CAP 939 CA 3

ROLL NUMBER:	**A20**
SECTION:	**1909**
REGISTRATION NUMBER:	**11919709**
NAME:	**GEORGINA ASUAH**
COURSE CODE:	**CAP939**
COURSE TITLE:	**DATA MINING AND DATA WAREHOUSING**

OPERATORS:

Multiply

Optimize Parameters (Grid)

Select Sub Process

Remember

Cross Validation

Decision Tree

Random Forest

Rule Induction

Apply Model

Performance (Classification)

Compare ROCs

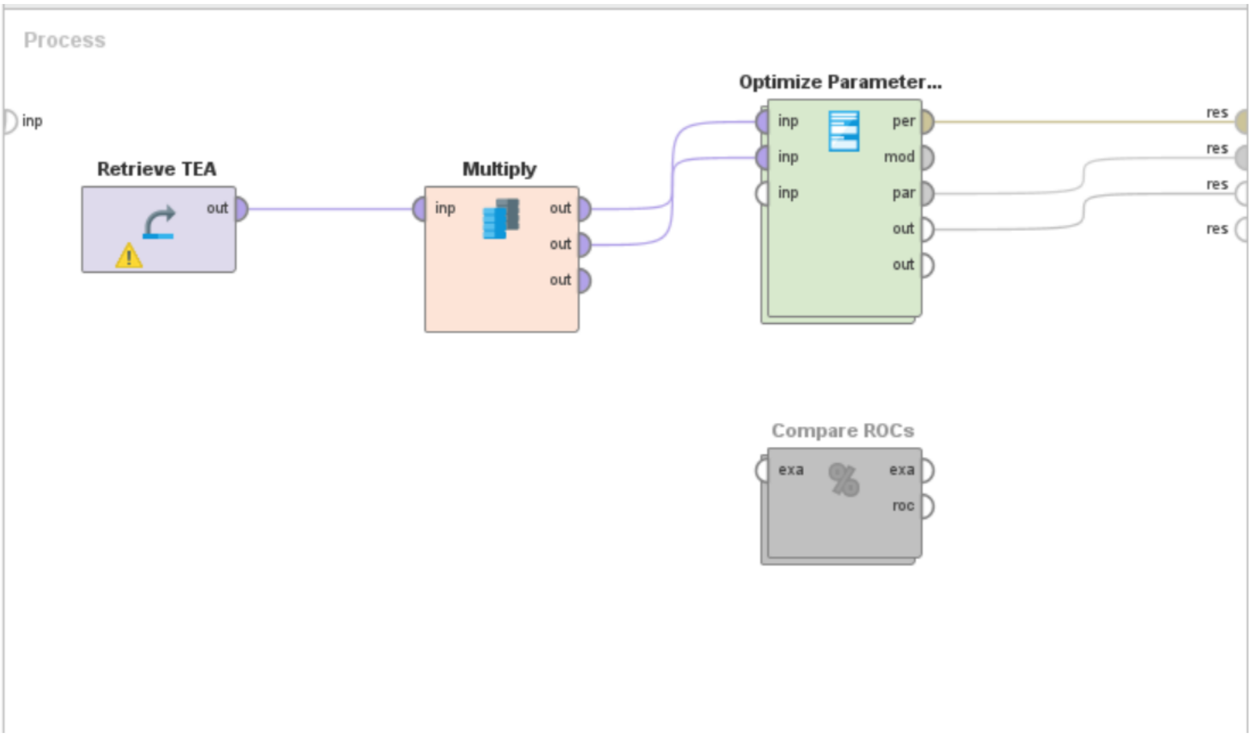
Recall

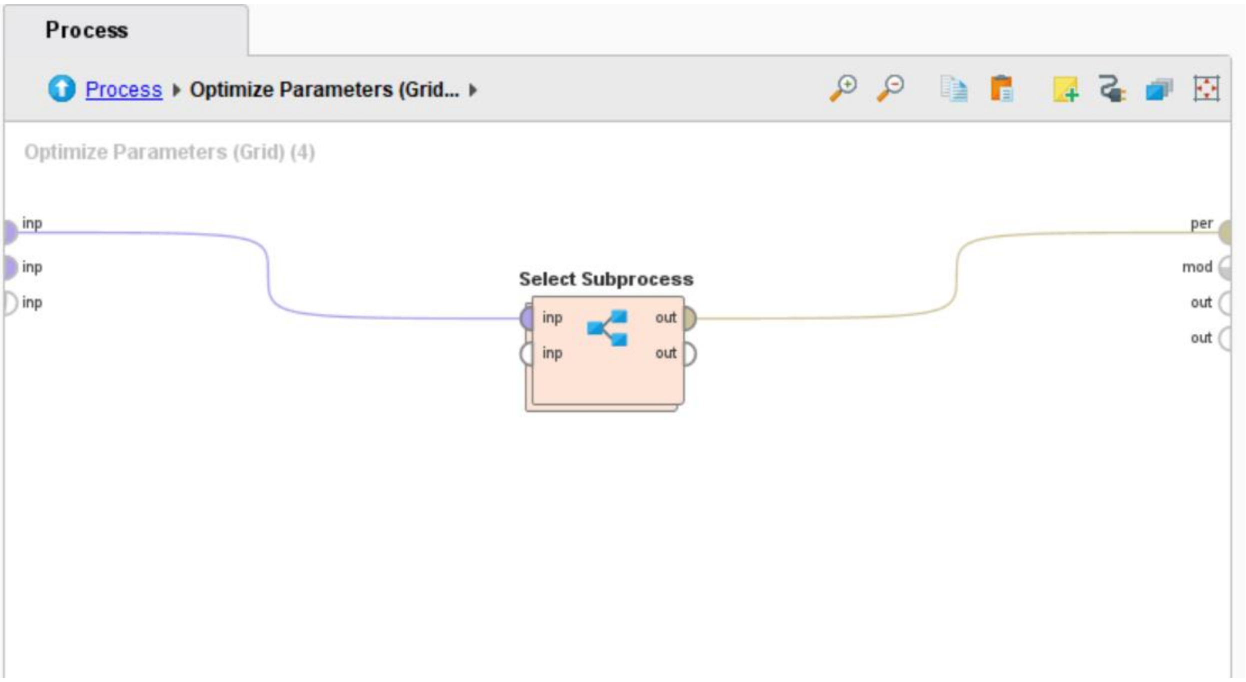
Set Parameters

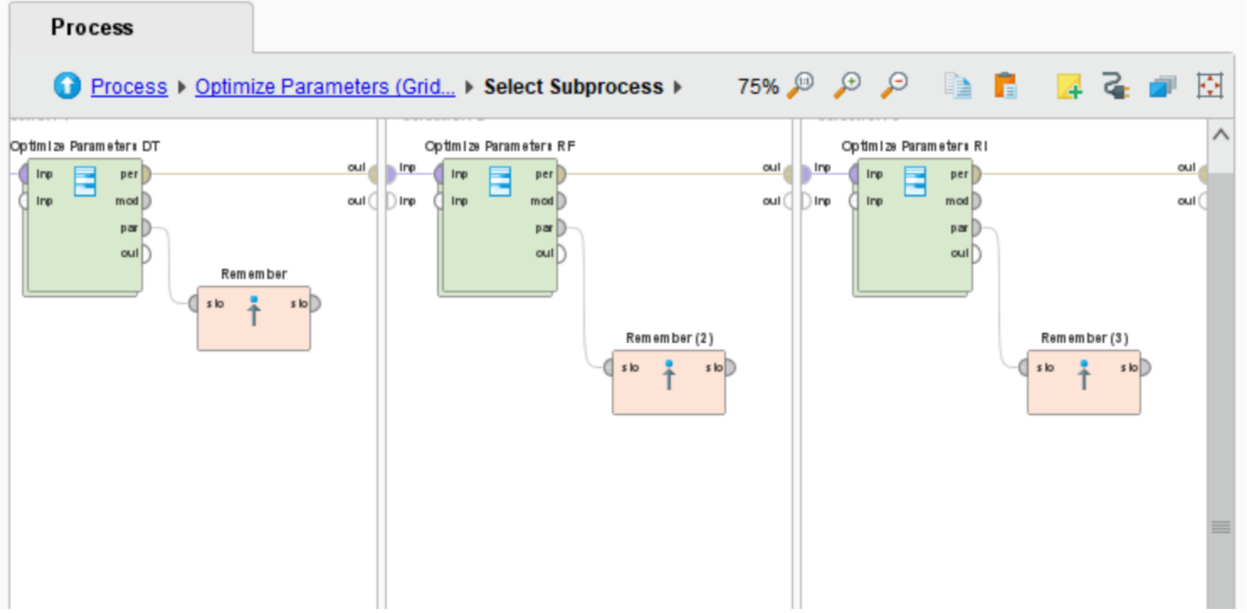
DATA SET:

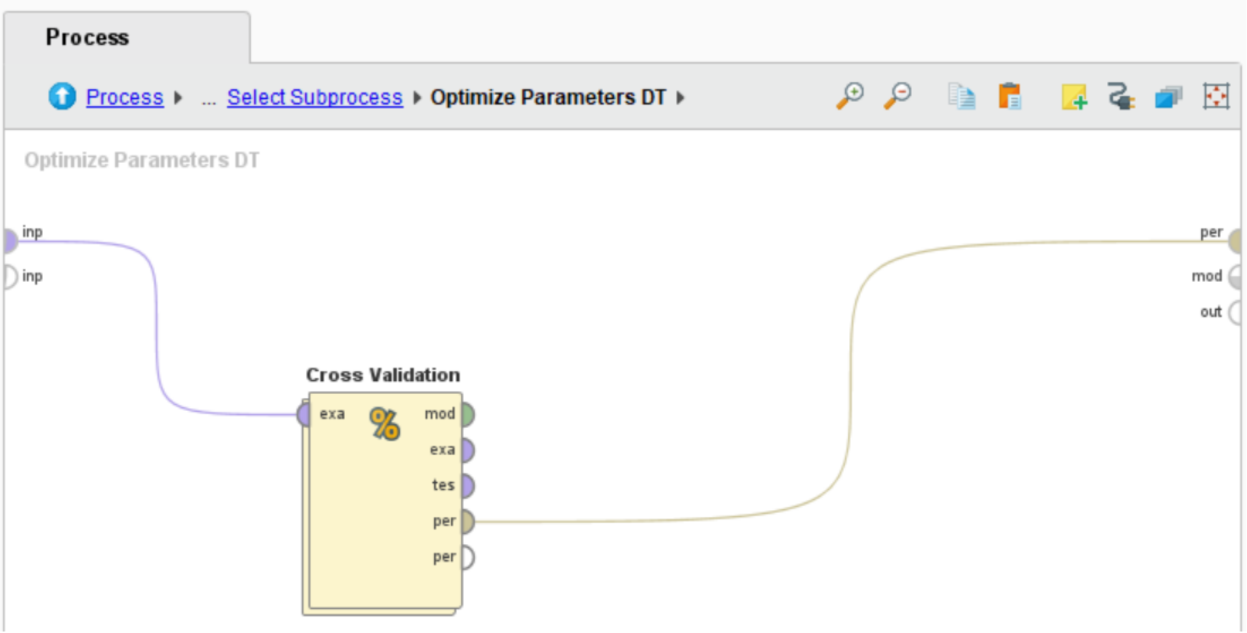
Teaching Assistant Evaluation Data Set (TAE)

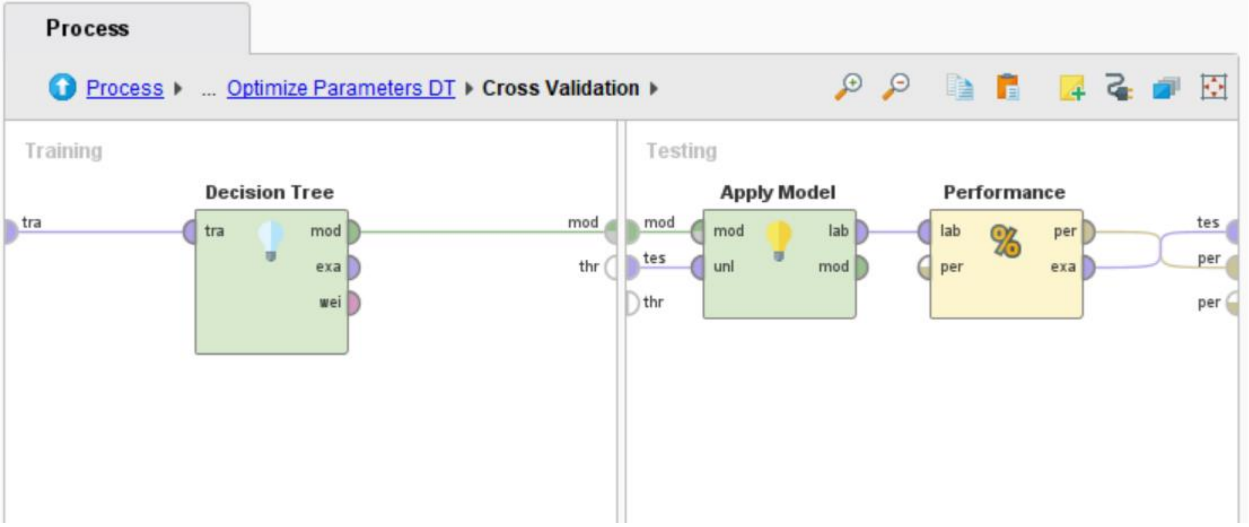
PROCESS:

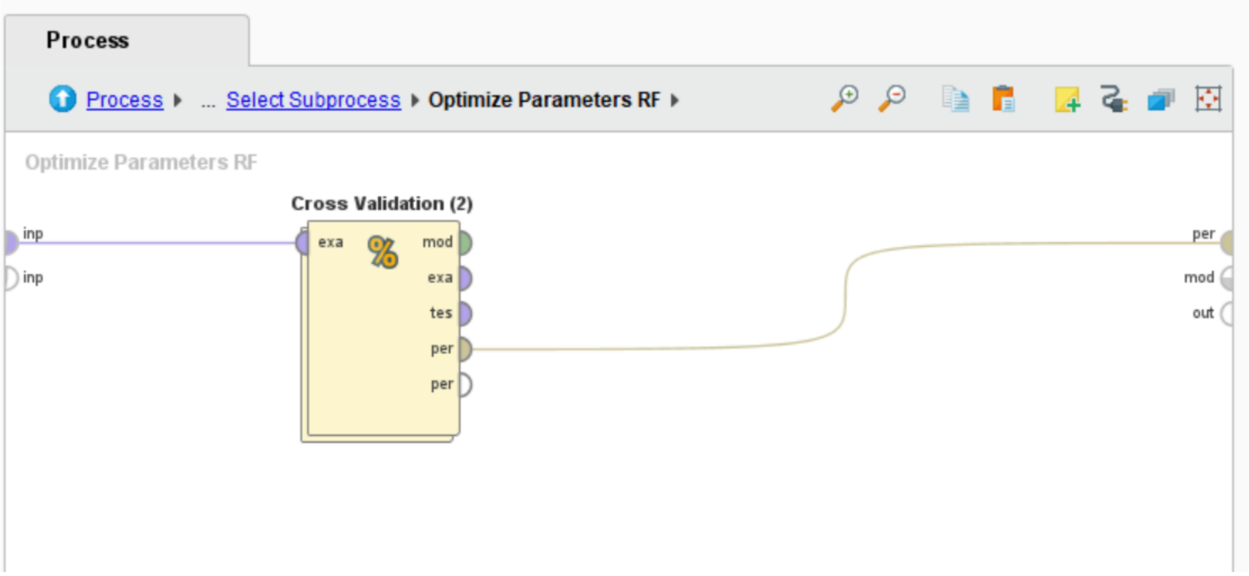

Process
inp
Retrieve TEA
out
Multiply
inp
out
out
out
Optimize Parameter...
inp
inp
inp
per
mod
par
out
out
res
res
res
res
Compare ROCs
exa
exa
roc

