

Taller de Análisis de datos - Problema de clasificación 1

Jésica Charaf e Ignacio Spiousas

12 de diciembre de 2023

Problema de clasificación 1

Estos datos son los resultados de análisis químicos de vinos provenientes de la misma región de Italia pero de 3 distintos cultivos. Cada una de las 178 filas contiene el número del cultivo seguido por los valores de 13 mediciones.

Aplique los métodos de clasificación que le parezcan convenientes y compare sus performances.

Los datos están en <http://archive.ics.uci.edu/ml/datasets/Wine>

Resolución

Análisis exploratorio

Los datos contienen 178 observaciones donde la primera variable indica el tipo de cultivo (1, 2 o 3) y las siguientes 13 variables corresponden a mediciones de: Alcohol, Malicacid, Ash, Alcalinity_of_ash, Magnesium, Total_phenols, Flavanoids, Nonflavanoid_phenols, Proanthocyanins, Color_intensity, Hue, OD280_OD315_of_diluted_wines y Proline.

Lo primero que vamos a ver es cómo se distribuyen las clases, es decir, cuántos datos pertenecientes a cada cultivo tenemos (figura 1).

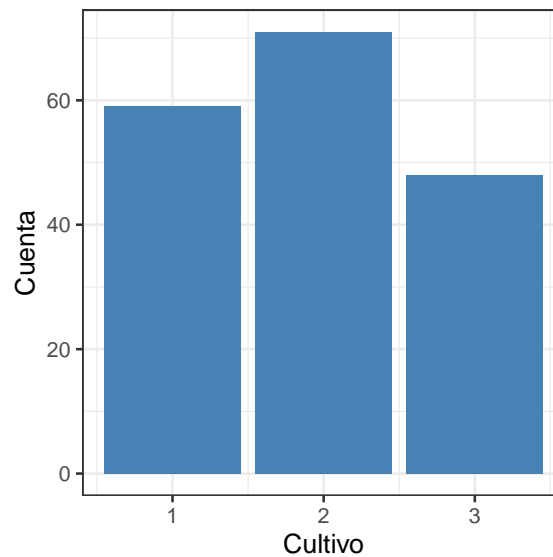


Figure 1: Cantidad de datos pertenecientes a cada clase (cultivo) en el dataset a utilizar.

Podemos ver que no contamos con grandes desbalances de clase. Se tienen 59 observaciones correspondientes al cultivo 1, 71 del cultivo 2 y 48 del cultivo 3.

NOTA: Acá poner un poco más de exploración de la relación de los cultivos con las variables.

Comparación de métodos de clasificación

El objetivo del trabajo consiste en explorar diferentes métodos de clasificación para determinar el tipo de cultivo y comparar sus desempeños. Para eso, vamos a considerar los enfoques de K vecinos cercanos, Random Forest y Regresión logística para modelos multinomiales.

Para analizar los distintos métodos de clasificación, separamos la muestra en un set de entrenamiento (dos tercios de los datos) y un set de testeo (un tercio de los datos) de forma estratificada según el cultivo, utilizando la función `initial_split` de `rsample`.

La métrica que vamos a utilizar para evaluar el modelo es el *accuracy* ya que los datos no presentan desbalances de clases marcados ni creemos que haya alguno de los errores (tipo I y tipo II) que debamos favorecer por sobre el otro.

K vecinos cercanos

El primer modelo que vamos a ajustar es el de K vecinos cercanos. Para esto consideramos una grilla de valores de k (cantidad de vecinos) entre 1 y 20. Para evaluar cuál es la cantidad de vecinos más conveniente realizamos validación cruzada separando la muestra de entrenamiento en 10 folds estratificando según la clase. Estos *folds* son generados utilizando la función `vfold_cv` del paquete `rsample`.

Para utilizar el modelo de KNN primero vamos a escalar los datos.

