# Predicción de popularidad de canciones

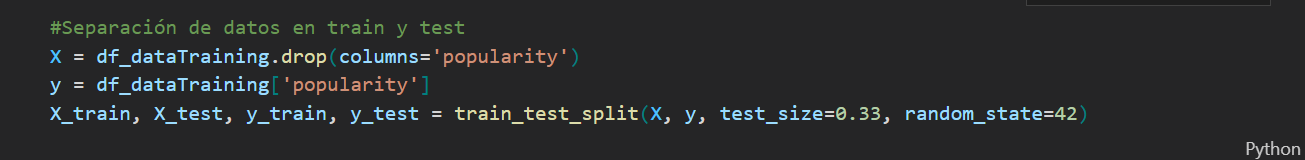
Daniel Montero, Juan Camilo Mejía, Simón Jaramillo

## Preprocesamiento de datos

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.



Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

## Calibración del modelo

Texto

El contenido generado por IA puede ser incorrecto.

Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

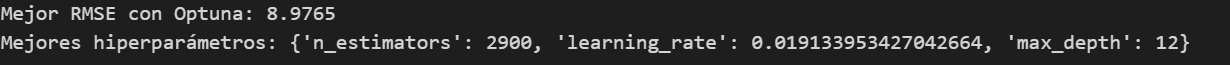
Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.

Optuna no evalúa exhaustivamente todas las combinaciones, sino que dirige las pruebas hacia las regiones con mejor desempeño. Es ideal cuando el numero de combinaciones posibles es muy grande.

Texto

El contenido generado por IA puede ser incorrecto.



Al optimizar los hiperparámetros por separado, se obtienen óptimos locales para cada uno de ellos. Sin embargo, algunos hiperparámetros tendrán mejor desempeño en conjunto con otros. Por ejemplo, *lernging rate* bajo necesitará más estimadores para converger.

**Resumen de hiperparámetros del modelo seleccionado:**

|  |  |  |
| --- | --- | --- |
| **Parámetro** | **Valor óptimo** | **Cómo afecta al modelo** |
| **n\_estimators** | 2900 | Un gran número de árboles permite que el modelo agregue poco a poco correcciones, reduciendo el **sesgo**. Con un learning\_rate pequeño (≈0.019), hace falta más estimadores para converger, pero se logra un ajuste más fino y estable. |
| **learning\_rate** | 0.0191 | El aprendizaje es lento, cada árbol aporta una pequeña corrección, lo que evita saltos gandres en el aprendizaje. Esta tasa de aprendizaje requiere más n\_estimators para que converja el modelo. |
| **max\_depth** | 12 | Profundidad alta que permite capturar variaciones pequeñas y complejas entre variables, pero sin caer en profundidades extremas que generen sobreajuste descontrolado. |

Texto

El contenido generado por IA puede ser incorrecto.

## Entrenamiento del modelo

Una captura de pantalla de un celular

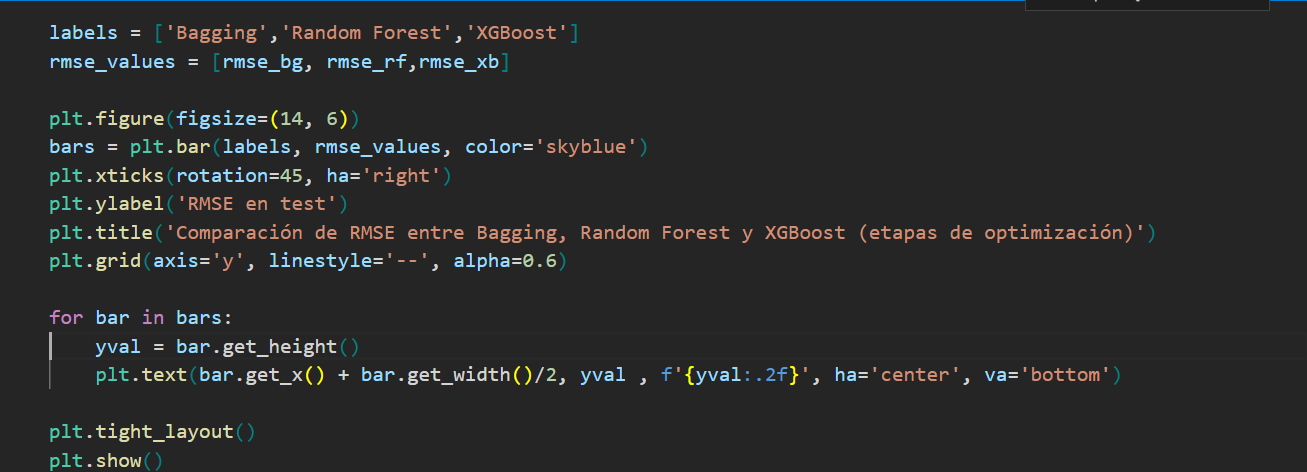
El contenido generado por IA puede ser incorrecto.

