Clasificación del varietal de una muestra de vino basado en medidas objetivas

Trabajo final de la Especialización en Estadística Matemática

Ignacio Spiousas

14 de febrero de 2024

Introducción

La identificación del varietal de uvas con las cual se produce un vino es una tarea compleja y que muchas veces se basa en la pericia de un sommelier experimentado. Este tipo de tarea se basa usualmente en descriptores subjetivos como "suave", "robusto" o "astringente". Estos descriptores son a menudo poco confiables y hay evidencia de inconsistensias sistemáticas en estas caracterizaciones (Cao (2014)). Sin embargo, estas predicciones subjetivas no deben ser pasadas por alto ya que son una herramienta altamente instalada en la industria del vino y con relativo valor predictivo. Por ejemplo, en Oczkowski (2016) se demostró que la influencia de estas opiniones personales expertas en la predicción de vinos de alta gama en Australia es similar a la influencia de mediciones objetivas como clima, año y productor del vino.

Sin embargo, muchas de estas cualidades subjetivas pueden relacionarse con medidas objetivas obtenidas a través del análisis químico. De hecho, en la literatura podemos encontrar muchos ejemplos de aplicaciones de aprendizaje automático a partir de este tipo de métricas para la clasificación de vinos. Por ejemplo, Beltran y colegas (Beltrán et al. (2008)) emplearon algoritmos de reducción de la dimensión (PCA[1]) en combinación con técnicas de clasificación (LDA, SVM y RBFNNs[2]) para clasificar el varietal de vinos chilenos (Cabernet Sauvignon, Merlot o Carmenere) a partir de los perfiles de aroma obtenidos por cromatografía. Otro ejemplo, muy relevante para el presente trabajo, es la propuesta de Barth y colegas (Barth et al. (2021)) en la que utilizan el mismo conjunto de datos que utilizaremos a continuación y, utilizando PCA y un algoritmo de clasificación (KNN[3]), clasifican el varietal de una muestra a partir de 13 métricas objetivas obtenidas a parti del análisis químico. En particular, los autores de este último artículo ponderan la interpretabilidad de su propuesta, que reduce las dimensiones a sólo dos, una relacionada con el perfil aromático y la otra relacionada con el alcohol y los niveles de fermentación. El resultado de este procedimiento nos permite comprender cómo se relacionan las características sensoriales de cada vino con su varietal,

[1] Principal component analysis.
[2] Linear Discriminant Analysis, Support Vector Machine y Radial Basis Function Neural Networks.
[3] K-nearest neighbor.

haciendo posible crear o verifican hipótesis sobre las condiciones geográfica, geológicas y climáticas de cada cultivo en el perfil aromático de cada vino.

En el presente trabajo vamos a evaluar estrategias de clasificación automática para determinar el varietal de una muestra de vino a partir de mediciones objetivas. En particular, vamos a estudiar el mismo conjunto de datos que Barth y colegas (Barth et al. (2021)), un conjunto de datos que contiene los resultados de análisis químicos de vinos provenientes de la misma región de Italia pero de tres varietales diferentes[4].

Métodos y resultados

La presente sección comienza con una descripción cualitativa de los datos y un análisis exploratorio de los mismos. Luego, vamos a detallar los métodos evaluados como posibles candidatos para construir un algoritmo que nos clasifique el varietal de cada muestra a partir de los resultados del análisis químico de forma confiable. Las cuatro alternativas de modelado propuestas son: K vecinos cercanos, Random Forest, Regresión logística y Redes neuronales. Los detalles de cada uno de estos modelos así como de su implementación se darán en la subsección correspondiente.

Todos los computos de estos modelos fueron realizados utilizando código R (R Core Team (2023)) y diversos paquetes específicos que serán mencionados en cada uno de los métodos[5].

Análisis exploratorio de datos

El conjunto de datos con el trabajaremos consiste en mediciones de un análisis químico de 178 vinos junto con el varietal al que pertenecen. El análisis químico está representado en 13 variables numéricas: **Alcohol**, **Ácido málico**, **Nivel de ceniza**, **Alcalinidad de la ceniza**, **Niveles de Magnesio**, **Fenoles totales**, **Fenoles flavonoides**, **Fenoles no flavonoides**, **Proantocianidinas**, **Intensidad del color**, **Matiz**, **OD280/OD315**, y **Proline**. A su vez, estas variables pueden agruparse en cuatro categorías: Alcohol/fermentación, Sabor, Contenido de minerales y Apariencia. En el cuadro 0 puede verse una descripción de cada una de las variables medidas y la categoría a la que pertenecen. El varietal en los datos originales es una variable númerica que puede tomar valores de 1 a 3, pero basado en lo propuesto en Barth et al. (2021), podemos inferir que los números correponden con los varietales *Barolo*, *Grignolino* y *Barbera*, respectivamente.

Vamos a comenzar el análisis exploratorio de datos observando cuán balanceada es la muestra repecto a los varietales. En la figura 1 podemos ver esto representado en un gráfico de barras en el que se observa que, si bien el varietal *Barbera* está ligeramente subrepresentado, no existen grandes desbalances de clase en el conjunto de datos a evaluar. Se tienen 59 observaciones correspondientes

[4]Los datos utilizados pueden consultarse en el siguiente link. Más información en Aeberhard et al. (1994).

[5]Todas las manipulaciones de los datos fueron realizadas utilizando {dplyr} (Wickham, François, et al. (2023)) y {tidyr} (Wickham, Vaughan, et al. (2023)) y todas las visualizaciones utilizando {ggplot2} (Wickham (2016))

Cuadro 0: Categorías y descripción de las variables medidas. Adaptada de Barth et al. (2021).

Categoria	*Variable*	*Descripción*
Alcohol/fermentación	Alcohol	Porcentaje de alcohol
	Proline	Aminoácidos que afectan el crecimiento de las levaduras
Sabor	Fenoles totales	Compuestos químicos que afectan el gusto
	Fenoles flavonoides	
	Fenoles no flavonoides	
	Ácido málico	Contribuye a la acidez
	Proantocianidinas	Afectan la astringencia y la "sequedad"
Contenido de minerales	Nivel de ceniza	Cantidad de minerales inorgánicos
	Alcalinidad de ceniza	pH de la ceniza
	Nivel de magnesio	
Apariencia	Matiz	Color general
	Intensidad del color	Claridad u oscuridad del color
	OD280/OD315	Contribuye a la opacidad

al varietal *Barolo*, 71 del *Grignolino* y 48 del *Barbera*.

Luego, observamos como se distribuye cada una de las variables medidas para cada uno de los varietales. Esto lo haremos combinando un boxplot, que nos permite ver ciertos parámetros representativos de la distribución de los datos, y un violin plot, que nos muestra una estimación no paramétrica de la densidad a partir de un *kernel* gaussiano. En la figura 2 pueden verse ambas representaciones para cada variable descrita anteriormente con el varietal al que pertenecen codificado con color.

