

UniversityHack 2024

Datathon



UNIVERSIDAD
DE GRANADA



UNIVERSITYHACK 2024[™]
DATATHON

21/11/2024

EQUIPO: DataRunnersGR

Álvaro Zorrilla Carriquí

Físico y Matemático

Máster en Ciencia de Datos e IA

Carlos Sánchez Muñoz

Ingeniero Informático y Matemático

Computer & Data Engineer

en Vircell S.L.

Introducción

El reto [UniversityHack 2024 Datathon](#) propone una tarea de predicción de la concentración en el antígeno final de una vacuna tras la conclusión de todas las fases de producción, explicadas en la web del reto. Para abordarlo, se ha desarrollado y entrenado un modelo de predicción para regresión, debido a la naturaleza numérica y continua de la variable objetivo.

El proceso de fabricación del antígeno consta de varias fases, en las que están involucradas distintos procesos, técnicas, materiales y factores. Este procedimiento se esquematiza en el siguiente diagrama para su mejor comprensión.

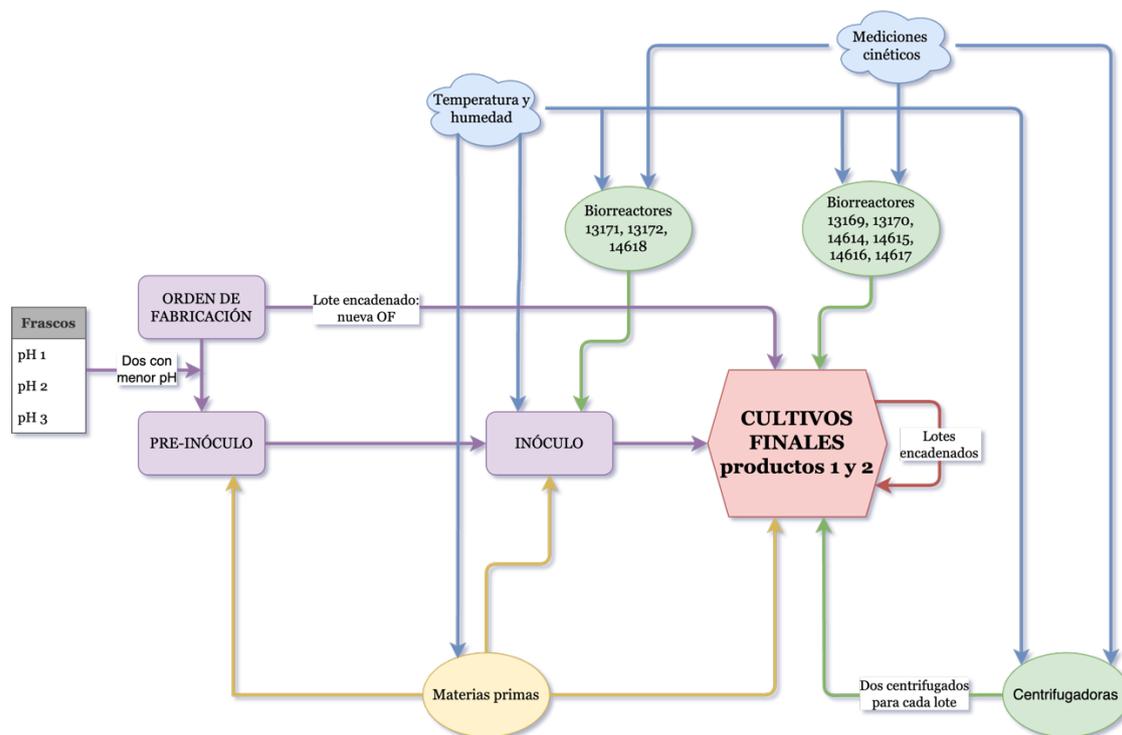


Figura 1. Diagrama del proceso de fabricación del antígeno.

A la hora de enfrentar este reto, distinguimos varias tareas. En primer lugar, se ha llevado a cabo un análisis exploratorio de todos los conjuntos de datos (EDA) proporcionados de forma individual, para así comprender las distintas variables involucradas en el proceso, cómo se relacionan entre sí y si existen errores en dichos conjuntos.

