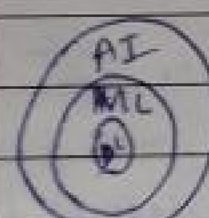
Date ______
Page ______

Machine Learning :- It is a boom in computer world.

Machine Learning is a set of data which make a machine to perform some specific task by using best suited algorithms.

In today's world, we found machine with limited task having.

ML is ⊂ AI which provide machines the ability to learn the action ...

→ 

Machine Learning :- "The field of study that gives computers the ability to learn without being explicitly programmed."

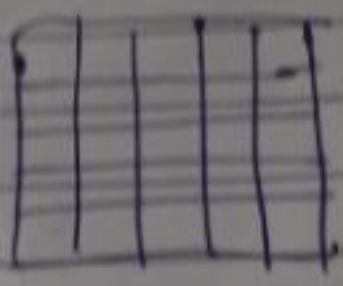
A computer program is said to learn from experience E, with respect to some task T, and some performance measure P, if its performance on T, as measured by P, improves with experience E.

Machine Learning :-

→ Well-defined Learning Problem :-

→ The class of tasks,
→ The measure of performance to be improved
→ The source of experience.

Example:- 1 checkers learning problem:

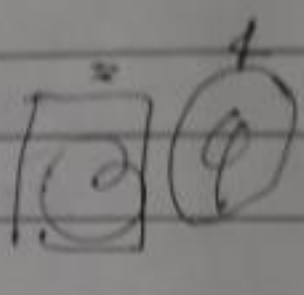


Find :-
T → To move /play the checker

E → where the move happen previously

P → win or loose

Find:
T → To detect the ~~to~~ digit



E → Have ~~to~~ previously digits ~~or~~ use detected.
→ a database of hand written words with given classification

P → percent How much correct digit detected

Find

T → ~~Find~~ robot drive the car "driving acar"

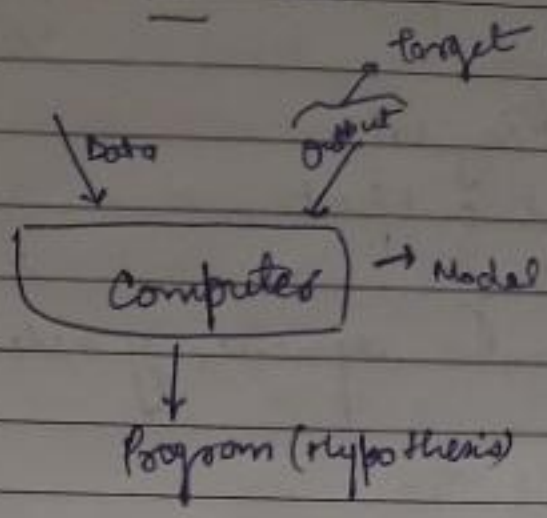
E → from dataset of how do drive acar or how ~~many time~~ robot drive previously.

P → percent of driving vehical successfully.

A sequence of image & steering commands recorded while observing a human driver.

Date
Page

ML cole



→ features to identify objects :-
- * Shape
- * Color
- * texture

* Supervised → when we have a target value.

* Unsupervised → when the result is uncertain.
i.e. no fix target.

⇒ Where ML use :-
- → when human experties does not exist
- → when we work with big data
- → Model must be customized.

Supervised (inductive) learning
- Given training data + desired output.
- → desired outputs (labels)

Unsupervised learning
- → Given training data
- → without desired output.

Semi supervised learning
→ Given : training data
→ few desired output.

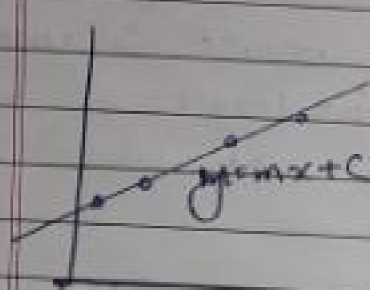
Reinforcement learning
→ Rewards from sequence of actions.

⇒ REGRESSION {Supervised}

→ Given $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
→ Learn a function $f(x)$ to predict y given x - y is real valued =.= regression.

for $x \to y$, y ∈ continuous values

$y = f(x)$
↓
to find without error.



Linear Regression

Polynomial regression

→ Regression gives continuous value as a target value

Classification

y ∈ discrete value, it is categorical

→ Learn a function $f(x)$ to predict y given x - y is categorical =.= classification

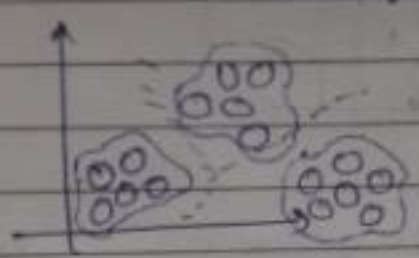
→ Regression gives descrete value as a target value.

- It depend on target value.

Unsupervised learning

- Given $x_1, x_2, \ldots x_n$ (without labels)
- Output hidden structure behind the x's
  - Clustering {Intensic}

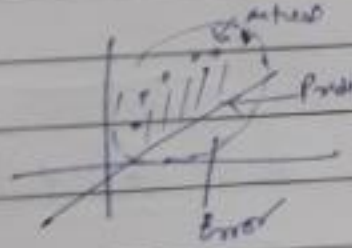
$x \rightarrow$ Astronomical data



Reinforcement learning

- Agent do action on rewards
- Agent also got penalty for wrong work.

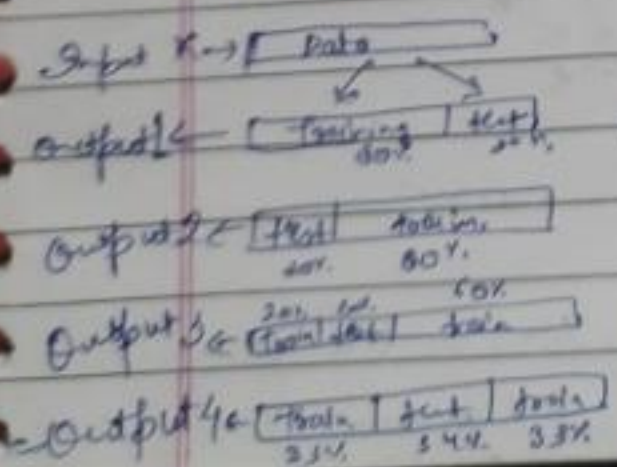
⇒ Bias :- Difference b/w predicted value & actual value i.e. Error.



High bias because error is underfit high.

Model Underfit ⇒ Sum of Error is high...

Model Overfit ⇒ Sum of error is low...



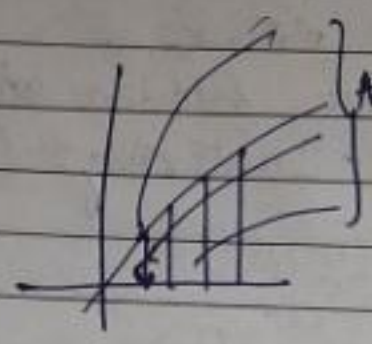
babba

Date
Page

image

Variance → difference in results.

Inductive bias → set of Assumption to predict the target



Choose a model (Assumptions)

x	y
2	3.4
3	5.9.
5	7.8
7.8	6.5
9.2	11.4
10.4	15.3
11.8	17.6

Model assumption

1- $y = bx + a$
2- $y = e^{-bx}$
3- $y = \sin(bx)$
4- $y = bx^2$
5- $y = \sqrt{abx}$

Ex
Let plot be of size price
$1m^2 = 1000 Rs$
$2m^2 = 2000 Rs$

Inductive bias also called learning bias

Inductive bias = $a + bx$.
↓ ↙
infinite

Best fit ⇒ Minimum error.

