

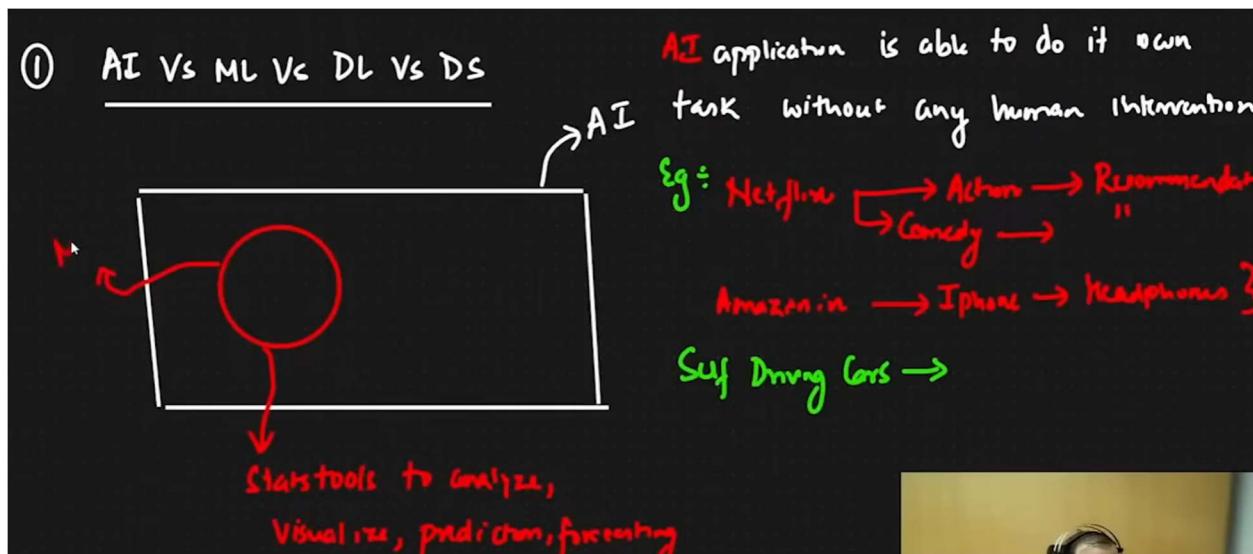
Table of Contents

AI Vs ML vs DL vs Data Science	2
Machine Learning and Deep Learning	3
Regression And Classification	4-6
Linear Regression Algorithm	6-14
Ridge And Lasso Regression Algorithms	14-19
Logistic Regression(classification) Algorithm.....	19-29
Linear Regression Practical Implementation.....	
Ridge And Lasso Regression Practical Implementation.....	
Naive Baye's Algorithms	30-
KNN Algorithm Intuition	
Decision Tree Classification Algorithms	
Decision Tree Regression Algorithms	
Practical Implementation Of Decision Tree Classifier.....	
Ensemble Bagging and Boosting Techniques.....	
Random Forest Classifier and Regressor.....	
Boosting, Adaboost Machine Learning Algorithms	
K Means Clustering Algorithm.....	
Hierarichal Clustering Algorithms	
Silhouette Clustering- Validating Clusters	
Dbscan Clustering Algorithms	
Clustering Practical Examples.....	
Bias And Variance Algorithms	
Xgboost Classifier Algorithms	
Xgboost Regressor Algorithms.....	
SVM Algorithm Machine LEarning Algorithm	

AI VS ML VS DL

Machine learning is basically subset of AI

- ML basically gives you stats tool to analyze visualizing , predicting and forecasting



- The goal is to create an AI application

MACHINE LEARNING AND DEEP LEARNING

Two algorithm

Supervised ML

Unsupervised ML

Supervised ML <ul style="list-style-type: none">• Regression Problem• Classification Problem	Unsupervised ML <ul style="list-style-type: none">• Clustering• Dimensionality reduction
---	---

Note: There is also one more type which is called reinforcement learning

Supervised ML Example of a data set :

Age	Weight
24	62
25	63
21	72
27	62



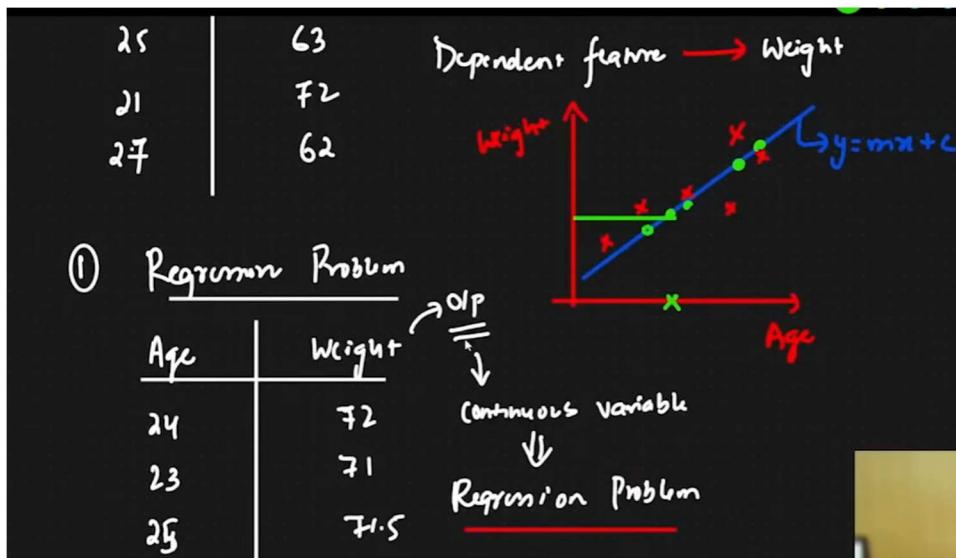
- INDEPENDENT FEATURE (
 - AGE

- DEPENDENT FEATURE
 - Weight

The value of weight is changing according to age so that's why they are categorized as dependent and independent

Regression vs Classification

Regression



- In regression there will be outputs of continuous variable

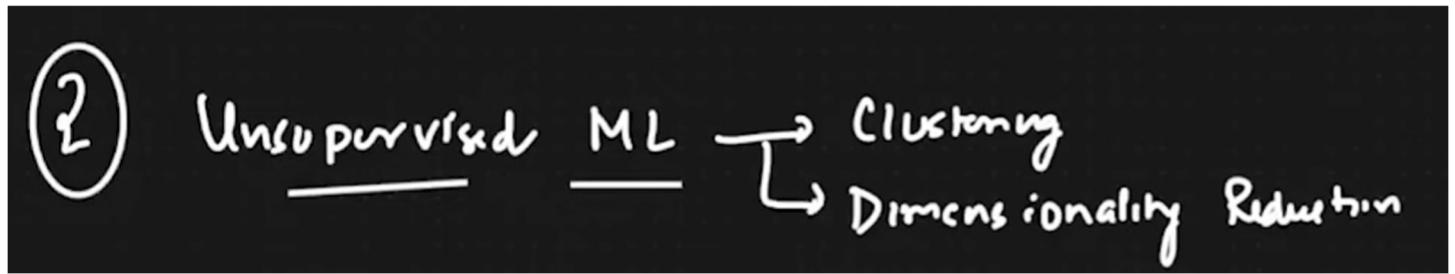
CLASSIFICATION

No of hrs	No of play hrs	No of sleep	Pass or Fail(dependent feature) AKA Putput
-	-	-	P
-	-	-	F