Posteriormente, se ha optado por un enfoque unificador de todos los *datasets*, de forma que, para cada lote de antígeno fabricado, tuviéramos la mayor cantidad de información recogida sobre todo el proceso de fabricación. De este modo, se obtienen 161 lotes o instancias de entrenamiento y 55 lotes o instancias de test. Sobre el conjunto de entrenamiento final, se ha realizado un breve EDA para corregir posibles valores perdidos, normalizar las variables y reducir la dimensionalidad, en busca de una mayor simplicidad y eficiencia computacional.

Finalmente, se han entrenado diversos algoritmos de aprendizaje supervisado para problemas de regresión como regresión lineal, modelos lineales generalizados, k-NN, máquinas de soporte vectorial para regresión, *Random Forest*, redes neuronales y *XGBoost* con un enfoque de *Cross-Validation* (CV).

También se consideró la posibilidad de discretizar la variable a predecir, y considerar el problema como uno de clasificación.

Metodología

Para este reto, se emplea como lenguaje de programación R, debido a su potencial y funcionalidad para resolver problemas de índole estadística y de ciencia de datos. En el trabajo se distinguen dos partes que han sido desarrolladas secuencialmente: el EDA y el entrenamiento de modelos de inteligencia artificial (IA). La gestión en equipo de todas las tareas de este proyecto se ha organizado bajo un enfoque ágil mediante el uso de [Trello](#).

Análisis Exploratorio Inicial

El punto inicial del trabajo ha consistido en un EDA individual para cada uno de los archivos dados, con el fin de comprender mejor el problema a resolver, detectar valores perdidos, así como posibles *outliers* y analizar las distribuciones de cada variable. Este proceso permite además esclarecer cómo se relacionan las variables de los distintos conjuntos de datos entre sí. Especialmente se ha estudiado la distribución de la variable a predecir, la cual se observa en la *Figura 2*, con el fin de sopesar una posible discretización de dicha variable que mejorase los resultados de los algoritmos. Señalar además que la concentración de antígeno del Producto 1 sigue una distribución normal, corroborada con el test de Shapiro.

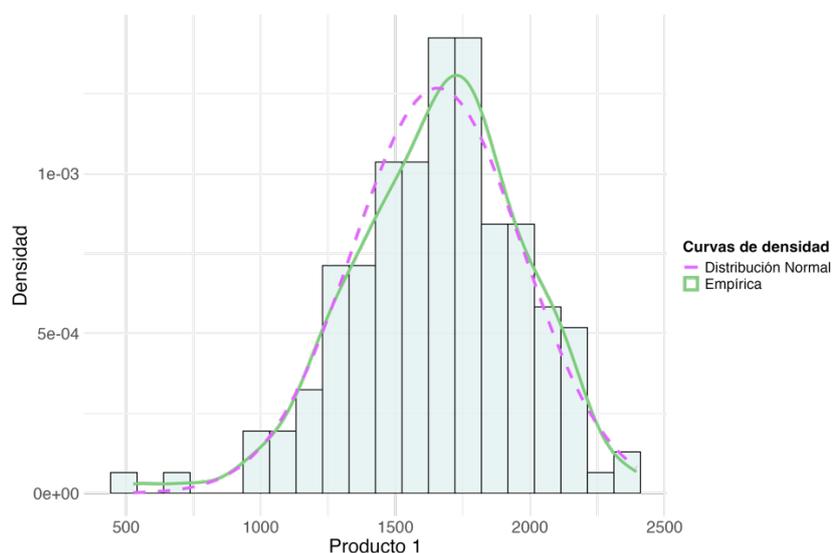


Figura 2. Distribución muestral de la variable a predecir (Producto 1). Se observa que la distribución presenta un notable parecido con una normal, lo cual ha sido corroborado mediante el test de Shapiro-Wilk.

Uno de los primeros pasos es formatear correctamente cada uno de los atributos del conjunto de datos, estableciendo a “NA” aquellos valores de los que no se dispone y puedan venir almacenados con otra configuración. Cada tabla se lee en formato *tibble* mediante la librería *readxl*, lo cual optimiza la lectura del fichero y el manejo en memoria de la información.