En la figura 2 se observa que hay algunas variables que presentan una clara separación por tipo de varietal y pueden ser relevantes en la clasificación, tales como **Alcohol**, **Alcalinida de la ceniza**, **Fenoles totales**, **flavonoides** y **no flavonoides**. A modo de ejemplo, en el caso de la variable **Fenoles totales** vemos que las mediciones son superiores para el varietal *Barolo*, seguido por el *Grignolino* y, por último, el *Barbera*. Por otra parte, en **Matiz**, **Ácido málico, Intensidad de color** y **OD280/OD315**, el varietal *Barbera* pareciera separarse del resto, mientras que en **Proline** es el *Barolo* el que se separa de los otros dos. Por último, se observan casos de mediciones como **Nivel de Ceniza**, **Niveles de Magnesio** y **Proantocianidinas** en las que no se distinguen claramente los varietales.

Otra cosa que puede verse en la figura 2 es que cuando **Fenoles flavonoides** bajan los **Fenoles no flavonoides**, lo que me generó la pregunta si la variable **Fenoles totales** es la suma de ambas

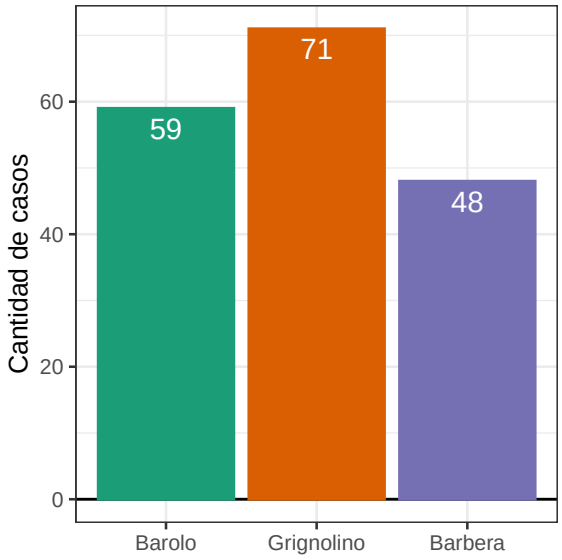

Figura 1: Cantidad de datos pertenecientes a cada clase (Varietal) en el dataset a utilizar. El color codifica el varietal.

magnitudes. Esto es relevante ya que, si fuera una variable la combinación lineal de otras dos, esto la transformaría en inútil para cualquier tarea de clasifiación. Para verificar esta hipótesis, grafiqué la variable **Fenoles totales** vs. la suma de los **flavonoides** y **no flavonoides**. En la figura 3 podemos ver un *scatterplot* que represeenta la relación antes planteada, codificando con color el varietal al que pertenece cada medición. Puede verse también que, si bien hay una alta correlación entre ambas magnitudes, la variable **Fenoles totales** no es simplemente la suma de los **flavonoides** y los **no flavonoides**.

Por último, vamos a indagar en la correrlación entre las medidas químicas. Para esto vamos a graficar la matriz de correlación ya que no estamos interesados en la información numérica exacta sino más bien en las estructuras esquemáticas de alta y baja correlación, ya sea positiva o negativa[6]. En la figura 4 puede verse que hay variables que están altamente correlacionadas positivamente, en especial el grupo de abajo a la izquierda que incluye a **Matiz**, **Proantocianidinas**, **OD280/OD315**, **Fenoles totales** y **Fenoles flavonoides**. Vale la pena notar que tres de estas cinco variables pertenecen a la categoría *Alcohol/fermentación*, mientras las restantes dos pertenecen a la categoría *Apariencia*. Sin embargo, no se observan agrupamientos generales con respecto a las categorías planteadas en la Tabla 1. Este puede tener que ver con que la definición de las categorías es conceptual y no necesariamente tiene porque haber una relación directa entre todas las variables que la componen.

[6]La figura fue realizada utilizando el paquete *{ggcorrplot}*(Kassambara (2023))

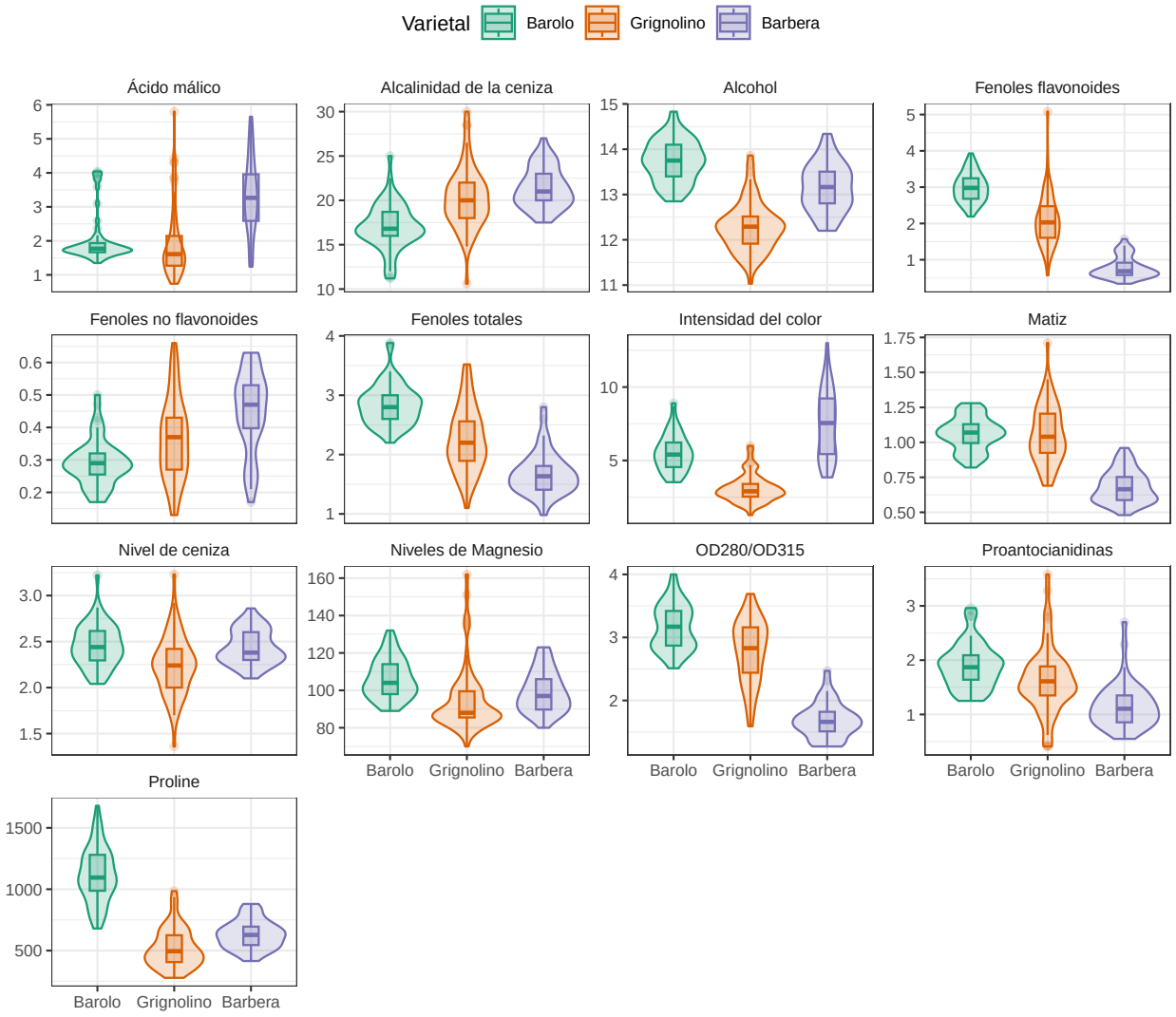

Figura 2: Boxplots y violin plots de las 13 variables según el tipo de cultivo. El color codifica el varietal.

