

GROUP RED

TRAINING, TESTING & VALIDATION OF A PREDICTIVE MODEL

JASPREET KAUR

AKSHAY JADHAV

ADITI JAIN

AISHWARYA KATE

RASHMI JAIN

SHIVANI JAIN

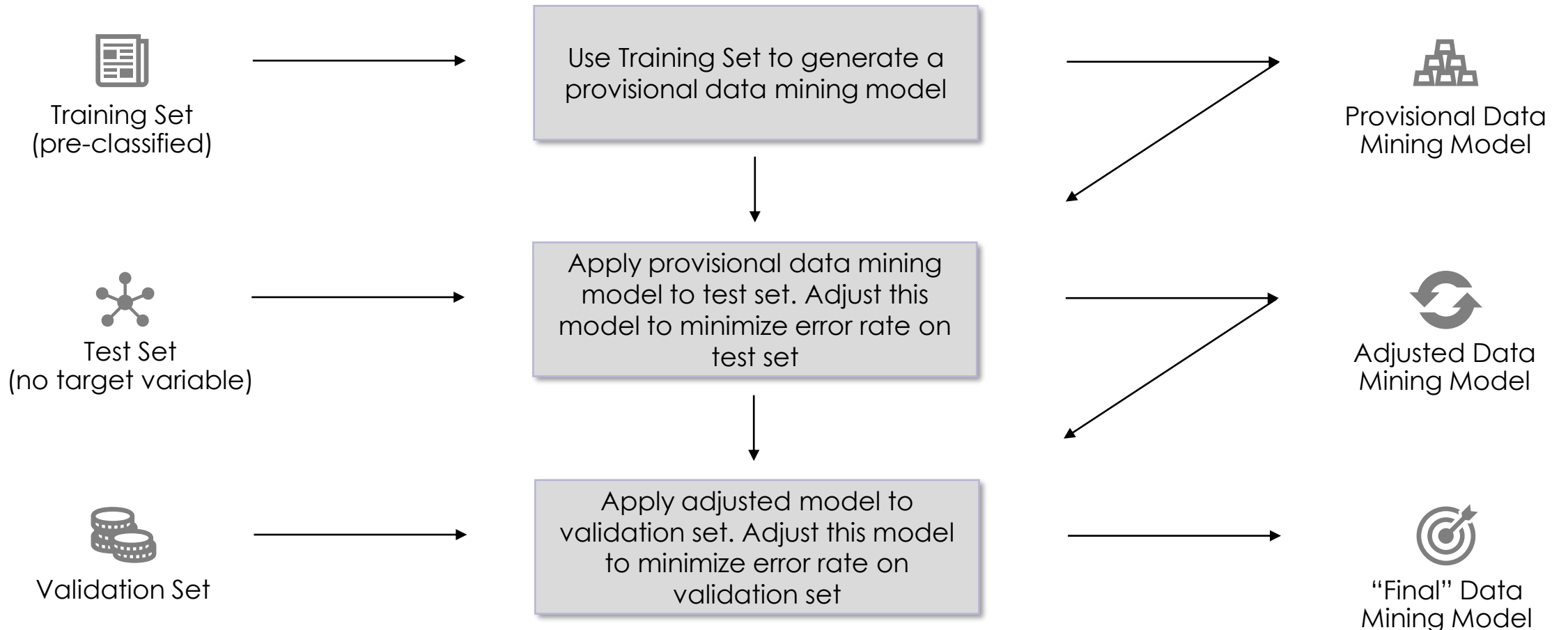
22-April-2020



GOAL



To choose a predictive model that can score records as True or False with at least 95% accuracy



DECISION TREE



A popular **supervised** classification method used in data mining



A decision tree is a collection of decision nodes, connected by branches, extending downward from **root** node to terminating **leaf** nodes



Begins with a root node, attributes are tested at decision nodes, and each possible outcome results in a branch; each branch leads to a decision node or a leaf node



Decision trees learn by **example**, hence the training set contains records with varied attribute values



Tool used for building the decision tree model: **RapidMiner Studio** 



RapidMiner is a very effective data science software platform that unites data prep, machine learning & predictive model deployment

Recommendation in EDA Phase:

....

Looks like it is extremely fast to build and learn and seems to provide an accuracy of 96.8%

DECISION TREE - USING THE TRAINING SET

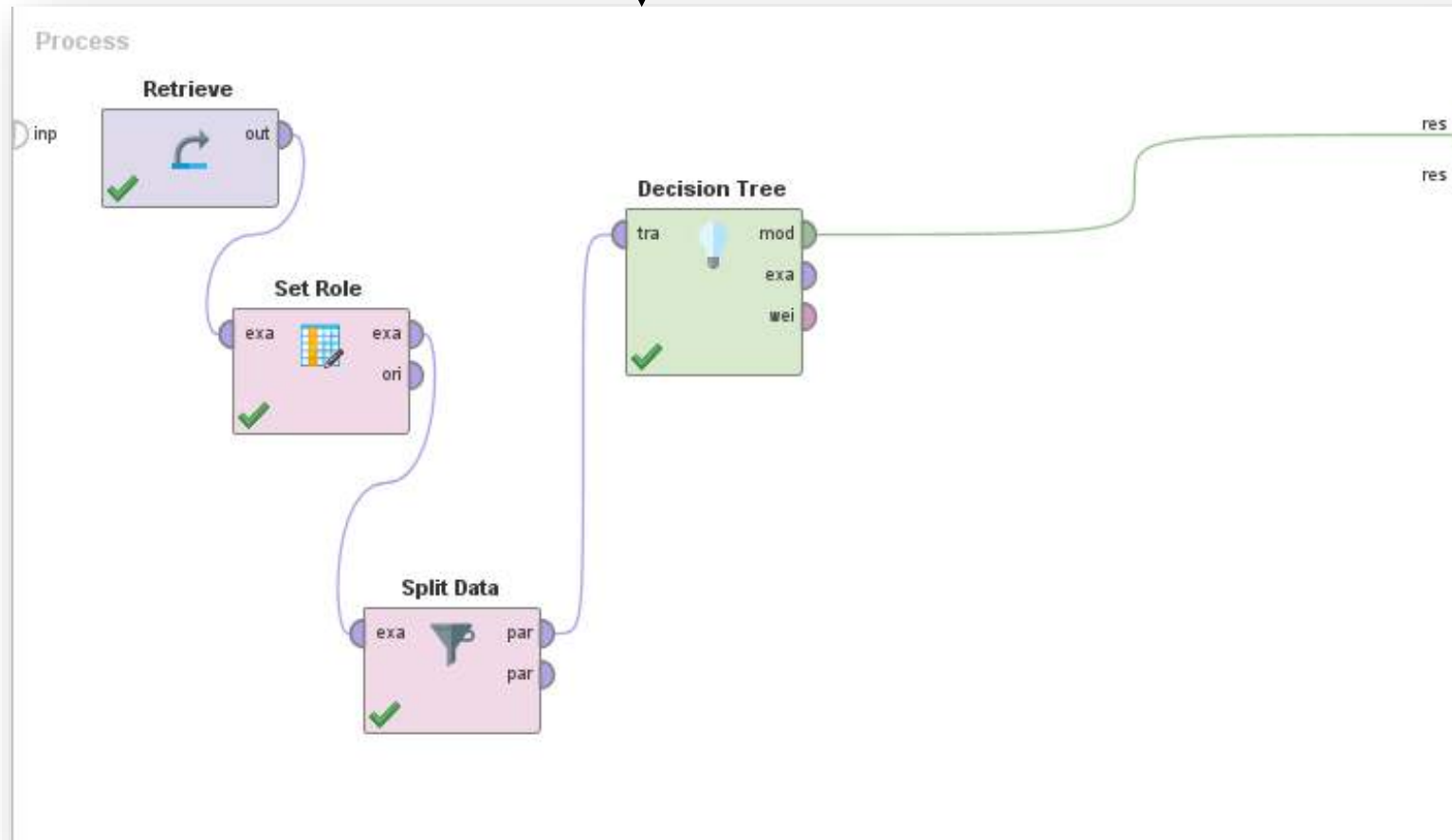
The screenshot displays the RapidMiner Studio Educational 9.6.000 interface. The main window is titled "Process" and shows a large empty canvas with the text: "Your process looks empty. Add some data first. Drag data or operators here." The interface includes several panels:

- Repository:** Lists various models and data sets, including "Building_Decision_Tree_Model_T", "Decision_Tree_Model_15-04-202", "Decision_Tree_Model_with_3_Pa", "DecisionTreeModel-1 (DELL - v1, 4", "Naive_Bayes_Classification_Mod", "Naive_Bayes_Model_Try (DELL - v", "Settings (DELL - v1, 4/17/20 1:40 AM", and "Simple_Naive_Bayes_Model_Try".
- Operators:** A search bar and a list of operator categories: Data Access (53), Blending (81), Cleansing (29), Modeling (165), and Scoring (14). The "Apply Model" operator is highlighted under the Scoring category.
- Parameters:** A panel for the "Process" operator, showing parameters like "logverbosity" (init), "logfile", "resultfile", "random seed" (2001), "send mail" (never), and "encoding" (SYSTEM). It also includes links for "Hide advanced parameters" and "Change compatibility (9.6.000)".
- Help:** A panel for the "Apply Model" operator, showing tags like "Predict, Predictions, Forecasts, Scores, Scoring, Trained, Test" and a synopsis: "This Operator applies a model on an ExampleSet."

The bottom of the interface features a status bar with the text "Leverage the Wisdom of Crowds to get operator recommendations based on your process design!" and a button to "Activate Wisdom of Crowds".

DECISION TREE - USING THE TRAINING SET

Process Editor



The first step is to retrieve the data i.e. Alarm file

'Set Role' operator is used to tell RapidMiner which is our target variable (Alarm)

To partition the data set, we use 'Split Data' operator

Dataset is split into 3 sets: Training, Test & Validation datasets (1/3rd each)

Now, we use 'Decision Tree' operator to build the provisional model

We connect the inputs with their relevant outputs and run the process

DECISION TREE MODEL

Decision Nodes
(Component Accessed,
Timestamp, Requestor)

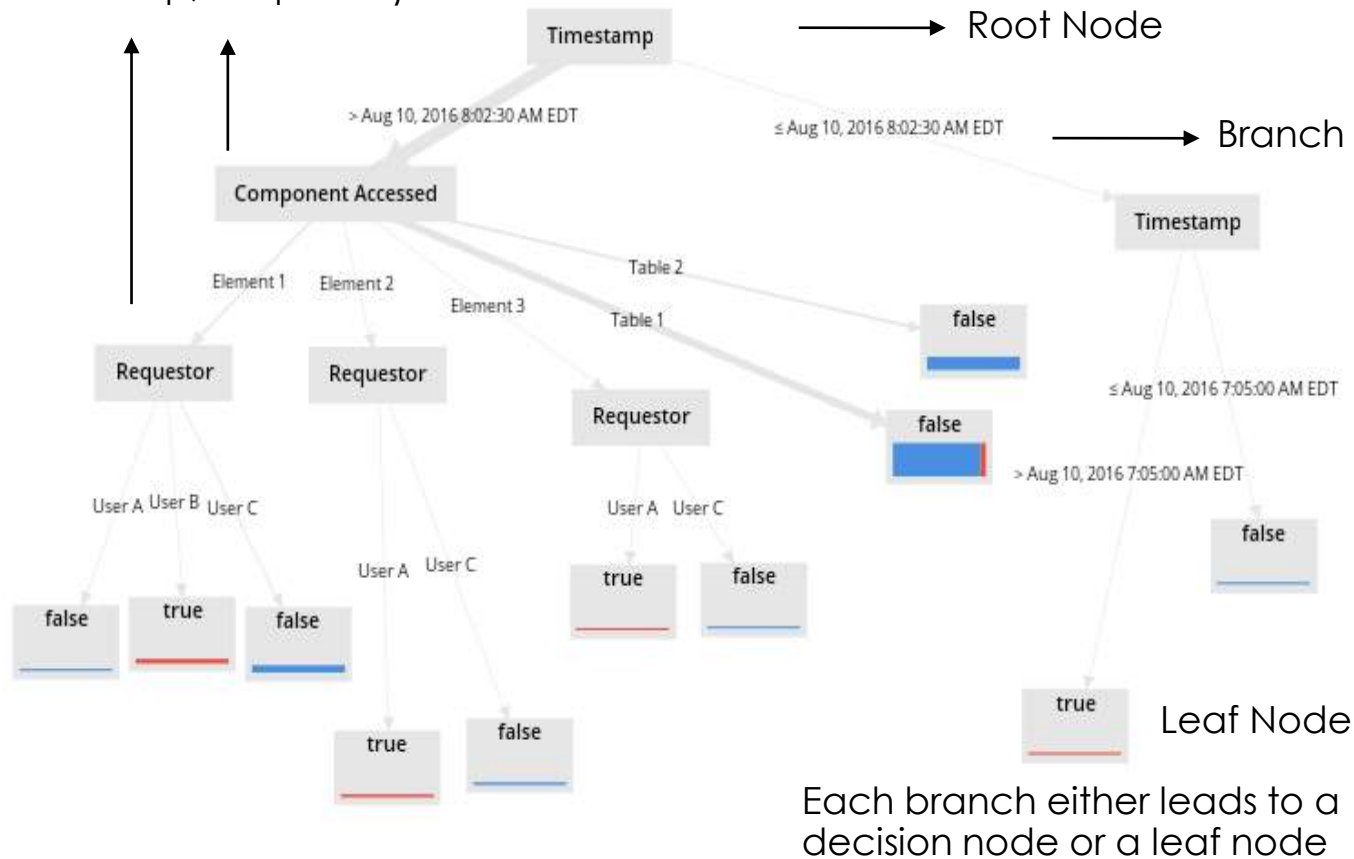


Figure 1.1 Decision Tree Model (provisional)

Tree

Timestamp > Aug 10, 2016 8:02:30 AM EDT

```
| Component Accessed = Element 1
| | Requestor = User A: false {false=4, true=0}
| | Requestor = User B: true {false=0, true=20}
| | Requestor = User C: false {false=40, true=0}
| Component Accessed = Element 2
| | Requestor = User A: true {false=0, true=9}
| | Requestor = User C: false {false=8, true=0}
| Component Accessed = Element 3
| | Requestor = User A: true {false=0, true=3}
| | Requestor = User C: false {false=3, true=0}
| Component Accessed = Table 1: false {false=207, true=12}
| Component Accessed = Table 2: false {false=80, true=0}
```

Timestamp ≤ Aug 10, 2016 8:02:30 AM EDT

```
| Timestamp > Aug 10, 2016 7:05:00 AM EDT: true {false=0, true=4}
| Timestamp ≤ Aug 10, 2016 7:05:00 AM EDT: false {false=4, true=0}
```

Description of the provisional Decision Tree Model

DECISION TREE - USING THE TEST SET

Local Repository/MyAlarmProject/DecisionTreeModel-1 - RapidMiner Studio Educational 9.6.000 @ JaspreetKaurPC

File Edit Process View Connections Settings Extensions Help

Views: Design Results Turbo Prep Auto Model Deployments

Find data, operators...etc All Studio

Repository

- Import Data
- Building_Decision_Tree_Model_T
- Decision_Tree_Model_15-04-202
- Decision_Tree_Model_with_3_Pa
- DecisionTreeModel-1 (DELL - v1, 4
- Naive_Bayes_Classification_Mod
- Naive_Bayes_Model_Try (DELL - v
- Settings (DELL - v1, 4/17/20 1:40 AM
- Simple_Naive_Bayes_Model_Try

Operators

Search for Operators

- Data Access (53)
- Blending (81)
- Cleansing (29)
- Modeling (165)
- Scoring (14)
 - Confidences (9)
 - Apply Model
 - Model Simulator

Get more operators from the Marketplace

Process

Process

inp

Retrieve

Set Role

Split Data

Decision Tree

res

res

Parameters

Process

- logverbosity: init
- logfile:
- resultfile:
- random seed: 2001
- send mail: never
- encoding: SYSTEM

[Hide advanced parameters](#)

[Change compatibility \(9.6.000\)](#)

Help

Apply Model

RapidMiner Studio Core

Tags: Predict, Predictions, Forecasts, Scores, Scoring, Trained, Test

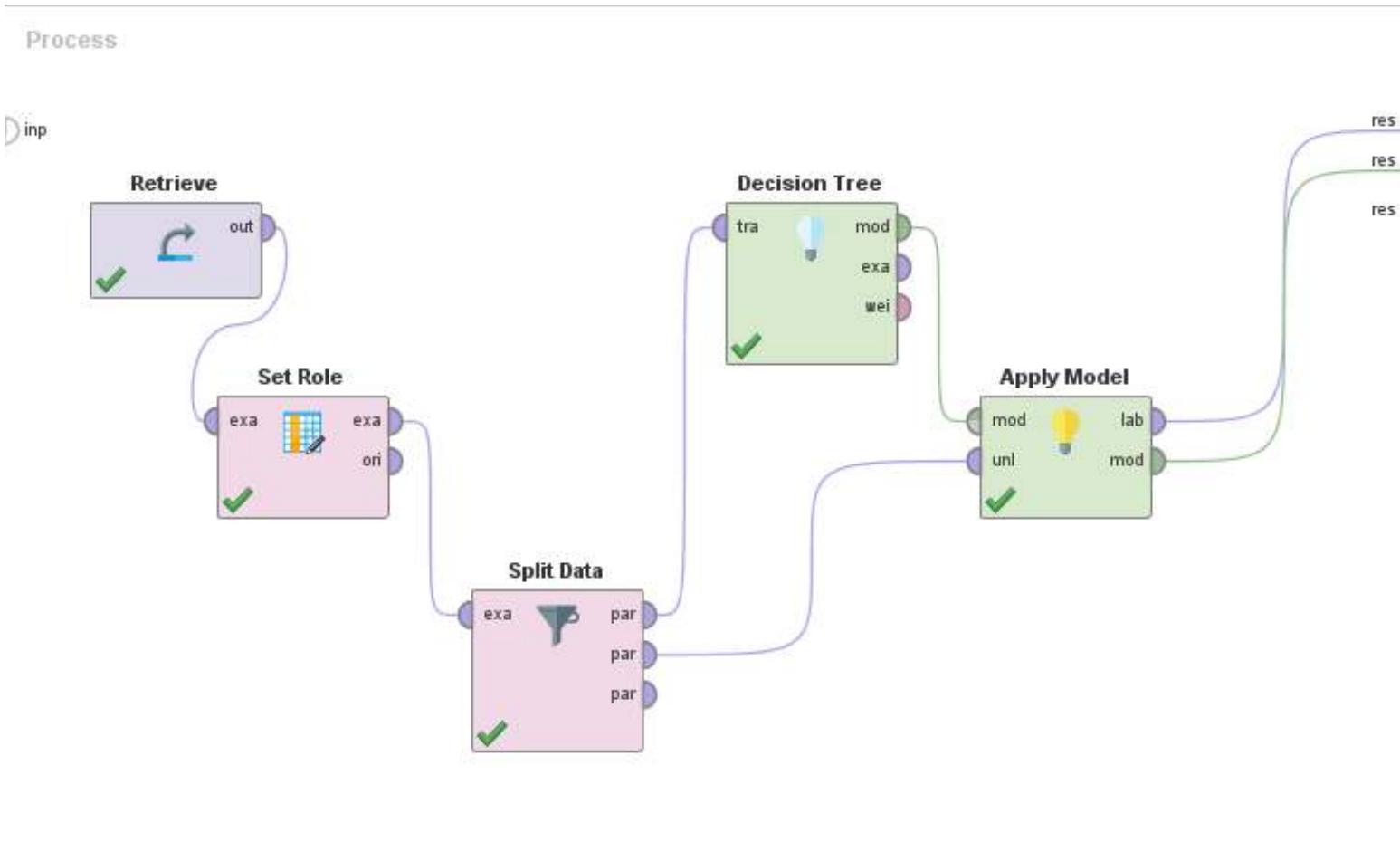
Synopsis

This Operator applies a model on an ExampleSet.

