Badge2-Lab-1 [Decision Tree]

Out date: July 11, 2022
Due date: July 17, 2022 at 11:59PM

Submission

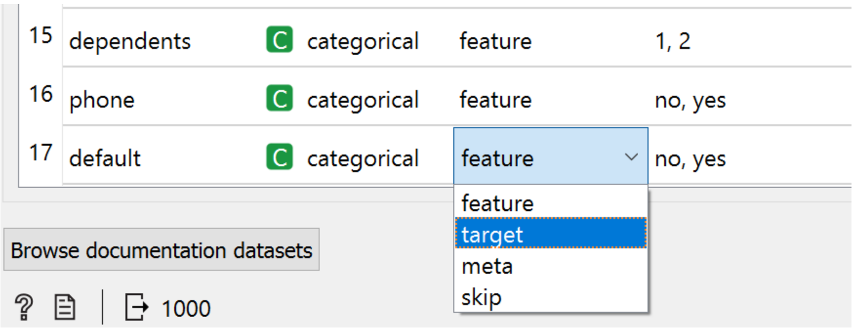
1. Prepare your solution in Orange and save the workspace for Problem 1 (e.g., Badge2_Lab-1_LastName.ows) **[20 points]**
2. Complete the tables given below and save the file (e.g., Badge2_Lab-1_LastName.docx). **[80 points]**
3. Upload the files to the Canvas.

Objective: To review and understand decision tree algorithm available in Orange for classification problems.

Data: For this lab, please download *credit.csv* from Canvas to your folder. The dataset contains information on loans obtained from a credit agency in Germany. Data is available on UCI ML website (http://archive.ics.uci.edu/ml/datasets/Statlog+%28German+Credit+Data%29)

Lab Instructions

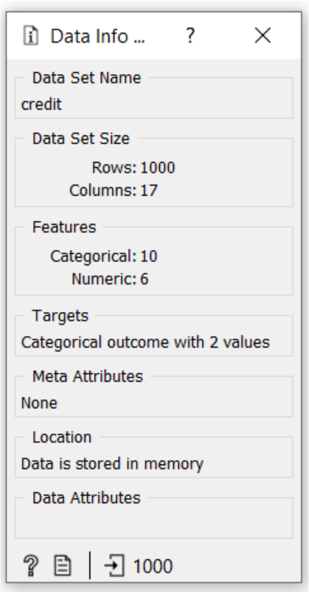
1. Load the *credit.csv*.
2. Open File window by double clicking on **File**.
3. Change the **default** feature to target as shown below.



15	dependents	C categorical	feature	1, 2
16	phone	C categorical	feature	no, yes
17	default	C categorical	feature	no, yes

4. Answer the following questions for this data:

How many objects/rows in the data set? 1000	
What is the dimensionality/columns of this data?	17
What are the class levels of the target feature?	no, yes



Data Info ...

