Trabajo Práctico 1 - Grupo 5

EJ1: Análisis Exploratorio:

El dataset analizado corresponde a los viajes en taxi amarillo de la ciudad de Nueva York durante el primer trimestre del año 2024 (enero, febrero y marzo). Estos dataset tienen 19 columnas cada uno, los mergeamos con el dataset de las zonas de los taxis. Quedando así un dataset que es con el que trabajaremos en este ejercicio.

Teniendo este 24 columnas y 9554778 registros, entre sus features más destacables tenemos:

Preprocesamiento de Datos:

tpep_pickup_datetime y tpep_dropoff_datetime

- **Qué representa**: Fecha y hora de inicio y fin del viaje.
- **Por qué es importante**: Permite calcular la duración del viaje y analizar patrones temporales (por día, por hora, días de semana vs. fines de semana).

trip_distance

- **Qué representa**: Distancia del viaje medida por el taxímetro (en millas).
- **Por qué es importante**: Variable clave para analizar la duración, el costo del viaje y detectar outliers o anomalías (distancias demasiado cortas o largas).

fare_amount, tip_amount, total_amount

- **Qué representan**: Tarifa base calculada por el taxímetro, propina y monto total cobrado.
- **Por qué son importantes**: Son esenciales para el análisis económico del servicio y para entender el comportamiento de pago de los usuarios (quién deja más propina, cómo varía el total con el tiempo y la distancia).

payment_type

- **Qué representa**: Método de pago (crédito, efectivo, etc.).
- **Por qué es importante**: Permite detectar tendencias en el uso de métodos de pago y analizar posibles relaciones con el monto o la propina (por ejemplo, propinas solo se registran si se paga con tarjeta).

passenger_count

- **Qué representa**: Número de pasajeros. Es un dato ingresado por el conductor.
- **Por qué es útil**: Ayuda a segmentar viajes según tamaño del grupo, y se puede usar para calcular el costo por pasajero, lo que da más perspectiva económica.

RatecodeID

- **Qué representa**: Código que indica la tarifa aplicada (tarifa estándar, aeropuerto JFK, tarifa negociada, etc.).
- **Por qué es útil**: Ayuda a diferenciar tipos de viaje, especialmente aquellos hacia/desde aeropuertos o con tarifas especiales.

Correlaciones que observamos destacables son:

- total_amount y fare_amount dando una relación de 0.98.
- Improvement_surcharge y MTA_tax dando una relación de 0.91.
- tip_amount y total_amount dando una relacion de 0.69

Generamos nueva features como las siguientes:

- **trip_time_min**: se calculó como la diferencia entre la hora de inicio (tpep_pickup_datetime) y la hora de finalización del viaje (tpep_dropoff_datetime), expresada en minutos. Esta variable permite analizar la duración efectiva de los trayectos, identificar viajes atípicamente largos o cortos, y explorar relaciones con la distancia o el monto pagado.
- **pickup_day**: se extrajo a partir del campo tpep_pickup_datetime, indicando el día de la semana en que ocurrió el viaje (por ejemplo, lunes, martes, etc.). Esta variable resulta útil para detectar variaciones en la demanda y en el comportamiento de los usuarios según el día, como diferencias entre días laborables y fines de semana.
- **pickup_hour**: también derivada de tpep_pickup_datetime, representa la hora del día (de 0 a 23) en que se realizó el viaje. Su inclusión permite analizar patrones horarios de uso del servicio, como los picos en horas laborales o nocturnas, y relacionarlos con la tarifa o la duración del viaje.
- **fare_per_passenger**: se creó dividiendo el monto total de la tarifa base (fare_amount) por la cantidad de pasajeros (passenger_count). Esta variable permite analizar el costo promedio por persona, útil para detectar posibles errores de carga (por ejemplo, un solo pasajero con tarifa grupal) o explorar decisiones económicas de los usuarios (como compartir viajes).

  Estas variables no sólo permiten profundizar en el análisis exploratorio, sino que también abren la posibilidad de construir modelos predictivos o clasificaciones más precisas al capturar aspectos temporales, económicos y de comportamiento del usuario.

Durante el proceso de limpieza y validación del dataset, se definieron una serie de reglas de consistencia y rangos aceptables con el objetivo de identificar y filtrar registros con valores atípicos o directamente inválidos. Estas reglas se establecieron considerando el conocimiento del dominio del problema (servicio de taxis en Nueva York) y la documentación oficial del dataset, permitiendo detectar inconsistencias tanto en variables numéricas como categóricas.

