Finding Missing Values Using Random Forest

Let's say we have the following dataset →

feature | target

Chest pain	Good Blood circulation	Blocked Arteries	Weight	Heart Disease
NO	NO	NO	125	NO
Yes	Yes	Yes	180	Yes
Yes	Yes	NO	210	NO
Yes	NO			NO

Missing value.

Dataset can be consist of missing values. Before training our model using this traing dataset, we need to fill up these missing values.

We do that by →

- making an (bad) initial guess for this missing values
- Gradually refining these bad guesses until it's a good guess.

1) Finding Initial guess:

Chest Pain	Good Blood circulation	Block Arteris	Weight	Heart Disease)
		NO	125	NO
		Yes	180	Yes
		NO	210	NO
		??	??	NO

4th sample has missing values

We will find intial guess for 'block Arteris'

target is NO

The 4th sample has target values 'NO' → Do not have any heart diseases. So the most ~~com~~ frequent/ common value found in 'Block Arteries' column that has target 'NO' should be the initial guess for Block Artenis.

Chest pain	Good Blood circulation	Block Arteris	Weight	Heart Disease)
		NO	125	NO
		Yes	180	Yes
		NO	210	NO
		??	??	NO

These two sample has 'NO'

The samples with 'NO' target has 2 'NO' in the

"Block Arteries" feature and '0 'Yes'. So, NO is the most common for "Heart Disease = NO" target.

So our initial guess for Block Arteries' 4th sample will be 'NO'.

Chest pain	Good Blood circulation	Block Arteries	Weight	Heart Diseases
		NO	125	NO
		Yes	180	Yes
		NO	210	NO
		NO	??	NO

Initial Guess ↓ — Numerical feature. ↓

As weight is the neumerical feature, the initial guess will be the median value of weight feature, sample is that has Heart disease = NO.

Sample (1) and (3) has heart diseases = NO.

So Median of this two data = $\frac{125 + 210}{2} = 167.5$.

[You need to know how to find Median of odd and even number of data]

2) Refining the initial Guess —

~~Iteration~~

we will refine the values until the ~~values~~ guesses become good. This will take multiple iteration. we will see the first iteration; all iterations will be same.

Iteration 1°:

	Chest pain	Good Blood circulation	Blocked Arteries	Weight	Heart Disease
(1)	No	NO	NO	125	NO
(2)	Yes	Yes	Yes	180	Yes
(3)	Yes	Yes	NO	210	NO
(4)	Yes	Yes	NO	167.5	NO

initial Guesses.

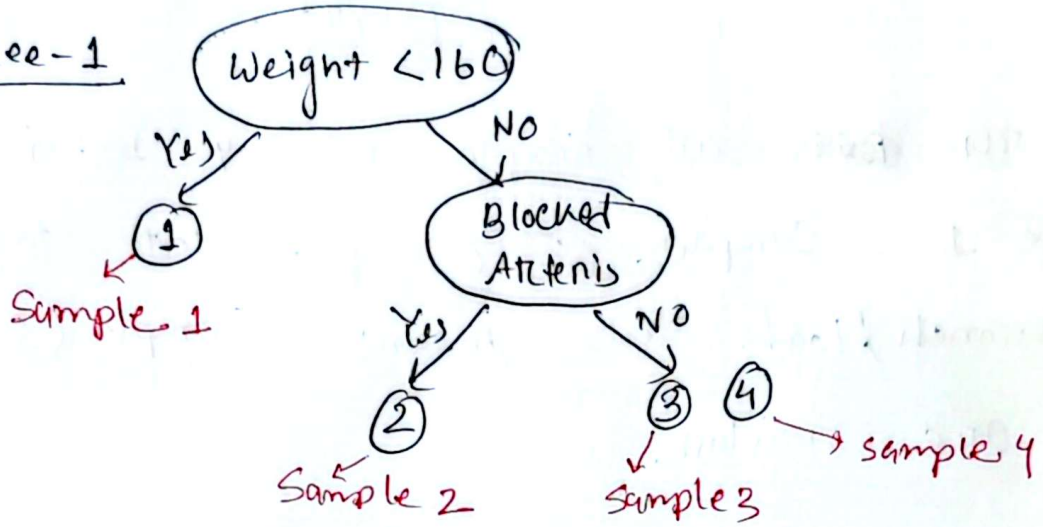
Now we need to find which samples (among 1, 2, 3) are similar to sample (4) [the one with missing data].

