

PRÁCTICA 8

Jesús S. Aguilar Ruiz
Catedrático de Universidad

REGISTRO



<https://www.drivendata.org/>

ALUMNO	USUARIO
AUNION DOMINGUEZ, CARLOS	UPOMD1901
BARCIELA RUEDA, DANIEL	UPOMD1902
FERNANDEZ LABRADOR, JOSE MANUEL	UPOMD1903
FIDALGO OLIVERES, FERNANDO	UPOMD1904
GANDUL PEREZ, MANUEL	UPOMD1905
GONZÁLEZ RUIZ, IRENE	UPOMD1906
HERRERA PULIDO, MANUEL	UPOMD1907
JUAREZ BOTE, ROBERTO	UPOMD1908
LAKIDAIN DE ARRIBA, ANDER	UPOMD1909
LÍPEZ-DAMAS OLIVERES, ARTURO (Erasmus)	UPOMD1910
LUZURIAGA RODRÍGUEZ, SERGIO	UPOMD1911
MACHADO GARCIA, ALBERTO	UPOMD1912
MUÑOZ ARENAS, CARLOS	UPOMD1913
PALOMINO GARCIA, ALEJANDRO	UPOMD1914
QUEVEDO RODRIGUEZ, ALBERTO	UPOMD1915
RODRIGUEZ RODRIGUEZ, JUAN ANTONIO	UPOMD1916
ROIZ PAGADOR, JOAQUIN	UPOMD1917
ROMERO FLORES, JESÚS SALVADOR	UPOMD1918
RUEDA MARIN, ANDRES	UPOMD1919
TERRERO LOPEZ, JOSE RAMON	UPOMD1920

Competition: **Pump it Up: Data Mining the Water Table**

Deadline: June 28, 2019, 11:59 p.m.

Examen (1ª Conv):	X 5 Junio	16:00
-------------------	-----------	-------

Examen (2ª Conv):	L 24 Junio	16:00
-------------------	------------	-------

FECHAS DE ENTREGA DEL PROYECTO

- Para presentarse a la 1ª Convocatoria:
 - 31 DE MAYO
- Para presentarse a la 2ª Convocatoria:
 - 19 DE JUNIO

ENTREGA DEL PROYECTO

- Debe enviar a AGUILAR@UPO.ES los siguientes archivos:
 - ARCHIVO .KNWF con modelo exportado de KNIME.
 - ARCHIVO .PDF con memoria técnica sobre el proyecto.
- Ambos archivos se denominarán con el nombre de usuario asignado. Por ejemplo:
 - UPOMD1901.KNWF
 - UPOMD1901.PDF
- El email debe indicar en ASUNTO: PROYECTO MD

FORMATO DE LA MEMORIA TÉCNICA

- Tipo de letra Arial 11pt;
- Márgenes 2.5cm;
- Interlineado 1.5;
- La primera página debe incluir únicamente el nombre completo y el DNI del autor.

ESTRUCTURA DEL INFORME TÉCNICO

- **Objetivo**

- Describa el objetivo que aborda el proyecto.

- **Datos**

- Describa las fuentes de datos usadas, internas y, si acaso, externas, mediante el formato de diccionario de datos.

- **Metodología**

- Describa, siguiendo el modelo CRISP-DM las fases que ha incluido en el proyecto, mostrando un diagrama realizado en KNIME.

- **Análisis**

- Para cada fase desarrollada en CRISP-DM, incluir un epígrafe que describa:
 - las técnicas o medidas usadas;
 - la justificación de su uso;
 - el diagrama en KNIME de esta fase (si existen metanodos, pueden omitirse sus desarrollos si es obvio lo que hacen y está descrito en el pie del metanodo). Es importante que las imágenes tengan calidad, es decir, que pueda leerse cualquier letra o número.

