Analysis Means

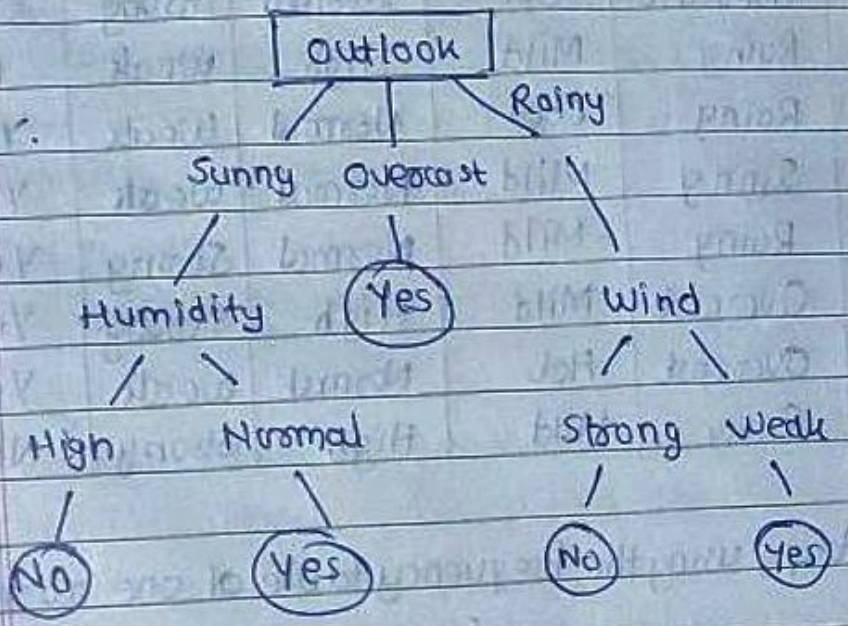
Page No. 67
Date

Decision Tree:

A decision tree is a diagram or chart that people used to determine a course of action or show a statistical probability. Each branch of decision tree represents a possible outcome, decision or reaction. The farthest branches represent the end results.

Individuals deploy decision trees in variety of situations like "Should I play golf?" or something industrial or complex.

Ex: Decision tree for "Play Golf"



Entropy:

- A measure or disorder or uncertainty and the goal of machine learning models and data scientists in general is to reduce uncertainty

Take a look at this table constructed from decision tree from last page:

Predictors | Target

Outlook	Temp	Humidity	Wind	Play Golf
Rainy	Hot	High	Weak	No
Rainy	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Sunny	Mild	High	Weak	Yes
Sunny	Cool	Normal	Weak	Yes
Sunny	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Rainy	Mild	High	Weak	No
Rainy	Cool	Normal	Weak	Yes
Sunny	Mild	Normal	weak	Yes
Rainy	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Sunny	Mild	High	Strong	No

a) Entropy using the frequency table of one attribute:

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

Play Golf	
Y	N
9	5

→ Entropy (Play Golf)

Entropy (5, 9)

Entropy (5/14, 9/14)

Entropy (0.36, 0.64)

$= -(0.36 \log_2 0.36) - (0.64 \log_2 0.64)$

$= 0.94$

Page No. 69
Date

1) Entropy using the frequency table of two attributes

$$E(T,X) = \sum_{c \in X} P(c) E(c)$$

		Play Golf		
		Y	N	
outlook	Sunny	3	2	5
	Over cast	4	0	4
	Rainy	2	3	5
				14

E (Play Golf, Outlook)

= P (Sunny) * E (3,2) + P (Over cast) * E (4,0)
+ P (Rainy) * E (3,2)

= 5/14 * (0.971) + 4/14 * 0 + 5/14 * 0.971

= 0.693

Random Forest:

- Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees
- The basic idea behind this is to combine multiple decision trees and determining the final output rather than relying on individual decision trees.
- A random forest algorithm, which is used for both classification & regression, creates decision trees from data samples and then gets a prediction from each of them and finally selects the best one by the means of voting.

