

# UNIVERSITYHACK 2024 - Equipo Named17

Ana Porras Garrido y Pedro José Lucas Guillén

*Nuestro trabajo ha consistido fundamentalmente en construir un gran número de características que potencialmente pudieran explicar la variación de la producción del antígeno en los distintos lotes de muestra. Se han construido casi 600 features distintas sobre las que hemos realizado un filtrado y una selección de las más relevantes, teniendo en cuenta además la interacción entre ellas. Nuestra propuesta final es un modelo de regresión, ElasticNet, que se basa en 13 características extraídas del proceso de producción. Son tres indicadores del final de cultivo, turbidez, glucosa y viabilidad, el peso que permanece en el biorreactor para el cultivo siguiente, cuatro medidas de los biorreactores, de inóculo y cultivo, y 5 relacionadas con condiciones medioambientales y materiales. El modelo desarrollado obtiene un error cuadrático medio de 216,53 en el conjunto de test.*

## DATASETS

El primer paso realizado fue un análisis de todos los datos suministrados orientado fundamentalmente a encontrar las **relaciones entre los distintos datasets**. Cada instancia queda definida por los lotes de fabricación. Nuestro punto de partida fue el subdataset **Cultivo**, en Fases de Producción, que contiene, además de características relativas a la fase de cultivo del lote, el dato a predecir para el test, Producto 1. Este conjunto original lo hemos relacionado con el resto de datasets, de la siguiente manera:

- Preinóculo e Inóculo: Son las fases de producción anteriores y, al igual que Cultivo, contienen un registro por lote. No todos los lotes tienen registro en estas fases, pues algunos cultivos se inician partiendo de cultivos de otros lotes parentales. Será necesario rellenar estos datos para obtener información de las fases iniciales de los lotes dependientes.
- OF123456: Contiene un registro por lote, que se alinea con Cultivo, relación 1:1. El Número de orden contenido en este conjunto será fundamental para obtener fechas de operación de centrifugas por lote.
- Cinéticos IPC: Contiene medidas de las fases de Inóculo, Cultivo y Centrifugación. En esta ocasión aparecen varios registros por lote que se agregan para construir un único registro por lote.
- Biorreactores: Cada lote tiene un campo de ID Biorreactor que indica la máquina en la que se realizó el Inóculo y un segundo ID de la máquina de Cultivo. Con este identificador se accede al subconjunto de series temporales que corresponde a cada lote. Además, se usan las fechas de inicio y final para acceder al fragmento necesario.
- Horas inicio fin centrifugas: La relación se obtiene con el Número de orden contenido en OF123456. Definen los intervalos de tiempo de centrifugación de los lotes. Algunos datos faltantes han sido aproximados teniendo en cuenta las fechas de las otras partes del proceso y el tiempo que de media transcurre entre ellos.
- Centrifugas: La relación se obtiene a través del lote que lleva a una ID de centrifuga (dataset Cultivo) y usando también los datos de Horas de inicio y fin de centrifuga. Con el identificador y el intervalo de tiempo se accede a las series temporales de operación de las centrifugas para cada lote.
- Movimientos componentes: Contiene varios registros por lote y material que hay que agregar. Los consideraremos para valorar si las cantidades de cada material o las condiciones de su almacenaje pueden influir en el resultado de la producción. Los materiales pasan por dos almacenes, el principal y el de producción.
- Temperaturas y humedades: Los datos ambientales suministrados para las distintas salas, son series temporales de las que extraemos los periodos de cada fase de fabricación de cada lote, así como las fechas de almacenaje de los materiales necesarios para la fabricación de cada lote.

## GENERACIÓN DE FEATURES

**Lotes parentales:** Hemos comprobado que todos los lotes independientes, i.e. sin lote parental, tienen datos de preinóculo e inóculo, el orden en el encadenado es 1, el tiempo que transcurre entre el inicio del cultivo y el final del cultivo anterior en el mismo biorreactor es suficientemente largo y que entre el final del cultivo previo y el inicio del independiente se vacía el reactor (variable Load\_Cell\_Net\_PV).