- **Discusión**

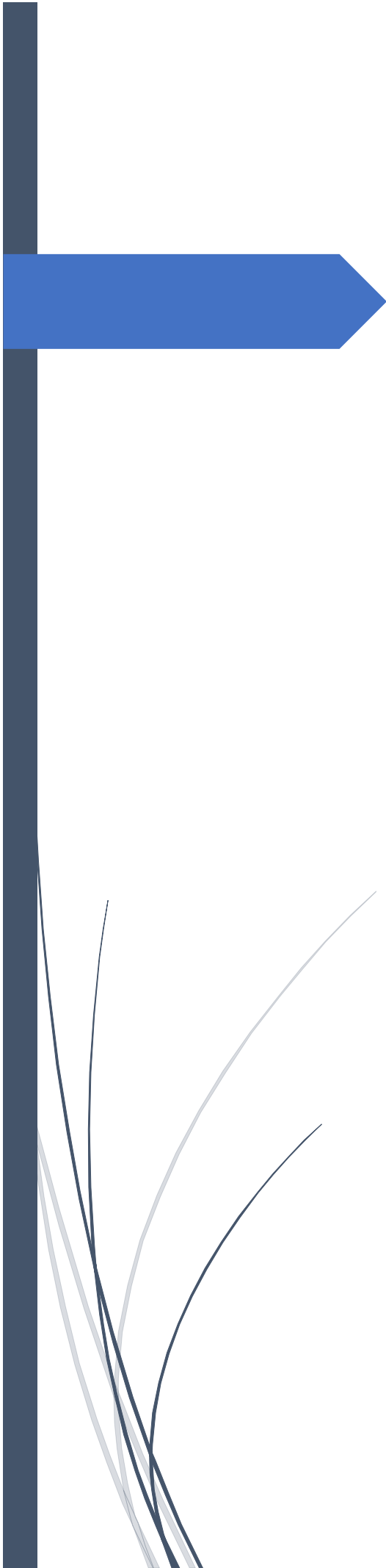
- Puede incluir un epígrafe que destaque el modelo final por decisiones que haya tomado durante el diseño, en cualquier fase, y que claramente hayan proporcionado mejores resultados frente a otras, con la justificación pertinente. (No se trata de novelar lo que no ha funcionado.)
- Los resultados obtenidos en función de los indicadores de calidad del modelo deben aparecer en este epígrafe (tanto en validación como test).

- **Conclusiones**

- Describa las conclusiones más relevantes del proyecto, desde un punto de vista científico, técnico o ingenieril.
- Describa, si existen, alternativas a su diseño que pudieren conducir a mejores resultados, con la justificación correspondiente.

- **Referencias**

- En caso de haber consultado material bibliográfico, documentación científica o docente, foros o, en general, información válida que haya contribuido a enriquecer sus conocimientos, inclúyala en este epígrafe.



José Manuel
Fernández
Labrador

31024029L

OBJETIVO:

El objetivo del proyecto es realizar un modelo de minería de datos que sea capaz de predecir qué bombas de agua son defectuosas, cuáles necesitan algunas reparaciones y cuáles funcionan. El país en el cual se realiza este estudio es la República de Tanzania.

Los datos para esta idea provienen del panel de control de puntos de agua de Taarifa, que agrega datos del Ministerio de Agua de Tanzania.

Tendremos que generar un archivo .csv con el identificador de fila de la bomba de agua, y su estado.

Para realizar dicho modelo, hemos utilizado el programa Knime Analytic Platform.

DATOS:

La fuente de datos usada es la que nos aporta el Ministerio de Agua de Tanzania, que nos aporta tres ficheros de datos .xlsx que usaremos en nuestro modelo:

- Train.xlsx: Nos proporciona un conjunto de información de los puntos de agua.
- Labels.xlsx: Nos proporciona el formato en base a:
 - Functional: El punto de agua está operativo y no se necesitan reparaciones.
 - functional needs repair: El punto de agua está operativo, pero necesita reparaciones.
 - non functional: El punto de agua no está operativo.
- Text.xlsx: Necesario para el ensamblado final del modelo y generación del .csv.

También nos aporta un fichero SubmissionFormat.xlsx que nos especifica un ejemplo del formato que debe tener nuestro fichero generado .csv.

A partir de estos datos, generamos nuestro propio Diccionario de datos, el cual usaremos para manejar toda la información del modelo.

FUENTE	TRAIN	NOMBRE VARIABLE IDENTIFICADA	SIMBOLO VARIABLE IDENTIFICADA	TIPO VARIABLE
ORIGEN	id	Identificador	ID	Number(Integer)
	amount_tsh	Cantidad_tsh	CA	String

date_recorded	Fecha	FE	Number(double)
funder	Financiador	FN	Number(double)
gps_height	GPS	GPS	String
installer	Instalador	IT	Number(Integer)
longitude	Longitud	LO	String
latitude	Latitud	LT	String
wpt_name	Nombre_wpt	NM	String
num_private	Numero privado	NP	Number(Integer)
basin	Cuenca	CU	Number(Integer)
subvillage	Subpoblación	SP	String
region	Region	RG	String
region_code	Codigo_Region	CR	Number(Integer)
district_code	Codigo_Distrito	CD	String
lga	LGA	LGA	String
ward	Sala	SL	String
population	Poblacion	PL	String
public_meeting	Reunion Publica	RP	String
recorded_by	Registrador	RG	Number(Integer)
scheme_management	Organización	OG	String
scheme_name	Nombre_Esquema	NE	String
permit	Permiso	PS	String
construction_year	Año_Construcion	AC	String
extraction_type	Tipo_Extracción	TE	String
extraction_type_group	Grupo_Tipo_Extraccion	GE	String
extraction_type_class	Clase_Tipo_Extraccion	CE	String
management	Administracion	AD	String
management_group	Grupo_Gestion	GG	String
payment	Pago	PG	String

payment_type	Tipo_Pago	TP	String
water_quality	Calidad_Agua	CA	String
quality_group	Calidad_Grupo	CG	String
quantity	Cantidad	CA	String
quantity_group	Cantidad_Grupo	CG	String
source	Fuente	FU	String
source_type	Tipo_Fuente	TF	String
source_class	Tipo_Clase	TC	String
waterpoint_type	Tipo_PuntoAgua	TPA	String
waterpoint_type_group	Tipo_Grupo_PuntoAgua	TGPA	String
status_group	Estado_Grupo	EG	String

