

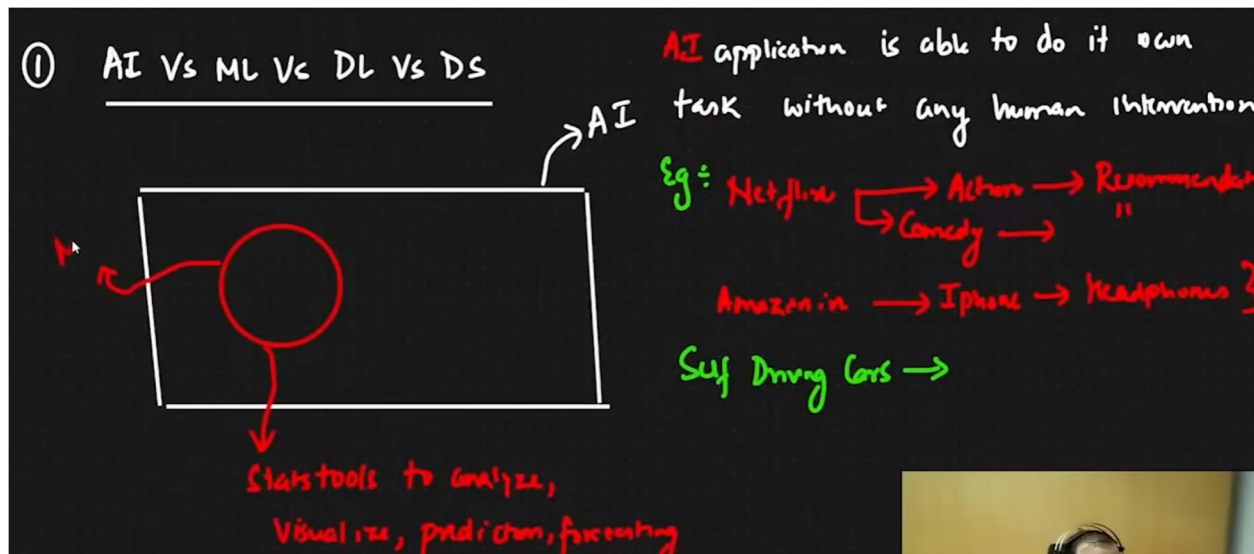
Table of Contents

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.....	
Linear Regression Practical Implementation.....	
Ridge And Lasso Regression Practical Implementation.....	
Naive Baye's Algorithms	
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	
Silhoutte 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

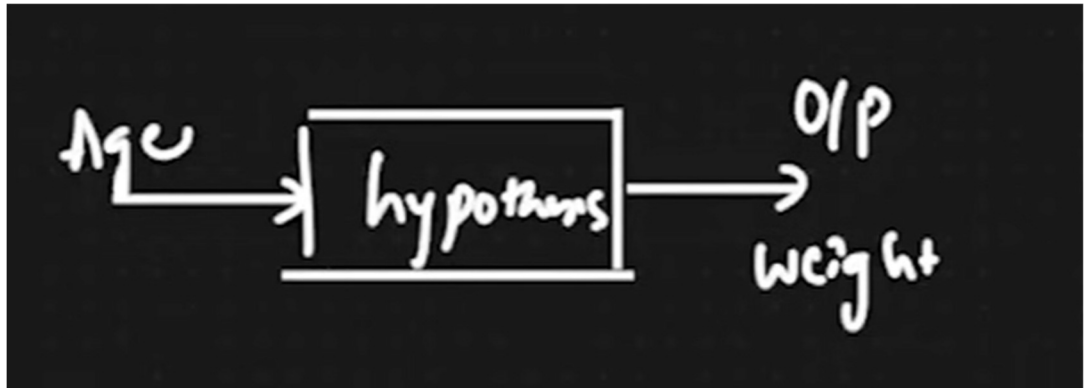
Unsupervised ML

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

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

Supervised ML Example of a data set :

