

Chapter 5

Knowledge Discovery with RapidMiner

Installing RapidMiner

- Go to the website <https://my.rapidminer.com/nexus/account/index.html#downloads> to download and install the latest version of RapidMiner Studio (Currently RapidMiner 8.2)
- This website also contains links to the RapidMiner Studio manual, operator reference guide, tutorials, and reference notes.
- Use the *RapidMiner Studio* shortcut on your desktop to open RapidMiner.
- Click on *New Process* and then on *Blank* to start creating your first process.

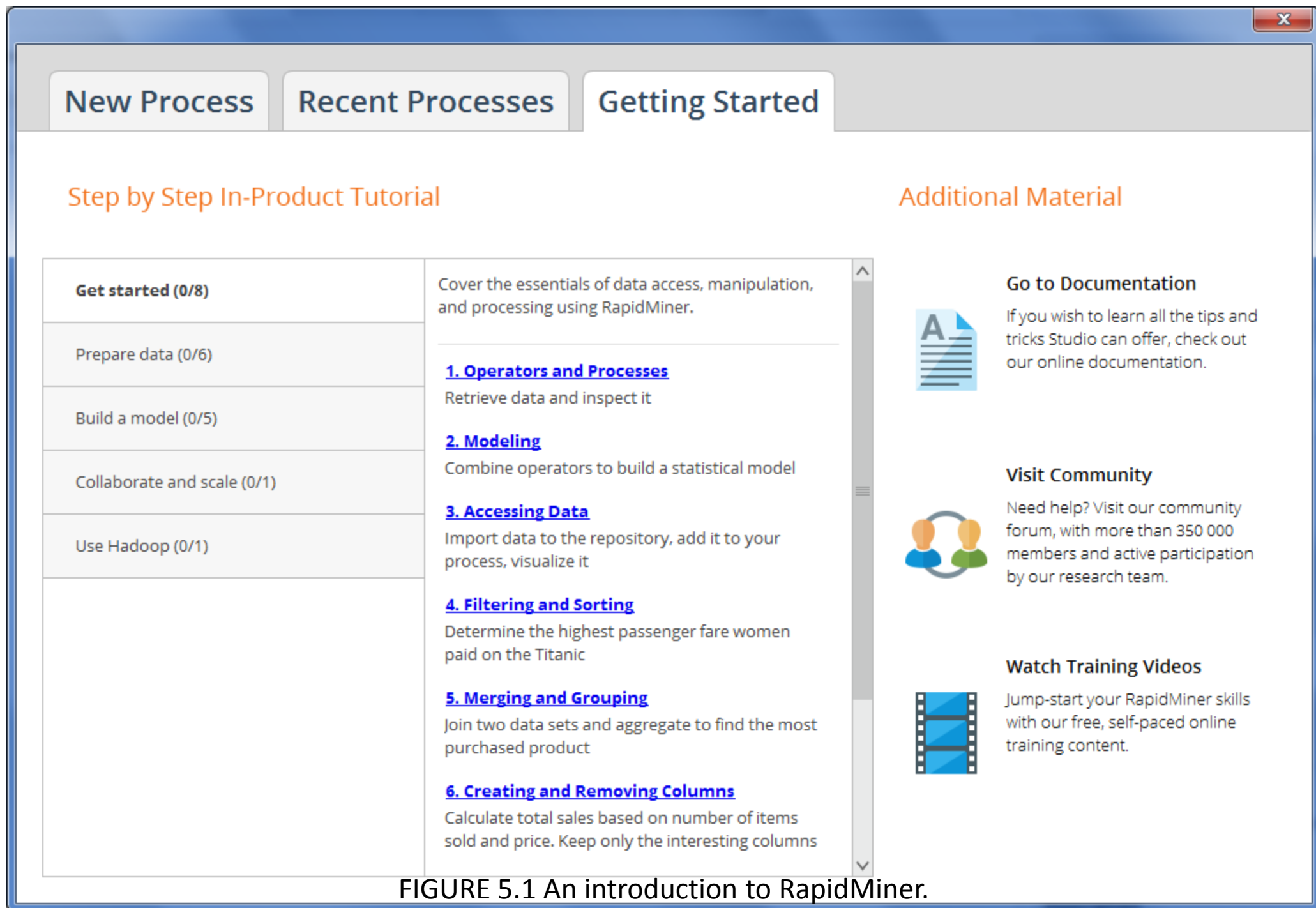


FIGURE 5.1 An introduction to RapidMiner.

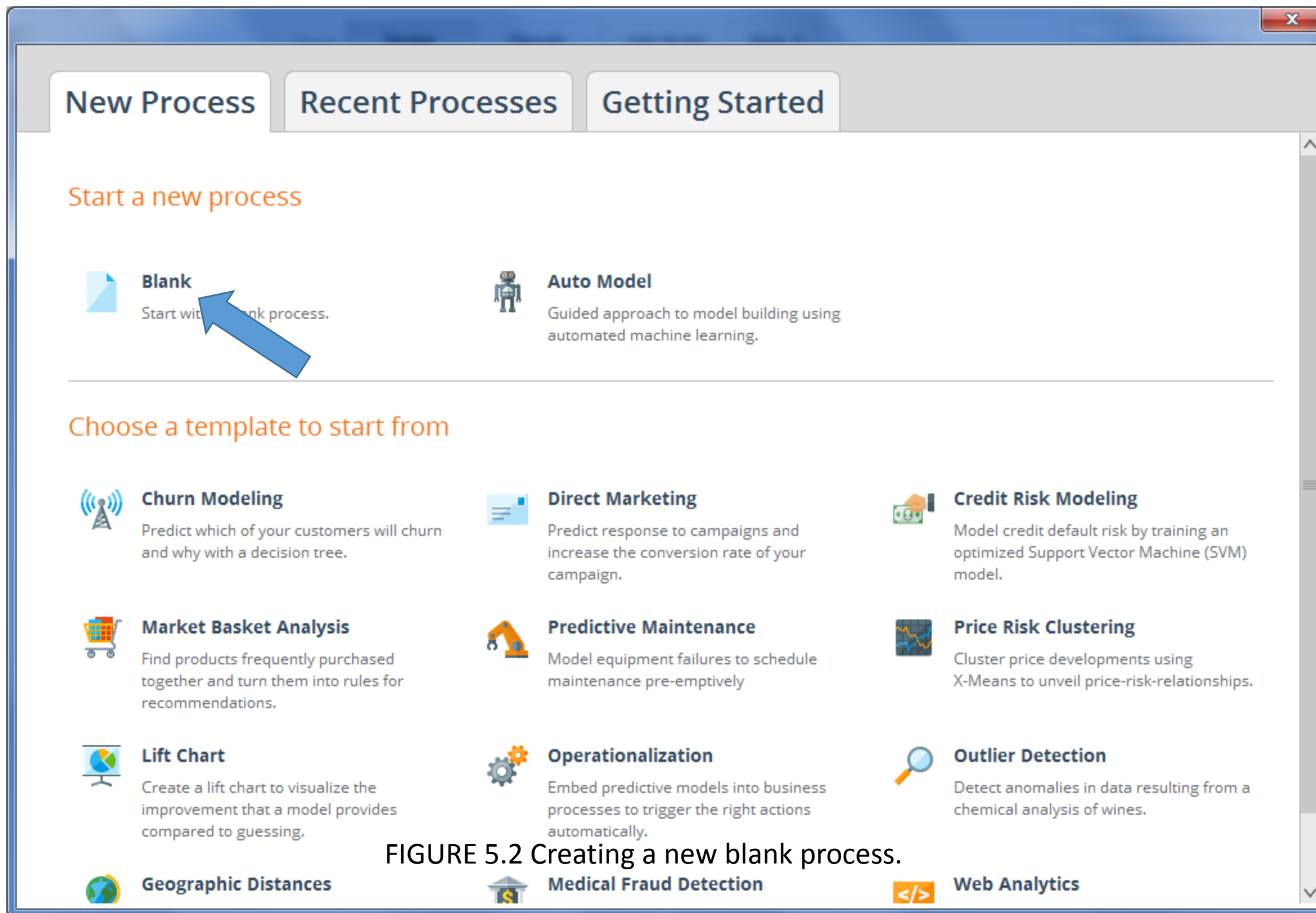
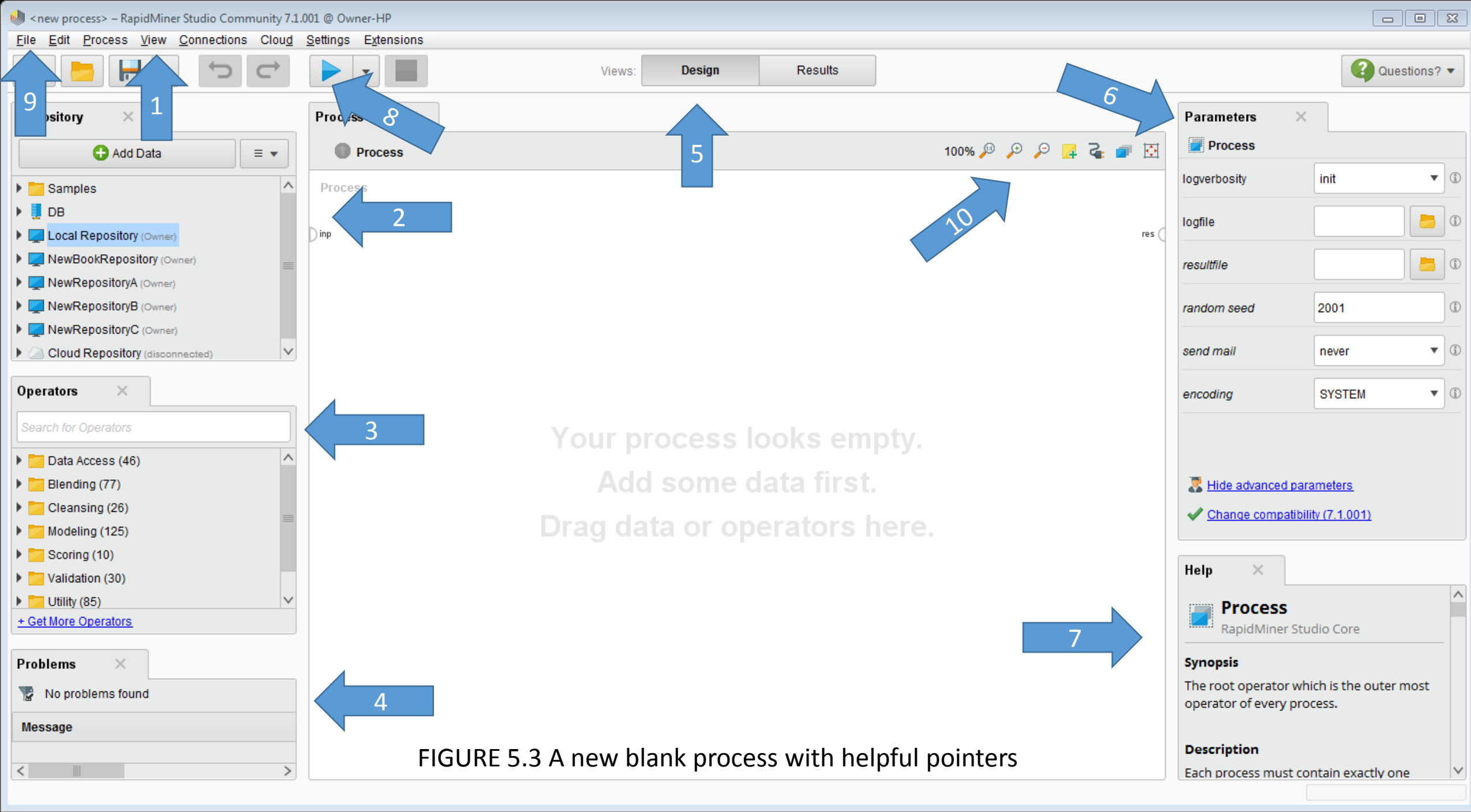


FIGURE 5.2 Creating a new blank process.



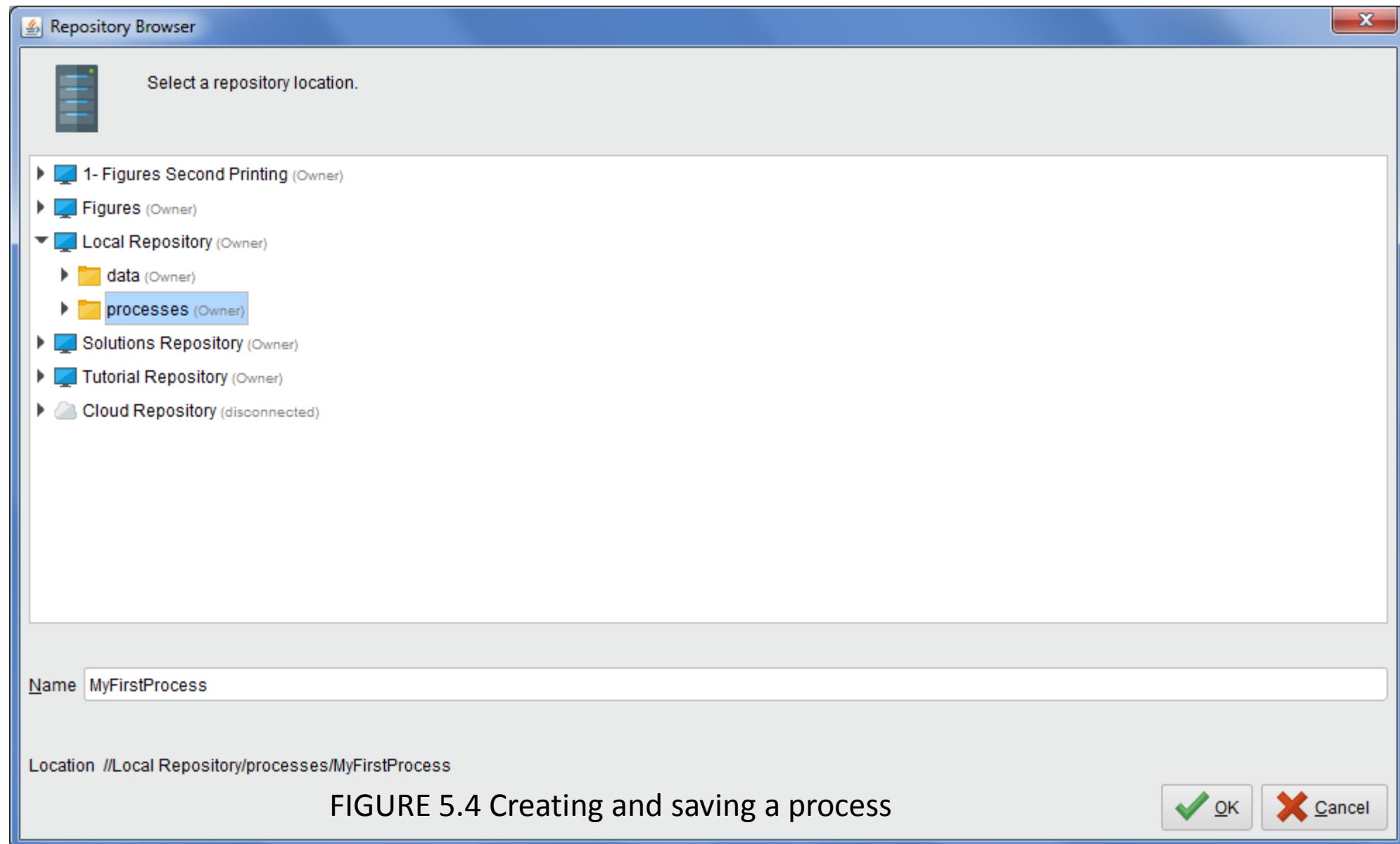


FIGURE 5.4 Creating and saving a process

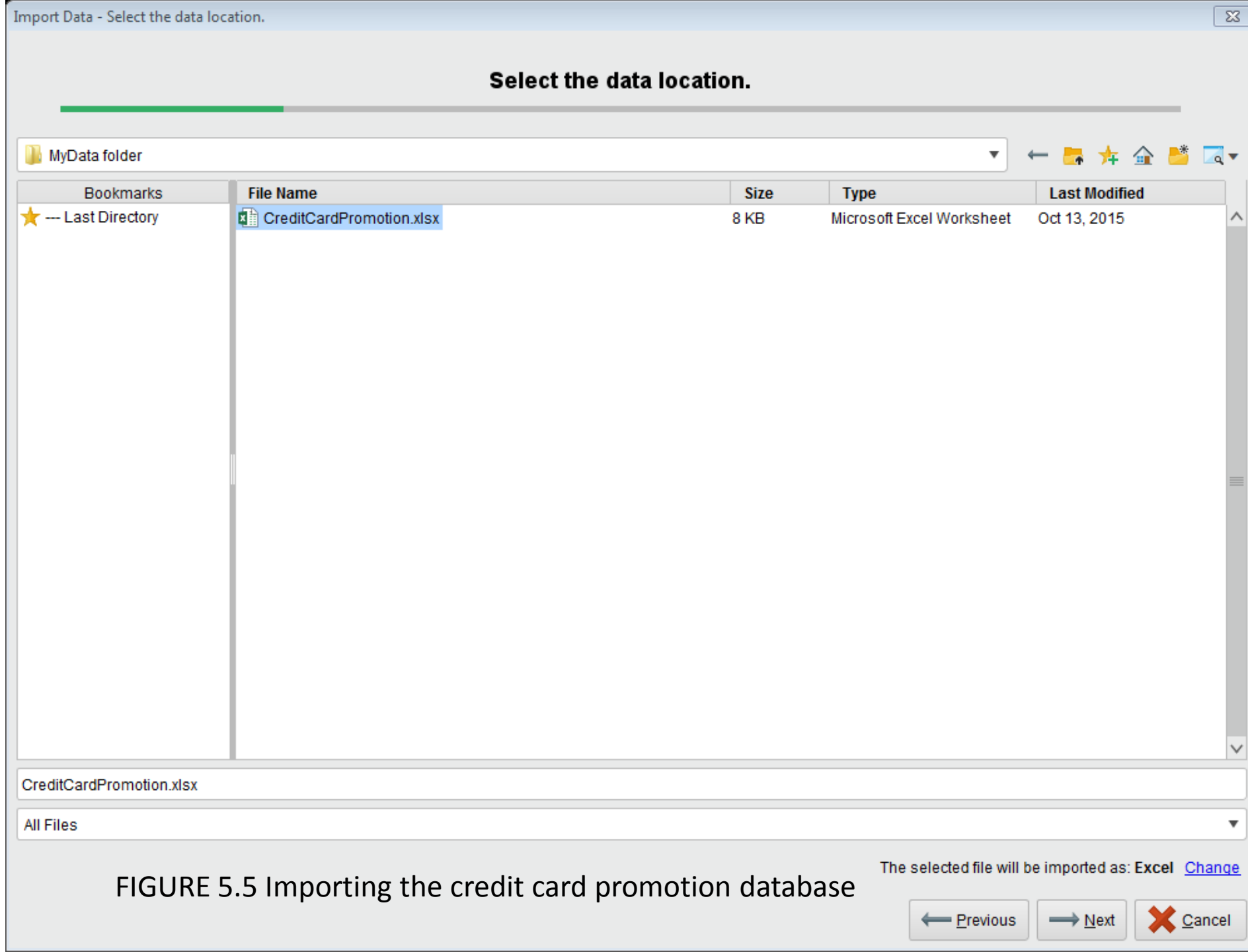


FIGURE 5.5 Importing the credit card promotion database

Select the cells to import.

Sheet: Sheet1 ▼

Cell range: A:G

Select All

☒ Define header row: 1 ▼

	A	B	C	D	E	F	G
1	Income Range	Magazine Promo	Watch Promo	Life Ins Promo	Credit Card Ins.	Gender	Age
2	40-50,000	Yes	No	No	No	Male	45.000
3	30-40,000	Yes	Yes	Yes	No	Female	40.000
4	40-50,000	No	No	No	No	Male	42.000
5	30-40,000	Yes	Yes	Yes	Yes	Male	43.000
6	50-60,000	Yes	No	Yes	No	Female	38.000
7	20-30,000	No	No	No	No	Female	55.000
8	30-40,000	Yes	No	Yes	Yes	Male	35.000
9	20-30,000	No	Yes	No	No	Male	27.000
10	30-40,000	Yes	No	No	No	Male	43.000
11	30-40,000	Yes	Yes	Yes	No	Female	41.000
12	40-50,000	No	Yes	Yes	No	Female	43.000
13	20-30,000	No	Yes	Yes	No	Male	29.000
14	50-60,000	Yes	Yes	Yes	No	Female	39.000
15	40-50,000	No	Yes	No	No	Male	55.000
16	20-30,000	No	No	Yes	Yes	Female	19.000

FIGURE 5.6 Selecting the cells to import

← Previous

→ Next

✕ Cancel

Format your columns.

Date format

☐ Replace errors with missing values ⓘ

	Income Range	Magazine Pro...	Watch Promo	Life Ins Promo	Credit Card Ins.	Gender	Age
	<i>polynominal</i>	<i>polynominal</i>	<i>polynominal</i>	<i>polynominal</i>	<i>polynominal</i>	<i>polynominal</i>	<i>integer</i>
1	40-50,000			No	No	Male	45
2	30-40,000			Yes	No	Female	40
3	40-50,000			No	No	Male	42
4	30-40,000	Yes		Yes	Yes	Male	43
5	50-60,000	Yes		Yes	No	Female	38
6	20-30,000	No		No	No	Female	55
7	30-40,000	Yes	No	Yes	Yes	Male	35
8	20-30,000	No	Yes	No	No	Male	27
9	30-40,000	Yes	No	No	No	Male	43
10	30-40,000	Yes	Yes	Yes	No	Female	41
11	40-50,000	No	Yes	Yes	No	Female	43
12	20-30,000	No	Yes	Yes	No	Male	29
13	50-60,000	Yes	Yes	Yes	No	Female	39
14	40-50,000	No	Yes	No	No	Male	55
15	20-30,000	No	No	Yes	Yes	Female	19

FIGURE 5.7 A list of allowable data types

✓ no problems.

← Previous

→ Next

✗ Cancel

Import Data - Format your columns. ✕

Format your columns.

Date format MMM d, yyyy h:mm:ss a z ☐ Replace errors with missing values ⓘ

	Income Range ⚙ <i>polynomial</i>	Magazine Pro... ⚙ <i>polynomial</i>	Watch Promo ⚙ <i>polynomial</i>	Life Ins Promo ⚙ <i>polynomial</i>	Credit Card Ins. ⚙ <i>polynomial</i>	Gender ⚙ <i>polynomial</i>	Age ⚙ <i>integer</i>
1	40-50,000	Yes	No	No	No	Male	45
2	30-40,000	Yes	Yes	Yes	No	Female	40
3	40-50,000	No					42
4	30-40,000	Yes					43
5	50-60,000	Yes					38
6	20-30,000	No					55
7	30-40,000	Yes					35
8	20-30,000	No					27
9	30-40,000	Yes					43
10	30-40,000	Yes					41
11	40-50,000	No					43
12	20-30,000	No	Yes	Yes	No	Male	29
13	50-60,000	Yes	Yes	Yes	No	Female	39
14	40-50,000	No	Yes	No	No	Male	55
15	20-30,000	No	No	Yes	Yes	Female	19

✎ Change role ✕

✎

Please enter the new role:

label

✓ OK ✗ Cancel

✓ no problems.

