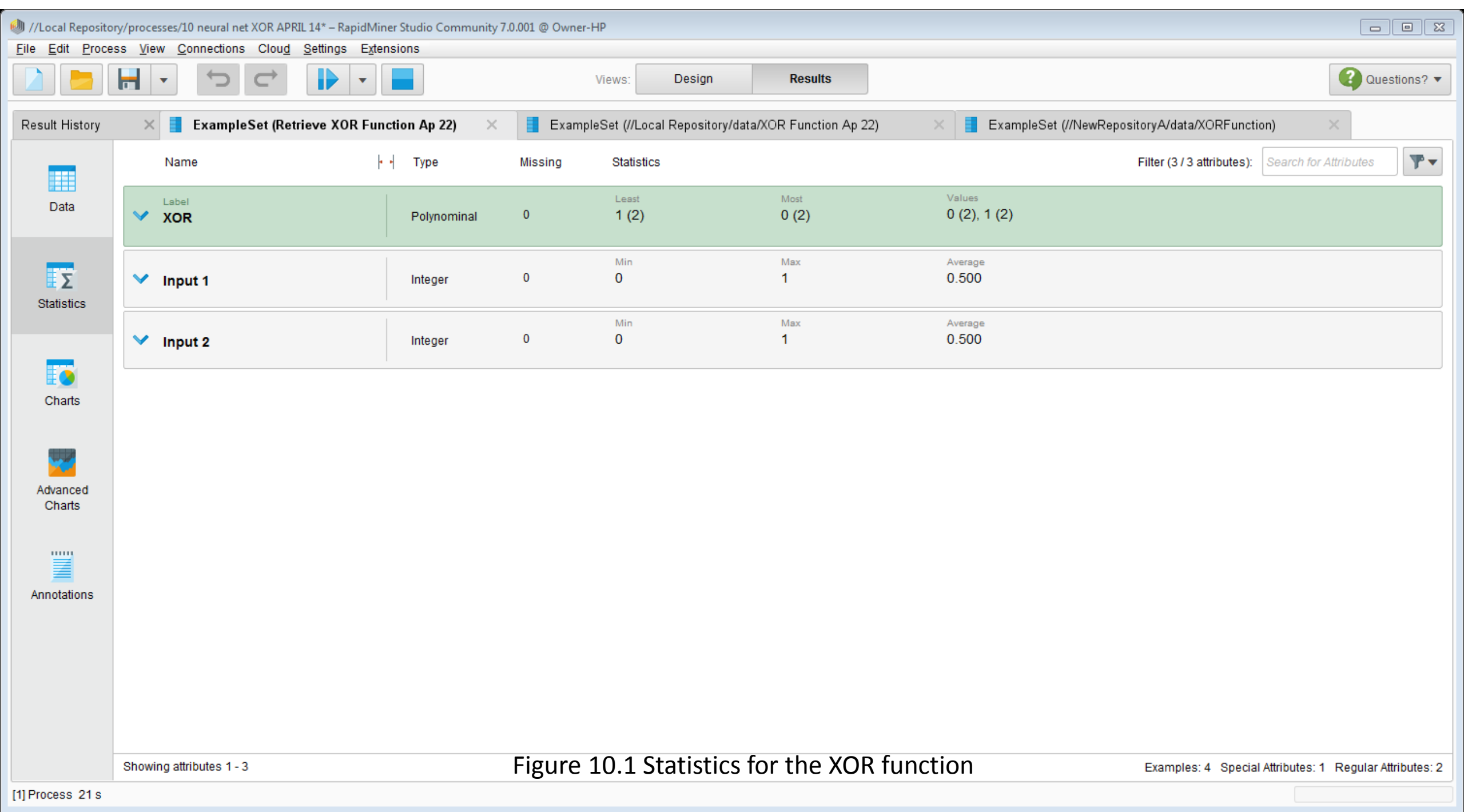


## Chapter 10

# Building Neural Networks with RapidMiner



Local Repository/processes/10 neural net XOR APRIL 14 – RapidMiner Studio Community 7.0.001 @ Owner-HP

File Edit Process View Connections Cloud Settings Extensions

Views: Design Results

Repository

- Add Data
- Samples
- DB
- Local Repository (Owner)
- LocalBookRepository (Owner)
- NewRepositoryA (Owner)
- NewRepositoryB (Owner)
- Repository for Chapter 5 (Owner)
- Cloud Repository (disconnected)

Operators

Search for Operators

- Data Access (46)
- Blending (77)
- Cleansing (26)
- Modeling (125)
- Scoring (10)
- Validation (30)
- Utility (85)
- Extensions (271)

+ Get More Operators

Process

Process

inp

Learning the XOR Function.

Retrieve XOR Functi...

Set Role

Multiply

Neural Net

Apply Model

Performance

Parameters

Neural Net

- hidden layers [Edit Lis...](#)
- training cycles 5000
- learning rate 0.3
- momentum 0.2
- ☐ decay
- ☒ shuffle
- ☒ normalize
- error epsilon 1.0E-5
- [Hide advanced parameters](#)

Help

Neural Net

RapidMiner Studio Core

Synopsis

This operator learns a model by means of a feed-forward neural network trained by a back propagation algorithm (multi-layer perceptron).

Figure 10.2 Main process for learning the XOR function

Edit Parameter List: hidden layers

Edit Parameter List: **hidden layers**  
Describes the name and the size of all hidden layers.

| hidden layer name | hidden layer sizes |
|-------------------|--------------------|
|                   | -1                 |

Add Entry Remove Entry Apply Cancel

Figure 10.3 Default setting for the *hidden layers* parameter

Edit Parameter List: hidden layers

Edit Parameter List: **hidden layers**  
Describes the name and the size of all hidden layers.

| hidden layer name | hidden layer sizes |
|-------------------|--------------------|
| one               | 5                  |

Add Entry Remove Entry Apply Cancel

Figure 10.4 A single hidden layer of five nodes

Local Repository/processes/10 neural net XOR APRIL 14\* - RapidMiner Studio Community 7.0.001 @ Owner-HP

File Edit Process View Connections Cloud Settings Extensions

Views: Design Results

Repository

- Add Data
- Samples
- DB
- Local Repository (Owner)
- LocalBookRepository (Owner)
- NewRepositoryA (Owner)
- NewRepositoryB (Owner)
- Repository for Chapter 5 (Owner)
- Cloud Repository (disconnected)

Operators

Search for Operators

- Data Access (46)
- Blending (77)
- Cleansing (26)
- Modeling (125)
- Scoring (10)
- Validation (30)
- Utility (85)
- Extensions (271)

+ Get More Operators

Process

Process

inp

Learning the XOR Function.

Retrieve XOR Functi...

Set Role

Multiply

Neural Net

Apply Model

Performance

Parameters

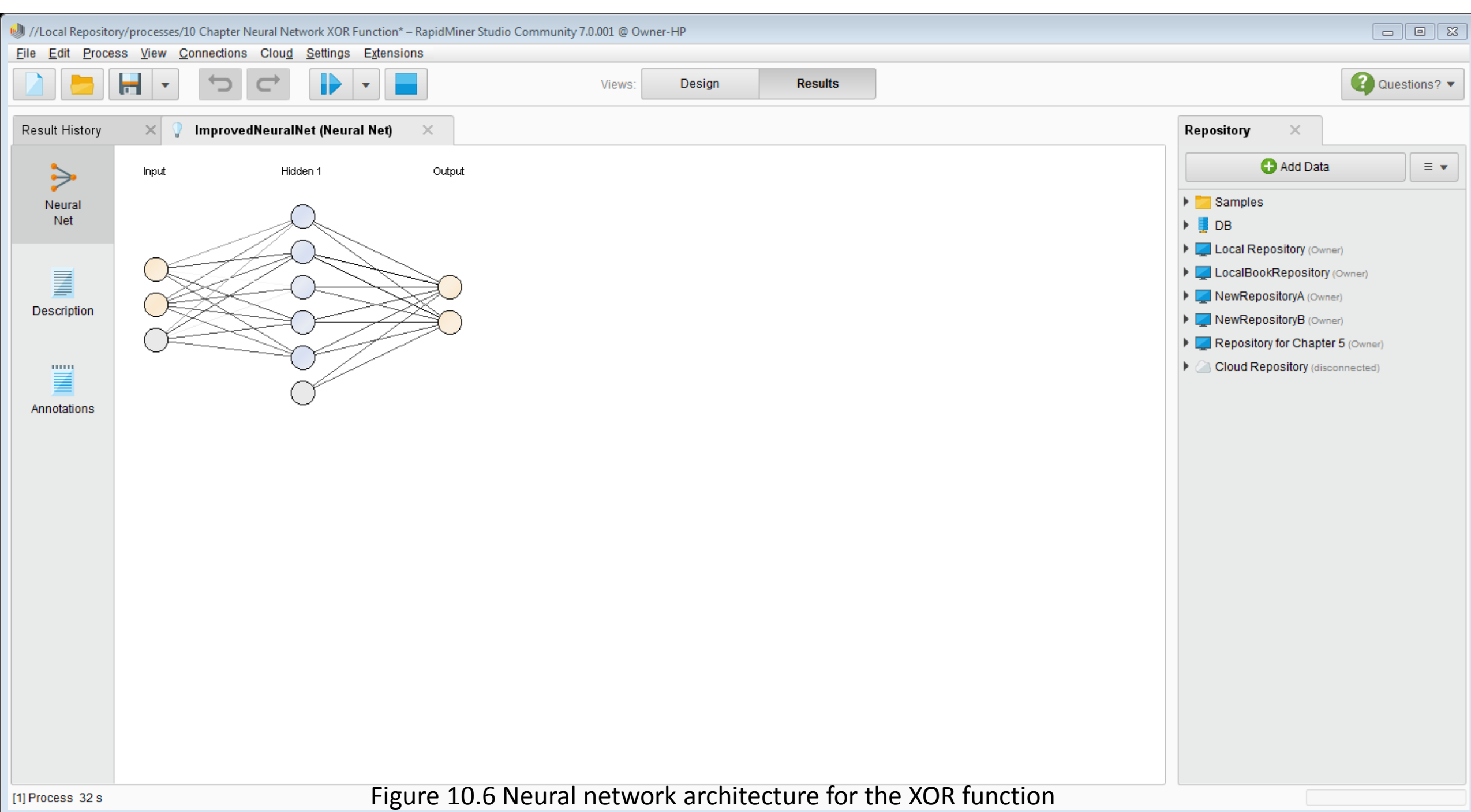
Performance (Performance...)