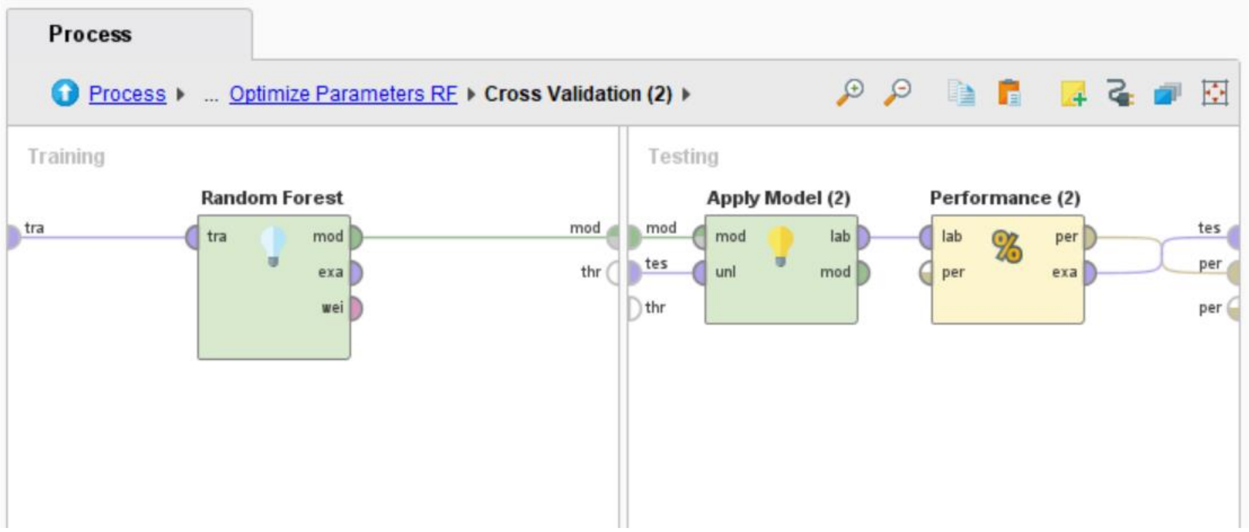

Process
Process
Optimize Parameters (Grid...
Optimize Parameters (Grid) (4)
inp
inp
inp
Select Subprocess
inp
inp
out
out
per
mod
out
out

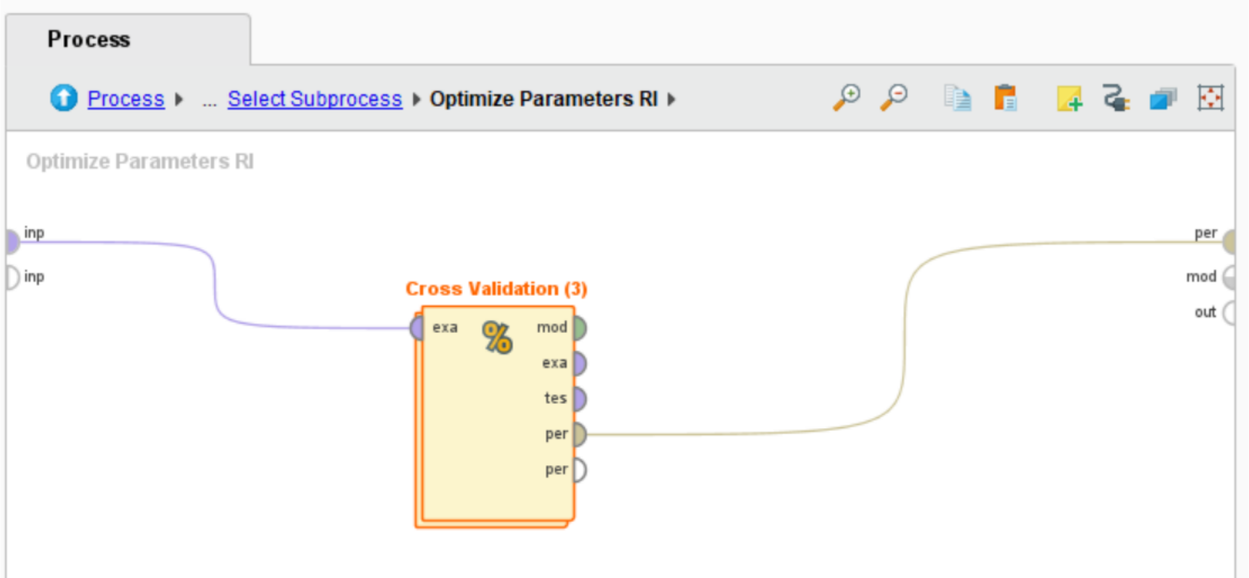

Process
Process
Optimize Parameters (Grid...
Select Subprocess
75%
Optimize Parameters DT
inp
inp
per
mod
par
out
out
out
Remember
sto
sto
Optimize Parameters RF
inp
inp
per
mod
par
out
out
out
Remember (2)
sto
sto
Optimize Parameters RI
inp
inp
per
mod
par
out
out
out
Remember (3)
sto
sto

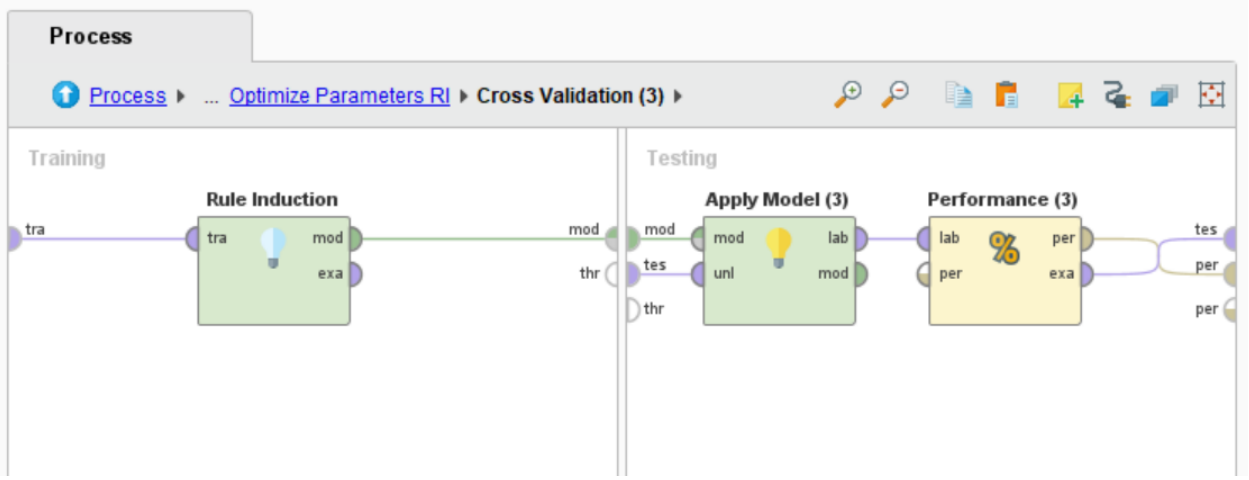

Process
Process
... Select Subprocess
Optimize Parameters DT
Optimize Parameters DT
inp
inp
per
mod
out
Cross Validation
exa
mod
exa
tes
per
per