Age	Weight
24	62
25	63
21	72
27	62



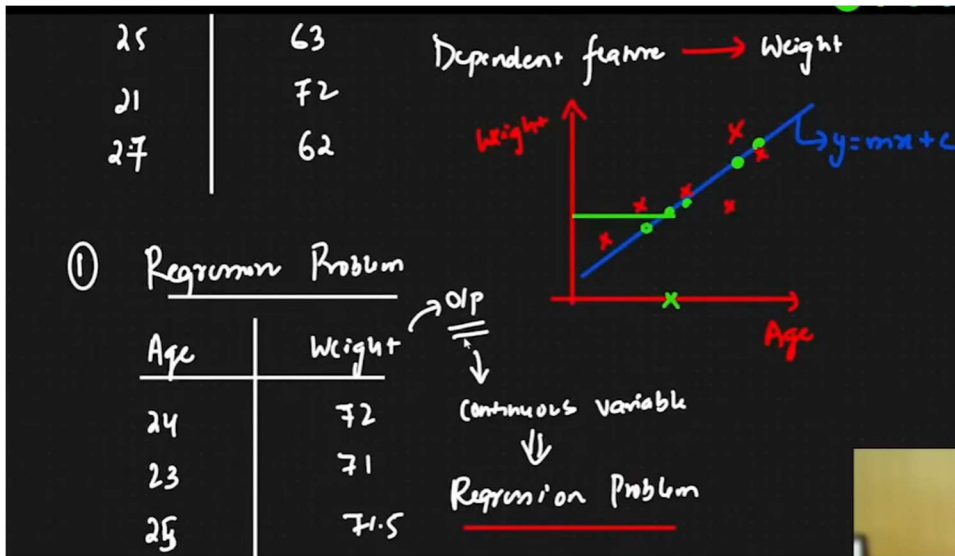
- INDEPENDENT FEATURE (
 - AGE

- DEPENDENT GEATURE
 - 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

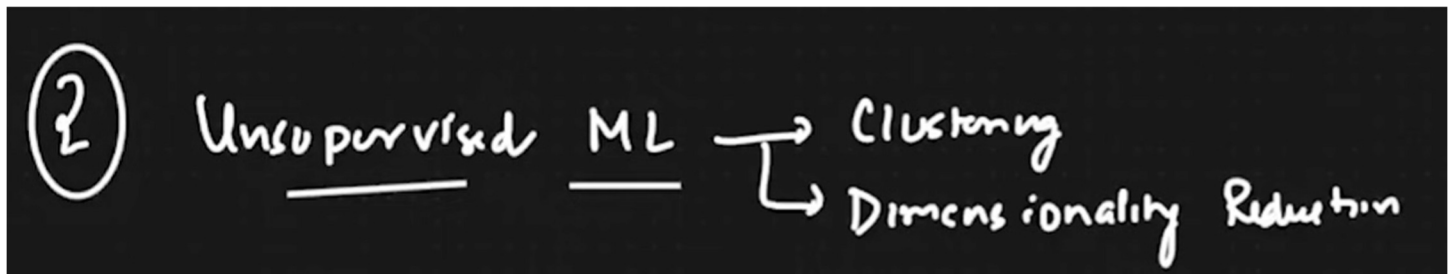
CLASSIFICATION

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

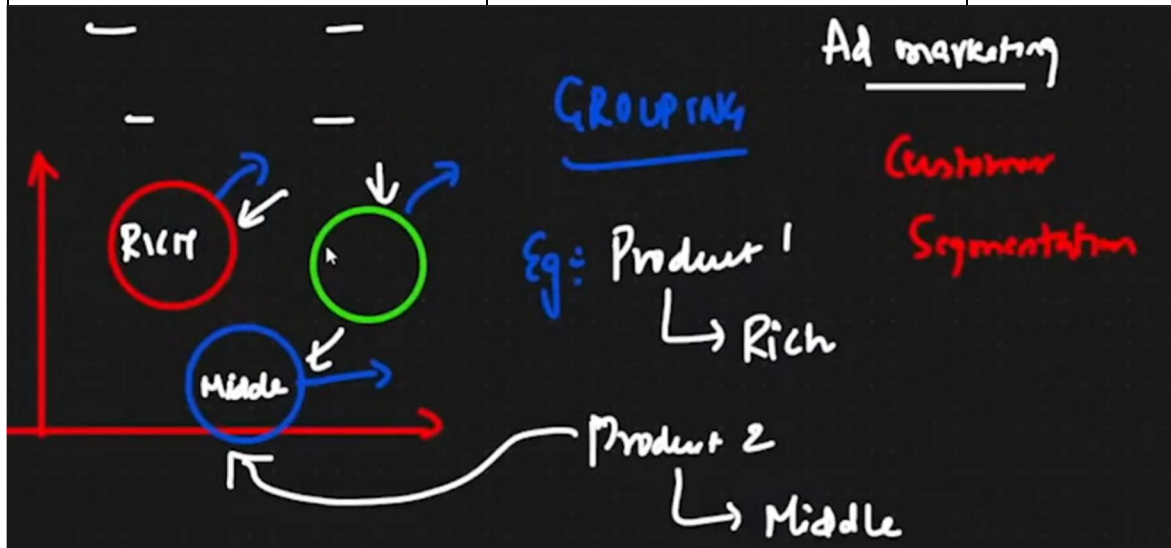
Classification : whenever you have fixed number of categorical output it becomes a classification output

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

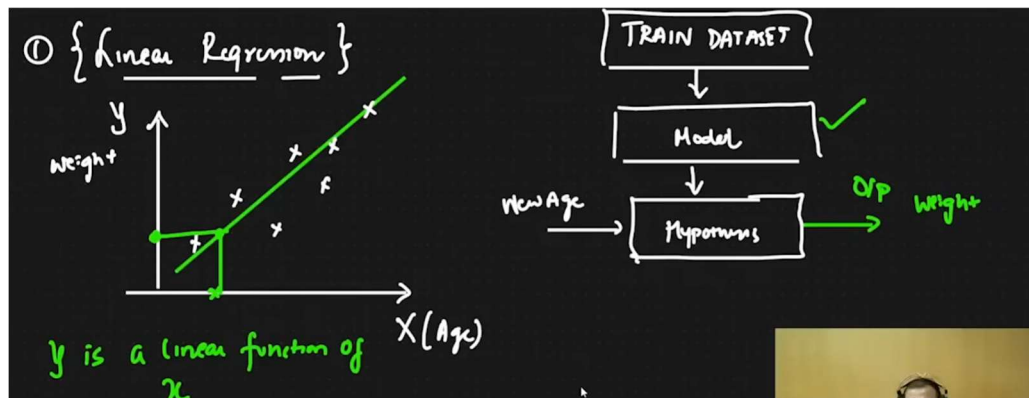
Dimensionality Reduction

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:

