ML, AL, DL

DL is a subset of ML.

Different types of Algorithm

→ Supervised Algorithm :- Needs a supervisor

→ Unsupervised Algorithm

→ Semi-supervised

→ Reinforcement

→ ~~Inputing labeled data~~

→ Supervised Algorithm

- Needs a supervisor
- Inputing labelled data
- Classification.

→ Unsupervised Algorithm

- No labelled data
- Automatically trains.
- Identifies the common features to group into diff categories

→ Semi-supervised

- Huge amount of supervised
- Small amt of unsupervised.

	Has heart disease	Does Not Have Heart Disease.
Has heart Disease	T P.	FP
Does not have Heart Disease	F N	TP.

Sensitivity

- A measure of how well a machine learning model can detect positive instances
- Sensitivity measures the ability of a model to correctly identify positive examples
- Sensitivity is used to evaluate model performance because it allows.
- Sensitivity = (TP) / (TP+FN)

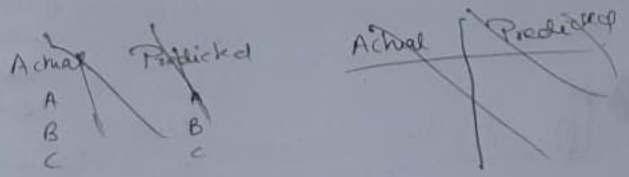
Specifity :-

Specifity

- Measures the proportion of how negative that are correctly identified by model.
- High specificity means that the model is correctly

Specificity = TN / T·N + F·N

Q. Consider the following 3 class confusion matrix. Calculate precision, recall, waited average, precision & recall.



Actual \ Predicted	A	B	C	Total.
A	15	2	3	20
B	4	15	8	30
C	2	3	45	50
Total	24	20	56	100

24
20
56
0

Precision = $\frac{TP}{TP+FP} = \frac{15}{15+9}$ class A

$= \frac{15}{24} = \frac{5}{8}$

$= \frac{15}{20} = \frac{3}{4}$ class B

$= \frac{45}{56}$ class C.

Recall = $\frac{TP}{TP+FN} = \frac{15}{20}$ → class A

$= \frac{15}{30}$ class B

$= \frac{45}{50}$ class C.

Accuracy = $\frac{15+15+45}{100}$

	Actual	Predicted
TP	+	+
TN	−	−
FP	−	+
FN	+	−

Precision : $\frac{TP}{TP+FP}$ Recall : $\frac{TP}{TP+FN}$

Confusion Matrix

- Given dataset of P positive instances & N negative

	Yes	No
Yes	TP	FN
No	FP	TN

accuracy : $\frac{TP+TN}{P+N}$

- Imagine using classifier to identify positive cases

precision = $\frac{TP}{TP+FP}$ recall = $\frac{TP}{TP+FN}$

Probability that a randomly selected +ve pdct is indeed

Probability that a randomly selected +ve is identified

Q. TP when the actual label is +ve & the machine learning label predicts +ve

TN when the actual label is -ve & the machine learning label predicted as -ve

FP when the actual label is -ve & the machine learning label predicted as +ve

FN when the actual label is +ve & machine learning label is -ve

Q. Consider a two class classification pblm of predicting whether a photograph contains man or woman. Suppose we have a test data set of 10 records with expected outcomes & a set of predictions from our classification algorithm. Compute the confusion matrix, accuracy, precision & recall

	Actual	~~Label~~ Predicted
1	Man	Woman
2	Man	Man
3	Woman	Woman
4	Man	Man
5	Man	Woman
6	Woman	Man
7	Woman	Man
8	Man	Woman
9	Man	Woman
10	Woman	Woman

Actual →

		Man	W
P	Man	TP	
↓	Wom		

TP =

Q Sales of a company (in Million Dollars) for each yr are shown in the table below

x (yr)	y (sales)
2005	12
2006	19
2007	29
2008	37
2009	45

a) Find the least square regression line

$y = ax + b$

t	y	t^2	ty
0	12	0	0
1	19	1	19
2	29	4	58
3	37	9	111
4	45	16	180
10	142	30	368

$a = \frac{5 \times 368 - 10 \times 142}{}$

b) Use the least square regression as a model to estimate the sales of the company in 2012.