METODOLOGÍA:

Para realizar el modelo del proyecto hemos seguido la metodología CRISP-DM en la que se incluyen las siguientes fases:

Comprensión del negocio:

En esta fase la hemos dedicado a la definición de los objetivos del proyecto y determinar su estructura. Lo primero fue comprender que teníamos que generar un modelo capaz de predecir si las bombas de agua podrían ser funcionales o no, y también si se pueden reparar. Una vez entendido esto, teníamos que estructurar nuestro modelo en siguiendo las fases del CRISP-DM, para adaptarlo a los requisitos del problema. La distribución final tenía que generar un fichero con un formato preestablecido.

Comprensión de los datos:

- Recopilación de datos iniciales: En nuestro caso, el Ministerio de Agua Tanzania aporta los datos con los cuáles tenemos que trabajar.
- Descripción de los datos: En su página web, teníamos una descripción del significado de cada uno de los datos que nos aporta, sirviendo de gran ayuda para elaborar nuestro diccionario de datos propio para nuestro modelo.

- Exploración de los datos: Al realizar el diccionario de datos, hemos tenido que explorar éstos, para analizar errores, valores faltantes, etc, que puedan contener y que debemos de controlar en la siguiente fase de preparación de los datos.

Preparación de los datos:

- Inclusión de datos: Utilizando los datos anteriormente descritos, hemos insertado la cabecera de nuestro propio diccionario de datos.
- Limpieza de datos: Primero hemos sustituido los outliers por missing, y a continuación hemos reemplazado esos valores faltantes empleando técnicas de imputación simple.
- Clasificación de datos: Finalmente clasificamos el conjunto de datos de entrenamiento numéricos utilizando el algoritmo k vecino más cercano.

Modelado:

- Estimar precisión del modelo: Para ello utilizamos la técnica de validación cruzada, la cual nos ayuda a estimar la precisión de nuestro modelo en este punto.
- Predecir valor de la variable de destino en función de nuestras variables de entrada: Utilizamos árboles de decisión para este fin.
- Identificar de forma automática agrupaciones de elementos: Para ello usamos técnicas de clustering, que nos permite esta agrupación de acuerdo a una medida de similitud entre ellos.

Evaluación:

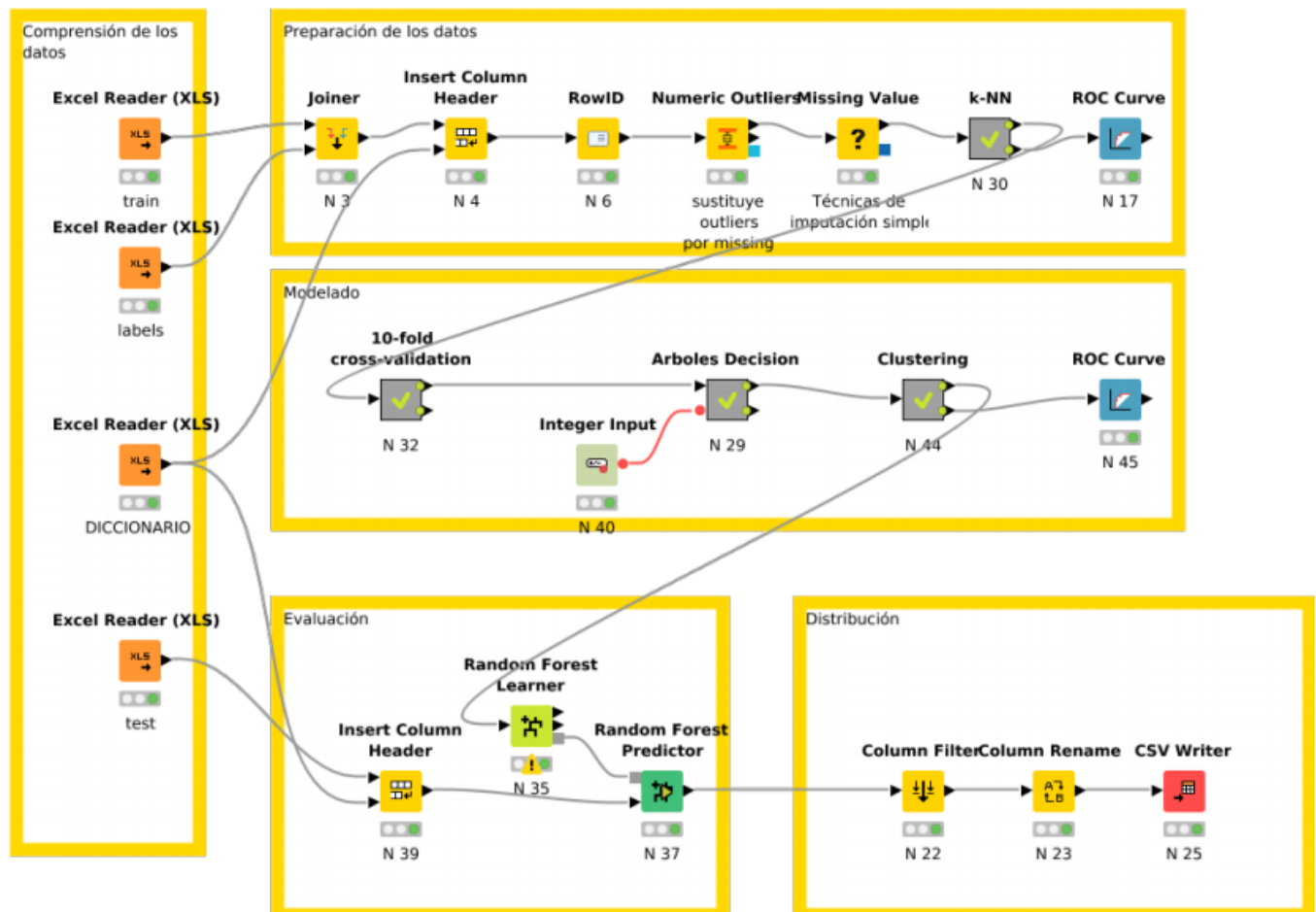
En esta fase nos dedicamos a evaluar los datos obtenidos del modelado de las anteriores fases con los datos originales. Para esta evaluación, utilizamos aprendizaje aleatorio usando algoritmos de árboles. Finalmente un predictor es el que nos genera la salida final de nuestros datos.

