Mo Tu We Th Fr Sa Su Date :

ML

MACHINE LEARNING ALGORITMS

Introduction to ML (AI Vs ML Vs DL Vs DS)

Machine learning is a subset or subfield of artificial Intelligence (AI) that focuses on developing algorithms and statistical models that enables Computers to learn and make Predictions or decisions without being explicitly programmed to perform specific tasks. In essence it is a method of teaching Computers to learn from data and improve their performance over time through experience.

What is AI?

AI, or Artificial intelligence is a branch of Computer Science and Technology that focuses on creating intelligent machines and systems that can simulate human-like intelligence and performs tasks that typically require human intelligence.

What is an "AI" application?

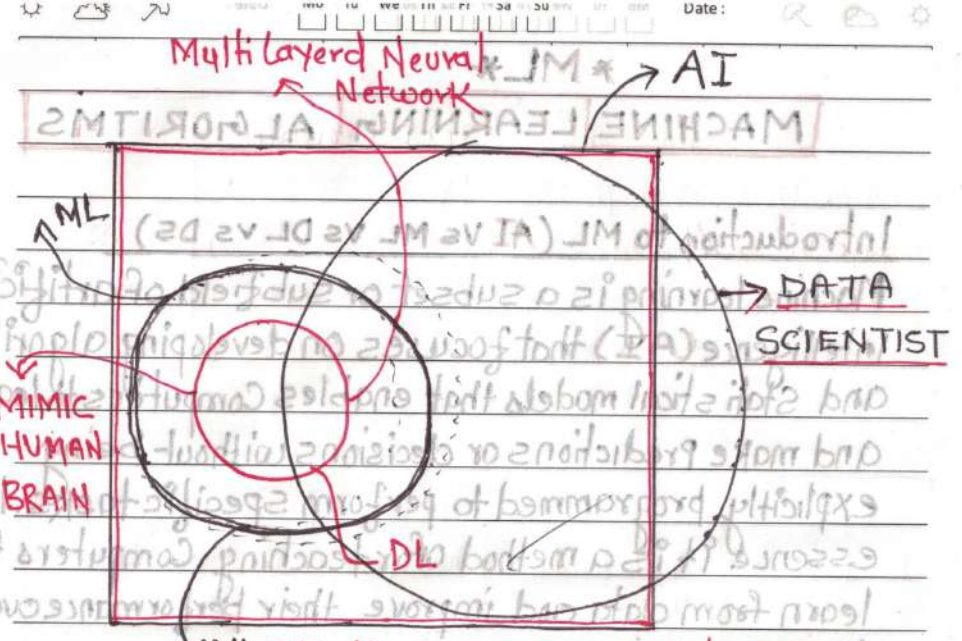
AI application is able to do it own task without any human intervention.

Example: Netflix [Action movie Recommendation
Comedy move Recommendation

This Recommendation is called Artificial Intelligence Application

: Amazon.in
↳ Iphone → Recommed → Headphone

Tesla → Self Driving Car (Elon musk)



ML ⇒ Machine learning is the subset of AI

Role of ML ⇒ It provides stats tool to analyze the data
Visualize the data, to do prediction, forecasting

What is Deep Learning?

Deep learning is a subset or subfield of machine learning (ML) that focuses on training artificial neural networks with many layers (hence the term "deep") to perform tasks such as pattern recognition, image and speech recognition, natural language processing, and more.

MACHINE & Deep Learning

Supervised ML
- Regression
- CLASSIFICATION

Unsupervised ML
- CLUSTERING
- Dimensionality Reduction

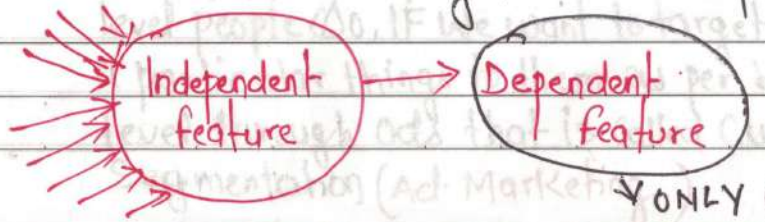
Age	Weight	← output
21	45	
24	59	
19	39	
16	36	

Take Age → Hypothesis → Weight → output
This model called (Ho)

* Dependent feature
⇓
Weight (output)

* Independent feature
⇓
Age (Input)

Note: Whenever we are solving Problems in the case of "Supervised ML" there one Dependent feature will be and there can be any number of Independent feature



Regression Problem

Age	Weight → O/P
21	72
23	71
24	71.5

When output have a Continuous Variable then these became Regression Problem Statement

Classification Problem

No. of hours	No. of playhour	No. of sleep	(O/P) Pass/Fail
→ INDEPENDENT FEATURE ←			Dependent
*	*	*	P
*	*	*	F
*	*	*	P
*	*	*	F

