Multivariate Adaptive Regression Splines

Calvo, Gil, Tame

11/28/2021

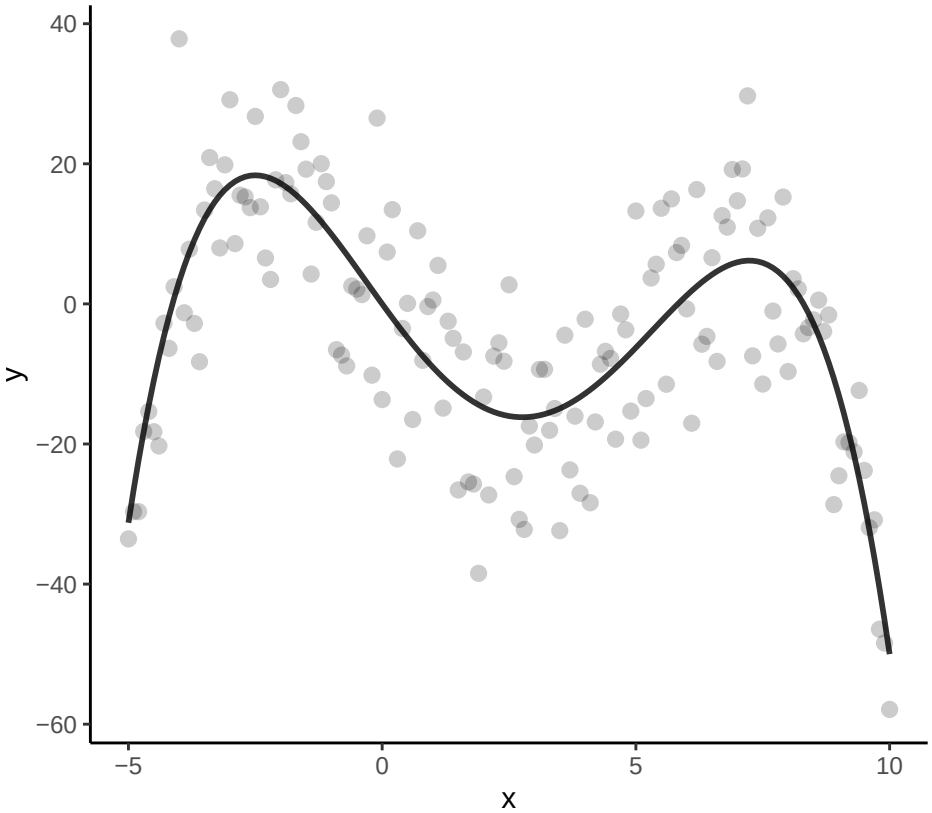
Introducción y Relevancia

El modelo MARS es un algoritmo de aprendizaje supervisado que funciona para regresión y clasificación. Es una versión generalizada de regresión lineal a pedazos [6], en el cual el modelo permite hacer diferentes pendientes para diferentes partes de la variable a estimar en función de las variables predictoras, y automáticamente modela términos no lineales e interacciones entre variables. Fue introducido originalmente en [1] por Friedman en 1991, y existen varias implementaciones del algoritmo, generalmente bajo el nombre de "Earth" [8].

