

RETO LA VIÑA WINE PREDICTION - LOS PALMERINES

1. Introducción

En esta sección se enumeran las principales etapas que hemos desarrollado a lo largo del hackathon y que serán descritas con mayor profundidad en los siguientes epígrafes.

El trabajo se ha dividido en dos partes claramente diferenciadas: en la primera se ha realizado un análisis previo y preprocesado de datos y en la segunda tareas relacionadas con la obtención del modelo final. Los pasos en la primera parte han sido:

1. Creación del primer modelo usando únicamente el dataset train, para poder realizar una comparativa a posteriori.
2. Análisis exploratorio de los datos, tanto del dataset train como de los otros dos de datos meteorológicos. Selección de variables relevantes.
3. Creación de un modelo simple utilizando las variables seleccionadas para establecer un nuevo baseline.
4. Creación de nuevas características para refinar los modelos.
5. Búsqueda bayesiana sobre el espacio de hiperparámetros para encontrar la mejor configuración de nuestro modelo.

Para la creación del modelo final se han seguido los siguientes pasos:

1. Entrenar un único modelo con todos los datos disponibles.
2. Entrenar un único modelo eliminando los valores outliers.
3. Entrenar múltiples modelos dotando a los datos anteriores a 2020 de superficie por inferencia de datos posteriores y eliminando aquellos que no puedan inferirse.
4. Lo mismo que el punto 3 pero incluyendo las variables relevantes a partir de las nuevas generadas del dataset meteo.
5. Lo mismo que el punto 3 pero incluyendo variables que recojan información histórica sobre la producción.

En función de los resultados obtenidos la opción 5 ha sido la mejor, pues es la que presenta menor error en el proceso de validación.

2. Análisis exploratorio de datos

Previo al análisis de los datos, hemos aplicado una recodificación habitual a las variables no numéricas: por un lado la **codificación one-hot** a las variables categóricas (*variedad*) al no presentar relación de orden y, por otro, una **codificación binaria** a las variables que solo tienen dos posibles valores (*color*, *tipo* y *modo*).

Tras un estudio previo de la relevancia de las variables basado en un análisis de correlación, se concluyó que las variables altitud y superficie influyen significativamente en la producción.

Sin embargo, ambas variables presentaban muchos valores nulos. Por lo tanto, se decidió crear nuevas variables para incluir esta información de la siguiente manera:

- Para cada finca se crean dos nuevas columnas *altitud_min* y *altitud_max*, con el mínimo y máximo de altitud, respectivamente. En caso de que el valor altitud sea nulo originalmente, dicho valor será sustituido por la media aritmética de otros años agrupando por estación meteorológica.
- Dado que sólo se disponía de los valores de superficie a partir de 2020, los valores de superficie de años anteriores se calcularon como la media de las superficies asociadas a cada finca, teniendo en cuenta la variedad, modo de cultivo y tipo y color de uva de la plantación. En 2022, hay nueve fincas de las que no se dispone información sobre la superficie y que no aparecen en ninguno de los años anteriores, por lo que se calculó su producción media a partir de lo que esas nueve fincas produjeron en años anteriores.

Para la entrega local analizamos los atributos del dataset train con el fin de determinar qué variables utilizar en los entrenamientos. Seguidamente, usamos el dataset meteo para generar métricas simples como precipitaciones y temperaturas medias, máximas y mínimas. Sin embargo, al realizar el análisis de atributos de estas nuevas variables junto con las de train, apenas presentaban relevancia. A pesar de ello, probamos a insertarlas en nuestro modelo con las de train, sin embargo el error aumentó considerablemente. Por este motivo, en la primera entrega no utilizamos ninguna información meteorológica.

De cara a la fase nacional, realizamos un análisis más profundo de la relevancia de cada una de las variables sin tener en cuenta ninguna de las creadas anteriormente a partir del dataset meteo. Con el fin de evitar posibles sobreajustes analizamos los atributos más relevantes mediante modelos de regresión **isolation forest** y el **coeficiente de correlación de Pearson**, obteniendo prácticamente el mismo resultado en ambos análisis. Así, seleccionamos 8 variables predictoras de entre las que sobresalen con diferencia *produccion_media* y *superficie*. (Figura 1)



Figura 1. Importancia de los atributos con Isolation Forest

3. Parametrización de los modelos

Una vez construido el conjunto de datos con las variables más significativas, al no disponer de los datos reales de la producción de 2022 para poder cuantificar la bondad de nuestras predicciones, definimos nuestro propio proceso de validación interna. Para ello, optamos por tomar el último año 2021 como test y los anteriores como entrenamiento.