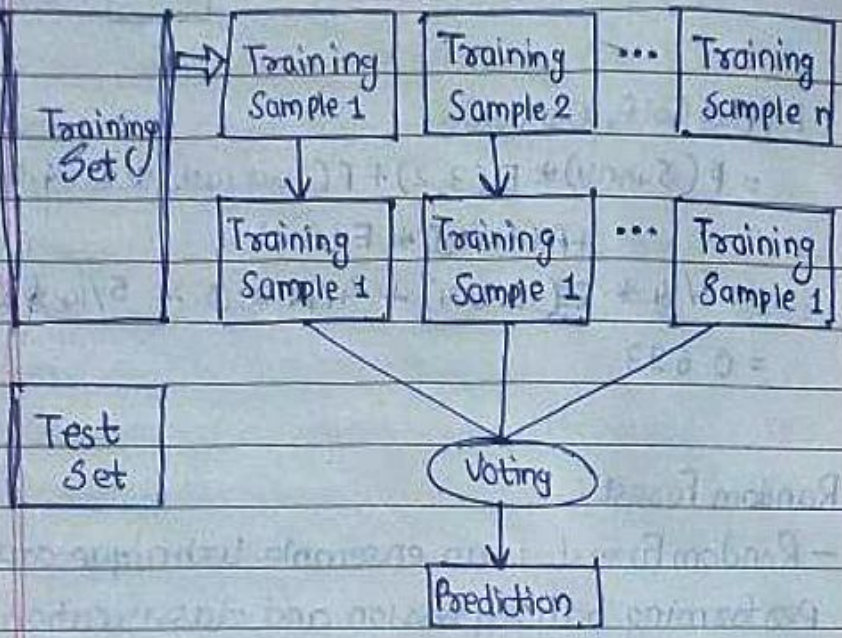
Working of random forest algorithm:

Step 1 → Start with the selection of random samples from the given data set.

Step 2 → A decision tree will be constructed. Then it will get prediction from every decision tree.

Step 3 → Voting will be performed for every predicted result.

Step 4 → Select the most voted result as final prediction result.

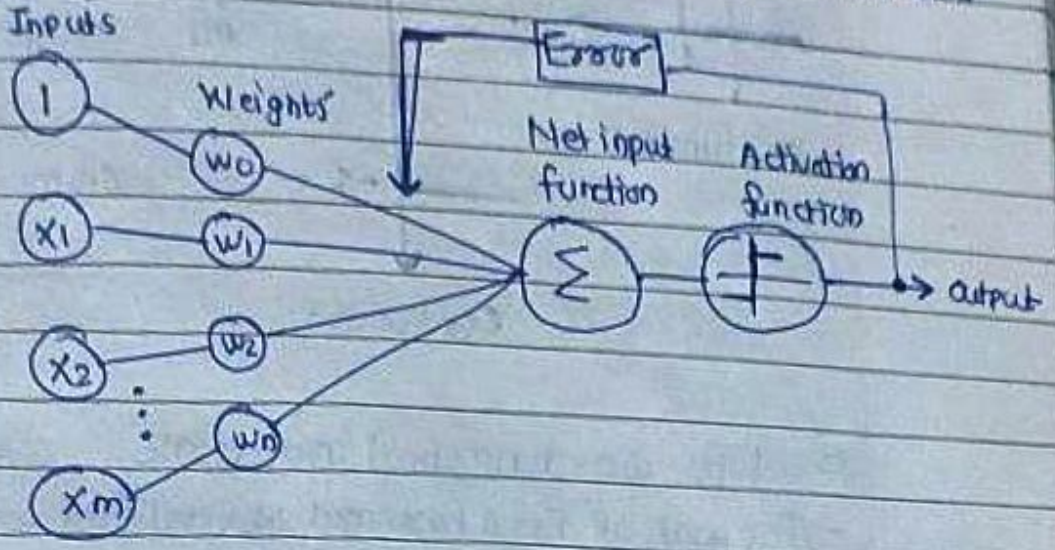


Neural Networks:

- Neural networks are artificial systems that are inspired by biological neural network.
- These systems learn to perform tasks by exposing to various datasets and examples without any task specific rules.
- The idea is that the system generates identifying characteristics from the data that they have been passed without being programmed/pre-programmed to understand these datasets.