Leverage the Wisdom of Crowds to get operator recommendations based on your process design!

Activate Wisdom of Crowds

DECISION TREE - USING THE TEST SET



To apply the provisional data mining model to the test set, we use 'Apply Model' operator

'Apply Model' operator requires two inputs:
i) Model ii) Unlabeled data

The 'Apply Model' operator pretends as if 33.33% of the data (which comes from the test set) is unlabeled and applies the model to create the labels (false/true alarms)

And, the model is going to come from the 'output' port of the Decision Tree operator

FEEDBACK FROM THE TEST SET

- 12 records show “**No Match**” in the feedback from the test set, out of the 393 records
- The security analysts had stated that the alarms were true (as shown below in yellow) but our model predicts that those alarms were false (as shown below in orange)
- Clearly, the 12 predicted values of Alarm (as shown below) are “**False Negatives**” - generated by **Business User (User B)** on **Table 1** using Select query between August 14, 2016 and August 17, 2016
- Let's find out the accuracy of the model!



Microsoft Excel
Worksheet

Timestamp	Request	Role	Component Access	Request type	Violation type	Alarm	confidence(false)	confidence(true)	prediction(Alarm)	Match/No match
2016-08-14 18:29:00	User B	Business user	Table 1	Select	No authorization	true	0.9	0.1	false	No Match
2016-08-14 22:05:00	User B	Business user	Table 1	Select	No authorization	true	0.9	0.1	false	No Match
2016-08-14 22:19:00	User B	Business user	Table 1	Select	No authorization	true	0.9	0.1	false	No Match
2016-08-14 22:34:00	User B	Business user	Table 1	Select	No authorization	true	0.9	0.1	false	No Match
2016-08-15 22:34:00	User B	Business user	Table 1	Select	No authorization	true	0.9	0.1	false	No Match
2016-08-16 02:10:00	User B	Business user	Table 1	Select	No authorization	true	0.9	0.1	false	No Match
2016-08-16 02:24:00	User B	Business user	Table 1	Select	No authorization	true	0.9	0.1	false	No Match
2016-08-16 02:39:00	User B	Business user	Table 1	Select	No authorization	true	0.9	0.1	false	No Match
2016-08-17 02:39:00	User B	Business user	Table 1	Select	No authorization	true	0.9	0.1	false	No Match
2016-08-17 06:15:00	User B	Business user	Table 1	Select	No authorization	true	0.9	0.1	false	No Match
2016-08-17 06:29:00	User B	Business user	Table 1	Select	No authorization	true	0.9	0.1	false	No Match
2016-08-17 06:43:00	User B	Business user	Table 1	Select	No authorization	true	0.9	0.1	false	No Match

DECISION TREE - ACCURACY CHECK

Local Repository/MyAlarmProject/DecisionTreeModel-1 - RapidMiner Studio Educational 9.6.000 @ JaspreetKaurPC

File Edit Process View Connections Settings Extensions Help

Views: Design Results Turbo Prep Auto Model Deployments

Find data, operators...etc. All Studio

Repository

- Import Data
- Building_Decision_Tree_Model_1
- Decision_Tree_Model_15-04-202
- Decision_Tree_Model_with_3_Pa
- DecisionTreeModel-1 (DELL - v1, 4
- Naive_Bayes_Classification_Mod
- Naive_Bayes_Model_Try (DELL - v
- Settings (DELL - v1, 4/17/20 1:40 AM
- Simple_Naive_Bayes_Model_Try

Operators

Search for Operators

- Data Access (53)
- Blending (81)
- Cleansing (29)
- Modeling (165)
- Scoring (14)
- Validation (30)
- Performance (21)

Get more operators from the Marketplace

Process

Process

inp

Retrieve

Set Role

Split Data

Decision Tree

Apply Model

res

res

res

res

Parameters

Process

logverbosity init

logfile

resultfile

random seed 2001

send mail never

encoding SYSTEM

Hide advanced parameters

Change compatibility (9.6.000)

Help

Performance (Classification)

RapidMiner Studio Core

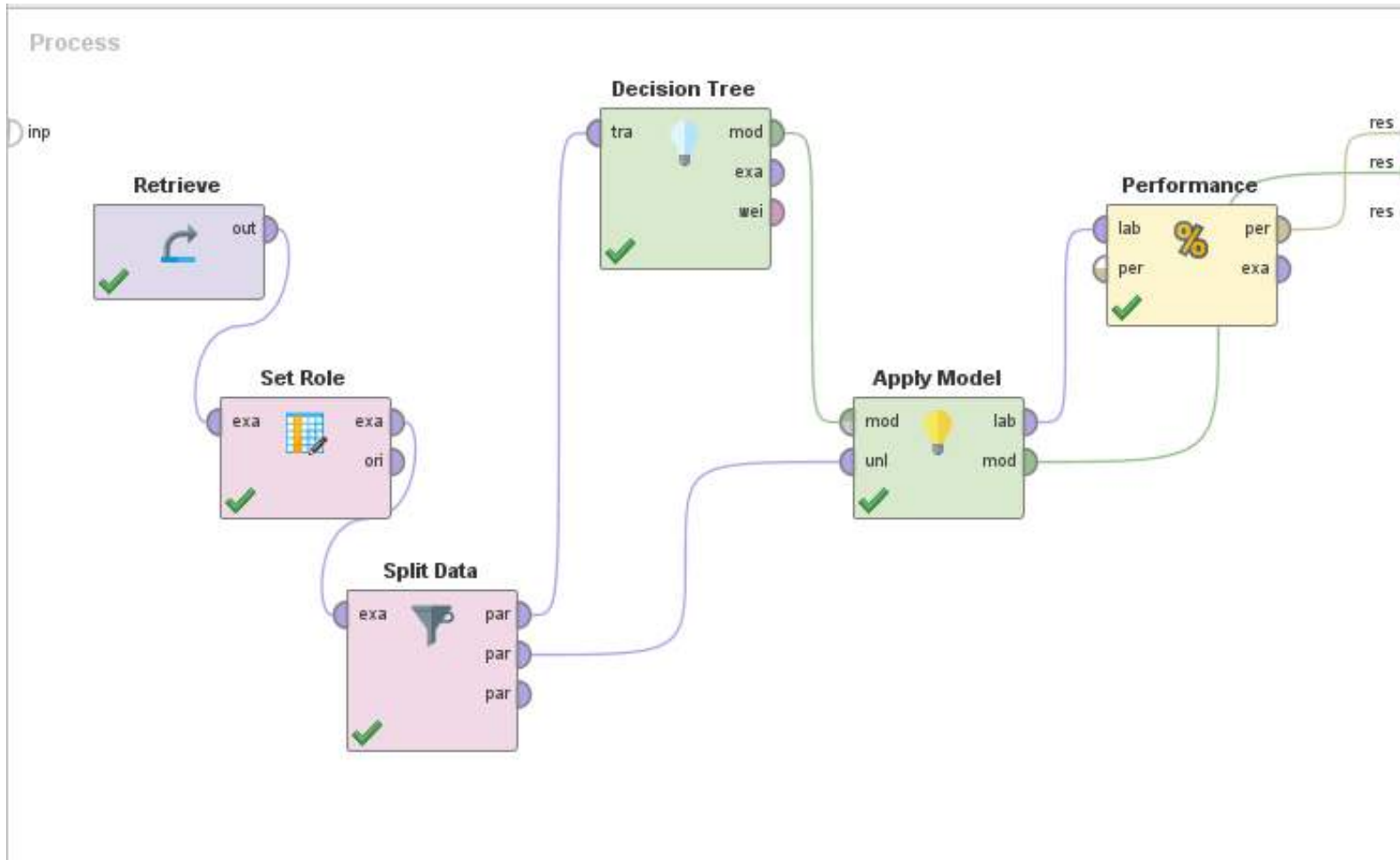
Tags: Accuracy, Errors, Precision, Recall, Kappa, Squared, Relative, Validations, Evaluations, Metrics, Confusion Matrix, Predictive

Synopsis

Leverage the Wisdom of Crowds to get operator recommendations based on your process design!

Activate Wisdom of Crowds

DECISION TREE - ACCURACY CHECK



'Performance (Classification)' Operator is used to evaluate the model that we are building in terms of the model accuracy, classification error etc.

'Performance (Classification) Operator' has one mandatory input: **labeled data** (which comes from the output port of the Apply Model)

We get a Performance (Classification) matrix and a decision tree model when we run the process

This table (or matrix) is also called '**Confusion Matrix**' as it describes the performance of a classification model (decision tree in our case) on a set of test data for which the true values are known

HOW GOOD OUR MODEL IS?

Tree (Decision Tree)		PerformanceVector (Performance)	
Criterion		Table View Plot View	
accuracy		accuracy: 96.95%	
classification error		Actual (or True) Values	
Predicted Values	true false	true true	class precision
pred. false	348 (TN)	12 (FN) - Type II error	96.67%
pred. true	0 (FP) - Type I error	33 (TP)	100.00%
class recall	100.00%	73.33%	

Here, True Negative (TN) = 348
False Negative (FN) = 12
False Positive (FP) = 0
True Positive (TP) = 33

Hence, **True Positive Rate (TPR)** can be calculated as: $TP / (TP + FN) = 73.33\%$

True Negative Rate (TNR) can be calculated as: $TN / (TN + FP) = 100.00\%$

Positive Predictive Value (PPV) can be calculated as: $TP / (TP + FP) = 100.00\%$

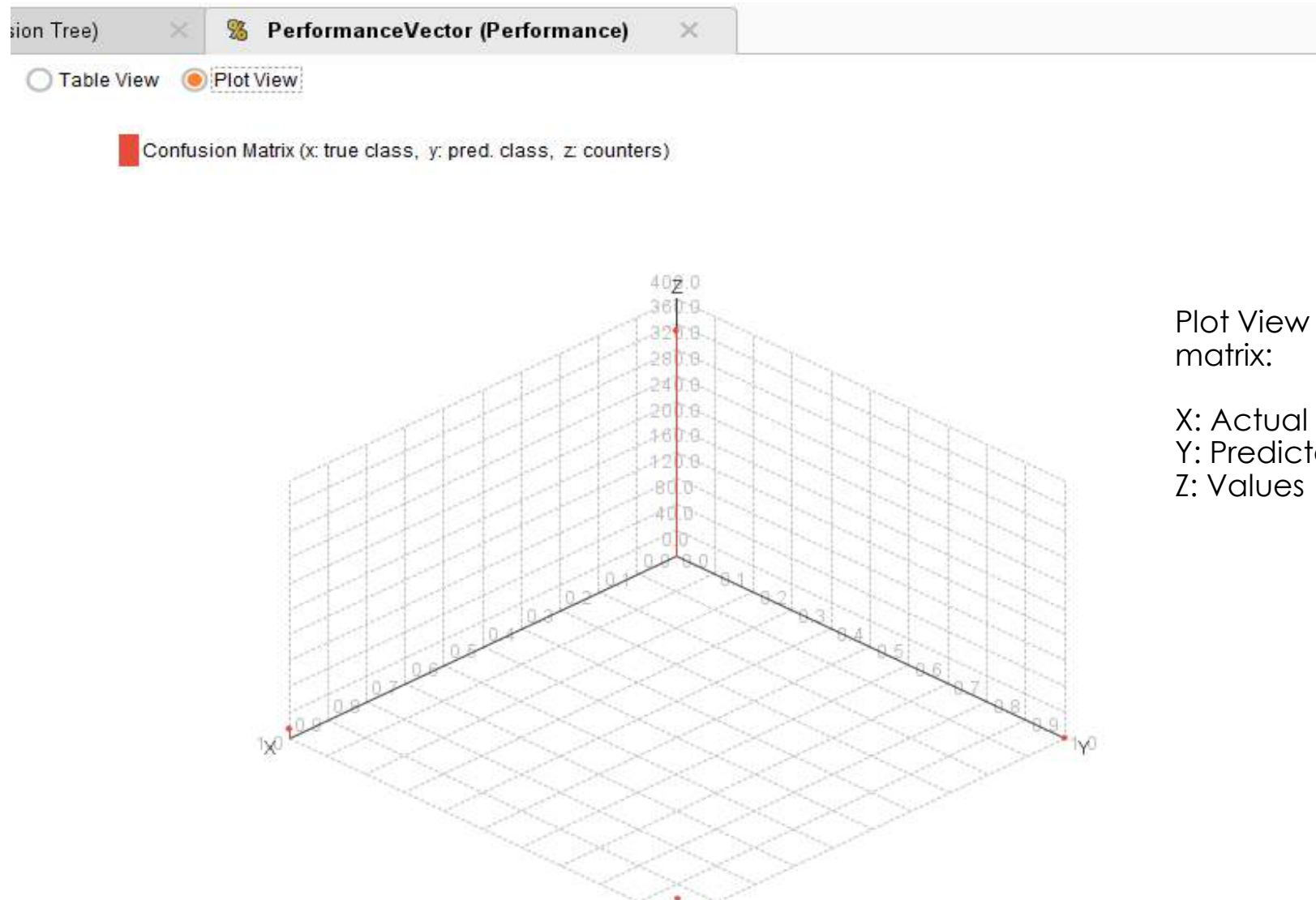
Negative Predictive Value (NPV) can be calculated as: $TN / (TN + FN) = 96.67\%$

Accuracy can be calculated as: $(TP + TN) / (TP + TN + FP + FN) = 96.95\%$

Tree (Decision Tree)		PerformanceVector (Performance)	
Criterion		Table View Plot View	
accuracy		classification_error: 3.05%	
classification error			
	true false	true true	class precision
pred. false	348	12	96.67%
pred. true	0	33	100.00%
class recall	100.00%	73.33%	

Classification error can be calculated as: $(FP + FN) / (TP + TN + FP + FN) = 3.05\%$

HOW GOOD OUR MODEL IS?



Plot View of Performance (Classification) matrix:

X: Actual Values
Y: Predicted Values
Z: Values (or Counters)

BUT, ARE WE SURE? CHECK FOR OVER-FITTING

The screenshot displays the RapidMiner Studio Educational 9.6.000 interface. The main window shows a process design in the 'Design' view. The process starts with a 'Retrieve' operator, followed by a 'Set Role' operator, and then a 'Split Data' operator. The 'Set Role' operator has two outputs: 'exa' and 'ori'. The 'Split Data' operator has two outputs: 'par' and 'par'. The 'Retrieve' operator has an 'out' output. The 'Set Role' operator has an 'exa' input and an 'ori' output. The 'Split Data' operator has an 'exa' input and two 'par' outputs. The 'Retrieve' operator has a green checkmark, indicating it is successful. The 'Set Role' operator has a green checkmark, indicating it is successful. The 'Split Data' operator has a green checkmark, indicating it is successful. The 'Repository' panel on the left shows a list of operators, including 'Building_Decision_Tree_Model_1', 'Decision_Tree_Model_15-04-202', 'Decision_Tree_Model_with_3_Pa', 'DecisionTreeModel-1 (DELL - v1, 4', 'Naive_Bayes_Classification_Mod', 'Naive_Bayes_Model_Try (DELL - v', 'Settings (DELL - v1, 4/17/20 1:40 AM', and 'Simple_Naive_Bayes_Model_Try'. The 'Operators' panel on the left shows a search bar and a list of operator categories: 'Data Access (53)', 'Blending (81)', 'Cleansing (29)', 'Modeling (165)', 'Scoring (14)', 'Validation (30)', 'Utility (85)', and 'Extensions (2)'. The 'Parameters' panel on the right shows the 'Process' operator parameters: 'logverbosity' (init), 'logfile' (empty), 'resultfile' (empty), 'random seed' (2001), 'send mail' (never), and 'encoding' (SYSTEM). The 'Help' panel on the right shows the 'Process' operator synopsis: 'The root operator which is the outer most operator of every process.' and a description: 'Leverage the Wisdom of Crowds to get operator recommendations based on your process design!'. A green checkmark and the text 'Activate Wisdom of Crowds' are visible at the bottom of the process design area.

Repository

- Import Data
- Building_Decision_Tree_Model_1
- Decision_Tree_Model_15-04-202
- Decision_Tree_Model_with_3_Pa
- DecisionTreeModel-1 (DELL - v1, 4
- Naive_Bayes_Classification_Mod
- Naive_Bayes_Model_Try (DELL - v
- Settings (DELL - v1, 4/17/20 1:40 AM
- Simple_Naive_Bayes_Model_Try

Operators

Search for Operators

- Data Access (53)
- Blending (81)
- Cleansing (29)
- Modeling (165)
- Scoring (14)
- Validation (30)
- Utility (85)
- Extensions (2)

Get more operators from the Marketplace

Process

Process

Retrieve

Set Role

Split Data

Parameters

Process

logverbosity: init

logfile:

resultfile:

random seed: 2001

send mail: never

encoding: SYSTEM

Hide advanced parameters

Change compatibility (9.6.000)

Help

Process

RapidMiner Studio Core

Synopsis

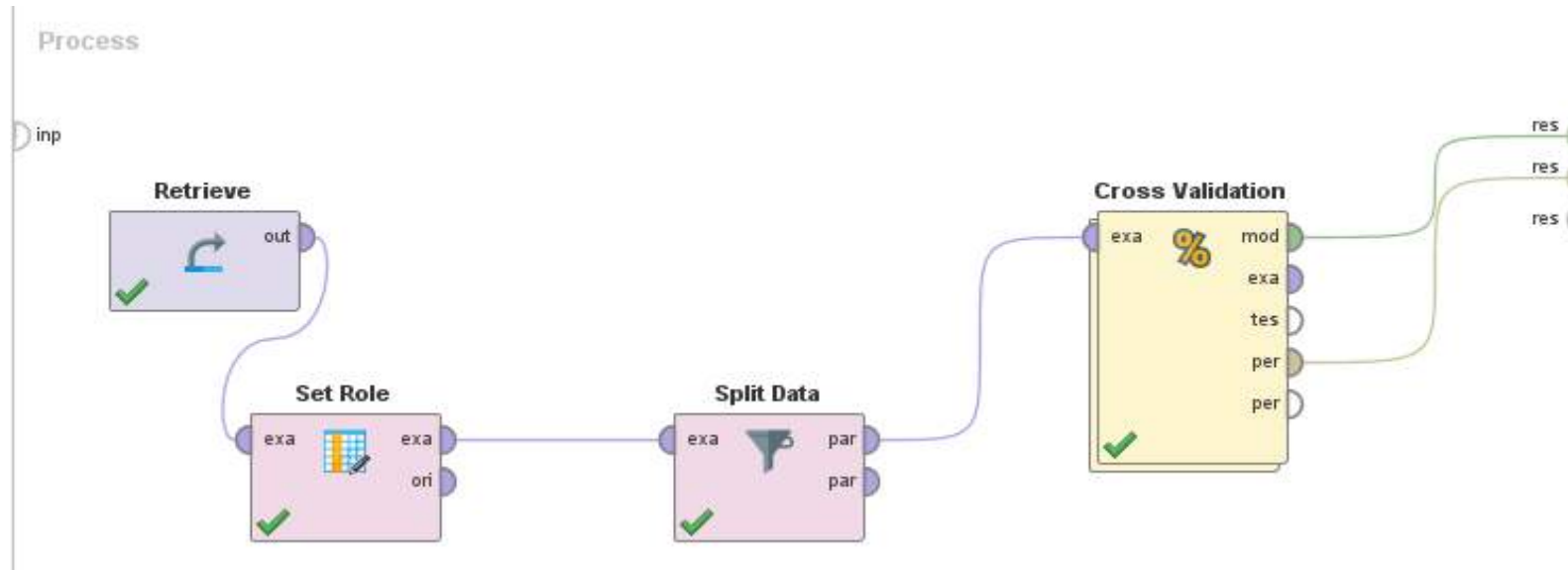
The root operator which is the outer most operator of every process.