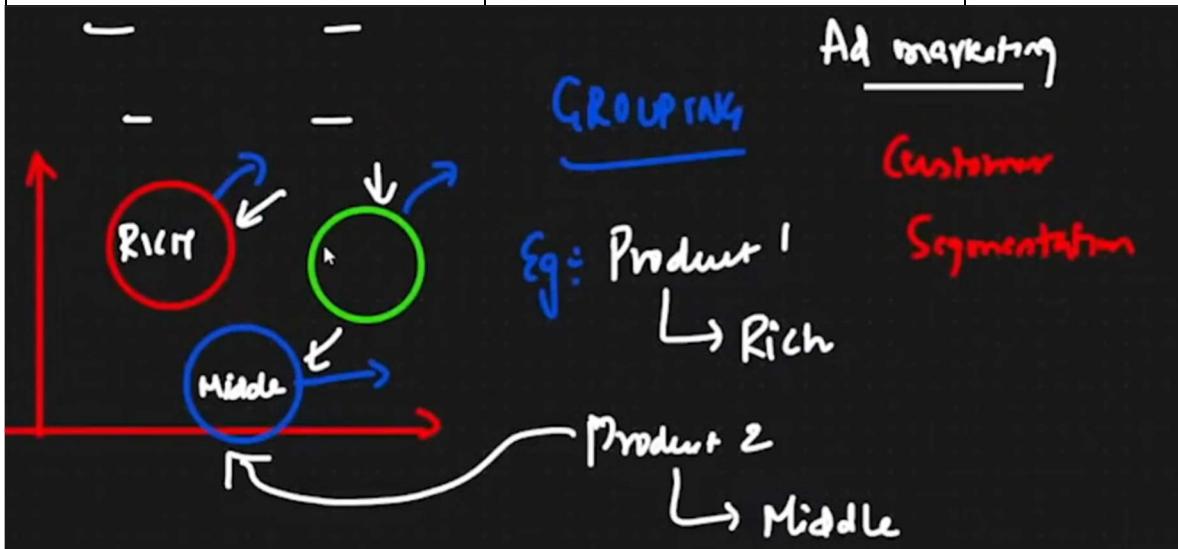
Classification : whenever you have fixed number of categorical output it becomes a classification output

Binary Classification: There are two categories of output Example P/F

Multi class classification: Multiple categories of output



Salary	Age	No dependent variable
-	-	
-	-	



- Clustering
 - Grouping
 - Example for Ad marketing : Product 1 for rich people product 2 for poor people etc

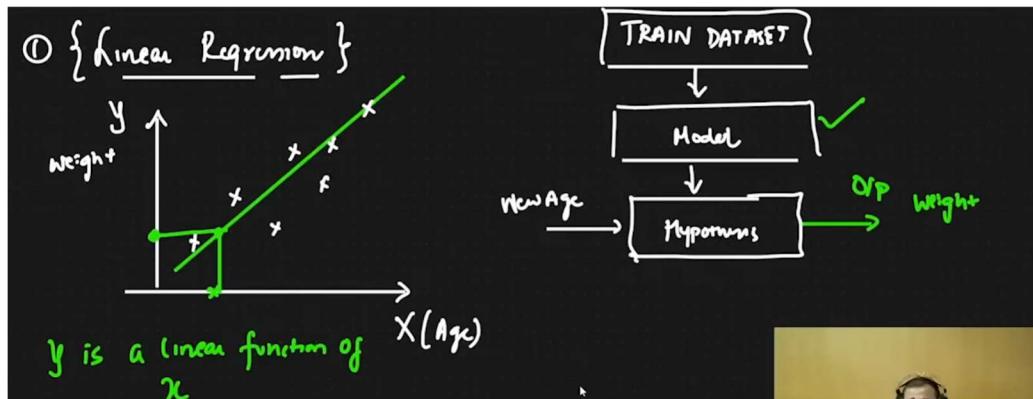
Dimensionality Reduction

Lower 100 feature of 100 feature examples of that is PCA and LDA

Supervised Learning Algorithms	Unsupervised Learning Algorithms
1. Linear Regression	1. K-Means
2. Ridge & Lasso	2. DBSCAN
3. Logistic Regression	3. Hierarchical Clustering
4. Decision Tree	4. K-Nearest Neighbor (Clustering)
5. AdaBoost	5. PCA (Principal Component Analysis)
6. Random Forest	6. LDA (Linear Discriminant Analysis)
7. Gradient Boosting	
8. XGBoost	
9. Naive Bayes	

Linear Regression

Find out the best fit line which will actually help in making prediction



Y is a linear function of X

The line equation could be given as:

$$Y = mx + c$$

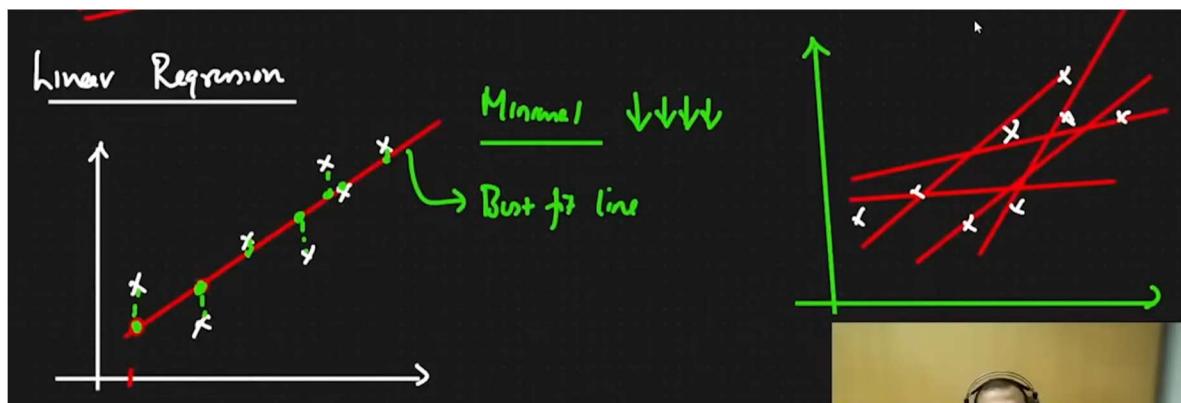
$$Y = B_0 + B_1 x$$

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

Etc

This is the best fit line eqn : θ_0 is the intercept when $x=0$

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



Our aim is to find the minimal cost line and to create a cost function The best red line possible

The distance in the submission should be minimal see the green lines

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

Cost function

Purpose $\{ \text{Derivation} \}$

$x^n = nx^{n-1}$

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

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

The entire equation is called as Squared error function, squaring is done to discard negative values

And m is the total number of points being compared, it is divided to get an average. It is divided by 2 for simplicity of mathematical equation

Time:30:00

What we need to actually solve

What we need to solve

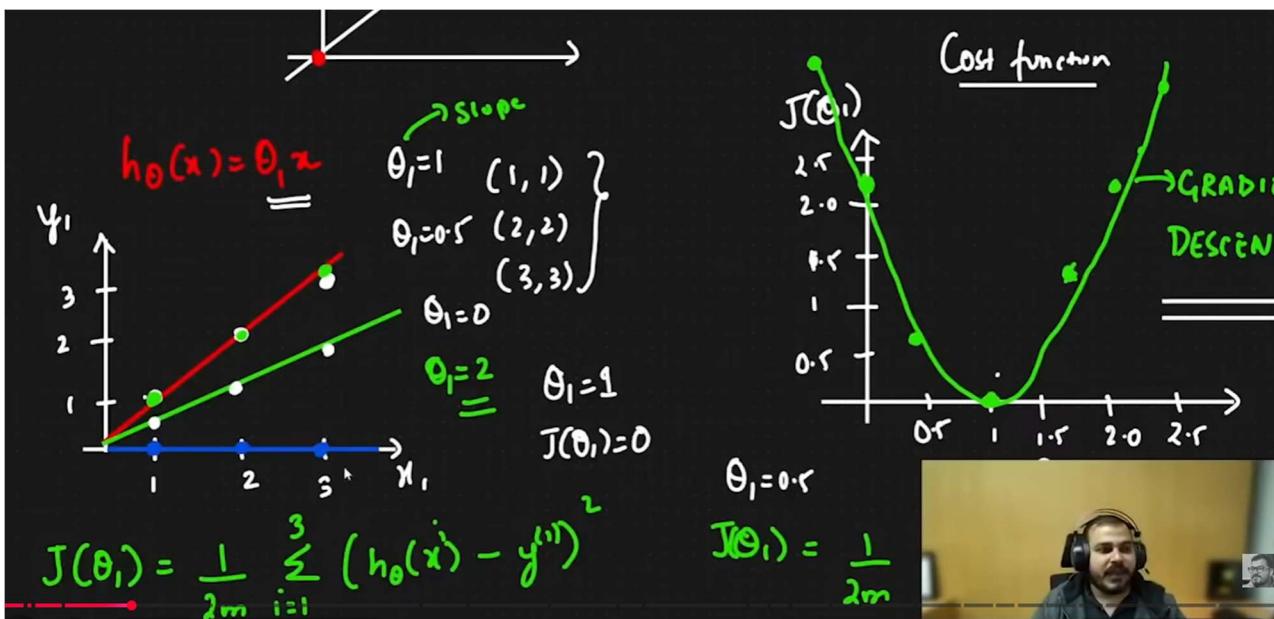
minimize θ_0, θ_1 $\frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$

