

Taller de Análisis de datos - Problema de clasificación 0

Jésica Charaf e Ignacio Spiousas

24 de noviembre de 2023

Problema de clasificación 0

El archivo Distrofia-info contiene una descripción de la Distrofia Muscular de Duchenne (DMD), para cuyo diagnóstico se realizó un estudio cuyos resultados están en el archivo Distrofia-Data. La primera fila es:

```
38    1      1      1 1007 22 6 0 079 52.0 83.5 10.9 176
```

Las primeras 5 columnas no sirven. “22” es la edad, “6” el mes, “0” no sirve, “079” el año, y las últimas cuatro son CK, H, PK y LD. El objetivo es proponer una regla para detectar la DMD usando las cuatro variables observadas (enzimas), y estimar su error de clasificación. Se plantean algunas preguntas:

- CK y H son más baratas de medir que PK y LD. ¿Cuánto aumenta el error si se prescinde de estas últimas?
- ¿Tiene sentido incluir la edad entre los predictores?
- La sensibilidad y la especificidad son respectivamente las probabilidades de identificar correctamente a sujetos enfermos y sanos. ¿Cómo elegir el balance entre ambas?
- Se sabe que la probabilidad de que una mujer sea portadora es $1/3200$. ¿Tiene alguna utilidad ese dato?

Resolución

Análisis exploratorio

El conjunto de datos contiene 209 observaciones dentro de las cuales 134 corresponden a la clase no portadora y 75 a la clase portadora. Las variables involucradas son:

- Edad
- Mes
- Año
- CK
- H
- PK
- LD

Para empezar, inspeccionamos los datos y detectamos observaciones con valores faltantes registrados como “-9999” dentro de las variables PK y LD. Estos valores fueron reemplazados por el promedio de los valores registrados en cada variable por grupo.

En la figura 1 realizamos boxplots de las edades según cada clase y podemos ver que aparentemente no hay representatividad de edades más grandes en la muestra sana, lo cual parece estar vinculado a cómo se tomó la muestra y no necesariamente a que haya una relación directa entre la edad y la pertenencia a alguna de las clases. Esta heterogeneidad podría enmascarar el efecto de los marcadores en el diagnóstico por la edad y es por este motivo que no incluiremos esta variable como predictora.

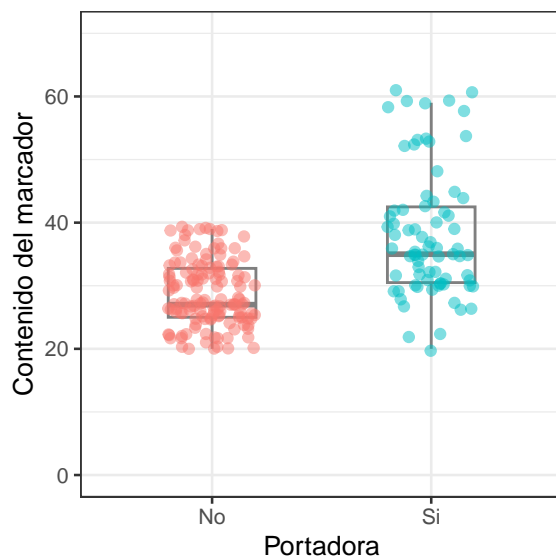


Figure 1: Dependencia de la medición de cada marcador con la condición de portadora de la persona. Los puntos indican los datos individuales mientras que la barra gris indica el promedio para cada categoría.

Una vez limpiados los datos, consideramos las 4 columnas que indican los valores detectados de ciertos marcadores (CK, K, PK y LD) y una columna con los valores “Sí” o “No” que indica si la persona es portadora o no.

En la figura 2 se observa la medición de cada marcador dependiendo si la persona es o no portadora. A simple vista podemos ver que, en promedio (barra gris), la medición de los cuatro marcadores es superior cuando la persona es portadora. Sin embargo, pareciera que **H** es el que separa menos eficientemente ambos grupos.

