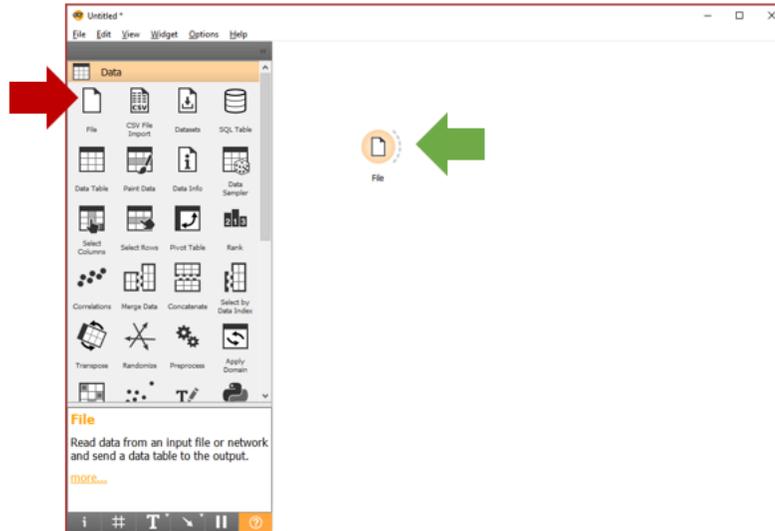




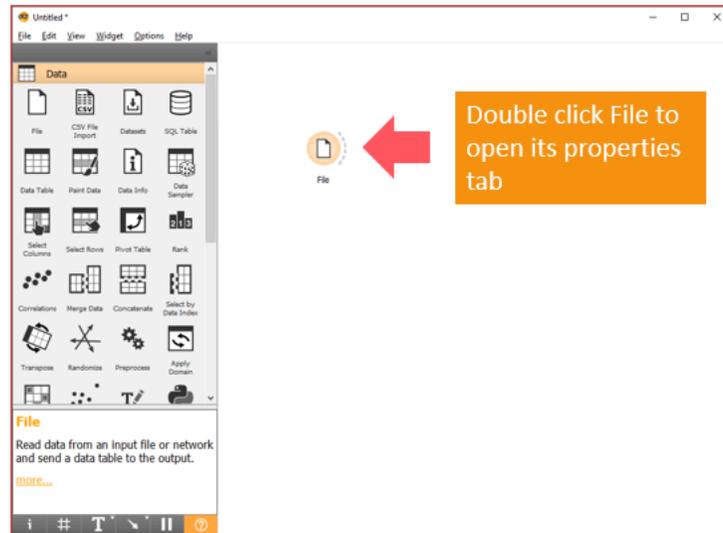
## Step 1(b)

Insert the File widget onto the canvas

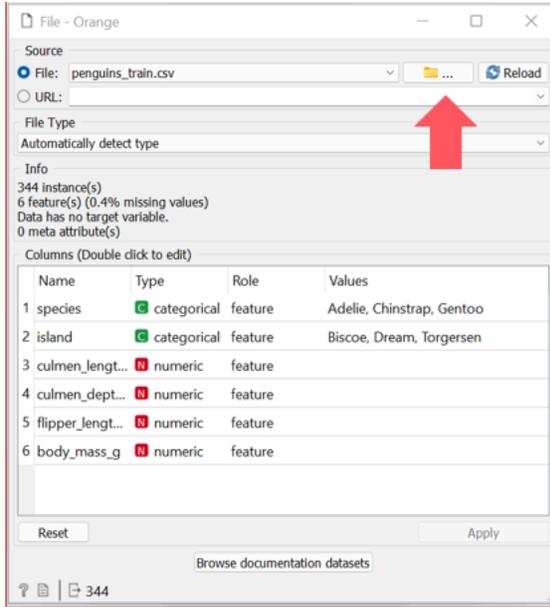


## Step 1(c)

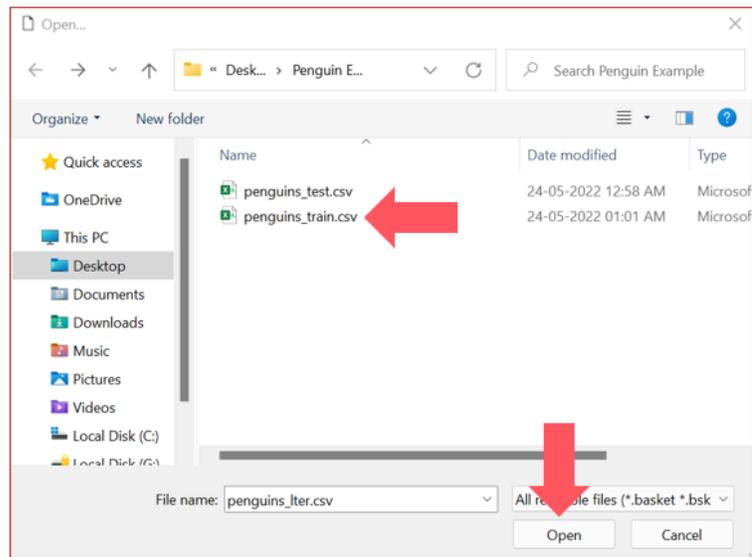
Double click File to open its properties tab



### Step 1(d)



### Step 1(e)



# Step 1(f)

penguins\_train.csv is uploaded



The screenshot shows the 'File' widget in the Orange software. The source is set to 'penguins\_train.csv'. The file type is 'Automatically detect type'. The info section shows 344 instances, 6 features (0.4% missing values), and no target variable or meta attributes. A table lists the columns:

Name	Type	Role	Values
1 species	categorical	feature	Adelie, Chinstrap, Gentoo
2 island	categorical	feature	Biscoe, Dream, Torgersen
3 culmen_lengt...	numeric	feature	
4 culmen_dept...	numeric	feature	
5 flipper_lengt...	numeric	feature	
6 body_mass_g	numeric	feature	

# Step 1(g)

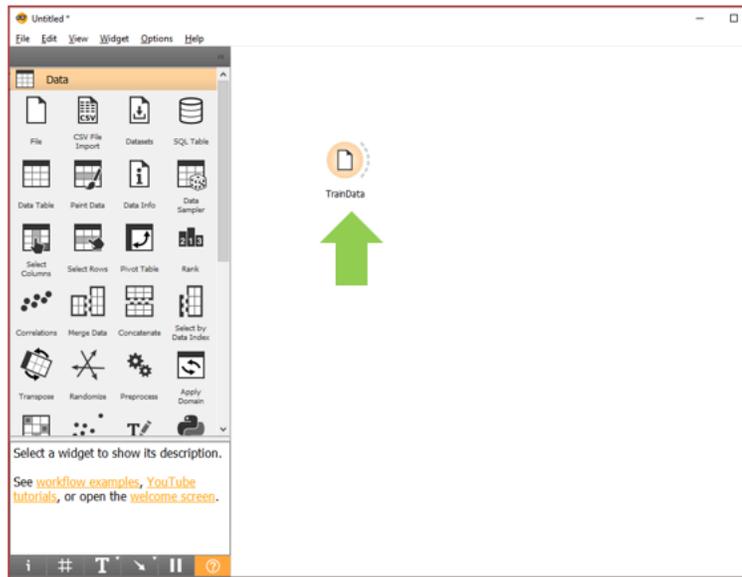
Right-click on the widget File and click on Rename. We will rename it to 'Train Data' so that we do not confuse it with the testing data later

The screenshot shows the 'Data' widget pane in Orange. A context menu is open over the 'File' widget icon. A red arrow points to the 'File' icon, and another red arrow points to the 'Rename' option in the menu.

- Open
- Rename F2
- Remove Backspace
- Duplicate Ctr+D
- Copy Ctr+C
- Help F1

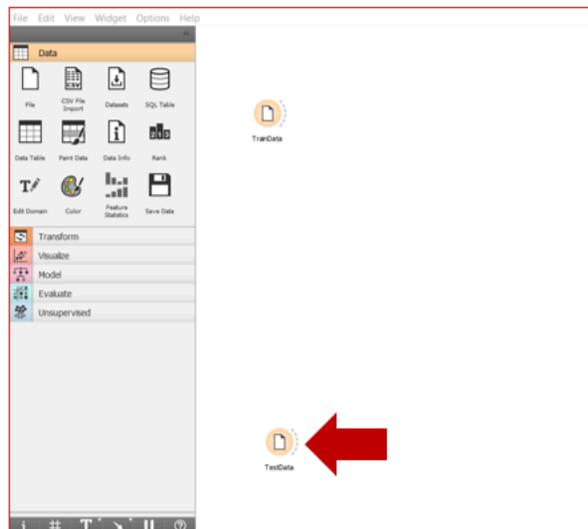
## Step 1(h)

The file name for the uploaded dataset has changed



## Step 1(i)

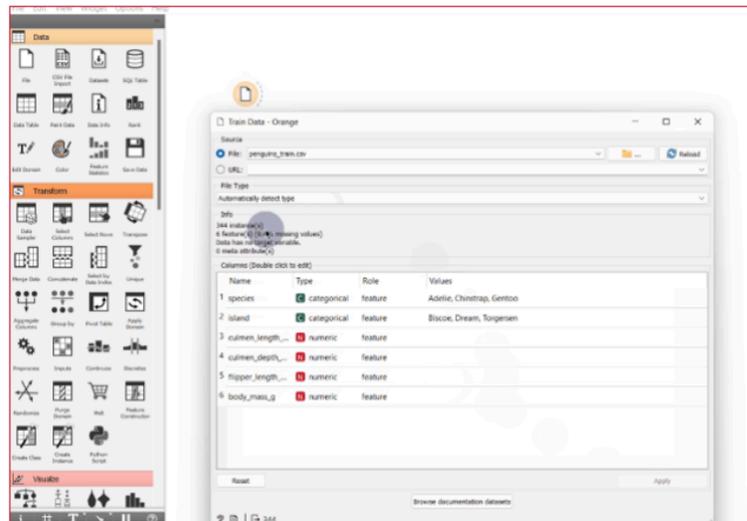
Repeat the same steps for 'Test Data' and upload testing dataset in the file widget



After Data Acquisition, what should we do next?

