



Aprendiendo 'XGBoost'

¡Hola!



Julia Lastra

Matemáticas – Data Scientist en **Lingokids**



@las_tra



linkedin.com/julialastr

Meet up - Aprendiendo “XGBoost”



Contexto



Introducción a XGBoost

- Ensamblado de modelos
- Gradient Boosting
- XGBoost

Caso práctico con R

- Datos: churn banco
- Ajustar un modelo con *xgboost*
- Interpretar resultados

<https://kahoot.it>

1

Gradient Boosting Machines

Ensamblado de modelos



- Combinar múltiples modelos en un único modelo.
- Cada modelo es un **predictor débil**, pero juntos forman un predictor **robusto**.



Predictor débil



Un predictor débil es...

- A). Un modelo con menos variables
- B). Un modelo con menos parámetros
- C). Un modelo con una estructura más sencilla
- D). Todas son ciertas



Gradient Boosting



- Técnica de ensamblado de modelos de forma **iterativa** y **aditiva**.
- Particularidad: opera en el espacio de funciones → en cada iteración, se busca una **función** (spoiler: el predictor débil), no un parámetro.

Boosting



- Otra técnica de ensamblado de modelos que también construye un modelo robusto de forma iterativa y aditiva.
- En cada iteración, se ponderan las observaciones de entrenamiento: las que producen una pérdida mayor (mal clasificadas/error grande en la predicción) cobran mayor importancia.
- **AdaBoost**



Gradient Boosting vs Boosting



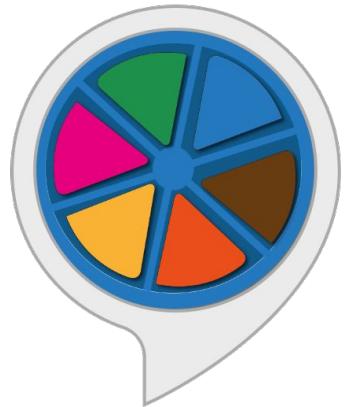
- Ambas son técnicas de ensamblado de modelos.
- Ambas construyen un modelo robusto de forma iterativa y aditiva.
- Buscan la función óptima en cada iteración de forma diferente:
 - *Boosting*: actualiza los pesos de las observaciones en función del predictor anterior.
 - *Gradient Boosting*: ajusta un modelo a los residuos del predictor anterior.

Predictor débil



Los predictores débiles pueden solo ser árboles de decisión.

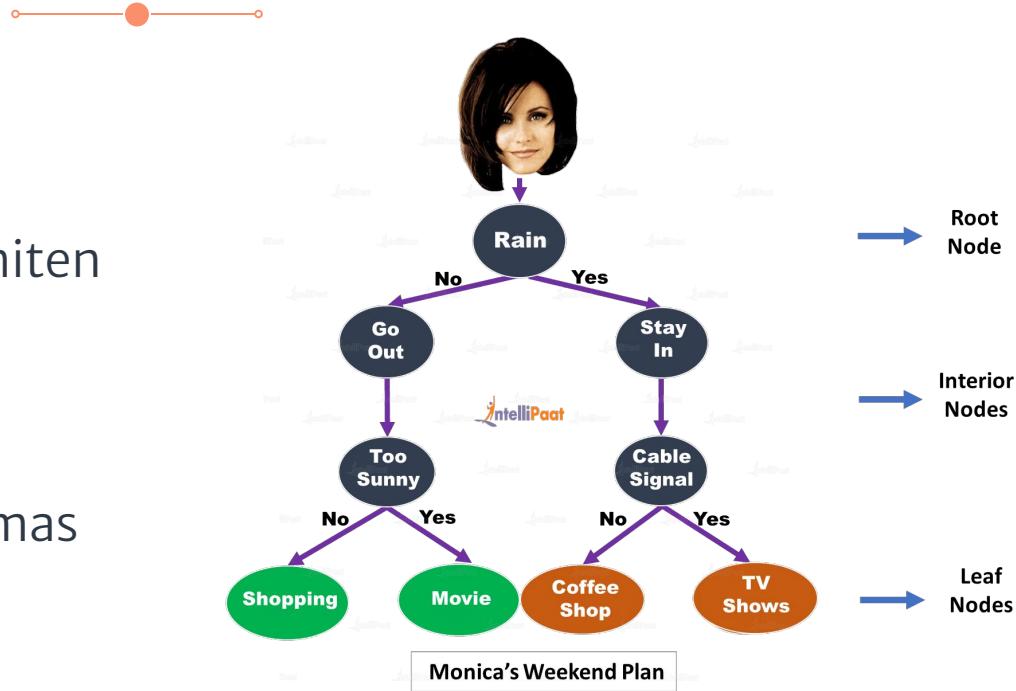
¿Verdadero o Falso?





