Lecture - 26, 27

Types of ML

UNIT-4

Supervised learning

Already've the data & answers (For eg- list of loans takers, defaulters is present and this 'data'll help in knowing that will that person will be able to repay the loan or not?

Unsupervised learning

Have no prior knowledge about the problem, no data nothing. Firstly, it'll try to group all the images together & form a cluster

Reinforcement learning

It has nothing with it & learns from experience. It has no knowledge about any data or info about problem

---

Problem in Machine learning

1) **Classification:** Problems with categorical solution — like Yes / No, True / False, etc (Naive Bayes, Logistic Regression, Decision Tree, Random forest)

2) **Regression:** Problems where continuous value needs to be predicted ("Product Prices", Profits)

3) **Clustering:** Problems wherein the data needs to be organized to find the specific patterns (Product Recommendation)

Decision Trees / Classification Trees. Lecture-28

→ Simplest method for supervised learning. Can be applied to both regression & classification.

→ It is a tree in which each internal node is labeled with an input feature.

→ Formation of tree: 1) The arcs coming from a node labeled with a feature are labeled with each of the possible values of the feature.

2) Each leaf of the tree is labeled with a class or a probability distribution over the classes.

3) Starting from ~~node~~ root node you move to one of next possible nodes.

→ Decision Trees operate in essentially the same manner, with every internal node in the tree being some sort of test criteria.

→ The nodes on the outside the endpoints of the tree, are the labels for the datapoint in question & they are dubbed "leaves".

→ The branches that lead from the internal nodes to the next node are feature or conjunction of feature.

→ The rules used to classify the datapoints are the paths that run from the root to the leaves.

→ Decision tree are often useful when classification needs to be carried out but computation time is a major constraint.

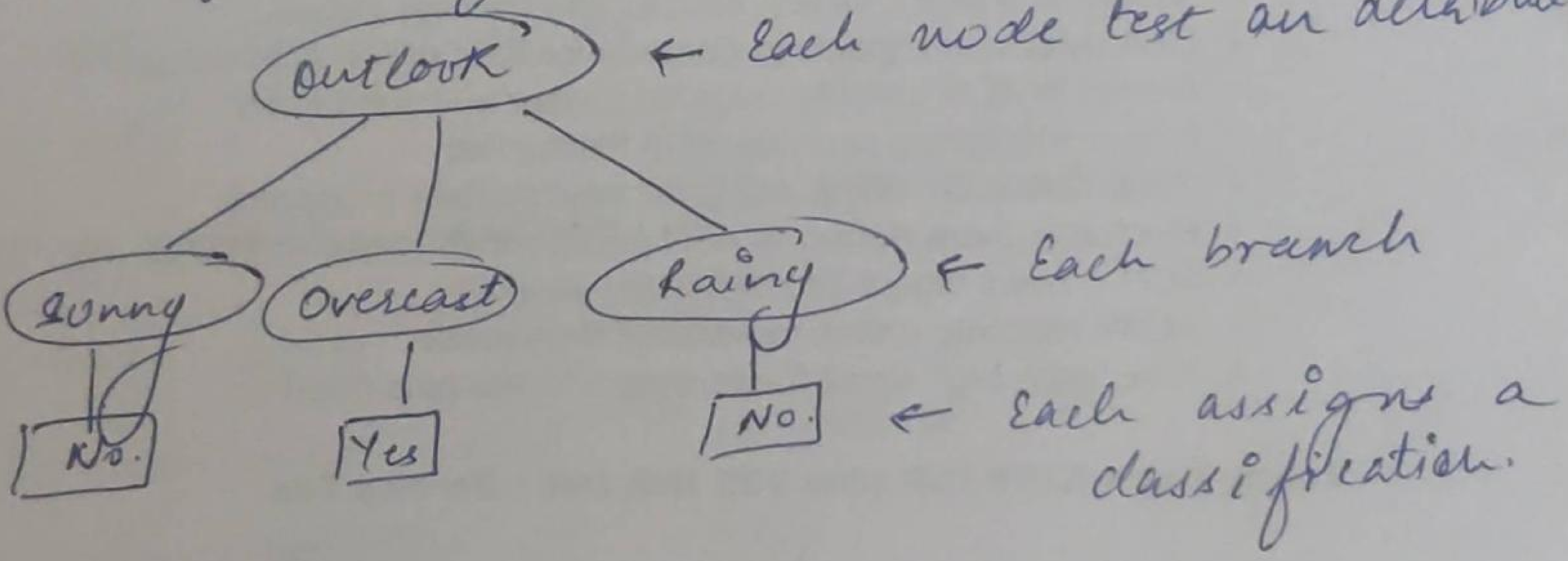
→ Decision Tree can make it clear which features in the chosen datasets wield the most predictive power.