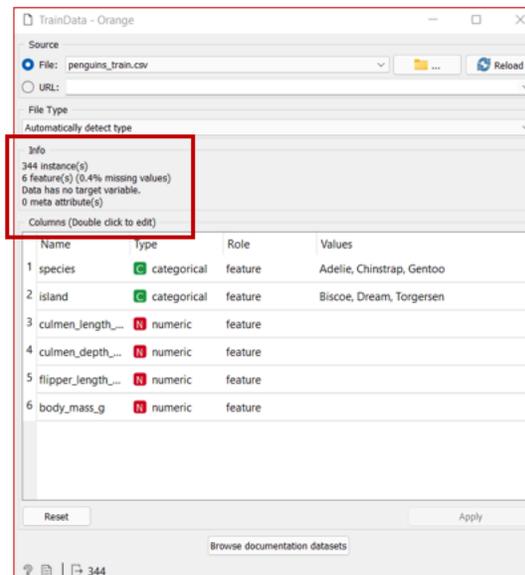
## Step 2: Clean Missing Data

Data  
Exploration



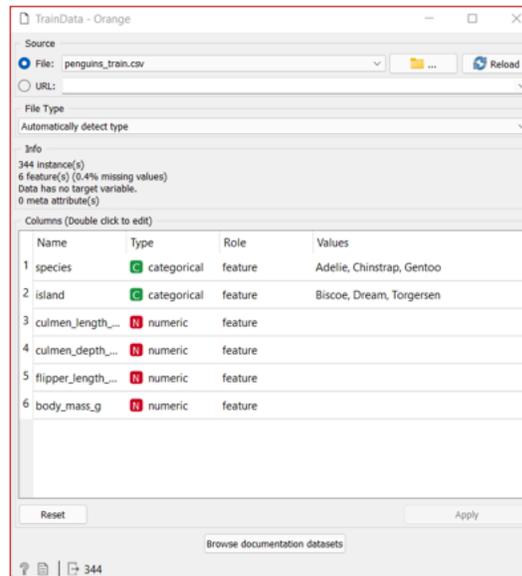
### Step 2(a)

Check if there are any missing values  
Notice that there are some missing values



## Step 2(b)

We will now look at another way to inspect on missing data. Click X to close the pop up.



TrainData - Orange

Source

File: penguins\_train.csv

File Type

Automatically detect type

Info

344 instance(s)  
6 feature(s) (0.4% missing values)  
Data has no target variable.  
0 meta attribute(s)

Columns (Double click to edit)

Name	Type	Role	Values
1 species	categorical	feature	Adelie, Chinstrap, Gentoo
2 island	categorical	feature	Biscoe, Dream, Torgersen
3 culmen_length_...	numeric	feature	
4 culmen_depth_...	numeric	feature	
5 flipper_length_...	numeric	feature	
6 body_mass_g	numeric	feature	

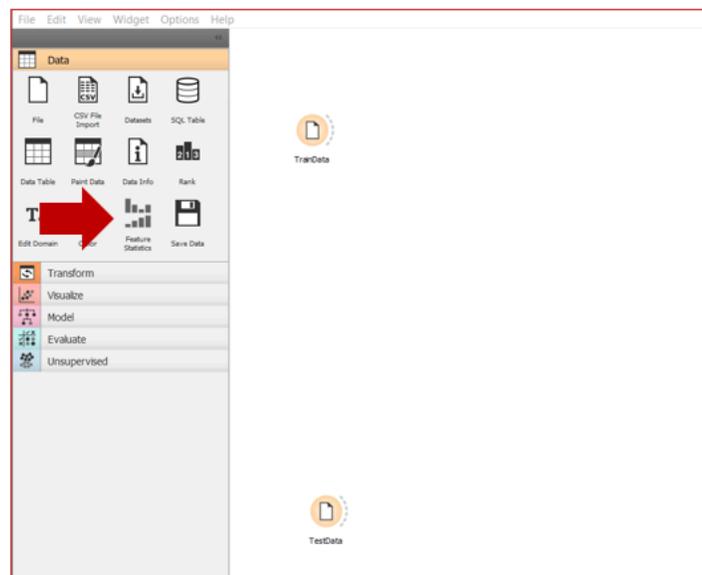
Reset Apply

Browse documentation datasets

344

## Step 2(c)

Insert the widget Feature Statistics onto the canvas



File Edit View Widget Options Help

Data

File CSV File Datasets SQL Table Support

Data Table Print Data Data Info Rank

Edit Domain Feature Statistics Save Data

Transform

Visualize

Model

Evaluate

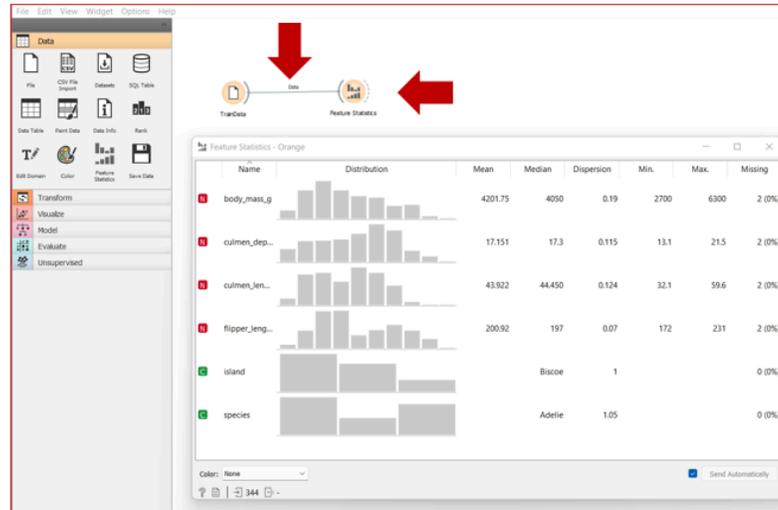
Unsupervised

TrainData

TestData

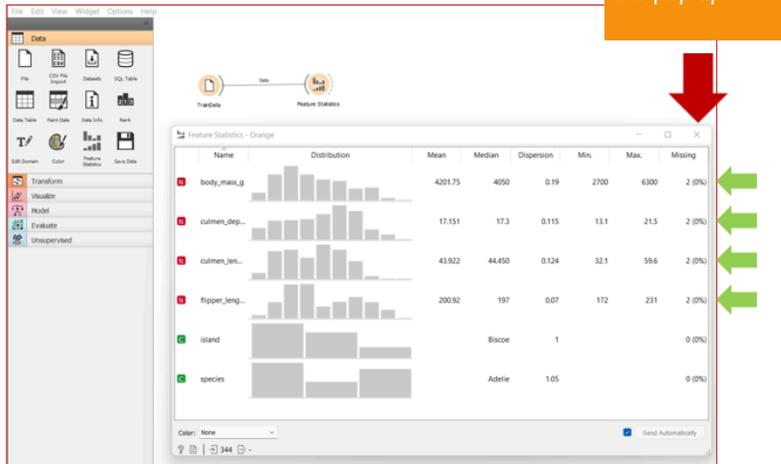
## Step 2(d)

1. Connect widget 'Train Data' to widget Feature Statistics. We can do that by dragging the output from 'Train Data' to the input of Feature Statistics.
2. Double-click on Feature Statistics to see the results.



What is the mean value of culmen\_length feature?