Árboles de decisión

- Múltiples divisiones secuenciales.
- Los nodos terminales emiten predicciones.
- Son interpretables.
- Un modelo basado en un único árbol tiene problemas para generalizar.



Ensamblar árboles de decisión



Bagging

- Los árboles se entran en paralelo.
- Todos los árboles tienen el mismo peso en la decisión final.
- Todas las observaciones tienen el mismo peso.
- Ejemplo: Random Forest

Boosting



Gradient Boosting



¿Siempre ha sido tan popular Gradient Boosting?

- A). ¡Claro! Lleva petándolo en la literatura científica más de 20 años.
- B). Meh, sobre todo los últimos 5 años, desde que se popularizaron algunos frameworks open source.
- C). No es nada popular, su uso es residual.





2

XGBoost

API en R, Python, Julia, Java



XGBoost



¿De dónde procede el nombre de este paquete?

- A). External Gradient Boosting
- B). Explode Gradient Boosting
- C). Extreme Gradient Boosting



XGBoost



- Gradient Boosting regularizado
 - Introduce regularización en la impureza (medida para encontrar las divisiones óptimas de un árbol).
- Altamente flexible: función objetivo, métricas de evaluación.
- Paralelización: la construcción es secuencial, pero se puede paralelizar el entrenamiento a nivel de variable.

XGBoost



Ventajas frente a las implementaciones previas

- Incorpora el objetivo global en cada iteración → mejores resultados.
- Menor coste computacional.
- Regularización → generaliza mejor.

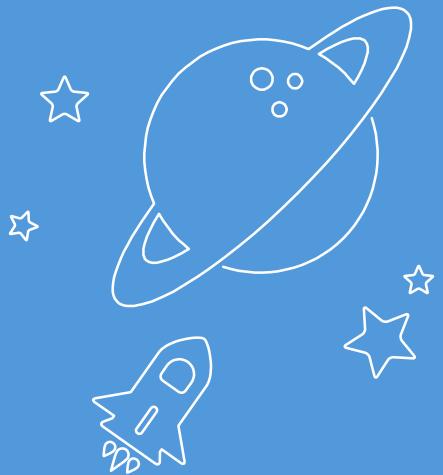
Otro framework



LightGBM

Su estrategia para crecer los árboles es diferente: mientras XGBoost se basa en niveles, *LightGBM* se basa en hojas.

- Regularización para reducir sobreajuste.
- Mejor rendimiento en computación.



xgboost

El maravilloso submundo de los hiperparámetros



Parámetros - Tipos



Generales

Determinan el funcionamiento general del método.

Predictor débil

Dependen del *booster* elegido (árbol de decisión, modelo lineal, ...).

Aprendizaje

Guían el escenario de aprendizaje.

Parámetros Generales



booster

[default = gbtree]

- Modelos basados en árboles: gbtree, dart.
- Modelos lineales: gblinear.

verbosity

[default = 1]

- Silent: 0
- Warning: 1
- Info: 2
- Debug: 3

nthread

[default = max threads available]

Número de hilos en paralelo a utilizar.

Más parámetros [aquí...](#)

Parámetros del booster (1/3)



eta

[default = 0.3]

Ayuda a evitar el sobreajuste contrayendo los pesos en cada paso.

- Valores habituales: 0.01 – 0.3

max_depth

[default = 6]

Profundidad máxima de un árbol.

- Valores habituales: 3-12

min_child_weight

[default = 1]

Suma mínima de los pesos de todas las observaciones requeridas en un nodo hijo.

Más parámetros [aquí...](#)

Parámetros del booster (2/3)



`subsample`

[default = 1]

Fracción del conjunto de entrenamiento a usar en cada iteración (filas)

- Valores habituales:
0.5 – 1

`colsample_bytree`

[default = 1]

Fracción del número de variables a usar en cada iteración (columnas)

- Valores habituales:
0.5–1

Más parámetros [aquí...](#)

`scale_pos_weight`

[default = 1]

Controla el balance entre clases (ideal en conjuntos desbalanceados).

