Deusto
Universidad de Deusto
Deustuko Unibertsitatea
University of Deusto

ANÁLISIS Y OPTIMIZACIÓN DE MÁQUINA DE IMPRESIÓN 3D

Análisis Avanzado de datos para la Industria

Índices

0.1 Índice de contenidos

1. Antecedentes

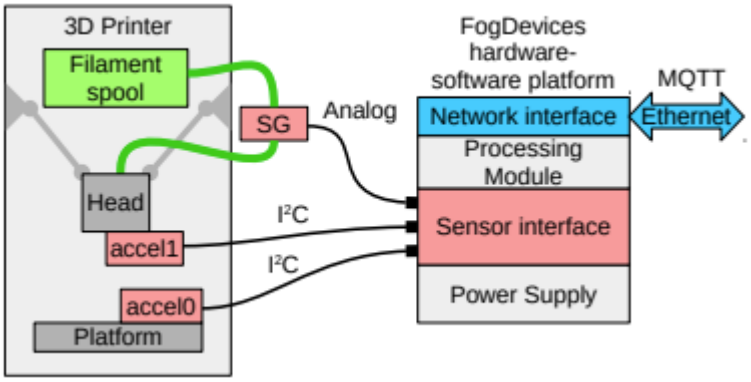
1.1 Proyecto de partida

Este proyecto parte del proyecto realizado por: Joanna Sendorek, Tomasz Szydlo, Mateusz Windak y Robert Brzoza-Woch sobre “Dataset for anomalies detection in 3D printing”. Agradecer por parte de los autores de esta memoria, la publicación de los datos del proyecto de partida.

[1] [2] [3] [4] [5]

Se parte de una máquina de fabricación aditiva, esta recibe un mismo diseño y un mismo material de entrada, la máquina transforma el material en el diseño. Durante el proceso, la máquina produce una serie de datos, además gracias al proyecto del que se parte se cuenta con más datos provenientes de la instalación de acelerómetros y un sensor de tensión en la máquina de forma externa.

En la siguiente imagen se puede ver el sistema IoT utilizado en el proyecto de partida para conseguir estos datos extra:



1.2 Explicación de la máquina y sus procesos.

El tipo específico de impresora 3D del que se extraen los datos utiliza un brazo robótico para realizar el proceso de impresión en lugar de los métodos más comunes, como la deposición de material fundido (FDM).

En lugar de tener una plataforma de impresión estacionaria, esta máquina cuenta con un brazo robótico articulado que se mueve en múltiples ejes para depositar el material de impresión capa por capa. El brazo robótico suele estar equipado con un extrusor o una boquilla especializada que permite la deposición precisa del material plástico.

La secuencia de impresión en una máquina de impresión 3D de plástico con brazo es similar a otros métodos de impresión 3D. Comienza con la preparación del modelo 3D en un software de diseño, donde se define la geometría del objeto que se va a imprimir. Luego, el software de impresión divide el modelo en capas y genera un camino de herramienta o ruta para que el brazo robótico siga durante la impresión.

Durante la impresión, el brazo robótico se mueve según la ruta predefinida y deposita el material plástico capa por capa para construir el objeto en 3D. Dependiendo del diseño y las características del brazo robótico, es posible lograr una mayor libertad de movimiento y mayor precisión en comparación con las impresoras 3D convencionales.

2. Objetivo de negocio

2.1 El problema

Se necesita conocer de antemano si el resultado final de la producción va a resultar correcto o no. Esta problemática reside en los largos tiempos de producción que tienen estas máquinas. En muchas ocasiones se compran varias máquinas para realizar el mismo diseño y producir piezas a mayor rapidez, pero esto requiere mantener más máquinas en funcionamiento.

2.2 Objetivo del sistema

Predecir la calidad de la pieza producida, para conocer de antemano si la máquina está realizando bien el proceso de impresión, teniendo en cuenta el histórico de procesos realizados con el mismo diseño y el proceso que se está realizando. Por otro lado, para los casos en los que la calidad prediga una pieza defectuosa, describir el motivo por el que ocurre, para así poder optimizar la máquina y adaptarla a ese diseño (mejora continua).

2.3 El sistema

El sistema se compone de:

- Subsistema predictivo
- Subsistema descriptivo

Con el mismo origen de datos, se trabajará por un lado el proceso de predicción de calidad y en el caso de que la calidad predecida sea defectuosa, describir el origen del fallo. Por lo que ambos sistemas se entrenan con los mismos datos.

2.4 Alcance

El proyecto incluye detectar los factores críticos a la hora de determinar el éxito de un diseño y ser capaz de calcular la probabilidad de éxito de cada elemento posible para la producción de un producto. Delimitar unos rangos de probabilidad de éxito en los cuales sea seguro trabajar, y otros en los que sea conveniente que no. En los casos en los que el producto haya salido defectuoso, se realizará una descripción del fallo ocurrido y cuáles son sus causas. Y por último, el software se realimentará constantemente con los nuevos resultados, para poder aumentar su precisión y detalle para cada nuevo diseño a trabajar.

2.5 Requisitos

El cliente pone los siguientes requisitos al sistema :

PRODUCT BACKLOG: Requisitos		
Autor:	Asier Vega y Gorka Berganza	
Proyecto:	Sistema predictivo - descriptivo	

ID	Historia de usuario	Ámbito
1	**Cómo** responsable de compras **quiero** conocer de antemano la calidad de la pieza de salida de esta máquina **para** decidir si comprar o no el material de entrada.	Desarrollo
2	**Cómo** responsable de producción **quiero** un proceso de mejora continua **para** que esta máquina se adapte mejor a nuevos materiales y a los actuales.	Desarrollo
3	**Cómo** responsable de producción **quiero** conocer los problemas que mi máquina da **para** corregirlos.	Desarrollo
4	**Cómo** responsable de informática **quiero** que el sistema se ejecute en la nube (Azure) y el envío de datos se realice de forma segura para ahorrar recursos de nuestro cpd.	Diseño
5	**Cómo** responsable de calidad **quiero** que el programa me realice una descripción del problema ocurrido y sus motivos **para** poder corregir mi máquina.	Desarrollo
6	**Cómo** cliente de este proyecto **quiero** que la precisión **para** la predicción sea del 70%.	Diseño
7	**Cómo** cliente de este proyecto **quiero** que la precisión **para** la descripción sea del 60%.	Diseño
8	**Cómo** responsable de la configuración de la máquina **quiero** que los resultados tengan el mismo formato de nuestra empresa **para** poder entenderlos fácilmente.	Usuarios

3. Desarrollo

3.1 Objetivos del desarrollo

3.1.1 Objetivo de negocio

Predecir la calidad del producto en procesamiento y describir el proceso con el objetivo de optimizar la máquina.

3.1.2 Objetivo de minería

Subsistema predictivo supervisado: predecir la calidad final de la pieza de antemano. Esto es el objeto de minería es el campo “ok” en los datos, en el que se especifica que ese registro ha resultado en defecto (0) o se ha producido de forma correcta (1).

Subsistema descriptivo supervisado: utilizar los valores de la máquina y los posibles defectos que esta puede tener, para describir el proceso que ha ocurrido. Esto es el objetivo de minería el campo “quality” en los datos, en el que se especifica que ese registro ha resultado en uno de estos defectos: “arm_failure”, “bowden”, “plastic”, “retraction” y “unstick” o en un proceso exitoso “proper”.

3.1.3 Objetivo de modelado

Modelo predictivo: Para cada diseño que la máquina procese, predecir si el proceso actual resultará en defecto o no. Con el objetivo de crear un sistema de alarma predictivo que avise si la máquina está generando una pieza que resultará defectuosa. El modelo más adecuado para esta solución se decide en la fase de modelado, siendo este el que más precisión de.

Modelo descriptivo: Para cada diseño que la máquina procese, una descripción gráfica de las variables más relevantes que intervienen en el proceso. El modelo adecuado para lograr esto es un árbol de decisión, ya que el resultado del modelo se les hará llegar a las personas encargadas de configurar la máquina.

3.1.4 Objetivo de calidad

Proyecto predictivo: la precisión solicitada de este sistema es de 80%. Al tratarse de un sistema de alarma se realizan dos predicciones: si la primera predice defecto, no se produce un aviso pero la alarma queda en pendiente, al producir la segunda predicción un tiempo más tarde, si ésta predice defecto se avisa para parar la producción, si no la alarma deja de estar en pendiente.

Proyecto descriptivo: la precisión solicitada de este sistema es de 70%. Se debe producir un árbol y unas reglas que puedan ser entendibles por un ser humano de manera sencilla.

3.2 Características de los datos

3.2.1 Características de los datos suministrados

Los datos suministrados parten de 6 procesos que la misma máquina de impresión 3D realiza para producir un mismo diseño.

Se cuenta con dos grupos de datos:

- Los propios de la máquina de impresión:
    1. 6 datasets (.csv) de diferentes tamaños, cada dataset cuenta con los datos de un único proceso. Por lo que en total son 5 procesos que terminan siendo defectuosos y uno que no.
    2. En total son 31 atributos, incluido índice, timestamp y fecha.
    3. El tamaño de cada dataset es de entre 200.000 y 500.000 registros, aproximadamente 80 registros cada segundo
- Los extraídos mediante el proyecto de partida.
    1. 6 datasets (.csv) de diferentes tamaños, cada dataset cuenta con los datos de un único proceso. Por lo que en total son 5 procesos que terminan siendo defectuosos y uno que no.
    2. En total 11 atributos, incluido índice, timestamp y fecha
    3. El tamaño de cada dataset es de 1.500.000 registros, aproximadamente 230 registros por segundo.

Al tratarse de datos separados es necesario desarrollar un proceso que los una.

3.2.2 Granularidad y cadencia

Se establece como unidad más pequeña de comparación: una muestra de cada **15 minutos** de proceso por cada **calidad**, con las medias para los valores numéricos y el sumatorio para los valores no numéricos. El cliente solicita que se evalúe la predicción cada 15 minutos, de aquí esta decisión.

3.2.3 Volumetría

Se orienta este proyecto a un proyecto de Small data por los siguientes motivos:

- Histórico de datos limitados, solo se cuenta con los datos de un diseño para 6 procesos distintos.
- Datos de calidad, se cuenta con hasta 36 atributos de posible peso para el sistema.

Pese a que de forma inicial no se cuenta con muchos datos, el proceso desarrollado es modular y aplicable para cualquier nuevo diseño.

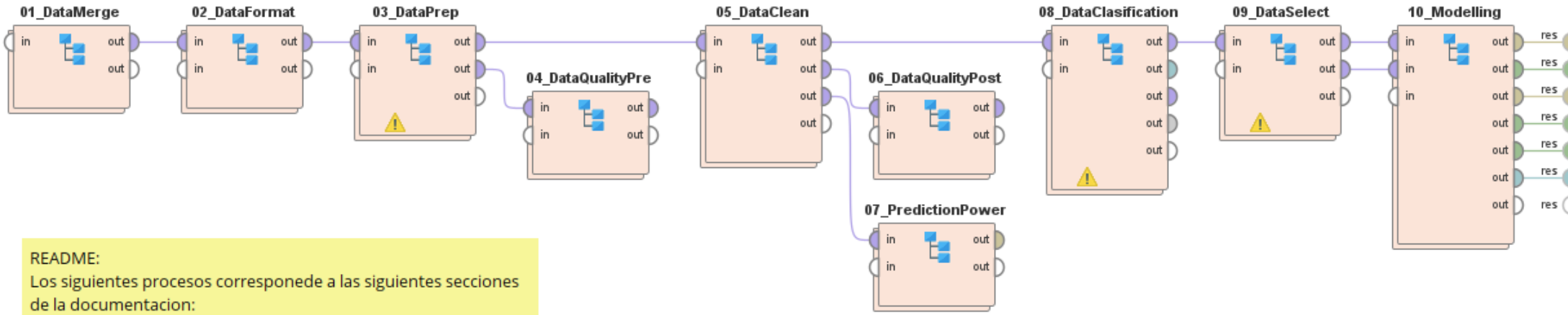
3.2.4 Inventario de datos

12 dataset con un peso total de 1,19 GB, con un formato común pero sin haberse realizado un proceso de limpieza con anterioridad.

