Clasificación

Barrera Borla, Gonzalo y Duarte, Octavio M.

27 de Noviembre

Epílogo

Ignorante que blasonas de sabio: te veo angustiado entre el infinito del pasado y el infinito del porvenir. Quisieras poner límite entre estos dos infinitos y detenerte… Siéntate antes bajo un árbol con un cántaro de vino y olvidarás tu impotencia.

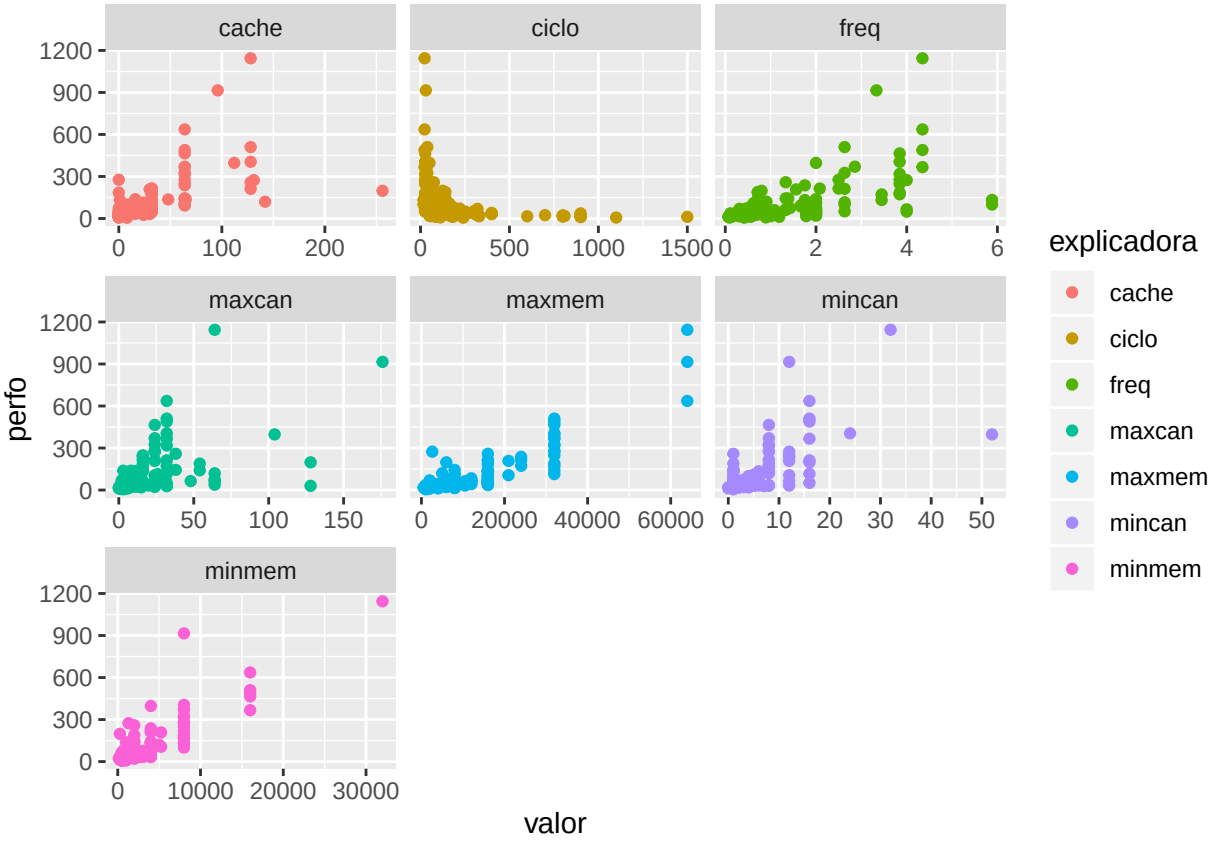
Omar Jayam, Rubaiyat.

Carga de los Datos

Exploración

Parece natural introducir una transformada del tiempo de ciclo de procesador que sea la inversa, "freq". Sobre todo porque la relación de "más es mejor" es más intuitiva.