## Step 2(f)



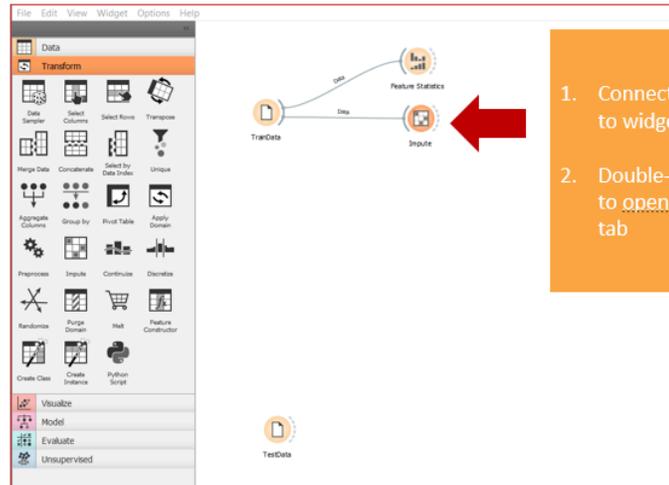
Notice that arrows are pointing to the features with missing values here

## Step 2(g)

Insert the impute widget onto the canvas



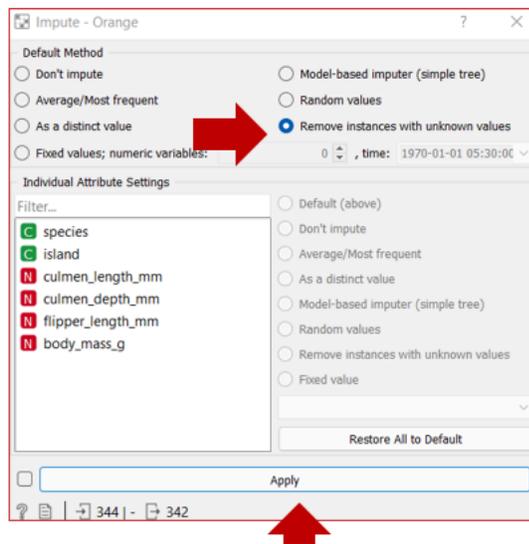
## Step 2(h)



The screenshot shows the Orange3 software interface. On the left is a widget palette with categories like Transform, Visualize, Model, Evaluate, and Unsupervised. The main workspace contains a workflow with three widgets: TrainData, Feature Statistics, and Impute. A red arrow points from the Impute widget to the right, towards a list of instructions.

1. Connect widget TrainData to widget Impute
2. Double-click on Impute to open up the properties tab

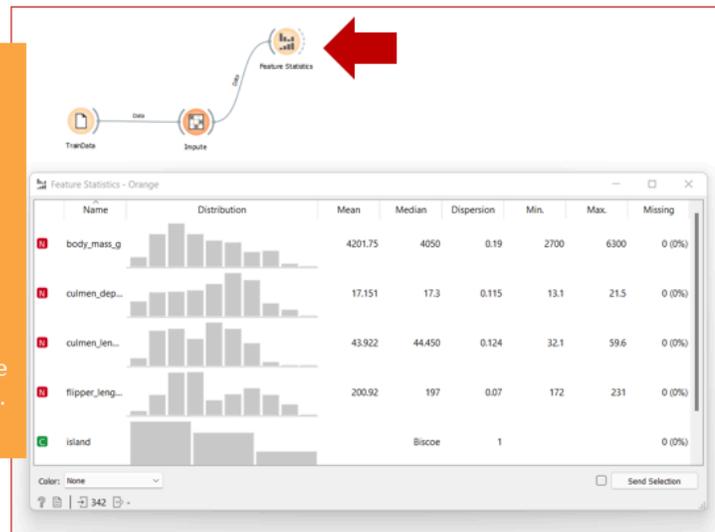
## Step 2(i)



The screenshot shows the 'Impute - Orange' dialog box. Under 'Default Method', the 'Remove instances with unknown values' option is selected. The 'Individual Attribute Settings' section lists variables: species, island, culmen\_length\_mm, culmen\_depth\_mm, flipper\_length\_mm, and body\_mass\_g. The 'Apply' button at the bottom is highlighted with a red arrow.

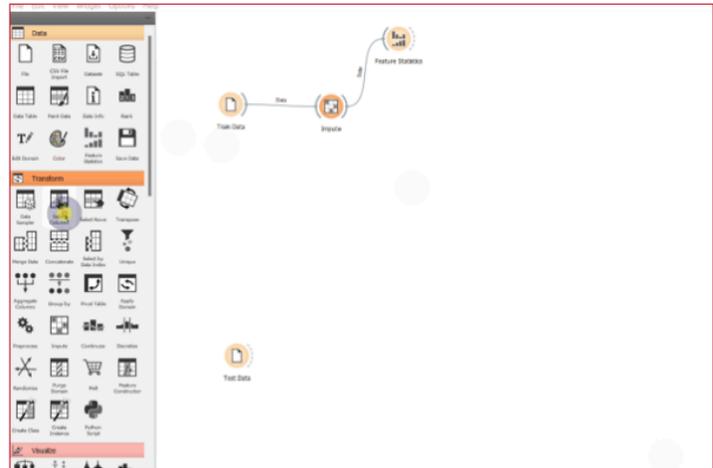
## Step 2(j)

1. Connect the output of Impute to the input of the existing Feature Statistics (the previous connection between TrainData and Feature Statistics has been removed because only accepts one input)
2. Double-click on the Feature Statistics to see the output.



Now that the data is clean and without any missing values, what next?

### Step 3: Select Target Label



#### Step 3(a)

Insert the Select columns widget onto the canvas



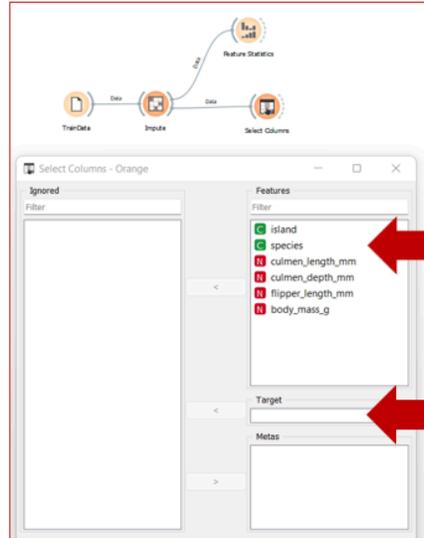
1. Connect with Impute widget
2. Double click on the widget

From TrainData, you would have noticed that the Feature Type for most of the columns is Numeric Feature. In supervised learning models, we have both the features and the labels. The labels are the output. Therefore, we need to define an output for our Palmer Penguin model. We will assign species as our label since that is what we want to identify.

Therefore, we will change the Feature Type for species, from Categorical Feature to Categorical Label. To do that, we will be using Select Columns.

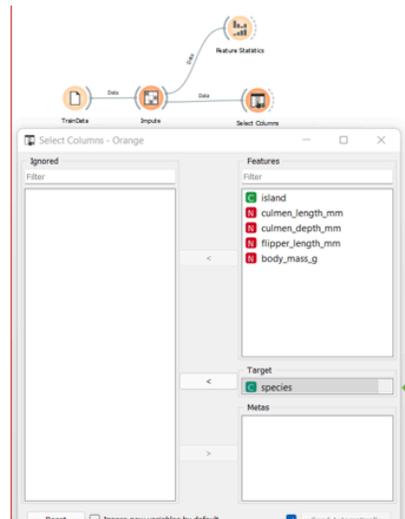
### Step 3(b)

A window displaying all the features will appear



Drag the 'species' feature to the 'Target' box

### Step 3(b)

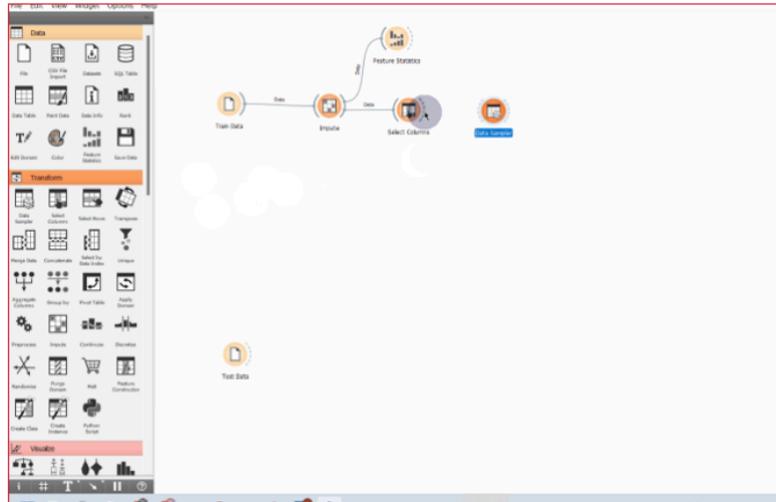


'species' is the Target label

After choosing a target label, we need to split the data

## Step 4: Data Sampler

Data  
Exploration

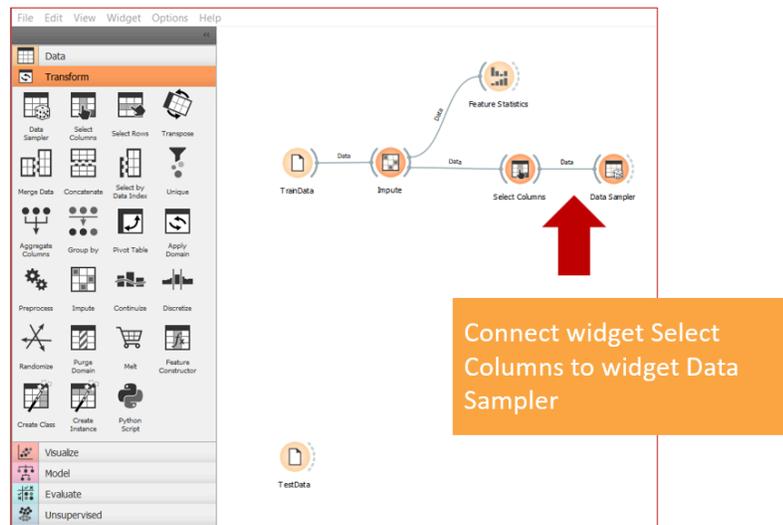


## Step 4(a)

Insert the Data Sampler widget onto the canvas



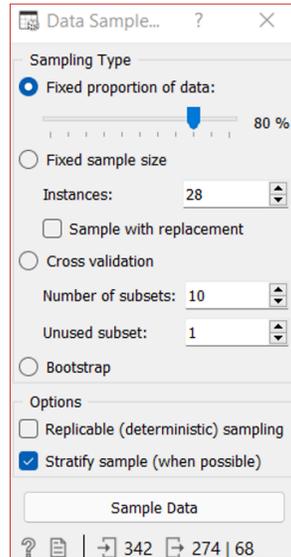
## Step 4(b)



Connect widget Select Columns to widget Data Sampler. We can do that by dragging the output from Select Columns to the input of Data Sampler.