Asier Vega y Gorka Berganza
31 de mayo de 2022

3.3 Análisis y modelado

El proceso creado en Rapidminer que se explica a continuación se divide en los siguientes subprocesos:



README:
Los siguientes procesos corresponede a las siguientes secciones de la documentacion:

- 01_DataMerge = 3.3.1 Unión de los datos
- 02_DataFormat = 3.3.2 Formato de los datos
- 03_DataPrep = 3.3.3 Preparación de los datos
- 04_DataQualityPre = 3.3.4 Calidad del dato inicial
- 05_DataClean = 3.3.5 Limpieza de los datos
- 06_DataQualityPost = 3.3.6 Calidad del dato final
- 07_PredictionPower = 3.3.7 Potencia predictiva
- 08_DataClasification = 3.3.8 Preparación del objetivo minería
- 09_DataSelect = 3.3.9 Preparación de los atributos
- 10_Modelling = 3.3.10 Modelado

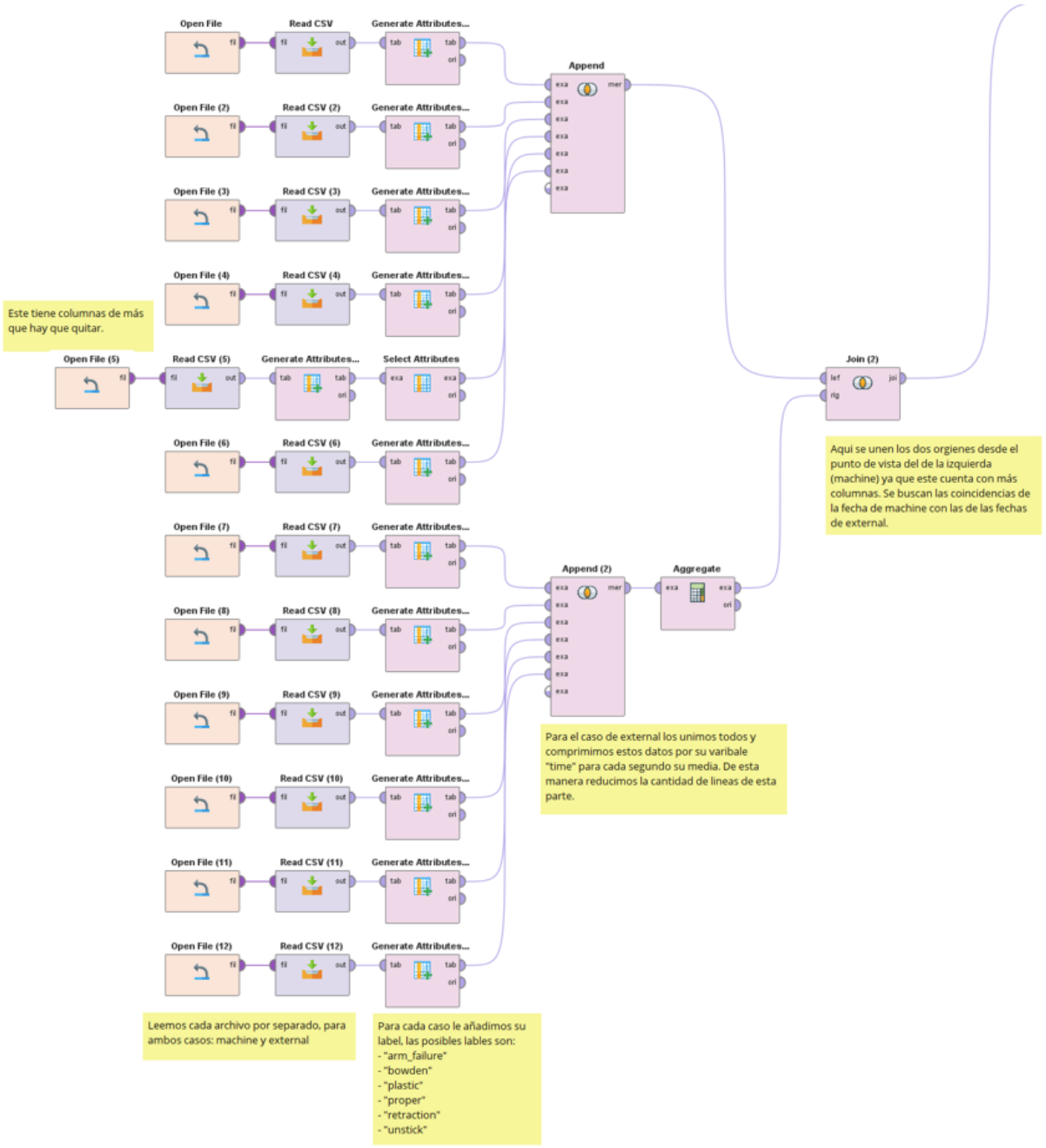
3.3.1 Unión de los datos

Al partir de 12 datasets separados es necesario unirlos todos de forma inicial. 6 de ellos pertenecen a la máquina y los otros 6 pertenecen a los datos extraídos del proyecto de partida, ambos grupos representan los mismos 6 procesos.

Para unir todos los datos primero, se genera un atributo “quality” en el que se introduce la label correspondiente a cada uno de los 6 dataset de cada grupo (labels posibles: arm_failure”, “bowden”, “plastic”, “retraction”, “unstick” y “proper”). Después se unen los grupos a través del operador “Append” generando dos datasets independientes, uno para cada origen.

Para unir ambos datasets, uno con 2.000.000 de registro y otro con 6.000.000 de registro. Se reducen los 6.000.000 (los datos del proyecto de partida unidos) a través de calcular la media para cada segundo. El dataset resultante se une a los 2.000.000 de registros (datos de la máquina) a través del operador “Join” fijando como atributo clave el tiempo.

De esta manera se consiguen 2.000.000 de registros con 35 atributos útiles en un solo dataset, siendo 7 de estos las medias extraídas de la reducción de los otros datasets.



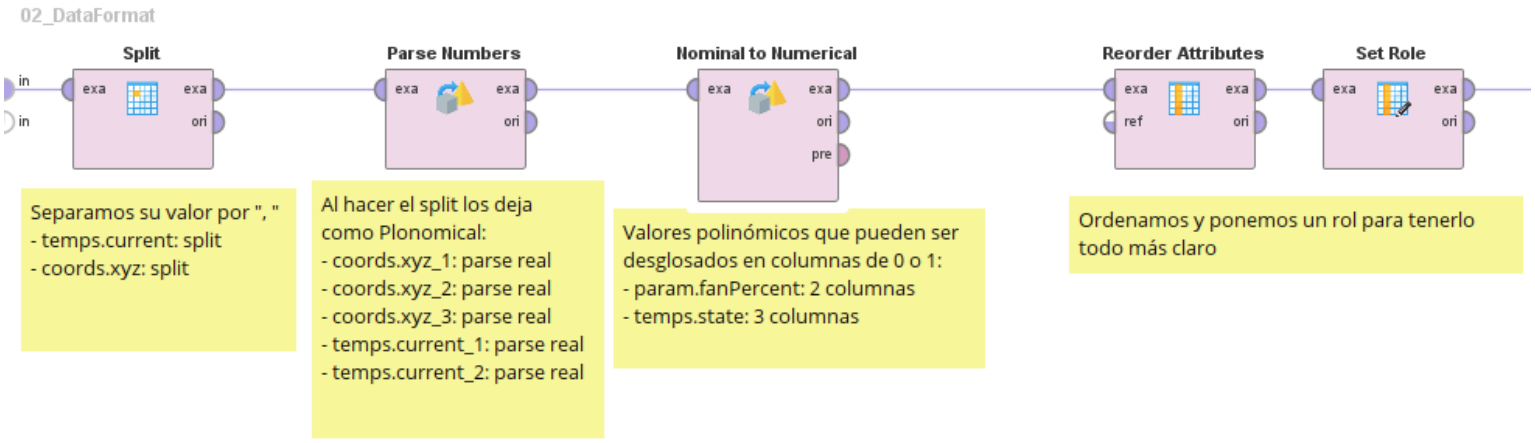
3.3.2 Formato de los datos

Por problemas en la lectura que realiza Rapidminer con el operador “Read CSV” es necesario formatear los datos de la siguiente manera:

Los atributos “temps.current” y “coords.xyz” no han sido exportados bien, ya que su valor lo conforman valores distintos (Ej: cords.xyz = “3, 2, 1”). Por lo tanto se separan y se convierte su tipo de dato en real.

Los tributos “param.fanPercent” y “temps.state” contienen demasiados valores agrupados por lo que hacer lo mismo que antes genera un dataset demasiado grande. Por lo que se hace “dummy coding” con ellos logrando un dataset más reducido.

Por último se reordenan los atributos de forma lógica y se establece como label el atributo generado en la sección “3.3.1 Unión de los datos” “quality”.

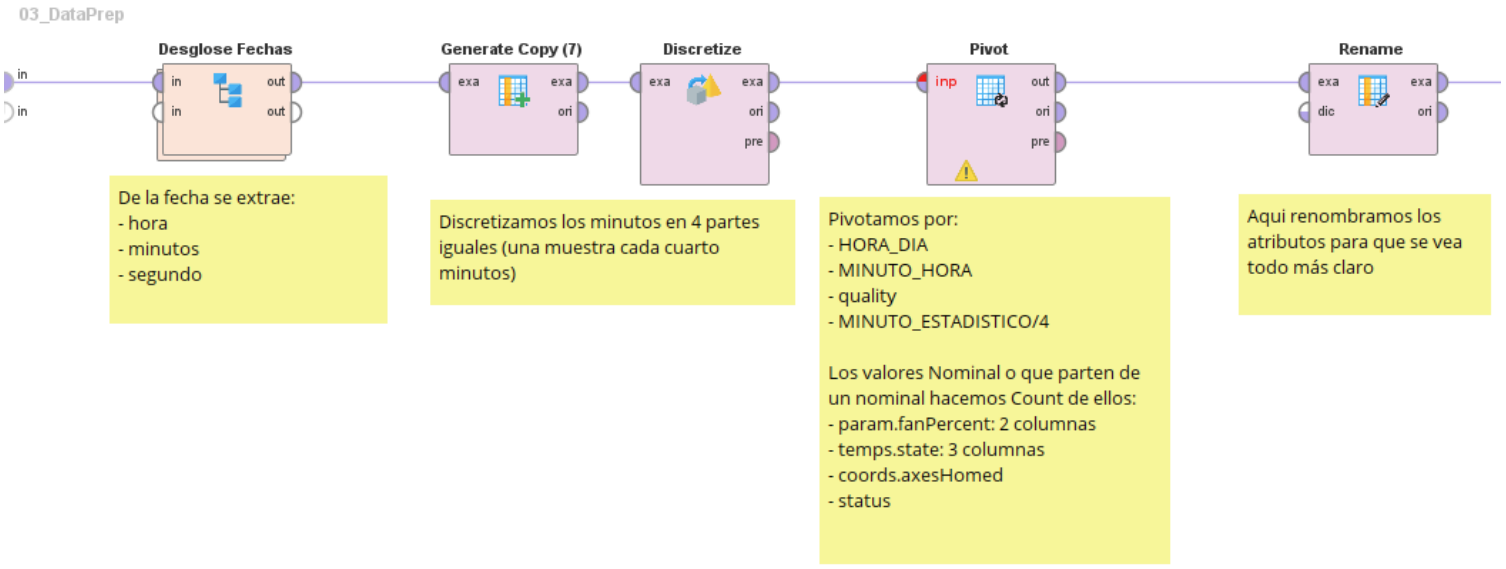


3.3.3 Preparación de los datos

Con el objetivo de preparar los datos en crudo para usarlos en sistemas de inteligencia artificial se desarrollan los siguientes procesos:

Se desglosa el campo “time2” para obtener 3 nuevos atributos con la hora, minuto y segundo. No se desglosa en cantidades más grandes debido a que no se poseen datos de más de un día. A continuación se discretizan los minutos en 4 partes iguales, obteniendo así el cuarto de hora estadísticos a través del operador “Discretize”.