Steps of finding Similarity:

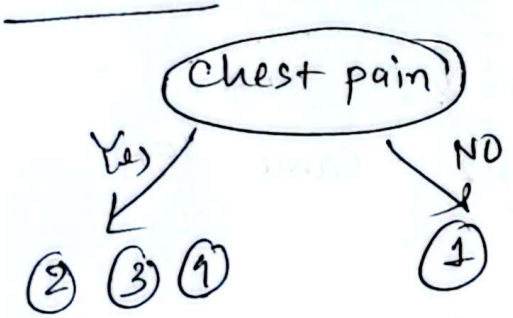
(1) Build a random forest:

we need to build a random forest consisting of multiple trees. Let's say we have total 10 trees in our random forest. I will draw only tree of them here for ~~your~~ example.

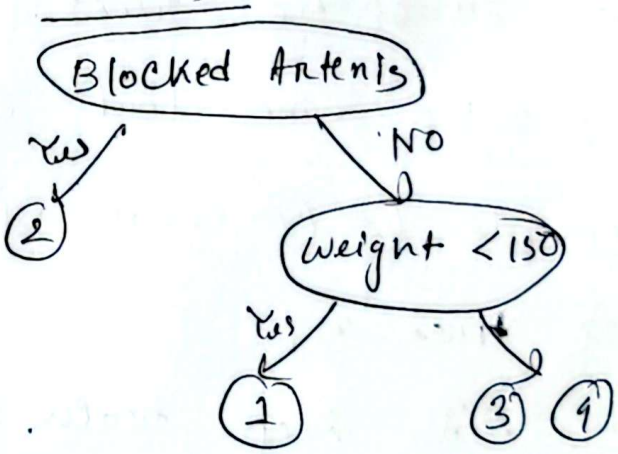
Tree-1



Tree-2



Tree -3

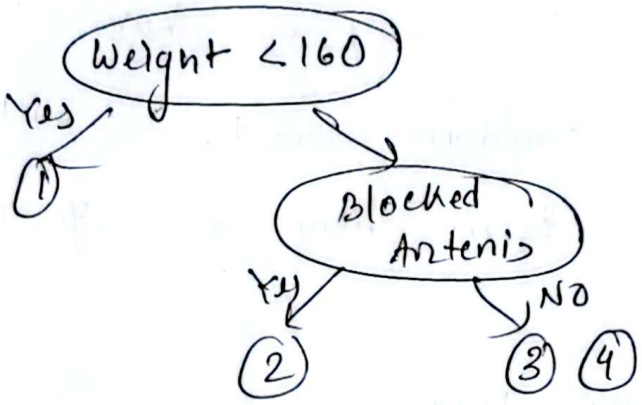


we will have total 10 trees like this for our example.

Step-2 : Run all data down each of the tree.

We have already seen that in Step-1 trees.

For example in tree-1 :