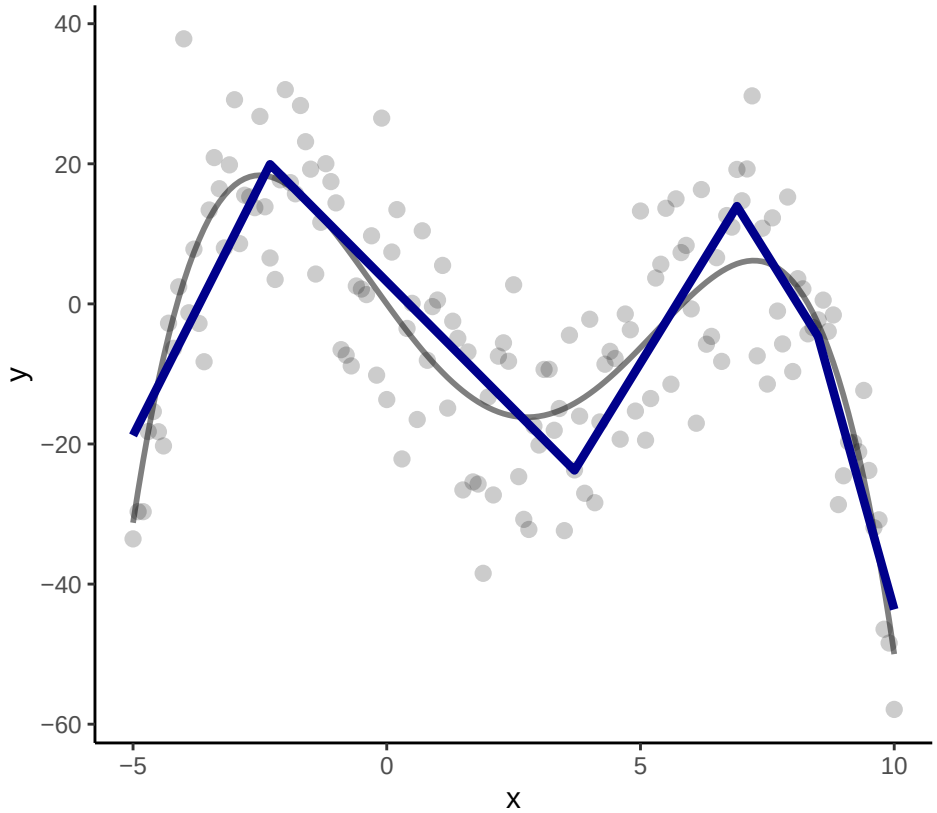
Comportamiento del Modelo

MARS busca los puntos de corte y las pendientes óptimas para aproximar a la variable objetivo. Se puede apreciar como en diferentes segmentos de la variable x existen pedazos de funciones lineales con diferentes pendientes.

