

Datathon Cajamar R-Commanders

Álvaro Añón Dosil

Marcos Gómez Rodríguez

Antón Quintela Ferreiro

19 de abril de 2023

1. Introducción

En este informe presentamos tanto el análisis exploratorio, como la transformación de los datos y la explicación de los modelos empleados durante el estudio de la producción de vino de la Cooperativa La Viña. El objetivo principal de este trabajo, el cual se llevó a cabo con el software R, fue conseguir la mejor predicción posible para las producciones del año 2022.

El informe está estructurado de la siguiente manera: en la Sección 2 se hace un breve repaso sobre el análisis exploratorio realizado; en la Sección 3 se comentan las variables externas usadas, las transformaciones aplicadas a los datos y cómo se imputaron los datos faltantes; finalmente, en la Sección 4 se explican los modelos considerados, la elección del modelo final y su interpretación.

2. Análisis exploratorio de datos

Lo primero que se examinó fue la serie histórica de producciones, detectando que para muchas fincas no se disponía de la serie completa. Al faltar un porcentaje tan elevado de producciones (en torno al 50%) se descartó el uso de series de tiempo y se optó por incluir en el modelo las producciones del año anterior, de hace dos años y la media histórica. Además de las producciones históricas, la variable más correlada con las producciones fue la superficie, con una correlación de 0.7180681, aunque solo se dispone de ella a partir del 2020.

A continuación, llevamos a cabo un análisis de cada una de las variables categóricas de UH_2023_TRAIN, estudiando sus distribuciones y las relaciones que estas mantenían entre sí, aunque sobre todo centrándonos en su relación con la variable PRODUCCION. Se observó que la variable categórica más relevante es MODO, mientras que el resto de variables categóricas apenas tenían diferencias en su producción media. Esto nos lleva a pensar que los datos no requieren una corrección para evitar sesgos, ya que las variables categóricas del modelo no parecen generar diferencias en la variable PRODUCCION, a excepción de la variable MODO. Aun así, esto puede deberse a que un modo de cultivo sea verdaderamente más eficiente que otro, y dado que en la información acerca de la base de datos no se explica cuáles son las diferencias entre los modos de cultivo, no podemos saber si estamos ante un sesgo en los datos o un fiel reflejo de la realidad. Es por ello por lo que asumiremos que los datos no están sesgados, pues no hay evidencias para pensar lo contrario.

Finalmente, llevamos a cabo un estudio de los NAs en los datos UH_2023_TRAIN, donde a primera vista, y observando los gráficos que incluimos en el código, parece que solo hay datos faltantes en algunos valores de las altitudes de las fincas. Sin embargo, al analizar la variable SUPERFICIE, nos damos cuenta de que algunos de los valores en los años 2020, 2021 y 2022 (aquellos para los cuales se proporcionaban las superficies de las fincas) tenían valor 0 con producción positiva (lo cual parece imposible). Consecuentemente, decidimos asignar estos valores

como NAs también.

Tras ello, hicimos un estudio de los datos faltantes en los datasets DATOS_ETO y DATOS_METEO. En ellos vimos que, en todos los años a excepción del 2022, faltaban la totalidad de los datos de distintas variables, por lo que en este caso se descartó totalmente la imputación. Además, este hecho imposibilitó el uso de todas estas variables en un modelo que tuviese datos de todas las campañas y limitó de forma importante la aportación de la meteorología en el modelo predictivo (la cual parecía bastante relevante).

3. Transformación y selección de variables

Con respecto a las variables climáticas, se han creado dos variables nuevas que pretenden recoger la aparición del mildew, una de las principales enfermedades de la vid. La primera es Goid, basada en la tabla del modelo de Goidànich (Goidànich, 1964, citado en Trilles et al., 1986), en la que se calculan puntuaciones de riesgo de infección diarias según factores climáticos. Estas puntuaciones se van acumulando, emitiendo un aviso al superar el valor 70, momento en el que es recomendable empezar el tratamiento contra el hongo. Al superar el valor 100, el riesgo acumulado vuelve a 0. El valor de Goid para cada campaña es el número de avisos producidos (días con riesgo mayor de 70). Cabe destacar que la tabla de Goidànich se incluye en el Goidànich.csv y, al estar discretizada, se le aplicó una interpolación.

