Part A

1.

Input Layer and Layer 1 = 7*16+16bias = 128

Layer 1 and Layer 2 = 16*8+ 8Bias = 136

Layer 2 and Output Layer = 8*1+1bias = 9

Total Parameters = 128 + 136 + 9 = 273

2.

In machine Learning paradigm “black box” is a special term which describes when using some Machine Learning Model for a specific problem some situations we are unable to interpret how the model predict or make this decision. For this kind of situations, we use this term.

As an example, I consider Artificial Neural Network which is very complex data processing paradigm inspired by Biological Neural Networks.

Neural Networks are very powerful and used some complex processes in training phrase to Learn from the data like a Human. There is a technique in ANN’s which called backpropagation it Minimizes the prediction error with fine tuning model parameters. This process is none interpretable, and this is the reason why I’m considering ANN’s are good examples for “black boxes”.

3.

Agglomerative is a hierarchical clustering technique which starts with considering all data points are considering as clusters and end up with whole dataset as a single cluster.

i)

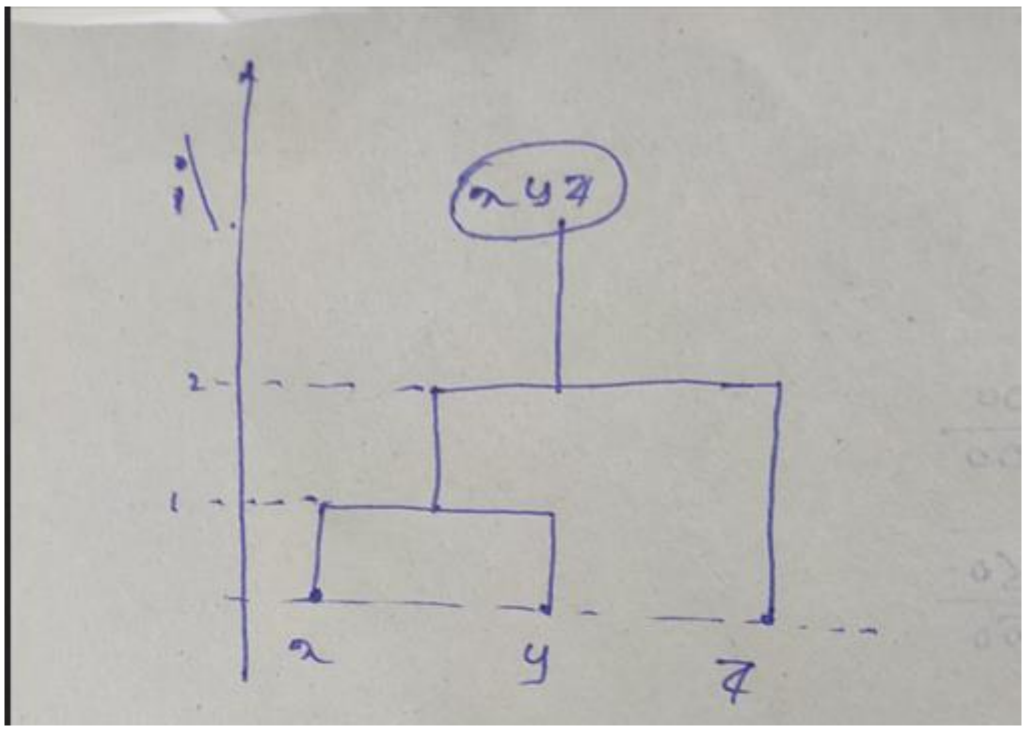
Distance Metrix : single linkage (closest points)

	X	Y	Z
X	0	1	5
Y		0	2
Z			0

Merged XY because those clusters are the closest

Calculating distance Metrix

	XY	Z
XY	0	2
Z	2	0



ii).