Note:

Whenever We have Fixed Number of Category then that became a Classification Problem statement

Suppose it has two output then it became Binary classification and if we have more than two different category it became Multiclass classification

Unsupervised ML

Salary	Age	No Dependent Variable → No O/P feature
*	*	-
*	*	-
*	*	-

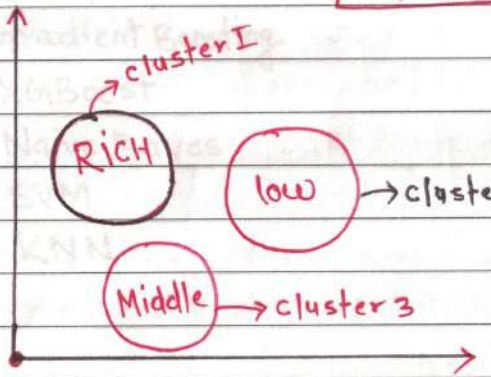
In this scenario We use "Clustering"

"Clustering is the Grouping Algorithm" ↓

Customer Segmentation

Clustering basically means that based on data we have to find out similar group : eg: Group of People.

GROUPING



We are grouping them together

Eg: Suppose A Company lunches a product "A" For Rich people and product "B" For Middle Class people and product "C" For low income level people Do, IF We want to target or sent a particular things to them as per their level through ads that is called Customer Segmentation (Ad. Marketing)

Dimensionality Reduction

Eg: Suppose IF we have 1000 Feature, can we reduce this feature to lower dimension

1000 Feature —convert→ Lower Dimension

Yes, It is possible with the help of Dimensionality Reduction Algoriths ⇓

PCA, LDA

Machine Learning

⇓

Algoriths

⇓

Supervised	Unsupervised
• Linear Regression	• K-means
• Ridge & Lasso	• DBSCAN
• Logistic Regression	• Hierarchical clustering
• Decision Tree	• K Nearest Neighbor clustering
• Ada Boost	• PCA
• Random Forest	• LDA
• Gradient Boosting	
• XGBoost	
• Naive Bayes	
• SVM	
• KNN	

Mo Tu We Th Fr Sa Su Date:

What is linear Regression?

- Linear regression analysis is used to predict the value of a variable based on the value of another variable. The variable we want to predict is called the dependent variable. The variable we are using to predict the other variable's value is called the Independent variable

- Linear regression is a fundamental machine-learning technique used for modeling the relationship between dependent variable (also called target or output) and one or more independent variable (also called features or predictors). It is commonly employed for tasks like predictive modeling and data analysis.

- In simple linear regression, there is only one independent variable, and the goal is to find a linear equation that best represents the relationship between the independent variable(s) and the dependent variable. Equation of Simple linear Regression

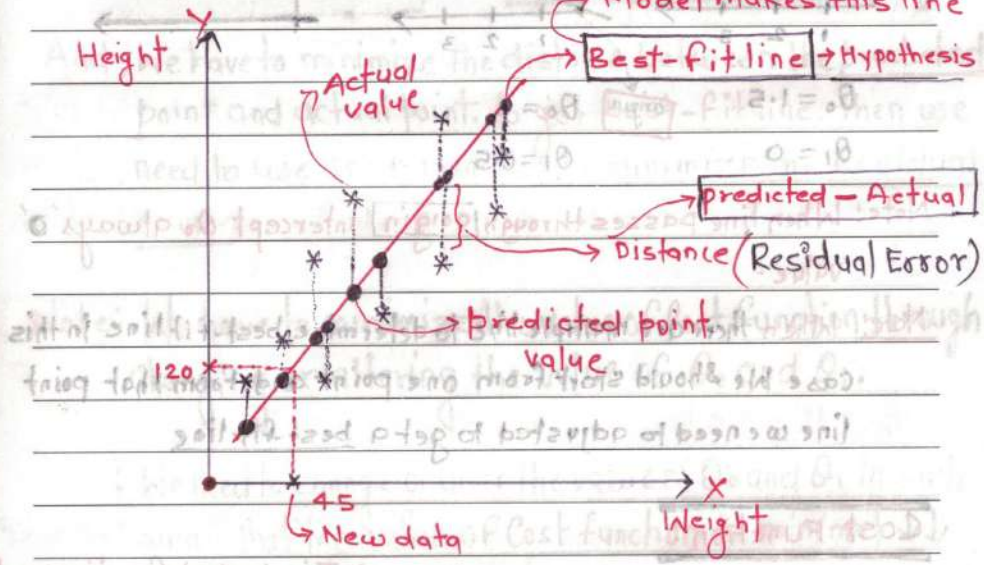
$$y = mx + c$$

- y → Dependent variable
- m → Slope of line
- x → Independent variable
- c → Y-intercept

Note: [c ⇒ is the y-intercept (The point where the line intersects the y-axis)]

Mo Tu We Th Fr Sa Su Date:

The objective of linear regression is to determine the values of [m] and [c] that minimize the difference between the predicted values ($mx+c$) and the actual values of the dependent variable 'y' in the dataset. This is typically done using a method called [Least squares], which find the values of m and c that minimize the sum of squared differences between the predicted and actual values.