⬅ Previous ➡ Next ✗ Cancel

FIGURE 5.8 Changing the role of *Life Ins Promo*

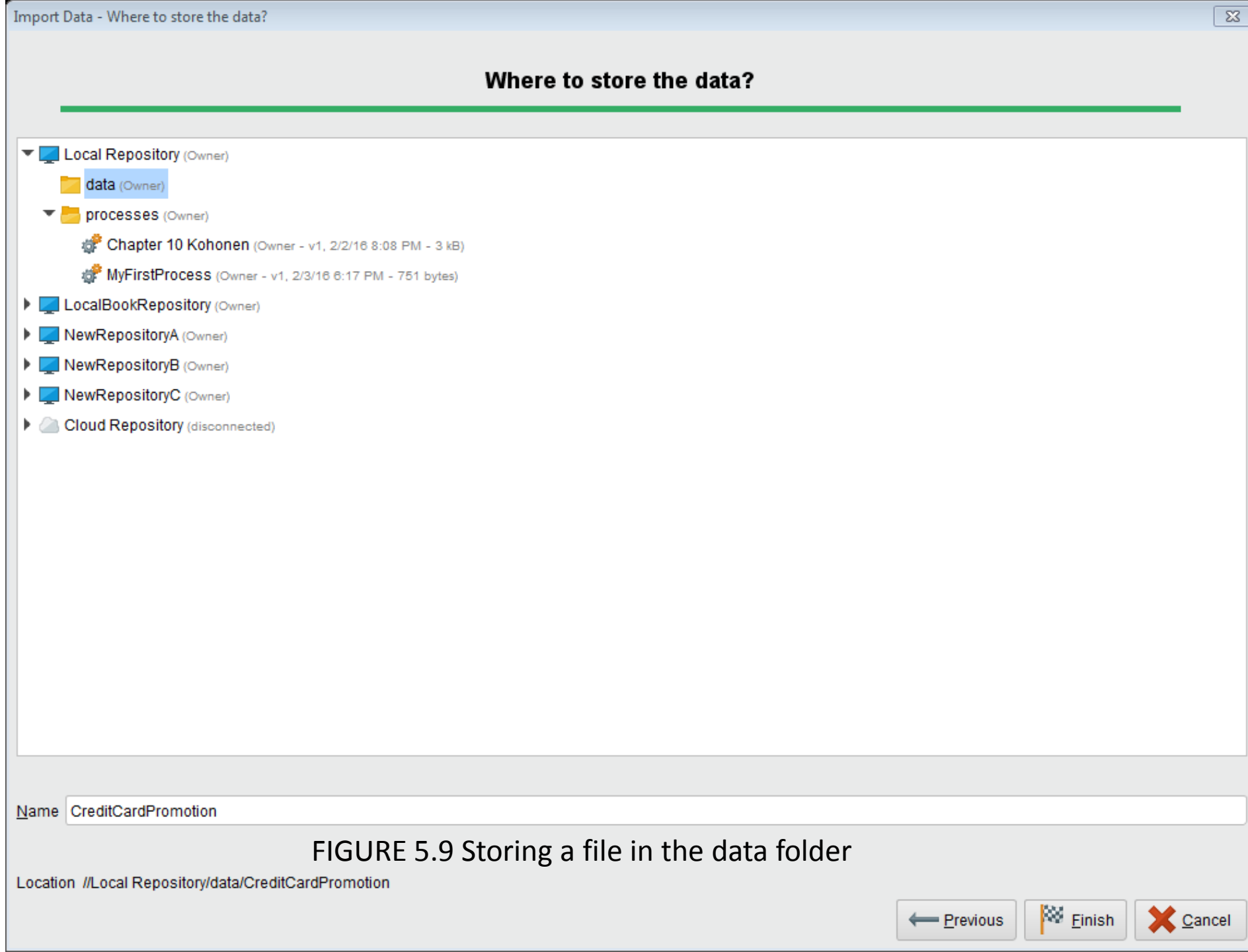


FIGURE 5.9 Storing a file in the data folder

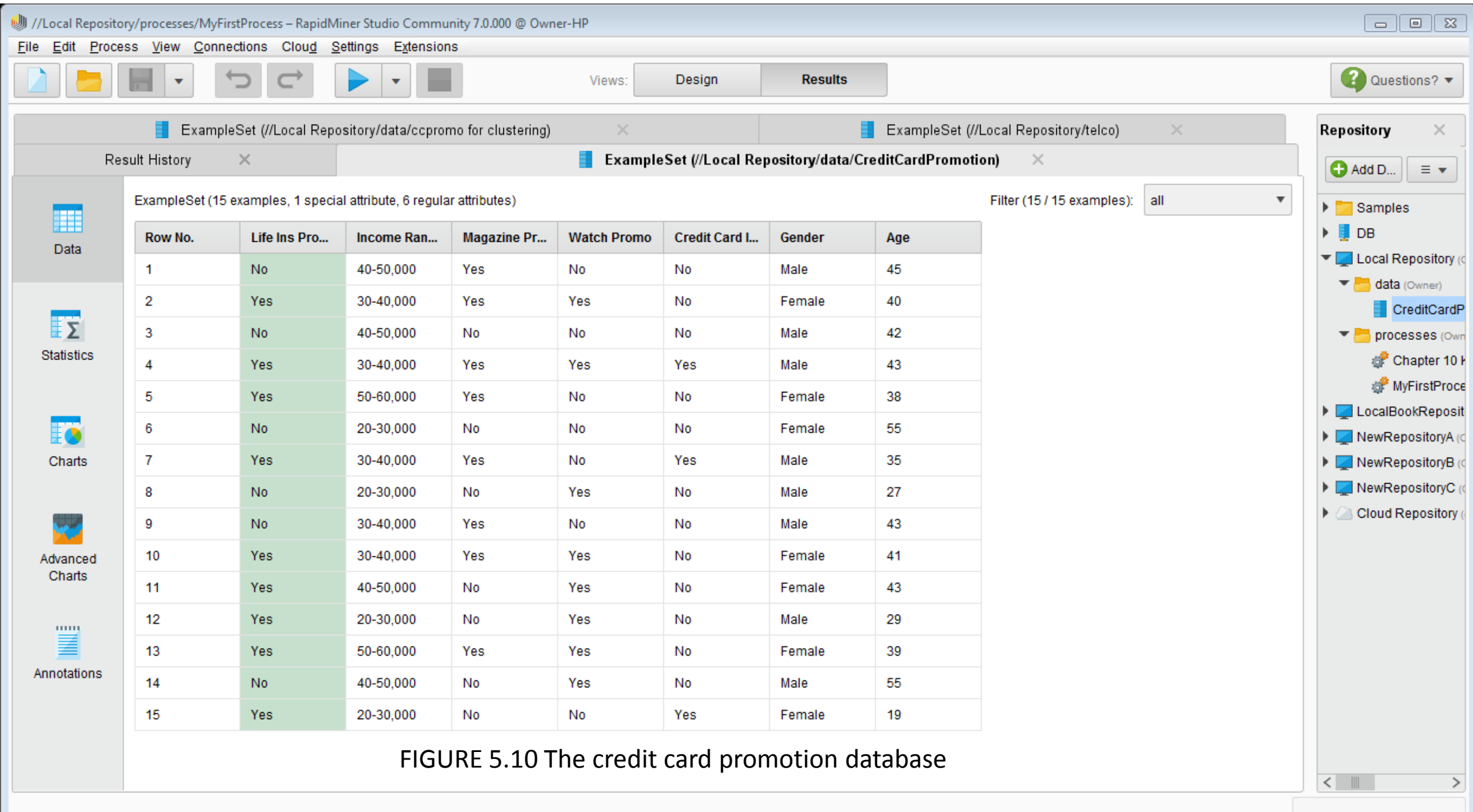


FIGURE 5.10 The credit card promotion database

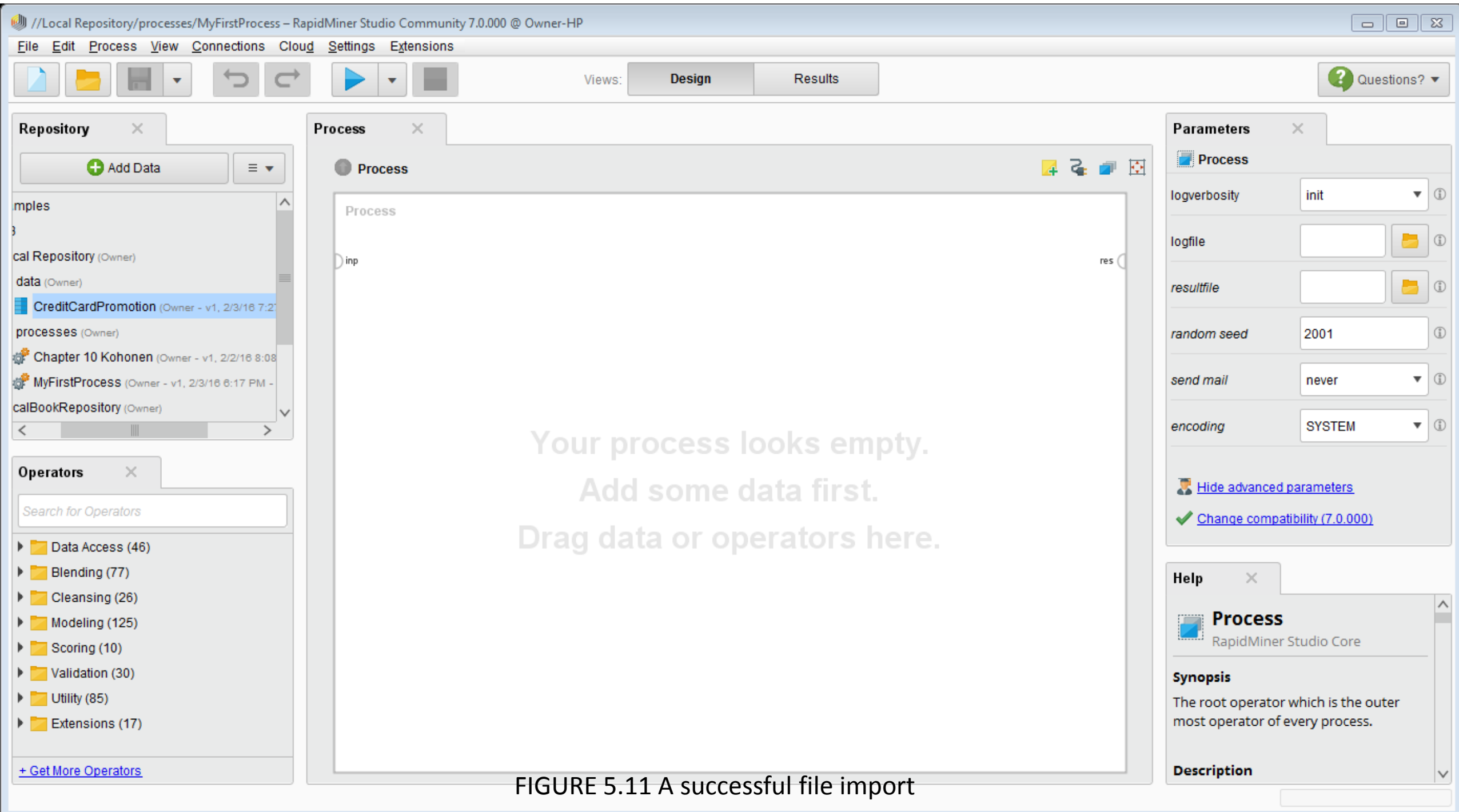


FIGURE 5.11 A successful file import

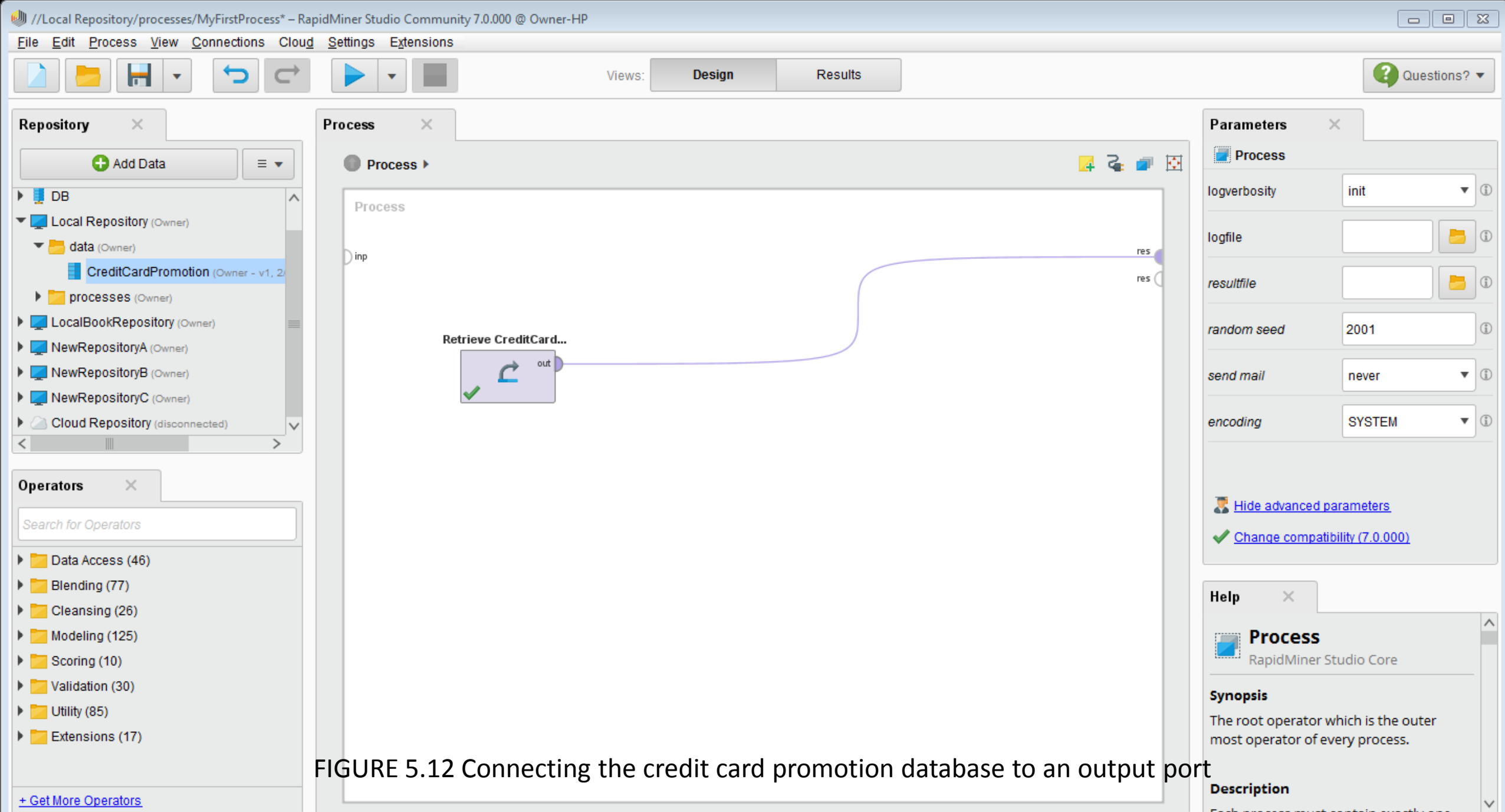
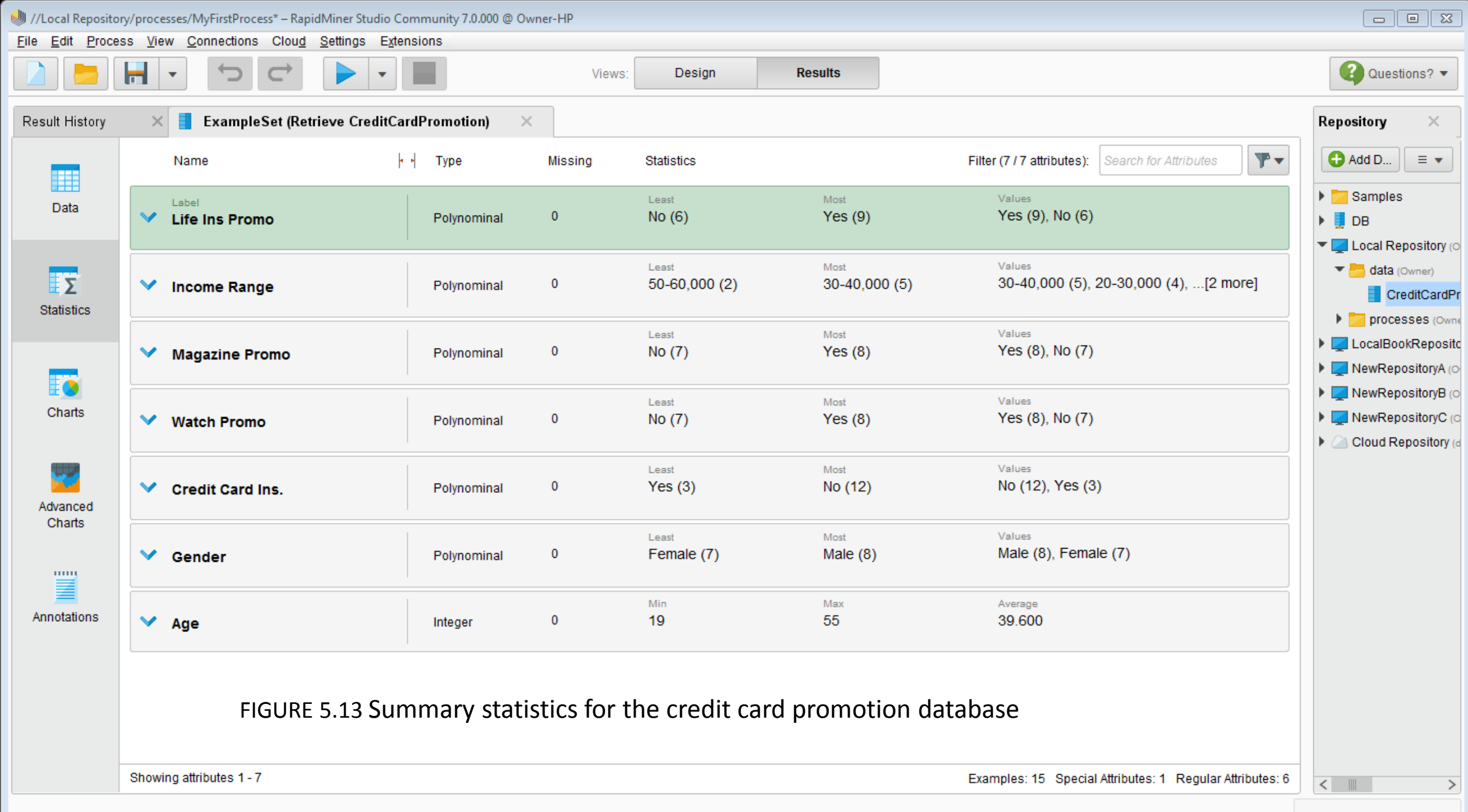