- ☒ accuracy
- ☐ classification error
- ☐ kappa
- ☐ weighted mean recall
- ☐ weighted mean precision
- ☐ spearman rho
- ☐ kendall tau
- ☒ absolute error
- ☐ relative error
- ☐ relative error lenient
- ☐ relative error strict
- ☐ normalized absolute error
- ☒ root mean squared error

[Hide advanced parameters](#)

Help

Figure 10.5 Performance parameters for neural network learning



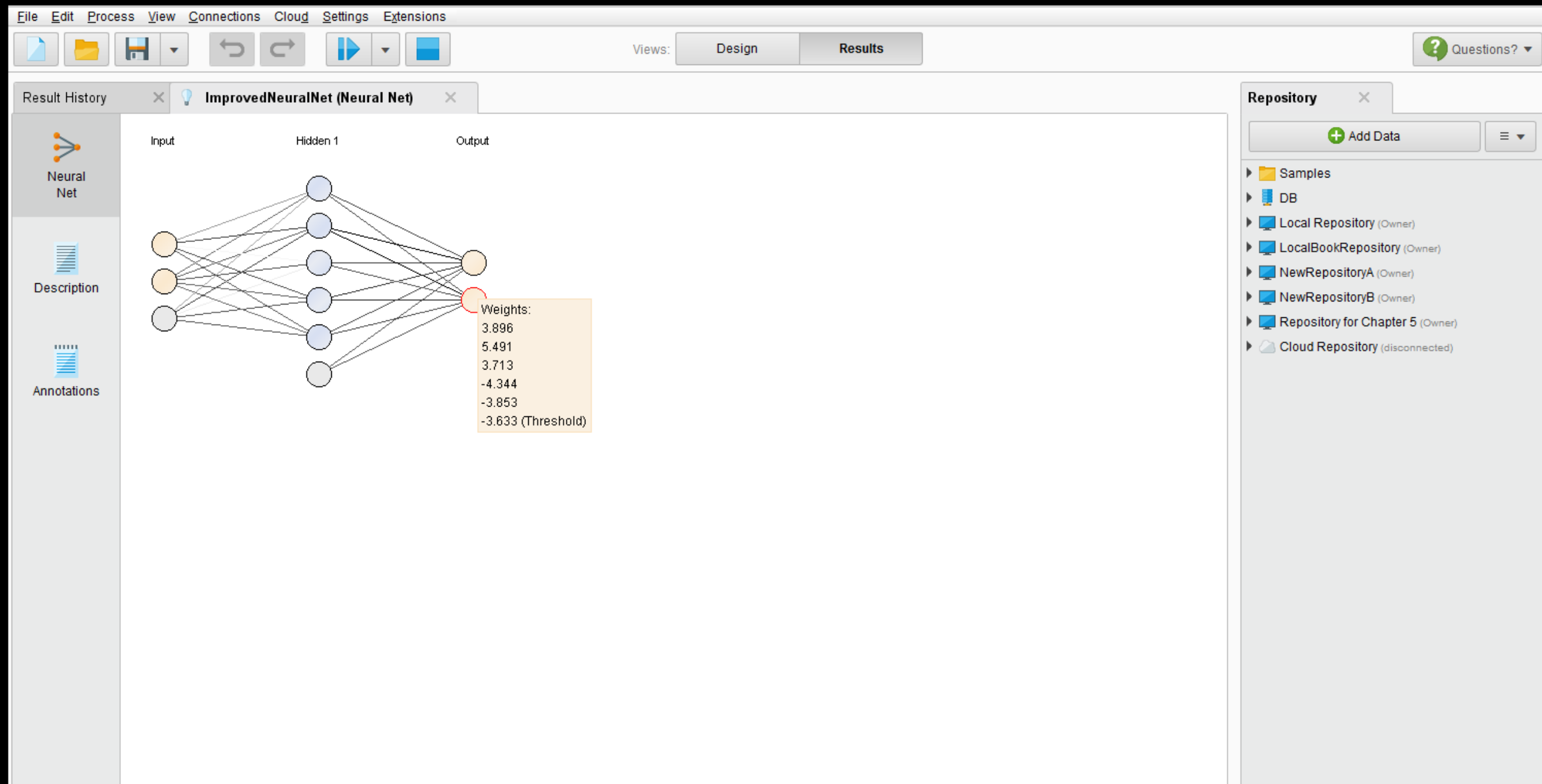


Figure 10.7 Hidden-to-output layer connection weights



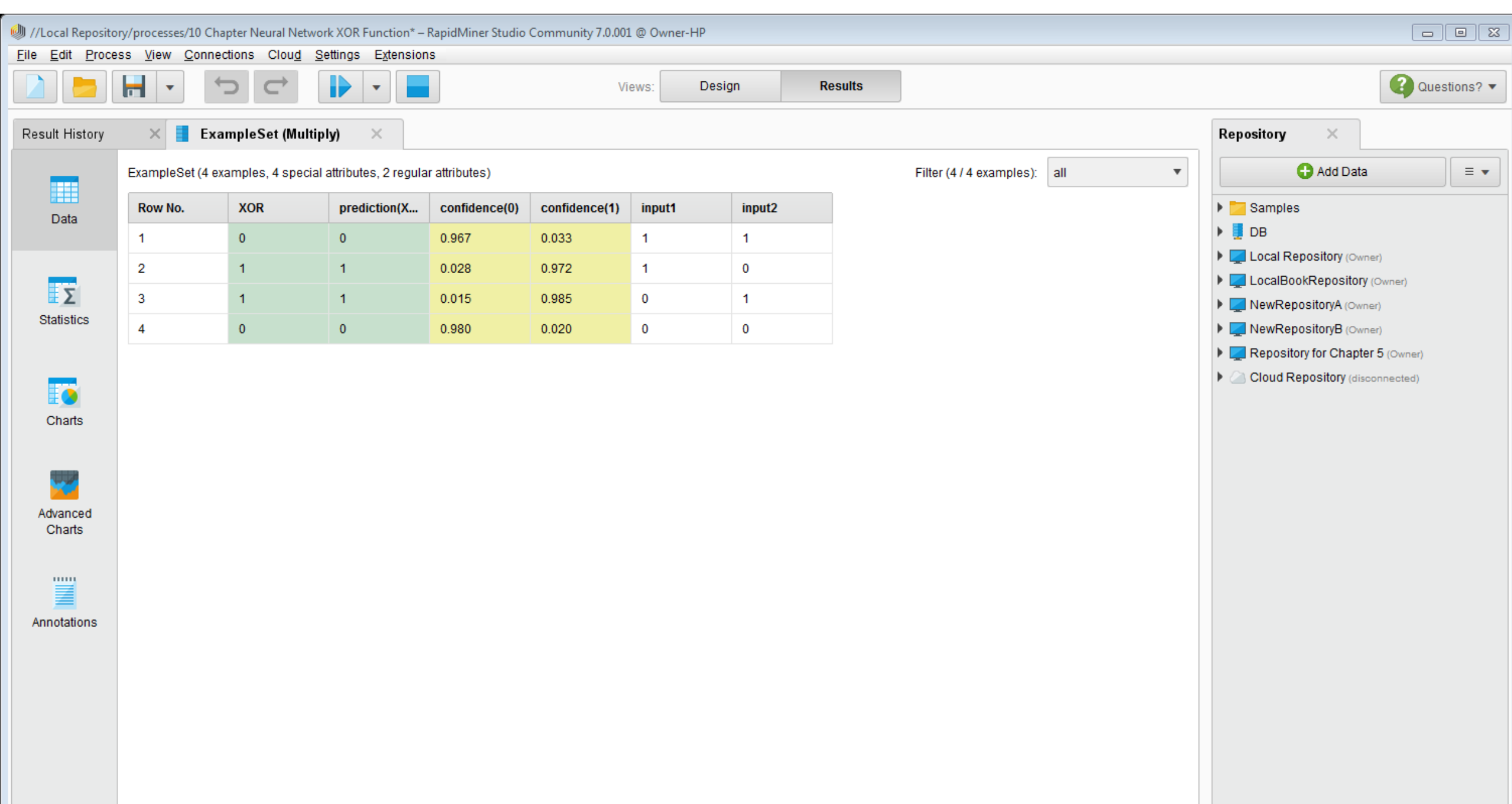


Figure 10.8 Prediction confidence values connection weights

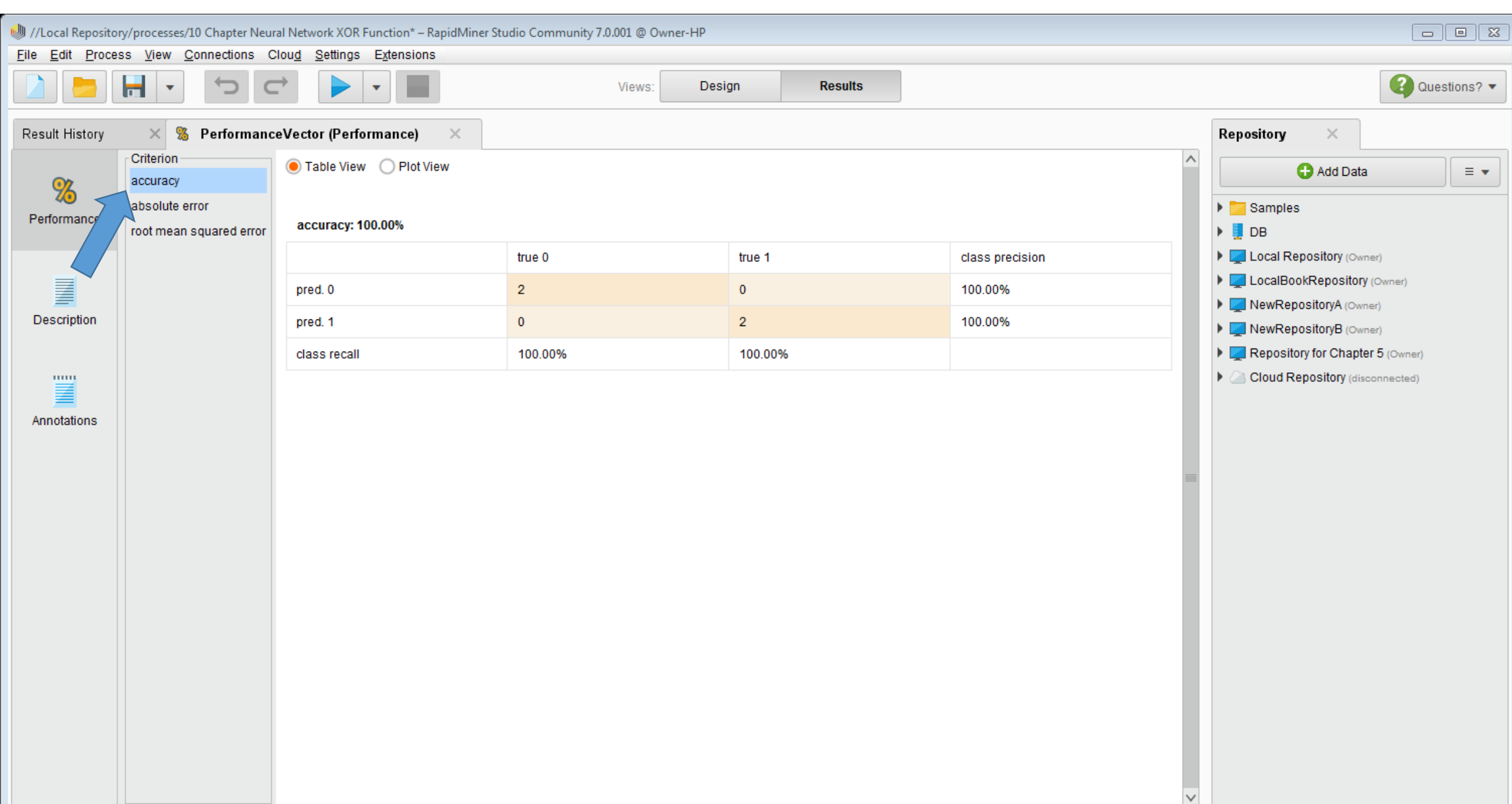
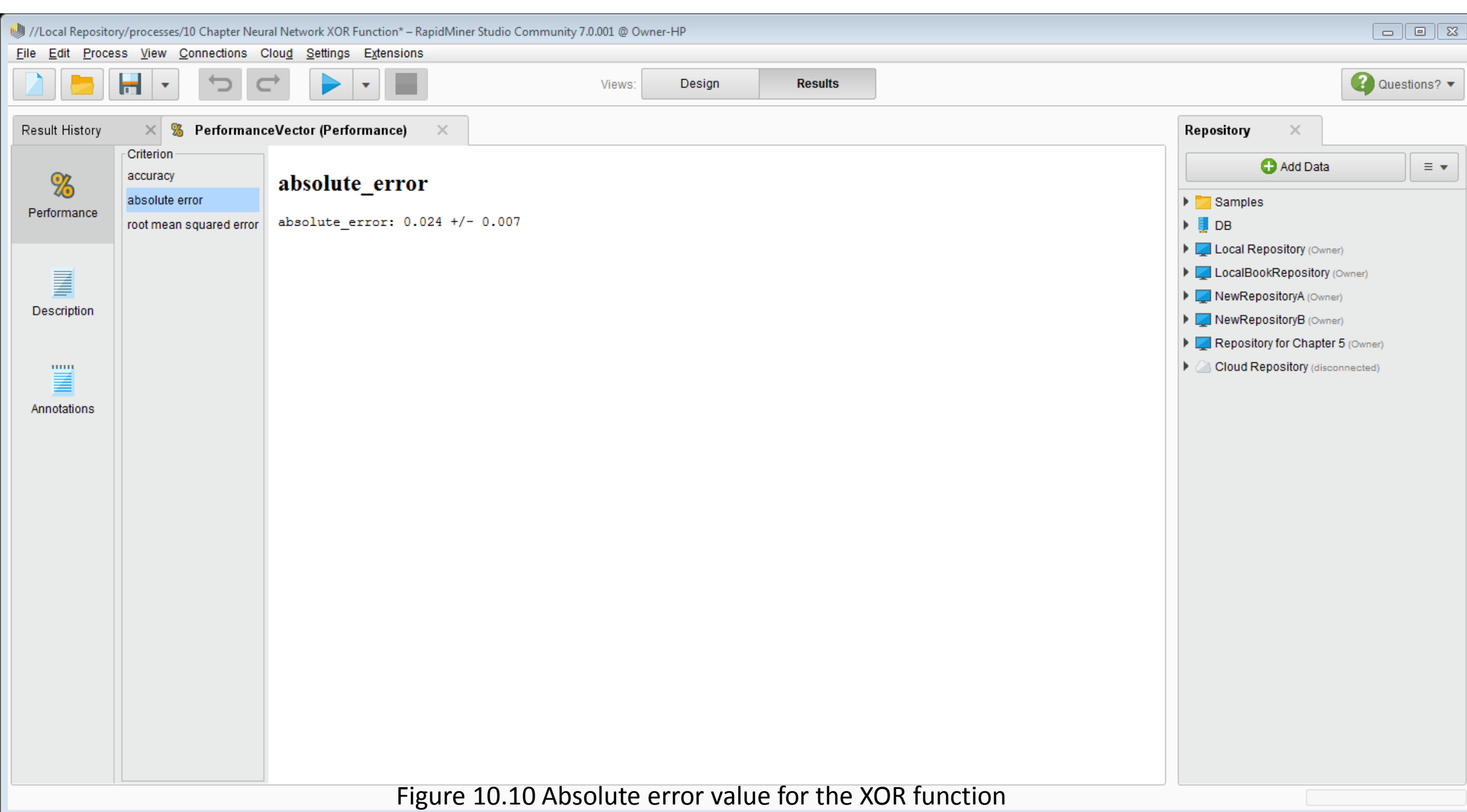