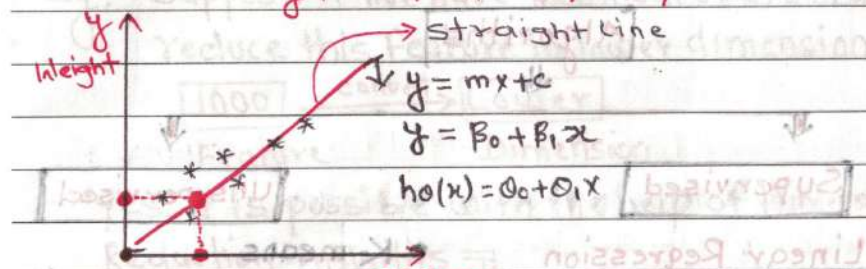
[Residual Error = Distance Between Actual value point and predicted point value]

Note:
- If we sum of all distances the sum or Residual Error value must be very minimal to be a best-fit line
- Minimal Residual Error value will be Accepted even there are many lines.

To be best-fit line ⇒ Minimal Residual Error

LINEAR REGRESSION

Taking two feature X & Y



{'y' is a linear function of 'X'}

IN Linear regression we try to create a Model, with the help of above training dataset

TRAIN DATASET

⇓

Model

⇓

Take as inp NEW AGE → Hypothesis → o/p (Weight)

In Linear Regression we are trying to find out a best-fit line which help to do prediction

LINEAR REGRESSION ⇒ It basically means we are going to create a linear line over there

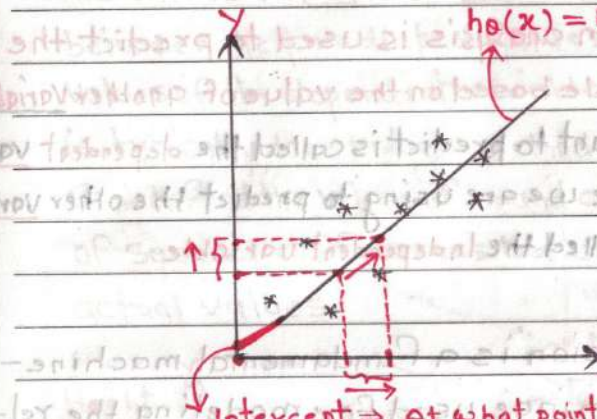
Equation of straight line : ⇓

$$y = mx^{(i)} + c$$

$$y = \beta_0 + \beta_1 x^{(i)}$$

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

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



When, $x = 0$

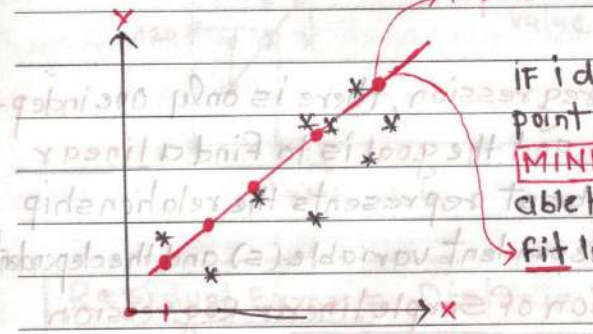
θ_0 = Intercept

θ_1 = slope or Coefficient

$x_{(i)}$ = data points

Intercept ⇒ At what point the straight line cut or touch the Y axis.

Note: Unit movement in X then what unit movement in Y is called Slope or Coefficient



IF i do the summation of these point those distance should be MINIMAL then only I will be able to say that this is the best-fit line

IF I take an example of LR I need to find out the best-fit line in such a way that the distance between these data point that I have above graph and the predicted point should be very-very less.

NOTE: IN Multiple line we use ⇒ start at point and for best-fit line

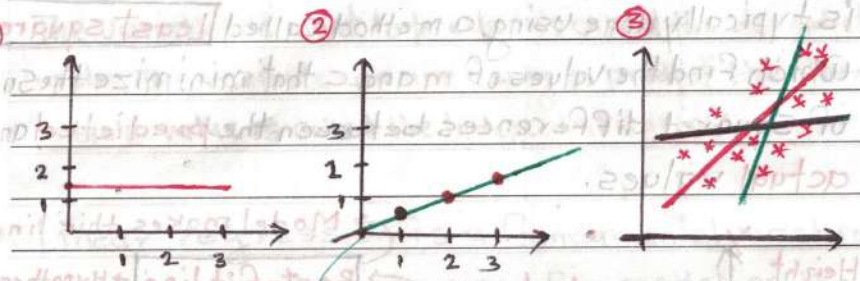
Hypothesis

{y is a linear function of x}

$h_\theta(x) = \theta_0 + \theta_1 x_1$ ⇒ This is a Simple Hypothesis

Which create or make the best-fit line once.