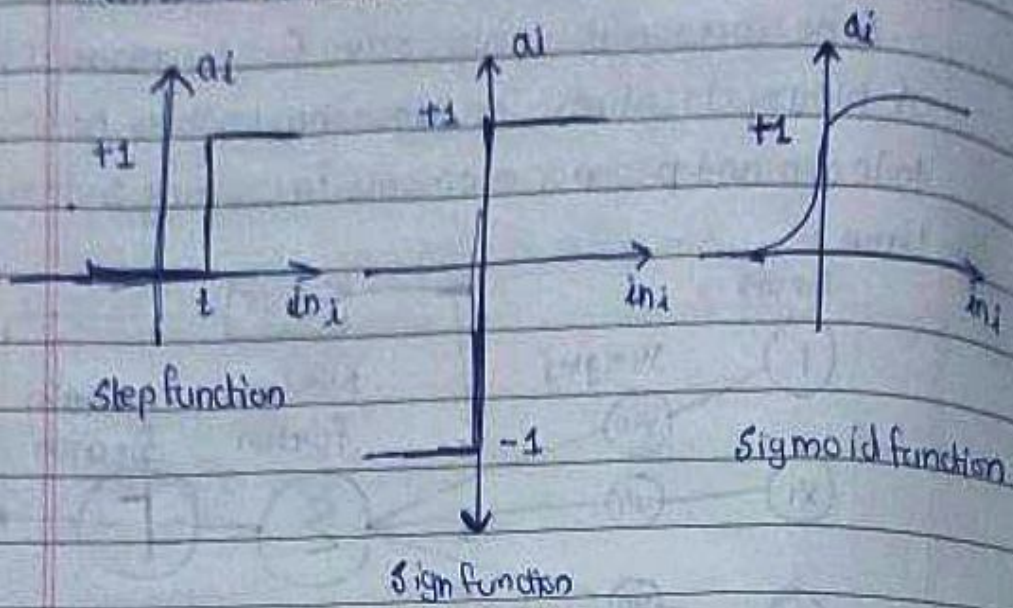
Perceptrons:

A perceptron is an algorithm for supervised learning of binary classifiers. The algorithm enables neurons to learn and process elements in training set one at a time.



- The process begins by taking all input values and multiplying them with their weights. Then all these multiplied values are added to create a weighted sum.
- The weighted sum is then applied to an activation function, producing the perceptron's output. The activation function ensures the output is mapped between required values such as (0,0) & (1,-1).
- It is important to note that the weight of an input is indicative of the strength of a node. Similarly, an input's bias value gives the ability to shift the activation function curve up or down.

Activation Functions of Perceptron



Feed forward neural network:

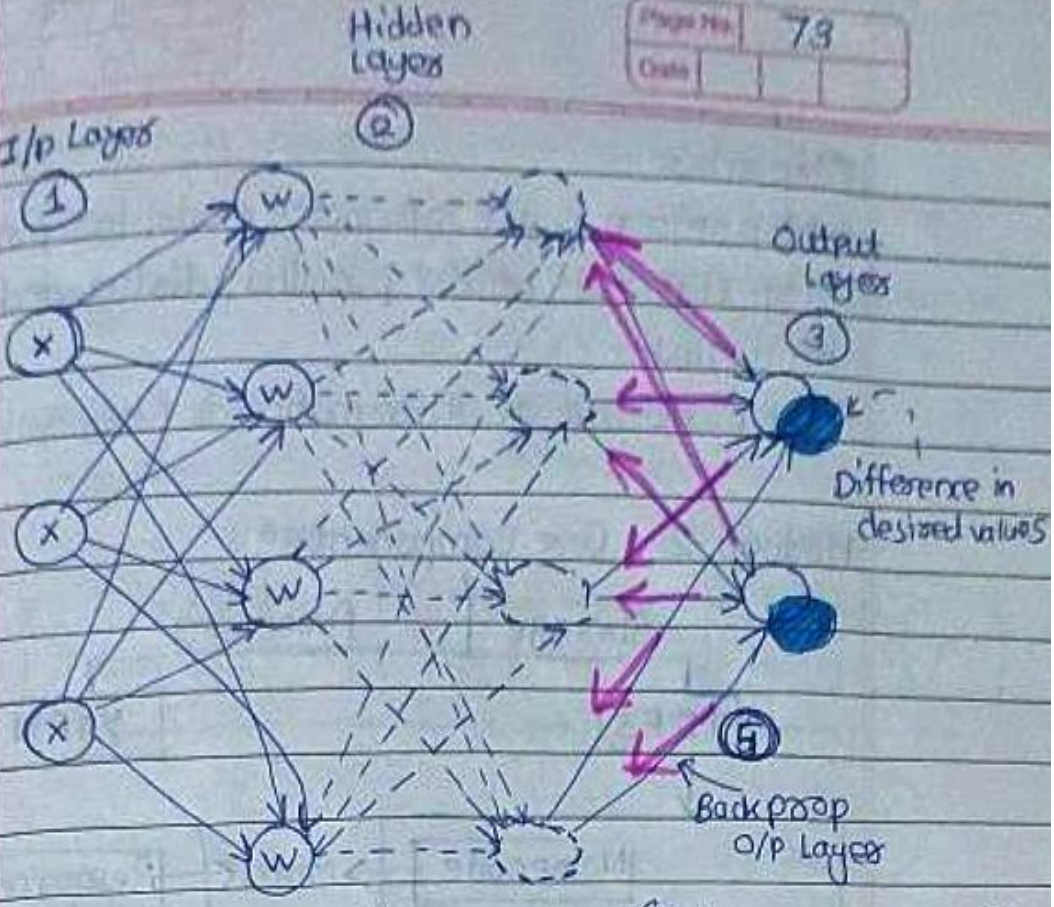
- The goal of Feed forward Neural Network is to approximate some function $f(x)$.