Una vez leídos los datos, se crearon varios *dataframes* que extrajeran la información relevante de los biorreactores y de las centrifugadoras según el lote que estuviera en ese momento. Además, para los

materiales empleados, se calculó la cantidad total de material por cada lote y el tiempo de almacenado del mismo.

En cuanto al tratamiento de los valores perdidos, se ha tomado como medida de imputación emplear, bien, la media de la columna correspondiente para variables numérica; o bien la moda, para variables categóricas, dado que eliminar los registros con valores perdidos llevaría una gran pérdida de información; y la imputación por k-NN no resultó adecuada ya que otros registros tenían valores perdidos en otras variables, lo cual afectaría a la imputación. No obstante, para el caso de los datos de temperatura y humedad de los distintos emplazamientos de fabricación, la imputación se ha hecho tomando el valor anterior y posterior en la misma columna, debido a la naturaleza continua de las variables temperatura y humedad.

Respecto al tratamiento de los *outliers*, no se llevó a cabo ningún tratamiento específico, debido al desconocimiento de la naturaleza de estos, salvo el caso de las variables que miden el pH, las cuales sabemos que deben tomar valores entre 0 y 14. Por lo que registros con valores fuera de ese rango fueron descartados.

Por último, se realizaron algunos gráficos para comprobar distintos comportamientos y correlaciones entre variables, no obteniendo ningún resultado destacable. Todo ello se puede encontrar en el *script_exploracion.qmd*.

Mergeado y análisis exploratorio final

Para conseguir juntar toda la información disponible sobre cada lote en cada una de las distintas etapas del proceso de fabricación del antígeno, se han “mergeado” los distintos datos en función del lote. En aquellos casos en los que había un conjunto de valores por lote, se ha optado por aplicar ingeniería de características (*feature engineering*), tomando la media, la desviación típica y el último valor que tomaba dicha variable, ya que consideramos plausible pensar que el valor final de una variable al terminar un proceso pueda influir en la variable objetivo.

Con todo ello, se obtuvo un *dataframe* total que contenía una fila por cada lote producido, y al que se realizó un EDA. Se imputaron los valores perdidos por la media/moda de la columna (según si la variable es numérica o categórica respectivamente). Las variables no numéricas que hacen referencia al ID de los biorreactores y de la centrifugadora empleados en la fabricación han sido descartadas. Según el algoritmo de predicción a entrenar, se llevó a cabo una selección de características mediante *Random Forest* para reducir la dimensionalidad de los datos (otros algoritmos como *Random Forest* o *XGBoost* ya implementan dicha selección). Por último, para las variables categóricas (almacenadas con un formato de tipo *factor*) se ha optado por un enfoque de *One-Hot-Encoding*.

Modelos de predicción

Una vez finalizado el preprocesamiento de los datos, aplicamos distintos algoritmos de aprendizaje supervisado para predecir el valor de la concentración del producto. El enfoque empleado para el entrenamiento y evaluación de los modelos fue la **validación cruzada** con 5 *folds* repetidos 3 veces. Este método implica dividir el conjunto de datos en cinco subconjuntos de tamaño aproximadamente igual. Para cada iteración, se utiliza un *fold* como conjunto de prueba y los otros cuatro como conjunto de entrenamiento, de forma que todos los datos se usen tanto para entrenar como para validar. Al repetir

este proceso tres veces con diferentes particiones aleatorias, se mejora la robustez de los resultados al reducir la variabilidad causada por una única partición. Esta aproximación otorga una mayor estabilidad en las métricas de evaluación, mayor generalización y reducción de sesgos. No obstante, para algunos algoritmos como XGBoost, no se pudo usar este enfoque debido a la falta de su implementación para R. Sobre los parámetros escogidos para los modelos, se optó por un *grid search*, que puede consultarse en el script *script_prediccion.qmd*. De este modo, el flujo de trabajo resultante es el siguiente:

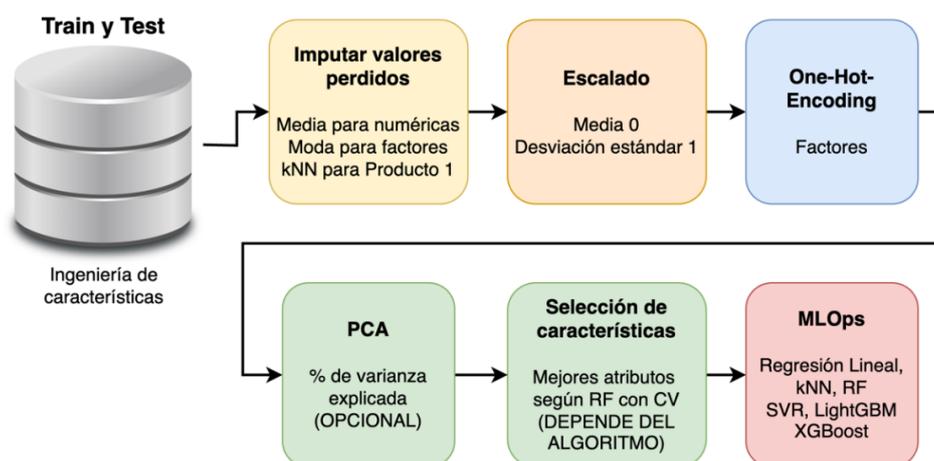


Figura 3. Flujo de trabajo de los datos (Data Workflow) en la tarea de predicción.

Los resultados recogidos tras los envíos al servidor de *Codabench* para conocer el error cuadrático medio sobre el conjunto de test se han recogido en la Tabla 1. Obsérvese cómo la evaluación en test permite establecer el algoritmo idóneo así como probar diferentes configuraciones para escoger la que provea un mejor rendimiento.

Tabla 1. Valores del RMSE para el conjunto de test según el algoritmo y la configuración empleada.

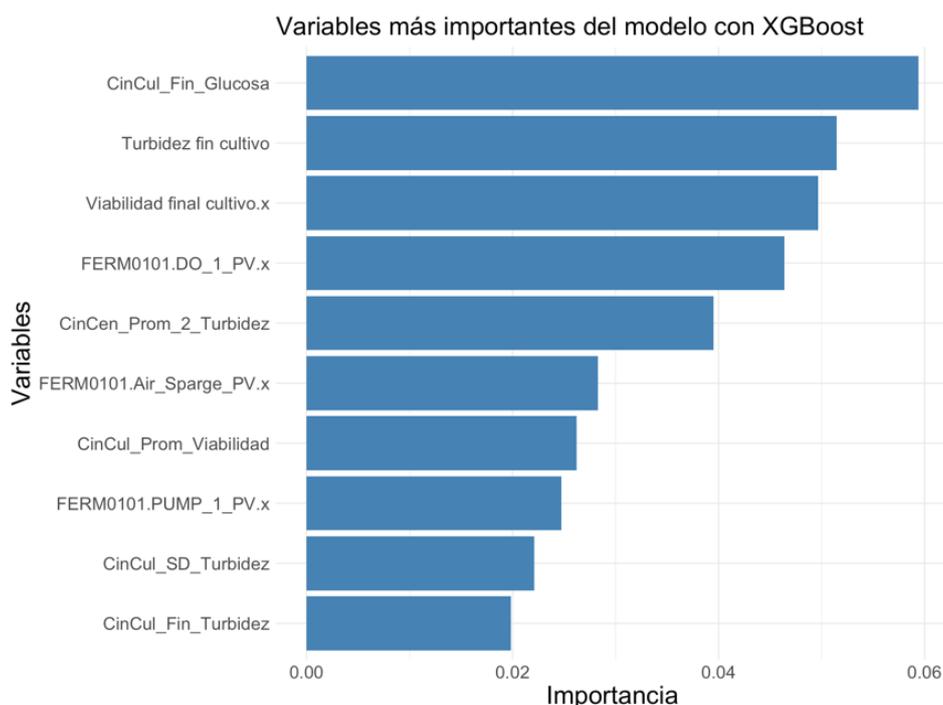
Envío	Algoritmo	PC	RMSE	Envío	Algoritmo	PCA	RMSE test
1	Random Forest	NO	253,72	11	XGBoost	NO	257,34
2	SVR	NO	259,23	12	XGBoost	NO	249,19
3	XGBoost	NO	224,12	13	XGBoost	NO	229,08
4	XGBoost	SI	232,71	14	XGBoost	NO	237,57
5	keras_selection	NO	ERROR	15	XGBoost	NO	219,30
6	keras_selection	NO	ERROR	16	XGBoost	NO	224,91
7	Stacking	-	234,65	17	XGBoost	NO	219,12
8	XGBoost	NO	239,83	18	XGBoost	NO	219,54
9	XGBoost + Feature	NO	260,80	19	XGBoost	NO	218,89
10	XGBoost	NO	255,94				

