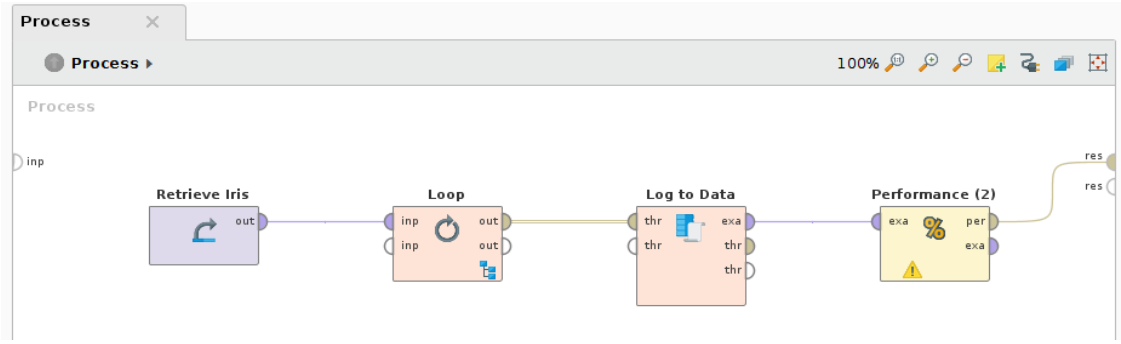
Tutorial básico de comparação de desempenho de algoritmos de classificação em RapidMiner

Neste tutorial, aprenderemos a aplicar diversos algoritmos de classificação com diferentes valores de parâmetros. Faremos isso através de amostragens *Random subsampling* e validação cruzada. Para isso, é necessário ter dominado o conteúdo do *Tutorial básico de classificação em RapidMiner*.

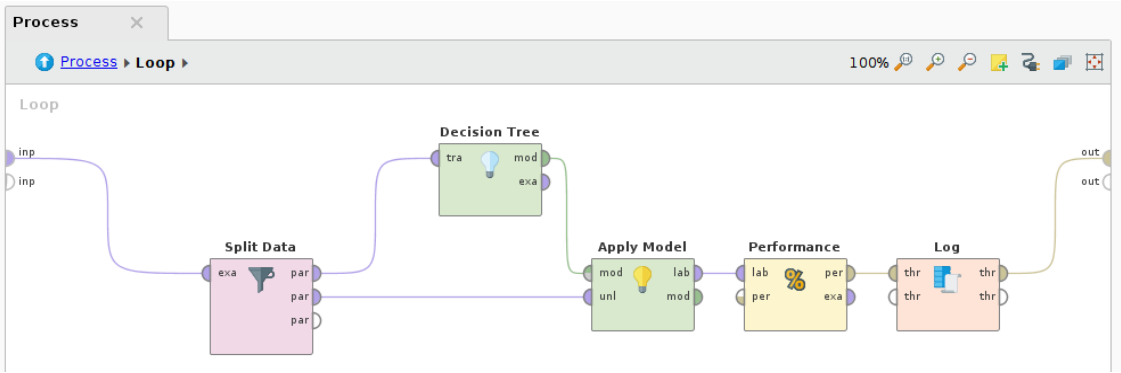
1. *Random subsampling* e três algoritmos de classificação

Nesta seção, aplicaremos árvore de decisão, rede neural e k-nn na base de dados Iris através de 5 iterações de *Random subsampling*. Poderíamos aplicar qualquer algoritmo de classificação em qualquer base suportada pelo algoritmo.

