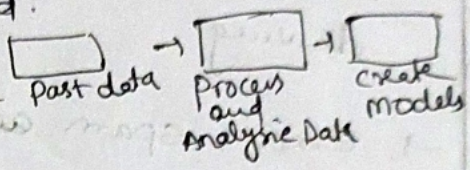
Machine learning

1) machine learning:-

→ machine learning is an application of AI that enables systems to learn and improve from experience without being explicitly Programmed.

→ Applications of M,L:-



1) Speech Recognition:-

→ while using Google, we get an option of search by voice, it comes under speech recognition.

→ Google Assistance, Siri, Alexa are using spech recognition technology to follow the voice instructions.

2) Product recommendations:-

→ machine learning is widely used by various e-commerce and entertainment companies such as Amazon, Netflex etc. for product recomendation to the user.

→ whenever we search for some product on Amazon, then we started getting an advertisement for the same product while internet surfing on the browser.

→ Self driving cars :-

→ machine learning plays a significant role in self driving cars (Tesla).

→ It is using unsupervised learning method to train the car models to detect people and objects while driving.

→ Email spam ~~and malw~~ filtering

→ whenever we recive a new email, it is filtered automatically as important normal and spam.

→ we always recive an important mail in our inbox with the important symbol and spam emails in our spam box.

→ Medical ~~dog~~ Diagnosis :-

→ In medical sciena, machine learning is used to diseases diagnoses.

→ with this, medical technology can predict the exact position of dieases in the brain.

→ Automatic language Translation:-

→ Translates the text from one language to another language.

→ Types of machine learning:-

1) Supervised learning
2) Unsupervised learning
3) Reinforcement Learning

1) Supervised machine Learning:-

→ The supervised learning technique, we train the machines using the labelled data set, and based on the training, the machine predicts the output.

→ We train the machine with input and corresponding output, and then we ask the machine to predict the output using the test dataset.

→ Suppose we have an input dataset of cats and dog images. So first, we will provide the training to the machine to understand the images, such as the shape and size of the tail of cat and dog, shape of eyes, height (dogs are taller, cats are smaller) etc.

→ After Completion of training, we input the picture of Cat and ask the machine to identify and predict the output.

→ the main goal of the supervised learning technique is to map the input variable (x) with the output variable (Y).

Ex:- Application: Fraud detection, Spam Filtering

→ Categories of supervised machine Learning

→ supervised machine learning can be classified into two types of Problems.

→ 1) Classification

→ 2) Regression.

→ 1) classification algorithms are used to solve the classification problem in which the output variable is Categorical. such as "Yes" or NO, male or Female etc.

→ 2) Regression:-

Regression finds correlation between dependent (target) and independent (predicted) variables. (co

→ Regression algorithm

Regression algorithms are used to solve regression problems in which there is a linear (straight line) relationship between input and output variables.

→ these are used to predict continous output variables, such as market trends, weather prediction, etc.

→ unsupervised machine learning:-

→ In unsupervised machine learning the machine is trained using the unlabeled dataset.

→ the main aim of the unsupervised learning algorithm is to group or categories the unsorted dataset according to the similarities, patterns and differences.

→ machines are instructed to find the hidden patterns from the input dataset.

→ suppose there is a basket of fruit images and we input it into the machine learning model.

→ the images are totally unknown to the model and the task of the machine is to find the patterns and categories of the objects.

→ So, now the machine will discover its patterns and such as colour, shape and predict the output when it is tested with test dataset.
Ex:- Network analysers :- identity plagarism
Recomendation system :- e-commerise web sites
→ Categories of unsupervised machine Learning

1) Clustering
2) Association.

→ clustering :-

→ The clustering technique is used when we want to find the inherent groups from the data.
→ It is a way to group the objects into a cluster such that the objects with the most similarities remain in one group and have fewer or no similarities with the objects of other groups.
→ An example of the clustering algorithm is grouping the customers by their purchasing behaviour.

→ Association:-

→ Association rule learning is a technique that checks for the dependency of one data item on another data item and maps accordingly.

→ It tries to find some interesting relations or associations among the variables of dataset.

→ It is based on different rules to discover the interesting relations between variables.

→ for example if a customer buys bread, he most likely can also buy butter, eggs or milk. so these products are stored within a shelf or mostly nearby.

→ Reinforcement Learning:-

→ It works on a feedback-based process, in which an AI agent (A software component) automatically explore its surrounding by hitting & trail, taking action, learning from experiences and improving its performance.

→ Agent gets rewarded for each good action and get punished for each bad action.

Hence the goal of reinforcement learning agent is to maximize the rewards.

→ In reinforcement learning there is no labelled data like supervised learning and agents learn from their experience only.

→ the reinforcement learning process is similar to a human being.
for example, a child learns various things by experiences in his day to day life.

→ Applications of RL are video Games, Robotics (intelligent Robots)

→ Basic concepts in machine learning

→ Features:- attributes that describes an instances in the dataset
Ex:- Customer — number of purchases
age
product brought

→ Feature selection:- (optimisation)
~~selecting best features~~
Process of selecting the optimal features to include in the training phase.

→ DATASET :- data used to train the system and detect patterns to design algorithms

→ Learning (or) Training :-

Patterns of a dataset are detected

→ Tunning :-

Optimising the parameters of an algorithm to find the best ~~combination for your specific data set~~ model performance. (Hyper parameter optimisation)

→ MODELS :-

~~models~~ models to make predictions

→ Validation :-

It is the process of evaluating a trained model on test data set.

→ Machine Learning Process

1) Gathering data
2) Preparing the data
3) Choosing a model
4) Training
5) Evaluation
6) ~~Hyper~~ parameter tunning
7) Prediction.

→ Testing Machine Learning Algorithms

→ Goal of ML testing:-

1) Quality assurance:-
It is required to make sure that the software system works accordiy to the requirement.

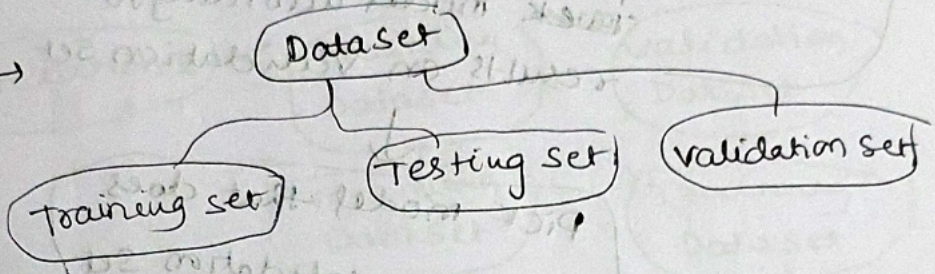
2) The purpose of ML Testing is first of all to ensure that the learned logic will remain consistent, no matter how many times we call the program.

→ Model evaluation in ML testing:-

1) Unit tests:- The program is broken down in blocks, and each unit is tested separately

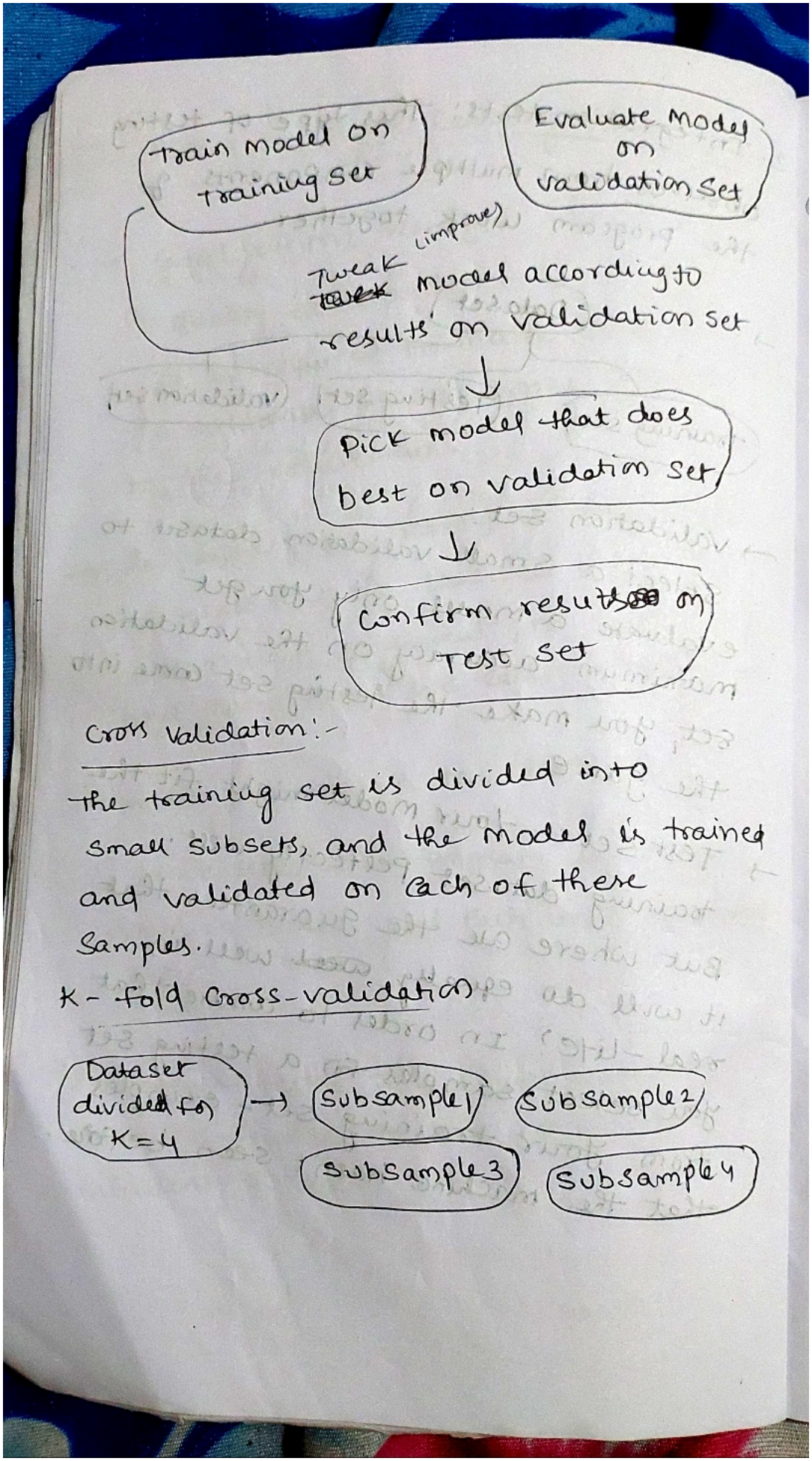
2) Regression tests: They cover already tested software to see if it doesn't suddenly break.

3. Integration tests:- This type of testing observes how multiple components of the program work together.

→ 

→ Validation set:-
Select a small validation dataset to evaluate a model. only you get maximum accuracy on the validation set, you make the testing set come into the game.