Por el contrario, en los dependientes o lotes con parental, no existen los datos de las fases previas, el encadenado es 2 ó 3 (y coincide con el encadenado del parental +1), transcurren solamente unas 3 horas desde el lote anterior en el mismo biorreactor y el biorreactor no se vacía en el tiempo que sucede entre ambos. Además, el biorreactor coincide con el del parental. En algunos lotes particulares se han corregido incoherencias en los identificadores de biorreactor. Para estos lotes dependientes, se construyen los datos de preinóculo e inóculo duplicando los de sus parentales, primero los de orden 2 y a continuación los de orden 3.

Del análisis de dependencias obtuvimos nuevas variables: el volumen en el reactor que un lote parental deja a sus dependientes y su variación porcentual, el peso que reciben los dependientes de sus parentales y el tiempo que transcurre desde que se finalizó el lote anterior hasta que se inicia el cultivo del lote.

**Preinóculo:** Se sustituyen las medidas de turbidez y pH de las tres líneas cultivadas por el resumen (media) de exclusivamente las líneas que se han utilizado en el inóculo.

**Biorreactores:** Se han sumado las variables (DO\_1, DO\_2) y (pH\_1, pH\_2), dando lugar a las nuevas DO y pH, porque parece claro que son complementarias, como si se tratara de dos sondas de medida.

**Catégoricas:** Se añaden como variables catégoricas los identificadores de biorreactores y centrifugas.

**Fechas:** Añadimos como nuevas variables las duraciones de todos los pasos de la producción del lote: preinóculo, inóculo, cultivo y centrifugación. Y también el tiempo transcurrido entre cada fase y la siguiente.

**Materiales:** Para cada lote y material, sumamos las cantidades (variable Qty) para crear las variables cantidades. Calculamos además la proporción de cada una teniendo en cuenta la cantidad total de materiales en cada lote. Por otro lado, utilizamos las fechas de recepción y traslado para definir la duración del periodo que cada uno de los materiales pasa en el almacén principal para cada lote. Algunos de los datos se han corregido cuando demostraban incoherencias. El tiempo en el almacén secundario, el de producción, se aproxima restando la fecha de final de cultivo del lote y la de traslado. Añadimos además el número de días desde la recepción del material hasta el final del cultivo.

**Agregaciones temporales:** Son dos diferentes:

1. Cinéticos IPC: Como se dispone de muy pocas medidas, no una serie temporal, como agregación hemos utilizado el primer dato, el último, la media, la variación entre el principio y el final.

2. Biorreactores, Centrifugas y Ambientales: Decidimos usar 4 parámetros de caracterización: la media de cada variable en el período temporal obtenido, la desviación típica, la media de la variación y la desviación típica de la variación. Las variables ambientales, temperaturas y humedades, se usan teniendo en cuenta las diferentes estancias por las que circulan los materiales y lotes:

- Almacén principal: Agregación por material y lote entre la recepción y el traslado.
- Almacén de producción: Agregación por material y lote entre el traslado y el final del cultivo.
- Sala de biorreactores: Agregación por lote para el periodo de preinóculo, el de inóculo y el de cultivo.
- Sala de centrifugas: Agregación por lote para el periodo de centrifugación.

La consecución de los puntos anteriores, sumado a las características ya existentes en los datasets originales, da lugar a un total de **563 potenciales features** para el problema de predicción.

## ELIMINACIÓN DE INSTANCIAS

Se ha descartado el uso de algunas de las instancias proporcionadas en el conjunto de train:

- Lotes sin output (Producto 1 faltante): 24026, 24034, 24039, 24048.
- Lotes que no están en la orden de fabricación: 23052, 23053, 23066, 23074, 24002, 24029, 24043.
- Lotes con datos incoherentes: Del estudio de dependencias se encontraron dos lotes que no se pudieron resolver dado el número de incongruencias entre fechas y el ID de los reactores asociados, estos fueron el 24011 y 24015.