Gráfico de Facetas



En este gráfico anticipamos algunos patrones que pueden parecer preocupantes. La forma cónica en que se disponen los puntos parece indicar que si se adaptara un modelo lineal (al menos uno monovariado respecto

a cualquiera de ellas) este sería heteroscedástico. Es muy probable que alguna transformada de la variable `perfo` nos de un mejor rendimiento.

Correlaciones

	ciclo	minmem	maxmem	cache	mincan	maxcan	perfo	freq
ciclo	1.0000	-0.3274	-0.3836	-0.3093	-0.3073	-0.2501	-0.3050	-0.5593
minmem	-0.3274	1.0000	0.7850	0.4982	0.5913	0.2761	0.8561	0.6785
maxmem	-0.3836	0.7850	1.0000	0.5112	0.5692	0.4445	0.8745	0.6555
cache	-0.3093	0.4982	0.5112	1.0000	0.6244	0.5467	0.6541	0.4364
mincan	-0.3073	0.5913	0.5692	0.6244	1.0000	0.5603	0.6729	0.5068
maxcan	-0.2501	0.2761	0.4445	0.5467	0.5603	1.0000	0.5434	0.2059
perfo	-0.3050	0.8561	0.8745	0.6541	0.6729	0.5434	1.0000	0.6064
freq	-0.5593	0.6785	0.6555	0.4364	0.5068	0.2059	0.6064	1.0000

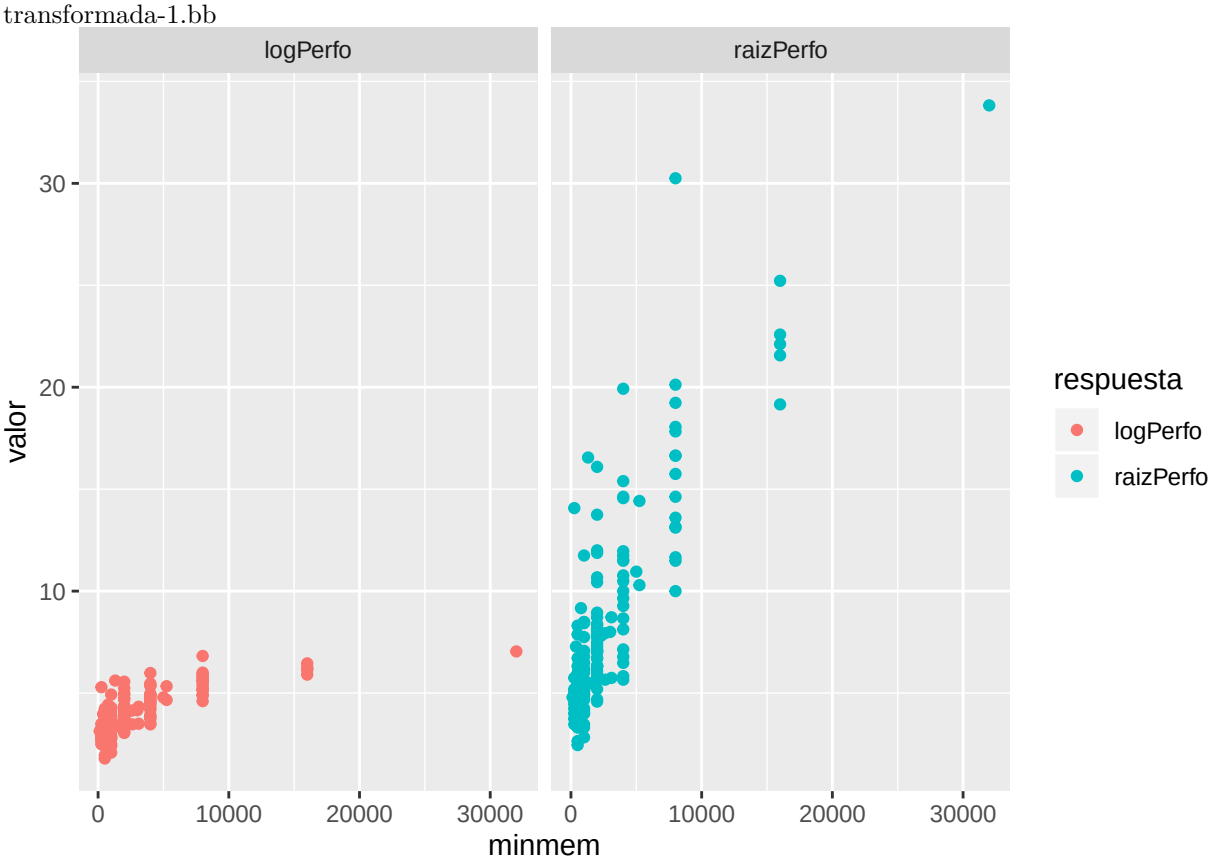
- La mejor correlación con la variable en estudio la presentan las variables asociadas a la memoria `minmem` con 0.8561 y `maxmem` 0.8745.
- La correlación entre ellas mismas es alta 0.785.
- Excepto `ciclo` todas las demás tienen una correlación suficiente como para suponer que se puede extraer información útil al modelo de ellas. La transformada de `ciclo`, `freq` sí muestra una correlación razonable.