$$Y = mx + c$$

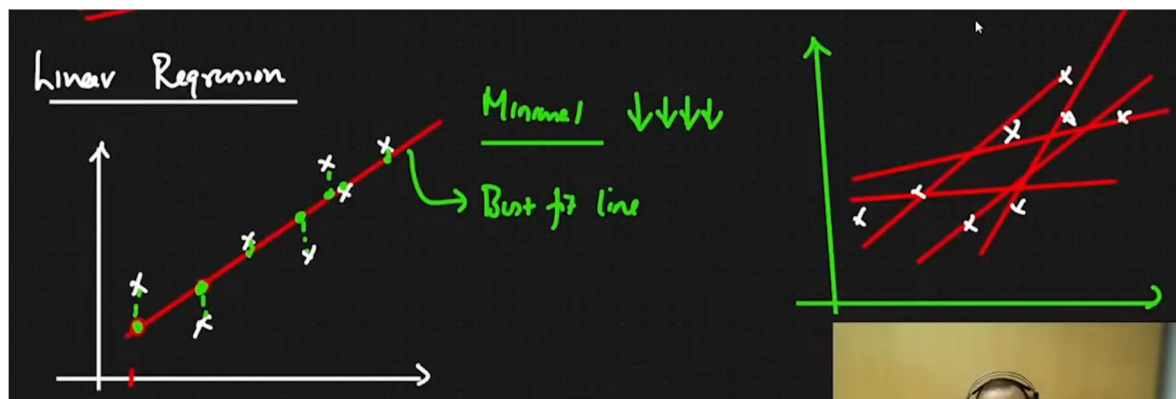
$$Y = B_0 + B_1x$$

$$H(x) = \theta_0 + \theta_1x$$

Etc

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

$$h_{\theta}(x) = \theta_0 + \theta_1x$$



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 $h_0(x) = \theta_0 + \theta_1 x$

Cost function

$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (h_0(x^{(i)}) - y^{(i)})^2$ → Cost function

Derivation

Purpose

$\frac{d(x^2)}{dx} = 2x$

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

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 $\frac{1}{2m} \sum_{i=1}^m (h_0(x^{(i)}) - y^{(i)})^2$

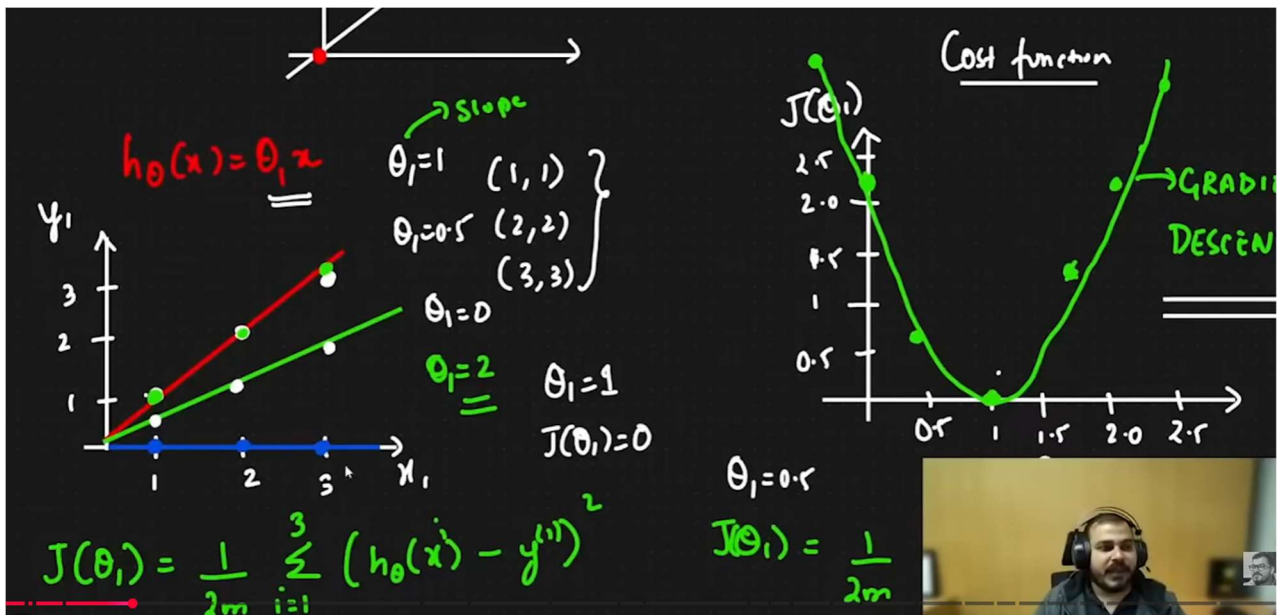
↓

minimize $J(\theta_0, \theta_1)$

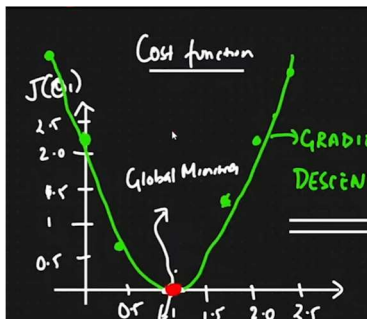
θ_0, θ_1

really need to minimize this so this is our task

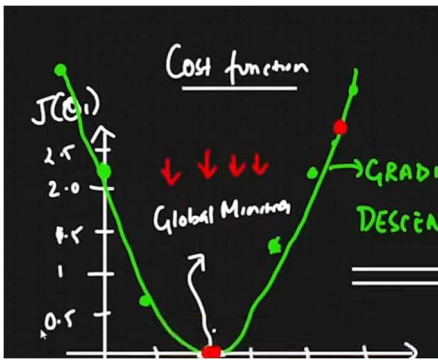
We need to make sure that we get the right theta 1 value