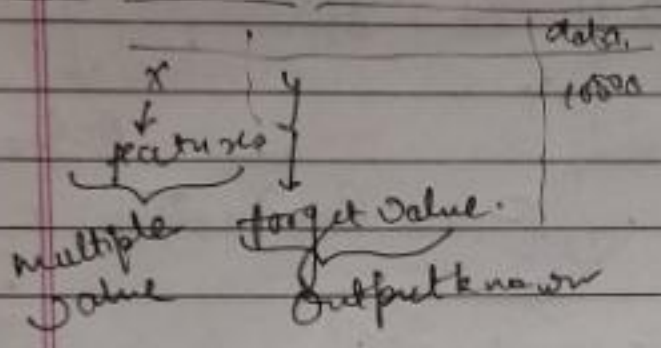
Hypothesis space → when there are no. of spaces set of possible or legal hypothesis.

Date
Page

Best solution = Hypothesis.

→ Bias - Error in training data
Variance - Error in test data.

For Best optimal result we should take care for low Bias & low Variance.



LINEAR REGRESSION WITH ONE VARIABLE →
{finding relationship b/w dependent & independent Variable}
of best fit line

$y = f(x)$

Dependent value. ← independent value

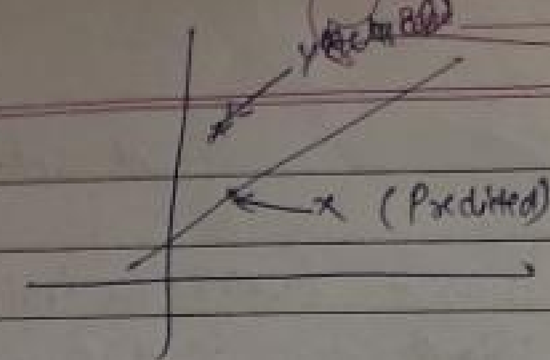
Best fit → Minimum error.

ERROR = (Actual data) − predicted data.
Given data
$e = y$ (actual) $- \hat{y}$ (predicted)

HYPOTHYSIS

m = no. of training examples
x's = "input" variable / features
y's = "output variable / target.

Date ______



Training set

↓

Learning Algo

↓

X: Size of ~~price~~ house → h (Hypothesis) → Estimated price Y

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

$h_\theta(x)$ → Predicted

Y → Actual

θ_i's: Parameters

$y = c + mx$

$J(\theta_0, \theta_1) \Rightarrow$ Actual - predicted.

Costfunction

square make absolute value

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=0}^{m} (h_\theta(x^i) - y^i)^2$$

(predicted: $h_\theta(x^i)$; actual: y^i; m: training set)

or called squared error function.

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

We set as data whose cost size is minimum to get best fit hypothesis.

single variable

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

Parameter → θ_0, θ_1

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

Goal → $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

ML Log i-

Training set { Dataset contain features }

Dataset → features
→ Output / label

Training set has both features & output
Testing set has only features. & we get output & we check accuracy.

Numpy → Library having inbuilt function for array related...

Libraryname
import 'numpy' as np → Objectname
importing library / variable name
a = np.array [1, 2, 3, 4] → Data
print (a) → Output command.
Typeof datastructure

ML
- SupervisedLearning
  - KNN
  - RandomForest
- Unsupervised

KNN - { K-NearestNeibours }

Distance = $\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$

calculate the higher frequency value & return predicted value

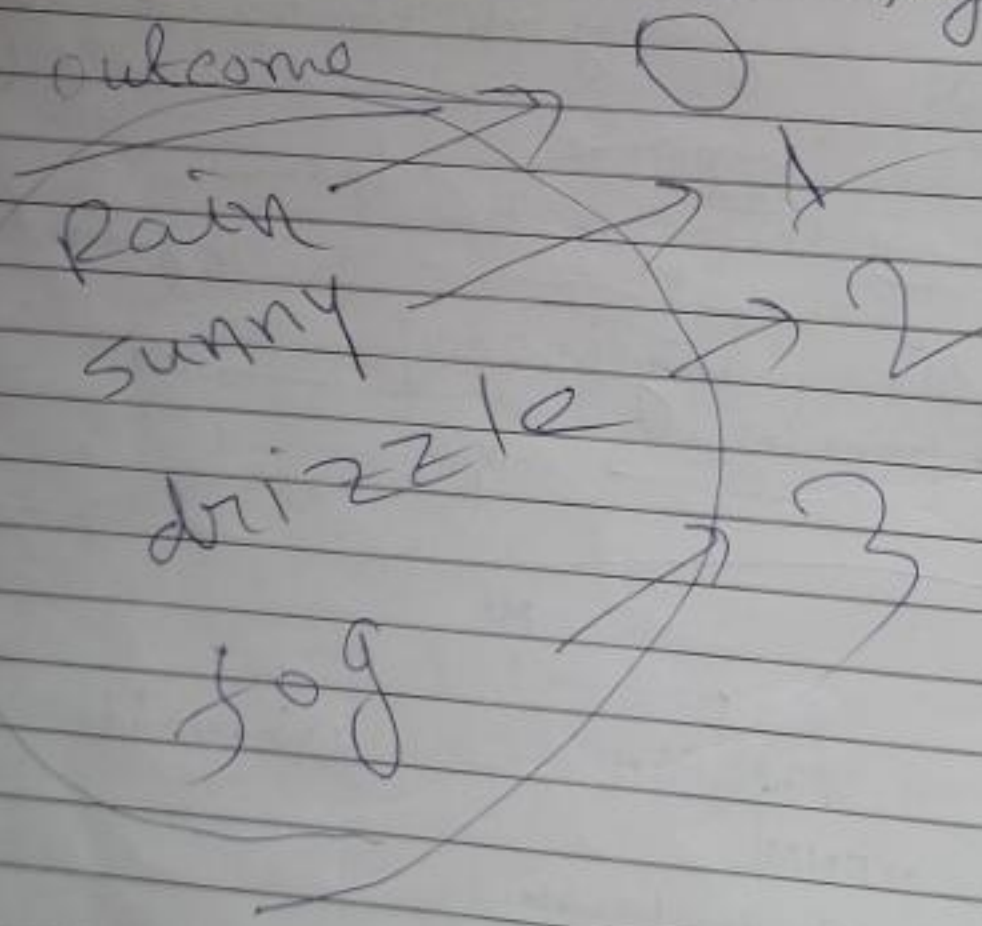
Date
Page

⇒ Kaggle → Use to get dataset :-

⇒ to read file - pd.read_csv(r"C:\...")