$$y = -\frac{1}{20}x^4 + \frac{1}{2}x^3 + \frac{1}{2}x^2 - 10x + \varepsilon$$

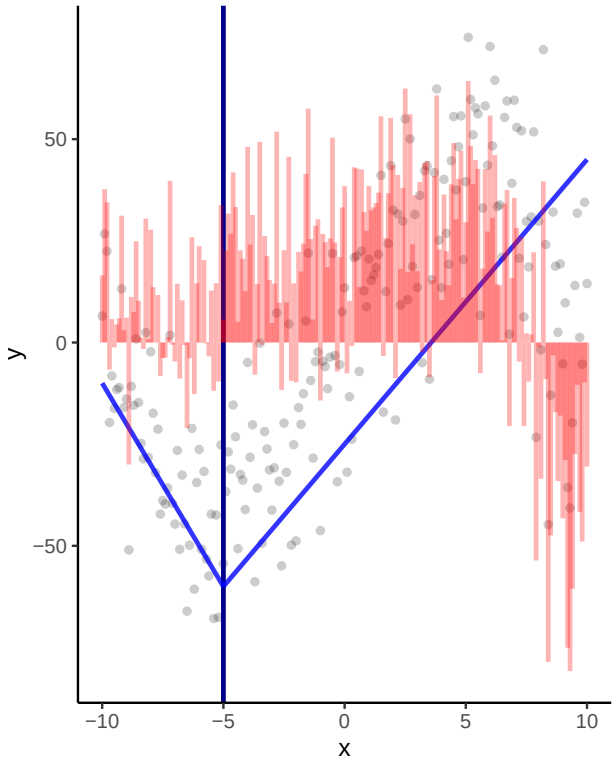


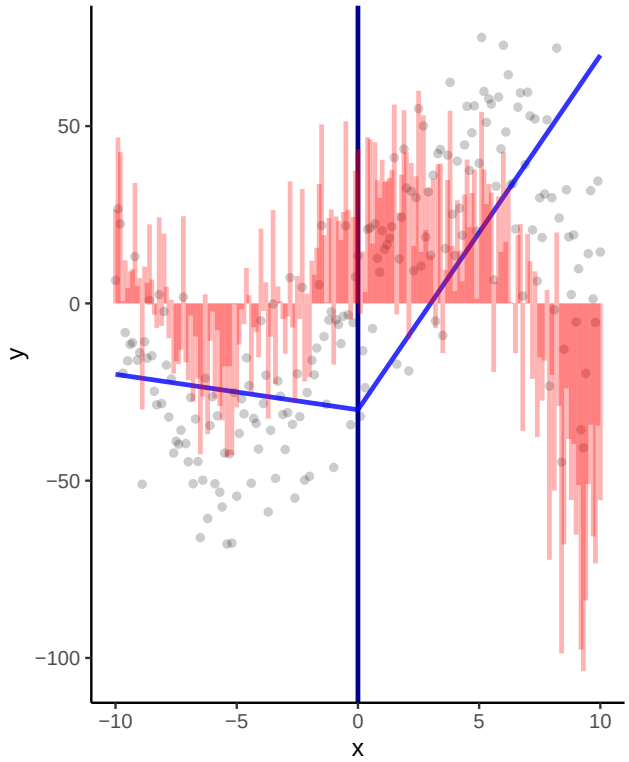
MARS con 4 nodos

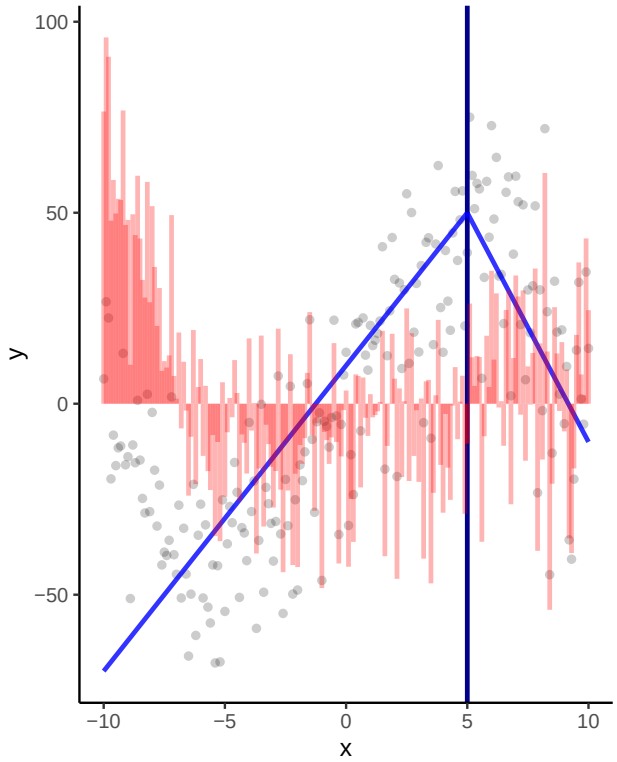


Comportamiento del Modelo

Como podemos ver, darnos a la tarea de encontrar manualmente cada punto de corte y pendientes óptimas se puede traducir a minimizar alguna métrica como el error cuadrático medio, o análoigamente, maximizar la R^2:

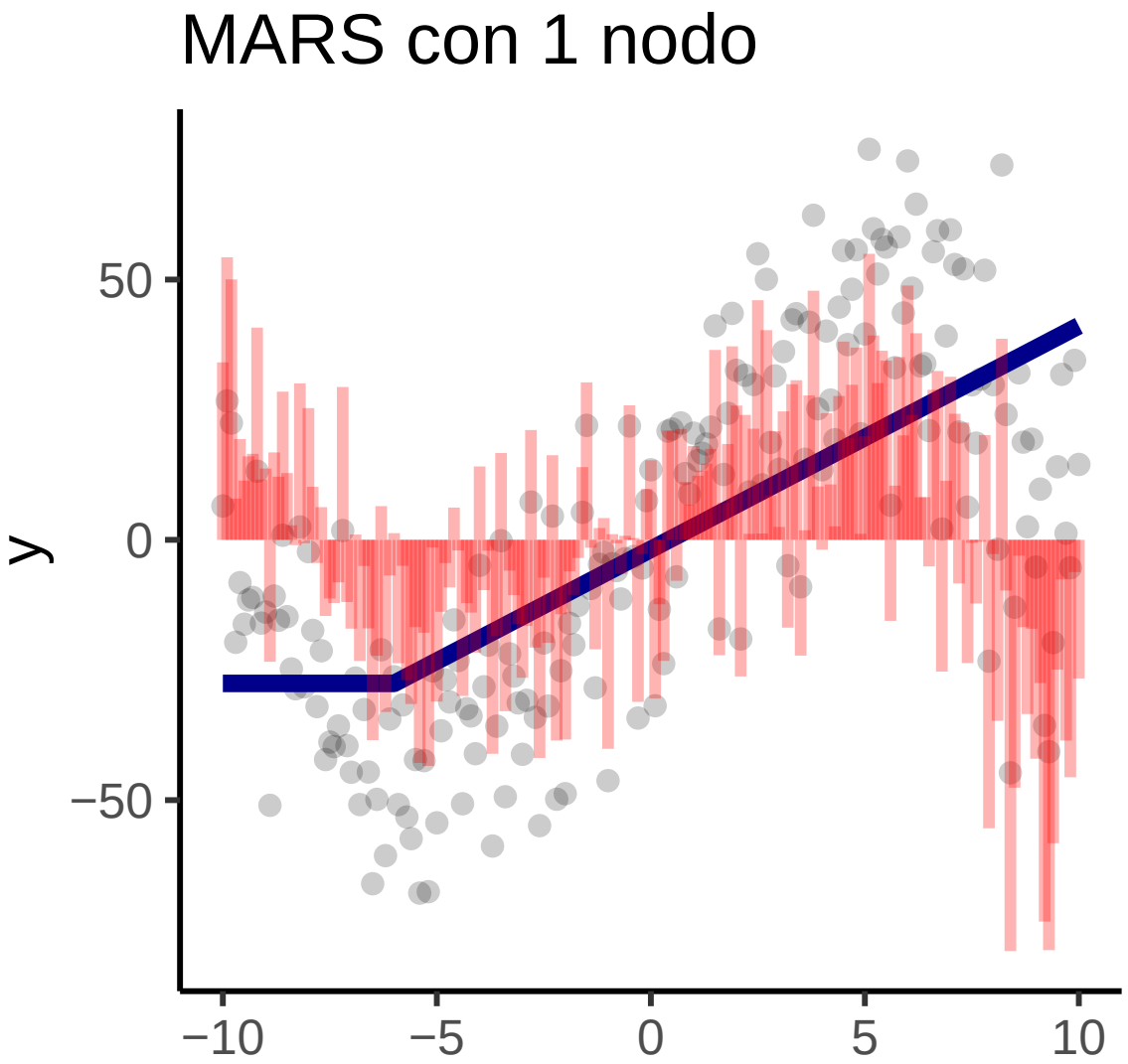






Comportamiento del Modelo

Lo que hace MARS es encontrar sistemáticamente los mejores puntos de corte y pendientes para minimizar el error de entrenamiento:



Descripción del Modelo

MARS es un modelo de regresión no paramétrico que a través de funciones bisagra (*hinge functions*).

Se puede considerar como una generalización de regresión lineal a pedazos o como modificación al método CART con el fin de mejorar su desempeño en el contexto de regresión [6].

Descripción del Modelo

MARS se define con el uso de funciones que son lineales por partes, de la forma $(x - t)_+$ y $(t - x)_+$. Estas funciones toman el máximo entre 0 y el valor dentro de la funcion, por lo tanto,

$$(x - t)_+ = \max\{0, x - t\} = \begin{cases} x - t & \text{si } x > t, \\ 0 & \text{en otro caso.} \end{cases}$$

Cada función es lineal a trozos con un cambio de pendiente en el valor t, lo cual las hace splines lineales, y a cada par dividido en el valor t se le llama un *par reflejado*.

Descripción del Modelo

La idea del método es formar pares reflejados para cada variable de entrada X_j con cambios de pendiente en cada valor observado x_{ij}.

Por lo tanto, para el conjunto de variables X_j, la colección de funciones base es:

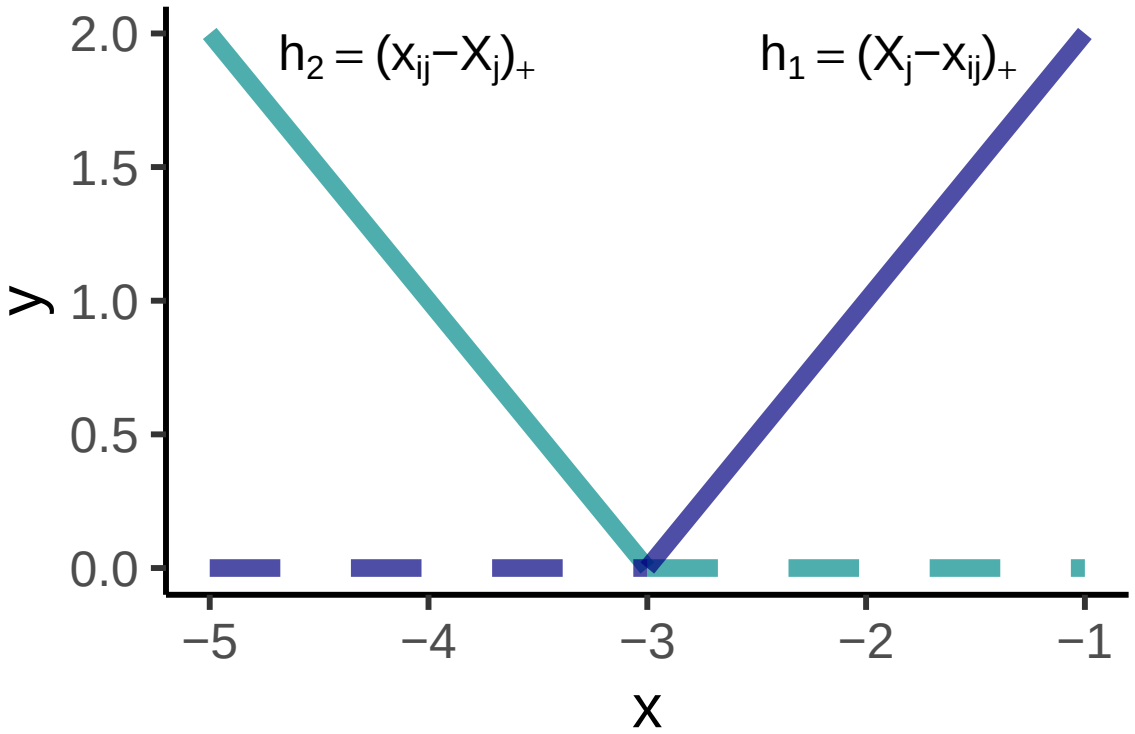
$$\mathcal{C} = \{(X_j - t)_+, (t - X_j)_+\}, \text{ con } t \in \{x_{1,j}, x_{2,j}, \ldots x_{N,j}\}, \text{ y } \ j = 1, 2, \ldots, p.$$

