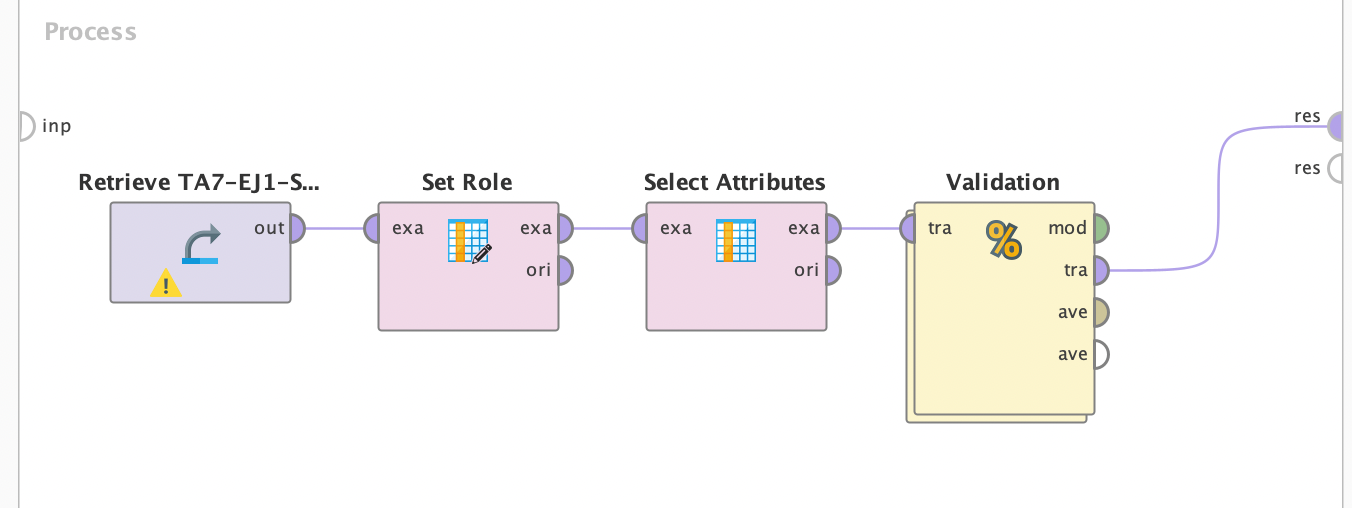
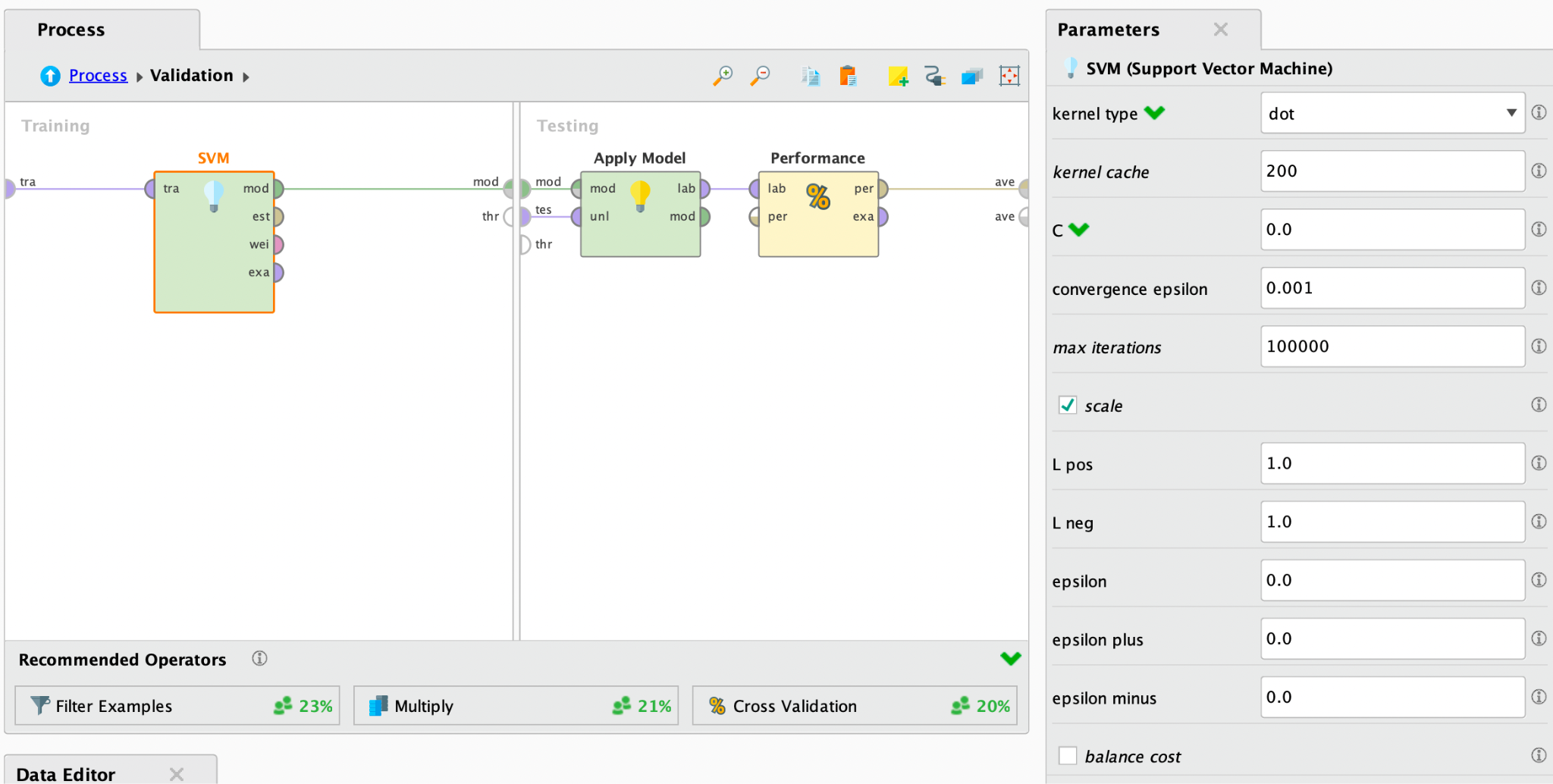
## Proceso



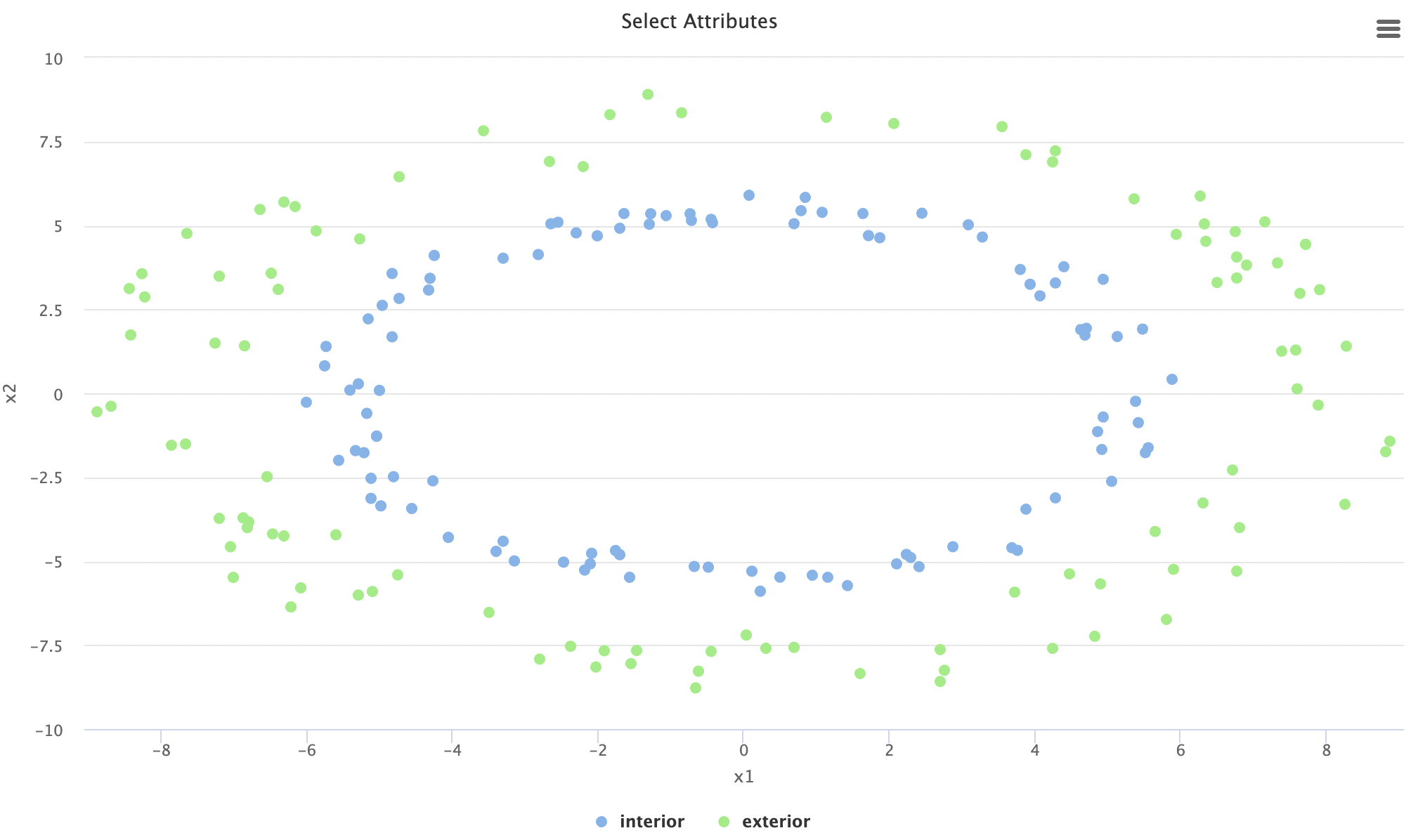
## SVM



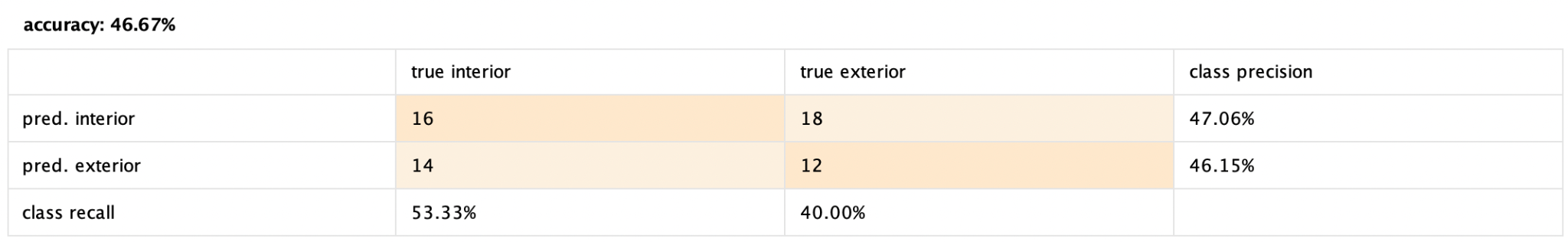
**Analizar brevemente la documentación del operador “SVM” y resumir qué significan y cuáles son los posibles valores.**

* kernel\_type
  + The type of the kernel function is selected through this parameter. Following kernel types are supported: dot, radial, polynomial, neural, anova, epachnenikov, gaussian combination, multiquadric
  + dot: The dot kernel is defined by k(x,y)=x\*y i.e. it is inner product of x and y.
  + radial: The radial kernel is defined by exp(-g ||x-y||^2) where g is the gamma, it is specified by the kernel gamma parameter. The adjustable parameter gamma plays a major role in the performance of the kernel, and should be carefully tuned to the problem at hand.
  + polynomial: The polynomial kernel is defined by k(x,y)=(x\*y+1)^d where d is the degree of polynomial and it is specified by the kernel degree parameter. The polynomial kernels are well suited for problems where all the training data is normalized.