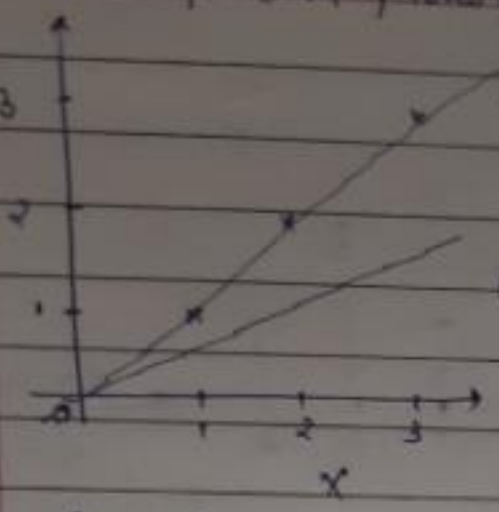
Ex⇒ dataset = pd.read_csv(r" address ")
dataset
// Display the data

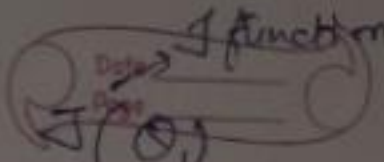
dis = dataset.columns → to select columns
print(d⇒)
Index(['PM2.5', ...], dtype='object')

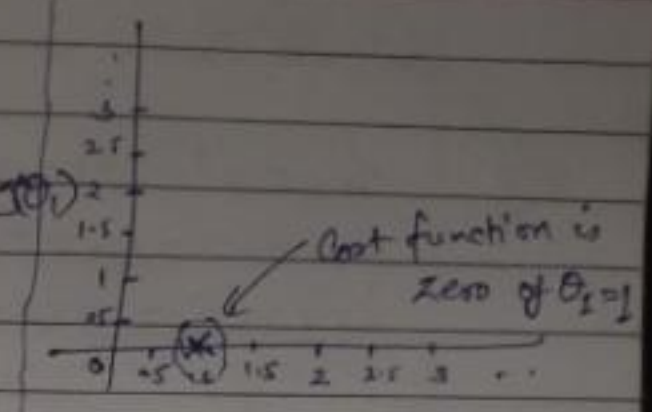


$h_\theta(x)$

for fixed θ_1, this is a function of x







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

$\theta_1 \downarrow$

$J(1) = 0$

$\theta_1 = 1$

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

$J(\theta_1) = \frac{1}{6} \times 0 = 0$ Cost function = 0 means, best fit

LINEAR

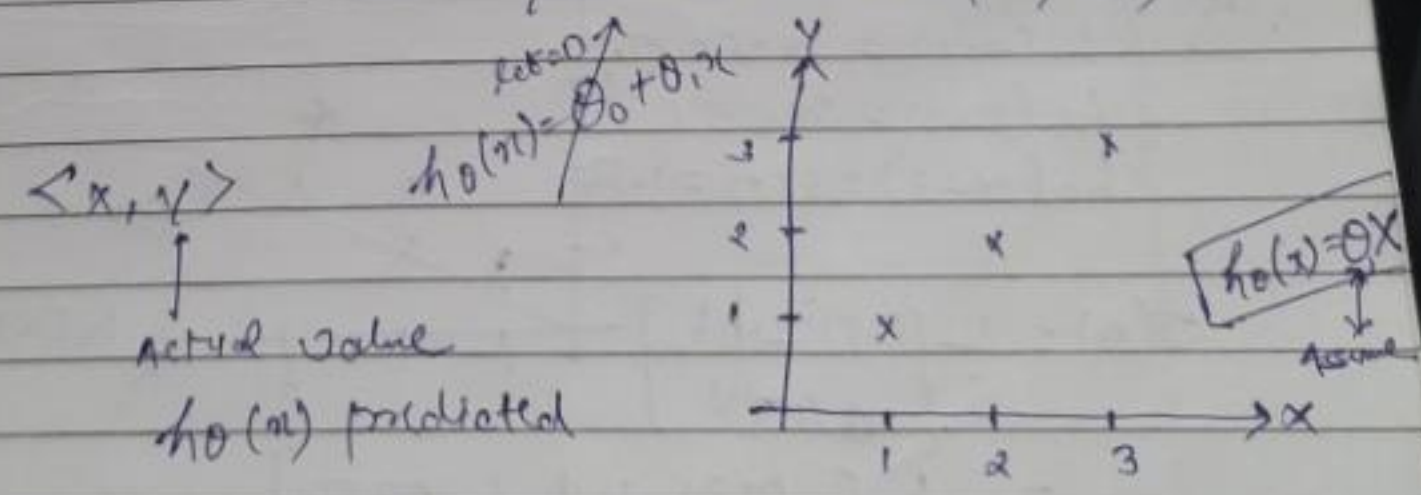
⇒ Hypothesis ① $h_\theta(x) = \theta_0 + \theta_1 x$

② θ_i parameter $i \in \{0, 1\}$

③ Cost function

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

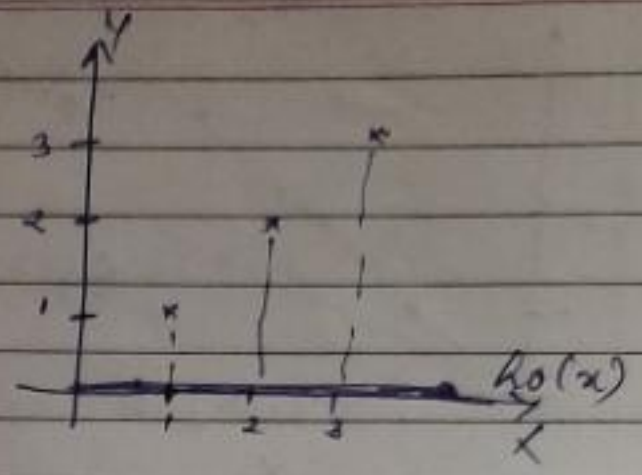
④ Goal Minimize $J(\theta_0, \theta_1)$



* Cost function → Error b/w cohole data set.
* Logs function → Error b/w single data point

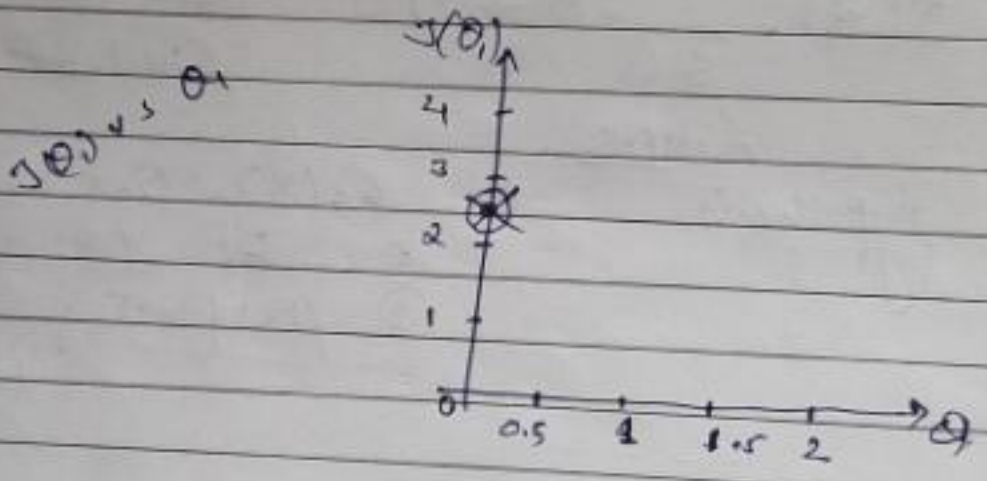
Hypothesis ⇒ for $\theta_1 = 0$

$h_\theta(x=1) = 0 \times 1 = 0$
$h_\theta(x=2) = 0 \times 2 = 0$
$h_\theta(x=3) = 0 \times 3 = 0$



$$J(\theta_1) = \frac{1}{2\times 3}\left[\begin{matrix}(1-0)^2 + (2-0)^2 + \\ (3-0)^2\end{matrix}\right] \quad h_\theta(x) = \theta_1 x$$