↓

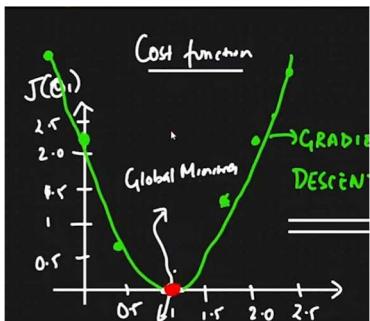
minimize θ_0, θ_1 $J(\theta_0, \theta_1)$

really need to minimize this so this is our task

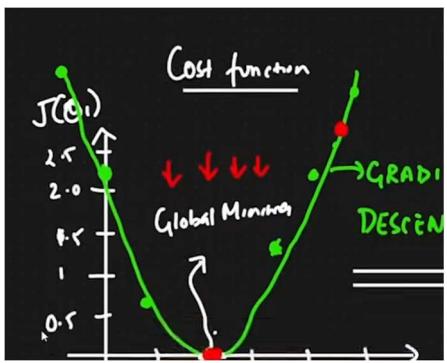
We need to make sure that we get the right theta 1 value



The best fit line represents global minimum



We need to try to find the red dot in the parabola that is in the upper section which can help us get to the global minimum



The Convergence Algorithm needs to be used to get to the

global minimum

Convergence Algorithm

Repeat until convergence derivative($\Sigma \hat{y}_i$)

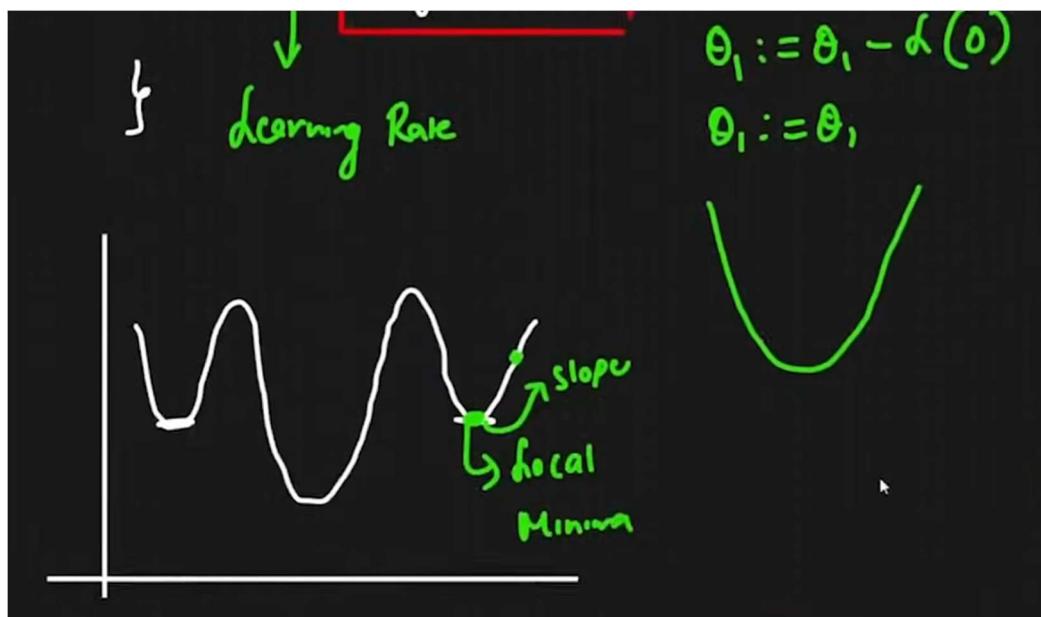
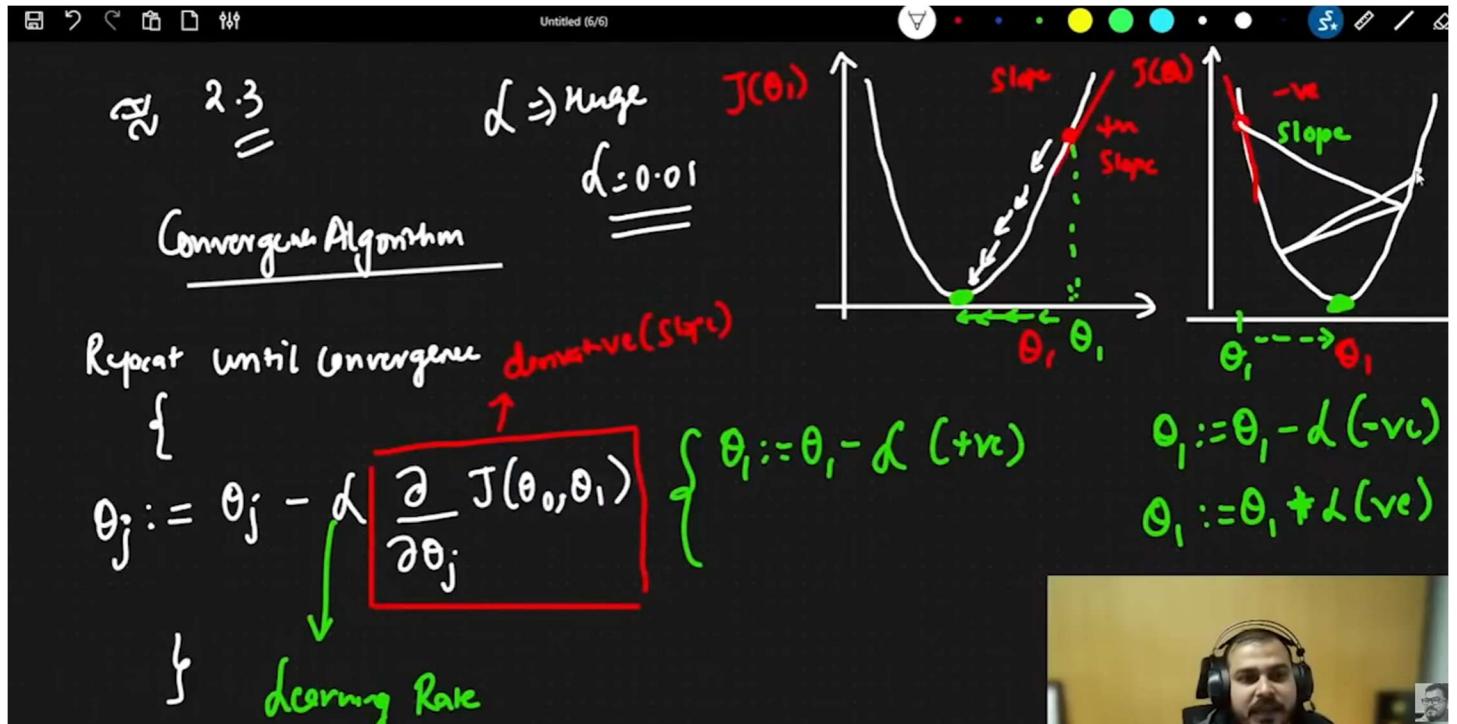
$$\left\{ \theta_j := \theta_j - \alpha \frac{\partial J(\theta_0, \theta_1)}{\partial \theta_j} \right.$$

↓

How does it find itself ?

if we find the point in right hand side of the parabola or the right side of the parabola the formula will adjust itself relatively and also if the point is on the left side it will adjust accordingly .