After the connection is made, double-click on Data Sampler to open the properties tab.

### Step 4(c)

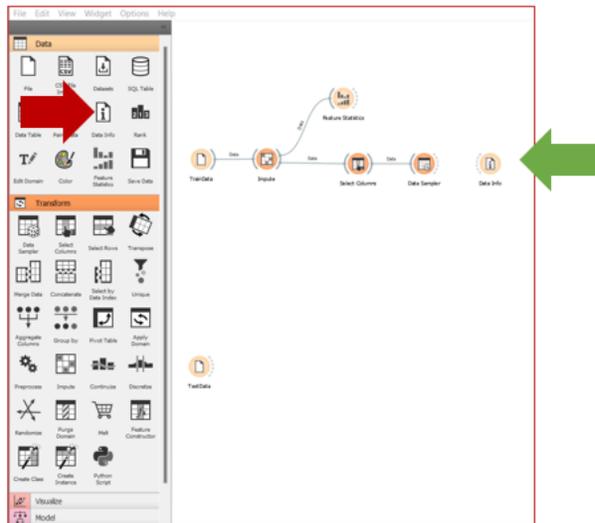


1. Set slider to 80%
2. Click on Sample Data to effect the changes
3. Click X to close the pop-up

How do we know if the data is actually split or not?

### Step 4(d)

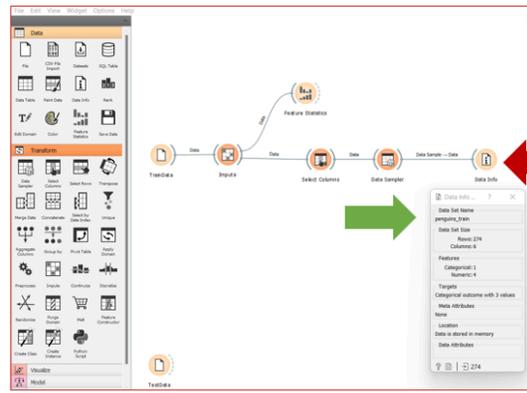
Insert the Data Info widget onto the canvas



Let's inspect on how the data is being split through Data Sampler.

We will be using Data Info.

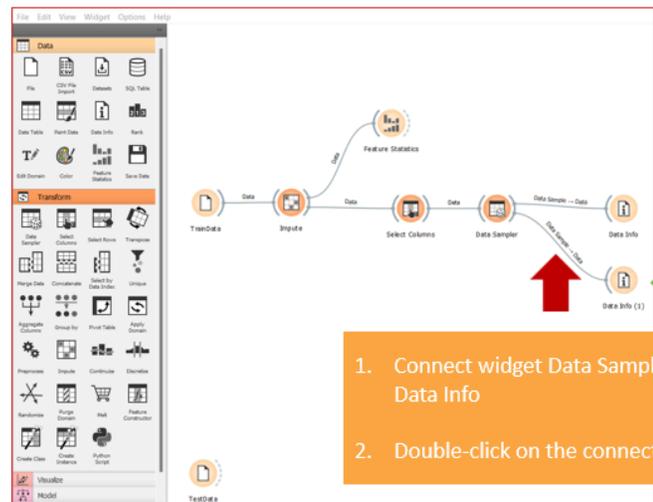
### Step 4(e)



1. Connect widget Data Sampler to widget Data Info

2. Double-click on Data Info to open the properties tab

### Step 4(f)



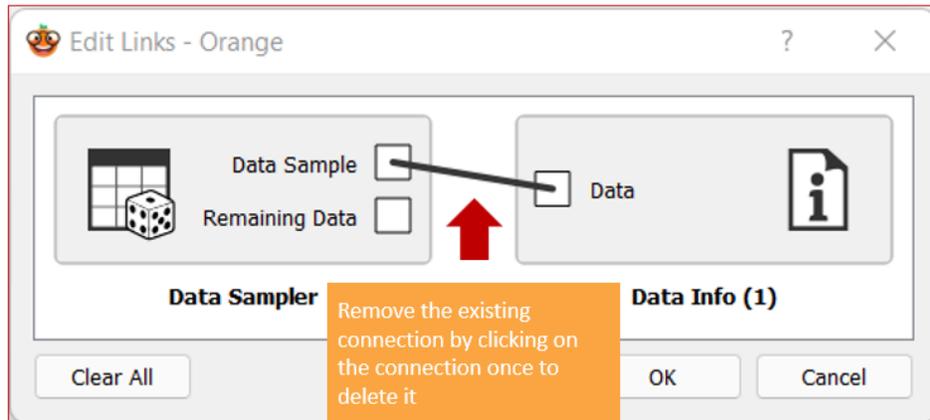
1. Connect widget Data Sampler to widget Data Info

2. Double-click on the connection

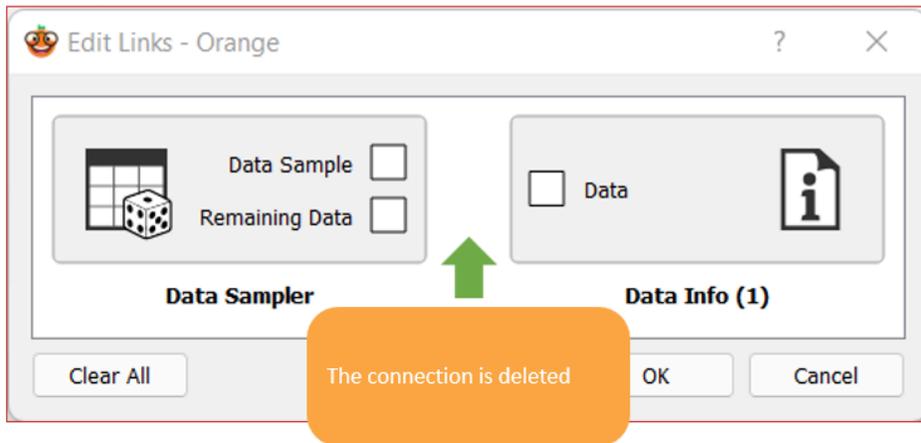
Connect widget Data Sampler to the second widget Data Info. We can do that by dragging the output from Data Sampler to the input of the second Data Info.

Take note of the connection name. We will change this. Double-click on the connection.

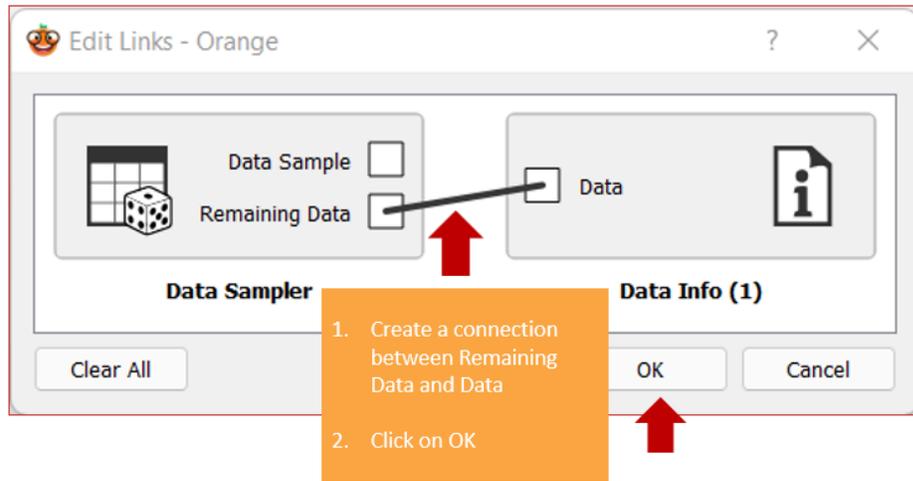
Step 4(g)