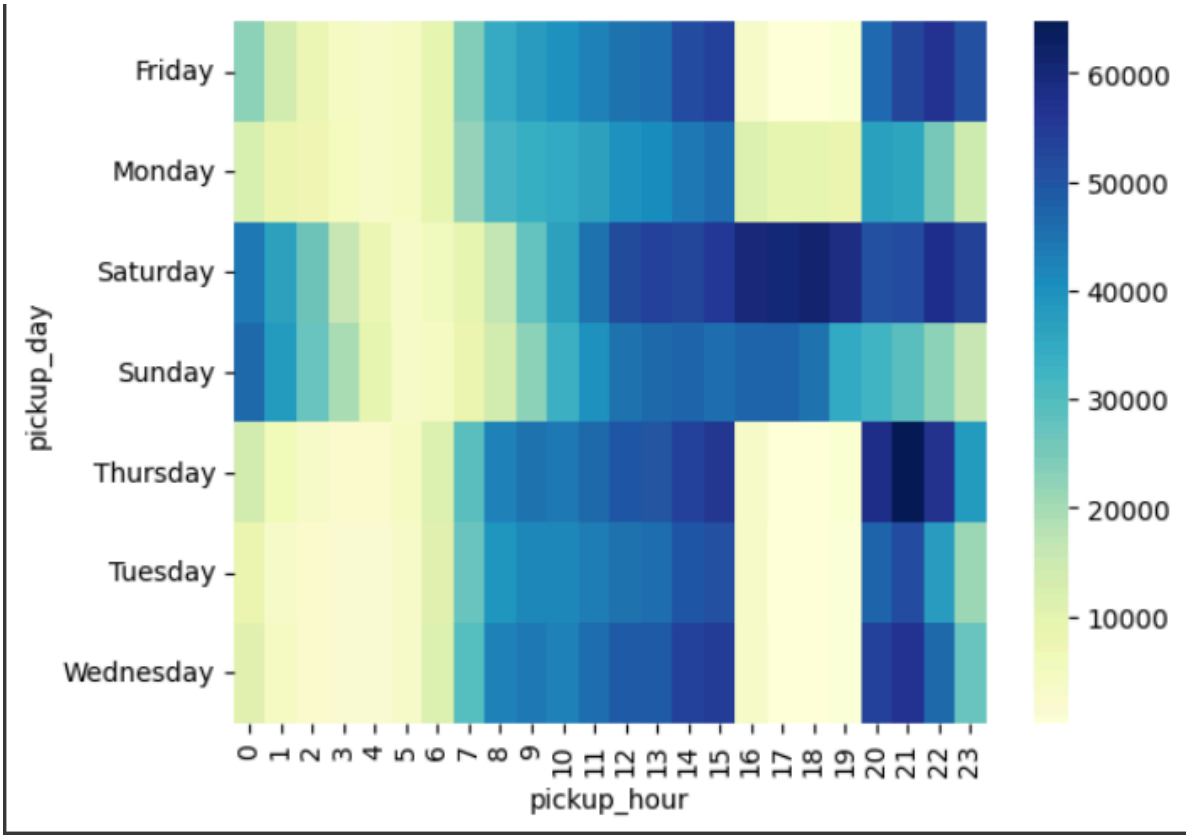
Se consideraron atípicos, por ejemplo, aquellos viajes con una cantidad de pasajeros igual a cero, distancias nulas o negativas, tarifas base o montos totales iguales o menores a cero, así como códigos fuera de los rangos definidos para variables como VendorID, RatecodeID o payment_type. También se descartaron registros con valores imposibles o fuera de lo esperado en campos como store_and_fwd_flag, extra o Airport_fee, cuya codificación está claramente delimitada.

Estas condiciones nos permitieron aplicar un filtro lógico al conjunto de datos y eliminar los que consideramos valores erróneos. Después si hay valores atípicos posibles que tienen sentido como algunos viajes largos en millas, algún viaje con demasiados pasajeros que se podría considerar con un vehículo más grande de lo normal, etc.

