

# Team 3 #Green Phase 2

Srushtiben Kankotiya



# Phase 2 Target



Phase 2 of project includes three essential stages.

- ➡ Training the data
- ➡ Testing the data
- ➡ Validating the data



The goal in this part of project is to choose a model that can classify records as Attrited or Existing with the highest accuracy and confidence possible

# Phase 2 Target



Apply the models chosen in Phase 1 to the test data set for confirming that the **accuracy** is at least 90% with **precision** of at least 90% and **recall** of at least 90%.

# Models Used



We suggested the below models in phase 1:

- ❖ Deep Learning (92%)
- ❖ Logistic Regression / Generalised Linear Model (90%)
- ❖ Naive Bayes (89%)



We have worked with Logistic Regression, Generalised Linear Model and Deep Learning.

# Deep Learning Model

- ❖ Deep learning is a Machine learning technique that teaches computers to do what comes naturally to humans i.e. learn by example. A computer model learns to perform classification tasks directly from images, text or sound. It is well explained through neural networks which makes it more understandable as a large neural network.
- ❖ Neural Networks and Deep Learning currently provide the best solutions to many problems in image recognition, speech recognition and natural language processing.

# Generalized Linear and Logistic Regression Models

- ❖ Generalized Linear Model (GLM) - An advanced statistical modelling technique, an umbrella term that encompasses many other models and, which allows the response variable  $Y$  (dependent) to have an error distribution other than a normal distribution i.e. it allows residuals to have other distributions from the exponential family of distributions.
- ❖ Logistic Regression Model - It is a predictive analysis used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables. It is the appropriate regression analysis to conduct when the dependent variable is dichotomous (Binary 1,0)

## What is Generalized linear model?

- Generalized Linear Model (GLiM, or GLM) is an advanced statistical modelling technique formulated by John Nelder and Robert Wedderburn in 1972.
- It is an umbrella term that encompasses many other models, which allows the response variable  $y$  to have an error distribution other than a normal distribution. The models include Linear Regression, Logistic Regression, and Poisson Regression.

# Why do we use the generalized linear model?

GLM models allow us to build a linear relationship between the response and predictors, even though their underlying relationship is not linear. The GLM operator is used to predict the Future customer attribute of the Deals sample data set



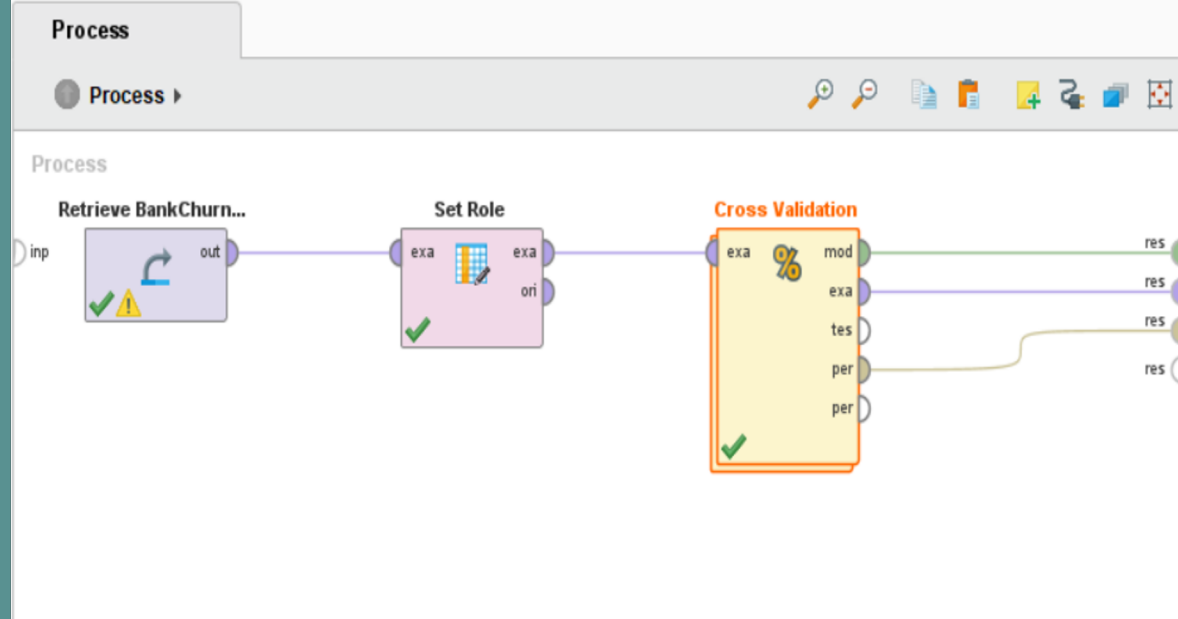
# Generalized linear model: Selected attributes