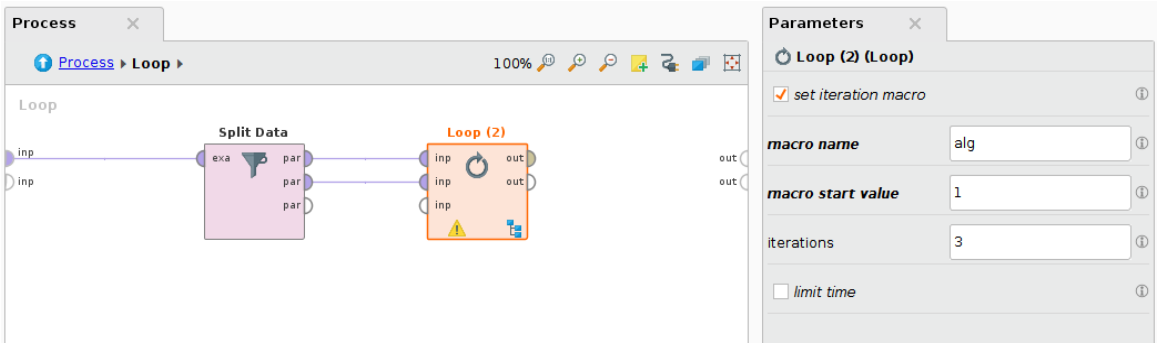
1.1 Aplique um *Random subsamplig* na base Iris, como feito no *Tutorial básico de classificação em RapidMiner*.



1.2 Entre no operador Loop.

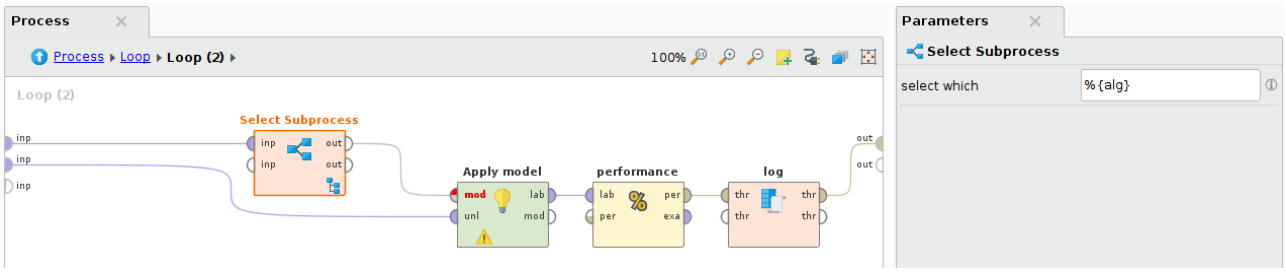


1.3 Queremos fazer com que cada partição gerada pelo operador *Split Data* seja utilizada como treinamento e teste para cada um dos algoritmos de classificação que desejamos avaliar. Para isso, remova o operador *Decisio Tree* e reserve os outros operadores seguintes ao *Split Data*. Inclua mais um operador *Loop*. Este novo *Loop* será responsável por aplicar cada um dos algoritmos de classificação. Nós definimos o parâmetro *iterations* como 3 porque desejamos avaliar o desempenho de três diferentes algoritmos de classificação.

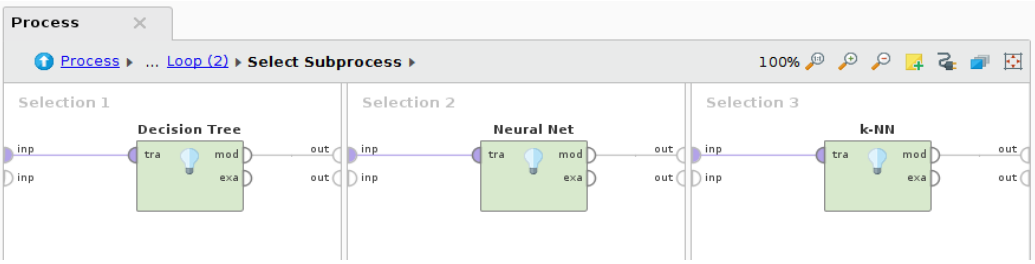


O que são macros?
Macros são variáveis de processo. Com elas conseguimos guardar valores ou acessar e controlar valores em nosso processo. Por exemplo, no nosso *Loop*, conseguimos definir uma macro de iteração com o nome "alg". A cada iteração no *Loop*, a macro alg terá seu valor alterado. Assim conseguimos saber, dentro do loop, em que iteração estamos e ligar cada iteração com um algoritmo de classificação diferente.