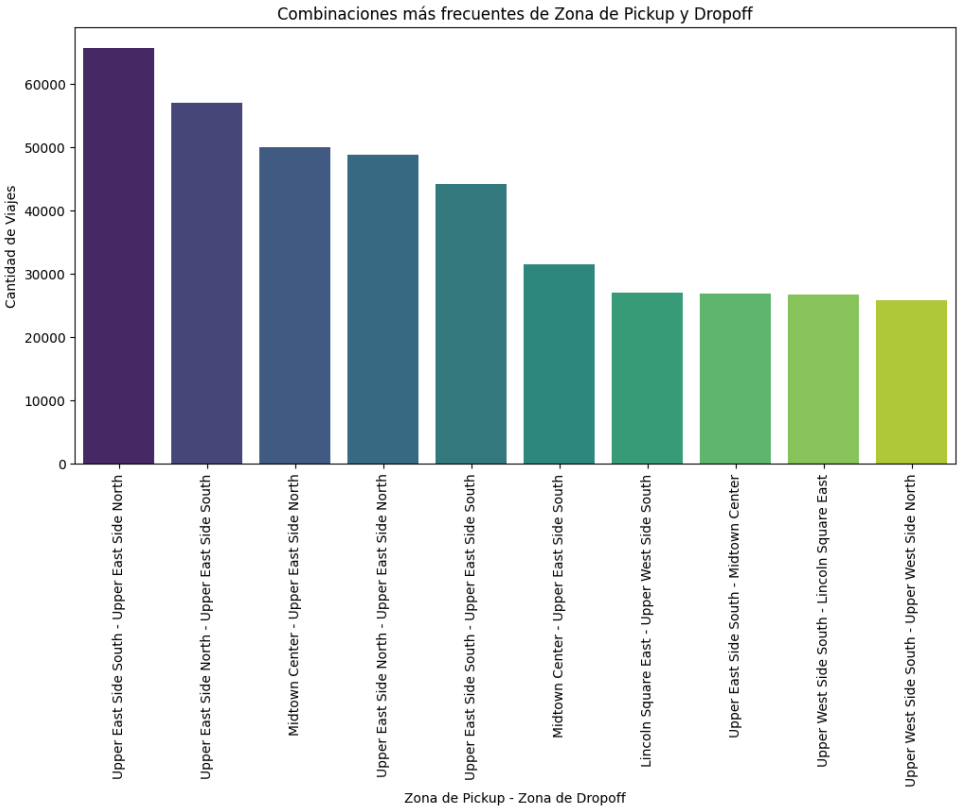
Las columnas con datos faltantes eran las siguientes: 'passenger_count', 'RatecodeID', 'store_and_fwd_flag', 'congestion_surcharge', 'Airport_fee

La mayoría con un porcentaje respecto al total de 7% aproximadamente, en base a esto la decisión que tomamos es imputarlos con el valor de la mediana en los variables numéricas y con el más frecuente en variables categóricas.

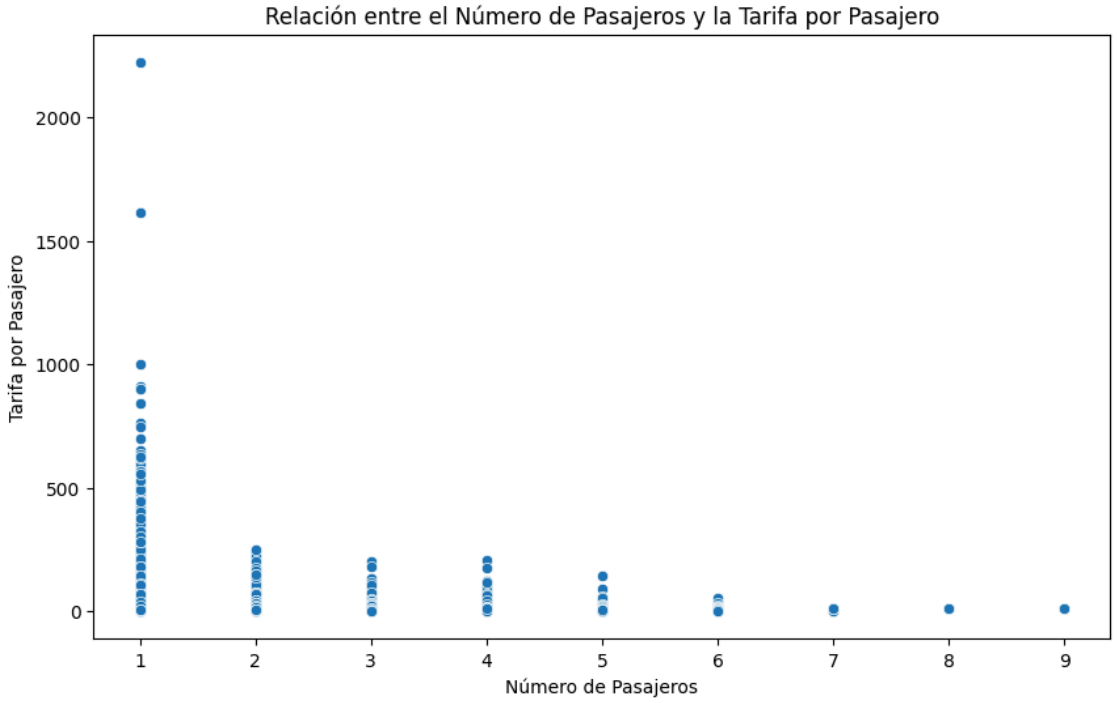
Visualizaciones:



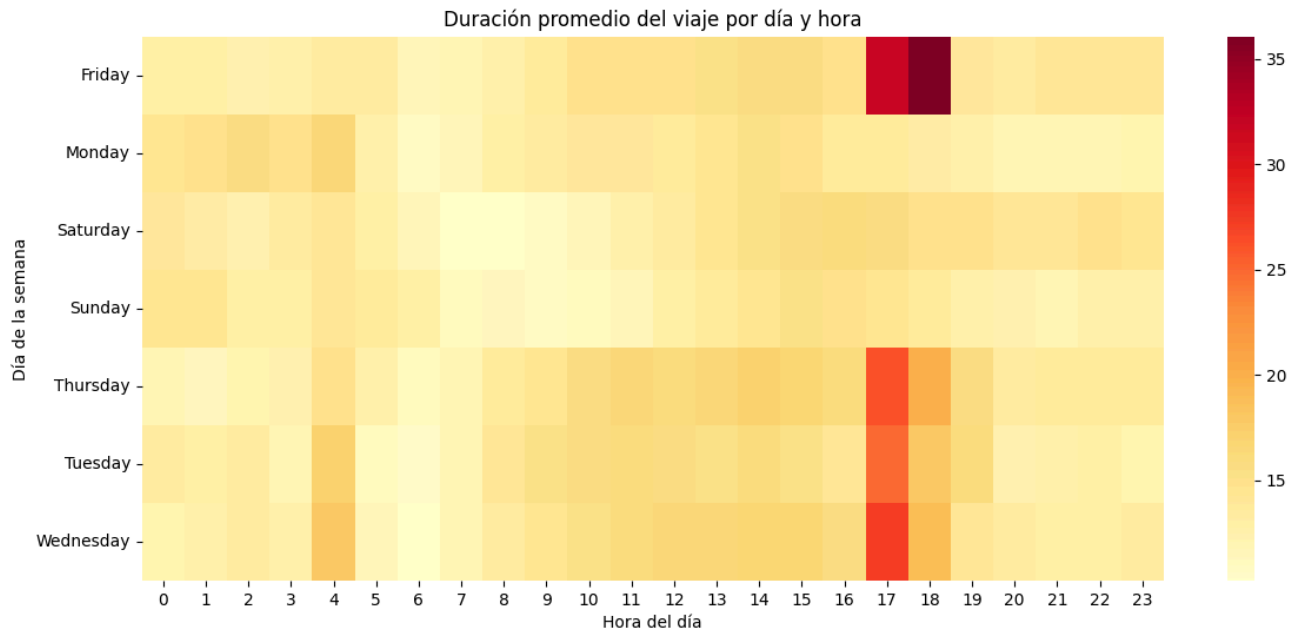
En este gráfico podemos observar según cual sea el dia de semana a qué hora se suelen tomar más taxis. Vemos que claramente todos los días en el horario de 9hs a 15hs y luego también de 20hs a 21hs hay un alto valor. También vemos que los fines de semana en la madrugada de 00hs a 2hs hay otro pico.