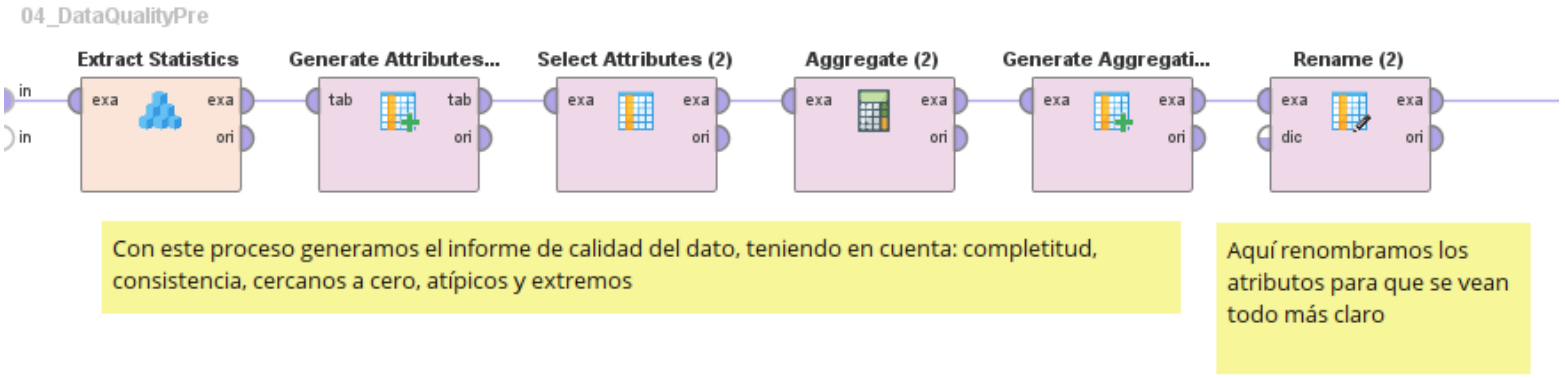
Para el pivotado se agrupan los datos empleando el campo definido en la granularidad (“quality”) y en la cadencia el atributo del cuarto de hora y la hora del día. Se pivotan los 25 atributos restantes, para 7 de ellos se utiliza el sumatorio debido a que son no numéricos, para el resto se emplea el valor medio. En total se generan 1.890 registros pivotados.



3.3.4 Calidad del dato inicial

Una vez se tienen los datos pivotados, se procede a evaluar la calidad de los datos suministrados por el cliente.

Para ello se emplea el operador "Extract Statistics" que genera un dataset con multitud de estadísticas de los tributos por separado.



Empleando estos datos, se genera el siguiente informe:

Informe de calidad de los datos INICIAL		
KPI	**Cálculo**	**Resultado**
Completitud	1-(Missing/1890)	0.994
Consistencia	if(Deviation/sqrt(1890) < 0.1, 0, 1)	0.689
Cercano a cero	if(Deviation < 0.01 && Average < 0.01, 0, 1)	1
Atípico	if(Maximum > (Average + (5*Deviation)), 0, 1)	0.828
Extremo	if(Maximum > (Average + (7*Deviation)), 0, 1)	0.862
Calidad de los datos:		0.875

Row No.	COMPLETITUD	ATIPICOS	CERCANO A CERO	EXTREMOS	CONSISTEN...	CALIDAD DA...
1	0.994	0.828	1	0.862	0.690	0.875

Como se puede ver en el informe de calidad del dato se tiene un 87,4% de calidad de los datos de forma inicial. El punto más débil de esta es la desviación de los datos con una medida del 68,9%. No se cuenta con datos a cero, pero sí extremos y atípicos. Para acabar, el número de datos vacíos es muy pequeño.

3.3.5 Limpieza de los datos

Partiendo del informe de calidad y de la investigación realizada a la distribución de estos. Se realizan los siguientes procesos con el objetivo de mejorar la calidad de los datos y lograr un mejor rendimiento en los siguientes procesos.

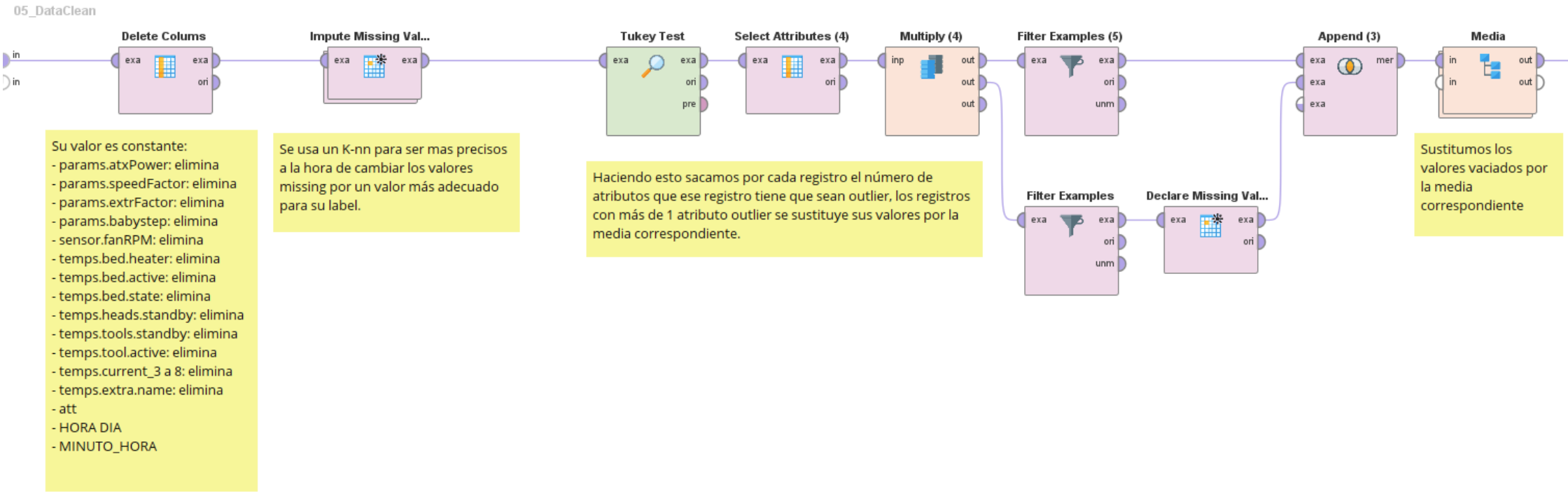
Se eliminan los siguientes atributos, debido a que son constantes o representan un índice : “params.atxPower”, “params.speedFactor”, “params.extrFactor”, “params.babystep”, “sensor.fanRPM”, ”temps.bed.heater”, “temps.bed.active”, “temps.bed.state”, “temps.heads.standby”, “temps.tools.standby”, “temps.tool.active”, “temps.current_3 a 8”, “temps.extra.name” y “att”

Para corregir el dato de COMPLETITUD se emplea el operador “Impute missing values” y un algoritmo K-nn para estimar el valor que debería ir en ese hueco.

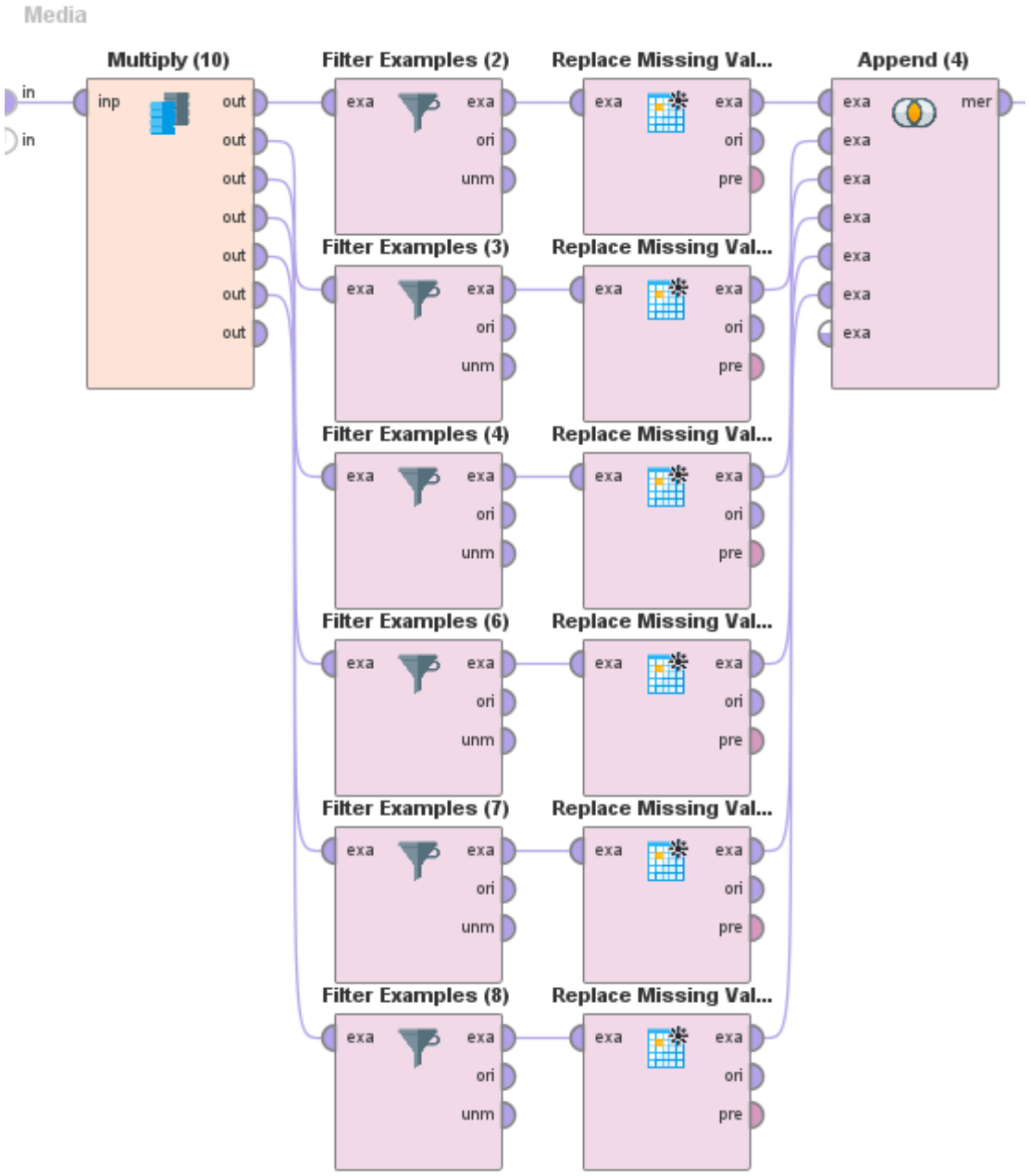
Para corregir los datos de ATÍPICO y EXTREMO se emplea el operador “Turkey test” que a través del cálculo: [Q_1 - 1.5*IQR,Q_3 + 1.5*IQR], genera un nuevo atributo, que indica, por cada registro el número de valores de ese registro que sobrepasan los rangos del cálculo. A continuación se sustituye, por la media de su respectivo label, los registros que cuente con más de un valor que sobrepase el cálculo. De esta manera casi se consigue arreglar los datos de ATÍPICO y EXTREMO.

Asier Vega y Gorka Berganza
31 de mayo de 2022

El proceso de limpieza de datos al completo, el detalle del subproceso “Media” se encuentra a continuación.



El subproceso “Media” sustituye los valores establecidos como nulos tras el “Turkey test”, este separa los datos por su label y cambia los valores nulos por las medias de cada grupo. Después se vuelven a unir todos los datos.