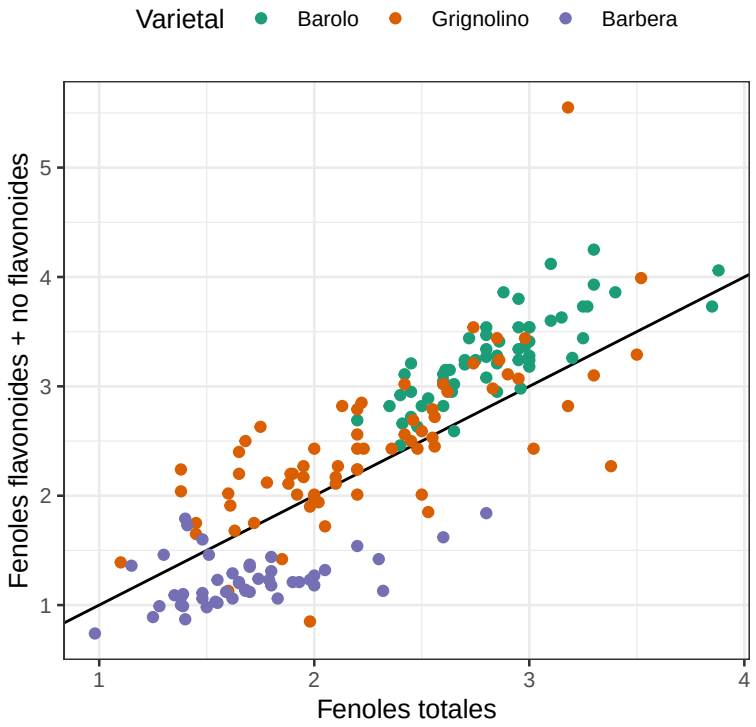

Figura 3: Fenoles totales vs. la suma de Fenoles flavonoides y no flavonoides. La línea negra representa una recta con ordenada al origen 0 y pendiente 1, que indicaría la igualdad entre ambas magnitudes. El color de los puntos codifica el varietal.

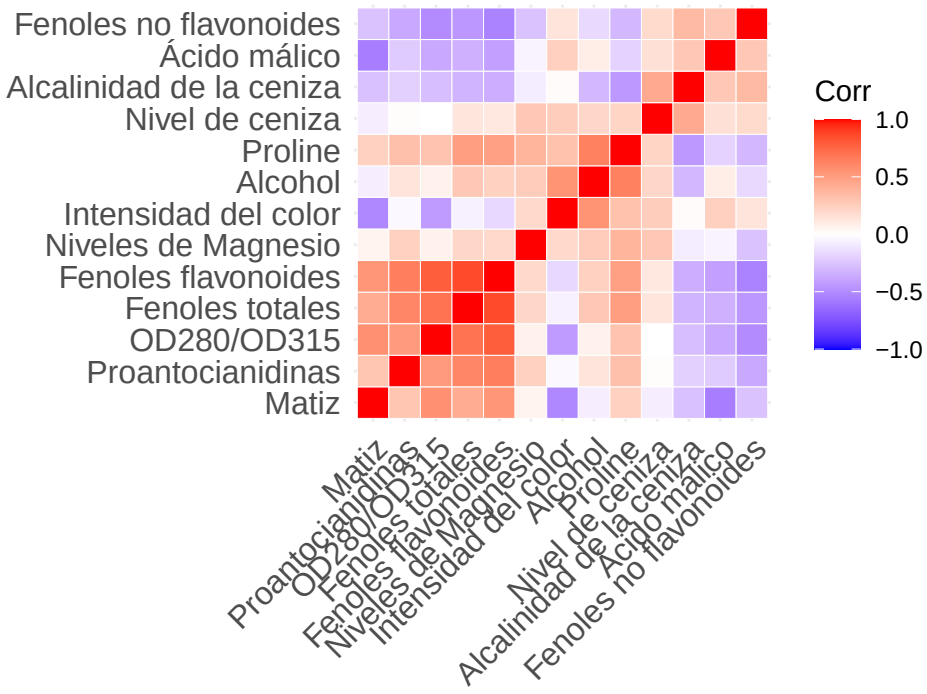

Figura 4: Matriz de correlación relacionando todas las mediciones químicas de las 178 muestas de vino.

Evaluación y selección de los métodos de clasificación

Como se mencionó en la Introducción, el objetivo del presente trabajo es el de explorar diferentes métodos de clasificación para determinar el tipo de varietal de una dada muestra. Como candidatos vamos a considerar métodos basados en K vecinos cercanos, Random Forest, Regresión logística (para modelos multinomiales) y Redes neuronales. La selección de estos cuatro métodos surge de la idea de considerar un método no parámetrico, uno basado en árboles, uno paramétrico (GLM) y uno basado en redes neuronales.

Para evaluar los distintos métodos de clasificación, separamos la muestra en un conjunto de entrenamiento (dos tercios de los datos) y un conjunto de testeo (un tercio de los datos) de forma estratificada según el Varietal [7].

La métrica que vamos a utilizar para evaluar los distintos modelos es el *accuracy*[8] ya que los datos no presentan desbalances de clases marcados ni creemos que haya alguno de los errores que debamos favorecer por sobre el otro. El *accuracy* se define como:

$$accuracy = \frac{Número\ de\ predicciones\ correctas}{Número\ total\ de\ predicciones}.$$

Para decidir cuál es el mejor método vamos a estimar el error utilizando el conjunto de datos de entrenamiento. Una forma sencilla de hacer esto sería dividir el conjunto de entrenamiento en dos subconjuntos (usualmente llamados de entrenamiento y de validación) y a partir de eso estimar la performance del modelo (en este caso calculando el *accuracy*). Sin embargo, para obtener una estimación menos sesgada, se presenta una alternativa conocida como validación cruzada (James et al. (2013)). La idea detrás de esta propuesta es la de separar el conjunto de testeo en n subconjuntos (llamado *folds*) y luego utilizar *n-1* para entrenar el modelo y el fold restante para evaluar su performance, repitiendo este procedimiento n veces, una por cada *fold*. Luego, la estimación del *accuracy* resulta de calcular el promedio de cada uno de estas estimaciones (una por *fold*). En nuestro caso realizamos la Validación cruzada separando la muestra de entrenamiento en 10 *folds* estratificando según el varietal[9]. La medida de performance, *accuracy* en este caso, será el promedio de la del las 10 estimaciones.

K vecinos cercanos

El primer modelo que vamos a evaluar es el de K vecinos cercanos (KNN de acá en adelante). Este método consiste en la determinación de la clase de un nuevo dato a partir de la clase mayoritaria en sus K puntos más cercanos, o vecinos (Hastie et al. (2009)).