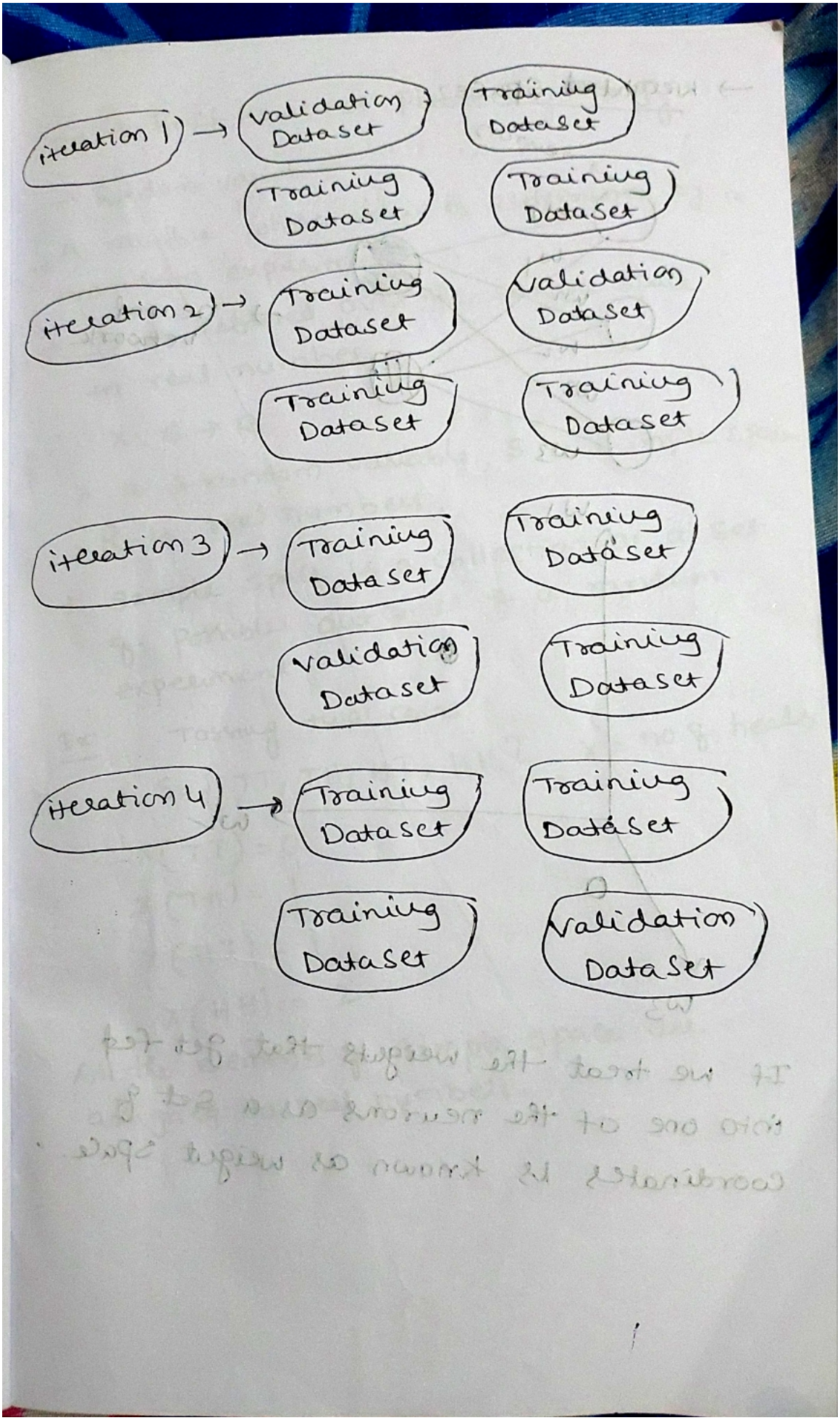
→ Test set:- Your model might fit the training dataset perfectly well. But where are the guarantees that it will do equally well in real-life? In order to assure that you select samples for a testing set from your training set - examples that the machine hasn't seen before.



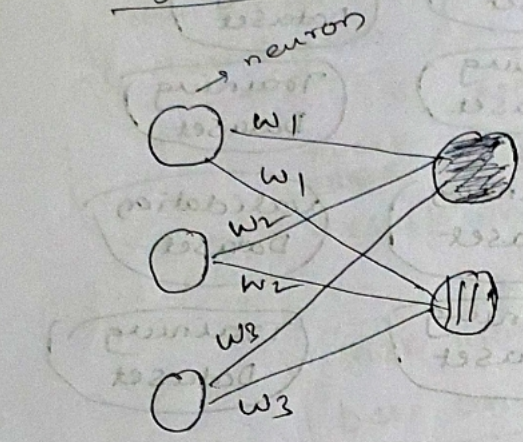
Cross Validation:-

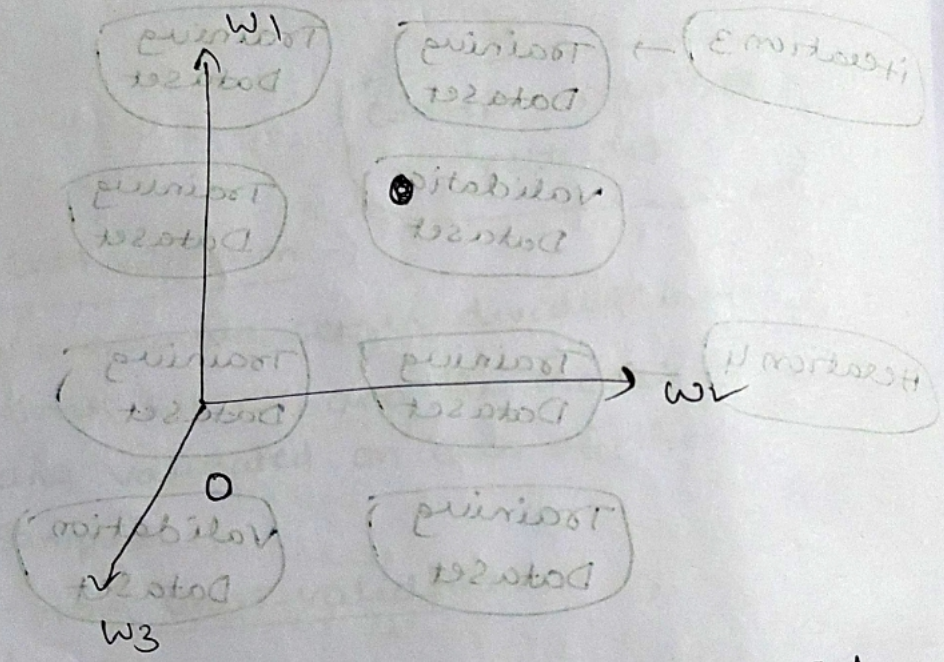
The training set is divided into small subsets, and the model is trained and validated on each of these samples.

K-fold Cross-validation


iteration 1
Validation Dataset
Training Dataset
Training Dataset
Training Dataset
iteration 2
Training Dataset
Validation Dataset
Training Dataset
Training Dataset
iteration 3
Training Dataset
Training Dataset
Validation Dataset
Training Dataset
iteration 4
Training Dataset
Training Dataset
Training Dataset
Validation Dataset

→ Weight Space:-

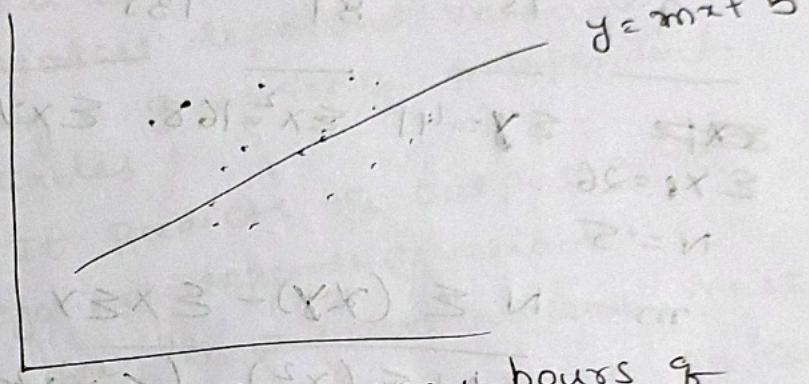




If we treat the weights that get fed into one of the neurons as a set of coordinates is known as weight space."

→ Lineal regression:-

It is a statistical method used to find the line of best fit of the form of an equation such as $y = mx + b$ to the given data



Ex:- Sam found how many hours of sunshine vs how many ice creams were sold at the shop from monday to friday:

X hours of Sunshine	Y Ice Creams Sold
2	4
3	5
5	7
7	10
9	15

Using least square method, find the regression line of the above data.

x	y	x^2	xy
2	4	4	8
3	5	9	15
5	7	25	35
7	10	49	70
9	15	81	135
	$\Sigma y = 41$	$\Sigma x^2 = 168$	$\Sigma xy = 263$

$\Sigma x_i = 26$

$N = 5$

$$m = \frac{N\Sigma(xy) - \Sigma x \Sigma y}{N\Sigma(x^2) - (\Sigma x)^2}$$

$$= \frac{5 \times 263 - 26 \times 41}{5 \times 168 - (26)^2}$$

$$= \frac{1315 - 1066}{840 - 676} = 1.5183$$

$$b = \frac{\Sigma y - m\Sigma x}{N}$$

$$= \frac{41 - 1.5183 \times 26}{5}$$

$$= 0.3049$$

$$y = mx + b$$

$$y = 1.518x + 0.305$$

suppose weather forcast says we expect 8 hours of sun tommorrow

$y = 1.518 \times 8 + 0.305 = 12.45$ Ice cream

→ Logistic regression:-

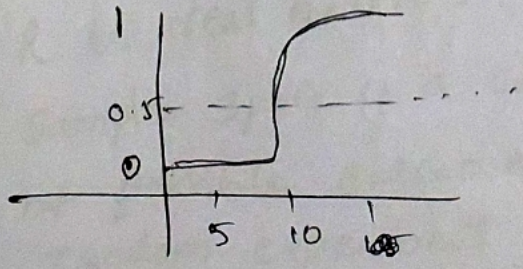
→ It is used to predict the categorical dependent variable using a given set of independent variables

→ It Predicts the output of a categorical dependent variable.

→ Therefore the outcome must be a categorical value. It can be either yes or NO, 0 & 1, true & false etc. but instead of giving the exact value as 0 and 1. It gives the probabilistic values which lies between 0 and 1.

(X) CGPA		(Y) Placement
8.36		yes
5.13		NO

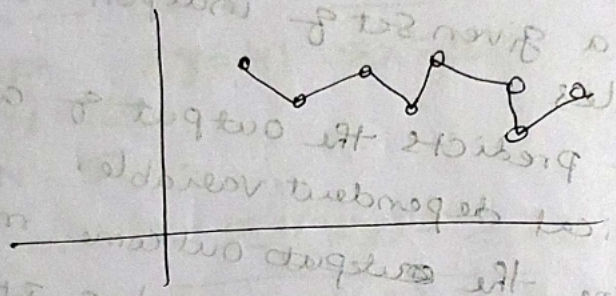
$$y = \frac{1}{1+e^{-x}}$$



→ Overfitting and Underfitting

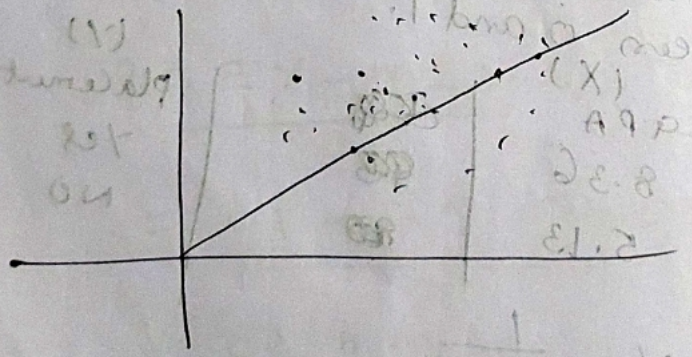
→ Overfitting occurs when our machine Learning model tries to cover all the more than the required data points present in the given dataset.

→ ML model works well on training data but not on test data.



→ underfitting occurs when our ML model is not able to capture the underlying trend of the data. (unable to capture the data points present in the plot)

→ ML model not able to capture the relationship in training data



→ curse of dimensionality.