Entre un amplio conjunto de algoritmos, entre los que tenemos k-NN, regresión lineal, *Random Forest*, XGBoost y redes neuronales, entre otros, elegimos aquellos que se muestran en la tabla anterior (RF, XGBoost, SVR y redes neuronales) debido a su explicabilidad y a sus buenos resultados sobre el conjunto de entrenamiento/validación, para ver su rendimiento sobre el conjunto de prueba. De entre ellos, debido a su potencia, optamos por centrarnos en XGBoost, ajustando los valores de sus parámetros para obtener el mejor rendimiento, tanto para entrenamiento/validación, como para test.

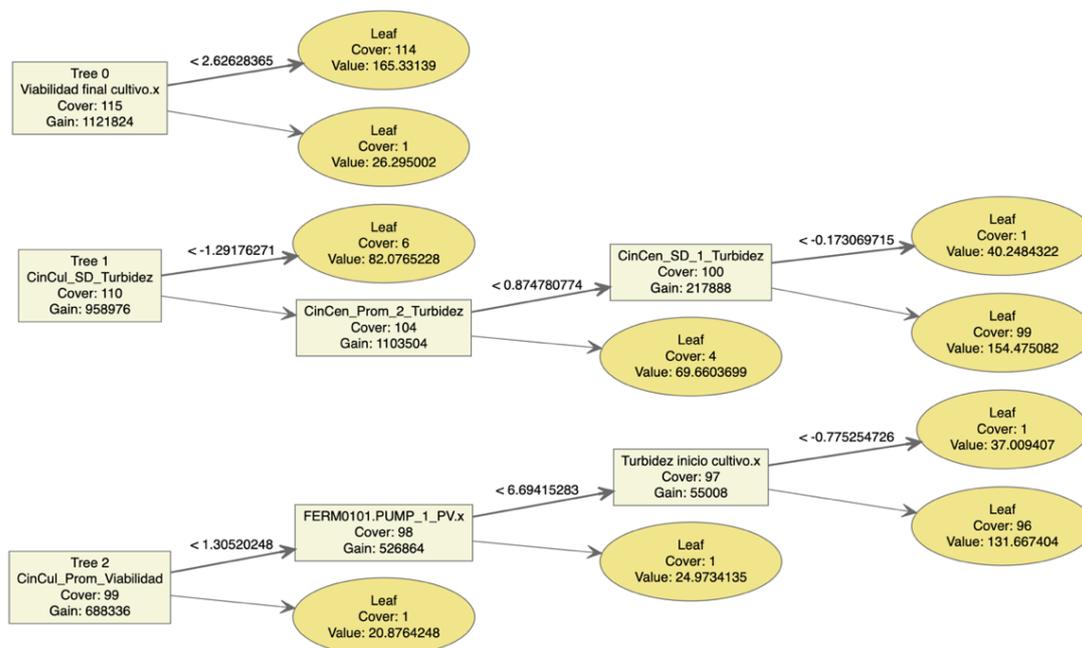
Nota: Se ha intentado resolver este problema como un problema de clasificación, discretizando la variable objetivo en grados de concentración de antígeno. Posteriormente, se asignaría el punto medio del intervalo como representante de la clase para el cálculo del RMSE. Tras realizar pruebas con varios algoritmos y 10 clases, los resultados han sido deficientes, por lo que se descartó ese enfoque.