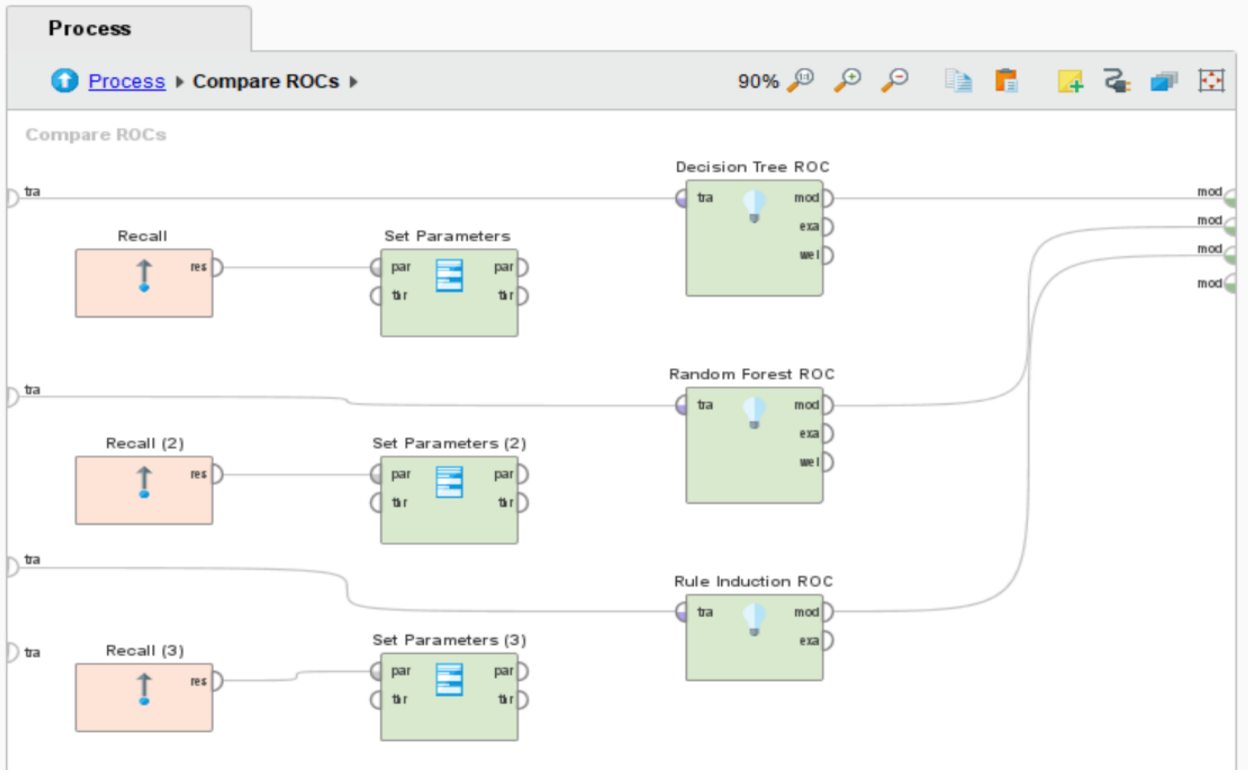

Process
Process
... Optimize Parameters DT
Cross Validation
Training
Decision Tree
tra
tra
mod
exa
wei
mod
thr
Testing
Apply Model
mod
tes
thr
mod
unl
lab
mod
Performance
lab
per
per
exa
tes
per
per


Process
Process ▸ ... Select Subprocess ▸ Optimize Parameters RF ▸
Optimize Parameters RF
Cross Validation (2)
inp
inp
exa
mod
exa
tes
per
per
per
mod
out


Process
Process ▸ ... Optimize Parameters RF ▸ Cross Validation (2) ▸
Training
Random Forest
tra
tra
mod
exa
wei
mod
thr
Testing
Apply Model (2)
Performance (2)
mod
tes
thr
mod
unl
lab
mod
lab
per
per
exa
tes
per
per

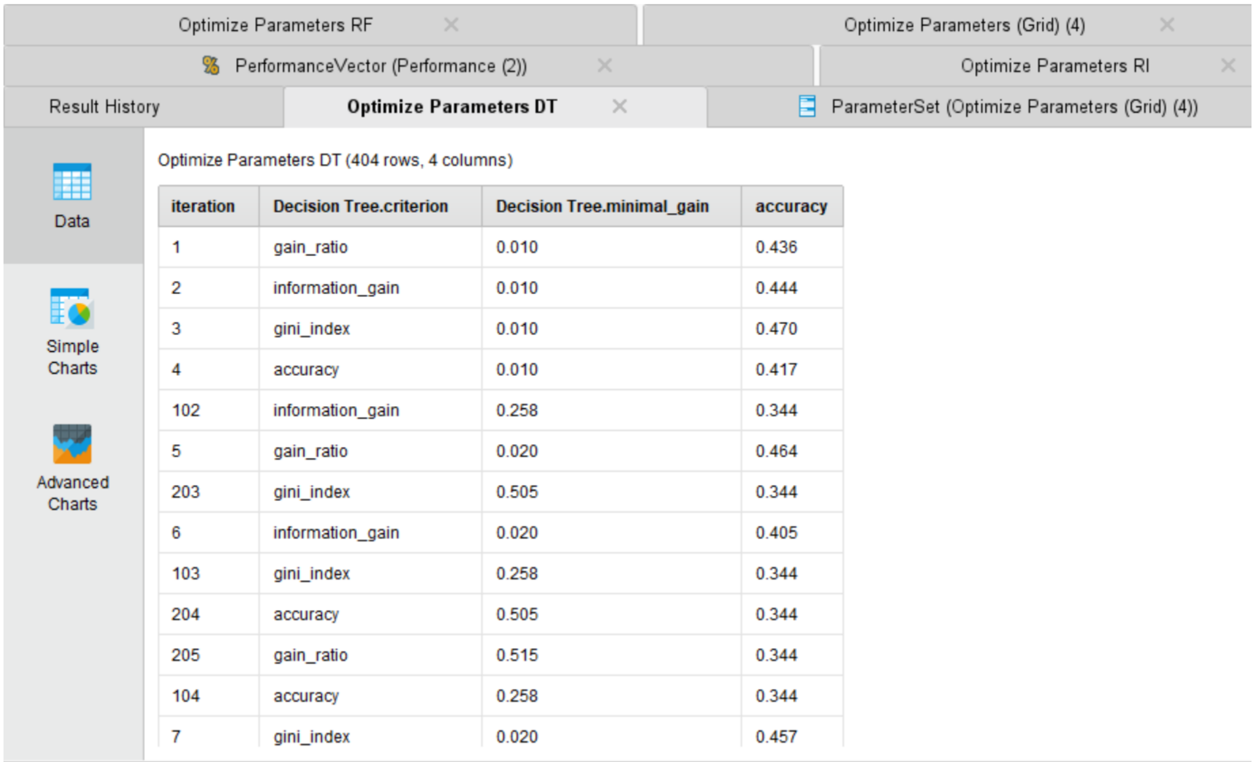

Process
Process ▸ ... Select Subprocess ▸ Optimize Parameters RI ▸
Optimize Parameters RI
inp
inp
Cross Validation (3)
exa
mod
exa
tes
per
per
per
mod
out





The compare ROCs Operator is not applicable to this data set because Compare ROCs only works on binomial attributes but in this case our label attribute (Performance) is non-binomial (Nominal).

STATISTICS:



Optimize Parameters DT (404 rows, 4 columns)

iteration	Decision Tree.criterion	Decision Tree.minimal_gain	accuracy
1	gain_ratio	0.010	0.436
2	information_gain	0.010	0.444
3	gini_index	0.010	0.470
4	accuracy	0.010	0.417
102	information_gain	0.258	0.344
5	gain_ratio	0.020	0.464
203	gini_index	0.505	0.344
6	information_gain	0.020	0.405
103	gini_index	0.258	0.344
204	accuracy	0.505	0.344
205	gain_ratio	0.515	0.344
104	accuracy	0.258	0.344
7	gini_index	0.020	0.457