Descripción del Modelo

Cada par reflejado se ve de la siguiente forma:

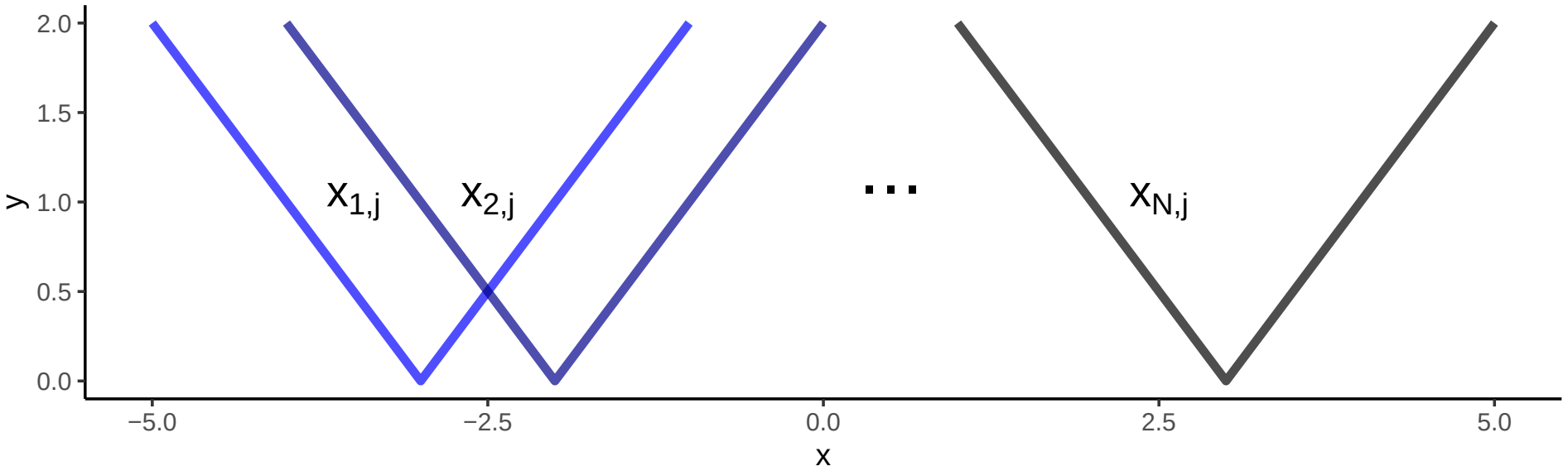
$$h_1(X) = h(X_j - x_{ij})$$

$$h_2(X) = h(x_{i,j} - X_j)$$



Descripción del Modelo

Entonces, para cada X_j, el conjunto de los pares reflejados candidatos que habitan en cada uno de los x_{ij} puntos de tal variable es:



Descripción del Modelo

El modelo que utilizamos entonces para juntar todas las variables es uno aditivo en β:

$$f(X) = \beta_0 + \sum_{m=1}^{M} \beta_m h_m(X),$$

donde cada función $h_m(X)$ es elemento de $\mathcal{C}$ o una combinación lineal de estas funciones base.

Descripción del Modelo

- El par reflejado que se agrega es aquel que minimice el error de entrenamiento.
- Este proceso se repite para cada uno de los pares reflejados restantes, resolviendo OLS (ordinary least squares) para determinar el nuevo β, y agregando aquel que continue minimizando el error.
- El criterio de paro puede ser ya sea que ninguno de los pares reflejados restantes reduzca suficiente el error (en términos absolutos o relativos), o hasta que tengamos un número determinado de variables en el modelo.