→ Unlike many machine learning algo. where the rules used to classify the data may be hard to interpret, decision tree can render interpretable rules.

→ It represents a function that takes as i/p a vector of attribute values & return a "decision" - a single o/p value. Also a supervised learning.

→ They use both Regression & Classification problems.

→ It performs sequence of test.



Decision Tree Algorithm : ID3 - Iterative Dichotomiser

→ dividing into 2 completely opposite things

→ Most common

→ Like Sunny & Rainy are opposite things.

→ Calculate Entropy & Gains of each attribute. In this way most dominant attribute can be founded. ↳ highest value

This step is performed iteratively.

→ After, the most dominant one is put up on the tree as decision node.

→ Entropy & Gains scores would be calculated again among the other attribute.

→ Process'll continue till it reaches the branch.

→ Calculate the Entropy of every

$$Entropy(S) = \Sigma - P(I) \cdot \log_2 P(I).$$

→ Split S into subset using the attribute the resulting Entropy (after splitting) is min.

→ $Gain(S, A) = Entropy(S) - \Sigma \{ P(S|A) \, Entropy[S|A] \}$

→ Construct a decision tree node that contains an attribute

→ Reuse the subsets using remaining attributes.

Statistical Learning Theory

→ statical learning in AI is a set of tools for machine that uses statistics & functional analysis. In simple words, is understanding from training data & predicting on unseen data. Used to build predictive models based on data.

→ It is one of the most beautifully developed branches of AI. It provides theoretical basis for many of today's ML Algorithm. The theory helps to explore what permits the one to draw the valid conclusions from an empirical data.

→ It begins with a class of hypotheses & uses empirical data to select one out of all of them.

→ It is a set of tools for understanding data. These come under supervised & unsupervised learning.

- supervised → predict or estimate an o/p based on 1 or more inputs
- unsupervised → provides a relationship or find a pattern within the given data without a supervised output

→ Suppose, response Y and p different predictors, $X = (X_1, X_2 \ldots X_p)$

$$Y = \underbrace{f(X)}_{\text{unknown fn.}} + \varepsilon \rightarrow \text{random error}$$

→ Thus, statistical learning refers to a set of approaches for estimating f.

→ Now, in cases where we've set of X readily available but the output Y not so much, the error averages to zero.

$\hat{Y} = \hat{f}(X)$ → estimate of f.
↓
resulting prediction

→ So, for set of predictors X,

Expected value of squared difference b/w actual & expected result ← $E[Y-\hat{Y}]^2 - E[f(X) - \varepsilon - \hat{f}(X)]^2$

$$\Rightarrow E(Y-\hat{Y})^2 = \underbrace{[f(X)-\hat{f}[X]]^2}_{\text{Reducible Error}} + \underbrace{Var(\epsilon)}_{\text{Irreducible Error}}$$

Reducible Error. (we can improve the accuracy of f by better modeling.)

Irreducible Error] (No, matter how well we estimate f, we cannot reduce the error introduced by variance in ε).

→ Some of the algorithms which uses Statistical learning models - Regression, Classification, Density Estimation, etc..