Ex:- for classifier $y = f(x)$ maps an input x to a category y.

A Feed Forward Neural Network defines a mapping $y = f(x; \theta)$ and learns the value of parameters θ that results in best function.

Back propagation:

- Essense of neural net training.
- method of fine tuning the weights of neural net based on error rate in the previous iteration.
- proper tuning of weights allows you to reduce error rates and to make the model reliable by increasing its generalization.



How back propagation works: (Refer upper numbers)

① Inputs x arrive through pre-connected path.

② Input is modelled using real weights W. These weights are usually randomly selected

③ Calculate the o/p for every neuron from input layer, to the hidden layers, to the output layers.

④ Calculate the error in outputs.

⑤ Travel back to output layers to the hidden layer to adjust the weights such as error is decreased

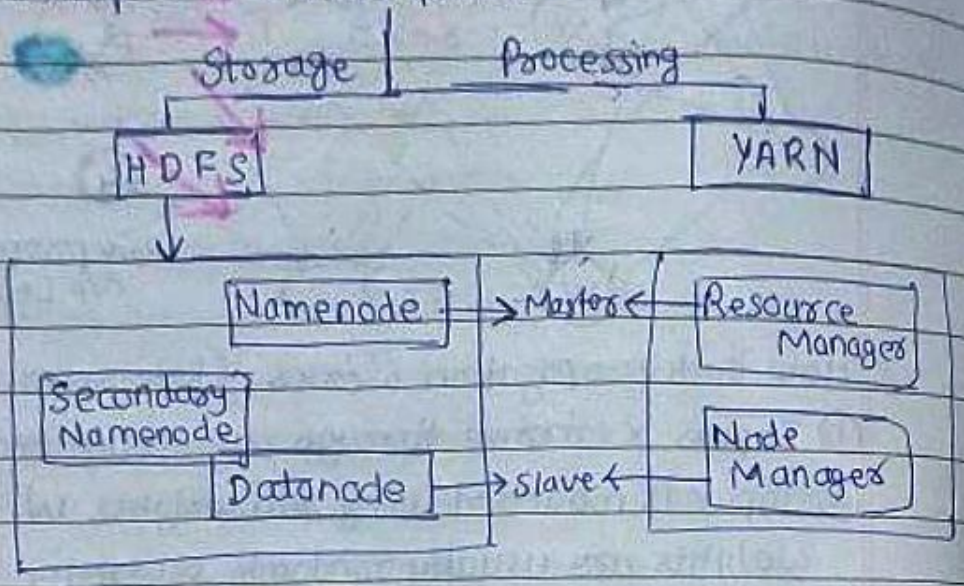
Keep repeating the process until the desired output is achieved.

- Back propagation is especially useful for deep neural networks working on error-prone projects, such as image or speech recognition.
- It takes advantage of the chain and power rule allows backpropagation to function with any no. of outputs.

Map reduce:

- Map reduce is a programming model for processing large data sets with a parallel, distributed algorithm on a cluster.
- The basic unit of information is (key-value) pair.

Hadoop 2.x Core Components:



Namenode:

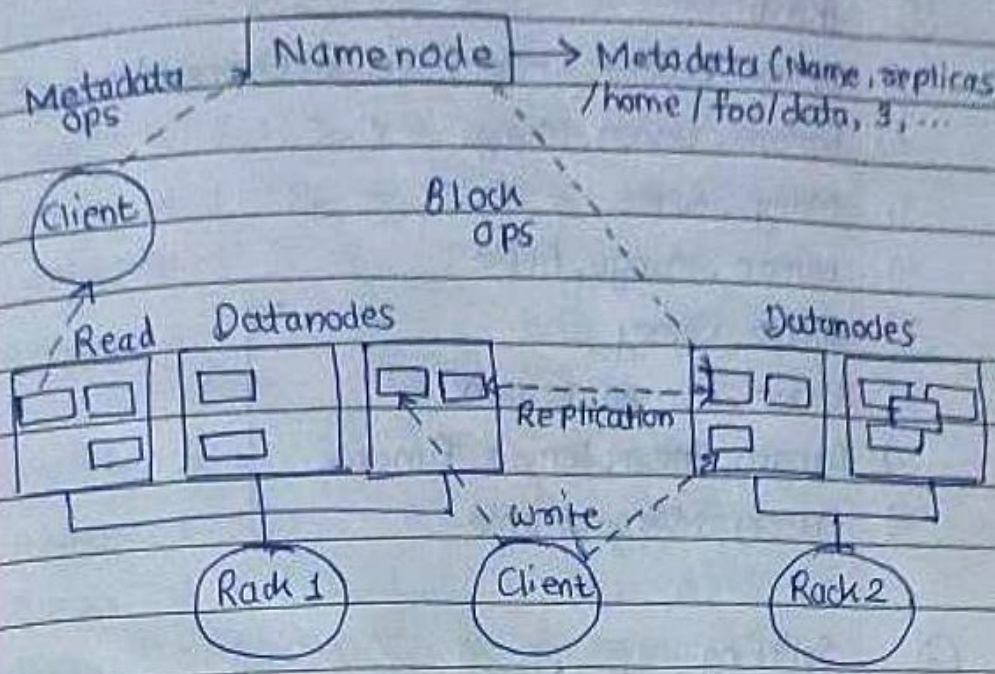
- Master of system.
- Maintains and manages the blocks which are present on data nodes.