En cuanto al modelo, la primera opción fue usar cuatro modelos de regresión **basados en árboles** con los valores de los parámetros por defecto para establecer un modelo base y poder comparar posteriormente. En una segunda etapa se implementó una **búsqueda bayesiana** sobre los cuatro modelos para optimizar sus parámetros. Esta búsqueda realiza un proceso de validación cruzada con los datos de entrenamiento de forma similar a una grid search, pero limitándose a un número máximo de modelos explorados. En nuestro caso la búsqueda se limitó a 400 posibles parametrizaciones por cada modelo. En la siguiente tabla se detalla el espacio de búsqueda y la mejor parametrización encontrada para el modelo **random forest** que presentó el menor error final:

Parámetro	Posibles valores	Mejor valor
bootstrap	{True, False}	True
max_depth	[10,110] con paso 1 (100 valores)	80
max_features	{auto, sqrt}	sqrt
min_samples_leaf	{1,2,4}	1
min_samples_split	{2,5,10}	5
n_estimators	[200,2000] con paso 18 (100 valores)	1000

Tabla 1. Grid search para el modelo RandomForest

Los mejores resultados para cada uno de los modelos se muestran en la siguiente tabla:

Modelo	Error RMSE
RandomForestRegressor	5772.92863834843
XGBRegressor	7374.909541541813
LGBMRegressor	5878.869834163633
CatBoostRegressor	6693.49886414591

Tabla 2. Errores de los cuatro modelos con la mejor parametrización

De esta manera el modelo presentado en la fase local obtuvo un RMSE sobre 2022 de 5.789,45. Como se puede comprobar, muy cercano al obtenido sobre 2021 en nuestro análisis previo.

4. Creación de nuevas variables

Después de lograr resultados prometedores en la fase local, identificamos una posible oportunidad de mejora mediante la aplicación de técnicas de "feature engineering". En concreto, decidimos crear una nueva variable que denominamos *produccion_media*, que resume la producción de cada finca en años anteriores mediante el cálculo de la media correspondiente. En los casos en los que no se dispone de valores de producción en años anteriores, se asignará a la variable el valor de la media correspondiente a las fincas que sí tengan datos disponibles. Esta estrategia permitió obtener información adicional relevante.

Una vez incluida *produccion_media*, se realizó el mismo estudio de relevancia aplicado en la fase local, obteniendo como variables más significativas: *produccion_media*, *superficie*, *modo*, *altitud_min*, *altitud_max*, *variedad_32*, *variedad_17*, *variedad_87*. La aplicación de los mismos cuatro modelos obtuvo una mejora del error sobre el análisis anterior superior al 5% para **random forest** y más del 15% para **XGB**, haciendo uso de la misma parametrización. En la Tabla 3 podemos observar los nuevos errores.

Modelo	Error RMSE
RandomForestRegressor	5475.194839645865
XGBRegressor	6257.578948770813
LGBMRegressor	5857.561153945297
CatBoostRegressor	6074.701899567947

Tabla 3. Errores de los cuatro modelos añadiendo *produccion_media*

En todas las pruebas realizadas para la selección de atributos tanto con las variables originales como con las variables construidas el mejor modelo ha sido mayoritariamente **random forest**.

5. Tratamiento del dataset meteo

Para la fase nacional, decidimos profundizar en la importancia de la información meteorológica haciendo un nuevo estudio sobre el dataset meteo. Después de la lectura del artículo [1] optamos por la generación de variables más complejas basándonos en la influencia de algunos índices meteorológicos extremos sobre el rendimiento de cultivos.

Con estas nuevas variables creadas a partir de los datos meteorológicos y con todos los atributos del dataset train se realizó un nuevo estudio de relevancia como los anteriores. Este análisis determinó que tres de las nuevas presentaban cierta relevancia: *GSP*, *damagedVineDays* y *longestDrySpell*. *GSP* (Growing Season Precipitation) es la media de las precipitaciones por año, *damagedVineDays* obtiene el número de días al año en los que la temperatura máxima fue mayor a 35 grados y *longestDrySpell* calcula el máximo número de días consecutivos por año en los que las precipitaciones que cayeron fueron inferiores a

1 mm. Hay que señalar que todas las nuevas variables utilizan exclusivamente los datos de los meses comprendidos entre marzo y junio (ambos inclusive) pues se pretende estimar la cosecha antes de que se inicie la recolección y deben ceñirse a los meses de crecimiento de la vid. Por este motivo y por tener solo datos de meteorología desde el penúltimo día de junio de 2015, el entrenamiento ha sido realizado usando datos desde 2016.

Como en pasos anteriores, una vez generadas las nuevas variables se comprobó su capacidad predictora entrenando y testando con los cuatro modelos de regresión citados. Los resultados, de nuevo, fueron peores que los obtenidos sin incluir esta información.