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

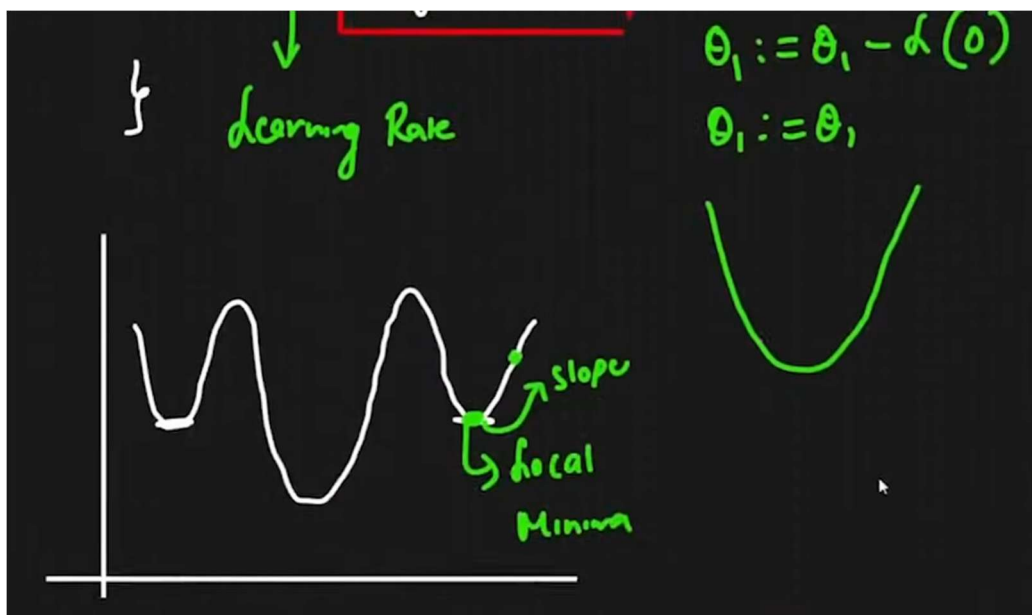
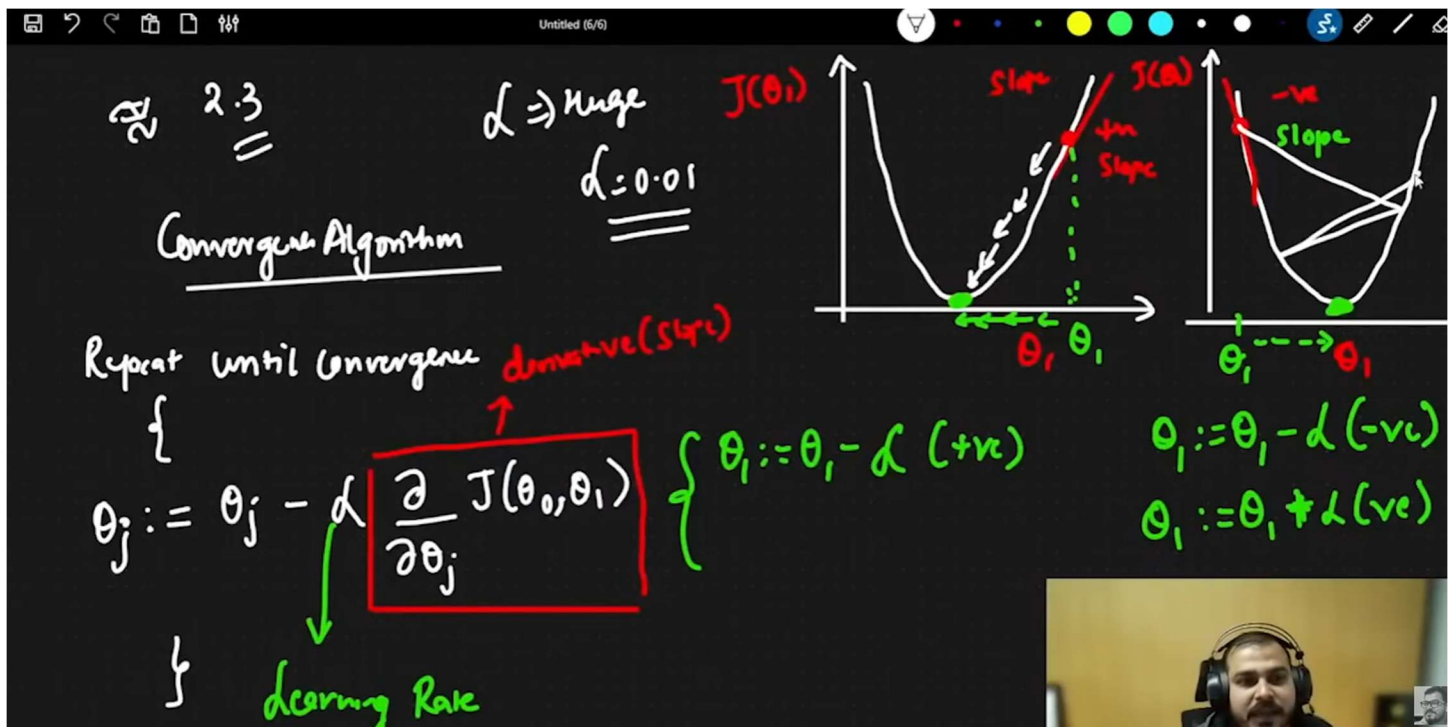
$$\theta_j := \theta_j - \alpha \frac{\partial J(\theta_0, \theta_1)}{\partial \theta_j}$$

derivative (slope)

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

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

$h_{\theta}(x) = \theta_0 + \theta_1 x$
 $\frac{x^2}{2x}$

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

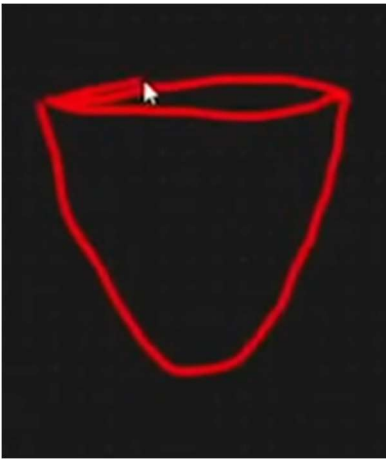
$\{d=0.001\}$ $d = \text{learning Rate}$