El único parámetro a determinar en un modelo de KNN es la cantidad de vecinos cercanos a

[7] Esto lo hice utilizando la función `initial_split` de *{rsample}* (Frick et al. (2023))
[8] Usaré la palabra en inglés ya que no hay una traducción específica al español.
[9] Estos *folds* son generados utilizando la función `vfold_cv` del paquete *{rsample}* (Frick et al. (2023)).

considerar en la predicción. Para seleccionar esté parámetro vamos a evaluar una grilla de de valores de K entre 1 y 30. Para evaluar cuál es la cantidad de vecinos más conveniente realizamos validación cruzada tal como se describió anteriormente.

Como el método de KNN puede verse afectado por variaciones de escala, en cada paso de validación cruzada estandarizamos los datos del subconjunto que se utiliza como entrenamiento (9 *folds*) y, con esa misma transformación, escalamos los datos del subconjunto sobre el cual se predice (el *fold* restante).

Como puede verse en la figura 5, el máximo *accuracy* para los modelos de KNN es de 0.984 para 21 vecinos.

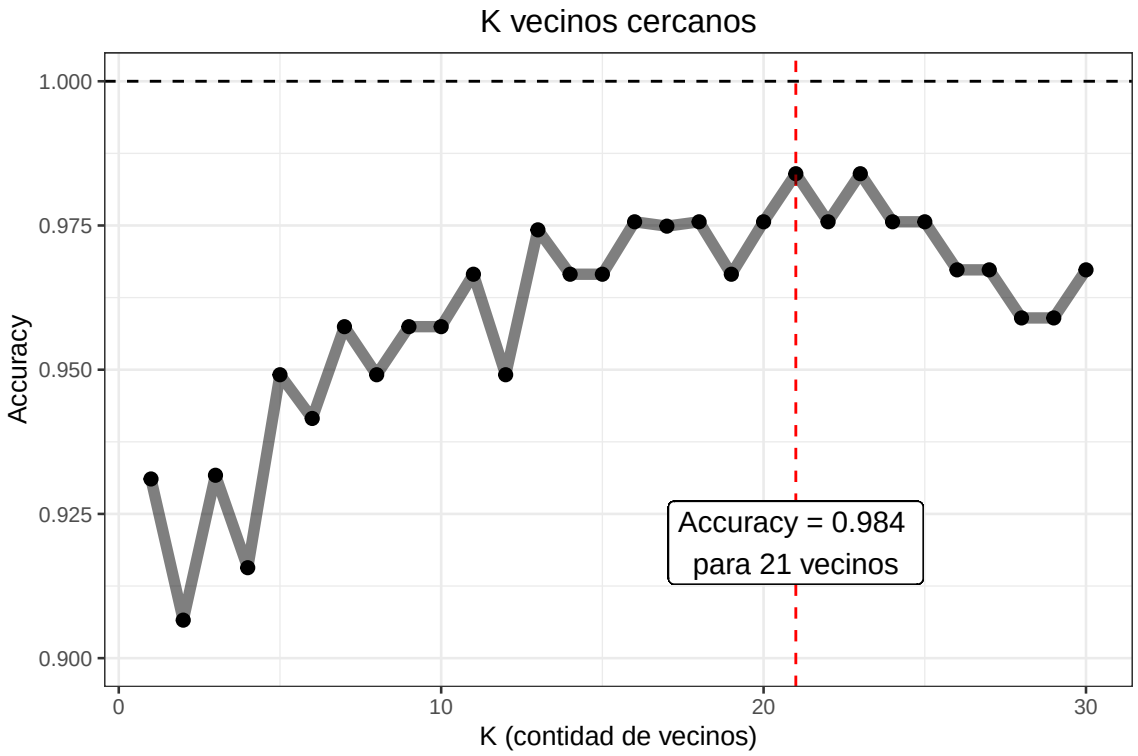

Figura 5: Accuracy en función de la cantidad de vecinos cercanos para la validación cruzada del modelo de K vecinos cercanos. La línea punteada roja indica la cantidad de vecinos que maximiza el accuracy.

Random Forest

La siguiente alternativa que vamos a considerar es un modelo denominado *Random Forest*. Este método se basa en árboles de decisión pero es considerado un método de ensamble, ya que el modelo ajustado realiza las predicciones basado en una combinación de las predicciones de varios árboles, de ahí que su nombre en español se podría traducir como bosque aleatorio (Hastie et al. (2009)).

En este caso también utilizaremos validación cruzada para hallar la combinación de parámetros que

maximice el *accuracy*, considerando los mismos *folds* que en KNN pero sin estandarizar los datos ya que para los métodos basados en árboles de decisión no resulta necesario (James et al. (2013)). Los hiperparámetros que vamos a optimizar en el modelo de Random Forest son: el número de variables que se consideran en cada split del árbol aleatorio (`mtry`); y el número mínimo de observaciones requeridas para que una hoja se bifurque (`min_n`).

Voy a calcular el accuracy para una grilla bidimensional que contenga valores de `mtry` entre 1 y 10 y valores de `min_n` entre 5 y 20. En total se trata de 160 combinaciones de hiperparámetros. El ajuste del modelo lo vamos a hacer utilizando la función `randomForest` del paquete *{randomForest}* (Liaw & Wiener (2002)).