## VALORES FALTANTES

En aquellas variables puntuales que no se disponía de información originalmente hemos probado distintos medios de imputación: usando la media de la columna, usando la mediana de la columna o usando imputación mediante los K vecinos más próximos (KNN Imputer,  $k=4$ ). KNN Imputer lo hemos probado para hacer la imputación de los datos de cada fase basada en el resto de variables de la misma fase. Aunque es cierto que KNN obtiene valores más realistas en los lotes, introduce una gran complejidad, pues la imputación de una característica requiere del cálculo de otras muchas variables. Este inconveniente no nos ha parecido que estuviera justificado por la mejora en el resultado final. Además, no era solución para aquellos lotes en los que no se disponía de ningún dato de la fase. Por este motivo, hemos optado por un Simple Imputer, utilizando la media de cada feature.

Al agregar los conjuntos de Cinéticos IPC se obtienen características que ya presentes en los conjuntos de Fases<sup>1</sup>. Hemos aprovechado esta redundancia para rellenar valores faltantes, antes de aplicar el Simple Imputer. En las etapas finales de la construcción de un modelo, sólo se mantiene una, pues filtraremos las redundantes.

---

<sup>1</sup> Esta comparación nos ayudó a detectar una desalineación en el conjunto de Test V2.

Tanto la imputación, como los procesos posteriores de filtrado y selección de features se realizan **utilizando únicamente el conjunto de train**. Sólo así podemos asegurar que el modelo construido, en caso de obtener buenos resultados, sería susceptible de poner en producción para estimar futuros lotes. Hemos comprobado que omitir cierta información que deriva del conjunto de test supone una desventaja, pero hemos priorizado la integridad del procedimiento.

## ESTANDARIZACIÓN DE FEATURES

Se han estandarizado las features sólo en los casos en los que era necesario, bien por funcionamiento coherente del modelo, bien por interpretabilidad. La estandarización se hace usando sólo el conjunto de train. No se modifican las variables binarias.

## SELECCIÓN DE FEATURES

Dado el alto número de potenciales predictores, se requiere de un proceso puramente automático que sea capaz de identificar las características relevantes y no redundantes.

Como hemos mencionado, para evitar data leakage durante el filtrado y la selección de características, eliminamos todas las instancias de los lotes del conjunto de test.

**Filtro:** El primer paso es la eliminación de variables utilizando algunos filtros básicos:

1. Variables que no aportan información para la correlación como fechas, Número de orden y similares (filtro manual).
2. Variables invariantes en todos los lotes, o bien con el mismo valor para todos o con un valor que se repite en el 90% de los casos.
3. Variables con más de un 25% de datos faltantes.
4. Variables redundantes, con correlación mayor que 0.9, en valor absoluto. Se utiliza la correlación de Spearman para evitar distorsiones en la de Pearson producidas por outliers. El orden de eliminación lo establecemos en función de la correlación en valor absoluto con la variable a predecir, Producto 1.

La criba inicial mantiene aún **355 características**. De ellas, un 25% aproximadamente muestran una correlación en torno a 0.2 o superior con el output.

**Selección:** Para seleccionar las más relevantes, hemos probado con varios algoritmos, basados fundamentalmente en la correlación con el output y en la eliminación de redundancia. Se trata de 6 algoritmos similares:

1. Se ordenan las variables por correlación de Spearman con el output. Se pasa al modelo un número fijo de features (5, 10, 20 ó 30).
2. Igual que el anterior, pero para cada una seleccionada se descartan las que tienen una correlación de Spearman superior a un límite de redundancia (0.6 ó 0.8).
3. Se ordenan las variables por su significancia estadística con respecto a esa variable objetivo usando el p-valor de un modelo de regresión lineal simple (se diferencia de 1. si usamos variables con datos faltantes).
4. Selección por p-valores en regresión múltiple. En cada paso se selecciona la variable si su p-valor es bajo en la regresión del output junto con el resto de variables explicativas previamente seleccionadas. Así, se incluyen en la propia selección las posibles interacciones entre variables. Se utiliza un límite de 0.05.
5. Igual que el anterior, pero en cada adición de variables se revisan los p-valores de las previamente seleccionadas. En este caso, como la selección es más restrictiva, ampliamos el límite a 0.15.