Modelización responsable

La elección del modelo, **XGBoost**, no solo se ha limitado a su precisión en las predicciones, sino que también a la interpretabilidad que puede ofrecer al usuario final. Este modelo contribuye a la **explicabilidad** y **transparencia** debido a que no es un modelo de caja negra, como sí podría considerarse a las redes neuronales, por ejemplo. De hecho, su implementación sobre R nos permite conocer qué variables del conjunto de entrenamiento considerado son más influyentes a la hora de la predicción, e incluso explorar qué árboles de decisión se construyen, como se muestra en la Figura 3 generada con el paquete *xgboost* de R.



(a) Las quince variables más importantes según el mejor modelo de XGBoost entrenado



(b) Visualización de los tres primeros árboles de decisión creados por el modelo de XGBoost. En cada uno se observa la variable del criterio, el criterio de decisión y su ganancia asociada.

Figura 4. Explicabilidad del modelo de XGBoost escogido en cuanto a (a) el análisis de las variables más influyentes en la predicción, y (b) la transparencia mediante la visualización de los primeros árboles de decisión

Por tanto, la propuesta metodológica para la predicción de antígeno otorga **explicaciones en 4 ejes**:

1. **Datos.** El proceso de mergeado y cálculo de algunos estadísticos sobre los conjuntos aportados, siguiendo la naturaleza del proceso de producción.
2. **Modelo.** Es inherentemente explicable por estar basado en árboles.
3. **Post-hoc.** Métodos de visualización que permiten explorar el funcionamiento de la arquitectura ajustada.
4. **Evaluación.** Confianza por parte de los usuarios en las predicciones debido a que pueden obtener el camino en los árboles y a que se han realizado pruebas sobre el conjunto de test.

En el desarrollo del modelo de IA, hemos velado por el principio de **justicia**, enfrentando las limitaciones propias al problema biotecnológico abordado. Aunque en este caso no contamos con variables protegidas específicas, se ha prestado especial énfasis a la **representatividad de los datos** y se ha trabajado para mitigar sesgos relacionados con la composición de la muestra. Dado el tamaño reducido de la base de datos, se ha intentado que los datos para el entrenamiento de los modelos abarquen adecuadamente la variabilidad presente en el fenómeno estudiado, **sin sesgos** que pudieran favorecer un subconjunto de muestras o excluir patrones relevantes. Para reforzar la equidad en la evaluación, se han usado técnicas de **validación cruzada** para maximizar la **equidad en la evaluación**, evitar que la escasez de datos afectara negativamente al desempeño y optimizar el comportamiento justo del modelo. De este modo, hemos validado que el proceso se ha llevado a cabo siguiendo un enfoque equitativo y riguroso.

Asimismo, hay que señalar que este desarrollo se ha llevado a cabo bajo un enfoque de **green computing**, con un compromiso firme hacia la **sostenibilidad** ambiental y la eficiencia energética. En este sentido, hemos seleccionado un algoritmo, XGBoost, considerado como **eficiente** y portátil, asegurando así un procesamiento eficiente con un consumo controlado de recursos. Además, para evitar

una demanda excesiva de energía, el código ha sido optimizado mediante el uso de la librería *dplyr*, aprovechando su capacidad para procesar datos de manera vectorizada y eficiente, evitando bucles innecesarios. Esta **optimización** reduce significativamente el tiempo de ejecución y, por ende, el consumo energético, sin comprometer la precisión del modelo.

También se ha optado por librerías oficiales y ampliamente reconocidas, que cuentan con optimizaciones internas, lo que contribuye aún más a la eficiencia computacional. Todo el desarrollo es compatible con la ejecución en equipos estándar, **eliminando la necesidad de infraestructura en la nube** y, con ello, reduciendo la huella de carbono. Este enfoque sigue la filosofía de *tiny machine learning*, que promueve modelos de IA eficientes en términos de energía y recursos, haciéndolos más sostenibles y accesibles para dispositivos de menor potencia.

Por tanto, considerando los motivos anteriormente expuestos, se propone este diseño como una potencial solución al reto planteado. Su implementación en la industria sería altamente viable gracias a un enfoque de modelización responsable, fundamentado en principios de explicabilidad, transparencia, equidad y sostenibilidad.