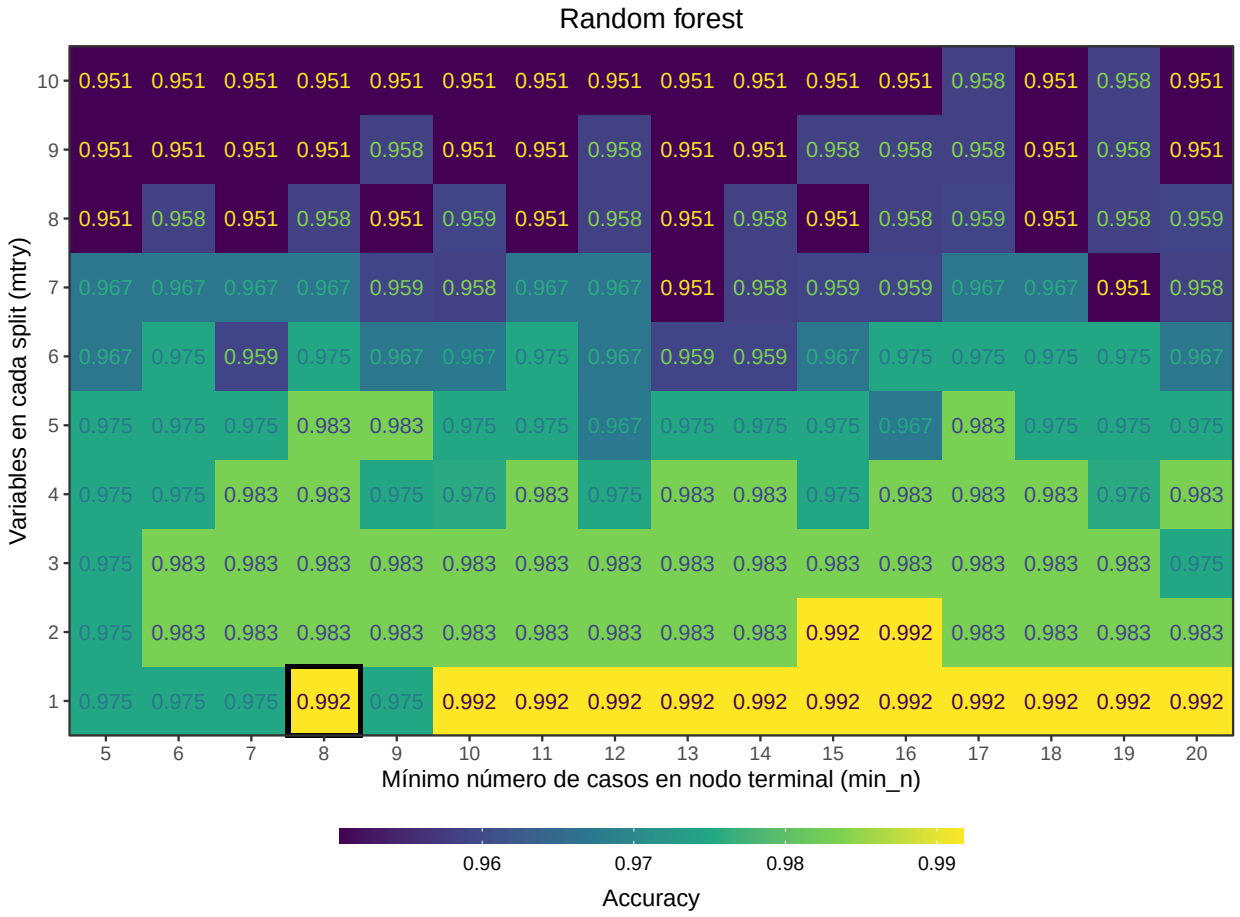

Figura 6: Accuracy en escala de colores en función de las variables en cada split y el número máximo de casos en cada nodo terminal para la validación cruzada del modelo de Random Forest. El cuadrado negro indica la combinación de hiperparámetros que maximiza el accuracy.

Como puede verse en la figura 6, el máximo valor de accuracy vale 0.992 para `m_try` igual a 1 y `min_n` igual a 8. Como el máximo accuracy se obtiene para varios valores de `m_try`, vamos a quedarnos con el menor para que se trate de árboles más simples (menos tiempo de cómputo y menos posibilidades de *overfitting*).

A continuación, vamos a ajustar un modelo de Random Forest a todo el conjunto de datos de entrenamiento, con los valores de hiperparámetros que maximizan el *accuracy*, para hacer un estudio de la importancia de las variables en la clasificación. Esto no permitirá ver si las ideas extraídas del análisis exploratorio tienen algún reflejo en la importancia de cada variable en la decisión. ´

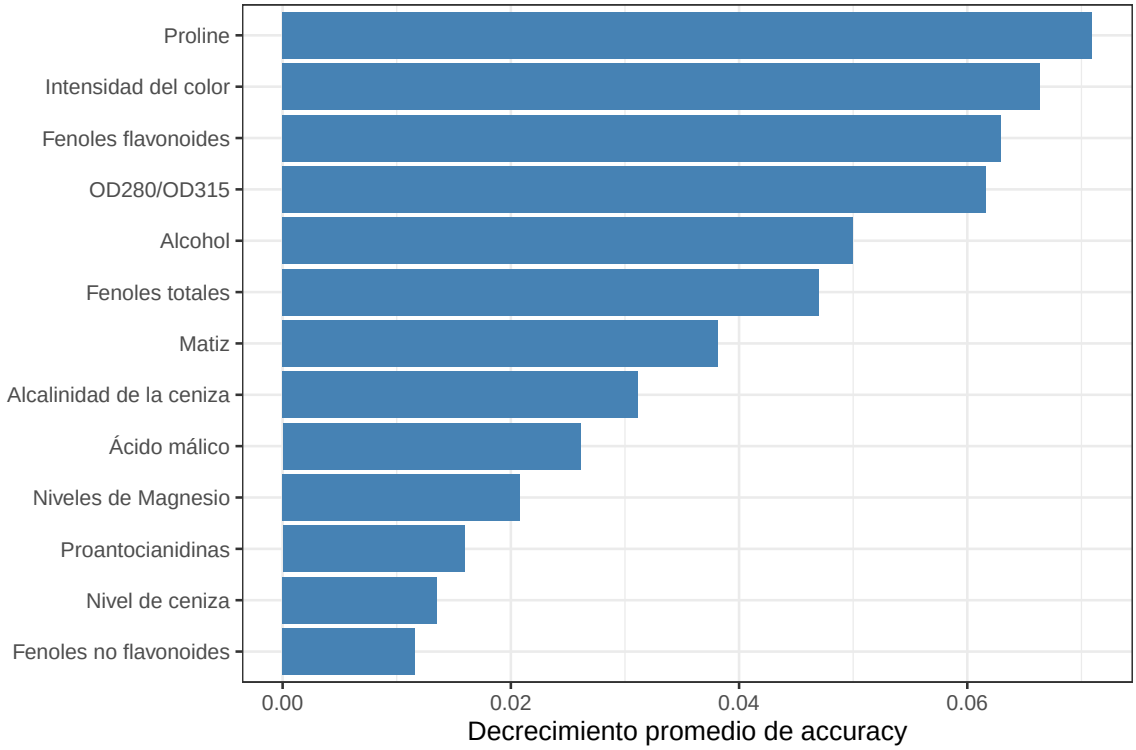

Figura 7: Importancia de las variables para el modelo de Random Forest ajustado a todo el conjunto de training con los hiperparámetros seleccionados en validación cruzada.

En la figura 7 podemos ver que, curiosamente, las tres variables más importantes para el modelo son las que habíamos identificado en el análisis exploratorio como variables que diferencian bien un varietal en particular de los otras dos. Por otro lado, las siguientes tres variables en importancia fueron identíficadas como buenas diferenciando los tres varietales.

Regresión logística

Para continuar, vamos a explorar una alternativa paramétrica y basada en modelos lineales para construir un clasificador. Se trata de un modelo lineal generalizado con *link function* logit(), es decir, lo que se conoce como una regresión logística. Como en este caso tenemos más de dos categorías debemos considerar una regresión logística multinomial (con tres categorías posibles) en lugar de la clásica binomial. Las ventajas de este modelo es la interpretabilidad de sus resultados. Al tratarse de un modelo paramétrico se puede asociar fácilmente las estimaciones de los parámetros con la dependencia de los *log-odds* con cada variable. Por tratarse de un problema con tantas variables

voy a considerar los modelos lineales generalizados con regularizaciones Ridge y Lasso[10].

Para esto, armamos una grilla de valores de λ tomando 100 valores entre 10^{-3} y 10^{0}. Para cada valor de λ, evaluamos el accuracy para las regularizaciones de Ridge y Lasso realizando validación cruzada de la misma forma que en los métodos previos.