———— x ———— x ————

Lecture 30 & 31

→ Regression & Classification Problem :

↓ Estimating Quantitative responses with the help of quantitative variables take on numerical values - age, height, income, price, and much more.

↳ Qualitative variables take on categorical values - gender, brand, parts of speech, etc. Estimating Qualitative responses are classification problems.

→ Variance & Bias :→ It refers to the error introduced by approximating a real-life problem which may be extremely complicated by simpler model.

→ Generally, when we over-fit a model on given dataset it results in very less bias.

↓ Amount by which 'f' would change if we estimated with diff. training dataset we get a model that has higher variance since any change in the data point would results in a different model.

→ Linear Regression : It is used for solving Regression problem whereas at times predict the continuous dependent variable with the help independent variables.

→ The goal of linear regression is to find the best fit line that can accurately predict the o/p for the continuous dependent variable.

If single independent variable is used for prediction then Simple Linear Regression and if there are more than 2 independent variable then it is Multiple Linear Regre

→ By finding the best fit line, algorithm establish the relationship b/w dependent variable & independent variable.

→ The o/p for Linear Regression should only be the continuous value such as price, age, salary, etc.

$$y = \underbrace{a_0 + a_1}_{\text{co-efficient}} x + \epsilon \rightarrow \text{error term}$$

* Logistic Regression: It is used for classification as well as for Regression Problem but mainly for classification problem.

→ Logistic Regression is used to predict the categorical dependent variable with the help of independent variable.

→ The o/p of Logistic Regression problem can only be b/w 0 & 1.

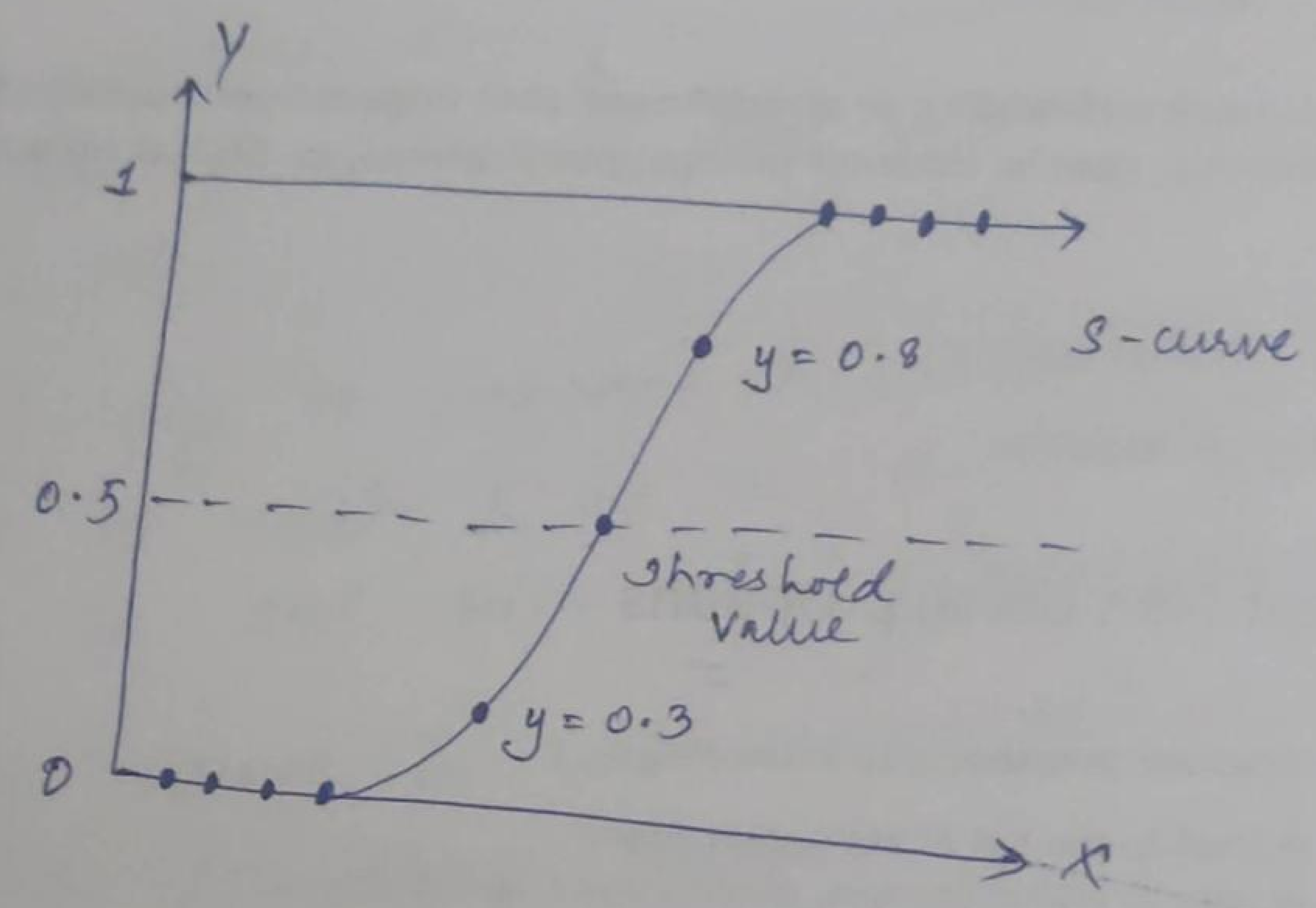
→ It can be used where the probabilities b/w 2 classes is required, such as whether it'll rain today or not either 0 & 1, true or false, etc.