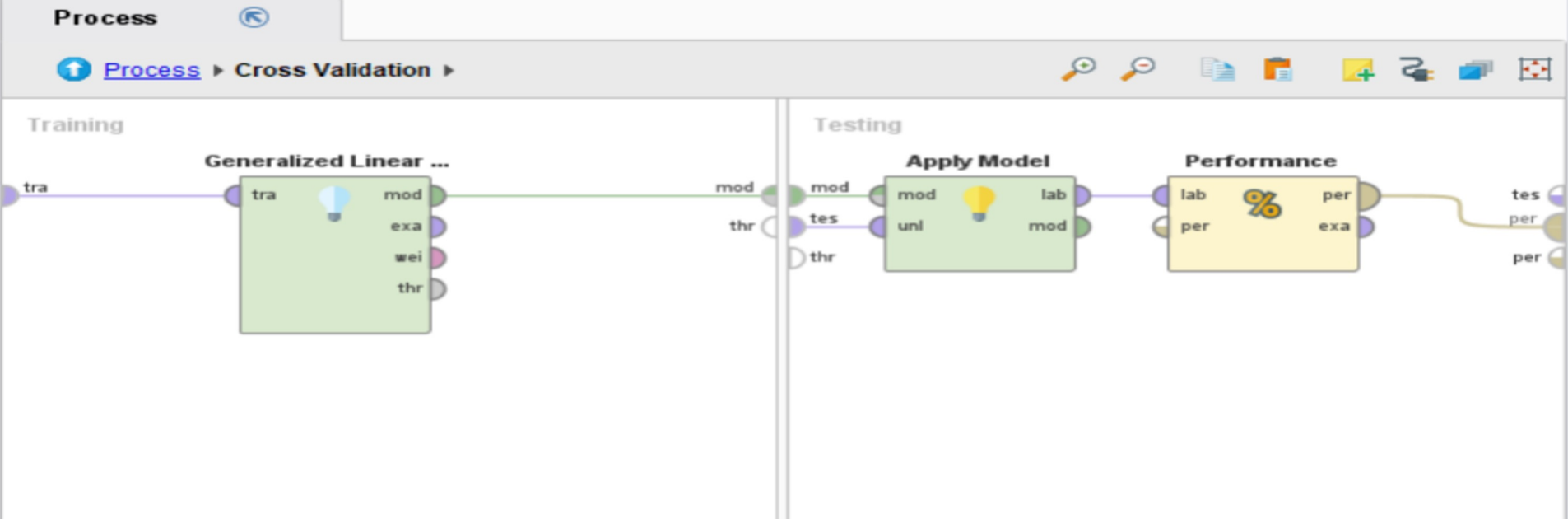
1. Average Utilisation Ratio
2. Education Level
3. Total Ct Chng Q4 to Q1
4. Card Category
5. Marital Status
7. Avg Open to buy
8. Total Revol balance
9. Dependants Count

# Steps performed for Generalised linear model

## Step-1

-First, the Data was extracted and retrieved and then imported from the Repository then Then, at that point, we associate it to set role to indicate the job for the extraordinary attribute "attrition flag" which it "Label" target role.the Cross Validation Operator was used. It performs cross-validation to assess the factual performance of a learning model.



