**Taller:** Clase 13

**Estudiante:** Danny Sebastián Díaz Padilla

**Fecha:** 20/11/19

**Tema:** Análisis del template **‘Predictive Maintenance’** de RapidMiner

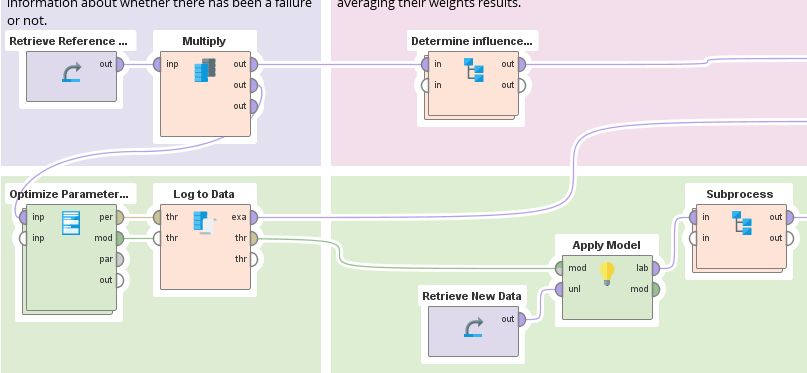


1. **Proceso**

Mantenimiento Predictivo

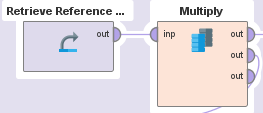
Modela fallas de equipos basadas en observaciones de referencias de datos de ejecuciones y fallas de máquinas anteriores. El modelo se aplica aplicar a situaciones actuales para anticipar las fallas de la máquina y programar el mantenimiento de manera preventiva.

El esquema del proceso completo es el siguiente:

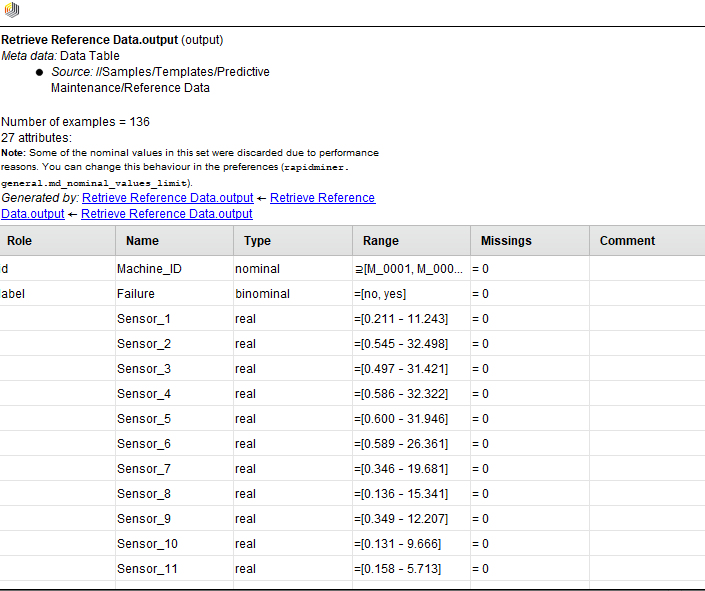


1. **Primer paso:**

Se carga los datos de las ejecuciones anteriores de la máquina, etiquetadas con información sobre si ha habido una falla o no.



Mediante el operador ‘**Retrieve’** se carga la base de datos con la información de las máquinas.

