Uso de minería de datos en Industria 4.0

Achraf Hmimou

Escuela Politécnica Superior de Ingeniería de Vilanova i la Geltrú

Resumen

La gran cantidad de datos generada por la cuarta revolución industrial (Industria 4.0) puede suponer un avance significativo en la mejora de los procesos. Extraer información útil de estos grandes volúmenes de datos es el principal reto y es también, donde la Minería de Datos juega un papel clave.

Este Trabajo Final de Grado parte de la idea de seguir un proceso de minería de datos en un escenario real de producción industrial. El escenario concreto es una célula de fabricación de piezas para el sector automovilístico mediante fundición a presión.

La aplicación central es la implementación de mantenimiento predictivo, una técnica que permite la detección de comportamientos anómalos para poder anticiparse a posibles fallos. La predicción se realiza mediante el análisis y aprendizaje del comportamiento de los sensores instalados en las máquinas o equipos.

Siguiendo el método estándar de minería de datos, los dos objetivos principales son: 1) Obtener modelos de aprendizaje automático para implementar el mantenimiento predictivo, y 2) Crear una aplicación que automatice el proceso que incluye la recogida de datos, el tratamiento de éstos y, finalmente, la visualización de los resultados en tiempo real.

Para llevar a cabo estos objetivos, se exploran primero diferentes propiedades y técnicas de tratamiento de datos obtenidos de sensores. Con los datos tratados, se crean y evalúan modelos que aprenden de los datos y puedan ser usados para predecir futuros comportamientos. Finalmente, se crea una aplicación combinando diferentes tecnologías para poder realizar análisis de resultados y monitorización de la evolución del estado de los sensores.

1. Introducción

La inmensa cantidad de información disponible hoy en día y los grandes avances en la informática hace que la **Minería de Datos** sea una de las disciplinas más populares. Encontrar patrones, tendencias, anomalías y extraer información útil de grandes volúmenes de datos puede ser clave para afrontar futuras decisiones.

Por otra parte, la cuarta revolución industrial, también conocida como **Industria 4.0**, supone un paso gigantesco en la mejora de los procesos industriales. Los sensores **IoT** (*Internet of Things*) son la base de esta revolución y permiten una gran recolección de datos que pueden ser explotados.

Este proyecto se titula "*Uso de minería de datos en industria 4.0*" porque el objetivo es utilizar la información generada por los sensores durante un proceso productivo para mejorarlo mediante minería de datos.

Este proyecto está realizado en CIE C.Vilanova [1], una fábrica que forma parte de la compañía *CIE Automotiv*e, uno de los principales proveedores de piezas para el sector de la automoción. La fábrica tiene una refinería, en la que se crea la aleación que forma las piezas, y cuenta además con las diferentes máquinas que permiten la creación de las piezas mediante inyección. También dispone de áreas donde se mecanizan las piezas. El producto final son piezas mecanizadas o sin mecanizar.

Concretamente, este proyecto se centra en los datos generados por las células de inyección. En éstas intervienen diferentes componentes para la elaboración de las piezas. El tener bajo control el funcionamiento de los componentes es vital para su mantenimiento así como asegurar la calidad del producto.

El **mantenimiento predictivo** [2] es una propuesta que se apoya en los avances de la Inteligencia Artificial para que la tarea de mantenimiento se realice sólo cuando sea necesario. La clave está en estudiar los patrones de comportamiento y detectar posibles anomalías que puedan afectar a futuro en la máquina.

El alcance de este proyecto abarca los siguiente puntos:

- Descripción de los elementos que intervienen en una célula de producción.
- Aplicación de la metodología de minería de datos (CRISP-DM) a un proceso productivo.
- Análisis de negocio y exploración de potenciales aplicaciones (Mantenimiento predictivo).
- Análisis de datos de naturaleza temporal y técnicas de análisis multivariado.
- Creación y evaluación de modelos de aprendizaje automático.
- Diseño y despliegue de una arquitectura que automatice el proceso de recolección, tratamiento y visualización de datos.
- Desarrollo web basado en frameworks modernos (Vue JS).

Adicionalmente también se describen conceptos contextuales como la fundición a presión.

2. Escenario y objetivos

La Célula 86, una de las células de las que dispone CIE C.Vilanova, ha sido el escenario del proyecto debido a que recientemente se le han incorporado una gran cantidad de sensores adicionales. Además, se trata de una célula nueva y dispone de una muy buena documentación.

La célula consta de diferentes elementos que se dividen en dos grupos: Inyección y Periféricos.

- **Inyección**: Elementos que intervienen en el proceso de inyección.
  - Máquina Inyectora
  - Horno de mantenimiento
  - Cargador de metal
  - Lubricador
- **Periféricos**: Elementos periféricos de acabado de la pieza de fundición.
  - Robot
  - Prensa
  - Cuba de refrigeración con elevación
  - Màquina de secado por vacío
  - Mesa de reintroducción de piezas
  - Granalladora
  - Atemperadores

El PLC Máster es quien se encarga de orquestar la máquina inyectora y el robot, y éstos, a su vez, se encargan de controlar los demás elementos (ver Figura 1).