Descripción del Modelo

- Este llevará a sobreajuste de los datos, pero esto es lo que se está buscando en esta parte del procedimiento al minimizar el error de entrenamiento.
- Ya teniendo $\mathcal{M}$, se empieza la segunda parte del ajuste del modelo, que es la parte de podarlo.
- En este caso, se van eliminando los términos $h_i(X)$ iterativamente, empezando por el que produce el menor incremento en el error cuadrático residual cuando se quita.
- El proceso de eliminación se hace utilizando la métrica de validación cruzada generalizada para ahorrar costo computacional:

$$GCV(\lambda) = \frac{\sum_{i=1}^{N}(y_i - \hat{f}_\lambda(x_i))^2}{\left(\frac{1-M(\lambda)}{N}\right)^2},$$

Implementación

Para nuestra implementación, utilizamos una base de datos de `spam`, tomada de [3]. Es una colección de palabras y caracteres que aparecen comúnmente en mensajes que son *spam*. La variable respuesta es una variable categórica que tiene dos niveles: `spam`, `no_spam`. Más información se puede ver en [4].

Presentamos una tabla de algunas de las variables predictoras y la variable respuesta.

Implementación

Podemos hacer predicciones con este modelo. Por ejemplo, en entrenamiento tenemos las predicciones de clase dan:

```
## # A tibble: 4 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary          0.96
## 2 sens     binary          0.97
## 3 spec     binary          0.94
## 4 roc_auc  binary          0.99
```

Implementación

En prueba:

```
## # A tibble: 4 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary          0.94
## 2 sens     binary          0.96
## 3 spec     binary          0.92
## 4 roc_auc  binary          0.98
```

Implementación

En la siguiente tabla, podemos ver los coeficientes que se usan para este modelo final (los primeros 10).

```
##                                                    spam
## (Intercept)                                  0.199122835
## h(0.257-cfexc)                              -0.245063537
## h(0.088-cfdollar)                            7.809551831
## h(wfremove-0.29)*h(0.088-cfdollar)           0.627252683
## h(0.29-wfremove)*h(0.088-cfdollar) -12.356605563
## h(wffree-0.41)*h(0.088-cfdollar)             0.238958028
## h(0.41-wffree)*h(0.088-cfdollar)            -4.768347722
## h(0.52-wfhp)*h(216-crltotal)                -0.001157403
## h(0.52-wfhp)*h(0.44-wfedu)                   1.012531128
## h(0.52-wfhp)*h(wfgeorge-0.08)                0.016298821
```

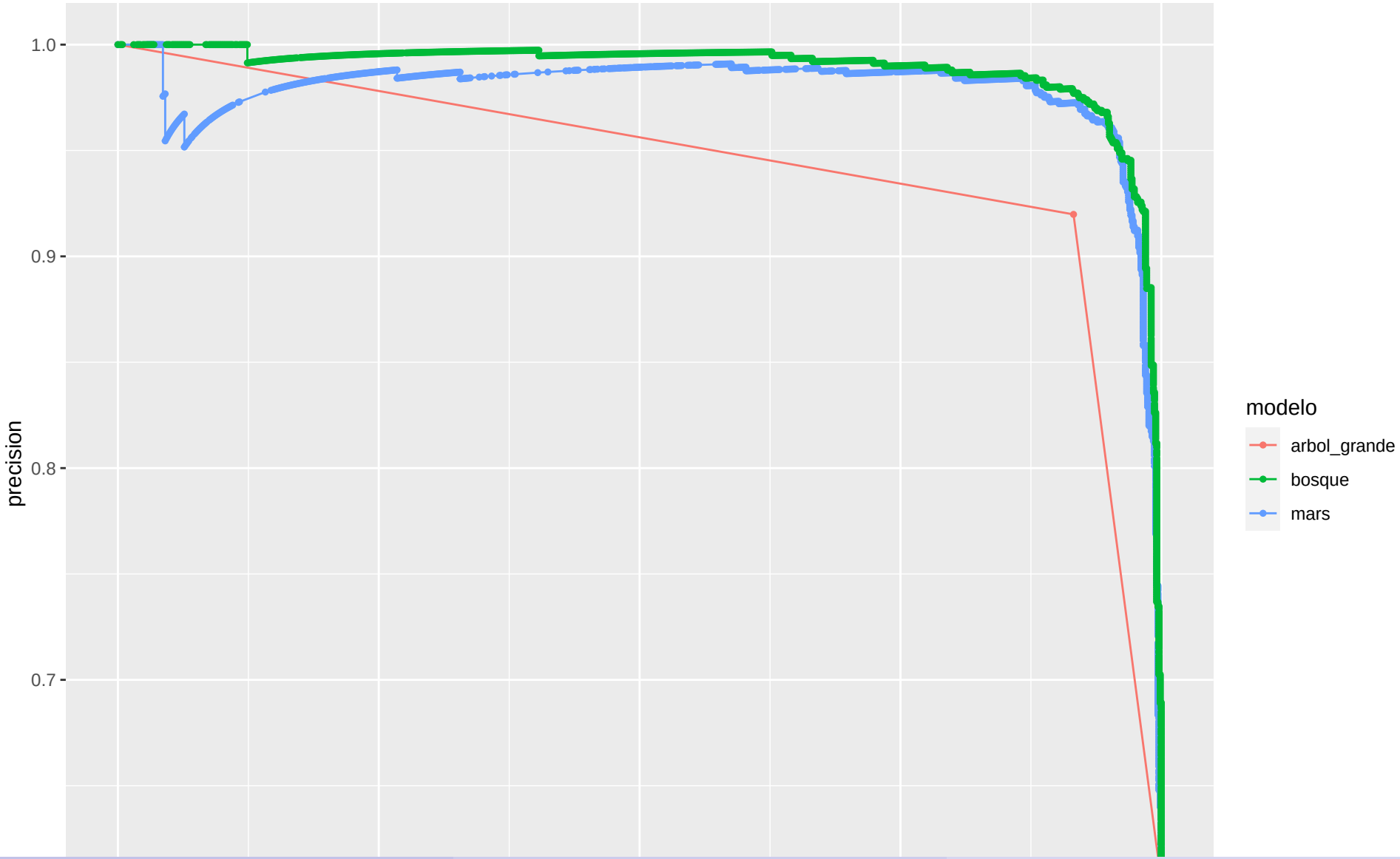
Comparación con Árboles y Bosques Aleatorios