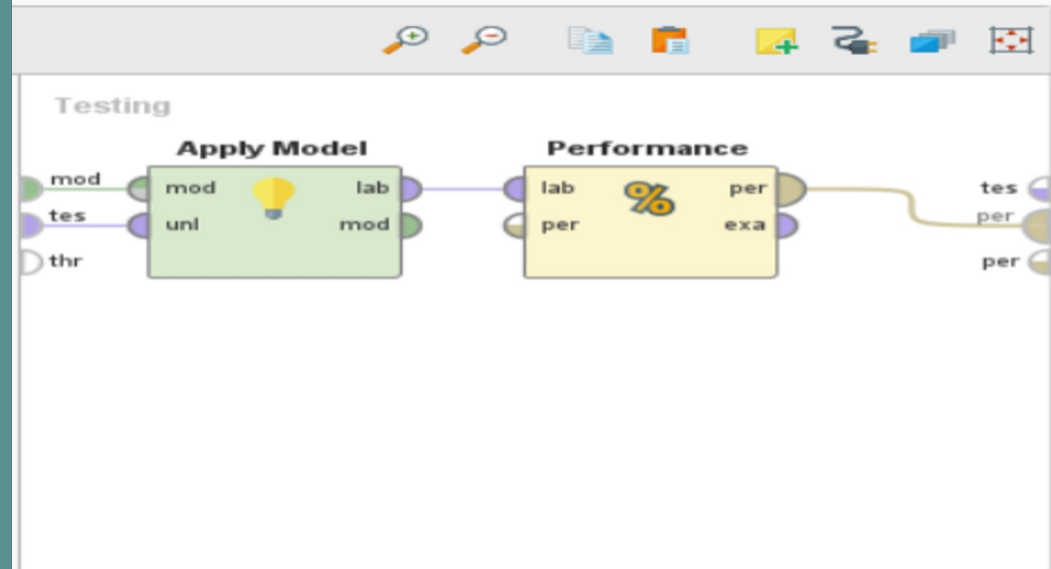


**Step2- how the process works inside Cross validation?**

Generalized linear model is used for for training purpose. While for testing the handling the information into model we associated model to the apply model

A model is first prepared on an ExampleSet by another Operator, which is much of the time a learning calculation.handling the information into model we associated model to the apply model

A model is first prepared on an ExampleSet by another Operator, which is much of the time a learning calculation.



### Step3-

Subsequent to interfacing with apply model named information shared to the Performance operator.

Here we have involved the presentation characterization in which we have applied our primary basis on accuracy, weighted mean recall, weighted mean precision.

In conclusion it's associated with the performance hub/node.

# Results

**Accuracy - 90.58%**

accuracy: 90.58% +/- 0.70% (micro average: 90.58%)

	true Existing Customer	true Attrited Customer	class precision
pred. Existing Customer	8213	672	92.44%
pred. Attrited Customer	281	949	77.15%
class recall	96.69%	58.54%	

## Precision - 84.86%

weighted\_mean\_precision: 84.86% +/- 2.02% (micro average: 84.80%), weights: 1, 1

	true Existing Customer	true Attrited Customer	class precision
pred. Existing Customer	8213	672	92.44%
pred. Attrited Customer	281	949	77.15%
class recall	96.69%	58.54%	

## Recall - 77.62%

**weighted\_mean\_recall: 77.62% +/- 1.13% (micro average: 77.62%), weights: 1, 1**

	true Existing Customer	true Attrited Customer	class precision
pred. Existing Customer	8213	672	92.44%
pred. Attrited Customer	281	949	77.15%
class recall	96.69%	58.54%	

# PerformanceVector

PerformanceVector:

accuracy: 90.58% +/- 0.70% (micro average: 90.58%)

ConfusionMatrix:

True:	Existing Customer	Attrited Customer
Existing Customer:	8213	672
Attrited Customer:	281	949

weighted\_mean\_recall: 77.62% +/- 1.13% (micro average: 77.62%), weights: 1, 1

ConfusionMatrix:

True:	Existing Customer	Attrited Customer
Existing Customer:	8213	672
Attrited Customer:	281	949

weighted\_mean\_precision: 84.86% +/- 2.02% (micro average: 84.80%), weights: 1, 1

ConfusionMatrix:

True:	Existing Customer	Attrited Customer
Existing Customer:	8213	672
Attrited Customer:	281	949



# Logistic Regression

- Logistic Regression is a Supervised machine learning algorithm that can be used to model the probability of a certain class or event.
- It is used when the data is linearly separable and the outcome is binary.

# Types of Logistic Regression

- Simple Logistic Regression: a single independent variable is used to predict the output
- Multiple logistic regression: multiple independent variables are used to predict the output