FIGURE 5.12 Connecting the credit card promotion database to an output port





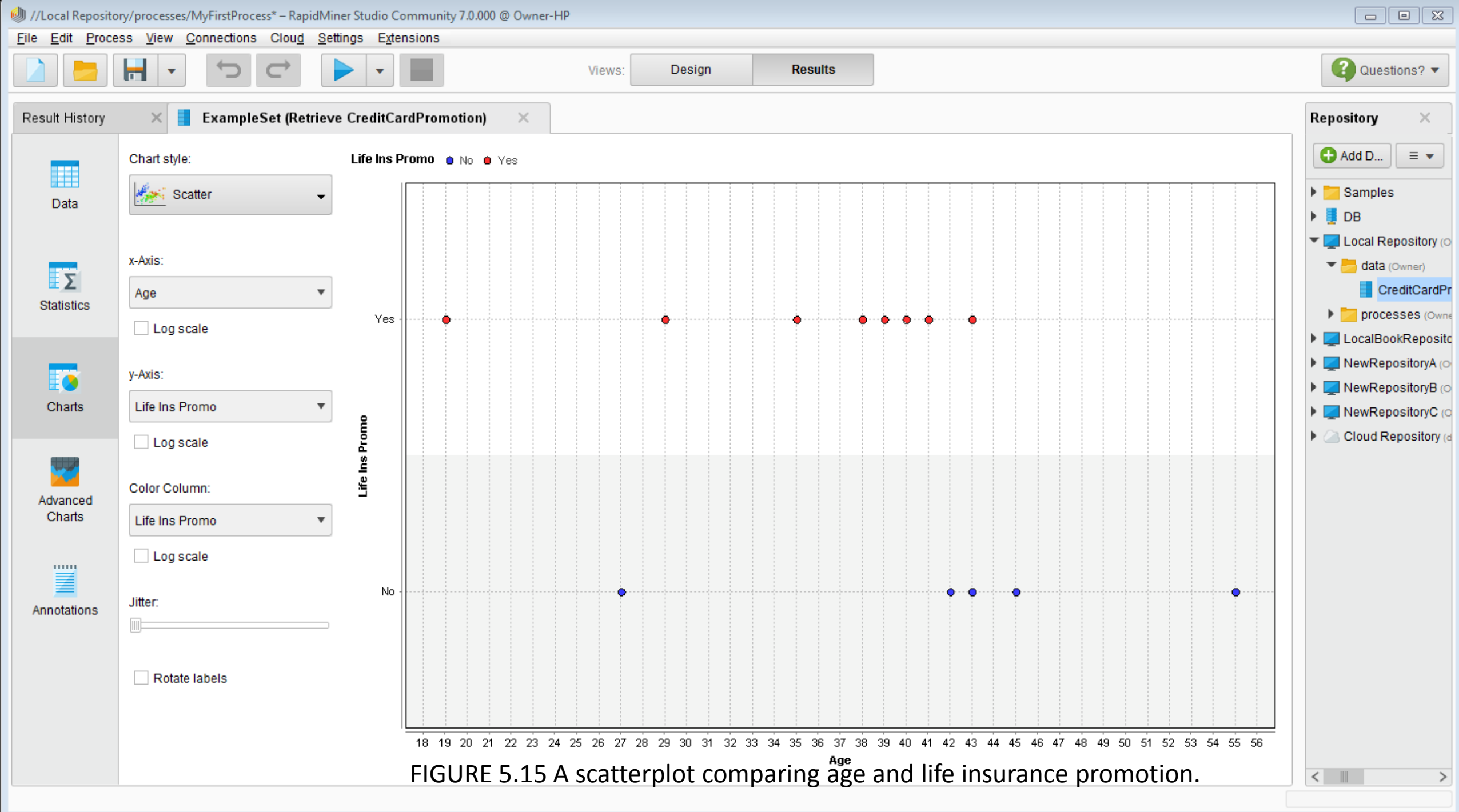
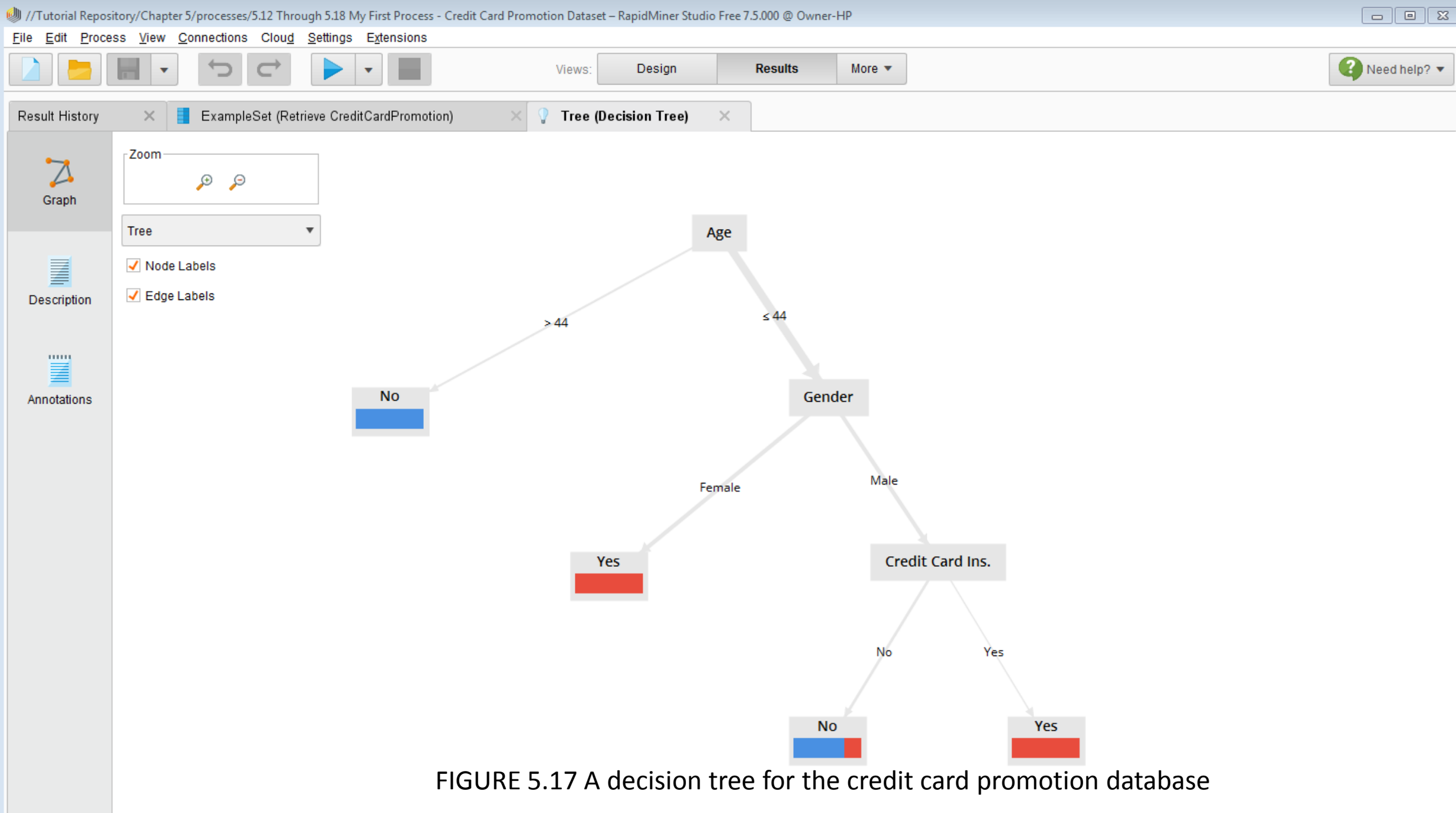


Figure 5.16 illustrates a decision tree process model within the RapidMiner Studio Community 7.0.000 interface. The interface is divided into several panels:

- Repository:** Displays a tree structure of data sources. The "Local Repository (Owner)" is expanded, showing a "data (Owner)" folder containing a "CreditCardPromotion (Owner - v1, 2)" dataset. Other repositories like "LocalBookRepository (Owner)", "NewRepositoryA (Owner)", "NewRepositoryB (Owner)", and "NewRepositoryC (Owner)" are also listed.
- Operators:** A list of available operators categorized by function. The "Blending (77)" category is selected. A search bar is provided for finding specific operators.
- Process:** The central workspace showing the workflow. It starts with an "inp" port connected to a "Retrieve CreditCard..." operator. The output of this operator is connected to a "Decision Tree" operator. The "Decision Tree" operator has four output ports: "tra", "mod", "exa", and "res". The "tra" and "mod" ports are connected to a "res" port, which then connects to a "res" port. The "exa" port is connected to a "res" port.
- Parameters:** A panel on the right showing the configuration for the "Decision Tree" operator. The parameters are:
 - criteria: gain_ratio
 - maximal depth: 20
 - apply pruning: ☒
 - confidence: 0.25
 - apply prepruning: ☒
 - minimal gain: 0.1
 - minimal leaf size: 2
 - minimal size for split: 4A link to "Hide advanced parameters" is also present.
- Help:** A panel on the right providing information about the operators. It states: "operator cannot be applied on ExampleSets with numerical attributes." and "Input: training set (Data Table)". It also notes: "This input port expects an ExampleSet. It is the output of the Retrieve operator in the attached Example Process. The

FIGURE 5.16 A decision tree process model



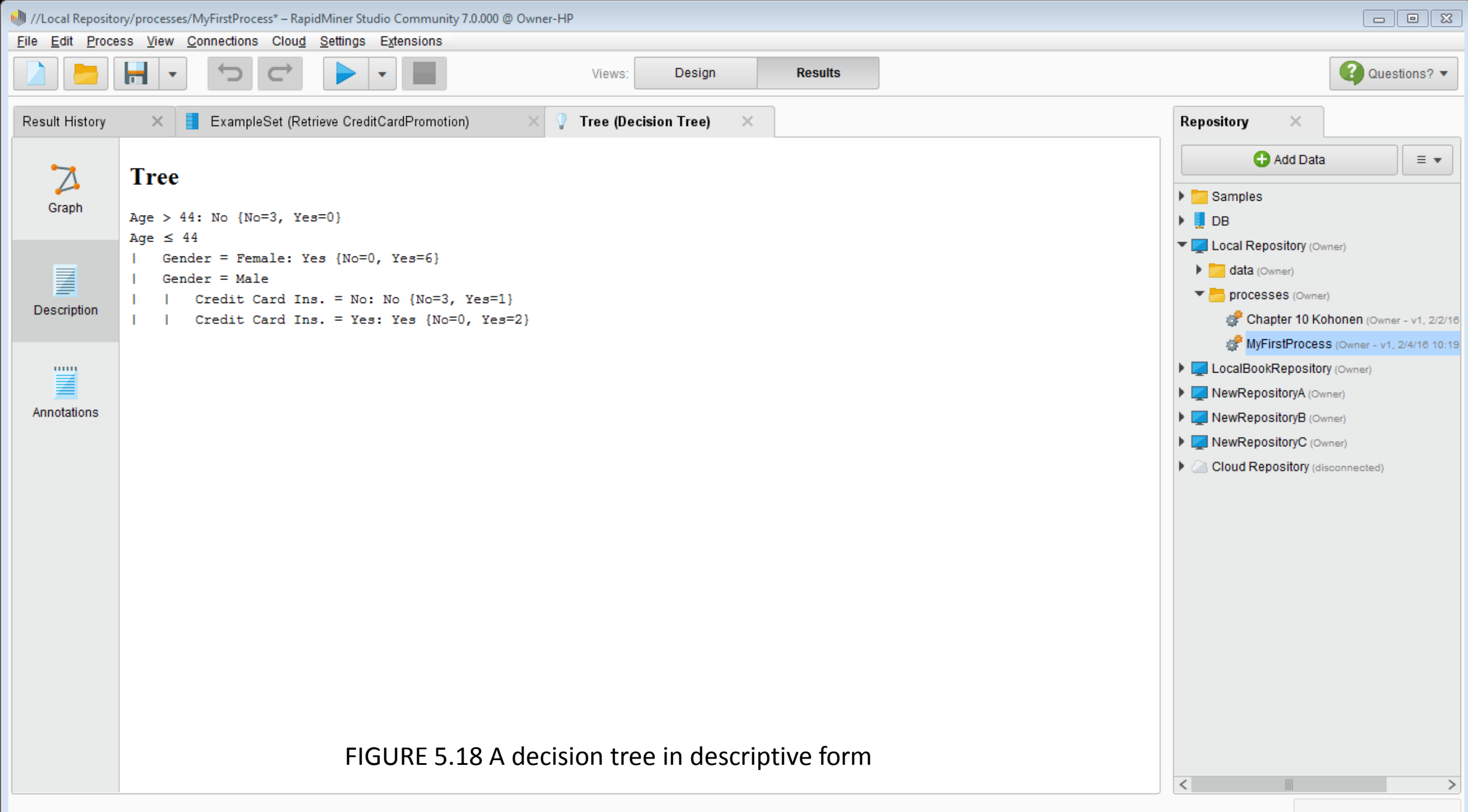


FIGURE 5.18 A decision tree in descriptive form

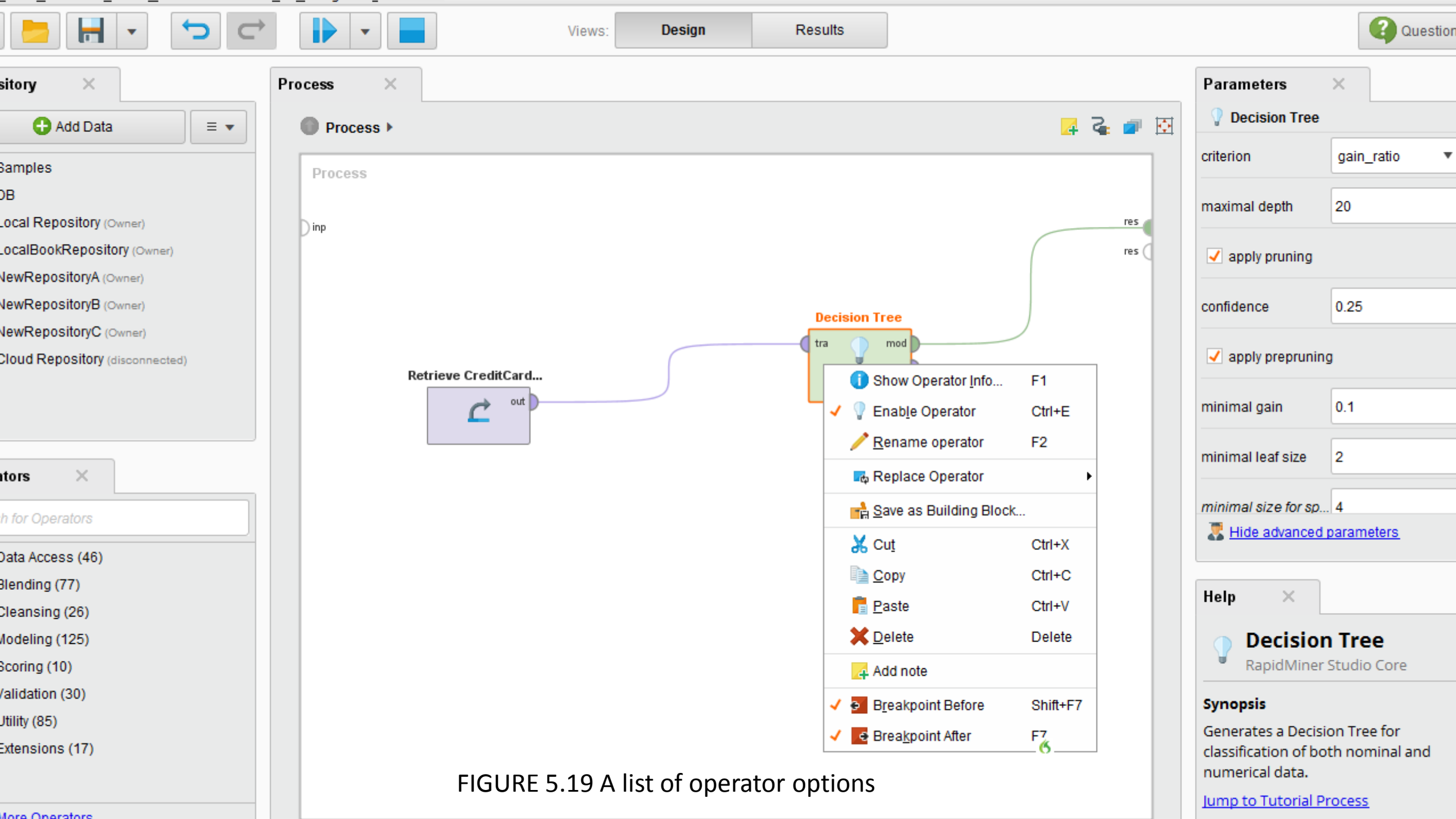


FIGURE 5.19 A list of operator options

Repository

+ Add Data

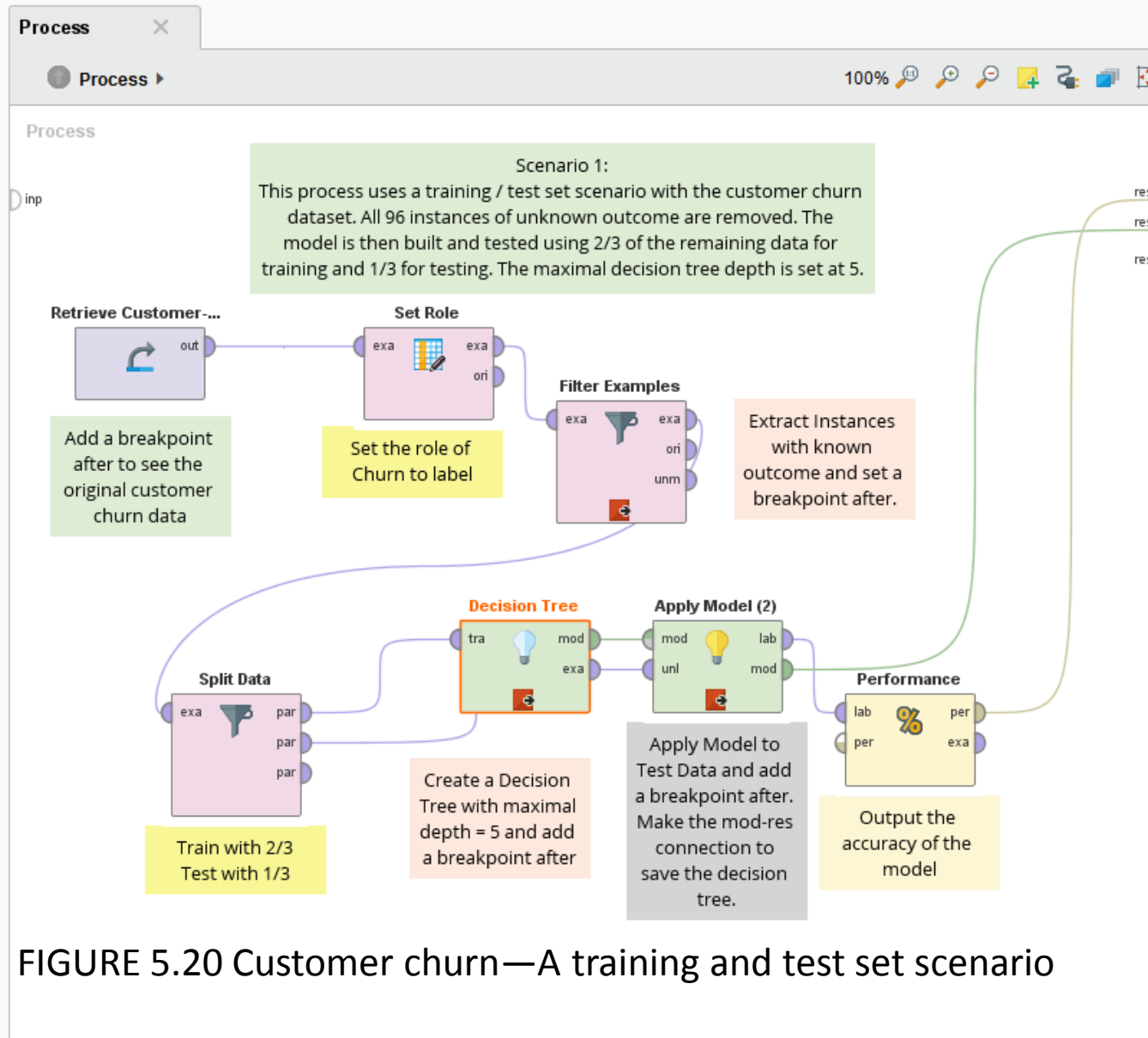
- Samples
 - DB
 - DM End Of Chapter Exercises (Owner)
 - DM Tutorials & Demos (Owner)
 - Figures (Owner)
 - Local Repository (Owner)
 - NewLocalRepository (Owner)
 - NewRepositoryA (Owner)
 - NewRepositoryBB (Owner)
 - Solutions Repository (Owner)
 - Tutorial Repository (Owner)**
 - Cloud Repository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



Parameters

Decision Tree

criterion: gain_ratio

maximal depth: 5

☒ apply pruning

confidence: 0.25

☒ apply prepruning

minimal gain: 0.1

minimal leaf size: 2

minimal size for sp...: 4

[Hide advanced parameters](#)

Help

Decision Tree
RapidMiner Studio Core

Tags: Supervised, Classification, Model, Id3, J48, J4.8, C45, C4.5, C50, C5.0, Cart, Chaid, Trees

Synopsis
Generates a Decision Tree for classification of both nominal and numerical data.

FIGURE 5.20 Customer churn—A training and test set scenario

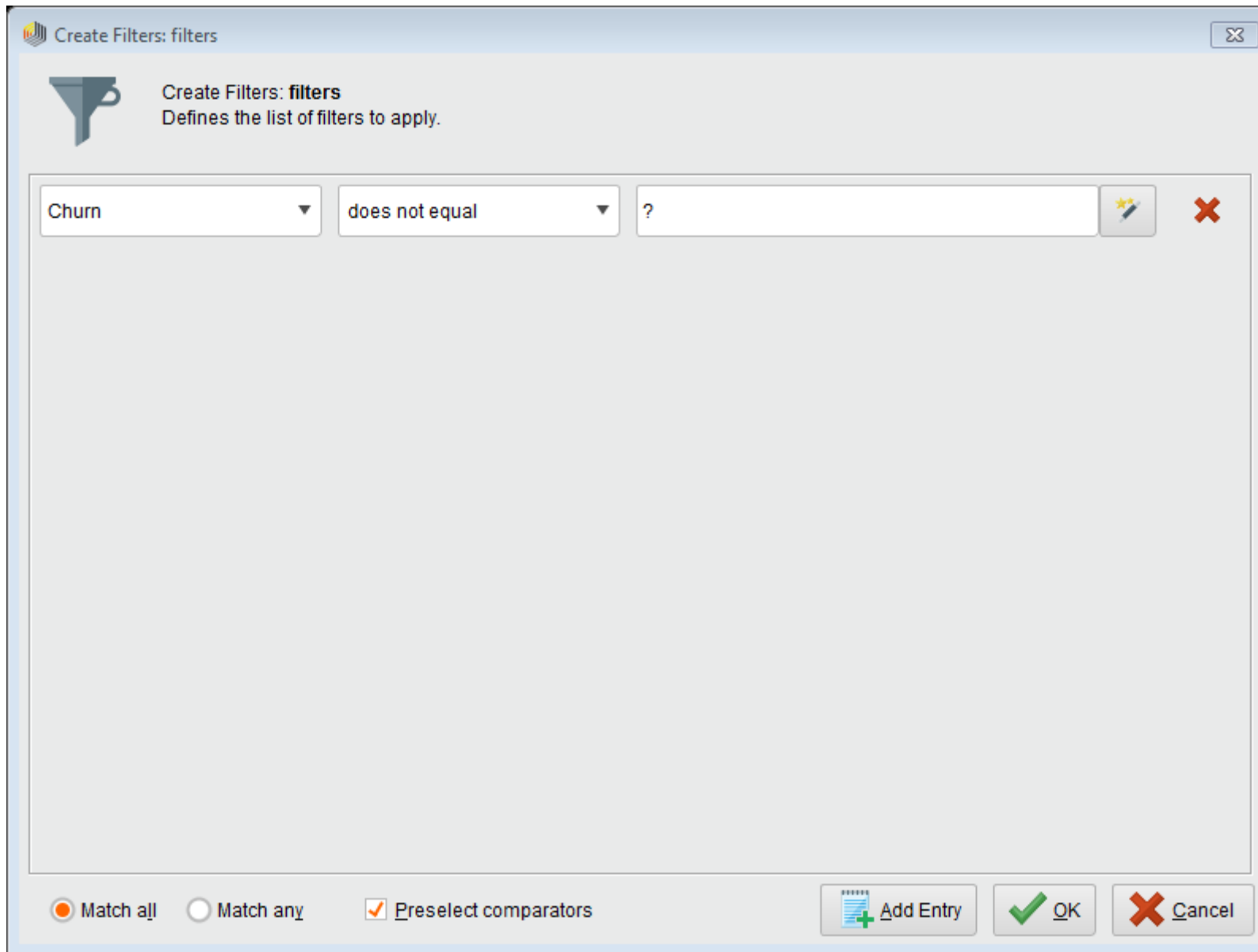



FIGURE 5.21 Removing instances of unknown outcome from the churn data set.