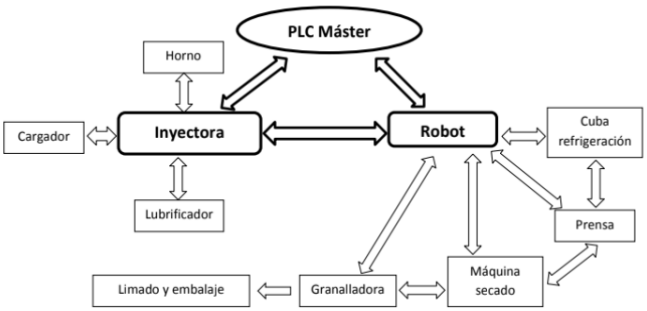

Fig. 1. *Esquema de comunicación Célula 86*

Mediante el PLC, es posible leer los datos de los sensores que forman parte de la célula. Con estos datos son muchas las aplicaciones que se pueden realizar. Algunas de las aplicaciones más comunes son la de monitorización y estudio de la evolución de los sensores así como otras aplicaciones que incluyen factores predictivos.

Este proyecto tiene dos objetivos principales para aplicar la minería de datos al proceso. Dichos objetivos son:

- El objetivo principal es aplicar minería de datos sobre los datos recogidos de los sensores para implementar el concepto de **mantenimiento predictivo.** Concretamente, el objetivo es obtener modelos de aprendizaje automático que aprendan del comportamiento de la célula y que sea posible detectar **anomalías** así como **predecir en el futuro** el comportamiento de ésta.

- Por otra parte, aprovechando la información generada, el otro objetivo principal es crear una **aplicación** que realice: 1) El proceso de recolección de datos, 2) su procesado mediante modelos y 3) su visualización. La aplicación final será un visualizador que permite ver los resultados obtenidos del estudio realizado. La aplicación está ideada para el departamento de ingeniería. Concretamente, para que se puedan realizar tareas de análisis y monitorización en tiempo real de los datos..

Ambos objetivos están relacionados entre sí aunque se puede considerar que la minería de datos es una capa superior de proceso añadida a la aplicación. Finalmente, recalcar que ambos objetivos se encuentran dentro de la metodología de minería de datos estándar.

3. Minería de datos

La minería de datos combina la informática, la matemática y el conocimiento del dominio para extraer información útil a través de datos. Normalmente, la disciplina se aplica sobre grandes volúmenes de datos y el objetivo es descubrir patrones, dependencias, tendencias, grupos, etc. Con los últimos avances tecnológicos, es posible procesar cantidades abismales de información pudiendo realizar cálculos y combinaciones de forma semiautomática o automática.

Mediante técnicas de aprendizaje automático (*machine learning*), es posible crear modelos que resuman el conocimiento observado y se puedan utilizar, por ejemplo, para predecir eventos futuros, clasificar información u obtener patrones recurrentes. Estos modelos pueden ahorrar costes y tiempo realizando tareas que el ser humano sería incapaz o que serían inviables por el tiempo requerido. También pueden ser un gran soporte para el análisis de datos puesto que se basan en la estadística y la matemática para obtener resultados.

3.1. Metodología: CRISP-DM

La metodología que se aplica en este proyecto se llama CRISP-DM (*Cross Industry Standard Process for Data Mining*). Se trata de la metodología más utilizada para proyectos de minería de datos. Esta metodología consiste en 6 etapas, cuyo desarrollo no es lineal sino más bien iterativo.

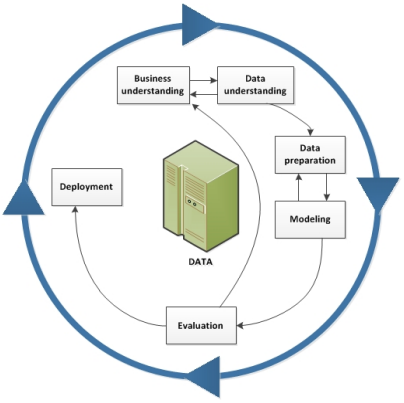

Fig. 2. *Esquema del ciclo de vida de un proyecto mediante la metodología CRISP-DM [3]*

Esta metodología es flexible y permite tener una concepción de las diferentes etapas necesarias para llevar a cabo un proyecto de este tipo. En este proyecto se ha intentado seguir esta metodología y en los siguientes apartados se describen

las decisiones y el desarrollo de las diferentes etapas mostradas en el esquema de la Figura 2.

3.2. Análisis de negocio

Este proyecto tiene un título genérico debido a que en el momento de hacer el registro, todavía se estaba estudiando la viabilidad de diferentes soluciones.

Las ideas principales fueron: el mantenimiento predictivo, la calidad predictiva y el estudio multivariado de parámetros de proceso para encontrar los parámetros más influyentes en el funcionamiento de la máquina.

La calidad predictiva requiere la trazabilidad de la pieza durante el proceso y al pasar los controles de calidad. Esta premisa requiere también que el muestreo sea lo suficientemente grande. Sin embargo, los controles de calidad son relativamente limitados respecto al gran volumen de muestras requeridos para obtener buenos resultados.

En cuanto al análisis de los parámetros más influyentes, se trata de una tarea complicada de la cual era difícil encontrar un enfoque adecuado. No obstante, podía formar parte de la optimización de otra solución debido a que está relacionado con la ingeniería de características.