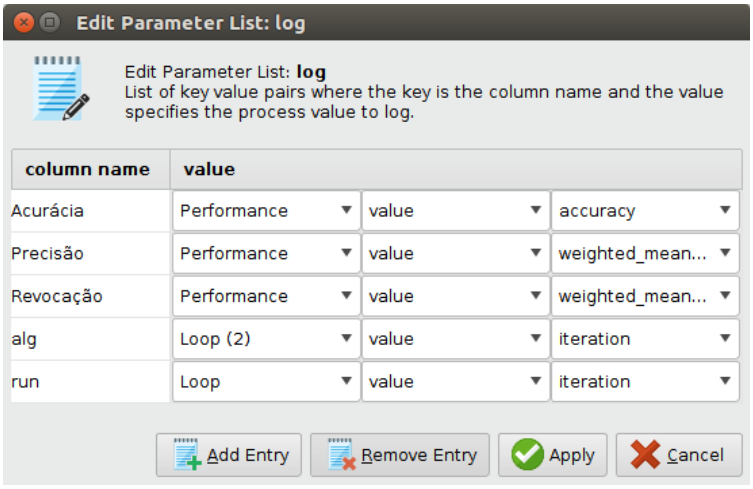
1.4 Dentro do novo *Loop* podemos treinar e aplicar cada um dos modelos que desejamos. Lembre-se: cada iteração do loop representará um algoritmo de classificação diferente. Para deixarmos explícito para o RapidMiner qual algoritmo de classificação desejamos aplicar em cada iteração, utilizamos o operador *Select Subprocess* e dizemos que o subprocesso escolhido dependerá da macro alg.



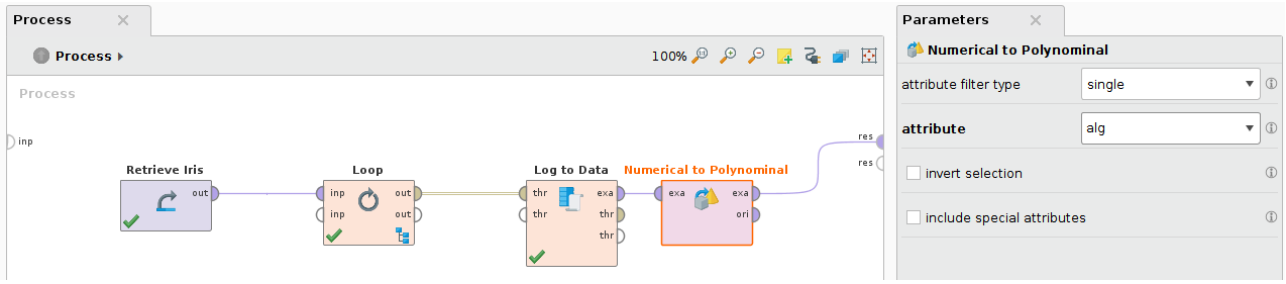
1.5 Dentro do subprocesso *Select Subprocess*, podemos colocar diversos caminhos possíveis que o processo poder tomar. Queremos que em cada um desses caminhos seja aplicado um algoritmo de classificação diferente. Na primeira iteração do *Loop (2)* queremos que ele aplique uma árvore de decisão; na segunda iteração, uma rede neural; e na terceira iteração, um k-nn.