Description

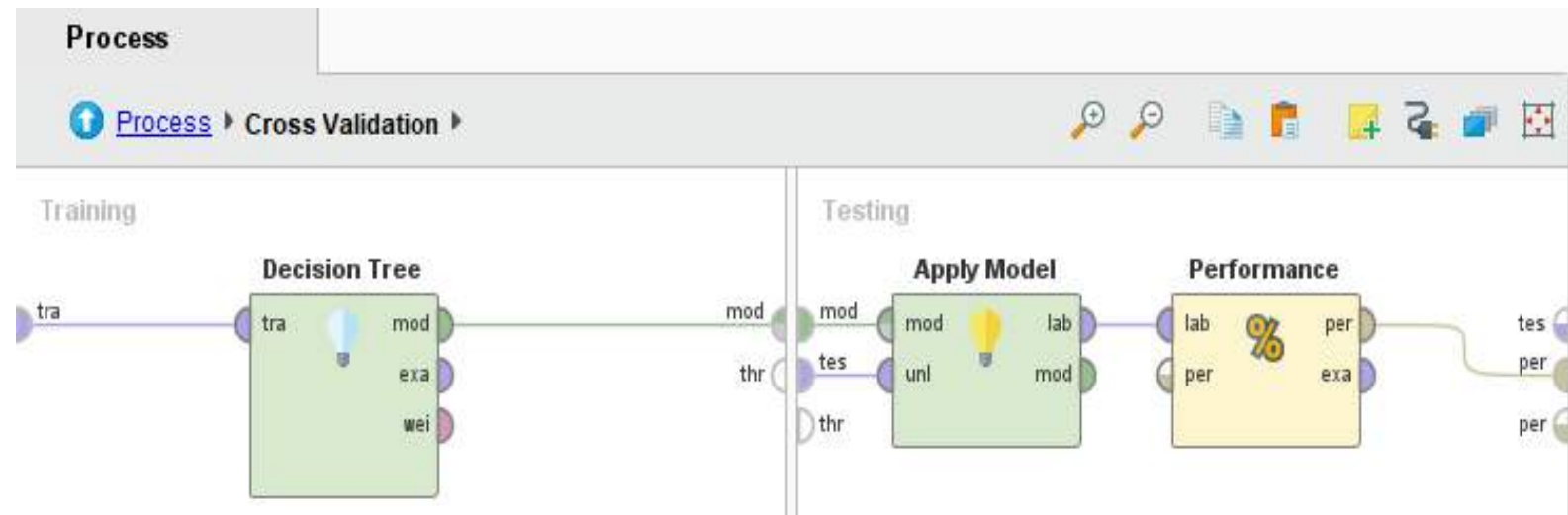
Leverage the Wisdom of Crowds to get operator recommendations based on your process design!

Activate Wisdom of Crowds

USING CROSS VALIDATION



Cross Validation is used to see if the model is over-fitting



ACCURACY RESULT AFTER USING CROSS VALIDATION

☒ Table View ☐ Plot View

accuracy: 96.20% +/- 3.40% (micro average: 96.19%)

	true false	true true	class precision
pred. false	343	12	96.62%
pred. true	3	36	92.31%
class recall	99.13%	75.00%	

There is a little difference in the accuracy of the model after using cross validation

The accuracy **before** using cross validation was **96.95%** whereas **now**, it is **96.20% +/- 3.40%** - hence, we can say that the model might be over fitting

☒ Table View ☐ Plot View

classification_error: 3.80% +/- 3.40% (micro average: 3.81%)

	true false	true true	class precision
pred. false	343	12	96.62%
pred. true	3	36	92.31%
class recall	99.13%	75.00%	

Similarly, there is a little difference in the classification error of the model after using cross validation

The classification error **before** using cross validation was **3.05%** whereas **now**, it is **3.80% +/- 3.40%** - hence, we can really say that the model might have a tendency to over fit

RE-TRAINING THE MODEL

//Local Repository/MyAlarmProject/DecisionTreeModel-1 – RapidMiner Studio Educational 9.6.000 @ JaspreetKaurPC

File Edit Process View Connections Settings Extensions Help

Views: Design Results Turbo Prep Auto Model Deployments

Find data, operators...etc All Studio

Repository

- MyAlarmProject (DELL)
 - 1 (DELL)
 - Myfirst (DELL)
 - MyFirstDeployment (DELL)
 - Building_Decision_Tree_Model_T
 - Decision_Tree_Model_15-04-202
 - Decision_Tree_Model_with_3_Pa
 - DecisionTreeModel-1 (DELL – v1, 4)
 - Naive_Bayes_Classification_Mod

Operators

Search for Operators

- Data Access (53)
- Blending (81)
 - Attributes (47)
 - Names & Roles (7)
 - Types (16)
 - Selection (7)
 - Select Attributes
 - Select by Weights

[Get more operators from the Marketplace](#)

Process

Process

inp

Retrieve

Set Role

Split Data

Cross Validation

res

res

res

Parameters

Process

logverbosity: init

logfile:

resultfile:

random seed: 2001

send mail: never

encoding: SYSTEM

[Hide advanced parameters](#)

[Change compatibility \(9.6.000\)](#)

Help

Select Attributes

RapidMiner Studio Core

Tags: [Filter](#), [Keep](#), [Remove](#), [Drop](#), [Delete](#), [Columns](#), [Variables](#), [Features](#), [Feature Set](#), [Selection](#)

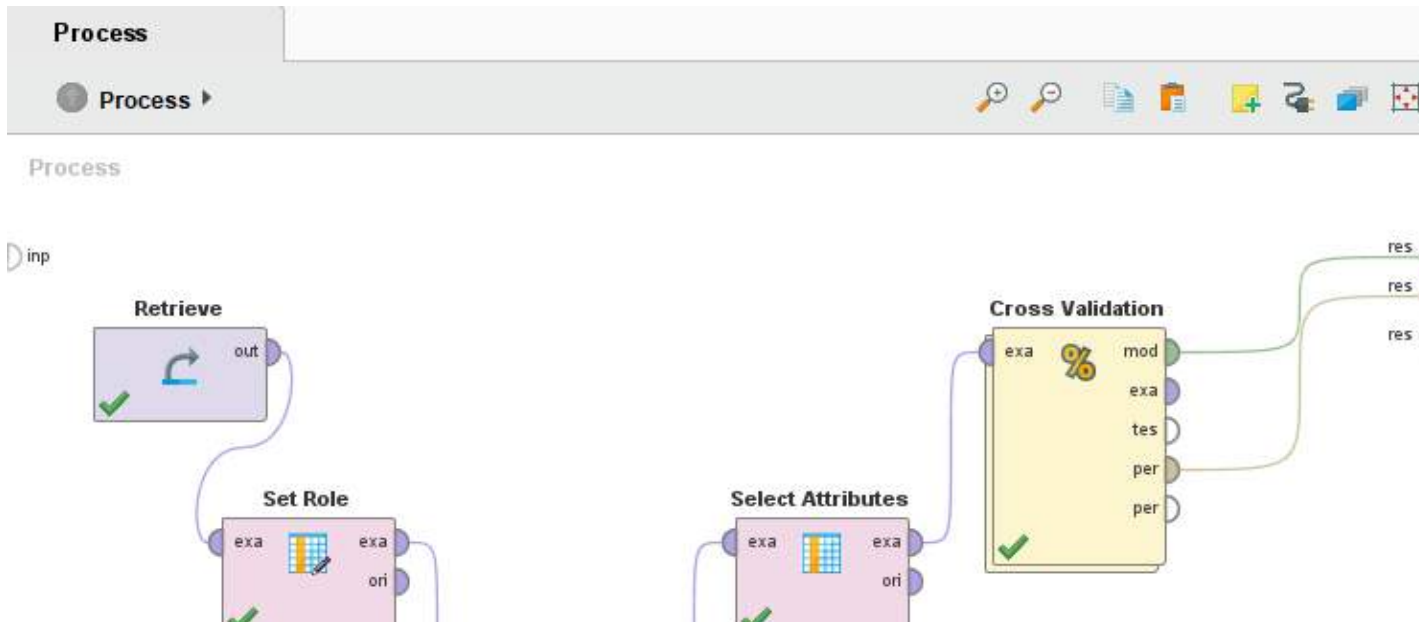
Synopsis

This Operator selects a subset of Attributes of an ExampleSet and removes the other Attributes.

Leverage the Wisdom of Crowds to get operator recommendations based on your process design!

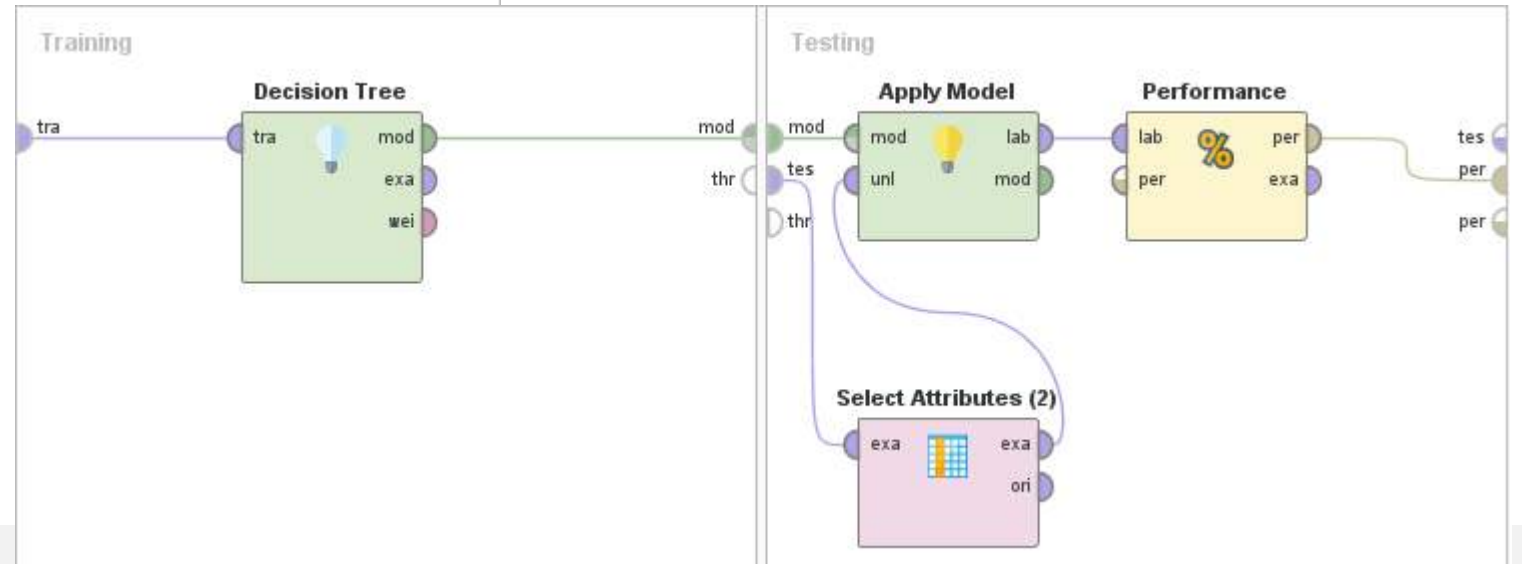
Activate Wisdom of Crowds

REMOVING TIMESTAMP & VIOLATION TYPE

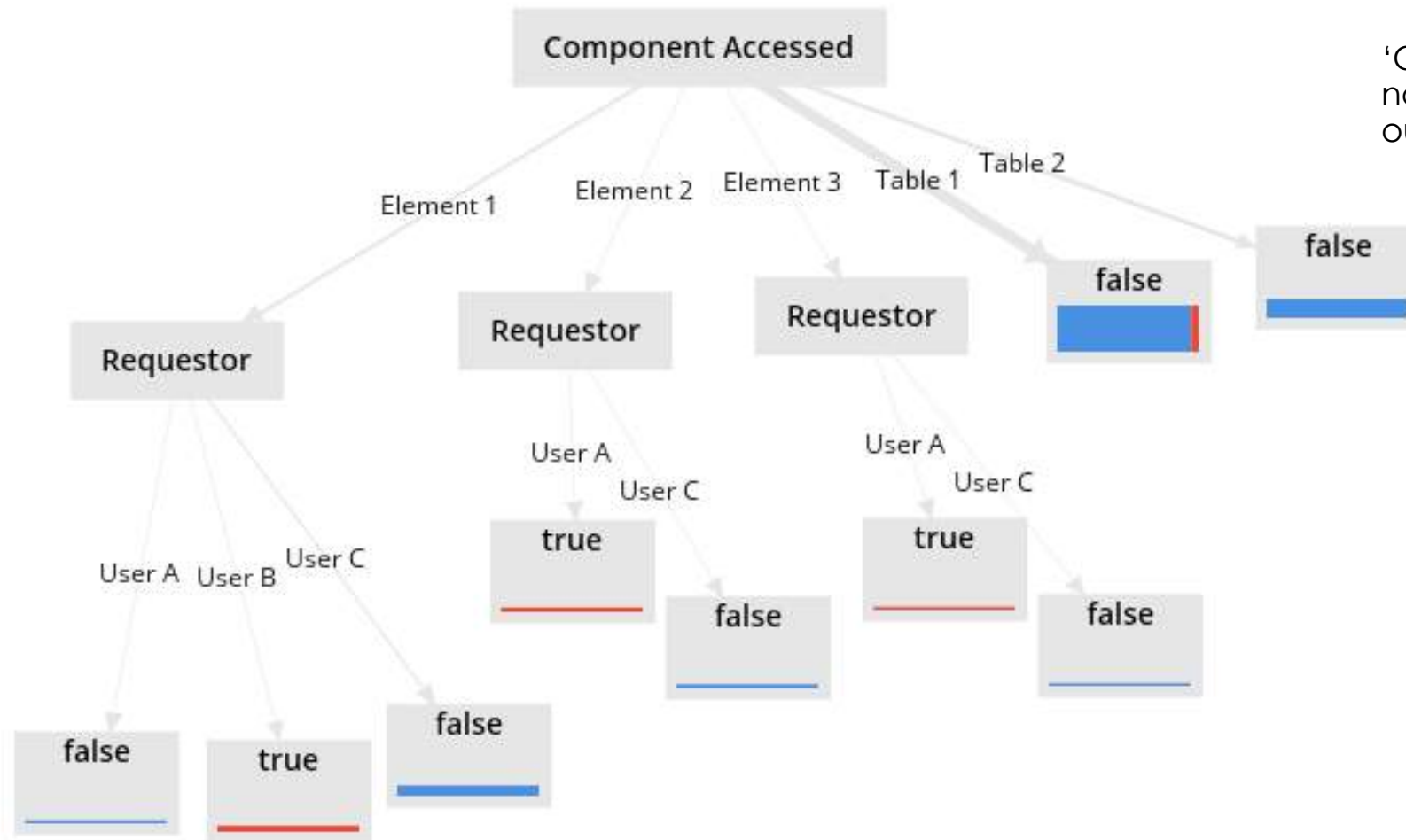


'**Select Attributes**' operator is used to select which attributes one would like to keep or remove in the dataset

As stated in EDA phase, we removed **Timestamp** (to speed up the model building) and **Violation Type** (since the column is practically constant) predictor variables to check if the model performs even better



DECISION TREE



'Component Accessed' (here, root node) seems to be the best predictor out of all the other predictor variables

Figure 1.2 Decision Tree Model (adjusted)

FEEDBACK - AFTER RE-TRAINING THE MODEL

- 12 records show “**No Match**” in the feedback from the test set, out of the 393 records
- The security analysts had stated that the alarms were true (as shown below in yellow) but our model predicts that those alarms were false (as shown below in orange)
- Clearly, the 12 predicted values of Alarm (as shown below) are “**False Negatives**”
- Let's find out how much accurate our model is!