$y = 8.4 \times 7 + 11.6$

$= 58.8 + 11.6$

$= 70.4$ M $

Q. The values of x & the corresponding values of y are given below

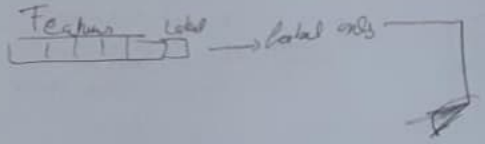
x	y	xy	x^2
0	2		
1	3		
2	5		
3	4		
4	6		

a) Find least square regression line

$y = ax + b$

$n = 10$

~~Start by role~~



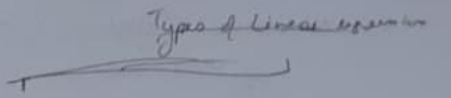
Simple Linear Regression

- supervised machine learning algo
- tries to find out the best linear relationship that describes the data you have.
- assumes that there exists a linear relationship between a dependent variable & independent variable.
- The value of the dependent variable of a linear regression model is a continuous value i.e, real nos

Representing Linear Regression Model

- Represent the linear relationship btw a dependent variable & independent variable via a sloped straight line.
- The sloped straight line representing the linear relationship that fits the given data best is called as a regression line.
- It is also called a best fit line

- Based on the no of independent variables, 2 types of linear regression



Simple Linear Regression

- dependent variable depends only on a single independent variable.

The model represented as:

$$Y = \beta_0 + \beta_1 X$$

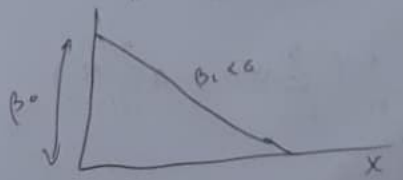
Y - dependent variable
X - independent variable.

β_0 & β_1 regression coefficient

β_0 - intercept or the bias that fixes the offset in a line.

β_1 - slope or wt that specifies the factor by which X has an impact on Y

- **Case 1: $\beta_1 < 0$**
  - indicates that variable X has -ve impact on Y.
  - If X increases, Y will decrease & Y via-versa



ML Life Cycle

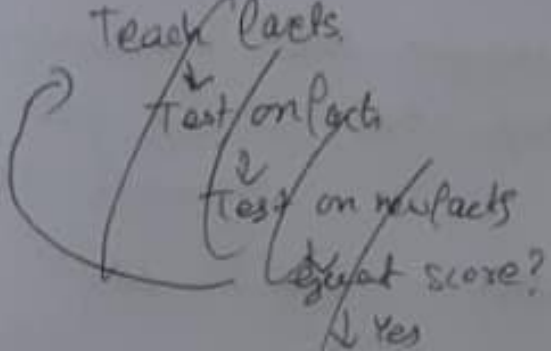
Model Selection
↓
Review the parameters
↓
Select the right searching method
↓
Apply cross validation approach
↓
Access the score of the model

Rule Based System

Where we'd amass a large no of rules for the computer to follow in order to imitate the human expert

This process of hand-crafting the rules to understand data is sometimes called Feature engineering.

Learning Strategy



Underfitting

- When it cannot capture the underlying trend of the data, i.e, it only performs well on training data but performs poorly on testing data
- Destroys the accuracy of our machine learning model
- The also doesn't fit the data well enough

Reasons for Underfitting

- High bias & low variance
- The size of the training dataset used is not enough
- The model is too simple
- Training data is not cleaned & also contains noise in it

Techniques to reduce underfitting :-

- Increase model complexity
- Increase the no. of features, performing feature engineering
- Remove noise from the data
- Increase the no of epochs or increase the duration of training to get better result

Overfitting

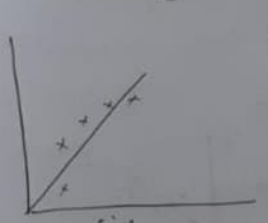
- When the

Reasons for Overfitting

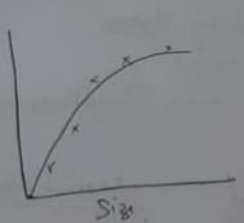
- High variance & low bias
- The model is too complex
- The size training data