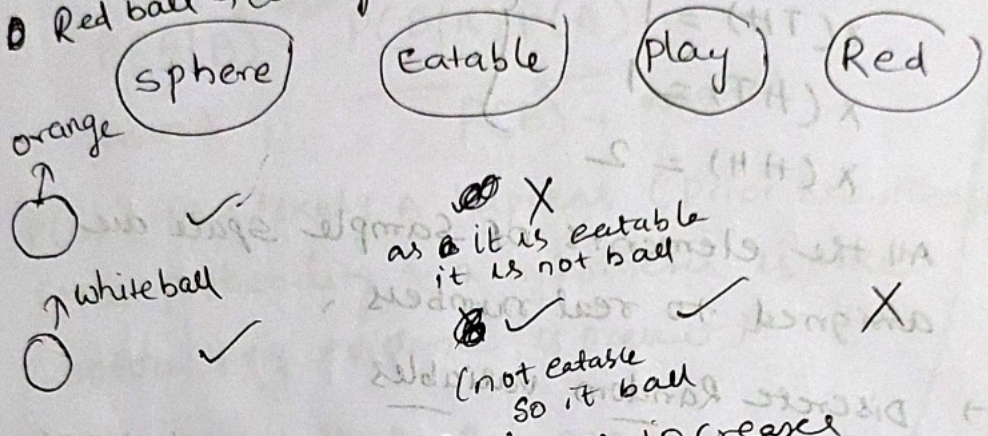
2 4 6 10 12 | 100 1000

↑ ↑ ↑ ↑ ↑ | ↑ ↑

m_1 m_2 m_3 m_4 m_5 | m_6 m_7

↳ threshold

Red ball → clarity



→ As the number features increases your model get confused.

→ A Brief Review of Probability theory

1) Random variable :-

→ A variable whose value is determined by a random experiment.

→ It is defined over the sample space to real number.

$X: S \to R$

X is a random variable, S is Sample space R is real numbers.

→ Sample space is a collection or a set of possible outcomes of a random experiment.

Ex:- Tossing two coins

$S = \{TT, TH, HT, HH\}$

X = no of heads

$X(TT) = 0$
$X(TH) = 1$
$X(HT) = 1$
$X(HH) = 2$

} 0, 1, 2

All the elements of sample space are assigned to real numbers.

→ Discrete Random variables

It takes only finite number of distinct values : Ex: 0, 1, 2, 3, 4...

→ Continous random variables

Infinite and uncountable set of values

Ex:- Interest rates of loans in a country.

→ Conditional probability :-

If A and B are two events in a sample space 'S', then Conditional Probability of A given B is

$$P(A/B) = \frac{P(A \cap B)}{P(B)} \quad \text{when } P(B) > 0$$

↓
event B is already occured.

Bayes rule & theorem:-

It determines the conditional probability of an event A given that event B has already occurred.

hypothesis → evidence / data

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

$P(A|B)$ → Posterior; $P(B|A)$ → likelihood; $P(A)$ → prior; $P(B)$ → marginal

$P(A)$ = how likely A happens (prior knowledge) (the probability of a hypothesis is true before any evidence is present).

$P(B)$ = how likely B happens
The probability of observing the evidence, (marginal)

$P(A|B)$ = how likely A happens given that B has happend (posterior)

$P(B|A)$ = how likely B happens given that A has happend. (likelihood)

→ Independence and Conditional Independence

→ Two random variables X and Y are said to be statistically independent if and only if

$$P(x, y) = P(x)\,P(y)$$

Ex:- Independent - X : Thow of a dice
Y : Toss of a coin

NOT Independent - X : Height Y : weight
In general as height increases weight increases

→ Conditional Independence :-

Two random variables X and Y are said to be independent given Z if and only if

$$P(x, y|z) = P(x|z)\,P(y|z)$$

Ex:- X : Height Y : vocabulary Z : Age

→ Not Independent unless I gave some condition. (If I tell you this person is just 2 feet tall, it automatically means it is most probably that this person must be child and have low vocabulary.

Mean and variance :-

→ mean is the average of given set of numbers

→ variance gives the variation from the expected value.

→ variance also measures amount of fluctuation of the variable.

$$\text{variance} = \frac{\sum (x_i - \bar{x})^2}{n-1}$$

x_i = value of the one observation
$\bar{x}$ = mean value of all observations
n = number of observations.

→ Confusion matrix:- It is a table that is used to define the performance of a classification algorithm.

Actual \ Prediction	1	0
1	True Positive (TP)	False Negative (FN)
0	False Positive (FP)	True Negative (TN)

(False Negative → actual; → Prediction)

True positive:- The model has predicted yes, and actual value was also yes

True Negative: model has predicted NO and actual value was also NO.

False positive:- The model has predicted yes but the actual value was NO.

False Negative: The model has predicted NO but actual value was yes.

→ Accuracy:- The accuracy is used to find the portion of correct classified values

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

(no of correct predictions / Total no of predictions)

Precision :- what proportion of Predicted Positives is truly Positive (suppose if I have to classify shoes Nike of the shoes classified Nike, How many are actualy Nike (true)

$$\text{Precision} = \frac{TP}{TP + FP}$$

Out of all dog Predicion how many you got it right.

Recall :- what proportion of actual Positives is correctly classified.

Of the shoes that are actualy Nike, how many are classified as Nike.

$$\text{Recall} = \frac{TP}{(TP + FN)}$$

Q. Recall is out of all dog truth how many you got it right

$$\text{F1 score} = \frac{2PR}{P+R}$$

(Harmonic mean of Precision and Recall)

↓ average of Precision and Recall.

→ Consider the given dataset Apply Naive-Bayes algorithm and predict that if a fruit has the following properties, then which type of Fruit it is

Fruit	Yellow	Sweet	Long	Total
orange	350	450	0	650
Banana	400	300	350	400
others	50	100	50	150
Total	800	850	400	1200

Sol:- $P(A|B) = \frac{P(B/A)\,P(A)}{P(B)}$

$P(\text{yellow}|\text{orange}) = \frac{P(\text{orange}|\text{yellow})\cdot P(\text{yellow})}{P(\text{orange})}$

Probability of yellow given that fruit is orange

$= \frac{\frac{350}{800} \times \frac{800}{1200}}{\frac{650}{1200}}$

$= 0.53$

$P(\text{sweet}|\text{orange}) = \frac{P(\text{orange}|\text{sweet})\cdot P(\text{sweet})}{P(\text{orange})}$

$= \frac{\frac{450}{850} \times \frac{850}{1200}}{\frac{650}{1200}} = 0.69$

$P(\text{long}|\text{orange}) = \frac{P(\text{orange}|\text{long})\,P(\text{long})}{P(\text{orange})}$

$= \frac{\frac{0}{400} \times \frac{400}{1200}}{\frac{650}{1200}} = 0$

$P(\text{Fruit}|\text{orange}) = 0.53 \times 0.69 \times 0 = 0$

Note:- Probability of event to happen

$P(E) = \frac{\text{No of favourable outcomes}}{\text{Total no of outcomes}}$

$$P(\text{yellow} \mid \text{Banana}) = \frac{P(\text{Banana} \mid \text{yellow})\, P(\text{yellow})}{P(\text{Banana})}$$

$$= \frac{\frac{400}{800} \times \frac{800}{1200}}{\frac{400}{1200}} = 1$$

$$P(\text{sweet} \mid \text{Banana}) = \frac{P(\text{Banana} \mid \text{sweet})\, P(\text{sweet})}{P(\text{Banana})}$$

$$= \frac{\frac{300}{850} \times \frac{850}{1200}}{\frac{400}{1200}} = 0.75$$

$$P(\text{long} \mid \text{Banana}) = \frac{P(\text{Banana} \mid \text{long})\, P(\text{long})}{P(\text{Banana})}$$

$$= \frac{\frac{350}{400} \times \frac{400}{1200}}{\frac{400}{1200}} = 0.87$$

$$P(\text{Fruit} \mid \text{Banana}) = 1 \times 0.75 \times 0.87 = 0.65$$

$$P(\text{yellow} \mid \text{others}) = \frac{P(\text{others} \mid \text{yellow})\, P(\text{yellow})}{P(\text{others})}$$

$$= \frac{\frac{50}{800} \times \frac{800}{1200}}{\frac{150}{1200}} = 0.33$$

$$P(\text{sweet} \mid \text{others}) = \frac{P(\text{others} \mid \text{sweet})\, P(\text{sweet})}{P(\text{others})}$$

$$= \frac{\frac{100}{850} \times \frac{850}{1200}}{\frac{150}{1200}} = 0.66$$

$$P(\text{long} \mid \text{others}) = \frac{P(\text{others} \mid \text{long})\, P(\text{long})}{P(\text{others})}$$

$$= \frac{\frac{50}{400} \times \frac{400}{1200}}{\frac{150}{1200}} = 0.33$$

$$P(\text{Fruit} \mid \text{others}) = 0.33 \times 0.66 \times 0.33 = 0.072$$

→ Fruit = Banana.

⇒ Bias :- while making predictions, a difference occurs between Expected values made by the model and actual values and this difference is known as bias.

bias of an estimator is

$$\text{bias}(\hat{\theta}_m) = E(\hat{\theta}_m) - \theta$$

$\hat{\theta}_m$: predicted, $E(\hat{\theta}_m)$: Expected, θ : actual

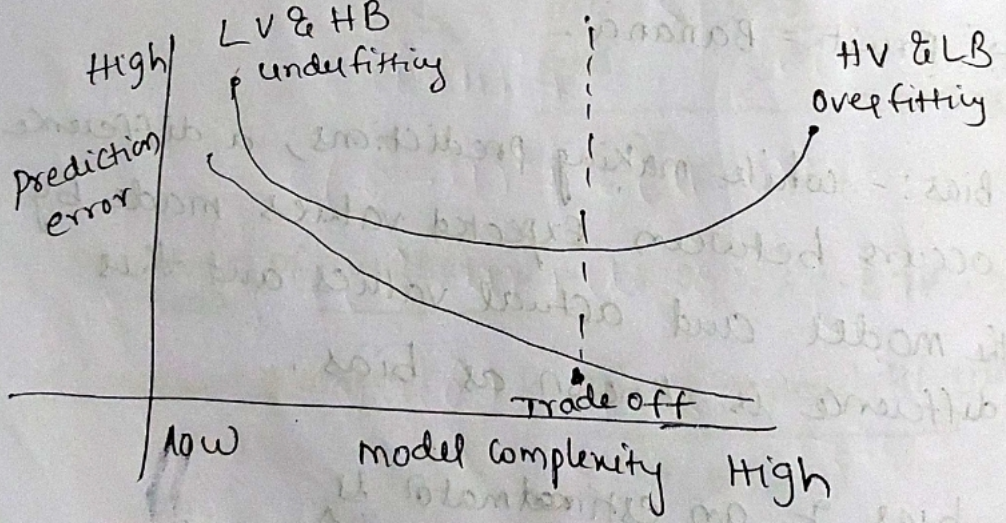
→ the Bias - variance Tradeoff :-

→ Bias and variance measure two different source of errors

→ while building the machine learning model it is really important to take care of bias and variance in order to avoid overfitting and underfitting in the model

→ Bias - variance trade-off is about finding the sweet spot (center) to make a balance between bias and variance errors.

→ the goal of any supervised machine learning algorithm is to achieve low bias and low variance



Unit-2

→ Linear models for Regression:-
(Linear Basis Function models)

→ the basic linear model for regression is a model that involves a linear combination of the input variables.

$$y(w,x) = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_D x_D$$

where $x = (x_1, \cdots x_D)^T$

→ the key attribute of this function is, that it is a linear function of the parameters $w_0, w_1, \cdots w_D$.

→ It is also a linear function of the input variable x.

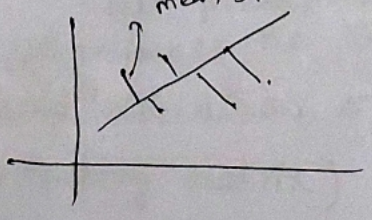
→ The Bias-variance Decomposition:-

The bias-variance decomposition is a useful theoretical tool for understanding a learning algorithm's performance characteristics.