Intento de Transformar la variable Objetivo

Vemos si disminuyen estos patrones gráficos preocupantes.

Transformando la Repuesta

transformada-1.bb



Ambas transformaciones parecen haber reducido drásticamente la forma cónica que ingenuamente sugiere heteroscedasticidad. Vale la pena comparar el rendimiento de estas transformaciones.

Elección de un Método

Armados de estas nociones para orientarnos, recurrimos a iterar una lista muy grande de modelos sobre los métodos que conocemos. Dadas las particulares circunstancias de una competencia, decidimos priorizar la precisión de las predicciones sobre otras cualidades usualmente deseables como la interpretabilidad.

Un paquete que homogeneiza esta la tarea de entrenar modelos para una gran cantidad de algoritmos es `caret`. Este separa la tarea de realizar una predicción en varias etapas, todas modulares y por lo tanto fue posible programar la métrica particular que estamos usando en esta trabajo, *alfa podada* al 80%. El paquete tiene interfases para una gran cantidad de librerías que abarcan regresiones de muchas clases. Intentamos con todas las que nos parecieron razonables, cubriendo la mayoría de los métodos que conocemos y algunos nuevos.

Además, se definió esta medición de tal forma que acepta una función inversa. De esta manera, pudimos realizar regresiones sobre modelos donde la variable está transformada (dado que las observaciones preliminares parecieron revelar la ventaja de estas transformaciones, $y_1 = \log y$ e $y_2 = \sqrt{y}$) pero medir el *Error Medio Cuadrático Alfa Podado* sobre nuestra variable respuesta original.

Probamos modelos sobre polinomios de hasta tercer grado respecto a las variables.

Una primera observación es que esta poda homogeneiza drásticamente el rendimiento de los modelos y por lo tanto fue una batalla cabeza a cabeza, al menos entre los modelos que nosotros conocemos.

Modelo Seleccionado

El modelo seleccionado fue "Bosques Aleatorios" en su adaptación para regresiones, implementado por la librería `randomForest`, la más clásica de las disponibles, con código de los desarrolladores originales del algoritmo. Si bien la estimación del error medio cuadrático alfa podado de simulación es aleatoria, esta medida osciló alrededor de 10.75 y fue el resultado más pequeño que pudimos lograr por un margen que podría llamarse relativamente grande.

$$\log Y \approx F\left(\sum_{i=1}^{6} \alpha_i \cdot x_i + \sum_{i=1}^{6} \beta_i \cdot x_i^2 + \sum_{j=1}^{6} \sum_{i=1;i<j}^{6} \gamma_{ij} \cdot x_i x_j\right)$$

Debido a que la función obtenida en este caso es el conjunto de árboles que en cada caso deben votar para determinar cómo asignamos el valor predicho, puede decirse que la inteligibilidad y comunicabilidad de este modelo son bajas, así como su parsimonia (es poco sensible a variables expurias pero no las elimina).