Figure 10.9 Performance vector for the XOR function



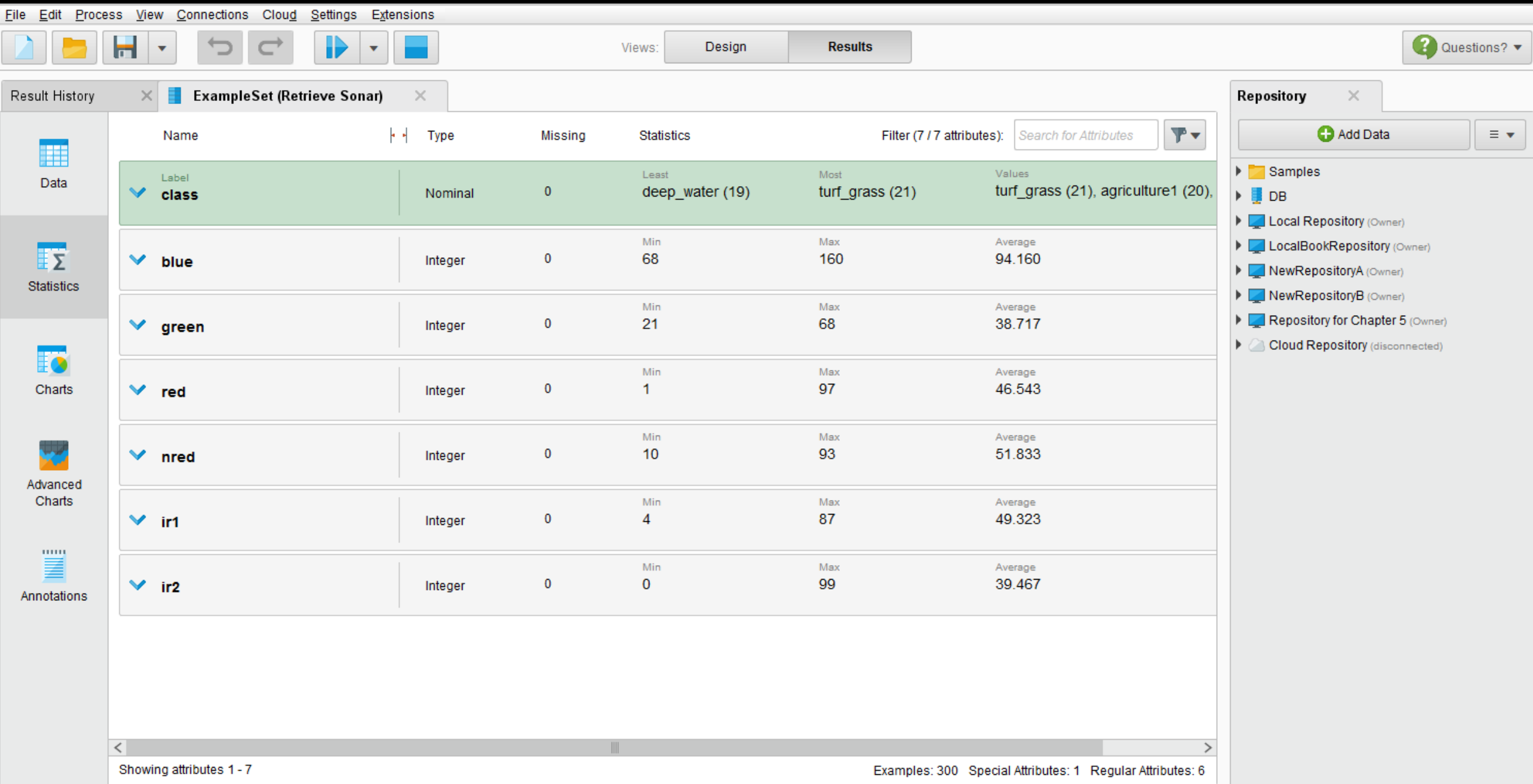


Figure 10.11 Attribute declarations for the satellite image data set

Figure 10.12 Main process for mining the satellite image data set

The screenshot displays the RapidMiner Studio interface. The top menu bar includes File, Edit, Process, View, Connections, Cloud, Settings, and Extensions. The main canvas shows a process model with two operators: 'Retrieve sonar' and 'Cross Validation'. A green callout box states: 'This process model uses the Neural Net operator to model the Satellite Image Dataset. The Cross Validation uses 2 folds.' The 'Cross Validation' operator has several output ports, with one labeled 'res' connected to a 'Neural Net' operator. The left sidebar contains a 'Repository' pane with a list of data sources and an 'Operators' pane with a search bar and a list of operator categories. The right sidebar contains a 'Parameters' pane for the 'Process' operator and a 'Help' pane for the 'Process' operator.

**Repository**

- DM End Of Chapter Exercises (Owner)
- DM Tutorials & Demos (Owner)
- Figures (Owner)
- Local Repository (Owner)
- NewLocalRepository (Owner)
- NewRepositoryA (Owner)
- NewRepositoryBB (Owner)
- Solutions Repository (Owner)
- Tutorial Repository (Owner)
- Cloud Repository (disconnected)

**Operators**

Search for Operators

- Data Access (46)
- Blending (77)
- Cleansing (26)
- Modeling (129)
- Scoring (9)
- Validation (29)
- Utility (85)
- Extensions (273)

[Get more operators from the Marketplace](#)

**Process**

Process

100%

This process model uses the Neural Net operator to model the Satellite Image Dataset. The Cross Validation uses 2 folds.

Retrieve sonar

out

Cross Validation

exa % mod

exa

tes

per

per

res

res

**Parameters**

Process

logverbosity init

logfile

resultfile

random seed 2001

send mail never

encoding SYSTEM

[Hide advanced parameters](#)

[Change compatibility \(6.0.002\)](#)

**Help**

Process

RapidMiner Studio Core

**Synopsis**

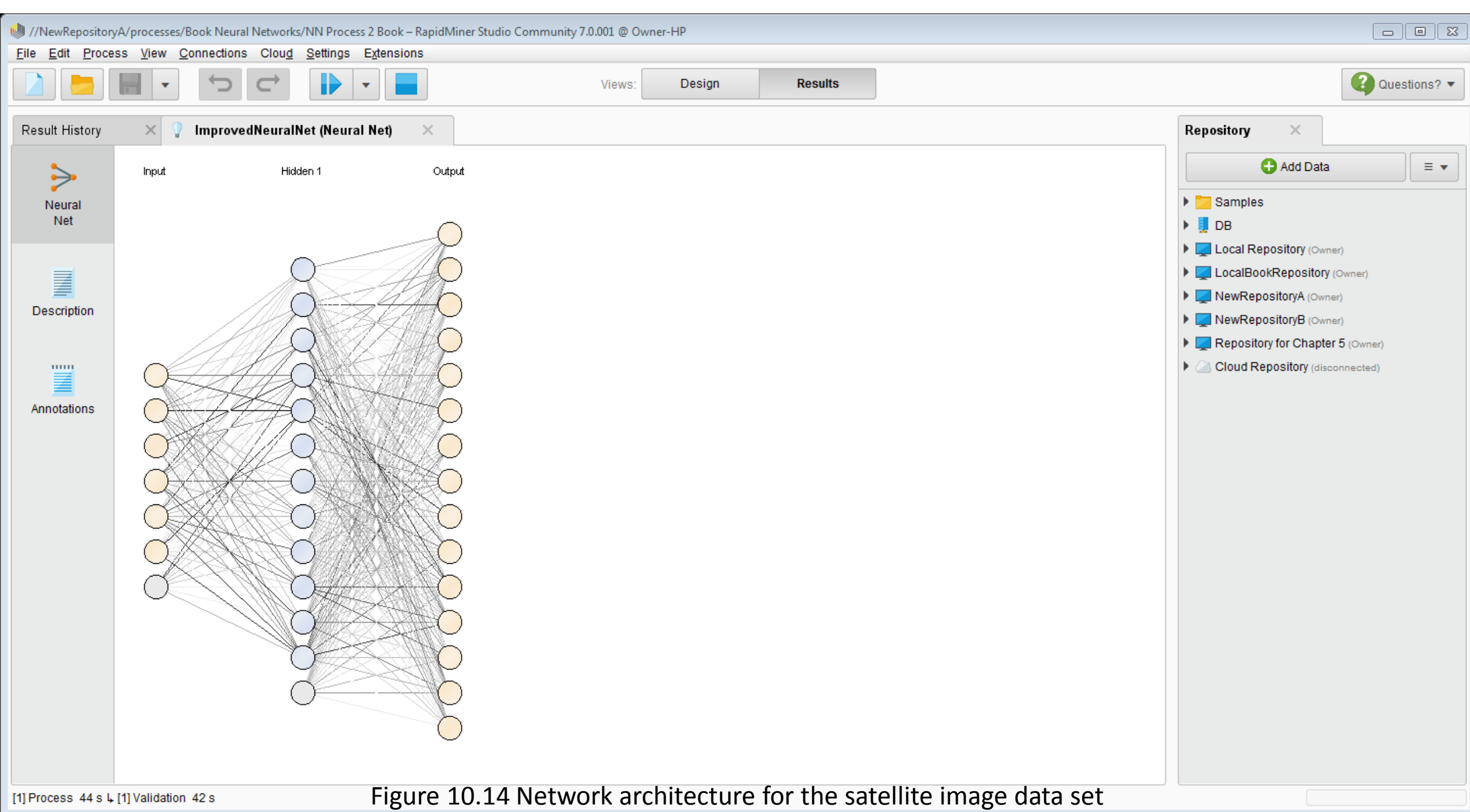
The root operator which is the outer most operator of every process.