① ② ③



$\theta_0 = 1.5$ origin $\theta_0 = 0$

$\theta_1 = 0$ $\theta_1 = 0.5$

Note: When line passes through origin intercept θ_0 always '0' value.

Note: When there are multiple line to determine best-fit line in this case we should start from one point and from that point line we need to adjusted to get a best-fit line

Cost Function:

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

⇒ This is Cost function and our main aim to minimize its value to get a best-fit line.

OR

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

→ Squared Error Function

m ⇒ No. of data point

Note: We have to find out Sub of error Average that's why We devide by 'm'.

Explanation:

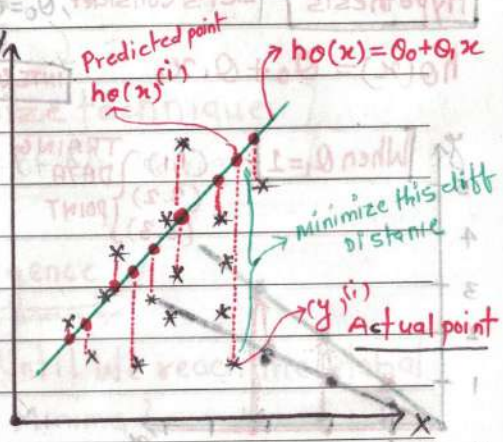
$h_\theta(x) = \theta_0 + \theta_1 (x)^{(i)}$

↓

Hypothesis

$h_\theta(x)^{(i)}$ ⇒ Predicted point

$y^{(i)}$ = Actual point



And, We have to minimize the distance between the predicted point and actual point. to get best-fit line. then we need to use cost function to minimize the Residual Error.

Note: We have to minimize the value of Cost function through changing or altering the value of θ_0 and θ_1.

: We need to change or alter the value of θ_0 and θ_1 in such away that the value of Cost function is minimal ↓↓

: Differentiation is used to find out that point slope

Mo Tu We Th Fr Sa Su Date:

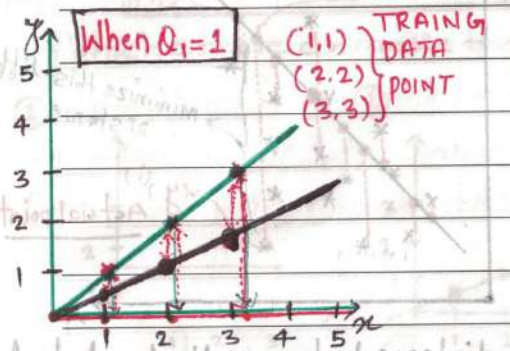
Hypothesis | Let's Consider, $\theta_0 = 0$ | **Cost Function**

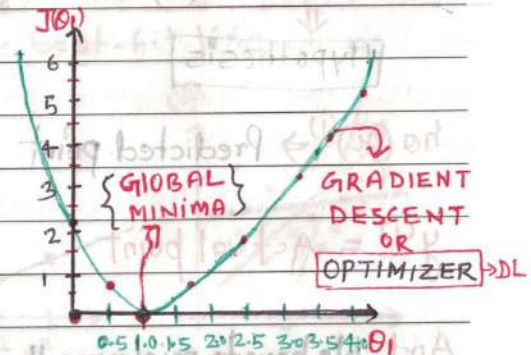
$h_\theta(x) = \theta_0 + \theta_1 x_1$ — INTERCEPT

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

When $\theta_1 = 1$

(1,1), (2,2), (3,3) } TRAING DATA POINT





GRADIENT DESCENT OR OPTIMIZER → DL

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

$h_\theta(x) = 1 \times 2$

$h_\theta(x) = 1 \times 3$

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

$= \frac{1}{2 \times 3} \left[(0)^2 + (0)^2 + (0)^2 \right]$

$J(\theta_1) = 0$ & $\theta_1 = 1$

Now plot in above $J(\theta_1) = 0$ & $\theta_1 = 1$

When, $\theta_1 = 0.5$

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

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

$J(\theta_1) = \approx 0.58$ & $\theta_1 = 0.5$

When, $\theta_1 = 0.0$

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

$= \frac{1}{6} \left[(0-1)^2 + (0-2)^2 + (0-3)^2 \right]$

$J(\theta_1) = \approx 2.3$ & $\theta_1 = 0.0$

Note: GRADIENT DESCENT automatically change the value of θ_1

Note: Unless and Until We have to change the θ_1 value when not reach at minimal point

Mo Tu We Th Fr Sa Su Date:

Convergence Algorithm

It is a Optimize technique

Optimize the changes of θ_1 values

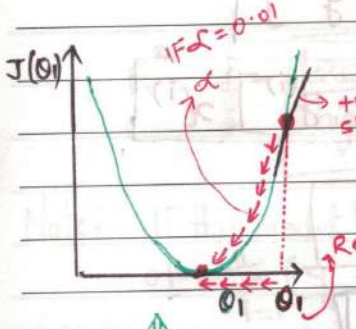
Repeat Until Convergence

→ Until We reach the Global Minima }

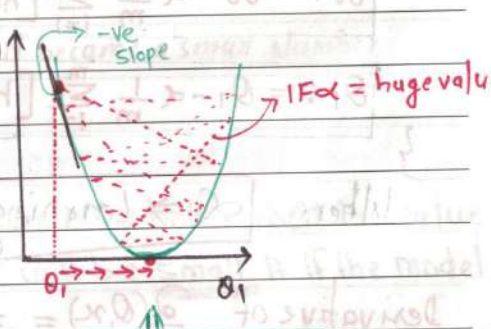
$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_j)$ OR $\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$

Learning Rate — Derivative (slope)

Note: This equation is actually help us to change our θ_1 value much more efficiently.



$\theta_1 := \theta_1 - \alpha (+ve)$



$\theta_1 := \theta_1 - \alpha (-ve)$

$\theta_1 := \theta_1 + \alpha (ve)$

Note: By What speed We should be Coming to near Global Minima → α (Alpha)

: Learning Rate usually a small value,

Mo Tu We Th Fr Sa Su Date:

Find out derivative of θ_j with respect to J of θ_0 & θ_1 respectively. $(J(\theta_0, \theta_1))$

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

When, $J = 0$ & 1

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

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

Note: Derivative of x^2 is $2x$.

$\Uparrow$ Convergence theorem $\Uparrow$

Repeat Until Convergence

{

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

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

}

Where, $\alpha \Rightarrow$ Learning Rate.

Derivative of $\frac{\partial}{\partial \theta_1}(\theta_1 x) = = x$

Mo Tu We Th Fr Sa Su Date:

R squared

$$R^2 = 1 - \frac{SS_{Res}}{SS_{Total}}$$

$\Rightarrow$ Sum of square Residual

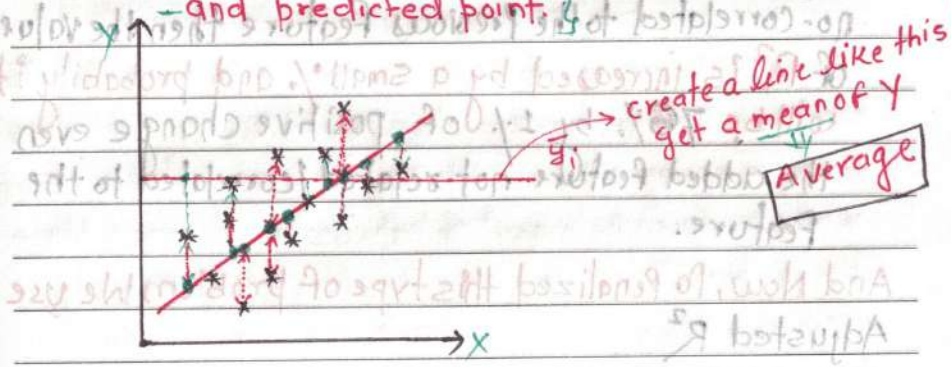
$\Rightarrow$ Sum of square Total

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

$h_\theta(x)$ $\Downarrow$ Predicted point

Mean of Y

Note: Sum of Residual Error is difference between Actual point and predicted point.



$$R^2 = 1 - \frac{\text{low}}{\text{high}}$$ → It became a small Number

Note: IF the model performance is better then the value of $\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$ will be small & If the model performance is bad it means that the value of $\sum_{i=1}^{n} (y_i - \bar{y})^2$ will be big.

Note: IF $R^2 = 100\%$ then it is called Overfitting but Not Possible probably.

Adjusted R squared

DATASET ↘ Let's Consider $R^2 = 71\%$.

Gender	Size of House	No. of Bedrooms	location House	Price of House

Explanation ↘ If We have three feature Our R^2 is 71%, If We add one feature more our R^2 is Changed to 75%. But the feature We have added that is correlated to the previous feature that's why R^2 increased by 4%. But If We add an additional feature that is no-correlated to the previous feature then the value of R^2 is increased by a small % and probabily it will be 76%. by 1% of positive change even the added feature not related/correlated to the feature.

And Now, To Penalized this type of problem, We use Adjusted R^2.

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

Where, N = No. of datapoints

P = No. of independent feature.