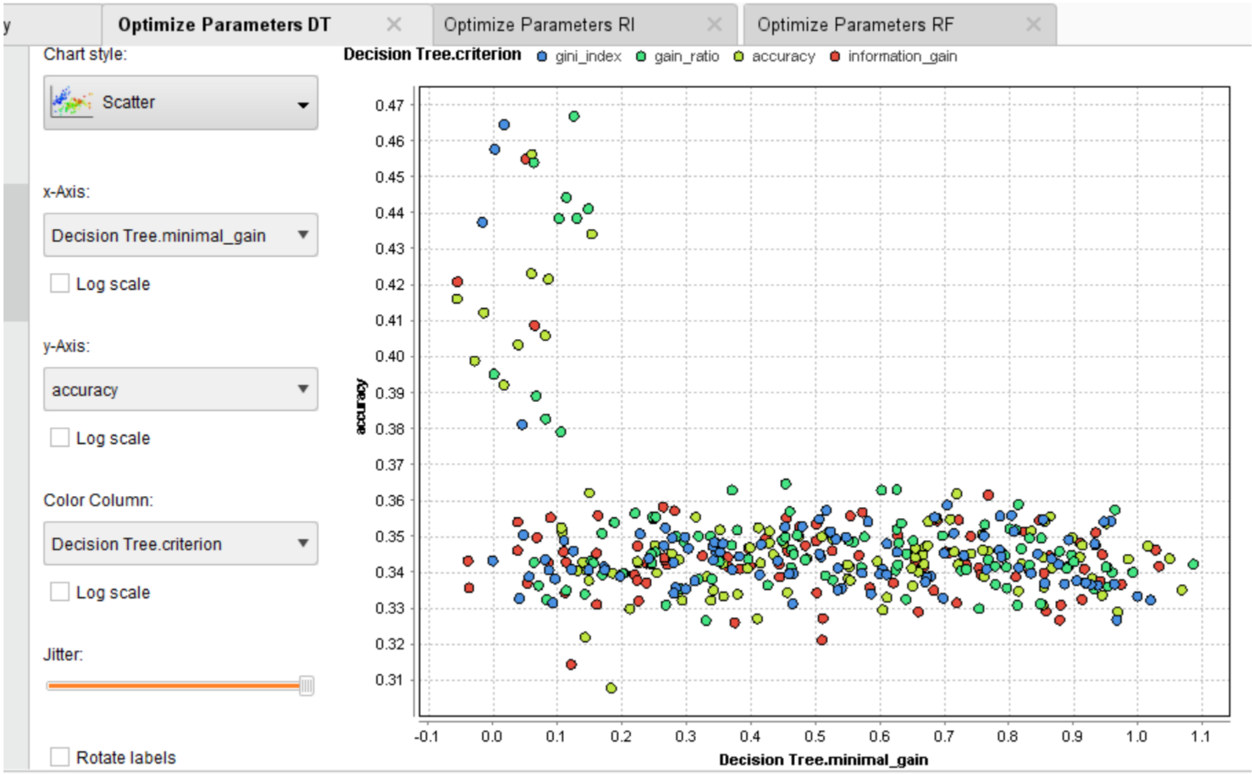


Chart style: Scatter

x-Axis: Decision Tree.minimal_gain

□ Log scale

y-Axis: accuracy

□ Log scale

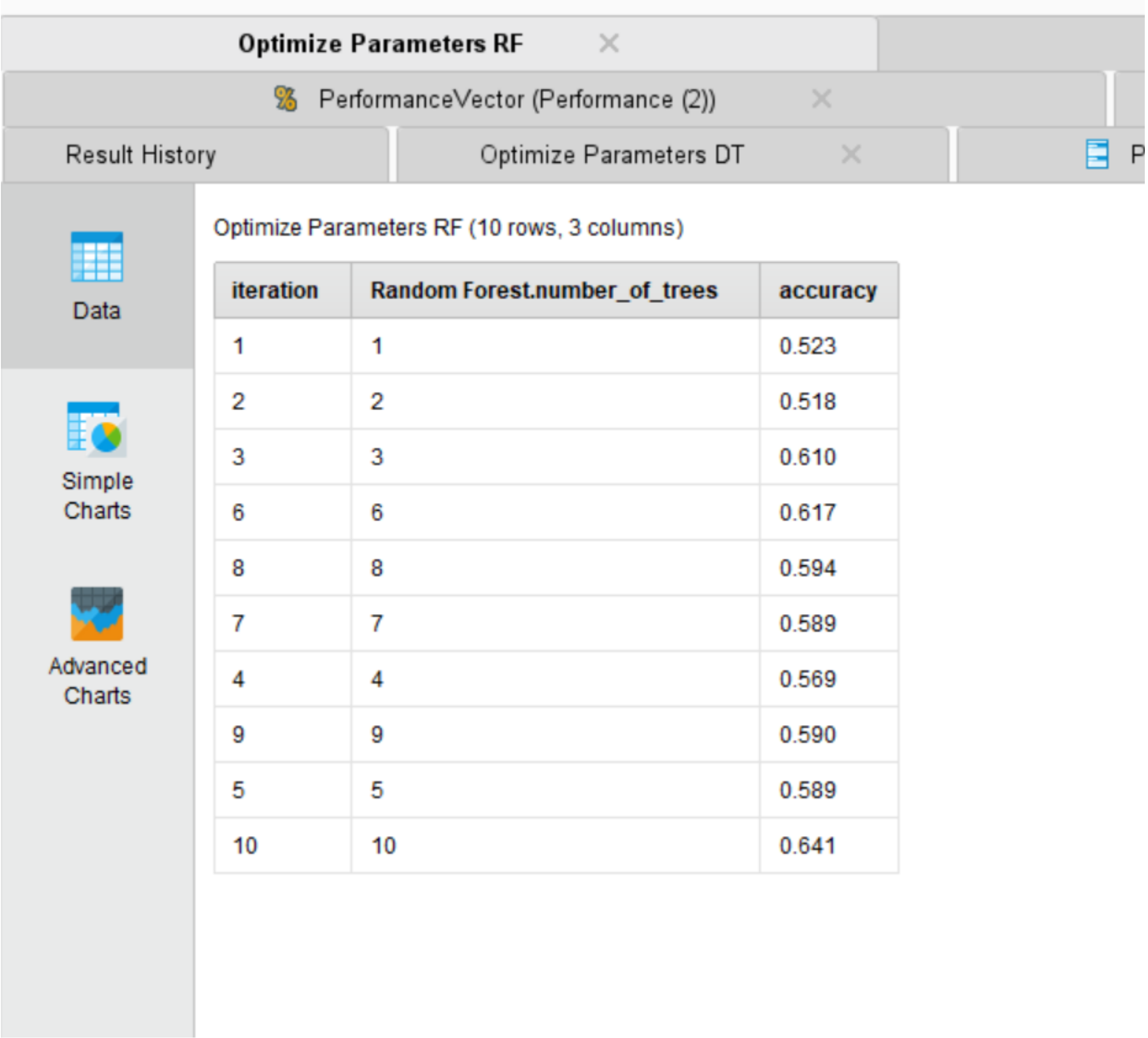
Color Column: Decision Tree.criterion

□ Log scale

Jitter:

□ Rotate labels

The scatter plot shows the results of the optimized parameters Decision Tree with minimal gain against accuracy and colored by criterion. It is visible that most of the criterions are achieving higher accuracies with minimal-gain between -0.1 to 0.2. The accuracies drop and remains within a certain range even when minimal-gain increases some more. The maximum accuracy is 0.47 given by gain-ratio.



Optimize Parameters RF (10 rows, 3 columns)

iteration	Random Forest.number_of_trees	accuracy
1	1	0.523
2	2	0.518
3	3	0.610
6	6	0.617
8	8	0.594
7	7	0.589
4	4	0.569
9	9	0.590
5	5	0.589
10	10	0.641