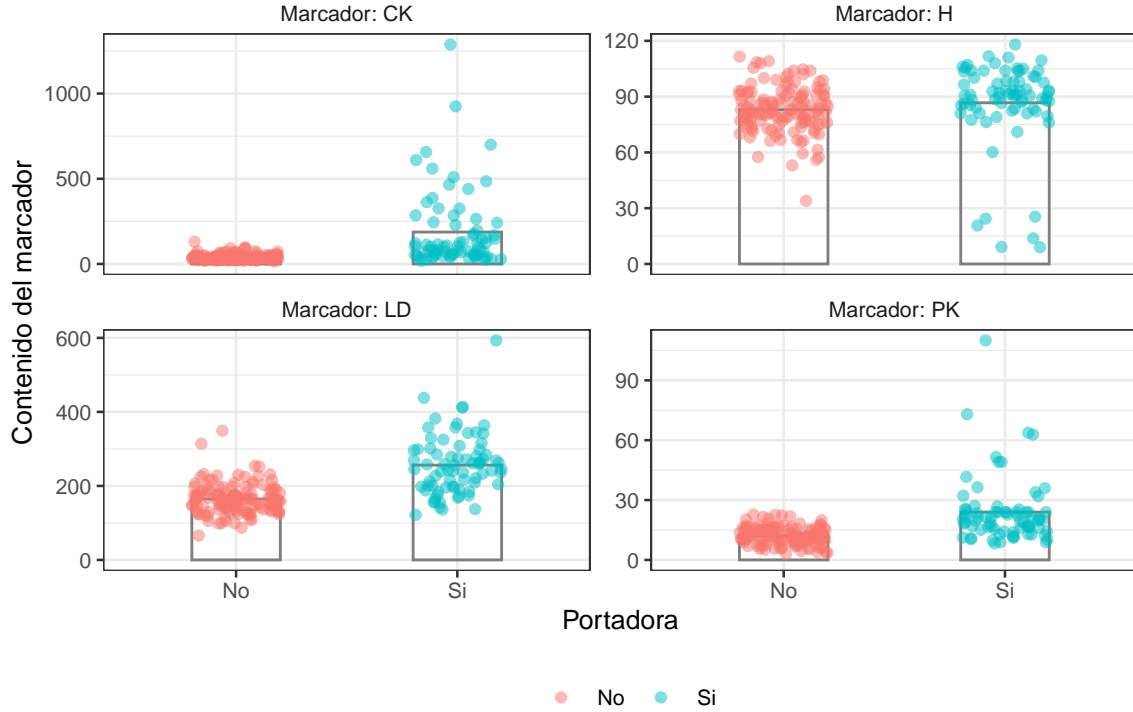


Figure 2: Dependencia de la medición de cada marcador con la condición de portadora de la persona. Los puntos indican los datos individuales mientras que la barra gris indica el promedio para cada categoría.

El objetivo del presente trabajo consiste en entrenar un modelo que, en principio, a partir de las cuatro mediciones de marcadores prediga si la persona es portadora o no. Para esto vamos a evaluar un número de modelos de clasificación: Regresión logística, K vecinos cercanos y Random Forest. Luego de elegir qué modelo es el más conveniente para el problema y ajustar sus hiperparámetros y parámetros evaluaremos su capacidad de predicción en un set de *testeo*.

Elección del método de clasificación

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 la clase, utilizando la función `initial_split` de `{rsample}`.

La métrica a utilizar para evaluar el modelo depende del caso en particular de estudio y del tipo de error que consideremos que es más grave cometer o se priorice evitar. En nuestro caso vamos a considerar la medida *F1* teniendo en cuenta que provee un balance entre *recall* y *precision*.

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 implementar este modelo utilizaremos las funcionalidades del paquete `{tidymodels}`.

Para cada valor de *k*, calculamos el promedio de los *F1* obtenidos en cada fold y seleccionamos el valor de *k* que maximice dicho promedio. Con este criterio, el valor de *k* obtenido es 9 con un valor de *F1* de 0.91. En

la figura 3 podemos ver los valores promedios de los F1 en función de la cantidad de vecinos utilizados.

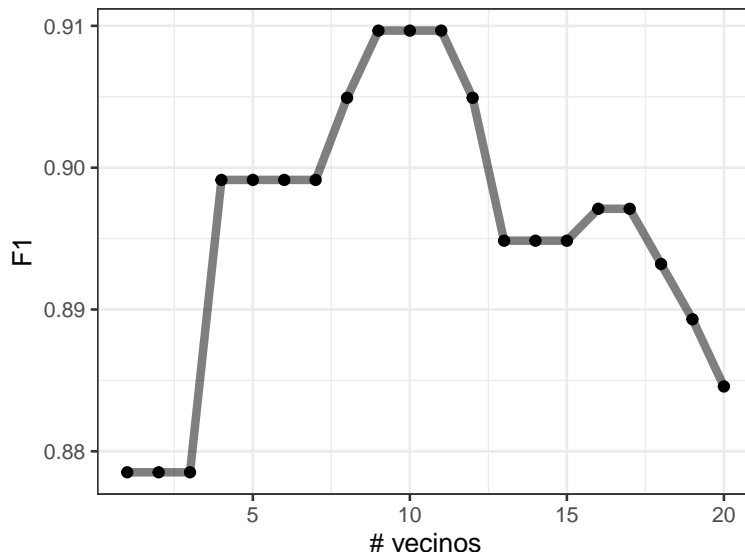


Figure 3: Dependencia de F1 con la cantidad de vecinos para el modelo de vecinos cercanos.

Regresión logística

Para continuar, otro enfoque que exploramos es el de regresión logística considerando una familia de modelos lineales generalizados con regularización Lasso.

En este caso, tomamos una grilla de 50 valores de λ equiespaciados en escala logarítmica entre 10^{-3} y 10^0 y, a la vez, consideramos distintos valores de umbral de clasificación p entre 0.2 y 0.8 con paso 0.1.