Parámetros del booster (3/3)



gamma

[default = 0]

Pérdida mínima
requerida para hacer una
división.

alpha

[default = 0]

Regularización L1 en los
pesos.

lambda

[default = 1]

Regularización L2 en los
pesos.

Más parámetros [aquí...](#)

Parámetros del booster



- A menor *eta*, más sobreajuste.
- A mayor profundidad del árbol, más sobreajuste.
- Muestrear (filas y columnas) → modelo más robusto.
- Valores altos de *gamma* → modelo más robusto.
- Los parámetros de regularización (*alpha*, *lambda*) ayudan a obtener un modelo más robusto.

Parámetros del aprendizaje



objective

[default =
reg:squarederror]

Se elige en función del
tipo de problema:
regresión, clasificación,
supervivencia, etc.

eval_metric

[default = depende del
objective]

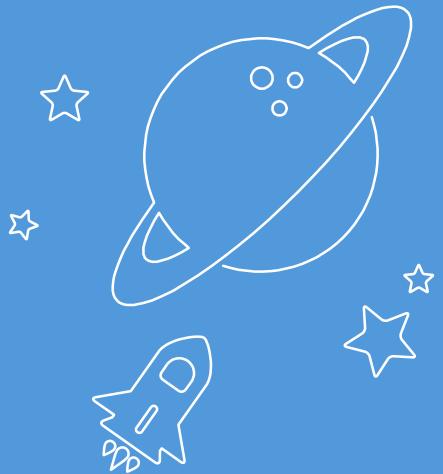
Métrica de evaluación
para la validación de datos
(ej. Rmse, mae, error, auc,
etc).

seed

[default = 0]

Semilla para obtener
resultados
reproducibles.

Más parámetros [aquí...](#)



xgboost

Evaluación del modelo – clasificación binaria

Matriz de confusión

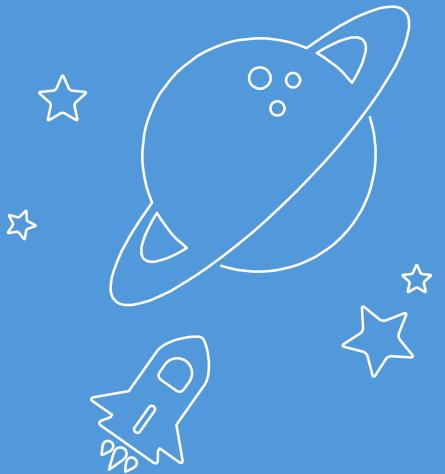


		REAL		
		0	1	
PREDICCIÓN	0	True negative (TN)	False negative (FN)	NPV $FN/(TN + FN)$
	1	False positive (FP)	True Positive (TP)	Precision $TP/(FP + TP)$
		Specificity $TN/(TN + FP)$	Recall $TP/(TP + FN)$	Accuracy $(TP + TN)/(TP + TN + FP + FN)$

Caso Churn - algunas métricas



- **Accuracy:** % de aciertos (sin distinguir entre chuners y no churners) → cuidado en conjuntos desbalanceados.
- **Precision:** % churners previstos que han sido finalmente churners.
- **Recall:** % churners que se han clasificado bien. 🔥



xgboost

Feature importance

Feature “importance”



gain

Mide la ganancia en la precisión del modelo al incorporar la variable para hacer una nueva división.

cover

Mide la cantidad relativa de observaciones en relación a una variable.

frequency

Frecuencia (%) con la que una variable aparece en el modelo para hacer una división .
(No recomendada)



2

Caso práctico

¡Vámonos!





Debugging Bingo

Extra comma	Misspelled variable	Confused factor variable with a numeric one	Extra quotation marks	Put code into a markdown cell
Missing a tilda	Extra parentheses	Used wrong case (upper vs. lower case)	Misspelled function name	Didn't import the data
Confused when using \$	Wrong argument(s) to a function	 FREE SPACE	Confused a data frame for a variable	Missing pipe operator (%>%)
Didn't load libraries	Missed a comma	Forgot to ask R to print the object	Didn't close parentheses	Misspelled data frame
Confused = with ==	Used color when you meant fill	Forgot to save	Put regular text into a code cell	Missing quotation marks

@cogscimom



**“La opinión de una multitud es
siempre más creíble que la de
una minoría.”**

Miguel de Unamuno

¡¡Gracias!!