Also we need to choose our alpha very carefully so that the points never reach the global minimum



In some cases we can get the local minimum as the actual point, but this problem is not solved in linear regression and this could be solved using deep learning algorithms, Gradient descent and ANN in dl has lot of local minima adam optimizers are used in such cases.

Gradient decent algorithm

Repeat until convergence

Converge Algorithm

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

$\alpha = 0.001$

$\alpha = \text{learning Rate}$

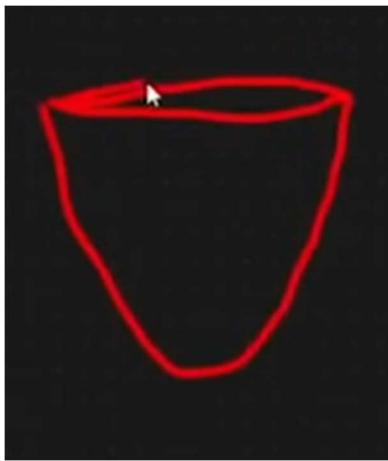
$\frac{\partial \theta_1}{\partial \theta_1} \quad \alpha = 0.001 \quad \alpha = \text{learning Rate}$

Repeat until converge

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

55:08

We will have lot of convex feature, at one point we will have a 3d curve: it will be coming down slowly just like you are descending from a mountain



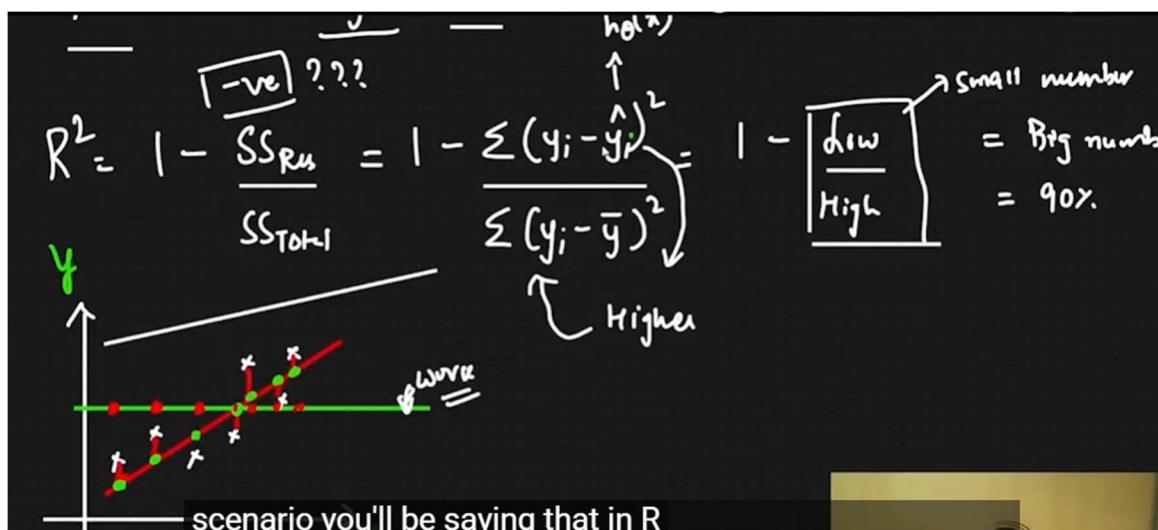
Two performance matrix

R^2 and adjusted R^2

$R^2 = 1 - \frac{\text{Sres}}{\text{Stotal}}$

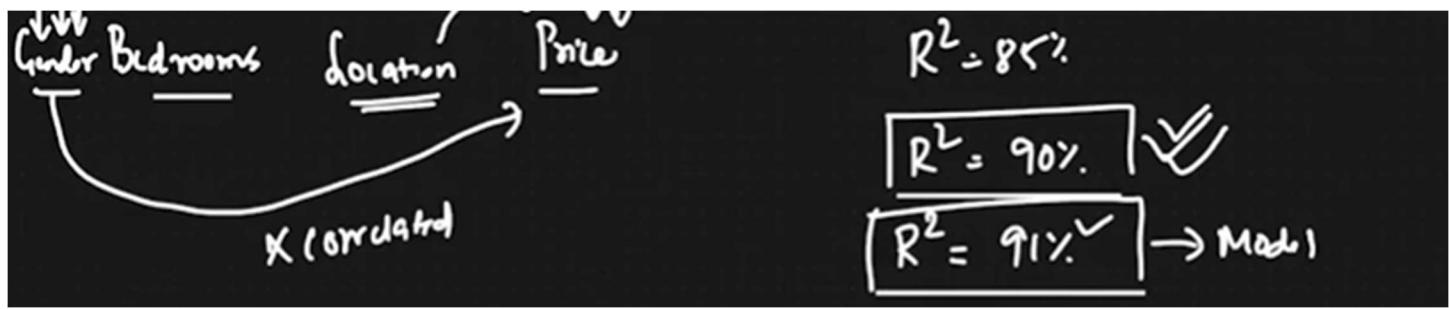
\hat{y} : green points y_i IS THE ACTUAL POINT

\bar{y} is the mean of y



If we keep on adding feature, then also there is a chance of increasing R^2 , so in order to not increase R^2 with irrelevant or related, so we need to fix that problem like the example here gender is not related at all with the pricing of the housing to increase or decrease, so we need to fix that issue with this corrected

formula



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

Note: N is the total number of samples, P is the number of features/predictors

When the features are correlated, the R^2 value will be greater than that of non correlated values

Adjusted R^2

$p = \text{features or predictors}$

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

$p=2 \quad = R^2 = 90\% \quad R^2_{\text{adjusted}} = 85\%$

$p=3 \quad = R^2 = 91\% \quad R^2_{\text{adjusted}} = 82\%$

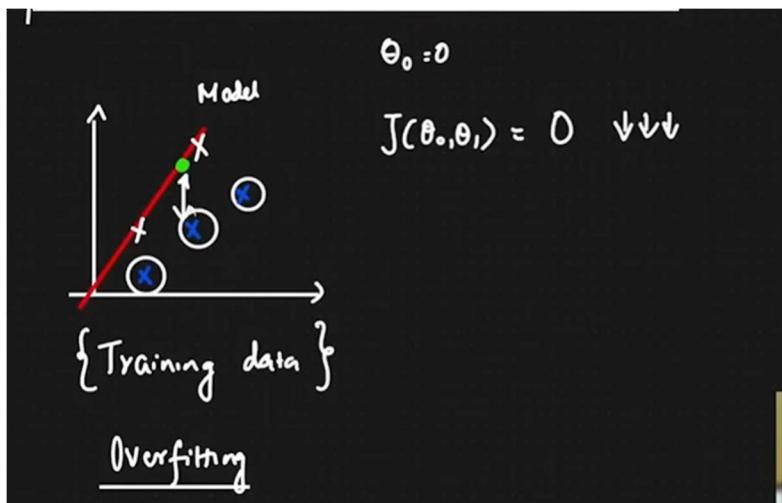
Interview Question: What is bigger R^2 or adjusted R^2

Ans: We know that R^2 will be bigger because R^2 does not take consideration of the correlation of the feature/Predictor

- Ridge And Lasso Regression
- Assumption of Linear Regression
- Logistic Regression
- Confusion Matrix
- Practicals for Linear Ridge Lasso and Logistic Regression

Ridge And Lasso Regression

Even though my model has trained well with the training data, it causes overfitting, which means my model performs well with training data and fails to perform well with the test data



Overfitting

Model performs well \rightarrow Training data

Fails to perform well \rightarrow Test Data

Overfitting

- When the model performs well with training data: Low Bias
- When it fails with the test data: High variance