Distribución:

- Planificación de la distribución: Tenemos que planificar y ordenar todas las fases de nuestro modelo, para que sea fácil el mantenimiento del mismo.
- Mantenimiento: Por supuesto, no se trata de un modelo final e inmodificable. Este modelo está preparado para realizar las modificaciones y/o mejoras futuras.

- Informe final: Este modelo genera una salida final la cual podrá ser evaluada en la plataforma la cual aporta los datos y destina este modelo. También este documento forma parte de este informe final.

Finalmente adjunto una imagen del modelo entero realizado:



ANÁLISIS:

Comprensión del negocio:

Esta fase no tiene diagrama en sí en Knime, su cometido lo hemos abordado en el apartado anterior.

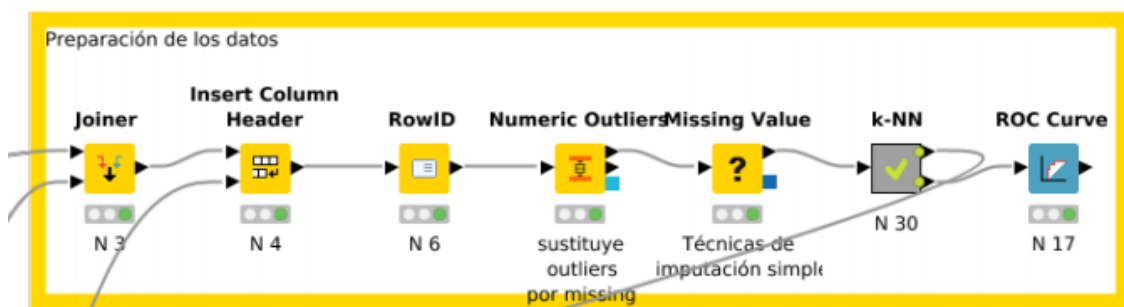
Comprensión de los datos:



En esta fase nos preocupamos de realizar un análisis de los datos que nos ofrecían para el modelo, de forma que construimos nuestro diccionario de datos.

Utilizamos 4 nodos Excel Reader (XLS) que realizan la función de cargar nuestros datos que se encuentran en formato .xlsx. Los ficheros train, labels y test son los que nos ofrecen la plataforma, mientras que PROYECTO_DIC trata de nuestro diccionario empleado para el modelo.

Preparación de los datos:



Joiner: une las tablas de train y labels.