Finalmente, al no incluir información del dataset meteo, probamos a usar los datos de train de 2014 y 2015 en el entrenamiento. Sin embargo, el error seguía siendo mayor que sin ellos, así que tampoco se han incluido en el modelo final.

6. Explicabilidad

Debido a la importancia creciente que la explicabilidad de los modelos tienen para una Inteligencia Artificial más cercana al problema y el experto, nuestro equipo descartó el uso de modelos basados en **redes neuronales** por su difícil interpretación. Por este motivo optamos por el uso de modelos basados en árboles como **random forest**, que genera modelos de ensemble compuestos por varios **random tree**, que operan particionando recursivamente los datos en subconjuntos basándose en la importancia de cada uno de los atributos. Esta estructura jerarquizada nos permite realizar visualizaciones más nítidas de los modelos, facilitando su explicación (Figura 2).

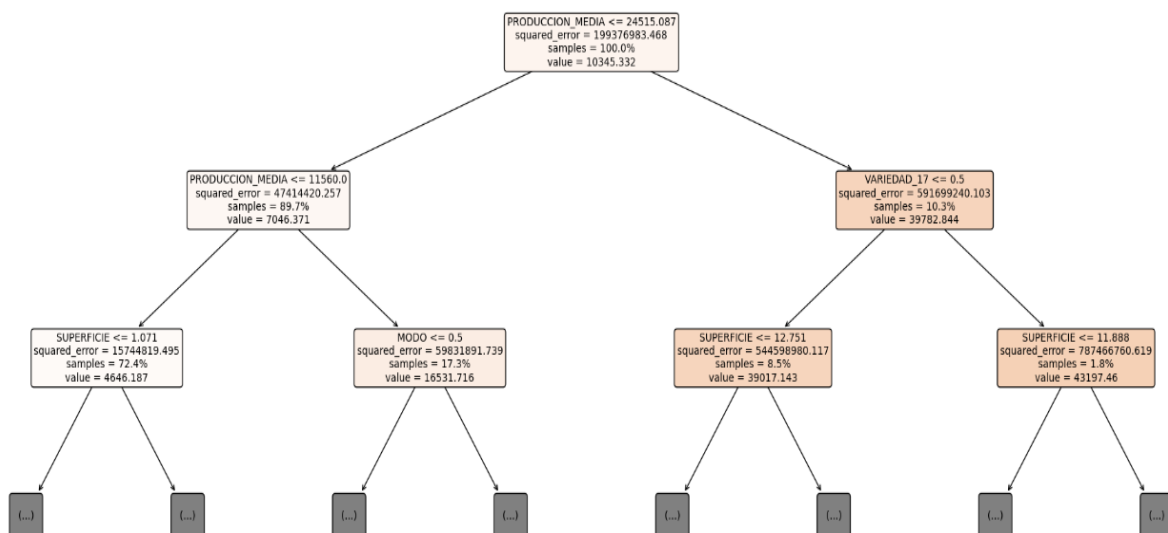


Figura 2. Primeros niveles de un árbol de un random forest

También es posible realizar una representación de la importancia asignada a cada una de las variables por el modelo (Figura 3), facilitando aún más su comprensión.