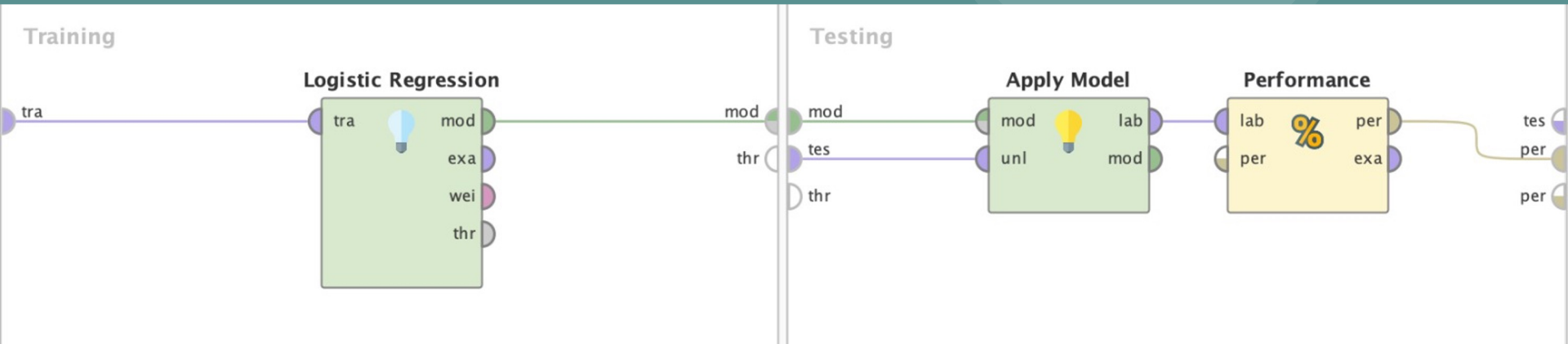
# Steps for Logistic Regression.

1. We imported the data set to the local repository.
2. Drag the data into the process.
3. Aim is to predict the target / dependent variable using the other variables. The target variable should be binomial, there must be only two possibilities.
  1. In next step, for better results, we have used cross validation to divide the data.
  2. After the cross validation we have connected the output with the model.

# Logistic Regression - Selected attributes

1. Average Utilisation Ratio
2. Education Level
3. Gender
4. Marital Status
5. Card Category
6. Total Revol balance
7. Total Trans Amt
8. Months on book

- After processing the data into model we connected model to the apply model.
- A model is first trained on an ExampleSet by another Operator. Afterwards, this model can be applied on another ExampleSet. Usually, the goal is to get a prediction on unseen data or to transform data by applying a preprocessing model.



# Performance Result

**Accuracy: - 90.43 %**

**accuracy: 90.43% +/- 0.70% (micro average: 90.43%)**

	true Existing Customer	true Attrited Customer	class precision
pred. Existing Customer	8195	669	92.45%
pred. Attrited Customer	299	952	76.10%
class recall	96.48%	58.73%	

# Weighted mean Recall:- 77.61 %

weighted\_mean\_recall: 77.61% +/- 1.43% (micro average: 77.60%), weights: 1, 1

	true Existing Customer	true Attrited Customer	class precision
pred. Existing Customer	8195	669	92.45%
pred. Attrited Customer	299	952	76.10%
class recall	96.48%	58.73%	

# Weighted mean Precision:- 84.33%

weighted\_mean\_precision: 84.33% +/- 1.88% (micro average: 84.28%), weights: 1, 1

	true Existing Customer	true Attrited Customer	class precision
pred. Existing Customer	8195	669	92.45%
pred. Attrited Customer	299	952	76.10%
class recall	96.48%	58.73%	



# Deep Learning - Srushtiben

- ❖ In deep learning, the algorithms are inspired by the structure of the human brain and known as neural networks. These neural networks are built from interconnected network switches designed to learn to recognize patterns in the same way the human brain and nervous system does.
- ❖ Rapidminer uses a multi layer feed forward neural network which uses back propagation for building the deep learning environment.

# Deep Learning

## ❖ Pseudo code for deep learning in Rapidminer

BEGIN

Initialize the neural network with the inputs and corresponding weights

For each hidden layer

The activation function is calculated and the weights are initialized

End For

The weights are updated in each hidden layer as per the gradient descent of the layer