La segunda variable es Espo, basada en el modelo de Fernández-González et al. (2019), que predice el crecimiento de las esporas según factores climáticos. Espo es la suma de los valores diarios de esporas predichos por el modelo en cada campaña.

El resto de las variables climáticas se han agrupado en temporadas según las etapas de desarrollo de la vid: Parada, desde octubre a febrero; Marzo, momento donde empiezan los brotes; Foliación, que comprende abril y las dos primeras semanas de mayo; y Floración, que comprende la segunda mitad de mayo y junio. Además, decidimos prescindir de las agrupaciones de Daytime y Nighttime, ya que se consiguieron resultados con errores similares usando únicamente la agrupación Day, con la ventaja de que al reducir en gran medida el número de variables, se reduce el tiempo de computación y se facilita la interpretabilidad.

Por otra parte, se creó una variable a la que denominamos ID_UNICO¹, esto se debe a que en un mismo año hay fincas que plantaron distintas variedades de uva, o mismas variedades pero de distinto tipo, y requerían de predicciones distintas. La variable ID_UNICO tenía como objetivo ayudar a capturar la dependencia temporal de las producciones, obteniendo para cada uno de sus valores su serie histórica de producciones. Utilizando la serie histórica se calcularon para cada ID_UNICO y cada campaña, las producciones del año anterior (PROD_1), de hace dos años (PROD_2) y la media histórica (PROD_mean). También se añadieron variables que indican la ausencia de alguno de esos datos (PROD_1na, PROD_2na y PROD_MEAN_NA). Cabe destacar que en la creación de este tipo de variables se llevó a cabo un esfuerzo para hacerlo lo más eficiente posible, empleando siempre que se pudiese la librería dplyr en lugar de bucles.

Finalmente, la última transformación que se llevó a cabo en los datos es la imputación de valores faltantes en las variables ALTITUD y SUPERFICIE, donde la variable ALTITUD se transformó de factor a numérico para dar más variabilidad a las imputaciones y evitar el sesgo

¹ID_UNICO es una combinación de ID_FINCA, VARIEDAD Y TIPO, para conseguir un identificador único de los cultivos.

en la medida de lo posible. Para la imputación, probamos dos modelos, los cuales se detallarán a continuación.

En primer lugar, se llevó a cabo la imputación con el paquete MICE. Este paquete crea varios datasets con diferentes imputaciones, las cuales genera a partir de un modelo de predicción al que le introduce ruido. Esta forma de imputar minimiza el sesgo, sin embargo, debido a que para maximizar la capacidad predictiva del modelo de imputación dividimos los datasets en 3 períodos (para emplear todas las variables posibles) nos vimos obligados a combinar posteriormente los datasets, generando 125 datasets diferentes. Esto provocaba un importante problema computacional, ya que por cada dataset habría que generar un modelo y posteriormente combinar los resultados, por tanto, decidimos descartar este modelo de imputación.

En segundo lugar, probamos el modelo finalmente escogido, el cual emplea el paquete missForest para llevar a cabo una imputación mediante random forest. En este modelo se llevaron a cabo las imputaciones de forma secuencial y separando en 2 etapas para cada una de las variables. Primeramente, se imputaron los valores de ALTITUD en los años 2014 - 2021, empleando todas las variables de UH_2023_TRAIN (quitando ID_FINCA, ID_ZONA y SUPERFICIE, pues no estaba imputada aún) y empleando las variables relativas a las producciones pasadas (nótese que en este caso sí se empleó la variable PRODUCCION). Seguidamente, se llevó a cabo la imputación de ALTITUD en 2022, empleando las mismas variables que en el modelo anterior, exceptuando la PRODUCCION.

A continuación, se imputó la variable SUPERFICIE, primero en los años 2020 – 2021, en los que se emplean todas las variables de UH_2023_TRAIN (quitando ID_FINCA e ID_ZONA) además de las variables relativas a la producciones pasadas. Cabe destacar que, al estar imputados los valores de ALTITUD, estos se pudieron emplear en el modelo de la SUPERFICIE. Nótese que en este caso sí se empleó la variable PRODUCCIÓN. Seguidamente, se llevó a cabo la imputación de SUPERFICIE en el año 2022, en el cual se utilizaron las mismas variables que en el anterior modelo, a excepción de la variable PRODUCCION, que en este año se desconoce.

