



Solution –

So the basic logic behind trees is that we divide our domain into sub-domains and we keep dividing it so that the final parts have majorly objects of one single class (squares and circles in this case). This way, while prediction, we can simply see in which sub-domain a particular object lies and tell what its class will be.

But there is a problem to this logic. The issue is that we will keep dividing the domain till there is only one object left in a sub-domain. Now this can be problematic as such domains are not very useful while prediction.

So, we create few rules to know where to stop (it’s important to know that where one should stop).

Here, we use two such rules. The first rule is, size threshold and the second rule is purity threshold.

The size threshold rule means that if a subdomain has let’s say only 5 objects, it should not be divided any further. This would ensure that we do not keep dividing till there is only one object in every sub-domain.

The purity threshold rule means that once we reach a certain level of purity, the model is good enough for us and we should stop.

Well purity is the percentage of a sub-domain is the number of right objects that are in it. Basically put, it is majority objects divided by total number of objects. For example, if a subdomain had 98 squares and 2 circles, we will say the purity is 98%.

The first part of this question is to create a decision tree. And indicate purity and size at leaf nodes. Leaf nodes are the final subdomains. Or as in a tree, leaves are the last part of a branch. Nothing branches out from a leaf.

So to create a decision tree, what according to you would have been the first split suggested by computer here? Look for the biggest split. The trick is to look for a split that can divide the entire domain into two sub-domain. Like here, the line x1 = 8 does not divide the entire domain. However, the line x2 = 4 does. So our first split happens on the basis of whether an object has x2 < 4 or x2 > 4. So our tree would have starting like this –

X2 <= 4

Now lets look at the first subdomain (X2 <= 4), in this subdomain, a split happens if X1 <= 8. So branch it out like this –

X2 <= 4

/ \

/ \

X1 <= 8

Now, there are not any further splits after this split in this subdomain. So lets look at the other subdomain, that is X2 > 4

Here, there can be two splits that happened first. You can choose any. You can choose X1 <=4 or X1<=9 as the prior split. I am choosing X1 <= 4, So now, the tree would look like,

X2 <= 4

/ \

/ \

X1 <= 8 X1 <= 4

Now, in the X1 > 4 subdomain, there is a further split at X1 = 9. So we will show this split in the tree as well. Remember, it is on the right side (greater than side).

X2 <= 4

(yes) / \ (no)

/ \

X1 <= 8 X1 <= 4

(yes) / \ (no) (yes) / \ (no)

Square Circle / \

Square X1 >= 9

(yes) / \ (no)

/ \

Square Circle

So, we see there are 5 final subdomains (leaf-nodes). Now we must write purity and size at leaf nodes as asked by the question. Size would be easy to calculate. Just count the number of objects. For purity, divide number of right objects with total objects. So, the final tree would look like this –

X2 <= 4

(yes) / \ (no)

/ \

X1 <= 8 X1 <= 4

(yes) / \ (no) (yes) / \ (no)

Square Circle / \

S: 22, P: 0.95 S:6, P:0.83 Square X1 >= 9

S: 17, P:0.88 (yes) / \ (no)

/ \

Square Circle

S: 12, P:0.91. S: 21, P: 0.95

Now, we need to determine if the tree requires any further splits. It can be done by checking conditions for the subdomains. There are five subdomains as of now. Let’s look from the left side. The leftmost subdomain in the tree is of square type with size = 22 and purity = 0.95. Now, the size threshold has not been satisfied (which is 5 in this question). But the purity threshold has been satisfied (0.90) in this question. Now even if one of these two conditions are satisfied, we do not further split a subdomain. So this subdomain does not require split. Second subdomain from left is a circle with Size equals to 6 and purity = 0.83. Now here, both the size and purity threshold are not satisfied. It means that we can split this subdomain. Similary, the square subdomain that has size = 17 and purity = 0.88 can also use one more split.

So, two subdomains require splits. Now there are three ways to make splits. The first one is based on error rate, the second on entropy and third on gini index. All three are different metrices to assess strength of a split. As question does not mention which split to use, I suggest to use the easiest one, that is the error rate one.

(In this question, we do not actually need to do that because splits are obvious. If they were not obvious, then we will have to calculate error rate and justify our choices.) But there, it is obvious that we can simply split the circle subdomain by a line at X1 = 10. This way, both newly created subdomains will have 5 or less than 5 objects so no subdivision will be required.

Now, the square subdomain can be spitted by a line at X1 = 3. This way, one of the new subdomain (right one), will have objects less than 5 so its size threshold will get satisfied. The left subdomain will now have 14 objects with 13 of them from dominant class. So the purity will be 0.92. So this subdomain’s purity threshold will satisfy. (in the exam, splits will be more or less obvious to human eye. If not, try 2-3 combinations and you will surely find the right split in at best 3rd attempt).

So after introducing these splits, new graph would look something like ->

X2 <= 4

(yes) / \ (no)

/ \

X1 <= 8 X1 <= 4

(yes) / \ (no) (yes) / \ (no)

Square X1 > 10 / \

S: 22, P: 0.95 / \ X1 <= 3 X1 >= 9

/ \ / \ (yes) / \ (no)

Square Circle | | / \

S:1, P:1 S:5, P:1 | | Square Circle

/ \ S: 12, P:0.91. S: 21, P: 0.95

Square. Square

S1: 14 P1: 0.91. S1:3, P1: 0.66