Techniques to reduce overfitting

- Increase training data
- Reduce model complexity
- Early stopping during training phase
- Ridge Regularization & Lasso Regularization
- Use dropout for neural networks to tackle overfitting



High bais (underfit)

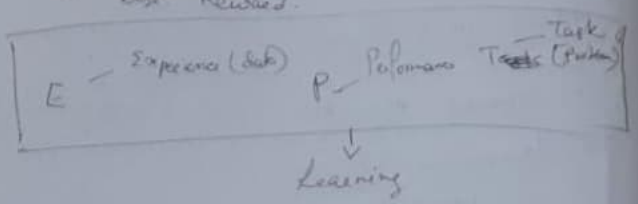


underfit

overfit

→ Reinforcement

- system learns from the environment
- using sensors
- we backtrack how if an obstacle (remarks)
- else Reward.



Supervised

Classification

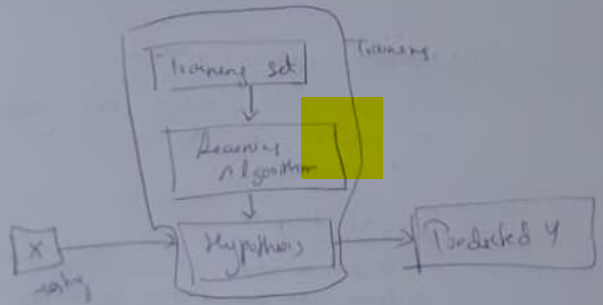
Regression (how variable changes w.r.t other variable)

continuous value

Supervised learning

Feature

→ Individual observations whether analyzed into a set of quantifiable properties



→ Learning a class from Examples / samples.

- Family & non-Family car
- Set of car: Class: Family & non
- People grouped based on Family cars

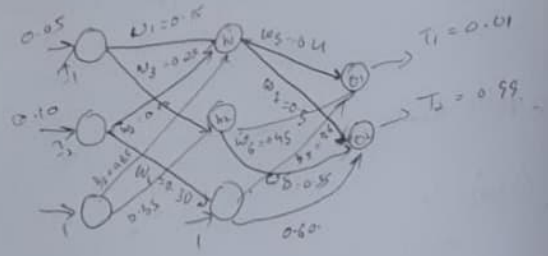
positive – Family car
negative – non-Family car.

Hypothesis: separates +ve & -ve values

Hypothesis class: Set of all hypothesis.

Empirical Error

→ the proportion of training instances where predictions of h do not match the required values given in X



Step1:- Assign random wts to each connection

Step2:- Pass the input values to the 1st layer.

Step3:- Calculate the outputs using sigmoid activation fn

$$f(x) = \frac{1}{1+e^{x}}$$

Output h_1 = $f(w_1 \times I_1 + w_2 \times I_2 + b_1)$

$= f(0.15 \times 0.05 + 0.20 \times 0.10 + 0.35)$

$= f(0.0075 + 0.002 + 0.35)$

$= f(1.0085)$

$= \frac{1}{1+e^{-1.3775}}$

$= 0.5932$

Output h_2 = $f(0.25 \times 0.05 + 0.35 \times 1)$

$= f(0.0125 + 0.35)$

$= f(0.3625)$

$= \frac{1}{1+e^{-0.3625}}$

$= 0.59688$

$O_1 = f(0.4 \times 0.5932 + 0.45 \times 0.596 + 0.6)$

$= f(1.60551)$

$= 0.75137$

$O_2 = f(0.5 \times 0.5932 + 0.55 \times 0.5912 + 0.6)$

$= f(0.7)$

Sum of the squares of output errors is given by

$e = \frac{1}{2}(T_1 - O_1)^2 + \frac{1}{2}(T_2 - O_2)^2$

$= \frac{1}{2}($

Limitation

- cannot provide multivalued outputs
- cannot be used for back propagation process.