3.3.6 Calidad del dato final

Una vez realizada la limpieza de los datos se vuelve a realizar el informe de calidad (con los mismos operadores que en la sección “3.3.4 Calidad del dato inicial”) de los datos para ver si las mejoras son efectivas:

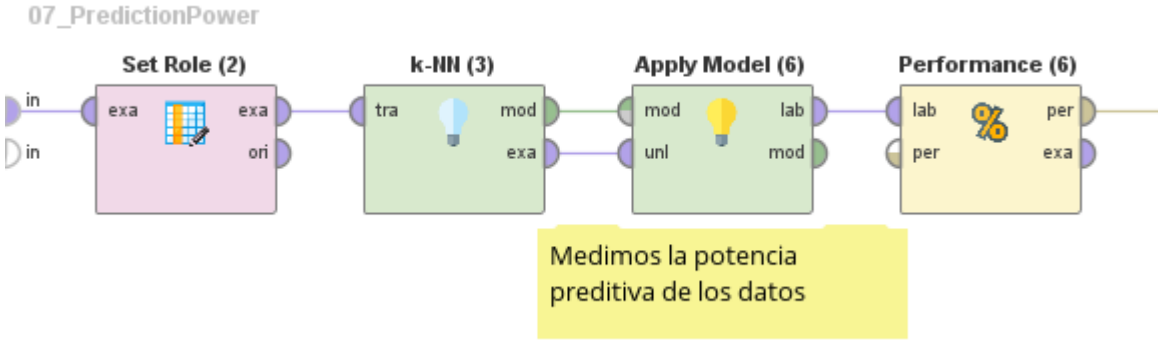
Informe de calidad de los datos INICIAL		
KPI	**Cálculo**	**Resultado**
Completitud	1-(Missing/1890)	1
Consistencia	if(Deviation/sqrt(1890) < 0.1, 0, 1)	0.652
Cercano a cero	if(Deviation < 0.01 && Average < 0.01, 0, 1)	1
Atípico	if(Maximum > (Average + (5*Deviation)), 0, 1)	0.957
Extremo	if(Maximum > (Average + (7*Deviation)), 0, 1)	1
Calidad de los datos:		0.922

Row No.	COMPLETITUD	ATIPICOS	CERCANO A CERO	EXTREMOS	CONSISTEN...	CALIDAD DA...
1	1	0.957	1	1	0.652	0.922

En conclusión, gracias al proceso de limpieza realizado se ha obtenido una mejora del 4,8% respecto al primer informe realizado.

3.3.7 Potencia predictiva

Con el objetivo de validar si será posible lograr la precisión final que el cliente exige (80% - 70%), ejecuta un algoritmo K-nn con los datos, después de la limpieza realizada. Se utiliza el operador de "K-nn" y tras aplicar el modelo y medir su precisión, se obtiene un 86,19% de precisión, suficiente para alcanzar el objetivo fijado.

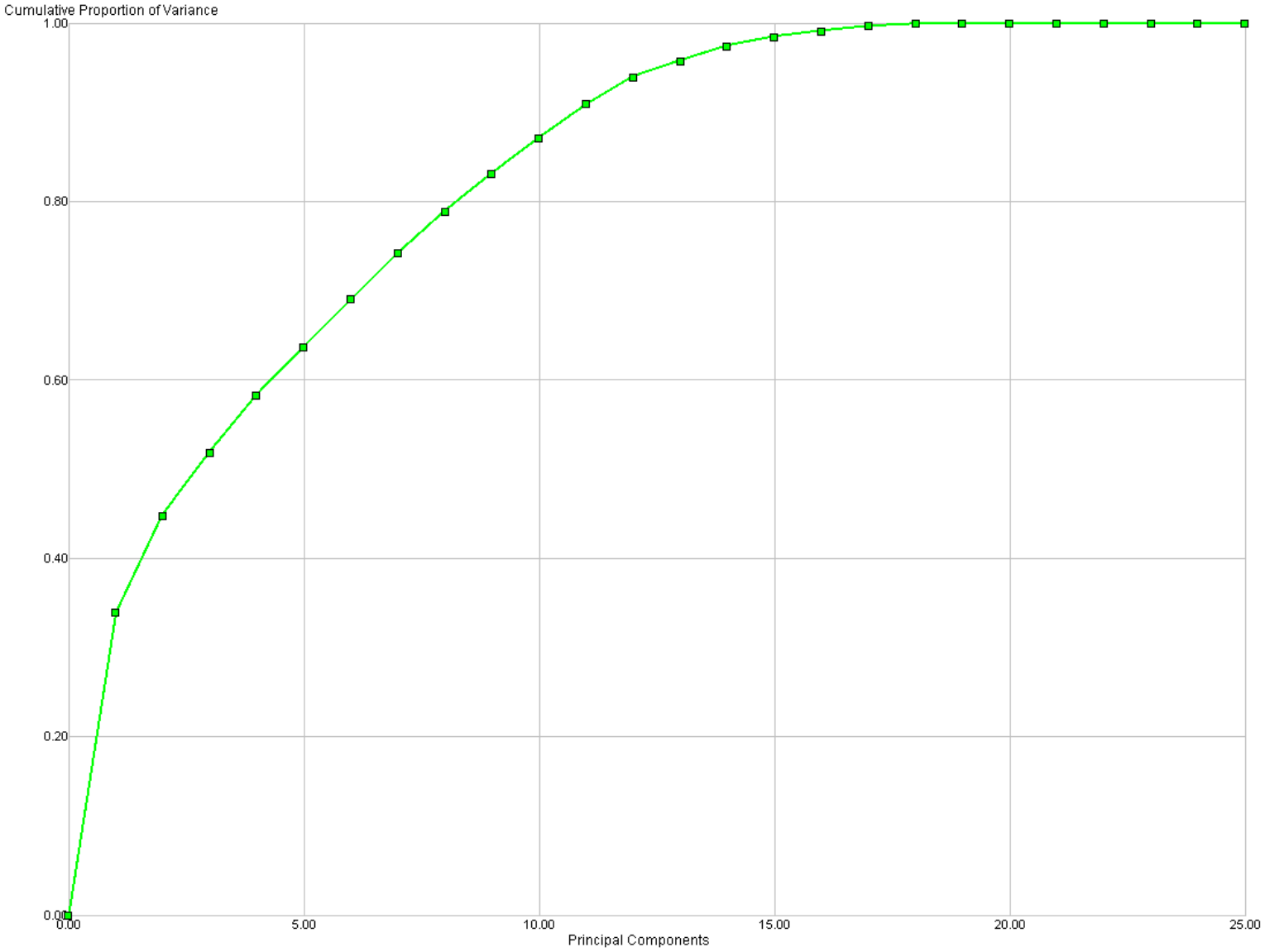


accuracy: 86.19%

	true bowden	true arm_failure	true plastic	true retraction	true unstick	true proper	class precision
pred. bowden	111	4	4	3	0	2	89.52%
pred. arm_failure	1	228	1	6	9	12	88.72%
pred. plastic	2	7	84	1	0	11	80.00%
pred. retraction	2	4	2	408	10	3	95.10%
pred. unstick	0	9	0	13	326	41	83.80%
pred. proper	5	25	12	3	69	472	80.55%
class recall	91.74%	82.31%	81.55%	94.01%	78.74%	87.25%	

3.3.8 Preparación del objetivo minería

Una vez terminadas las fases de ETL, se procede a analizar los datos para reformular el objetivo a predecir/describir. Para ello se normalizan los datos a través de de la "Transformación-Z", con los que se crea una PCA de dos componentes y con las que se extrae el siguiente gráfico:



En este se puede observar que el número de componentes para conseguir una buena precisión es de 15. Por lo que este proyecto se clasifica como un nivel de dificultad intermedio.

Además, partiendo de estos datos normalizados se utiliza el operador "X-means" para realizar un cluster empleando el algoritmo K-means, con 6 mínimos grupos y empleando la distancia cosena, esta es la configuración que mejor distribución ha dado. Para poder visualizar mejor los resultados se utiliza el operador "Cluster model visualizer", con el que se obtienen las reglas y atributos más relevantes utilizados para realizar la clasificación:

Deusto
Universidad de Deusto
Deustuko Unibertsitatea
University of Deusto

Asier Vega y Gorka Berganza
31 de mayo de 2022

Number of Clusters: 6

Cluster 0 384

coords.axesHomed is on average **36.27%** smaller, **params.fanPercent = [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]** is on average **36.27%** smaller, **params.fanPercent = [100.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]** is on average **36.27%** smaller

Cluster 1 414

sensors.probeValue is on average **222.99%** larger, **SEGUNDO_ESTADISTICO/4 = range2 [14.750 - 29.500]** is on average **85.64%** larger, **SEGUNDO_ESTADISTICO/4 = range3 [29.500 - 44.250]** is on average **42.19%** smaller

Cluster 2 288

sensors.probeValue is on average **99.59%** smaller, **SEGUNDO_ESTADISTICO/4 = range1 [-∞ - 14.750]** is on average **82.68%** larger, **accel1Z** is on average **61.76%** smaller

Cluster 3 400

coords.xyz_3 is on average **82.70%** smaller, **SEGUNDO_ESTADISTICO/4 = range3 [29.500 - 44.250]** is on average **76.41%** larger, **sensors.probeValue** is on average **65.74%** smaller

Cluster 4 221

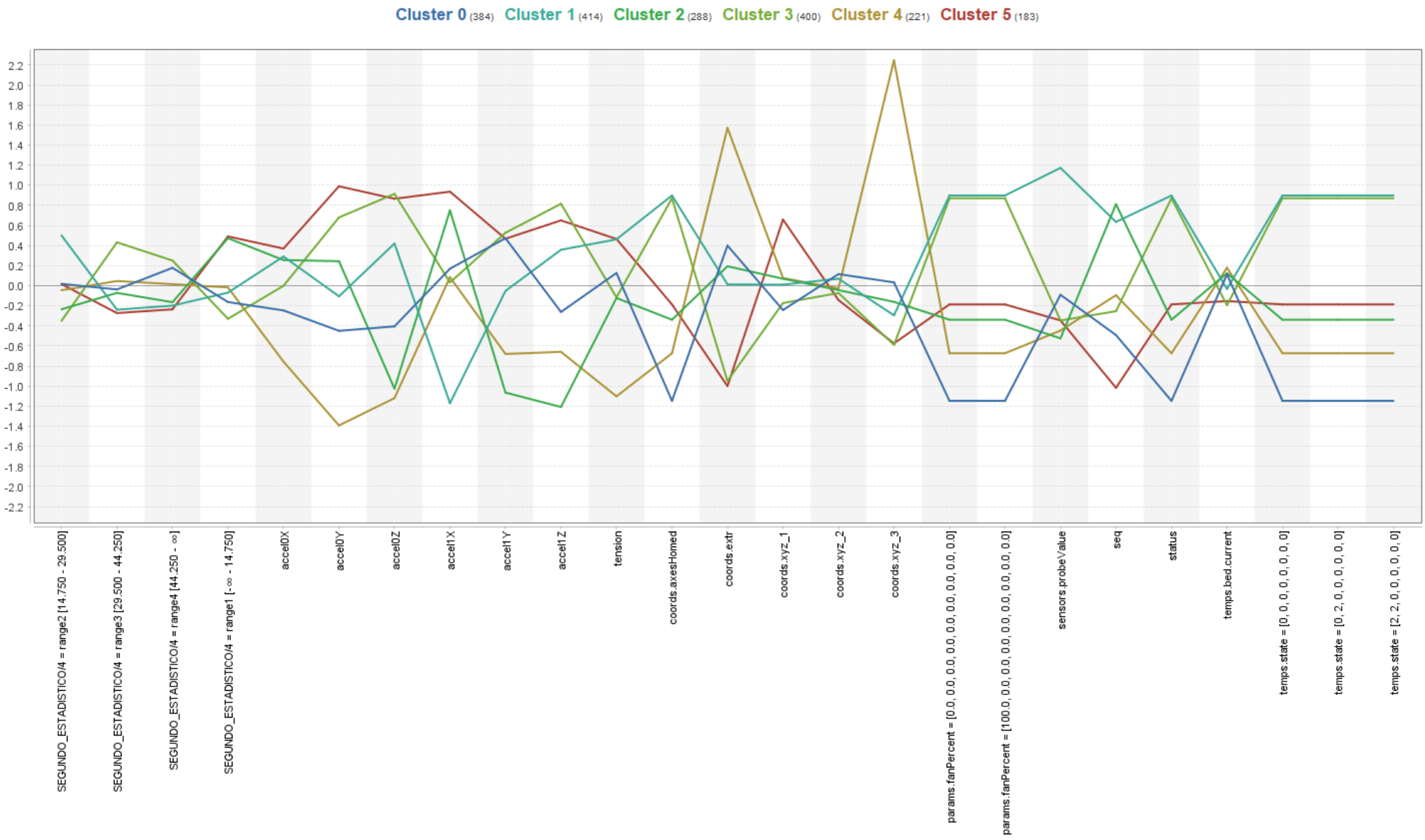
coords.xyz_3 is on average **315.53%** larger, **coords.extr** is on average **85.55%** larger, **sensors.probeValue** is on average **84.83%** smaller