Es preciso señalar que también se probó una imputación análoga a la anterior, pero sin las variables de nueva creación asociadas a la producción en años anteriores. No obstante, esta opción se descartó porque, aunque era más rápida (35.30s frente a 269.64s) y los resultados tenían pequeñas diferencias, en la Sección 6.3 de van Buuren (2018) se recomienda emplear todas las variables que se pueda, siempre y cuando se vayan a utilizar en el modelo final. Por último, cabe resaltar que, con el objetivo de llevar a cabo las imputaciones de forma más eficiente se emplearon paquetes de computación en paralelo, que redujeron el tiempo de computación considerablemente.

4. Modelos ajustados

En esta sección se presentarán cuatro modelos, siendo el último de ellos aquel con el que se obtuvieron las predicciones de esta entrega. Los modelos mencionados son los siguientes: en primer lugar, un modelo XGBoost `hist`; en segundo lugar, un modelo creado a partir de agrupar las fincas en clusters y llevar a cabo un ajuste por random forest para cada cluster; en tercer lugar, un modelo GAM empleando únicamente las campañas 2020 – 2021; y finalmente el modelo definitivo, un XGBoost `exact`.

Cabe destacar que uno de los principales problemas del conjunto de datos de entrenamiento era el hecho de que faltasen los valores de la superficie para los años anteriores al 2020, así como la falta de datos climatológicos para los años anteriores a 2016. Consecuentemente, se aprovechó la creación de las variables PROD_1, PROD_2 y PROD_mean para prescindir de los años 2014 y 2015. Con ese mismo motivo se decidió que para todos los modelos creados, a excepción del GAM, se ajustarían dos submodelos encadenados: el primero, con datos del 2016 al 2019, sin considerar superficies; y el segundo, con datos a partir del 2020, añadiendo la variable SU-

PERFICIE y las predicciones del primer modelo para el año 2022. Se denotará como submodelo 16/19 a la primera parte y submodelo 20/21 a la segunda. El esquema seguido se puede ver en la Figura 1. Una gran ventaja de esta estructura es su eficiencia a nivel computacional, pues si se quisiese emplear para predecir, por ejemplo, la producción del año 2023, tan solo habría que entrenar el segundo de los modelos. De esta forma, bastaría con obtener las predicciones del submodelo 16/19 para los datos de 2023 e introducirlos en el segundo submodelo como variable, donde este último submodelo sí habría que entrenarlo de nuevo, añadiendo los datos del año 2022 a la muestra de entrenamiento.

Para evaluar el rendimiento de los submodelos 16/19 se emplearon dos medidas, la primera es el RMSE calculado mediante CV por k -folds de la librería caret, en el cual cada fold es uno de los 4 años empleados en el submodelo 16/19. Además, se tuvo en cuenta el RMSE obtenido al predecir sobre los años 2020 y 2021, con el fin de comprobar que los resultados excepcionalmente buenos en k -folds no fuesen fruto del sobreajuste. En cuanto a los submodelos 20/21, la medida que empleamos para llevar a cabo la validación es la obtenida por CV en la librería caret, pues a lo largo de las distintas entregas realizadas comprobamos que, buenos resultados en CV de caret, se traducían en buenos resultados en las predicciones de 2022. Una vez aclarada cuál fue la métrica empleada para comparar los modelos, se explicarán con mayor detalle los modelos presentados, comparándolos con el modelo final.

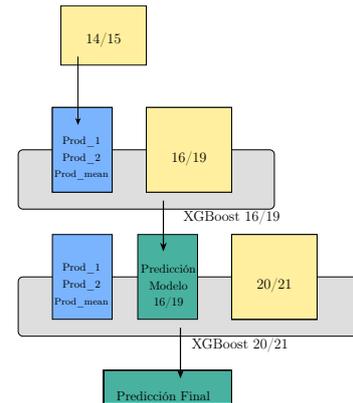


Figura 1: Estructura del modelo final.