→ Linear:- $f(x) = a$

→ Non-linear Activation Fn

- Sigmoid Fn / Logistic fn :-
  - takes any real values as input & output is in the range 0 to 1
  - The larger the input the output will be closer to 1. whereas smaller the input the output will be closer to

$f(x) = \frac{1}{1+e^{-x}}$

- Most widely used fn.

- Tanh fn :-

  input :- real-valued
  output :- -1 to 1

- ~~ReLU Fn~~ (Rectified Linear

Q. Consider the following network with 2 inputs, 2 outputs & 1 hidden layer.

Sample	Input 1 I_1	Input 2 I_2	Output Target T_1	O/P Target T_2
1	0.05	0.10	0.01	0.99
2	0.25	0.18	0.25	0.79

Back Propagation Algorithm

1. Initialize connection weights with small random values

2. Present the pth sample input vector of pattern $X_p = (x_{p1}, x_{p2} \cdots x_{pn})$ and the output target $T_p = (T_{p1}, T_{p2} \cdots T_{pM})$ to the network.

3. Pass the input from values to the 1st layer for every input node i in layer 0 perform

$$Y_{0i} = p_i$$

4. For every neuron i in layer $j = 1, 2 \cdots M$. Find the output from the neuron.

$$Y_{ij} = \sum_{k=1}^{N-1} Y_{(j-1)k} W_{jik}, \text{ where}$$

$$f(x) = \frac{1}{1 + \exp(x)}$$

5. Obtain output values for every output node i in layer M perform $O_{pi} = Y_{Mi}$

6. Calculate error value δ_{ji} for every neuron i in every layer in backward order $j = m, m-1 \cdots 1$ from output to input layer. followed by the wt adjustments

7. For the output layer the error value is

$$\delta_{Mi} = Y_{Mi}(1 - Y_{Mi})(T_{Pi} - Y_{Mi})$$

hidden layers :- $\delta_{ji} = Y_{ji}(1 - Y_{ji}) \sum_{k=1}^{N_{j+1}} \delta_{(j+1)k} W_{(j+1)k}$.

8. The weight adjustment can be done for every connection from neuron k in layer j-1 to every neuron j in every layer i

$$W_{jik} = W_{jik} + \eta \delta_{ji} Y_{ji}.$$

- η is the learning rate normalized b/w 0 & 1.

9. Steps through 8 will be repeated for every training sample pattern p & repeated for those sets until the sum of the squares of the errors is minimized.

Activation fn decides whether neuron should be fired or not

Determined to fire the output.

Linear Non-linear

Linear Fn

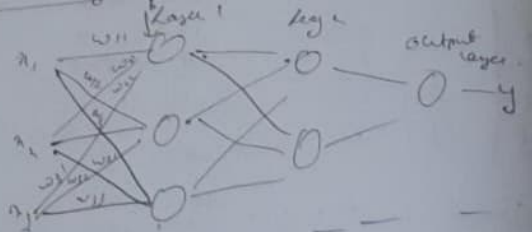
Binary → Step function :-

Depends upon the threshold value.

$f(x) = 0$ for $x < 0$

1 for $x \geq 0$

Multilayer Perceptron



→ Feedforwarded network

→ Network which allows signals to travel in both directions using loops are called recurrent networks.

Number of nodes

→ No of input nodes – no of features in input data

→ No of output nodes – no of outcomes to be modeled or the no of class levels in the outcome

→ No of hidden nodes – decided prior to training the model

→ ~~Appropriate no depends on no of inputs~~

Training Algorithm

→ 2 algos are used for training a single perceptron

- Perceptron rule →
  used when the training data set is linearly seperable
- Delta rule :-
  used when the training data set is not linearly seperable

→ The algorithm which is now commonly used train an ANN is known as back propagation

→ Cost Function
- a fn that measures how well the algo maps the target fn that it is trying to guess or a fn that determines how well the algo performs in an optimization problm

Back propagation

- Initially the wts are assigned at random
- Then the algo generates through many cycles of two process

OR.