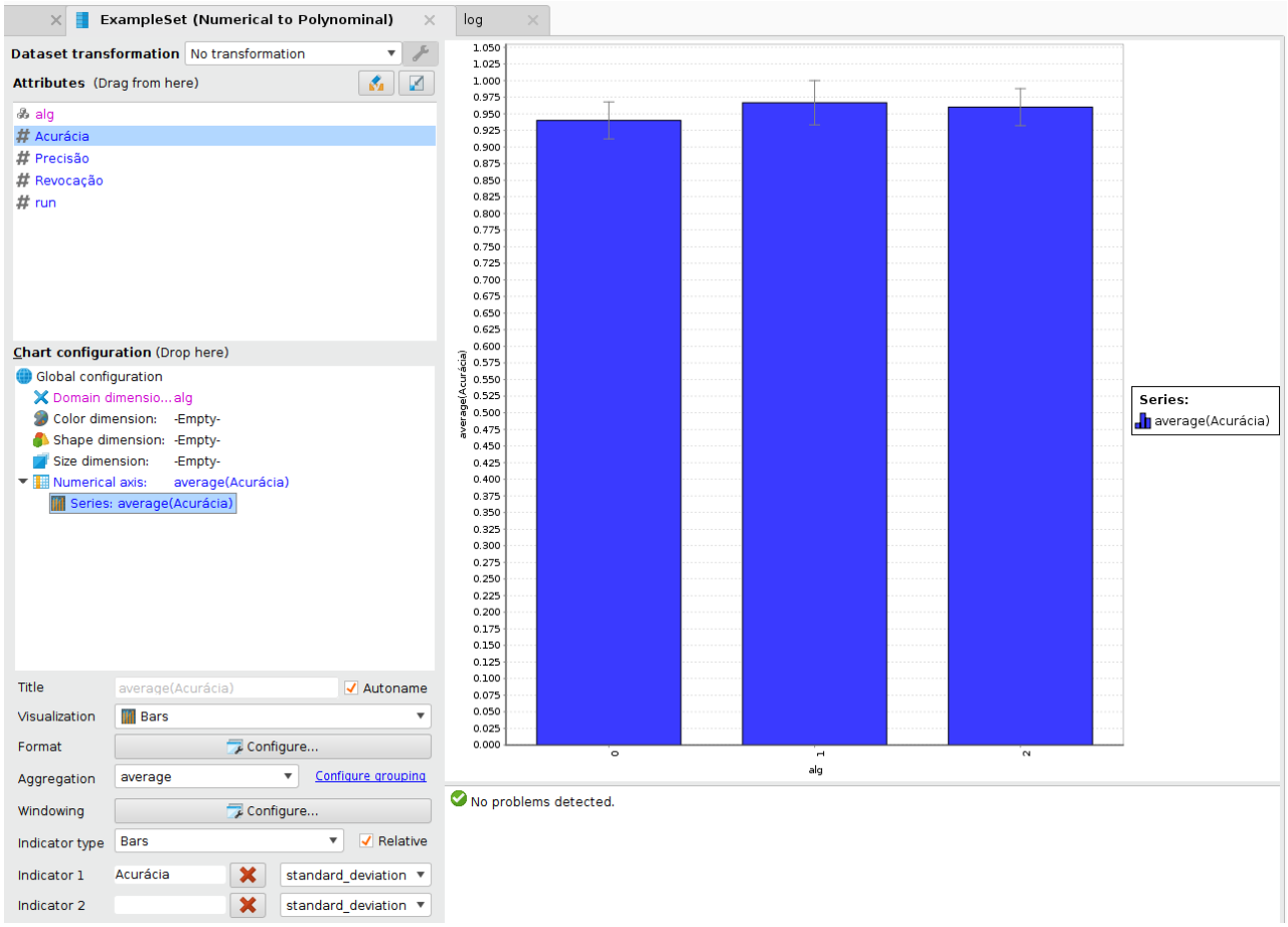
1.6 Voltando ao subprocesso do *Loop (2)*, é interessante que nossa saída contenha a informação sobre qual algoritmo de classificação obteve determinada performance. Para isso, incluiremos no operador *Log* essa informação. Incluiremos também o número da iteração do *Random subsampling* com o nome de *run*.