Data Set Name
credit

Data Set Size
Rows: 1000
Columns: 17

Features
Categorical: 10
Numeric: 6

Targets
Categorical outcome with 2 values

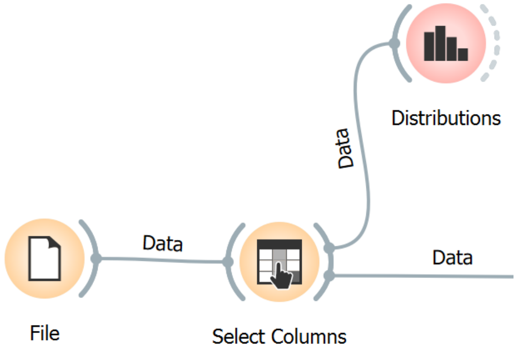
Meta Attributes
None

Location
Data is stored in memory

Data Attributes

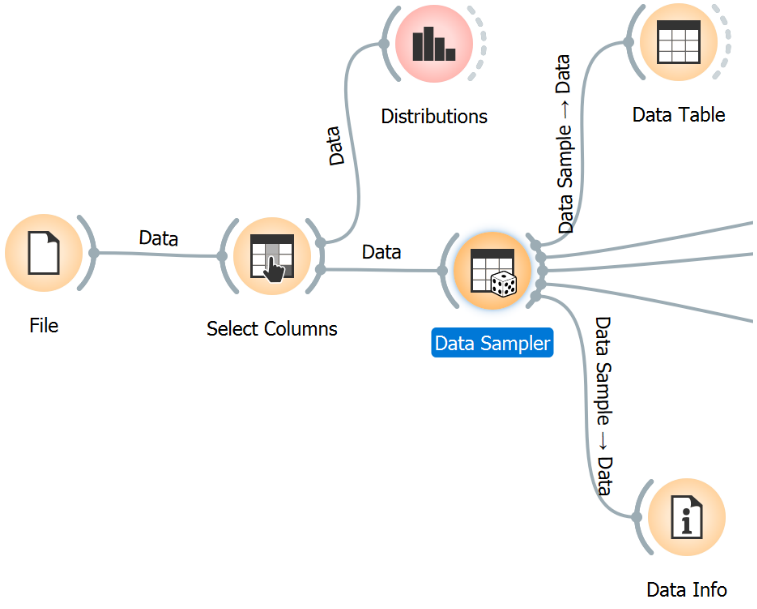
1000

5. Add the **Select Columns** and **Distribution** widget as shown below.



Comment on the class distribution.	Target (default) has bar graphs, not equally balanced
	No = 70% (700/1000)
	Yes = 30% (300/1000)

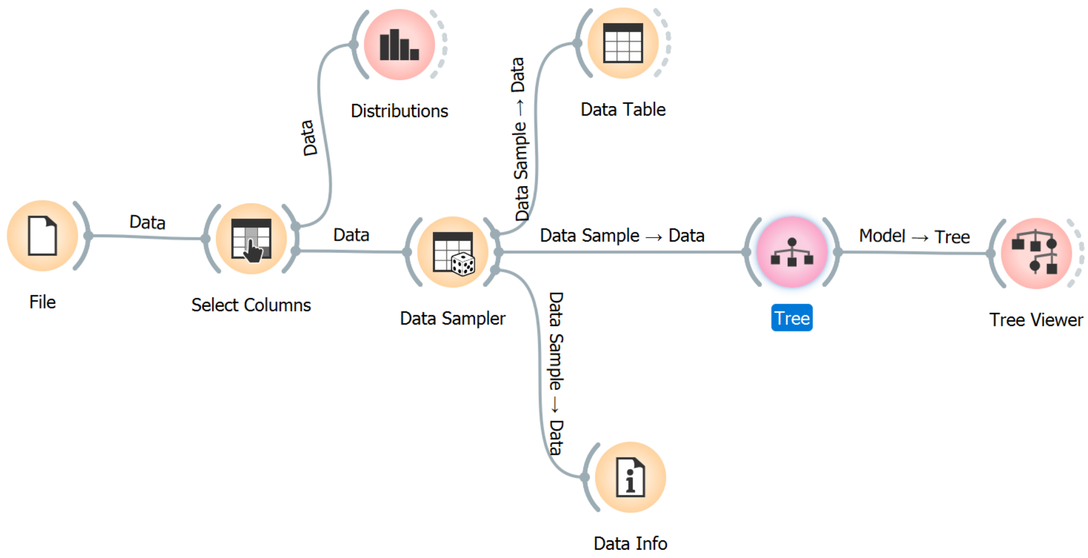
6. Add the **Data Sampler, Data Table,** and **Data Info** widget as shown below:



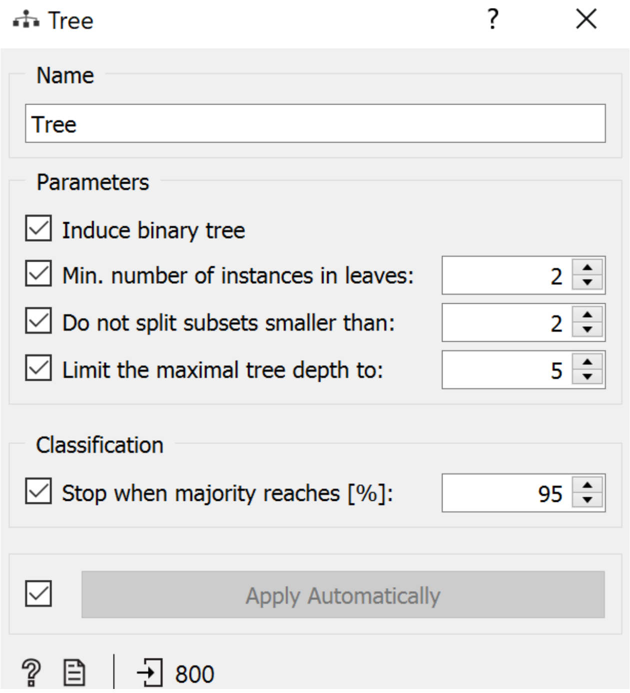
7. Double click on **Data Sampler** and set the ***Fixed proportion of data*** to 80%.
8. Complete the following table.