Mean squared error $MSE(\hat{\theta}) = bias(\hat{\theta})^2 + var(\hat{\theta})$

$= E\left[(\hat{\theta} - \theta)^2\right] =$ ~~$Bias^2(\hat{\theta}, \theta) + var(\hat{\theta})$~~

(s is random variable
$\hat{\theta}$ is predicted, θ is actual



mean squared error measures how close a regression line is to a set of data points.

$$E[(\hat{\theta}_s - \theta)^2] = E[\hat{\theta}_s^2] + \theta^2 - 2E[\hat{\theta}_s]$$

($E(\theta)$ is constant which is θ)

$$Bias^2(\hat{\theta}_s, \theta) = (E[\hat{\theta}_s] - \theta)^2$$

$$= E^2[\hat{\theta}_s] + \theta^2 - 2E[\hat{\theta}_s]\theta$$

$$var(\hat{\theta}_s) = E[\hat{\theta}_s^2] - E^2[\hat{\theta}_s]$$

$$mse(\hat{\theta}) = bias(\hat{\theta})^2 + var(\hat{\theta})$$

$$\Rightarrow E(\hat{\theta} - \theta)^2 = E((\hat{\theta} - \mu) + (\mu - \theta))^2$$

$$= E((\hat{\theta} - \mu)^2 + 2(\hat{\theta} + \mu)(\mu - \theta) + (\mu - \theta)^2)$$

($\because$ Let $\mu = E\hat{\theta}$, $E(\hat{\theta} - \mu) = 0$)

$$= E(\hat{\theta} - \mu)^2 + E(\mu - \theta)^2$$

$$= E(\hat{\theta} - \mu)^2 + (\mu - \theta)^2$$

$\underbrace{E(\hat{\theta} - \mu)^2}_{var(\hat{\theta})}$ $\underbrace{(\mu - \theta)^2}_{bias^2}$

(Expected value of Constant is Constant)

$E(\mu - \theta)^2 = (\mu - \theta)^2$

→ Bayesian Linear Regression :-

→ In the Bayesian view point, we formulate linear regression using probability distribution rather than point estimates.

→ The response, y, is not estimated as a single value, but is assumed to be drawn from a probability distribution.

→ the model for Bayesian Linear Regression with the response sampled from a normal distribution is

$$y \sim N(\beta^T X, \sigma^2 I)$$

→ the output y is generated from a normal (Gaussian) Distribution characterized by a mean and variance.

→ The mean for linear regression is the transpose of the weight matrix multiplied by the predictor matrix.

→ the variance is the square of the standard deviation σ (multiplied by the identity matrix)

→ the aim of Bayesian linear regression is not to find the single best value of the model parameters, but rather to determine the posterior distribution for the model parameters.

→ the posterior Probability (updated probability of an event obtained by applying the new evidence formed) of the model parameters is conditional upon the training inputs and outputs.

$$P(\beta|y,X) = \frac{P(y,X|\beta)\,P(\beta)}{P(y,X)}$$

→ Here $P(\beta|y,X)$ is the posterior Probability distribution of the model parameters given the inputs and outputs.

→ $P(\beta|y,X)$ = Posterior of model parameters (best data distribution given specific situation in data)

$P(y,X|\beta)$ = Likelihood of the response Features Given the model and predictors Features.

$P(\beta)$ = prior Probability of the model parameters

$P(y,X)$ = Normalizing constant independent of Model.

→ Naive Bayes

1) For the given dataset Apply Naive-Bayes Algorithm and predict the outcome for a car = {Red, Domestic, SUV}

Color	Type	Origin	stolen
Red	sports	Domestic	yes
Red	sports	Domestic	NO
Red	sports	Domestic	yes
yellow	sports	Domestic	NO
yellow	sports	imported	yes
yellow	SUV	imported	NO
yellow	SUV	imported	yes
yellow	SUV	Domestic	NO
Red	SUV	imported	NO
Red	sports	imported	yes

$$P(A/B) = \frac{P(B/A)\,P(A)}{P(B)}$$

(likelihood → $P(B/A)$; prior → $P(A)$; marginal → $P(B)$; posterior → $P(A/B)$)

$X = \{Red, Domestic, SUV\}$ → Is it stolen

$$P(Red/yes) = \frac{P(yes/Red)\,P(Red)}{P(yes)}$$

$$= \frac{\frac{3}{5} \times \frac{5}{10}}{\frac{5}{10}} = 3/5$$

$$P(\text{Domestic}|\text{Yes}) = \frac{P(\text{Yes}|\text{Domestic}) \cdot P(\text{Domestic})}{P(\text{Yes})}$$

$$= \frac{\frac{2}{5} \times \frac{5}{10}}{\frac{5}{10}} = \frac{2}{5}$$

$$P(\text{SUV}|\text{Yes}) = \frac{P(\text{Yes}|\text{SUV}) \cdot P(\text{SUV})}{P(\text{Yes})}$$

$$= \frac{\frac{2}{5} \times \frac{5}{10}}{\frac{5}{10}} = \frac{2}{5}$$

$$= \frac{\frac{1}{4} \times \frac{4}{10}}{\frac{5}{10}} = \frac{1}{5}$$

$$P(\text{Red}|\text{NO}) = \frac{P(\text{NO}|\text{Red}) \, P(\text{Red})}{P(\text{NO})}$$

$$= \frac{\frac{2}{5} \cdot \frac{5}{10}}{\frac{5}{10}} = \frac{2}{5}$$

$$P(\text{Domestic}|\text{NO}) = \frac{P(\text{NO}|\text{Domestic}) \cdot P(\text{Domestic})}{P(\text{NO})}$$

$$= \frac{\frac{3}{5} \times \frac{5}{10}}{\frac{5}{10}} = \frac{3}{5}$$

$$P(\text{SUV}|\text{NO}) = \frac{P(\text{NO}|\text{SUV}) \cdot P(\text{SUV})}{P(\text{NO})}$$

$$= \frac{\frac{3}{4} \times \frac{4}{10}}{\frac{5}{10}} = \frac{3}{5}$$

c)

~~P(X|Yes) = 3/5 × 2/5 × 1/5 = 6/125 = 0.6~~

P(Yes|X)

P(X|Yes) = P(Yes) · P(Red|Yes) · P(Domestic|Yes) · P(SUV|Yes)

= 5/10 × 3/5 × 2/5 × 1/5

= 3/125 = 0.024

P(NO|X) = P(NO) · P(Red|NO) · P(Domestic|NO) · P(SUV|NO)

= 5/10 × 2/5 × 3/5 × 3/5

= 9/125 = 0.072

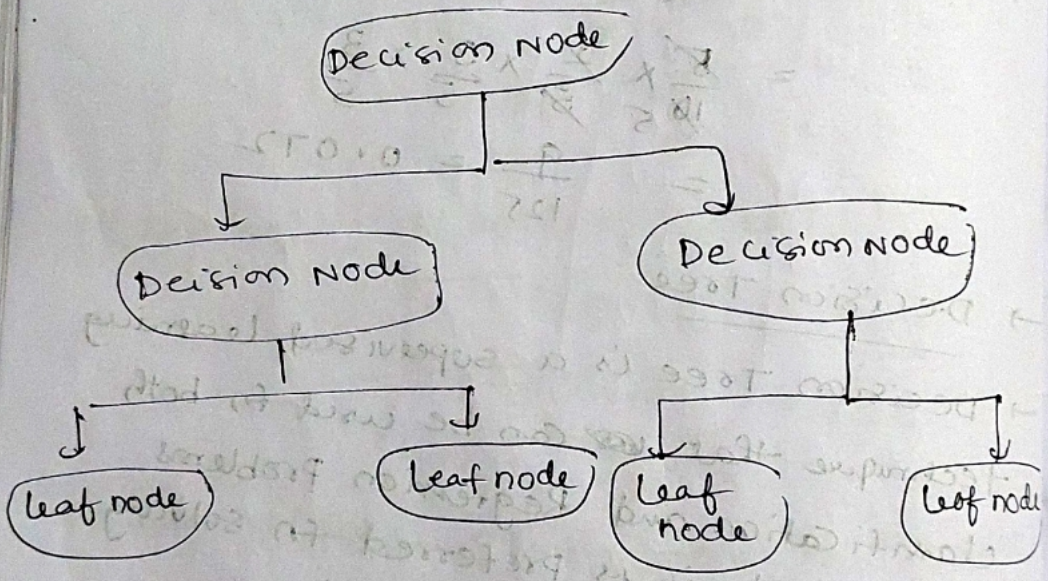
→ Decision Tree

→ Decision Tree is a supervised learning technique that can be used for both classification and Regression problems but mostly it is preferred for solving classification problems.

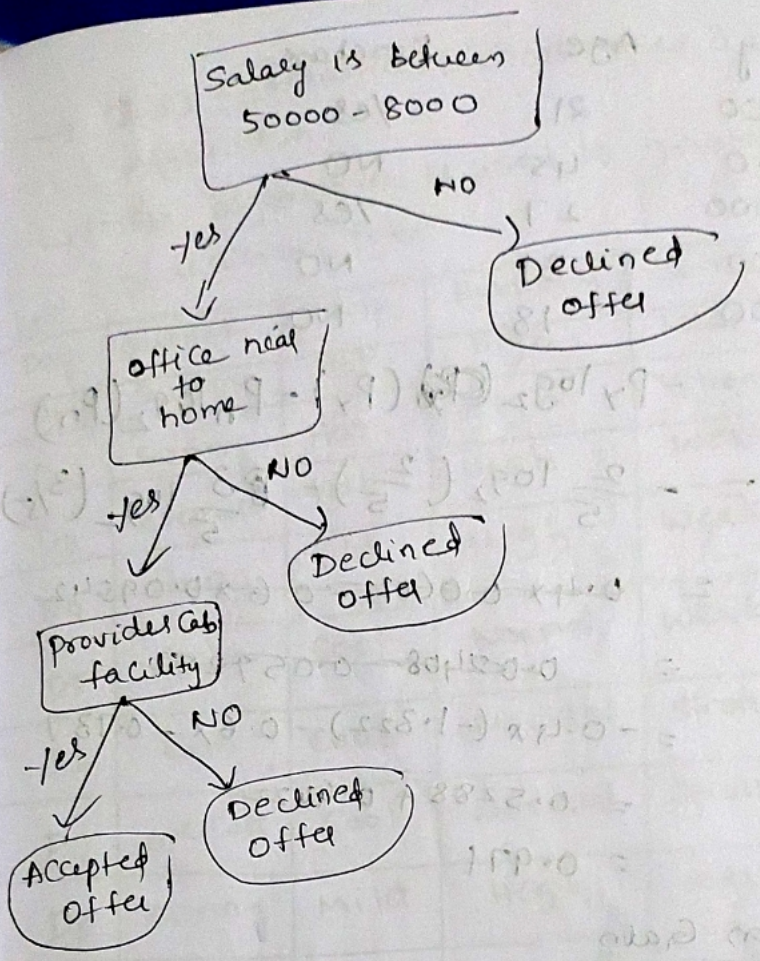
→ It is a tree structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

→ In a Decision tree, there are two nodes which are the Decision Node, and leaf node. Decision nodes are used to make any decision and have multiple branches, where as leaf nodes are the output of those decisions and donot contain any further branches

→ It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.



Ex:- Suppose there is a Candidate who whas job offer and wants to decide whether he should accept the offer or Not.



→ Entropy :-

Entropy is nothing but the measure of disorder or measure of impurity.

$$E(S) = \sum_{i=1}^{c} -P_i \log_2 P_i$$

P_i is frequentist probability of an element "i" in our data.

Salary	Age	Purchase
20000	21.	yes
10000	45	NO
60000	27	yes
15000	31	NO
12000	18	NO.

$$E(S) = -P_y \log_2(P_y) - P_n \log_2(P_n)$$

$$= -\frac{2}{5}\log_2\left(\frac{2}{5}\right) - \frac{3}{5}\log_2\left(\frac{3}{5}\right)$$

$$= 0.4 \times 0.0602 = 0.6 \times 0.09542$$

$$= 0.02408 - 0.057252$$

$$= -0.4 \times (-1.322) - 0.6 \times -0.737$$

$$= 0.5288 + 0.4422$$

$$= 0.971$$

Information Gain

→ Information Gain is a metric used to train Decision Trees.

→ Constructing a decision tree is all about finding attribute that returns the highest information gain

$$Gain(S, A) = Entropy(S) - \sum_{v \in values(A)} \frac{|S_v|}{|S|} Entropy$$

(S: set, A: Attribute, S_v: subset)

→ ID3 (Iterative Dichotomiser 3) Algorithm

→ This algorithm is used to generate a decision tree.