**Description**

Figure 10.13 Subprocesses for satellite image data set

The screenshot displays the RapidMiner Studio interface with the following components:

- Repository:** A list of data sources on the left, including 'DM End Of Chapter Exercises', 'DM Tutorials & Demos', 'Figures', 'Local Repository', 'NewLocalRepository', 'NewRepositoryA', 'NewRepositoryBB', 'Solutions Repository', 'Tutorial Repository', and 'Cloud Repository (disconnected)'. The 'Tutorial Repository' is selected.
- Process:** The central workspace showing a workflow titled 'Cross Validation'. The workflow is divided into two phases:
  - Training:** A 'Neural Net' operator (green box) with inputs 'tra' and 'mod', and outputs 'mod' and 'exa'.
  - Testing:** Two operators: 'Apply Model' (green box) with inputs 'mod' and 'tes', and outputs 'mod' and 'thr'; and 'Performance' (yellow box) with inputs 'lab' and 'per', and outputs 'per' and 'exa'.
- Parameters:** A panel on the right showing the configuration for the 'Cross Validation' operator. The parameters are:
  - split on batch attribute:** ☐
  - leave one out:** ☐
  - number of folds:** 2
  - sampling type:** linear sampling
  - enable parallel execution:** ☒
- Help:** A panel at the bottom right showing the 'Cross Validation' operator's help text, including tags and a summary.



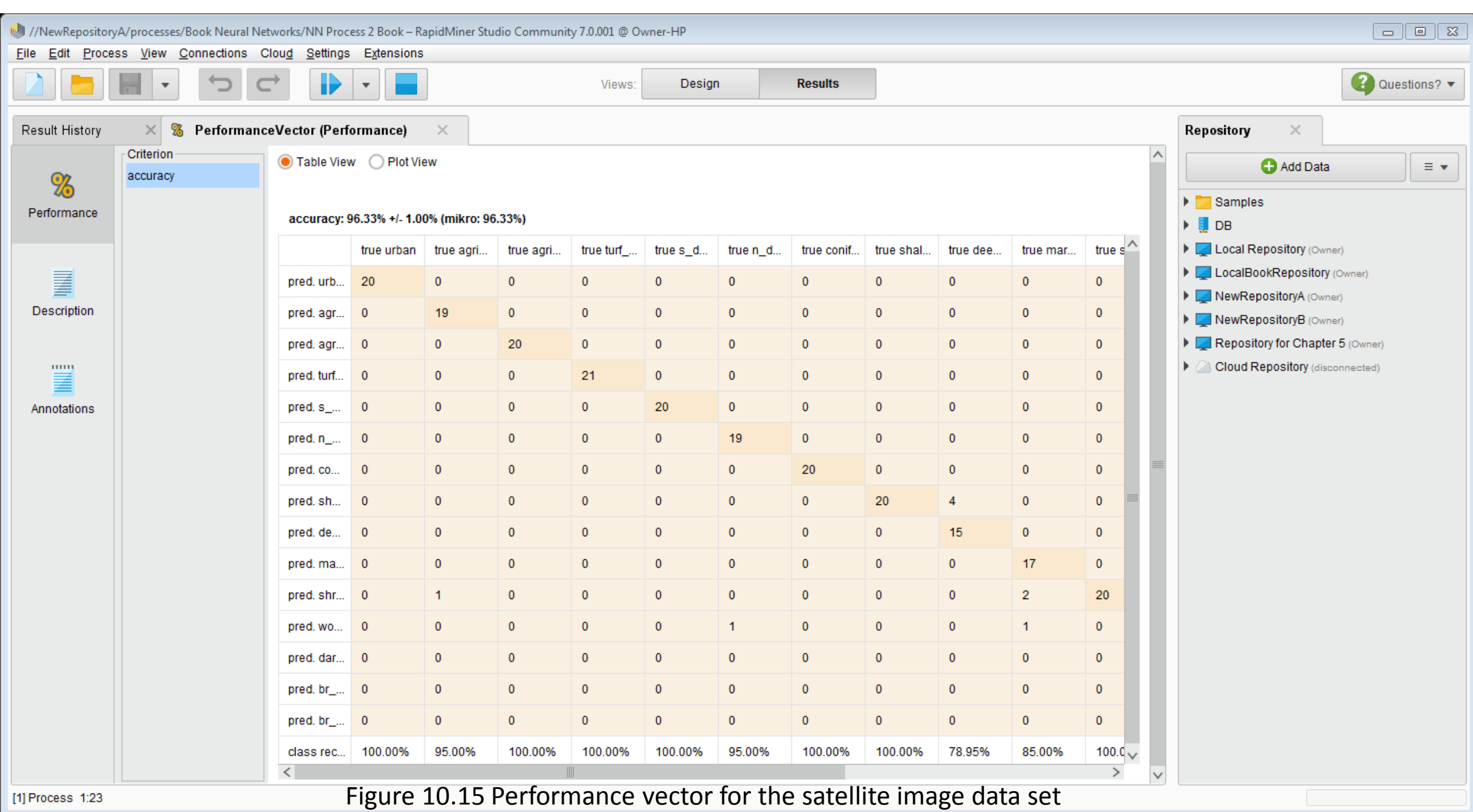




Figure 10.16 shows the RapidMiner Studio interface for removing correlated attributes from a satellite image dataset. The process model is displayed in the central canvas, and the parameters for the 'Remove Correlated Attributes' operator are shown on the right.

**Process Model:**

- The process starts with the 'Retrieve sonar' operator.
- The output is connected to the 'Multiply' operator.
- The output of 'Multiply' is connected to the 'Correlation Matrix (2)' operator.
- The output of 'Correlation Matrix (2)' is connected to the 'Remove Correlated Attributes' operator.
- The output of 'Remove Correlated Attributes' is connected to the 'Cross Validation' operator.

**Remove Correlated Attributes Parameters:**

- correlation: 0.85
- filter relation: greater
- attribute order: original
- use absolute correlation: ☒
- use local random seed: ☐

**Help Text:**

**Remove Correlated Attributes**  
RapidMiner Studio Core

Tags: Filter, Keep, Remove, Drop, Delete, Correlations, Selection

**Synopsis**  
This operator removes correlated attributes from an ExampleSet. The correlation threshold is specified by the