Cluster 5 183

SEGUNDO_ESTADISTICO/4 = range1 [-∞ - 14.750] is on average **85.77%** larger, **coords.xyz_3** is on average **80.72%** smaller, **sensors.probeValue** is on average **65.64%** smaller

Deusto
Universidad de Deusto
Deustuko Unibertsitatea
University of Deusto

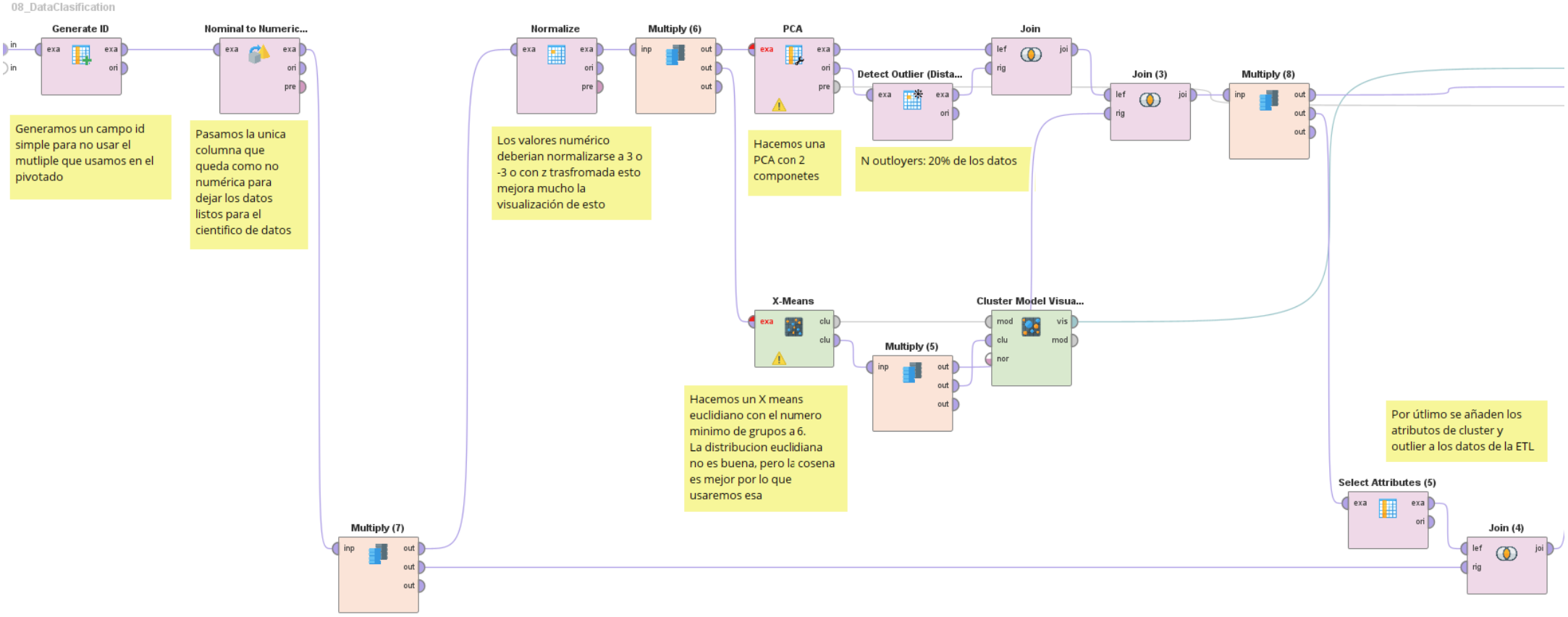
Asier Vega y Gorka Berganza
31 de mayo de 2022



Asier Vega y Gorka Berganza
31 de mayo de 2022

Por último se utiliza el operador “Detecte outlier” a través de la distancia cosena para marcar el 20% de los registros (378) que difieren más de la mayoría. Estos serán eliminados en procesos futuros.

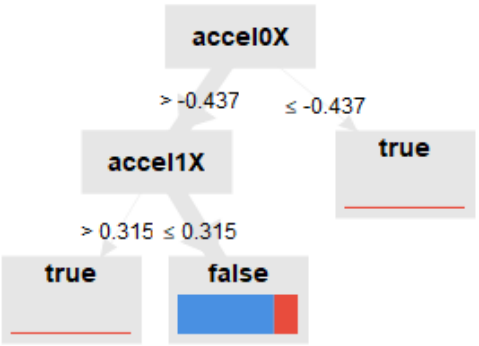
El proceso empleado:



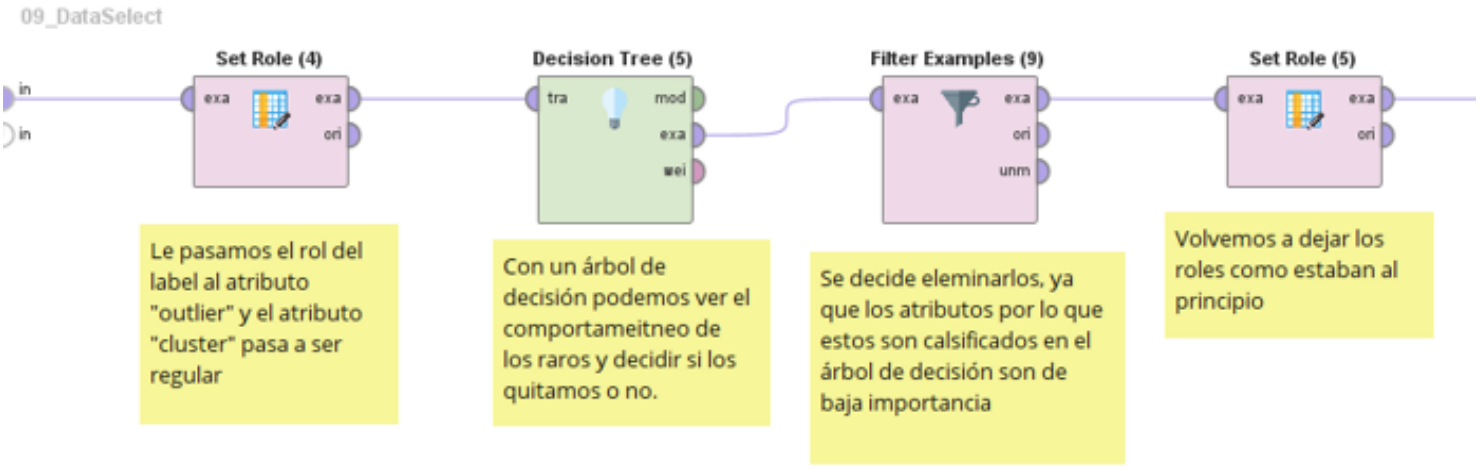
El proceso no supervisado realizado no ha dado conclusiones demasiado claras sobre los datos, pero se van a añadir los atributos generados para el cluster y los outliers a los datos debido a que pueden ser de utilidad más adelante.

3.3.9 Preparación de los atributos para el modelado

Lo primero que se realiza en este proceso es ver qué hacer con los registros marcados como outliers del proceso anterior. Se utiliza una árbol de decisión para ver el motivo de este comportamiento dispar.



En él se observa que solo depende de dos variables asociadas a la aceleración del eje X, se decide eliminar estos outlier debido que estos datos podrían ser resultantes de un fallo en el sensor del eje X de la máquina. Además estas variables no fueron relevantes a la hora de realizar el cluster.



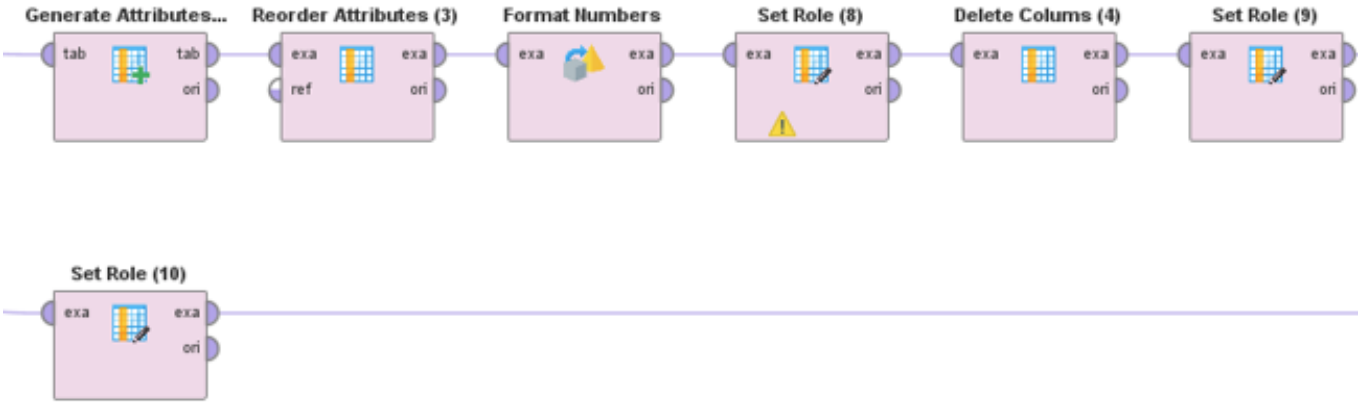
Continuando, como se busca entrenar dos modelos, uno predictivo y otro descriptivo, se realiza el siguiente proceso para dividir el dataset resultante de la ETL, teniendo en cuenta el objetivo de minería.

- Por un lado la label actual es el atributo “quality” el cual se categoriza en los siguientes 5 casos: “arm_failure”, “bowden”, “plastic”, “retraction”, “unstick” y “proper”. Teniendo en cuenta que el objetivo de minería es predecir la calidad del producto, no es necesario predecir el tipo de error. Por lo que se cambia el atributo “quality” por nuevo atributo “ok”. Este tiene un valor de 1 si el atributo “quality” es igual a “proper” y 0 si no lo es. Este nuevo atributo se establece como nuevo label.
- Por otro lado para describir el proceso se emplean tal cual el campo “quality” clasifica los datos. Esto es: “arm_failure”, “bowden”, “plastic”, “retraction”, “unstick” y “proper”.

Como se puede ver se parte del mismo dataset pero se prepara de dos maneras distintas para cada objetivo de minería.

