



MSBA 265

ASSIGNMENT 5

CLASSIFICATION

Using Iris Dataset and RapidMiner

Phuc Minh Thao Pham - 989468542

Dataset

Existing dataset in RapidMiner, Iris.

It consists of 150 samples of iris flowers, with three different species:

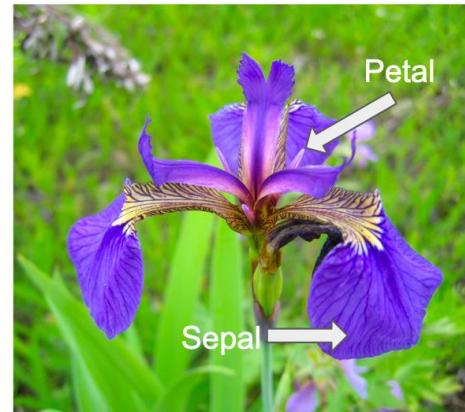
- setosa
- versicolor
- virginica

Each sample includes four features measured in centimeters:

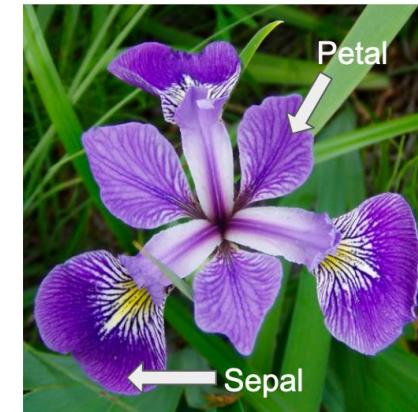
- a2: The **length** of the **sepal** of the flower
- a3: The **width** of the **sepal** of the flower
- a4: The **length** of the **petal** of the flower
- a5: The **width** of the **petal** of the flower

These features are used to distinguish between the three species.

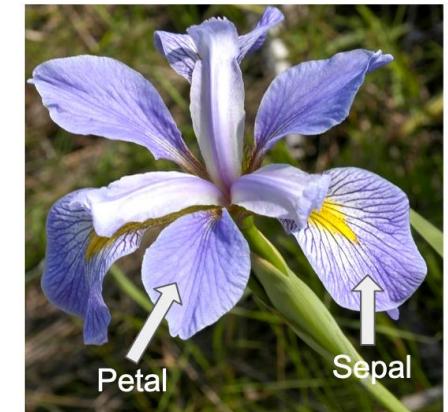
Iris setosa



Iris versicolor



Iris virginica

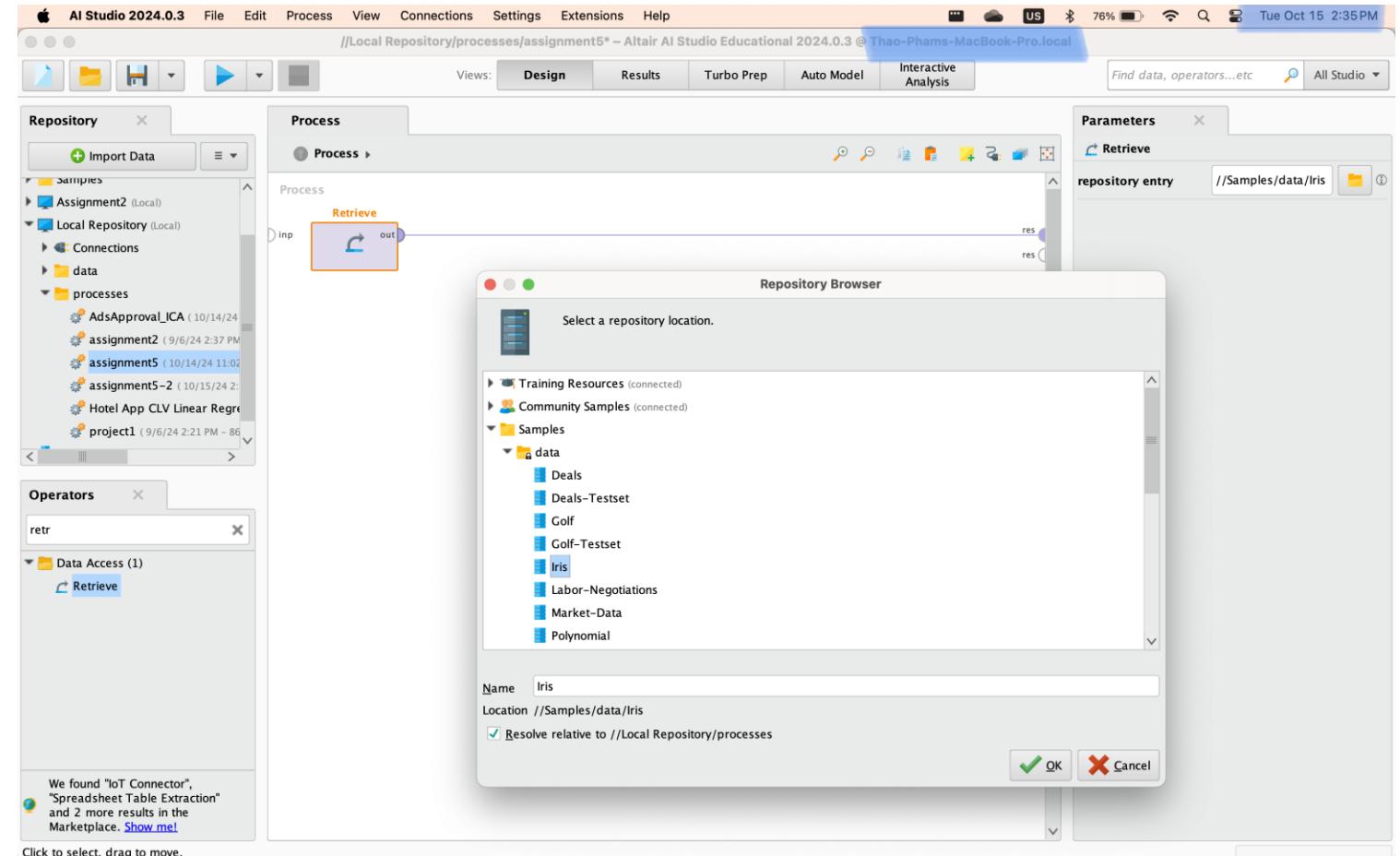


A screenshot of the AI Studio 2024.0.3 software interface. The top menu bar includes File, Edit, Process, View, Connections, Settings, Extensions, and Help. The toolbar below has icons for Load Data, Transform, Cleanse, Generate, Pivot, Merge, Model, Charts, Create Process, and History. The main window shows the "Turbo Prep" interface with a "Data Sets" panel on the left containing a list of datasets and a preview of the Iris dataset. The Iris dataset preview shows four numerical columns (a1, a2, a3, a4) and two categorical columns (id and label). The label column contains values like "Iris-setosa". Below the preview are histograms for each column. The status bar at the bottom right shows the date and time: Tue Oct 15 2:23PM.

Retrieve Operator

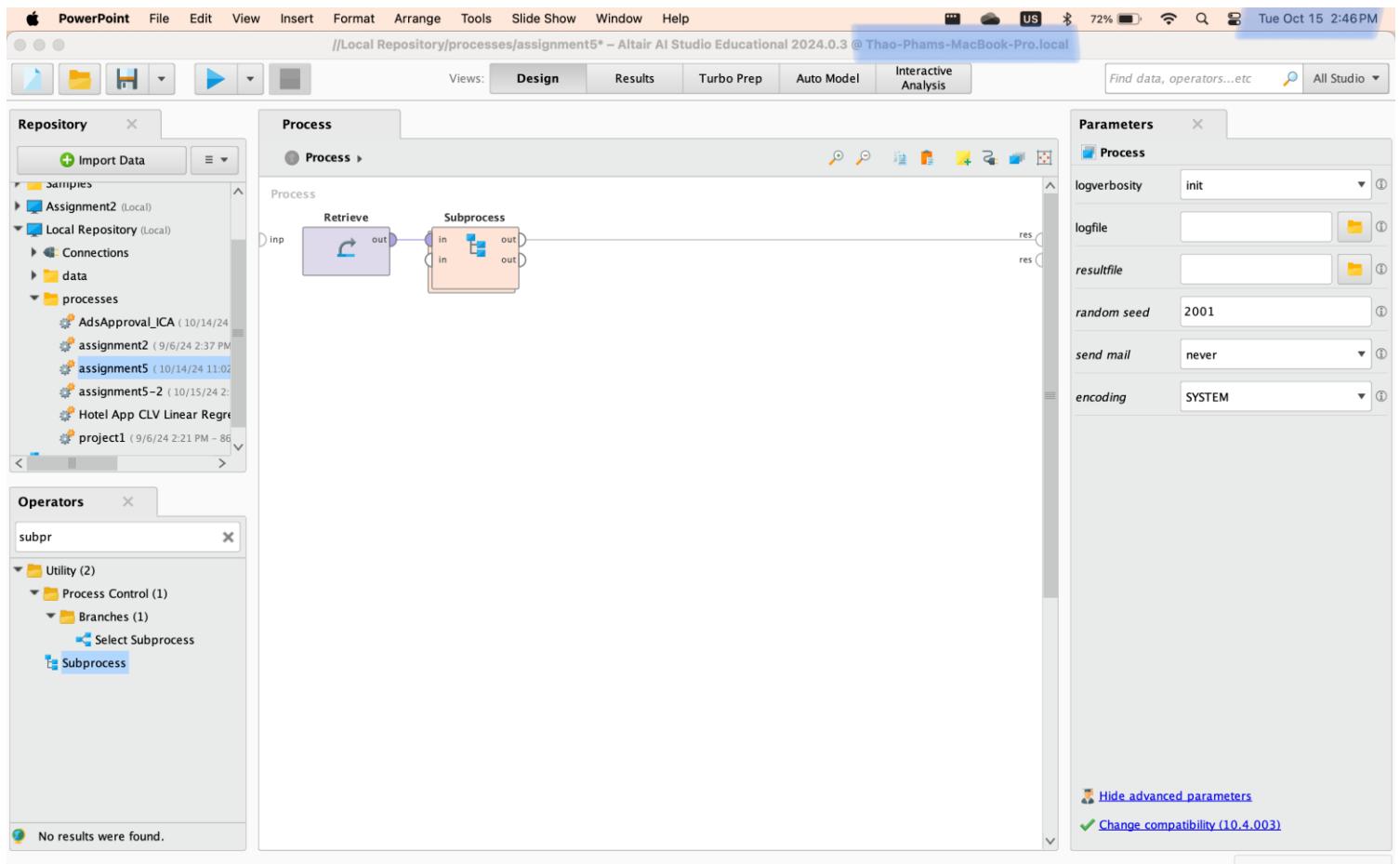
1. Operators tab, type **Retrieve** then drag and drop the operator to the Process.
2. Selecting a dataset:
 - Either double click on the **Retrieve** operator for the **Repository Browser** to pop up and select the **Iris** dataset.
 - Or click 1 time on the **Retrieve** operator and choose the **Iris** dataset in the **repository entry** of the operator's **Parameters** tab on the right-hand side.
3. Connection:

Retrieve.output - Process.result 1



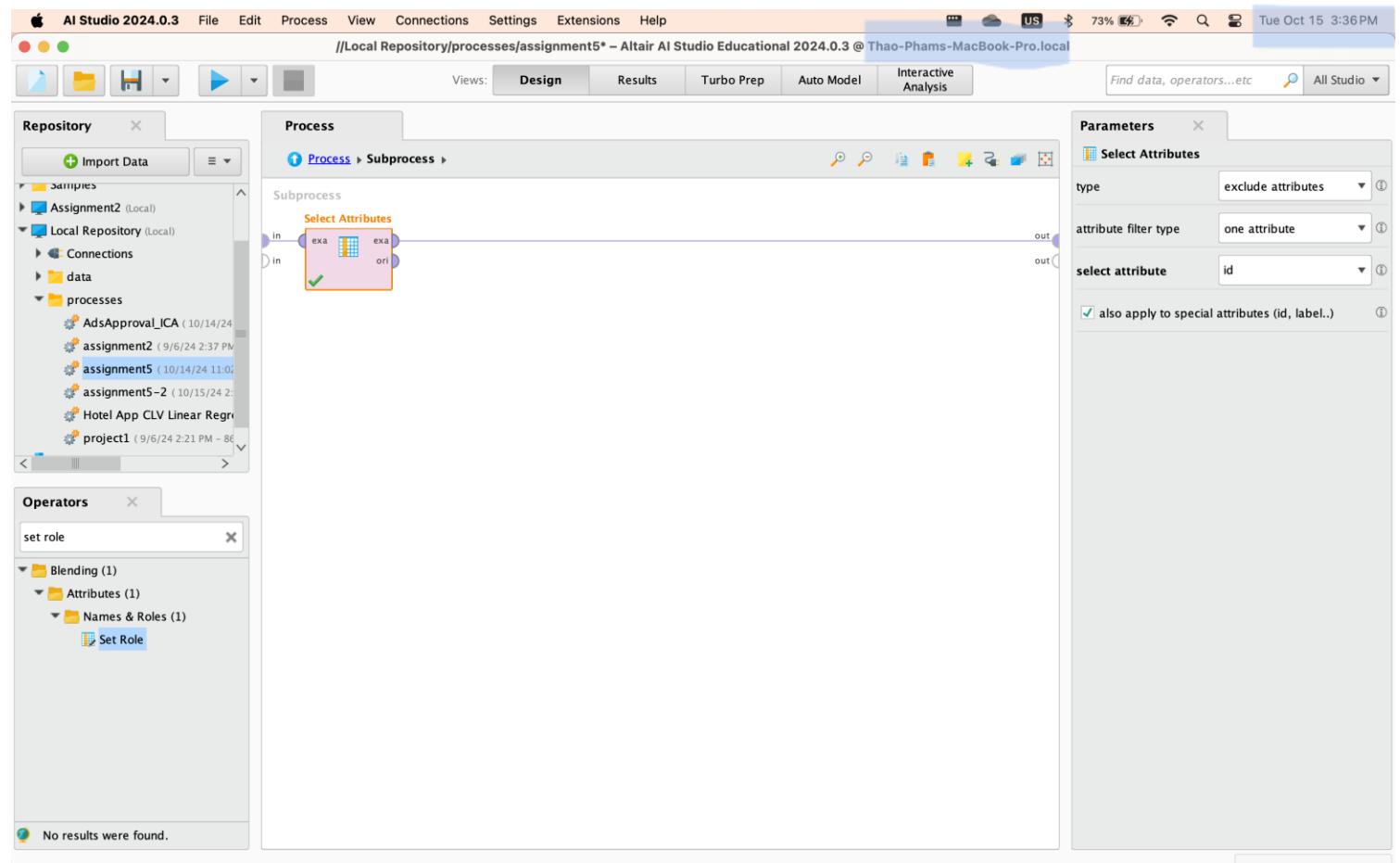
Subprocess Operator

1. Operators tab, type **Subprocess** then drag and drop the operator to the Process.
2. Connection:
Retrieve.output – Subprocess.in 1
Subprocess.out 1 – Process.result 1
3. Double-click on the **Subprocess** operator to get inside it.