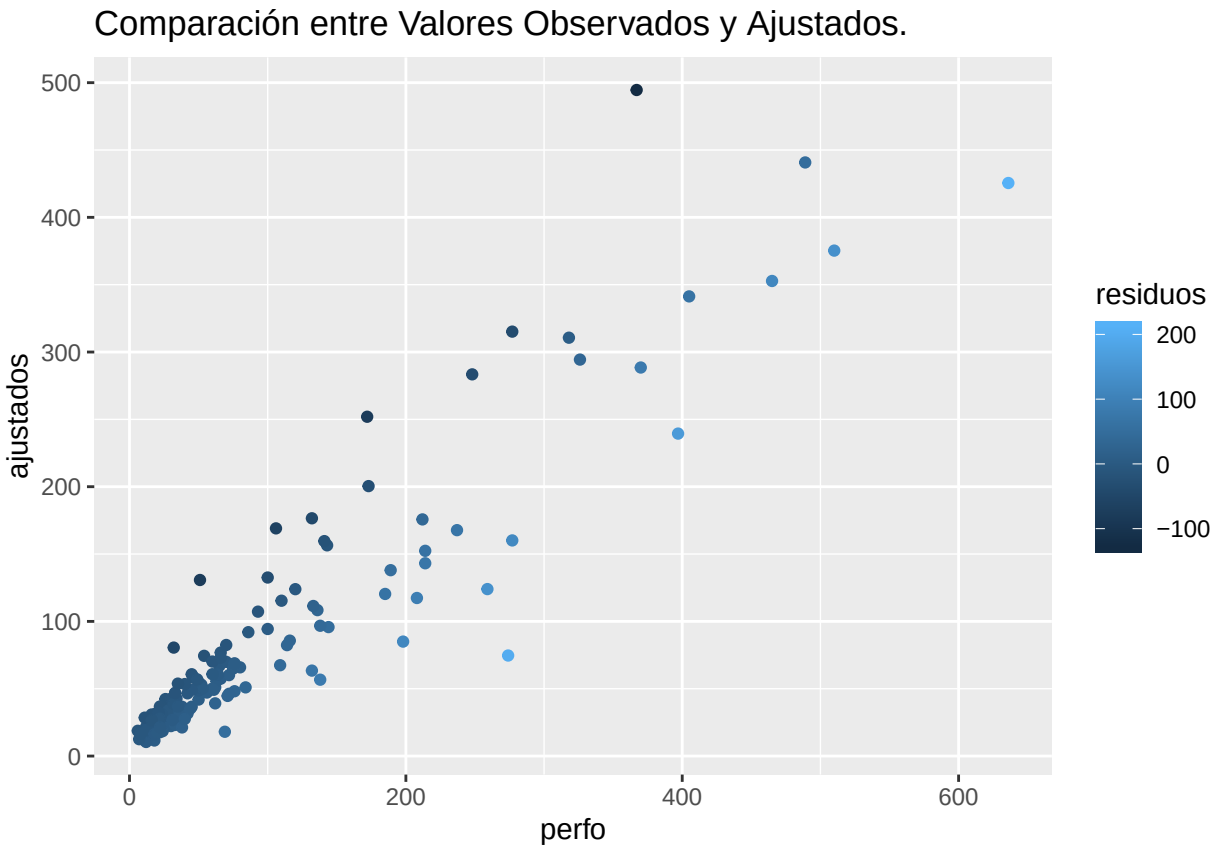
Podemos observar la medida de *importancia* de cada variable. Esta cuantifica el incremento en el error al excluir la regresora en cuestión.