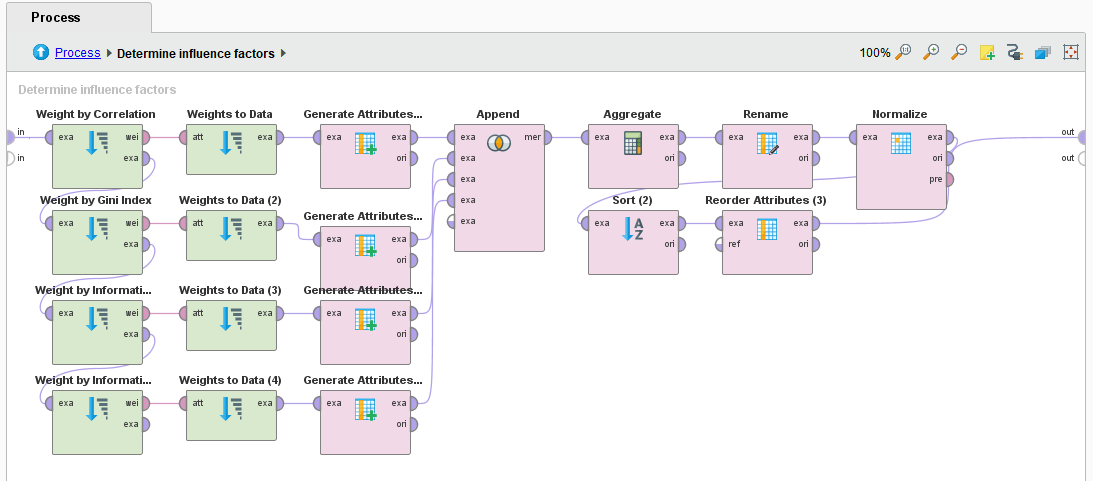


Se toma esta entrada y se agrega al operador ‘**Multiply’** el cual crea una copia de la información enviándola al paso 2 y al paso 3.

1. **Segundo paso:**

Se determina los factores de influencia utilizando varios algoritmos de ponderación de atributos y promediando sus resultados de ponderaciones.

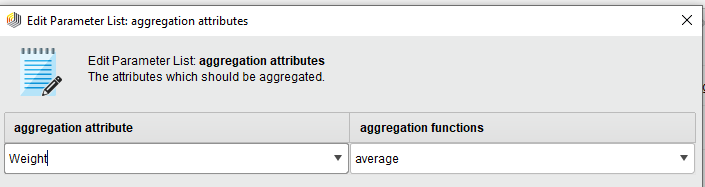
Usando la entrada del primer paso, se realiza un subproceso que se crea un peso (relevancia) por cada atributo de acuerdo a: la correlación índice Gini, ganancia de información e índice de ganancia de información.



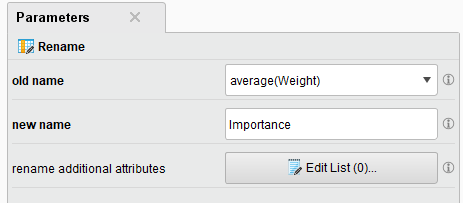
Luego para cada peso se genera un conjunto de datos de ejemplo, el operador Generar atributos construye nuevos atributos a partir de los atributos de la entrada y constantes arbitrarias utilizando expresiones matemáticas.

Luego usando el operador ‘Append’ se combinan todos los conjuntos de ejemplos.

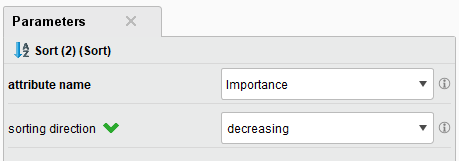
A este conjunto combinado se le aplica una función de agregación (Aggregate) para obtener el promedio de los pesos.



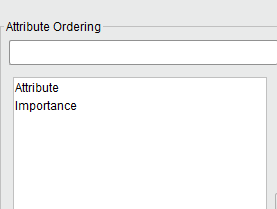
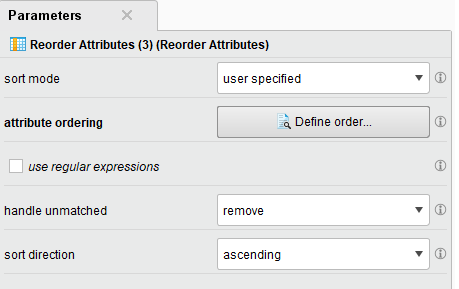
Y le da un nuevo nombre usando el operador ‘**Rename’:**



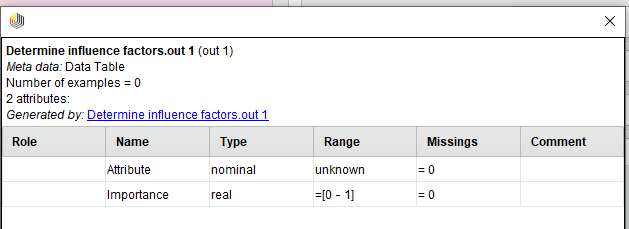
Y por un segundo canal envía la información original que recibió (sin realizar la agregación) para ordenarla en forma descendente por **importancia.**



Y finalmente redistribuir sus atributos.