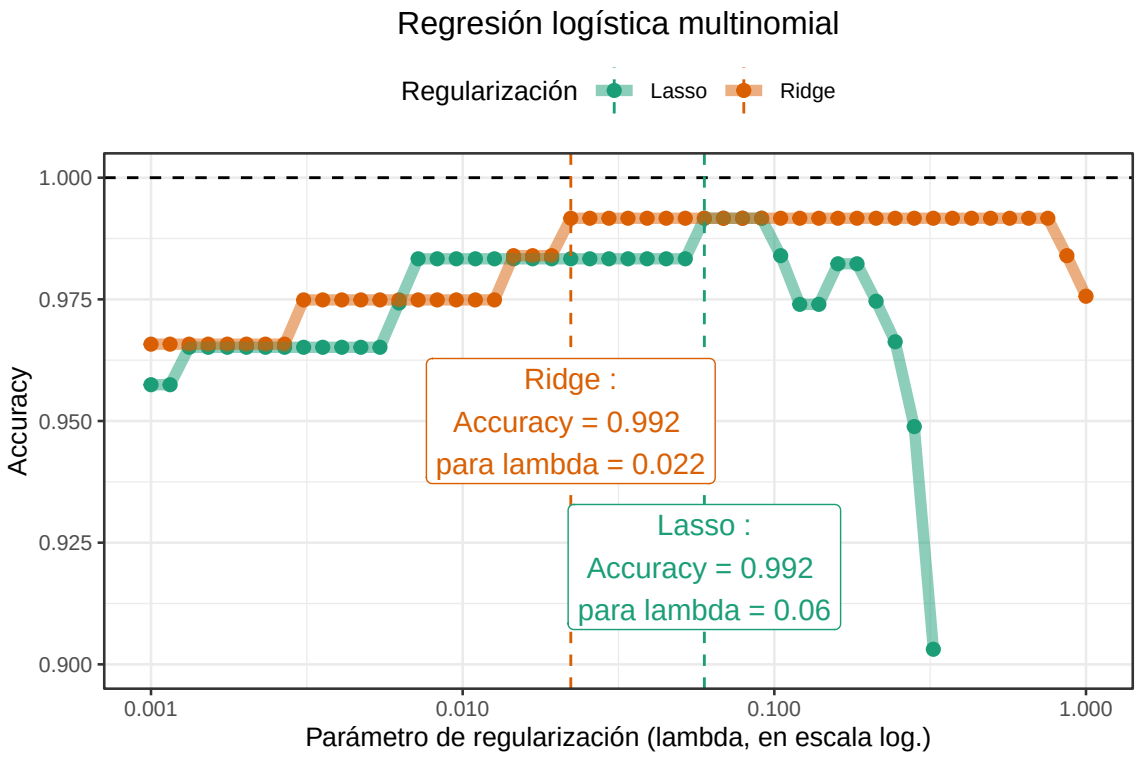

Figura 8: Accuracy en función del factor de regularización (lambda) la cantidad de vecinos cercanos para la validación cruzada del modelo de K vecinos cercanos. La línea punteada roja indica la cantidad de vecinos que maximiza el accuracy.

En la figura 8 observamos los resultados del accuracy en función de λ y vemos que, para la regularización Lasso el máximo se alcanza en λ igual a 0.06 con un valor de 0.9917 y para la regularización Ridge, el valor máximo de accuracy es el mismo, pero en este caso para λ igual a 0.022. Ya que ambas opciones de regularización llegan a la misma accuracy, voy a elegir trabajar con Lasso ya que, al permitir que los valores de los parámetros ajustados valgan cero, el modelo ajustado resulta más "chico" y, por ende, más fácilmente interpretable.

Una forma de tener una idea de la complejidad del modelo es ver los valores de los parámetros estandarizados. Para eso ajustamos el modelo elegido (Regularización Lasso con λ igual a 0.06) a los datos de entrenamiento completos y luego obtuvimos sus coeficientes. En el cuadro 1, podemos ver los valores de los parámetros para cada clase. De los trece parámetros originalmente presentes en el modelo de GLM, hay cinco que valen cero, es decir, el modelo utiliza sólo 8 de las variables

[10]Los ajustes los voy a realizar con la implementación del paquete {*glmnet*} (Friedman et al. (2010), Tay et al. (2023))

para predecir el varietal. Resulta interesante remarcar que **Alcalinidad de la ceniza**, **Niveles de Magnesio**, **Proantocianidinas**, **Fenoles totales** y **Proantocianidinas** también se encuentran entre las variables menos importantes para el modelo de Random Forest[11].

Cuadro 1: Parámetros estandarizados del ajuste de GLM logístico multinomial con los datos de training con regularización Lasso y lambda igual a 0.06.

Parametros	Barbera	Barolo	Grignolino
beta 0	0.6839014	-11.5929968	10.9090954
Alcohol	0.1448880	0.5199215	-0.6648095
Ácido málico	0.0389831	-0.0007265	-0.0382566
Nivel de ceniza	0.0624565	0.0948414	-0.1572979
Alcalinidad de la ceniza	0.0000000	0.0000000	0.0000000
Niveles de Magnesio	0.0000000	0.0000000	0.0000000
Fenoles totales	0.0000000	0.0000000	0.0000000
Fenoles flavonoides	-0.5558840	0.4317502	0.1241338
Fenoles no flavonoides	0.0000000	0.0000000	0.0000000
Proantocianidinas	0.0000000	0.0000000	0.0000000
Intensidad del color	0.1969894	0.0205050	-0.2174944
Matiz	-0.6325045	0.2127527	0.4197518
OD280/OD315	-0.8333040	0.5597984	0.2735056
Proline	-0.0002828	0.0028099	-0.0025271

Redes neuronales

Finalmente, vamos a considerar un modelo de redes neuronales para predecir el cultivo. Debido a que el problema es relativamente simple, exploraremos sólo modelos con una capa intermedia. Al igual que en los métodos anteriores, vamos a determinar la cantidad de neuronas de esta capa por

[11]Son seis de las ocho variables menos importantes.

medio de validación cruzada. Para esto, consideramos una grilla de valores entre 1 y 10 para la cantidad de neuronas y buscamos el valor que maximice el accuracy de validación cruzada.

En este caso, al igual que en K vecinos cercanos, estandarizamos los datos en cada paso de validación cruzada. El modelo de redes se ajusta utilizando la función `neuralnet` del paquete {*neuralnet*} (Fritsch et al. (2019)).

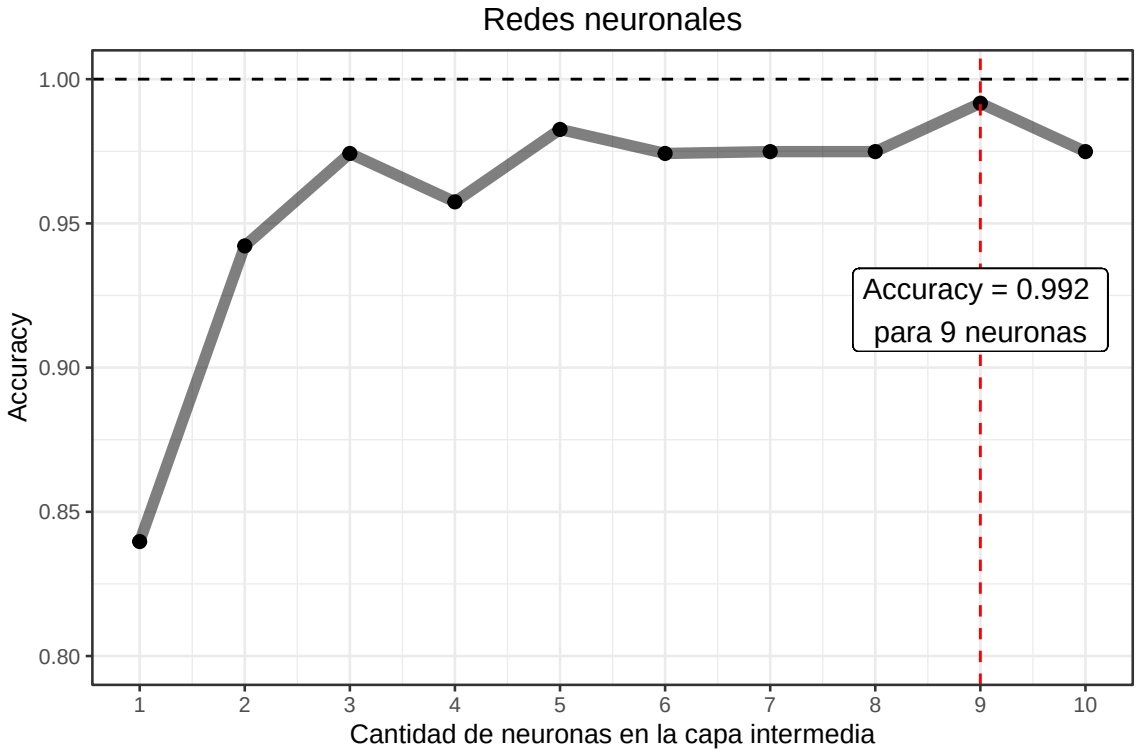

Figura 9: Accuracy en función de la cantidad de neuronas de la capa intermedia en un modelo de redes neuronales. La línea punteada roja indica la cantidad de neuronas que maximiza el accuracy.

En la figura 9 se observa los resultados del accuracy en función de la cantidad de neuronas de la capa intermedia y vemos que el máximo se alcanza utilizando 9 neuronas con un valor de 0.992.

Para tener un idea de la complejidad de la red propuesta, podemos ajustar la red con la cantidad de neuronas obtenidas mediante validación cruzada a todos los datos de entrenamiento. En la figura 10 podemos ver una representación esquemática de este modelo.

Selección del mejor clasificador basado en los resultados de la validación cruzada

En el Cuadro 2 podemos ver el *accuracy* obtenido en cada método luego de ajustar los hiperparámetros correpondientes por validación cruzada. Todos los métodos tienen un nivel de *accuracy* alto, siendo K vecinos cercanos el único ligeramente más bajo. Por este motivo, seguiremos considerando como candidatos a los otros tres métodos.

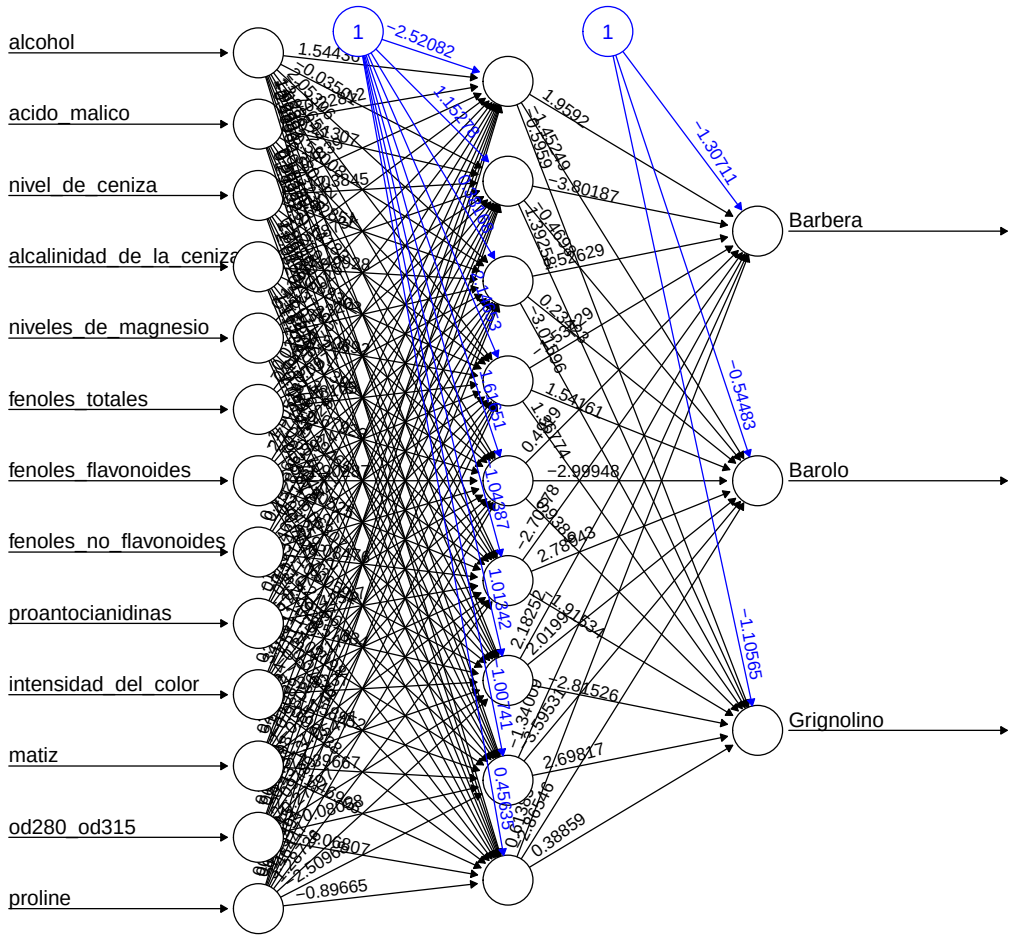

Figura 10: Representación esquemática de la red neuronal ajustada a partir de la selección de parámetros con validación cruzada.

Cuadro 2: Accuracy para cada método propuesto luego de hallar los hiperparámetros que la maximicen por validación cruzada.

Metodo	Accuracy de VC
KNN	0.9840
Random Forest	0.9917
GLM-Lasso	0.9917
Redes neuronales	0.9917

Paara elegir entre los otros tres métodos podemos considerar varios criterios entre los cuales elegí: Simplicidad del modelo, interpretabilidad del modelo y tiempo de cómputo. Por supuesto que estos tres criterios no son independientes (por ejemplo, modelos más simples suelen ser más interpretables).

Para comparar los tiempos de cómputo de cada uno de los tres modelos competidores, creamos un conjunto de datos de 10000 muestras, repitiendo aleatoriamente filas de los datos de entrenamiento. Luego, calculamos el tiempo en milisegundos que tarda cada modelo en predecir todos esas muestras. En el Cuadro 3 podemos ver que claramente los modelos basados en Random Forest y en Redes neuronales son más rápidos que la regresión logística multinomial. Siendo el más rápido de estos el Rándom Forest. Vale la pena aclarar que esta prueba de tiempo de cómputo es una estimación que sólo nos permite observar diferencias grandes.