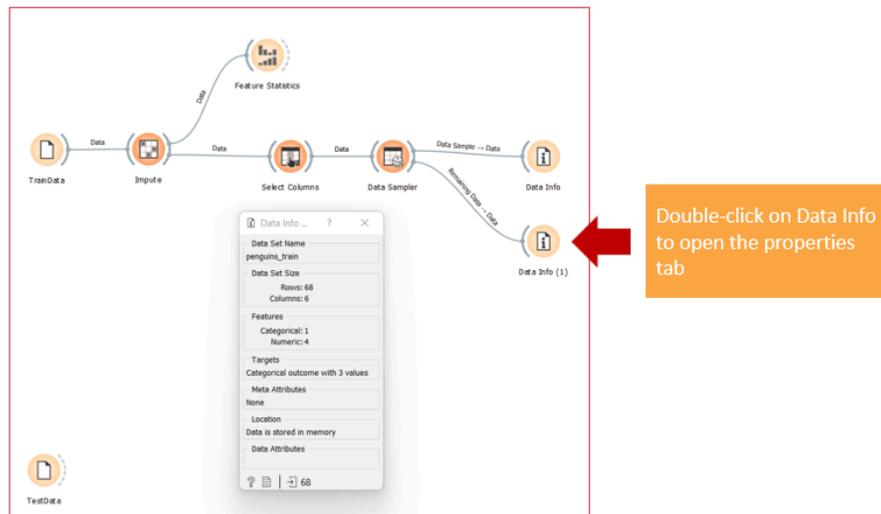
Step 4(g)



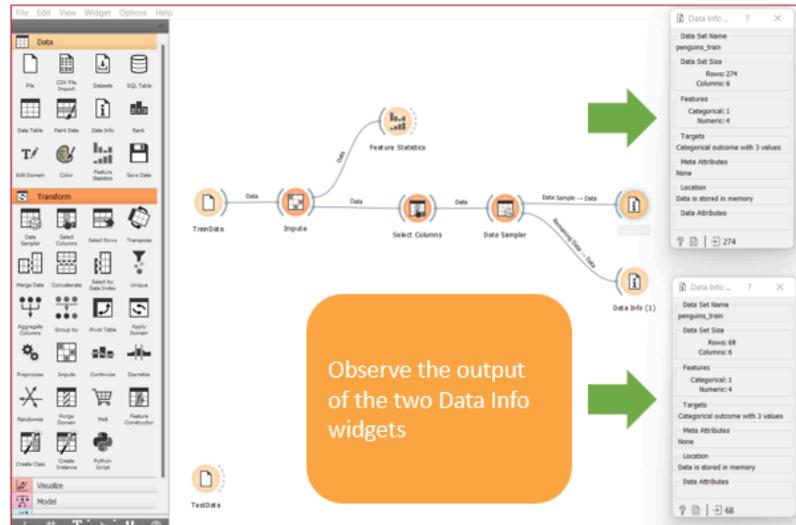
### Step 4(h)



### Step 4(i)



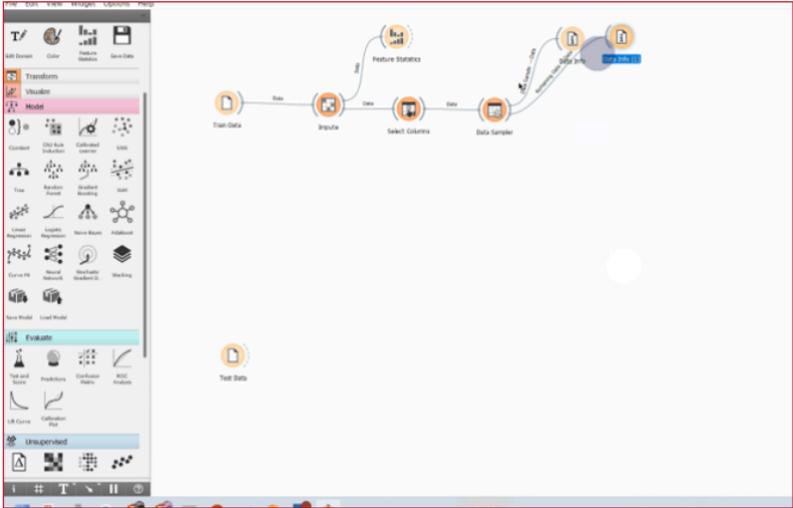
## Step 4(j)



What do we do after having split the data?

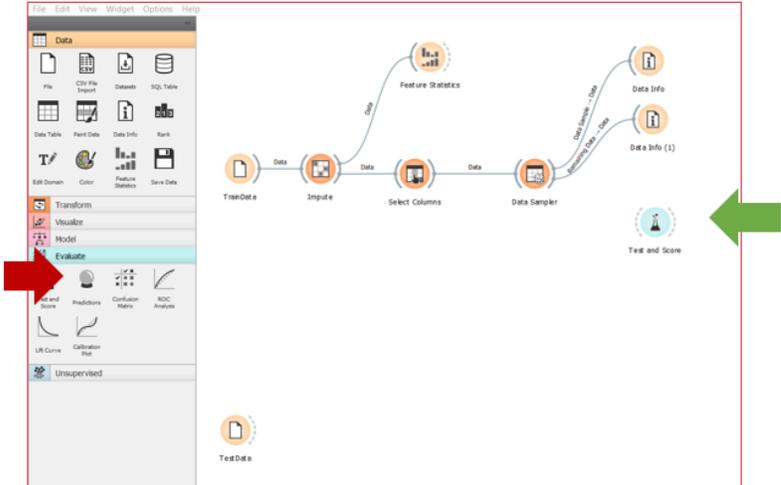
# Step 5: Train Model

Modeling

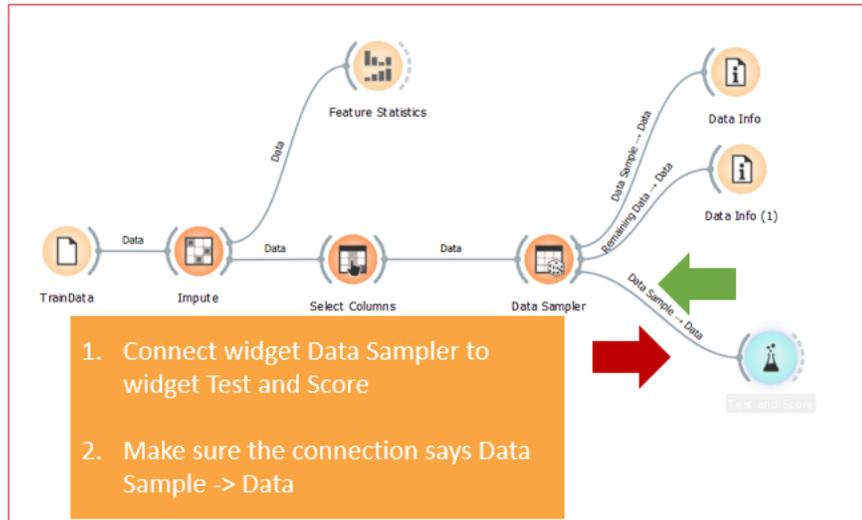


## Step 5(a)

Insert the Test and Score widget onto the canvas

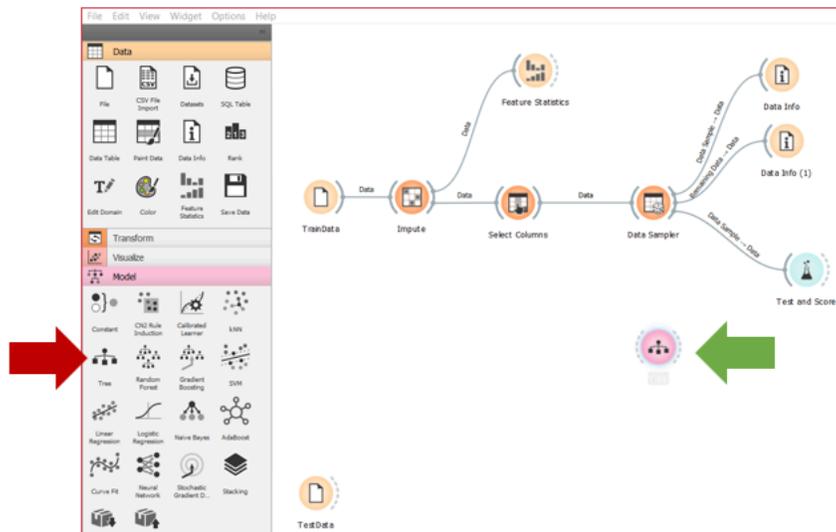


## Step 5(b)



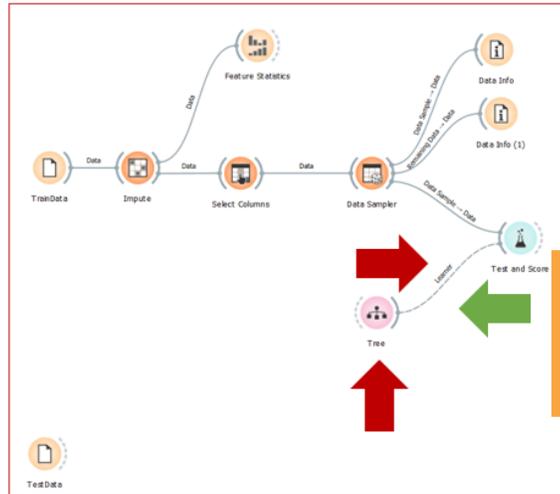
## Step 5(c)