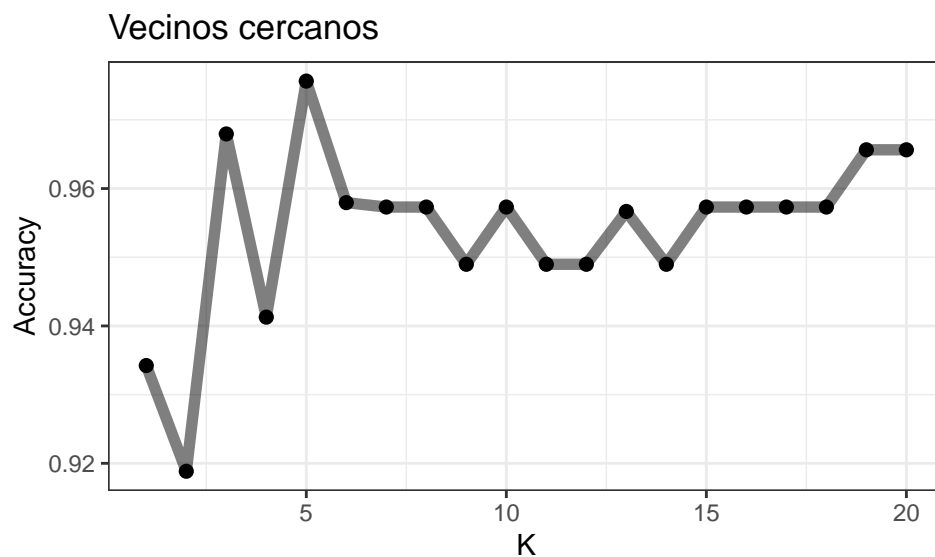


Figure 2: Accuracy en función de la cantidad de vecinos cercanos.

Como puede verse en la figura 2, el máximo de Accuracy para los modelos de KNN es de 0.98 para 5 vecinos.

Random Forest

La siguiente alternativa que vamos a considerar es un modelo basado en ensambles de árboles conocido como Random Forest. En este caso también utilizaremos validación cruzada para hallar la combinación de parámetros que maximice el Accuracy. Los hiperparámetros a optimizar en un 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`).

Vamos a calcular el Accuracy para una grilla de 160 filas, con $1 \leq \text{mtry} \leq 10$ y $5 \leq \text{min_n} \leq 20$. Al igual que en KNN, utilizaremos los datos estandarizados (aunque en este caso no debería afectar a los resultados).