Repeat until the error is minimized to the minimum level

END

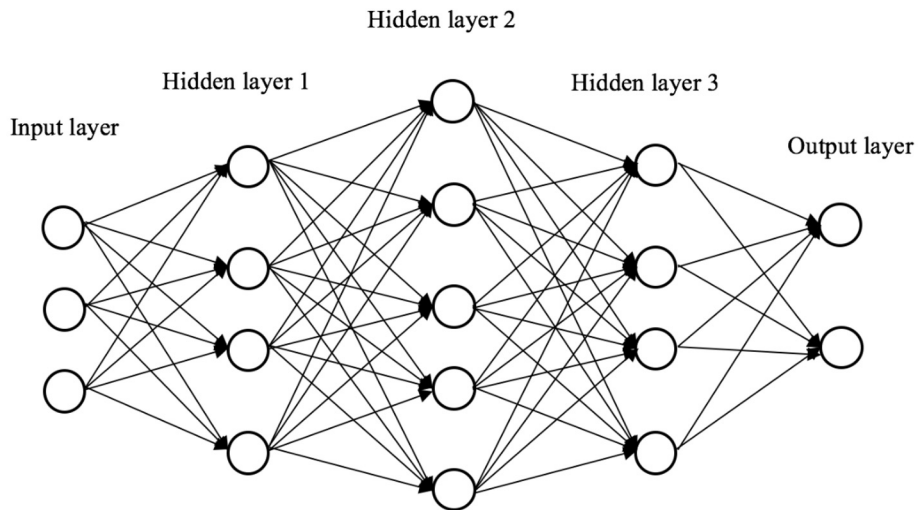
# Deep Learning

- ❖ Deep learning algorithms continue to optimize performance regardless of how much data is added to them. This means their optimal performance is much higher than previous learning algorithms, it also means that it takes vast amounts of data to outperform old models. In other words, they're one of the most data-hungry models we have.
- ❖ Another challenge of deep learning models is their inability to effectively deal with data that is different from their training set's distribution. When deep learning models are fed data that has some variations from their training data, they often perform poorly.

# Deep Learning

It connect perceptron units together to create a neural network; it has 3 sections:

1. Input Layer
2. Hidden Layer
3. Output Layer



A Deep Learning Model

# Deep Learning: data set

We started modeling by importing the bank churners dataset into repository.

**Retrieve BankChurners set for EDA**



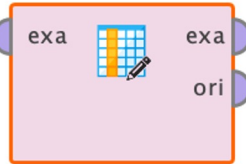
# Deep Learning: Set Role

Then we connect it to set role to specify the role for the special attribute “attrition flag” which it “Label” target role.


Retrieve BankChurn...



Set Role



**Parameters** ✕

 **Set Role**

attribute name

Attrition Flag ▼ ⓘ

target role

label ▼ ⓘ

regular

id

label

prediction

cluster

weight

batch

set additional roles

# Deep Learning: Selected attributes

1. Average Utilisation Ratio
2. Education Level
3. Total Amnt Chng Q4 to Q1
4. Total Ct Chng Q4 to Q1
5. Card Category
6. Income Category
7. Credit Limit
8. Total Revol balance
9. Total Trans Amt
10. Total Trans Ct
11. Months on book

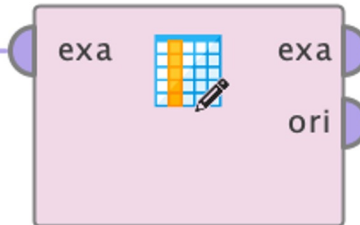
# Deep Learning: Filter Examples

- After setting role we have proceed with the next connect which is filter Examples in which we have selected no\_missing\_attributes for conditional class.
- Filter examples must be here because without it we can't perform cross validation.

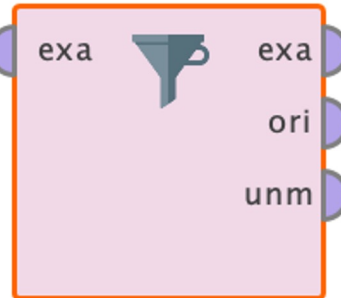
Retrieve BankChurn...