	IncNodePurity
freq	13.61
mincan	16.50
maxcan	11.05
minmem	23.59
maxmem	39.82
cache	40.08

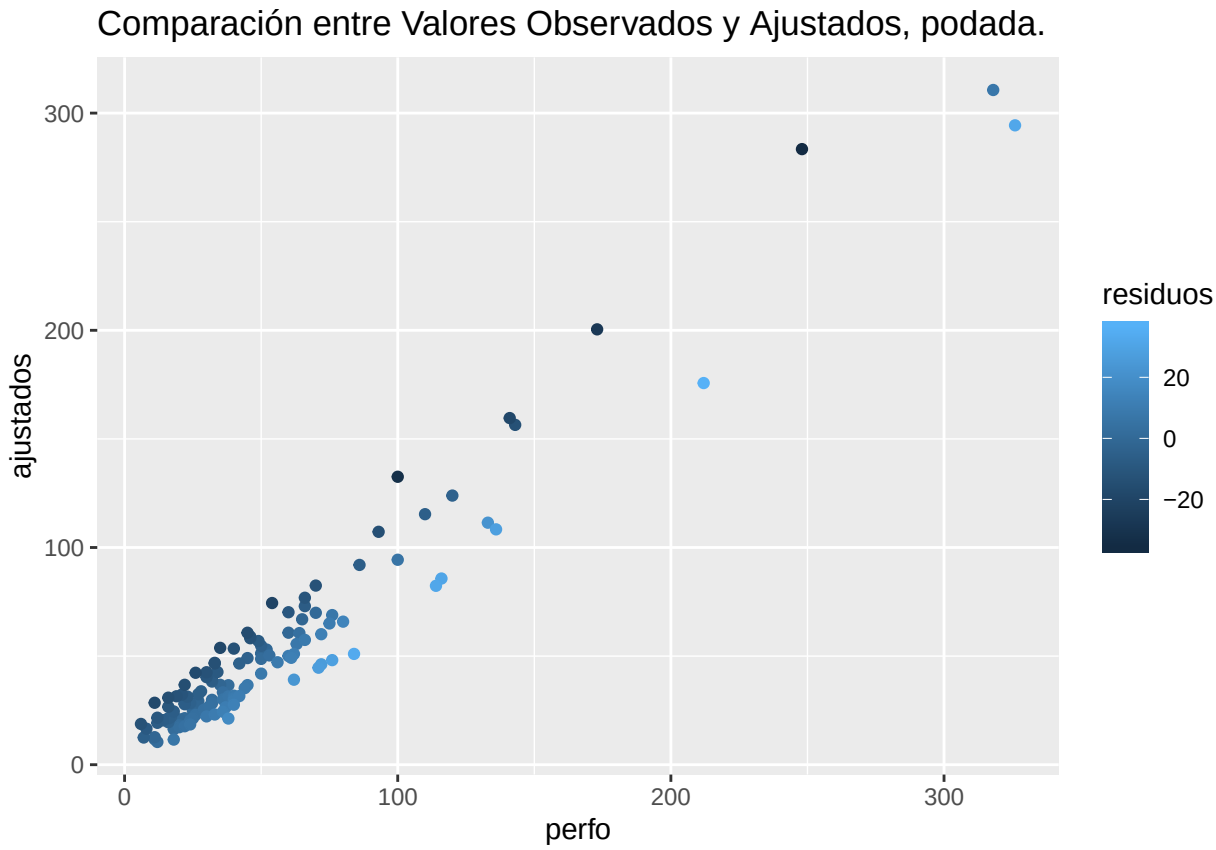
Como se puede ver, si bien hay variables de mayor importancia ninguna es irrelevante.