Finalmente, la opción más viable con los datos que se podían obtener, es la del mantenimiento predictivo que además podía suponer una buena herramienta para mejorar el rendimiento de la máquina. Aún así, es importante destacar que se tuvo en cuenta que se trataba de un objetivo donde es difícil obtener buenos resultados debido a la dimensionalidad del problema.

3.2. Análisis de datos

El paso inicial al aplicar minería de datos es estudiar los datos y entender su naturaleza. La exploración de datos es fundamental para las próximas decisiones puesto que permite entender los datos antes de realizar hipótesis.

El estudio se ha realizado en el entorno de ejecución *Google Colab***.** Este entorno permite ejecutar código en lenguaje *Python* utilizando un formato de libreta interactiva, lo que lo convierte en un entorno ideal para el estudio de datos. Se puede visualizar la libreta utilizada en el enlace de la referencia [4].

El estudio y modelaje inicial se realizará sobre una muestra de aproximadamente 24 horas extraída de la base de datos. También es importante recalcar que para simplificar el análisis y modelaje en este proyecto, solo se tienen en cuenta las señales **críticas**, definidas por ingeniería. No obstante, el proceso de análisis puede ser extrapolado al conjunto total de señales en un futuro. El *dataset* de la muestra descrita tiene la estructura mostrada en la Figura 3.

La muestra es un *dataset* de **naturaleza temporal**, ya que se trata de un conjunto de muestras/datos ordenados cronológicamente observados en distintos momentos, y **multivariante** por el hecho de que hay diferentes variables a observar.

```
[ ] criticas.info()

<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 86042 entries, 2022-06-13 07:10:04.084000+00:00 to 2022-06-14 07:10:03.068000+00:00
Data columns (total 17 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Caudal_Piston           86042 non-null  int64
 1   Flow_CIR_1              86042 non-null  float64
 2   Flow_CIR_2              86042 non-null  float64
 3   Flow_CIR_3              86042 non-null  float64
 4   Grueso_colada           86042 non-null  float64
 5   Pres_Entrada_Agua_Maquina  86042 non-null  float64
 6   Pres_Entrada_Aire       86042 non-null  float64
 7   Pres_Final_Mult         86042 non-null  int64
 8   Pres_Maxima_Mult        86042 non-null  int64
 9   Pres_Retorno_Agua_Maquina  86042 non-null  float64
 10  Pres_Retorno_Agua_Molde  86042 non-null  float64
 11  Temp_Cuba               86042 non-null  float64
 12  Temp_Horno              86042 non-null  float64
 13  Tiempo_Subida_RT        86042 non-null  float64
 14  Vel_1a_Fase_Media       86042 non-null  float64
 15  Vel_2a_Fase_Maxima      86042 non-null  float64
 16  Vel_2a_Fase_Media       86042 non-null  float64
dtypes: float64(14), int64(3)
memory usage: 11.8 MB
```

Fig. 3. *Muestra de 24h de señales críticas*

3.2.1. Análisis de series temporales

Los datos de los sensores de la máquina son de naturaleza **temporal** ya que son un conjunto de mediciones que se realizan periódicamente en el tiempo. Este tipo de datos constituyen las series temporales y se pueden encontrar en muchos ámbitos como el mercado de stocks, datos meteorológicos o provenientes de sensores, entre otros.

En este proyecto se ha estudiado la naturaleza de este tipo de datos y sus propiedades principales. También se han explorado algunas técnicas para la extracción de características y procesado de los datos.

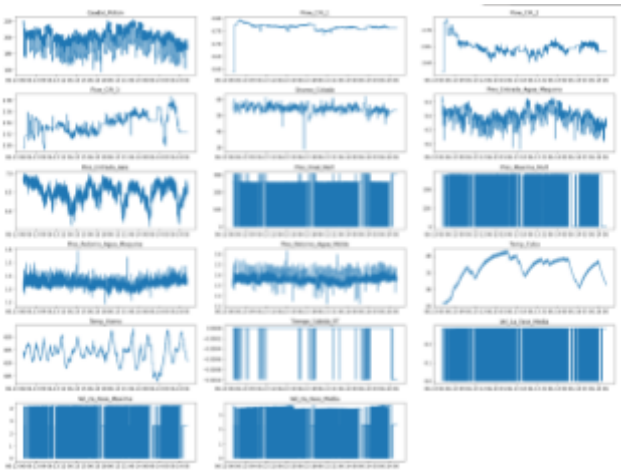

Fig. 4. *Gráficas temporales de la muestra de señales críticas*

En la Figura 4, se muestran todos los puntos (mediciones) del dataset en el tiempo, obteniendo la evolución de cada uno de los sensores. Como se puede observar, hay muchos sensores con comportamiento muy homogéneo y también hay algunos comportamientos irregulares que podrían ser anomalías.

Para entender mejor la naturaleza de este tipo de datos, se han estudiado las diferentes propiedades de una serie temporal. La tendencia, estacionalidad, componente cíclica,

ruido, autocorrelación son propiedades de una serie temporal que permiten entender el comportamiento de las señales.