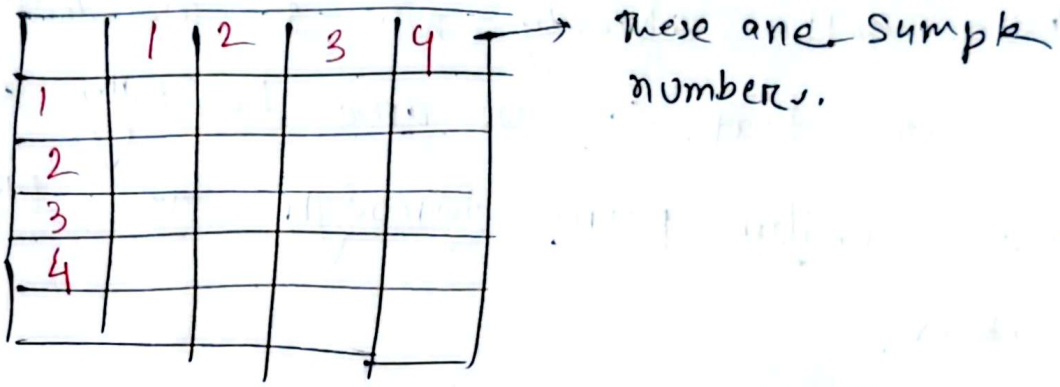
If we run all sample 1, 2, 3, 4 down the tree-1, sample (3) (4) goes into the same branch / leaf. This means sample (3) and (4) are similar.

similarly we will run the samples down all of the trees. Sample (2), (3), (4) ends up in the same leaf of tree-2 and sample (3) (4) ends up in the same leaf of tree-3.

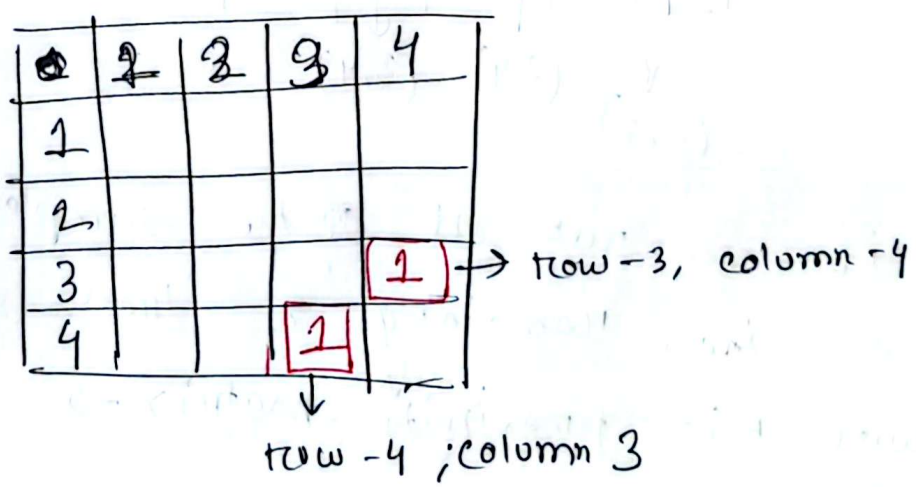
We will keep track of this similarity using a proximity matrix.

Step-3 : Building Proximity Matrix.

As there are 4 samples in our data set, there will be 4 rows and columns in the proximity matrix - which looks like —

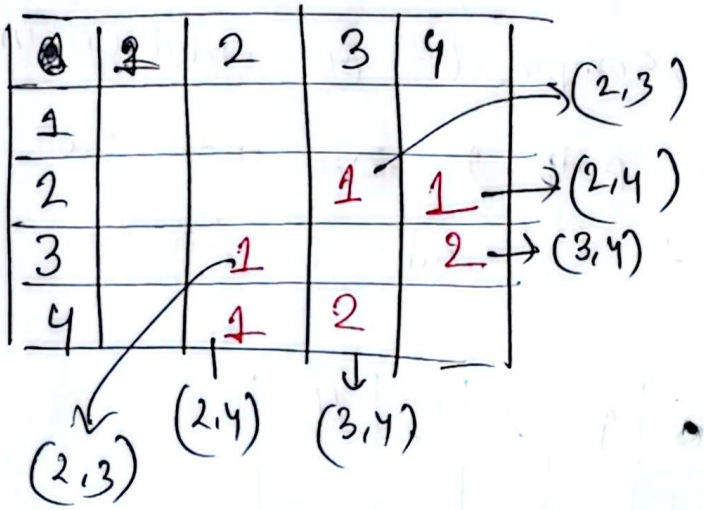


In tree - 1, sample (3) (4) ended up in same leaf so we put 1 ~~that~~ into the cells that corresponds to 3 and 4 in the matrix



If there were other pairs of samples ended in the same leaf node, we will keep track of that as well.