→ It is based on the concept of MLE (Maximum Likelihood Estimation). According to this estimation the observed data should be most probable.

→ In Logistic, we pass the weighted sum of i/p through an activation fn. that can map value in b/w 0 & 1. Such activation fn is Sigmoid Fn. & the curve is Sigmoid Curve | S-curve.

→ Equation for logistic regression is :

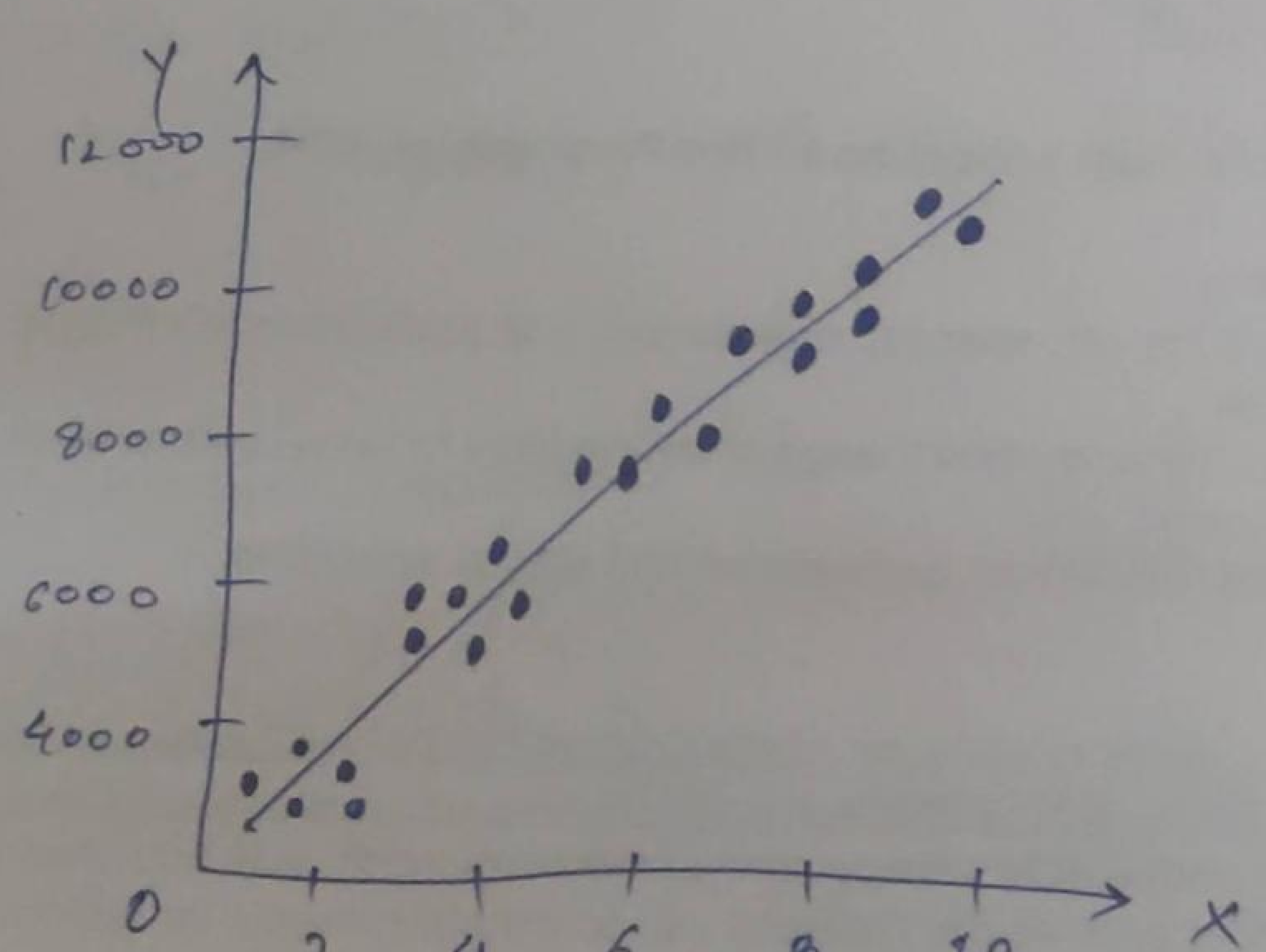
$$\log\left[\frac{y}{1-y}\right] = b_0 + b_1 x_1 + b_2 x_2 + \dots + b_n x_n$$



logistic Regression

→ Equation of linear regression is :

$$y = a_0 + a_1 x + \epsilon$$



Linear Regression

Naive Bayes Model

→ Need of Naive Bayes?

- One of the simplest density estimation methods from which we can form one of the standard classifications method in ML.
  - → Very easy to program & intuitive.
  - → Fast to train & use as a classifier.
  - → Easy to deal with missing attributes.
  - → Very popular in NLP / computational linguistics.
  - → used for large datasets.

Ex:-

→ A fruit may be considered to be an apple if it is Red, Round and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that it why it is known as 'Naive'.

→ Bayes theorem provides a way of calculating posterior probability $P(c|x)$ from $P(c)$; $P(x)$ & $P(x|c)$

Likelihood

$$P(c|x) = \frac{P(x/c)\,P(c)}{P(x)} \rightarrow \text{Class Prior Probability}$$

Posterior Probability

Predictor Prior Probability

$$P(c|x) = P(x_1|c) \times P(x_2|c) \times \ldots \times P(x_n|c) \times P(c)$$

→ Consider a training set of weather & corresponding target variable 'play'. Now need to classify whether players'll play or not based on weather condition.

Step 1: Convert the dataset into frequency table.

Step 2: Create likelihood table by finding the probabilities like overcast probability = 0.29 & probability of playing is 0.64.

Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table

Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Total	5	9

Likelihood Table.

Weather	No.	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36.
All	5	9		
	=5/14	=9/14		
	=0.36	=0.64		

Scanned by TapScanner

Step 3: Now use Bayesian eq. to calculate posterior prob. for each class. The class with the highest posterior probability is the outcome of prediction.