El primer modelo presentado es el XGBoost **hist**, que hace referencia a la implementación del LightGBM en la librería xgboost, con la estructura de la Figura 1. Este fue el presentado en la fase local debido a que es computacionalmente menos costoso que un XGBoost normal, (con el método **exact**) aunque a costa de tener errores ligeramente mayores. Este modelo obtuvo un RMSE de 5078.95, lo que nos valió para clasificarnos para la fase nacional. No obstante, debido a que este tiene casi 300 más de RMSE que el modelo final, a que la diferencia del tiempo de computación entre uno y otro es insignificante, (en lo que al entrenamiento del modelo se refiere) y a que su explicabilidad es idéntica, nos decantamos por el XGBoost **exact**.

El segundo modelo que explicaremos es el random forest separado en clusters, el cual fue una de las dos entregas intermedias que llevamos a cabo en esta fase final. Este modelo consistía en separar los datos en clusters empleando las variables $prop_1$ y $prop_2$ ² y posteriormente, crear un modelo basado en random forest para cada uno de los clusters (con la estructura de la Figura 1). Este modelo obtuvo un RMSE de 5.268,66, que es significativamente peor que el del modelo anterior y casi 500 unidades peor que el modelo final. Además, aunque algunos de los submodelos de los clusters eran ligeramente más rápidos que el XGBoost **exact**, al sumar los tiempos de todos los clusters, estos eran significativamente más lentos. A mayores, este modelo planteaba un problema de interpretación, y también de sesgo, debido a que la introducción de una finca en un cluster determinaba como se calculaba su predicción. Por último, un modelo random forest no es mucho más sencillo de interpretar que un XGBoost, por lo que todos estos factores, unidos a su mal desempeño hicieron que este modelo fuese descartado en favor del XGBoost **exact**.

²Proporción de la producción total que la finca en cuestión había representado hacía uno y dos años respectivamente.

	Tiempo Ajuste	RMSE
GAM	0.08s	7229.07
RF	63.78 / 210.15	5268.66
XGB-hist	6.5s / 18.66s	5078.95
XGB-exact	10.94s / 24.98s	4787.51

(a) Comparativa tiempos de ejecución y RMSE obtenidos con hp victus 16-d1021ns con 12th Gen Intel(R) Core(TM) i7-12700H 2.30 GHz.

Hiperparam.	Mod. 16/19	Mod. 20/21
eta	0.0075	0.02
max depth	9	9
min child weight	1	1
subsample	0.25	0.61
nrounds	400	275

(b) Hiperparámetros para los submodelos que generan las predicciones finales, utilizando XGBoost exacto en ambas.

Cuadro 1: En (a), separados con una barra están los modelos 16/19 y 20/21. El modelo GAM es un modelo conjunto, por eso solo hay un tiempo. En el random forest por clusters, se han sumado los tiempos de cada cluster. Además, los RMSE de los tres últimos modelos son los obtenidos en las entregas, mientras que el del GAM fue obtenido por métricas internas.

El tercer modelo es un modelo GAM, que emplea regresión por splines y es más interpretable que el resto de modelos considerados. Sin embargo, este no tenía ningún poder predictivo en comparación con los modelos basados en árboles, tal como se puede observar en el Cuadro 1a. En consecuencia, a pesar de ser altamente eficiente e interpretable no habría sido capaz de pasar el corte de los 10 mejores modelos y dichas propiedades no habrían sido valoradas.

Por último, el modelo final consiste en un modelo muy similar al de la entrega local, pero usando la versión exacta del XGBoost. En la entrega intermedia de la fase nacional se envió este modelo utilizando los hiperparámetros obtenidos para el XGBoost `hist` de la fase local, y conseguimos un RMSE de 4787.51, por lo que, con la intención de mejorarlo, para esta entrega se ha optado por emplear los hiperparámetros óptimos para el XGBoost `exact`. Cabe señalar que en la búsqueda de hiperparámetros adecuados sí que hay una diferencia mayor de tiempo de computación entre el modelo exacto y el aproximado, aunque en ninguno de los casos es exagerado (para el exacto se tardaron 3921.22 s para el submodelo 16/19 y 7794.11 s para el submodelo 20/21).