Figure 10.16 Removing correlated attributes from the satellite image data set

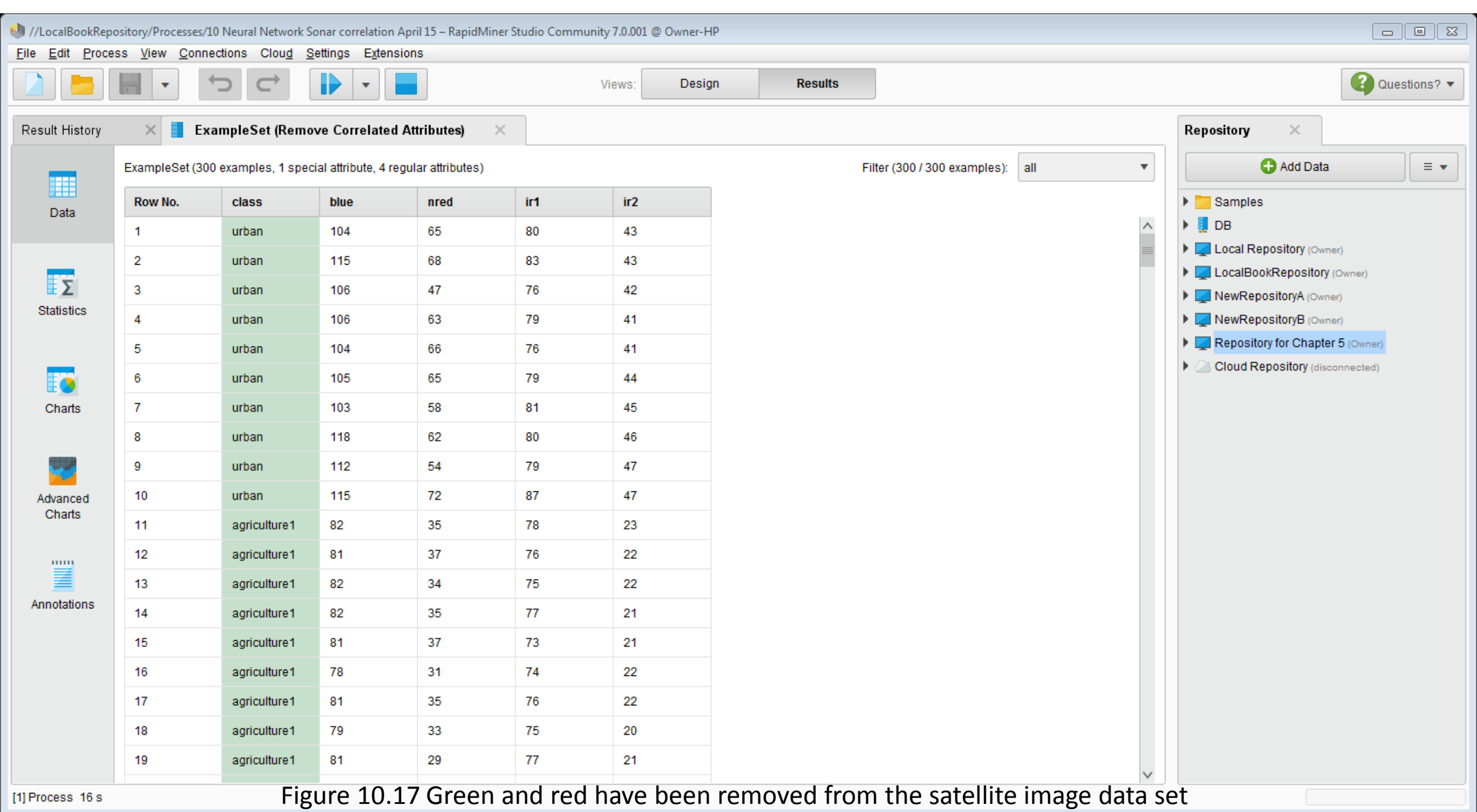


Figure 10.17 Green and red have been removed from the satellite image data set

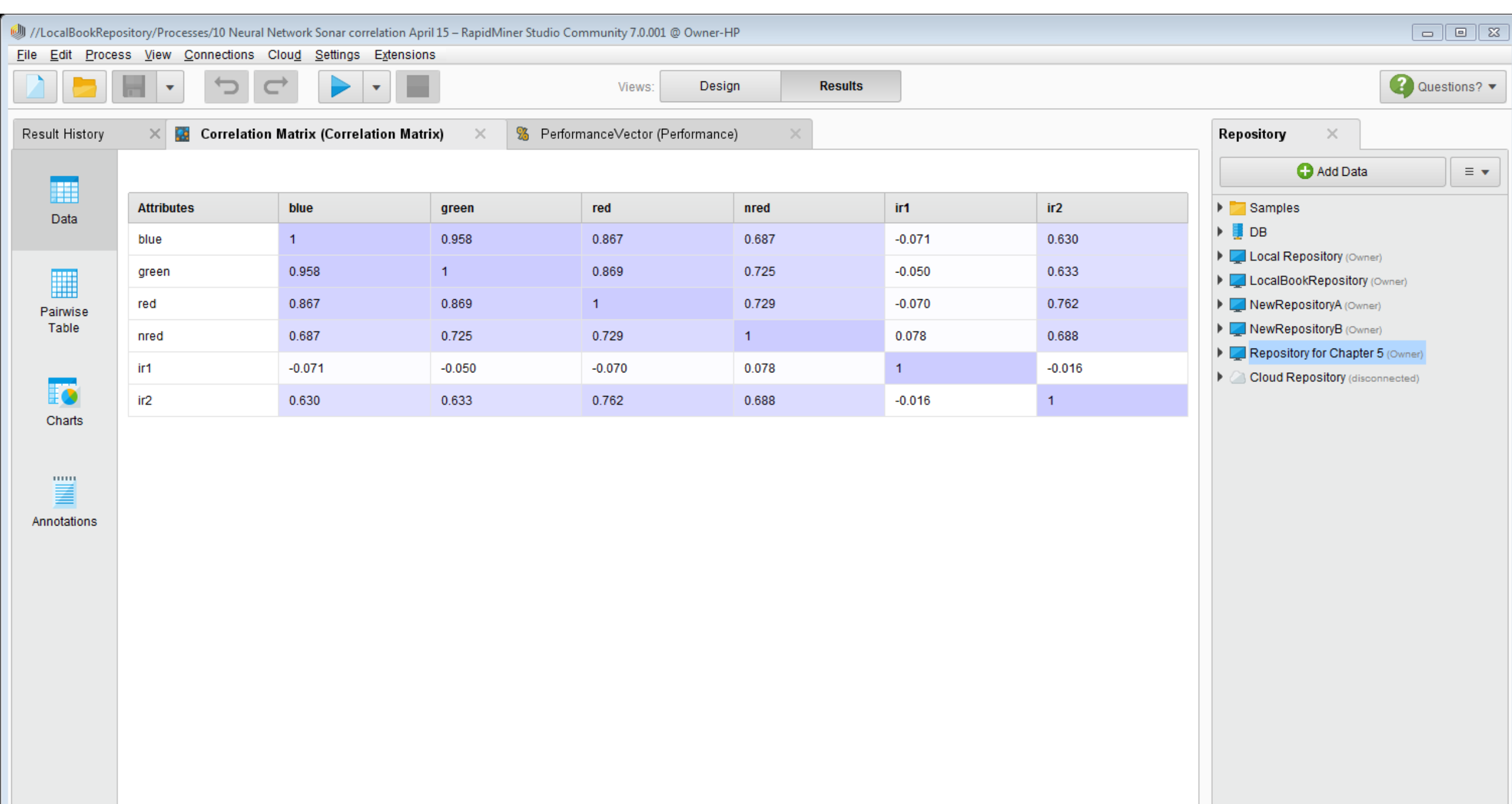


Figure 10.18 Correlation matrix for the satellite image data set

//Tutorial Repository/Chapter 10/processes/10.19 Through 10.22 Customer Churn – RapidMiner Studio Free 7.3.000 @ Owner-HP

File Edit Process View Connections Cloud Settings Extensions

Views: Design Results More

Repository

- + Add Data
- DM End Of Chapter Exercises (Owner)
- DM Tutorials & Demos (Owner)
- Figures (Owner)
- Local Repository (Owner)
- NewLocalRepository (Owner)
- NewRepositoryA (Owner)
- NewRepositoryBB (Owner)
- Solutions Repository (Owner)
- Tutorial Repository (Owner)**
- Cloud Repository (disconnected)

Operators

Search for Operators

- Data Access (46)
- Blending (77)
- Cleansing (26)
- Modeling (129)
- Scoring (9)
- Validation (29)**
- Utility (85)
- Extensions (273)

Get more operators from the Marketplace

Process

Process

100%

THIS PROCESS CREATES AND TESTS VIA CROSS VALIDATION A NEURAL NETWORK MODEL FOR PREDICTING CUSTOMER CHURN.

Retrieve Customer...

PreProcess Data

Cross Validation

Uses the Neural Net Operator

Parameters

**Cross Validation**

- ☐ split on batch attribute
- ☐ leave one out
- number of folds: 10
- sampling type: automatic
- ☐ use local random seed
- ☒ enable parallel execution

Hide advanced parameters

Help

**Cross Validation**

Concurrency

Tags: Cross-Validations, Cross-validations, Folds, K-Folds, K-folds, Validations, Estimations, Evaluations, Performances, Splitting, X-Validation, X-Prediction, Validation

Summary

Figure 10.19 Neural network model for predicting customer churn

Figure 10.20 shows the RapidMiner Studio interface for preprocessing customer churn data. The main workspace displays a process diagram titled "PreProcess Data" containing two operators: "Nominal to Numerical" and "Filter Examples".

The "Nominal to Numerical" operator is configured with the following parameters:

- Input: in
- Output: out
- Example set: exa
- Output type: ori
- Preprocessing: pre

The "Filter Examples" operator is configured with the following parameters:

- Input: in
- Output: out
- Example set: exa
- Output type: ori
- Unknown outcome: unkm

The process flow is as follows:

```
graph LR; in((in)) --> N2N[Nominal to Numerical]; N2N --> FE[Filter Examples]; FE --> out((out))
```

The "Parameters" panel on the right shows the "PreProcess Data (Subprocess)" configuration. The "Help" panel on the bottom right provides a synopsis of the Subprocess operator:

**Subprocess**  
RapidMiner Studio Core

**Synopsis**

This operator introduces a process within a process. Whenever a Subprocess operator is reached during a process execution, first the entire subprocess is executed. Once the subprocess execution is complete, the flow is returned to the process (the parent).

Figure 10.20 Preprocessing the customer churn data

Figure 10.21 shows the Cross-validation subprocess for customer churn in RapidMiner Studio Free 7.3.000. The interface displays the Process view, showing the workflow for training and testing a Neural Net (2) model.