El algoritmo número 6, seleccionado finalmente, es una variación del 4. en el que hemos separado conceptualmente las variables. En primer lugar, se realiza la selección sobre todas las características, salvo las de ambientales y materiales y después se continúa evaluando las pertenecientes a estos dos conjuntos. Nuestra motivación para establecer este criterio de separación es la naturaleza de las características ambientales y materiales. Entre ellas, las más relevantes son las basadas en fechas. El conjunto de train y el de test son disjuntos en cuanto a temporalidad, todos los lotes de train son anteriores a los de test.

Ambientales y Materiales pueden contener diferencias grandes entre train y test y no hay manera de evaluar si esas diferencias son relevantes para resolver el problema. Por una parte, queríamos mantener el uso de la totalidad de los datasets proporcionados, y por otra parte, consideramos arriesgado basar el modelo en gran medida en features temporales cuando train y test son disjuntos en este sentido. Este último algoritmo busca el equilibrio entre los dos extremos.

## MODELO DE PREDICCIÓN

Las pruebas de modelos se hacen utilizando **validación cruzada, de 5 splits y con shuffle**, barajamos los datos para garantizar heterogeneidad en los datos de manera que se facilite la capacidad de generalización de los modelos.

En el primer paso, probamos diferentes algoritmos de Machine Learning variando el conjunto de features con las selecciones comentadas en el apartado anterior. Se han probado **RandomForest, XGBoost, SVM y ElasticNet**<sup>2</sup>. Originalmente, elegimos los árboles por su explicabilidad en términos de importancia de características y porque no era necesario entrar en profundidad en la imputación y estandarización de las features. Sin embargo, pensamos que podría ser un problema que no manejaran bien los datos fuera del rango de entrenamiento, por lo que decidimos explorar también modelos con capacidad de extrapolación, que podría hacer falta para la predicción del test. SVM se probó porque funciona bien en problemas de baja dimensionalidad y ElasticNet por su simplicidad, que lo hace mucho más explicable y rápido.

Para darnos una referencia sobre cuál sería el máximo error admisible, añadimos además un modelo benchmark que usaba la media para realizar predicciones sobre la variable objetivo.

---

<sup>2</sup> Hiperparámetros:

RandomForest: max\_depth=3, max\_features=1, n\_estimators=100

XGBoost: max\_depth=3, n\_estimators=100, learning\_rate=0.05, alpha=0.5

SVM: C=0.5, kernel='rbf'

ElasticNet: alpha=1, l1\_ratio=0.5



La selección final del modelo tuvo en cuenta los siguientes factores:

- Poder de predicción, medido por el RMSE (Root Mean Square Error) obtenido en media de los 5 splits de la validación cruzada.
- Poder de generalización de la solución, medido por la desviación estándar del error entre los 5 splits y sobre todo la diferencia en el error entre la media del error en train y en el test de los 5 splits.
- Explicabilidad del modelo. Se considera la sencillez del modelo a la hora poder entender la importancia de la aportación de cada variable a las predicciones.
- Economía de computación. Se considera el posible tiempo de computación en entrenamiento y predicción del modelo, muy asociado con la sencillez del mismo.

En la tabla 1 se muestran los mejores resultados de cada tipo de modelo.