Ex:-

Day	Outlook	Temp	Humidity	Wind	PlayTennis :
D1	Sunny	Hot	High	Weak	NO
D2	Sunny	Hot	High	Strong	NO
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	yes
D5	Rain	Cool	Normal	Weak	yes
D6	Rain	Cool	Normal	Strong	NO
D7	Overcast	Cool	Normal	Strong	yes
D8	Sunny	Mild	High	Weak	NO.
D9	Sunny	Cool	Normal	Weak	yes
D10	Rain	Mild	Normal	Weak	yes
D11	Sunny	Mild	Normal	Strong	yes
D12	Overcast	mild	High	Strong	yes
D13	Overcast	~~Mild~~ Hot	~~High~~ Normal	~~Strong~~ Weak	yes
D14	Rain	Mild	High	Strong	NO

→ Attribute : outlook =

values(outlook) = sunny, overcast, Rain

$S = [9+, 5-]$

(S: whole dataset)

$$Entropy(S) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right)$$

$$= 0.94$$

$S_{sunny} = [2+, 3-]$

$$Entropy(S_{sunny}) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}$$

$$= 0.971$$

$S_{overcast} = [4+, 0-]$

$$Entropy(S_{overcast}) = -\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4} = 0$$

$S_{Rain} = [3+, 2-]$

$$Entropy(S_{rain}) = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}$$

$$= 0.971$$

Gain(S, outlook) = Entropy(S) − $\sum_{v \in \text{sunny, overcast, Rain}} \frac{|S_v|}{|S|}$ Entropy(S_v)

[S_v = sunny, overcast, Rain]

= Entropy(S) − $\frac{5}{14}$ Entropy(S_{sunny}) − $\frac{4}{14}$ Entropy($S_{overcast}$) − $\frac{5}{14}$ Entropy(S_{Rain})

= 0.94 − $\frac{5}{14}$ × 0.971 − $\frac{4}{14}$ × 0 − $\frac{5}{14}$ × 0.971

= 0.3464

→ Attribute: Temp

values(Temp) = Hot, mild, Cool

S_{Hot} = [2+, 2−]

Entropy(S_{Hot}) = $-\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4}$ = 1.0

S_{mild} = [4+, 2−]

Entropy(S_{mild}) = $-\frac{4}{6}\log_2\frac{4}{6} - \frac{2}{6}\log_2\frac{2}{6}$

= 0.9183

S_{cold} = [3+, 1−]

Entropy(S_{cool}) = $-\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4}$

= 0.8113

Gain(S, Temp)

$$= \text{Entropy}(S) - \sum_{v \in \{Hot, mild, Cool\}} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

$$= \text{Entropy}(S) - \frac{4}{14}\text{Entropy}(S_{Hot}) - \frac{6}{14}\text{Entropy}(S_{mild}) - \frac{4}{14}\text{Entropy}(S_{Cool})$$

$$= 0.94 - \frac{4}{14} \times 1.0 - \frac{6}{14} \times 0.9183 - \frac{4}{14} \times 0.8113$$

$$= 0.0289$$

→ Attribute (Humidity) = High, Normal

$S_{High} = [3+, 4-]$

$$\text{Entropy}(S_{High}) = -\frac{3}{7}\log_2\frac{3}{7} - \frac{4}{7}\log_2\frac{4}{7} = 0.9852$$

$S_{Normal} = [6+, 1-]$

$$\text{Entropy}(S_{Normal}) = -\frac{6}{7}\log_2\frac{6}{7} - \frac{1}{7}\log_2\frac{1}{7}$$

$$= 0.5916$$

Gain(S, Humidity)

$$= \text{Entropy}(S) - \sum_{v \in \{High, Normal\}} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

Gain(S, Humidity)

$= Entropy(S) - \frac{7}{14} Entropy(S_{High}) -$

$\frac{7}{14} Entropy(S_{Normal})$

Gain(S, Humidity)

$= 0.94 - \frac{7}{14} \times 0.9852 - \frac{7}{14} \times 0.5916$

$= 0.1516$

→ values(wind) = strong, weak

$S_{strong} = [3+, 3-]$

$Entropy[S_{strong}] = -\frac{3}{6} \log_2 \frac{3}{6} - \frac{3}{6} \log_2 \frac{3}{6}$

$= 1$

$S_{weak} = [6+, 2-]$

$Entropy[S_{weak}] = -\frac{6}{8} \log_2 \frac{6}{8} - \frac{2}{8} \log_2 \frac{2}{8}$

$= 0.8113$

Gain(S, wind)

$= Entropy(S) - \sum_{V \in \{strong, weak\}} \frac{|S_V|}{|S|} Entropy(S_V)$

Gain(S, wind)

$= Entropy(S) - \frac{6}{14} Entropy(S_{strong}) -$

$\frac{8}{14} Entropy(S_{weak})$

$$Gain(S, wind) = 0.94 - \frac{6}{14} \times 1.0 - \frac{8}{14} \times 0.811$$

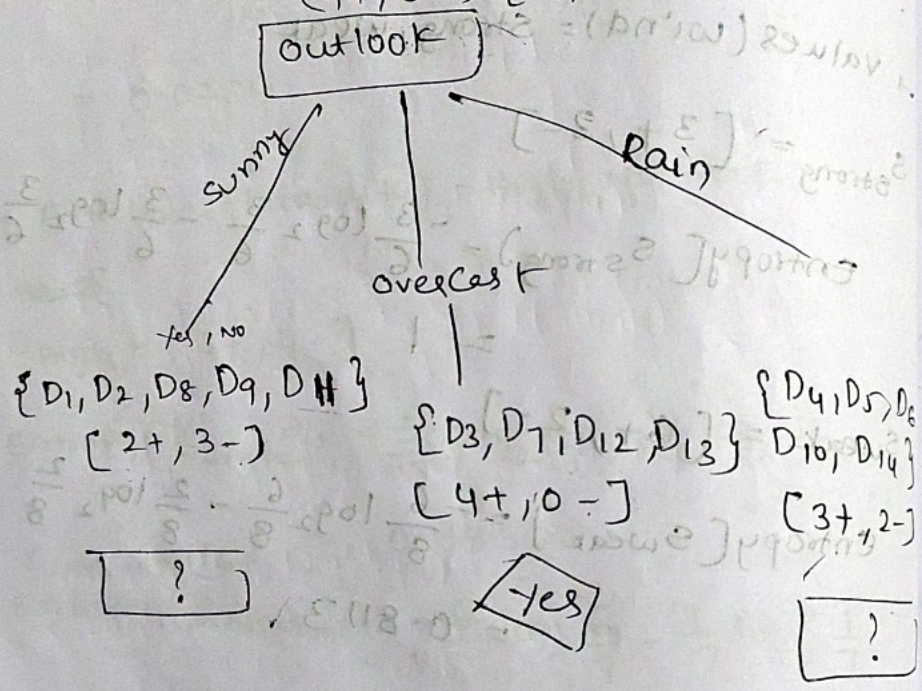
$$= 0.0478$$

→ Gain (S, outlook) = 0.2468 → (highest so root node)

Gain (S, Temp) = 0.0289

Gain (S, Humidity) = 0.1516

Gain (S, wind) = 0.0478



Day	Temp	Humidity	wind	Play Tennis
D1	Hot	High	Weak	NO
D2	Hot	High	Strong	NO
D8	Mild	High	Weak	NO
D9	Cool	Normal	Weak	Yes
D11	Mild	Normal	Strong	Yes

→ Attributes Temp

values (Temp) = Hot, mild, Cool

$S_{sunny} = [2+, 3-]$

$Entropy(S_{sunny}) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}$

$= 0.97$

$S_{Hot} = [0+, 2-]$

$Entropy(S_{Hot}) = -\frac{0}{2}\log_2\frac{0}{2} - \frac{2}{2}\log_2\frac{2}{2}$

$= 0$

$S_{mild} = [1+, 1-]$

$Entropy(S_{mild}) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log\frac{1}{2}$

$= 1$

S_{cool} [1+, 0-]

$Entropy(S_{cool}) = -\frac{1}{1}\log_2 1 - \frac{0}{1}\log 0$

$= 0$

$Gain(S_{sunny}, Temp)$

$= Entropy(S) - \sum_{v \in \{Hot, mild, Cool\}} \frac{|S_v|}{|S|} Entropy(S_v)$

$= Entropy(S) - \frac{2}{5} Entropy(S_{Hot}) - \frac{2}{5} Entropy(S_{mild}) - \frac{1}{5} Entropy(S_{cool})$

$= 0.97 - \frac{2}{5}\times 0 - \frac{2}{5}\times 1 - \frac{1}{5}\times 0 = 0.570$

Attribute : Humidity

Values (Humidity) = High, Normal

$S_{high} = [0+, 3-]$

$Entropy(S_{High}) = -\frac{0}{3}\log_2\frac{0}{3} - \frac{3}{3}\log\frac{3}{3}$

$= 0$

$S_{Normal} = [2+, 0-]$

$Entropy(S_{Normal}) = 0$

$Gain(S_{sunny}, Humidity)$

$= Entropy(S) - \sum_{v \in (High, Normal)} \frac{|S_v|}{|S|} Entropy(S_v)$

$= Entropy(S) - \frac{3}{5} Entropy(S_{high}) - \frac{2}{5} Entropy(S_{Nor}$

$= 0.97 - \frac{3}{5} \times 0 - \frac{2}{5} \times 0 = 0.97$

→ Attribute : Wind

Values (wind) = strong, weak

$S_{strong} = [1+, 1-]$

$Entropy(S_{strong}) = 1$

$S_{weak} = [1+, 2-]$

$Entropy(S_{weak}) = -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3}$

$= 0.9183$

Gain(S sunny, wind) →

= Entropy(S) − $\sum_{v \in (strong, weak)} \frac{|S_v|}{|S|}$ Entropy(S_v)

= Entropy(S) − $\frac{2}{5}$ Entropy(S_{strong})

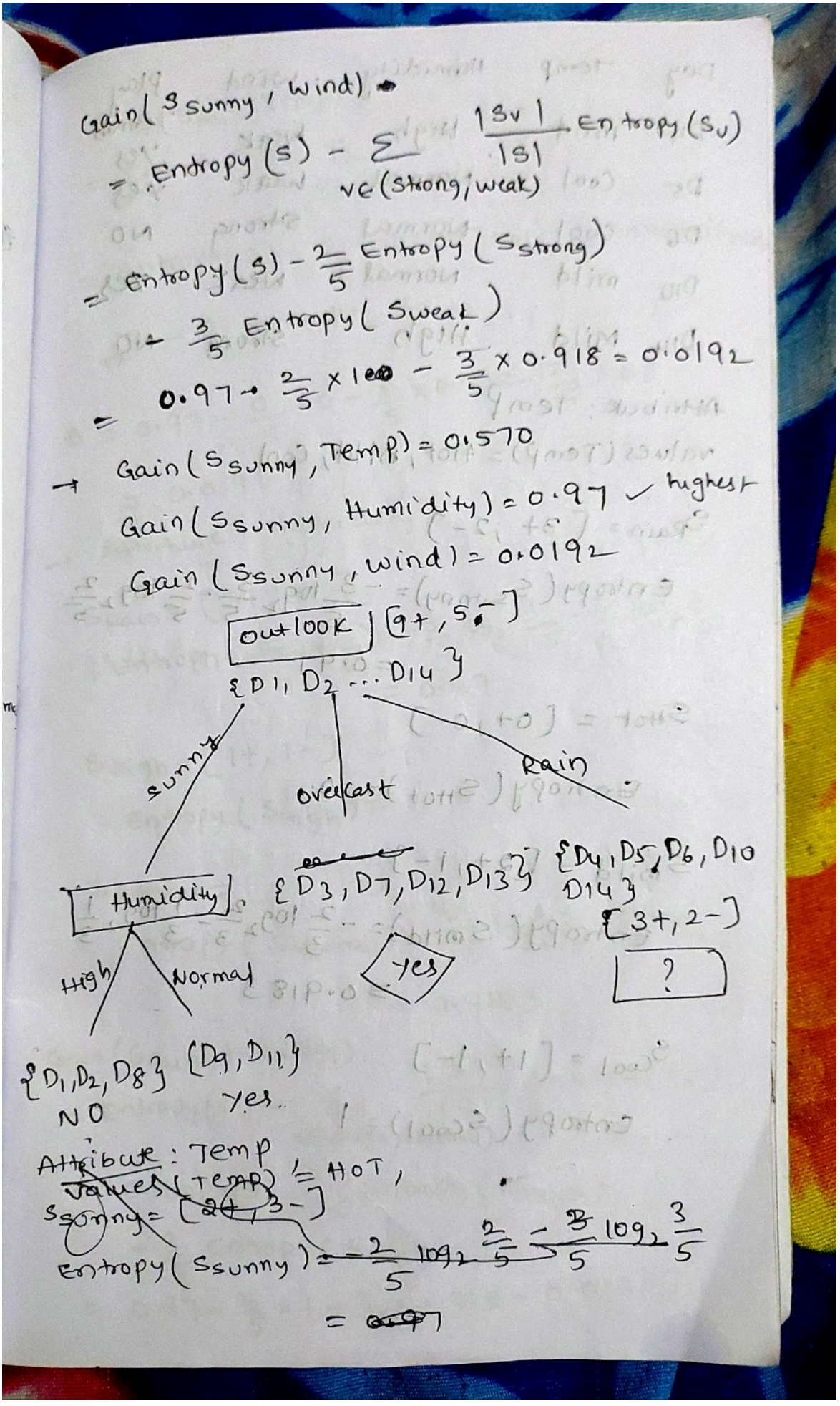
= $\frac{3}{5}$ Entropy(S_{weak})