Asier Vega y Gorka Berganza
31 de mayo de 2022

Aquí tenemos los datos pero se hace un proceso que sustituye la label por 1 (si es correcto) y 0 (si tiene fallo)

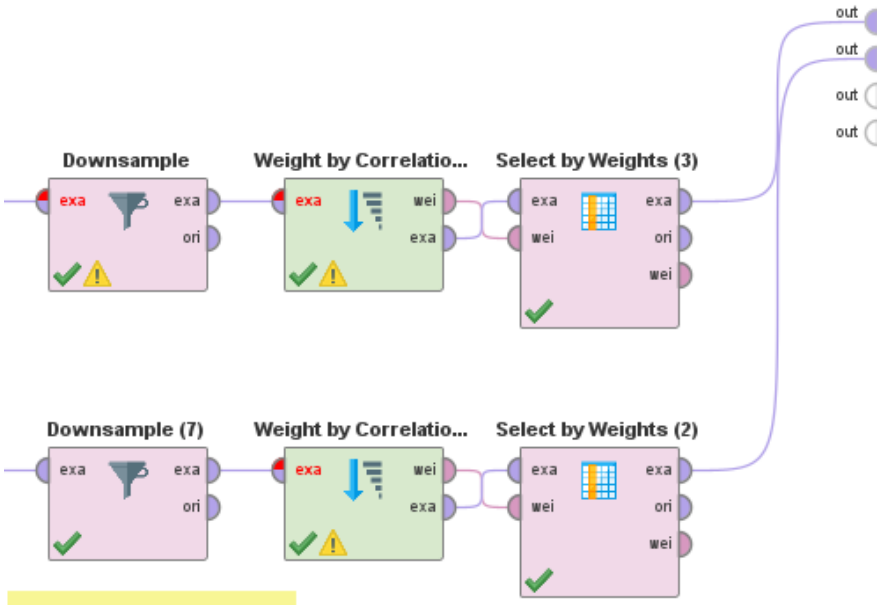


Aquí tenemos los datos tal cual salen del proceso anterior, esto es la label e indica el tipo de fallo o si es correcto

Para acabar con el objetivo de eliminar el sesgo de los datos debido a la presencia mayoritaria de una label respecto de otra, se realiza el siguiente proceso de "downsample":

- "ok" = 1:395 0:1069 -> 1:395 0:395
- "quality" = arm_failure:222 bowden:91 plastic:105 retraction:344 unstick:375 proper:487 -> arm_failure:91 bowden:91 plastic:91 retraction:91 unstick:91 proper:91

Por último para agilizar el tiempo de modelado se emplea el operador "Weight by correlation" para solo coger el top 20 de los 47 atributos más relevantes según un análisis de correlación.



Recuperamos la misma cantidad datos para todas las labels posibles

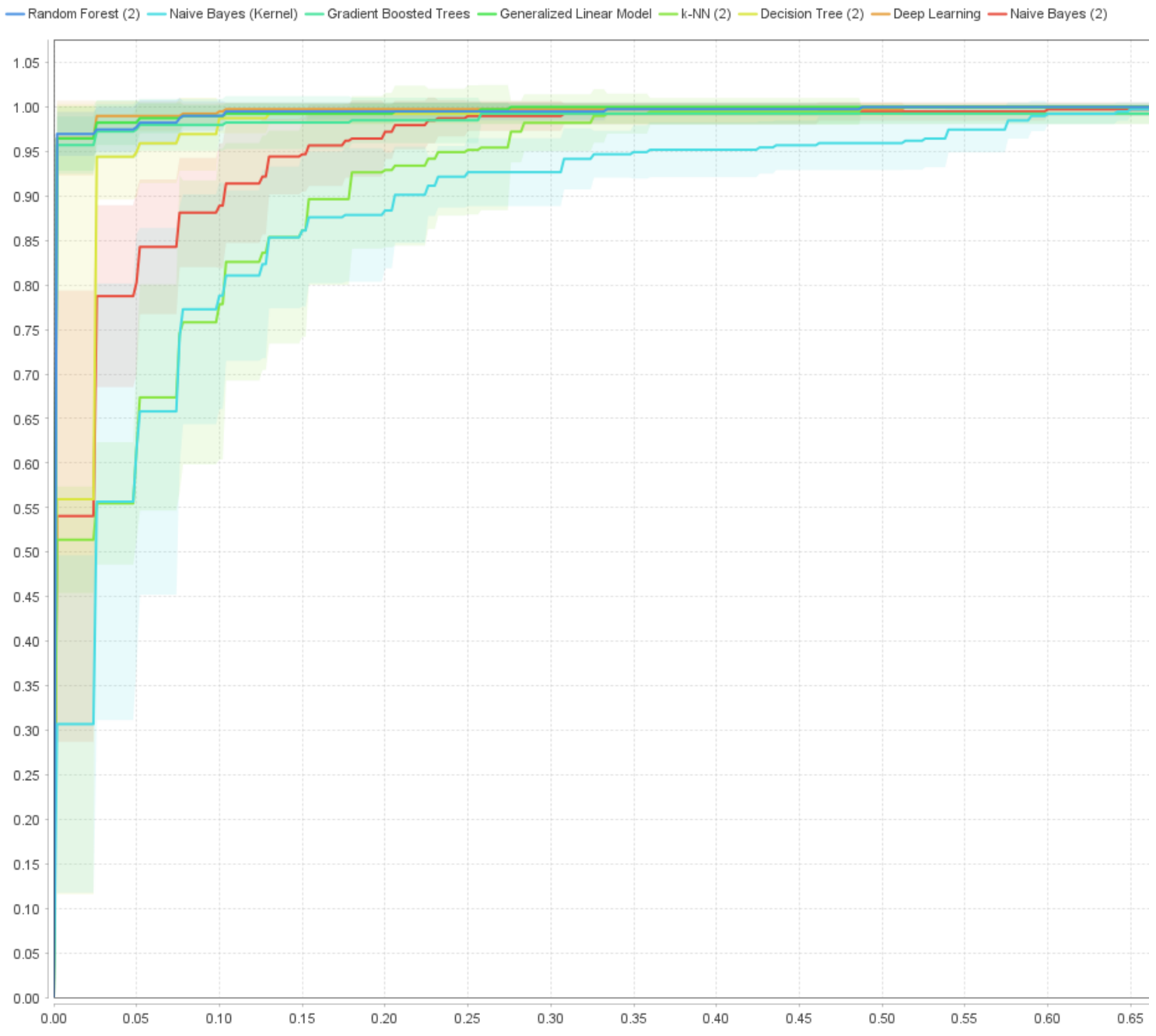
En total son 47 atributos y seleccionamos los 20 mejores

3.3.10 Modelado y resultados

La fase de modelado parte de los dos datasets creados en el proceso anterior, con cada uno se genera un modelo para resolver el problema planteado por el cliente.

3.3.10.1 Predicción

Para la parte predictiva, la label es “1” correcto y “0” fallo, al tratarse de un campo binario es posible utilizar el operador “Copare ROCs” para generar doscientos gráficos ROC de distinto modelo y compararlos, este es el resultado:



Como se puede ver los mejores modelos son el “Random Forest” (con 83% de precisión) y el “Deep Learning” (con 83% de precisión). Se decide usar el segundo debido a que es un poco más preciso.

A continuación se divide el dataset en dos grupos, uno con el 80% de los datos y otro con el 20% de los datos. El grupo más grande se utiliza en el operador “Optimize parameters” que contiene un operador de “Cross Validation” con un modelo “Deep Learning”. El 20% restante se utiliza para generar una medición de la precisión más real.

El operador “Cross Validation” utiliza el 80% de los datos para generar 10 iteraciones y obtener la precisión media a través de repartir esos 80% en dos grupos diferentes por cada una de las 10 iteraciones.

El operador “Optimize parameters” se utiliza para buscar los mejores valores para las propiedades de “learning rate” (rango de aprendizaje) y “epochs” (número de veces que se iteran los datos). En total se generan 5 muestras de 0 a 1 para “learning rate” y 5 muestras 0 a 10000 para “epochs”, que combinados al valor de activación “Rectifier” se generan un total de 36 modelos. Siendo el resultado de modelo con más precisión:

- learning_rate = 0.6
- epochs = 6000.0

Una vez encontrado el mejor modelo, se aplica al 20% de los datos restantes a este y se mide su precisión. Esta es de 98.10%.

accuracy: 98.10%

	true 0	true 1	class precision
pred. 0	77	1	98.72%
pred. 1	2	78	97.50%
class recall	97.47%	98.73%	

3.3.10.2 Descripción

Para la parte descriptiva se cuenta con los mismos datos, pero en esta caso la label cuenta con los 5 casos de error que esta pueda dar además de la label si el resultado es correcto.

En este caso el cliente solicita un árbol de decisión, por lo que se busca optimizar este lo máximo posible.

Se parte de la misma idea de separar los datos en un grupo de 80% y otro de 20%, para obtener una medición de la precisión más real. El grupo más grande pasa por un operador “Optimize parameters” con un árbol de decisión dentro.

El operador “Optimize parameters” se utiliza para buscar los mejores valores para las propiedades de “criterion” (criterio de división de las ramas) 5 casos , “maximal_depth” (profundidad máxima del árbol) 10 muestras de 1 a 100 y “minimal_leaf_size” (profundidad mínimas de ramas) 10 muestra de 1 a 100. En total se generan 484 modelos. Siendo el resultado de modelo con más precisión:

- criterion = gain_ratio
- maximal_depth = 60
- minimal_leaf_size = 1

Asier Vega y Gorka Berganza
31 de mayo de 2022

Una vez encontrado el mejor modelo, se aplica al 20% de los datos restantes a este y se mide su precisión. Esta es de 88.89%.

accuracy: 88.89%

	true bowden	true arm_failure	true plastic	true retraction	true unstick	true proper	class precision
pred. bowden	16	1	3	0	0	0	80.00%
pred. arm_failure	0	16	0	0	0	1	94.12%
pred. plastic	1	0	15	0	0	0	93.75%
pred. retraction	1	1	0	16	0	0	88.89%
pred. unstick	0	0	0	2	16	0	88.89%
pred. proper	0	0	0	0	2	17	89.47%
class recall	88.89%	88.89%	83.33%	88.89%	88.89%	94.44%	

Además del árbol de decisión, se generan las siguientes reglas que permiten entender mejor los atributos que generan estos errores en la máquina.

if accel0Z > 9.645 and sensors.probeValue > 54.068 then retraction

if accel0Z > 9.645 and sensors.probeValue ≤ 54.068 and coords.axesHomed > 1139 then arm_failure

if accel0Z > 9.645 and sensors.probeValue ≤ 54.068 and coords.axesHomed ≤ 1139 and coords.extr > 1238.153 then arm_failure

if accel0Z > 9.645 and sensors.probeValue ≤ 54.068 and coords.axesHomed ≤ 1139 and coords.extr ≤ 1238.153 and accel0X > -0.184 then plastic

if accel0Z > 9.645 and sensors.probeValue ≤ 54.068 and coords.axesHomed ≤ 1139 and coords.extr ≤ 1238.153 and accel0X ≤ -0.184 then bowden