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

$w_1 = 0.6$
$w_2 = 0.6$
$\eta = 0.5$
$t = 1$

$\sum x_i w_i = 0 + 0 = 0 \geq 1 \Rightarrow 0$

$\sum x_i w_i = 0 + 0.6 = 0.6 \geq 1 \Rightarrow 0$

$y \neq t$

$\Delta w_2 = 0.5(0-1)1$

$= -0.5$

$w_2 = 0.6 + 0.5 = 1.1$ //

$\Delta w_1 = 0.5(0-0)1$

$= 0$ //

$w_1 = 0.6$

$\sum x_i w_i = 0 + 1.1 \geq 1 \Rightarrow 1$

$\sum x_i w_i = 0.6 + 0 \geq 1 \Rightarrow 0$

$\Delta w_1 = 0.5(0-1)1$

$= -0.5$

$w_1 = 0.6 + 0.5 = 1.1$ //

$w_2 = 0 + 1$ //

$\sum x_i w_i = 0 + 1.1 \geq 1 \Rightarrow 1$

$\sum x_i w_i = 1.1 + 0 \geq 1 \Rightarrow 1$

$\sum x_i w_i = 1.1 + 1.1 \geq 1 \Rightarrow 1$

$$\sum_{i=1}^{n} w_i x_i \geq \theta$$

$$\sum_{i=1}^{n} w_i a_i - \theta = 0$$

$$\sum_{i=0}^{n} w_i a_i = 0$$

$-\theta \rightarrow$ bias

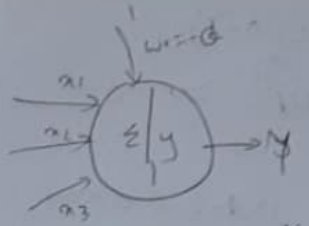
Assignment

Ps

1. Write a python pgm to demonstrate single linear regression and logistic regression on a data set

2. Design a multilayer perceptron for the given binary classification problm

XOR fn in which the 2 pts $(0,0) \& (1,1)$ belong to 1 class and other pts $(0,1) \& (1,0)$ belongs to the 2nd class.



$$y = 1 \quad \sum_{i=1}^{n} x_i w_i \geq \theta$$

otherwise $y = 0$

1. Initially assign random weights
2. Iterate and check if $y = t$ ($t \rightarrow$ target value; $y \rightarrow$ predicted output)

3. If $y \neq T$ then change the weights $w_1 = w_1 + \Delta w_i$

$\Delta w_i = \eta (t - y) x_i$

AND fn

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

$w_1 = 1.2$
$w_2 = 0.6$
threshold : 1
$\eta = 0.5$

$\sum x_i w_i = 0 \times 1.2 + 0 \times 0.6$

$= 0 \not\geq 1 \Rightarrow 0$

$y = t$

$\sum x_i w_i = 0 \times 1.2 + 1 \times 0.6$

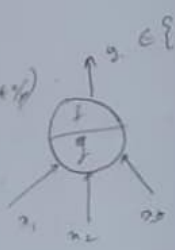
$= 0.6 \not\geq 1 \Rightarrow 0$

$y = t$

Initial structure of neuron

- Mc-Culloch Pitts Neuron (MP Neuron)

- Simplified model
- No wt -
- for boolean values (0/F & 1/T)



- $g = \sum_{i=1}^{n} x_i$ (aggregation fn)

- Inhibitory Input :- Irrespective what the other inputs are, the decision is made by this particular input.

- Excitatory Input :-

Aggregate of inputs taken to make a decision

or

$y = 1$ if $\sum_{i=1}^{n} x_i \geq \theta$, $g(n) \geq \theta$ → threshold

$y = 0$ if $\sum_{i=1}^{n} x_i < \theta$, $g(n) < \theta$

$g(n) \geq \theta$

AND

$\sum_{i=1}^{n} x_i \geq \theta$

≥ 2

$\theta = 2$

if $n = 3$

$\theta = 3$ ...

OR

$\theta = 1$

NOT

$\theta = 0$

Disadvantage

- Used only for boolean values
- linearly separable prbms

Perceptron

if $y = 1$

$\sum_{i=1}^{n} x_i w_i \geq \theta$

Dendrites : Input
Cell body : Processor

Synapse : output

Once inputs exceeds a critical level, the neuron discharges a spike - an electrical pulse that travels from the body, down the axon.