Microsoft Excel
Worksheet

Request	Role	Component Access	Request type	Alarm	confidence(false)	confidence(true)	prediction(Alarm)	Match/No Match
User B	Business user	Table 1	Select	true	0.9	0.1	false	No Match
User B	Business user	Table 1	Select	true	0.9	0.1	false	No Match
User B	Business user	Table 1	Select	true	0.9	0.1	false	No Match
User B	Business user	Table 1	Select	true	0.9	0.1	false	No Match
User B	Business user	Table 1	Select	true	0.9	0.1	false	No Match
User B	Business user	Table 1	Select	true	0.9	0.1	false	No Match
User B	Business user	Table 1	Select	true	0.9	0.1	false	No Match
User B	Business user	Table 1	Select	true	0.9	0.1	false	No Match
User B	Business user	Table 1	Select	true	0.9	0.1	false	No Match
User B	Business user	Table 1	Select	true	0.9	0.1	false	No Match
User B	Business user	Table 1	Select	true	0.9	0.1	false	No Match
User B	Business user	Table 1	Select	true	0.9	0.1	false	No Match

ACCURACY - AFTER RE-TRAINING THE MODEL

☒ Table View ☐ Plot View

accuracy: 96.46% +/- 3.38% (micro average: 96.45%)

	true false	true true	class precision
pred. false	344	12	96.63%
pred. true	2	36	94.74%
class recall	99.42%	75.00%	

☒ Table View ☐ Plot View

After removing **Timestamp** and **Violation Type** predictor variables, the accuracy **increased** from **96.19%** to **96.45%** (if we consider the micro average)

Since, RapidMiner tells us the range of the accuracy of the model, hence, it might be difficult for us to explicitly say if removing Timestamp & Violation Type variables will improve the accuracy

Similarly, the Classification Error after removing **Timestamp** and **Violation Type** variables **reduced** from **3.81%** to **3.55%** - if we consider the micro average method, but this method can sometimes be misleading

classification_error: 3.54% +/- 3.38% (micro average: 3.55%)

	true false	true true	class precision
pred. false	344	12	96.63%
pred. true	2	36	94.74%
class recall	99.42%	75.00%	

APPLYING THE ADJUSTED MODEL ON VALIDATION SET

//Local Repository/MyAlarmProject/New - Final - RapidMiner Studio Educational 9.6.000 © JaspreetKaurPC

File Edit Process View Connections Settings Extensions Help

Views: Design Results Turbo Prep Auto Model Deployments

Find data, operators...etc. All Studio

Repository

- Import Data
- Naive_Bayes_Model_Try (DELL - v1, 4/21/20 1:08)
- New - Final (DELL - v1, 4/21/20 1:08)
- Settings (DELL - v1, 4/20/20 3:43 PM)
- Simple_Naive_Bayes_Model_Try
- processes (DELL)
 - Alarm File (DELL - v1, 3/24/20 1:05 AM)
 - Alarm File - for Test Set (DELL - v1, 4/1)
 - Validating_Decision_Tree_Model_15
- Community Samples (connected)

Operators

apply mod

- Modeling (1)
 - Time Series (1)
 - Forecasting (1)
 - Apply Forecast
 - Scoring (1)
 - Apply Model

No results were found.

Process

Process

Retrieve Alarm File

Set Role

Select Attributes

Cross Validation

Split Data

Parameters

Process

logverbosity: init

logfile:

resultfile:

random seed: 2001

send mail: never

encoding: SYSTEM

[Hide advanced parameters](#)

[Change compatibility \(9.6.000\)](#)

Help

Process

RapidMiner Studio Core

Synopsis

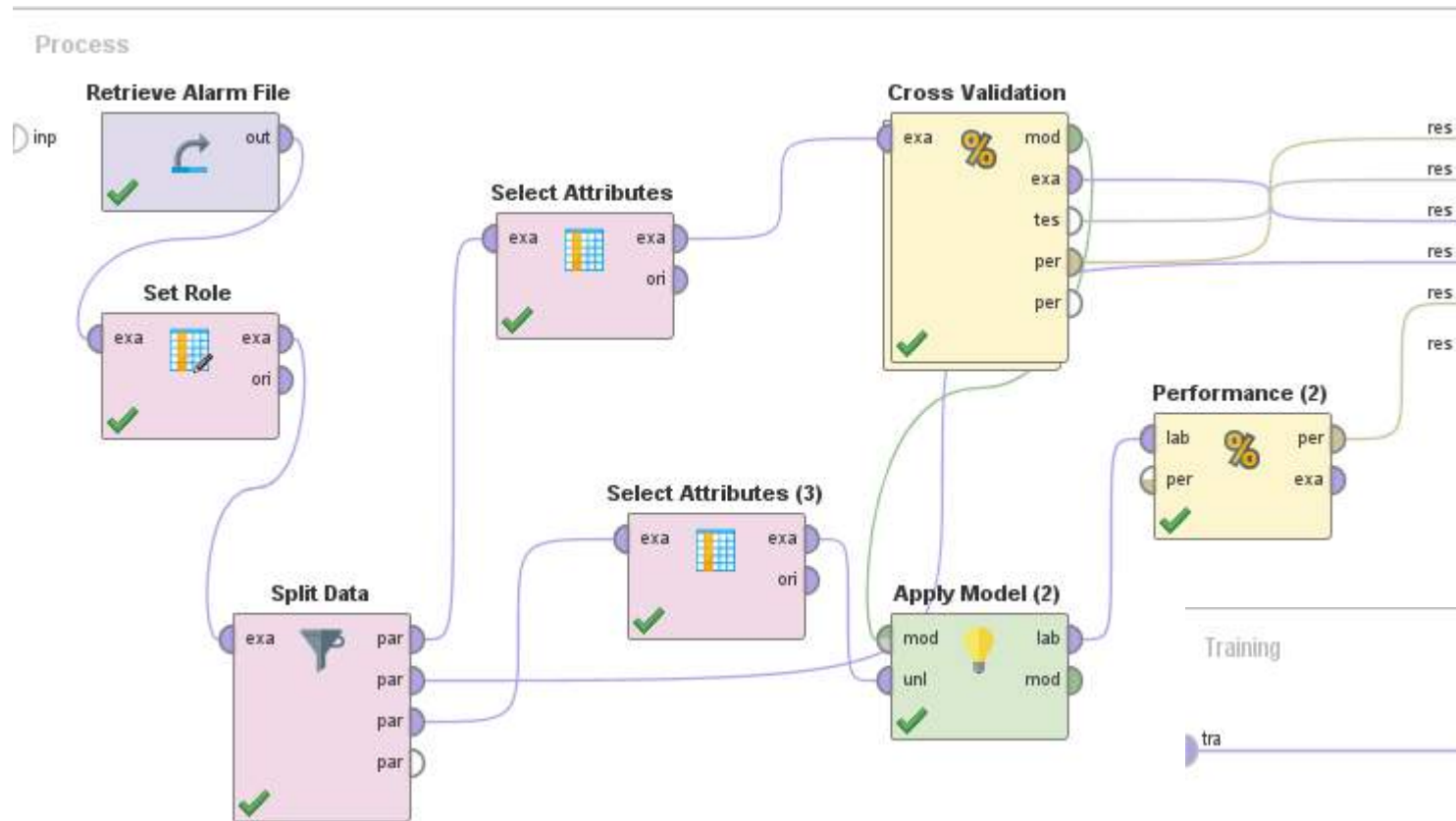
The root operator which is the outer most operator of every process.

Description

Leverage the **Wisdom of Crowds** to get operator recommendations based on your process design!

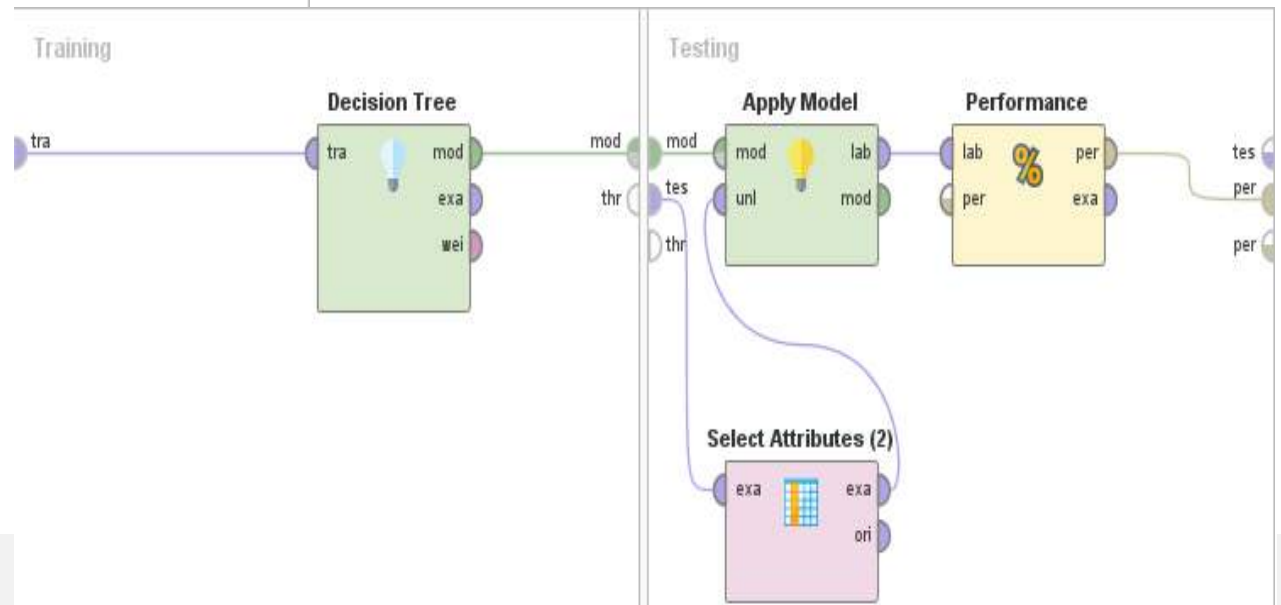
☒ Activate Wisdom of Crowds

DECISION TREE - FINAL DATA MINING MODEL



We finally apply the adjusted model (in which Timestamp & Violation Type variables are removed) to the validation set (as shown in the process editor)

Let's check the feedback from the validation set and the accuracy of the final data mining model!



FEEDBACK FROM THE VALIDATION SET

- 16 records show “**No Match**” in the feedback from the validation set, out of the 406 records
- The security analysts had stated that the alarms were true (as shown below in yellow) but our model predicts that those alarms were false (as shown below in orange)
- Clearly, the 16 predicted values of Alarm (as shown below) are “**False Negatives**” - generated by **Business User (User B)** on **Table 1** using *Select* query
- Let's find out how much accurate our model is!



Microsoft Excel
Worksheet

Request	Role	Component Access	Request type	Alarm	confidence(false)	confidence(true)	prediction(Alarm)	Match/No Match
User B	Business user	Table 1	Select	true	0.9	0.1	false	No Match
User B	Business user	Table 1	Select	true	0.9	0.1	false	No Match
User B	Business user	Table 1	Select	true	0.9	0.1	false	No Match
User B	Business user	Table 1	Select	true	0.9	0.1	false	No Match
User B	Business user	Table 1	Select	true	0.9	0.1	false	No Match
User B	Business user	Table 1	Select	true	0.9	0.1	false	No Match
User B	Business user	Table 1	Select	true	0.9	0.1	false	No Match
User B	Business user	Table 1	Select	true	0.9	0.1	false	No Match
User B	Business user	Table 1	Select	true	0.9	0.1	false	No Match
User B	Business user	Table 1	Select	true	0.9	0.1	false	No Match
User B	Business user	Table 1	Select	true	0.9	0.1	false	No Match
User B	Business user	Table 1	Select	true	0.9	0.1	false	No Match
User B	Business user	Table 1	Select	true	0.9	0.1	false	No Match
User B	Business user	Table 1	Select	true	0.9	0.1	false	No Match
User B	Business user	Table 1	Select	true	0.9	0.1	false	No Match
User B	Business user	Table 1	Select	true	0.9	0.1	false	No Match

PERFORMANCE CLASSIFICATION MATRIX

☒ Table View ☐ Plot View

accuracy: 96.46% +/- 3.38% (micro average: 96.45%)

	true false	true true	class precision
pred. false	344	12	96.63%
pred. true	2	36	94.74%
class recall	99.42%	75.00%	

Test set performance classification result shows **96.46% +/- 3.38%** accuracy

Sensitivity or Recall of all positive classes or **True Positive Rate (TPR)** = **75.00%**

☒ Table View ☐ Plot View

Validation set performance classification result shows **96.06%** accuracy

Sensitivity or Recall of all positive classes or **True Positive Rate (TPR)** = **65.96%**

accuracy: 96.06% classification_error: 3.94%

	true false	true true	class precision
pred. false	359	16	95.73%
pred. true	0	31	100.00%
class recall	100.00%	65.96%	

DECISION TREE - GOOD MODEL OR NOT?



The decision tree model gives us an **overall accuracy** of **96.46% +/- 3.38%** on the **test set** and **96.06%** on the **validation set**



The assumption that alarm inspection team should focus on inspecting the alerts generated by Business User on Table 1 (since the number of True alarms generated by this user or role is the highest), from the EDA phase is **confirmed**



The assumption that CISO should investigate alarm process & system for Administrator role (since the number of False alarms generated by this user or role is the highest), from the EDA phase is **rejected**.



Removing Timestamp definitely increases the execution time of the model (0s execution time)



Removing Violation Type has no impact on the decision tree model, since the column is unary or constant



The model chooses only one variable out of Role and **Requestor**. Here, both the variables are the same and hence, we can say that they are perfectly correlated since one user has only one role and hence, the model omits Role - but in real life, many different users can have one role










Component Accessed has strong correlation with Requestor

Model Accuracy: 93.00% to 96.06%

A good model!



NAÏVE BAYES

-  A popular **supervised** classification method used in data mining, which comes under the Bayesian Classification
-  Uses probability for doing predictive analysis
-  Works on the assumption that the predictor variables are independent - but is not so naïve!
-  A low-variance classifier and works well even on small data sets
-  Tool used for building the Naïve Bayes classification model:
RapidMiner 
-  RapidMiner is a very effective data science software platform that unites data prep, machine learning & predictive model deployment

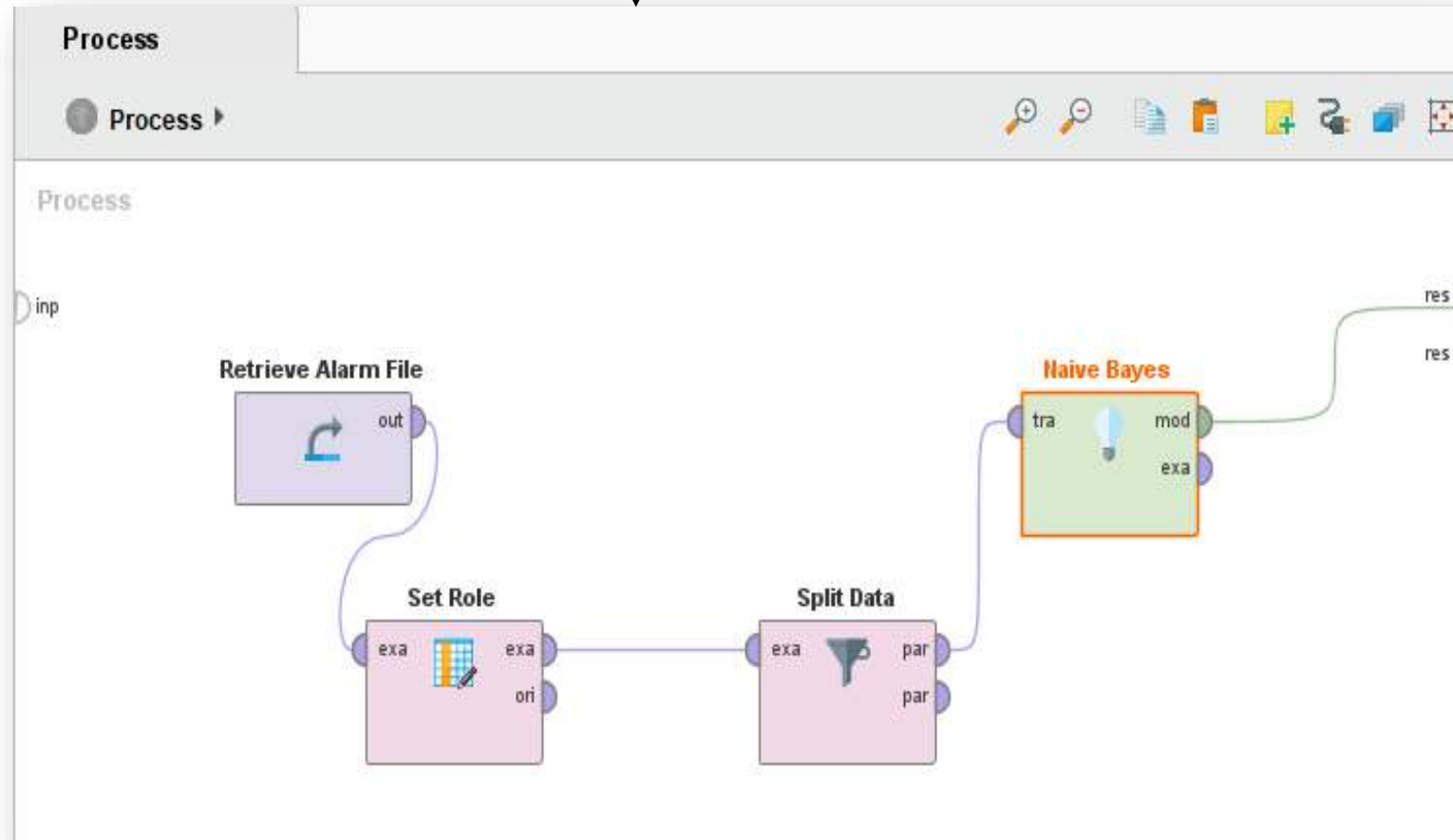
Recommendation in EDA Phase:

....