En este proyecto se han estudiado estas propiedades en la muestra de datos debido a que es posible optar por unos u otros modelos dependiendo de este análisis. Por ejemplo, algunos modelos como *ARIMA/SARIMA* [5], parten de la premisa de que los datos son estacionarios. Hay tests que permiten verificar la estacionalidad de una muestra como el *Augmented Dickey-Fuller Test* [6] y también hay técnicas de preprocesado como del *detrending* que permiten convertir un conjunto no estacionario a estacionario eliminando la componente tendencia de la serie. Por otra parte, otros modelos como los vectores autorregresivos [7] funcionan mejor cuando hay altas auto correlaciones, por lo que también se ha realizado un análisis de esta propiedad.

3.2.1. Análisis multivariado

Otra de las características principales de los datos con los que se trabaja es que se trata de un *dataset* multivariado, ya que hay que tener en cuenta múltiples variables diferentes. Estudiar cada una y la relación entre ellas, permite tener una mejor conciencia sobre los datos y puede habilitar a tomar algunas decisiones de preprocesado.

Un buen análisis multivariado es una tarea compleja aunque hay algunas técnicas estadísticas populares que facilitan la tarea. En este proyecto se han aplicado las siguientes técnicas a los datos de la muestra:

- **Análisis correlativo:** Se ha creado una matriz de correlación de *Pearson* [8] y se han detectado variables con correlación muy alta por lo que también se han probado métodos de reducción de dimensionalidad.
- **Análisis de componentes principales (PCA [9]):** Con las correlaciones detectadas, se ha estudiado y aplicado esta técnica de reducción de dimensionalidad y se ha obtenido que el 90% de la variabilidad de la muestra, puede ser explicada mediante las 6 primeras componentes principales.
- ***Clustering*:** Para estudiar la distribución de los datos, se ha estudiado y aplicado el algoritmo *K-Means* [10]. Mediante el análisis de silueta se ha detectado que la mejor distribución es agrupando los datos en 2 clústeres. Aplicando previamente *PCA*, se han mejorado los resultados obtenidos.

Este análisis ha sido muy importante para el modelado debido a que se han observado las relaciones internas en los datos y el comportamiento en distintos algoritmos no supervisados.

3.3. Preprocesado

A partir del análisis de datos realizado, se han considerado diversas técnicas generales de preprocesado para mejorar la calidad de los datos y garantizar la compatibilidad con algunos modelos.

En el contexto de este trabajo, los datos provienen de sensores de una célula que podría estar parada o tener problemas de comunicación, situación en la cual se podrían producir datos nulos. Aunque la muestra extraída en el análisis no indique valores nulos, es muy importante incluir este paso en el preprocesado porque es esperable la posible pérdida de los sensores o la aparición de errores de lectura. El criterio para tratar los datos nulos ha sido eliminar las columnas donde el 70% o más de los datos sean nulos. En el resto de los casos, se interpolan linealmente los datos.

La reducción de ruido, en este caso, no es buena idea debido a que uno de las tareas es encontrar las inconsistencias de comportamiento y estas inconsistencias, podrían confundirse con ruido.

3.4. Modelado

Una vez hecho el análisis de datos y el preprocesado, se han creado modelos que intentan responder a las hipótesis planteadas.

Dentro del aprendizaje automático, los dos enfoques clásicos son el aprendizaje supervisado y el no supervisado, aunque hay más formas como el aprendizaje semisupervisado y el aprendizaje por refuerzo.

- **Aprendizaje supervisado**: Este enfoque, normalmente, tiene los datos etiquetados (se conoce la entrada y la salida de cada ejemplo o observación) con los que se entrena un modelo. El modelo, aprendido con un algoritmo que ha identificado patrones en los datos, se utiliza para predecir la etiqueta en una nueva observación.
- **Aprendizaje no supervisado:** Este enfoque no tiene los datos etiquetados por lo que se basa solamente en ellos para realizar predicciones. Las técnicas más habituales son las regresiones y el *clustering*.

En este proyecto se crean dos modelos no supervisados para la detección de anomalías (clustering) y predicción a futuro (regresión).

3.4.1. Predicción temporal (Regresión)

La aplicación principal para implementar el mantenimiento predictivo es la predicción temporal del comportamiento de los sensores. Predecir el comportamiento de los sensores puede ayudar a evitar prevenir fallos o comportamientos no deseados.

En este proyecto se han estudiado varios modelos de predicción temporal. Se ha tenido en cuenta rendimiento, escalabilidad y eficacia para determinar el mejor modelo. Los principales modelos probados son los vectores autoregresivos y las redes neuronales. Las redes neuronales han proporcionado mejores resultados. Además, se trata de un modelo más flexible y escalable para el tipo de problema que se quiere solucionar. Se trata de un modelo que es capaz de adaptarse a prácticamente cualquier problema gracias a su algoritmo de reducción de error (*backpropagation* [11]) en el entrenamiento.

Este modelo puede ser descrito como un conjunto de neuronas artificiales que reciben datos de entrada y devuelven valores formados por combinaciones lineales con pesos. La arquitectura de estos modelos puede incluir capas intermediarias que proporcionan la capacidad de dar respuesta a problemas no lineales.