ANN

— consist of pool of simple processing units

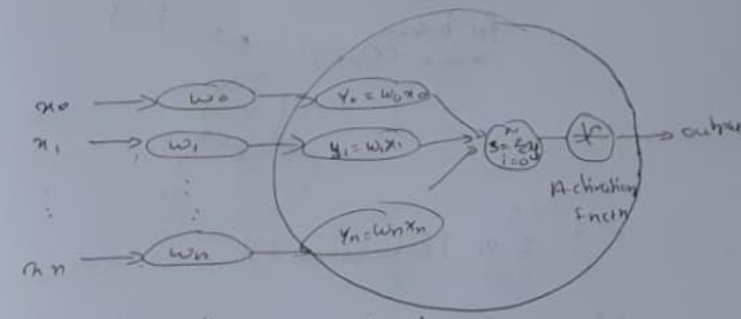
→ A set of major aspects of parallel distributed model include:

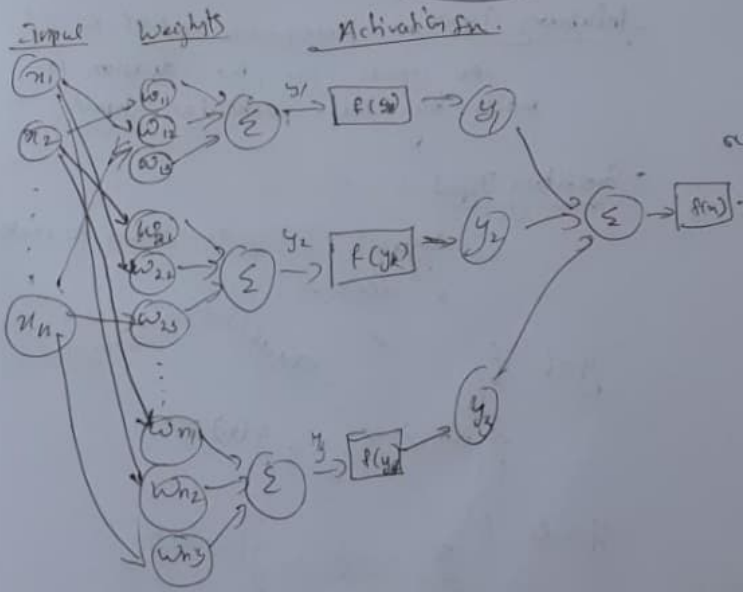
- A set of processing units
- A state of activation for every unit, which equivalent to the output of the unit.
- connections b/w the units. Generally each connection is defined by the weight.
- a propagation rule, which determines the eff. input of a unit from its external inputs.
- an activation fn, which determines the new level of activtn based on effective input

Computers vs Neural Networks

— One CPU
— fast processing units
— reliable units

- parallel processing





Weighted avg precision = (Actual A class instance × Precision of class A + Actual class B instance × Precision of class B + Actual class C instance × Precision of class C) / no. of observation

$= \left(\frac{20 \times 5}{8}\right) + (30 \times 0.75) + \left(\frac{50 \times 45}{56}\right)$

$= \frac{75}{100} = 0.75$

Average Recall : (Actual class A instance × Recall class A + Actual class B instance × Recall class B + Actual class C × Recall) / Total no. of ins.

$= 0.18$

Q. We have a true class classifier, the confusion matrix is given as

Actual \ predict	TP	FN
	1484	447
	FP	TN
	336	107

Q. Suppose you working on a ~~face~~ spam detection. The pblm is formulated as a classification task where spam is +ve and non-spam is -ve class.

The training set contains 1000 emails 99% of which are not spams and 1% are spam.

a) What accuracy has the classifier that always predicts non-spam.

b) The fraction of spam mails that are correctly classified is measured by recall value. What is the recall of always non-spam classifier.

Artificial Neural Networks

→ The most fundamental unit of deep neural network is an artificial neuron.