Repeat until converge

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

55:08

We will have lot of 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



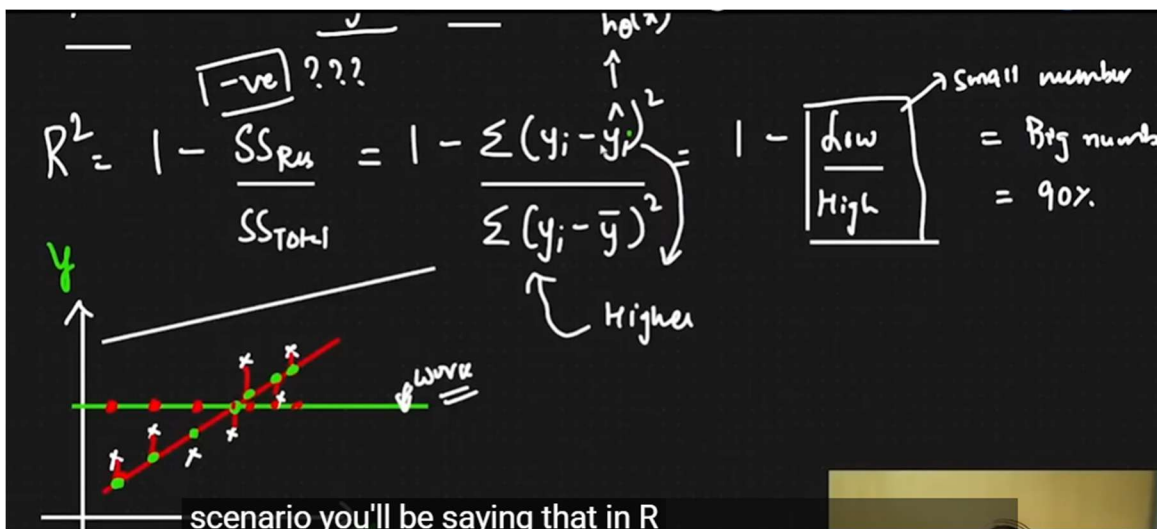
Two performance matrix

R² and adjusted R²

$$R^2 = 1 - \frac{S_{res}}{S_{total}}$$

Y-hat : green points YI IS THE ACTUAL POINT

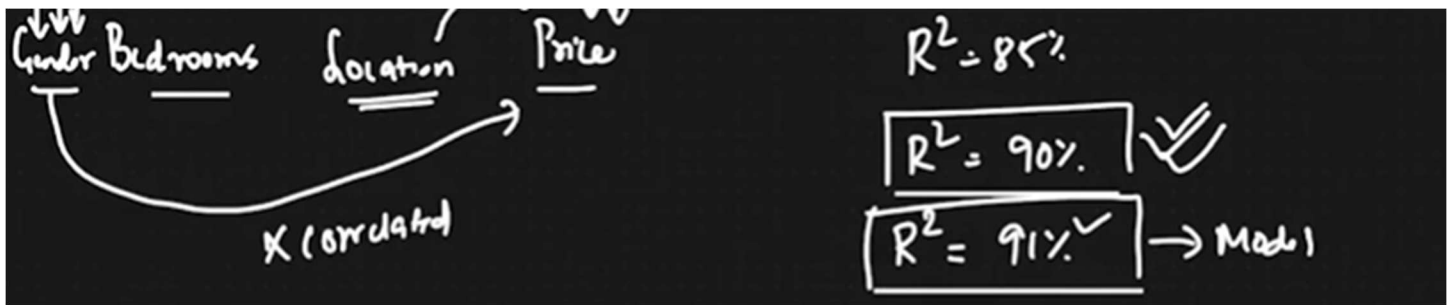
Y bar is the mean of y



If we keep on adding feature, then also there is a chance of increasing R², so in order to not increase

R² 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

$p = \text{features or predictors}$

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

$p = 2 \quad R^2 = 90\% \quad R^2_{\text{adjusted}} = 8$

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

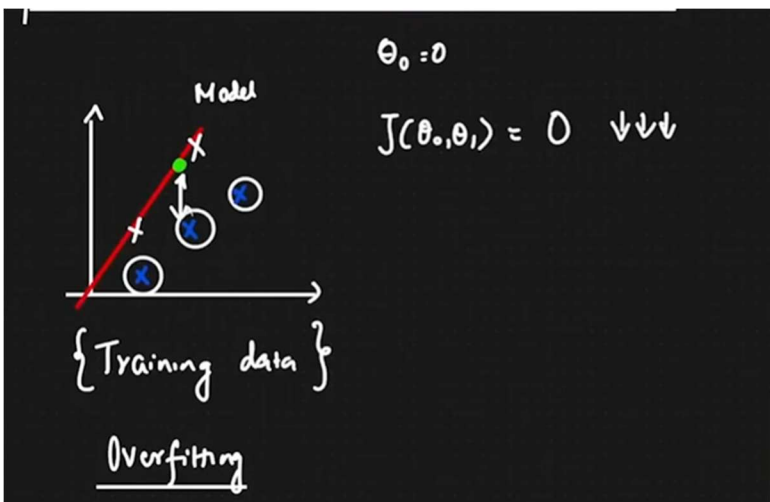
Interview Question: What is bigger R^2 or adjusted R^2