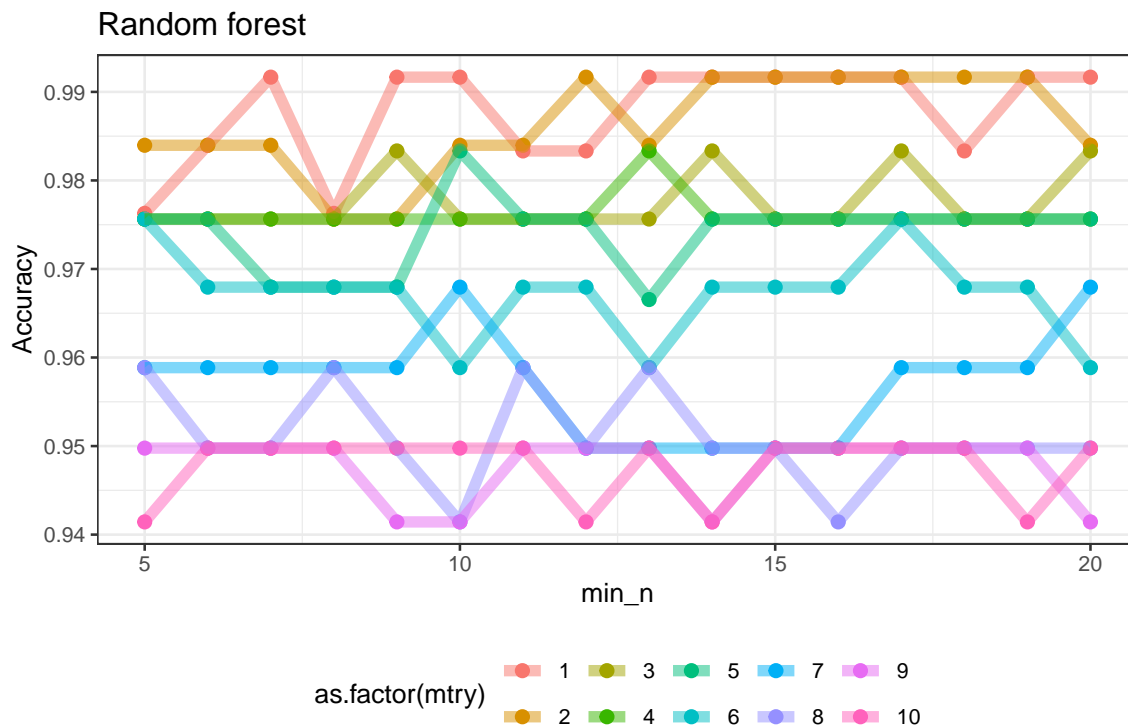


Figure 3: Accuracy en función de mtry y minn para Random Forest.

Como puede verse en la figura 3, el máximo valor de accuracy vale 0.992 para `m_try` igual a 1 y `min_n` igual a 7.

Regresión logística

El último enfoque que vamos a considerar es el de regresión logística para modelos multinomiales. Ajustaremos una familia de modelos con regularización Lasso utilizando el paquete `glmnet`.

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 realizando validación cruzada de la misma forma que en los otros métodos.

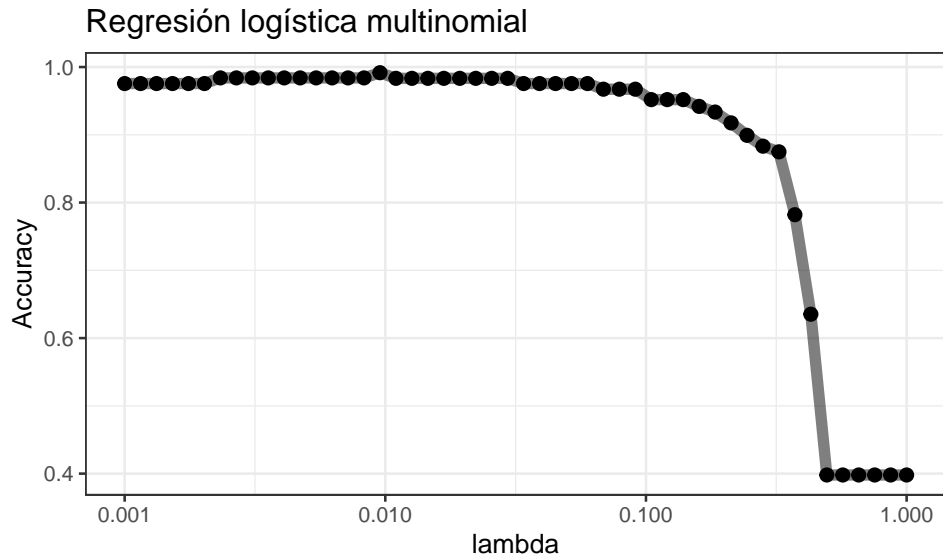


Figure 4: Accuracy en función del valor de lambda.

En la figura 4 observamos los resultados del Accuracy en función de λ y vemos que el máximo se alcanza en λ igual a 0.01 con un valor de 0.992.

Redes neuronales

Finalmente vamos a considerar un modelo de redes neuronales para preecir el cultivo. Por simplicidad el modelo contiene una sola capa intermedia y por medio de validación cruzada vamos a determinar cuántas neuronas tiene esta capa.