Size of the training Samples	800 rows, 17 columns
Size of the test Samples	200 rows, 17 columns

9. Add the **Tree** and **Tree Viewer** widgets as shown below.



10. Double click on **Tree** and change the settings as shown below.



Tree

Name

Tree

Parameters

☑ Induce binary tree

☑ Min. number of instances in leaves: 2

☑ Do not split subsets smaller than: 2

☑ Limit the maximal tree depth to: 5

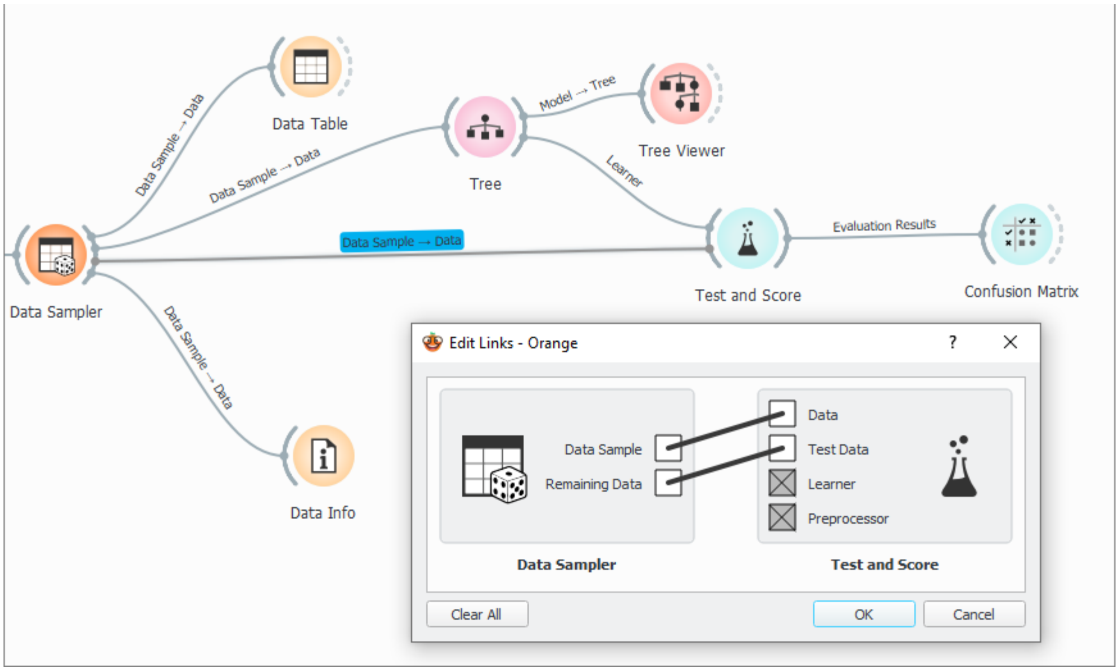
Classification

☑ Stop when majority reaches [%]: 95

☑ Apply Automatically

800

11. Add the **Test and Score and Confusion Matrix** widgets as shown below.



Open **Test and Score** widget and select sampling method as 10-fold cross validation.

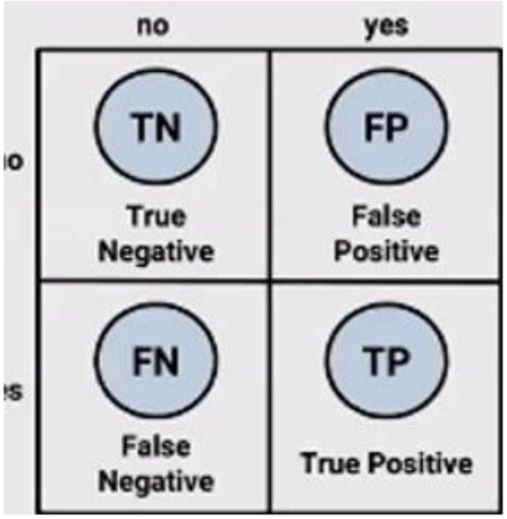
***** Model is Ready → Now let's understand the model *****

12.Complete the following table. Positive class is default – **'yes'**

Min # of instances in leaves	Don't split subsets smaller than	Depth	Stop when majority reaches	CA (accuracy)	F1	TP (M→M) Bottom Right	FP (B→M) Upper Right	Comments
2	2	2	95%	0.719	0.247	37	22	0.141 Train Time 37/240 yes
2	2	5	95%	0.750	0.429	75	35	0.713 Train Time 75/240 yes

2	2	20	95%	0.728	0.509	113	91	3.265 Train Time 113/240 yes (predicted correctly) But test on test data CA =0.695 (worse than others; overfitting)
10	10	20	95%	0.715	0.470	101	89	1.688 Train Time 101/240 yes
10	10	20	60%	0.700	0.000	0	0	0.008 Train Time 0/240 yes No info gained (ZeroR)

Baseline: ZeroR → Label every test sample with majority class. → No-class is 66.5% and Yes-class is 33.5% → ZeroR accuracy is 66.5%



13.Answer the following question

Which model are you going to put in production?				
2	2	20	95%	assuming cross validation (10 folds) only

Lowest false positive better, usually