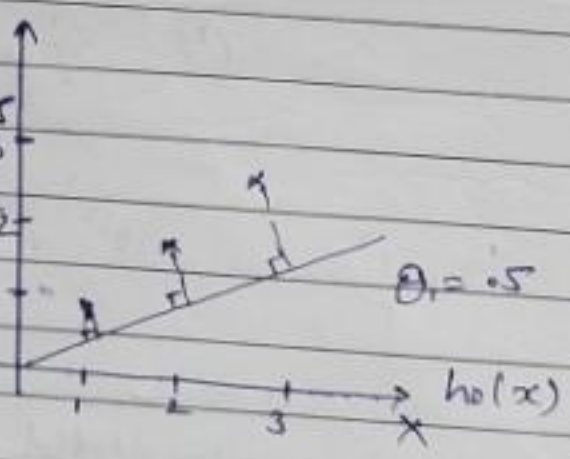
$$J(\theta_1) = \frac{1}{6} \times 14 \Rightarrow \frac{7}{3} = 2.\bar{3}$$



Hypothesis ⇒ 2

for $\theta_1 = 0.5$

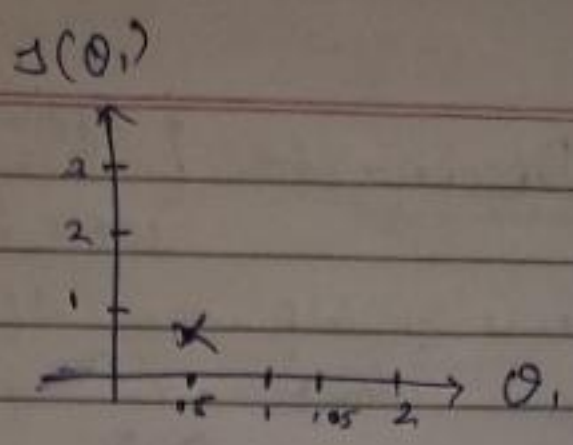
$h_\theta(x=1) = 1 \times .5 = 0.5$
$h_\theta(x=2) = 2 \times .5 = 1$
$h_\theta(x=3) = 3 \times .5 = 1.52$



$$J(\theta_1) = \frac{1}{6}\left[\begin{matrix}(.5)^2 + 1^2 + \\ (1.5)^2\end{matrix}\right]$$

$$= \frac{1}{6}\left[0.25 + 1 + 2.25\right]$$

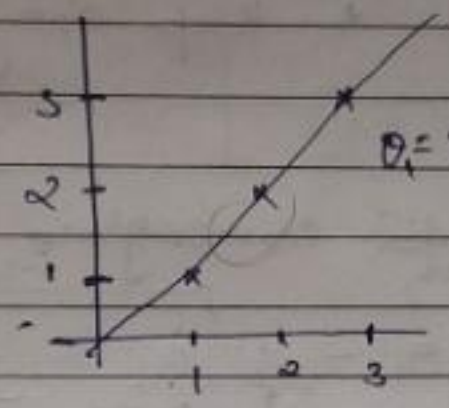
$$\rightarrow \frac{1}{6} \times 3.5 \Rightarrow \frac{3.5}{6} \Rightarrow .58\bar{3}$$

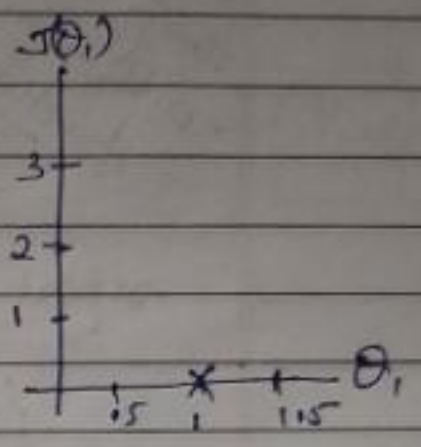


Hypothesis :-

for $\theta_1 = 1$

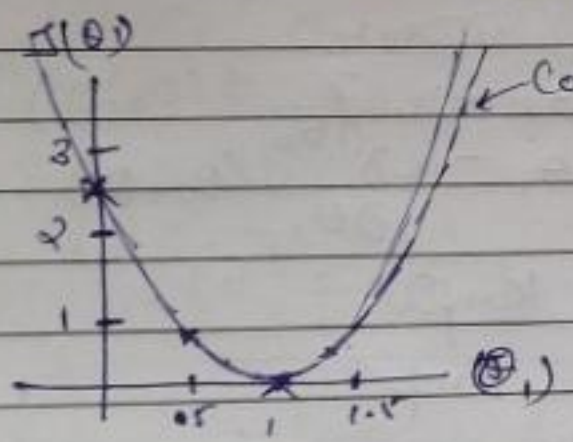
$J(\theta_1) = 1$





Now

Combine Cost function graph:-



⇒ * the best fit hypothesis is of $\theta_1 = 1$ because it is the most bottom point (min) of θ_1 V/s $J(\theta_1)$ graph.

* Also the error is zero.

Date ______
Page ______

GRADIENT DESCENT :- { optimization technique to find out best parameter w.r.to minimum cost function }

→ Have some function $J(\theta_0, \theta_1)$

Want $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$

Alg for finding best parameter with min. cost function.

- Start with θ_0, θ_1
- Keep changing θ_0, θ_1

down to the slope to reach bottom.

repeat until convergence

$$\{ \theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \quad \text{for } j=0 \text{ \& } j=1$$
$$\}$$

(value; change; α ↑ Learning Rate)

correct : Simultaneous update

$$temp0 := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$
$$temp1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$

$$\theta_0 := temp0$$
$$\theta_1 := temp1$$

→ Learning Rate { generally optimal i.e. neither big value nor small value }

→ baby step to find bottom line.

eqn of line Hypothesis $h_\theta(x) = \theta_0 + \theta_1 x$

for $j = 0, 1$

for 2D it is line; for 3D it is a plane

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

Linear Regression with Multiple Variables →

nD > 3D with multiple plane :- Hyperplane

Parameter matrix $\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_{n-1} \\ \theta_n \end{bmatrix}$ $X = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix}$ forall $x_0 = 1$

features

for Multiplication change order :-

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

$$h_\theta(x) \; [\theta_0 \; \theta_1 \; \theta_2 \cdots \theta_n] \times \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$\boxed{h_\theta(x) = \theta^T . X}$$

for one features $h_\theta(x) = [\theta_0 \; \theta_1] \begin{bmatrix} 1 \\ x_1 \end{bmatrix}$

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

$h_\theta(x) = \underbrace{\theta^T}_{\downarrow \text{transpose}} . X$ $\quad h_\theta(x) = \theta_0 + \sum_{n=1}^{n} [\theta_n x_n]$

Simple / Single Linear Regression - {Univariate}

Multiple / Multi Linear Regression - {Multivariate}

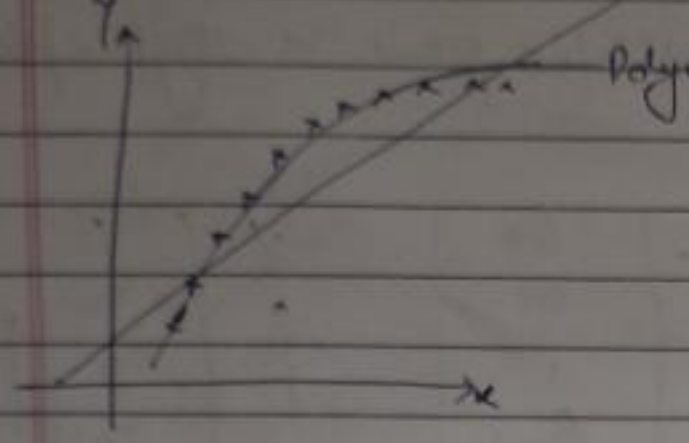
power more than one ...

:- POLYNOMIAL REGRESSION :-

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

Polynomial $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \dots$

Way to get polynomial hypothesis.