After running samples into the second tree, we saw sample ② ③ ④ ends up in the same leaf. So we will put 1 for each pair of (2,3), (3,4) and (2,4). As (3,4) pair cells already has 1, we will add 1 with that. So we are basically counting the similar pairs through the proximity matrix.



Now if we run all the samples down all of the remaining 8 trees we end up with this proximity matrix →

	1	2	3	4
1		2	1	1
2	2		1	1
3	1	1		8
4	1	1	8	

NO we divide each proximity value (each cells) with total number of trees (which is 10 here) and end up with →

	1	2	3	4	
1		0.2	0.1	0.1	→ 1/10
2	0.2		0.1	0.1	
3	0.1	0.1		0.8	→ 8/10
4	0.1	0.1	0.8		

NO we will use this proximity values to refine our initial guesses.

Step-4: updating the missing data

① updating categorical Features

Now we need to find the weight freaveney for each category (Here Yes and NO are the categories of Blocked arteries) and check which one is higher / highest.

Weighted Freavency of a 'category' = Freaveney of 'category' × weight of 'category'.

we have to find weighted frequency of Yes and weighted frequency of NO.

Weighted frequency of 'Yes'

∴ weighted frequency of 'Yes' = Frequency of 'Yes'. × weight of 'Yes'

Frequency of 'Yes' → From dataset

weight of 'Yes' → from proximity matrix

Frequency of Yes –

Chest pain	Good Blood circulation	Blocked Arteries	weight	heart Diseases
		NO		
		Yes		
		NO		
		???		

There is 2 NO and 1 Yes

∴ Frequency of 'Yes' = $\frac{1}{3}$ → 1 Yes
→ 3 samples without missing data

Similarly

Frequency of 'NO' = $\frac{2}{3}$

[Weight of Yes]

$$\text{Weight of 'category'} = \frac{\text{Proximity of 'category'}}{\text{All proximities}}$$

$$\therefore \text{Weight of Yes} = \frac{\text{Proximity of Yes}}{\text{All proximities}}$$

Here,

	1	2	3	4
1		0.2	0.1	0.1
2	0.2		0.1	0.1
3	0.1	0.1		0.8
4	0.1	0.1	0.8	

→ All proximities

Proximity of Yes → Proximity value of the samples with only Yes (in Blocked arteries). Here only sample-2 has Yes in Blocked arteries. We will take the proximity from the row of sample

(4) as sample 4 has the missing value.

∴ Proximity of Yes = 0.1

All proximities → All of the proximities of the sample that has missing value. the sample (4) has missing value here.

∴ All proximities = 0.1 + 0.1 + 0.8

∴ The weight for Yes = $\frac{0.1}{0.1+0.1+0.1} = 0.1$

The weight for NO →

the weight of NO = $\frac{\text{proximity of NO}}{\text{All proximities}}$

	1	2	3	4
1		0.2	0.1	0.1
2	0.2		0.1	0.1
3	0.1	0.1		0.8
4	0.1	0.1	0.8	

→ All proximities

Proximity of NO = 0.1 + 0.8

Sample (1) and (3) has NO in Blocked antenies column.

∴ Weight of NO = $\frac{0.1+0.8}{0.1+0.1+0.8} = 0.9$

∴ weighted frequency of Yes

= Frequency of Yes × Weight of Yes

= $\frac{1}{3} \times 0.1 = 0.03$

∴ weighted frequency of NO

= Frequency of NO × Weight of NO

= $\frac{2}{3} \times 0.9 = 0.6$ → large

So, our now revised guess for Blocked Arteries will be NO

Chest pain	Good Blood circulation	Blocked Arteries	Weight	Heart disease
		NO	125	
		Yes	180	
		NO	210	
		NO	??	