Problem: Players will play if weather is sunny, Is this statement is correct.

We can solve it using above discussed method of posterior probability.

$$P(Yes \mid Sunny) = P(Sunny \mid Yes) * P(Yes) \,/\, P(Sunny)$$

Here we have $P(Sunny \mid Yes) = \frac{3}{9} = 0.33$;

$P(Sunny) = \frac{5}{14} = 0.36$;

$P(Yes) = \frac{9}{14} = 0.64$;

$$P(Yes \mid Sunny) = \frac{0.33 * 0.64}{0.36} = 0.60.$$

→ Naive Bayes uses a similar method to predict the probability of different class based on various attributes. This algorithm is mostly used in text classification & with problems having multiple classes.

VARIOUS CLASSIFICATION METHODS

Naive Bayes - posterior prob. - find the class.

$P(C|X)$ - prob that same tuple $X = \langle x_1, \ldots x_2 \rangle$ is of class C.

P(class = N| outlook = sunny, windy = true ..)

Bayes Theorem - $P(C/X) = \dfrac{P(X|C) \cdot P(C)}{P(X)}$

$P(C/X)$ → post prob

$P(X|C)$ → conditional prob. given a hypothesis tuple satisfies it

$P(C)$ → prior prob. associated with hypothesis C → class sample

$P(X)$ → prob. of occurrence of each value X & is const for all classes

MAP - maximum a post. prob rule

$$\boxed{\cancel{P(C|X)}} \quad \theta_{MAP} = \underset{\theta}{\arg\max}\, f(\theta | x_1, x_2, \ldots x_n)$$

$$= \underset{\theta}{\arg\max} \left(\log g(\theta) + \sum_i \log f(x_i/\theta) \right)$$

MLE - maximum likelihood estimation - $\theta_{MLE} = \underset{\theta}{\arg\max}\, f(x_1, x_2, \ldots x_n/\theta)$

$$= \underset{\theta}{\arg\max} \sum_i \log f(x_i/\theta)$$

Neural Network

Linear Regression

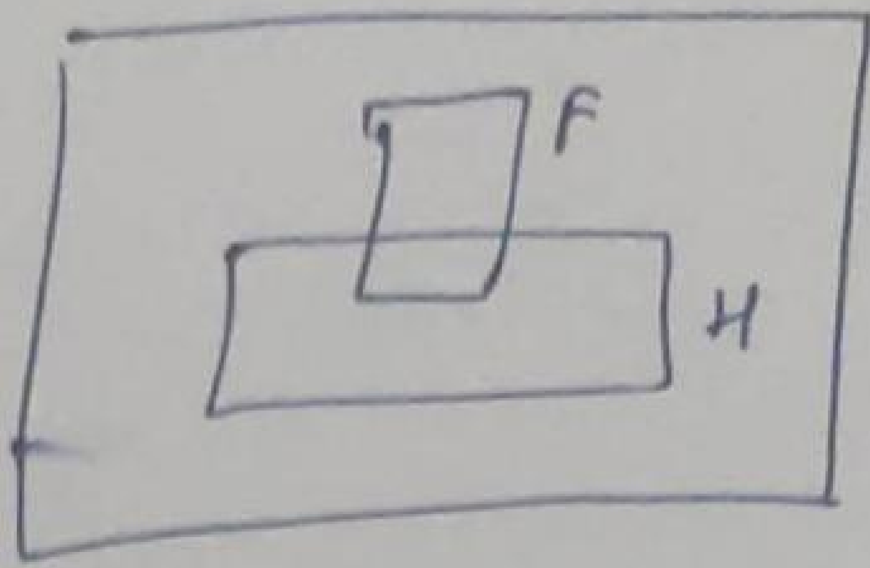
Naive Bayes

Logistic Regression

Parameter Estimation (PCA) and (LDA)

* **Conditional Probability:**

→ H – "Have a headache"
f – "Coming down with flu".