1.7 Uma sugestão para ajudar na visualização dos resultados é, no processo principal, transformar o atributo “alg” em polinomial.



1.8 Podemos gerar um gráfico de barras comparando a acurácia média e o desvio padrão de cada um dos algoritmos. Lembrando que, como esquematizamos no processo, o algoritmo zero é a árvore de decisão, o algoritmo 1 é a rede neural e o algoritmo 2 é o k-nn.



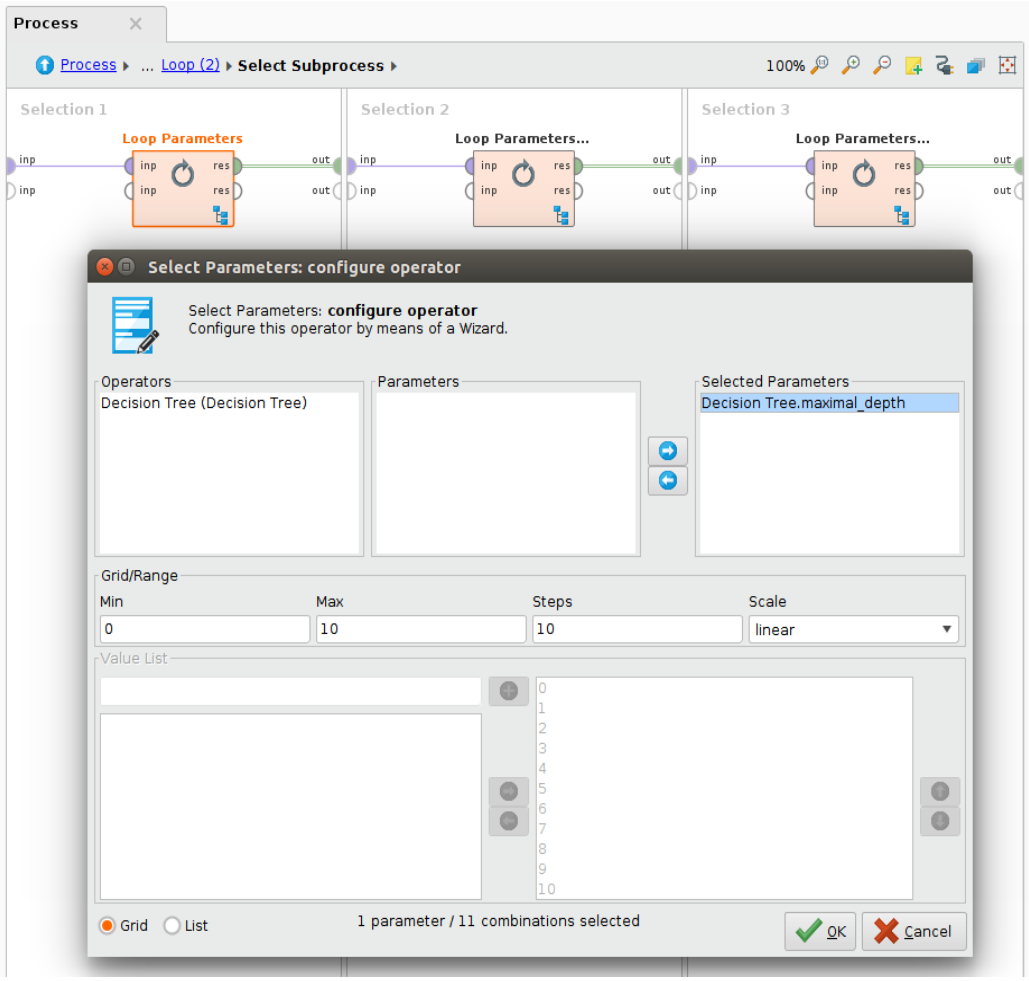
Questionamento: Escolhemos três algoritmos diferentes porque eles possuem vieses diferentes, pois cada algoritmo de classificação "corta" o espaço de uma maneira diferente. Através dos resultados obtidos, podemos dizer que o viés da árvore de decisão é o menos adequado para o problema em questão?