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 :

<u>Model 1</u>	<u>Model 2</u>	<u>Model 3</u>
Training Acc = 90%	Training Acc = 92%	Training Acc = 70%
Test Acc = 80%	Test Acc = 91%	Test Acc = 65%
↓	↓	↓
Overfitting	Generalized Model	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

how Variance

Overfitting

$J(\theta_1) = 0$

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

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

$h_{\theta}(x) = \hat{y}$

$\theta_0 = 0$

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

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

$\rightarrow \text{slope}$

Ridge (L2 Regularization)

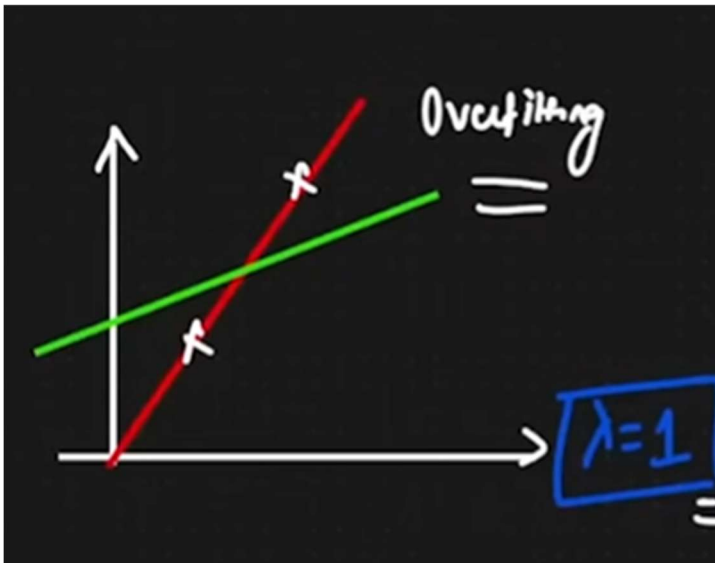
= 0

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

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

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:

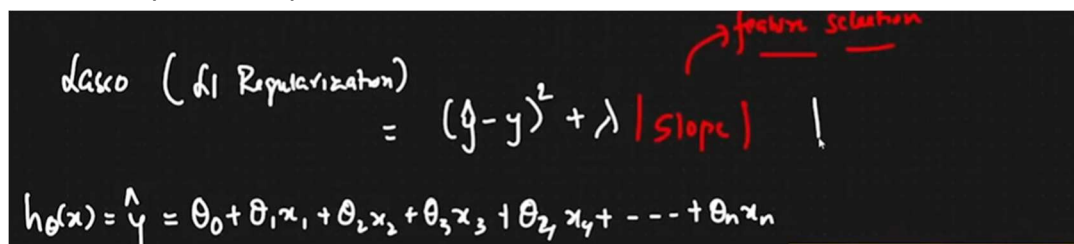
$$\begin{aligned}
 &= (\hat{y}^{(i)} - y^{(i)})^2 + \lambda (\text{slope})^2 \\
 &\quad \downarrow \\
 &= (\text{Small value}) + 1(1.5)^2 \\
 &= (\text{Small value}) + 2.25 \\
 &\quad \downarrow \quad \downarrow \\
 &\quad \underline{\underline{3}}
 \end{aligned}$$

- 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

Lambda: 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.

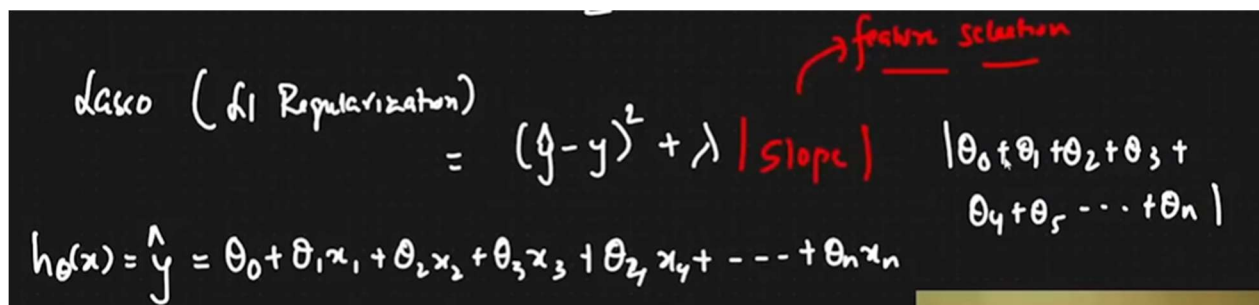
Lasso Regularization = $(\hat{y} - y)^2 + \lambda |\text{slope}|$

- Mod of slope will help us do feature selection



$$\text{Lasso (L1 Regularization)} = (\hat{y} - y)^2 + \lambda |\text{slope}|$$

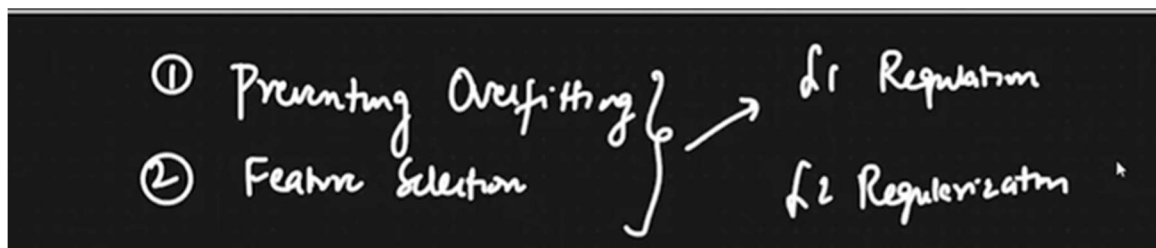
$$h_{\theta}(x) = \hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 + \dots + \theta_n x_n$$



$$\text{Lasso (L1 Regularization)} = (\hat{y} - y)^2 + \lambda |\text{slope}|$$

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

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



Lambda 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 regressuion funcgtn with different cost function