Neurons

→ Dendrite :- receives signals from other neurons
→ Synapse : pt of connection to other neuron
→ Soma : processes the info.
→ Axon : transmits the output of this neuron

Ordinal Logistic Regression :-

" " " with some order

Age	Wt
22	72
25	64
47	52
52	78
46	61
55	58
60	49
62	55
28	70
27	63
29	60
40	51
45	58

Flow of Batch Machine Learning

Given: labeled training data, $X, Y = \{\langle x_i, y_i \rangle\}^n$

Assumes each $x_i \sim D(X)$ with

$y_i = f_{target}(x_i)$

Train the model :-

model ← classifier.train(X, Y)

Apply the model to new data:

- Given new unlabeled instance $x \sim D(X)$

prediction ← model.predict(x)

Q Calculate the regression coeff & obtain the lines of regression for the following data

x	y	xy	x^2
1	9	9	1
2	8	16	4
5	10	50	9
4	12	48	16
5	11	55	25
6	13	78	36
7	14	98	49
28	77	334	140

$a = 101/63 = 0.57$

$b = 6.87$

(relation between person & movie they saw)

person	xi	yi
1	4	3
2	2	4
3	3	2
4	5	5
5	1	3
6	3	1

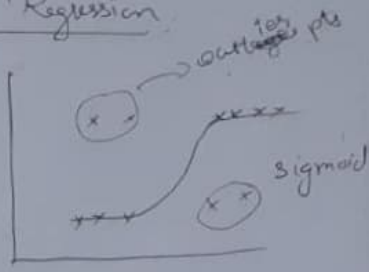
i) Find the values of a & b w.r.t linear regression model which best fits the given data

ii) Find regression line that fits best for the given sample data.

iii) Interpret & explain the eqtn of regression line

iv) If a new person rates the movie 1 as 3 then predict the rating of the same person for movie 2.

Logistic Regression



$$y = P(x) = \frac{1}{1+e^{-z}} \Rightarrow z = -\log\left(\frac{1}{y} - 1\right)$$

Logistic Regression is a classification algorithm which is used when the variable is categorical in nature (in classes)

The main objective is to find the relationship b/w features & probability of a particular outcome

Binary logistic Regression - 0 & 1 - interclass (not interclass)

Multinominal logistic Regression - The target can have 3 or more possible values without any order.

Q8: Consider the following set of points

$\{(-2,-1), (1,1), (3,2)\}$

Find the least square regression line for the given data points

x	y	xy	x^2
-2	-1	2	4
1	1	1	1
3	2	6	9
2	2	9	14

$$a = \frac{3 \times 9 - 2 \times 2}{3 \times 14 - 4}$$

$$= \frac{27 - 4}{38}$$

$$= \frac{23}{38}$$

$$b = \frac{4}{3} \cancel{(0) = 0} \quad \frac{3}{9}$$

Generalization

- How well our hypothesis will correctly classify future examples that are not part of the training set

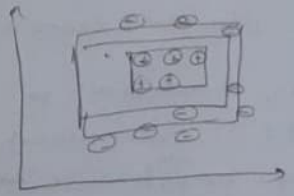
→ **Most specific hypothesis (S)**

- the tightest rectangle that includes all the positive examples & non of the -ve examples

→ **Most General hypothesis (G)**

- largest rectangle we can draw that includes all the positive examples & non of the -ve examples.

→



A mode is said to be a good machine learning model if it generalizes any new input data from the problem domain in a proper way.

Reasons for poor performance

- Overfitting
- Underfitting

→ **Training error**

- Error measure on Training data
- This must be minimized

→ **Test Error**

- Expected value of the error on new input
- Expectation is taken from across diff. possible inputs drawn from the distribution of inputs

→ Generalization error is estimated by measuring its performance on a test set of examples.

→ Data generating process

- probability distribution over datasets
- Assumptions
  - Examples in each dataset that are independent from each other.

Bias.

- Assumptions made by a model to make a function easier to learn.
  - Low error rate of train data.
  - High error rate - high bias.
  - Low error rate : Low bias.

Variance

- diff b/w error rate of training data & testing data
- low variance is good for generalized model.