= 0.97 − $\frac{2}{5} \times 1$ − $\frac{3}{5} \times 0.918 = 0.0192$

→ Gain(S_{sunny}, Temp) = 0.570

Gain(S_{sunny}, Humidity) = 0.97 ✓ highest

Gain(S_{sunny}, wind) = 0.0192



Attribute : Temp

Values (Temp) = HOT,

S_{sunny} = [2+, 3−]

Entropy(S_{sunny}) = $-\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5}$

= 0.97

Day	Temp	Humidity	Wind	Play Tennis
D_4	mild	High	Weak	Yes
D_5	Cool	Normal	Weak	Yes
D_6	Cool	Normal	Strong	NO
D_{10}	mild	Normal	Weak	Yes
D_{14}	Mild	High	Strong	NO

Attribute: Temp

values (Temp) = Hot, mild, Cool

$S_{Rain} = [3+, 2-]$

$$Entropy(S_{sunny}) = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}$$

$$= 0.97$$

$S_{Hot} = [0+, 0-]$

$$Entropy(S_{Hot}) = 0$$

$S_{mild} = [2+, 1-]$

$$Entropy(S_{mild}) = -\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3}$$

$$= 0.9183$$

$S_{Cool} = [1+, 1-]$

$$Entropy(S_{Cool}) = 1$$

Gain(S_{Rain}, Temp.)

$= \text{Entropy}(S) - \sum_{v \in (Hot, mild, Cool)} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$

$= \text{Entropy}(S) - \frac{0}{5}\text{Entropy}(S_{Hot}) - \frac{3}{5}\text{Entropy}(S_{mild}) - \frac{2}{5}\text{Entropy}(S_{cool})$

$= 0.97 - \frac{0}{5} \times 0 - \frac{3}{5} \times 0.918 - \frac{2}{5} \times 0$

$= 0.0192$

→ Attribute : Humidity

$S_{Rain} = [3+, 2-]$

$\text{Entropy}(S_{sunny}) = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}$

$= 0.97$

$S_{High} = [1+, 1-]$

$\text{Entropy}(S_{High}) = 1$

$S_{Normal} = [2+, 1-]$

$\text{Entropy}(S_{Normal}) = -\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3}$

$= 0.9183$

Gain(S_{Rain}, Humidity)

$= \text{Entropy}(S) - \sum_{v \in (High, Normal)} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$

$= \text{Entropy}(S) - \frac{2}{5}\text{Entropy}(S_{High}) - \frac{3}{5}\text{Entropy}(S_{Normal})$

$= 0.97 - \frac{2}{5} \times 1 - \frac{3}{5} \times 0.918 = 0.0192$

→ Attribute : Wind
values(wind) = strong, weak

$S_{Rain} = [3+, 2-]$

$Entropy(S_{sunny}) = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}$

$= 0.97$

$S_{strong} = [0+, 2-]$

$Entropy(S_{strong}) = 0$

$S_{weak} = [3+, 0-]$

$Entropy(S_{weak}) = 0$

$Gain(S_{Rain}, Wind) = Entropy(S) - \sum_{v \in (Strong, weak)} \frac{|S_v|}{|S|} Entropy$

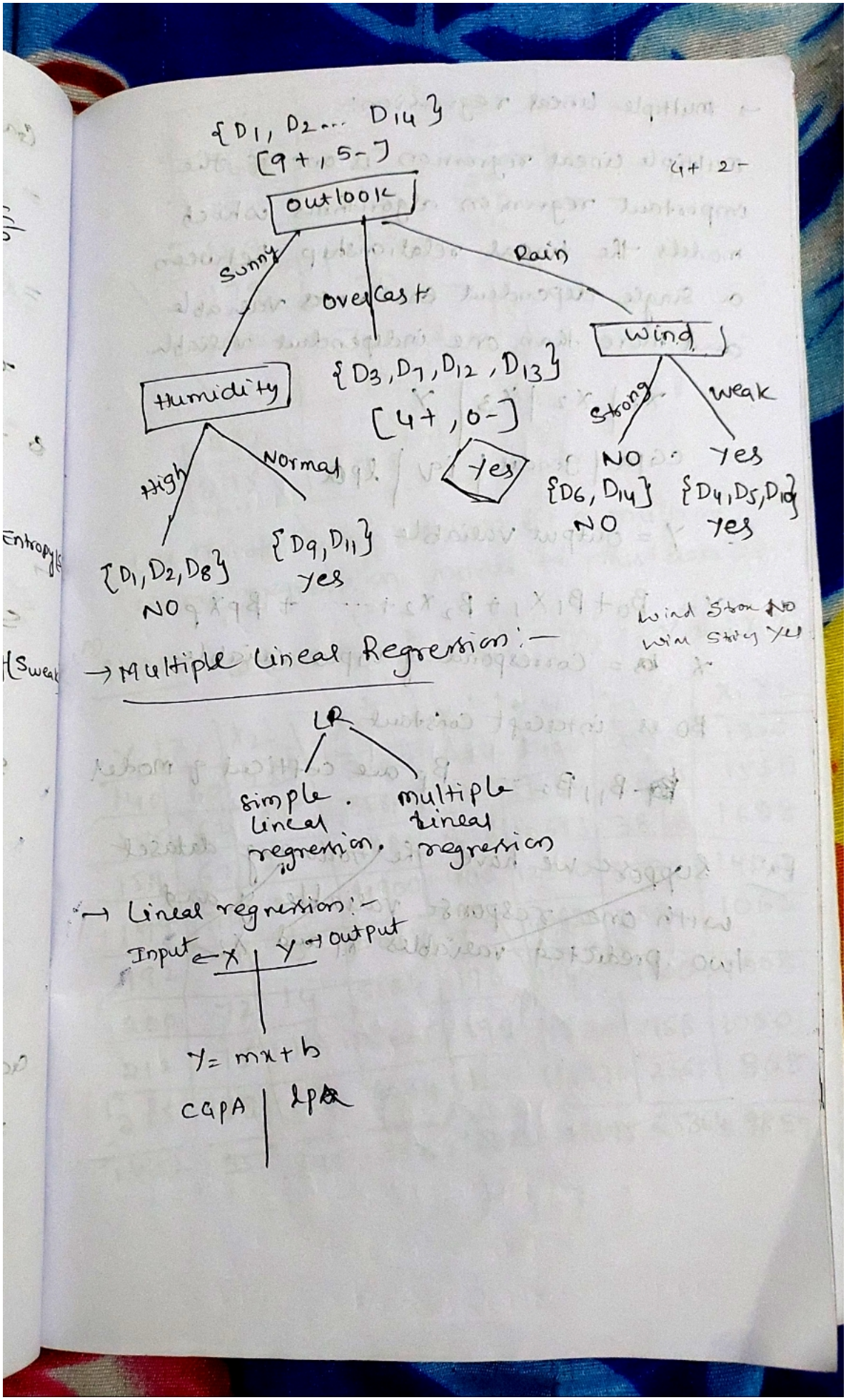
$= Entropy(S) - \frac{2}{5} Entropy(S_{strong}) - \frac{3}{5} Entropy(S_{weak})$

$= 0.97 - \frac{2}{5} \times 0 - \frac{3}{5} \times 0 = 0.97$

→ $Gain(S_{Rain}, Temp) = 0.0192$

$Gain(S_{Rain}, Humidity) = 0.0192$

$Gain(S_{Rain}, Wind) = 0.97$ ✓


{D1, D2 ... D14}
[9+, 5-]
Outlook
Sunny
Overcast
Rain
Humidity
{D3, D7, D12, D13}
[4+, 0-]
Wind
High
Normal
Strong
Weak
Yes
{D1, D2, D8}
NO
{D9, D11}
yes
NO
{D6, D14}
NO
{D4, D5, D10}
yes
Wind Strong NO
Wind Strong YES
→ Multiple Linear Regression:-
LR
simple linear regression
multiple linear regression
→ Linear regression:-
Input ← X | Y → output
y = mx + b
CGPA | LPA

→ Multiple linear regression:-

Multiple linear regression is one of the important regression algorithms which models the linear relationship between a single dependent ~~continuous~~ variable and more than one independent variable.

x_1	x_2	x_3	y
cgpa	gender	iq	lpa

y = output variable

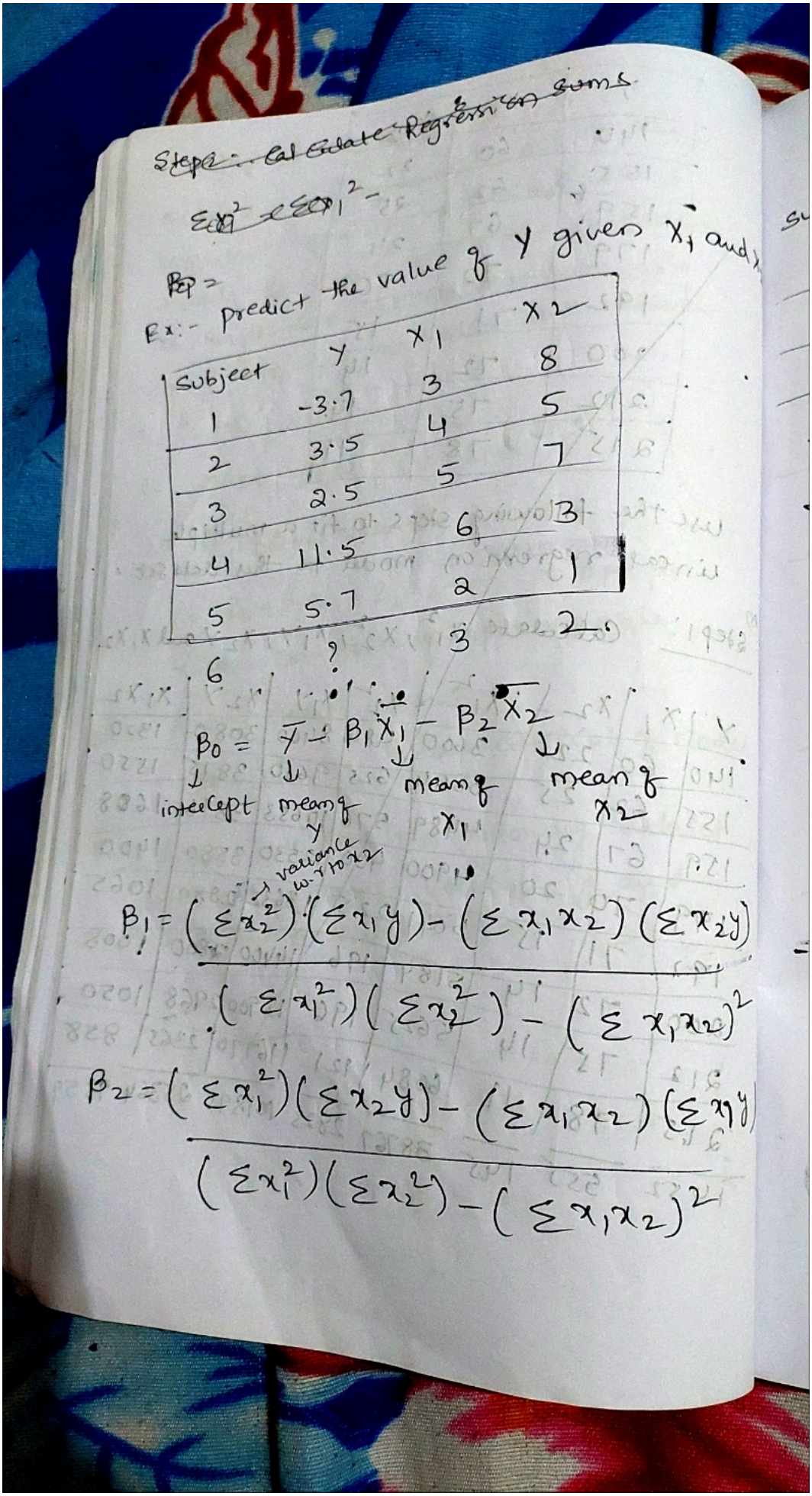
$$y = B_0 + B_1 x_1 + B_2 x_2 + \cdots + B_p x_p$$