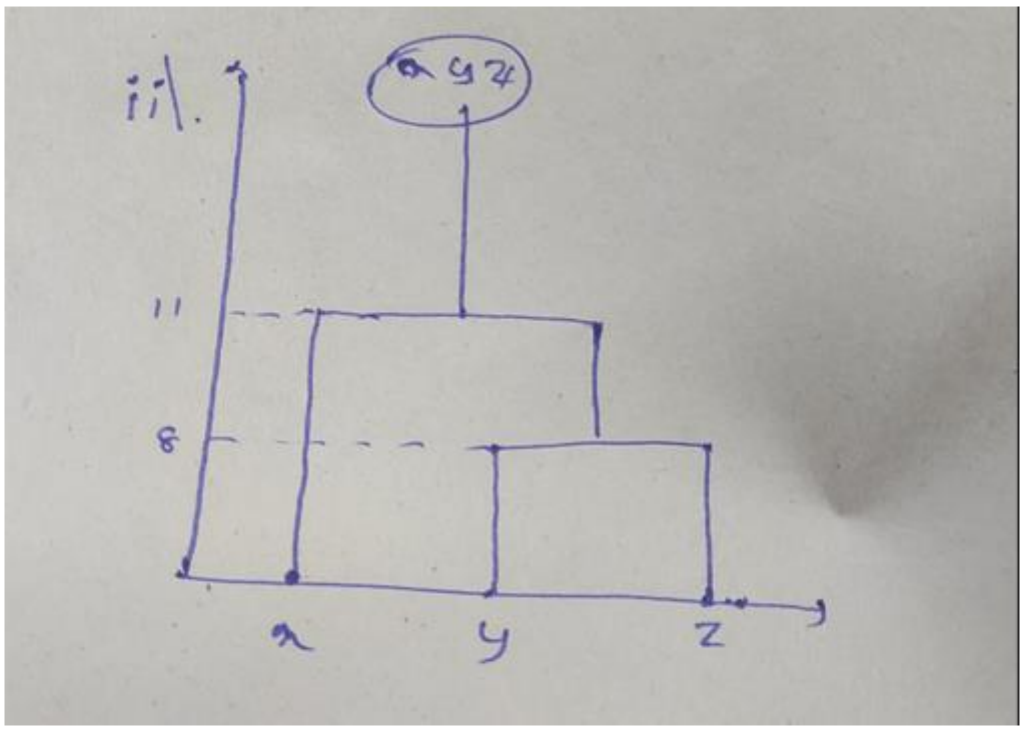
Distance Metrix: Complete linkage (Longest points)

	X	Y	Z
X	0	9	11
Y		0	8
Z			0

Merged XY because those clusters are the closest

Calculating distance Metrix

	X	YZ
X	0	11
YZ	11	0



4.

Overall Accuracy = (TP+TN) / (TP+TN+FP+FN) = (6+9+10) / (6+9+10+3+1+2+1) = 25/32 = 78.125%

actual		Predicted	
		Apple	Not
	Apple	TP = 6	FN = 2
	Not	FP = 4	TN = 20

actual		Predicted	
		Pears	Not
	Pears	TP = 9	FN = 4
	Not	FP = 0	TN = 19

actual		Predicted	
		Grapes	Not
	Grapes	TP = 10	FN = 1
	Not	FP = 3	TN = 18

Sensitivity = TP /(TP+FN)

Apple Sensitivity = 6 / 8 = 0.75

Pears Sensitivity = 9 / 13 = 0.692

Grapes Sensitivity = 10 / 11 = 0.909

Specificity = TN / (TN + FP)

Apple Specificity = 20 / 24 = 0.8333

Pears Specificity = 19 /19 = 1

Grapes Specificity = 18 / 21 = 0.857

5.

Overfitting is one of the major problems in Machine Learning World which means Machine Learning Algorithm Going do a Perfect job with Training Data and fit with perfectly but when in Testing phrase it will perform with poor predicting accuracy.