Al igual que antes, realizamos validación cruzada tomando 10 folds y evaluando para cada valor de λ y de p los resultados del F1 obtenido a partir del ajuste del modelo lineal generalizado correspondiente asignándole pesos $1 - \frac{\# \text{casos de la clase}}{\# \text{casos totales}}$ a las observaciones correspondientes a cada clase.

El valor máximo obtenido para el F1 es 0.923 y se alcanza para un umbral de $p = 0.3$ y un $\lambda = 0.02223$.

Random forest

Como última alternativa vamos a considerar un modelo basado en árboles, más precisamente un modelo de random forest. En este caso también utilizaremos validación cruzada para hallar la combinación de parámetros que maximice F1. Los parámetros que vamos a optimizar son: El número de variables que se consideran en cada split del árbol aleatorio que puede tomar valores enteros de 1 a 4 (**mtry**), y el número máximo muestras en cada hoja que puede tomar valores enteros mayores a 1 (**min_n**). Para esto calcularemos el F1 para cada combinación de **mtry** y **min_n** de una grilla que considera valores enteros y rangos $1 \leq \text{mtry} \leq 4$ y $1 \leq \text{min_n} \leq 40$.

Para implementar este modelo, y al igual que en el caso de K vecinos cercanos, utilizaremos las funcionalidades del paquete *{tidymodels}*.

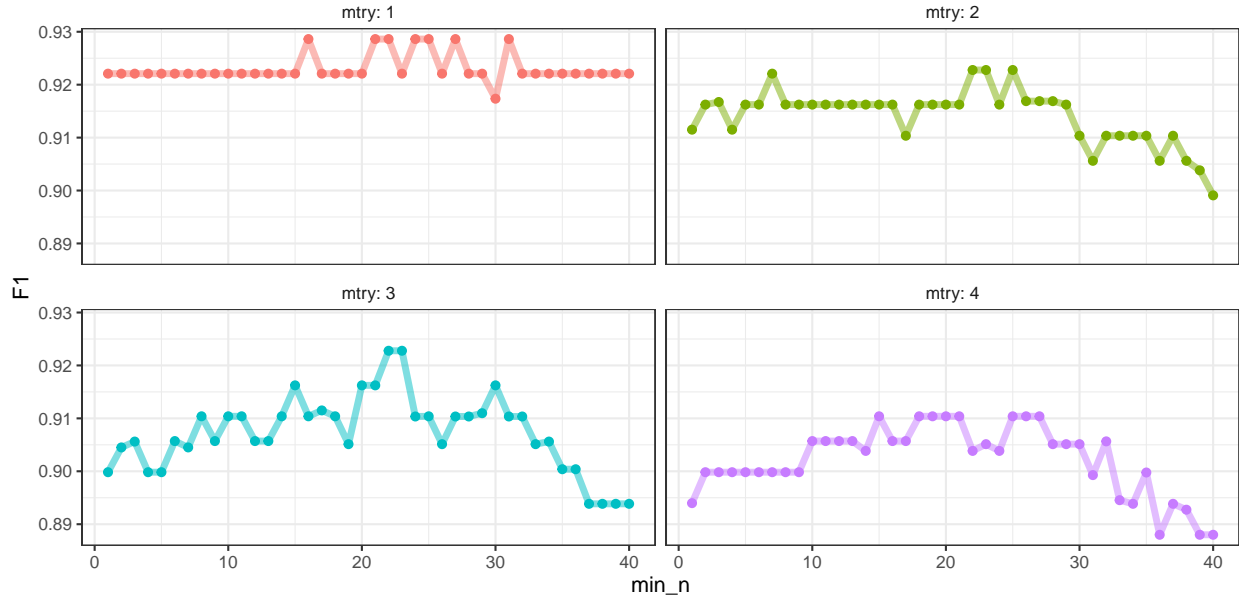


Figure 4: Dependencia de F1 con los parámetros de random forest.

En la figura 4 puede verse la dependencia de F1 con los parámetros de la grilla. Si bien no pareciera haber una clara dependencia con `min_n` si la habría con `mtry`, con valores de F1 más altos para `mtry` igual a 1 y decreciendo para valores más grandes. El valor máximo obtenido para el F1 es 0.929 y se alcanza para `mtry` igual a 1 y `min_n` igual a 16.

Evaluación del mejor modelo en los datos de *testeo*

En nuestro caso, los mejores modelos de cada tipo tuvieron una performance similar. Es por eso que seleccionaremos el modelo más simple e interpretable, es decir, GLM con regularización LASSO con parámetros $p = 0.3$ y $\lambda = 0.02223$.

Finalmente, vamos a entrenar el modelo con todos los datos de entrenamiento y evaluar su performance con los datos de testeo. La matriz de confusión que se oobtiene es la siguiente:

	0	1
0	43	8
1	2	17

A la que le corresponde un valor de F1 de 0.896.

Conclusiones