Insert the widget Tree into the canvas and put it to the left of widget Test and Score



## Step 5(d)

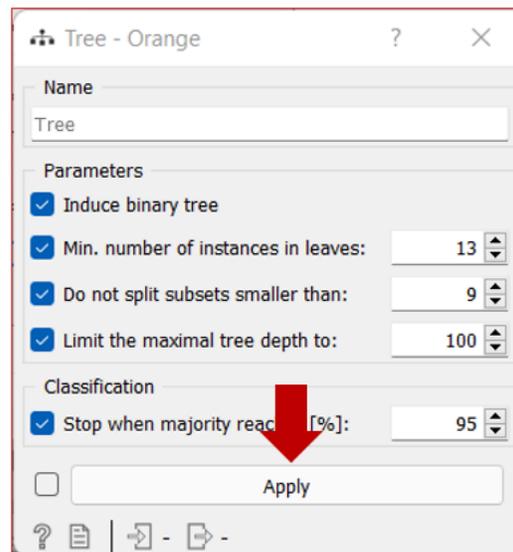
1. Connect widget Tree to widget Test and Score
2. Double click on the model icon to open its properties



If the connection is dotted it means the model has not been applied yet

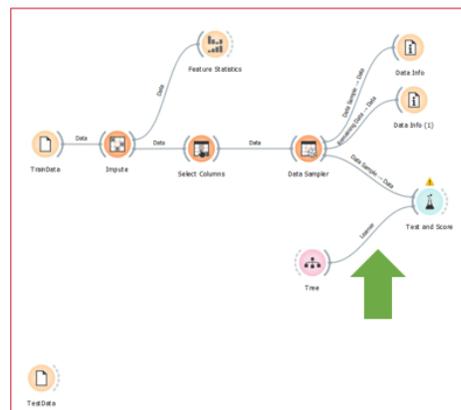
## Step 5(e)

1. Click on Apply if automatically apply has not been checked
2. Click X to close the pop-up window



## Step 5: Train Model

The model is ready

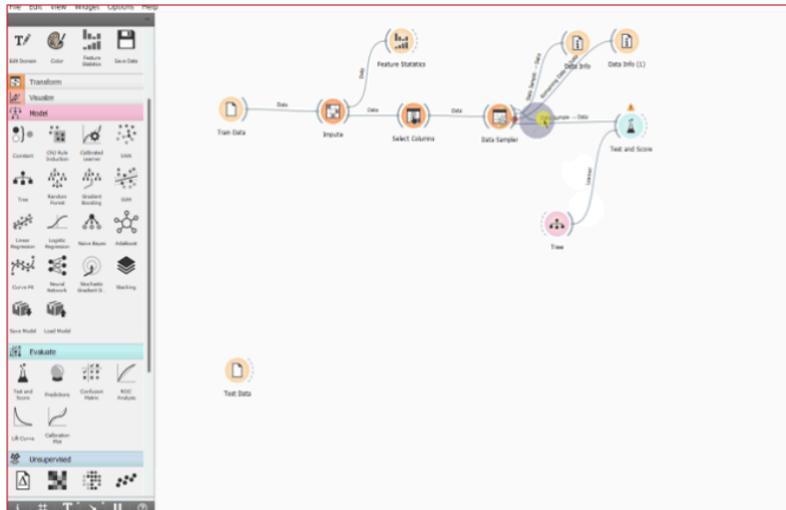


If there is a warning sign on the test and score widget, we will need to change its properties. Let's talk about it next

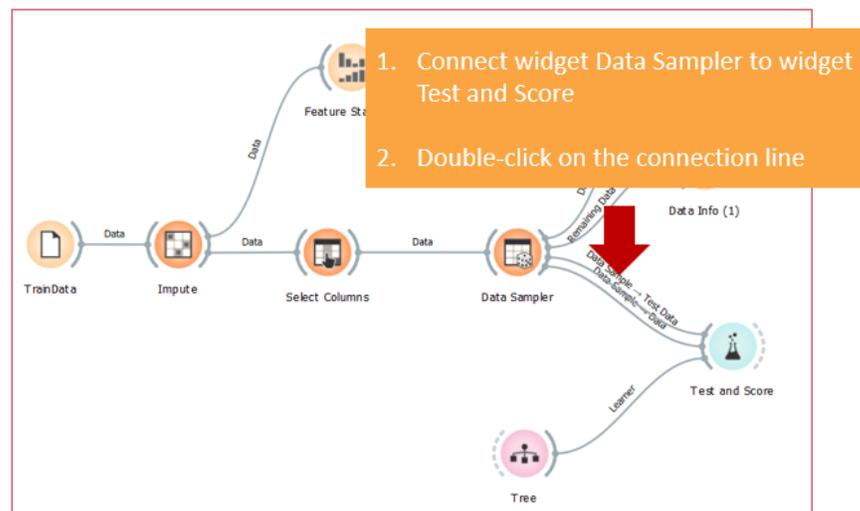
After creating a model, we need to test the model and check its accuracy

## Step 6: Evaluate Model

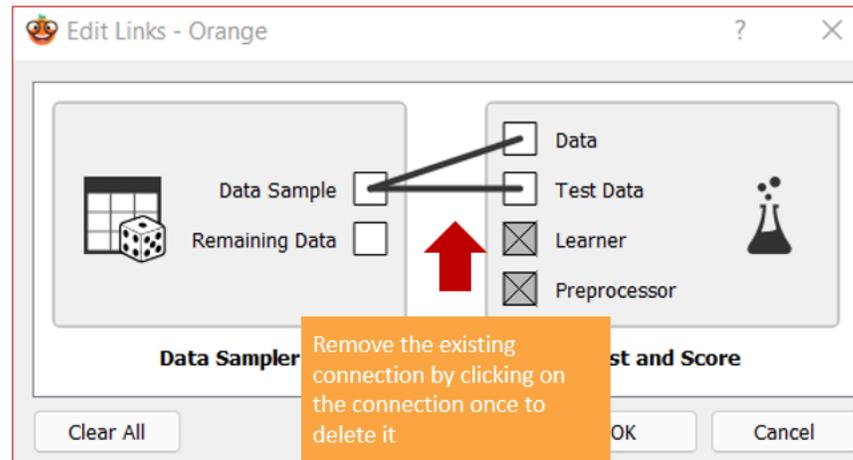
Evaluation



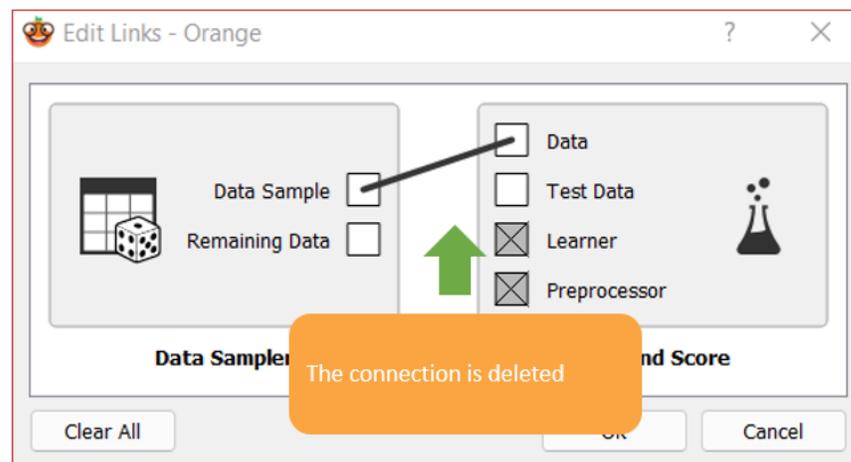
### Step 6(a)



Step 6(b)



Step 6(b)



Step 6(c)