 Edit Parameter List: **partitions**
The partitions that should be created.

ratio
0.67
0.33





 Add Entry  Remove Entry  OK  Cancel

FIGURE 5.22 Partitioning the customer churn data

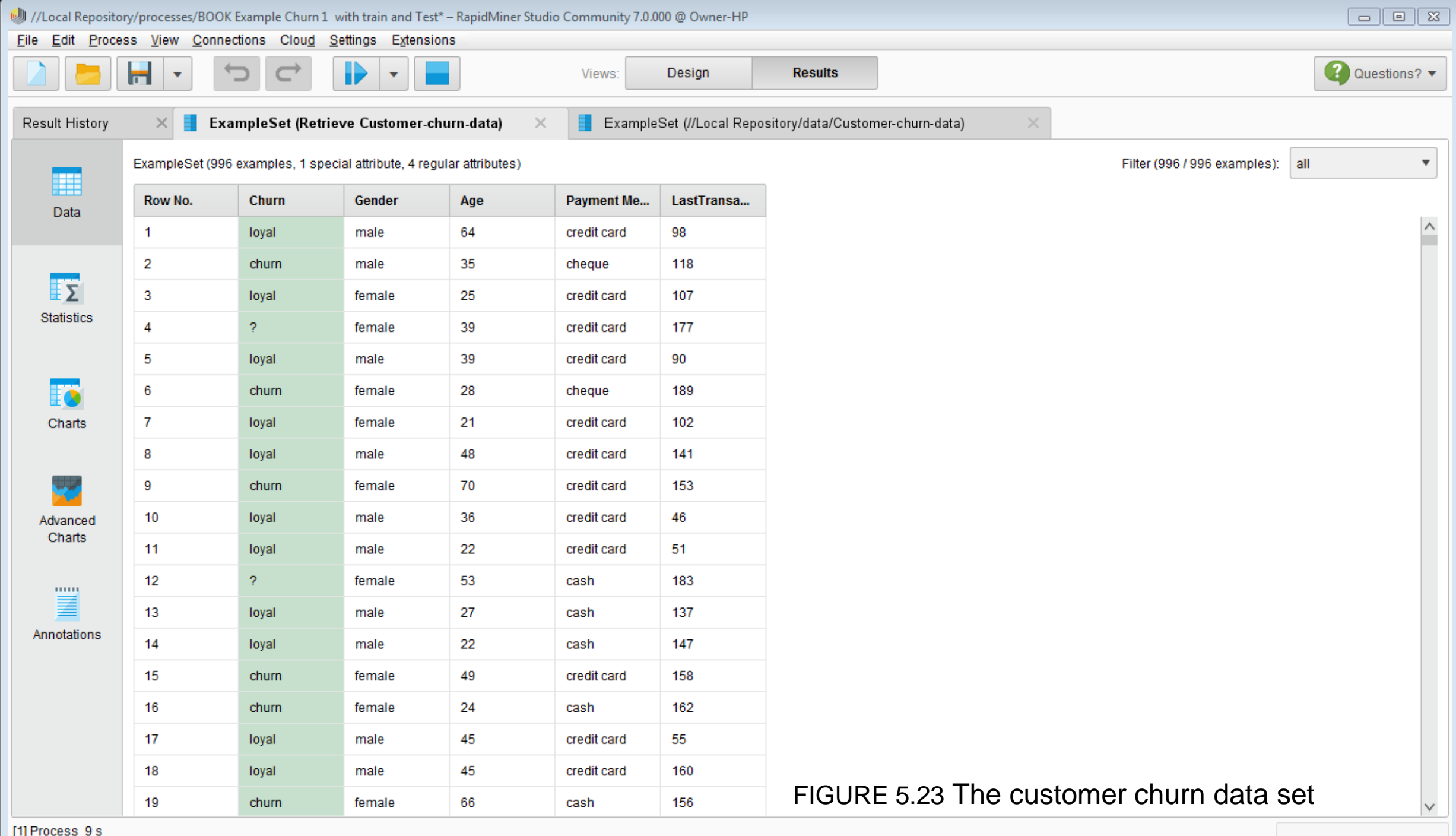
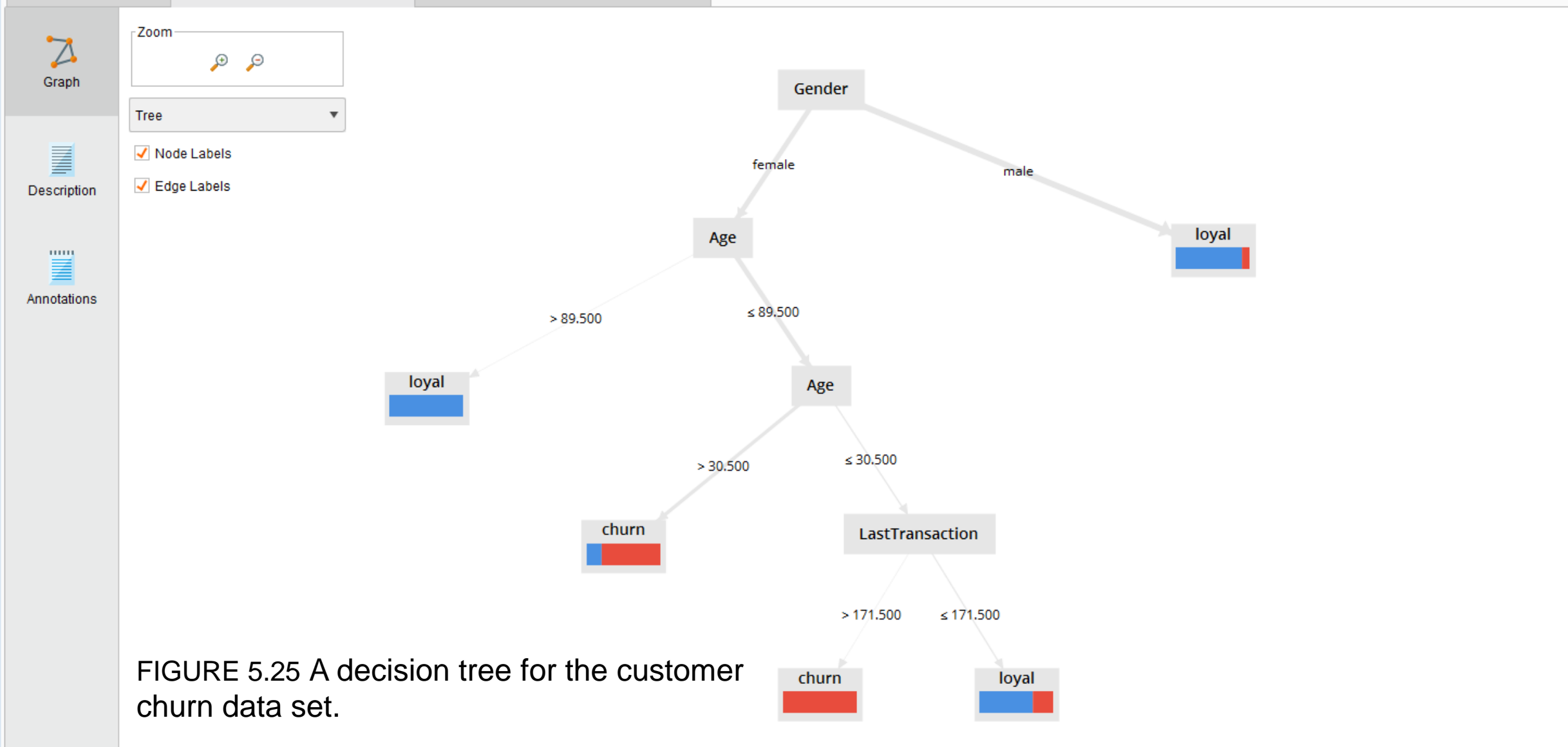


Figure 5.24 shows the RapidMiner interface displaying the results of the *Filter Examples* process. The process has successfully removed all instances of unknown outcome, leaving 900 examples. The interface shows the *ExampleSet (Filter Examples)* tab selected, displaying a table of 19 rows (labeled 1 to 19) with 6 columns: Row No., Churn, Gender, Age, Payment Me..., and LastTransa... (truncated). The *Churn* column contains values 'loyal' and 'churn'. The *Gender* column contains 'male' and 'female'. The *Age* column contains numerical values. The *Payment Me...* column contains 'credit card' and 'cash'. The *LastTransa...* column contains numerical values. The *Filter (900 / 900 examples)* dropdown menu is set to 'all'.

Figure 5.24 *Filter Examples* has removed all instances of unknown outcome



ExampleSet (297 examples, 4 special attributes, 4 regular attributes)

Filter (297 / 297 examples): all

Row No.	Churn	prediction(C...	confidence(L...	confidence(...	Gender	Age	Payment Me...	LastTransa...
1	loyal	loyal	0.899	0.101	male	64	credit card	98
2	churn	loyal	0.899	0.101	male	35	cheque	118
3	loyal	loyal	0.725	0.275	female	25	credit card	107
4	churn	churn	0	1	female	28	cheque	189
5	churn	loyal	0.725	0.275	female	24	cash	162
6	loyal	loyal	0.899	0.101	male	45	credit card	55
7	churn	churn	0.203	0.797	female	82	cash	177
8	loyal	churn	0.203	0.797	female	35	credit card	176
9	loyal	loyal	0.725	0.275	female	17	credit card	133
10	loyal	churn	0.203	0.797	female	84	cash	195
11	loyal	loyal	0.899	0.101	male	34	cheque	96
12	churn	churn	0.203	0.797	female	49	cash	188
13	loyal	loyal	0.725	0.275	female	22	credit card	72
14	churn	churn	0.203	0.797	female	37	credit card	162
15	churn	loyal	0.899	0.101	male	55	cash	126
16	loyal	churn	0.203	0.797	female	60	credit card	142
17	churn	churn	0.203	0.797	female	47	cheque	212
18	loyal	churn	0.203	0.797	female	33	credit card	30
19	loyal	churn	0.203	0.797	female	35	credit card	153

FIGURE 5.26 Output of the *Apply Model* operator

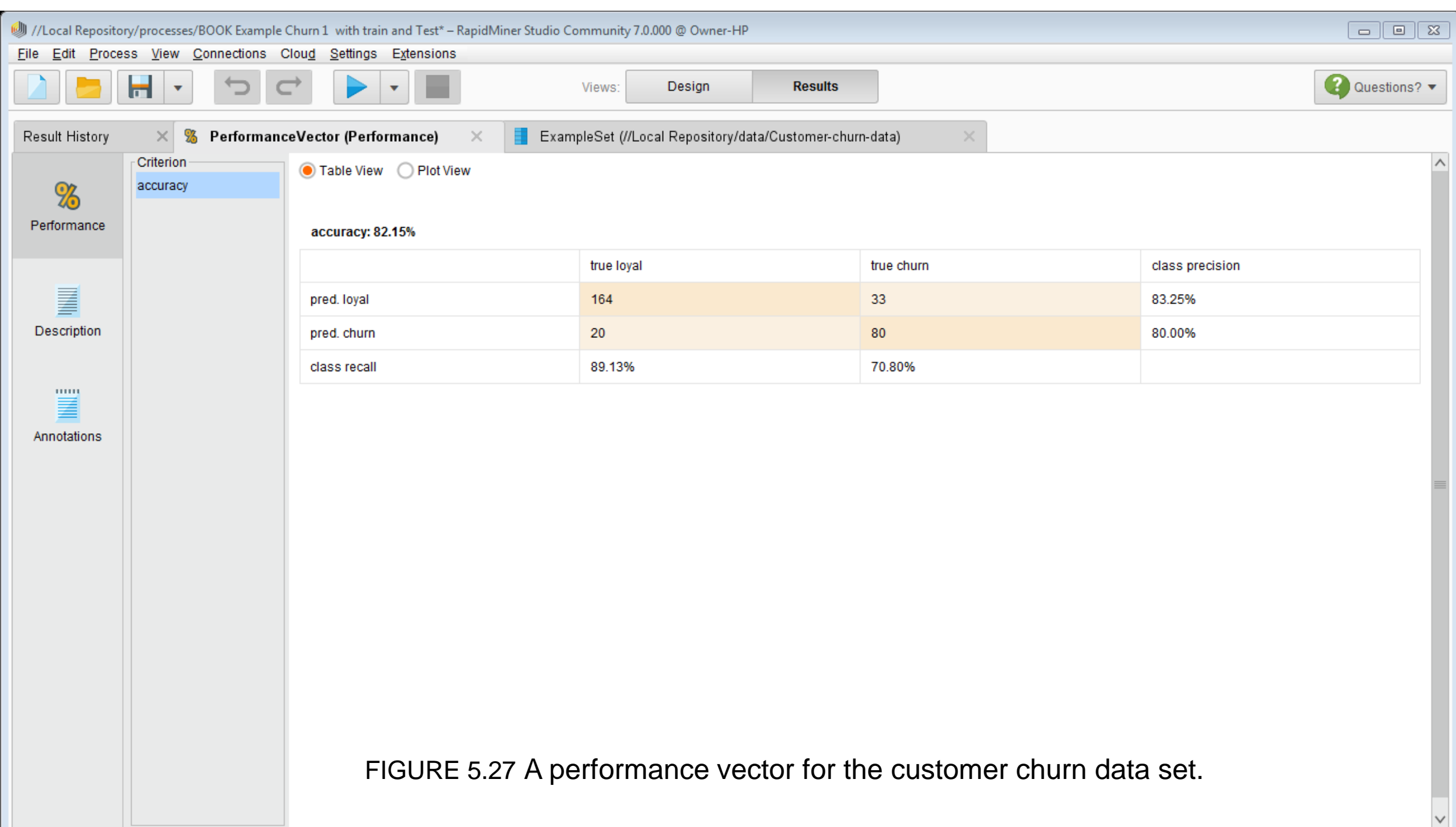


FIGURE 5.27 A performance vector for the customer churn data set.

Figure 5.28 illustrates the process of adding a subprocess to the main process window in RapidMiner Studio. The interface shows the Repository, Process, Parameters, and Help panels.

Repository: Lists various data sources, including Samples, DB, DM End Of Chapter Exercises, DM Tutorials & Demos, Figures, Local Repository, NewLocalRepository, NewRepositoryA, NewRepositoryBB, Solutions Repository, Tutorial Repository (selected), and Cloud Repository (disconnected).

Process: Displays the main process flow:

- Scenario 2:** Building a model to predict customer churn. Here we use a training / test set scenario and a subprocess to preprocess the data prior to presenting it to the decision tree operator. Maximal decision tree depth is set at 5.
- Subprocess:** Preprocess Customer Churn Data.
- Decision Tree:** Create Decision Tree with maximal depth = 5 using gain_ratio. Add a Breakpoint After.
- Apply Model (2):** Apply Model to Test Data. Apply a breakpoint after. The mod-res connection saves the Decision Tree.
- Performance:** Output the accuracy of the model.

Parameters: Configures the Decision Tree operator:

- Criterion: gain_ratio
- Maximal depth: 5
- Apply pruning: ☒
- Confidence: 0.25
- Apply prepruning: ☒
- Minimal gain: 0.1
- Minimal leaf size: 2
- Minimal size for sp...: 4

Help: Provides information about the Decision Tree operator, including tags (Supervised, Classification, Model, Id3, J48, J4.8, C45, C4.5, C50, C5.0, Cart, Chaid, Trees) and a synopsis: Generates a Decision Tree for classification of both nominal and numerical data.

FIGURE 5.28 Adding a subprocess to the main process window

Figure 5.29 displays a screenshot of the RapidMiner Studio interface, showing a subprocess for data preprocessing.

The interface includes a top menu bar (File, Edit, Process, View, Connections, Cloud, Settings, Extensions), a toolbar with icons for file operations and execution, and a status bar indicating the current view (Design or Results).

The main workspace is divided into several panels:

- Repository:** Lists data sources such as Samples, DB, Local Repository (Owner), LocalBookRepository (Owner), NewRepositoryA (Owner), NewRepositoryB (Owner), Repository for Chapter 5 (Owner), and Cloud Repository (disconnected).
- Operators:** A search bar and a list of operator categories (Data Access (46), Blending (77), Cleansing (26), Modeling (125), Scoring (10), Validation (30), Utility (85), Extensions (271)).
- Process:** The central area showing the subprocess flowchart.
- Parameters:** A panel for configuring the selected operator.
- Help:** A panel providing documentation for the selected operator.

The subprocess flowchart, titled "Subprocess", illustrates the data preprocessing steps:

- Retrieve Customer...** (Input operator) connects to the **Set Role** operator.
- Set Role** (Operator) outputs to the **Filter Examples** operator.
- Filter Examples** (Operator) outputs to the **Split Data** operator.
- Split Data** (Operator) outputs to three separate output ports labeled "out".

The **Filter Examples** operator is configured with the role "Remove Unknown Instances". The **Split Data** operator is configured with the role "2/3 Train 1/3 Test".

The **Help** panel for the **Split Data** operator provides the following synopsis:

Split Data
RapidMiner Studio Core

Synopsis
This operator produces the desired number of subsets of the given ExampleSet. The ExampleSet is partitioned into subsets according to the specified relative sizes.

[Jump to Tutorial Process](#)

FIGURE 5.29 A subprocess for data preprocessing

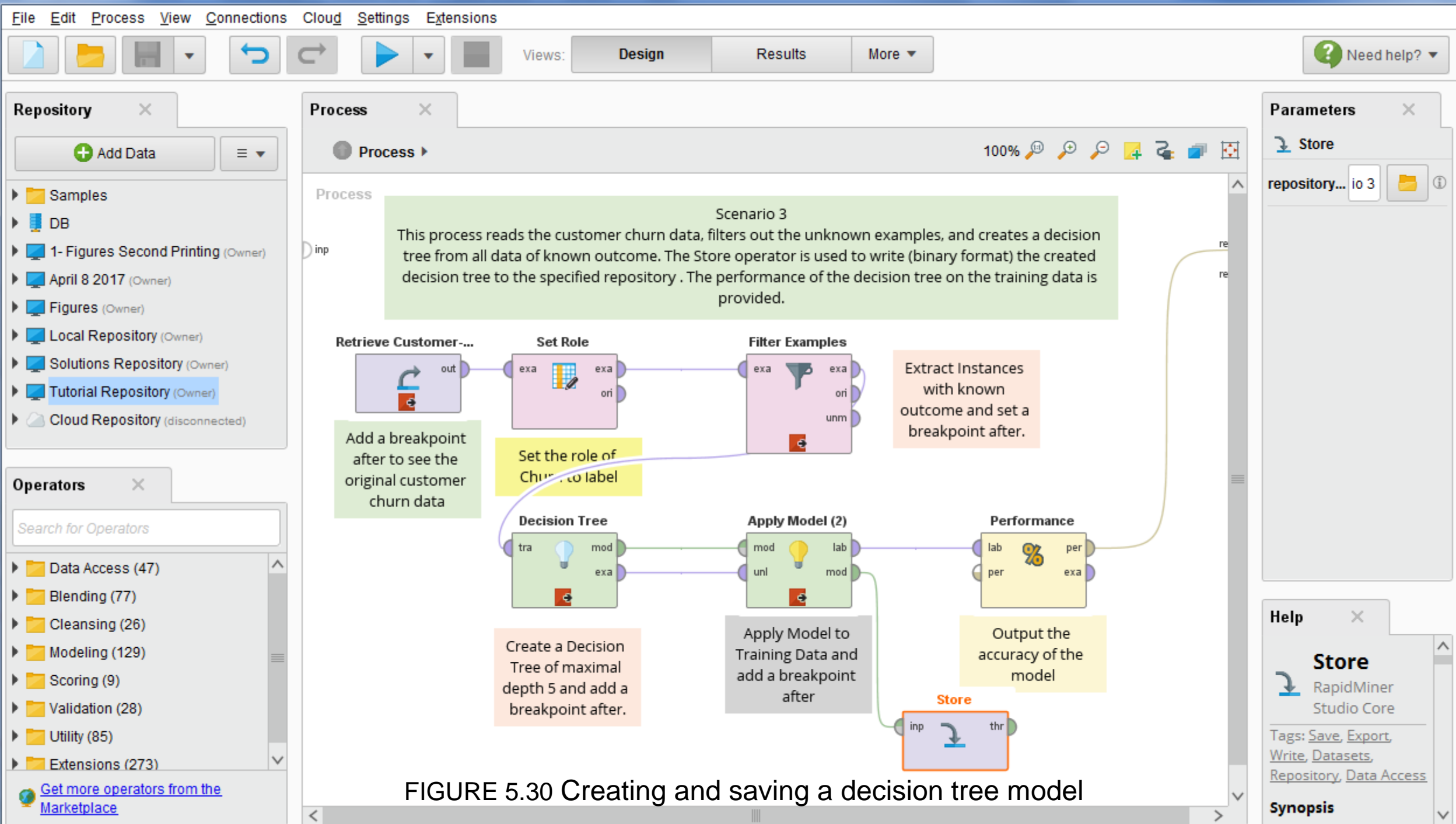


FIGURE 5.30 Creating and saving a decision tree model

//Tutorial Repository/Chapter 5/processes/5.30 Scenario 3 Creating and saving a decision tree using all known instances - customer churn – RapidMiner Studio Free 7.4.000 @ Owner-HP

File Edit Process View Connections Cloud Settings Extensions

Views: Design Results More

Questions?

Repository

+ Add Data

- Solutions Repository (Owner)
- Tutorial Repository (Owner)
 - Chapter 5 (Owner)
 - data (Owner)
 - processes (Owner)
 - Chapter 6 (Owner)
 - Chapter 7 (Owner)
 - Chapter 10 (Owner)
 - Chapter 11 (Owner)
 - Chapter 12 (Owner)
 - Chapter 13 (Owner)
- Cloud Repository (disconnected)

Operators

Search for Operators

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

Process

Process

100%

inp

Scenario 3: continued

This process reads the customer churn data set, filters out all examples of known outcome, uses the Retrieve operator to read the previously created decision tree model (Figure 5.30) and applies the tree model to the instances of unknown outcome. The output is then written to an Excel file. Process 5.30 must first be executed to create the file (Saved Model Scenario 3) containing the decision tree.

Retrieve the decision tree created by the model given in Figure 5.30.

Retrieve

out

Apply Model (2)

mod unl lab mod

Write Excel

inp thr fil

Write the output to an Excel file.

Retrieve Customer-...

out

Retrieve the original customer churn data

Filter Examples

exa ori unkn

Extract Instances with unknown outcome and set a breakpoint after.