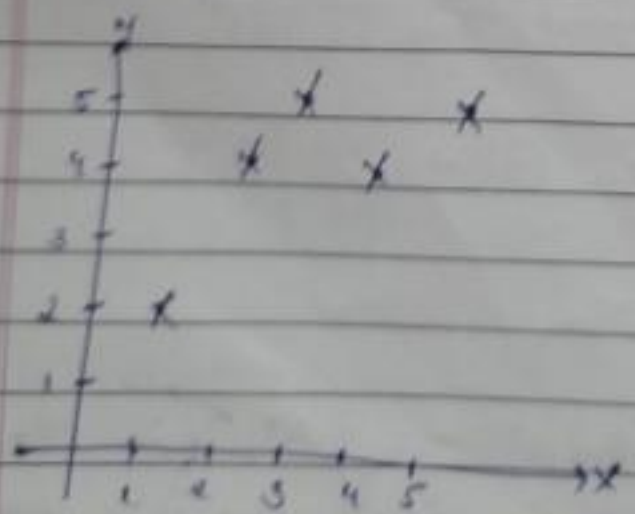
⇒ in hypothesis feature have power greater than 1. (at least one)

Hypothesis for Polynomial regression. (power > 1.)

$$h\theta(x) = \theta_0 + \theta_1 x^2 + \dots$$

Eg. For the given point find best fit. Given $x = 6$

x	y
1	2
2	4
3	5
4	4
5	5



Least square Method :-

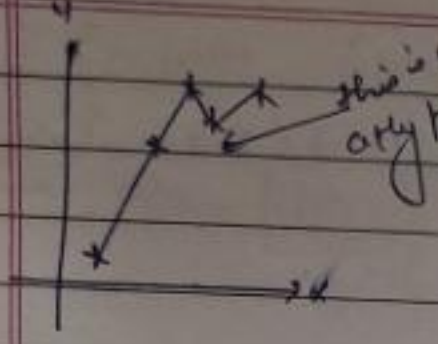
$h_\theta(x) = \theta_0 + \theta_1 x$

$y = mx + c$

$m = \theta_1$, $c = \theta_0$

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

$c = y_1 - m x_1$



Now for single line.

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

$$\bar{x} = \frac{\Sigma x}{n} \qquad \bar{y} = \frac{\Sigma y}{n}$$

$$C = \bar{y} - m \cdot \bar{x}$$

Now calculating
$\bar{x}$, $\bar{y}$ & m & C

$\bar{X} = 3$ $\bar{Y} = 4$

~~$m = \frac{(x \times y)}{y}$~~

$x - \bar{x}$	$y - \bar{y}$	$(x - \bar{x})(y - \bar{y})$	$(x - \bar{x})^2$
-2	-2	4	4
-1	0	0	1
0	1	0	0
1	0	0	1
2	1	2	4
Σ 0	Σ 0	Σ 6	Σ 10

$$m = \frac{6}{10} \Rightarrow 0.6$$

$C = 4 - 0.6 \times 3$

$= 4 - 1.8$

$C \Rightarrow 2.2$

$\boxed{h_\theta(x) = 0.6X + 2.2}$ Hypothesis.

$h_\theta(1) = 0.6 \times 6 + 2.2$

$\Rightarrow 3.6 + 2.2 \Rightarrow 5.8$

Linear target value is continuous

Date
Page

Ex:

Hours spent in Std X	Grade Y	$x-\bar{x}$	$y-\bar{y}$	$(x-\bar{x})(y-\bar{y})$	$(x-\bar{x})^2$
6	82	0.37			
10	88	4.37			
2	56	-3.63			
4	64	-1.63			
6	77	0.37			
7	92	1.37			
0	23	-5.63			
1	41	-4.63			
8	80	2.37			
5	59	0.63			
3	47	-2.63			

What is the line for the given data & predict the grade for those who spent 9 hrs.

Sol: Here $\bar{x} = \frac{52}{11}$ $\bar{y} = \frac{709}{11}$

$\Rightarrow 5.63$ $= 64.45$

$\Rightarrow 4.72$

$y = mx + c$

$y = 6.49x + 33.82$

$y = 6.49 \times 9 + 33.82$

$y \Rightarrow 58.41 + 33.82$

$y \Rightarrow 92.23$

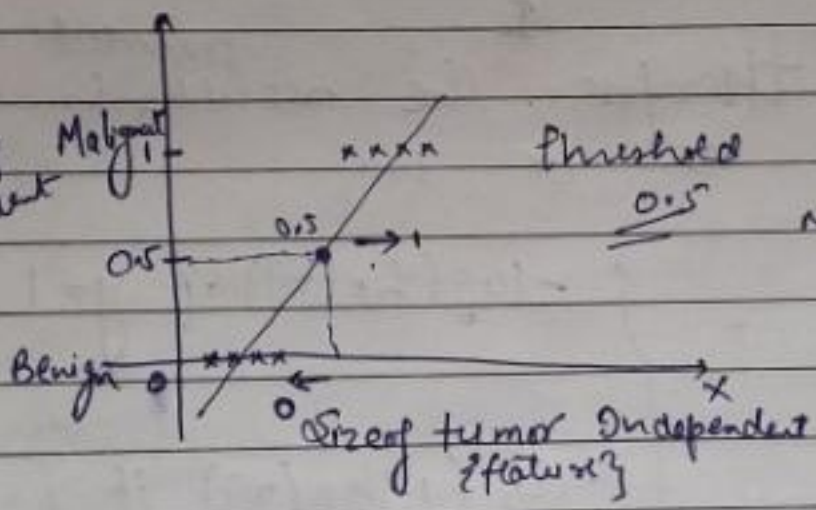
Swiggy

Date
Page

Logistic Regression → {Classification technique}

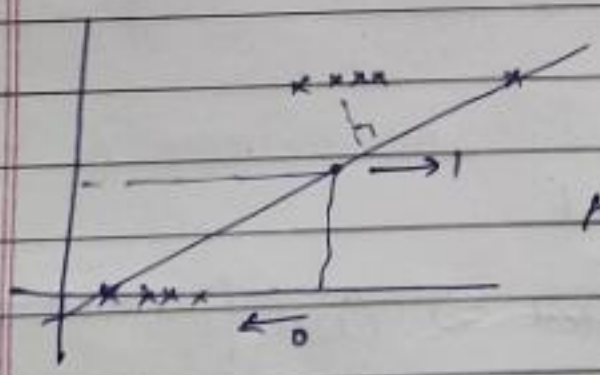
classification
- ↳ Spam filtering — spam/not spam — yes/no, 0/1, Binary, Binary
- ↳ Fraud detection — Fraud / No fraud — Binary
- ↳ Cancer — Malignant / Benign — Binary.

$y \in \{0, 1\}$



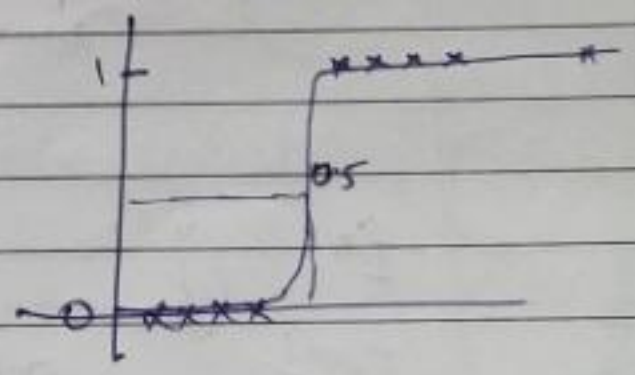
Threshold 0.5

mid point left ⇒ 0 {No}
Mid point right = 1 {Yes}



Heavy Benign
Heavy error

↓ logistic



$$g(z) = \frac{1}{1+e^{-z}}$$ Sigmoid

Hypothesis of logistic regression

$$h_\theta(x) = g(\theta^T x)$$

$$1 \geq h_\theta(x) \geq 0$$

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

Cost function

Here for result there is only two choice either 0 or 1

Therefor the predicted result is either 0 or 1.