	Error (avg)	Error (std)	Error (train - test)
Benchmark	313,16	32,13	4,72
RandomForest	267,60	29,48	51,15
XGBoost	248,59	35,16	176,28
SVM	309.07	34,33	1,21
ElasticNet	244,61	34,34	29,89

Tabla 1. Modelos probados y sus mejores resultados

A la vista de los resultados mostrados y aplicando los criterios comentados anteriormente decidimos usar como nuestro modelo para obtener la mejor solución posible para el problema planteado el **ElasticNet usando variables seleccionadas mediante p-values en dos etapas**. ElasticNet como metamodelo, que usa las características de los modelos lineales Ridge y Lasso con sus capacidades de regularización, tenían un error sobre test muy bueno y lo complementaban muy bien con una diferencia de errores entre train y test muy ajustada lo que indica una buena capacidad de generalización, y en cuanto

a explicabilidad y tiempo de cálculo no tienen rival.

Los motivos de exclusión del resto de modelos son, en el caso de los basados en árboles, la poca capacidad de generalización observada, pese a ajustar los parámetros para reducir overfitting; y, en SVM, no conseguimos un aprendizaje adecuado, además de que el modelo resultaría poco interpretable.

## MODELO PROPUESTO

Una vez decidido el modelo, contemplamos algunas posibilidades más antes de dar la solución por definitiva.

Inicialmente el modelo utilizaba 18 características. No obstante, algunas de ellas tenían un coeficiente final en la regresión muy bajo, por lo que contemplamos la opción de reducir el número de features. Como el resultado seguía siendo igualmente bueno, concluimos que la complejidad de un mayor número de variables no se justificaba y planteamos un nuevo candidato basado sólo en 13 features.

Sobre esta nueva versión, comprobamos qué variación de las métricas se producía al variar los hiperparámetros de ElasticNet, alpha y l1\_ratio. Aumentar alpha y reducir l1 devolvía modelos con error mayor y sobre-ajuste menor y reducir alpha y aumentar l1, modelos error menor y sobre-ajuste mayor, más o menos en la misma medida. Sin un punto de equilibrio claro, decidimos mantener los parámetros originales.

Tras estas últimas decisiones, nuestro modelo final es una ElasticNet con parámetros alpha=1, l1\_ratio=0.5, basado en las 13 características que se definen a continuación:

1. amb\_cultivo\_H\_bios\_std: Desviación estándar de la Humedad en la sala de producción durante el periodo de cultivo.
2. biocultivo\_ID Bioreactor\_13170: Indica si el cultivo se realiza en el biorreactor con número de identificador 13170.
3. biocultivo\_PUMP\_2\_PV: Promedio de la adición solución base durante el periodo de cultivo.

4. bioinoculo\_Air\_Sparge\_PV\_std: Desviación estándar del aporte de aire por sparger durante la fase de inóculo.
5. bioinoculo\_Gas\_Overlay\_PV\_ret\_std: Desviación estándar de las diferencias del aire por cúpula durante el inóculo.
6. cultivo\_Glucosa g/L\_fin: Indicador de glucosa al final del cultivo.
7. cultivo\_Turbidez fin cultivo: Indicador de turbidez al final del cultivo.
8. cultivo\_Viabilidad\_fin: Indicador de viabilidad al final del cultivo.
9. dependencias\_peso\_siguiente: Volumen que se preserva en el biorreactor para realizar el cultivo del siguiente lote.
10. materiales\_traslado\_100008: Tiempo que pasa el material 100008 en el almacén de producción.
11. amb\_almacen\_2\_by\_material\_T\_alma\_prod\_ret\_100005: Diferencia media de temperaturas en el almacén de producción que se produce mientras se almacena el material 100005.
12. amb\_almacen\_2\_by\_material\_T\_alma\_prod\_ret\_100012: Diferencia media de temperaturas en el almacén de producción que se produce mientras se almacena el material 100012.
13. amb\_almacen\_by\_material\_T\_alma\_princ\_100004: Temperatura media en el almacén principal mientras que se almacena el material 100004.