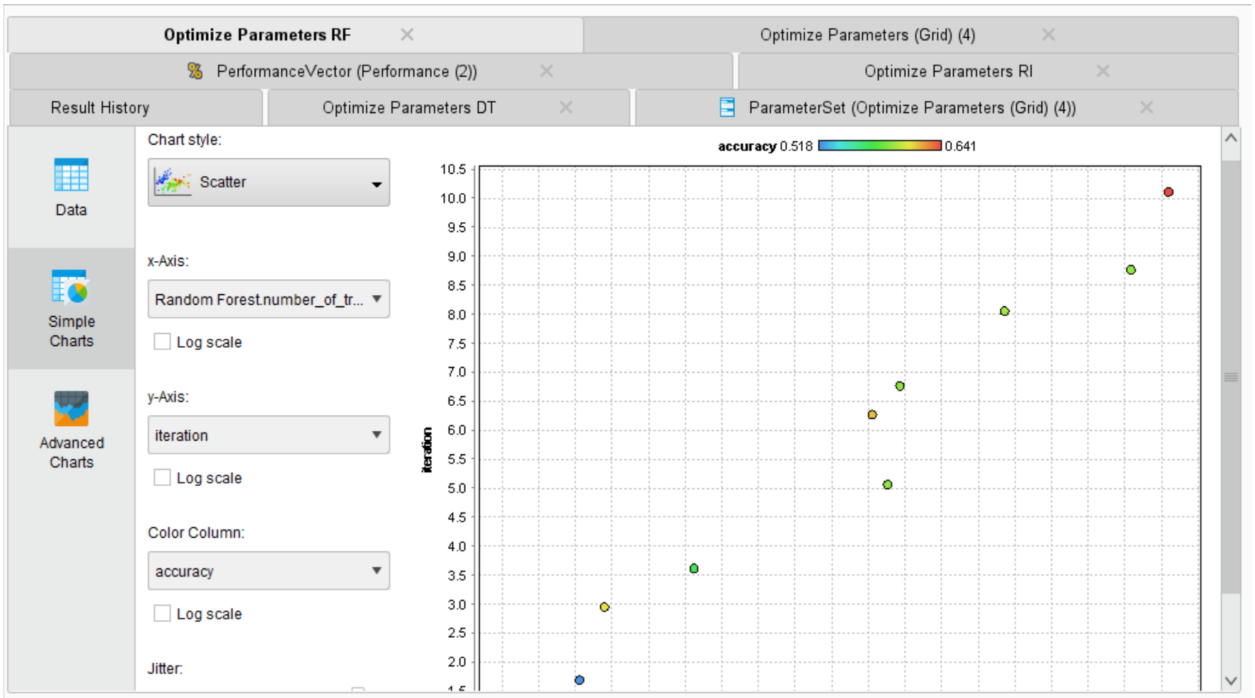


Chart style: Scatter

x-Axis: Random Forest.number_of_tr...

□ Log scale

y-Axis: iteration

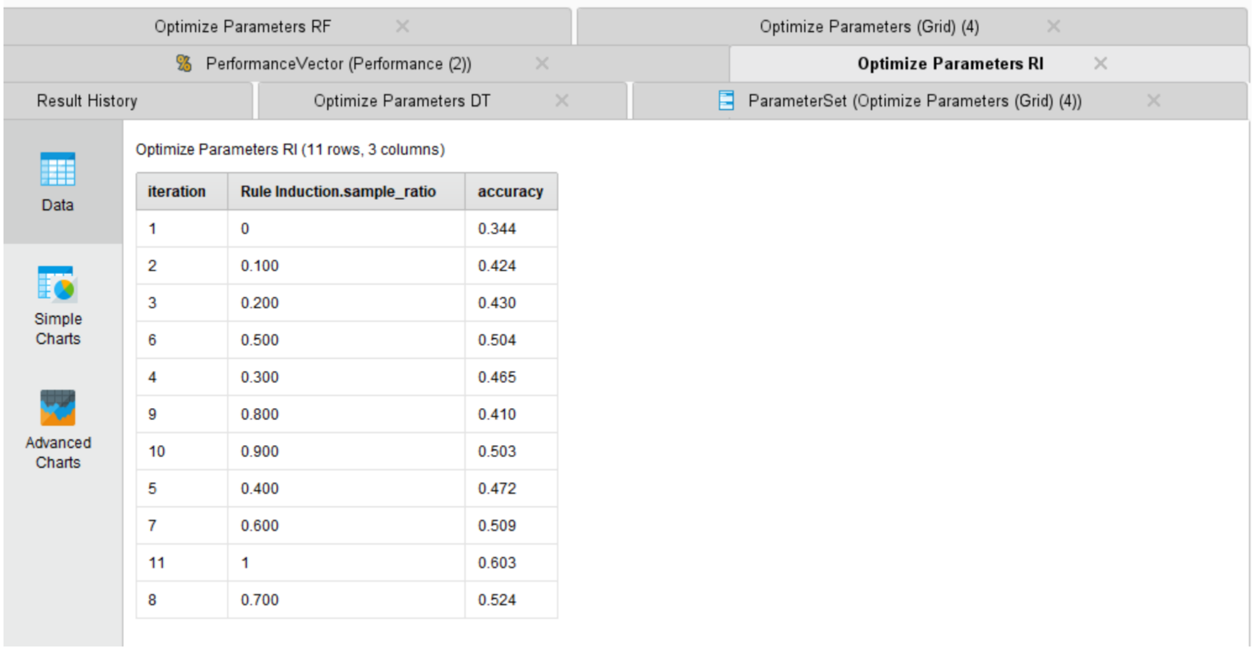
□ Log scale

Color Column: accuracy

□ Log scale

Jitter:

The scatter plot above dipicts the Randon Forrest after the parameters have been optimized. With 10 ietrations on the number of trees and colored by accuracy, the accuracies ranges from 0.51 to 0.64. The number of trees giving the maximum accuracy is 10.



Optimize Parameters RI (11 rows, 3 columns)

iteration	Rule Induction.sample_ratio	accuracy
1	0	0.344
2	0.100	0.424
3	0.200	0.430
6	0.500	0.504
4	0.300	0.465
9	0.800	0.410
10	0.900	0.503
5	0.400	0.472
7	0.600	0.509
11	1	0.603
8	0.700	0.524