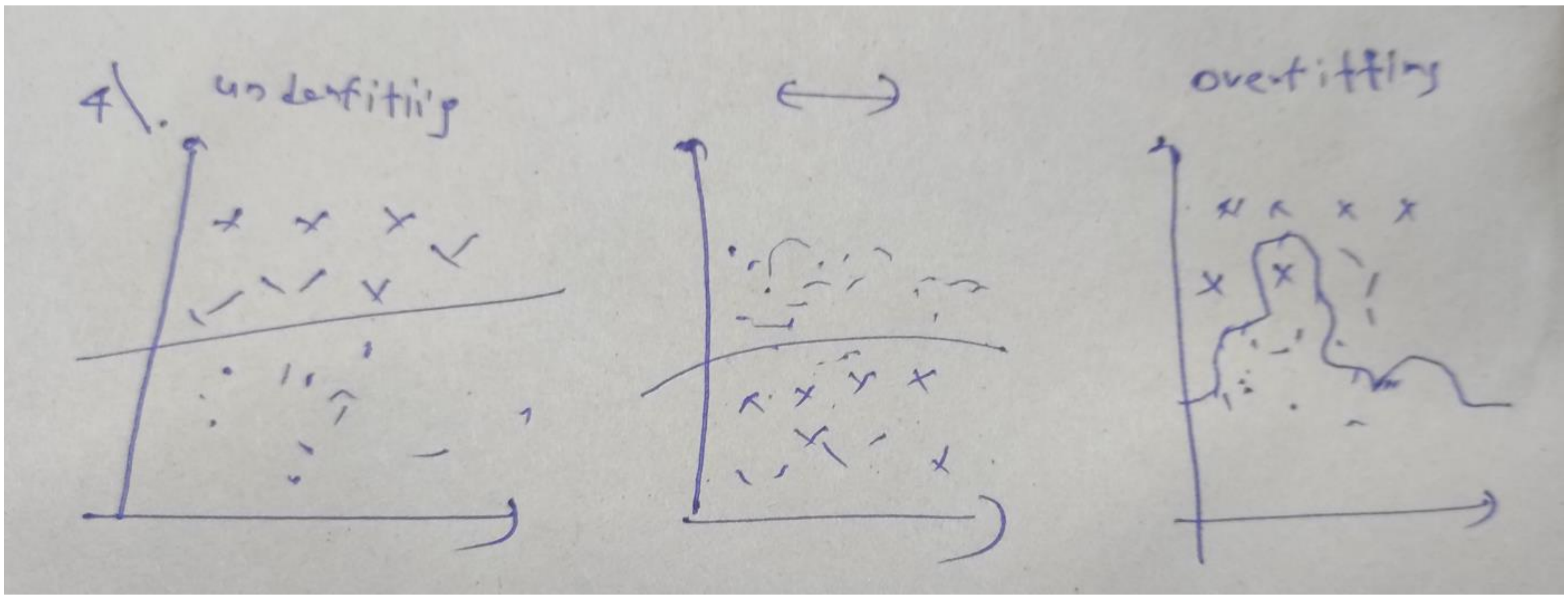


Figure 1 underfitting and overfitting

For avoid this overfitting problem these techniques can be used.

Early stopping

Dropping

Regularization

6.

7.

Cat Class

P(Cat) = 500/1500

P (Swim=True | Cat) = 450/500

P (Wings = False | Cat) = 500/500

P (Green = True | Cat) = 0/500

P (Sharp Teeth = False | Cat) = 0 /500

P (Cat | [Swim=True, Wings=False, Green=True, Sharp Teeth=False]) = P ([Swim=True, Wings=False, Green=True, Sharp Teeth=False] | Cat) * P (Cat)

P (Cat | [Swim=True, Wings=False, Green=True, Sharp Teeth=False]) = 450/500 * 0 * 1 * 0 * 500/1500 = 0

Parrot Class

P(Parrot) = 500/1500

P (Swim=True | Parrot) = 50/500

P (Wings = False | Parrot) = 0/500

P (Green = True | Parrot) = 400/500

P (Sharp Teeth = False | Parrot) = 500/500

P (Parrot | [Swim=True, Wings=False, Green=True, Sharp Teeth=False]) = P ([Swim=True, Wings=False, Green=True, Sharp Teeth=False] | Parrot) * P (Parrot)

P (Parrot | [Swim=True, Wings=False, Green=True, Sharp Teeth=False]) = 50/500 * 0 * 400/500 * 1 * 500/1500= 0

Turtle Class

P(Turtle) = 500/1500

P (Swim=True | Turtle) = 500/500

P (Wings = False | Turtle) = 500/500

P (Green = True | Turtle) = 100/500

P (Sharp Teeth = False | Turtle) = 450/500

P (Turtle| [Swim=True, Wings=False, Green=True, Sharp Teeth=False]) = P ([Swim=True, Wings=False, Green=True, Sharp Teeth=False] | Turtle) * P (Turtle)

P (Turtle| [Swim=True, Wings=False, Green=True, Sharp Teeth=False]) = 1*1*100/500*450/500*500*1500 > 0

So highest probability is in Turtle class because

Input ([Swim=True, Wings=False, Green=True, Sharp Teeth=False]) Is classified as "Turtle"

Question B-1

Toral Parent Entropy = S

S = - (4/7) log_2 (4/7) – (3/7) log_2 (3/7)

S = 0.4613 + 0.5234

S = 0.9847

Starter Class

	yes	no	Si
S1 - salad	xx	xxx	[2/5, 3/5]
S2 - soup	xx		[2/2, 0]

S1= -(2/5) log_2 (2/5) – (3/5) log_2 (3/5)

S1= 0.5288 + 0.4422

S1= 0.971

S2 = - (2/2) log_2 (2/2) – 0 log_2 (0)

S2= 0

P (S | Starter) = (5/7) * 0.971 = 0.6936

Main Course Class

	yes	no	Si
S1 - steak	x		[1, 0]
S2 - salmon	xx		[2/2, 0]
S3 - variety-roast	x	xx	[1/3, 2/3]
S4 - surprise-bake		x	[0, 1]

S1 = - log_2(1) = 0

S2 = -(2/2) log_2(2/2) = 0

S3 = -(1/3) log_2 (1/3) – (2/3) log_2 (2/3)

S3 = 0.5283 + 0.3899

S3 = 0.9182

S4 = 0

P (S | Main Course) = 3/7 * 0.9182 = 0.3935

Gain: Starter = 0.9847 – 0.6936

Gain: Main Course = 0.9847-0.3935

Information Gain of Main Course is Higher and Main Course is root node.

Question B-2

For k =2

Cluster 1 centroid = ((2+2+3)/3, (1+3+2)/3)

Cluster 2 centroid = ((5+5+4)/3, (1+2+3)/3)

Centroid for all points = ((2+2+3+5+5+4)/6, (1+3+2+1+2+3)/6)

WSS each data point distance with cluster centroid

BSS each datapoint with all cluster centroid