  + neural: The neural kernel is defined by a two layered neural net tanh(a x\*y+b) where a is alpha and b is the intercept constant. These parameters can be adjusted using the kernel a and kernel b parameters. A common value for alpha is 1/N, where N is the data dimension. Note that not all choices of a and b lead to a valid kernel function.
  + anova: The anova kernel is defined by raised to power d of summation of exp(-g (x-y)) where g is gamma and d is degree. gamma and degree are adjusted by the kernel gamma and kernel degree parameters respectively.
  + epachnenikov: The epachnenikov kernel is this function (3/4)(1-u2) for u between -1 and 1 and zero for u outside that range. It has two adjustable parameters kernel sigma1 and kernel degree.
  + gaussian\_combination: This is the gaussian combination kernel. It has adjustable parameters kernel sigma1, kernel sigma2 and kernel sigma3.
  + multiquadric: The multiquadric kernel is defined by the square root of ||x-y||^2 + c^2. It has adjustable parameters kernel sigma1 and kernel sigma shift.
  + Range: selection
* kernel\_gamma
  + This is the SVM kernel parameter gamma. This is available only when the kernel type parameter is set to radial or anova.
  + Range: real
* kernel\_sigma1
  + This is the SVM kernel parameter sigma1. This is available only when the kernel type parameter is set to epachnenikov, gaussian combination or multiquadric.
  + Range: real
* kernel\_sigma2
  + This is the SVM kernel parameter sigma2. This is available only when the kernel type parameter is set to gaussian combination.
  + Range: real
* kernel\_sigma3
  + This is the SVM kernel parameter sigma3. This is available only when the kernel type parameter is set to gaussian combination.
  + Range: real
* kernel\_shift
  + This is the SVM kernel parameter shift. This is available only when the kernel type parameter is set to multiquadric.
  + Range: real
* kernel\_degree
  + This is the SVM kernel parameter degree. This is available only when the kernel type parameter is set to polynomial, anova or epachnenikov.
  + Range: real
* kernel\_a
  + This is the SVM kernel parameter a. This is available only when the kernel type parameter is set to neural.
  + Range: real
* kernel\_b
  + This is the SVM kernel parameter b. This is available only when the kernel type parameter is set to neural.
  + Range: real
* kernel\_cache
  + This is an expert parameter. It specifies the size of the cache for kernel evaluations in megabytes.
  + Range: real
* C
  + This is the SVM complexity constant which sets the tolerance for misclassification, where higher C values allow for 'softer' boundaries and lower values create 'harder' boundaries. A complexity constant that is too large can lead to over-fitting, while values that are too small may result in over-generalization.
  + Range: real
* convergence\_epsilon
  + This is an optimizer parameter. It specifies the precision on the KKT conditions.
  + Range:
* max\_iterations
  + This is an optimizer parameter. It specifies to stop iterations after a specified number of iterations.
  + Range: integer
* scale
  + This is a global parameter. If checked, the example values are scaled and the scaling parameters are stored for a test set.
  + Range: boolean
* L\_pos
  + A factor for the SVM complexity constant for positive examples. This parameter is part of the loss function.
  + Range: real
* L\_neg
  + A factor for the SVM complexity constant for negative examples.This parameter is part of the loss function.
  + Range: real
* epsilon
  + This parameter specifies the insensitivity constant. No loss if the prediction lies this close to true value. This parameter is part of the loss function.
  + Range: real
* epsilon\_plus
  + This parameter is part of the loss function. It specifies epsilon for positive deviation only.
  + Range: real
* epsilon\_minus
  + This parameter is part of the loss function. It specifies epsilon for negative deviation only.
  + Range: real
* balance\_cost
  + If checked, adapts Cpos and Cneg to the relative size of the classes.
  + Range: boolean
* quadratic\_loss pos
  + Use quadratic loss for positive deviation. This parameter is part of the loss function.
  + Range: boolean
* quadratic\_loss\_neg
  + Use quadratic loss for negative deviation. This parameter is part of the loss function.
  + Range: boolean

## Ejecucion e interpretacion



Usando los parámetros por defecto, el resultado es:



Al cambiar la configuración del operador de SVM de lineal a polynomial, los resultados son los siguientes:

