Trabajo Practico Integrador - Actividad nº 3

Algoritmos de Machine Learning: Regresión, Clasificación y Clustering.

Valentino Francisco, Juan Strauss y Matías Costamagna.

Contexto de los Datos

Objetivo Principal

Entender y predecir la popularidad (Streams) de una canción utilizando técnicas avanzadas de machine learning.

Dataset

Más de 20.000 canciones de Spotify y Youtube con características técnico-musicales complementarias de ambas plataformas.

Atributos

Popularidad

Views, Likes, Comments, Stream

Musicales

Danceability, Energy, Loudness, Valence, Speechiness

Metadatos

Key, Licensed, Duración, Artista

Preparación de Datos

Realizamos una preparación robusta aplicada a todos los modelos posteriores, estableciendo una base sólida para el análisis.

01

Limpieza de Datos

Eliminamos filas nulas de Stream e imputamos valores nulos, Mediana para Views y Likes.

02

Feature Engineering

Creamos 3 atributos nuevos: Youtube_Popularity, Like_per_View_Ratio y Acoustic_Danceability.

03

Transformación logarítmica

Transformamos logaritmicamente los atributos que sesgan a los demas datos.

04

Definición del Conjunto de Entrenamiento y Prueba

Registros para Entrenamiento: 16111

Registros para Prueba: 4028

05

Pre-procesamiento

StandardScaler para features numéricas.

OneHotEncoder para categóricas (Key, Licensed).

Modelos de Regresión — ¿Cuánta Popularidad Tendrá?

Predecir el valor numérico exacto de Streams mediante transformación logarítmica para manejar outliers extremos.

Desafío Principal

Los outliers (canciones con miles de millones de streams) sesgaban los datos, invalidando modelos lineales.

Solución Implementada

Transformación logarítmica (np.log1p) aplicada a Stream y todas las features de popularidad, normalizando la distribución.

Métrica de Éxito

RMSE (Raíz del Error Cuadrático Medio). Modelos probados con K-Fold: Regresión Lineal, Árbol de Decisión, SVR, Random Forest y K-NN.

Conclusión A: El Modelo Ganador

RandomForestRegressor

RMSE = 1.11 (K-Fold). Un 16% más preciso que Regresión Lineal (RMSE 1.29), justificando su mayor complejidad computacional.

Resultado Final: $R^2 = 0.53$. El modelo explica el 53% de la varianza en popularidad de canciones.

Hallazgo Clave

`log_Likes: 35% de importancia`. La mejor predicción del éxito en Spotify es conocer el éxito en YouTube.

Hiperparámetros (RandomizerSearch)

- `n_estimators` **(Árboles):** Se aumentó de 100 a **300**. Esto le dio al modelo un "bosque" más grande para promediar las predicciones.
- `max_depth` **(Profundidad):** Se limitó a **30** (en vez de `None`, que es ilimitado).
- `min_samples_leaf` / `split` **(Muestras Mínimas):** Se aumentaron a **4** y **5** (en lugar de 1 y 2).

- 🥇 Random Forest: R^2 Promedio 0.5359 (RMSE: 1.1137)
- 🥈 KNN (Regresión): R^2 Promedio 0.3867 (RMSE: 1.2810)
- 🥉 Regresión Lineal: R^2 Promedio 0.3783 (RMSE: 1.2898)
- 🏅 Linear SVR: R^2 Promedio 0.3707 (RMSE: 1.2976)
- 🏅 Árbol de Decisión: R^2 Promedio 0.0850 (RMSE: 1.5634)

Conclusión B: El Crítico Musical

Prueba: ¿Qué ocurre si le prohibimos al modelo acceder a variables de popularidad (log_Likes, log_Views, entre otras)?

Objetivo

Forzar al modelo a predecir usando solo características musicales.

Resultado

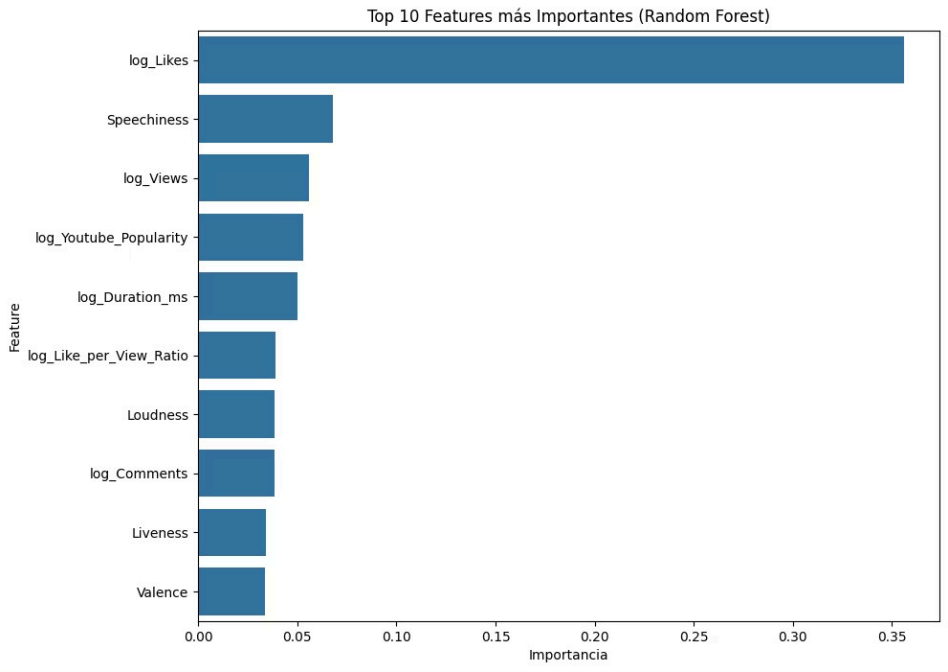
RMSE = 1.47 (peor). Pero revelador sobre qué características musicales importan.

Top Features Musicales

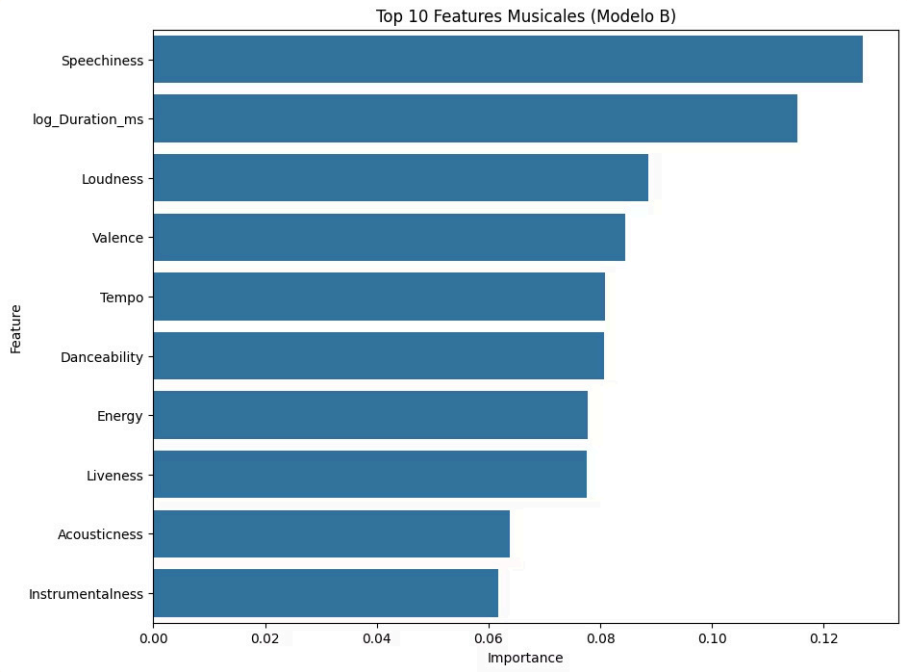
Speechiness (12.7%), log_Duration (11.5%), Loudness (8.8%). Danceability, Energy y Valence entre 7% y 8%.

Speechiness diferencia "música" de "podcasts". El modelo aprendió a penalizar contenido hablado.

Comparacion de caracteristicas importantes para el modelo (conclusión A vs B)



Modelo conociendo las "variables de popularidad".



Modelo solo conociendo las variables "tecnico-musicales"

Conclusión alternativa - Regresión por tramos:

Objetivo: Lo que esperamos ver es que los valores de **R^2** para cada "tramo" sean **más altos** que el 0.5338 de nuestro modelo generalista (RandomForest).

Paso 1: Segmentación (Crear los "mini-datsets")

Usamos la función `pd.cut` de Pandas para etiquetar cada canción en una de estos tramos:

- `1 - Alcance_Bajo` (0 - 100k)
- `2 - Popular` (100k - 1M)
- `3 - Exito_Relativo` (1M - 10M)
- `4 - Exito_Importante` (10M - 100M)
- `5 - Exito_Global` (100M - 1B)
- `6 - Fenomeno_Mundial` (> 1B)

Paso 2: El Aislamiento

Una vez que cada canción tuvo su etiqueta de "Tramo", creamos un bucle `for` que iteró 6 veces, una por cada etiqueta. En cada iteración, el código aisló los datos de esa liga, creando 6 "mini-datasets" temporales.

- Iteración 2 (tramo 2 Popular): Creó un "mini-dataset" que contenía solo las 254 canciones que tenían entre 100k y 1M de streams.
- Iteración 3 (tramo 3 Exito_Relativo): Creó un "mini-dataset" con las 2226 canciones de ese tramo. (y así sucesivamente)

Paso 3: El Entrenamiento "Especialista"

Entrenamos un modelo separado e independiente para cada "mini-dataset". Esto significa que no terminamos con 1 modelo, sino con 6:

Un modelo especialista en predecir "Alcance Bajo", Un modelo especialista en predecir "Popular", ...etc.

La hipótesis era que el especialista de "Popular" sería mucho más preciso en su rango (100k-1M) que el modelo "generalista".

Paso 4: El Resultado (La Sorpresa)

Medimos el **R^2** (la precisión) de cada modelo especialista *solo* en su propio tramo. Los resultados fueron claros:

- **R^2 Modelo Generalista (RandomForest): 0.5338 (53.4%)**
- **R^2 Modelo Especialista (Tramo `2_Popular`):** 0.2914 (29.1%)
- **R^2 Modelo Especialista (Tramo `4_Exito_Importante`):** 0.0799 (8.0%)

El "experimento" **fracasó espectacularmente**. Los modelos especialistas fueron *mucho peores* que el modelo generalista.

¿Por qué falló?

Porque le quitamos a los modelos lo más importante que necesitaban para aprender: la **varianza**.

- **Nuestro Modelo Generalista (R^2 0.53)** tenía "éxito" porque podía ver la diferencia masiva entre una canción de 10k *streams* y una de 1B *streams*.
- **El Modelo Especialista (R^2 0.08)** falló porque le pedimos una tarea casi imposible: Tomar solo canciones que ya son similares (todas en el tramo de 10M a 100M) y explicar la diferencia entre ellas.

Algunas Predicciones Realizadas Por el Modelo Generalista

Expectativa: El Modelo Predice de Manera Aceptable:

```
--- 🚨 ¡RESULTADO DE LA PREDICCIÓN! 🚨 ---
Modelo (Random Forest Ajustado) predice: 42,092,171 streams
La canción REALMENTE tuvo:               49,205,438 streams
--------------------------------------------------
Error en esta predicción específica: 7,113,267 streams
```

Acierto: 85.5% - Error: 14.5%

```
--- 🚨 ¡RESULTADO DE LA PREDICCIÓN! 🚨 ---
Modelo (Random Forest Ajustado) predice: 251,082,323 streams
La canción REALMENTE tuvo:               228,525,496 streams
--------------------------------------------------
Error en esta predicción específica: 22,556,827 streams
```

Acierto: 90.1% - Error: 9.9%

```
--- 🚨 ¡RESULTADO DE LA PREDICCIÓN! 🚨 ---
Modelo (Random Forest Ajustado) predice: 18,572,883 streams
La canción REALMENTE tuvo:               20,371,761 streams
--------------------------------------------------
Error en esta predicción específica: 1,798,878 streams
```

Acierto: 91.2% - Error: 8.8%

Realidad: El Modelo También Predice Muy Erroneamente:

```
--- 🚨 ¡RESULTADO DE LA PREDICCIÓN! 🚨 ---
Modelo (Random Forest Ajustado) predice: 248,591 streams
La canción REALMENTE tuvo:                10,798 streams
--------------------------------------------------
Error en esta predicción específica: 237,793 streams
```

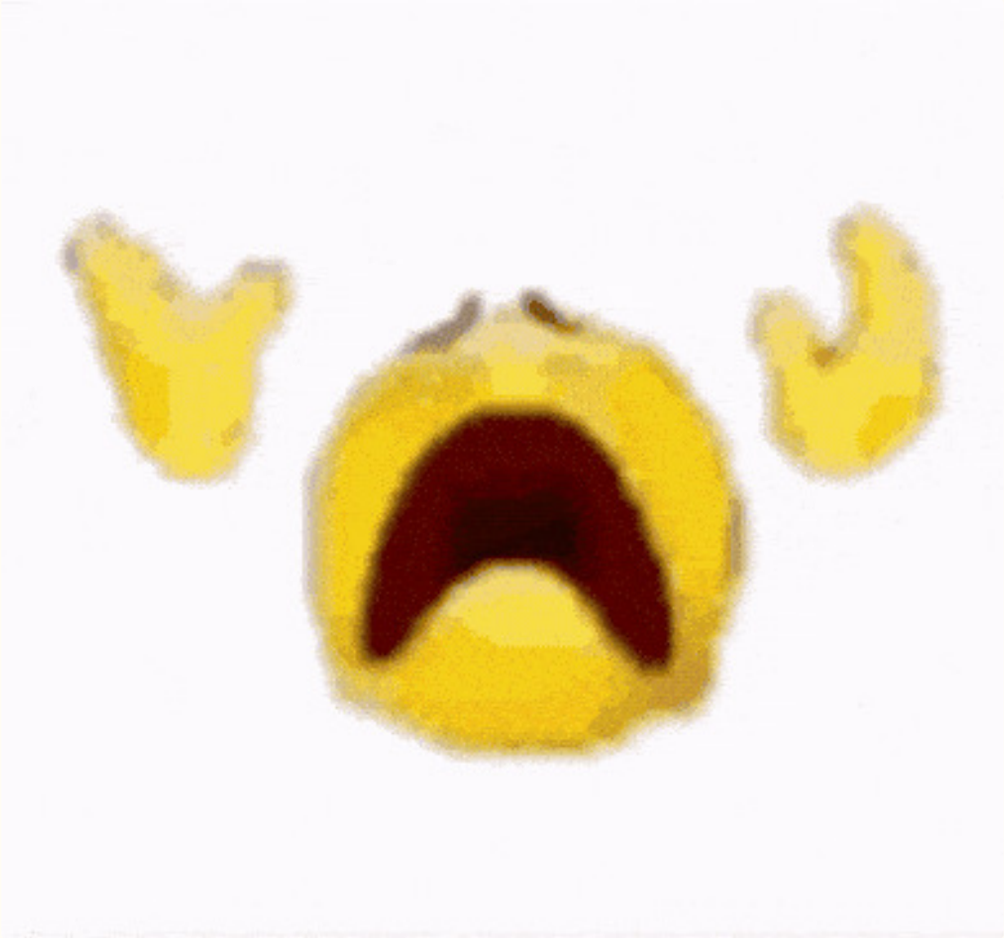
Acierto: -2102.2% - Error: 2202.2%

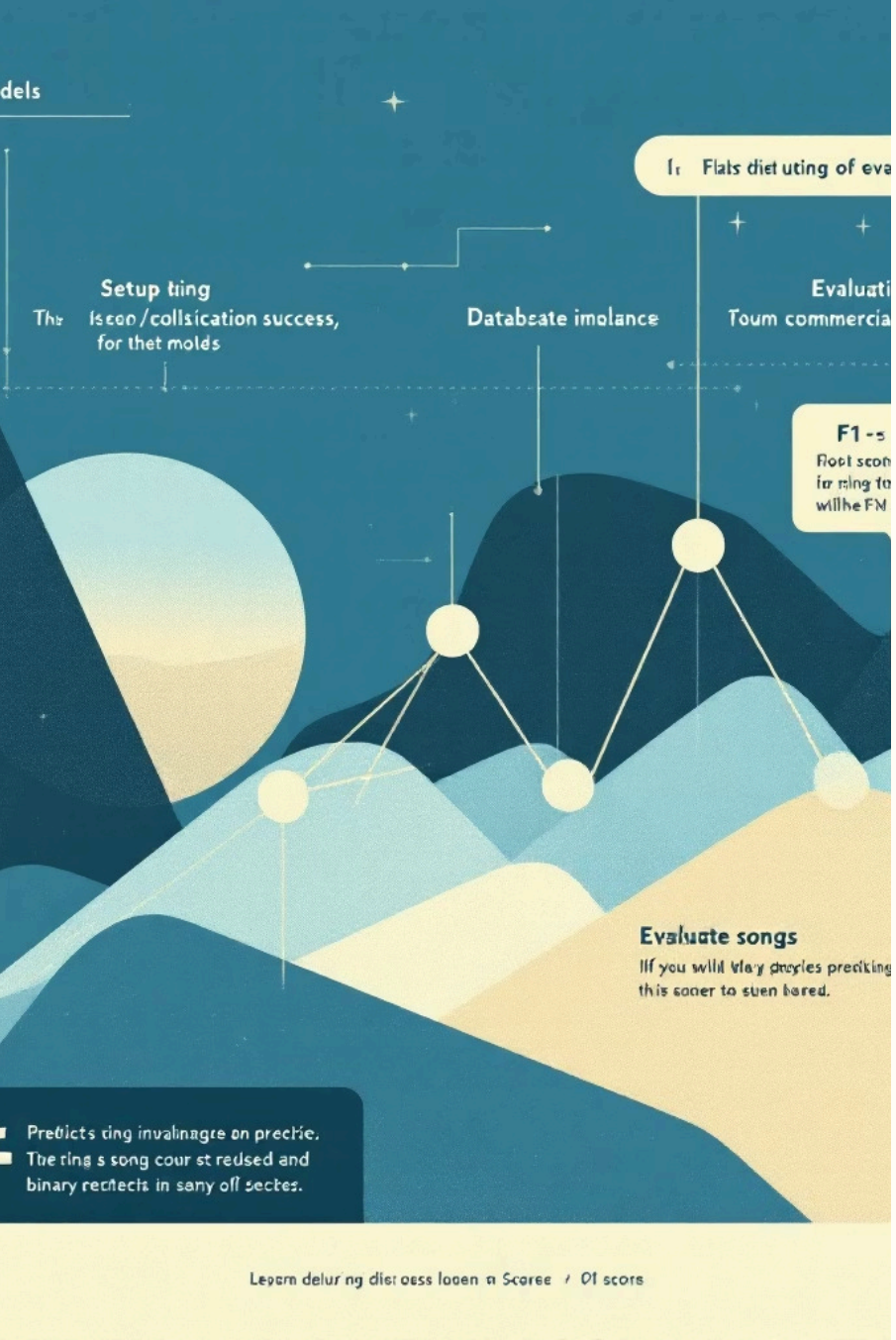
```
--- 🚨 ¡RESULTADO DE LA PREDICCIÓN! 🚨 ---
Modelo (Random Forest Ajustado) predice: 37,176,381 streams
La canción REALMENTE tuvo:               345,671,408 streams
--------------------------------------------------
Error en esta predicción específica: 308,495,027 streams
```

Acierto: 10.8% - Error: 89.2%

```
--- 🚨 ¡RESULTADO DE LA PREDICCIÓN! 🚨 ---
Modelo (Random Forest Ajustado) predice: 18,660,643 streams
La canción REALMENTE tuvo:                2,106,873 streams
--------------------------------------------------
Error en esta predicción específica: 16,553,770 streams
```

Acierto: -685.7% - Error: 785.7%





Modelos de Clasificación — ¿Será un Éxito?

Objetivo: Clasificar las canciones en éxito y no éxito.

Setup del Problema

- Target: (1=Éxito, 0=No Éxito)
- Dataset: 80% No Éxito, 20% Éxito (desbalanceado)
- Métrica: F1-Score (robusto al desbalance)

Modelos Evaluados

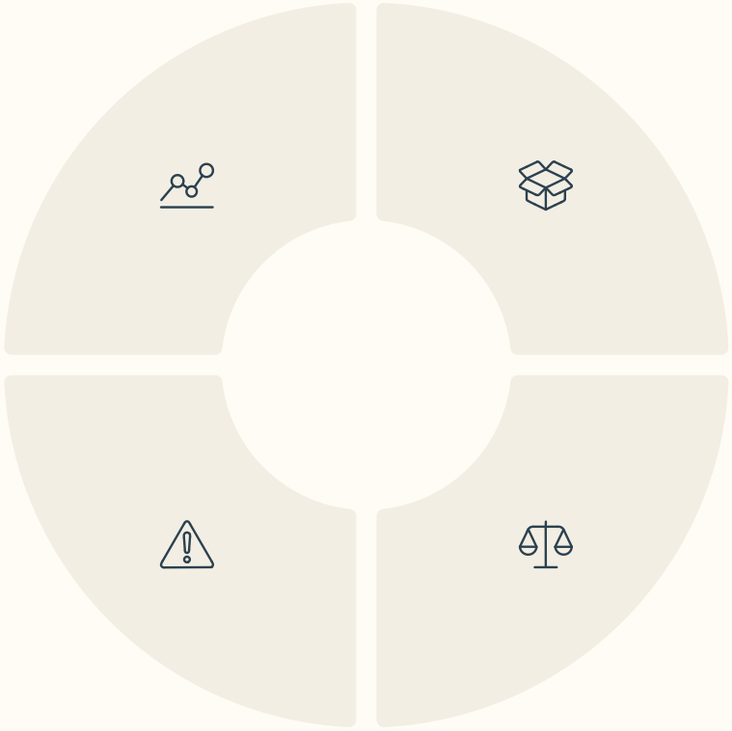
- Regresión Logística
- Random Forest (F1=0.78)
- Árbol de Decisión ajustado (F1=0.76 y Accuracy (Exactitud) = 86.37%)

Modelo Ganador: Interpretabilidad sobre Precisión

Elegimos Árbol de Decisión (F1=0.76) sobre Random Forest, priorizando interpretabilidad y estabilidad para insights accionables.

Rendimiento

F1-Score: 0.76. Una diferencia mínima de 2% es un precio justo por transparencia.



Interpretabilidad

100% "caja blanca", nos da una interpretabilidad clara.

Limitación

Conservador: Recall 95% para "No Éxitos" pero solo 51% para "Éxitos" (falsos negativos).

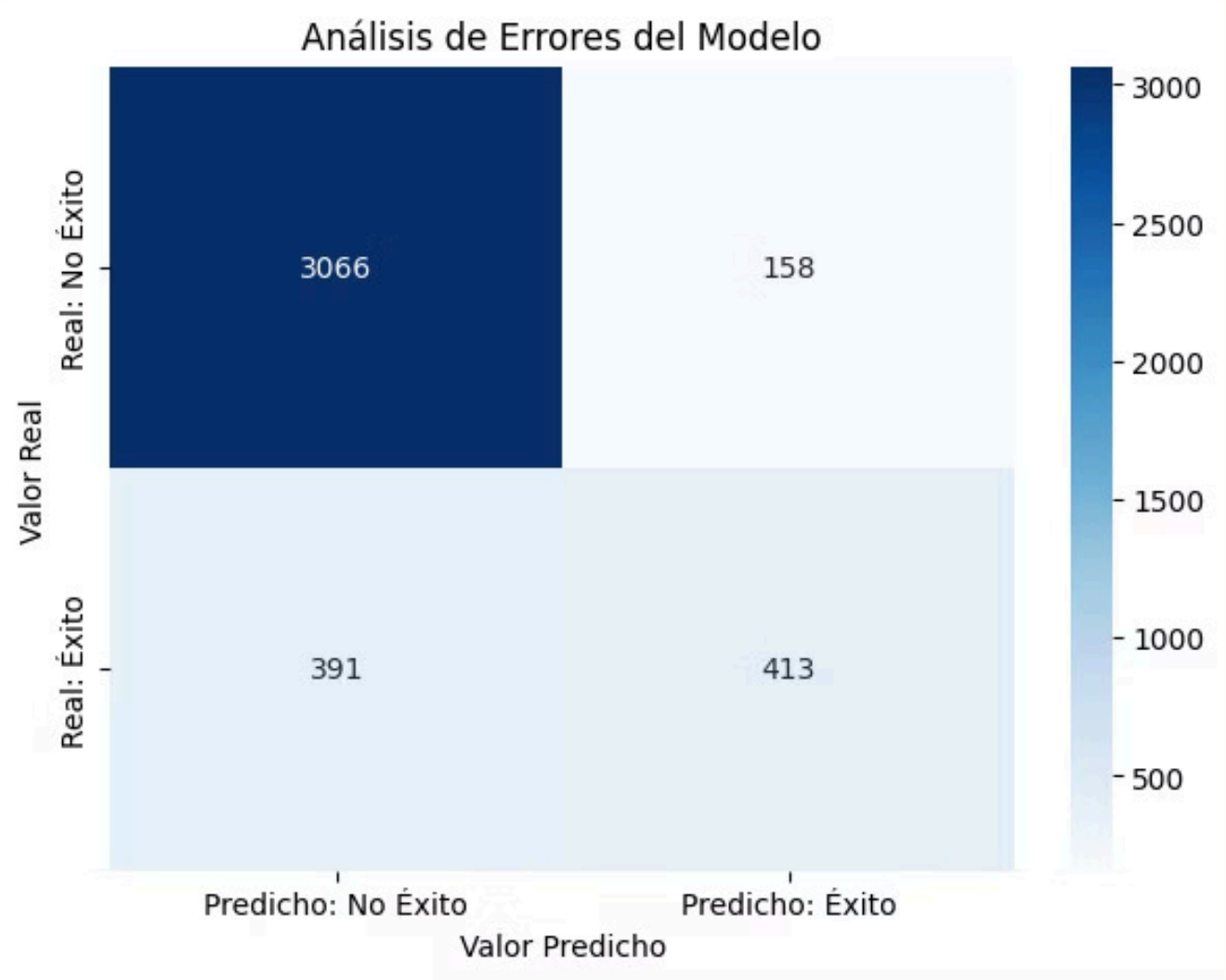
Características Clave

Modelo estable con menor desviación estándar entre folds.

Hiperparámetros (RandomizedSearch)

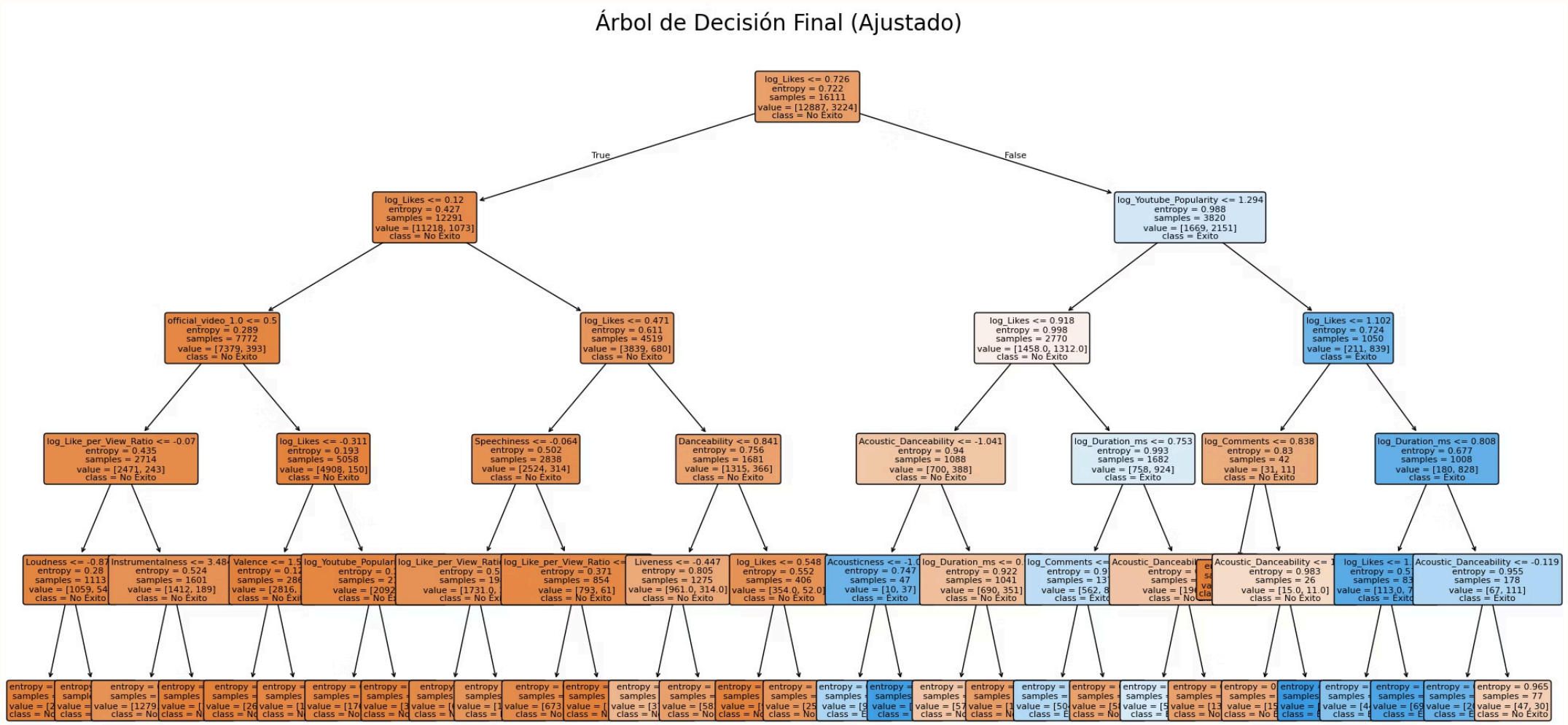
- `max_depth` **(Profundidad):** Se redujo de `None` (ilimitado) a solo **5**.
- `min_samples_leaf` **(Muestras Mínimas):** Se aumentó a **4**.

Matriz de Confusión



- El modelo es muy **conservador**. Es fantástico para identificar canciones 'No Éxito' (acertó 3,066).
- Sin embargo, su principal debilidad es el **Falso Negativo**. El modelo **se perdió 392 'Éxitos'**, que es casi la misma cantidad que los 412 'Éxitos' que sí encontró.
- Esto confirma lo que vimos en el **Reporte anterior de Clasificación**: nuestro modelo tiene un **Recall de 0.51** (51%), lo que significa que solo es capaz de 'encontrar' a la mitad de los 'Éxitos' reales."

Árbol de decisión



¿Cómo "Piensa" el Árbol de Decisión?

Funciona como un diagrama de flujo de "preguntas" (Si/No) para decidir si una canción es un **"Éxito"** o un **"No Éxito"**.

1. **La Pregunta #1 (La Raíz):** La pregunta más importante que el árbol aprendió a hacer es: **"¿Esta canción tiene muchos 'Likes'?"**
2. **El Camino Fácil (La Rama Izquierda):** Si la respuesta es **NO** (pocos *likes*), el árbol predice inmediatamente **"No Éxito"**. Esta simple regla clasifica correctamente a la gran mayoría de las canciones no populares.
3. **El Camino Difícil (La Rama Derecha):** Si la respuesta es **SÍ** (muchos *likes*), el árbol no está seguro (el grupo es 50/50). Por lo tanto, debe hacer más preguntas.
4. **Preguntas de Refinamiento:** El árbol continúa preguntando por *features* secundarias como `Liveness` (si suena en vivo) o `Speechiness` (si es hablada) para separar a los "Éxitos" de los "No Éxitos" dentro de ese grupo de *likes* altos.

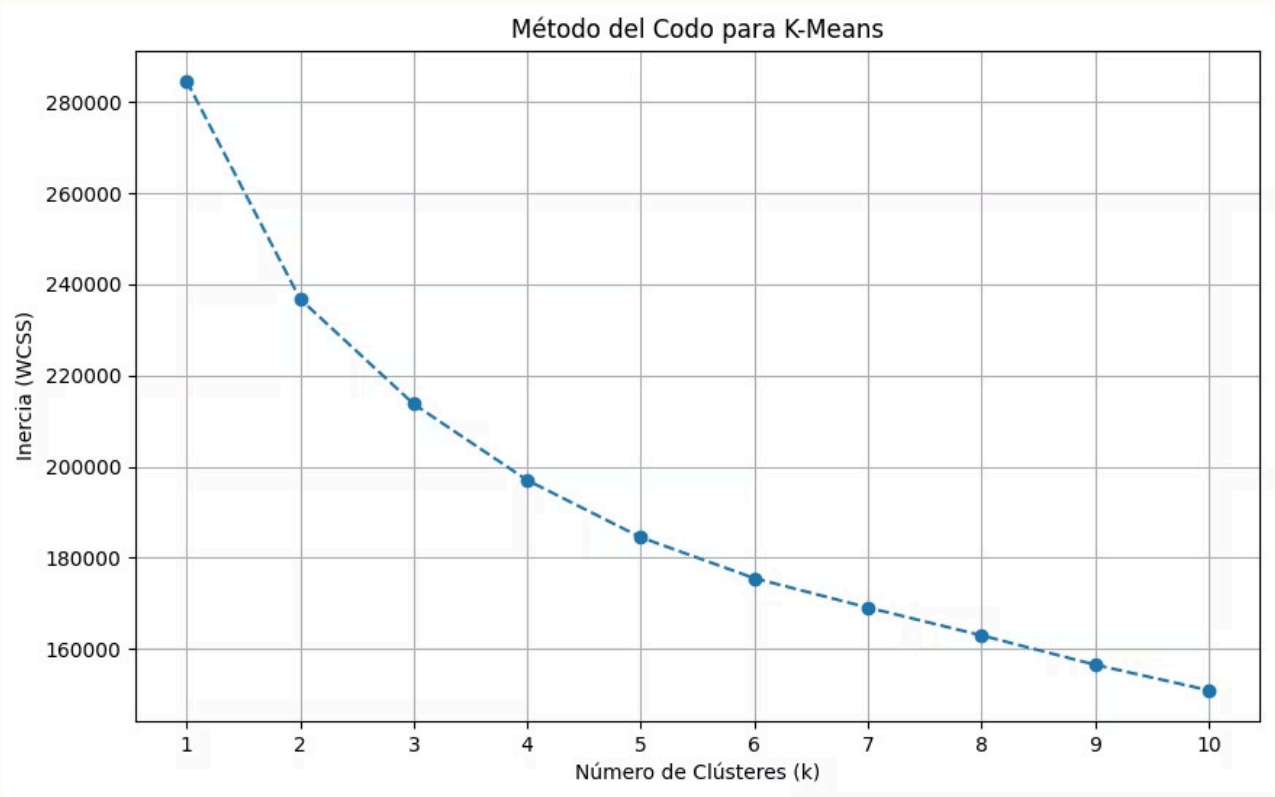
Modelos de Clustering — ¿Qué Grupos Naturales Existen?

Enfoque no supervisado para descubrir segmentaciones en los datos sin etiquetas previas.

Modelos probados:

K-Means, Clustering Jerárquico, DBSCAN y HDBSCAN con validación K-Fold.

Metodo del codo



- El número óptimo de clústeres (k) utilizados para los modelos de K-Means y Clustering Jerárquico, obtenido con el metodo del codo, fue K = 4.

- Modelo: K-Means (k=4)
  Silhouette Score: 0.1579
  Tiempo: 9.39s

- Modelo: Jerárquico (k=4)
  Silhouette Score: 0.1011
  Tiempo: 51.85s

- DBSCAN encontró 5 clústeres y 15854 puntos de ruido.
  Silhouette Score (sin ruido): 0.1864
  Tiempo: 1.63s

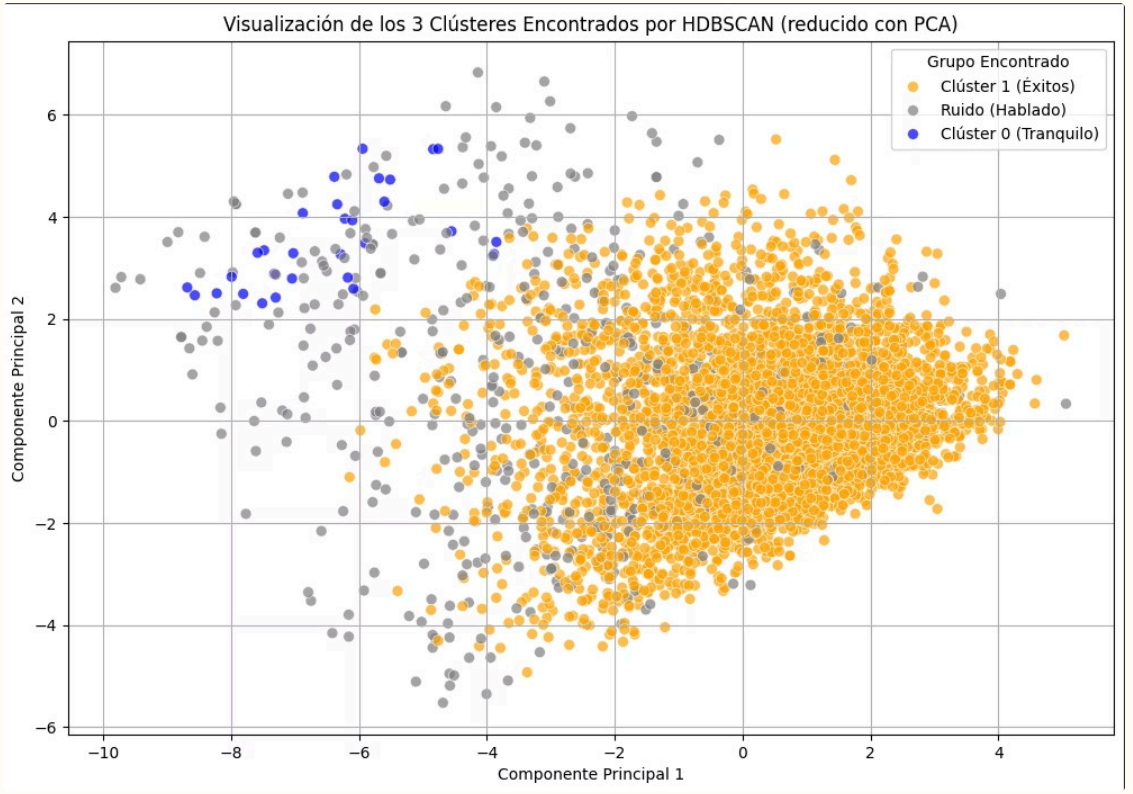
- HDBSCAN encontró 2 clústeres y 1721 puntos de ruido.
  Silhouette Score (sin ruido): 0.4697
  Tiempo: 17.13s

Ganador: HDBSCAN

Silhouette = 0.47. Masivamente superior a K-Means (0.15), demostrando que los grupos no son esféricos.

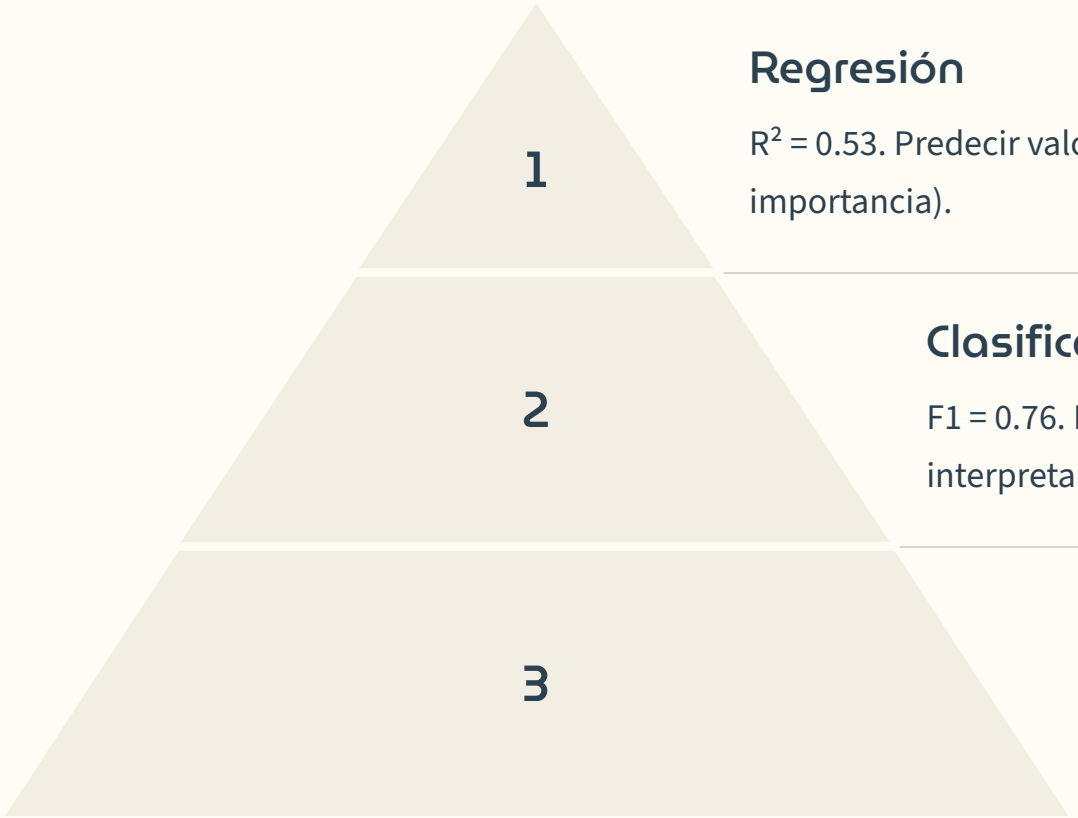
Los 3 Grupos Descubiertos

- **Clúster 1:** Éxitos Mainstream (Pop/Bailable)
- **Clúster 0:** Música Tranquila (Acústica, Clásica)
- **Clúster -1 (ruido):** Contenido Hablado (Podcasts, Comedia)



Conclusión final:

Abordamos la predicción de popularidad musical desde perspectivas complementarias, cada una respondiendo una pregunta específica.



Regresión

$R^2 = 0.53$. Predecir valores exactos de Streams. Demostró que log_Likes domina (35% importancia).

Clasificación

F1 = 0.76. Identificar "Éxitos" Top 20%. Árbol de Decisión ofrece reglas interpretables y confiables.

Clustering

Silhouette = 0.47. Descubrir patrones naturales. HDBSCAN identificó 3 segmentos musicales distintos y coherentes.