---

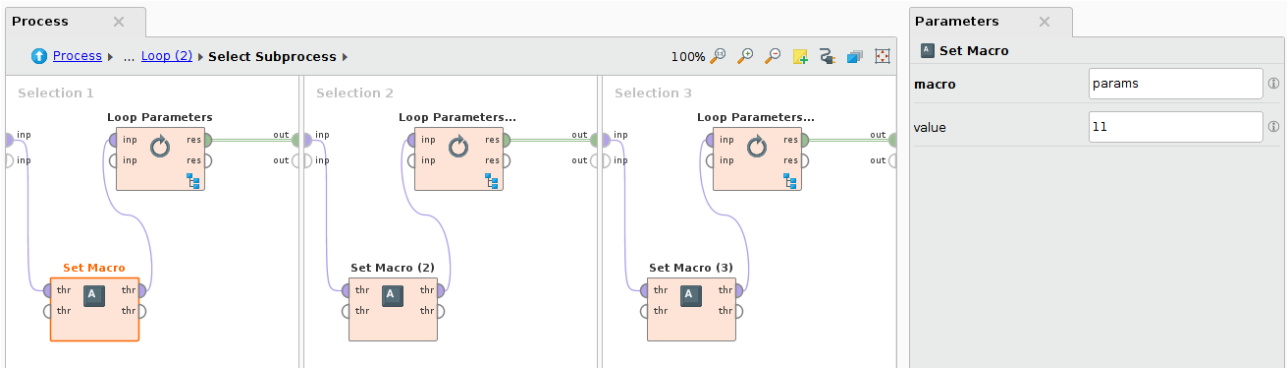
2. *Random subsampling* com variação de parâmetro nos algoritmos de classificação

Nesta seção variaremos os valores de um parâmetro de cada algoritmo de classificação. Compararemos o desempenho do algoritmo de classificação nos diferentes valores de parâmetro.

2.1 Dentro do subprocesso *Select subprocess*, faça um *Loop Parameter* para cada um dos algoritmo de classificação. Varie os parâmetros como já fizemos em outras aulas. A figura abaixo apresenta um exemplo de variação de parâmetro para a árvore de decisão. Varie os parâmetros da rede neural e do k-nn.

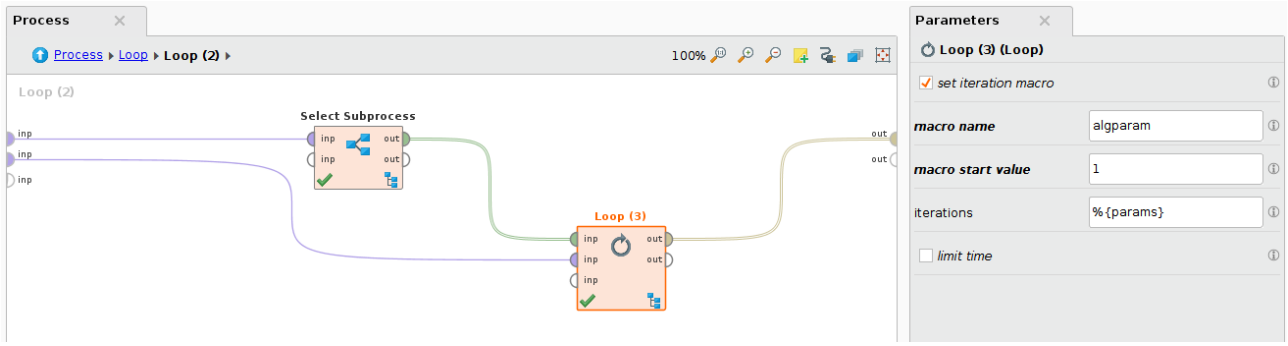


2.2 Para cada alteração em um parâmetro no algoritmo de classificação, é gerado um modelo diferente. Cada um desses modelos será aplicado no conjunto de teste. Portanto, precisamos informar ao processo quantas vezes ele deverá aplicar um modelo em um conjunto de teste. Por exemplo, no nosso caso ele aplicará 11 vezes para a árvore de decisão (pois definimos 11 valores diferentes de parâmetro). Para isso utilizaremos uma macro e a definiremos como 11. Faça o mesmo para cada um dos algoritmos de classificação, definindo a mesma macro (com o mesmo nome) com o número de valores de parâmetros utilizado.