Una vez explicados los distintos modelos y justificada la elección del modelo final veamos como se escogen los hiperparámetros del mismo. Aunque es habitual elegir una rejilla o *grid* de hiperparámetros y probarlos todos o hacer búsquedas aleatorias, en este trabajo se optó por usar optimización bayesiana, que obtiene resultados similares a búsquedas por rejillas con menor tiempo de computación (Wu et al., 2019). En este caso, los hiperparámetros `gamma` y `colsample_bytree` del XGBoost no se incluyeron en la búsqueda, dejando sus valores por defecto (0 y 1 respectivamente). En el Cuadro 1b se muestran los hiperparámetros utilizados en la entrega final.

A continuación, llevaremos a cabo una interpretación del modelo final, centrándonos en el submodelo 20/21, pues es el que proporciona las predicciones. Sin embargo, cabe destacar que el modelo 16/19 tiene unas interpretaciones análogas (ya que está construido para ser muy similar) y se dejan en el script de predicción todas las gráficas que lo respaldan, junto con algunos comentarios. Para analizar la influencia de las variables usamos el método de SHAP, como se hace en Zhang et al. (2020).

Seguidamente, tal y como se puede ver en la Figura 2, las variables que tienen mayor peso en la predicción son: `xgboost`, `SUPERFICIE`, `PROD_mean`, `PROD_2`, y `PROD_1`, en ese mismo orden. De esta forma, la variable `xgboost` modela la esperanza para el año 2022 y `PROD_mean` la de la producción histórica. Por otra parte, la `SUPERFICIE` es importante, ya que es obvio que la superficie cultivada ha de influir significativamente en la producción. Por último, están

PROD_1 y PROD_2, que sirven para medir la tendencia temporal a corto plazo (si asciende o desciende). Es importante destacar que los efectos sobre la respuesta de todas las variables mencionadas hasta el momento, son los esperados, a mayor valor de estas variables, mayor será la predicción, tal como se puede observar en la Figura 2.

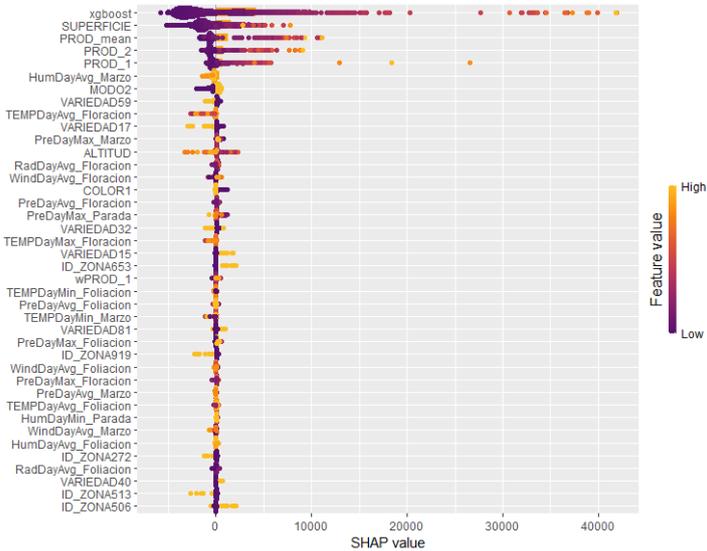


Figura 2: Gráficos de enjambre de abeja (*bee swarm*) para los valores de Shapley de las variables más relevantes en el modelo 20/21.