if accel0Z ≤ 9.645 and coords.axesHomed > 1104 then retraction

if accel0Z ≤ 9.645 and coords.axesHomed ≤ 1104 and accel1Y > 0.335 then unstick

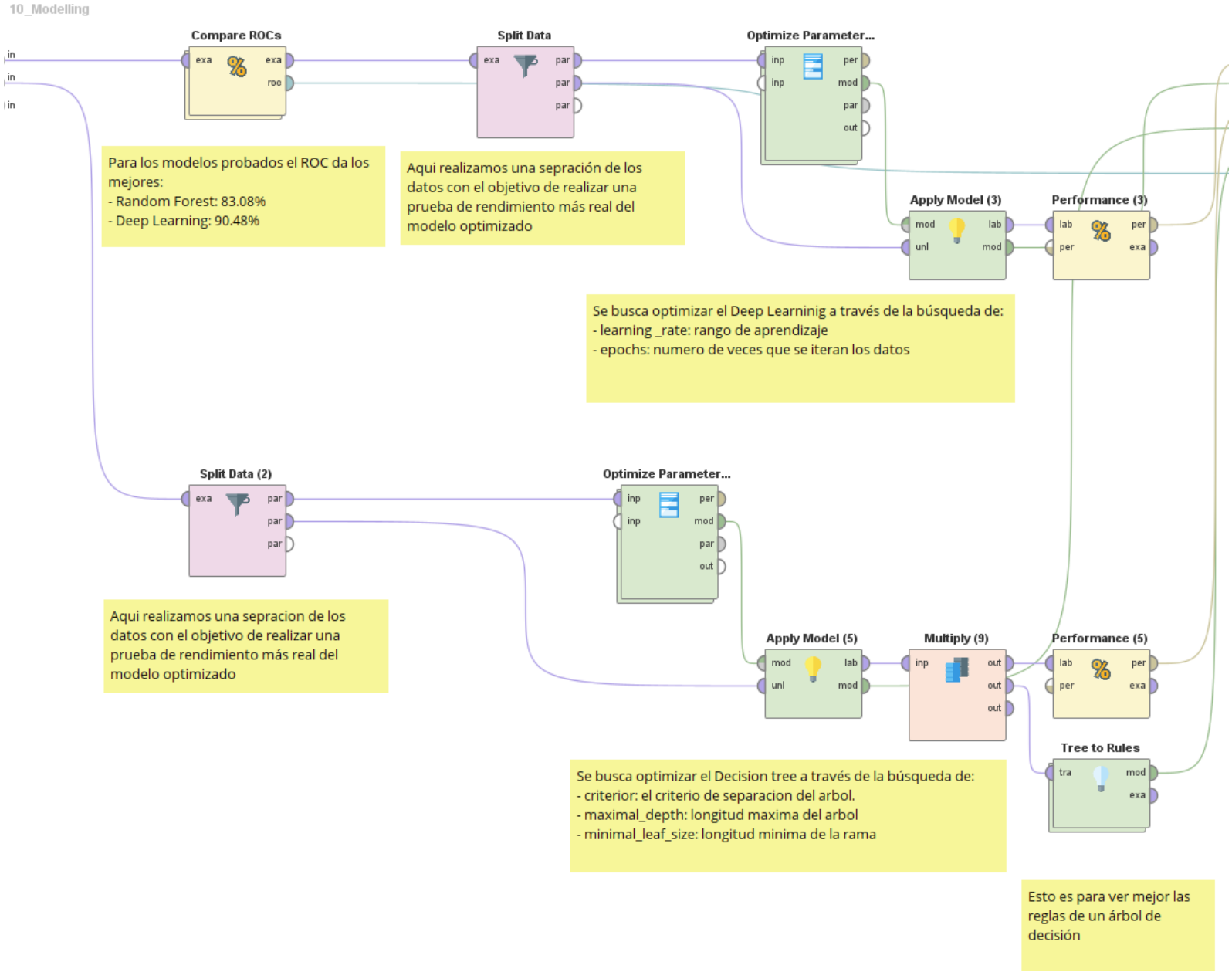
if accel0Z ≤ 9.645 and coords.axesHomed ≤ 1104 and accel1Y ≤ 0.335 then proper

Debido a que el árbol generado es demasiado grande como para incluirlo en esta memoria se deja la descripción del mismo:

```
accel0Z > 9.635
|   sensors.probeValue > 60.687
|   |   accel1Y > 0.360: unstick {bowden=0, arm_failure=0, plastic=0, retraction=0, unstick=1, proper=0}
|   |   accel1Y ≤ 0.360
|   |   |   accel0Y > -0.164: plastic {bowden=0, arm_failure=0, plastic=2, retraction=0, unstick=0, proper=0}
|   |   |   accel0Y ≤ -0.164: retraction {bowden=0, arm_failure=0, plastic=1, retraction=59, unstick=0, proper=0}
|   sensors.probeValue ≤ 60.687
|   |   coords.extr > 2746.761: retraction {bowden=0, arm_failure=0, plastic=0, retraction=7, unstick=0, proper=0}
|   |   coords.extr ≤ 2746.761
|   |   |   coords.extr > 1184.579
|   |   |   |   accel1Y > 0.358: arm_failure {bowden=0, arm_failure=1, plastic=0, retraction=0, unstick=1, proper=0}
|   |   |   |   accel1Y ≤ 0.358
|   |   |   |   |   temps.bed.current > 60.524: arm_failure {bowden=0, arm_failure=47, plastic=0, retraction=0, unstick=0, proper=0}
|   |   |   |   |   temps.bed.current ≤ 60.524
|   |   |   |   |   |   temps.bed.current > 60.175: retraction {bowden=0, arm_failure=0, plastic=0, retraction=2, unstick=0, proper=0}
|   |   |   |   |   |   temps.bed.current ≤ 60.175: arm_failure {bowden=0, arm_failure=7, plastic=0, retraction=1, unstick=0, proper=0}
|   |   |   coords.extr ≤ 1184.579
|   |   |   |   accel0Y > -0.213
|   |   |   |   |   accel0Z > 9.647
|   |   |   |   |   |   accel0Z > 9.720: retraction {bowden=0, arm_failure=0, plastic=0, retraction=1, unstick=0, proper=0}
|   |   |   |   |   |   accel0Z ≤ 9.720
|   |   |   |   |   |   |   accel0X > -0.180
|   |   |   |   |   |   |   |   accel0Z > 9.693: retraction {bowden=0, arm_failure=0, plastic=0, retraction=2, unstick=0, proper=0}
|   |   |   |   |   |   |   |   accel0Z ≤ 9.693
|   |   |   |   |   |   |   |   |   accel0Y > -0.199
|   |   |   |   |   |   |   |   |   |   accel1Y > 0.372: bowden {bowden=2, arm_failure=0, plastic=0, retraction=0, unstick=0, proper=0}
|   |   |   |   |   |   |   |   |   |   accel1Y ≤ 0.372: plastic {bowden=1, arm_failure=0, plastic=64, retraction=0, unstick=0, proper=0}
|   |   |   |   |   |   |   |   |   accel0Y ≤ -0.199: bowden {bowden=3, arm_failure=0, plastic=0, retraction=0, unstick=0, proper=0}
|   |   |   |   |   |   |   accel0X ≤ -0.180
|   |   |   |   |   |   |   |   temps.bed.current > 61.434: retraction {bowden=0, arm_failure=0, plastic=0, retraction=1, unstick=0, proper=0}
|   |   |   |   |   |   |   |   temps.bed.current ≤ 61.434
|   |   |   |   |   |   |   |   |   accel0Y > -0.167: plastic {bowden=0, arm_failure=0, plastic=5, retraction=0, unstick=0, proper=0}
|   |   |   |   |   |   |   |   |   accel0Y ≤ -0.167
|   |   |   |   |   |   |   |   |   |   accel0X > -0.197: bowden {bowden=61, arm_failure=0, plastic=1, retraction=0, unstick=0, proper=0}
|   |   |   |   |   |   |   |   |   accel0X ≤ -0.197
|   |   |   |   |   |   |   |   |   |   |   coords.extr > 350.925: arm_failure {bowden=0, arm_failure=17, plastic=0, retraction=0, unstick=0, proper=0}
|   |   |   |   |   |   |   |   |   |   |   coords.extr ≤ 350.925: bowden {bowden=6, arm_failure=0, plastic=0, retraction=0, unstick=0, proper=0}
|   |   |   |   |   accel0Z ≤ 9.647: proper {bowden=0, arm_failure=0, plastic=0, retraction=0, unstick=0, proper=2}
|   |   |   |   accel0Y ≤ -0.213: proper {bowden=0, arm_failure=0, plastic=0, retraction=0, unstick=0, proper=7}
accel0Z ≤ 9.635
|   accel1Z > -9.896
|   |   accel0X > -0.182: arm_failure {bowden=0, arm_failure=1, plastic=0, retraction=0, unstick=0, proper=0}
|   |   accel0X ≤ -0.182
|   |   |   coords.extr > 1521.900
|   |   |   |   accel0X > -0.238
|   |   |   |   |   accel0Z > 9.589
|   |   |   |   |   |   accel1Y > 0.325
|   |   |   |   |   |   |   coords.axesHomed > 829
|   |   |   |   |   |   |   |   accel0X > -0.229: unstick {bowden=0, arm_failure=0, plastic=0, retraction=0, unstick=58, proper=0}
|   |   |   |   |   |   |   |   accel0X ≤ -0.229
|   |   |   |   |   |   |   |   |   accel0X > -0.229: proper {bowden=0, arm_failure=0, plastic=0, retraction=0, unstick=0, proper=1}
|   |   |   |   |   |   |   |   |   accel0X ≤ -0.229: unstick {bowden=0, arm_failure=0, plastic=0, retraction=0, unstick=10, proper=1}
|   |   |   |   |   |   |   coords.axesHomed ≤ 829: unstick {bowden=0, arm_failure=0, plastic=0, retraction=0, unstick=1, proper=1}
|   |   |   |   |   |   accel1Y ≤ 0.325: proper {bowden=0, arm_failure=0, plastic=0, retraction=0, unstick=1, proper=2}
|   |   |   |   |   accel0Z ≤ 9.589: proper {bowden=0, arm_failure=0, plastic=0, retraction=0, unstick=0, proper=1}
|   |   |   |   accel0X ≤ -0.238: proper {bowden=0, arm_failure=0, plastic=0, retraction=0, unstick=0, proper=1}
|   |   |   coords.extr ≤ 1521.900: proper {bowden=0, arm_failure=0, plastic=0, retraction=0, unstick=0, proper=2}
|   accel1Z ≤ -9.896
|   |   accel0Z > 9.592: proper {bowden=0, arm_failure=0, plastic=0, retraction=0, unstick=0, proper=50}
|   |   accel0Z ≤ 9.592
|   |   |   accel0Y > -0.237: proper {bowden=0, arm_failure=0, plastic=0, retraction=0, unstick=0, proper=5}
|   |   |   accel0Y ≤ -0.237: unstick {bowden=0, arm_failure=0, plastic=0, retraction=0, unstick=1, proper=0}
```

Asier Vega y Gorka Berganza
31 de mayo de 2022

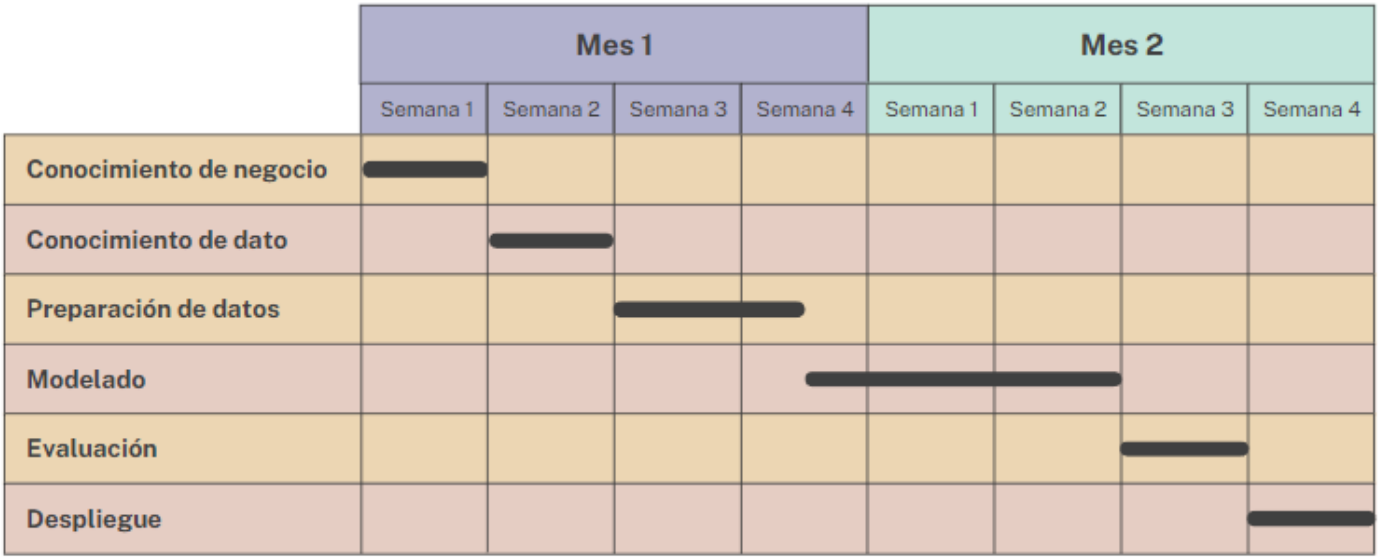
El proceso completo empleado para la creación de los dos modelos es el siguiente:



4. Organización

4.1 Planificación temporal

Se ha fijado la siguiente planificación temporal para el proyecto: 31 de marzo al 31 de mayo.



Conocimiento del negocio: fase para conocer en detalle el ámbito de trabajo del sistema. Se pretende comprender los conceptos de ingeniería mecánica que concierne a la máquina a optimizar.