En la Figura 1 se presentan los coeficientes en el modelo final de cada una de las variables. En la Figura 2 se muestra la comparativa gráfica entre predicciones y el valor real de Producto 1. Se aprecia un muy buen comportamiento en general salvo algunos puntos extremos. La correlación es de 0,65.

## SCORE

El resultado en el conjunto de test obtenido a través de Codabench ha sido de **216,53**.

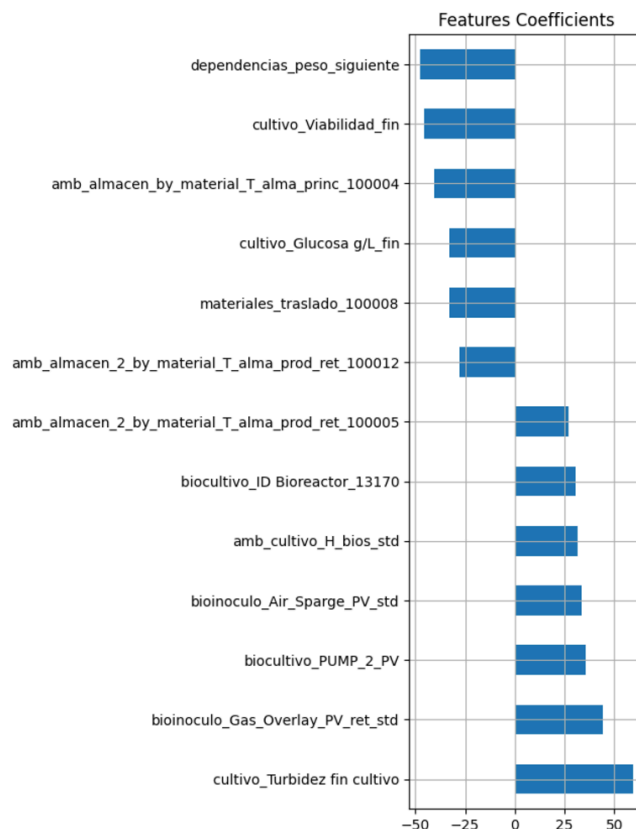


Figura 1. Variables seleccionadas y sus coeficientes

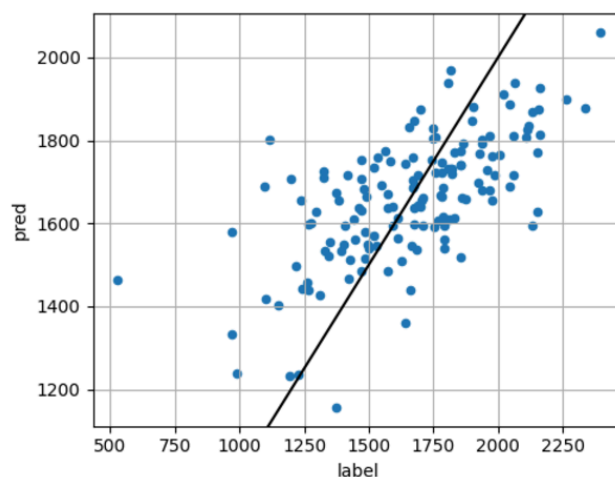


Figura 2. predicciones vs Producto 1 (out of sample)

## EXPLICABILIDAD

El modelo elegido es 100% interpretable. Sus predicciones se explican de manera sencilla ya que la aportación a la predicción de cada una de las variables se ve determinada por su coeficiente en la ecuación lineal del modelo. En la Figura 2 se muestran gráficamente estos coeficientes.

## TRANSPARENCIA

En el presente estudio hemos tratado de explicar resumidamente, pero sin obviar

detalles, cómo ha sido la metodología completa de resolución del problema y las motivaciones que nos han llevado a tomar las decisiones ejecutadas durante todo el proceso.