↓ revised

No we need to revise the guess of weight (numerical feature)

(2) Updating Numerical Feature:

To find revised weight for sample (4), we need to calculate Weighted Average.

Weighted Average = (Sample (1)'s value (from dataset) x sample (1)'s weighted average (from proximity matrix)) + (sample (2)'s value x sample (2)'s weighted average) + (sample (3)'s value x sample (3)'s weighted average)

→ we sum all samples without the missing data.

Sample x's weighted average

$$w = \frac{\text{proximity of sample x}}{\text{Sum of Proximity}}$$

∴ sample 1's weighted average

$$= \frac{\text{Proximity of sample 1}}{\text{sum of proximity}}$$

(From sample (4) row.)

→ of sample (4)

here sample (4) has the missing data, so we will consider sample (4)

∴

	1	2	3	4
1		0.2	0.1	0.1
2	0.2		0.1	0.1
3	0.1	0.1		0.8
4	0.1	0.1	0.8	

→ All proximity

↓ proximity of sample (1)

∴ Sample (1)'s weighted average = $\frac{0.1}{0.1+0.1+0.8}$

$= 0.1$

Similarly "(2)'s " " $= \frac{0.1}{0.1+0.1+0.8} = 0.1$

and "(3)'s " " $= \frac{0.8}{0.1+0.1+0.8} = 0.8$

sample (1)'s value from dataset ↑

∴ weighted Avenge $= (125 \times 0.1) + (180 \times 0.1) + (240 \times 0.8)$

↓ Sample (1)'s weighted avenge.

$= 198.5$

So this is our revised guess for sample (4)'s weight.

So, after first iteration →

Chest Pain	Good Blood circulation	Blocked Arteries	Weight	Heart Disease
		No	125	
		Yes	180	
		No	210	
		NO	198.5	

Now we do this steps over and over again until the missing value converges, when there is no changes happening in the missing values each time we recalculate.

This process is for when the dataset has missing value. However, if the testing data/ new unknown data has missing value we will need to do a few steps extra.

Finding missing value in new Sample'.

chest pain	Good Blood circulation	Block arteries	Weight	Heart Disease
Yes	NO	??	168	We will predict the target

missing data.

We want to classify this new patient, But we have a missing value in Block arteries feature. So we need to make a guess about Block arteries.

Here we already have the Random Forest Model. We are trying to predict for a new data. this data ~~this data~~ So we run down all the trees of the Random Forest.

Step-1 : Create two copies of the data for the target.

copy-1

Chest pain	Good Blood circu	Blocked Arteries	weight	Heart Disease
Yes	NO	??	168	Yes

Copy - 2

chest pain	Good Blood Circu	Blocked Arteries	weight	Heart Diseases
Yes	NO	??	168	NO

Step-2 : Fill the missing values:

Now, we will run the previous method of finding missing value for this two copies of data and fill the missing values.

Let assume, we end up with →

Copy-1

chest pain	Good Blood cine	Blocked Arteries	weight	Heart Diseases
Yes	NO	Yes	168	Yes

↓

After refining

Copy-2

Chest pain	Good Blood cine	Blocked Arteries	weight	Heart Disease
Yes	NO	NO	168	NO

Step-3 : Run the two data down the trees of Random Forest

We will run ~~down~~ these two copy of new data ~~two~~ down the ~~Trees~~ of random forest. and count ~~each~~ how many times each ~~Data~~ copy was correctly Labelled (Yes, NO)

= Lets say copy-1 was correctly Labelled by the trees [3 times]. (Actual target is Yes and trees ~~gave~~ predicted Yes)

= copy-2 was correctly Labelled by the trees [1 time] (Actual target is NO and trees predicted NO)

So, copy-1 wins as it was correctly labelled more time than copy-2.

So, our filled in missing data with prediction will be copy-1 →

Chest pain	Blood circ	Blocked arteries	weight	Heart Diseases
Yes	NO	Yes	168	Yes.