Diagrama QQ

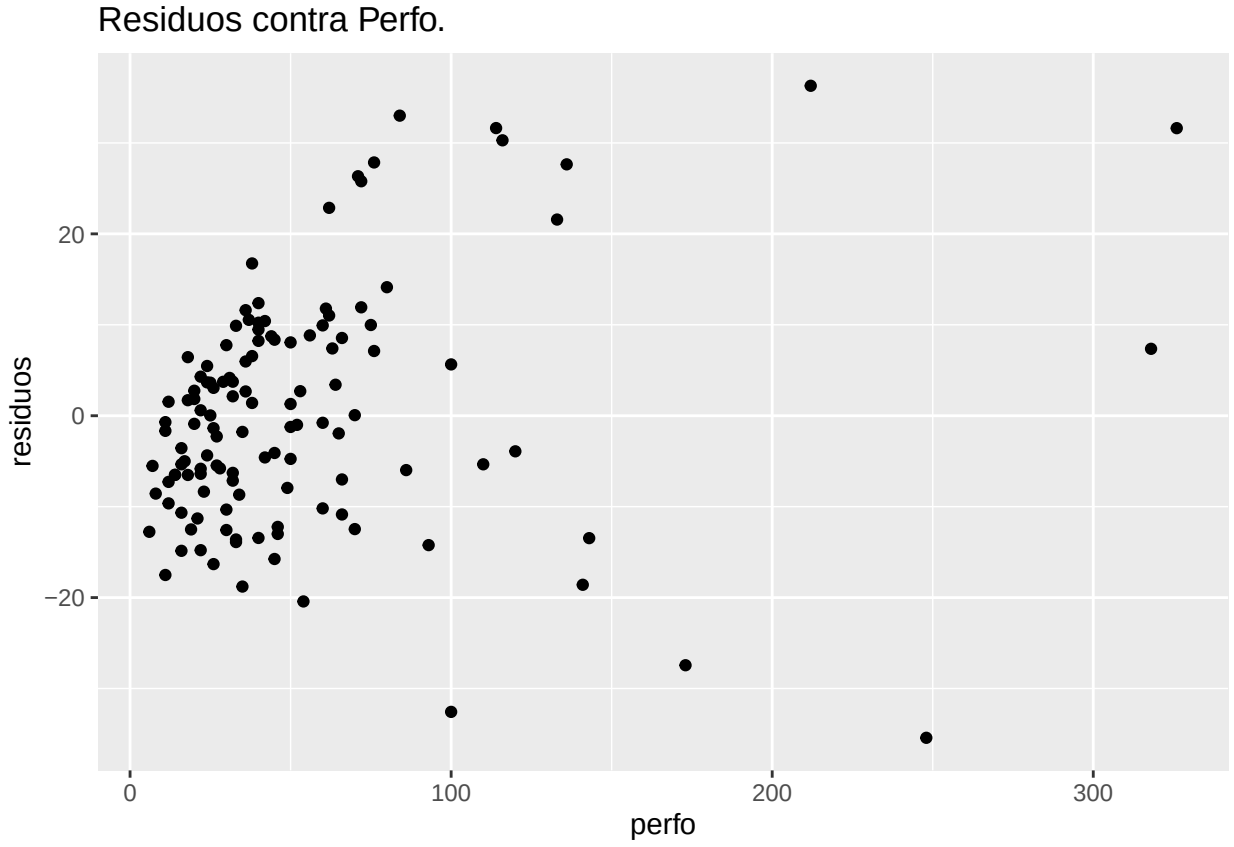
Comparamos las performances relativas ajustadas y las observadas.



En línea con la métrica propuesta para esta competencia, repetimos el mismo gráfico con los datos de mayor residuo eliminados.


Comparación entre Valores Observados y Ajustados, podada.
ajustados
perfo
residuos
300
200
100
0
0
100
200
300
20
0
−20

Residuos contra Perfo



Uso de la Función Enviada, Predecir.R

Al ejecutar la función sin parámetros, por ejemplo desde la terminal `Rscript predecir.R`, esta espera encontrar en el directorio donde se halle un archivo llamado `tests.txt` con el mismo formato del archivo que `Concurso_Estima.txt` que recibimos. En caso de hallarlo levanta la sesión llamada `modelo.RData` que contiene el modelo seleccionado entrenado y genera las predicciones. Estas quedan almacenadas en un nuevo archivo llamado `preds.txt`. También acepta nombres para el archivo de entrada y el de salida, `Rscript predecir.R <test_data_file> <preds_file>`.

Alternativamente, se puede directamente cargar la sesión `modelo.RData` y predecir usando la clásica interface de R ,en este caso con `predict(modelo,DatosNuevos)`.

Nuestro desarrollo está resumido en el guión `entrenar.R`.