Apply Model to the instances whose outcome is to be determined

res

res

Parameters

Process

logverbosity init

logfile

resultfile

random seed 2001

send mail never

encoding SYSTEM

[Hide advanced parameters](#)

[Change compatibility \(7.4.000\)](#)

Help

Process

RapidMiner Studio Core

Synopsis

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

Description

Each process must contain exactly one instance of this class and it must be

FIGURE 5.31 Reading and applying a saved model

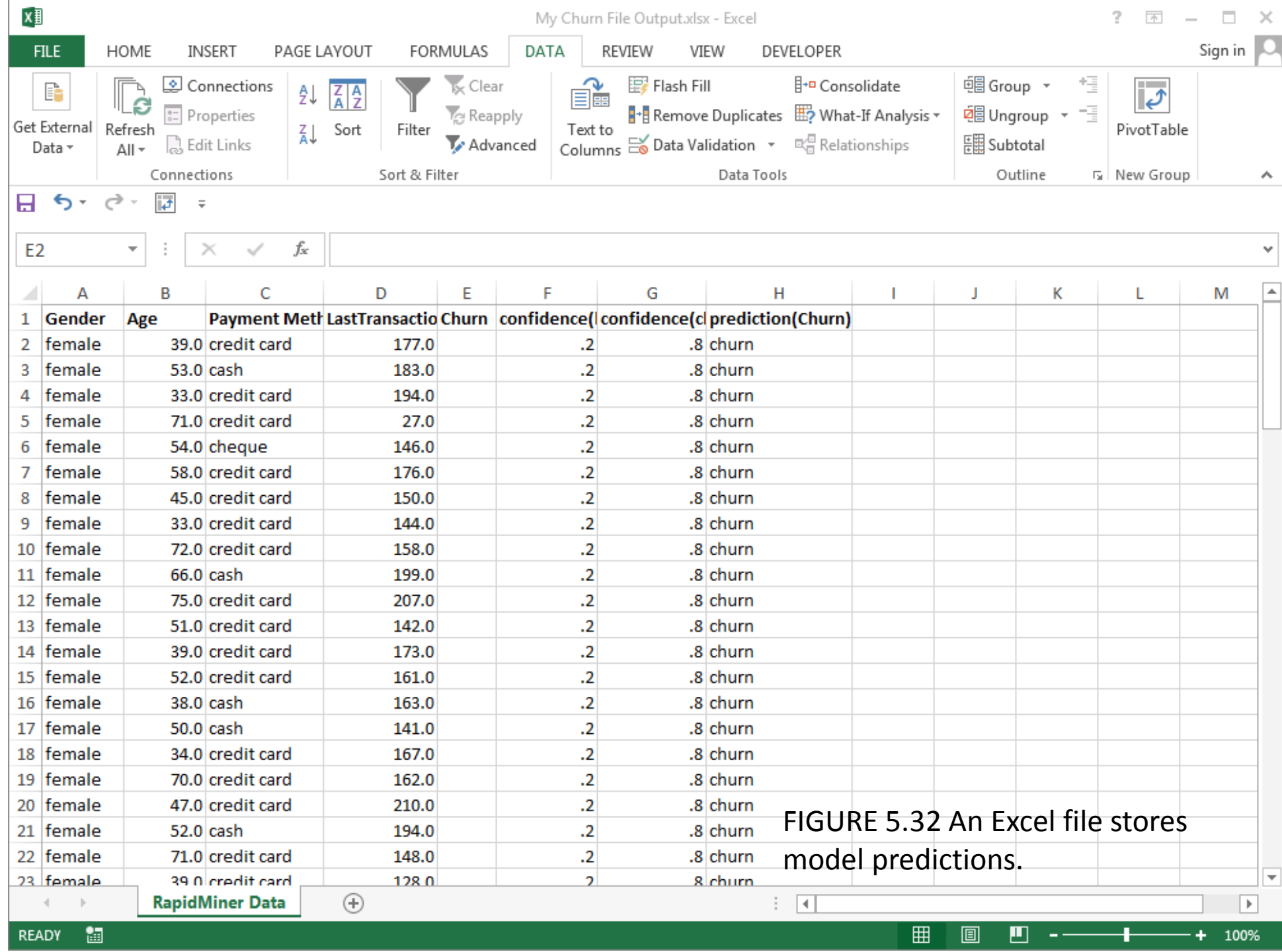


FIGURE 5.32 An Excel file stores model predictions.

//Tutorial Repository/Chapter 5/processes/5.33 Through 5.36 Scenario 4 Cross Validation Customer Churn Data – RapidMiner Studio Free 7.3.000 @ Owner-HP

File Edit Process View Connections Cloud Settings Extensions

Views: Design Results More

Repository

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

Get more operators from the Marketplace

Process

Process

Scenario 4

Here we use the customer churn data together with 10-fold cross validation to obtain a measure of decision tree model accuracy. The Cross Validation operator implements the 10-fold cross validation. Cross Validation is a nested operator containing two subprocesses. One subprocess is for model training and the second for testing.

PreProcess

Preprocess Customer Churn Data

Cross Validation

Cross Validation 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

Synopsis

This operator performs a cross-validation in

FIGURE 5.33 Testing a model using cross-validation

Figure 5.34 shows a screenshot of the RapidMiner Studio interface, illustrating a subprocess designed to read and filter customer churn data.

The interface displays the following components:

- Repository:** Lists data sources including Samples, DB, Local Repository (Owner), LocalBookRepository (Owner), NewRepositoryA (Owner), NewRepositoryB (Owner), Repository for Chapter 5 (Owner), and Cloud Repository (disconnected).
- Process:** Shows the workflow within the PreProcess subprocess:
 - Retrieve Customer-...:** Retrieves data from the repository.
 - Set Role:** Configures the role of the 'Churn' attribute as the label.
 - Filter Examples:** Removes unknown instances from the dataset.
- Parameters:** Displays the configuration for the PreProcess (Subprocess).
- Help:** Provides documentation for the Subprocess operator, stating: "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)".

The status bar at the bottom indicates the execution progress: [1] Process 1:36, [1] PreProcess 1:36.

Figure 5.35 illustrates the nested subprocesses for cross-validation in RapidMiner Studio Free 7.3.000. The interface shows the **Process** view with a workflow titled **Cross Validation**.

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

Training: A **Decision Tree** operator is used to train the model. A yellow box notes: "A decision tree using gain_ratio with a maximum depth of 5".

Testing: The trained model is applied using the **Apply Model** operator. A yellow box notes: "Apply the model to one of the 10 partitions." The output of the **Apply Model** operator is then evaluated using the **Performance** operator. A yellow box notes: "Output the accuracy of the model."

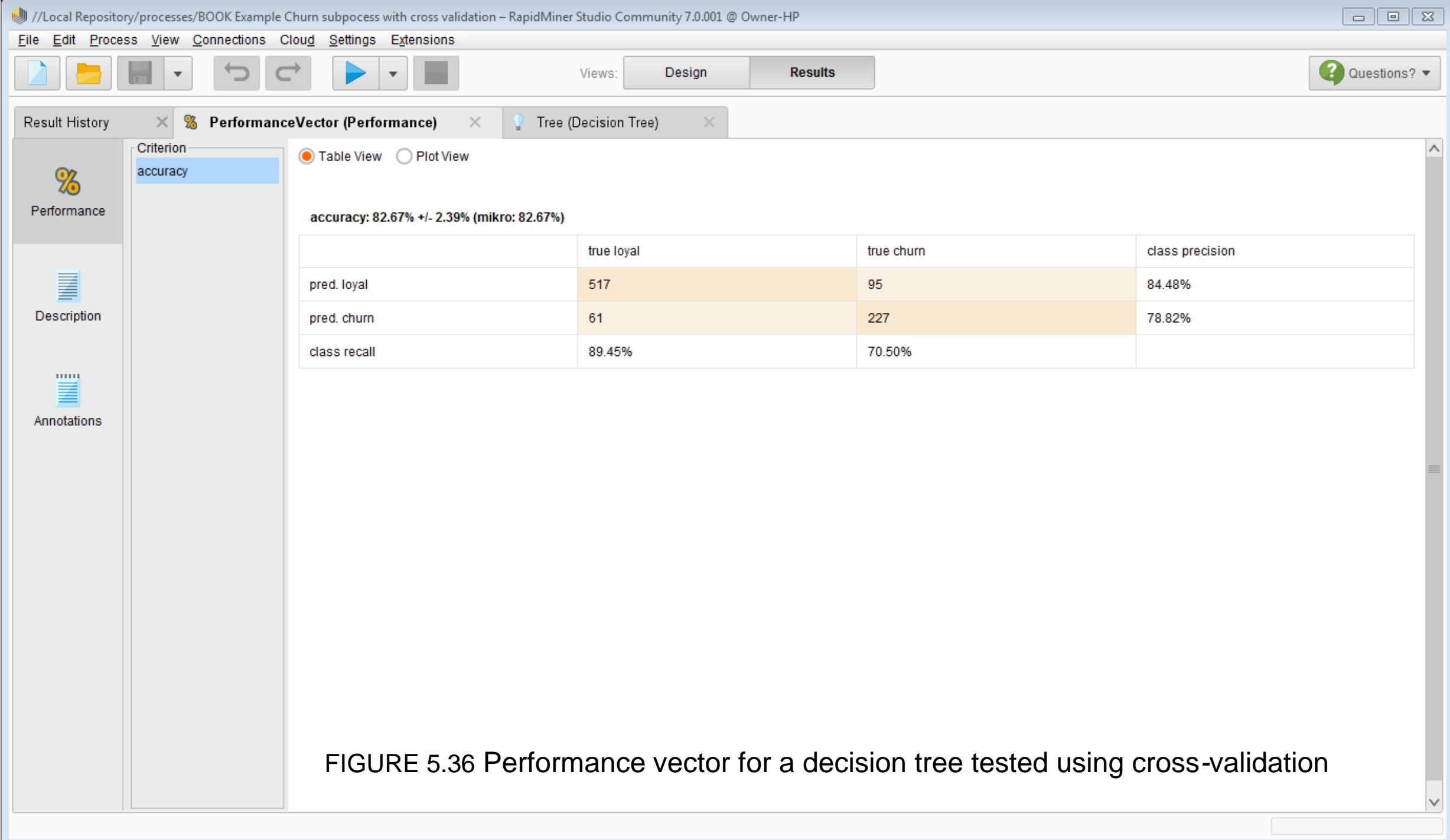
The **Parameters** panel on the right shows the configuration for the **Decision Tree** operator:

- criterion:** gain_ratio
- maximal depth:** 5
- apply pruning:** ☒
- confidence:** 0.25
- apply prepruning:** ☒
- minimal gain:** 0.1
- minimal leaf size:** 2
- minimal size for sp...:** 4

The **Help** panel on the right provides additional information about the **Decision Tree** operator:

- Decision Tree** (RapidMiner Studio Core)
- Tags:** Supervised, Classification, Model, Id3, J48, J4.8, C45, C4.5, C50, C5.0, Cart, Chaid, Trees
- Synopsis:** Generates a Decision Tree for classification of both nominal and numerical data.

FIGURE 5.35 Nested subprocesses for cross-validation



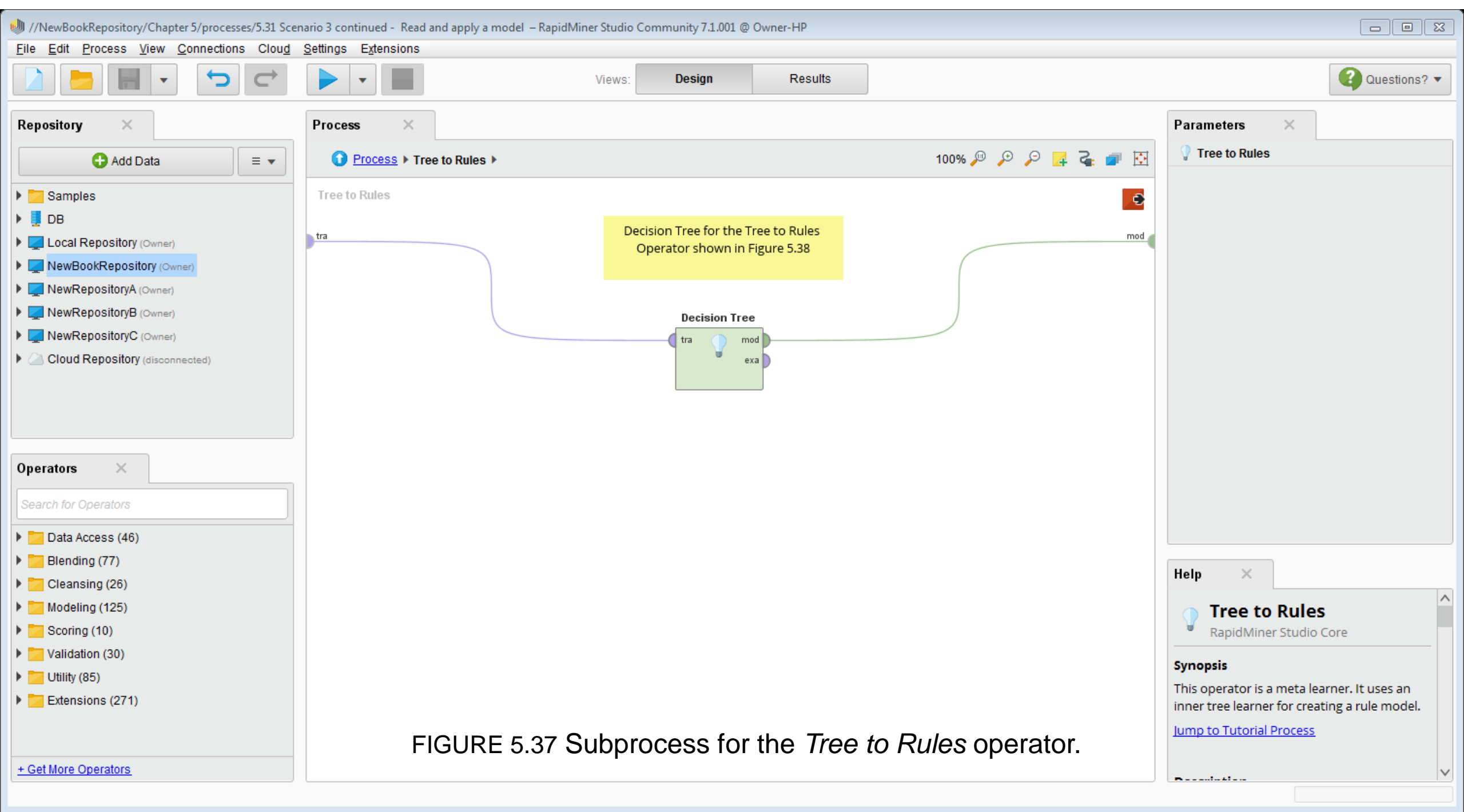
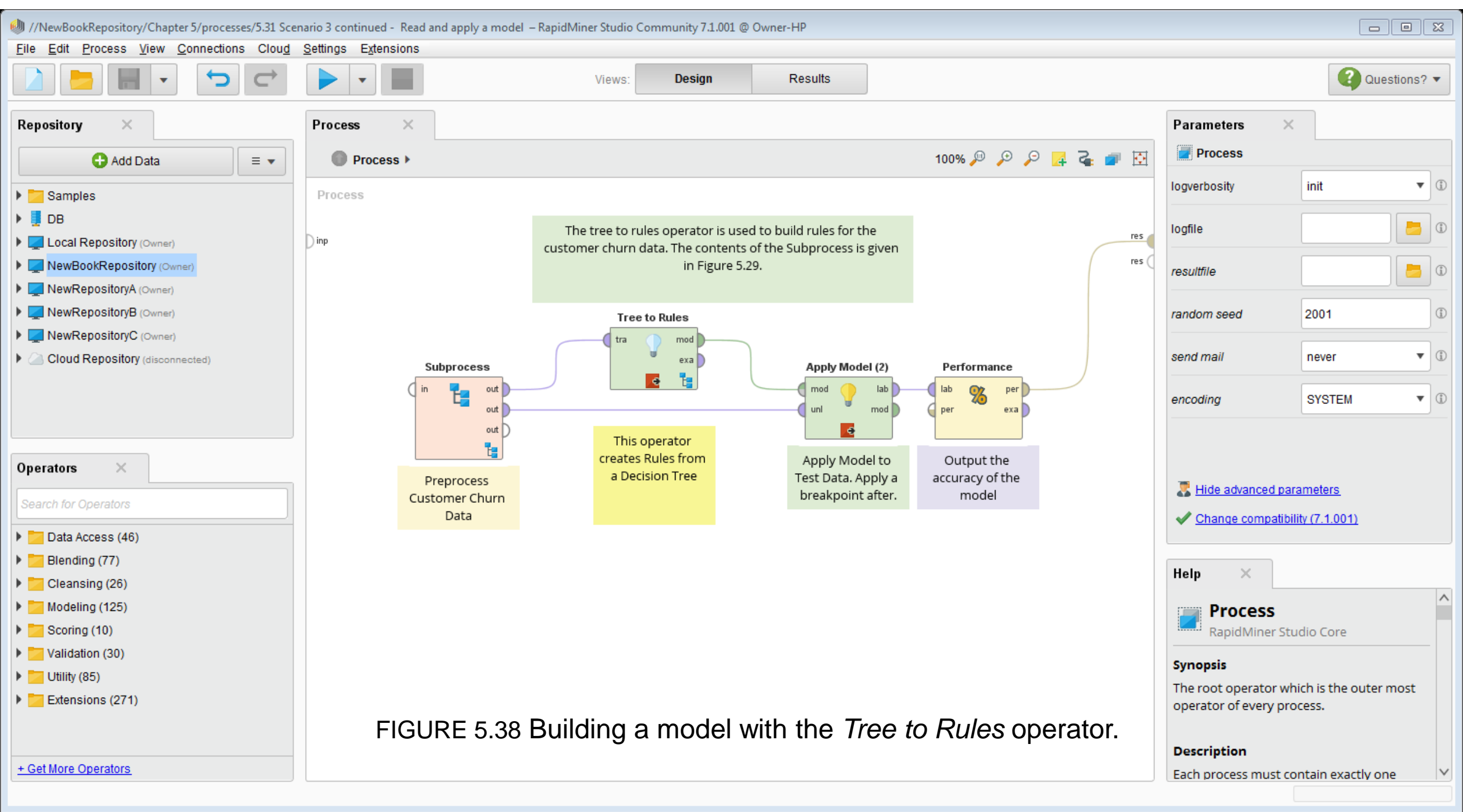


FIGURE 5.37 Subprocess for the *Tree to Rules* operator.



Local Repository/processes/BOOK Example Churn tree to rules – RapidMiner Studio Community 7.0.001 @ Owner-HP

File Edit Process View Connections Cloud Settings Extensions

Views: Design Results

Questions?

Result History RuleModel (Tree to Rules)

RuleModel

Description

```
if Gender = female and Age > 89.500 then loyal (2 / 0)
if Gender = female and Age ≤ 89.500 and Age > 30.500 then churn (39 / 153)
if Gender = female and Age ≤ 89.500 and Age ≤ 30.500 and LastTransaction > 171.500 then churn (0 / 3)
if Gender = female and Age ≤ 89.500 and Age ≤ 30.500 and LastTransaction ≤ 171.500 and Payment Method = cash then churn (2 / 9)
if Gender = female and Age ≤ 89.500 and Age ≤ 30.500 and LastTransaction ≤ 171.500 and Payment Method = cheque then churn (1 / 2)
if Gender = female and Age ≤ 89.500 and Age ≤ 30.500 and LastTransaction ≤ 171.500 and Payment Method = credit card then loyal (47 / 8)
if Gender = male then loyal (303 / 34)
```

Annotations

correct: 519 out of 603 training examples.

[1] Process 11 s

FIGURE 5.39 Rules generated by the *Tree to Rules* operator.

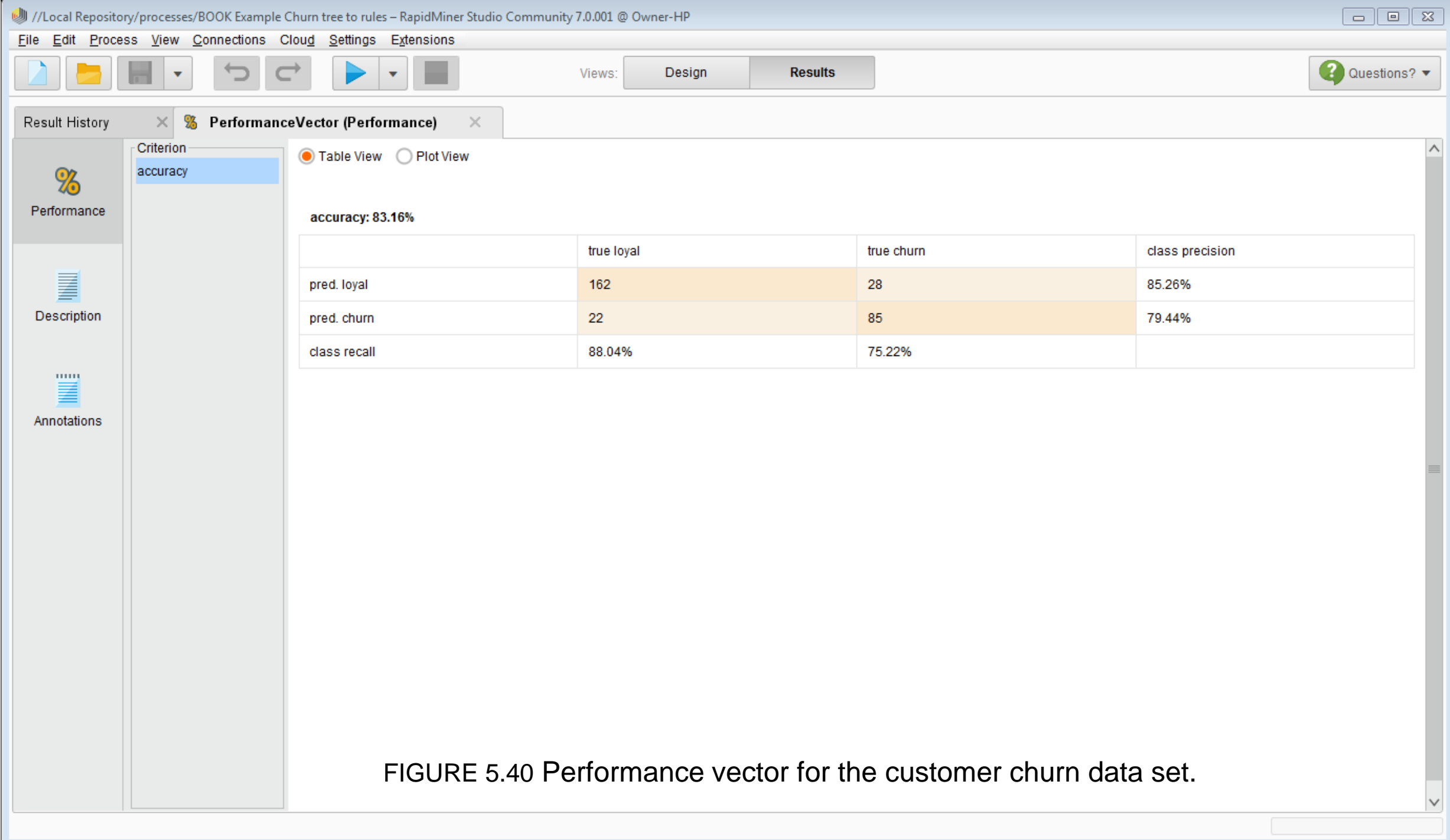


FIGURE 5.40 Performance vector for the customer churn data set.