Set Role



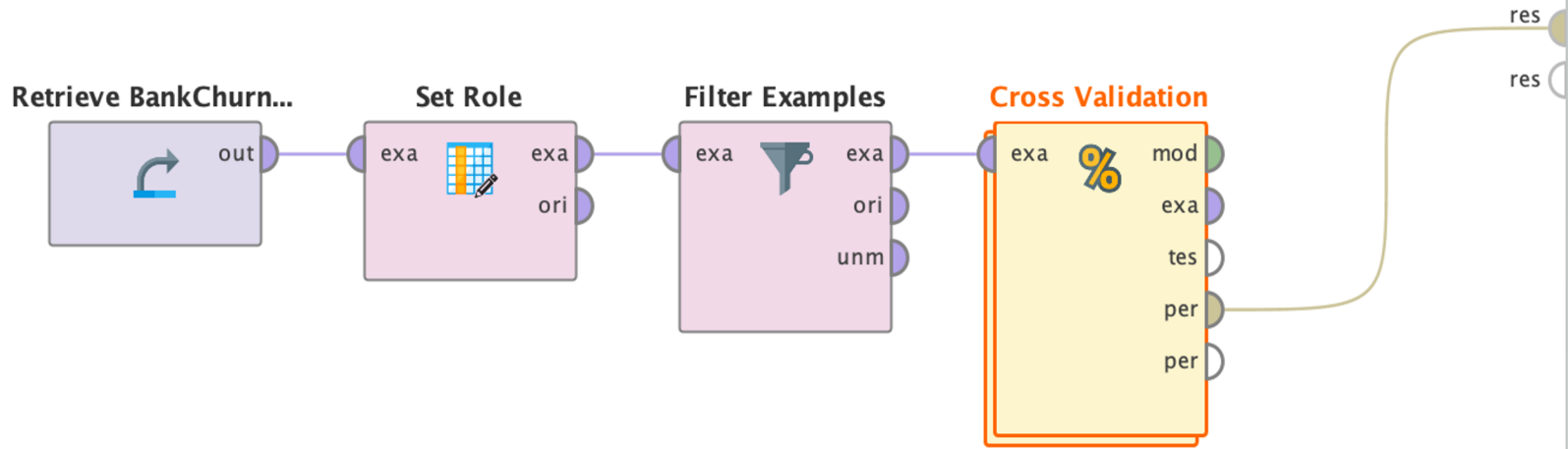
Filter Examples





# Deep Learning: Cross Validation

- Cross Validation Operator connected through Filter Examples and then connected with the results set.
- It performs a cross validation to estimate the statistical performance of a learning model.

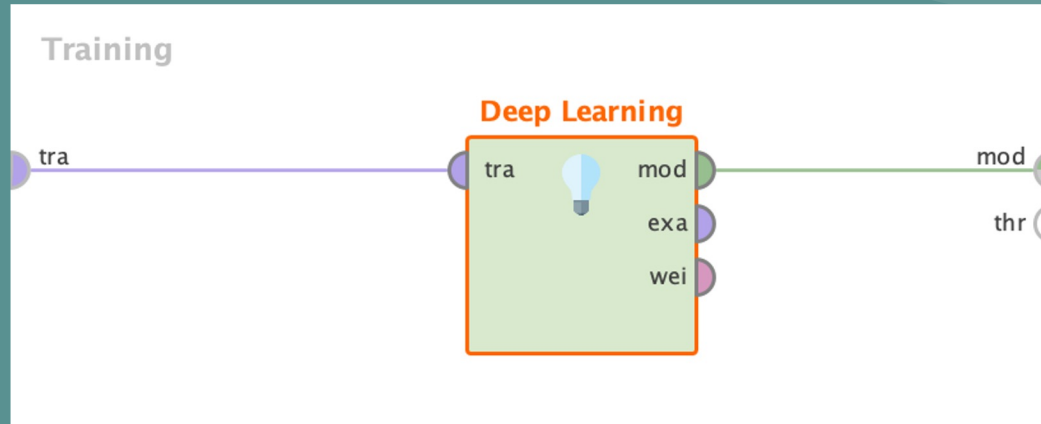


# Deep Learning: Cross Validation

- The Cross Validation Operator is a nested Operator. It has two subprocesses: a Training subprocess and a Testing subprocess.
- WE have added 10 fold in number of fold which means the number of folds (number of subsets) the ExampleSet should be divided into.
- Also the number of iterations that will take place is the same as the number of folds.
- If the model output port is connected, the Training subprocess is repeated one more time with all Examples to build the final model.

