

① Leave P out CV

| P=10 P=20 P=30 |

TRAINING → 500 Records ↑↑ Complexity of Training Model

Exp 1 | TRAINING | → Acc 1

Validation

Exp 2 | TR | | TRAINING | → Acc 2

⋮

Exp 500 | TRAINING | V | → Acc_{500}

① Overfitting → TRAINING ↑↑ Acc → New Test Data → Acc ↓↓

Validation Acc ↓↓

③ K Fold CV

K=5   n=500

Binary Classification

1 or 0's

Test size = $\frac{500}{5}$ = 100

CV=1   Exp 1 | TRAINING | → Acc 1

Validation

Exp 2 | /// / / / / / / / / / | → Acc 2

Avg Accuracies

⋮

TRAINING   Validation

Exp 5 | | → Acc 5

④ Stratified K Fold CV

K=5

60 1's
40 0's

60:40

Exp 1 | TRAINING |

Validation

| TRAINING |

Test size = $\frac{500}{5}$ = 100

⇓

Validation Data

⑤ Time Series CV

Reviews

Product Sentiment Analysis

Time

JAN → DEC

TRAINING   Validation

DAY 1   DAY 2   DAY 3   DAY 4 | . - - - - DAY N

Time Series Application

Performance Metrics, Accuracy, Precision, Recall And F-Beta

Topics to be covered

1. Confusion matrix
2. Accuracy
3. Precision
4. Recall
5. F-Beta Score

Logistic Regression

R squared

Adjusted R squared

Dataset

x_1	x_2	y (O/P)	$\hat{y}$ (pred by model)
–	–	0	1
–	–	1	1
–	–	0	0
–	–	1	1
–	–	1	1
–	–	0	1
–	–	1	0
–	–		

1. Confusion Matrix

Actual Values

Predicted Values	1	0
1	3	2
0	1	1

Actual

Predicted	1	0
1	TP	FP
0	FN	TN

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN}$$

$$= \frac{3+1}{3+2+1+1}$$

$$= \frac{4}{7}$$

Dataset — Binary classification

→ 1000 datapoints → 900 → 1 ; 100 → 0 } Imbalanced Dataset

⇓

90% Accuracy

(1) Precision $= \dfrac{TP}{TP+FP\downarrow}$ } Out of all the actual value how many are correctly predicted

	1	0	Actual
1	TP	FP	
0	FN	TN	

Predicted

(2) Recall $= \dfrac{TP}{TP+FN\downarrow}$ $\Rightarrow$ Out of all the predicted value how many are correctly predicted

Usecase (1)

Spam classification

Mail $\rightarrow$ Spam
Model $\rightarrow$ Spam } Good

		Spam 1	Not Spam 0	Actual
Spam	1	TP	FP ✓ $\Downarrow$	
Not Spam	0	FN	TN	

Mail $\rightarrow$ Not Spam
Model $\rightarrow$ Spam } Blunder

$\Downarrow$

Precision $= \dfrac{TP}{TP+FP}$

Usecase 2

To predict whether person has diabetes or not

✓ Truth $\rightarrow$ diabetes
✓ Model $\rightarrow$ Doesn't diabetes } Blunder

	Diabetes	No Diabetes
Diab	TP	FP
No Diab	FN	TN

$\downarrow$ $\Downarrow$

usecase of diseases

Recall $= \dfrac{TP}{TP+\boxed{FN}\Downarrow}$

Truth $\rightarrow$ diabetes
Model $\rightarrow$ " } good

Truth → Not diabetes
Model → Diabetes

$\Rightarrow$ 2nd opinion
⇓
check

Assignment

(*) Tommorow the stock market will crash or not

Reducing FP↓ or FN↓

(*) F-Beta Score : $(1+\beta^2)\dfrac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$

① If FP & FN are both important

$\beta = 1$

$$\text{F1 Score} = 2 \times \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

$\Rightarrow$ Harmonic Mean

② If FP is more important FN

$\beta = 0.5$

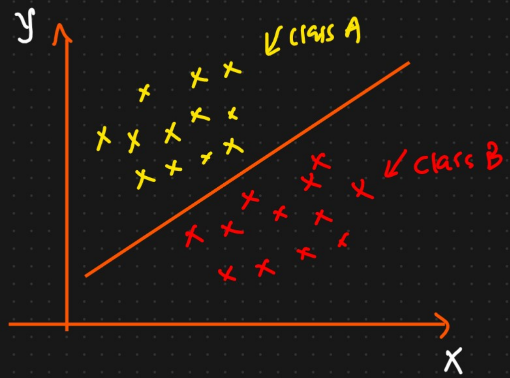
$$\text{F 0.5 Score} = (1+0.25)\frac{P * R}{P + R}$$