↓ No. of Predictors

Note: If any feature are added to our dataset that is not important and not correlated for that dataset and due to this unnecessary feature R^2 increased then after We need to calculate Adjusted R square (R^2) value which minimize the value of R^2

Note: Adjusted $R^2 < R^2$

Ridge Regression ⇒

Ridge regression, also known as L2 regularization, is a technique used to prevent overfitting in linear regression, the objective is to minimize the sum of squared

Mo Tu We Th Fr Sa Su Date:

Ridge Regression → Ridge regression, also known as L2 Regularization, is a technique used to prevent overfitting in linear regression models. by adding a penalty term to the loss function. In traditional linear regression, the objective is to minimize the sum of squared differences between the predicted value and actual values (least squares). However, this can lead to overfitting when there are many input features or when some features are highly correlated.

Ridge Regression adds a regularization term to the least squares loss function, which is a proportional to the sum of the squared coefficients of the linear regression model. The regularization term encourages the model to keep the coefficients small, which in turn reduces the impact of individual features on the prediction. This helps in mitigating multicollinearity (High correlation between features) and improves the Generalization of the model.

The Ridge Regression loss function ↓

⇒ Loss = least squares loss + λ*(Sum of Squared (oefficients)

Where,

λ (lambda) is a regularization parameter, which controls the strength of regularization. A higher λ value results in more Aggressive Regularization.

Cost Function = $(h_\theta(x)^{(i)} - y^{(i)})^2 + \lambda(\text{slope})^2$ → Ridge Regression (L2 Norm)

↳ Purpose: preventing Overfitting

Mo Tu We Th Fr Sa Su Date:

Lasso Regression

((L1 Regularization

Lasso Regression, or L1 regularization is a technique used to prevent overfitting and perform feature selection in linear regression models. It is similar to Ridge Reg. It adds a Penalty term to the loss function, but instead of using the Sum of Squared Coefficients, it uses the Sum of the absolute values of the coefficients.

The Lasso Regression loss function ↓

↳ Loss = Least squares Loss + λ*(Sum of absolute values of Coefficients)

Note: Like Ridge Regression, λ controls the strength of regularization in Lasso. However, one key differences is that Lasso tends to drive the coefficients of less important features to exactly zero. This property makes Lasso useful for feature selection. Because it effectively eliminates irrelevant features from the model.

Cost function = $(h_\theta(x)^{(i)} - y^{(i)})^2 + \lambda|\text{slope}|$ → L1 Reg.

↳ Purpose: Prevent Over fitting. : Feature Selection.

↳ mode

↳ slope

GRADIENT DESCENT

Gradient Descent is an algorithm that finds best fit line for given training data set.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_{predicted})^2 \rightarrow \Delta^2$$

→ Mean Squared Error

Cost function ⇒ $\frac{1}{n}\sum_{i=1}^{n}(y_i - (mx_i + b))^2$

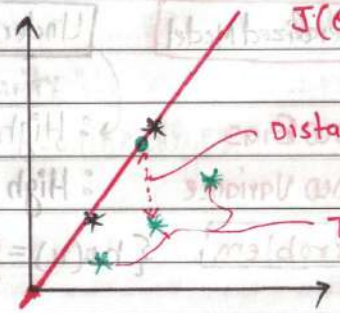
Note:

The Algorithm that are used to find out MSE is called Gradient Descent.

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

$\theta_0 = 0$

$J(\theta_0, \theta_1) = 0$



Note: Ploted two points are called Training data

↳ Model is trained by Training Data.

Overfitting ⇒ : Model performs well → Training data.

: Fails to performs well → Test data

→ High variance

→ Low Bias

Underfitting ⇒ : Model Accuracy is bad with → Training data.

: Model Accuracy is also bad with → Test data.

→ High variance

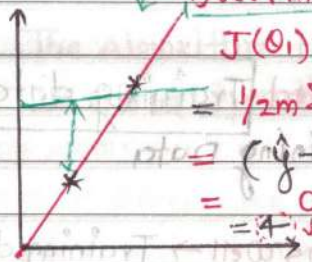
→ High Bias

Mo Tu We Th Fr Sa Su Date:

Example

Model 1	Model 2	Model 3
Training Accuracy = 90%	92%	70%
Test Accuracy = 80%	91%	65%
⇓	⇓	⇓
Overfitting	Generalized Model	Underfitting
: low Bias	: low Bias	: High Bias,
: High Variance	: low Variance	: High Variance.

Overfitting Problem



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

$J(\theta_1) = 0$

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

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

$= 0 + 1(2)^2$

$= 4$ ↓↓

Note:

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

When, $\theta_0 = 0$

then, $h_\theta(x) = \theta_1 x$ → slope

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

$= (\text{small value}) + 1(1.5)^2$

$= \text{small value} + 2.25$

≈ 3 ↓↓

Note: We have to reduce the value unless and until we get a line which is/will be able to handle the Generalized Model.