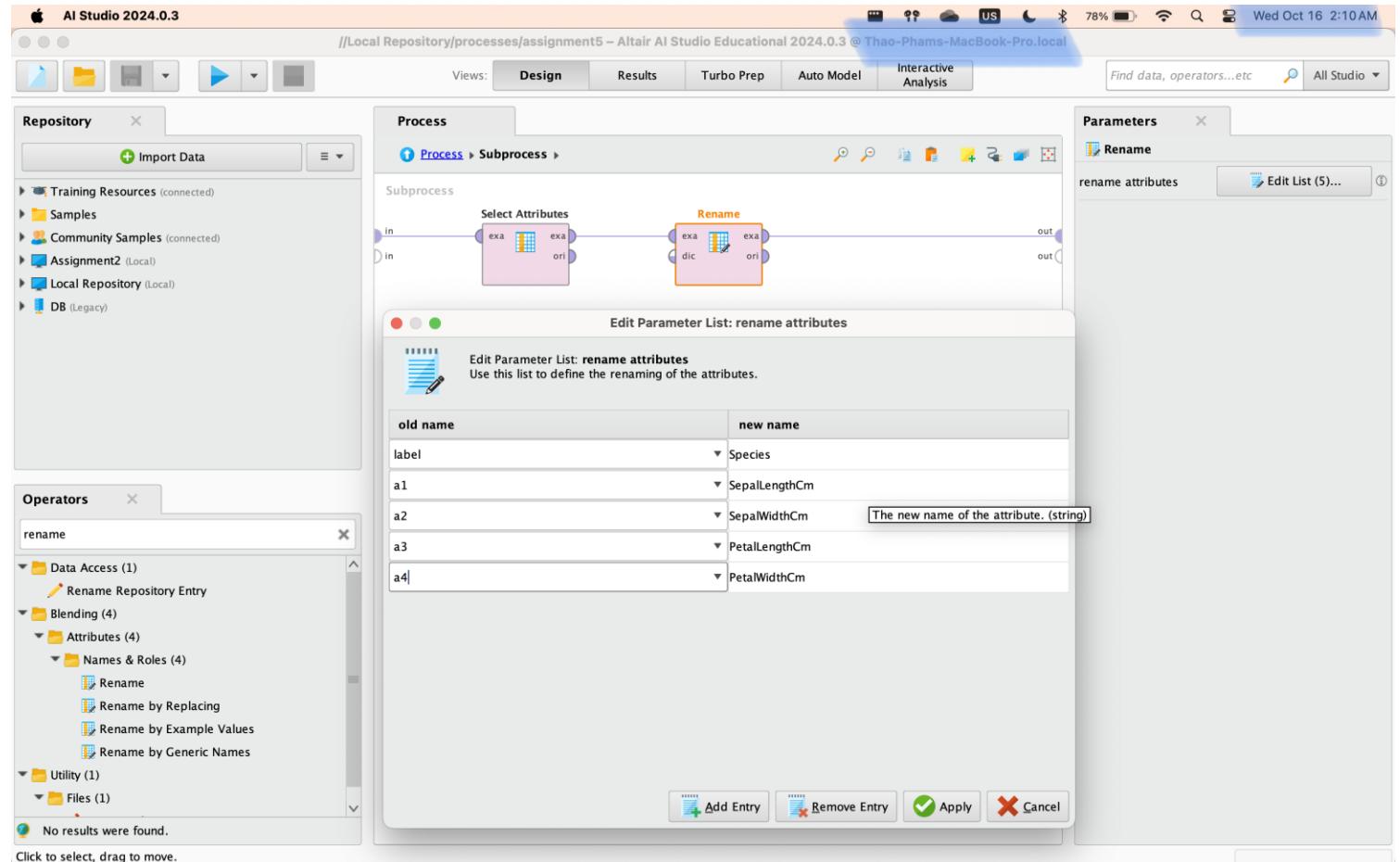
Select Attributes Operator

1. Operators tab, type **Select Attributes** then drag and drop the operator to the Subprocess.
2. Connection:
Subprocess.in 1 – Select Attributes.example set input
3. Selecting the same things from in the Parameters tab.
 - type : **exclude attributes**
 - attribute filter type : **one attribute**
 - select attribute : **id**
 - Selecting **also apply to special attributes (id, label..)**



Rename Operator

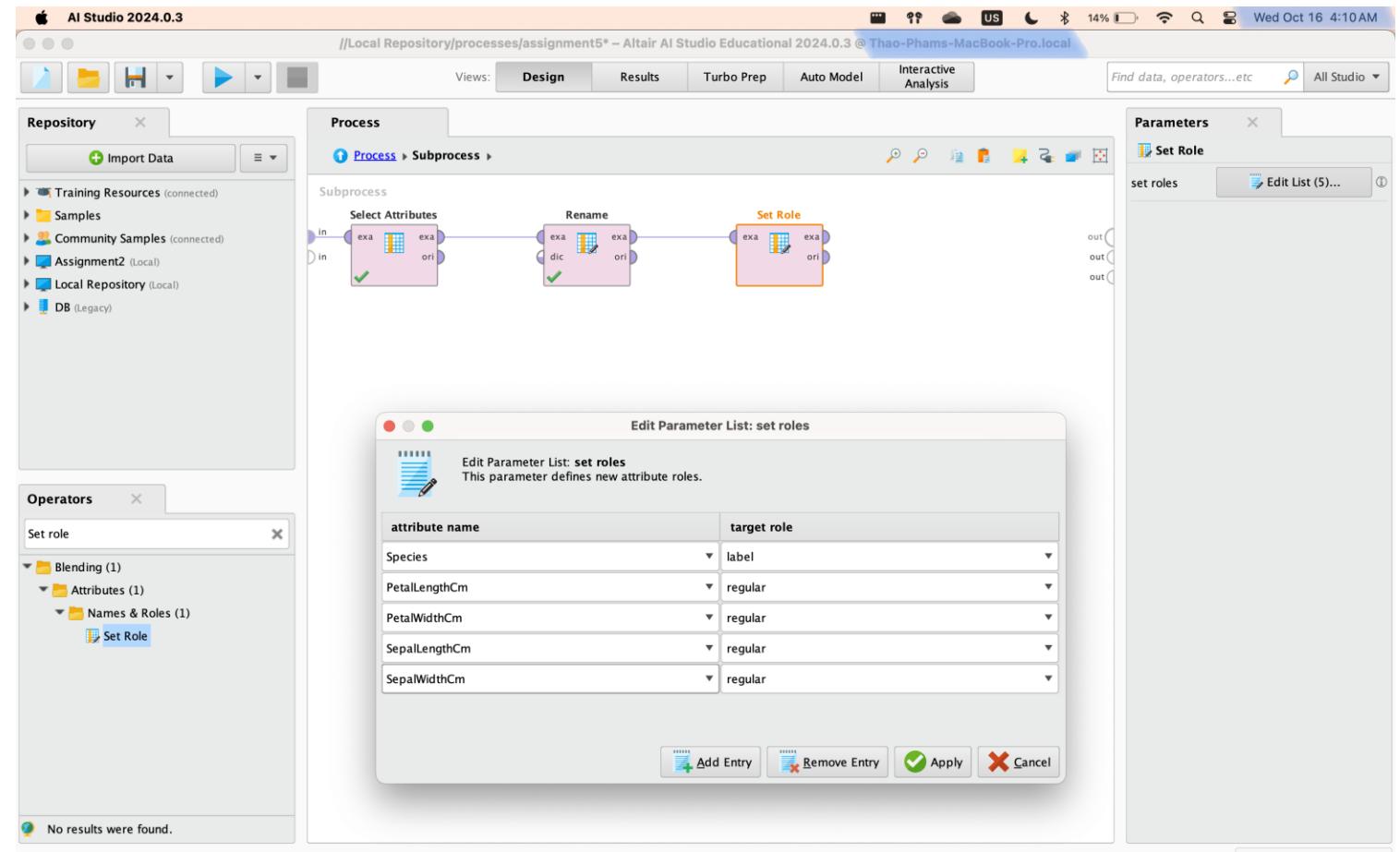
1. Operators tab, type **Rename** then drag and drop the operator to the **Subprocess**.
2. Renaming attributes by either double click on the **Rename** operator or click on the **Edit List** in the **Parameters** tab on the right-hand side. A browser will pop up and add entry like in the image. Click **Apply** when finish.
3. Connection:
Select Attributes.example set output - Rename.example set input



Set Role Operator

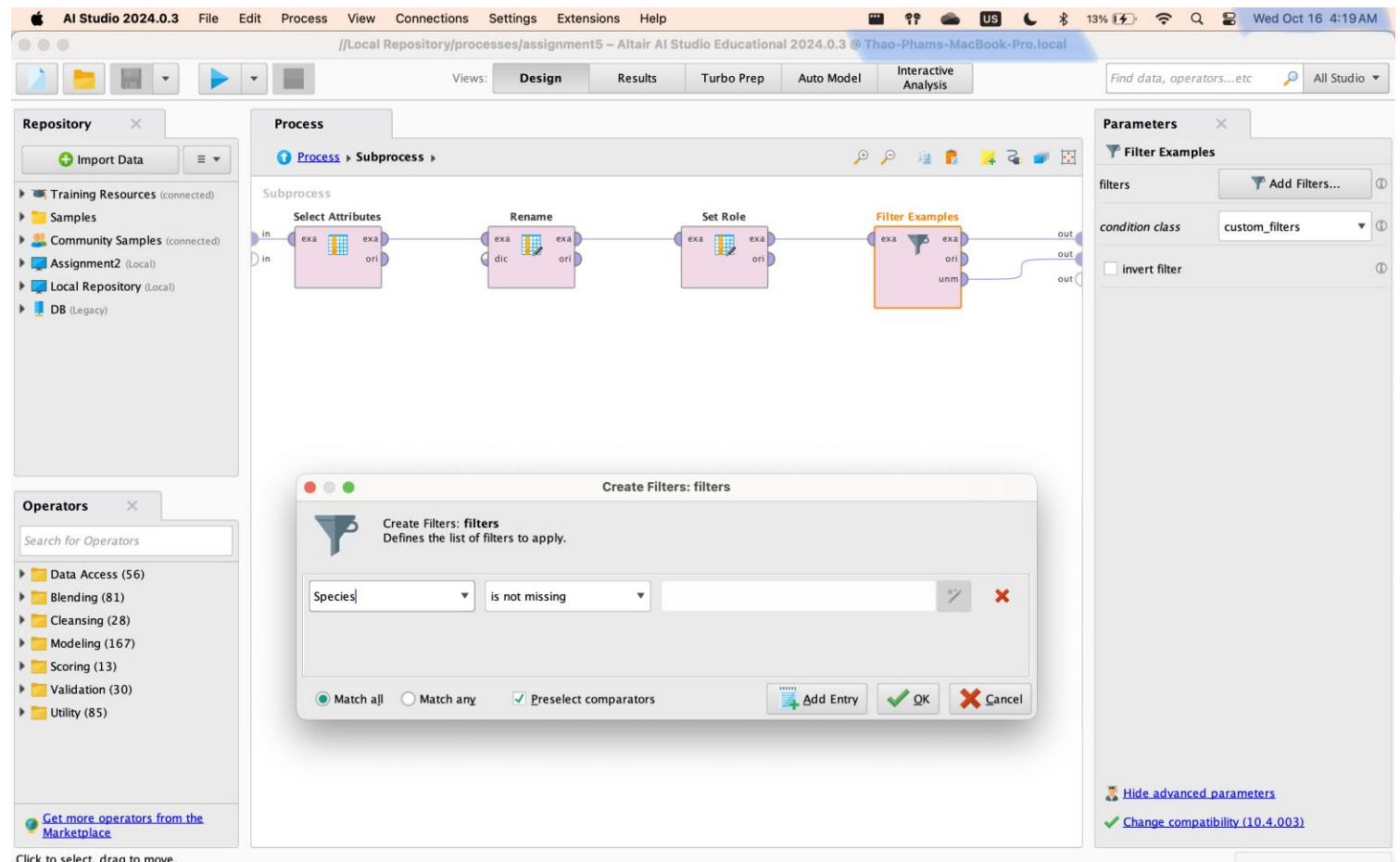
1. Operators tab, type **Set Role** then drag and drop the operator to the Subprocess.
2. Parameters tab, **Edit List**:
 - Species: *label* (dependent variable)
 - Other attributes: *regular* (independent variable)
3. Connection:

Rename.example set output - Set Role.example set input



Filter Examples Operator

1. Operators tab, type **Filter Examples** then drag and drop the operator to the Subprocess.
2. Renaming attributes by either double click on the **Filter Examples** operator or click on the **Add filters** in the **Parameters** tab on the right-hand side.
 - **Species : is not missing.**
 - Click **OK** when finish.
 - Setting the **condition class** as **custom_filters**.
3. Connections:
 - Set Role.example set output - Filter Examples.example set input
 - Filter Examples.example set output - Subprocess.out 1
 - Filter Examples.unmatched example set - Subprocess.out 2



Results of Subprocess Operator

- **Select Attributes:** excluding the **id** column/attribute.
- **Rename:** changing the remaining columns/attributes to understandable names.
 - $a_2 \rightarrow \text{SepalLengthCm}$
 - $a_3 \rightarrow \text{SepalWidthCm}$
 - $a_4 \rightarrow \text{PetalLengthCm}$
 - $a_5 \rightarrow \text{PetalWidthCm}$
- **Set Role:** assigning dependent and independent variables to the attributes.
 - **Specie \rightarrow label**
 - **SepalLengthCm \rightarrow regular**
 - **SepalWidthCm \rightarrow regular**
 - **PetalLengthCm \rightarrow regular**
 - **PetalWidthCm \rightarrow regular**
- **Filter Examples:** selecting what we don't want as part of the data split. What is missing in the Species column will go to validation set. If y is missing, it will be considered in the example set.

The screenshot shows the AI Studio 2024.0.3 interface with the 'Results' tab selected. A central window displays an 'ExampleSet (Filter Examples)' table with 150 rows. The table has columns: Row No., Species, SepalLengthCm, SepalWidthCm, PetalLengthCm, and PetalWidthCm. The 'Species' column contains values like 'Iris-setosa'. The table is part of a larger interface with tabs for Design, Results, Turbo Prep, Auto Model, and Interactive Analysis. On the left, there's a sidebar with icons for Data, Statistics, Visualizations, and Annotations. The status bar at the bottom indicates 'ExampleSet (150 examples, 1 special attribute, 4 regular attributes)'.

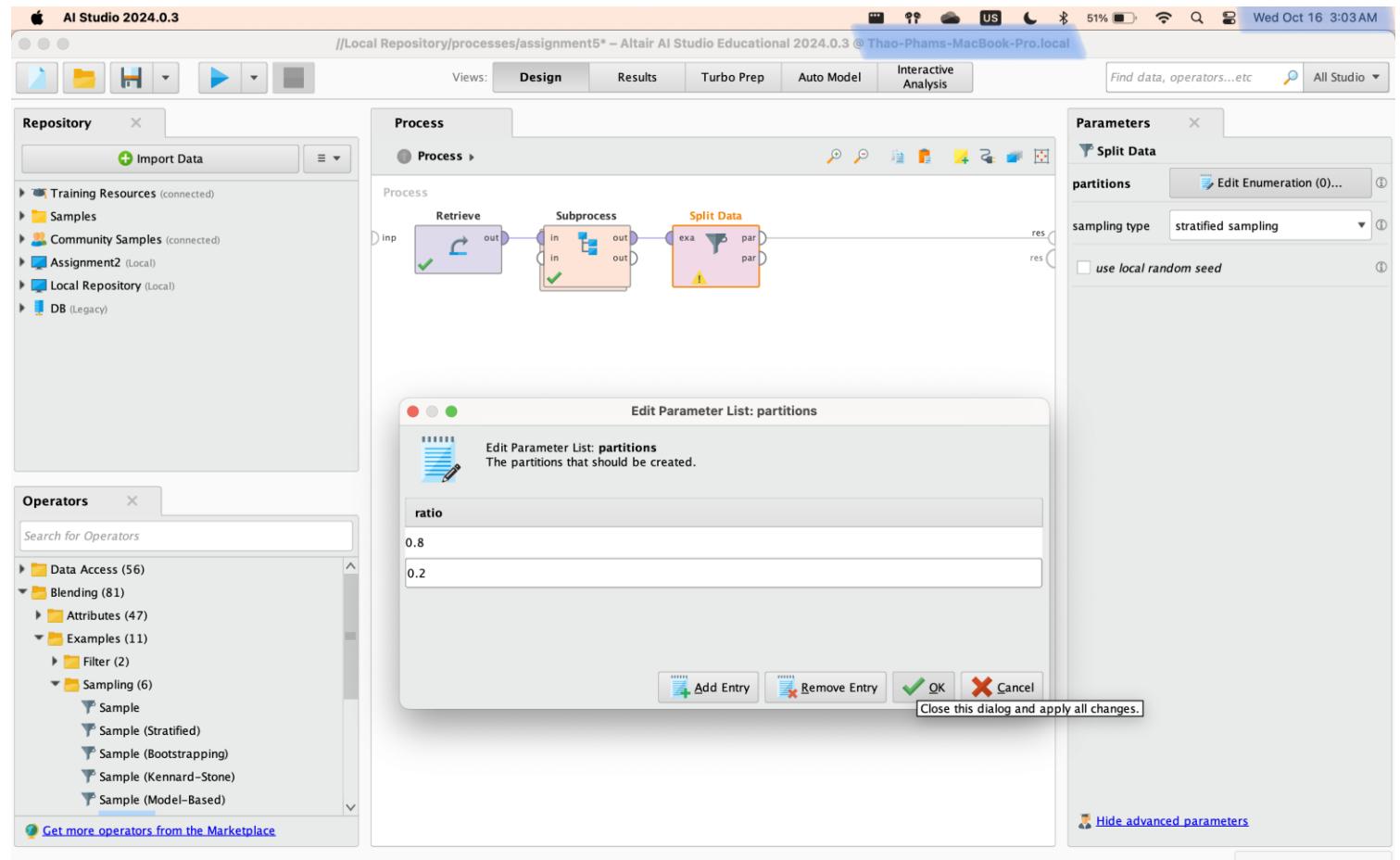
Row No.	Species	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
1	Iris-setosa	5.100	3.500	1.400	0.200
2	Iris-setosa	4.900	3	1.400	0.200
3	Iris-setosa	4.700	3.200	1.300	0.200
4	Iris-setosa	4.600	3.100	1.500	0.200
5	Iris-setosa	5	3.600	1.400	0.200
6	Iris-setosa	5.400	3.900	1.700	0.400
7	Iris-setosa	4.600	3.400	1.400	0.300
8	Iris-setosa	5	3.400	1.500	0.200
9	Iris-setosa	4.400	2.900	1.400	0.200
10	Iris-setosa	4.900	3.100	1.500	0.100
11	Iris-setosa	5.400	3.700	1.500	0.200
12	Iris-setosa	4.800	3.400	1.600	0.200
13	Iris-setosa	4.800	3	1.400	0.100
14	Iris-setosa	4.300	3	1.100	0.100
15	Iris-setosa	5.800	4	1.200	0.200
16	Iris-setosa	5.700	4.400	1.500	0.400
17	Iris-setosa	5.400	3.900	1.300	0.400
18	Iris-setosa	5.100	3.500	1.400	0.300
19	Iris-setosa	5.700	3.800	1.700	0.300

Selecting Design to go back to the **Subprocess**

Exiting the **Subprocess** by clicking on the **Process**

Split Data Operator

1. Operators tab, type **Split Data** then drag and drop the operator to the Process.
2. Parameters tab,
 - **partitions** : Edit Enumeration
 - Add Entry : 0.8
 - Add Entry : 0.2
 - Click OK
 - **sampling type** : Stratified sampling
3. Connection:
 - Subprocess.out 1 - Split Data.example set
 - Split Data.partition 1 - Process.result 1



Result of Split Data Operator

The operator divides the dataset into 2 parts.

- 80%: training and testing
- 20%: validation

The image displays 120 rows as the result of the splitting for training and testing.

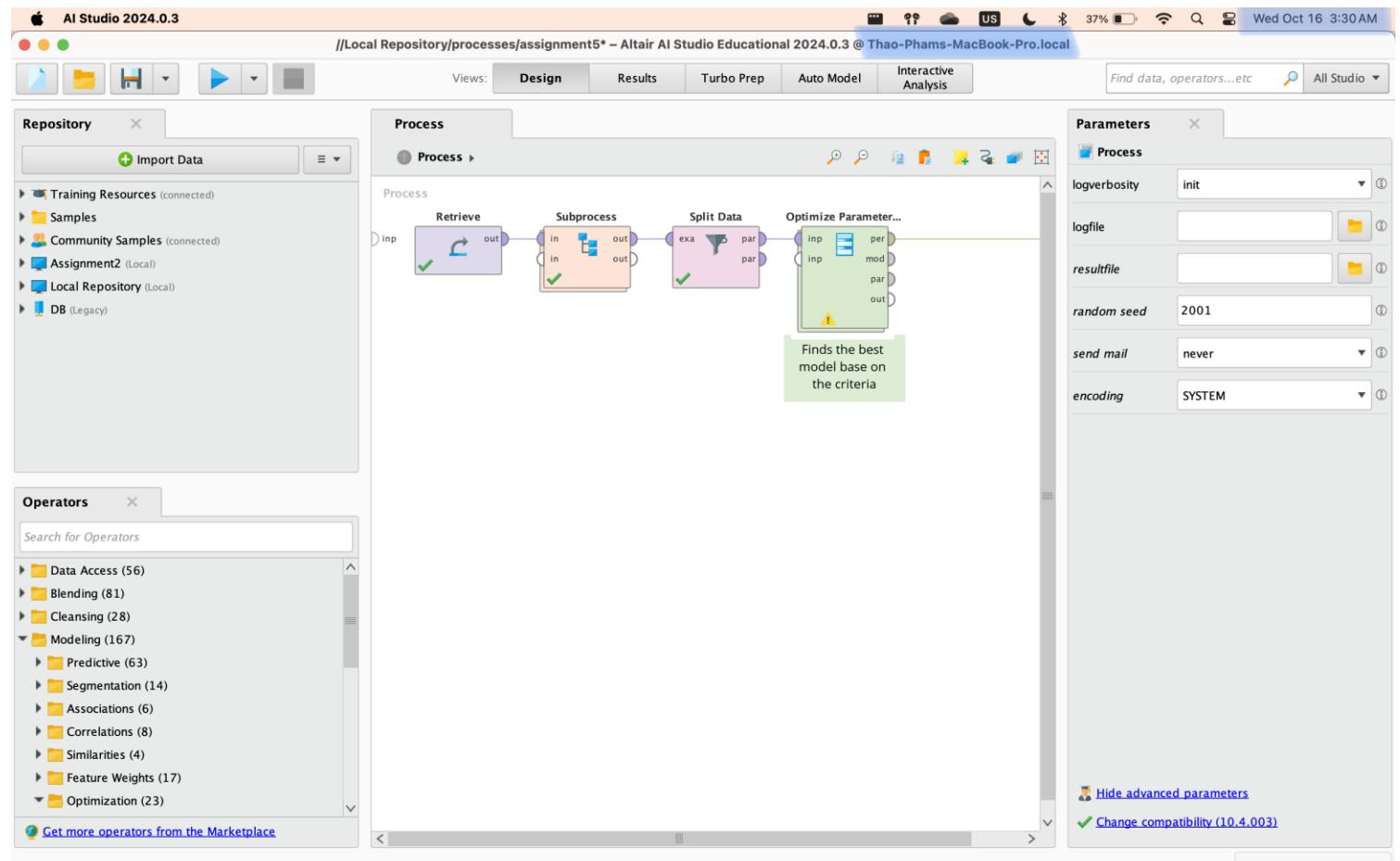
Selecting Design to go back to the Process

The screenshot shows the AI Studio 2024.0.3 interface with the title bar "AI Studio 2024.0.3 //Local Repository/processes/assignment5* – Altair AI Studio Educational 2024.0.3 @ Thao-Phams-MacBook-Pro.local" and the date "Wed Oct 16 3:18 AM". The top menu bar includes "Views: Design, Results, Turbo Prep, Auto Model, Interactive Analysis", "Find data, operators...etc", and "All Studio". The main area is titled "Repository" and shows the "Result History" tab selected. A sub-tab "ExampleSet (Split Data)" is open. Below it, there are buttons to "Open in" "Turbo Prep", "Auto Model", and "Interactive Analysis". A filter bar indicates "Filter (120 / 120 examples): all". On the left, there are four categories: "Data" (selected), "Statistics", "Visualizations", and "Annotations". The main content area is a table titled "ExampleSet (120 examples, 1 special attribute, 4 regular attributes)". The table has columns: Row No., Species, SepalLength..., SepalWidth..., PetalLength..., and PetalWidth... . The data consists of 120 rows of Iris-setosa measurements. The first few rows are:

Row No.	Species	SepalLength...	SepalWidth...	PetalLength...	PetalWidth...
1	Iris-setosa	5.100	3.500	1.400	0.200
2	Iris-setosa	4.900	3	1.400	0.200
3	Iris-setosa	4.700	3.200	1.300	0.200
4	Iris-setosa	4.600	3.100	1.500	0.200
5	Iris-setosa	5	3.600	1.400	0.200
6	Iris-setosa	5.400	3.900	1.700	0.400
7	Iris-setosa	4.600	3.400	1.400	0.300
8	Iris-setosa	4.400	2.900	1.400	0.200
9	Iris-setosa	4.900	3.100	1.500	0.100
10	Iris-setosa	4.800	3.400	1.600	0.200
11	Iris-setosa	4.300	3	1.100	0.100
12	Iris-setosa	5.800	4	1.200	0.200
13	Iris-setosa	5.700	4.400	1.500	0.400
14	Iris-setosa	5.400	3.900	1.300	0.400
15	Iris-setosa	5.100	3.500	1.400	0.300
16	Iris-setosa	5.700	3.800	1.700	0.300
17	Iris-setosa	5.100	3.800	1.500	0.300
18	Iris-setosa	5.100	3.700	1.500	0.400
19	Iris-setosa	4.600	3.600	1	0.200

Optimize Parameters (Grid) Operator

1. Operators tab, type **Optimize Parameters (Grid)** then drag and drop the operator to the Process.
2. Connection:
Split Data.partition 1 - Optimize Parameters (Grid).input 1
Optimize Parameters (Grid).performance –
Process.result1
3. Double click on the **Optimize Parameters (Grid)** operator to get inside it.