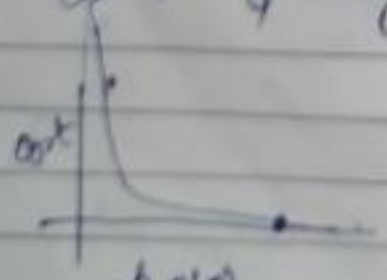
$$\text{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y=1 \\ -\log(1-h_\theta(x)) & \text{if } y=0. \end{cases}$$

↑ Predicted ↑ Actual

Ex $y=1$ $h_\theta(x)=1$

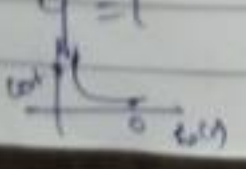
Cost 1 actual = predicted ⇒ Cost function = 0

$\text{Cost}(h_\theta(x), y) = \log(1) = \frac{1}{\log(1)} \Rightarrow 0$



Cost 2 $y=1$ $h_\theta(x) = 0$

$-\log(0) \Rightarrow \frac{1}{\log(0)} = \infty$



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

for $y=1$ putting $y=1$ in

$$\text{Cost}(h_\theta(x)) = -\log(h_\theta(x)) \quad -\text{①}$$

for $y=0$ putting $y=0$ in

$$\text{Cost}(h_\theta(x)) = 0 - (1) \log(1-h_\theta(x))$$

$$\text{Cost}(h_\theta(x)) = -\log(1-h_\theta(x)) \quad -\text{②}$$

⇒ Error is high when we get opposite result.

Logistic { Regression → Because we use the concept of line $h_\theta(x)$
Classification → Because result is either 0 or 1.

GRADIENT DESCENT

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

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

where $J(\theta)$ = Cost function

use to evaluate

Confusion Matrix → ^ for classification model

Evaluation matrix

Size of Matrix ∝ depend on the no. of classes

ex = n = 2

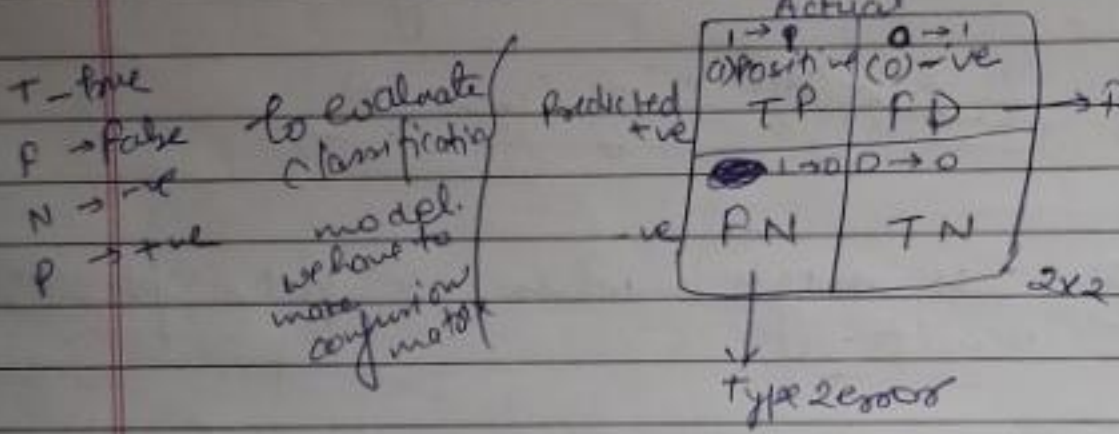
↓

dealing with 2 class

Binary classification

above 2 → Multiclass

→ class is 2 means matrix size 2×2.



① Accuracy → $\frac{TP + TN}{TP + TN + FP + FN}$

② Precision → Actual positive from total +ve.

$\frac{TP}{TP + FP}$

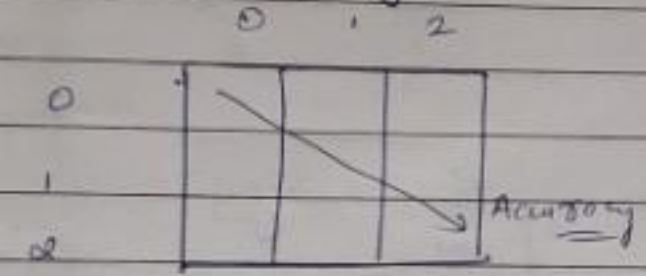
③ Recall → Actual predicted from total ↓

$\frac{TP}{TP + FN}$

(4) F1 score → Harmonic mean of precision & recall.

$$= \frac{1}{\left(\frac{1}{\text{Preci}} + \frac{1}{\text{Recall}}\right)} \Rightarrow \frac{2 \text{ Precision} \times \text{Recall}}{(\text{Precision} + \text{recall})}$$

Ex) For the given values of cell in matrix, calculate all 4 evaluating parameters.



TP	FP
30	30
FN	TN
10	930

$A = \frac{960}{1000} \Rightarrow 0.96 \Rightarrow 96\%$.

$P = \frac{30}{60} \Rightarrow 0.5 \Rightarrow 50\%$.

$R = \frac{30}{40} \Rightarrow .75$

$F1 = \frac{2 \times .5 \times .75}{1.25} \Rightarrow \frac{.75}{1.25} \Rightarrow .6$

there is a huge variance b/w Accuracy & precision because of unbalanced dataset. 30 ⟹ 930

precision → when we check false positive (High FP value)

Recall → when we check FN. (High FN value)

→ For the medical purpose we check recall value ...

KNN - { K-Nearest Neighbors Algorithm }

- Supervised
- Classification
- Similarity or distance function

→ Stores all available cases & classifies new cases based on a similarity measure (distance function).

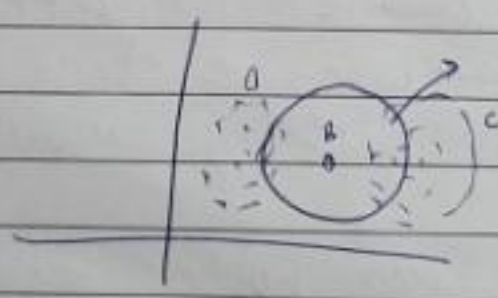
→ Generally used for statistical estimation & pattern recognition.

⇒ if K=1 then assigned to neighbor i.e. min from distance value.

Euclidean function ⇒ $\sqrt{\sum_{i=1}^{k} (x_i - y_i)^2}$

Manhattan ⇒ $\sum_{i=1}^{k} |x_i - y_i|$

Minkowski $\left(\sum_{i=1}^{k} (|x_i - y_i|)^q \right)^{1/q}$



for points are maximum in some distance B go to C

Day →
- → Calculate distance
- → ~~sort~~
- → Get top k
- → Get most frequent
- → Return predicted class.

$\sqrt{4^2+6^2}$
$\sqrt{16+36}$

Date
Page

Q1

Gender	Height (cm)	Distance	Rank		Sorted
M	178	3	2	M-1	
M	174	1	1	M-3	K=3
F	163	12	8	F-4	
F	168	7	6		5
M	181	6	5		6
F	170	5	4		7
M	184	9	7		9
F	171	4	3		12

for feature-1

Predict class of Gender / person whose height (175) where K=3

Distance matrix used is Euclidean & find the distance with other

for K=3 the sorted one is
M - 174 - 1
M - 178 - 3
F - 171 - 4

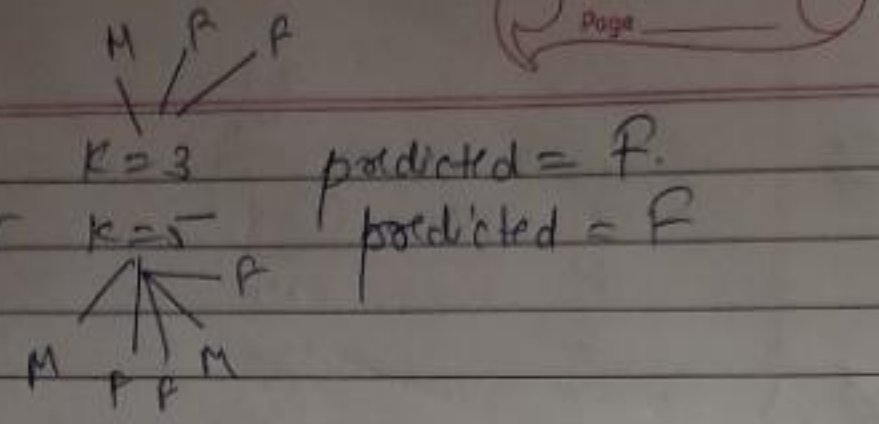
Since Male are in higher frequency therefore

the Gender of person is Male 'M'.

Q1 Q2

for K	Gender	Height (cm)	Weight (kg)	Distance (Q2+4)
	1- M.	178	72	$\sqrt{(170-178)^2+(72-70)^2}=\sqrt{64+4}=\sqrt{68}$
to	4 M	174	81	$\sqrt{137} = 11.70$
feature	7 F	163	55	$\sqrt{274} \Rightarrow 16.55$
= 5	f	168	58	$\sqrt{148} \Rightarrow 12.16$
	8 M	181	90	$\sqrt{905} \Rightarrow 30.08$.
	2- f.	170	70	$\sqrt{100} \Rightarrow 10$
	6 M	184	78	$\sqrt{260} \Rightarrow 16.12$
	3- f.	171	59	$\sqrt{122} \Rightarrow 11.04$

Predict for 170cm & 70kg for K=3 & K=5



for K=3 predicted = F.
for K=5 predicted = F

How we Choose K :-

* K value must be Odd.
* By Elbow Method.

Feature Scaling → we want to normalise the features.
↓
बड़ा को छोटा
छोटा को बड़ा

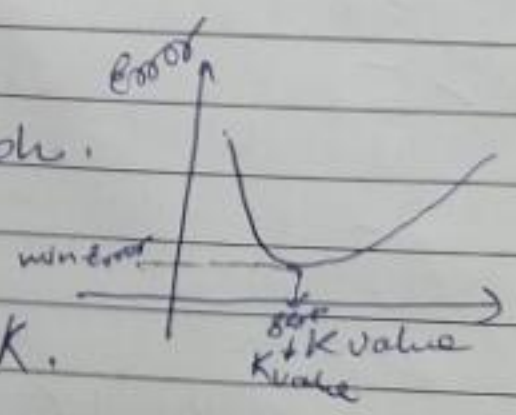
divide the max value with min value.

→ By default euclidean distance.
→ confusion matrix → create matrix
→ Classification-report → Precision
recall
f1 score
support.

ELBOW METHOD

→ Based on the graph.

Lowest point of elbow is best value for K.



Date
Page

NAIVE BAYES ALGORITHM / CLASSIFER

Probability

→ Classification
↓
→ supervised

* It assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

* It is very useful for large data set.

Experiment → Planned activity under certain condition
Sample space → total outcome.
Event → Set of outcome
Exhaustive → at least one event exist
Independent → occurance of one doesn't effect other

$$P(A \cap B) = P(A|B) = P(A) * P(B)$$

CONDitional probability - for a, b already occur.

$$P(A|B) = P(A \cap B) / P(B)$$

BAYES theorem →

$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)}$$

↓
We process new feature and refine our hypothesis at each step...

P(A) is called Prior probability & P(B) is called Evidence.
P(B|A) is called Likelihood and P(A|B) is called posterior probability...

Posterior Likelihood Prior

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Evidence

Probability of A when B is already occur.

"All the features are independent to each other."

$$P(Y|X) = \frac{P(X|Y) \cdot P(Y)}{P(X)}$$

$X = [x_1\ x_2 \dots x_n]$ → Features

→ Independent features.

$P(X) = P(x_1, x_2, x_3 \dots x_n)$

$P(X) = P(x_1) * P(x_2) * P(x_3) * \dots P(x_n)$

Based on conditional independence

$$P(Y|X) = \frac{P(x_1|Y)\,P(x_2|Y)\,P(x_3|Y) \dots P(x_n|Y)\,P(Y)}{P(x_1)\,P(x_2) \dots P(x_n)}$$

finally the Y for which $P(Y|X)$ is maximum is our predicted class.

* the application of Bayes theorem is Naive Bayes classifiers.

PCA - { PRINCIPAL COMPONENT ANALYSIS }

Feature Selection
→ imp features {Property}
↓
When we take subset of feature from whole set.

* Algo work with features subset.

ex Information gain, randomforest

Features Extraction
→ Creation of new feature from the previous/original features.

* Algo work for new features.

* PCA - { A technique for features extraction }
* SVD - { Singular Value Decomposition }

(PCA) Acc increase the no of features, the dimension got increase.

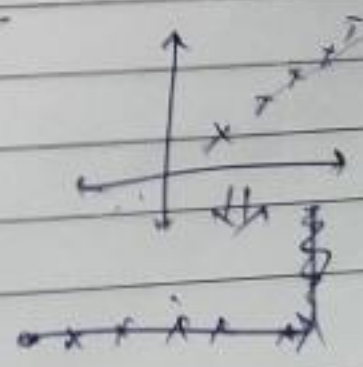
⇒ Dimensionality reduction $r < d$

Remove inconsistencies → Highly correlated data

Redundant data

$x^n \in \mathbb{R}^2$ (2-d)

$\Rightarrow x^n \in \mathbb{R}$ (1-d)



$x^{(i)} \in \mathbb{R}^3$ (3-d)

PCA

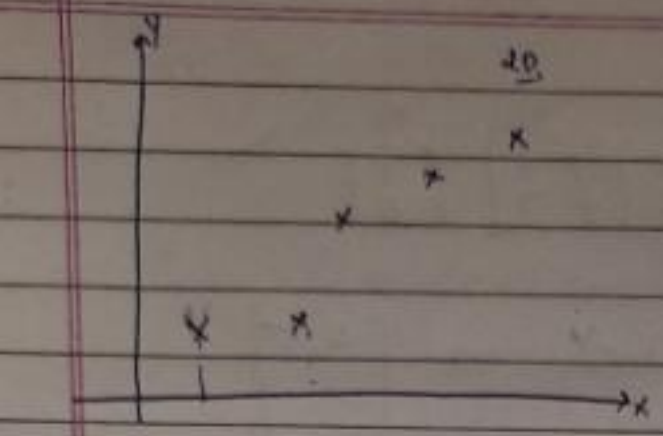
PC_1 PC_2

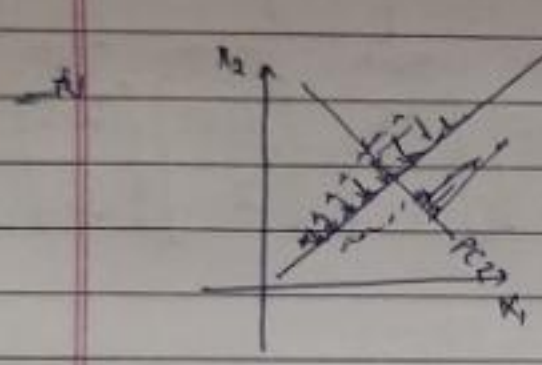
↓

we get all features without having too much loss.