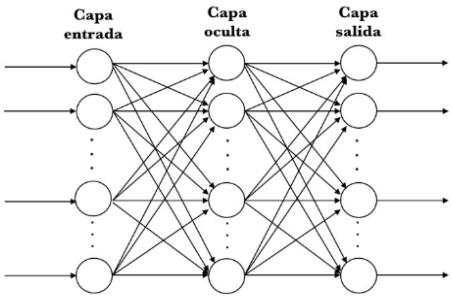

Fig. 5. *Arquitectura genérica de una red neuronal* [12]

Hay varias clases y arquitecturas de redes neuronales. En este proyecto se han escogido las **redes neuronales recurrentes (RRN)** que implementan una memoria interna para recordar información pasada gracias a que contienen neuronas que implementan conexiones cíclicas, es decir, que se utilizan predicciones antiguas en el input. Esta naturaleza la hace una arquitectura ideal para la tarea que se quiere realizar, que es predecir el funcionamiento de la máquina en base a observaciones antiguas.

Dentro de las redes neuronales recurrentes hay una variante especial muy popular llamada **LSTM** (*Long Short Term Memory*) [13]. Este tipo de redes se caracteriza por tener mecanismos que simulan una memoria y son capaces de recordar muchos pasos en el pasado.

En este proyecto, el objetivo es poder predecir el comportamiento de todos los sensores con una ventana de tiempo a futuro. Se trata de una tarea compleja puesto que se quiere obtener un modelo que sea capaz de predecir varios pasos en el futuro de todas las variables a la vez. Teniendo en cuenta estas condiciones, es importante recalcar que no hay unas reglas fijas para obtener buenos resultados modelando. Esto hace que el modelado sea una tarea que requiere intuición, experiencia y mucho prueba y error.

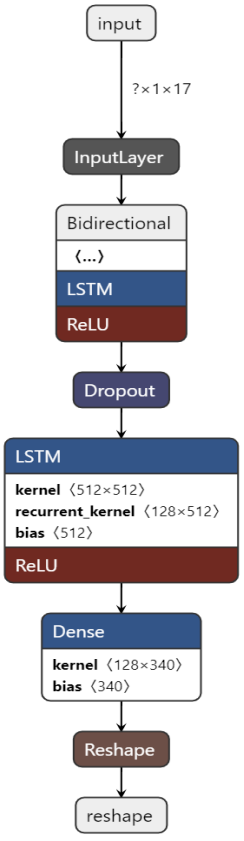

Fig.6. Arquitectura del modelo final

El modelaje, en este caso, ha consistido en encontrar modelos base de partida e, iterativamente, ir añadiendo complejidad al modelo y evaluando, hasta encontrar un modelo que mantenga una buena relación entre resultados y cumpla las condiciones establecidas.

Para la implementación de la red neuronal, se ha utilizado el módulo `Keras`, una API de alto nivel que forma parte de la biblioteca `TensorFlow` (Google). El módulo `Keras` contiene las clases necesarias para crear diferentes modelos utilizando distintas arquitecturas.

En la Figura 6 se puede observar que el modelo se ha entrenado de manera que recibe 1 paso en el tiempo con 17 variables y es capaz de predecir **20 pasos en el futuro** de estas.

No ha sido posible ver más al futuro debido a problemas con el tiempo de inferencia. No obstante, con mejores capacidades de procesamiento se podría intentar mirar más a futuro.

3.4.1. Detección de anomalías (Clustering)

Otra de las aplicaciones que se ha implementado en este proyecto es la detección de anomalías o comportamientos inconsistentes de la célula de inyección.

Para la detección se ha propuesto un enfoque multivariado tratando todos los sensores a la vez, verificando si la combinación de éstos es inconsistente en relación a lo observado.

Teniendo en cuenta, principalmente, el estudio multivariado realizado previamente, se han estudiado diferentes modelos para la detección no supervisada de anomalías. Finalmente, el modelo escogido se basa en una combinación de *PCA* y *clustering*. Para la implementación del modelo se ha utilizado el módulo `adtk` [14]. Se ha utilizado la herramienta *Pipenet* que permite combinar transformadores, modelos y agregados para formar un modelo más complejo. En la Figura 7 se puede visualizar el *pipenet* creado:

Fig.7. *pipenet* generado que combina PCA + K-means

3.5. Evaluación

Al ser modelos no supervisados, la tarea de evaluación es más ambigua que al tratar con modelos supervisados. Esto es debido a que se suele tener en cuenta el comportamiento teórico de los modelos para evaluarlos.

Para evaluar el modelo de predicción al futuro, el método que mejor ha permitido visualizar los resultados ha sido utilizar *walk validation* conjuntamente con la métrica de error cuadrático medio (*RSME*) [15]. Concretamente, se ha

recorrido el conjunto de test y se ha ido prediciendo 20 pasos al futuro punto a punto calculando la métrica en el proceso.