Una captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.



Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.

Además de que **XGBoost** muestra el mejor RMSE evaluándolo en los datos X\_test, comparado con el modelo de random forest y bagging, en general presentó una velocidad de entrenamiento y predicción mucho mayor, lo que permitió que su calibración utilizara menos recursos computacionales y tiempo. Este factor resultó importante de cara a poder ser utilizado en la competencia, ya que el tiempo de entrenamiento con los datos completos y los de test para la competencia siempre fue mucho menor y ayudó en su calibración.

Este modelo, de igual forma, presenta una mayor robustes y mayor capacidad para interpretar y predecir las relaciones no lineales presentes en los datos, lo que se traduce en una mayor capacidad de escalabilidad que los otros dos modelos.

## Disponibilización del modelo

Se disponibiliza el modelo usando AWS.

Texto

El contenido generado por IA puede ser incorrecto.

Predicciones de 2 registros del *data set* test:



Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.



Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

## Conclusiones

* Variables numéricas presentan distribuciones muy sesgadas y colas largas (ej: duration\_ms, popularity).
* La matriz de correlaciones mostró relativamente baja multicolinealidad, lo que permite usar varias variables sin excesiva redundancia.
* track\_genre y artists tienen cardinalidad moderada; track\_name y album\_name son casi únicas.
* Para estas últimas, un one-hot sería inviable (dimensionalidad excesiva), por lo que se opta por codificación basada en target.
* Separación 67 %/33 % **antes** de cualquier transformación para evitar fugas de información.
* Uso de CatBoostEncoder en variables categóricas. La ventaja de este categorizador es que se obtiene una sola columna numérica por categoría y con regularización hacia la media global.
* **Log(1+x)** en duration\_ms, instrumentalness, num\_artists: comprime colas y estabiliza la varianza.
* **Exp(x)** en speechiness, liveness: amplifica diferencias sutiles en rangos [0–1].
* **Raíz cuadrada** en tempo y en las codificaciones del CatBooster, para moderar valores extremos.
* StandardScaler sobre todas las variables: media ≈ 0, desviación ≈ 1, mejor convergencia y comparabilidad de coeficientes.
* **BaggingRegressor**: Óptimo local de RMSE en ~660 árboles, max\_samples=1.0, max\_features=0.6.
* **RandomForestRegressor**: Mejor desempeño a n\_estimators=500, max\_depth=25, max\_features=1.0.
* **XGBoostRegressor**: mínimo RMSE cerca de learning\_rate≈0.15–0.2 y max\_depth≈6–7.
* **Optimización con Optuna** RMSE = 8.9765 n\_estimators = 2900 learning\_rate =0.01913 max\_depth = 12. La mayor ventaja de Optuna es que captura interacciones entre parámetros y dirige la búsqueda hacia combinaciones globalmente óptimas sin hacer búsqueda exhaustiva.
* XGBoost gana con la menor métrica de error, compensando el mayor coste computacional con mejor precisión.