Conocimiento de dato: fase para conocer al detalle la representación y rangos de los datos del inventario de datos. Aquí se elaborará un informe de la calidad de los datos, por lo que será false que determine si seguir o no con el proyecto.

Preparación de datos: Fase para preparar los datos. Partiendo de la granularidad y cadencia, generan datos útiles y realizan el análisis de calidad del dato.

Modelado: Fase de modelado, se trabajarán los nuevos datos creados para obtener los dos subsistemas dentro de la precisión solicitada por el cliente.

Evaluación: Fase de pruebas del modelo, se probará de forma simulada el funcionamiento del sistema intentando buscar posibles fallos.

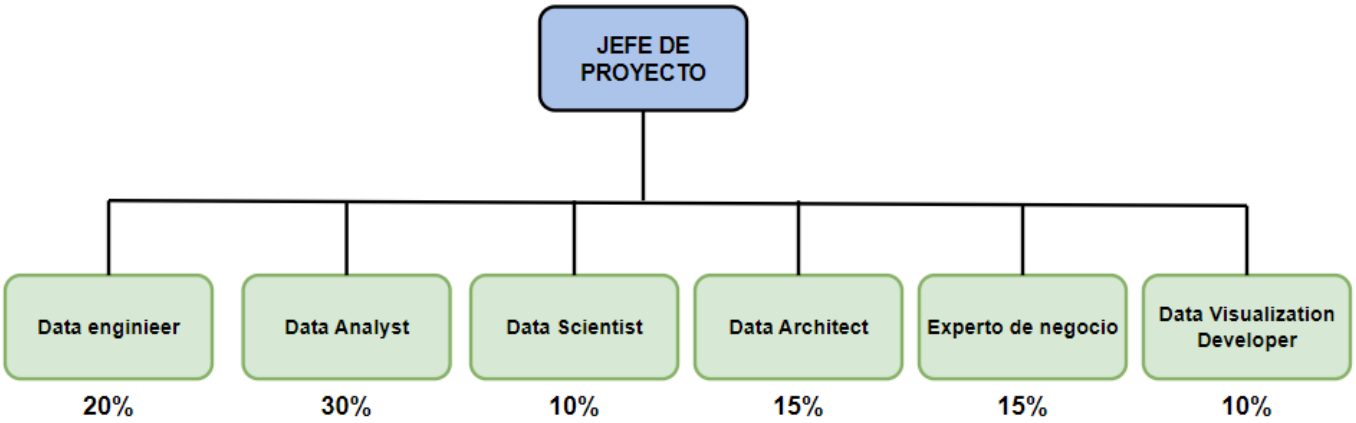
Despliegue: Fase de despliegue en las instalaciones del cliente.

4.2 Entregable

Teniendo en cuenta la planificación fijada, el entregable del proyecto se realiza el 24 de mayo (coincidiendo con el inicio de la fase de despliegue) y los días posteriores se utilizará para realizar las pruebas pertinentes de funcionamiento.

4.3 Organigrama del proyecto

A continuación se muestra el organigrama de las personas implicadas en el proyecto con su porcentaje de participación en el mismo.



Se han estimado los siguientes porcentajes de participación en el proyecto y estas serán sus tarea a desarrollar:

- Data engineer 20%: el trabajo de esta persona se realiza durante las primeras fases críticas del proyecto donde al final de la fase de conocimiento elabora un informe de calidad de datos, que justifica la continuidad del proyecto.

- Data analyst 30% y Data scientist 10%: el trabajo de estas personas está orientado a la mejora de la calidad de los datos, por lo que será crucial a la hora de alcanzar la precisión solicitada por el cliente.

- Data architect 15%: el trabajo de esta persona está orientado a la arquitectura del sistema y la implantación de este sistema dentro de las instalaciones del cliente, así como será el principal responsable durante la fase de despliegue.

- Experto de negocio 15%: el trabajo de esta persona está orientado a trabajar con el cliente para comprender al detalle el ámbito del sistema así como los procesos de la máquina involucrada.

- Data visualization developer 10%: el trabajo de esta persona está orientado al desarrollo de la interfaz de usuario del sistema. Esta deberá mostrar los datos clave de una forma clara para el cliente.

4.4 Presupuesto

Se ha fijado el siguiente presupuesto para el proyecto:

Presupuesto:				
Presupuestado por: Asier y Gorka			**Cliente:** A&G	
Responsable: Asier y Gorka			**Proyecto:** Sistema predictivo - descriptivo	
Duración del proyecto: 2 meses			**Solicitante:** Responsable de producción	
Elemento	**Tipo de recurso**	**Tipo de unidad**	**Unidades**	**Coste**
Data engineer	Sueldos	participación	20%	5.000
Data analyst	Sueldos	participación	30%	7.000
Data science	Sueldos	participación	10%	3.000
Data architect	Sueldos	participación	15%	4.000
Expt. negocio	Sueldos	participación	15%	4.000
Visualization	Sueldos	participación	10%	3.000
Jefe proyecto	Sueldos	mes	2	5.000
Desarrollo	Materiales	mes	2	2.000
Licencias	Materiales	mes	2	6.000
Despliegue	Arquitectura	mes	1	1.000
			Total:	40.000

4.5 Retorno de la inversión

Rendimiento del capital invertido:

Se estima que el sistema, entre la mejora de los resultados y prevención de errores, puede ayudar a tener un beneficio de 7.500€ anuales.

Retorno de la inversión (1 años): $\frac{7500}{40000} * 100 = 18{,}5\%$

Retorno de la inversión (6 años): $\frac{45000}{40000} * 100 = 112{,}5\%$

Retorno de la inversión (10 años): $\frac{75000}{40000} * 100 = 187{,}5\%$

Los resultados del retorno de la inversión son los siguientes:

Inversión inicial: 40.000€

Ganancias anuales: 7500€

PRI = 40.000/7500 = (5,3) 5 años 3 meses y 3 semanas se tardaría en recuperar la inversión según lo estimado

5. Despliegue y arquitectura

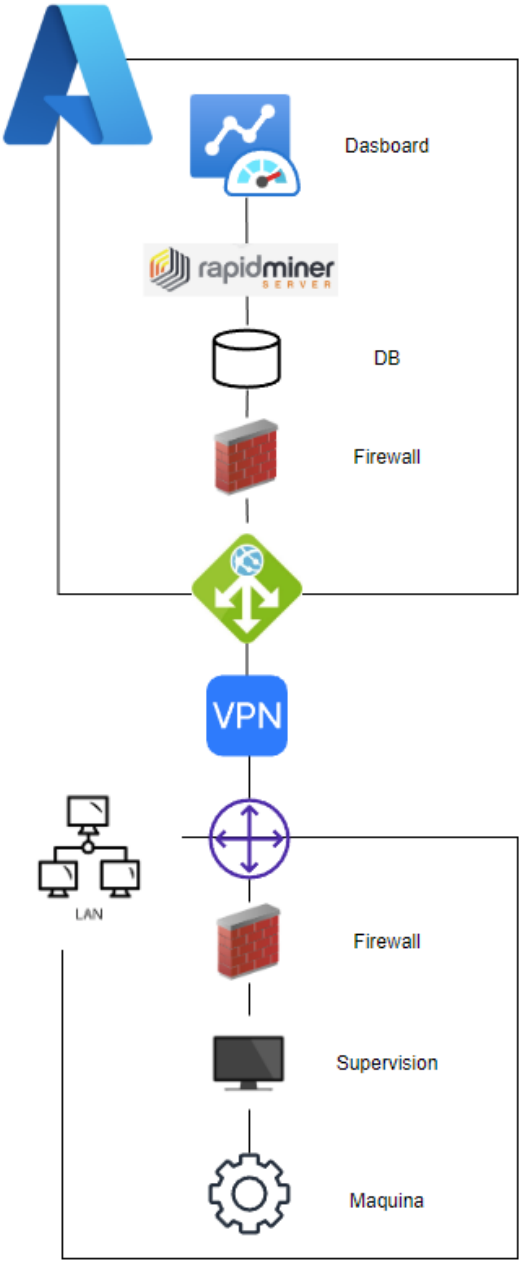
El cliente ha solicitado que este sistema esté ubicado en la nube por lo tanto la arquitectura propuesta es la siguiente:

Partiendo de la red local del cliente, donde se extraen los datos de la máquina a través del sistema de supervisión. Se envían por una VPN de forma segura los datos a la plataforma Azure.

En Azure lo primero que se hará será exportar los datos que el cliente tiene en azure a una DB con la que nosotros trabajemos, después, se procesarán vía Rapidminer entrenando así nuestro modelo.

Una vez listo el modelo, el funcionamiento normal sería el siguiente: cada vez que se planifique una petición de producción para nuestra máquina, con antelación se enviarán esos datos a Azure y se realizará la predicción, esta se mostrará en un dashboard (donde se verán las alarmas de forma gráfica) accesible por el usuario vía navegador. En el caso de que la calidad predecida no coincida con la final se ejecutará automáticamente el sistema descriptivo mostrando el resultado en el dashboard.

Esta última parte se deja como propuesta de arquitectura de este sistema. Queda como trabajo futuro realizarlo.



6. Conclusiones

Tras finalizar la elaboración del proyecto, se han extraído las siguientes conclusiones.

Ha sido posible cumplir con los requisitos del cliente:

Por un lado se ha obtenido un modelo predictivo, que predice cada 15 minutos si el proceso actual de impresión 3D resultará en fallo o no. Avisando en caso negativo al responsable de la máquina para que interrumpa o corrija el proceso en marcha. La precisión de este sistema supera el 80% solicitado por el cliente.

Por otro lado el modelo descriptivo, describe a través de un árbol de decisión gráfico y una serie de reglas, los motivos por los que la máquina produce piezas con defecto, y el tipo de defecto, así como el camino a seguir para tener piezas correctas siempre. De esta manera se puede lograr optimizar la máquina a través del ajuste de sus parámetros. La precisión de este sistema supera el 70% solicitado por el cliente.

Ha sido posible entender los siguientes conceptos académicos:

Se ha trabajado partiendo de un caso real, dado que los datos utilizados así como la problemática bien planteados por el el proyecto de partida utilizado para este proyecto.

Se ha podido comprender el concepto detrás de la extracción, transformación y carga de los datos, así como la limpieza, validación y potencia de los datos para sistemas de inteligencia artificial.

Se ha podido comprender los conceptos detrás de la elección y optimización de un modelo de inteligencia artificial para dos de sus problemáticas más comunes, la predicción y la descripción.

Se ha podido comprender los conceptos detrás del análisis de datos de procesos y máquinas industriales, a través del desarrollo de dos modelos partiendo de datos reales de una máquina real.

7. Bibliográfica

[1] J. Sendorek, T. Szydlo, M. Windak, y R. Brzoza-Woch, «Dataset for anomalies detection in 3D printing». arXiv, 19 de abril de 2020. Accedido: 31 de marzo de 2023. [En línea]. Disponible en: http://arxiv.org/abs/2004.08817

[2] «3D-Printing-Data-OpenDataLab». https://opendatalab.com/3D-Printing-Data (accedido 15 de mayo de 2023).

[3] «joanna-/3D-Printing-Data: This repository contains data gathered from 3D printing machine with different printing failures.» https://github.com/joanna-/3D-Printing-Data (accedido 31 de marzo de 2023).

[4] «Papers with Code - 3D-Printing-Data Dataset». https://paperswithcode.com/dataset/3d-printing-data (accedido 31 de marzo de 2023).

[5] Szydlo T., Sendorek J., Windak M., Brzoza-Woch R. (2021) Dataset for Anomalies Detection in 3D Printing. In: Paszynski M., Kranzlmüller D., Krzhizhanovskaya V.V., Dongarra J.J., Sloot P.M. (eds) Computational Science – ICCS 2021. ICCS 2021. Lecture Notes in Computer Science, vol 12745. Springer, Cham. https://doi.org/10.1007/978-3-030-77970-2_50

8. Anexos

Adjunto esta documentación:

- El proceso al completo
- Las fotos con el detalle de cada parte del proceso
- La foto del árbol de decisión al completo