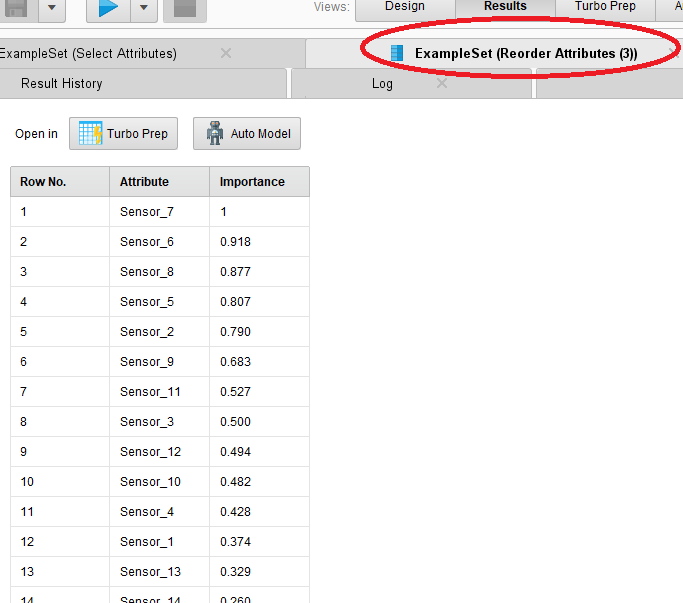


La salida de este paso son estos dos conjuntos de datos ya explicados.



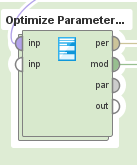
Además, su resultado es final, es decir, ya no pasa por más procesos.

Dando como salida:

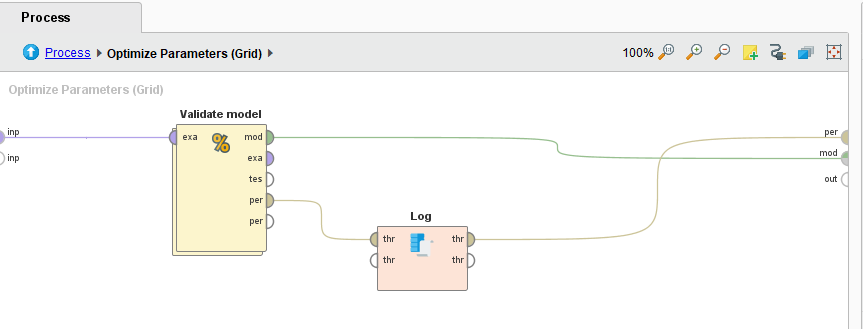


1. **Tercer paso:**

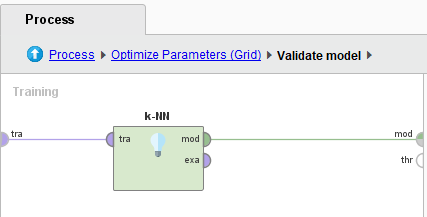
Se entrena un modelo k-NN: se optimiza para k (el número de situaciones de referencia a tener en cuenta para la predicción) para producir una precisión de predicción de falla máxima.



Usando la salida del primer paso como entrada, se crea un proceso interno que validará el modelo.



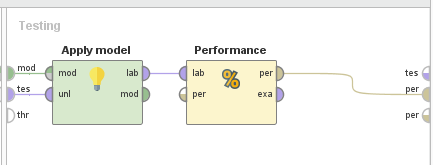
Dentro del proceso interno del subproceso se usa el operador ‘**k-NN’** para realizar el entrenamiento del modelo, operador que genera un modelo de k vecinos más cercano, que se utiliza para clasificación o regresión. Se basa básicamente en la comparación de un ejemplo desconocido con los ejemplos de entrenamiento k que son los vecinos más cercanos del ejemplo desconocido. Los cuales corresponden a una parte de la entrada del primer paso.