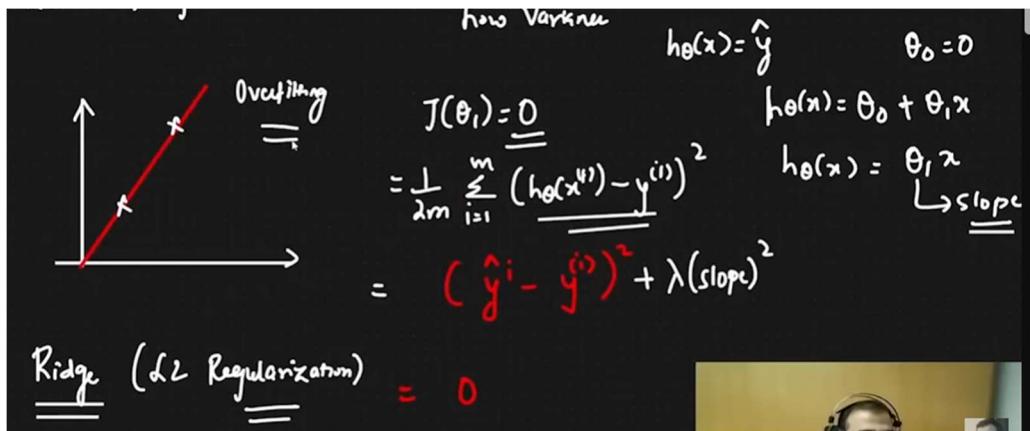
Underfitting

- Model accuracy is bad with training data: (High Bias)
- Model accuracy is also bad with testing data: (High Variance)

So here is an example of different scenarios :

Model 1	Model L	Model >
Training Acc = 90%.	Training Acc = 92%.	Training Acc = 70%.
Test Acc = 80%.	Test Acc = 91%.	Test Acc = 65%
\Downarrow Overfitting	\Downarrow Generalized Model	\Downarrow Underfitting

IN our example we used the equation and we tried to predict the J with it, and it has caused an overfitting condition, so to change that 0 into something else we need to use Ridge regression also known as the L2 regularization

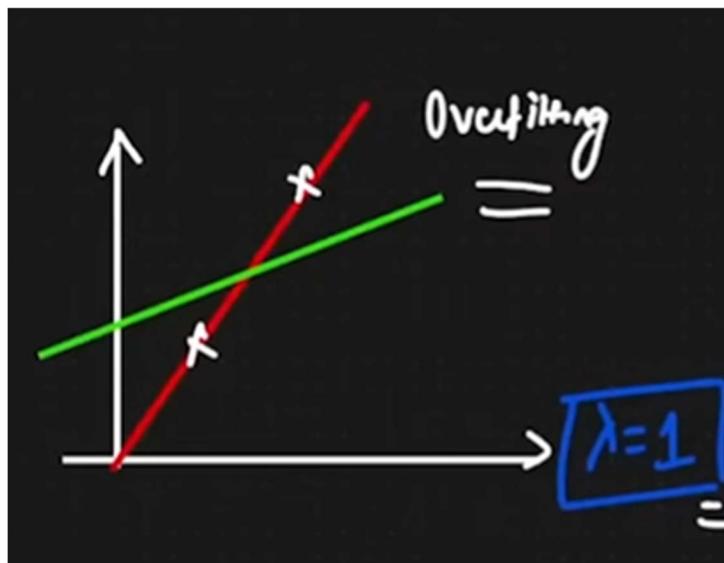


Let us consider our lamda as 1 and consider slope as 2 and the result becomes 3(Lamda is the hyper parameter)

$$\underline{\text{Ridge}} \quad (\underline{\text{L2 Regularization}}) = 0 + 1(2) = 3w$$

So due to this it will try to again change the best fit line to something else, and therefore it will change the theta 1 value

Now to change that we will again change the best fit line equation to something else :



Let us consider the value of slope to be changed from 2 to 1.5

This is why we are adding ridge L2 regularization, as the value is decreasing:

$$\begin{aligned}
 &= (y^{(i)} - \hat{y}^{(i)})^2 + \lambda (\text{slope})^2 \\
 &\quad \Downarrow \\
 &= (\text{Small value}) + 1(1.5)^2 \\
 &= (\text{Small value}) + 2.25 \\
 &\approx 3 \downarrow \downarrow
 \end{aligned}$$

- This is used to prevent overfitting
- The steep should not be steep, and it should help us create a generalized value
- Now we might have to specify the iterations{Hyperparameter}
- We can not get 0 because it could be an overfitting model

Lamda: How fast we can grow or lessen the steepness, this is changed by checking the hyperparameter, the examples will be shown later in the notes.

Lasso Regularizartion = $(\hat{y} - y)^2 + \lambda |\text{slope}|$

- Mod of slope will help us do feature selection

$$\text{Lasso (f1 Regularization)} = (\hat{y} - y)^2 + \lambda |\text{slope}|$$

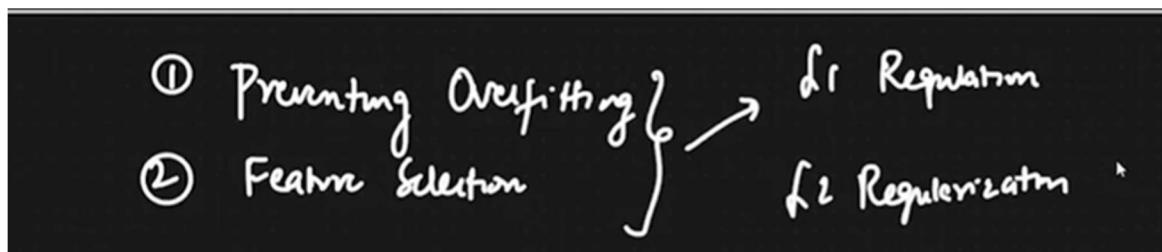
$h_{\theta}(x) = \hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 + \dots + \theta_n x_n$

$$\text{Lasso (f1 Regularization)} = (\hat{y} - y)^2 + \lambda |\text{slope}|$$

$h_{\theta}(x) = \hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 + \dots + \theta_n x_n$

| $\theta_0 + \theta_1 + \theta_2 + \theta_3 + \theta_4 + \theta_5 + \dots + \theta_n$ |

We will try to remove those feature which are not at all useful in this problem statement



Lamda basically means cross validation, cross validation is the technique where we will try to train our model

- In short we are trying to reduce the cost function on the basis of lamda and slope value

We should try both algorithms and see which has the best regressuin funcgtion with different cost function

- Ridge regression prevents overfitting

Ridge Regression (L2 Norm)

$$\text{Cost function} = (h_{\theta}(x^{(i)}) - y^{(i)})^2 + \lambda (\text{slope})^2$$

④ Purpose : Preventing Overfitting

- Lasso regression prevents users from using unnecessary features/ help in feature selection

Lasso Regression (L1 Reg)