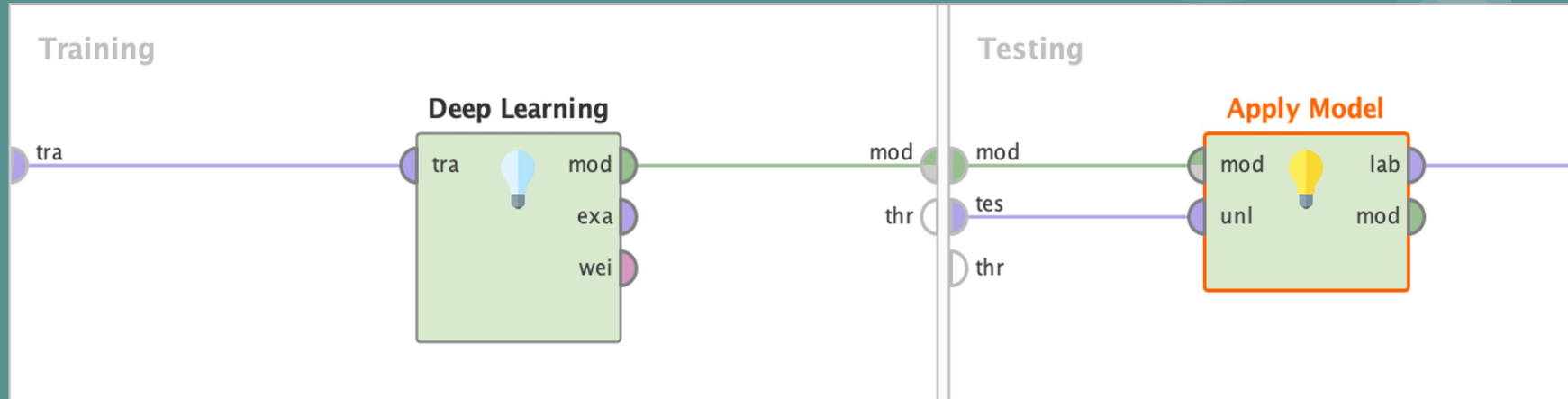
# Deep Learning: Model training part

- Moved forward with the training data connected to the Deep Learning model .
- Here we have changed to the suitable values for epochs, activation and for hidden layers and train sample per iteration for getting appropriate result of this model.
- Activation module selected to 'Tanh'.



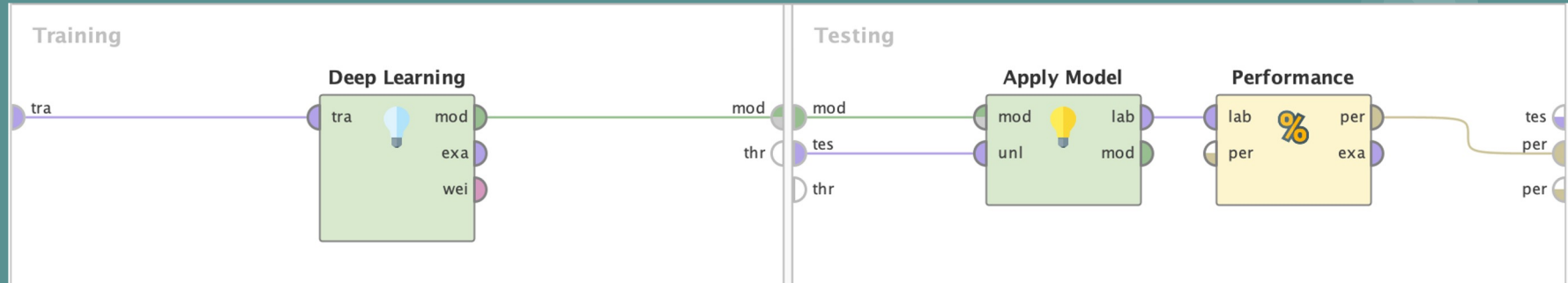
# Deep Learning: Model testing part [Apply Model]

- After processing the data into model we connected model to the apply model
- A model is first trained on an ExampleSet by another Operator, which is often a learning algorithm. Afterwards, this model can be applied on another ExampleSet. Usually, the goal is to get a prediction on unseen data or to transform data by applying a preprocessing model.