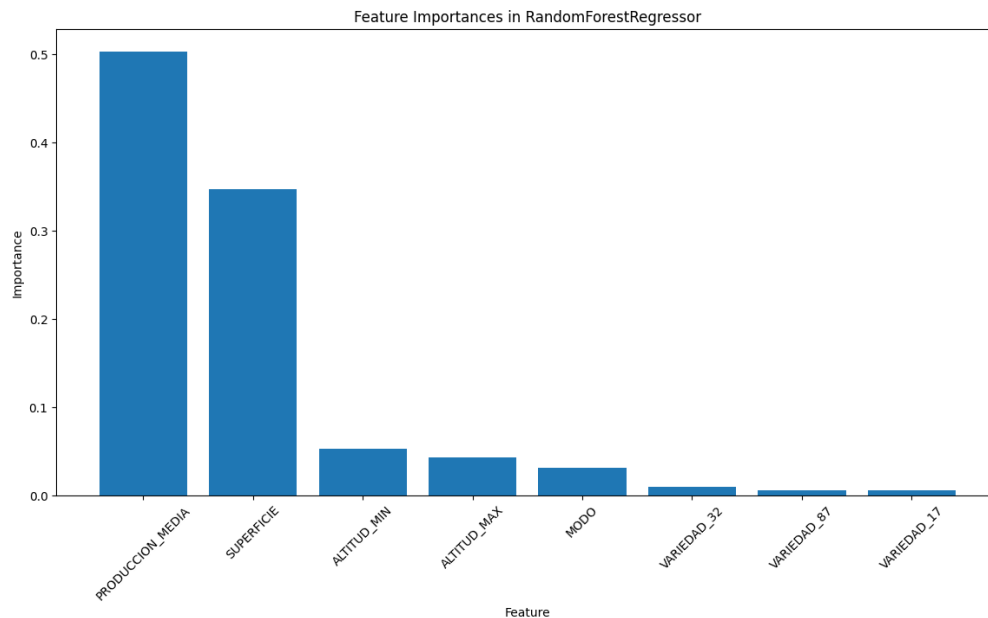


Figura 3. Importancia asignada por random forest a las variables

7. Transparencia

La transparencia de los resultados alcanzados se consigue mediante la aportación de todo el código usado para su desarrollo junto a este documento. Para la ejecución del modelo y la obtención del dataset de predicción simplemente se debe ejecutar el fichero ***prediccion.py***. En cuanto al procesado de los datos, todo el código de este tratamiento se encuentra recogido en el archivo ***prediccion_utils.py***, que recibe los datasets originales y se encarga de todo su procesamiento. Además, también se encarga de dividirlo en los conjuntos de entrenamiento (datos hasta 2021) e instancias a predecir (datos de 2022). Una vez generadas todas las variables seleccionamos únicamente las más significativas. Todo el proceso de selección de variables está recogido en el fichero ***EDA.ipynb***.

8. Justicia

Desde un principio se ha buscado un modelo lo menos sesgado posible. Desde el punto de vista de los datos, en todo momento se ha procurado eliminar el mínimo número posible de filas, ya que esto aumentaría el sesgo. Asimismo, desde el punto de vista del modelo, las arquitecturas de tipo ensemble (como son los **random forest**) tienen menos sesgo que los modelos individuales. Cada árbol de random forest es entrenado, mediante la técnica de **bagging**, con un subconjunto de los datos obtenidos. Además, cada árbol proporciona aleatoriedad a las particiones seleccionando atributos de forma aleatoria. Una vez cada árbol realiza su predicción, **random forest** une las predicciones mediante **ensemble**, ayudando a reducir el sesgo y el overfitting de cada uno de los modelos individuales.

Debido a esta forma de construcción mencionada anteriormente, los random forest también son robustos ante valores outliers y datasets desbalanceados.

9. Sostenibilidad Ambiental

En cuanto a la elección del modelo, hemos utilizado aquellos basados en árboles en lugar de los de aprendizaje profundo, debido a que estos últimos requieren del uso de **GPUs** para su entrenamiento y despliegue, lo que se traduce en un mayor consumo energético. Además, los modelos basados en árboles proporcionan resultados similares o incluso mejores en comparación con los de aprendizaje profundo, ofreciendo también un tiempo de ejecución mucho más bajo [2]. Esto significa que, en caso de desplegar el modelo en producción, el tiempo necesario para realizar predicciones será muy reducido, lo que contribuirá a mejorar la eficiencia energética del sistema en el que se integre.

Dentro de los modelos basados en árboles, **random forest** requiere generalmente de un menor tiempo de entrenamiento en comparación con otros modelos basados en **gradient boosting**. Esto significa que una actualización futura del modelo presentado con nuevos datos requerirá de menos recursos. Por todo lo anterior consideramos que el modelo final presentado es una excelente opción tanto en términos de precisión como de eficiencia.

10. Conclusiones

Tras todo el estudio realizado, resulta importante destacar la relevancia que han tenido en los modelos la generación de nuevas variables del dataset train a partir de las existentes, como *produccion_medio* y la ampliación de datos sobre la *superficie*, siendo ambas variables las más influyentes en la predicción final del modelo. Asimismo, a pesar de añadir información sobre el dataset meteo mediante variables simples y más complejas, el error obtenido era siempre mejor prescindiendo de dicha información.

Por otro lado, el uso de **random forest** encaja con la idea de modelo simple, eficiente y explicable que se buscaba desde un inicio, además de ser insesgado y tener una gran capacidad de resistencia al sobreajuste.

Como trabajo futuro, se podría volver a intentar extraer información del dataset meteo, creando variables nuevas que relacionen los atributos más relevantes (sobre la temperatura o precipitación) con otros que tienen poca relevancia pero que sí son importantes para el cultivo de la vid (como la humedad o el punto de rocío). En general, la construcción de atributos es un mecanismo que podría mejorar los modelos de regresión obtenidos, pero el proceso puede requerir de bastante tiempo y numerosas pruebas.

Referencias

[1] KONDIRI, Venkata Shashank, et al. Data science for weather impacts on crop yield. *Frontiers in Sustainable Food Systems*, 2020, vol. 4, p. 52.

[2] ROßBACH, Peter. Neural networks vs. random forests—does it always have to be deep learning. *Germany: Frankfurt School of Finance and Management*, 2018.