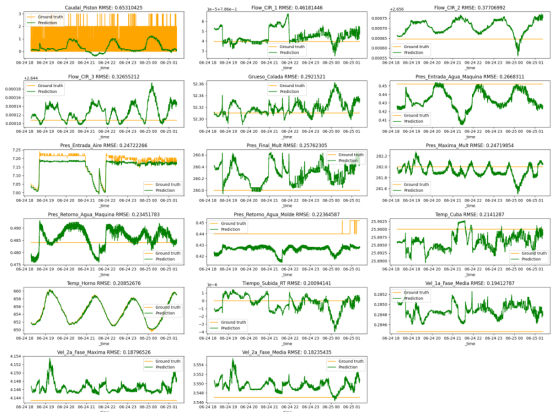

Fig.8. *Gráficos generados mediante el método walk validation sobre el conjunto de test*

En la Figura 8 se pueden observar los valores predichos (verde) y los observados (amarillo). Los resultados, en general, han sido buenos. No obstante, se puede ver que el comportamiento predicho de los sensores no coincide en muchos casos. Aunque son los mejores resultados obtenidos, todavía hay mucho camino de mejora a recorrer.

En cuanto al modelo de detección de anomalías, la selección de parámetros se ha realizado mediante el estudio multivariado explicado anteriormente. La evaluación de los resultados se ha hecho graficando las anomalías detectadas en el conjunto de datos.

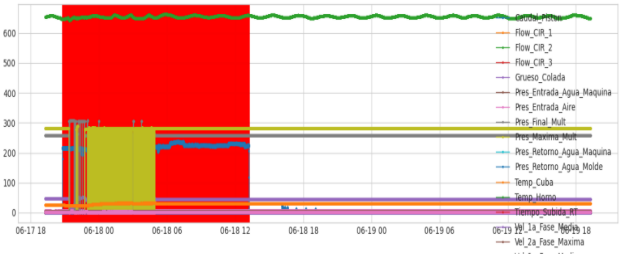

Fig.9. *Resultado de detección de anomalía multivariada sobre una muestra de 24h*

En la Figura 9, se puede decir que el modelo funciona correctamente. No obstante, su eficacia dependerá mucho de la muestra con la que se haya entrenado.

4. Desarrollo y despliegue aplicación

Con los modelos creados y evaluados, el siguiente paso es crear la aplicación que se encargará de recoger los datos, tratarlos y, finalmente, poder visualizar los resultados.

Las herramientas principales para el desarrollo de la aplicación han sido las siguientes:

- **Node-RED**: Se trata de un framework basado en *Node.JS* que se basa en programación de flujos de mensajes y nodos. Con esta herramienta es posible integrar hardware, *APIs* y funcionalidades de una manera muy accesible. La parte del cliente web se ha creado mediante el complemento *UIBuilder* [16] utilizando el framework web *VueJS* [17].
- **Python:** Para el tratamiento de los datos, *Python* será la herramienta principal. Es un lenguaje *open source* con una gran cantidad de módulos para todo tipo de aplicaciones. Es la herramienta preferida para tareas de ciencia de datos e inteligencia artificial.
- **InfluxDB**: Base de datos de series temporales (*Time Series Database*, *TSDB*). Este tipo de bases de datos están optimizadas para las series temporales. Mediante el concepto de "cubos", permite almacenar grandes cantidades de datos de forma eficiente teniendo la posibilidad de especificar un tiempo de retención para cada cubo. El tiempo de retención indica que cada medición, pasada ese tiempo, es eliminada.
- **Docker:** Se trata de una aplicación que permite el despliegue de aplicaciones dentro de contenedores, proporcionando una capa de virtualización, permitiendo que las aplicaciones se puedan aislar y automatizar.

La arquitectura utilizada combinando todas las herramientas anteriores es la que se muestra en la Figura 10.

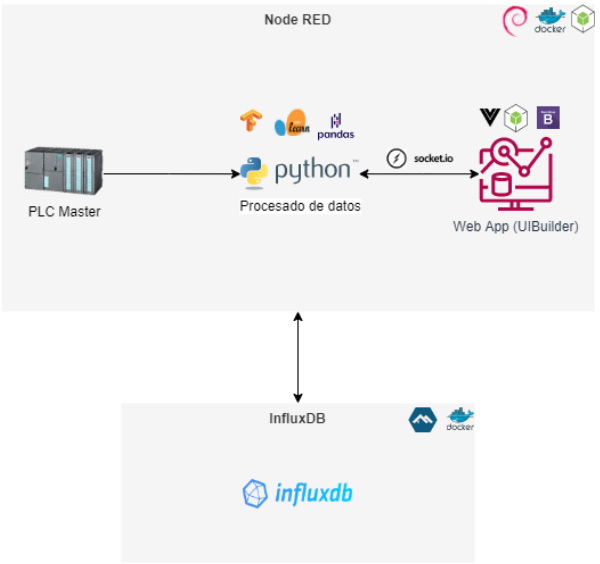

Fig.10. *Diagrama de la arquitectura de la aplicación final*

Como se puede observar en el diagrama de la Figura 10, hay dos contenedores (servicios) que contendrán las diferentes funcionalidades de la aplicación. Cada contenedor tiene su propio sistema operativo.

El contenedor de *InfluxDB* se ejecuta en el puerto 8086 y será donde se almacenarán los datos de los sensores y servirá para realizar consultas y para entrenar los modelos.

Por otra parte, el contenedor de *Node-RED* se ejecuta en el puerto 1880 y contiene toda la lógica e integra todos los elementos de la aplicación. Como se puede ver en el mismo

diagrama, desde *Node-RED* se lee la información del *PLC*, se realizan todos los procesos de minería de datos, se guardan los resultados en la base de datos y también se comunica con la aplicación web que permite visualizar los datos.