La salida de esta parte es el modelo ya entrenado.

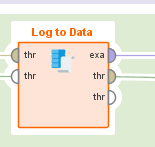
Para la parte del testeo se usa el operador ‘**Apply model’** para aplicar el modelo sobre un subconjunto de testeo previamente obtenido de la entrada del primer paso.

Y luego evalua el desempeño del modelo usando el operador ‘**Performance’** la única salida de esta parte es las medidas de desempeño

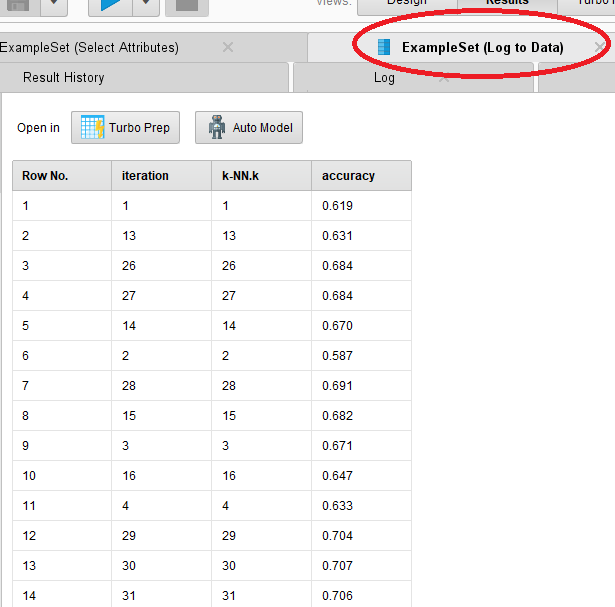


Finalmente se utiliza el operador ‘**Log’** para recuperar las medidas de desempeño.

La salida de este subprocesso consiste en un modelo entrenado y en las mediciones de desempeño.



Finalmente usando el operador ‘**Log to Data’** se genera un conjunto de ejemplos usando la información del ‘**Log’** previamente recuperado.



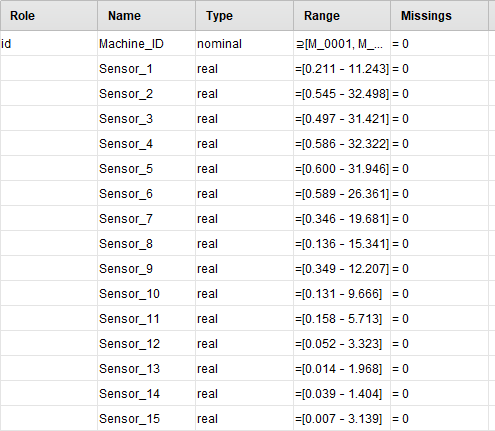
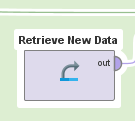
La salida de este paso consiste en la data generada desde el log y la salida del subproceso anteriormente detallado.

Además, el conjunto de ejemplos es un resultado final.

1. **Cuarto paso:**

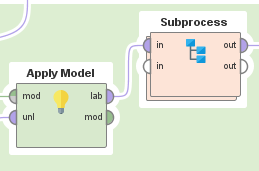
Se carga nuevos datos y se aplica el modelo de falla de la máquina a las ejecuciones actuales de la máquina para predecir posibles fallas.

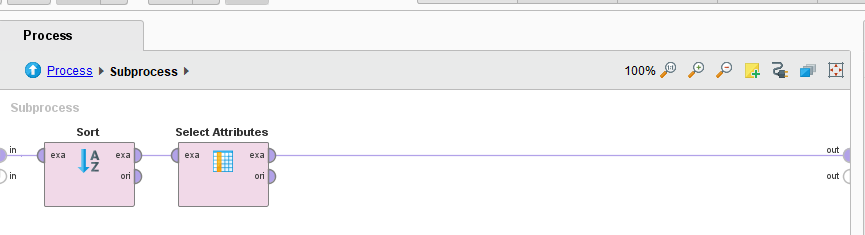
Primero se carga de nuevo los datos con el operador ‘**Retrieve’**



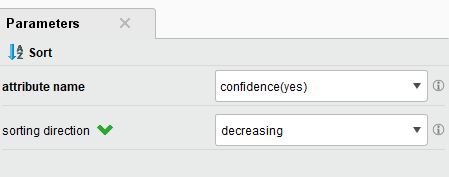
Se aplica el modelo usando ‘**Apply model’** al conjunto de datos generados desde el Log utilizando el modelo generado en el tercer paso.