Se adjunta además el código, muy extenso, pero suficientemente organizado e hilado con este documento para facilitar su uso. Se incluyen un breve README con las instrucciones de uso, la versión de Python utilizada y la lista de librerías utilizadas también con sus versiones.

## JUSTICIA

Durante la selección del modelo final, se ha prestado atención a la distribución de errores para comprobar si existían sesgos en el modelo. En particular, hemos valorado los siguientes:

- Temporal: Los lotes se pueden ordenar por fecha de inicio. Comprobamos que el error se distribuye homogéneamente a lo largo del tiempo. No hemos encontrado ningún modelo que pareciera tener sesgo en este sentido.
- Parentales: Comprobamos que el error es independiente de que el lote sea parental o no lo sea. Los modelos que no incluían suficiente peso en características del subconjunto de dependencias sí presentaban diferencias relevantes en los errores. El modelo seleccionado asigna un peso importante a la variable `dependencias_peso_siguierte`, que indica que el lote es parental.
- Dependientes: Repetimos el mismo ejercicio esta vez distinguiendo si los lotes tienen parental o no. En algunos conjuntos de features sí se producía diferencia apreciable, pero no en el modelo final propuesto.

En la Figura 3 se muestra el resumen de este análisis de sesgos sobre los resultados del modelo final.

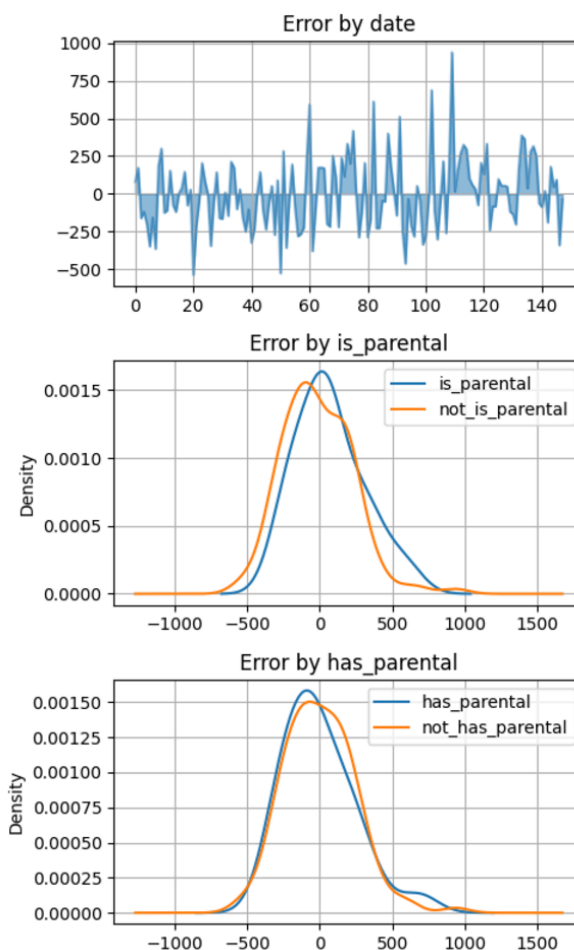


Figura 3. Análisis de sesgos en el modelo propuesto

## SOSTENIBILIDAD AMBIENTAL

El modelo es de computación muy rápida; al tratarse de un modelo de regresión lineal los tiempos de computación tanto en entrenamiento como en predicción son extremadamente bajos.

Aparte de la elección del modelo, se ha procurado simplificar los requerimientos en cuanto a cantidad de inputs necesarios:

- Eliminación de características innecesarias: una vez obtenido un modelo base se ha valorado si el resultado se preservaba y se eliminó una proporción importante de features, favoreciendo la simplicidad del modelo.
- Descarte de KNN Imputer: Análogamente, comprobamos que no había una mejora sustancial que justificara el requerimiento de muchas otras features con el objetivo de imputar los valores faltantes de las relevantes. Usamos la media como método suficientemente eficiente y robusto.