x = Corresponding input variable

B_0 is intercept constant

$B_1, B_2 \cdots B_p$ are cofficient of model

Ex:- Suppose we have the following dataset with one response variable y and two predicted variables x_1 and x_2



Steps:- ~~Calculate Regression sums~~

Σx_1^2 Σx_2^2 $\Sigma x_1 y$ $\Sigma x_2 y$ $\Sigma x_1 x_2$

Rep =

Ex:- predict the value of Y given X_1 and X_2

Subject	Y	X_1	X_2
1	-3.7	3	8
2	3.5	4	5
3	2.5	5	7
4	11.5	6	3
5	5.7	2	1
6	?	3	2

$$\beta_0 = \bar{Y} - \beta_1 \bar{X}_1 - \beta_2 \bar{X}_2$$

β_0 ↓ intercept; $\bar{Y}$ ↓ mean of Y; $\bar{X}_1$ ↓ mean of X_1; $\bar{X}_2$ ↓ mean of X_2

β_1 ↓ variance w.r.t. X_2

$$\beta_1 = \frac{(\Sigma x_2^2)(\Sigma x_1 y) - (\Sigma x_1 x_2)(\Sigma x_2 y)}{(\Sigma x_1^2)(\Sigma x_2^2) - (\Sigma x_1 x_2)^2}$$

$$\beta_2 = \frac{(\Sigma x_1^2)(\Sigma x_2 y) - (\Sigma x_1 x_2)(\Sigma x_1 y)}{(\Sigma x_1^2)(\Sigma x_2^2) - (\Sigma x_1 x_2)^2}$$

$\sum x^2 = \sum X^2 - \frac{(\sum X)^2}{N}$

subject	Y	X_1	X_2	X_1^2	X_2^2	X_1X_2	X_1Y	X_2Y
1	-3.7	3	8	9	64	24	-11.1	-29.6
2	3.5	4	5	16	25	20	14	17.5
3	2.5	5	7	25	49	35	12.5	17.5
4	11.5	6	3	36	9	18	69	34.5
5	5.7	2	1	4	1	2	11.4	5.7
	19.5	20	24	90	148	99	95.8	45.6

$\sum x_1^2 = \sum X_1^2 - \frac{(\sum X_1)^2}{N}$ → regression sum

↓ variance w.r.t to x_2

$= 90 - \frac{20 \times 20}{5} = 10$

$\sum x_2^2 = \sum X_2^2 - \frac{(\sum X_2)^2}{N}$

$= 148 - \frac{24 \times 24}{5} = 32.8$

$\sum x_1 y = \sum X_1 Y - \frac{(\sum X_1)(\sum Y)}{N}$

$= 95.8 - \frac{(20)(19.5)}{5} = 17.8$

$\sum x_2 y = \sum X_2 Y - \frac{(\sum X_2)(\sum Y)}{N}$

$= 45.6 - \frac{(24)(19.5)}{5} = -48$

$$\Sigma x_1 x_2 = \Sigma X_1 X_2 - \frac{(\Sigma x_1)(\Sigma x_2)}{N}$$

$$= 99 - \frac{(20 \times 24)}{5} = 3$$

$$\beta_1 = \frac{(\Sigma x_2^2)(\Sigma x_1 y) - (\Sigma x_1 x_2)(\Sigma x_2 y)}{(\Sigma x_1^2)(\Sigma x_2^2) - (\Sigma x_1 x_2)^2}$$

$$= \frac{(32.8)(17.8) - (3)(-48)}{(10)(32.8) - (3)^2}$$

$$= 2.28$$

$$\beta_2 = \frac{(\Sigma x_1^2)(\Sigma x_2 y) - (\Sigma x_1 x_2)(\Sigma x_1 y)}{(\Sigma x_1^2)(\Sigma x_2^2) - (\Sigma x_1 x_2)^2}$$

$$= \frac{(10)(-48) - (3)(17.8)}{(10)(32.8) - (3)^2}$$

$$= -1.67$$

$$\beta_0 = \bar{Y} - \beta_1 \bar{X}_1 - \beta_2 \bar{X}_2$$

$$= \frac{19.5}{5} - \frac{2.28 \times 20}{5} - \frac{-1.67 \times 24}{5}$$

$$= 2.796$$

Final regression equation is

$$Y = 2.796 + 2.28X_1 - 1.67X_2$$

Now given $X_1 = 3$ and $X_2 = 2$ $Y = ?$

$$Y = 2.796 + 2.28 \times 3 - 1.67(2)$$

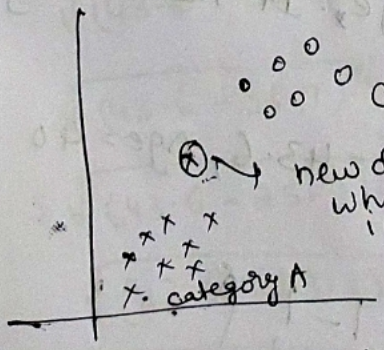
$$= 6.296$$

→ KNN algorithm :-
K Nearest Neighbour (K values must be odd)

→ It is a supervised learning algorithm that classifies a new data points into the target class, depending on the features of its neigbouring data points.

→ K-NN is a non-parametric algorithm which means it does not make any assumption on underlying data.

→ It is called a lazy learner algorithm



→ we will calculate distance of new data point from every data point and assume particular K=3 (nearest three)

→ nearest three data points. Then we will choose the ones which are of maximum type.

→ The K values will be decided based on trail and error so that reduces the bias and variance.

Ex:-

	BMI	Age	Sugar	Rank
1	33.6	50	1	2
2	26.6	30	0	
3	23.4	40	0	
4	43.1	67	0	
5	35.3	23	1	
6	35.9	67	1	
7	36.7	45	1	
8	25.7	46	0	3
9	23.3	29	0	
10	31	56	1	

→ Apply K nearest neighbor classifier to predict the diabetic patient with the given features BMI, Age. If the training example are,

- Assume K = 3
- Test example BMI = 43.6, Age = 40 Sugar = ?

36.7 - 1 - 1

33.6 2 - 1

3 - 0

→ ~~Distance~~ =

→ First Calculate the distance between the test instance and training instances

Test example

BMI = 43.6, Age = 40, Sugar = ?

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

x_2: Test data variable; x_1: Training data variable; y_2: Test data variable; y_1: Training data variable

	Distance		Rank
1	$\sqrt{(43.6-33.6)^2 + (40-50)^2}$	14.14	2
	$\sqrt{(43.6-26.6)^2 + (40-30)^2}$	19.72	
	$\sqrt{(43.6-23.4)^2 + (40-40)^2}$	20.20	
	$\sqrt{(43.6-43.1)^2 + (40-67)^2}$	27.00	
	$\sqrt{(43.6-35.3)^2 + (40-23)^2}$	18.92	
	$\sqrt{(43.6-35.9)^2 + (40-67)^2}$	28.08	
	$\sqrt{(43.6-36.7)^2 + (40-45)^2}$	8.52	1
	$\sqrt{(43.6-25.7)^2 + (40-46)^2}$	18.88	3
	$\sqrt{(43.6-23.3)^2 + (40-29)^2}$	23.09	
	$\sqrt{(43.6-31)^2 + (40-56)^2}$	20.39	

→ ~~out of 3 cases '2' cases having Sugar (maximum~~ Sug

→ out of 3 nearest neighbours 2 one and one is zero, so the new test example is classified as

sugar = 1

Ex:- K = 3 → Normal

	Height (cm)	weight (KG)	Class
1	167	51	Underweight
2	182	62	Normal
3	176	69	Normal
4	173	64	Normal
5	172	65	Normal
6	174	56	Underweight
7	169	58	Normal 1
8	173	57	Normal 3
9	170	55	Normal 2
10	170	57	?

→

~~Rank~~	Distance	Rank	
169	1.4	1	→ Normal
170	2	2	→ Normal
173	3	3	→ Normal

Maximum likelihood and Least Squared.

→ To find the maximum likelihood hypothesis in bayesian learning

$$h_{ML} = \underset{h \in H}{\operatorname{argmax}} \; p(D|h)$$

probability density function

Let us take training instances $(x_1, x_2 \dots x_n)$ and consider target values $D = (d_1, d_2 \dots d_m)$

∴ we write $P(D|h)$ as product of $P(d_i|h)$

$$h_{ML} = \underset{h \in H}{\operatorname{argmax}} \prod_{i=1}^{n} p(d_i|h)$$

By assuming normal distribution

$$f(x|\mu) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

(x: variable, μ: mean, σ: standard deviation)

$$h_{ML} = \underset{h \in H}{\operatorname{argmax}} \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-1}{2\sigma^2}(d_i-\mu)^2}$$

$$= \underset{h \in H}{\operatorname{argmax}} \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(d_i-h(x_i))^2}$$

$\log e^{-x} = -x$

→ use log function

$$= \underset{h \in H}{\operatorname{argmax}} \sum_{i=1}^{n} \ln \frac{1}{\sqrt{2\pi\sigma^2}} \left(-\frac{1}{2\sigma^2}(d_i - h(x_i))^2\right)$$

$$= \underset{h \in H}{\operatorname{argmax}} \sum_{i=1}^{n} \ln \frac{1}{\sqrt{2\pi\sigma^2}} \left(-\frac{1}{2\sigma^2}(d_i - h(x_i))^2\right)$$

$$= \underset{h \in H}{\operatorname{argmax}} \sum_{i=1}^{m} -\frac{1}{\sqrt{2\pi\sigma^2}} (d_i - h(x_i))^2$$

↳ Constant

$$= \underset{h \in H}{\operatorname{argmax}} \sum_{i=1}^{m} -\frac{1}{\sqrt{2\pi\sigma^2}} \frac{1}{2\sigma^2} (d_i - h(x_i))^2$$

↳ Constant
Can remove

$$= \underset{h \in H}{\operatorname{argmax}} \sum_{i=1}^{m} (d_i - h(x_i))^2$$

$$= \underset{h \in H}{\operatorname{argmin}} \sum_{i=1}^{m} (d_i - h(x_i))^2$$

→ square

$$h_{ML} = \underset{h \in H}{\operatorname{argmin}} \sum_{i=1}^{m} (d_i - h(x_i))^2$$

↓ minimum error

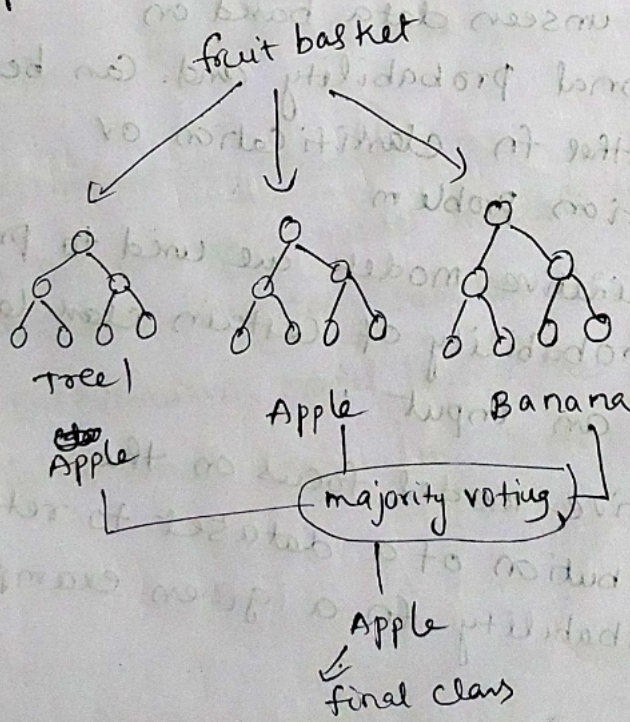
The maximum likelihood hypothesis which has minimum least squared error between the hypothesis.