Cross Validation Operator

1. Operators tab, type **Cross Validation** then drag and drop the operator to the **Optimize Parameters (Grid)**.
2. Connections:

Optimize Parameters (Grid).input 1 - Cross Validation.example set

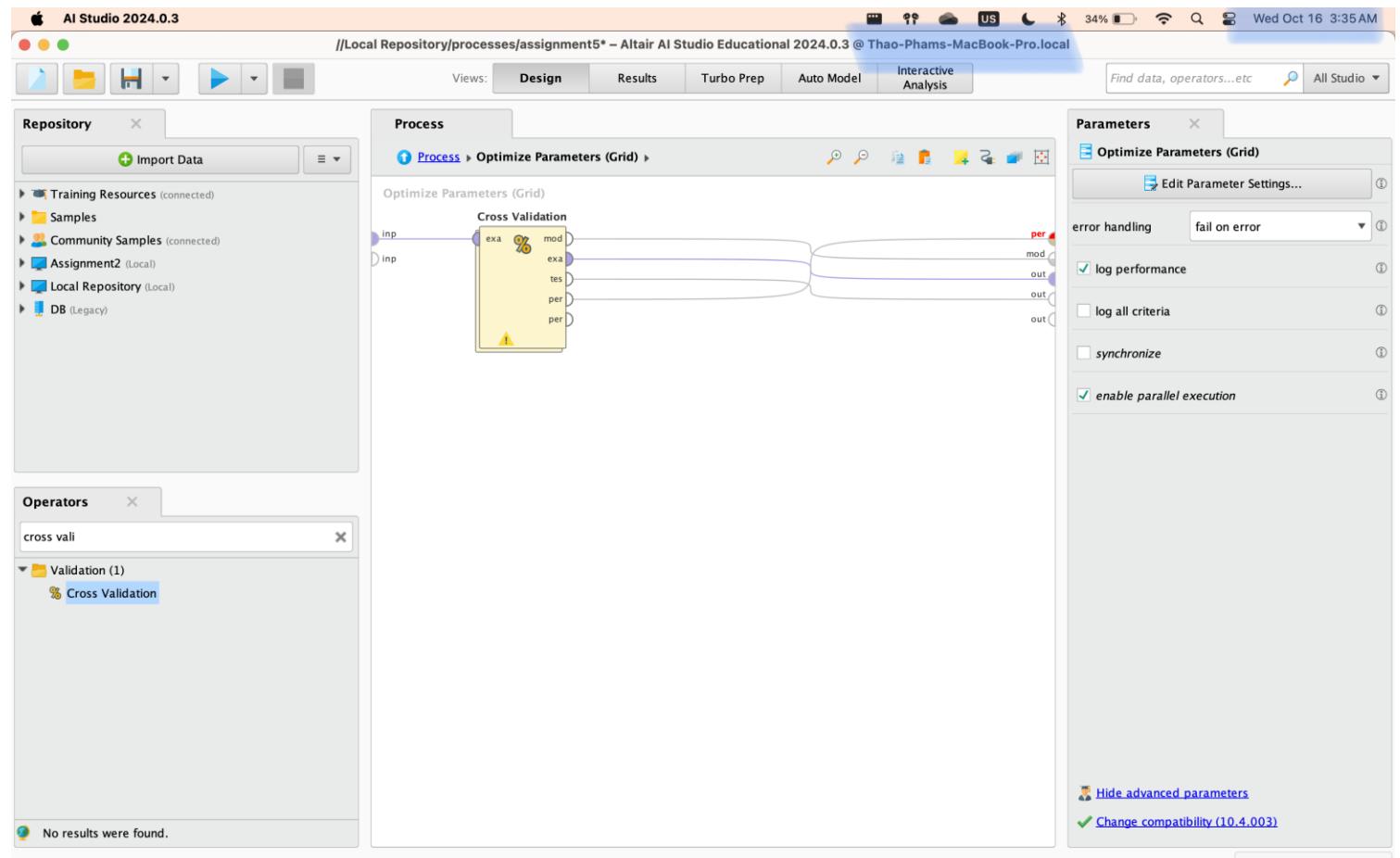
Cross Validation.model - Optimize Parameters (Grid).model

Cross Validation.example set - Optimize Parameters (Grid).output 1

Cross Validation.test result set - Optimize Parameters (Grid).output 2

Cross Validation.performance 1 - Optimize Parameters (Grid).performance

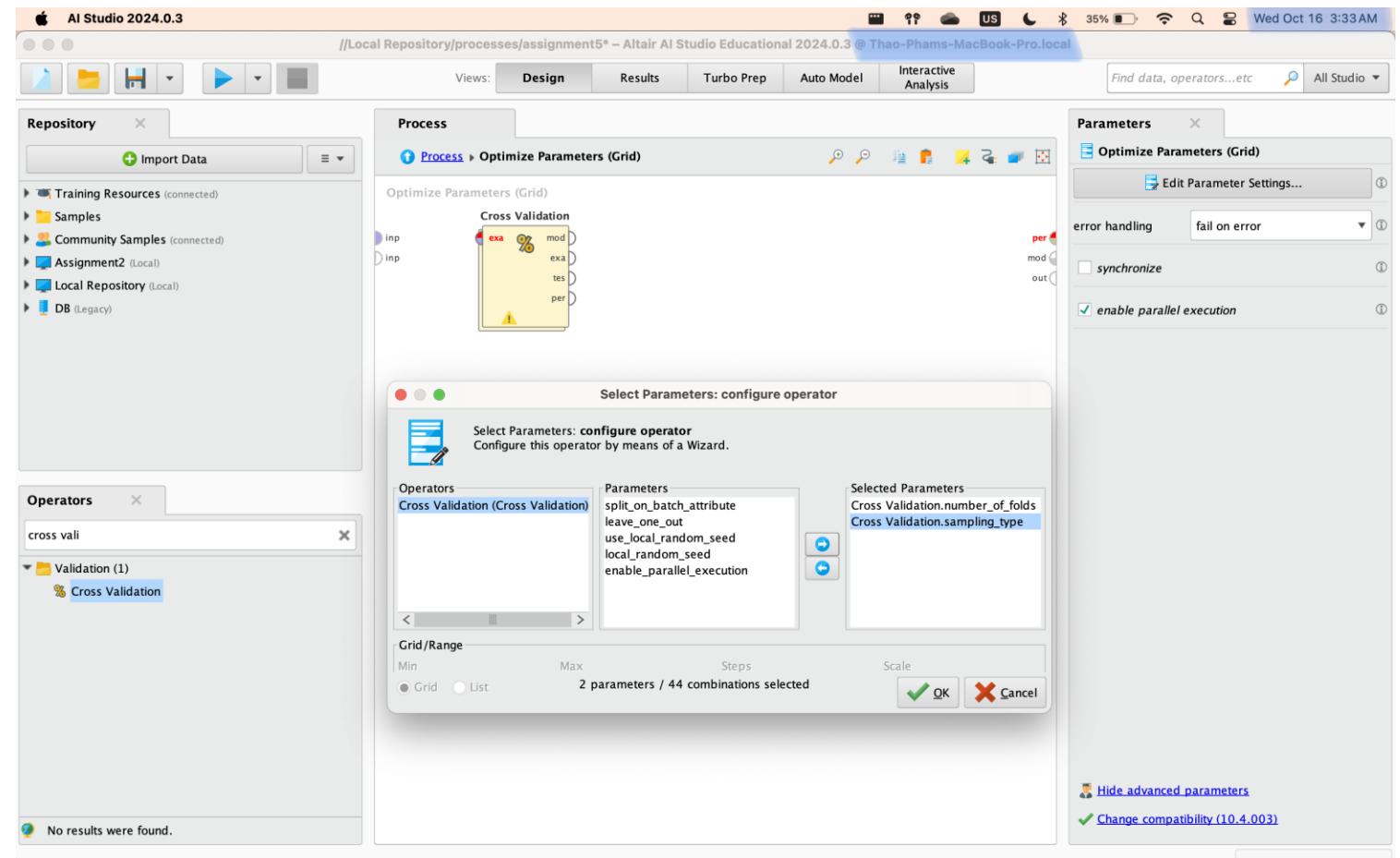
Click on the white space of the **Optimize Parameters (Grid)**.



Optimize Parameters (Grid) Operator cont.

1. Optimize Parameters (Grid)'s Parameters tab, Edit Parameters Setting and selecting:
 - Cross Validation.number_of_folds
 - Cross Validation.sampling_type
 - Click OK
2. **error handling** : fail on error
3. Selecting **enable parallel execution**

Double click on **Cross Validation** to get inside it

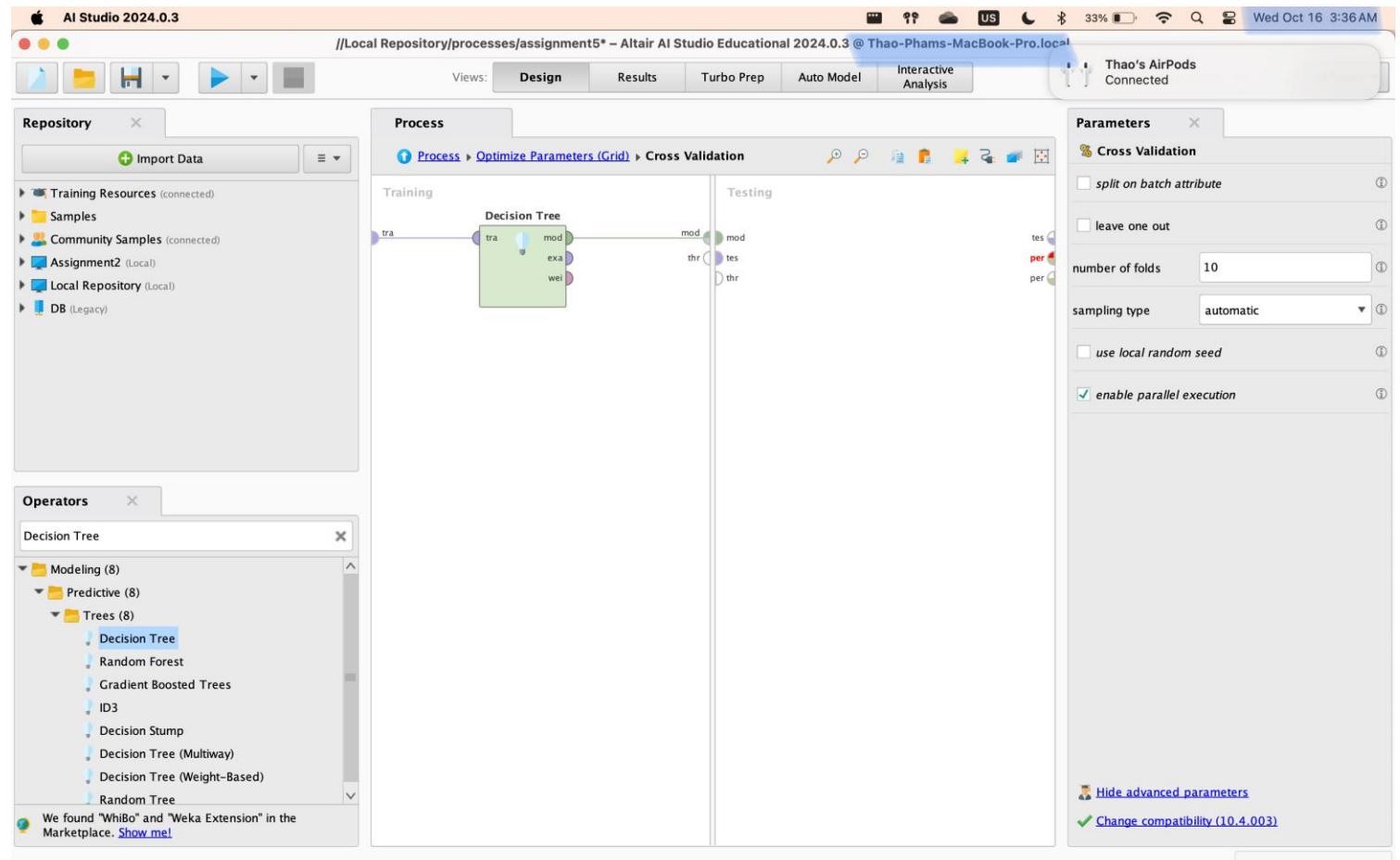


Decision Tree Operator

1. Operators tab, type **Decision Tree** then drag and drop the operator to the **training** subprocess of the **Cross Validation**.
2. Connections:

Cross Validation.training set - Decision Tree.training set

Decision Tree.model – Cross Validation.model



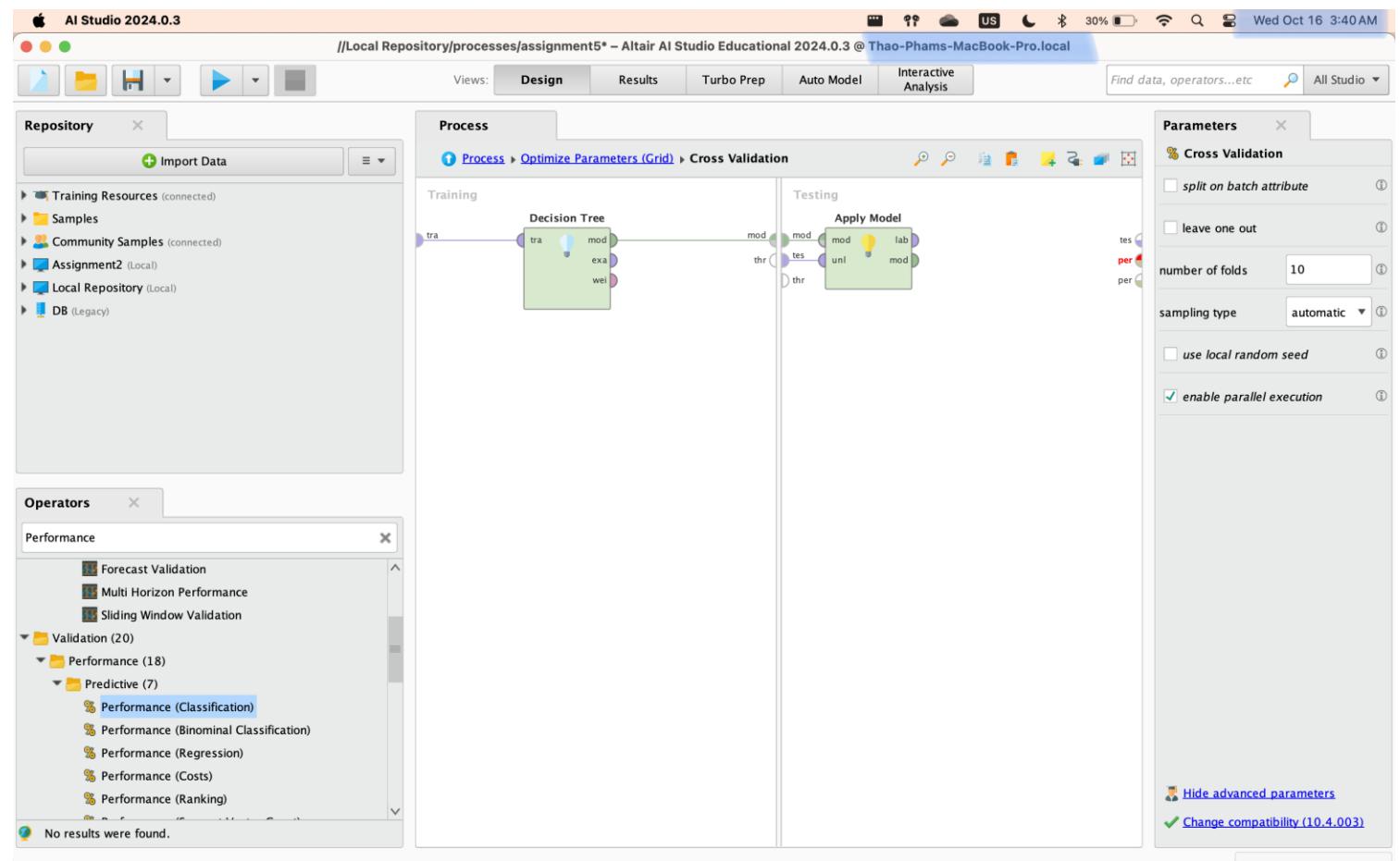
Apply Model Operator – Training

1. Operators tab, type **Apply Model** then drag and drop the operator to the **testing** subprocess of the **Cross Validation**.
2. Renaming the operator by double click on its name and change to **Apply Model Training**

3. Connections:

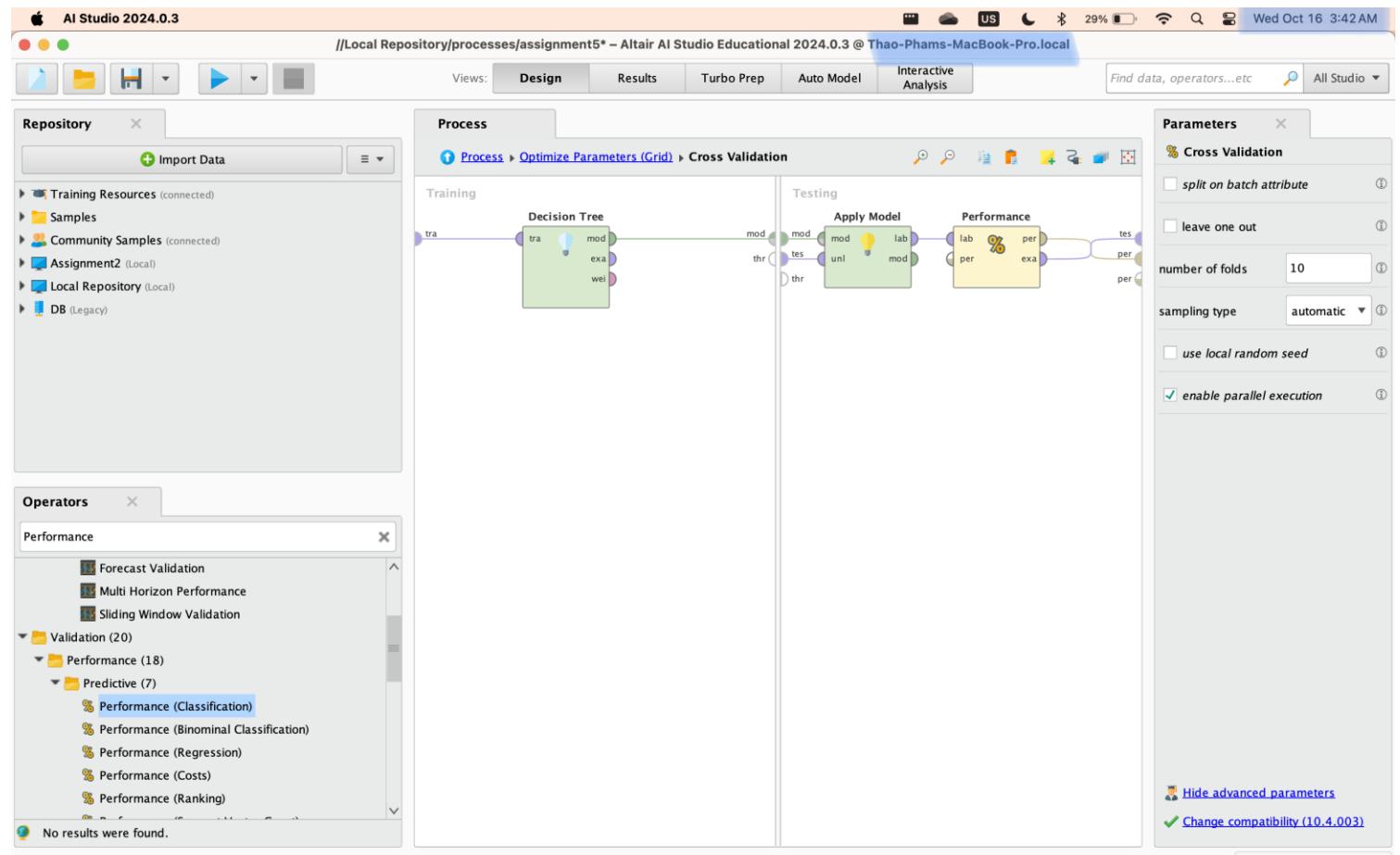
Cross Validation.model – **Apply Model Tranining.model**

Cross Validation.test set – **Apply Model Training.unlabelled data**



Performance (Classification) Operator – Training

1. Operators tab, type **Performance (Classification)** then drag and drop the operator to the **testing** subprocess of the **Cross Validation**.
2. Renaming the operator by double click on its name and change to **Performance Training**
3. Parameters tab, selecting:
 - **accuracy**
 - **classification error**
 - **skip undefined labels**
 - **use example weights**
4. Connections:
 - Apply Model Training.labelled data – Performance Training.labelled data
 - Performance Training.performance - Cross Validation.performance 1
 - Performance Training.example set - Cross Validation.test set results



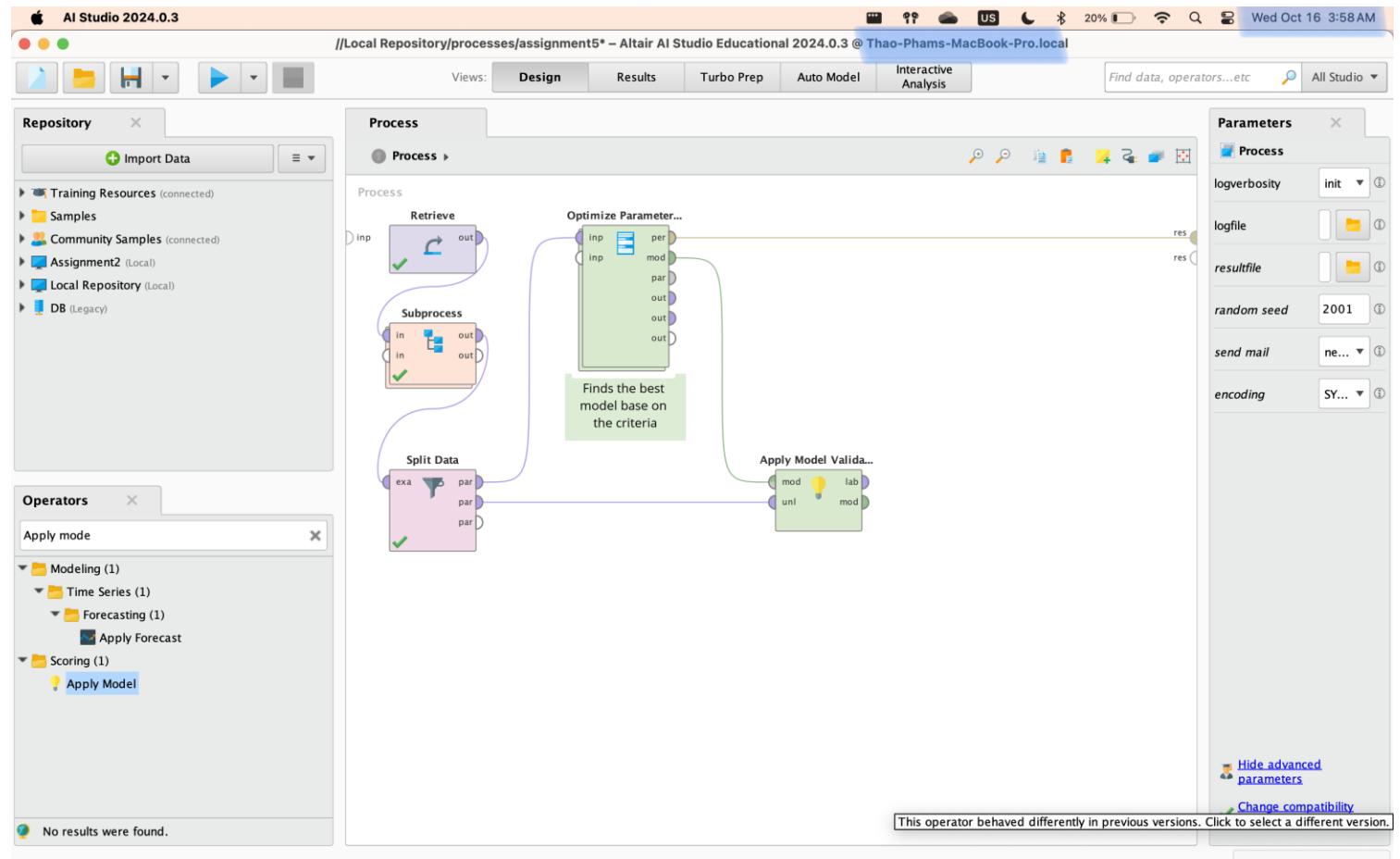
Click back to the Process.

Summary for Optimize Parameters (Grid)

- **Optimize Parameters (Grid)** operator: used to find the best combination of hyperparameters for a model by searching through a predefined set of parameter values (a "grid")
 - Cross Validation:
 - **Cross Validation.number_of_folds**: the data is divided into several equal-sized "folds." The model is trained on some of these folds and tested on the remaining one, repeating the process for each fold. This helps assess the model's performance across different subsets of data.
 - **Cross Validation.sampling_type**: This defines how the data is split during cross-validation, such as randomly shuffling the data or using stratified sampling to maintain class distribution.
 - **Training subprocess**: Within each fold of the cross-validation, a decision tree model is trained using the training data.
 - **Testing subprocess**: The trained model is then applied to the test data, and its performance is evaluated based on metrics like accuracy and classification error.