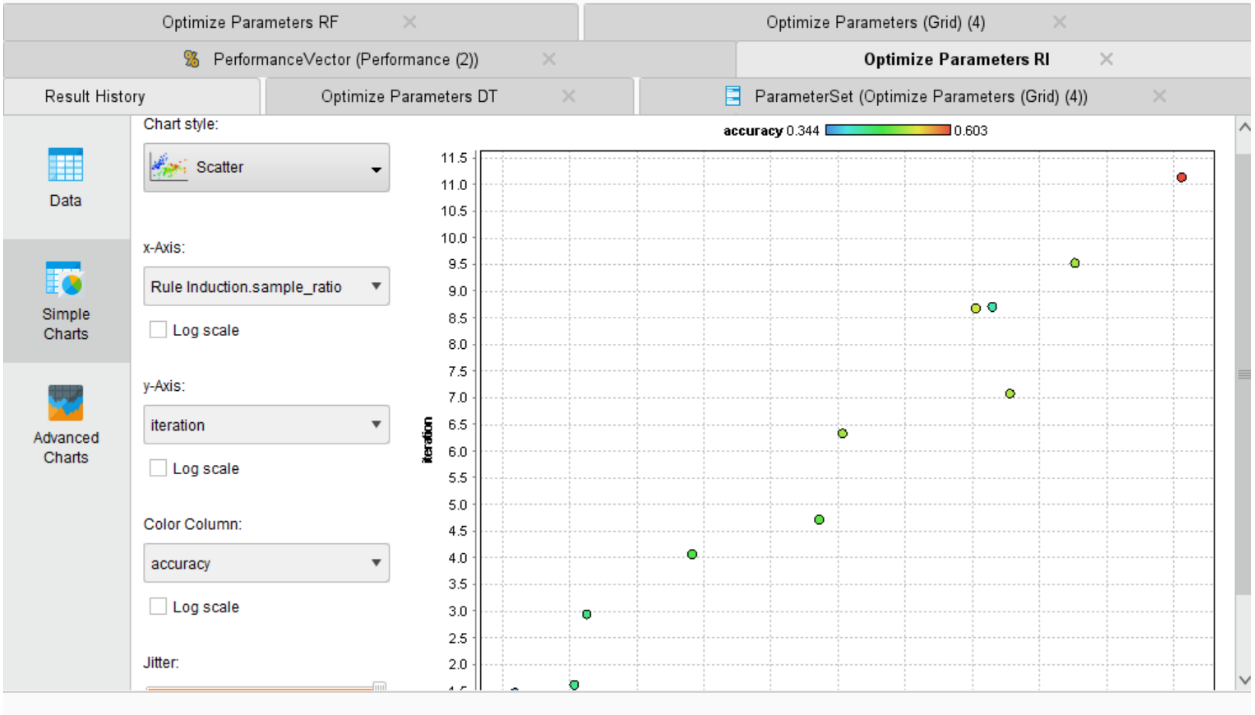


Chart style: Scatter

x-Axis: Rule Induction.sample_ratio

☐ Log scale

y-Axis: iteration

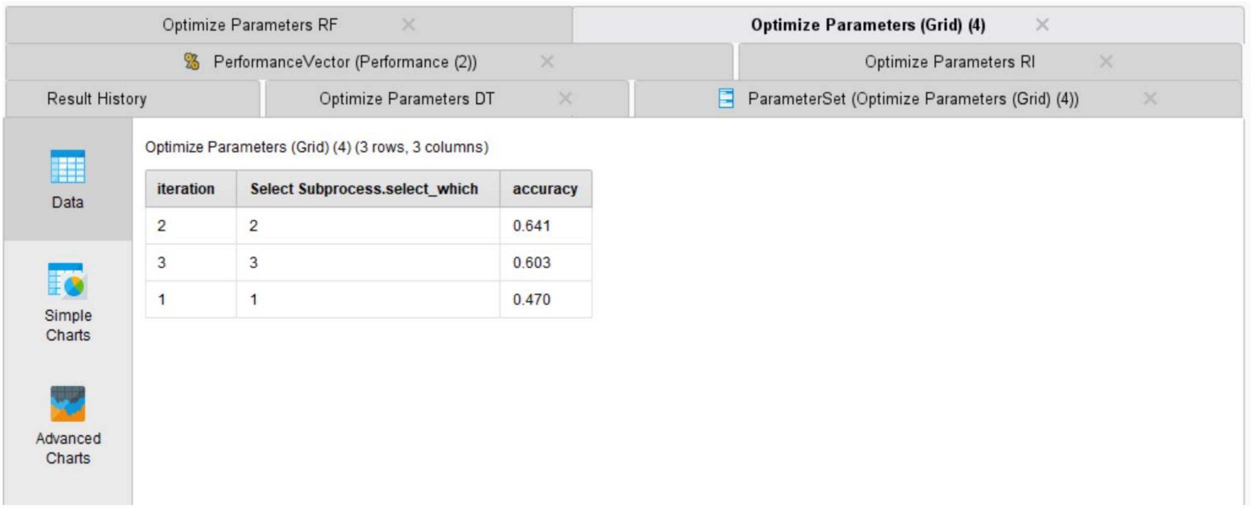
☐ Log scale

Color Column: accuracy

☐ Log scale

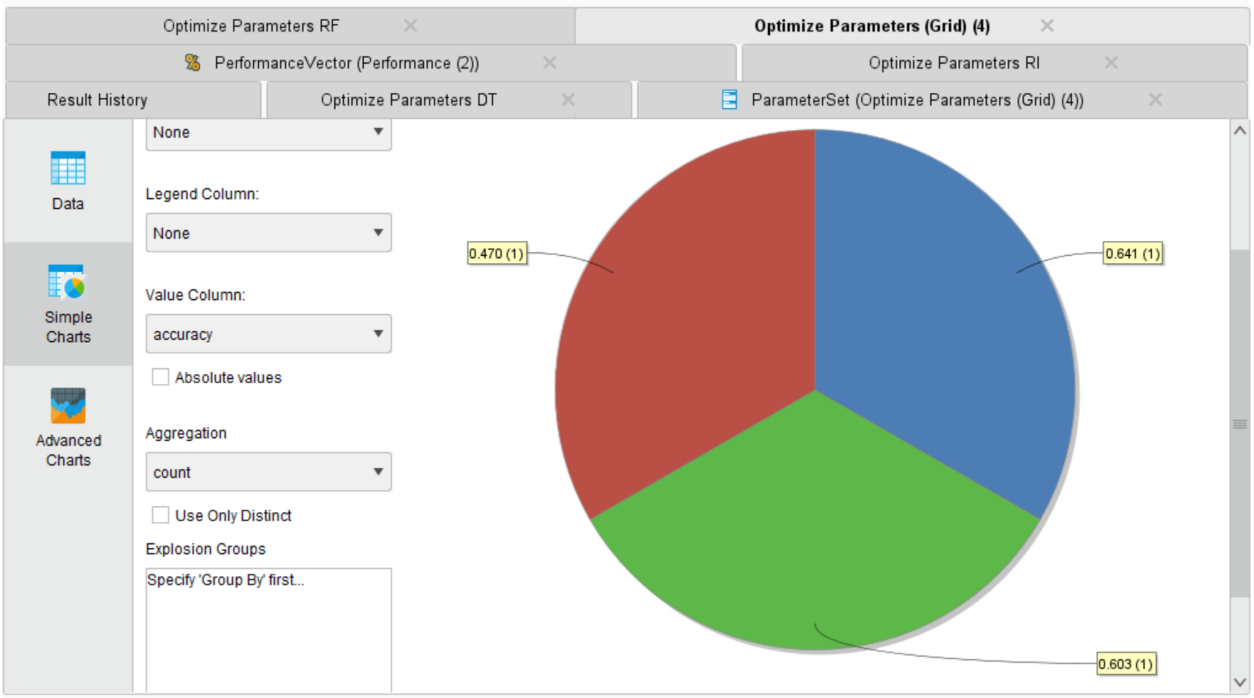
Jitter:

This scatter plot shows the Rule Induction after the parameters have been optimized. Sample-Ratio is plated against iteration and colored by accuracy. The accuracies range from 0.34 to 0.60, for 11 iterations. Sample -Ratio of 1 is giving the maximum accuracy.



Optimize Parameters (Grid) (4) (3 rows, 3 columns)