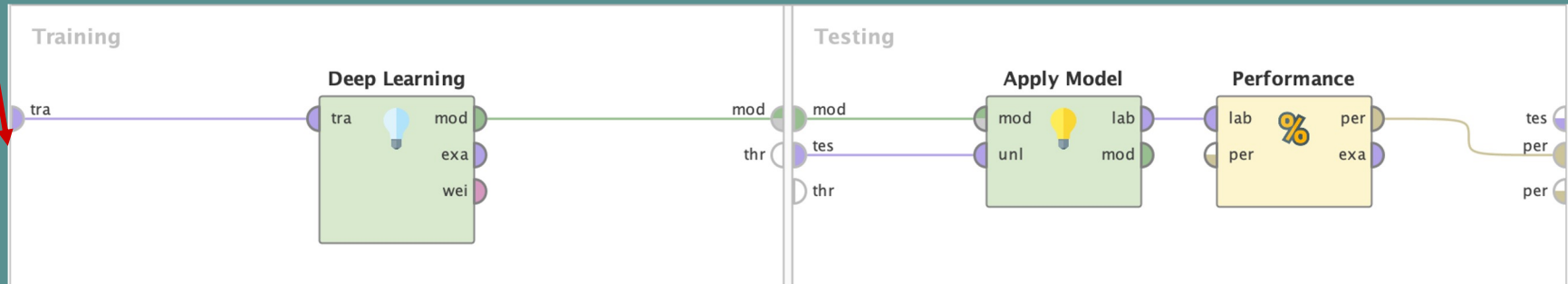
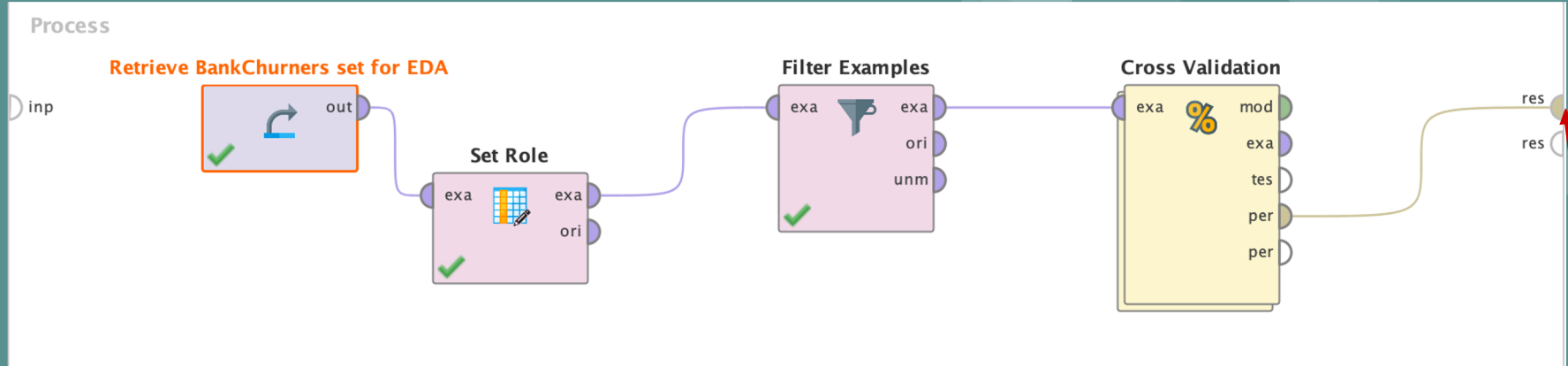


# Deep Learning: Model testing part [Performance]

- After connecting to apply model labeled data shared to the Performance operator.
- Here we have used the performance classification in which we have applied our main criterion, accuracy, weighted mean recall, weighted mean precision.
- Lastly it's connected to the performance node.



# Deep Learning: overall visual of process



# Performance Results

## 1. Accuracy 95.73%

accuracy: 95.73% +/- 0.54% (micro average: 95.73%)

	true Existing Customer	true Attrited Customer	class precision
pred. Existing Customer	8287	225	97.36%
pred. Attrited Customer	207	1396	87.09%
class recall	97.56%	86.12%	

## 2. Weighted mean precision 92.24%

**weighted\_mean\_precision: 92.24% +/- 0.94% (micro average: 92.22%), weights: 1, 1**

	true Existing Customer	true Attrited Customer	class precision
pred. Existing Customer	8287	225	97.36%
pred. Attrited Customer	207	1396	87.09%
class recall	97.56%	86.12%	



### 3. Weighted mean recall 91.84%

**weighted\_mean\_recall: 91.84% +/- 1.46% (micro average: 91.84%), weights: 1, 1**

	true Existing Customer	true Attrited Customer	class precision
pred. Existing Customer	8287	225	97.36%
pred. Attrited Customer	207	1396	87.09%
class recall	97.56%	86.12%	

# Performance Vector

## PerformanceVector

PerformanceVector:

accuracy: 95.73% +/- 0.54% (micro average: 95.73%)

ConfusionMatrix:

True:	Existing Customer	Attrited Customer
-------	-------------------	-------------------

Existing Customer:	8287	225
--------------------	------	-----

Attrited Customer:	207	1396
--------------------	-----	------

weighted\_mean\_recall: 91.84% +/- 1.46% (micro average: 91.84%), weights: 1, 1

ConfusionMatrix:

True:	Existing Customer	Attrited Customer
-------	-------------------	-------------------

Existing Customer:	8287	225
--------------------	------	-----

Attrited Customer:	207	1396
--------------------	-----	------

weighted\_mean\_precision: 92.24% +/- 0.94% (micro average: 92.22%), weights: 1, 1

ConfusionMatrix:

True:	Existing Customer	Attrited Customer
-------	-------------------	-------------------

Existing Customer:	8287	225
--------------------	------	-----

Attrited Customer:	207	1396
--------------------	-----	------

# Conclusions

- As the results of the deep learning model and the phase 2 requirement of getting accuracy, precision and recall at least 90% which is only fulfilled through DL model.
- All the requirements satisfied with getting performance of accuracy, precision and recall above the 90% so we look forward to go with “Deep Learning Model” rather than going with other suggested models.

# Thank You