③ If FN >> FP

$\beta = 2$

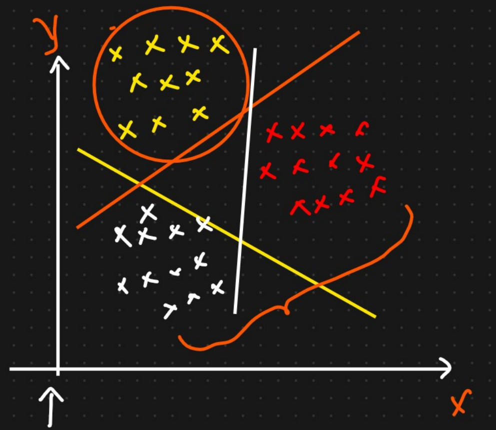
$$\boxed{\text{F 2 Score} = (1+4)\frac{P \times R}{P + R}}$$

Logistic Regression (One Versus Rest)



① Binary Classification

One Versus One



Multiclass Classification {Logistic Regression}

→ M_1 → Binary classification
→ M_2 → Binary classification
→ M_3 → Binary classification

One Versus Rest (OVR) → Logistic Regression

O_1, O_2, O_3

f_1	f_2	f_3	O/p	$\boxed{O_1}$	O_2	$\boxed{O_3}$
–	–	–	O_1	1	0	0
–	–	–	O_2	0	1	0
–	–	–	O_3	0	0	1
–	–	–	O_1	1	0	0
–	–	–	O_3	0	0	1
–	–	–	O_2	0	1	0

Model

M_1 ← I/p $\{f_1, f_2, f_3\}$ O/p $\{O_1\}$
M_2 ← I/p $\{f_1, f_2, f_3\}$ O/p $\{O_2\}$
M_3 ← I/p $\{f_1, f_2, f_3\}$ O/p $\{O_3\}$

0.55 → O/P = $\boxed{O_3}$ → Category 3

$[0.25, 0.20, \boxed{0.55}]$

M1 M2 $\boxed{M3}$

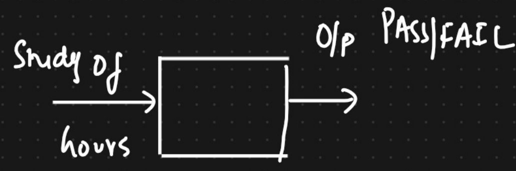
O/p

New Test Data → M1 → 0.25 ✓
→ M2 → 0.20 ✓
→ M3 → 0.55 ✓

Logistic Regression (Binary classification)

Dataset

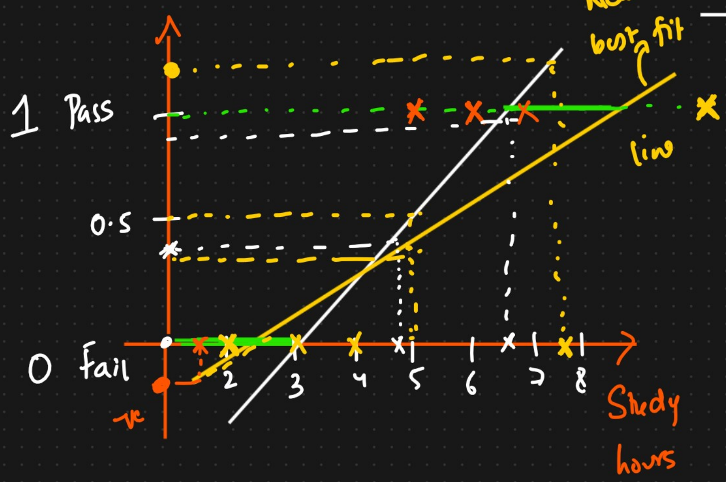
Study hours	Pass/Fail O/p {Binary categories}
2	Fail
3	Fail
4	Fail
5	Pass
6	Pass
7	Pass
12	Pass



Logistic Regression

0 to 1

Linear Regression



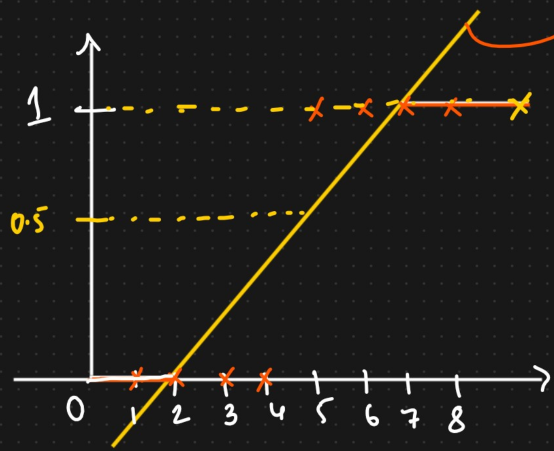
$$\begin{cases} \le 0.5 \Rightarrow 0 \\ > 0.5 \Rightarrow 1 \end{cases}$$

{ > 1 and < 0 }

Why we cannot use Linear Regression for Classification?