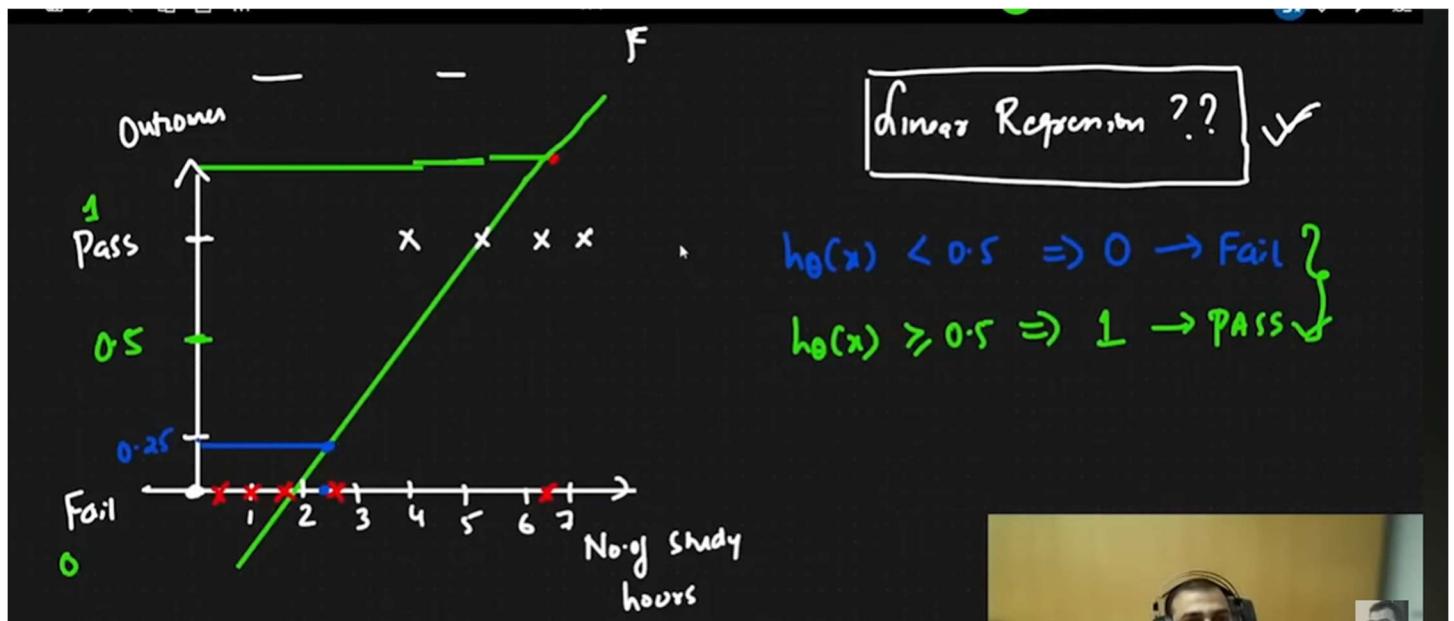
$$\text{Cost function} = (h_{\theta}(x^{(i)}) - y^{(i)})^2 + \lambda |\text{slope}|$$

Assumption of Linear Regression



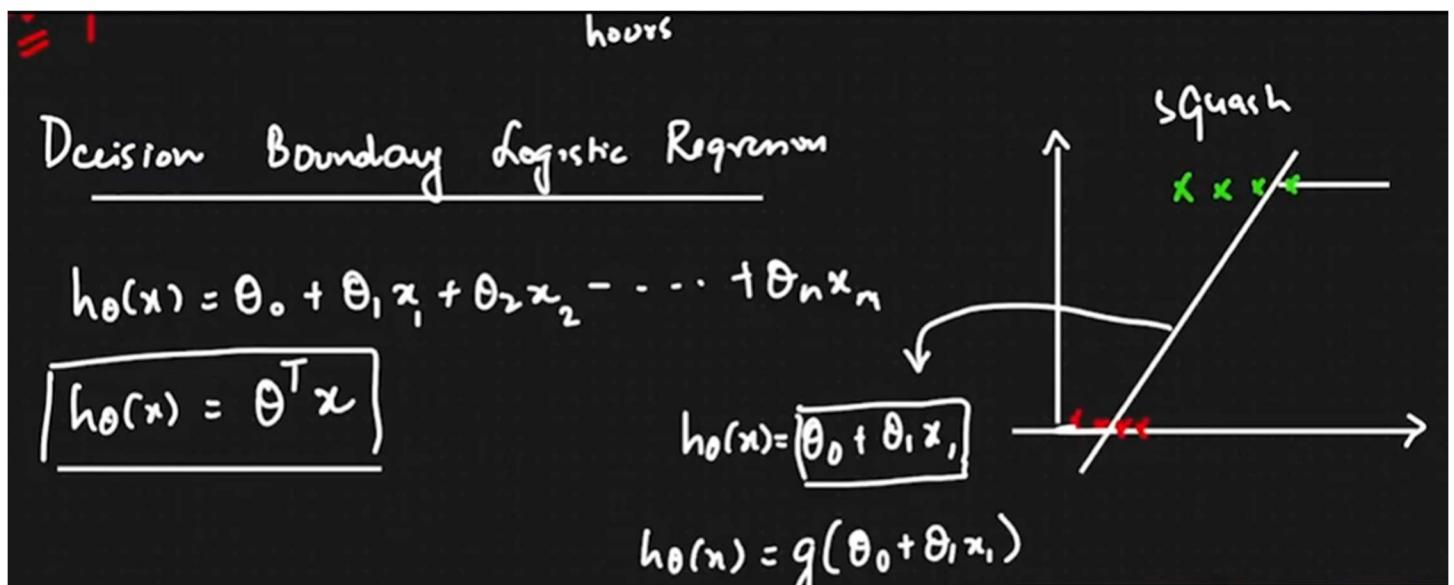
- ① Normal / Gaussian Distribution → Model will get trained well
- ② {Standardization} {Scaling data} → Z-score $\mu=0, \sigma=1$
- ③ Linearity X_3 $\boxed{X_1 \quad X_2}$ \boxed{Y} Variation Inflation factor?
- ④ Multi Collinearity

LOGISTIC REGRESSION is the first example for classification -> works very well with Binary Classification



To pass >3(basic logic)

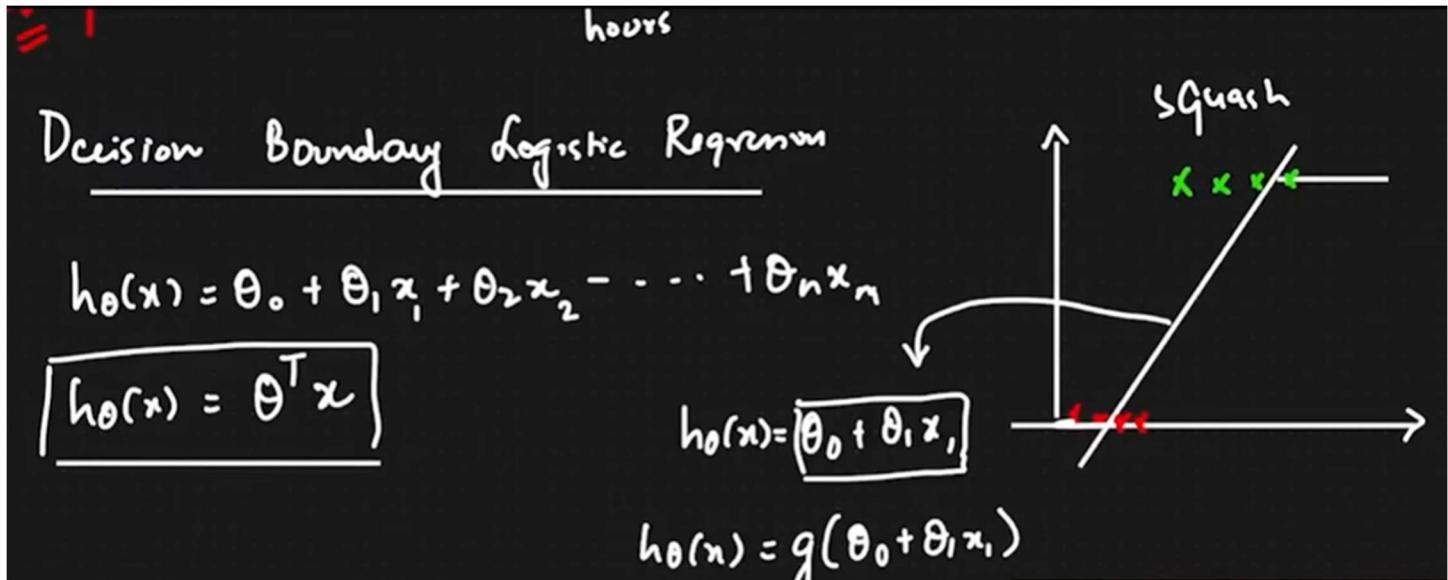
- Why not use linear regression then ?
 - With outlier the entire bestfit line would change, now for even 5 hrs study some people could fail according to this new model
- We could have to squash the functionality and this would be not good. Sepcially for negative inputs