1. Create a connection between Remaining Data and Test Data

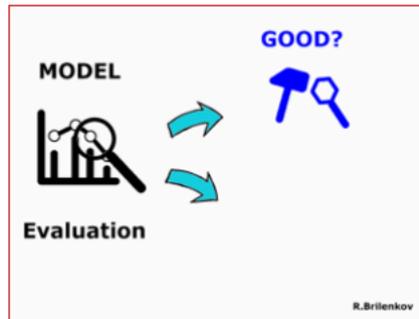
2. Click on OK

## Step 6(d)

We can observe the different scores for the model

Model	AUC	CA	F1	Precision	Recall
Tree	0.970	0.926	0.928	0.931	0.926

Click on X to close the pop-up

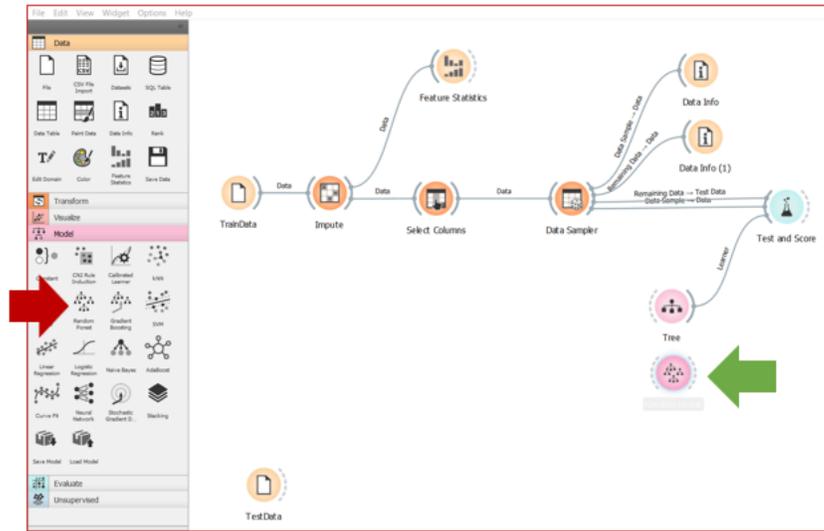


Through Evaluation we know if a model is good or bad

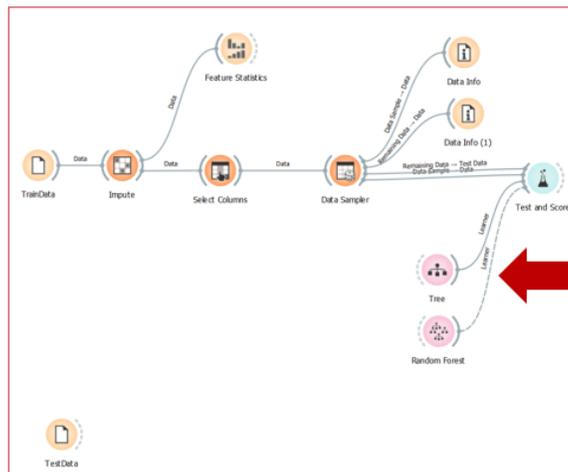
Let's try a couple of other classification algorithms

### Step 6(h)

Insert the widget  
Random Forest into  
the canvas



### Step 6(i)



Connect the Random Forest  
widget to the Test and  
Score widget

## Step 6(j)

This pop up will appear  
We can observe the individual scores for each algorithm here

Model	AUC	CA	F1	Precision	Recall
Tree	0.970	0.926	0.928	0.931	0.926
Random Forest	1.000	1.000	1.000	1.000	1.000

Double click on the  
Test and Score widget

## Step 6(k)

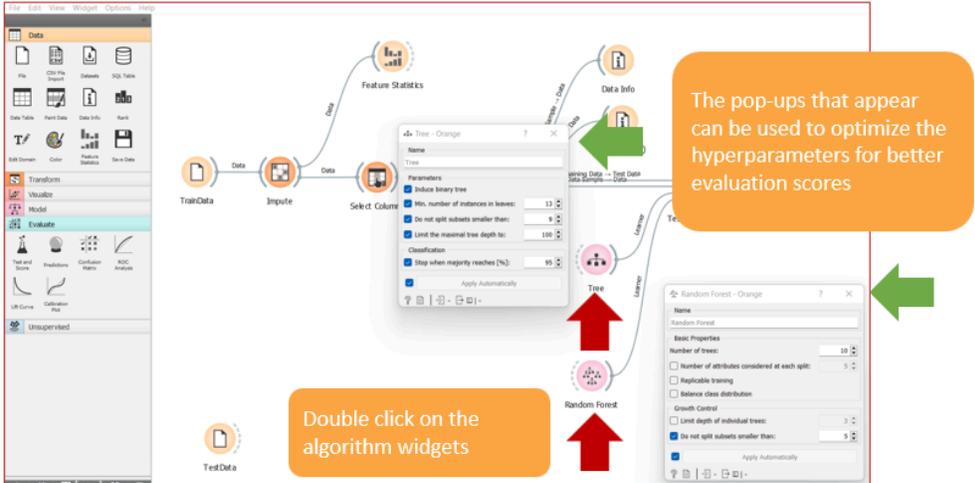
We can choose the  
methods for  
evaluation in this  
pop up

Model	AUC	CA	F1	Precision	Recall
Tree	0.958	0.931	0.930	0.930	0.931
Random Forest	0.997	0.982	0.982	0.982	0.982

Compare models by:	Area under ROC curve	Negligible diff.:	0.1
Tree	Tree	Random...	0.071
Random Forest	0.929		

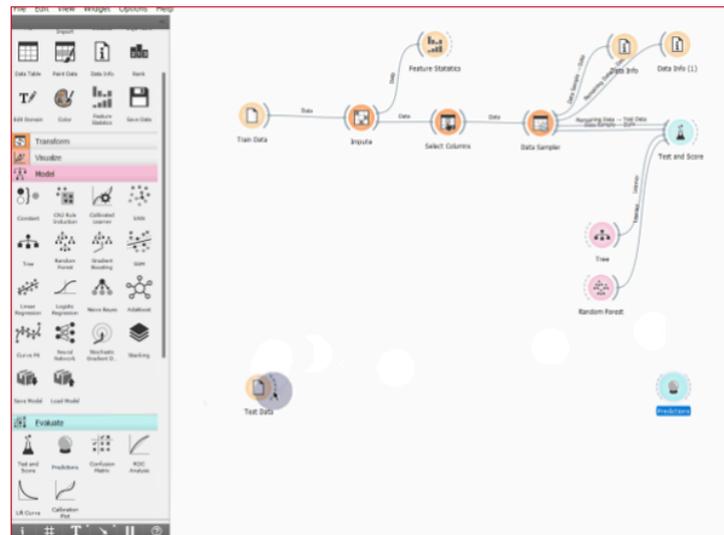
# Step 6(I)



Now that we have found which model gives us the best results, we can use that one!

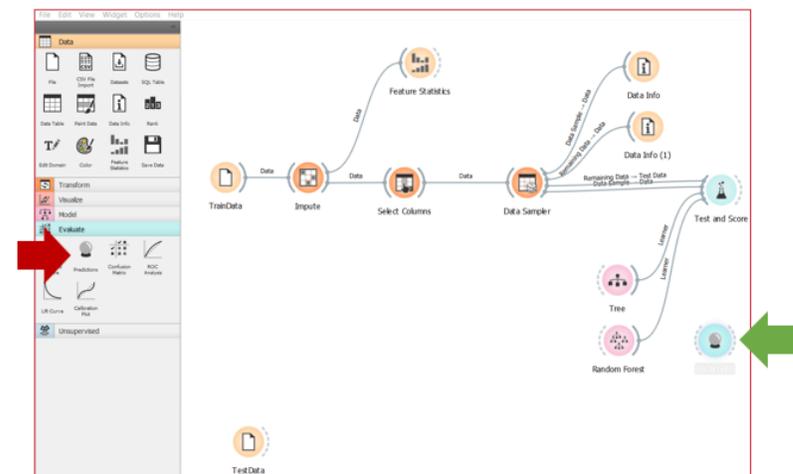
## Step 7: Predictions

Predictions

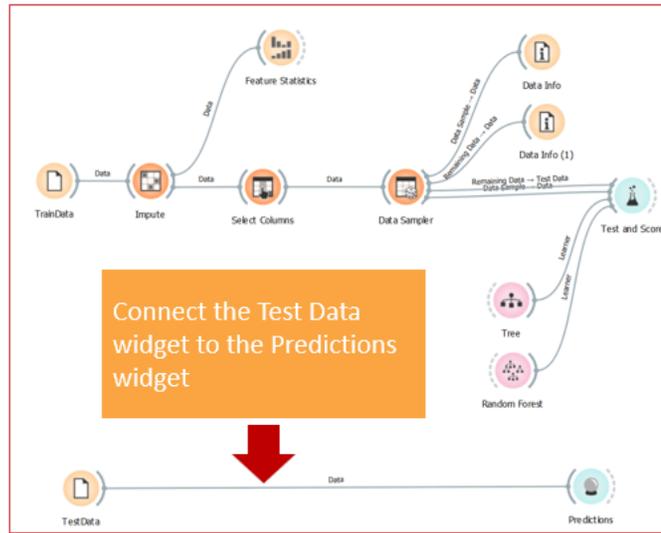


### Step 7(a)

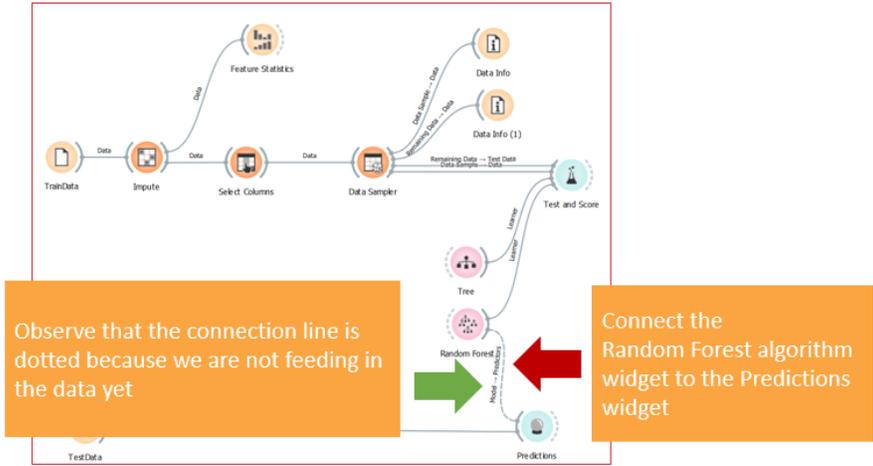
Insert the widget Predictions into the canvas