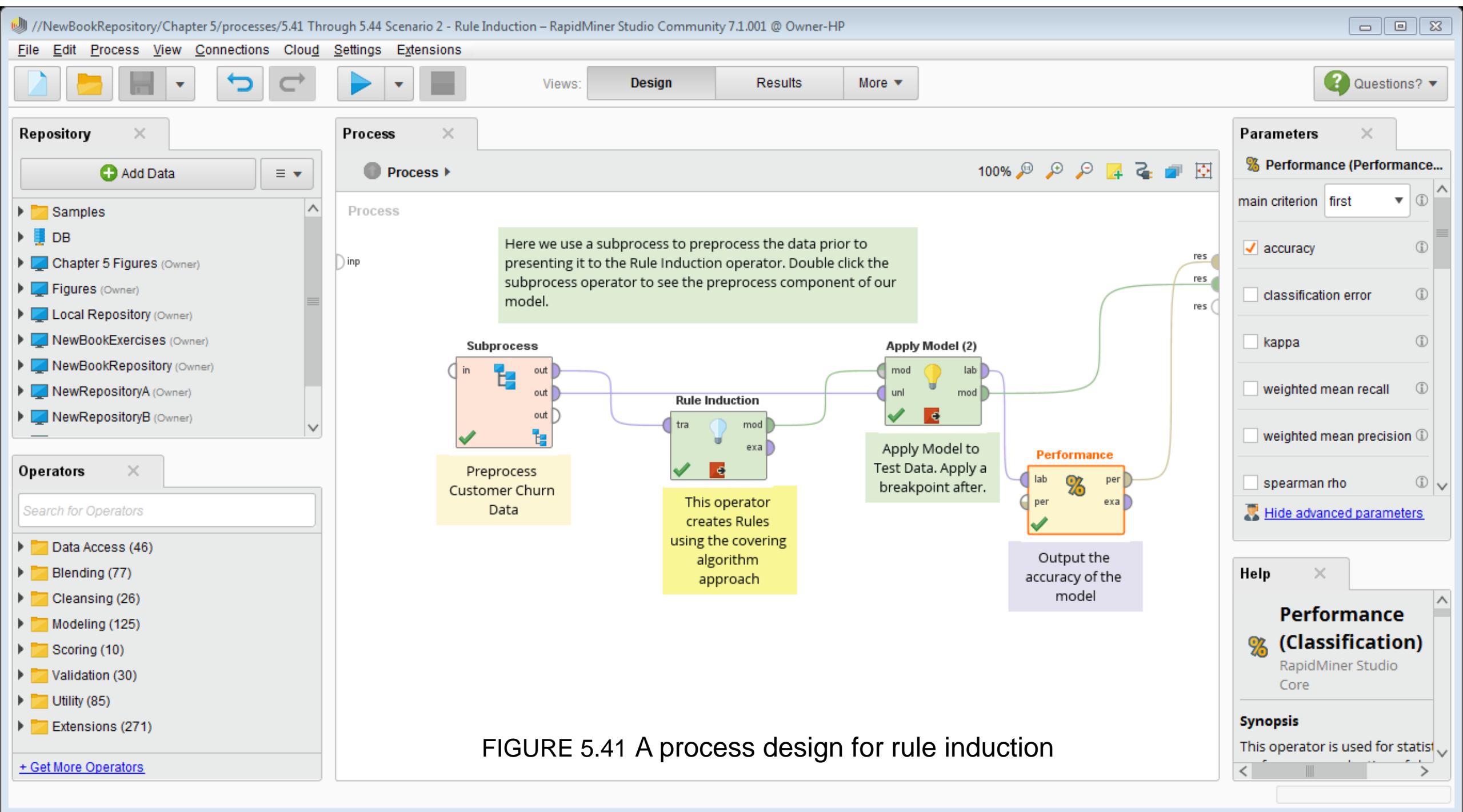
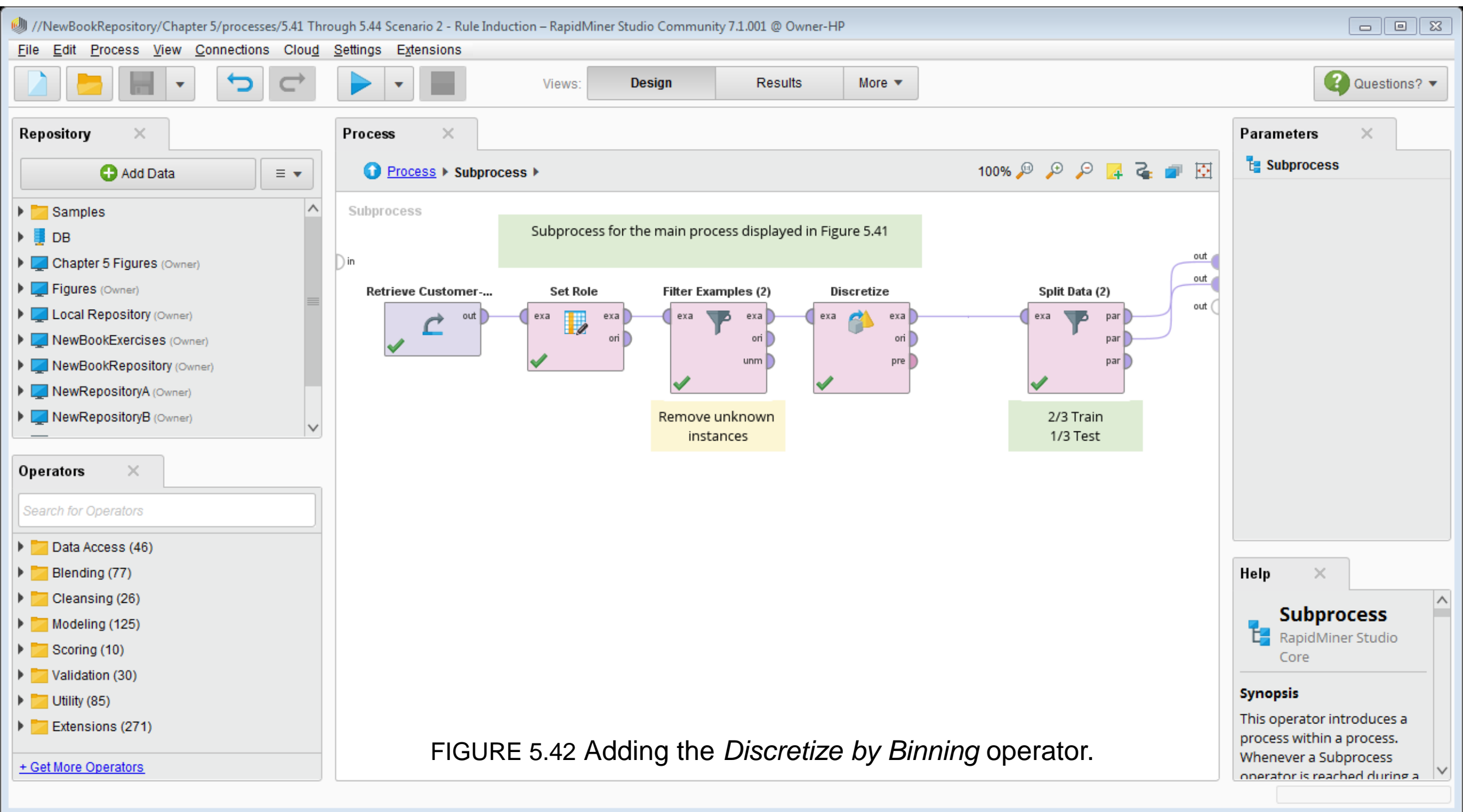


FIGURE 5.41 A process design for rule induction



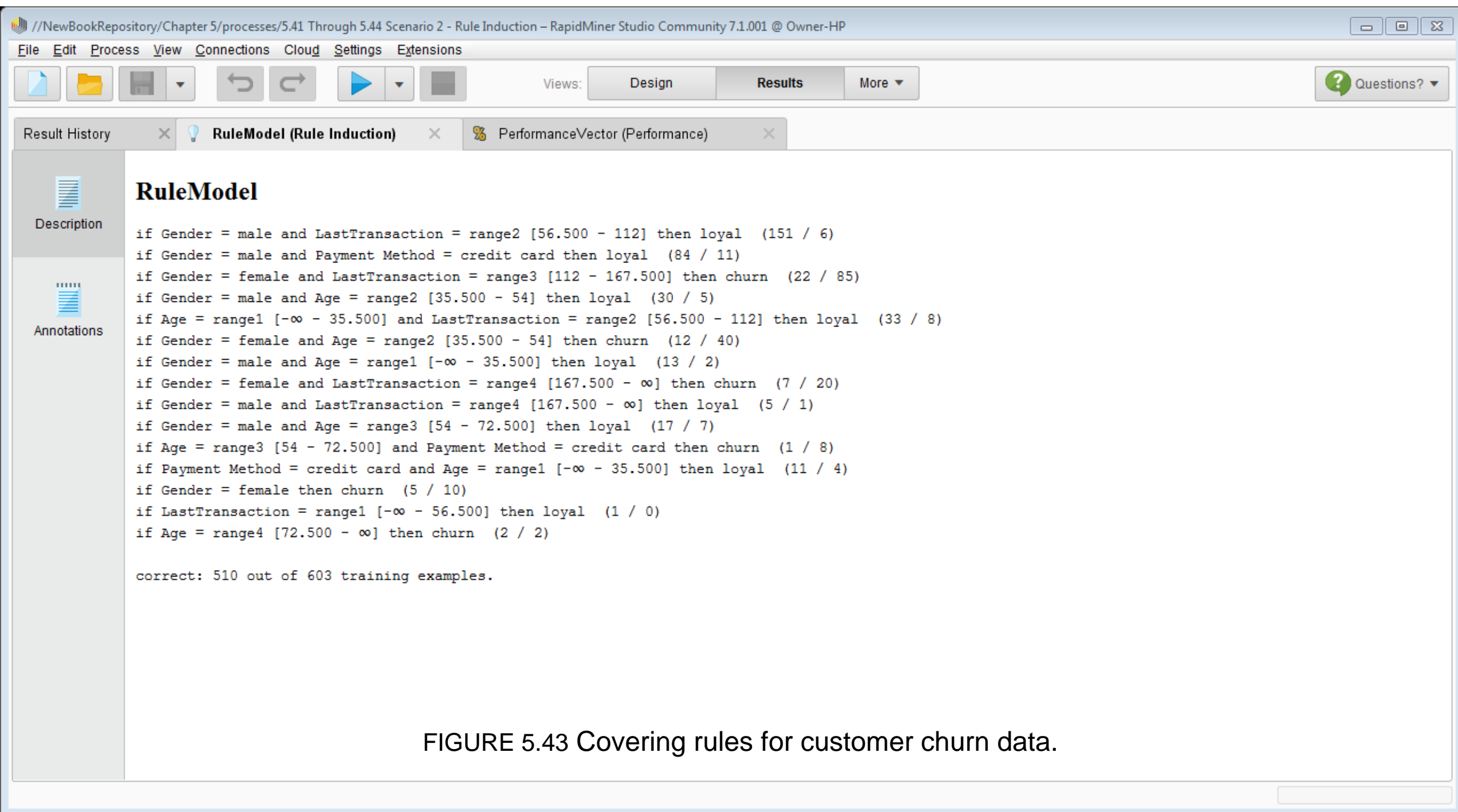


FIGURE 5.43 Covering rules for customer churn data.

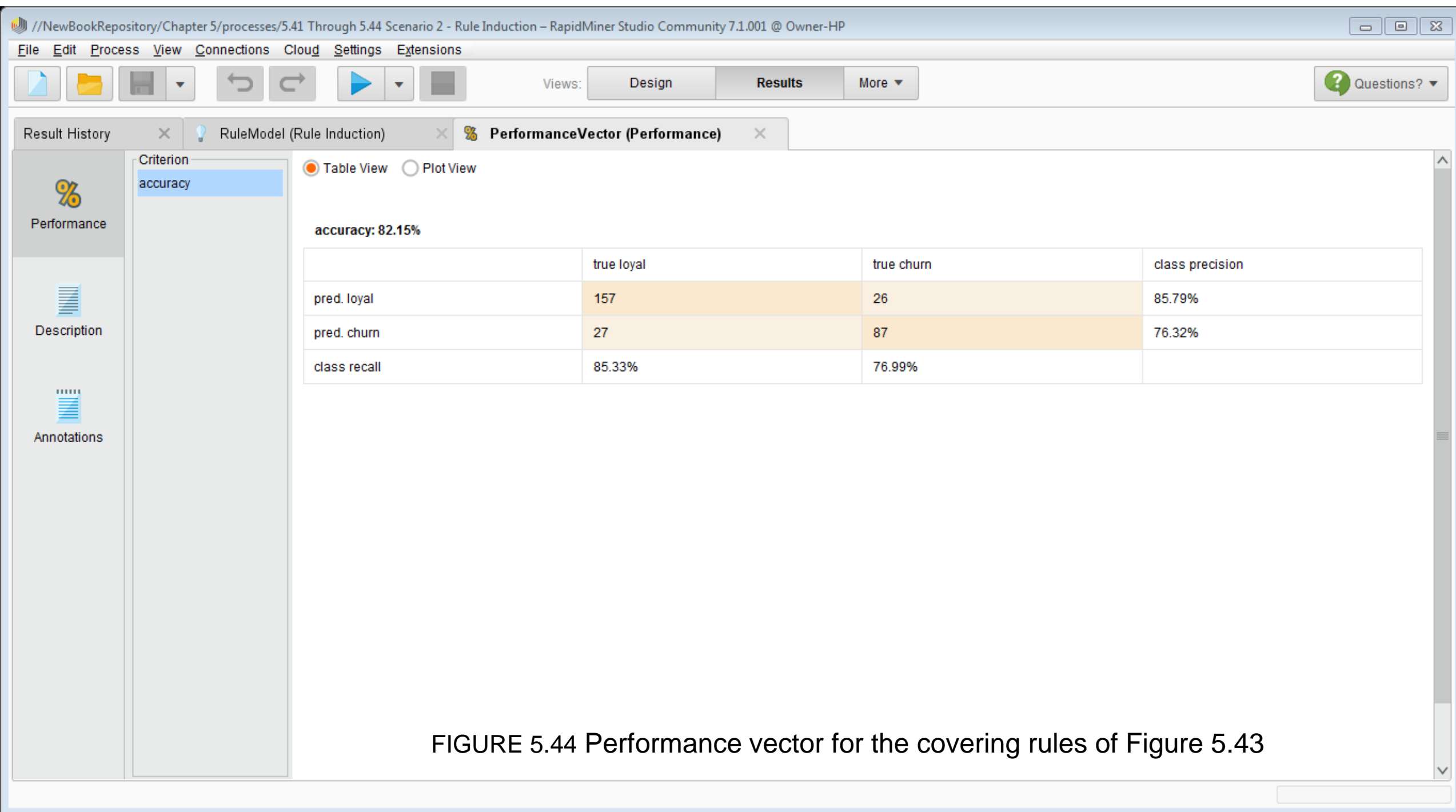


Figure 5.45 illustrates the process design for subgroup discovery in RapidMiner Studio. The interface shows the Repository, Process, Operators, Parameters, and Help panels.

Repository Panel: Lists data sources including Samples, DB, Local Repository (Owner), LocalBookRepository (Owner), NewRepositoryA (Owner), NewRepositoryB (Owner), Repository for Chapter 5 (Owner), and Cloud Repository (disconnected).

Process Panel: Displays the workflow design. The process starts with an input (inp) leading into a Subprocess. The Subprocess is labeled "Preprocess Customer Churn Data" and contains a green box explaining: "Here we use the Subgroup Discovery operator with the customer churn dataset. A subprocess is used for data preprocessing." The output of the Subprocess (out) connects to the Subgroup Discovery operator. The Subgroup Discovery operator is labeled "Subgroup Discovery" and has a yellow box explaining: "This operator creates Rules using the Subgroup Discovery Operator". The output of the Subgroup Discovery operator (mod) connects to the final output (res).

Parameters Panel: Shows the configuration for the Subgroup Discovery operator:

- mode: k best rules
- utility function: WRAcc
- min utility: 0.0
- k best rules: 10
- rule generation: both
- max depth: 5
- min coverage: 0.0
- max cache: -1

Help Panel: Provides a synopsis of the Subgroup Discovery operator:

Subgroup Discovery
RapidMiner Studio Core

Synopsis
This operator performs an exhaustive subgroup discovery. The goal of subgroup discovery is to find rules describing subsets of the population that are sufficiently large and statistically

FIGURE 5.45 Process design for subgroup discovery.

Figure 5.46 displays the Subprocess design for subgroup discovery in RapidMiner Studio Community 7.0.001. The interface shows the Process Designer with a Subprocess design.

The Subprocess design consists of the following operators and connections:

- Retrieve Customer...** (Data Access) connects to **Set Role** (Data Access).
- Set Role** (Data Access) connects to **Filter Examples (2)** (Filtering).
- Filter Examples (2)** (Filtering) connects to **Discretize** (Modeling).
- Discretize** (Modeling) connects to the **out** output port.

The **Filter Examples (2)** operator has a yellow box labeled "Remove unknown instances" below it.

The interface also includes the Repository panel on the left, the Operators panel on the bottom left, the Parameters panel on the right, and the Help panel on the bottom right.

FIGURE 5.46 Subprocess design for subgroup discovery.



Views:

Design

Results

Questions? ▾

Result History

RuleSet (Subgroup Discovery)



Data



Description



Annotations

Premise	Conclusion	Pos	Neg	Size	Coverage	Precision	Accuracy	Bias	Lift	Bi...
Gender=female	churn	268	139	407	0.452	0.658	0.786	0.3...	1.840	0....	0...
Gender=male	loyal	54	439	493	0.548	0.890	0.786	0.2...	1.387	0....	0...
Gender=male , Payment Method=credit card	loyal	18	297	315	0.350	0.943	0.668	0.3...	1.468	0....	0...
LastTransaction=range2 [56.500 - 112] , Gender=male	loyal	11	221	232	0.258	0.953	0.591	0.3...	1.483	0....	0...
LastTransaction=range3 [112 - 167.500] , Gender=female	churn	130	38	168	0.187	0.774	0.744	0.4...	2.163	0....	0...
Age=range2 [35.500 - 54] , Gender=female	churn	113	27	140	0.156	0.807	0.738	0.4...	2.256	0....	0...
Gender=female , Payment Method=credit card	churn	157	108	265	0.294	0.592	0.697	0.2...	1.656	0....	0...
LastTransaction=range2 [56.500 - 112] , Gender=male , Payment Method=cre...	loyal	5	176	181	0.201	0.972	0.548	0.3...	1.514	0....	0...
LastTransaction=range2 [56.500 - 112]	loyal	70	285	355	0.394	0.803	0.597	0.1...	1.250	0....	0...
LastTransaction=range2 [56.500 - 112] , Payment Method=credit card	loyal	41	230	271	0.301	0.849	0.568	0.2...	1.322	0....	0...