BINARY CLASSIFICATION

- Our output value should always be 0 or 1

Decision Boundary Logistic regression



We will try to apply a mathematical formula which would squash this line, it is also called sigmoid/Logistic activation function

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

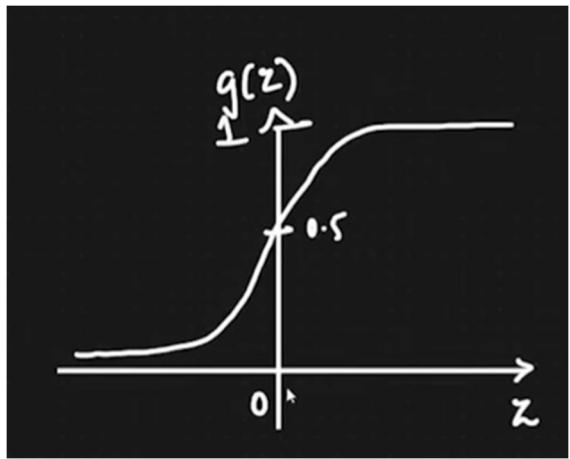
Let $z = \theta_0 + \theta_1 x_1$

$$h_{\theta}(x) = g(z)$$

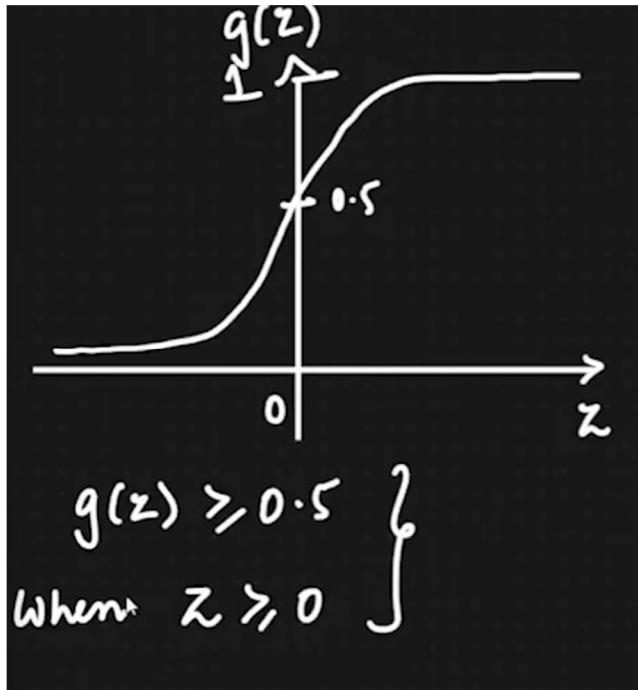
$$h_{\theta}(x) = \frac{1}{1 + e^{-z}}$$

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

Now this will be able to squash the regular line to desired value / this function is also known as logistic or sigmoid function and this will look something like



So we can make the assumption that



Training Set

$$\{(x^1, y^1), (x^2, y^2), (x^3, y^3), \dots, (x^n, y^n)\}$$

$$y \in \{0, 1\} \rightarrow 2 \text{ o/p}$$

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

$$z = \theta_0 + \theta_1 x$$

Change parameter theta 1 to get the best fit line and the sigmoid activation formula

Cost function =

Linear Regression $J(\theta_0) = \frac{1}{m} \sum_{i=1}^m \frac{1}{2} (h_{\theta}(x^{(i)}) - y^{(i)})^2$

Logistic Regression

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

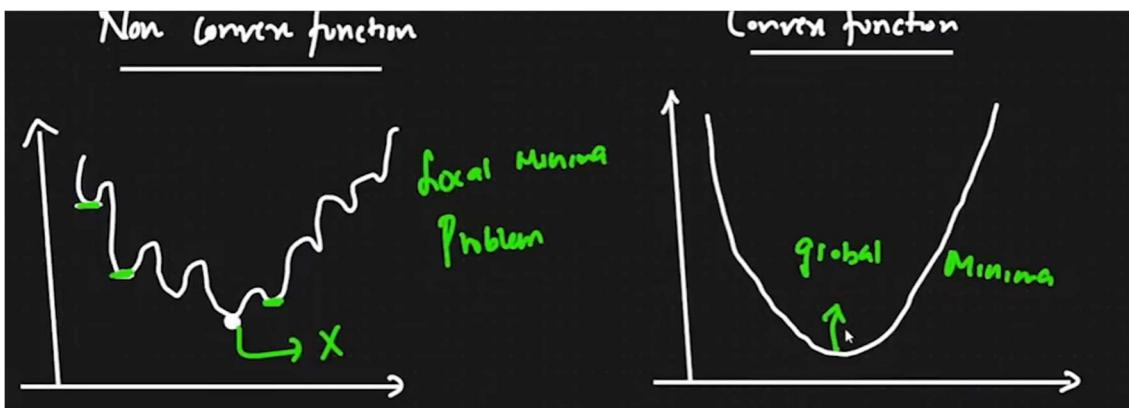
Logistic Regr

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

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

This can not be used because it is a non convex function:

Non Convex Function	Convex Function
Lots of local minima	No local minimas
Never reach global minima	Reach global Minima



Logistic regression cost function:

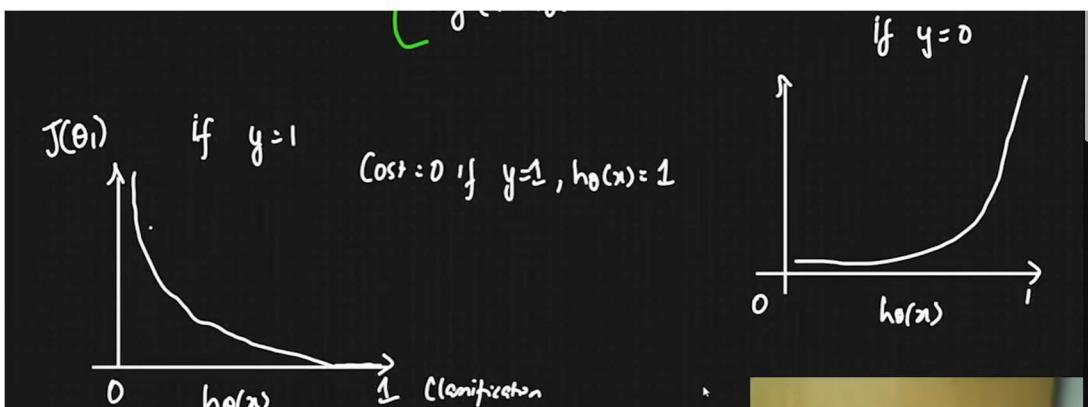
New type of CF is suggested=

Logistic Regression Cost function

$$J(\theta_0) = \begin{cases} -\log(h_{\theta}(x^i)) & y=1 \\ -\log(1-h_{\theta}(x^i)) & y=0 \end{cases}$$

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

So if $y=1$, $h(\theta)=1$ then the left side graph is made and the right side graph is for when $y=0$



Now if we combine both we will get a convex function and the final cost function will be :

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

If we replace y with 1 then you will only get a particular value :

If y will be 0 then the following function will be executed:

$$\begin{aligned} \widehat{\text{Cost}}(h_\theta(x^i), y) &= -y \log(h_\theta(x^i)) - (1-y) \log(1-h_\theta(x^i)) \\ \text{if } y=1 &\quad \downarrow \text{cost function.} \\ \text{Cost}(h_\theta(x^i), y) &= -\log(h_\theta(x^i)) \\ \text{if } y=0 & \\ \text{Cost}(h_\theta(x^i), y) &= -\log(1-h_\theta(x^i)) \end{aligned}$$

$$\begin{aligned} J(\theta_0) &= \frac{1}{2m} \sum_{i=1}^m (y^i \log(h_\theta(x^i)) + (1-y^i) \log(1-h_\theta(x^i))) \\ \downarrow \text{cost} \quad h_\theta(x^i) &= \frac{1}{1+e^{-\theta_0 \cdot x}} \\ \rightarrow \quad \text{repeat until convergence} \quad & \\ \left\{ \quad \theta_j := \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j} \right. \end{aligned}$$