Mo Tu We Th Fr Sa Su Date:

Assumption of linear Regression

⇒ If our features are in: Normal/Gaussion Distribution.
Or If our features follows these Distribution then H Obviously good a model will get trained value.

⇒ Standardization: scaling data by using Z-score, $\mu = 0$, $\sigma = 1$,

⇒ Linearity:

⇒ Multi Collinearity:

* Variation Inflation factor

Logistic Regression

- Logistic Regression is a statistical and Machine learning model used for Binary classification tasks. It is a type of Regression analysis that predicts the probability of a binary outcome (1/0, YES/No, True/False) based on one or more predictor variables. Despite its name, logistic regression is used for classification, not regression.
- Logistic regression is a type of Machine learning — Algorithm that are used for classification tasks, and models the probability that a sample belongs to a certain class using a logistic function.
- This is the first type of Algorithm in Classification

Example:

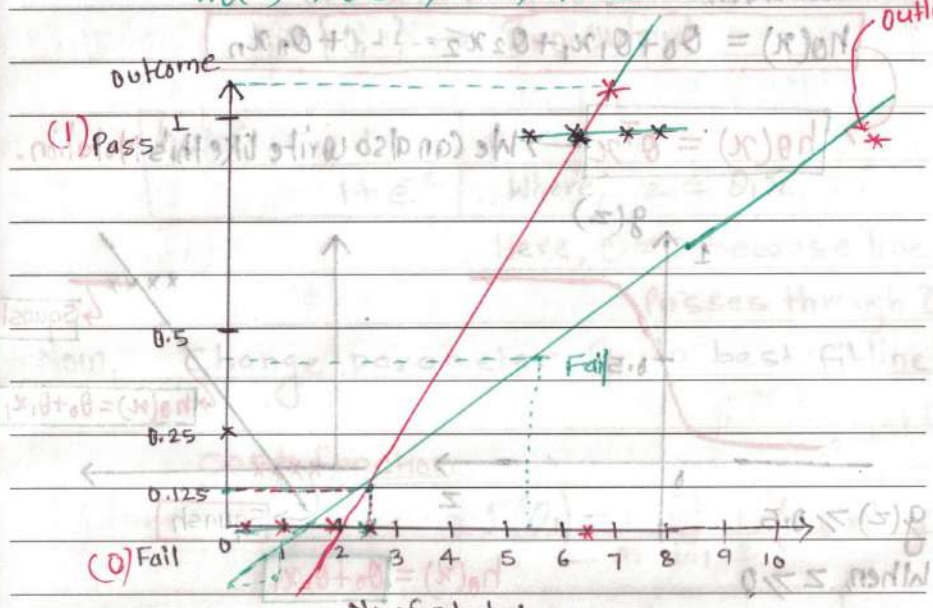
No. of study(hr.)	No. of play(hr.)	P/F (Two category ↘ Binary (Y/N))
-	-	P
-	-	F

Note: logistic Regression is very well with Binary classification

- We can solve multiclass classification with logistic Regression.

Condition: $h_\theta(x) < 0.5 \Rightarrow 0 \rightarrow$ Fail

$h_\theta(x) \geq 0.5 \Rightarrow 1 \rightarrow$ PASS



Sigmoid Function

(Linear Regression can not be used to solve the above Problem)

Note: The above problem is related to Binary classification Problem that's why its value lies between 0 to 1

: Squash ⇒ to make straight

xxxx Squash

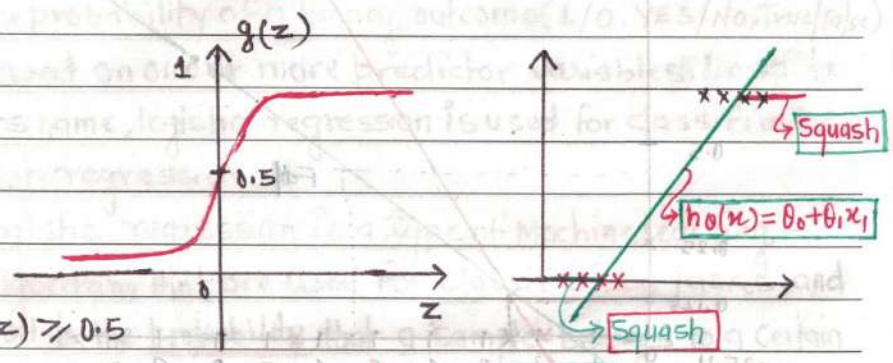
xxxx squash

Decision Boundary in case of logistic Regression

Lets Consider,

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

$h_\theta(x) = \theta^T x$ → We can also write like this Notation.



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

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

let,

$z = \theta_0 + \theta_1 x$

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

$h_\theta(x) = \frac{1}{1+e^{-z}}$ → Sigmoid or logistic function.