Looks like it is simple to use and seems to provide an accuracy of 96.5%

NAÏVE BAYES - USING THE TRAINING SET

Process Editor



The first step is to retrieve the data i.e. Alarm file

'Set Role' operator is used to tell RapidMiner which is our target variable (Alarm)

To partition the data set, we use 'Split Data' operator

Dataset is split into 3 sets: Training, Test & Validation datasets (1/3rd each)

Now, we use 'Naïve Bayes' operator to build the provisional model

We connect the inputs with their relevant outputs and run the process

NAÏVE BAYES CLASSIFICATION

SimpleDistribution

Distribution model for label attribute Alarm

Class false (0.878)

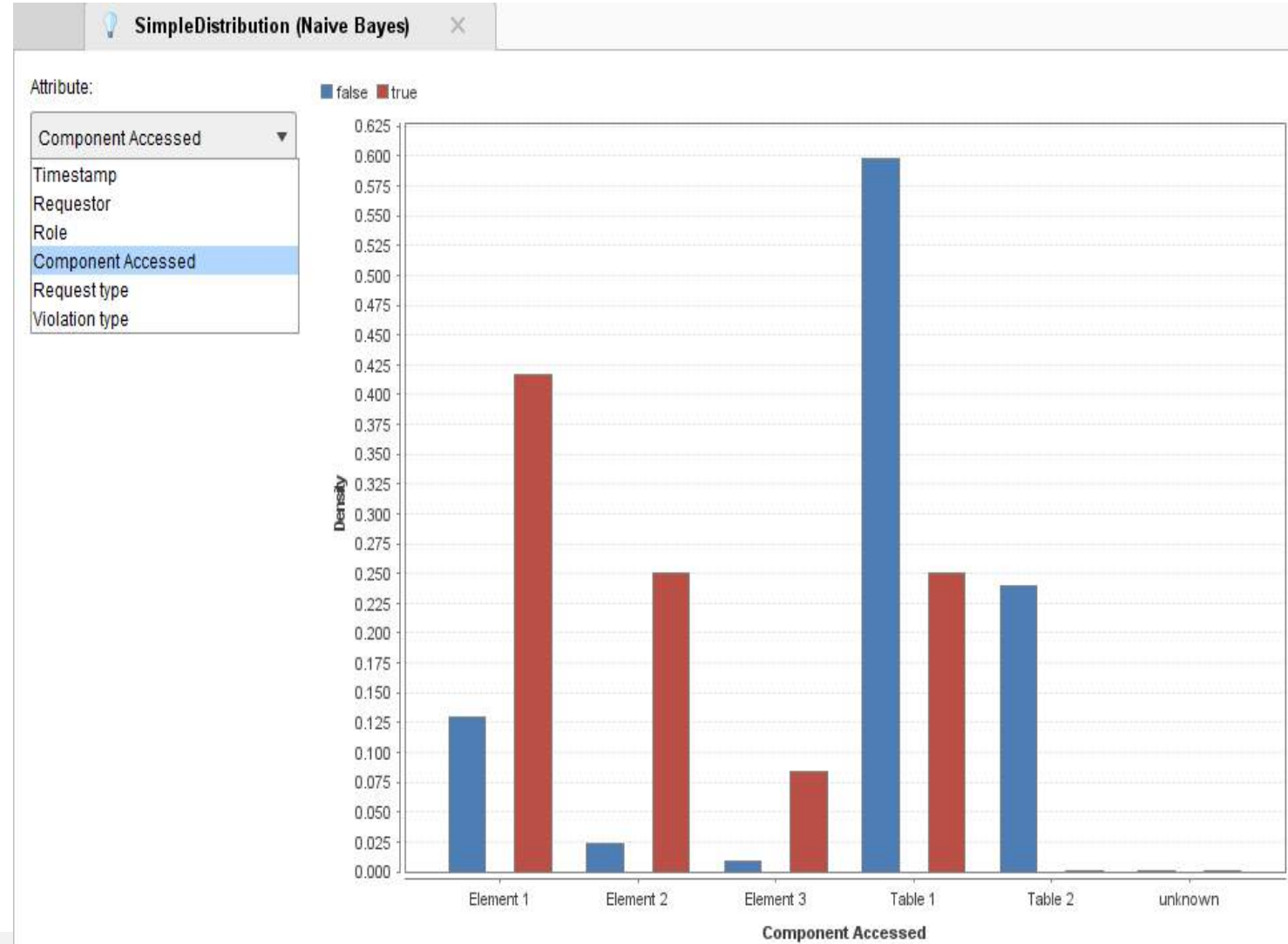
6 distributions

Class true (0.122)

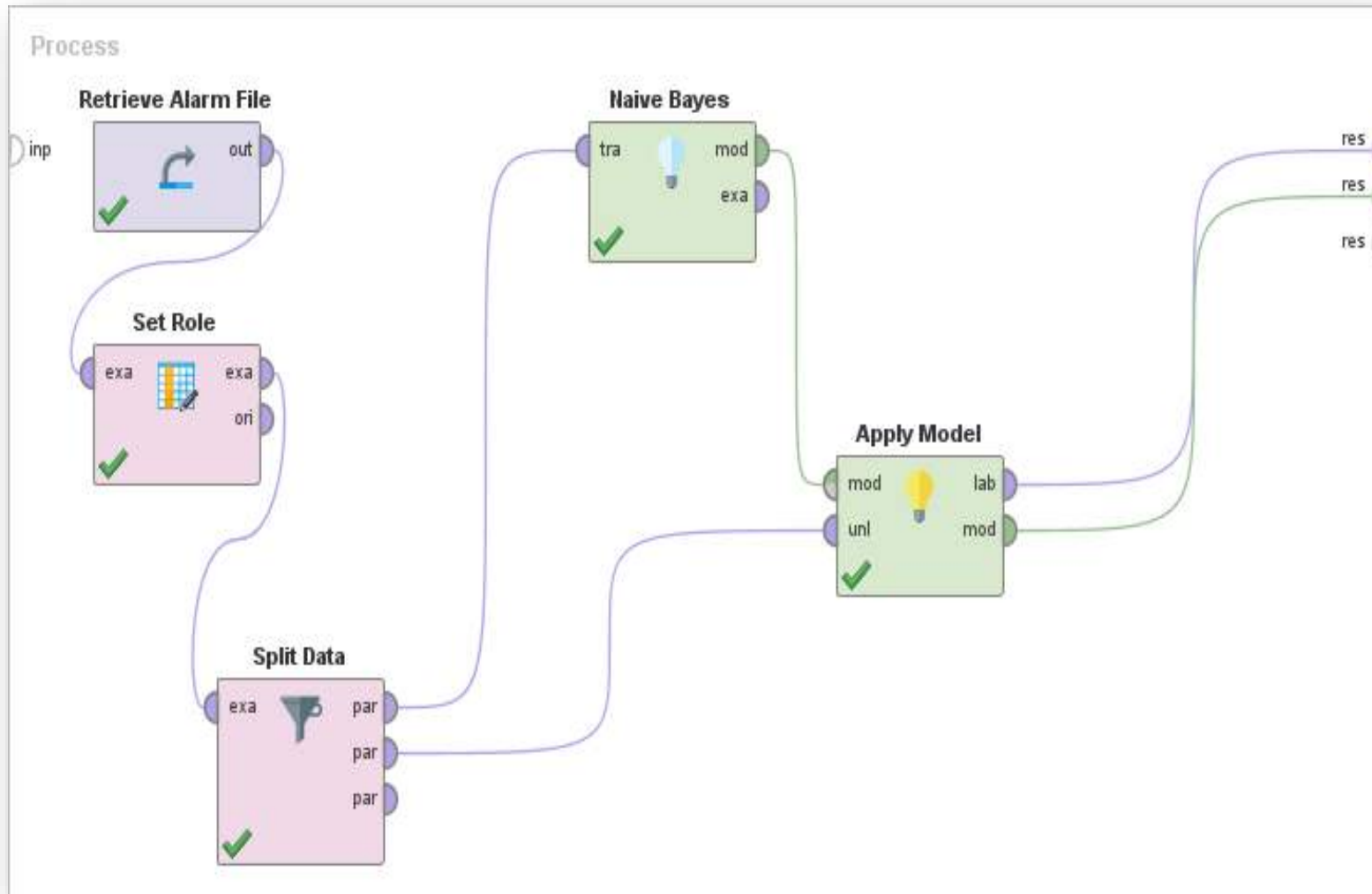
6 distributions

Naïve Bayes Classification Model gives the overall probability of true and false classes

It also shows the individual probability of true and false classes in each of the predictor variables – as shown in the chart on the right



NAÏVE BAYES - USING THE TEST SET



To apply the provisional data mining model to the test set, we use 'Apply Model' operator

'Apply Model' operator requires two inputs:
i) Model ii) Unlabeled data

The 'Apply Model' operator pretends as if 33.33% of the data (which comes from the test set) is unlabeled and applies the model to create the labels (false/true alarms)

And, the model is going to come from the 'output' port of the Naïve Bayes operator

FEEDBACK FROM THE TEST SET

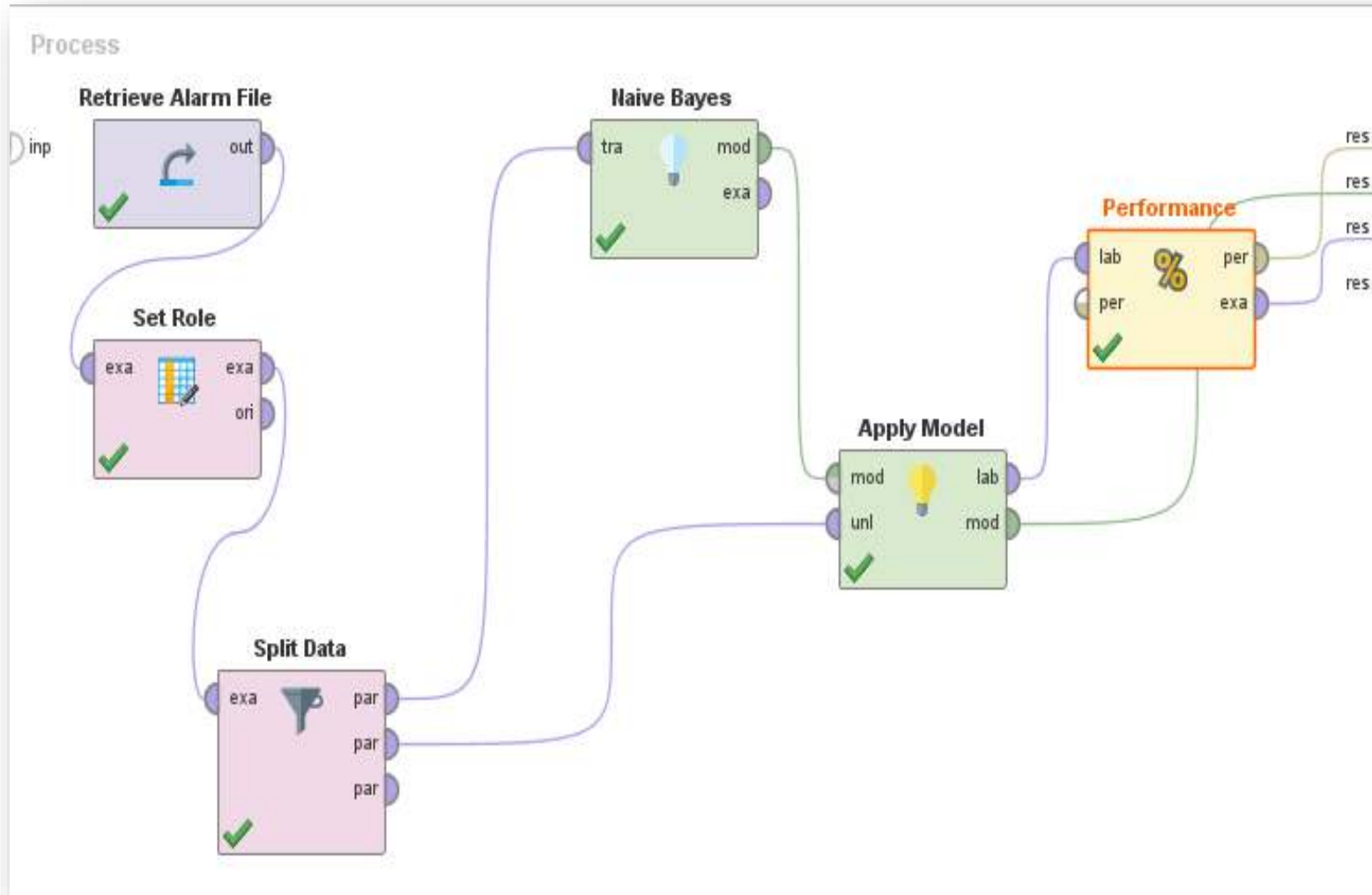
- 17 records show “**No Match**” in the feedback from the test set, out of the 393 records
- The security analysts had stated that the alarms were true (as shown below in yellow) but our model predicts that those alarms were false (as shown below in orange)
- Clearly, the 17 actual values of Alarm (as shown below) are “**False Negatives**” – out of which 8 were generated by Business User (or User B) on Table 1 using Select query and 9 were generated by Analyst (or User A) on Element 2 (using Select) and Element 3 (using Append)
- Let's find out the accuracy of the model!



Microsoft Excel
Worksheet

Timestamp	Request	Role	Component Accessed	Request type	Violation type	Alarm	confidence(false)	confidence(true)	prediction(Alarm)	Match/No Match
2016-08-14 18:29:00	User B	Business user	Table 1	Select	No authorization	true	0.6	0.4	false	No Match
2016-08-14 22:05:00	User B	Business user	Table 1	Select	No authorization	true	0.6	0.4	false	No Match
2016-08-14 22:19:00	User B	Business user	Table 1	Select	No authorization	true	0.6	0.4	false	No Match
2016-08-14 22:34:00	User B	Business user	Table 1	Select	No authorization	true	0.6	0.4	false	No Match
2016-08-14 23:31:00	User A	Analyst	Element 2	Select	No authorization	true	0.6	0.4	false	No Match
2016-08-14 23:46:00	User A	Analyst	Element 2	Select	No authorization	true	0.6	0.4	false	No Match
2016-08-15 00:00:00	User A	Analyst	Element 2	Select	No authorization	true	0.6	0.4	false	No Match
2016-08-15 00:15:00	User A	Analyst	Element 3	Append	No authorization	true	0.7	0.3	false	No Match
2016-08-15 22:34:00	User B	Business user	Table 1	Select	No authorization	true	0.5	0.5	false	No Match
2016-08-16 02:10:00	User B	Business user	Table 1	Select	No authorization	true	0.5	0.5	false	No Match
2016-08-16 02:24:00	User B	Business user	Table 1	Select	No authorization	true	0.5	0.5	false	No Match
2016-08-16 02:39:00	User B	Business user	Table 1	Select	No authorization	true	0.5	0.5	false	No Match
2016-08-16 03:36:00	User A	Analyst	Element 2	Select	No authorization	true	0.5	0.5	false	No Match
2016-08-16 03:51:00	User A	Analyst	Element 2	Select	No authorization	true	0.5	0.5	false	No Match
2016-08-16 04:05:00	User A	Analyst	Element 2	Select	No authorization	true	0.5	0.5	false	No Match
2016-08-16 04:19:00	User A	Analyst	Element 3	Append	No authorization	true	0.6	0.4	false	No Match
2016-08-17 08:24:00	User A	Analyst	Element 3	Append	No authorization	true	0.5	0.5	false	No Match

NAÏVE BAYES - ACCURACY CHECK



'Performance (Classification)' Operator is used to evaluate the model that we are building

'Performance (Classification) Operator' has one mandatory input: labeled data (which comes from the output port of the Apply Model

We get a Performance (Classification) matrix and a distribution table showing probabilities, when we run the process

This table (or matrix) is also called 'Confusion Matrix' as it describes the performance of a classification model (naïve bayes in our case) on a set of test data for which the true values are known

HOW GOOD OUR MODEL IS?

95.67% accuracy! The model is not bad!

et (Apply Model) × SimpleDistribution (Naive Bayes) × PerformanceVector (Performance) ×

☒ Table View ☐ Plot View

Actual (or True) Values

accuracy: 95.67%

Predicted Values ↓	true false	true true	class precision
pred. false	348 (TN)	17 (FN) - Type II error	95.34%
pred. true	0 (FP) - Type I error	28 (TP)	100.00%
class recall	100.00%	62.22%	

Here, True Negative (TN) = 348
False Negative (FN) = 17
False Positive (FP) = 0
True Positive (TP) = 28

Hence, **True Positive Rate (TPR)** can be calculated as: $TP / (TP + FN) = 62.22\%$

True Negative Rate (TNR) can be calculated as: $TN / (TN + FP) = 100.00\%$

Positive Predictive Value (PPV)

can be calculated as: $TP / (TP + FP) = 100.00\%$

Negative Predictive Value (NPV)

can be calculated as: $TN / (TN + FN) = 95.34\%$

Accuracy can be calculated as: $(TP + TN) / (TP + TN + FP + FN) = 95.67\%$

t (Apply Model) × SimpleDistribution (Naive Bayes) × PerformanceVector (Performance) ×

☒ Table View ☐ Plot View

classification_error: 4.33%

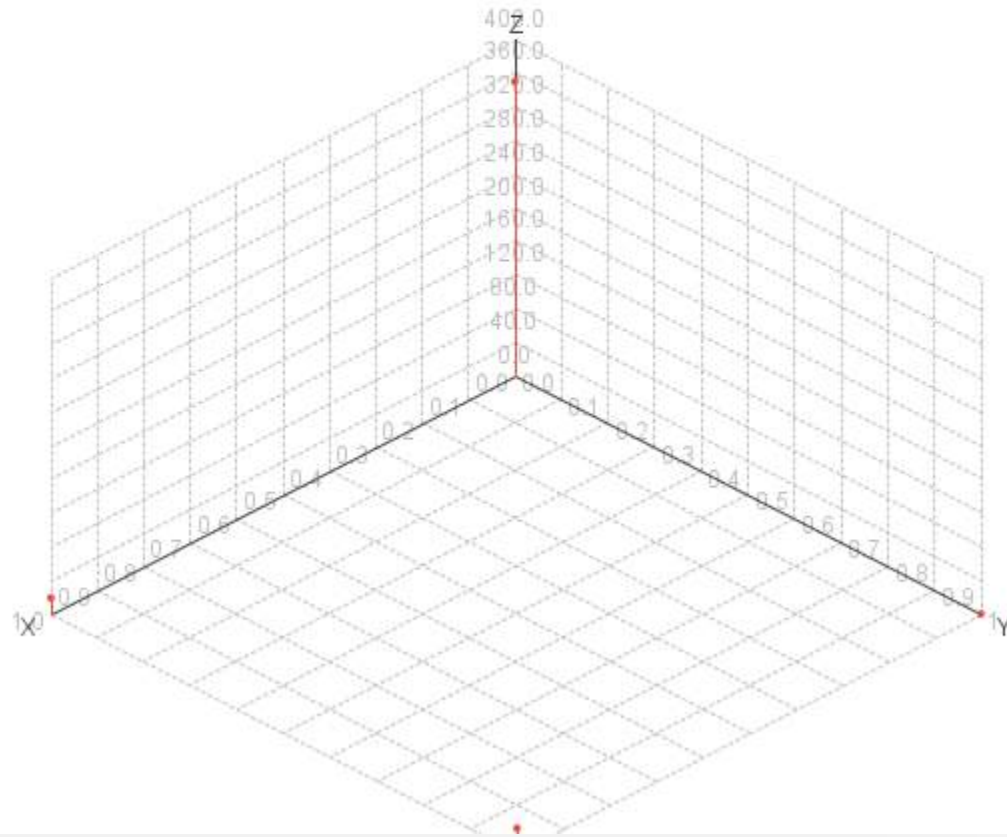
	true false	true true	class precision
pred. false	348	17	95.34%
pred. true	0	28	100.00%
class recall	100.00%	62.22%	

Classification error can be calculated as: $(FP + FN) / (TP + TN + FP + FN) = 4.33\%$

HOW GOOD OUR MODEL IS?