Cuadro 3: Tiempo en milisegundos que le lleva a cada modelo clasificar 10000 muestras de vino.

Metodo	Tiempos (ms)
Random Forest	0.5340576
GLM-Lasso	9.4580650
Redes neuronales	2.4359226

Sin embargo, y como ya mencionamos anteriormente, la regresión logística tiene la ventaja de ser más interpretable. En cuanto a simplicidad, creo que tanto el Random Forest como la regresión logística son modelos más simples que las redes neuronales. Por estos motivos, sumados a la ligera

ventaja de Random Forest sobre redes neuronales en tiempos de cómputo, es que vamos a considerar como modelos más adecuados apra la tarea de clasificación propuesta a Random Forest y Regresión logística multinomial con regularización Lasso.

Comparación de los métodos en un conjunto de testeo

De los cuatro modelos propuestos inicialmente conservamos dos basándonos en la estimación de accuracy de validación cruzada y criterios más "blandos" como el tiempo de cómputo y la interpretabilidad de los modelos. A continuación vamos a comparar el desempeño de los modelos seleccionados para cada método ajustándolos con todo el set de entrenamiento y evaluándolos en el conjunto que reservamos para testeo.

Random Forest

El *accuracy de testeo* es de 0.983 para el modelo de Random Forest con la cantidad de variables en cada *join* del árbol aleatorio igual a 1 y la catidad máxima de casos en un nodo terminal igual a 8.

Adicionalmente, en el cuadro 4 pondemos ver la matriz de confusión para las 3 categorías del modelo de Random Forest al clasificar el conjunto de datos de *testeo*. Podemos ver que, de las 60 mediciones presentes, sólo una de las muestras fue mal clasificada: Una muestra de *Grignolino* fue clasificada como *Barolo*.

Cuadro 4: Matriz de confusión para el modelo de Random Forest al clasificar el conjunto de datos de testeo. En las filas vamos la verdadera catergoría de las muestras y en las columnas las categorías predichas por el modelo.

	Barolo	Grignolino	Barbera
Barolo	20	0	0
Grignolino	1	23	0
Barbera	0	0	16

Regresión logística

En la regresión logística multinomial con regularización Lasso y λ igual a 0.06, el accuracy luego de predecir el varietal a partir del conjunto de datos de *testeo* es de 0.95. En este caso, el *accuracy de testeo* es notablemente inferior que el de validación cruzada.

Para este modelo también calculamos la matriz de confusión para las 3 categorías (Cuadro 5). En este caso podemos ver que son tres las muestras que están mal clasificadas: Una muestra

de *Grignolino* clasificada como *Barolo* y dos mustras de *Grignolino* clasificadas como *Barbera*. De hecho, la muestra mal clasificada como *Barolo* es la misma que clasifica mal el modelo de Random Forest (la medición correspondiente a la fila 27). Entonces, el modelo de regresión logística multinomial, clasifica erroneamente dos muestras más del conjunto de testeo que el de Random Forest.

Cuadro 5: Matriz de confusión poara la regresión logística multinomial con regularización Lasso al clasificar el conjunto de datos de testeo. En las filas vamos la verdadera catergoría de las muestras y en las columnas las categorías predichas por el modelo.

	Barolo	Grignolino	Barbera
Barolo	20	0	0
Grignolino	1	21	2
Barbera	0	0	16

Conclusiones

Como puede verse en la sección anterior, ambos modelos preseleccionados a partir de la validación cruzada tienen una excelente performance con el conjunto de datos de *testeo*, sin embargo, el modelo basado en Random Forest clasifica ligeramente mejor las muestras de este conjunto de datos. Si bien el criterio de selección de modelo no puede depender de los datos de testeo, sino de la validación cruzada, esta ligera diferencia, sumada a la diferencia en tiempo de procesamiento hace que **el modelo basado en Random Forest con la cantidad de variables en cada *join* del árbol aleatorio igual a 1 y la cantidad máxima de casos en un nodo terminal igual a 8 sea el considerado como más adecuado para la tarea de clasificación de varietales basado en las mediciones químicas de muestras de vinos**.

Una de las posibles limitaciones del modelo elegido es que no tenemos información concreta de cuándo ni dónde se tomaron las muestras con las cuales se entrenaron y evaluaron los modelos. Esto no sólo significa que al clasificar muestras con distintas características (medias, varianzas, covarianzas, etc.) el modelo seleccionado podría tener una peor performance, sino que, dada esta nueva muestra, el modelo elegido como **más adecuado** podría ser otro aún siguiendo el mismo razonamiento que en este trabajo.

Referencias

Aeberhard, S., Coomans, D., & De Vel, O. (1994). Comparative analysis of statistical pattern recognition methods in high dimensional settings. *Pattern Recognition*, *27*(8), 1065-1077.

Barth, J., Katumullage, D., Yang, C., & Cao, J. (2021). Classification of wines using principal component analysis. *Journal of Wine Economics*, *16*(1), 56-67.

Beltrán, N. H., Duarte-Mermoud, M. A., Vicencio, V. A. S., Salah, S. A., & Bustos, M. A. (2008). Chilean wine classification using volatile organic compounds data obtained with a fast GC analyzer. *IEEE Transactions on Instrumentation and Measurement*, *57*(11), 2421-2436.

Cao, J. (2014). Quantifying randomness versus consensus in wine quality ratings. *Journal of Wine Economics*, *9*(2), 202-213.

Frick, H., Chow, F., Kuhn, M., Mahoney, M., Silge, J., & Wickham, H. (2023). *rsample: General Resampling Infrastructure*. https://CRAN.R-project.org/package=rsample

Friedman, J., Tibshirani, R., & Hastie, T. (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software*, *33*(1), 1-22. https://doi.org/10.18637/jss.v033.i01

Fritsch, S., Guenther, F., & Wright, M. N. (2019). *neuralnet: Training of Neural Networks*. https://CRAN.R-project.org/package=neuralnet

Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction* (Vol. 2). Springer.

James, G., Witten, D., Hastie, T., Tibshirani, R., et al. (2013). *An introduction to statistical learning* (Vol. 112). Springer.

Kassambara, A. (2023). *ggcorrplot: Visualization of a Correlation Matrix using 'ggplot2'*. https://CRAN.R-project.org/package=ggcorrplot

Liaw, A., & Wiener, M. (2002). Classification and Regression by randomForest. *R News*, *2*(3), 18-22. https://CRAN.R-project.org/doc/Rnews/

Oczkowski, E. (2016). Identifying the effects of objective and subjective quality on wine prices. *Journal of Wine Economics*, *11*(2), 249-260.

R Core Team. (2023). *R: A Language and Environment for Statistical Computing*. R Foundation

for Statistical Computing. https://www.R-project.org/

Tay, J. K., Narasimhan, B., & Hastie, T. (2023). Elastic Net Regularization Paths for All Generalized Linear Models. *Journal of Statistical Software*, *106*(1), 1-31. https://doi.org/10.18637/jss.v106.i01

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org

Wickham, H., François, R., Henry, L., Müller, K., & Vaughan, D. (2023). *dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr

Wickham, H., Vaughan, D., & Girlich, M. (2023). *tidyr: Tidy Messy Data*. https://CRAN.R-project.org/package=tidyr