FIGURE 5.47 Rules generated by the *Subgroup Discovery* operator

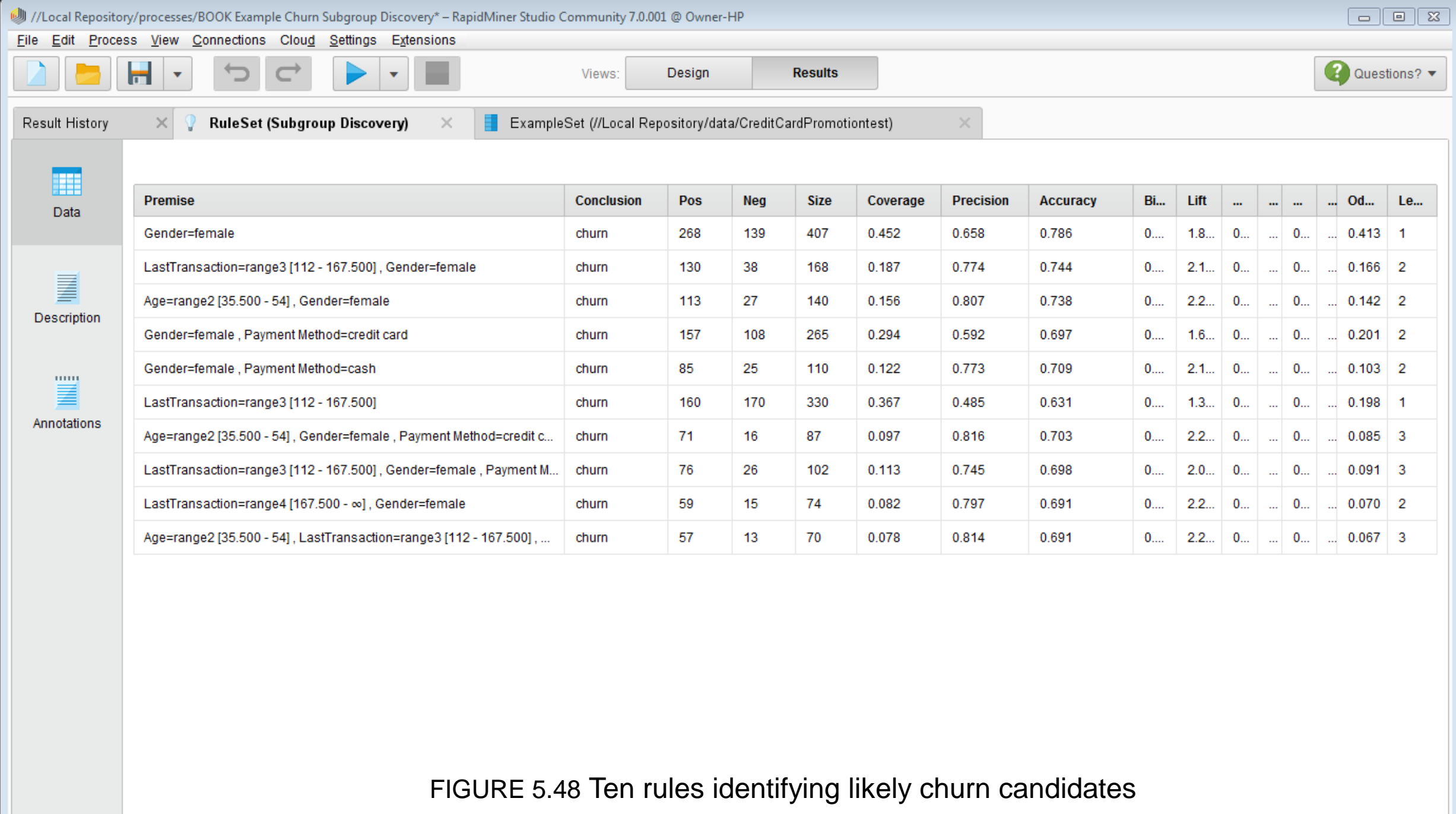
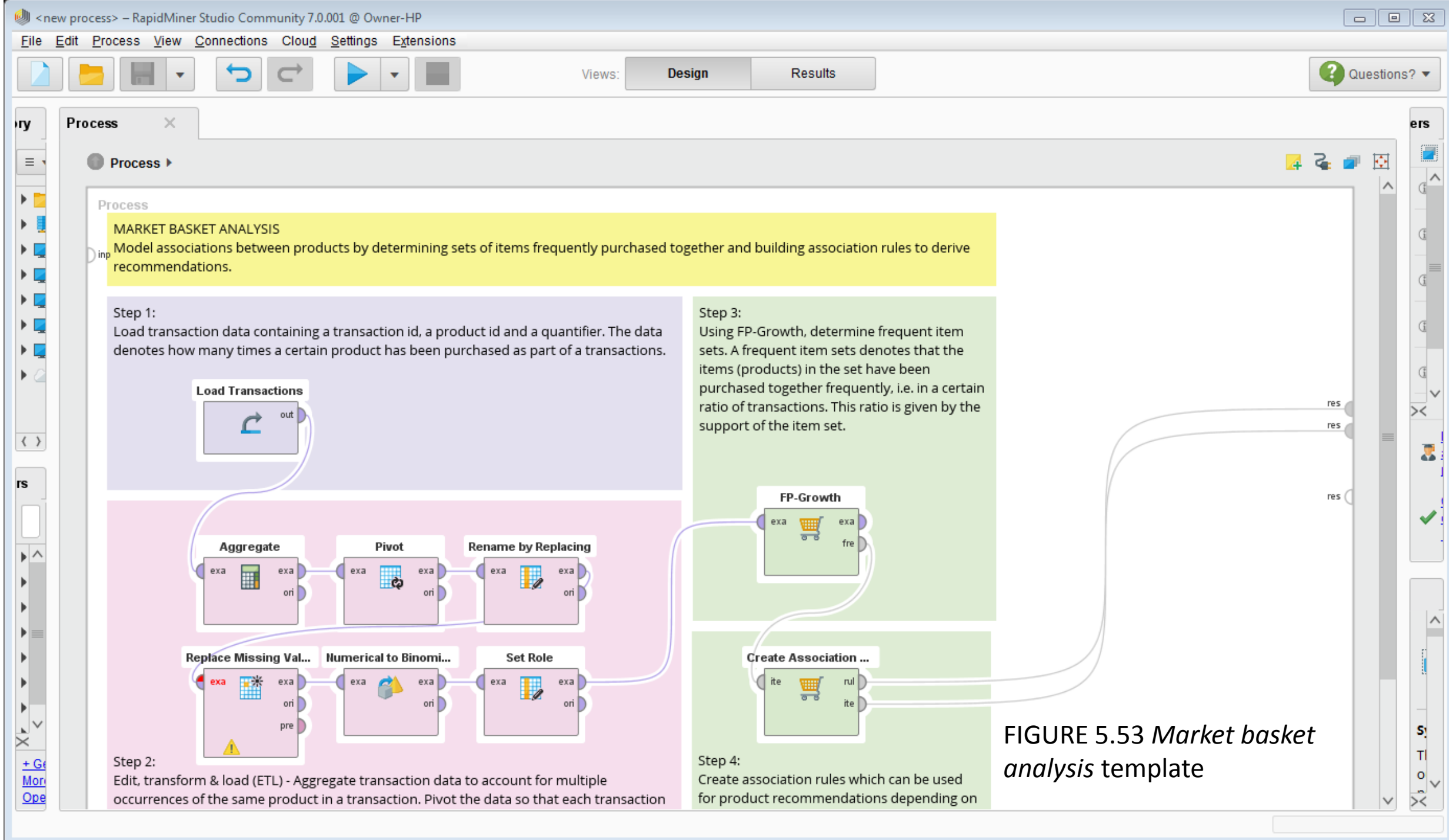
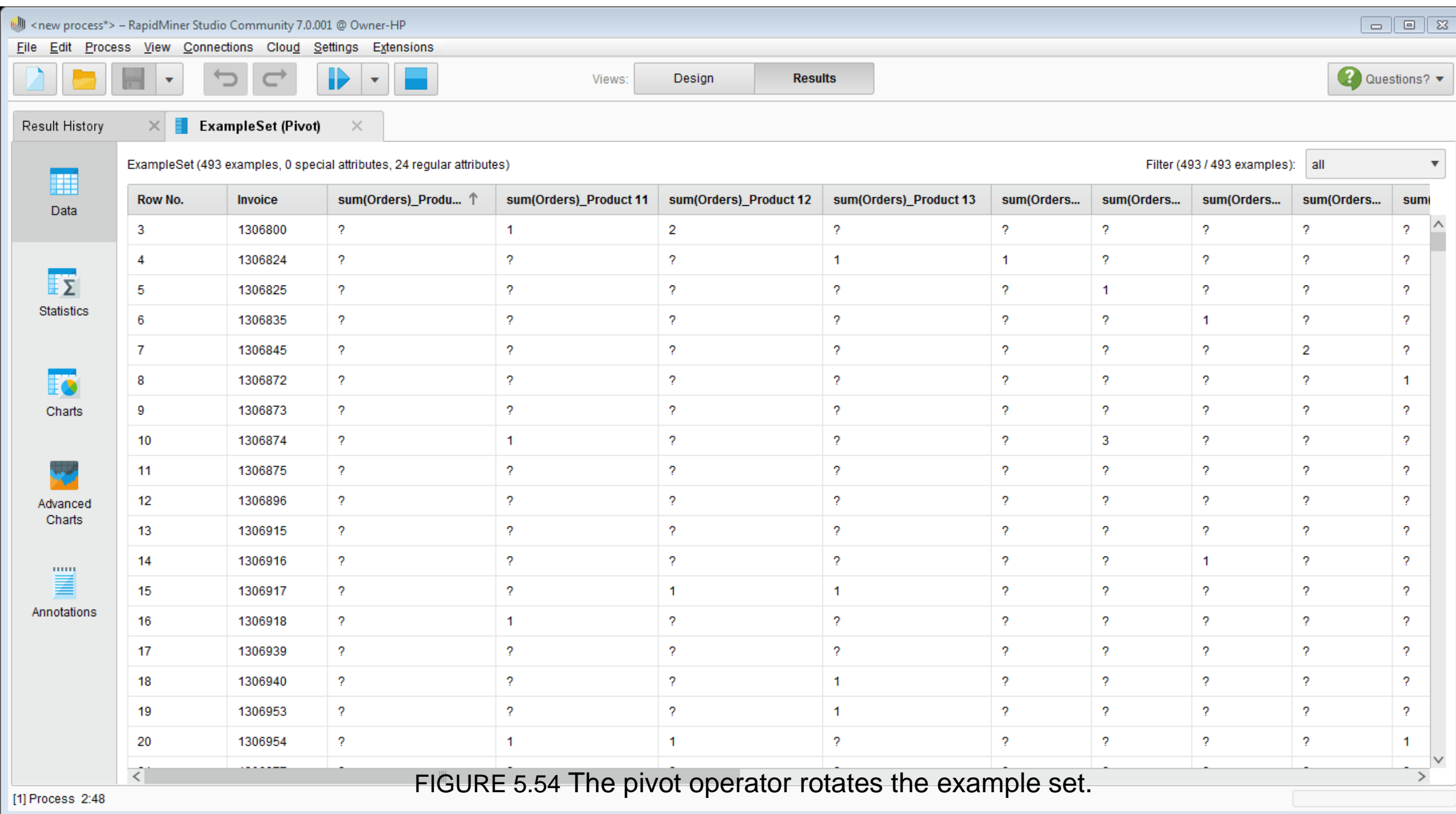
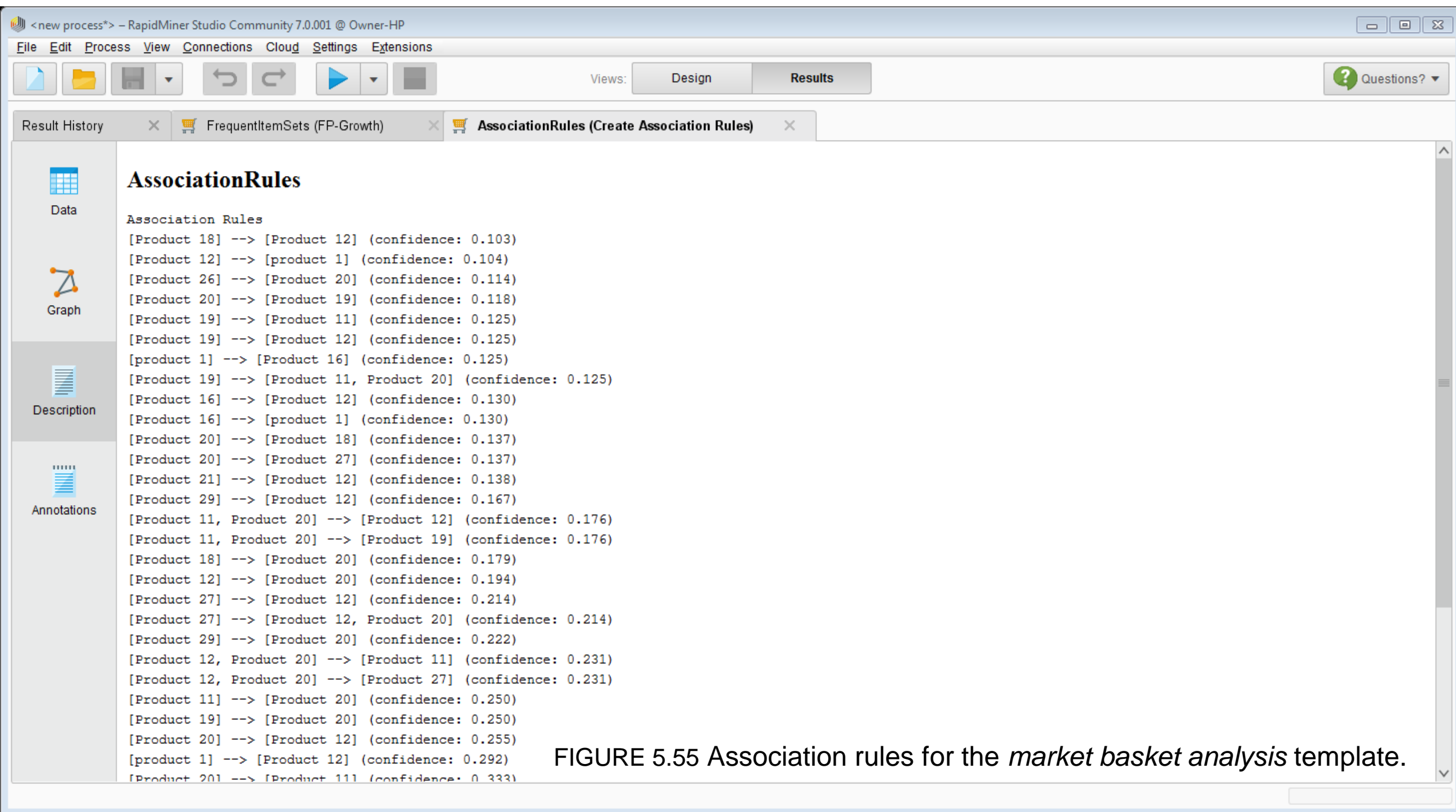


FIGURE 5.48 Ten rules identifying likely churn candidates







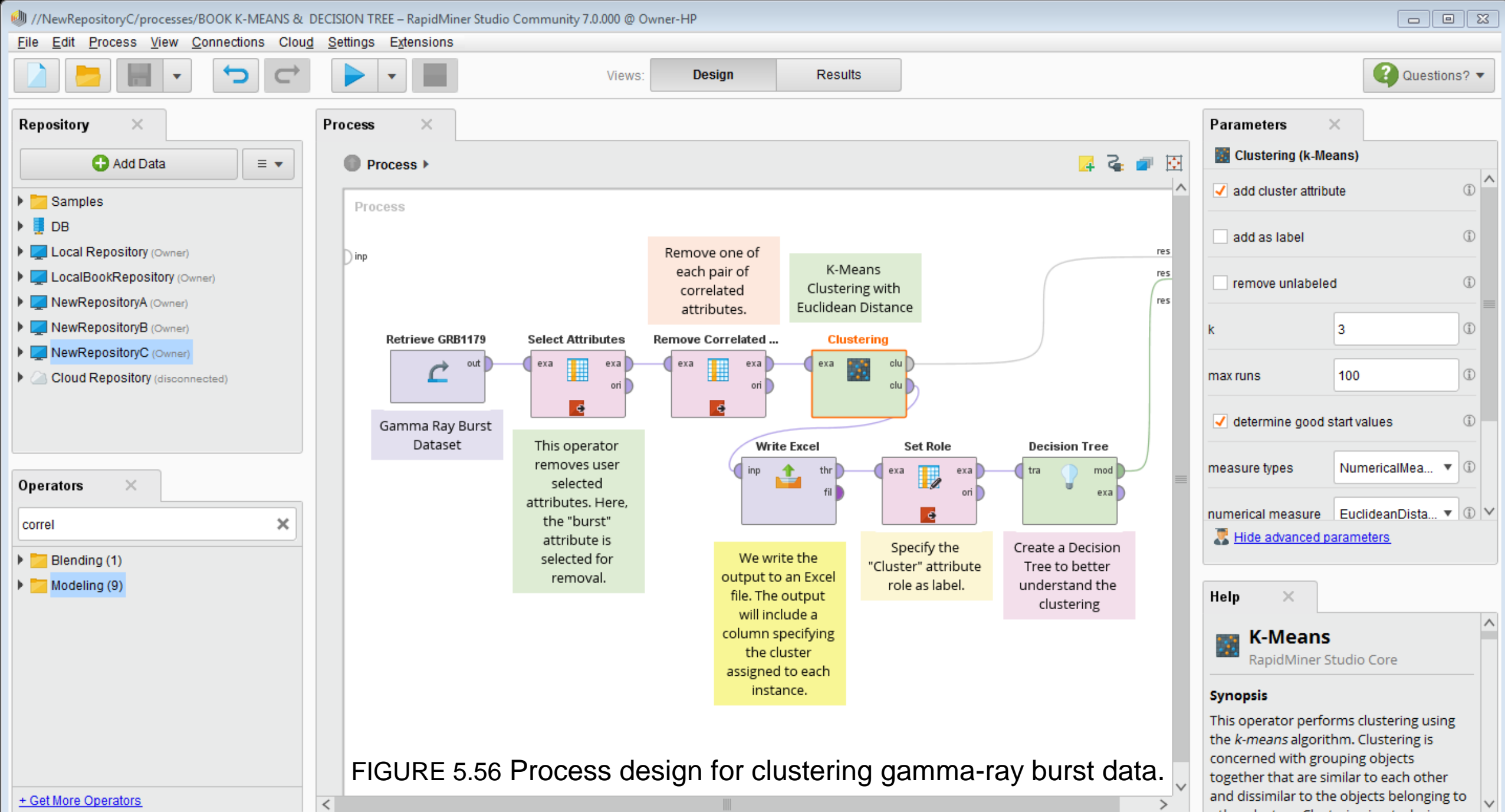


FIGURE 5.56 Process design for clustering gamma-ray burst data.

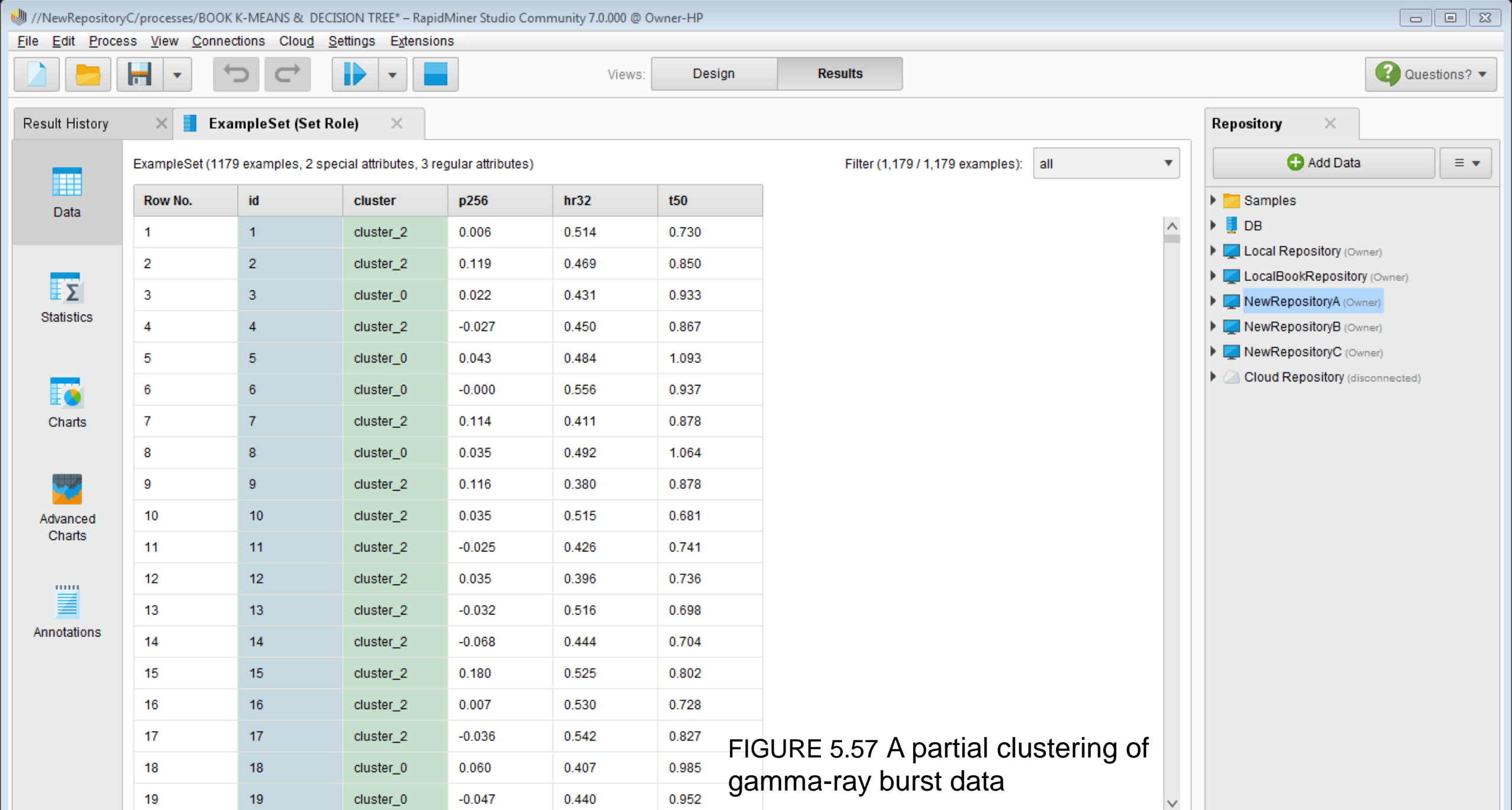
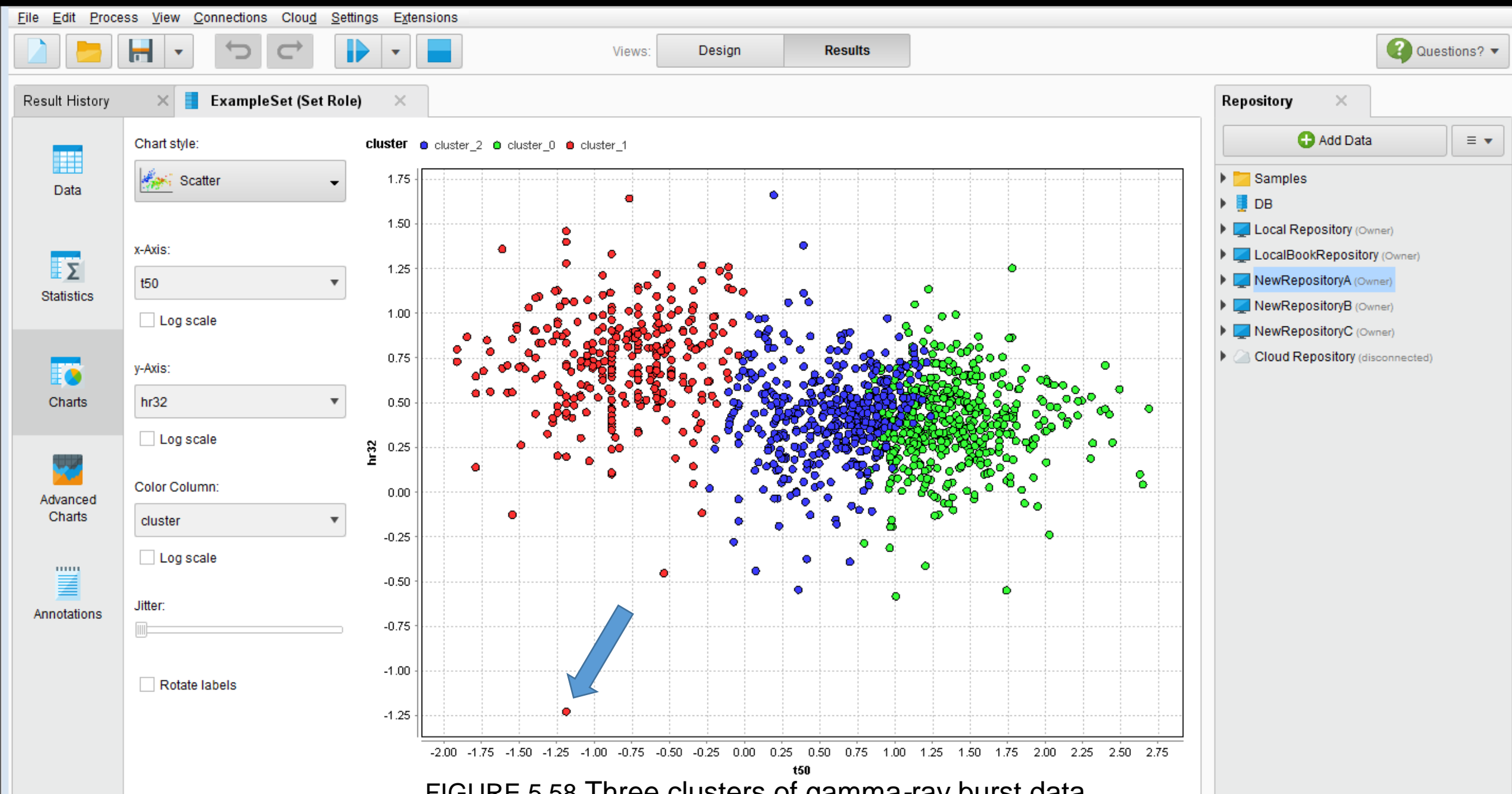