iteration	Select Subprocess.select_which	accuracy
2	2	0.641
3	3	0.603
1	1	0.470



None

Legend Column: None

Value Column: accuracy

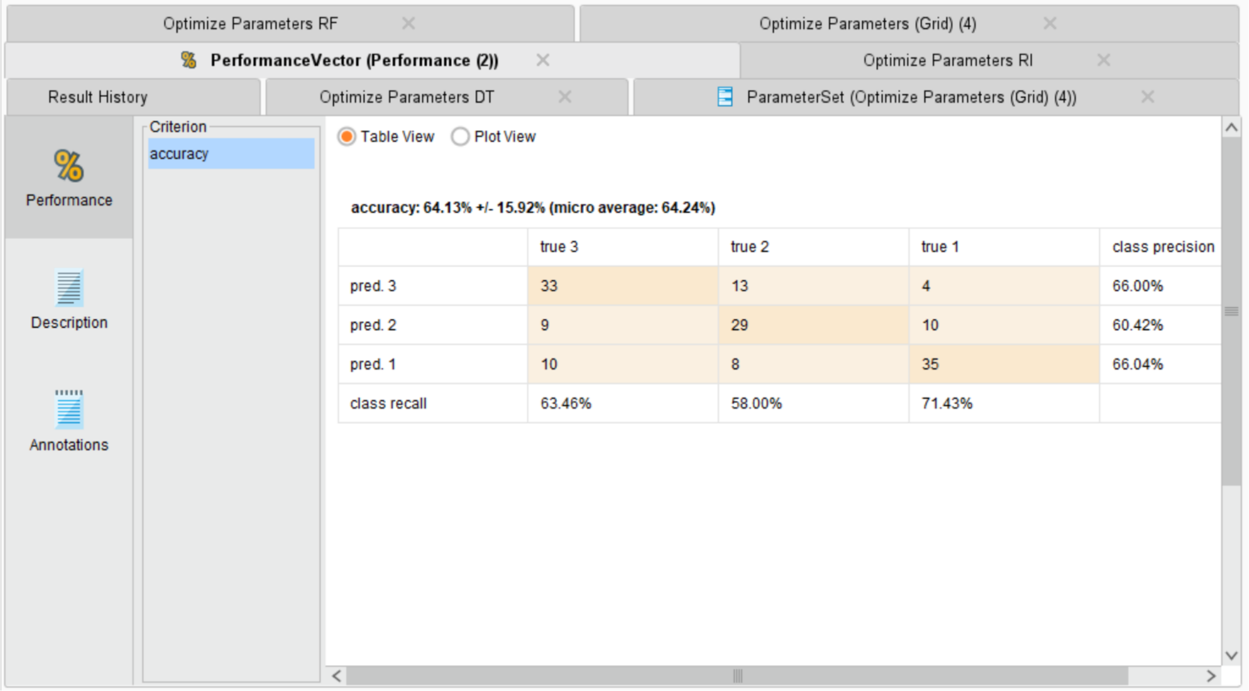
☐ Absolute values

Aggregation: count

☐ Use Only Distinct

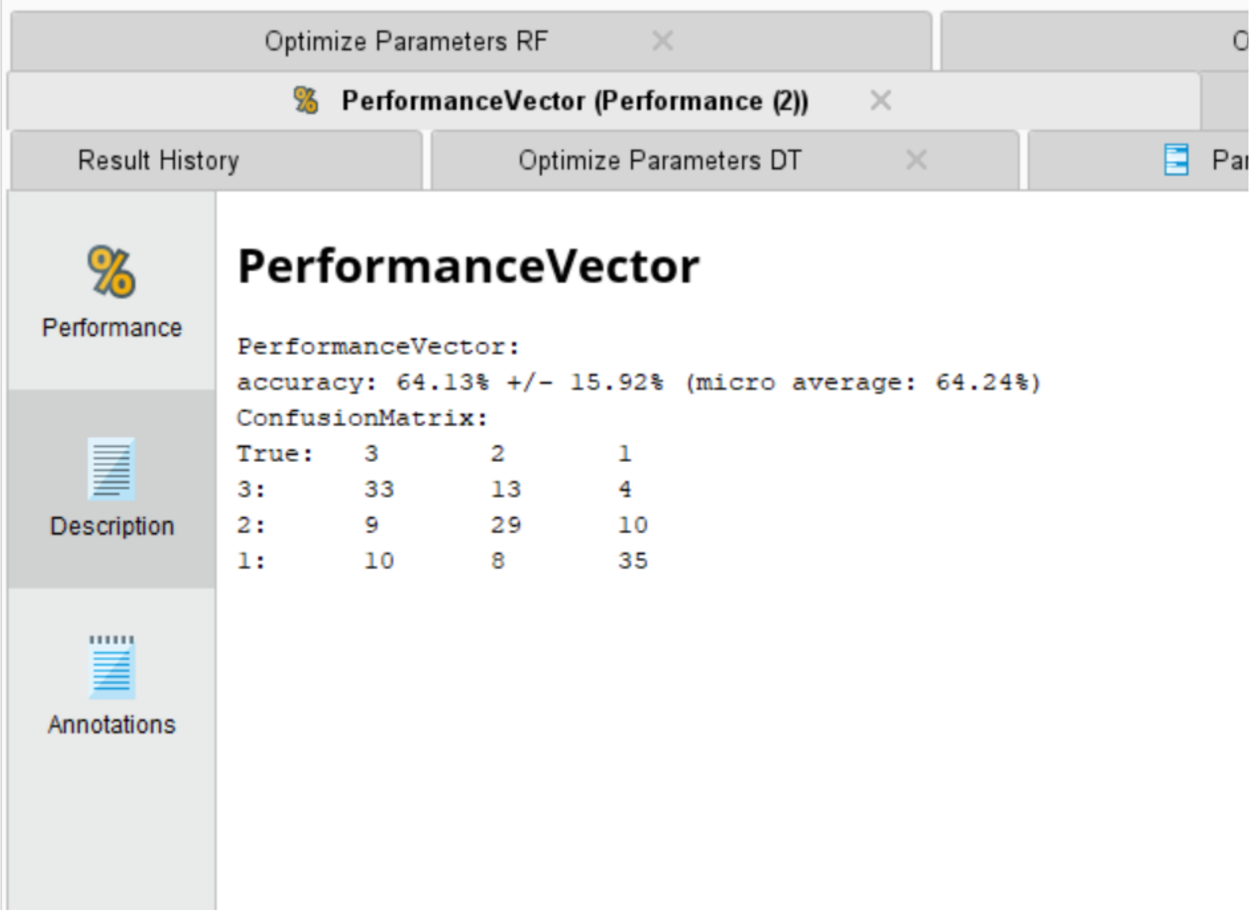
Explosion Groups: Specify 'Group By' first...

The pie chart shows the maximum accuracies from each optimized model. With Random Forest giving the highest accuracy as 0.641 followed by Rule Induction with accuracy of 0.603 and Decision Tree being last with accuracy of 0.470.



accuracy: 64.13% +/- 15.92% (micro average: 64.24%)

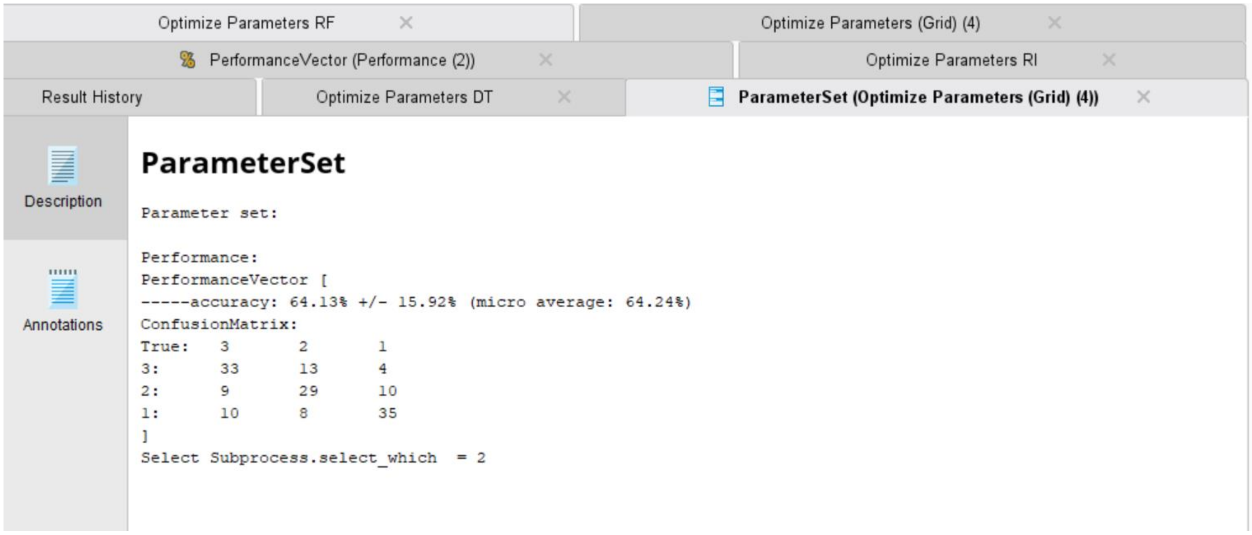
	true 3	true 2	true 1	class precision
pred. 3	33	13	4	66.00%
pred. 2	9	29	10	60.42%
pred. 1	10	8	35	66.04%
class recall	63.46%	58.00%	71.43%	



PerformanceVector

```
PerformanceVector:
accuracy: 64.13% +/- 15.92% (micro average: 64.24%)
ConfusionMatrix:
True:   3       2       1
3:      33      13      4
2:      9       29      10
1:      10      8       35
```

The above is the performance vector showing the accuracy and confusion Matrix of the best performing model (Random Forest).


Optimize Parameters RF
Optimize Parameters (Grid) (4)
PerformanceVector (Performance (2))
Optimize Parameters RI
Result History
Optimize Parameters DT
ParameterSet (Optimize Parameters (Grid) (4))
Description
Annotations

ParameterSet

```
Parameter set:

Performance:
PerformanceVector [
-----accuracy: 64.13% +/- 15.92% (micro average: 64.24%)
ConfusionMatrix:
True:   3       2       1
3:      33      13      4
2:      9       29      10
1:      10      8       35
]
Select Subprocess.select_which  = 2
```

The parameter set shows that it is appropriate to select subprocess 2 which is Random Forest with accuracy of 64.13%, which proves to be performing better than the other classifiers on this data set.