Random Forest model :-

→ Random Forest is a supervised machine learning algorithm that is used widely in classification and Regression problem.

→ It builds decision trees on different samples and takes their majority vote for classification

Ex:- Consider the fruit basket. ~~as the~~ Now 'n' number of samples are taken from the fruit basket and individual decision tree is constructed for each sample. Each decision tree will ~~do~~ generate an output. the final output is considered based on majority voting.



Step1: Select random samples from a given data or training set

Step2: This algorithm will construct a decision tree for every training data

Step3: voting will take place by averaging the decision tree.

Step4: Final select the most voted prediction result as the final prediction result.

→ Discriminative and Generative models:-

→ machine learning models can be classified into two types of models.

→ Discriminative model makes predictions on the unseen data based on conditional probability and can be used either for classification or regression problem

→ Discriminative models are used to predict the probability of certain class label given an input.

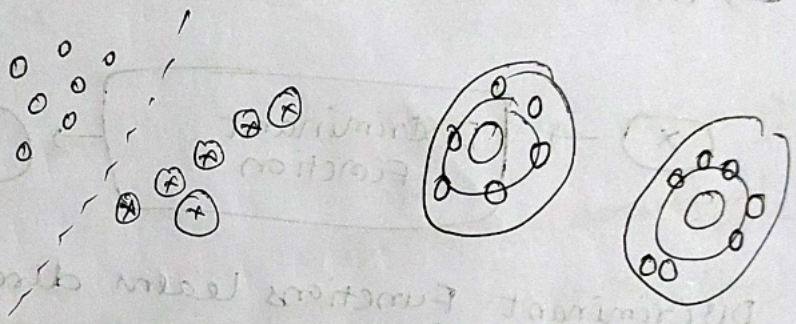
→ Generative model focus on the distribution of a dataset to return a probability for a given example

→ Discriminative models focus on predicting labels, while generative model focus on modeling the distribution of data.

→ ~~Generative vs~~ Discriminative vs Generative models

	Discriminative	Generative model
Goal	Directly estimate $P(Y\|x)$	estimate $P(x\|y)$ to then decide $P(Y\|x)$
what's learned	Decision boundary	Probability of the data

~~Illustration~~ illustration



Example Regression, SVMs · Naive Bayes

→ Discriminative classifier.

→ A Discriminative model models the decision boundary between the classes

→ Discrinative model learns the Conditional probability Distribution $P(Y|x)$

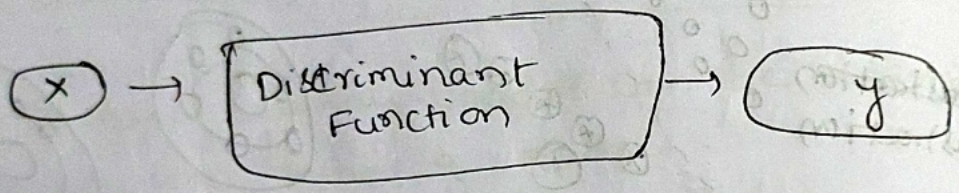
→ Generative classifiers

→ A Generative model learns the joint Probability distribution $P(x, y)$

→ It Predicts the Conditional Probability with the help of Bayes theorem

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

→ Discriminant Function :-



→ Discriminant Functions learns direct mapping between feature vector x and label y.

→ In a binary classification set up with m features, the simplest discriminant function is very similar to the linear regression.

$$y = w_0 + w_1 x_1 + \cdots + w_m x_m$$

$$y = w_0 + w^T x$$

label ↓ bias ↓ weight vector ← feature vector

→ Geometrically, the simplest discriminant function $y = w_0 + w^T x$ represents a hyperplane in $m-1$ dimensional space where m is the number of features

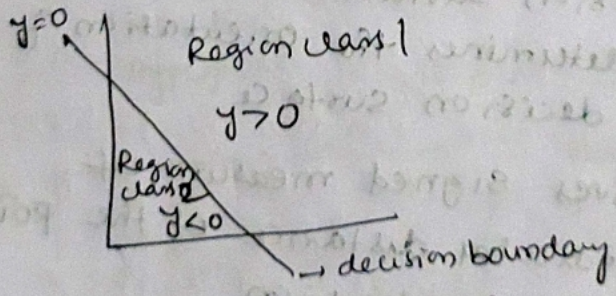
features (m)	Discriminant Function
1	point
2	Line
3	plane.
4	Hyperplane in 3D space.
⋮	
m	Hyperplane in (m-1)D space

→ (m-1) subspace in a m dimensional space.

→ (m-1) D hyperplane (which is line in this case) divides features space into two regions one for each class.

Region for class 1, where $y > 0$

Region for class 2, where $y < 0$



→ The decision boundry between two classes is represented with (m-1) D hyperplane.

$w_0 + w^T x = 0$

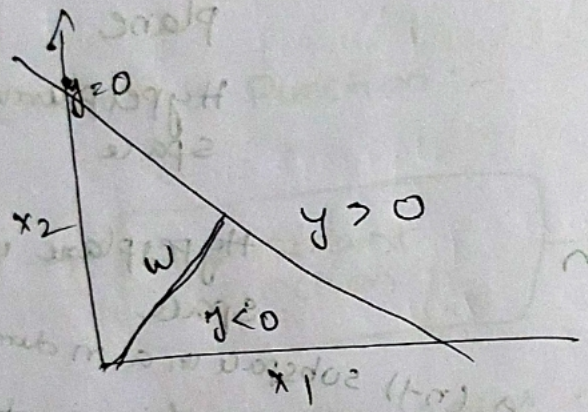
→ Consider two points $x^{(A)}$ and $x^{(B)}$ on the decision surface, we will have

$$y^{(A)} = w_0 + w^T x^{(A)} = 0$$

$$y^{(B)} = w_0 + w^T x^{(B)} = 0$$

$$y^{(A)} - y^{(B)}$$

$$\Rightarrow w^T (x^{(A)} - x^{(B)}) = 0$$



→ the vector w is orthogonal to every vector lying within the decision surface, hence it determines the orientation of the decision surface

→ w_0 determines the location of decision surface

→ w determines the orientation of the decision surface.

→ y gives signed measure of perpendicular distance of the point x from the decision

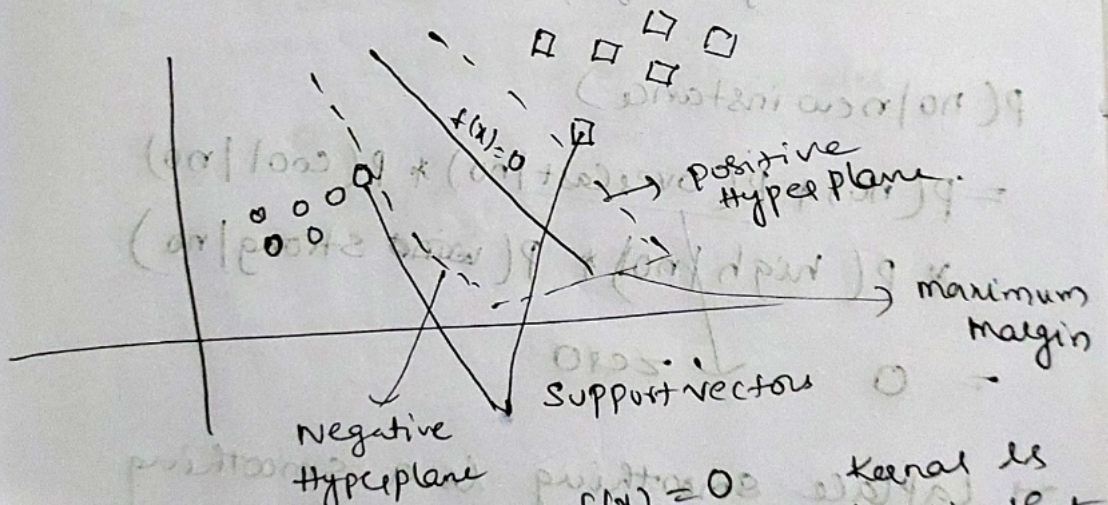
→ Decision surface divides feature space into two regions.

Support vector machine :

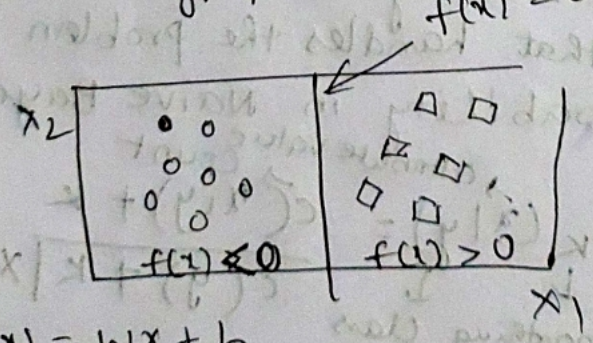
→ SVM is machine learning algorithm which can be used for both classification or regression challenges.

→ In this algorithm, we ~~plot~~ plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate.

→ Then, we perform classification by finding the hyper-plane that differentiate the two classes very ~~well~~ well.



→



Kernal is a function that maps input data into a highes dimensional space.

$f(x) = wx + b$

w is the normal to the line, x is input vector and b is bias, w is known as the weight vector

→ Laplace smoothing or

→ zero probability problem

	yes	NO
outlook		
sunny	2/9	3/5
overcast	4/9	0/5
Rainy	3/9	3/5

→ clarity new example

(outlook = overcast, temp = cool,
Humidity = High, wind = strong)

P(no | new instance)

= P(no) * P(overcast | no) * P(cool | no)
* P(high | no) * P(strong | no)

= 0 → zero

→ Laplace smoothing is a smoothing technique that handles the problem of zero probability in Naive Bayes

$$P_{LAPK}(\vec{x}|y) = \frac{c(\vec{x}, y) + k}{c(y) + k|X|}$$

$\vec{x}$ → attribute value, count

P_{LAPK} → Laplace Approximation Probability

k → Smoothing Parameter, $k > 0$

y → class

$c(x|y)$ is the number of times x appears in y examples

$c(y)$ is number of y examples
(suppose here y is no)
if y is yes number of yes examples

X is number of values of that feature
Ex:- outlook ~~no~~ no of values
↓
sunny, overcast, Rainy = 3

$$P_{LAP,k}(\text{outlook} = \text{overcast} | \text{no}) = \frac{c(\text{outlook} = \text{overcast} | \text{no}) + k}{c(\text{no}) + k|X|} \quad |X| \rightarrow \text{modules}$$

$$= \frac{0+1}{5+1*3} = \frac{1}{8}$$