FIGURE 5.57 A partial clustering of gamma-ray burst data



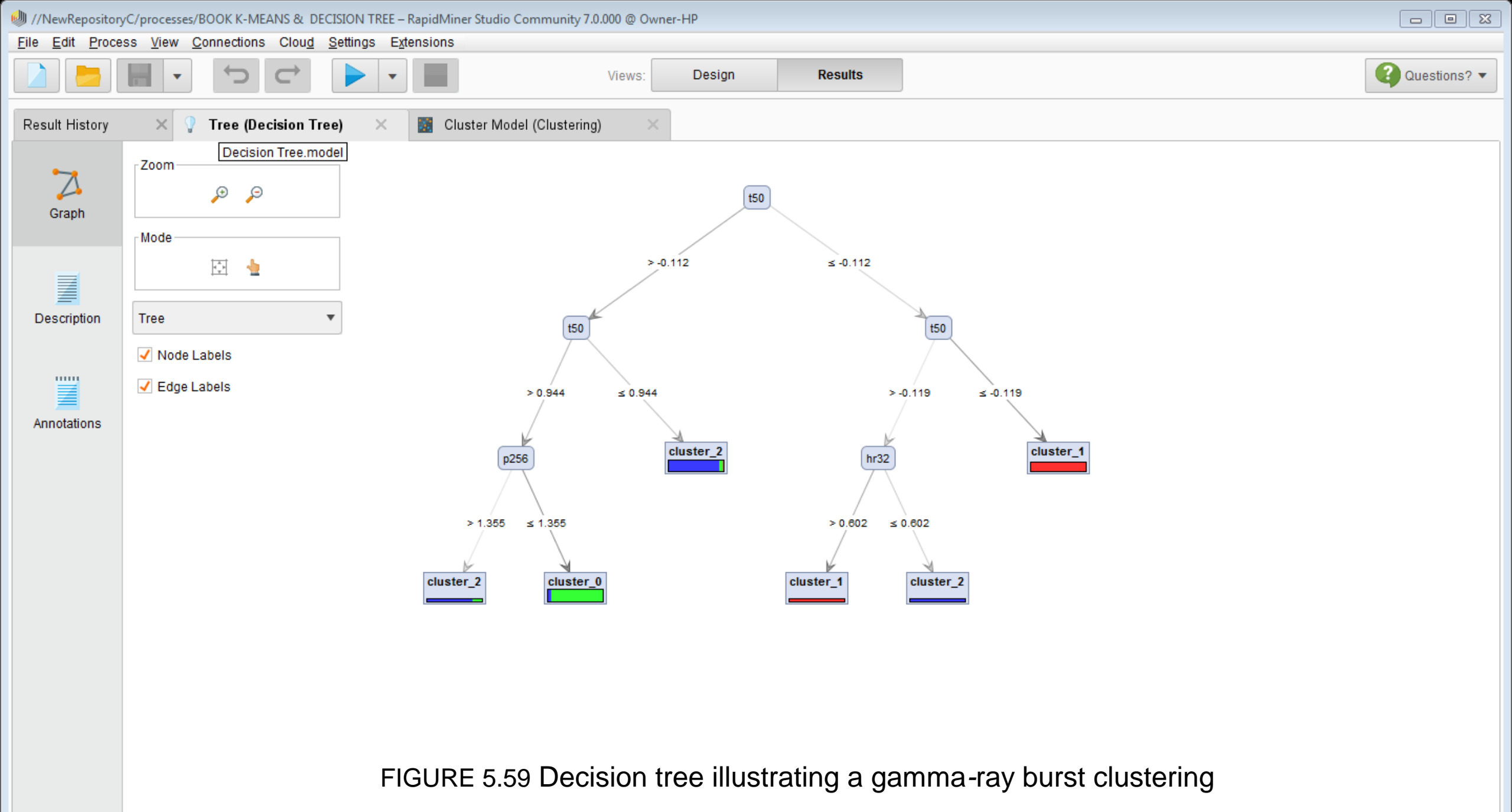


FIGURE 5.59 Decision tree illustrating a gamma-ray burst clustering

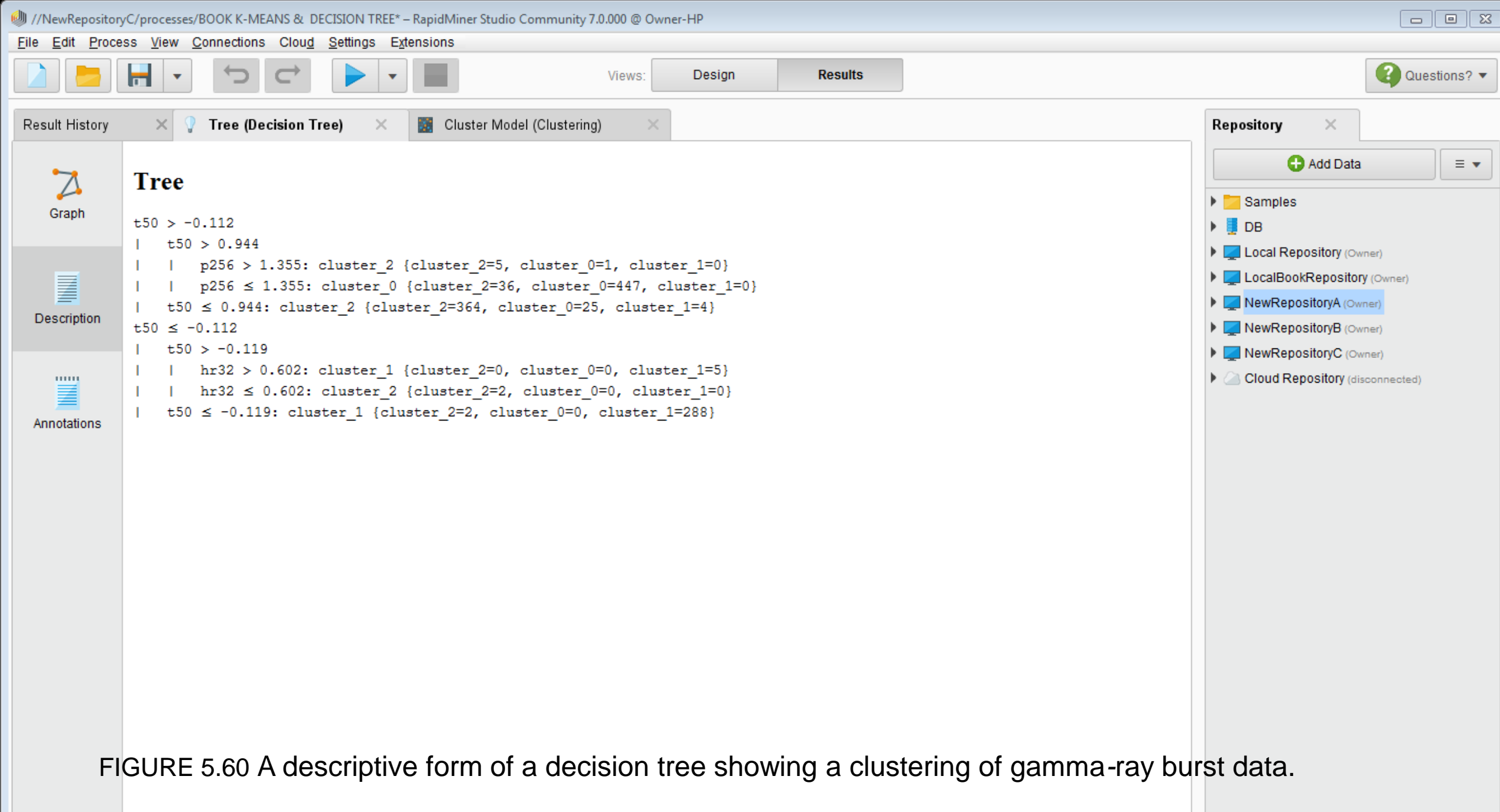


FIGURE 5.60 A descriptive form of a decision tree showing a clustering of gamma-ray burst data.

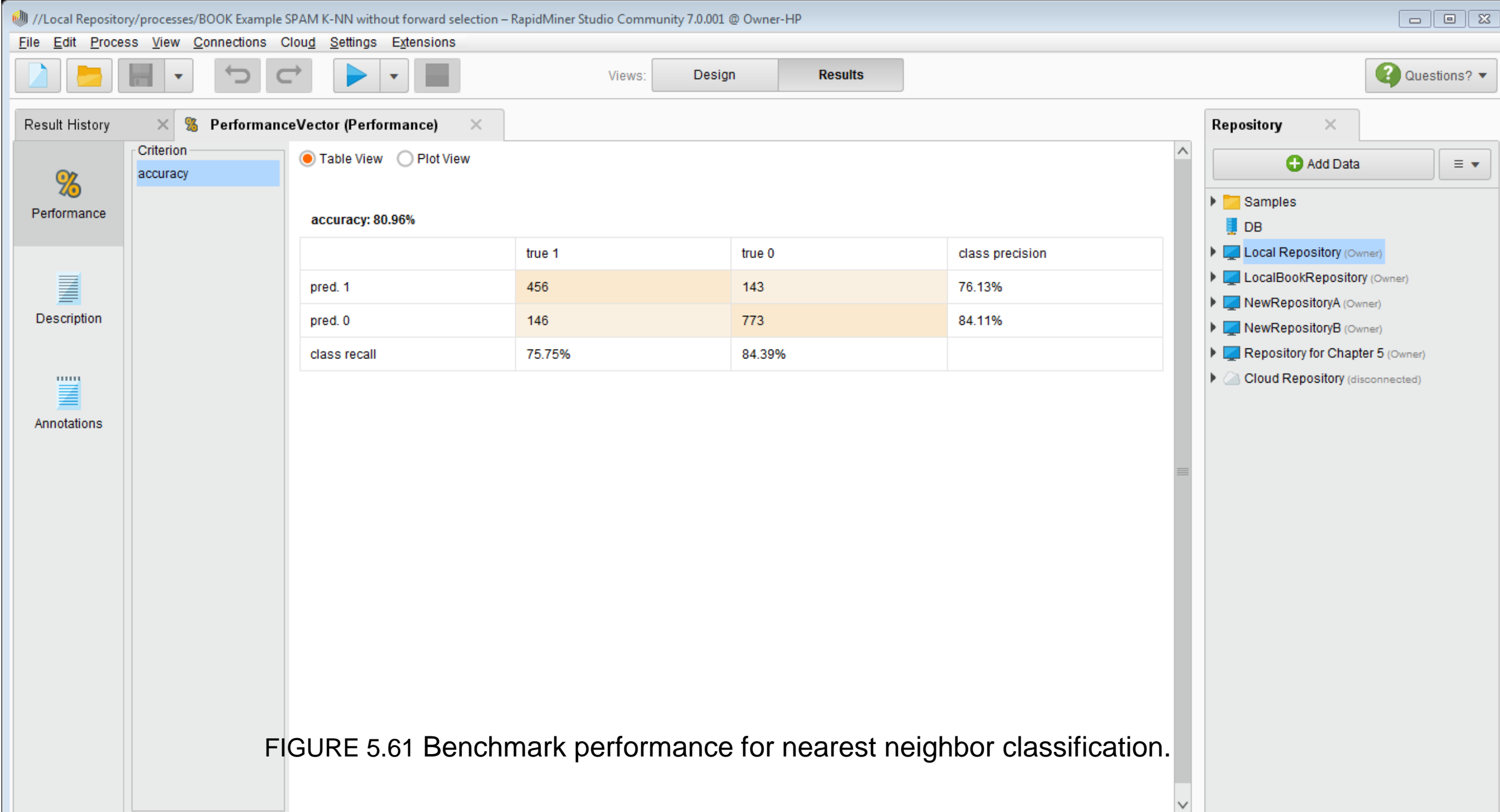


FIGURE 5.61 Benchmark performance for nearest neighbor classification.

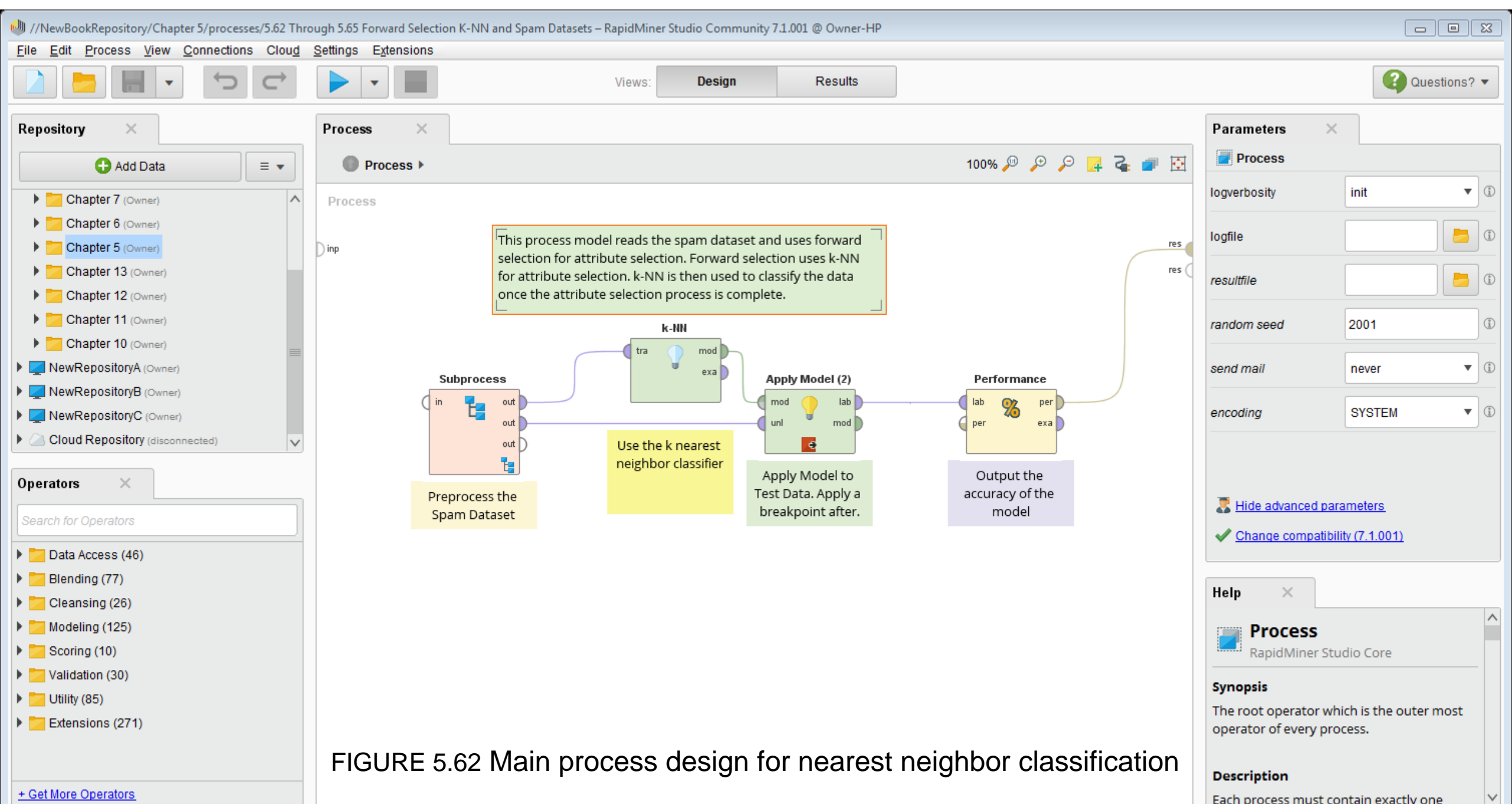


FIGURE 5.62 Main process design for nearest neighbor classification

Figure 5.63 shows a screenshot of the RapidMiner Studio interface, illustrating a subprocess for preprocessing data using forward selection for nearest neighbor classification.

The interface displays the following components:

- Repository:** Lists data sources including Samples, DB, Local Repository (Owner), NewBookExercises (Owner), NewBookRepository (Owner), NewRepositoryA (Owner), NewRepositoryB (Owner), NewRepositoryC (Owner), TestJuly12 (Owner), and Cloud Repository (disconnected).
- Process:** Shows a subprocess titled "Subprocess for preprocessing the data using forward selection." The workflow includes:
 - Retrieve spam data...** (Input operator)
 - Forward Selection** (Operator): Labeled "Use k-NN with Forward Selection".
 - Split Data (2)** (Operator): Labeled "2/3 Train 1/3 Test".
- Parameters:** Configures the Forward Selection operator with:
 - maximal number ... 10
 - speculative round... 0
 - stopping behavior without incre...
- Help:** Provides a synopsis of the Forward Selection operator: "This operator selects the most relevant attributes of the given ExampleSet t... highly efficient implementation of th... selection scheme".

FIGURE 5.63 Subprocess for nearest neighbor classification.

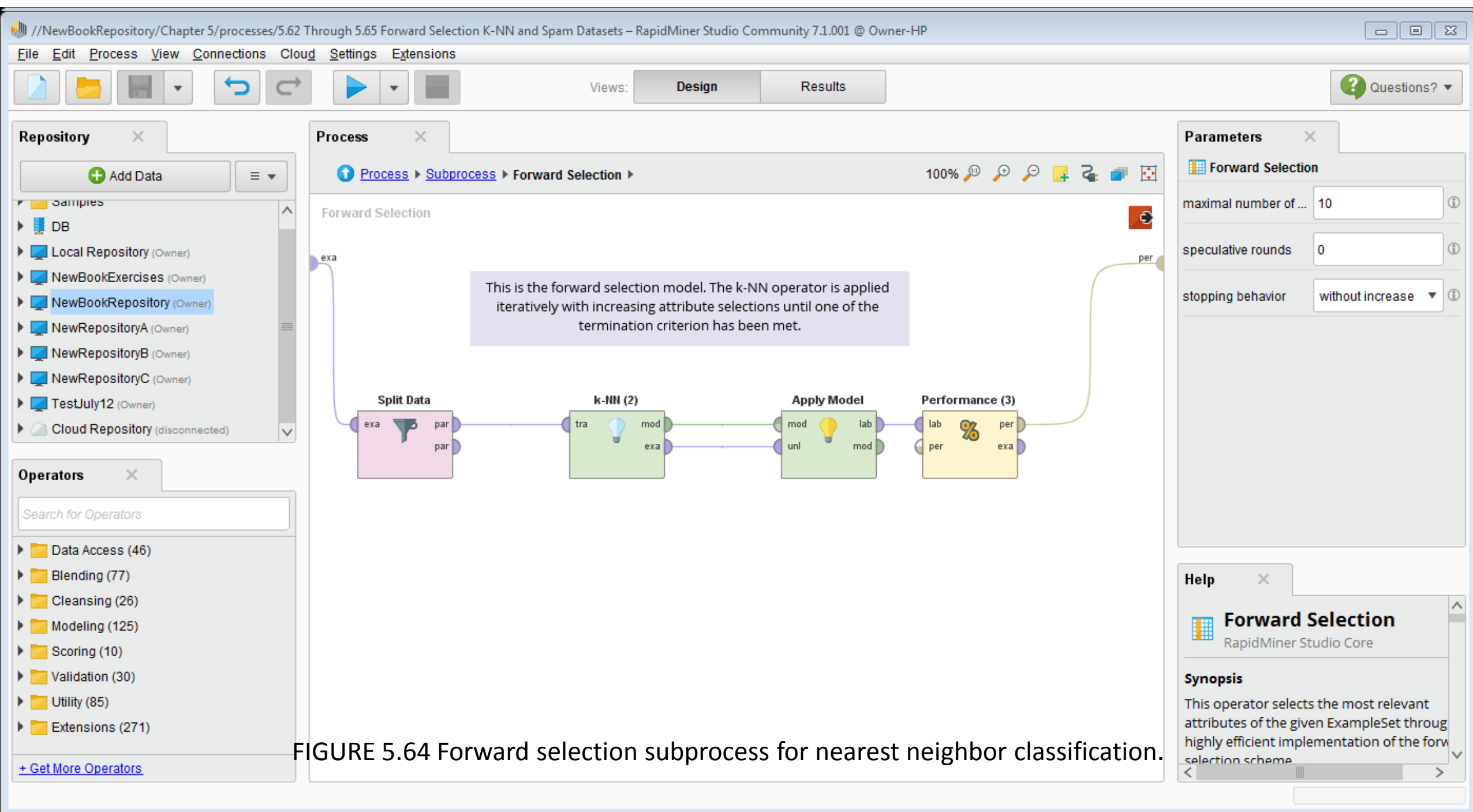


FIGURE 5.64 Forward selection subprocess for nearest neighbor classification.

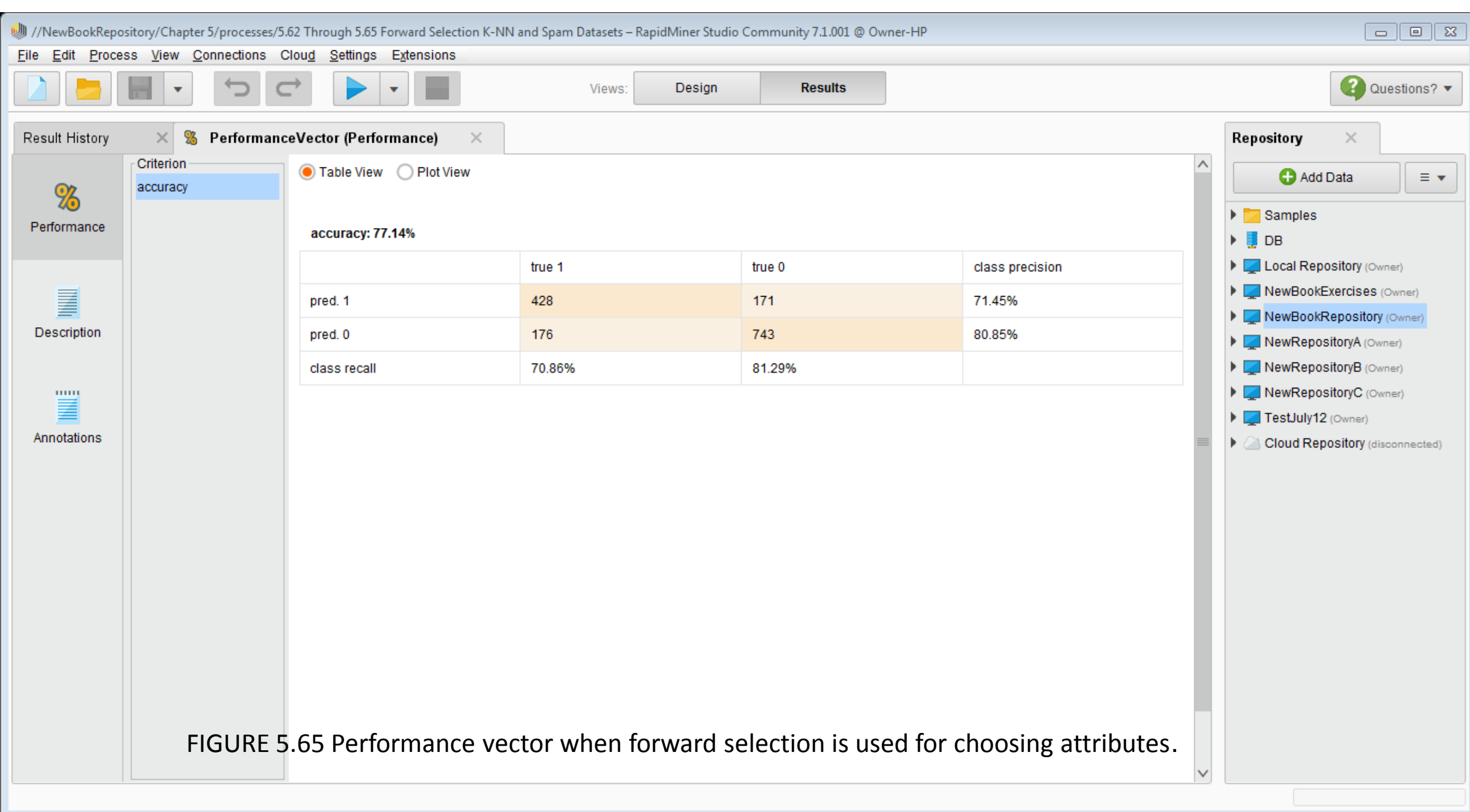


FIGURE 5.65 Performance vector when forward selection is used for choosing attributes.

Local Repository/processes/EXERCISE 5.12 Chapter 5 exercise with k-means and cardiology numerical – RapidMiner Studio Community 7.0.001 @ Owner-HP

File Edit Process View Connections Cloud Settings Extensions

Views: Design Results

Questions?

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 (256)

+ Get More Operators

Process

Process

Exercise 5.12 Chapter 5
Using unsupervised Clustering with the cardiology patient data to determine if a the set of input attributes are a viable choice for supervised learning.

inp

Retrieve cardiology...

Clustering

Write Excel

Multiply

Filter Examples

Filter Examples (2)

Filter Examples (3)

Filter Examples (4)

res

res

res

res

res

Parameters

Process

logverbosity init

logfile

Show advanced parameters

Change compatibility (7.0.001)

Help

Process

RapidMiner Studio Core

Synopsis

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

Description

FIGURE 5.66 Unsupervised clustering for attribute evaluation.