Datanodes:

- Slaves which are deployed on each machine and provide the actual storage.
- Responsible for serving read and write requests for the clients.

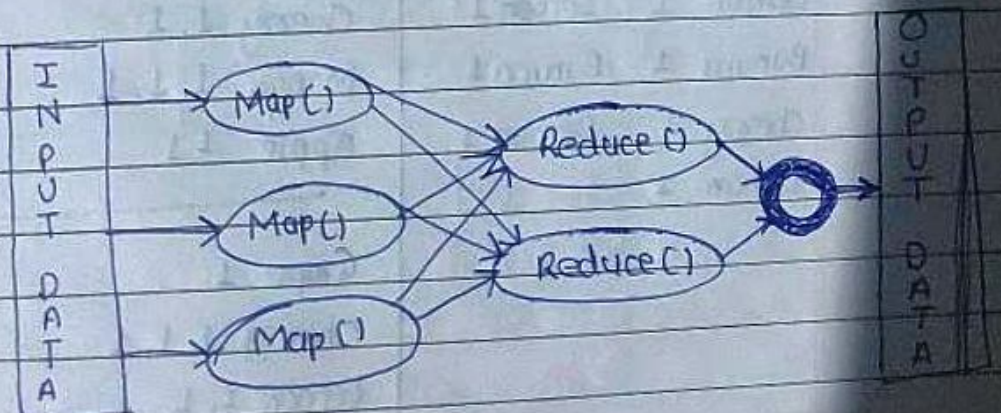
Page No. 75
Date

HDFS Architecture:



MapReduce:

- A processing technique and a program model for distributed computing based on java.
- Contains two important tasks: Map & Reduce
- Map: takes a set of data and converts it into another set of data, where individual data is broken ~~Reduce~~ in key-value pairs.
- Reduce: takes output of map as input and combines those data tuples into a smaller set of tuples.



Mapreduce example:

① Input:

1) Lemon, Banana, Lemon, Banana, Cherry, Lemon, Banana, Cherry
2) Banana, Lemon, Mango
3) Mango, Apple
4) Lemon, Mango, Apple
5) Grape, Cherry
6) Lemon, Banana, Lemon
7) Banana, Cherry, Lemon, Banana
8) Cherry, Mango, Apple

② Splitting

1) 2) 3) 4)

5) 6) 7) 8)

③ Mapping

Lemon 1
Banana 1
Lemon 1
Banana 1
Cherry 1
Lemon 1

Grape 1
Cherry 1
Lemon 1
Banana 1
Lemon 1

④ Merging + 1

Lemon 1, 1, 1, 1, 1
Banana 1, 1, 1, 1
Cherry 1, 1
Mango 1, 1, 1
Apple 1 1

Grape 1
Cherry 1, 1, 1
Lemon 1, 1
Banana 1, 1, 1
Mango 1
Apple 1

⑤ Merging - 2

Lemon 1, 1, 1, 1, 1, 1, 1

Banana 1, 1, 1, 1, 1, 1, 1

Cherry 1, 1, 1, 1, 1

Mango 1, 1, 1, 1

Apple 1, 1, 1

Grape 1

⑥ Sorting:

Apple 1, 1, 1

Banana 1, 1, 1, 1, 1, 1, 1

Cherry 1, 1, 1, 1, 1

Grape 1

Lemon 1, 1, 1, 1, 1, 1, 1

Mango 1, 1, 1, 1

⑦ Reducer:

Apple : 3

Banana : 7

Cherry : 5

Grape : 1

Lemon : 7

Mango : 4