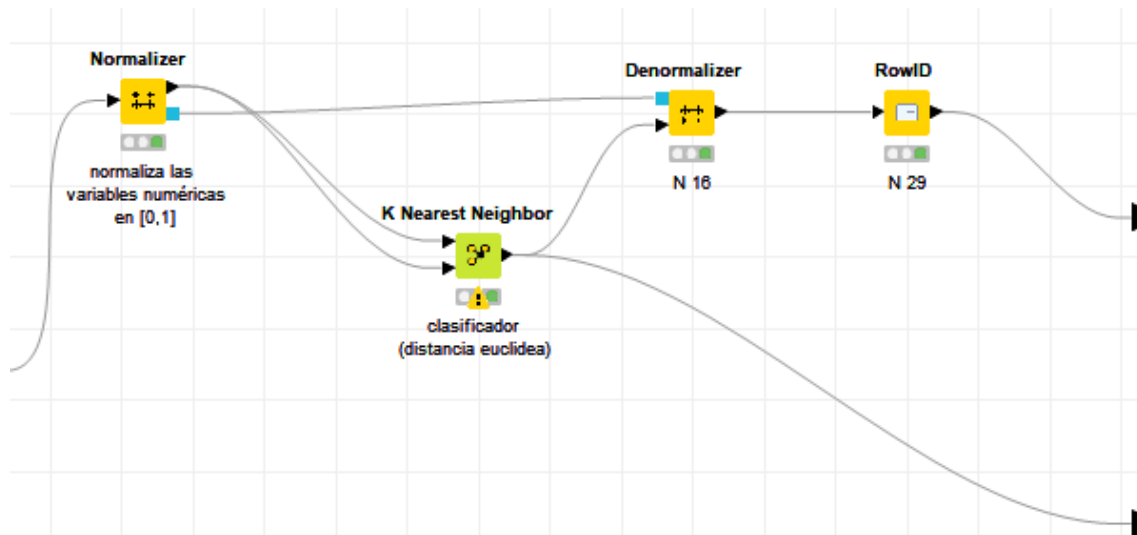
Insert Column Header: a la salida de la unión de las dos tablas anteriores, le añadimos la cabecera del diccionario de datos para trabajar con él.

RowID: tan solo volvemos a enumerar las filas de la tabla de datos.

Numeric Outliers: sustituye los outliers que superan 1.5 veces el rango intercuartílico por encima de Q3 y por debajo de Q1 por valores ausentes (missing).

Missing Value: aplicamos técnicas de imputación simple sobre los valores ausentes (missing). Esto es, aplicar la mediana a las variables numéricas, y la más frecuente a las variables de String.

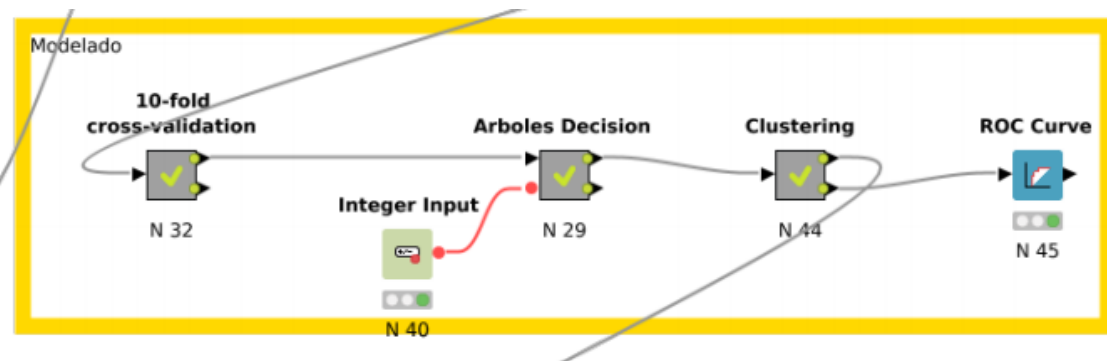
k-NN: metanodo que clasificamos el conjunto de datos de entrenamiento numéricos utilizando el algoritmo k vecino más cercano. Genera dos salidas, una la de datos clasificados por k-NN, y otra para analizar la curva ROC:



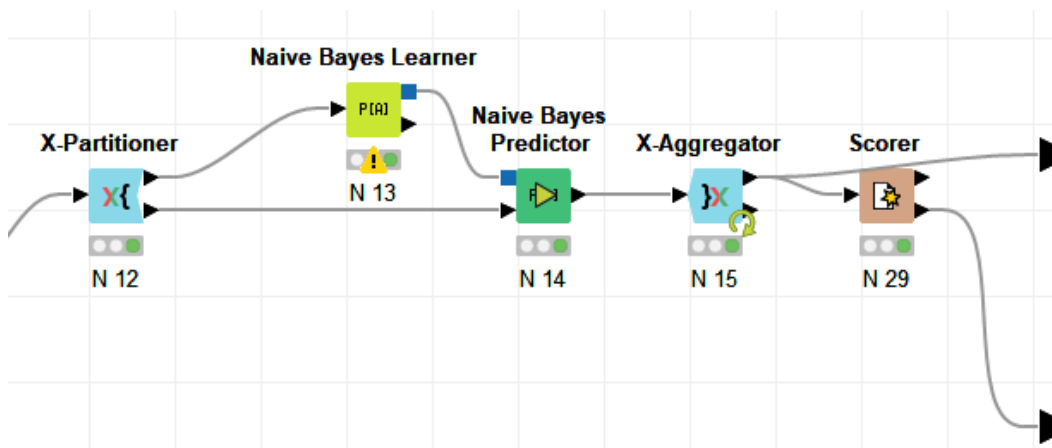
- **Normalizer:** normaliza las variables numéricas en el intervalo [0,1] para el correcto funcionamiento de K-NN.
- **K Nearest Neighbor:** se trata del clasificador del algoritmo del vecino más cercano. Utiliza los datos de entrenamiento numéricos, y calcula la distancia euclídea entre ellos con respecto a la variable Estado_Grupo. Su valor de vecinos considerados está asignado a 11.
- **Denormalizer:** devuelve los valores originales a las variables numéricas normalizadas.
- **RowID:** volvemos a enumerar las filas de la tabla de datos.