☐ Table View ☒ Plot View

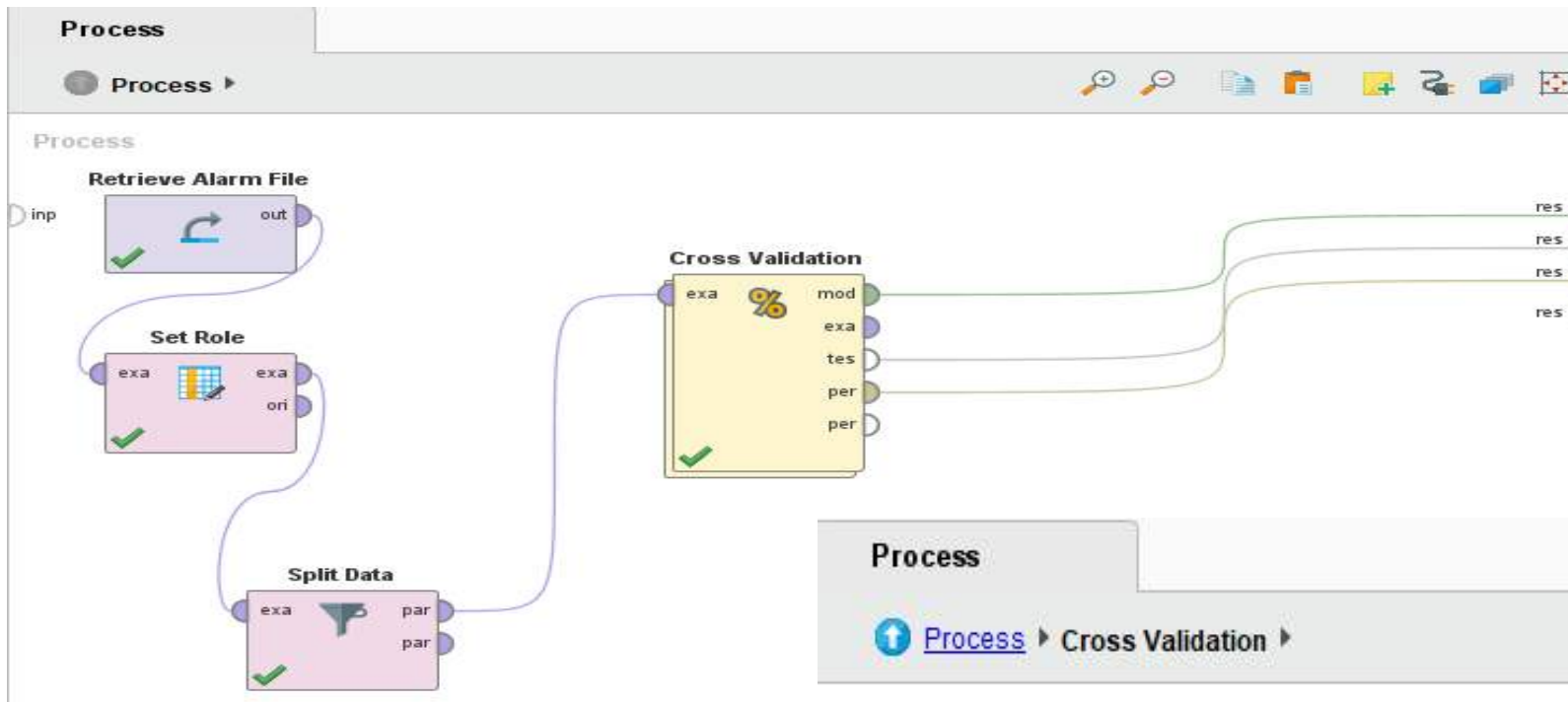
Confusion Matrix (x: true class, y: pred. class, z: counters)



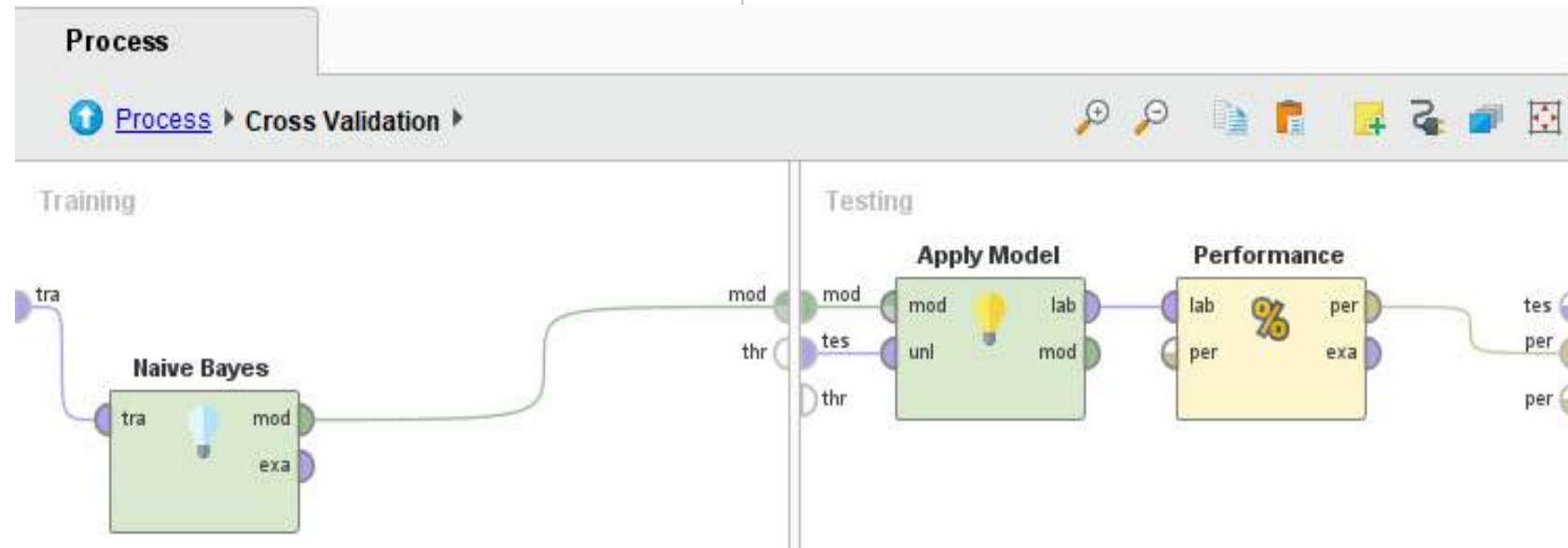
Plot View of Performance (Classification) matrix:

X: Actual Values
Y: Predicted Values
Z: Values (or Counters)

USING CROSS VALIDATION TO CHECK OVER-FITTING



Cross Validation is used to see if the model is over-fitting



FEEDBACK FROM THE TEST SET

Timestamp	Request	Role	Component Access	Request ty	Violation type	confidence(fals	Ala	confidence(tru	prediction(Alar	Match/No Ma
2016-08-11 10:05:00	User B	Business use	Table 1	Select	No authorization	0.7	true	0.3	false	No Match
2016-08-12 14:24:00	User B	Business use	Table 1	Select	No authorization	0.7	true	0.3	false	No Match
2016-08-13 19:55:00	User A	Analyst	Element 2	Select	No authorization	0.7	true	0.3	false	No Match
2016-08-12 15:51:00	User A	Analyst	Element 2	Select	No authorization	0.7	true	0.3	false	No Match
2016-08-10 07:41:00	User A	Analyst	Element 2	Select	No authorization	0.8	true	0.2	false	No Match
2016-08-11 11:17:00	User A	Analyst	Element 2	Select	No authorization	0.8	true	0.2	false	No Match
2016-08-13 19:27:00	User A	Analyst	Element 2	Select	No authorization	0.8	true	0.2	false	No Match
2016-08-13 19:41:00	User A	Analyst	Element 2	Select	No authorization	0.8	true	0.2	false	No Match
2016-08-10 07:12:00	User A	Analyst	Element 2	Select	No authorization	0.7	true	0.3	false	No Match
2016-08-10 07:55:00	User A	Analyst	Element 3	Append	No authorization	0.9	true	0.1	false	No Match
2016-08-13 14:24:00	User B	Business use	Table 1	Select	No authorization	0.6	true	0.4	false	No Match
2016-08-12 15:22:00	User A	Analyst	Element 2	Select	No authorization	0.7	true	0.3	false	No Match
2016-08-12 15:36:00	User A	Analyst	Element 2	Select	No authorization	0.7	true	0.3	false	No Match
2016-08-11 10:19:00	User B	Business use	Table 1	Select	No authorization	0.7	true	0.3	false	No Match
2016-08-13 20:10:00	User A	Analyst	Element 3	Append	No authorization	0.8	true	0.2	false	No Match
2016-08-11 11:46:00	User A	Analyst	Element 2	Select	No authorization	0.7	true	0.3	false	No Match
2016-08-11 12:00:00	User A	Analyst	Element 3	Append	No authorization	0.9	true	0.1	false	No Match
2016-08-12 13:55:00	User B	Business use	Table 1	Select	No authorization	0.7	true	0.3	false	No Match
2016-08-10 07:27:00	User A	Analyst	Element 2	Select	No authorization	0.7	true	0.3	false	No Match
2016-08-11 06:15:00	User B	Business use	Table 1	Select	No authorization	0.8	true	0.2	false	No Match
2016-08-11 09:51:00	User B	Business use	Table 1	Select	No authorization	0.8	true	0.2	false	No Match
2016-08-12 10:19:00	User B	Business use	Table 1	Select	No authorization	0.7	true	0.3	false	No Match
2016-08-11 11:31:00	User A	Analyst	Element 2	Select	No authorization	0.7	true	0.3	false	No Match
2016-08-12 14:10:00	User B	Business use	Table 1	Select	No authorization	0.7	true	0.3	false	No Match
2016-08-12 16:05:00	User A	Analyst	Element 3	Append	No authorization	0.8	true	0.2	false	No Match
2016-08-13 18:00:00	User B	Business use	Table 1	Select	No authorization	0.7	true	0.3	false	No Match
2016-08-13 18:15:00	User B	Business use	Table 1	Select	No authorization	0.7	true	0.3	false	No Match
2016-08-13 18:29:00	User B	Business use	Table 1	Select	No authorization	0.7	true	0.3	false	No Match

- 28 records show **"No Match"** in the feedback from the test set after cross validation
- The security analysts had stated that the alarms were true (as shown in yellow) but our model predicts that those alarms were false (as shown in orange)
- Clearly, the 28 predicted values of Alarm (as shown below) are "False Negatives"
- Since, the model over fitted, hence, we now have more records showing "No Match"
- Let's find out how much accurate our model prediction is after using cross validation!



Microsoft Excel
Worksheet

ACCURACY RESULT

☒ Table View ☐ Plot View

accuracy: 92.90% +/- 2.61% (micro average: 92.89%)

	true false	true true	class precision
pred. false	346	28	92.51%
pred. true	0	20	100.00%
class recall	100.00%	41.67%	

Clearly, there is a lot of difference in the accuracy after doing cross validation

The accuracy before using cross validation was 95.67% whereas now, it is 92.89% - if we consider the micro average method, hence, we can say that the model is over fitting

☒ Table View ☐ Plot View

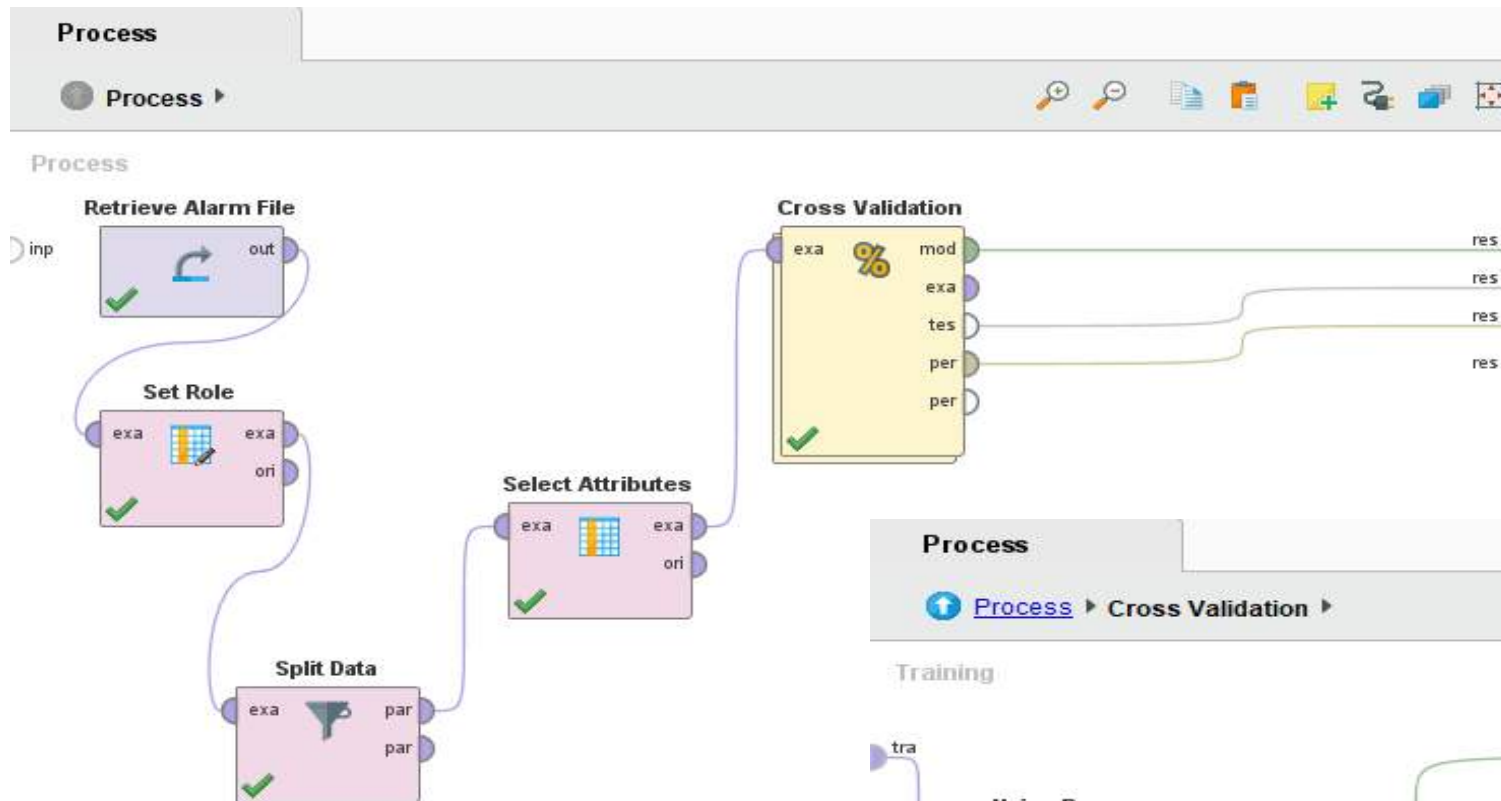
classification_error: 7.10% +/- 2.61% (micro average: 7.11%)

	true false	true true	class precision
pred. false	346	28	92.51%
pred. true	0	20	100.00%
class recall	100.00%	41.67%	

Similarly, there is again a lot of difference in the classification error of the model after using cross validation

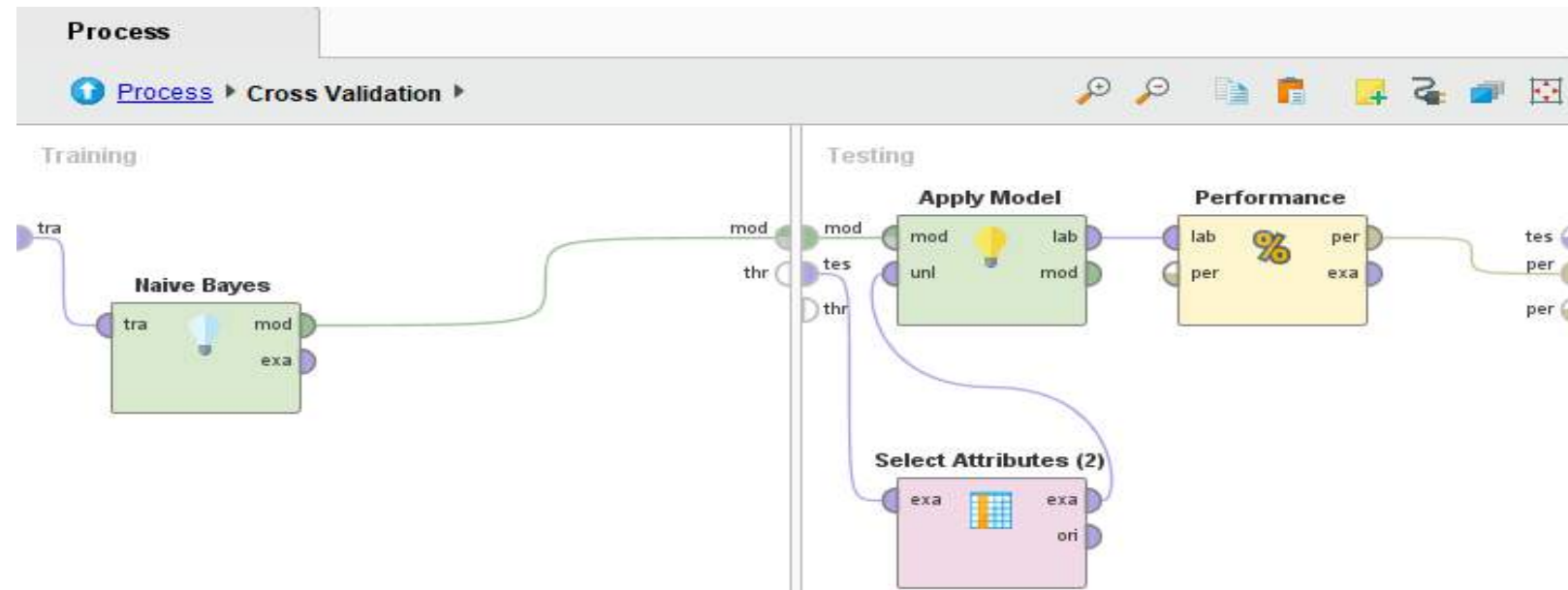
The classification error before using cross validation was 4.33% whereas now, it is 7.11% - if we consider the micro average method, hence, we can say that the model over fits

RE-TRAINING THE MODEL



'Select Attributes' operator is used to select which attributes one would like to keep or remove in the dataset

As stated in EDA phase, we removed Timestamp (to speed up the model building) and Violation Type (since the column is practically constant) predictor variables to check if the model performs even better



FEEDBACK FROM THE TEST SET

Request	Role	Component Access	Request ty	confidence(tru	confidence(fals	prediction(Alarm	Ala	Match/No Ma
User B	Business use	Table 1	Select	0.3	0.7	false	true	No Match
User B	Business use	Table 1	Select	0.3	0.7	false	true	No Match
User A	Analyst	Element 2	Select	0.3	0.7	false	true	No Match
User A	Analyst	Element 2	Select	0.3	0.7	false	true	No Match
User A	Analyst	Element 2	Select	0.2	0.8	false	true	No Match
User A	Analyst	Element 2	Select	0.2	0.8	false	true	No Match
User A	Analyst	Element 2	Select	0.2	0.8	false	true	No Match
User A	Analyst	Element 2	Select	0.2	0.8	false	true	No Match
User A	Analyst	Element 2	Select	0.3	0.7	false	true	No Match
User A	Analyst	Element 3	Append	0.1	0.9	false	true	No Match
User B	Business use	Table 1	Select	0.3	0.7	false	true	No Match
User A	Analyst	Element 2	Select	0.3	0.7	false	true	No Match
User A	Analyst	Element 2	Select	0.3	0.7	false	true	No Match
User B	Business use	Table 1	Select	0.3	0.7	false	true	No Match
User A	Analyst	Element 3	Append	0.1	0.9	false	true	No Match
User A	Analyst	Element 2	Select	0.3	0.7	false	true	No Match
User A	Analyst	Element 3	Append	0.1	0.9	false	true	No Match
User B	Business use	Table 1	Select	0.3	0.7	false	true	No Match
User A	Analyst	Element 2	Select	0.4	0.6	false	true	No Match
User B	Business use	Table 1	Select	0.3	0.7	false	true	No Match
User B	Business use	Table 1	Select	0.3	0.7	false	true	No Match
User B	Business use	Table 1	Select	0.3	0.7	false	true	No Match
User A	Analyst	Element 2	Select	0.3	0.7	false	true	No Match
User B	Business use	Table 1	Select	0.3	0.7	false	true	No Match
User A	Analyst	Element 3	Append	0.2	0.8	false	true	No Match
User B	Business use	Table 1	Select	0.3	0.7	false	true	No Match
User B	Business use	Table 1	Select	0.3	0.7	false	true	No Match
User B	Business use	Table 1	Select	0.3	0.7	false	true	No Match