- Ridge regression prevents overfitting

Ridge Regression (L2 Norm)
 Cost function = $(h_0(x^{(i)}) - y^{(i)})^2 + \lambda (\text{slope})^2$
 Purpose : Preventing Overfitting

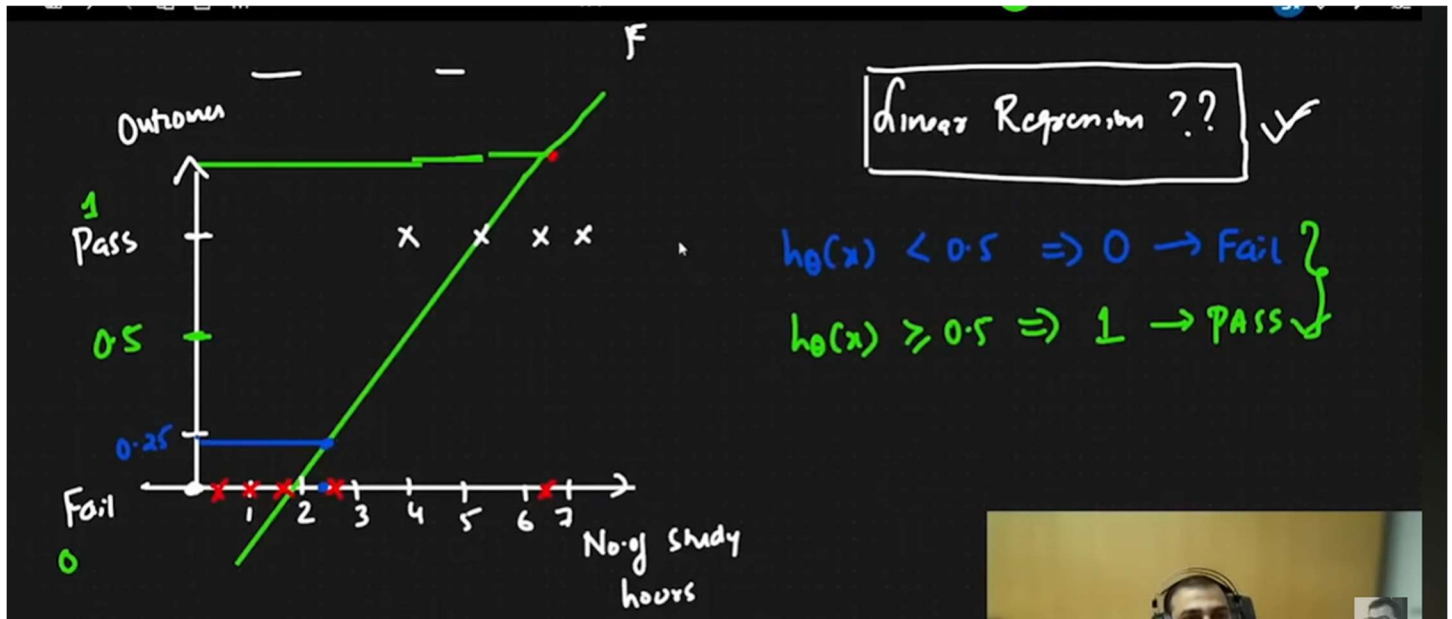
- Lasso regression prevents users from using unnecessary features/ help in feature selection

Lasso Regression (L1 Reg)
 Cost function = $(h_0(x^{(i)}) - y^{(i)})^2 + \lambda |\text{slope}|$

Assumption of Linear Regression

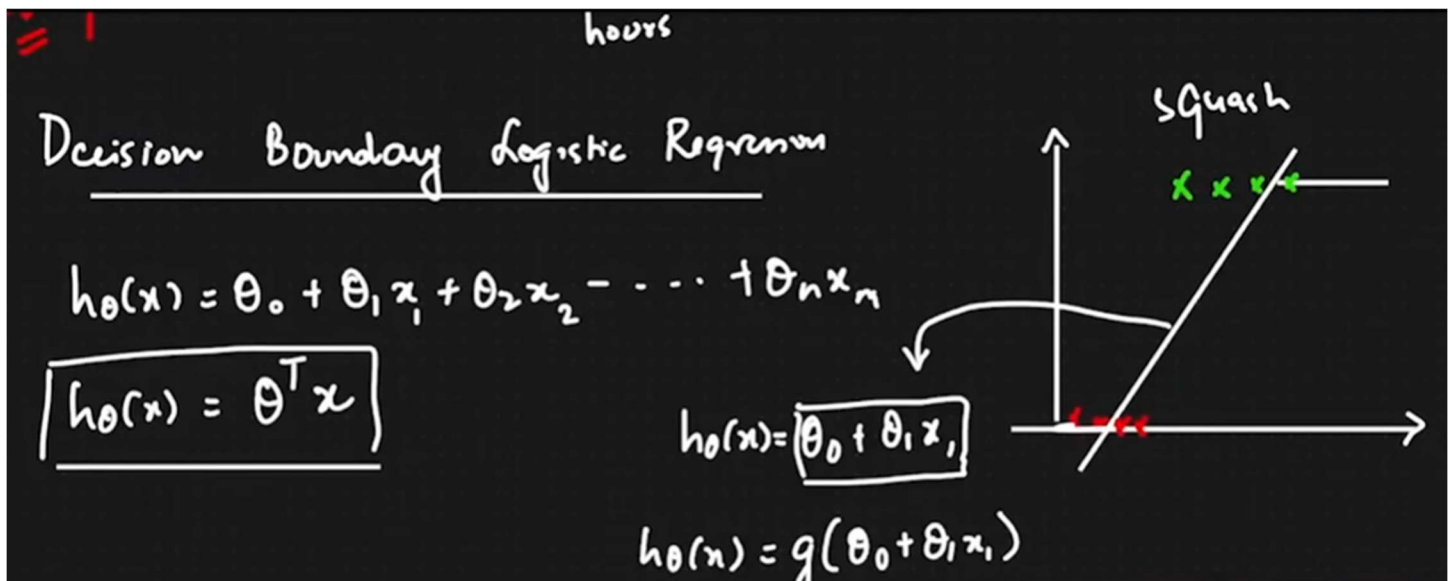
- ① Normal / Gaussian Distribution → Model will get trained well
- ② {Standardization {Scaling data} → Z-score $\mu=0, \sigma=1$ }
 ↗ 95%
- ③ Linearity X_3 $\boxed{X_1 \quad X_2}$ \boxed{Y} Variance Inflation factor?
- ④ Multi Collinearity