1. Outlier {Best fit line Change}
2. > 1 and < 0 {Squash line}

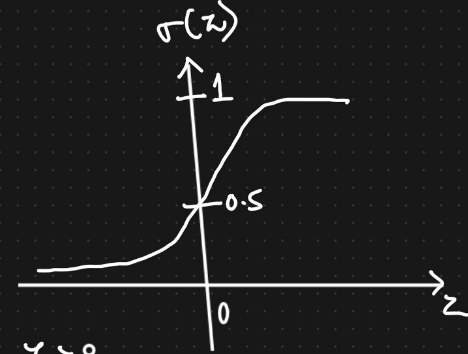
How Logistic Regression Solves Classification Problem



$$\boxed{h_\theta(x) = \theta_0 + \theta_1 x_1}$$

$\Downarrow$

Best fit line



Sigmoid Activation

$$\frac{1}{1+e^{-z}}$$

$z \geq 0$

$\sigma(z) \geq 0.5$

$z = \theta_0 + \theta_1 x_1$

$\Uparrow$

$$h_\theta(x) = \sigma(\theta_0 + \theta_1 x_1) \qquad \sigma = \frac{1}{1+e^{-z}}$$

$$= \sigma(z)$$

$$\boxed{h_\theta(x) = \frac{1}{1+e^{-z}}}$$

$\Rightarrow$ Logistic Regression hypothesis

$z = \theta_0 + \theta_1 x_1$

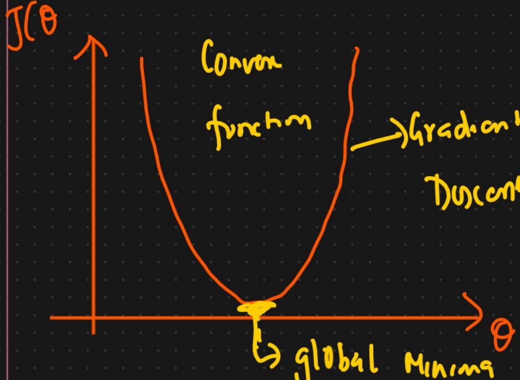
Linear Regression Cost function

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x)^{(i)} - y^{(i)}\right)^2$$

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

$\Downarrow$

Convex function



Logistic Regression Cost function

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x)^{(i)} - y^{(i)}\right)^2$$

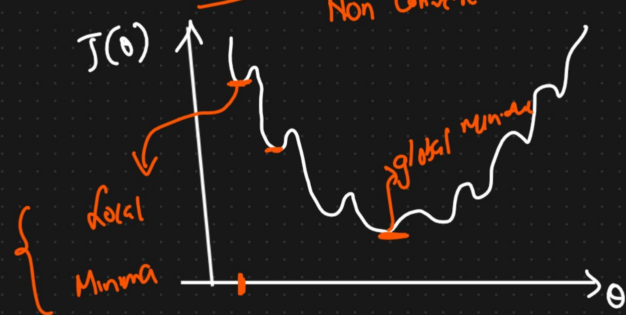
Non Convex function

$\Downarrow$

$$h_\theta(x) = \frac{1}{1+e^{-z}} \qquad z = \theta_0 + \theta_1 x_1$$

Sigmoid

Non Convex



$$J(\theta_0,\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \underbrace{\left(h_\theta(x)^{(i)} - y^{(i)}\right)^2}$$

$h_\theta(x)^{(i)} = \frac{1}{1+e^{-z}}$ $z = \theta_0 + \theta_1 x_1$

$\Downarrow$

Lets Denote $Cost(h_\theta(x)^{(i)}, y^{(i)})$

$\Downarrow$

{ Log Loss }

$$Cost(h_\theta(x)^{(i)}, y^{(i)}) = \begin{cases} -\log(h_\theta(x)) & \text{if } y=1 \\ -\log(1-h_\theta(x)) & \text{if } y=0 \end{cases}$$

Convex function

$\Downarrow$

$$Cost(h_\theta(x)^{(i)}, y^{(i)}) = -y \log(h_\theta(x)) - (1-y)\log(1-h_\theta(x))$$

$$\boxed{J(\theta_0,\theta_1) = -\frac{1}{2m} \sum_{i=1}^{m} \left(y^{(i)} \log(h_\theta(x)^{(i)}) - (1-y^{(i)}) \log(1-h_\theta(x)^{(i)}) \right)}$$

Minimize Cost function $J(\theta_0,\theta_1)$ by changing θ_0 & θ_1

Convergence Algorithm

Repeat

{ $j = 0$ and 1

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

}