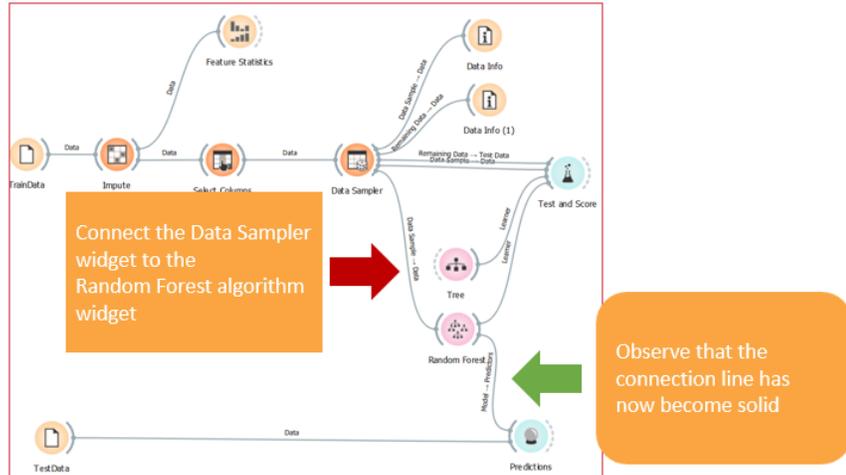
## Step 7(b)



### Step 7(b)



### Step 7(c)



## Step 7(d)

Predictions pop up appears

Double click on the Predictions widget

Random Forest	actual species	Island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g
1	Adelie	Torgersen	39.1	18.7	181	3750
2	Adelie	Torgersen	39.5	17.4	186	3800
3	Adelie	Torgersen	40.3	18.0	195	3250
4	Adelie	Torgersen	?	?	?	?
5	Adelie	Torgersen	36.7	19.3	193	3450
6	Adelie	Torgersen	39.3	20.6	190	3650
7	Adelie	Torgersen	38.9	17.8	181	3625
8	Adelie	Torgersen	39.2	19.6	195	4675
9	Adelie	Torgersen	34.1	18.1	193	3475
10	Adelie	Torgersen	42.0	20.2	190	4250
11	Adelie	Torgersen	37.8	17.1	186	3300
12	Adelie	Torgersen	37.8	17.3	180	3700
13	Adelie	Torgersen	41.1	17.6	182	3200
14	Adelie	Torgersen	38.6	21.2	191	3800
15	Adelie	Torgersen	34.6	21.1	198	4400
16	Adelie	Torgersen	36.6	17.8	185	3700
17	Adelie	Torgersen	38.7	19.0	195	3450
18	Adelie	Torgersen	42.5	20.7	197	4300
19	Adelie	Torgersen	34.4	18.4	184	3325
20	Adelie	Torgersen	46.0	21.5	194	4200
21	Adelie	Biscoe	37.8	18.3	174	3400
22	Adelie	Biscoe	37.7	18.7	180	3600
23	Adelie	Biscoe	35.9	19.2	189	3800
24	Adelie	Biscoe	38.2	18.1	185	3950
25	Adelie	Biscoe	38.8	17.2	180	3800
26	Adelie	Biscoe	35.3	18.9	187	3800

## STEP 7.1. PREDICTIONS

Random Forest Predictions

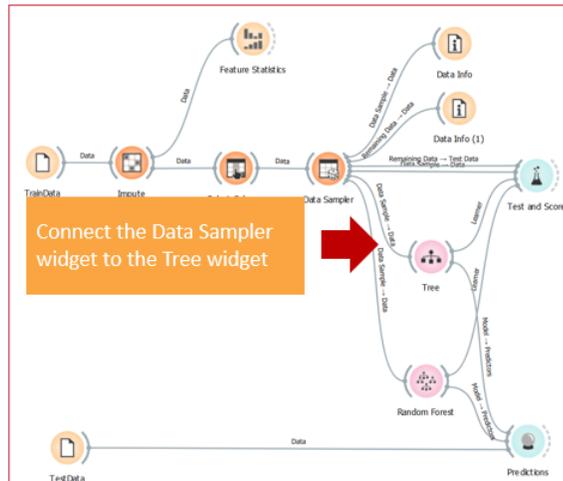
Actual Species

Observe that the predictions made for Chinstrap by Random Forest are false  
Random Forest is classifying Chinstrap as Adelie

Random Forest	actual species	Island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g
154	Adelie	Dream	45.4	18.7	188	3525
155	Adelie	Dream	52.7	19.8	197	3725
156	Adelie	Dream	45.2	17.8	198	3950
157	Adelie	Dream	46.1	18.2	178	3250
158	Adelie	Dream	51.3	18.2	197	3750
159	Adelie	Dream	46.0	18.9	195	4150
160	Adelie	Dream	51.3	19.9	198	3700
161	Adelie	Dream	46.6	17.8	193	3800
162	Adelie	Dream	51.7	20.3	194	3775
163	Adelie	Dream	47.0	17.3	185	3700
164	Adelie	Dream	52.0	18.1	201	4050
165	Adelie	Dream	45.9	17.1	190	3575
166	Adelie	Dream	50.5	19.6	201	4050
167	Adelie	Dream	50.3	20.0	197	3300
168	Adelie	Dream	58.0	17.8	181	3700
169	Adelie	Dream	46.4	18.6	190	3450
170	Adelie	Dream	49.2	18.2	195	4400
171	Adelie	Dream	42.4	17.3	181	3600
172	Chinstrap	Dream	48.5	17.5	191	3400
173	Adelie	Dream	43.2	16.6	187	2900
174	Adelie	Dream	50.6	19.4	193	3800
175	Adelie	Dream	46.7	17.9	195	3300
176	Adelie	Dream	52.0	19.0	197	4150
177	Adelie	Dream	50.5	18.4	200	3400
178	Adelie	Dream	49.5	19.0	200	3800
179	Adelie	Dream	46.4	17.8	191	3700

Since the Random Forest algorithm is not working well with one of the species, let's use another algorithm

## Step 7(e)



Now we are using multiple models at the same time

## Step 7: Predictions

Random Forest Predictions

Tree Predictions

Actual Species

Random Forest	Tree	actual species	Island	ulmen_length_mm	ulmen_depth_mm	tipper_length_mm
Adelie	Chinstrap	Chinstrap	Dream	52.0	18.1	201
Adelie	Chinstrap	Chinstrap	Dream	45.9	17.1	190
Adelie	Chinstrap	Chinstrap	Dream	50.5	19.6	201
Adelie	Chinstrap	Chinstrap	Dream			
Adelie	Chinstrap	Chinstrap	Dream			
Adelie	Chinstrap	Chinstrap	Dream			
Chinstrap	Chinstrap	Chinstrap	Dream			
Adelie	Chinstrap	Chinstrap	Dream			
Chinstrap	Chinstrap	Chinstrap	Dream			
Adelie	Chinstrap	Chinstrap	Dream	48.5	17.5	191
Chinstrap	Chinstrap	Chinstrap	Dream	43.2	16.6	187
Adelie	Chinstrap	Chinstrap	Dream	50.6	19.4	193
Adelie	Chinstrap	Chinstrap	Dream	46.7	17.9	195
Adelie	Chinstrap	Chinstrap	Dream	52.0	19.0	197
Adelie	Chinstrap	Chinstrap	Dream	50.5	18.4	200
Adelie	Chinstrap	Chinstrap	Dream	49.5	19.0	200
Adelie	Chinstrap	Chinstrap	Dream	46.4	17.8	191
Chinstrap	Chinstrap	Chinstrap	Dream	52.8	20.0	205
Adelie	Adelie	Chinstrap	Dream	40.9	16.6	187
Adelie	Chinstrap	Chinstrap	Dream	54.2	20.8	201
Adelie	Chinstrap	Chinstrap	Dream	42.5	16.7	187
Adelie	Chinstrap	Chinstrap	Dream	51.0	18.8	203
Adelie	Chinstrap	Chinstrap	Dream	49.7	18.6	195
Adelie	Chinstrap	Chinstrap	Dream	47.5	16.8	199
Adelie	Chinstrap	Chinstrap	Dream	47.6	18.3	195
Adelie	Gentoo	Chinstrap	Dream	52.0	20.7	210
Adelie	Chinstrap	Chinstrap	Dream	46.9	16.6	192

Observe that the predictions made for Chinstrap by Tree are correct

This suggests that some models give better results than others