$h_\theta(x) = \frac{1}{1+e^{-(\theta_0+\theta_1 x)}}$ → It squash the function

This is Hypothesis

Note:
- With the help of Regression We are creating the straight line.
- With the help of the Concept of Sigmoid We are able to Squashing.

Training set

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

$y \in \{0, 1\}$ → Two output

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

Here, $\theta = 0$, because line passes through Origin

Now, Change parameter θ_1 to best fitline.

Cost function

Linear Regression → $J(\theta_1) = \frac{1}{m}\sum_{i=1}^{m} \frac{1}{2}\left(h_\theta(x)' - y'\right)^2$ ← cost function

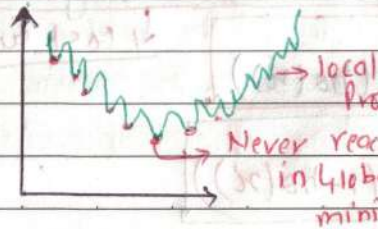
logistic Regression ⇒ $h_\theta(x) = \frac{1}{1+e^{-(\theta_1 x)}}$

Cost function = $1/2\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$ } We can not use this cost function for logistic Regression

Replacing $h_\theta(x) = \frac{1}{1+e^{-(\theta_1 x)}}$ } Non convex function.

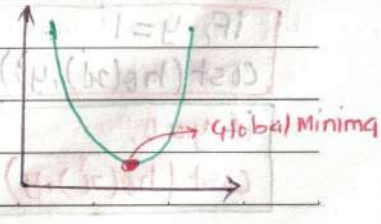
Non Convex function

Gradient Descent



Convex function

Gradient Decent



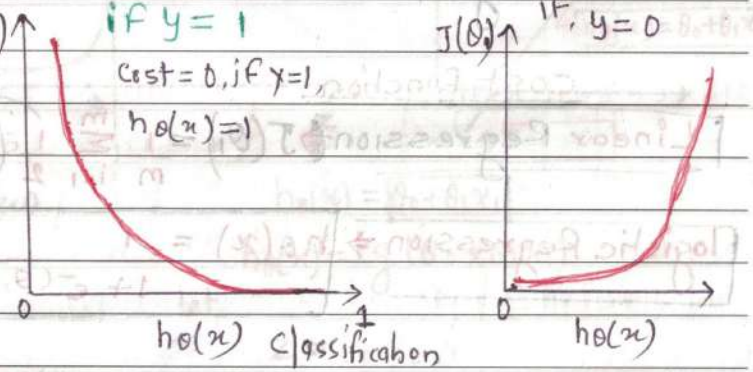
Mo Tu We Th Fr Sa Su Date:

LOGISTIC REGRESSION Cost Function

$$J(\theta_1) = \begin{cases} -\log(h_\theta(x^i)) & y=1 \\ -\log(1-h_\theta(x^i)) & y=0 \end{cases}$$

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

Now:→



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

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

↓ cost function

if, $y=1$

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

if $y=0$,

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

Mo Tu We Th Fr Sa Su Date:

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

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

((Repeat it Until Convergence))

{

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

}

Performance Metrics

- Performance metrices in machine learning are meg-sures used to evaluate the performance of a machine learning model. These metrices help assess how well a model is making predictions or classifications on a given dataset.
  - The choice of Performance metrices depends on the specific task and goals of the Machine learning Project.

:⇒ Some of the Common Performance metrices used in Machine learning.

#Accuracy::

This is the most straightforward metric and is simply the ratio of correctly predicted instances to the total instances. It is suitable for balanced datasets, where the classes are roughly equally distributed.

$$\text{Accuracy} = \frac{\text{Number of correct Predictions}}{\text{Total Number of Predictions}}$$

Confusion Matrix → A confusion matrix is a table used to evaluate the performance of a classification algorithm. It provides a more detailed breakdown of the model's predictions compared to a single accuracy score. The confusion matrix is especially useful when dealing with imbalanced datasets.

The confusion matrix is a matrix used to determine the performance of the classification models for a given set of test data. It can only determined if the true values for test data are known. The matrix itself can be easily understood, but the related terminologies may be confusing. Since it shows the errors in the model Performance in the form of a matrix, hence also known as an error matrix. : source → (javatpoint.com) → (only this one)

Components of a confusion matrix

Suppose we have a binary classification problem (two classes: positive and negative)

- True Positive (TP): Instances that are actually Positive and are correctly classified as positive.
- True Negative (TN): Instances that are actually negative and are correctly classified as Negative.
- False Positive (FP): Instances that are actually negative but are incorrectly classified as positive. (Type I error)
- False Negative (FN): Instances that are actually positive but are incorrectly classified as Negative. (Type II error)

Confusion Matrix Presentation → 2X2 table ⇒

		Predicted class	
		Negative	Positive
Actual class	Negative	TN	FP
	Positive	FN	TP