Es importante destacar que la comunicación entre *Node-RED* y la aplicación web se realiza mediante *sockets* web (Socket.IO [18]). Esto permite visualizar los datos en tiempo real sin tener que pasar por la base de datos.

El resultado del desarrollo es una aplicación web donde se puede realizar un análisis histórico de los resultados de los modelos y también se puede realizar una monitorización en tiempo real donde también se incluyen los resultados de los modelos (Figura 11).

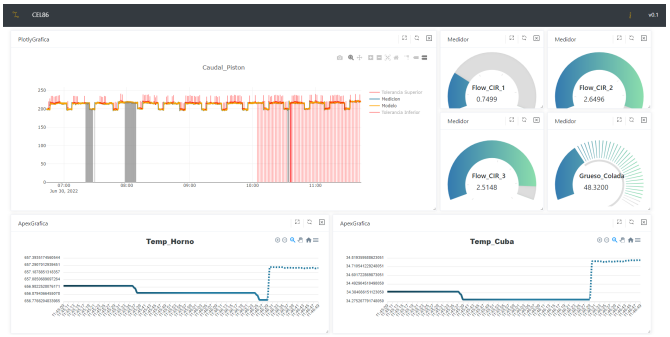

Fig.11. *Dashboard web final de la aplicación*

5. Trabajo para futuro

Aunque en el proyecto se han descrito e implementado las diferentes etapas de la metodología de minería de datos, todavía hay mucho margen de mejora en cada una de ellas.

Estudiar detenidamente la fundición a presión y también trabajar conjuntamente con ingenieros de producto, puede enriquecer mucho el análisis de datos y preprocesado. Concretamente, en el enfoque multivariado del análisis, quedan todavía pendientes muchas técnicas y enfoques por estudiar.

En cuanto al modelaje, hay todavía mucho margen de mejora e investigación para optimizar los modelos actuales. El modelo de regresión está basado en *Deep learning*, una rama del aprendizaje automático (*machine learning*) muy poderosa dónde actualmente hay mucha investigación. Mejorar el modelo de regresión en sí es un reto que daría para varios proyectos nuevos. Por otra parte, el modelo de detección de anomalías también puede ser mejorado añadiendo más casos de detección y trabajando conjuntamente con ingeniería para supervisar la efectividad del modelo obtenido.

Finalmente, la aplicación también tiene un largo recorrido por delante. Aunque, de momento, es una aplicación de análisis de datos sobre una ventana reducida de tiempo, es posible ir añadiendo variabilidad con más sensores, piezas y células. También se pueden añadir módulos y sistemas de manera que sea una aplicación que incluya sistemas de control de supervisión y control del proceso.

6. Conclusiones

En este proyecto se han obtenido dos modelos de naturaleza diferente y con resultados también distintos. El modelo de regresión para predicción al futuro, ha supuesto un problema muy complejo y los resultados indican que todavía hay mucho margen de mejora para ser capaz de predecir el comportamiento de los sensores. Por otra parte, en el modelo de clasificación para detección de anomalías se han obtenido mejores resultados teóricos. Aun así, será importante mejorarlo y verificar su efectividad conjuntamente con los ingenieros de producto.

Generalmente, la parte de minería de datos requiere de mucha más iteración para obtener mejores resultados. Estudiar mejor los fundamentos y explorar con mayor profundidad las técnicas disponibles será fundamental para las decisiones técnicas. Por otra parte, entender mejor el proceso y trabajar conjuntamente con los ingenieros de proceso permitirá un análisis mucho más enriquecido y, consecuentemente, mejores resultados.

En cuanto la parte de la aplicación, aun siendo muy limitada al escenario, implementa toda la parte de minería de datos estudiada y se trata de un buen punto de partida para el análisis de los parámetros de proceso. En algunas decisiones de diseño se han tenido en cuenta factores de escalabilidad por lo que futuramente se podrían añadir más sensores, células, piezas o modelos, por ejemplo.

Desde una perspectiva global, el proyecto ha resultado en una prueba de concepto o punto de partida para la implementación de mantenimiento predictivo. Aunque los resultados no han sido excelentes, el proceso seguido ha sido todo un reto donde he aprendido mucho en todos los aspectos.

Finalmente, quiero decir que ha sido una gran oportunidad para iniciarse en la minería de datos. Trabajar en un escenario real me ha enseñado el potencial que tiene la disciplina y me ha empujado a entrar en el Máster de Ciencia de Datos en la Facultat d'Informàtica de Barcelona.

7. Gestión del proyecto

Toda la gestión del proyecto se ha realizado mediante la herramienta *Notion*. Se ha creado inicialmente un *dashboard* [19] donde se han ido anotando las etapas, tareas, referencias, roles, recursos, etc.

La temporalidad de este trabajo está concentrada entre los meses de febrero y julio de 2022. Al ser un proyecto realizado mediante convenio de cooperación educativa para prácticas académicas externas, ha sido posible aprovechar las horas en la empresa para llevarlo a cabo. El convenio realizado tenía fecha de inicio el día 25 de febrero de 2022 y fecha de finalización el día 14 de julio de 2022, con un total 484 horas. Aparte de las horas en la empresa, también se han realizado horas extra fuera de ella para poder llegar a cumplir los objetivos. La mayoría del tiempo invertido en esas horas extra ha sido dedicado a aprender o a documentar. La gran parte del desarrollo y proceso de minería de datos se ha

realizado en la empresa. El total de horas realizadas ha sido de **590 horas**.