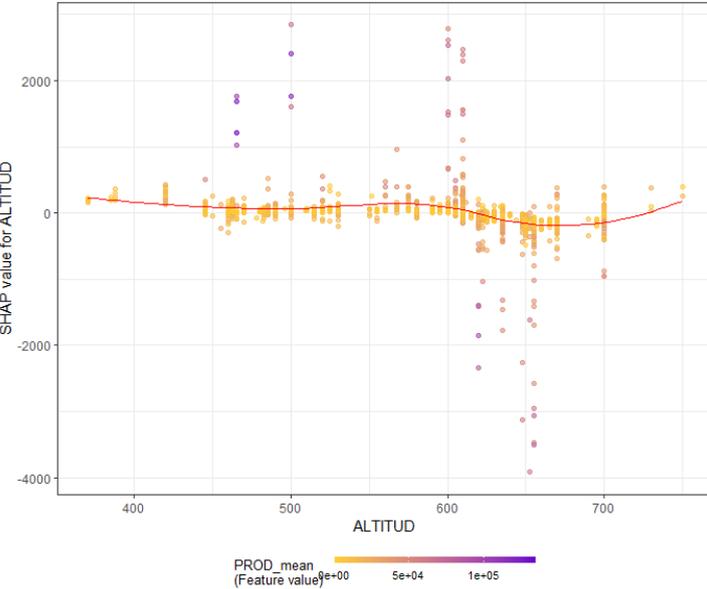


Figura 3: Gráfico de ALTITUD vs Shapley de ALTITUD, con los puntos coloreados según la variable PROD_mean.

A mayores, cabe señalar que en el modelo existe una interacción relevante entre las variables ALTITUD y PROD_MEAN (y en consecuencia también existe con cualquier variable con una alta correlación con PROD_mean). Esta interacción se puede apreciar en la Figura 3 en donde el modelo da a los valores altos de PROD_mean valores de Shapley positivos (aumentan la

predicción) hasta que la ALTITUD supera los 600m. A partir de dicha altitud, comienza a asignarles valores negativos, es decir, corrige la predicción, reduciéndola ligeramente. Esto se puede deber a que por cada 100m de altitud, la temperatura media desciende un grado, por lo que las probabilidades de heladas y temperaturas perjudiciales para la vid aumentan y, en consecuencia, el modelo reduce su predicción. De todas formas, a excepción de esta interacción, el modelo es bastante explicable para ser un XGBoost.

Una vez hemos arrojado luz sobre el funcionamiento del modelo, pasaremos a otro aspecto importante del mismo, que es su tratamiento del sesgo. Tal y como se puede observar en los perfiles de dependencia parcial del script de predicción, parece claro que el modelo creado no tiene un problema de sesgo. Esto se debe a que en estas gráficas se aprecia que, tanto para la variable MODO, como para cualquiera de las variables categóricas escogidas, el modelo parece bastante equilibrado, obteniendo predicciones medias muy similares para las diferentes categorías de las variables.

Finalmente, queremos resaltar que con el objetivo de que este modelo sea más accesible y transparente para el usuario final, se ha creado una aplicación en R-shiny al final del script de predicción. En esta aplicación, introduciendo el ID_UNICO, se devuelven las gráficas con los valores de Shapley (medidas de importancia) de cada una de las variables en la predicción, así como un gráfico en el que se puede apreciar la aportación concreta de cada una de las variables al valor de la predicción de dicha finca. De esa forma, el usuario puede entender de forma rápida y visual qué factores están influyendo más en la predicción de su producción y de que manera.

Referencias

- Fernández-González, M., Piña-Rey, A., González-Fernández, E., Aira, M. J., & Rodríguez-Rajo, F. J. (2019). First assessment of Goidanich Index and aerobiological data for Plasmopara viticola infection risk management in north-west Spain. *The Journal of Agricultural Science*, 157, 129-139. <https://doi.org/10.1017/S0021859619000376>
- Goidanich, G. (1964). *Manuale di patologia vegetale* (Vol. 2).
- Trilles, S., Torres-Sospedra, J., Belmonte-Fernández, Ó., Zarazaga-Soria, F. J., González-Pérez, A., & Huerta, J. (1986). Development of an open sensorized platform in a smart agriculture context: A vineyard support system for monitoring mildew disease. <https://repositori.uji.es/xmlui/handle/10234/181833>
- van Buuren, S. (2018). *Flexible Imputation of Missing Data* (2.^a ed.).
- Wu, J., Chen, X.-Y., Zhang, H., Xiong, L.-D., Lei, H., & Deng, S.-H. (2019). Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimization. *Journal of Electronic Science and Technology*, 17(1), 26-40. <https://doi.org/10.11989/JEST.1674-862X.80904120>
- Zhang, K., Xu, P., & Zhang, J. (2020). Explainable AI in Deep Reinforcement Learning Models: A SHAP Method Applied in Power System Emergency Control, 711-716. <https://doi.org/10.1109/EI250167.2020.9347147>