ROC Curve: evalúa la curva ROC tras utilizar el clasificador K-NN. Lo realiza a la variable Estado_Grupo y la clase positiva valorada es “functional”.

Modelado:



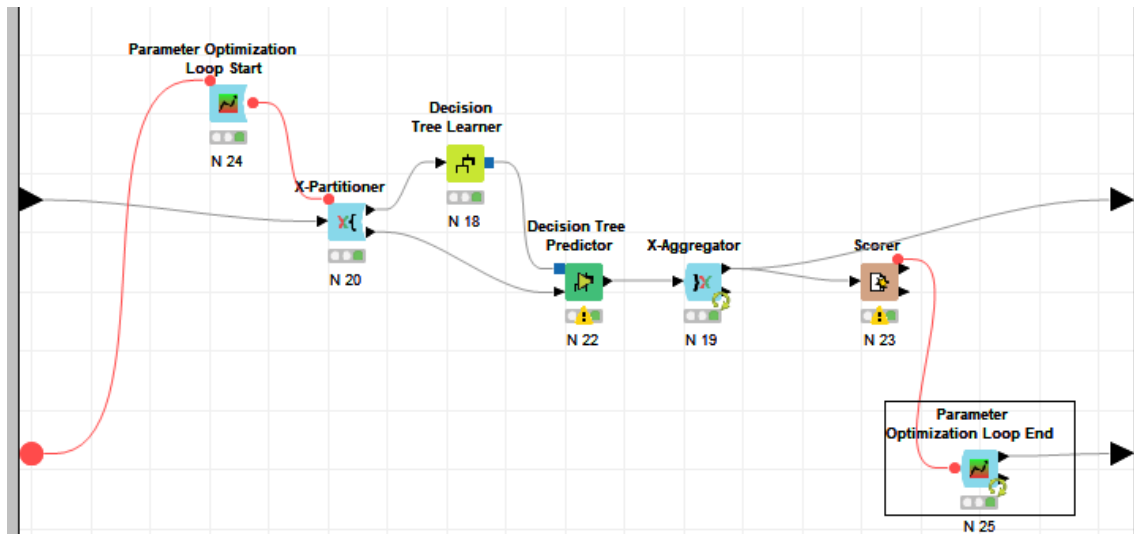
10-fold cross-validacion: metanodo que usa la técnica de validación cruzada que nos ayuda a estimar la precisión del modelo. Genera dos salidas, una nuestra tabla de datos añadiendo una columna con la predicción y otra de exactitud de estadísticas:



- **X-Partitioner:** realiza el inicio de la validación cruzada, con un número de 10 validaciones y sobre la columna Estado_Grupo.
- **Naive Bayes Learner:** predice la probabilidad de de posibles resultados de Estado_Grupo.
- **Naive Bayes Predictor:** realiza la predicción anterior en una nueva columna Prediction(Estado_Grupo).
- **X-Aggregator:** fin del ciclo de validación cruzada.
- **Scorer:** genera la matriz de confusión entre Estado_Grupo y su Prediction(Estado_Grupo).

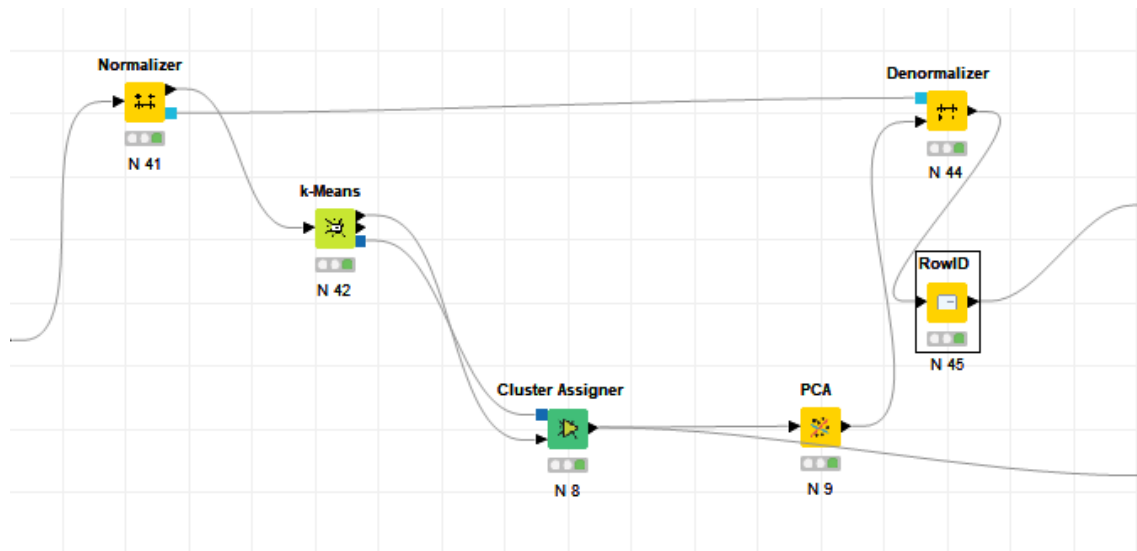
Integer Input: produce un valor entero asignado a 5 en forma de variable de flujo para el metanodo de los árboles de decisión.