- 28 records show “No Match” in the feedback from the test set after cross validation
- The security analysts had stated that the alarms were true (as shown in yellow) but our model predicts that those alarms were false (as shown in orange)
- Clearly, the 28 predicted values of Alarm (as shown below) are “False Negatives”
- Let's find out how much accurate our model prediction is after removing Timestamp & Violation Type variables



Microsoft Excel
Worksheet

PERFORMANCE RESULT

☒ Table View ☐ Plot View

accuracy: 92.90% +/- 2.61% (micro average: 92.89%)

	true false	true true	class precision
pred. false	346	28	92.51%
pred. true	0	20	100.00%
class recall	100.00%	41.67%	

Clearly, there is no difference in the accuracy of the model even after retraining the provisional model

The accuracy of the model still remains 92.89% - which is quite low and hence, we might want to rethink before considering such a model

☒ Table View ☐ Plot View

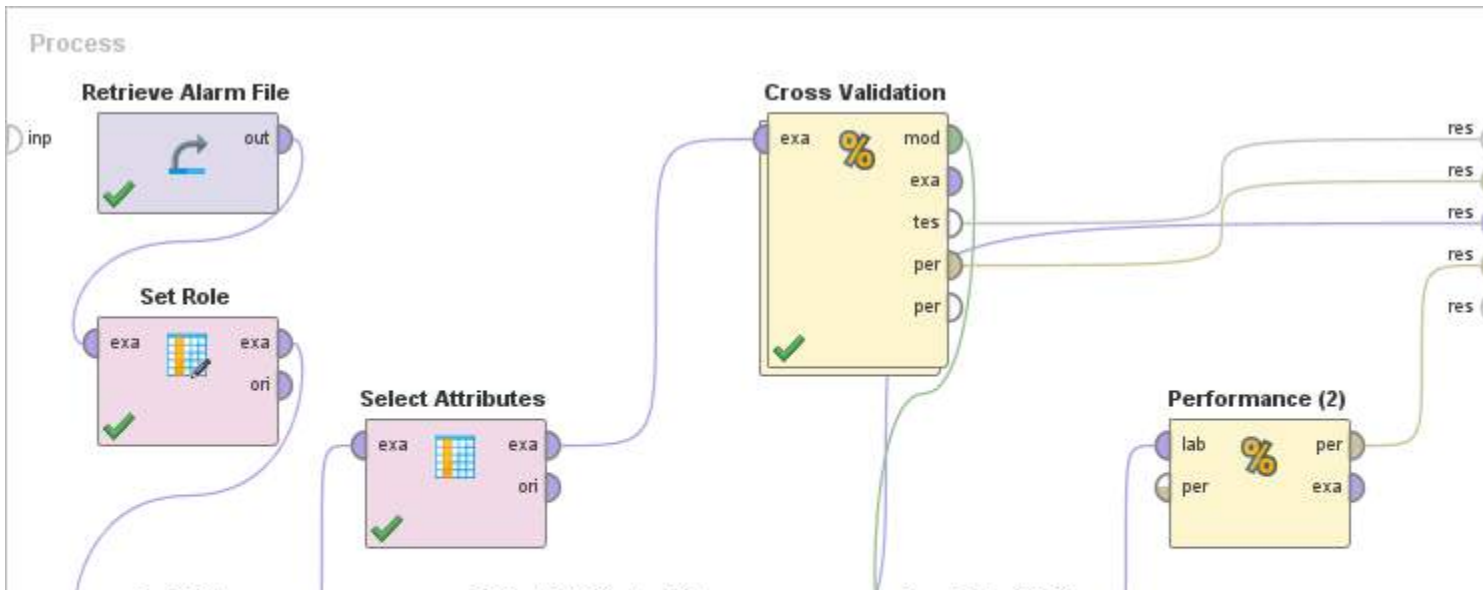
Similarly, there is no difference in the classification error of the model even after retraining the provisional model

The classification error still remains 7.11% - which is quite high and hence, we might want to rethink before considering such a model

classification_error: 7.10% +/- 2.61% (micro average: 7.11%)

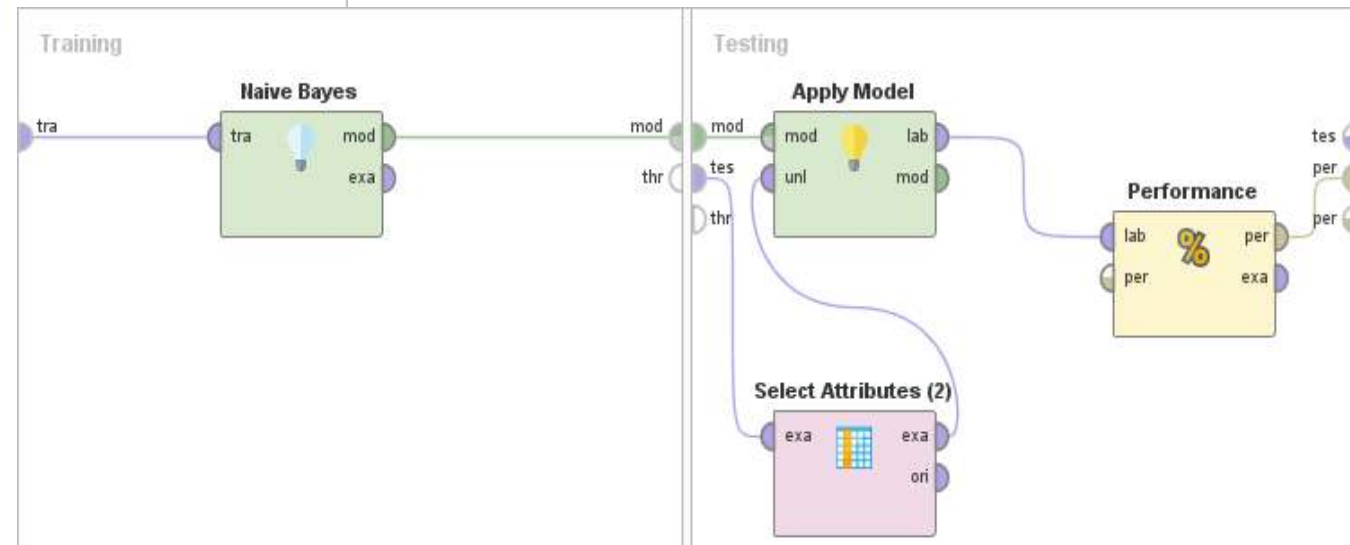
	true false	true true	class precision
pred. false	346	28	92.51%
pred. true	0	20	100.00%
class recall	100.00%	41.67%	

APPLYING THE ADJUSTED MODEL ON VALIDATION SET



We finally apply the adjusted model (in which Timestamp & Violation Type variables are removed) to the validation set (as shown in the process editor)

Let's check the feedback from the validation set and the accuracy of the final data mining model!



FEEDBACK FROM THE VALIDATION SET

Request	Role	Component Access	Request type	Alarm	confidence(false)	confidence(true)	prediction(Alarm)	Match/No Match
User B	Business use	Table 1	Select	true	0.7	0.3	false	No Match
User B	Business use	Table 1	Select	true	0.7	0.3	false	No Match
User B	Business use	Table 1	Select	true	0.7	0.3	false	No Match
User B	Business use	Table 1	Select	true	0.7	0.3	false	No Match
User A	Analyst	Element 2	Select	true	0.7	0.3	false	No Match
User A	Analyst	Element 2	Select	true	0.7	0.3	false	No Match
User A	Analyst	Element 2	Select	true	0.7	0.3	false	No Match
User A	Analyst	Element 3	Append	true	0.8	0.2	false	No Match
User B	Business use	Table 1	Select	true	0.7	0.3	false	No Match
User B	Business use	Table 1	Select	true	0.7	0.3	false	No Match
User B	Business use	Table 1	Select	true	0.7	0.3	false	No Match
User B	Business use	Table 1	Select	true	0.7	0.3	false	No Match
User A	Analyst	Element 2	Select	true	0.7	0.3	false	No Match
User A	Analyst	Element 2	Select	true	0.7	0.3	false	No Match
User A	Analyst	Element 2	Select	true	0.7	0.3	false	No Match
User A	Analyst	Element 3	Append	true	0.8	0.2	false	No Match
User B	Business use	Table 1	Select	true	0.7	0.3	false	No Match
User B	Business use	Table 1	Select	true	0.7	0.3	false	No Match
User B	Business use	Table 1	Select	true	0.7	0.3	false	No Match
User B	Business use	Table 1	Select	true	0.7	0.3	false	No Match
User A	Analyst	Element 2	Select	true	0.7	0.3	false	No Match
User A	Analyst	Element 2	Select	true	0.7	0.3	false	No Match
User A	Analyst	Element 2	Select	true	0.7	0.3	false	No Match
User A	Analyst	Element 3	Append	true	0.8	0.2	false	No Match
User B	Business use	Table 1	Select	true	0.7	0.3	false	No Match
User B	Business use	Table 1	Select	true	0.7	0.3	false	No Match
User B	Business use	Table 1	Select	true	0.7	0.3	false	No Match
User B	Business use	Table 1	Select	true	0.7	0.3	false	No Match

- 28 records show “No Match” in the feedback from the validation set, out of the 406 records
- The security analysts had stated that the alarms were true (as shown below in yellow) but our model predicts that those alarms were false (as shown below in orange)
- Clearly, the 28 predicted values of Alarm (as shown below) are “False Negatives” – some generated by Business User (User B) on Table 1 using Select query and some generated by Analyst (User A) on Element 2 and Element 3
- Let's find out how much accurate our model prediction is!



PERFORMANCE CLASSIFICATION MATRIX

☒ Table View ☐ Plot View

accuracy: 92.90% +/- 2.61% (micro average: 92.89%)

classification_error: 7.10% +/- 2.61% (micro average: 7.11%)

	true false	true true	class precision
pred. false	346	28	92.51%
pred. true	0	20	100.00%
class recall	100.00%	41.67%	

Test set performance classification result shows 92.90% accuracy

Sensitivity or Recall of all positive classes or True Positive Rate (TPR) = 41.67%

☒ Table View ☐ Plot View

Validation set performance classification result shows 93.10% accuracy

Sensitivity or Recall of all positive classes or True Positive Rate (TPR) = 40.43%

accuracy: 93.10%

classification_error: 6.90%

	true false	true true	class precision
pred. false	359	28	92.76%
pred. true	0	19	100.00%
class recall	100.00%	40.43%	

NAÏVE BAYES - GOOD MODEL OR NOT?



The decision tree model gives us an **overall accuracy** of **92.90% +/- 2.61%** on the **test set** and **93.10%** on the **validation set**



Removing Timestamp and Violation Type do not make any difference on the accuracy of the model










Since, the model accuracy is below 95% and the error rate is also high (6.90%), hence, we might want to rethink before considering such a model

Model Accuracy: 90.00% to 93.10%

Not recommended!



LOGISTIC REGRESSION

-  A popular go-to **supervised** classification method used in data mining
-  Uses a predictive analysis algorithm which is based on the concept of probability
-  Used to assign observations to a discrete set of classes (here, True and False)
-  Required polynomial variables to be converted to numerical variables (also called dummy variables)
-  Tool used for building the Logistic Regression classification model: **RapidMiner Studio** 
-  RapidMiner is a very effective data science software platform that unites data prep, machine learning & predictive model deployment

Recommendation in EDA Phase:

....

Seems to give 100.00% accuracy results and seems to be very efficient

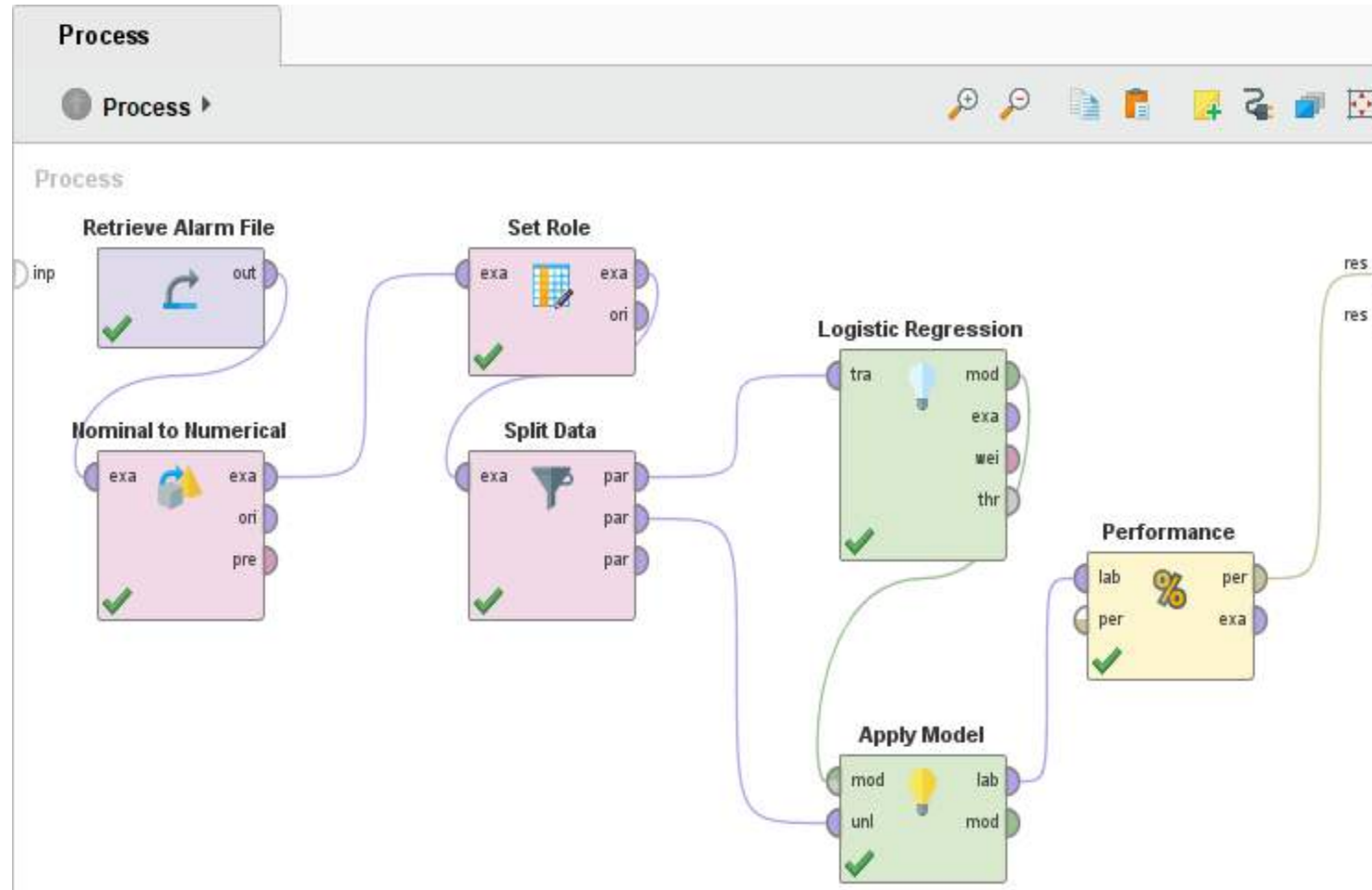
LOGISTIC REGRESSION - USING THE TRAINING AND TEST SET

The first step is to retrieve the data i.e. Alarm file

The second step is to convert polynomial variables into numeric variables by using 'Nominal to Numerical' operator

The 'Nominal to Numerical' operator converts the nominal variables into dummy variables depending on the number of values of the nominal variable

'Set Role' operator is used to tell RapidMiner which is our target variable (Alarm)



Dataset is split into 3 sets: Training, Test & Validation datasets (1/3rd each) using the 'Split Data' operator

Now, we use 'Logistic Regression' operator to build the provisional model

Now, we apply the provisional model to the test set using 'Apply Model' operator

Lastly, 'Performance (Classification)' operator is used to evaluate the model by telling us the accuracy of the model

ACCURACY RESULT



Microsoft Excel
Worksheet

100.00% accuracy, The best model so far!