Ajustamos otros dos modelos, un árbol de decisiones y bosques aleatorios, y podemos comparar el ajuste de MARS contra estos modelos. Las implementaciones para estos datos la tomamos de [5].

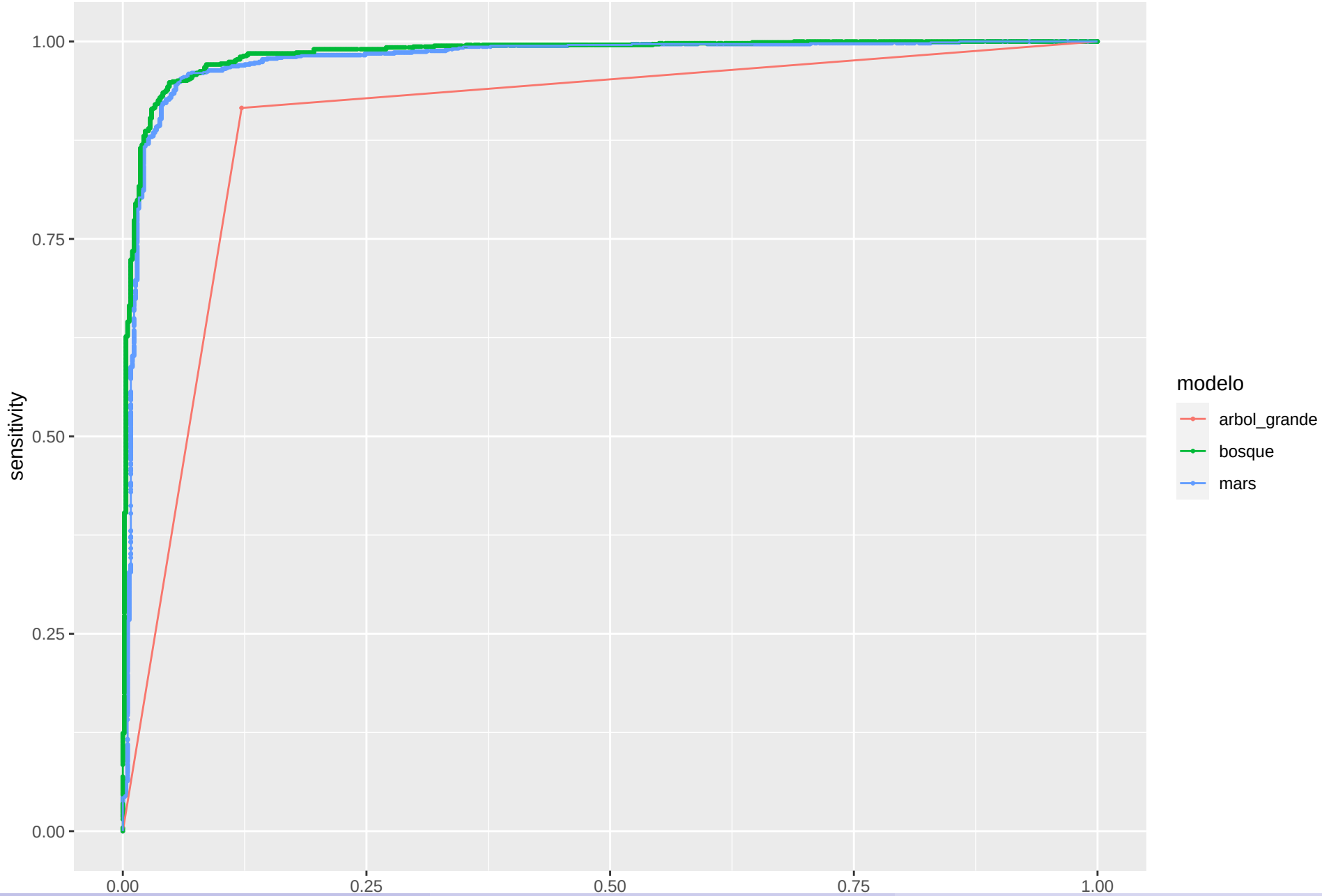
```
## # A tibble: 4 x 5
##   .metric  .estimator estimate_mars estimate_arboles estima
##   <chr>    <chr>              <dbl>            <dbl>
## 1 accuracy binary              0.94              0.9
## 2 sens     binary              0.96              0.92
## 3 spec     binary              0.92              0.88
## 4 roc_auc  binary              0.98              0.9
```