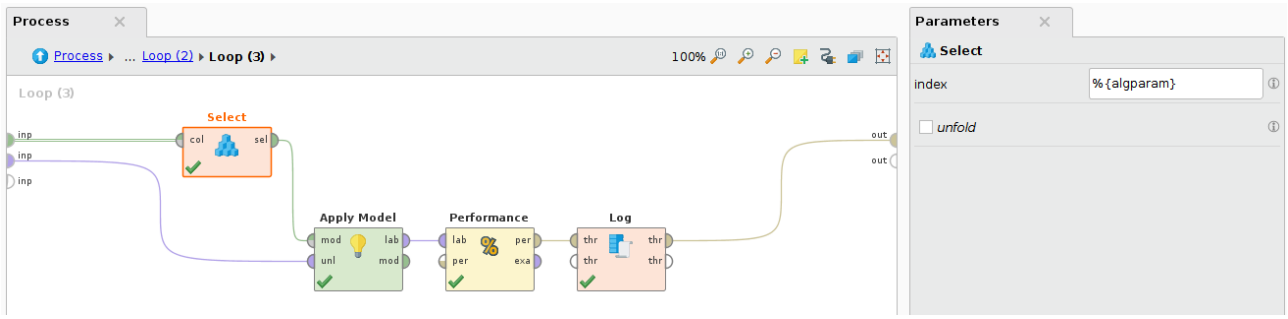


2.3 Volte ao subprocesso *Loop (2)*. Agora a saída do operador *Select Subprocess* não é mais um modelo, e sim uma coleção de modelos. No caso da árvore de decisão do nosso exemplo, uma coleção de 11 modelos. Por isso, não podemos utilizar diretamente o operador *Apply Model*, pois ele espera apenas um modelo como entrada e não uma coleção de modelos. Para aplicarmos os

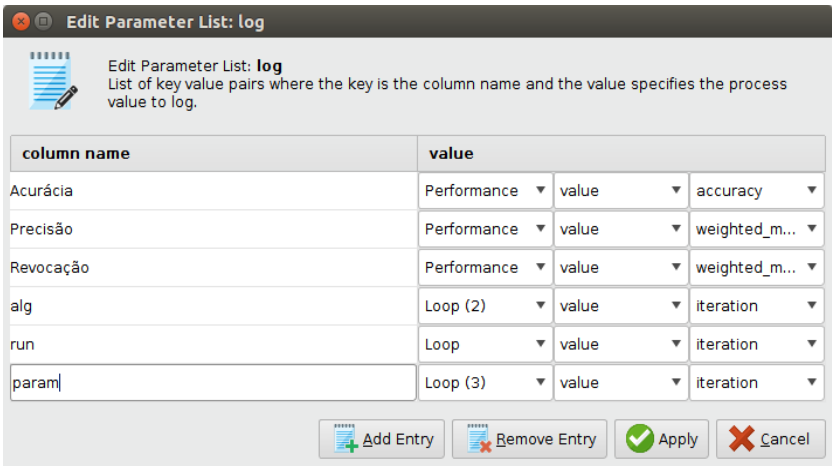
modelos um a um, faremos outro *Loop* e ele terá *params* iterações, sendo *params* a macro que define o número de valores diferentes utilizados no parâmetro do algoritmo de classificação.