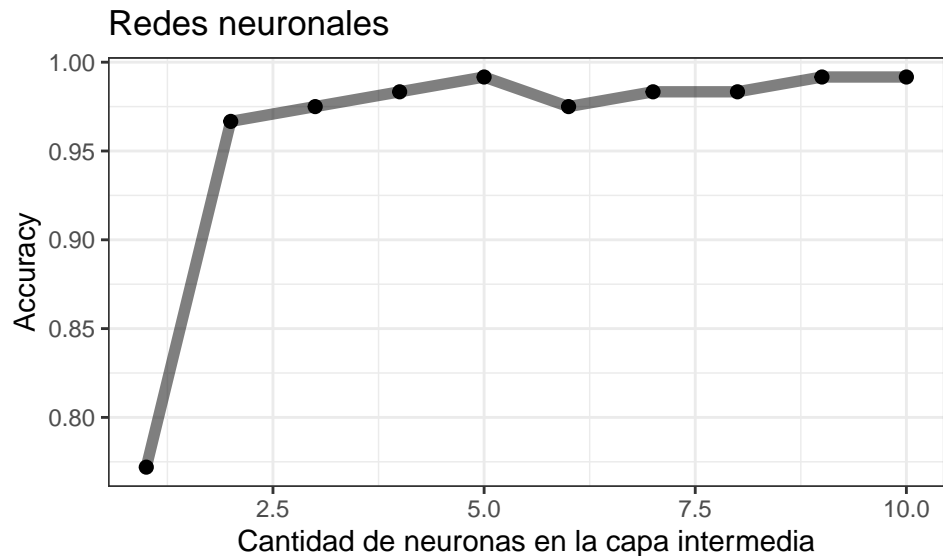


Figure 5: Accuracy en función de la cantidad de neuronas en la capa intermedia.

En la figura 5 observamos los resultados del Accuracy en función de la cantidad de neuronas de la capa intermedia y vemos que el máximo se alcanza en 5 con un valor de 0.992.

En la figura 6 podemos ver una representación del modelo que elegimos mediante validación cruzada.

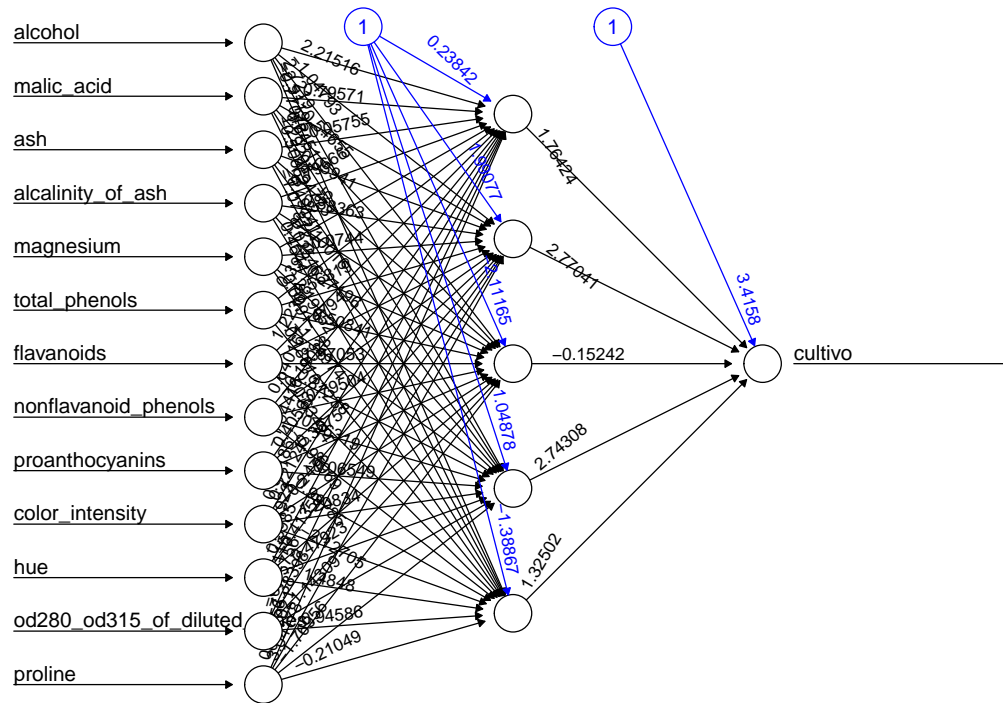


Figure 6: Representación esquemática de la red neuronal ajustada a partir de la selección de parámetros con validación cruzada.

Evaluación del modelo elegido