The workflow is divided into two main sections: Training and Testing.

**Training Section:**

- The process starts with a **Neural Net (2)** operator.
- The output of the Neural Net (2) is connected to the **Apply Model (2)** operator.

**Testing Section:**

- The **Apply Model (2)** operator is connected to the **Performance (2)** operator.
- The **Performance (2)** operator outputs performance metrics (tes, per, per).

The **Parameters** panel on the right shows the configuration for the **Neural Net (2) (Neural Net)** operator:

- hidden layers:** Edit List (0)...
- training cycles:** 500
- learning rate:** 0.3
- momentum:** 0.2
- ☐ **decay**
- ☒ **shuffle** (indicated by a blue arrow)
- ☒ **normalize**
- [Hide advanced parameters](#)

The **Help** panel on the right provides information about the **Neural Net** operator:

- Neural Net**  
RapidMiner Studio Core
- Tags: [Supervised](#), [Classification](#), [Regression](#), [Model](#), [Deep](#), [Backpropagation](#), [Back-propagation](#), [Perceptron](#), [Network](#), [Artificial](#), [Anns](#), [Neural Nets](#)
- Synopsis**

The **Repository** panel on the left shows the data sources available:

- DM End Of Chapter Exercises (Owner)
- DM Tutorials & Demos (Owner)
- Figures (Owner)
- Local Repository (Owner)
- NewLocalRepository (Owner)
- NewRepositoryA (Owner)
- NewRepositoryBB (Owner)
- Solutions Repository (Owner)
- Tutorial Repository (Owner)**
- Cloud Repository (disconnected)

The **Operators** panel on the left shows the available operators:

- Data Access (46)
- Blending (77)
- Cleansing (26)
- Modeling (129)
- Scoring (9)
- Validation (29)**
- Utility (85)
- Extensions (273)

The **Process** panel at the top shows the current process: **Cross Validation**.

Figure 10.21 Cross- validation subprocess for customer churn

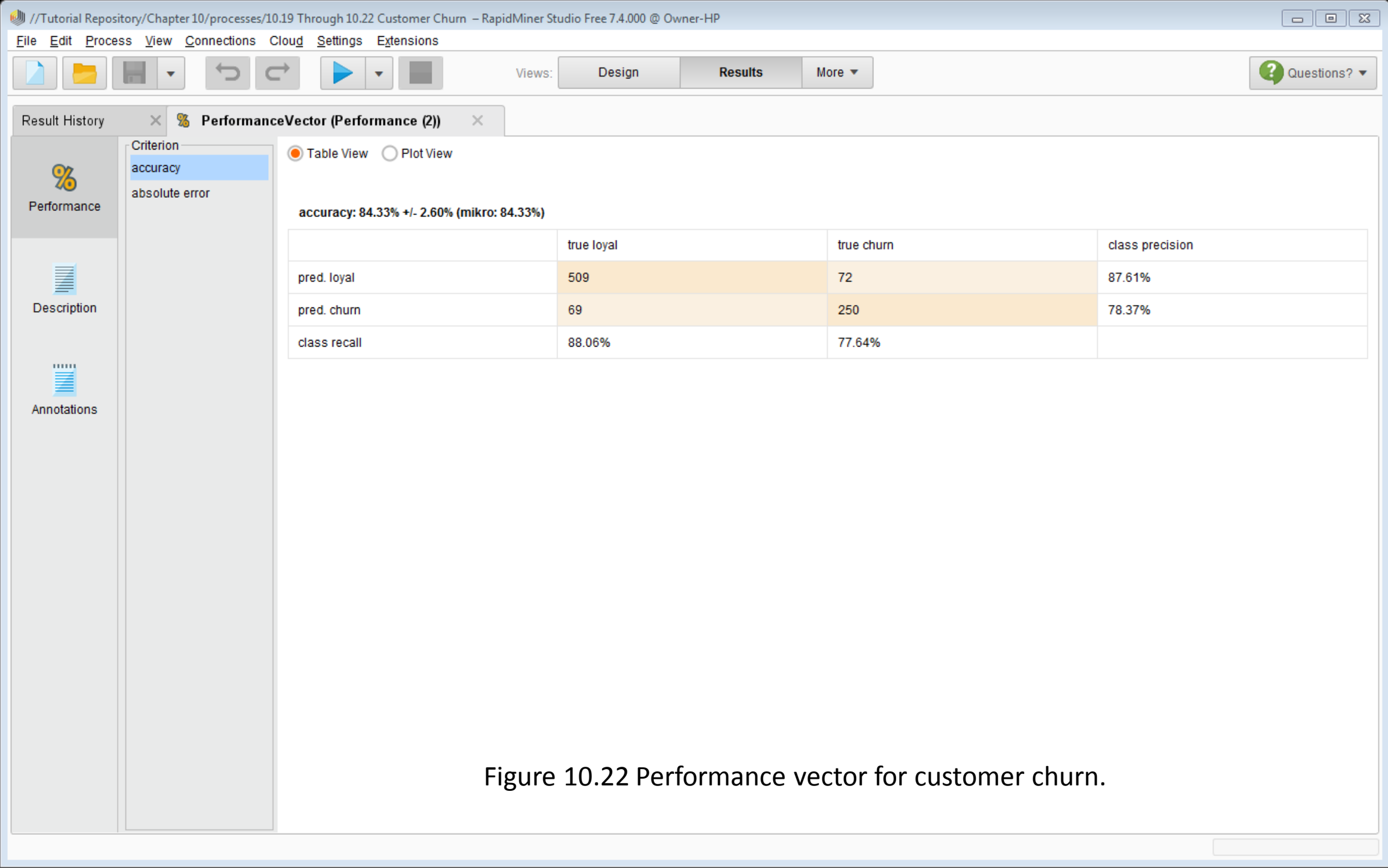


Figure 10.23 shows the RapidMiner Studio interface for creating and saving a neural network model. The main workspace displays a process diagram with the following steps:

- Retrieve Customer...**: Retrieves the RapidMiner Churn Dataset (900 instances whose class is known and 96 unknown instances).
- PreProcess Data**: Preprocesses the data.
- Neural Net**: Creates a neural network model.
- Store**: Stores the created model in the chosen repository.

The process is described as: "THIS PROCESS CREATES AND SAVES A NEURAL NETWORK MODEL TO PREDICT CUSTOMER CHURN."

The interface includes the following panels:

- Repository**: Lists available data sources (Samples, DB, 1- Figures Second Printing, DM End Of Chapter Exercises, DM Tutorials & Demos, Figures, Local Repository, NewLocalRepository, NewRepositoryA, NewRepositoryBB, Solutions Repository).
- Operators**: Lists available operators (Data Access, Blending, Cleansing, Modeling, Scoring, Validation, Utility, Extensions).
- Parameters**: Shows the configuration for the **Store** operator, including the repository entry (Figure 10.23).
- Help**: Provides documentation for the **Store** operator, including tags (Save, Export, Write, Datasets, Repository, Data Access) and a synopsis (This operator stores an IO Object in the data repository).

Figure 10.23 Process for creating and saving a neural network model



//Tutorial Repository/Chapter 10/processes/10.24 Through 10.25 Apply Model - Customer Churn Data – RapidMiner Studio Free 7.4.000 @ Owner-HP

File Edit Process View Connections Cloud Settings Extensions

Views: Design Results More

Repository

- Add Data
- Samples
- DB
- 1- Figures Second Printing (Owner)
- DM End Of Chapter Exercises (Owner)
- DM Tutorials & Demos (Owner)
- Figures (Owner)
- Local Repository (Owner)
- NewLocalRepository (Owner)
- NewRepositoryA (Owner)
- NewRepositoryBB (Owner)
- Solutions Repository (Owner)

Operators

Search for Operators

- Data Access (45)
- Blending (77)
- Cleansing (26)
- Modeling (129)
- Scoring (9)
- Validation (29)
- Utility (85)
- Extensions (273)

Get more operators from the Marketplace

Process

Process

100%

This process applies the model built with the Neural Network operator to determine customers likely to churn.

Retrieve Customer...

RapidMiner Churn Dataset 900 instances whose class is known and 96 unknown instances.

PreProcess Data

PreProcess converts nominal attributes to numeric equivalents and removes all instances of KNOWN outcome from the data.

Retrieve

Apply Model

Predict outcomes for current customers

Parameters

Retrieve

repository entry

el for Figure 10.23

Help

Retrieve

RapidMiner Studio Core

Tags: [Load](#), [Import](#), [Read](#), [Datasets](#), [Examples](#), [Example Set](#), [Table](#), [Repository](#), [Data Access](#)

Synopsis

This operator reads an object from the data repository.