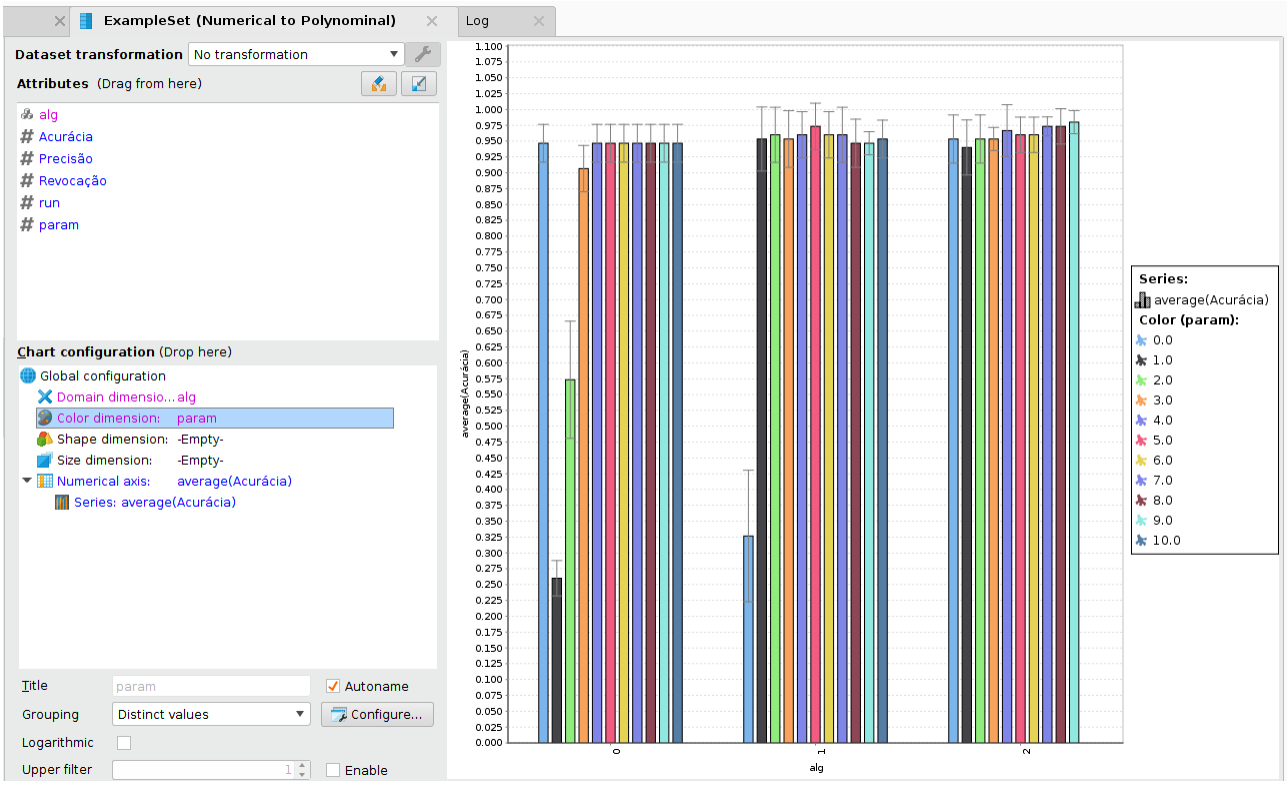
2.4 No subprocesso do operador *Loop (3)*, precisamos apenas selecionar qual modelo da coleção nós desejamos aplicar e aplicá-lo da mesma forma que já fazemos. Para isso, utilizaremos o operador *Select* da pasta *Collections* utilizando a macro *algparam*.



2.5 Podemos incluir as informações de parâmetro no *Log*.



2.6 Se rodarmos nosso processo, podemos analisar a tabela de resultados e gerar um gráfico comparativo como o da figura abaixo



Questionamentos: Qual algoritmo de classificação e com quais parâmetros você escolheria para esta base de dados? Podemos garantir com certeza absoluta que a função escolhida é igual a função real que define o problema na natureza?