- Standardization of data
- Computing the covariance matrix
  - pxp (symmetric Matrix) $f_1, f_2, f_3 \cdots f_n$ — p dimension
- Calculating the eigenvectors & eigenvalues
  - Principal Components $p = \begin{bmatrix} Cov(x,x) & Cov(x,y) \\ Cov(y,x) & Cov(y,y) \end{bmatrix}$
- Computing the principal components
- Reducing the dimensions of the data

+ve - correlated
-ve - inversely correlated





PC 1
PC 2

n Dimension = n PC

Max Information - PC 1

Ex: Given data
{2, 3, 4, 5, 6, 7 ; 1, 5, 3, 6, 7, 8} compute the PC using PCA algorithm.

⇒ Steps involved in PC Algorithm

① Get data
② Compute the mean vector (μ)
③ Subtract mean from the given data
④ Calculate the co-variance matrix
⑤ Calculate the eigen vectors & eigen values of the covariance matrix
⑥ Choose components & forming a feature vector.
⑦ Deriving the new data set.

Data → 2, 3, 4, 5, 6, 7
1, 5, 3, 6, 7, 8.

Mean = sum of no. / position no.

① vector

$x_1 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ $x_2 = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$ $x_3 = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$ $x_4 = \begin{bmatrix} 5 \\ 6 \end{bmatrix}$ $x_5 = \begin{bmatrix} 6 \\ 7 \end{bmatrix}$ $x_6 = \begin{bmatrix} 7 \\ 8 \end{bmatrix}$

② $M \Rightarrow$ M(mean vector) $\Rightarrow \begin{bmatrix} 4.5 \\ 5 \end{bmatrix}$

③ $x_1 - M$ $x_2 - M$ $x_3 - M$ $x_4 - M$ $x_5 - M$ $x_6 - M$

$\begin{bmatrix} -2.5 \\ -4 \end{bmatrix} \begin{bmatrix} -1.5 \\ 0 \end{bmatrix} \begin{bmatrix} -0.5 \\ -2 \end{bmatrix} \begin{bmatrix} .5 \\ 1 \end{bmatrix} \begin{bmatrix} 1.5 \\ 2 \end{bmatrix} \begin{bmatrix} 2.5 \\ 3 \end{bmatrix}$

④ formula Covariance

$\text{Covariance matrix} \; \frac{\sum (x_i - M)(x_i - M)^T}{n}$

$m_1 = \begin{bmatrix} -2.5 \\ -4 \end{bmatrix} \begin{bmatrix} -2.5 & -4 \end{bmatrix} = \begin{bmatrix} 6.25 & 10 \\ 10 & 16 \end{bmatrix}$

$m_2 = \begin{bmatrix} -1.5 \\ 0 \end{bmatrix} \begin{bmatrix} -1.5 & 0 \end{bmatrix} = \begin{bmatrix} 2.25 & 0 \\ 0 & 0 \end{bmatrix}$

$m_3 = \begin{bmatrix} -0.5 \\ -2 \end{bmatrix} \begin{bmatrix} -0.5 & -2 \end{bmatrix} = \begin{bmatrix} 0.25 & 1 \\ 1 & 4 \end{bmatrix}$

$m_4 = \begin{bmatrix} .5 \\ 1 \end{bmatrix} \begin{bmatrix} .5 & 1 \end{bmatrix} = \begin{bmatrix} 0.25 & .5 \\ .5 & 1 \end{bmatrix}$

$m_5 = \begin{bmatrix} 1.5 \\ 2 \end{bmatrix} \begin{bmatrix} 1.5 & 2 \end{bmatrix} = \begin{bmatrix} 2.25 & 3 \\ 3 & 4 \end{bmatrix}$

$m_6 = \begin{bmatrix} 2.5 \\ 3 \end{bmatrix} \begin{bmatrix} 2.5 & 3 \end{bmatrix} = \begin{bmatrix} 6.25 & 7.5 \\ 7.5 & 9 \end{bmatrix}$

$$m = \begin{bmatrix} 17.5 & 22 \\ 28 & 34 \end{bmatrix}$$

Covariance matrix $= m/n$; $n = 6$

$$M = \begin{bmatrix} 2.92 & 3.67 \\ 3.67 & 5.67 \end{bmatrix}$$

① Eigen value $= \lambda$

$$|M - \lambda I| = 0 \Rightarrow \left| \begin{matrix} 2.92 & 3.67 \\ 3.67 & 5.67 \end{matrix} \right| - \left| \begin{matrix} \lambda & 0 \\ 0 & \lambda \end{matrix} \right|$$

$$\begin{vmatrix} 2.92 - \lambda & 3.67 - 0 \\ 3.67 - 0 & 5.67 - \lambda \end{vmatrix} = 0$$

~~5.67, 2.92 = λ~~

$$(2.92 - \lambda)(5.67 - \lambda) - (3.67)(3.67) = 0$$

$$16.55 - 2.92\lambda - 5.67\lambda + \lambda^2 - 13.46 = 0$$

$$\lambda^2 - 8.59\lambda + 3.09 = 0$$

$$\lambda = \frac{+8.59 \pm \sqrt{(8.59)^2 - 4 \times 3.09}}{2} \quad \frac{8.59 \mp 7.05}{2}$$

$\lambda_1 = 8.22$, $\lambda_2 = 0.38$

Clearly λ_2 is very small so we neglect λ_2.

So, Eigen vector for λ, i.e. for 8.22 value.

features

↓

eigen vector $MX = \lambda X$

$$\begin{bmatrix} 2.92 & 3.67 \\ 3.67 & 5.67 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = 8.22 \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$$

After solving $X_1 = 2.55$

$X_2 = 3.67$

PC_1

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} 2.55 \\ 3.67 \end{bmatrix}$$

Data (

x

1

2.

2

5

8

8

9.

4

$\bar{x} = 146/8$

(X) - R

Yes-1

No-0

No=0

of single →

Matched

Divorc

8 Data (10, 7) K=3

x	y			
1	1	$\sqrt{81+36}$	10.81	A
2	3	$\sqrt{64+16}$	8.94	A
2	4	$\sqrt{64+9}$	8.54	A
5	3	$\sqrt{25+16}$	6.4	A
8	6	$\sqrt{4+1}$	2.14	B
8	8	$\sqrt{4+1}$	2.14	B
9	6	$\sqrt{1+1}$	1.41	B
11	7	$\sqrt{1+0}$	1	B

$\frac{146}{18}$

(10, 7) → B

9

(X) - R-NO 0
M-M 2
T-120K 120
} NO (code)

Yes → Yes-1
No → 0 No→0
Single → 1
Married → 2
Divorce → 3

x	y	z	code	Distance	
1	1	125	0	$\sqrt{27502}$	(2)
0	2	100	0	20	(3)
0	1	70	0	50.-	(7)
1	2	120	0	1	(1)
0	3	95	1	25.2	(5)
0	2	60	0	60	(8)
1	3	220	0	100.--	(10)
0	1	85	1	35.--	(4)
0	2	75	0	45.--	(6)
0	1	90	1	30.--	(9)