Árboles Decisión: metanodo que predice el valor de las variable de salida en función de nuestras variables de entrada:



- **Parameter Optimization Loop Start:** inicia un bucle de optimización de parámetros. Variará estos parámetros, siguiendo la estrategia de búsqueda por fuerza bruta. Introduce la variable de flujo MIN_EXAMPLE entre 6 y 12 a X-Partitioner.
- **X-Partitioner:** realiza el inicio de la validación cruzada, con un número de 3 validaciones y sobre la columna Estado_Grupo teniendo en cuenta la variable de flujo MIN_EXAMPLE.
- **Decision Tree Learner:** utiliza la técnicas de los árboles de decisión para predecir la variable de salida en función de las variables de entradas. Esto nos permite generar unos valores en el modelo que nos ayuda a predecir las bombas de agua. Actúa sobre Estado_Grupo.
- **Decision Tree Predictor:** realiza la predicción anterior y genera una nueva columna llamada Prediction(Estado_Grupo).
- **X-Aggregator:** fin del ciclo de validación cruzada.
- **Scorer:** genera la matriz de confusión entre Estado_Grupo y su Prediction(Estado_Grupo)#1.
- **Parameter Optimization Loop End:** finaliza el bucle de optimización de parámetros.

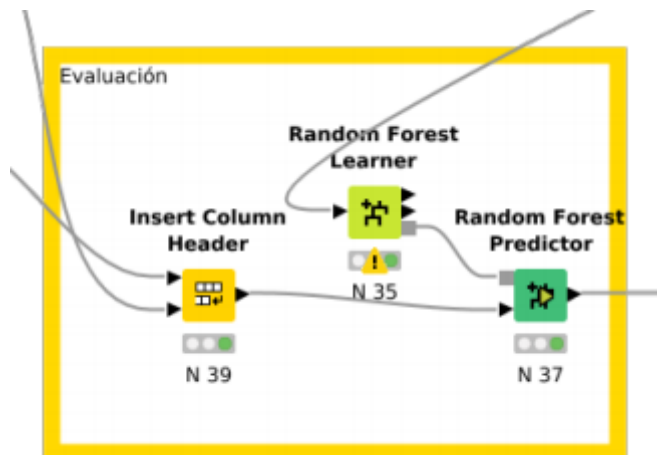
Clustering: metanodo que identifica de forma automática agrupaciones entre elementos mediante una medida de similitud entre ellos. Tiene dos salidas, una para los datos de nuestro modelo añadiendo las columnas con los datos agrupados, y otra para analizar la curva ROC:



- **Normalizer:** normaliza las variables numéricas en el intervalo [0,1] para el correcto funcionamiento de k-Means.
- **K-Means:** realiza un agrupamiento nítido que asigna un vector de datos a exactamente un clúster. El algoritmo termina cuando las asignaciones de clúster ya no cambian. El algoritmo de agrupamiento utiliza la distancia euclidiana en los atributos seleccionados. Le hemos asignado un número de cluster igual a 3 y un máximo de 99 iteraciones.
- **Cluster Assigner:** asigna nuevos datos al agrupamiento realizado con k-Means. Cada punto de datos se asigna a su prototipo más cercano.
- **PCA:** realiza un análisis de los componentes principales de los datos que recibe con la mínima pérdida de información.
- **Denormalizer:** devuelve los valores originales a las variables numéricas normalizadas.
- **RowID:** volvemos a enumerar las filas de la tabla de datos.

ROC Curve: evalúa la curva ROC tras utilizar el clasificador K-NN. Lo realiza a la variable Estado_Grupo y la clase positiva valorada es “functional”.

Evaluación:

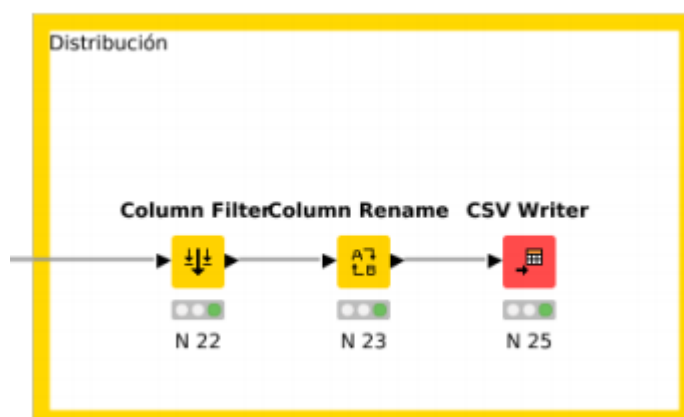


Insert Column Header: le añadimos la cabecera del diccionario de datos a la tabla de datos que resulta de cargar el dataset test.xlsx.