Performance Measure

TP represents true positive

TN represents true negative

FP represents False Positive

FN represents False Negative

Performance Metrics {Classification problem}			
		{Actual}	
		Pred	{Actual}
x_1	x_2	y_{Actual}	y_{Pred}
-	-	0	1
-	-	1	1
-	-	0	0
-	-	1	1
-	-	0	1
-	-	1	0
		Confusion matrix	
		Predicted	Actual
		1	0
		0	1
		TP	FP
		FN	TN

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

$0 \rightarrow 900$	$1 \rightarrow 100$	Imbalanced Data
$0 \rightarrow 600$	$1 \rightarrow 400$	Balanced Data
$\left\{ \begin{array}{l} \text{Model} \rightarrow 0 = \frac{900}{1000} = 90\% \\ \text{Model} \rightarrow 1 = \frac{100}{1000} = 10\% \end{array} \right\}$		

If there is imbalanced data set then there will be problems in the accuracy and the algorithm, so we can use

- **Precision**
 - $TP / (TP + FP)$
- **Recall:** Out of all true positives how many of them have predicted positive, now we try to reduce False negatives

		1	0
Pred	1	TP	FP
	0	FN	TN
		↓↓↓	

$$\text{RECALL} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Precision means False positive is given more priority to be reduced

		1	0	Actual
Pred	1	TP	FP	↑
	0	FN	TN	
		↓↓↓		

{ Spam classification } → Precision

{ Has CANCER OR NOT } → Recall

F-Beta/F-score(3 different formula) The generic one is also termed as Harmonic mean

$$\underline{\underline{F-\text{Beta}}} = (1 + \beta^2) \frac{\text{Precision} \times \text{Recall}}{\beta^2 \times \text{Precision} + \text{Recall}}$$

For the generic case we have the following explanation :

$$\begin{aligned}\beta = 1 &= (1+1) \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \\ &= \frac{2(\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}} \\ \text{Harmonic Mean } &\uparrow \\ &= \frac{2xy}{x+y}\end{aligned}$$

Lets say that the false positive is more important than false negative, so we consider the B as 0.5, so decreasing means we are creating more importance on False positive than false negative:

F0.5 score is used

$$\beta = 0.5 \quad (1 + (0.5)^2) \frac{P \times R}{(0.25) P + R}$$

Now if we do the other thing it becomes the **F2 score**

$$\beta = 2 \quad FN \gg FP .$$

F2 Score

DAY3: Please look at the .pynb file in github from time 2:35 continue with Lasso regression to see the results

Agenda:

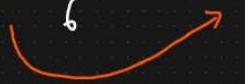
Naive Bayes Algorithm (Classification)

↳ Probability and Bayes's Theorem

Independent Event

Rolling a Dice $\{1, 2, 3, 4, 5, 6\}$

$$P(1) = \frac{1}{6} \quad P(2) = \frac{1}{6} \quad P(3) = \frac{1}{6}$$



Dependent Event



What is the probability of removing
a white marble and then a
yellow marble?

$$\hookrightarrow P(w) = \frac{3}{5} \rightarrow 1^{\text{st}} \text{ Event}$$



$$P(y/w) = \frac{2}{4} = \frac{1}{2} \rightarrow 2^{\text{nd}} \text{ Event}$$

↳ conditional probability

$$P(w \text{ and } y) = P(w) * P(y/w) \Rightarrow \text{Independent Event}$$

↑ ↑

↳ Conditional

probability

Baye's Theorem

$$P(A \text{ and } B) = P(B \text{ and } A)$$

$$P(A) * P(B/A) = P(B) * P(B/A)$$

$$\boxed{P(B/A) = \frac{P(B) * P(B/A)}{P(A)}} \Rightarrow \text{Baye's Theorem}$$

Data set

Independent feature :-

$$x_1 \ x_2 \ x_3 \ x_4 \dots x_n \quad y$$

Dep

$$P(y/x_1, x_2, x_3, \dots, x_n) = \frac{P(y) * P(x_1, x_2, x_3, x_4, \dots, x_n/y)}{P(x_1, x_2, x_3, \dots, x_n)}$$

$$= \frac{P(y) * P(x_1/y) * P(x_2/y) * P(x_3/y) \dots * P(x_n/y)}{P(x_1, x_2, x_3, \dots, x_n)}$$

x_1	x_2	x_3	x_4	y
-	-	-	-	Yes
-	-	-	-	No
-	-	-	-	Yes

$$\left\{ \begin{array}{l} P(\text{Yes}/x_1, x_2, x_3, x_4) = \frac{P(y_{\text{Yes}}) * P(x_1/y_{\text{Yes}}) * P(x_2/y_{\text{Yes}}) * P(x_3/y_{\text{Yes}}) * P(x_4/y_{\text{Yes}})}{\text{Constant } \leftarrow P(x_1) * P(x_2) * P(x_3) * P(x_4) \times} \\ P(\text{No}/x_1, x_2, x_3, x_4) = \frac{P(\text{No}) * P(x_1/\text{No}) * P(x_2/\text{No}) * P(x_3/\text{No}) * P(x_4/\text{No})}{\text{Constant } \leftarrow P(x_1) * P(x_2) * P(x_3) * P(x_4) \times} \end{array} \right.$$

Let's Solve this Problem

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Outlook

Sunny	Yes	No	$P(E Yes)$	$P(E No)$
↓	2	3	$\frac{2}{9}$	$\frac{3}{5}$
↓ Overcast	4	0	$\frac{4}{9}$	0
↓ Rain	3	2	$\frac{3}{9}$	$\frac{4}{5}$

Temperature → Test (Sunny, Hot) → Play (y)

	Yes	No	$P(E Yes)$	$P(E No)$		$P(Yes)$	$P(No)$
Hot	2	2	$\frac{2}{9}$	$\frac{4}{5}$	Yes 9	$\frac{9}{14}$	$\frac{5}{14}$
Mild	4	2	$\frac{4}{9}$	$\frac{2}{5}$	No 5		
Cool	3	1	$\frac{3}{9}$	$\frac{1}{5}$			

$$P(Yes | (Sunny, Hot)) = \frac{P(Yes) * P(Sunny|Yes) * P(Hot|Yes)}{P(Sunny) + P(Hot)}$$

$$= \frac{9}{14} * \frac{2}{9} * \frac{2}{9}$$

$$= \frac{2}{63} = 0.031$$

$$P(No | (Sunny, Hot)) = \frac{P(No) * P(Sunny|No) * P(Hot|No)}{P(Sunny) + P(Hot)}$$

$$X \leftarrow P(Sunny) + P(Hot)$$

$$= \frac{5}{14} * \frac{3}{9} * \frac{2}{5}$$

$$= \frac{3}{35} = 0.085$$

$$P(\text{Yes} | (\text{Sunny}, \text{hot})) = \frac{0.031}{0.031 + 0.085} = 0.27 = 27\%$$

$$P(\text{No} | (\text{Sunny}, \text{hot})) = \frac{0.085}{0.031 + 0.085} = 0.73 = 73\%$$

Test data (\downarrow Sunny, \downarrow hot) = 73%. They will not play Tennis

\Rightarrow Person is not going
to play Tennis
==

QUESTIONS:

TOP 10 MOST COMMON INTERVIEW QUESTIONS

1. Tell us a bit about yourself?
2. What are your strengths & weaknesses?
3. Why do you want to work here?
4. Where do you see yourself in 5 years?
5. Why should we hire you?
6. Challenge at work & how you dealt with it?
7. Greatest professional achievement?
8. Why are you leaving your current job?
9. How do you handle stress & pressure?
10. Do you have any questions for us?

global minimum

Convergence Algorithm

Repeat until convergence derivative(5%)

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

How does it find itself ?

If we find the point in right hand side of the parabola or the right side of the parabola the formula will adjust itself relatively and also if the point is on the left side it will adjust accordingly.

Questions Like how is Logistic Regression different from linear regression ?