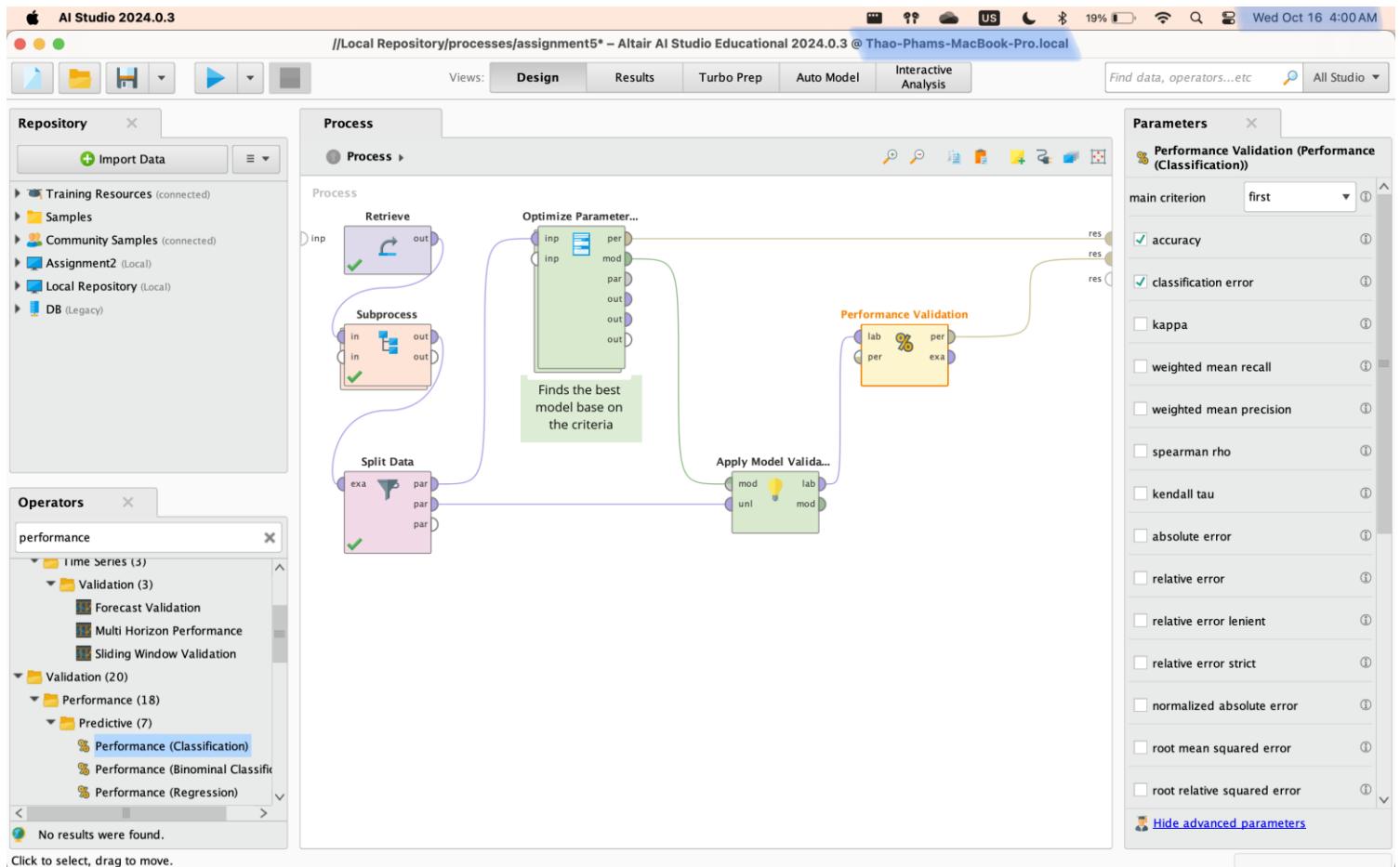
Apply Model Operator - Validation

1. Operators tab, type **Apply Model** then drag and drop the operator to the Process.
2. Renaming the operator by double click on its name and change to **Apply Model Validation**
3. Connections:
 - **Optimize Parameters (Grid).model** - **Apply Model Validation.model**
 - **Split Data.partition 2** - **Apply Model Validation.unlabelled data**



Performance (Classification) Operator – Validation

1. Operators tab, type **Performance (Classification)** then drag and drop the operator to the Process.
2. Renaming the operator by double click on its name and change to **Performance Validation**
3. Parameters tab, selecting:
 - **accuracy**
 - **classification error**
 - **skip undefined labels**
 - **use example weights**
4. Connections:
 - **Apply Model Validation.labelled data** – **Performance Validation.labelled data**
 - **Performance Validation.performance** – **Process.result 2**

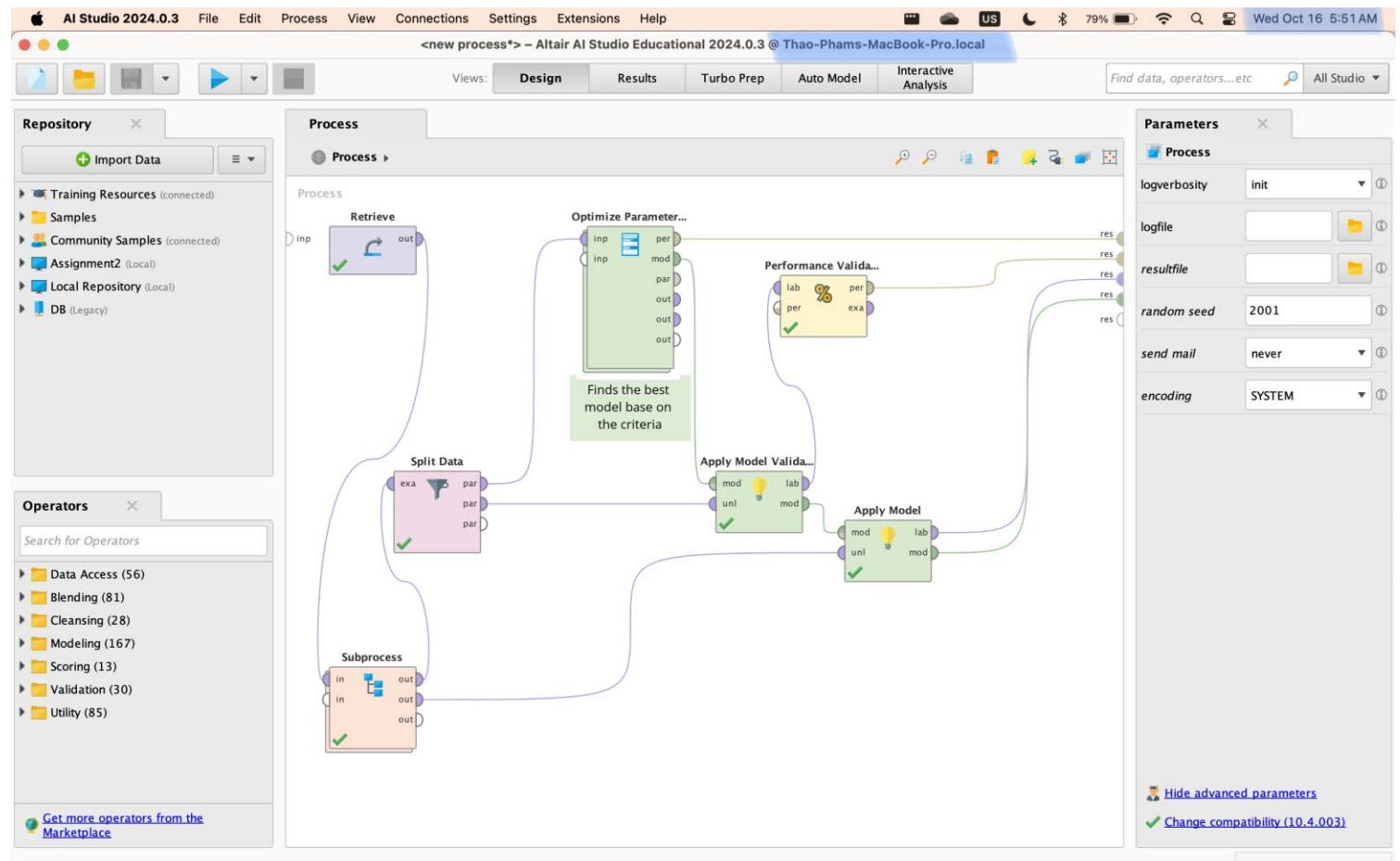


Apply Model Operator

1. Operators tab, type **Apply Model** then drag and drop the operator to the Process.

2. Connections:

- Apply Model Validation.model - Apply Model.model
- Subprocess.out 2 – Apply Model.Unlabelled data
- Apply Model.labelled data – Process.result 3
- Apply Model.model - Process.result 4

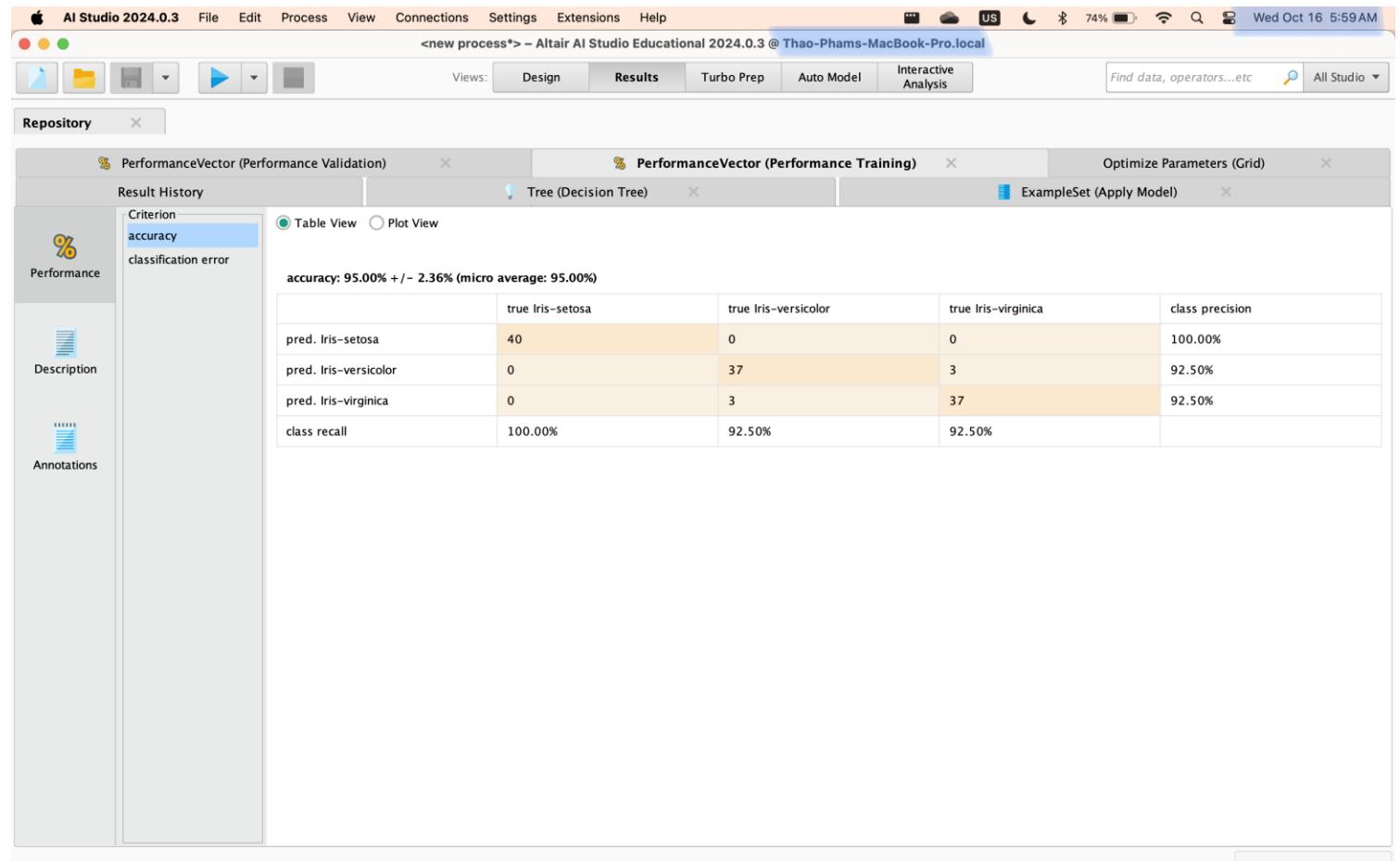


Summary Validation

- **Apply Model Validation:** This operator uses the 20% split data (validation set) from the previous step to validate the trained model (from the "Optimize Parameters (Grid)" operator). It applies the model to this validation set to generate predictions for evaluation.
- **Performance Validation:** This operator evaluates the model's performance using the predicted results from **Apply Model Validation** by comparing them to the true labels. It provides metrics such as accuracy, confusion matrix, classification error, etc.
- **Apply Model (Final Step):** This step is used to apply the trained decision tree model to a new dataset where the target variable (Species) is missing. This process generates predictions, which can be used to visualize or analyze the decision tree structure.

Interpretation – Performance Training

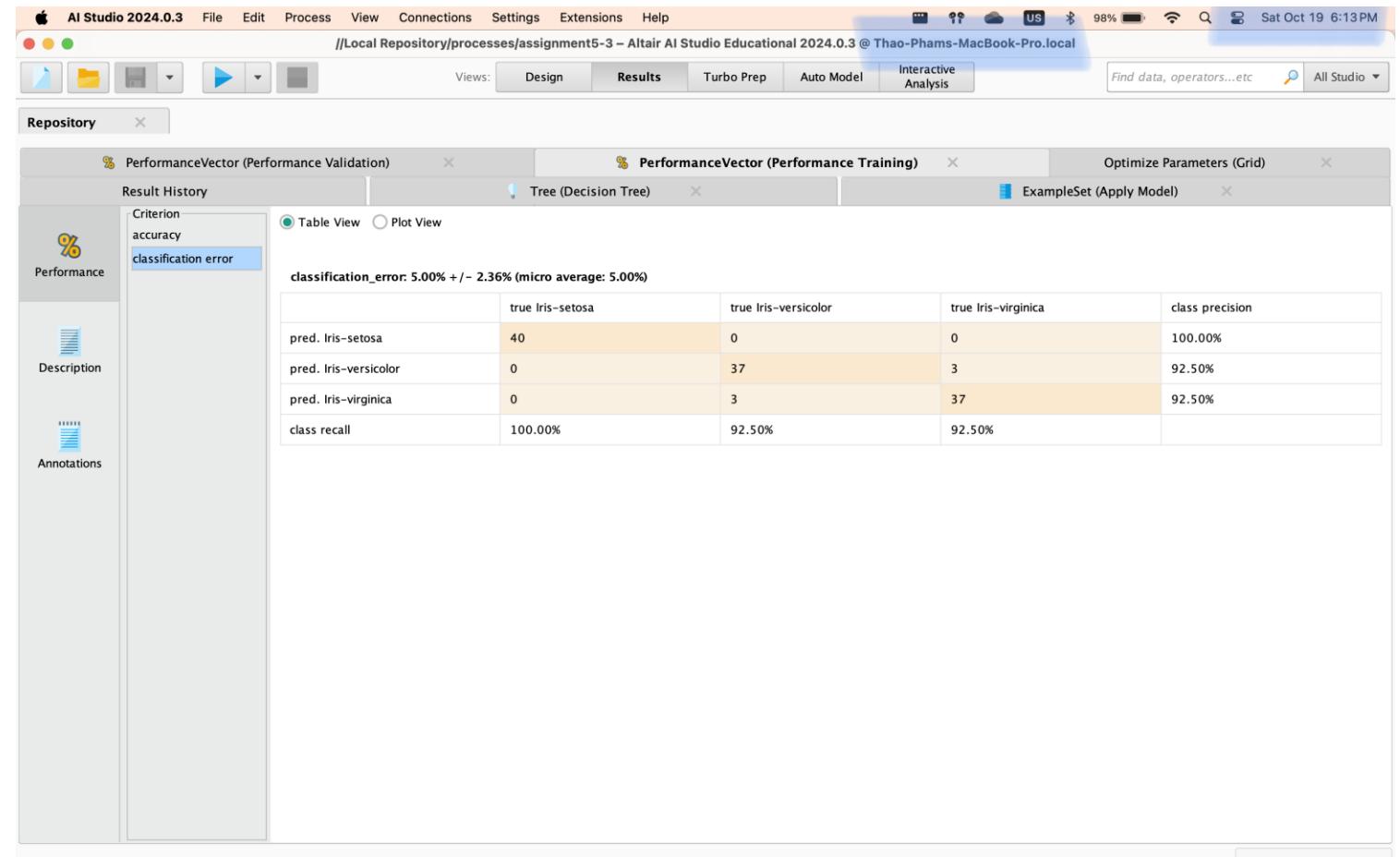
- **Accuracy:** 95.00% indicates the model performs consistently well on the testing set.
- **Confusion Matrix:**
 - **Iris-setosa:** All 40 instances were correctly classified.
 - **Iris-versicolor:** 37 out of 40 instances were correctly classified, with 3 misclassified as "Iris-virginica".
 - **Iris-virginica:** 37 out of 40 instances were correctly classified, with 3 misclassified as "Iris-versicolor".
- **Accuracy Variability:** The variation of $\pm 2.36\%$ suggests that accuracy is relatively stable across different testing subsets.