Curva Precision-Recall

Podemos visualizar esta diferencia con una gráfica de precision-recall:



Curva ROC


1.00
0.75
0.50
0.25
0.00
sensitivity
0.00
0.25
0.50
0.75
1.00
modelo
arbol_grande
bosque
mars

Ventajas

- Al ser lineal, resulta ser relativamente parsimonioso; preserva bastante interpretabilidad de los coeficientes, aún en presencia de interacciones.
- Funciona bien tanto para baja como alta cardinalidad.
- No es computacionalmente costoso.
- Hace selección de variables automáticamente, tanto por si solas como interacciones.
- Preciso si localmente es correcto hacer aporximaciones lineales.

Desventajas

- Existen modelos que tienen mejor desempeño predictivo; vimos arriba que bosques aleatorios funcionó casi igual, aunque en general este supera a MARS.
- Paquetes como earth no incluyen funciones de órdenes mayores (aunque, de acuerdo a [6], el hacer interacciones de un orden ayuda a la interpretabilidad al hacer superficies igual a 0 donde no queremos aproximar).
- Poco preciso si en general hacer las relaciones lineales, aún localmente, es incorrecto.

Conclusiones

El modelo de MARS es un modelo bastante bueno y aplicable a varios tipos de problemas. Mientras que su uso principal es en problemas de regresión, presentamos un ejemplo en el que se aplica para clasificación. Aún así, tuvo un desempeño comparable con Bosques Aleatorios.

Si se tienen interacciones relativamente lineales y una variable a predecir continua, entonces el algoritmo tiene muchas ventajas relativo a por ejemplo árboles o una regresión lineal simple.

Referencias

- [1] Friedman, J. H. (1991). "Multivariate Adaptive Regression Splines". The Annals of Statistics. *19* (1): 1–67.
- [2] Notas sobre MARS
- [3] Base de datos Spam
- [4] Datos de Spam
- [5] Árboles aleatorios y bosques
- [6] Hastie, T., et al. "The Elements of Statistical Learning", Springer Series in Statistics, ISBN 0172-7397, 745 pp. (2009): 291.

Referencias

- [7] Stone, C. J., et al. "Polynomial splines and their tensor products in extended linear modeling: 1994 Wald memorial lecture." The Annals of Statistics *25.4* (1997): 1371-1470.
- [8] Stephen Milborrow. Derived from mda:mars by Trevor Hastie and Rob Tibshirani. Uses Alan Miller's Fortran utilities with Thomas Lumley's leaps wrapper. (2021). earth: Multivariate Adaptive Regression Splines. R package version 5.3.1. https://CRAN.R-project.org/package=earth
- [9] Video del cual basamos algunas gráficas
- [10] Multivariate adaptive regression splines
- [11] How to improve a linear regression with Mars
- [12] Wikipedia de MARS