$P(H) = 1/10.$

$P(F) = 1/40.$

$P(H|F) = 1/2$

$$\boxed{P(A|B) + P(\sim A|B) = 1}$$

→ "Headache are rare & flu is rarer, but if you're coming down with 'flu there's a 50-50 chance you'll have a headache".

→ $P(H|F)$ = Fraction of flu-inflicted world in which you have a headache.

$$= \frac{\text{\# worlds with flu and headache}}{\text{\# worlds with flu}}$$

$$= \frac{\text{Area of "H \& F" region}}{\text{Area of F region}} = \frac{P(H \wedge F)}{P(F)}$$

⇒ Theory:

$P(A|B)$ = fraction of world in which B is true that also have A true

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

$P(A \wedge B) = P(A|B)\,P(B)$

$P(A|B) + P(\sim A|B) = 1$

$\sum_{k=1}^{n} P(A = v_k | B) = 1$

Example of EM Algorithm   Lecture: 33

→ Imagine you've 2 coins A & B.
One is more likely to get Heads, the other tails.
You pick one at random & toss it. which one was it.

→ Let's do this 5 times:

1) Pick a coin randomly
2) toss it 10 times
3) Record the no. of heads & tails
4) Get the average no. of heads for each coin

Coin	5 set, 10 Tosses / set	Coin A	Coin B	Average Heads.
B	H T T T H H T H T H		5H, 5T	$\hat{\theta}_A = \frac{24}{24+6} = 0.80$
A	H H H H T H H H H H	9H, 1T		
A	H T H H H H H T H H	8H, 2T		$\hat{\theta}_B = \frac{9}{9+11} = 0.45$
B	H T H T T T H H T T		4H, 6T	
A	T H H H T H H H T H	7H, 3T		
		24H, 6T	9H, 11T	

→ What is the probability that one is likely to get pick from A or B.

→ The higher the probability the more likelihood of that coin to be choosen.

→ Compute the likelihood that it was coin A & B using Binomial Distribution with mean probability θ on 'n' trials with 'k' success. $\boxed{p(k) = \binom{n}{k} \theta^{k} (1-\theta)^{n-k}}$

Learning with hidden data

EM Algorithm

→ **Numerical:** other way to think about this:

1) Assign random averages to both coins.
2) For each of 5 round of 10 coin tosses.
   a) check the %age of heads.
   b) find the prob. of it coming from each coin.
   c) compute the expected no of heads, using that prob. as a weight, multiply by the no of head
   d) Record those numbers.
   e) Re-compute new means for coin A & B.
3) with these new means go back to step 2.

→ 5 Round 10 coin tosses with $\theta_A = 0.6$, $\theta_B = 0.5$

1	H T T T H H T H T H
2	Refer prev. table only
3	
4	
5	

for 1st round, coin B $\frac{5}{10}$ Heads & $\frac{5}{10}$ Tails.

likelihood of "A" $= p_A(h)^h(1-p_A(h))^{10-h} = 0.000796$

" " "B" $= p_B(h)^h(1-p_B(h))^{10-h} = 0.000976$

Normalizing, 'll get probabilities $= 0.45$ & 0.55

(by adding both likelihood & then divide by 2)

→ B was the one to win

Estimating likely no. of heads & tails from:

A = Heads = 0.45 × 5 heads = 2.2 heads

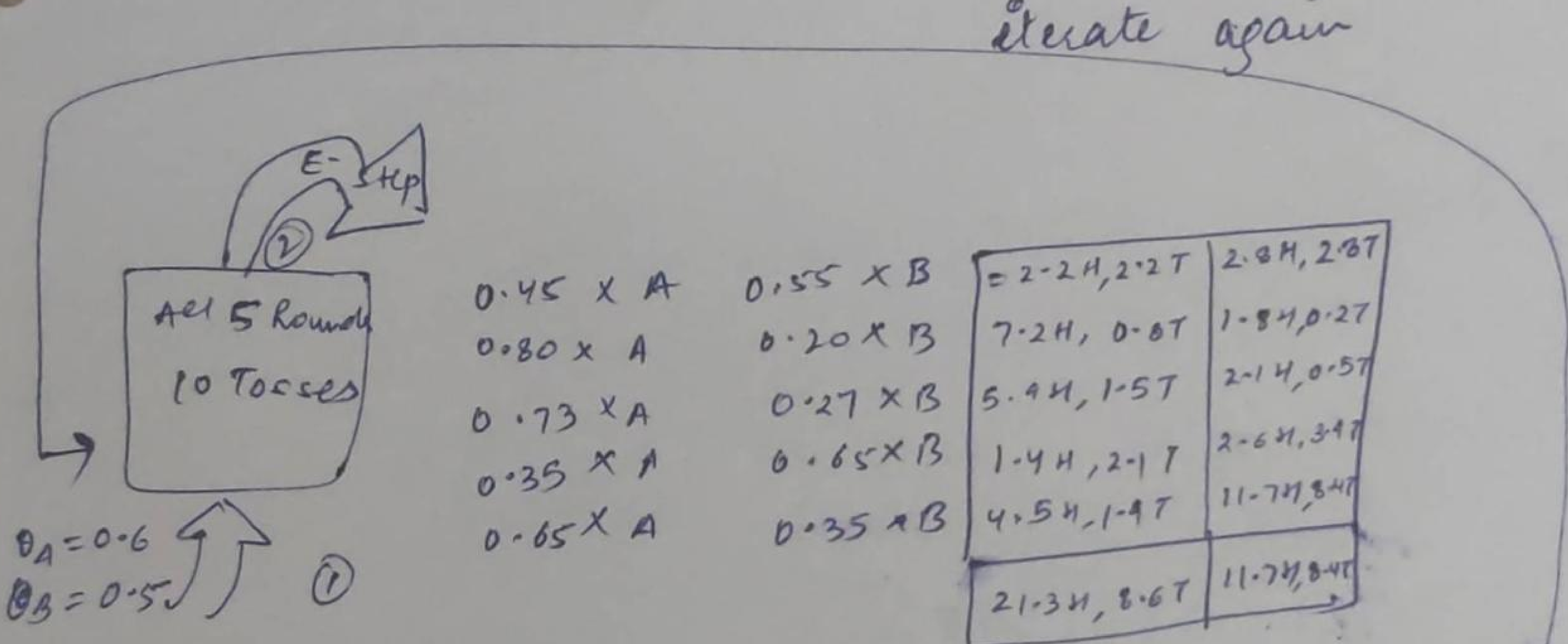
B = Heads = 0.55 × 5 heads = 2.8 heads

A = Tails = 0.45 × 5 Tail = 2.2 Tail

B = Tails = 0.55 × 5 Tail = 2.8 Tail

Do the same for all five runs.

* Expectation step:



0.45 × A	0.55 × B	= 2.2H, 2.2T	2.8H, 2.8T
0.80 × A	0.20 × B	7.2H, 0.8T	1.8H, 0.2T
0.73 × A	0.27 × B	5.9H, 1.5T	2.1H, 0.5T
0.35 × A	0.65 × B	1.4H, 2.1T	2.6H, 3.9T
0.65 × A	0.35 × B	4.5H, 1.9T	11.7H, 8.4T
		21.3H, 8.6T	11.7H, 8.4T

3 M-step

$\hat{\theta}_A = \frac{21.3}{21.3 + 8.6} = 0.71$

$\hat{\theta}_B = \frac{11.7}{11.7 + 8.4} = 0.56$

*. Compute the new prob. of each coin $= \frac{H}{H+T}$

$\hat{\theta}_A = .80$

$\theta_B = 0.52$

This gives new maximized parameter θ for each coin. 4.

* Repeat E & M step untill convergence

— X —