Interpretation – Performance Training

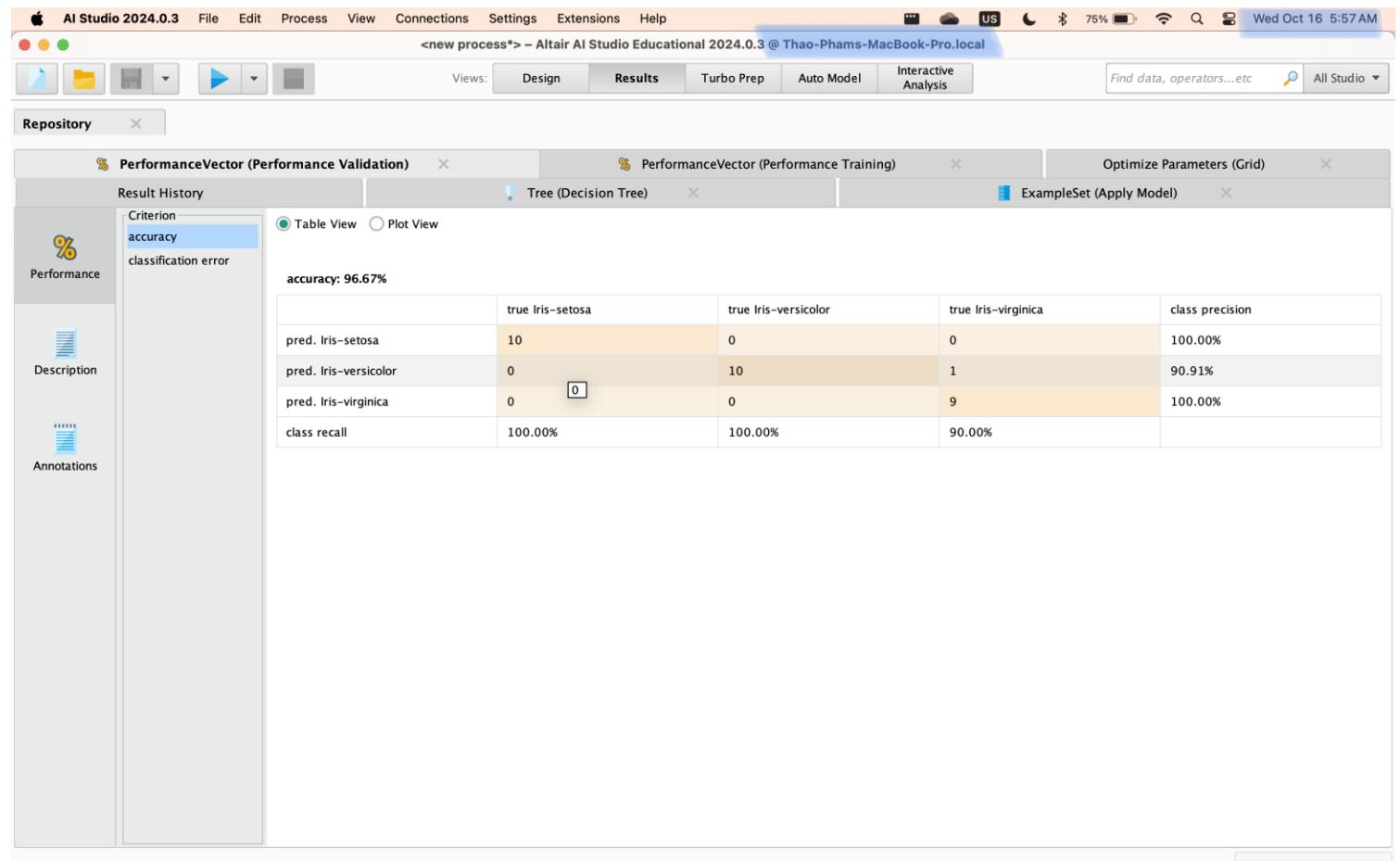
- **Classification Error:** 5% indicates the predictions on the training/testing set were 5% incorrect.

The model performed well during training but had some misclassification, likely due to the overlap between **Iris-versicolor** and **Iris-virginica**.



Interpretation – Performance Validation

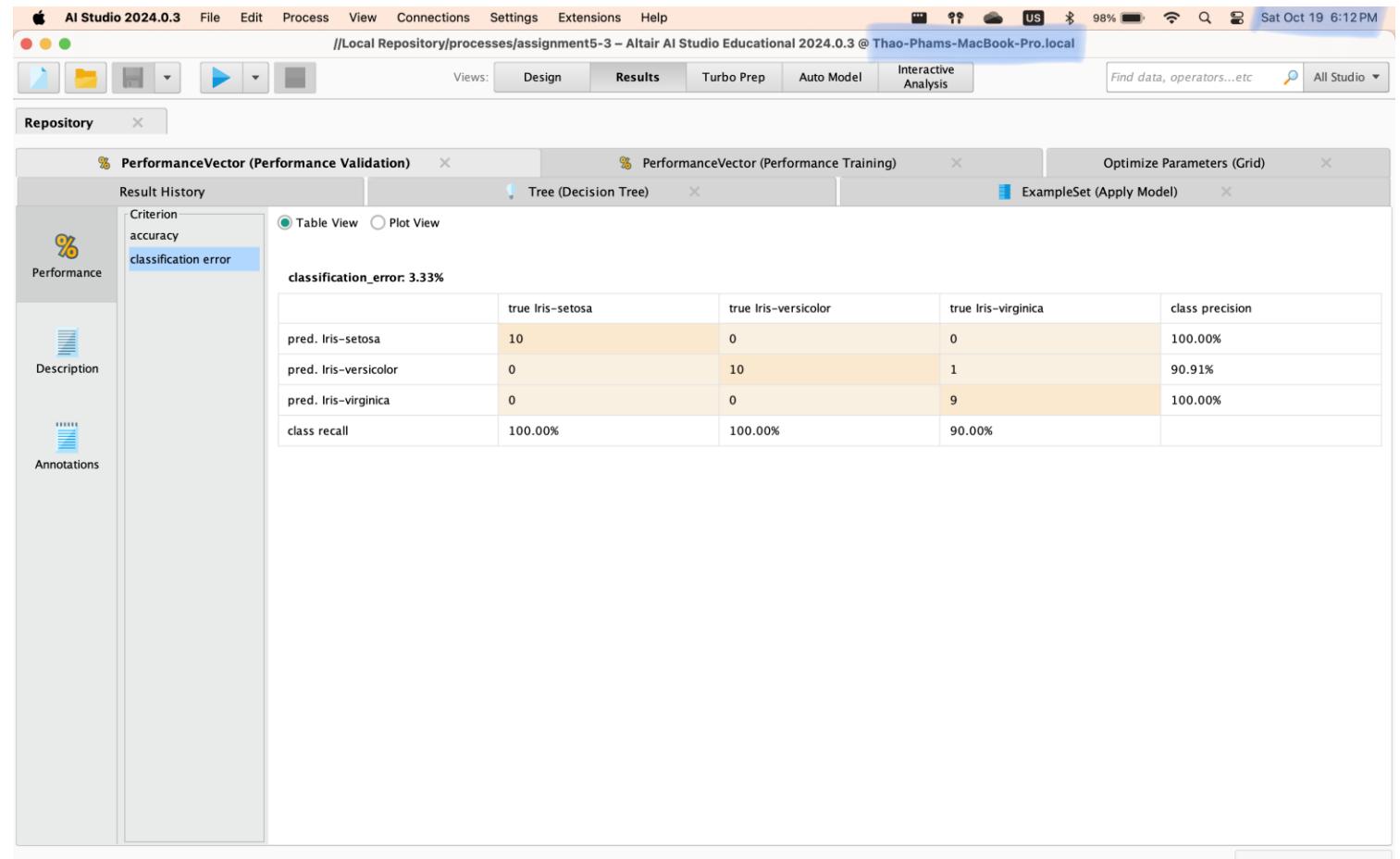
- **Accuracy:** 96.67% shows the model's high performance on the validation set, with a high proportion of correct predictions.
- **Confusion Matrix:**
 - **Iris-setosa:** All 10 instances were correctly classified, showing the model performs very well for this class.
 - **Iris-versicolor:** 10 out of 11 were correctly classified, while 1 was misclassified as "Iris-virginica".
 - **Iris-virginica:** All 9 instances were correctly classified.



Interpretation – Performance Validation

- **Classification error:** 3.33% indicates the predictions on the validation set were 3.33% incorrect.

A lower error rate on the validation set indicates that the model generalized well to unseen data, making fewer mistakes on new samples. The small difference between training and validation errors suggests the model is not overfitting and performs consistently across different data splits.

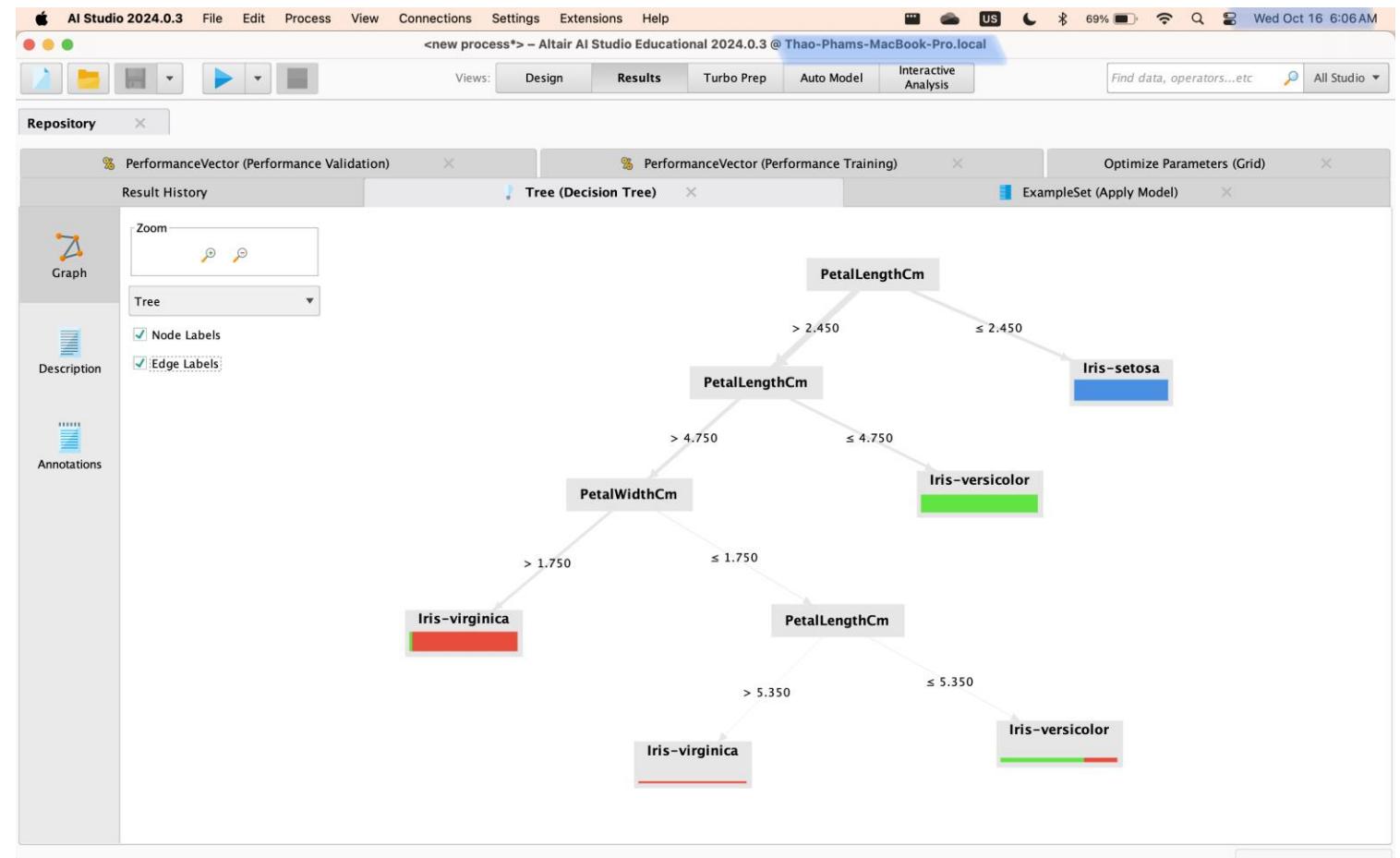


Interpretation - Decision Tree Graph

The decision tree separates the Iris species based on petal length and width:

1. **Iris-setosa:** Classified when $\text{PetalLengthCm} \leq 2.45$ cm, indicating shorter petals.
2. **Iris-versicolor:** Predicted for $2.45 < \text{PetalLengthCm} \leq 4.75$ cm or for smaller petal widths when $\text{PetalLengthCm} > 4.75$ cm.
3. **Iris-virginica:** Identified with wider petals ($\text{PetalWidthCm} > 1.75$ cm) when $\text{PetalLengthCm} > 4.75$ cm, or with longer petals ($\text{PetalLengthCm} > 5.35$ cm).