Fig.12. *Proporción de tiempo dedicado a cada etapa*

De la Figura 12 se puede extraer que las etapas de análisis, procesado, modelaje y evaluación constituyen el 33% del trabajo y la parte de la aplicación es del 35% de las horas totales, por lo que hay un cierto balance de dedicación a los dos objetivos principales. Por otra parte, se ha requerido el 32% para el estudio y realización de documentación.

En cuanto a la gestión económica del proyecto, en cada tarea se ha ido apuntando el rol realizado y mirando el salario base de cada rol [20] para obtener una estimación de los costes humanos. Por otra parte, también se ha tenido en cuenta los costes de recursos (*hardware* y *software*). El total del proyecto ha sido la suma de los costes humanos y de recursos (Figura 13).

Rol	Salario base (€/hora)	Horas realizadas	Costes totales (€)
BackEnd Developer	18,75	110	2062,5
Data Analyist	16,42	58	952,36
Data Scientist	19,21	79	1517,59
Front End Developer	16,37	53	867,61
Full Stack Developer	16,82	30	504,6
ML Engineer	23,77	22	522,94
Project Manager	18,72	10	187,2
Student	0	228	0
TOTAL		590h	6614,8€

	Recurso	Precio
Hardware	PC gama media	400€
	Servidor	1.200€
Software	Python	0€
	Node-RED	0€
	Docker	0€
	InfluxDB	0€
TOTAL		1.600€

Costes	Precio Total
Humanos	6.614,80€
Recursos	1.600€
TOTAL	8.214,80€

Fig.13. Tablas para la estimación económica del proyecto

Agradecimientos

Primero quiero agradecer a mi familia y a mis amigos por su constante apoyo y cariño. Son mi fuente de inspiración y energía.

Agradecer especialmente a Neus Català i Roig por dirigir este trabajo. Sus consejos, seguimiento y dedicación han sido clave para la realización de este proyecto.

También agradecer a Ángel Avilés por darme esta oportunidad y codirigir este trabajo y a Sebastian Sarrias por poner a disposición los recursos que necesitaba y sus consejos en diseño web.

Por último, agradecer a la *UPC* y a *CIE Automotive* por enseñarme tanto en tan poco tiempo.

Referencias

[1] Apartado de CIE C.Vilanova en la web de CIE Automotive https://cieautomotive.com/-/cie-vilanova

[2] Definición y características de mantenimiento predictivo https://www.predicagroup.com/blog/predictive-maintenance/

[3] Conceptos básicos de ayuda y ejemplos de CRISP-DM https://www.ibm.com/docs/es/spss-modeler/SaaS?topic=dm-crisp-help-overview

[4] Libreta Interactiva utilizada para el análisis y modelado https://colab.research.google.com/drive/1v7YI_yHnaAYRnBX9SnEULuJzoBqDPqAH?usp=sharing

[5] Teoría de los modelos ARIMA Y SARIMA con ejemplos https://neptune.ai/blog/arima-sarima-real-world-time-series-forecasting-guide

[6] Definición y teoría del Augmented Dickey-Fuller Test https://es.wikipedia.org/wiki/Prueba_de_Dickey-Fuller_aumentada

[7] Referencia al apartado de vectores autorregresivos (memoria) https://docs.google.com/document/d/1jM6q_rTnAV86NFboC-aGgZZnWB5c_Ia-/edit#heading=h.bj69ym8z8wwx

[8] Teoría y definición del coeficiente de correlación de Pearson https://es.wikipedia.org/wiki/Coeficiente_de_correlaci%C3%B3n_de_Pearson

[9] Teoría y ejemplos de Análisis de componentes Principales https://www.cienciadedatos.net/documentos/35_principal_component_analysis

[10] Teoría, definición del algoritmo de clusterning K-Means https://en.wikipedia.org/wiki/K-means_clustering

[11] Algoritmo Backpropagation para Redes Neuronales https://en.wikipedia.org/wiki/Backpropagation

[12] Jordi Torres. Introducción práctica con Keras [en línea] 2018. https://torres.ai/deep-learning-inteligencia-artificial-keras/

[13] Explicación de Redes Neuronales tipo LSTM https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[14] Referencia a los detectores del módulo adtk de Python https://adtk.readthedocs.io/en/stable/api/detectors.html

[15] Definición y explicación de la métrica error cuadrático medio https://acolita.com/que-es-el-error-cuadratico-medio-rmse

[16] Referencia al módulo UIBuilder de Node-RED https://flows.nodered.org/node/node-red-contrib-uibuilder

[17] Referencia al framework web VueJS (v2) https://es.vuejs.org/v2/guide/index.html

[18] Referencia a Socket.IO (Sockets WEB) https://socket.io/

[19] Dashboard creado para la gestión del proyecto (Notion) https://aacraf.notion.site/TFE-b75f1fd04ea64fc2a85d2989df0576d9

[20] Referencia a la plataforma utilizada para consultar sueldos https://www.glassdoor.es/Sueldos/index.html