Y luego por medio de un subproceso se trata la data generada.

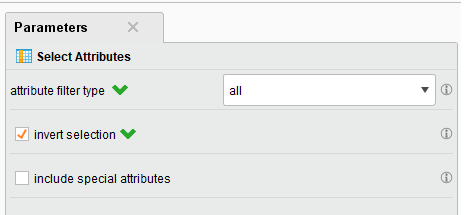




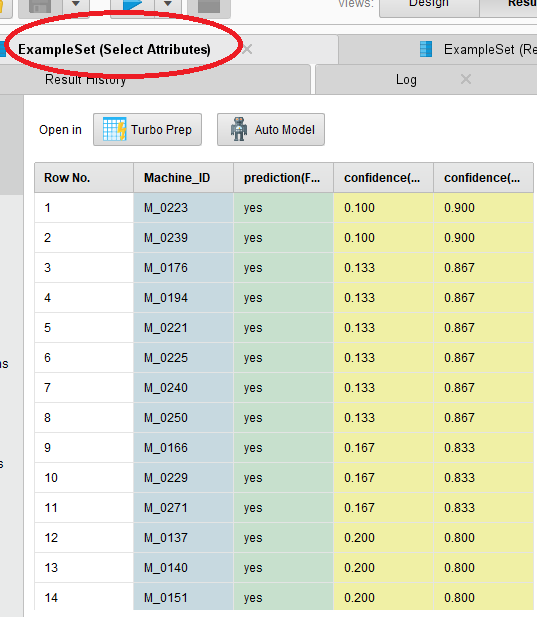
Ordenando a los datos con ‘**Sort’** por confiabilidad de forma decreciente



Y seleccionando todos los atributos utilizando ‘**Select Attributes’**:



Finalmente, la salida de las predicciones de fallas

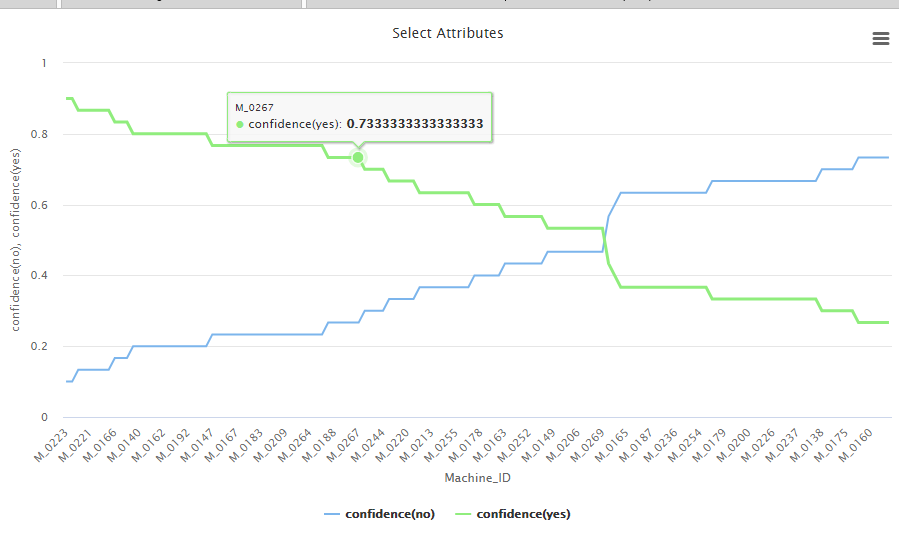


1. **Resultados estadísticos**



1. **Visualizaciones**

En este caso se determina que hay un cambio brusco cuando la confiabilidad está alrededor de 0.5.



Y que el conjunto de datos poseía más información sobre máquinas que efectivamente han fallado.