Random Forest Learner: evaluamos la tabla resultante del modelado y nos quedamos solo con las columnas que también teníamos en nuestro dataset de datos original. Funciona mediante un conjunto de árboles de decisión aleatorios (Random Forest). Lo realizamos con respecto a la columna Estado_Grupo.

Random Forest Predictor: realiza la predicción en un pequeño subconjunto tras recibir los datos seleccionados del modelo a través del Random Forest Learner y el dataset de datos originales. Genera una salida que será el modelo final.

Distribución:


















Column Filter: filtramos el modelo que recibimos de forma que solo nos quedamos con el identificador y la predicción del estado del grupo puesto que sería el resultado de si una bomba está defectuosa, necesita reparación, o está correcta.

Column Rename: renombramos las columnas anteriores a “id” y “status_group” respectivamente para mantener el formato que necesitamos en la salida para evaluar el modelo en la plataforma DrivenData. Tenemos que convertir a int la primera y en String la segunda.

CSV Writer: genera una salida en formato CSV llamada salidaModelo.csv que será el archivo en el formato correcto el cual tendrá el resultado final.

DISCUSIÓN:

Este modelo se ha ido comprobando a medida de su elaboración, mediante una evaluación de calidad que nos proporciona la plataforma DrivenData. En la siguiente captura adjunto la valoración obtenida en las subidas de los últimos días, y como ha ido evolucionando y/o variando el modelo en consecuencia:

0.6136	UPOMD1903 	2019-04-25 14:45:25 UTC
!	UPOMD1903 	2019-05-27 17:50:51 UTC
0.8041	UPOMD1903 	2019-05-27 17:53:07 UTC
0.6133	UPOMD1903 	2019-05-27 18:12:05 UTC
0.4646	UPOMD1903 	2019-05-27 18:31:21 UTC
0.8046	UPOMD1903 	2019-05-28 09:52:56 UTC
0.8046	UPOMD1903 	2019-05-28 10:13:15 UTC
0.8046	UPOMD1903 	2019-05-28 10:21:25 UTC
0.7846	UPOMD1903 	2019-05-29 07:18:32 UTC
0.7846	UPOMD1903 	2019-05-29 07:52:33 UTC
0.7846	UPOMD1903 	2019-05-29 07:53:57 UTC
0.8046	UPOMD1903 	2019-05-30 07:40:37 UTC
0.8046	UPOMD1903 	2019-05-30 10:05:24 UTC
0.8046	UPOMD1903 	2019-05-30 10:06:27 UTC
0.8046	UPOMD1903 	2019-05-31 09:04:55 UTC

Finalmente mostramos en la siguiente captura, la valoración final y la posición final (ésta puede variar hasta el cierre de la competición) a escasas semanas de finalización de la competición:

Submissions

BEST	CURRENT RANK	# COMPETITORS
0.8046	1221	7231

CONCLUSIONES:

Aunque la finalidad de los datos y datasets aportados por la plataforma y dicho modelo están destinados a una competición en la cual consiguen generar una predicción, el objetivo de este proyecto no ha sido más que una forma de aprender y adentrarnos en el mundo de la minería de datos.

Este proyecto ha aportado conocimientos de análisis de datos. Conocer los datos que tenemos, y extraer información de ellos con las distintas herramientas que nos aportaba la presente asignatura. También la utilización de técnicas de preprocesado de datos, aprendizaje y validación de modelos.

Podríamos decir, que hemos generado un pequeño modelo con una parte predictiva, pero también enfocándonos en otra parte más descriptiva.

Los conocimientos de estadística también han sido necesarios para elaborar este modelo.

Por supuesto, considero alternativas a este modelo que puedan mejorarlo, pero me remito a lo comentado en los párrafos anteriores, este proyecto ha servido para introducción a la minería de datos, la idea era realizar el mejor modelo posible con las herramientas que disponíamos.

REFERENCIAS:

El principal material ha sido sacado de la asignatura de Minería de Datos impartida en la Universidad Pablo de Olavide, en las clases presenciales impartidas por el docente Jesús S. Aguilar-Ruiz.

Otras referencias que han servido para investigar:

<https://www.drivendata.org/>

<https://forum.knime.com/c/knime-analytics-platform>

<https://www.knime.com/forum?destination=forum>

<http://mavir.net/docs/jlb-MineriaDatos.pdf>

<https://data.sngular.com/es/art/25/crisp-dm-la-metodologia-para-poner-orden-en-los-proyectos-de-data-science>

https://es.wikipedia.org/wiki/Cross_Industry_Standard_Process_for_Data_Mining