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



To pass >3(basic logic)

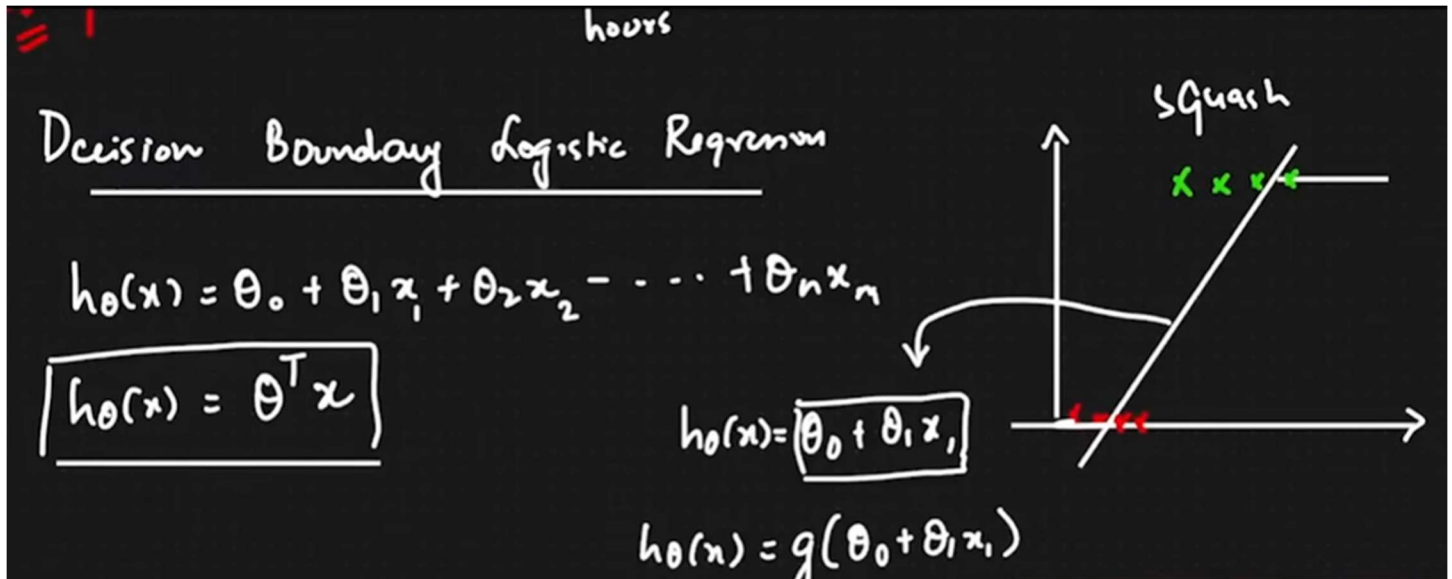
- Why not use linear regression then ?
 - With outlier the entire bestfit line would change, now for even 5 hrs study some people could fail according to this new model
- We could have to squash the functionality and this would be not good. Sepcially for negative inputs



BINARY CLASSIFICATION

- Our output value should always be 0 or 1

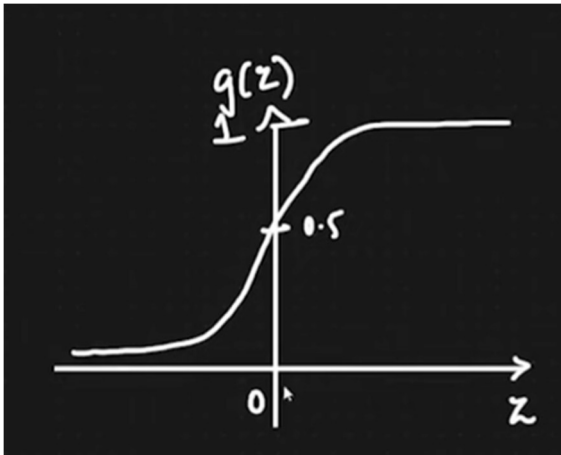
Decision Boundary Logistic regression



We will try to apply a mathematical formula which would squash this line

$$h_0(x) = g(\theta_0 + \theta_1 x)$$
$$\text{let } z = \theta_0 + \theta_1 x$$
$$h_0(x) = g(z)$$
$$h_0(x) = \frac{1}{1 + e^{-z}}$$
$$\boxed{h_0(x) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x)}}}$$

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



1:44