[Jump to Tutorial Process](#)

Figure 10.24 Process for reading and applying a neural network model

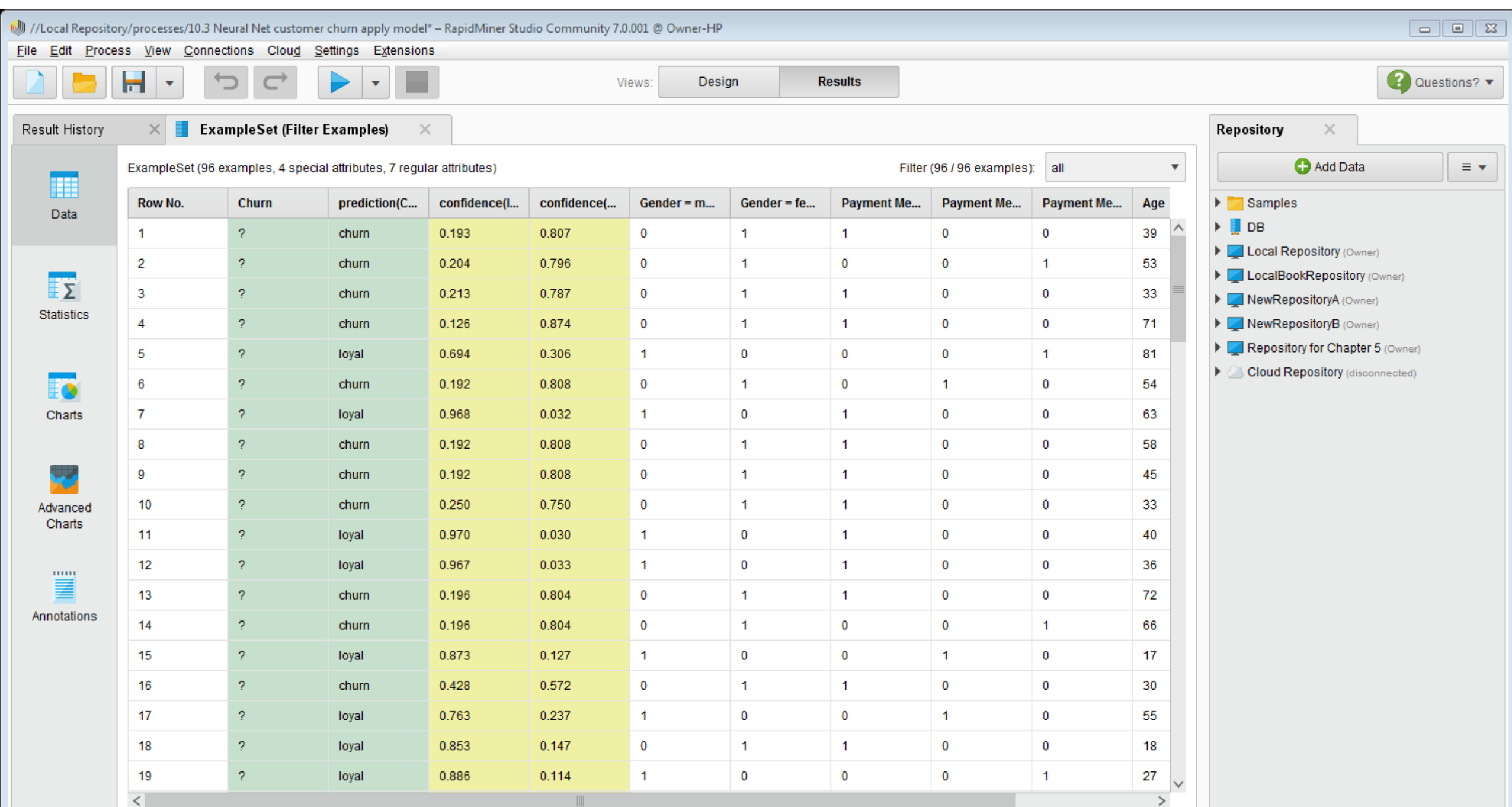


Figure 10.26 shows the SOM process model for the cardiology patient data set in RapidMiner Studio. The interface includes a Repository, Process, Parameters, Operators, and Help panels.

**Repository:** Lists data sources including Samples, DB, Local Repository, LocalBookRepository, NewRepositoryA, NewRepositoryB, and Repository for Chapter 5. The data folder under Repository for Chapter 5 contains CardiologyMixed, CreditScreening, and Customer-churn-data.

**Process:** Displays the workflow for clustering the cardiology patient dataset using a Kohonen Network. The process flow is:

- Retrieve cardiology... (Input)
- Nominal to Numerical (Transformation)
- SOM (Modeling)
- Output (res)

The central text box states: "Kohonen Network for clustering the cardiology patient dataset."

**Parameters:** Configures the SOM (Self-Organizing Map) operator with the following settings:

- return preprocessing model: ☐
- number of dimensions: 1
- net size: 2
- training rounds: 300
- learning rate start: 0.8
- learning rate end: 0.01
- adaption radius start: 10.0
- adaption radius end: 1.0

**Help:** Provides information about the Self-Organizing Map operator, including its synopsis: "This operator performs a dimensionality..."

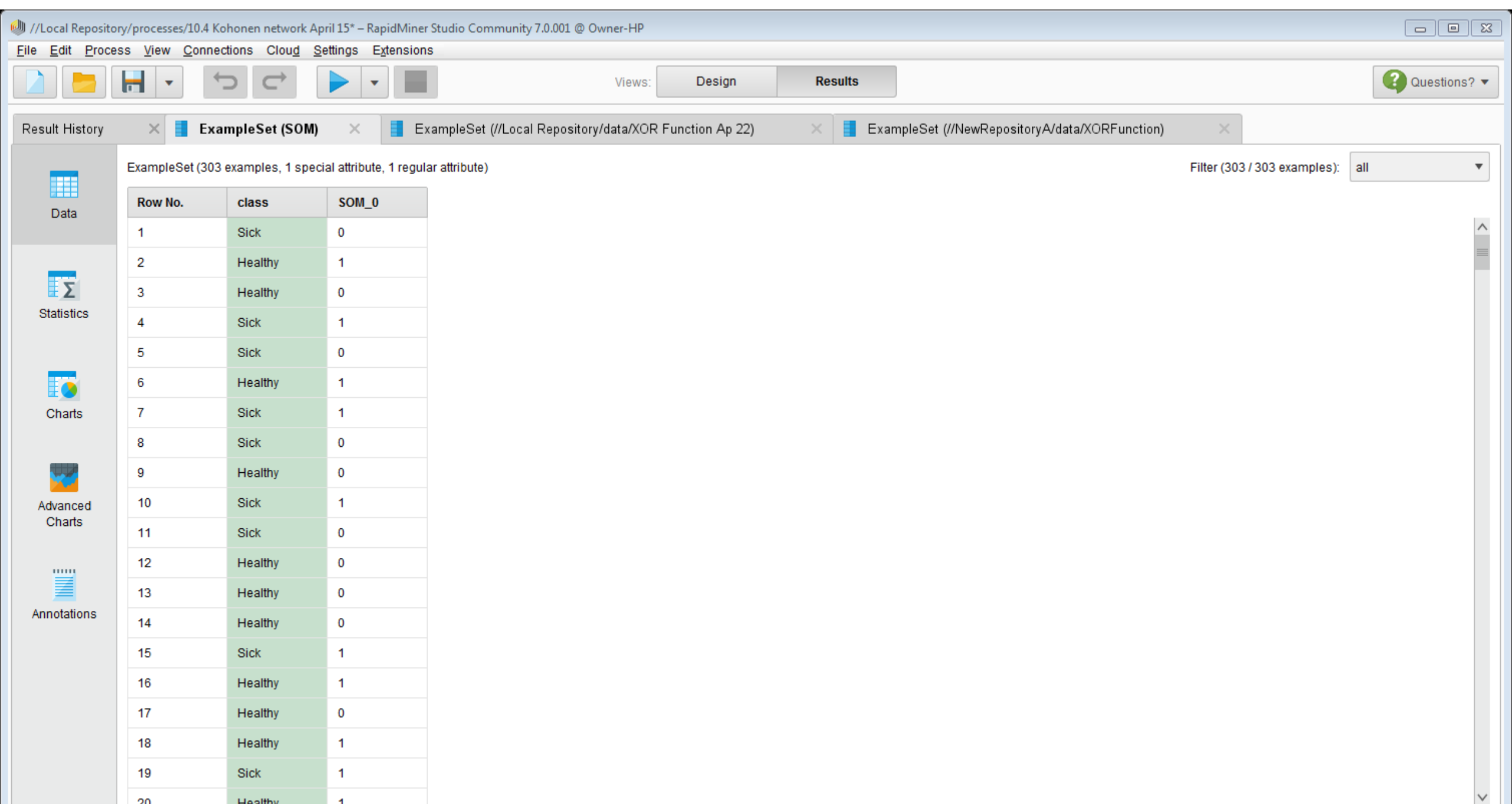


Figure 10.27 Clustered instances of the cardiology patient data set