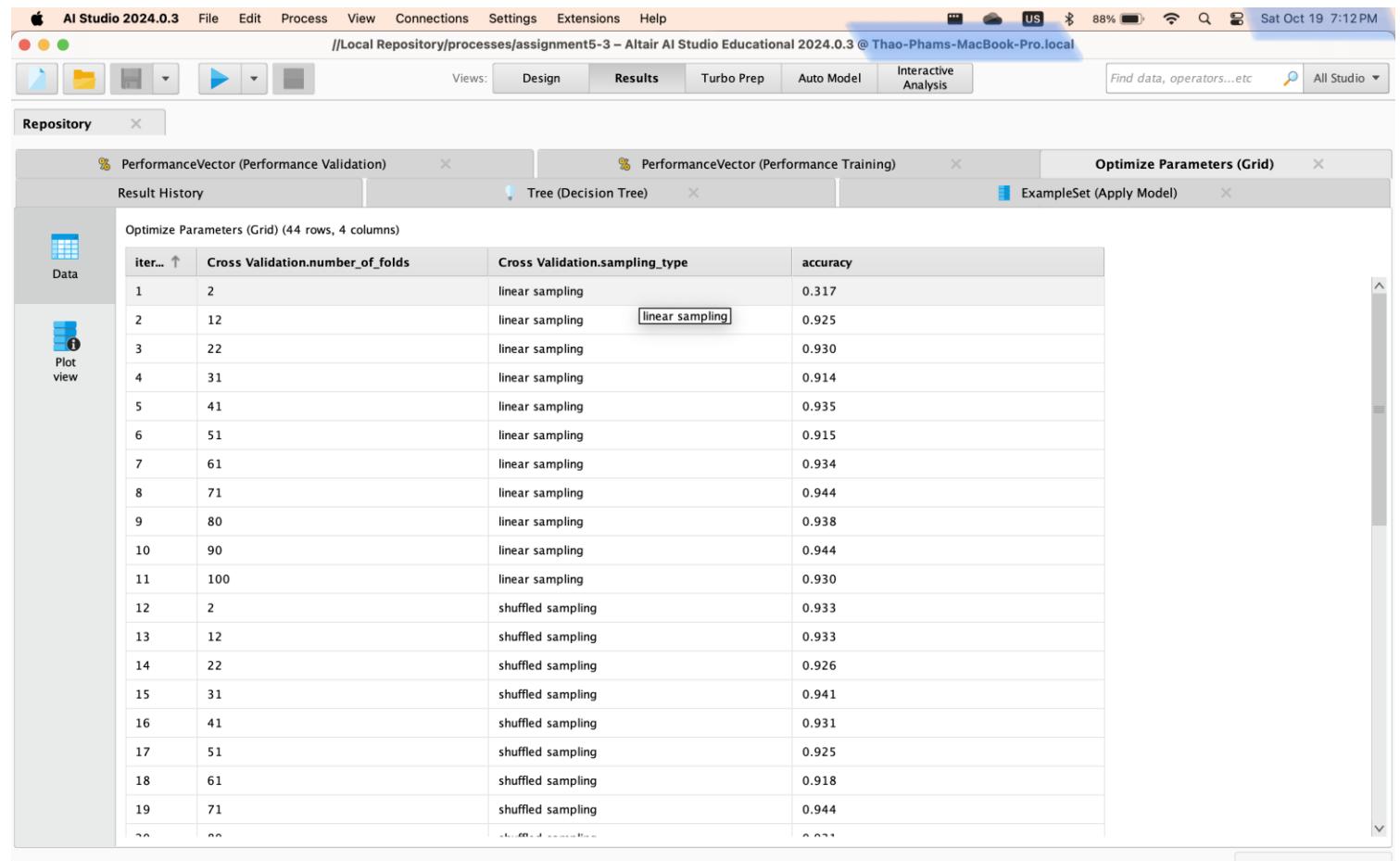
This tree effectively distinguishes Iris-setosa but shows some overlap between Iris-versicolor and Iris-virginica.



Optimize Parameter (Grid) Table

The table aims to identify the best combination of cross-validation settings to maximize model accuracy during the grid search.

- **Iteration:** Represents the number of times different parameter combinations were tested during the grid search.
- **Cross Validation.number_of_folds:** This represents the number of folds used for cross-validation in each iteration. Higher numbers indicate more folds, leading to smaller test sets in each fold.
- **Cross Validation.sampling_type:** Two sampling types are used: linear and shuffled. Linear sampling keeps the original order of the data, while shuffled sampling randomly shuffles the data before splitting, which can help improve model generalization by reducing bias.
- **Accuracy:** The accuracy column shows the performance of the model for each configuration. Generally, higher accuracy indicates better performance for that particular combination of number of folds and sampling type.



The screenshot shows the Altair AI Studio interface with the 'Optimize Parameters (Grid)' tab selected. The main area displays a data table titled 'Optimize Parameters (Grid) (44 rows, 4 columns)'. The columns are labeled 'iter...', 'Cross Validation.number_of_folds', 'Cross Validation.sampling_type', and 'accuracy'. The data shows various iterations with different fold counts (2, 12, 22, 31, 41, 51, 61, 71, 80, 90, 100) and sampling types (linear and shuffled). Accuracy values range from 0.918 to 0.937. The 'Plot view' icon on the left indicates there is a corresponding plot available for this data.

iter...	Cross Validation.number_of_folds	Cross Validation.sampling_type	accuracy
1	2	linear sampling	0.317
2	12	linear sampling	0.925
3	22	linear sampling	0.930
4	31	linear sampling	0.914
5	41	linear sampling	0.935
6	51	linear sampling	0.915
7	61	linear sampling	0.934
8	71	linear sampling	0.944
9	80	linear sampling	0.938
10	90	linear sampling	0.944
11	100	linear sampling	0.930
12	2	shuffled sampling	0.933
13	12	shuffled sampling	0.933
14	22	shuffled sampling	0.926
15	31	shuffled sampling	0.941
16	41	shuffled sampling	0.931
17	51	shuffled sampling	0.925
18	61	shuffled sampling	0.918
19	71	shuffled sampling	0.944
20	80	shuffled sampling	0.931

Conclusion

- The model achieved high accuracy, with 95.00% on the training/testing set and 96.67% on the validation set, demonstrating effective generalization.
- The tree structure successfully distinguished Iris-setosa and showed good performance in separating Iris-versicolor and Iris-virginica, despite some overlap.
- Overall, the results indicate that the decision tree is a reliable model for classifying the Iris dataset.

Research Items - Operators

- Subprocess: Introducing a process within a process for better manage when the process increase in size (GmbH, n.d.-k).
 - Input can have multiple inputs. When one input is connected, another input port becomes available which is ready to accept another input (if any). The order of inputs remains the same (GmbH, n.d.-k).
 - Output can have multiple outputs. When one output is connected, another output port becomes available which is ready to deliver another output (if any). The order of inputs remains the same (GmbH, n.d.-k).
- Rename: Changing name of one or more attributes of the input Examples (GmbH, n.d.-f).
 - The *Rename* operator has no impact on the type or role of an attribute (GmbH, n.d.-f).
 - Input: examples set and dictionary (data table) as optional (GmbH, n.d.-f).
 - Output: The ExampleSet with renamed attributes and original (GmbH, n.d.-f).
 - Parameters: **rename attributes** allows selecting attributes and assign new names to them (GmbH, n.d.-f).

Research Items - Operators

Cross Validation: estimating the statistical performance of a learning model. Cross Validation is a nested operator. It has two subprocesses: a Training subprocess and a Testing subprocess. The Training subprocess is used for training a model. The trained model is then applied in the Testing subprocess. The performance of the model is measured during the Testing phase (GmbH, n.d.-b).

- Input: get example set to apply the cross validation (GmbH, n.d.-b).
- Output:
 - **model** delivers the prediction model trained on the whole example set (GmbH, n.d.-b).
 - **example** returns the same example set which as been given as input. (GmbH, n.d.-b).
 - **test** delivers only an Example Set if the test set results port of the inner Testing subprocess is connected. (GmbH, n.d.-b).
 - **performance** an expandable port (GmbH, n.d.-b).

Parameters:

- **number of folds** This parameter specifies the number of folds (number of subsets) the example set should be divided into. Each subset has equal number of Examples. Also, the number of iterations that will take place is the same as the **number of folds**. If the model output port is connected, the Training subprocess is repeated one more time with all Examples to build the final model (GmbH, n.d.-b).
- **sampling type** The Cross Validation Operator can use several types of sampling for building the subsets. Following options are available: linear sampling, shuffled sampling, stratified sampling and automatic (GmbH, n.d.-b).

Research Items - Operators

- Decision Tree: generating a decision tree model, which can be used for classification and regression (GmbH, n.d.-b).
 - Input: **Training set** is the input data which is used to generate the decision tree model (GmbH, n.d.-b).
 - Output: **model** (Decision Tree): The decision tree model is delivered from this output port (GmbH, n.d.-b).
- Filter Examples: Selecting which examples to include or exclude (GmbH, n.d.-c).
 - Input: examples set (GmbH, n.d.-c).
 - Output: filtered example set and original (GmbH, n.d.-c).
 - Parameters:
 - **Filters** allows the definition of a custom filter condition. A condition consists of an Attribute, and comparison function and a value to match (GmbH, n.d.-c).
 - **condition_class : custom_filters** is the same as the default filters parameter (GmbH, n.d.-c).

Research Items - Operators

Optimize Parameters (Grid): finding the optimal values of the selected parameters for the Operators in its subprocess (GmbH, n.d.-d).

- Input can have multiple inputs. When one input is connected, another input port becomes available which is ready to accept another input (if any). The order of inputs remains the same.
- Output:
 - **performance** delivers the Performance Vector for the optimal values of the selected parameters. A Performance Vector is a list of performance criteria values (GmbH, n.d.-d).
 - **model** delivers the Model for the optimal values of the selected parameters (GmbH, n.d.-d).
- Parameter: **error handling** allows selecting the method for handing error occurring during the execution of the inner process (GmbH, n.d.-d).
 - ***fail on error***: In case an error occurs, the execution of the process will fail with an error message (GmbH, n.d.-d).

Research Items - Operators

Performance (Classification): finding statistical performance evaluation of classification tasks. This operator delivers a list of performance criteria values of the classification task (GmbH, n.d.-e).

- Input: **labelled data** (GmbH, n.d.-e).
- Output:
 - **performance** port delivered a Performance Vector which is a list of performance criteria values (GmbH, n.d.-e).
 - **example set** that was given as input is passed without changing to the output through this port (GmbH, n.d.-e).
- Parameters:
 - **Accuracy**: Relative number of correctly classified examples or in other words percentage of correct predictions (GmbH, n.d.-e).
 - **classification error**: Relative number of misclassified examples or in other words percentage of incorrect predictions (GmbH, n.d.-e).

References

GmbH, R. (n.d.-a). *Apply Model - RapidMiner Documentation*. https://docs.rapidminer.com/10.2/studio/operators/scoring/apply_model.html

GmbH, R. (n.d.-b). *Cross Validation - Altair RapidMiner Documentation*. https://docs.rapidminer.com/2024.1/studio/operators/validation/cross_validation.html

GmbH, R. (n.d.-c). *Decision Tree - Altair RapidMiner Documentation*.

https://docs.rapidminer.com/2024.1/studio/operators/modeling/predictive/trees/parallel_decision_tree.html

GmbH, R. (n.d.-d). *Filter Examples - Altair RapidMiner Documentation*.

https://docs.rapidminer.com/2024.0/studio/operators/blending/examples/filter/filter_examples.html

GmbH, R. (n.d.-e). *Optimize Parameters (Grid) - Altair RapidMiner Documentation*.

https://docs.rapidminer.com/2024.1/studio/operators/modeling/optimization/parameters/optimize_parameters_grid.html

GmbH, R. (n.d.-f). *Performance (Classification) - Altair RapidMiner Documentation*.

https://docs.rapidminer.com/latest/studio/operators/validation/performance/predictive/performance_classification.html

References

GmbH, R. (n.d.-g). *Rename - Altair RapidMiner Documentation*.

https://docs.rapidminer.com/latest/studio/operators/blending/attributes/names_and_roles/ rename.html

GmbH, R. (n.d.-h). *Retrieve - Altair RapidMiner Documentation*. https://docs.rapidminer.com/latest/studio/operators/data_access/retrieve.html

GmbH, R. (n.d.-i). *Select Attributes - Altair RapidMiner Documentation*.

https://docs.rapidminer.com/latest/studio/operators/blending/attributes/selection/select_attributes.html#:~:text=The%20first%20Select%20Attributes%20Operator,choosing%20binomial%20and%20non%2Dbinomial

GmbH, R. (n.d.-j). *Set Role - RapidMiner Documentation*. https://docs.rapidminer.com/9.1/studio/operators/blending/attributes/names_and_roles/set_role.html

GmbH, R. (n.d.-k). *Split Data - RapidMiner Documentation*. https://docs.rapidminer.com/9.6/studio/operators/blending/examples/sampling/split_data.html

GmbH, R. (n.d.-l). *Subprocess - RapidMiner Documentation*. <https://docs.rapidminer.com/9.3/studio/operators/utility/subprocess.html>



**Thank you
for reading**

Thao Pham