Here, True Negative (TN) = 348
False Negative (FN) = 0
False Positive (FP) = 0
True Positive (TP) = 46

Hence, **True Positive Rate (TPR)**
can be calculated as: $TP / (TP + FN) = 100.00\%$

True Negative Rate (TNR) can be
calculated as: $TN / (TN + FP) = 100.00\%$

accuracy: 100.00%

Actual (or True) Values

Predicted Values ↓	true false	true true	class precision
pred. false	348	0	100.00%
pred. true	0	46	100.00%
class recall	100.00%	100.00%	

Positive Predictive Value (PPV)
can be calculated as: $TP / (TP + FP) = 100.00\%$

Negative Predictive Value (NPV)
can be calculated as: $TN / (TN + FN) = 100.00\%$

Accuracy can be calculated as:
 $(TP + TN) / (TP + TN + FP + FN) = 100.00\%$

et (Apply Model) × PerformanceVector (Performance) ×

☒ Table View ☐ Plot View

classification_error: 0.00%

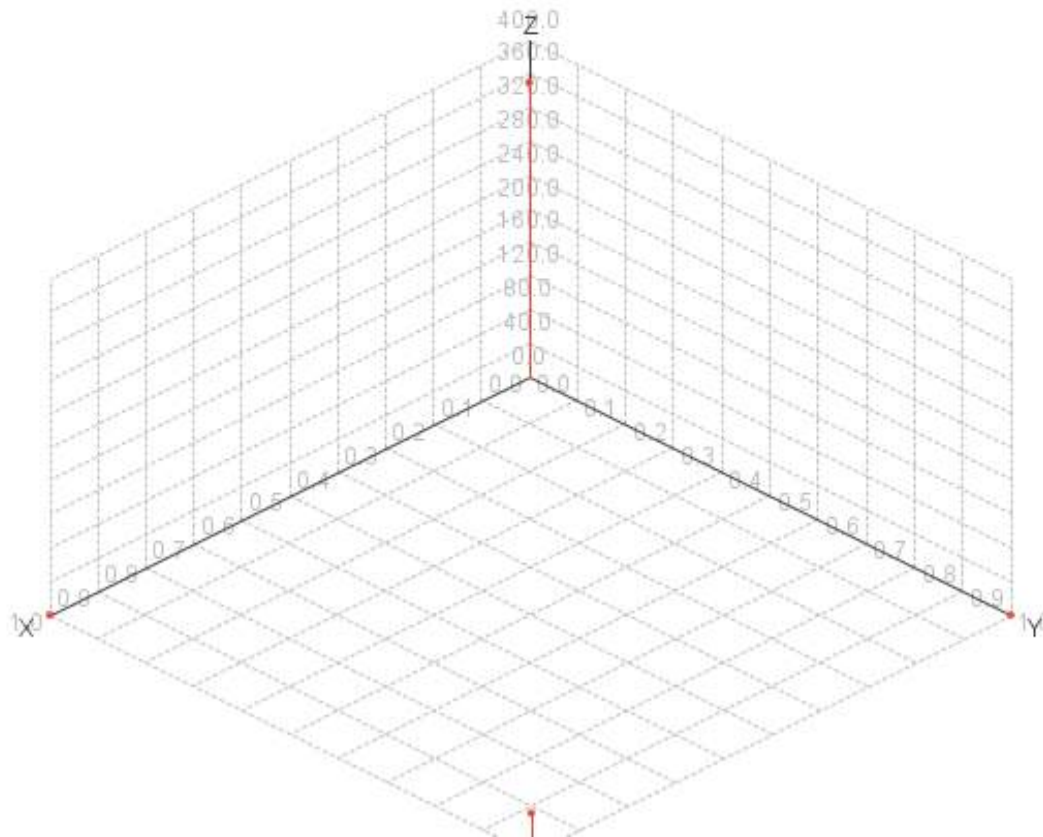
	true false	true true	class precision
pred. false	348	0	100.00%
pred. true	0	46	100.00%
class recall	100.00%	100.00%	

Classification error can be calculated as:
 $(FP + FN) / (TP + TN + FP + FN) = 0.00\%$

ACCURACY RESULT

☐ Table View ☒ Plot View

Confusion Matrix (x: true class, y: pred. class, z: counters)



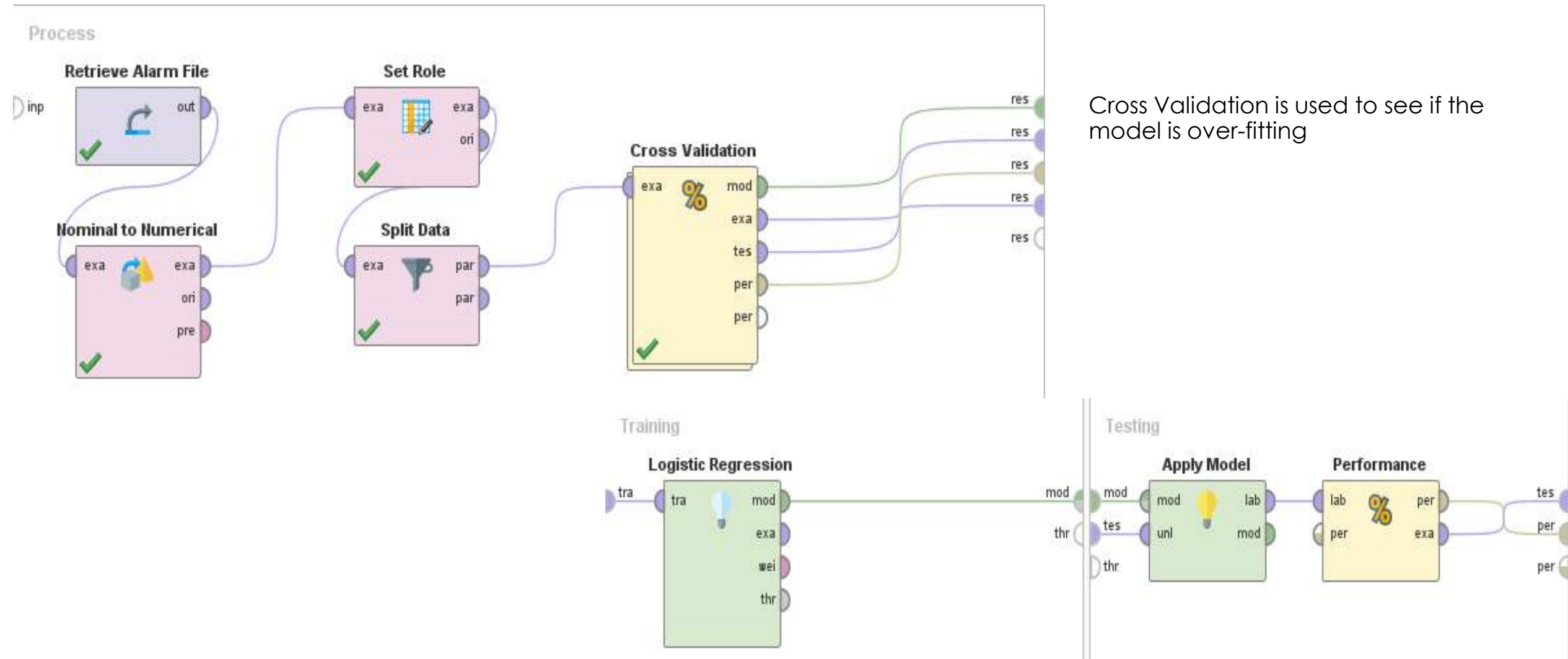
Plot View of Performance (Classification) matrix:

X: Actual Values

Y: Predicted Values

Z: Values (or Counters)

USING CROSS VALIDATION TO CHECK OVER-FITTING



FEEDBACK FROM THE TEST SET

Reques	Reques	Reques	Role_B	Role_A	Role_A	Compo	Compo	Compo	Compo	Compo	Reques	Reques	confide	predict	Alarm	confidence(tru	Match/No Mat
0.0	0.0	1.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	true	false	1.0	No Match

- Only one record shows 'No Match'
- Let's find out how much accurate our model prediction is after using cross validation!



Microsoft Excel
Worksheet

ACCURACY RESULT

☒ Table View ☐ Plot View

accuracy: 99.75% +/- 0.79% (micro average: 99.75%)

	true false	true true	class precision
pred. false	346	0	100.00%
pred. true	1	46	97.87%
class recall	99.71%	100.00%	

Clearly, there is not much difference between the accuracy of the model before and after using cross validation

The accuracy before using cross validation was 100.00% whereas now, it is 99.75% - hence, we cannot explicitly say if the model over fits

☒ Table View ☐ Plot View

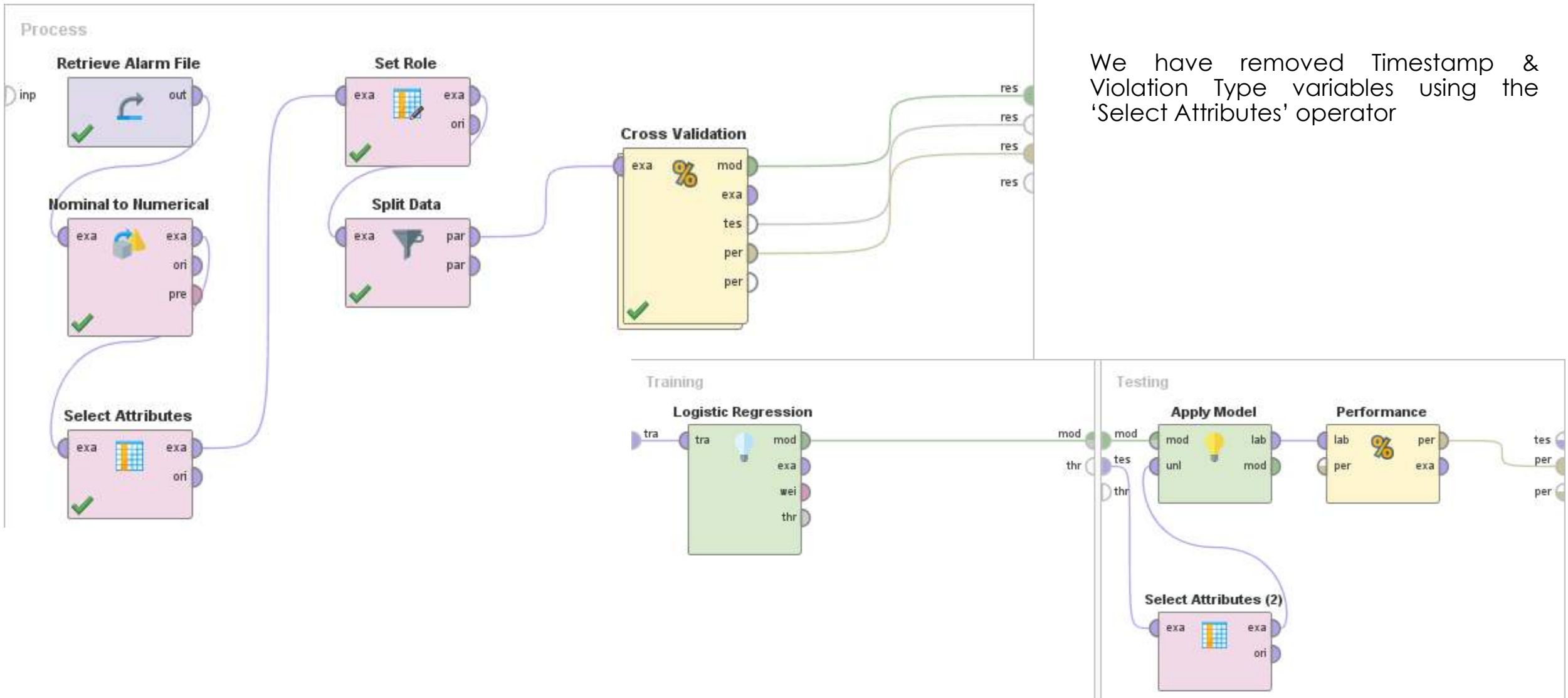
classification_error: 0.25% +/- 0.79% (micro average: 0.25%)

	true false	true true	class precision
pred. false	346	0	100.00%
pred. true	1	46	97.87%
class recall	99.71%	100.00%	

Similarly, there is not much difference in the classification error of the model before and after using cross validation

The classification error before using cross validation was 0.00% whereas now, it is 0.25% - hence, we cannot explicitly say that the model over fits

RE-TRAINING THE MODEL



PERFORMANCE RESULT

☒ Table View ☐ Plot View

accuracy: 99.75% +/- 0.79% (micro average: 99.75%)

	true false	true true	class precision
pred. false	346	0	100.00%
pred. true	1	46	97.87%
class recall	99.71%	100.00%	

There is no difference in the accuracy of the model even after removing Timestamp & Violation Type variables

The accuracy still remains 99.75%

Again, there is no difference in the classification error of the model even after removing Timestamp & Violation Type variables

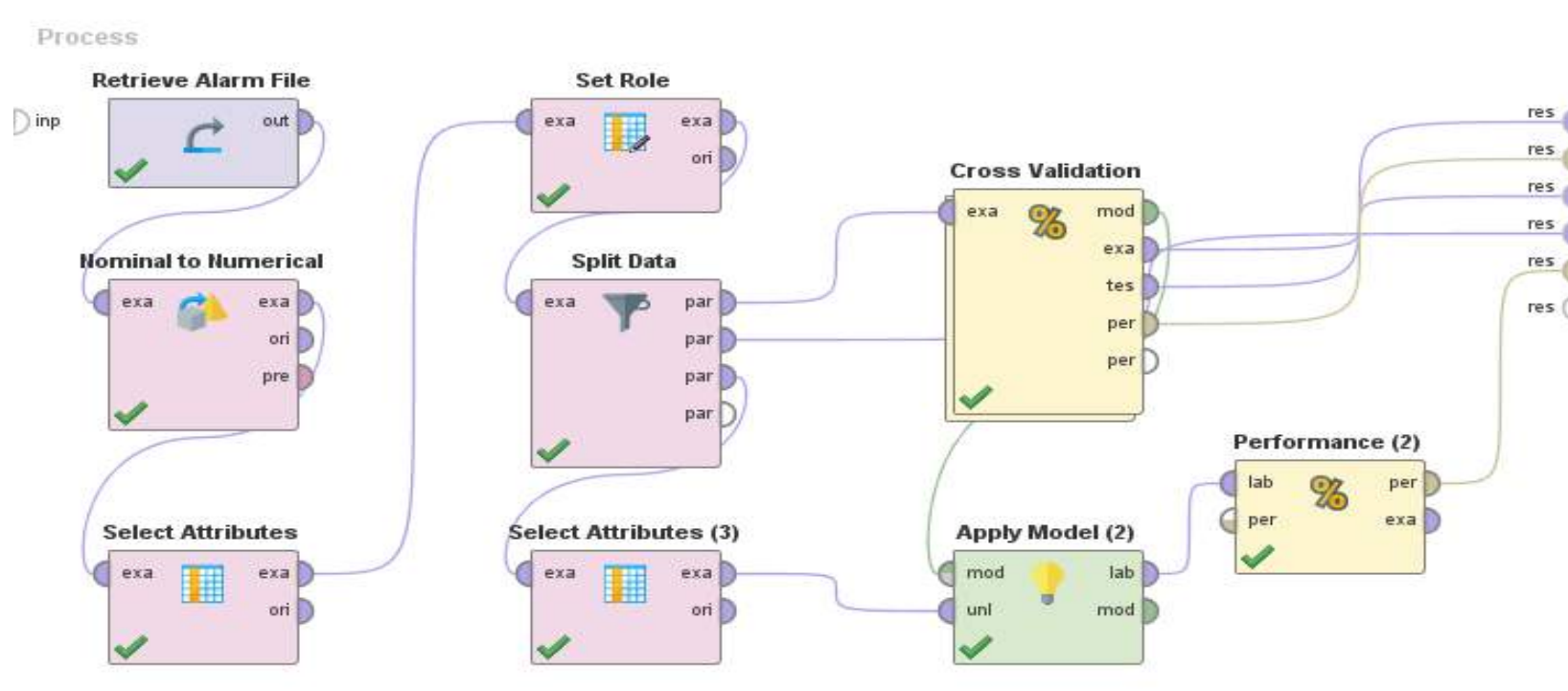
The classification error rate still remains 0.25%

☒ Table View ☐ Plot View

classification_error: 0.25% +/- 0.79% (micro average: 0.25%)

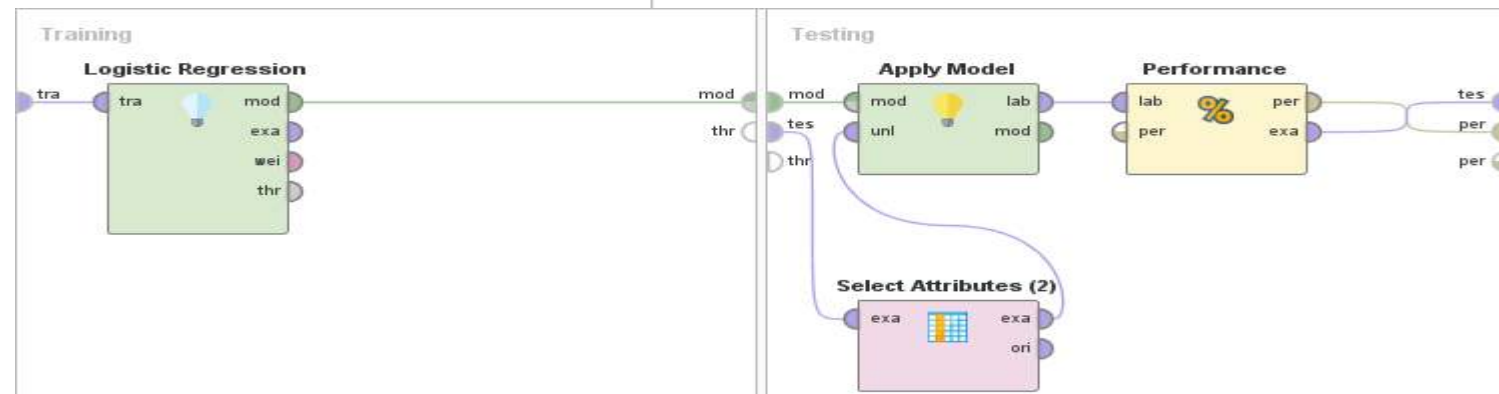
	true false	true true	class precision
pred. false	346	0	100.00%
pred. true	1	46	97.87%
class recall	99.71%	100.00%	

APPLYING THE ADJUSTED MODEL ON VALIDATION SET



We finally apply the adjusted model (in which Timestamp & Violation Type variables are removed) to the validation set (as shown in the process editor)

Let's check the feedback from the validation set and the accuracy of the final data mining model!



PERFORMANCE CLASSIFICATION MATRIX

☒ Table View ☐ Plot View

accuracy: 99.75% +/- 0.79% (micro average: 99.75%)

	true false	true true	class precision
pred. false	346	0	100.00%
pred. true	1	46	97.87%
class recall	99.71%	100.00%	

Test set performance classification result shows 99.75% accuracy

Sensitivity or Recall of all positive classes or True Positive Rate (TPR) = 100.00%

☒ Table View ☐ Plot View

Validation set performance classification result shows 100.00% accuracy

Sensitivity or Recall of all positive classes or True Positive Rate (TPR) = 100.00%

accuracy: 100.00%

	true false	true true	class precision
pred. false	358	0	100.00%
pred. true	0	48	100.00%
class recall	100.00%	100.00%	

LOGISTIC REGRESSION - GOOD MODEL OR NOT?



The decision tree model gives us an **overall accuracy** of **99.75% +/- 0.79%** on the **test set** and **100.00%** on the **validation set**



Removing Timestamp and Violation Type do not make any difference on the accuracy of the model but definitely reduce the execution time of the model building



Since, the model accuracy is above 95% and the error rate is also very low(0.00% to 0.25%), hence, we can definitely think of considering this model as the best one!

Model Accuracy: 98.00% to 99.99%

The best model so far!



SUMMARY AND RECOMMENDATIONS



Based on the percentages calculated, we must consider the following scenarios:

- i) **False Positive Rate (FPR):** FPR is equal to the level of significance ($\alpha = 0.05$). Hence, a large False Positive Rate (or Fall-out rate) can present a poor performance of the Data Loss Detection Engine/System
- ii) **False Negative Rate (FNR):** A large False Negative Rate (FNR) can make CISO or any other organization an easy target to sabotage since the result is erroneously marked as 'False' alarms
- iii) **True Positive Rate (TPR):** A large True Positive Rate (or Sensitivity) can give confidence to CISO or any other organization about the rising number of 'true' alarms and hence, can take appropriate actions at an early stage
- iv) **True Negative Rate (TNR):** A large True Negative Rate (or Specificity) may present a higher efficiency of security analysts at work and hence, CISO can improve workforce management



The False Positive Rate (**FPR**) in all the 3 models is: **0.00%** $\Rightarrow FP/(FP+TN)$ - which is good



The False Negative Rate (**FNR**) in all the 3 models is calculated as $\Rightarrow FN/(FN+TP)$

Decision Tree: 34.04%
(rethink)

Naïve Bayes: 59.57%
(too high)

Logistic Regression: 0.00%
(best to consider)