En este gráfico podemos ver las combinaciones más frecuentes entre las distintas zonas, destacando las zonas que tienen mayor cantidad de viajes entre sí



Acá podemos observar que la feature que decidimos generar puede ser útil ya que estamos relacionando el precio por pasajero con la cantidad de pasajeros. Se ve claramente que a medida de que haya más pasajeros el precio baja notoriamente, beneficiando compartir el viaje.



En este podemos observar la relación entre el día, la hora y su duración, se puede observar claramente que en el horario de 17hs a 18hs los días martes,miércoles,jueves y viernes. Los viajes son más largos, esto se puede dar debido al tráfico y la hora pico, lo llamativo es que el lunes no esté en estos niveles.

EJ2: Clasificación:

El dataset utilizado contiene registros meteorológicos históricos de distintas estaciones climáticas distribuidas en Australia. En su forma original, el archivo weatherAUS.csv presenta 65298 registros y 23 columnas, abarcando variables climáticas como temperatura, humedad, velocidad del viento, precipitaciones, entre otras, así como una variable de interés llamada RainTomorrow, que indica si se espera lluvia al día siguiente (valor binario: Yes/No).

Para el presente análisis, se realizó un filtrado geográfico con el fin de acotar el estudio a dos estados australianos específicos: Victoria y Nueva Gales del Sur. Para ello, se seleccionaron únicamente las observaciones correspondientes a 18 ubicaciones (8 de Victoria y 10 de Nueva Gales del Sur), lo que redujo el conjunto a 37667 registros. Se

incorporó además una nueva columna llamada Estado, que clasifica explícitamente cada ciudad en uno de los dos estados, facilitando comparaciones y análisis por región.

Decidimos eliminar las columnas "Evaporation", "Sunshine", "Cloud9am", "Cloud3pm" ya que tenian un alto porcentaje de valores nulos, arriba del 40%

Modelos:

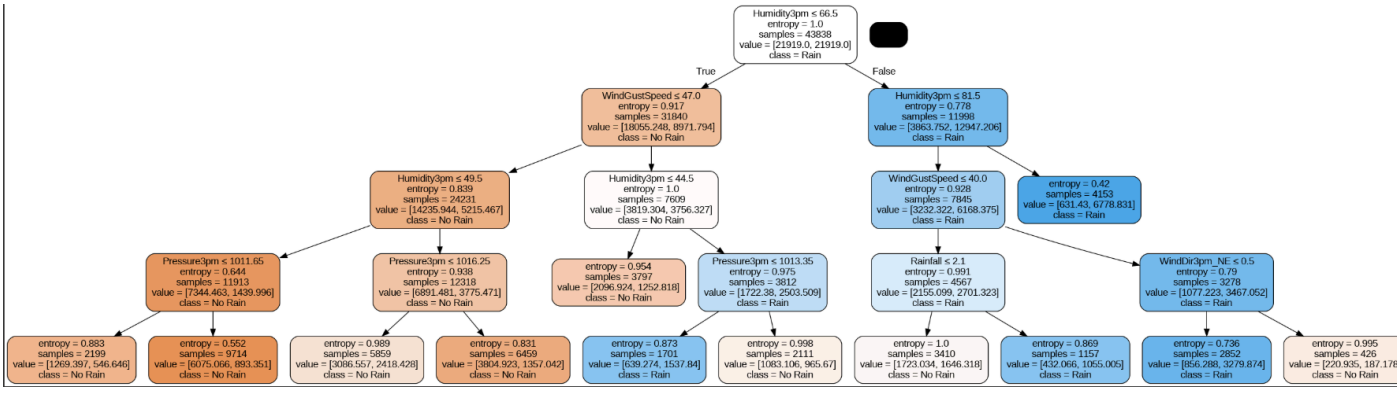
- ## Arbol de Decision

Sí, en esta etapa del trabajo se realizó un proceso de optimización de hiperparámetros con el objetivo de mejorar el rendimiento del modelo de clasificación. Para ello, se utilizó un enfoque de búsqueda aleatoria (RandomizedSearchCV) sobre un conjunto definido de combinaciones posibles de parámetros para el algoritmo DecisionTreeClassifier. Entre los hiperparámetros evaluados se encuentran: el criterio de división (criterion), el parámetro de poda mínima (ccp_alpha), la profundidad máxima del árbol (max_depth), así como los mínimos valores requeridos para realizar una división interna (min_samples_split) y para ser considerado una hoja (min_samples_leaf).

La evaluación de las distintas configuraciones se realizó mediante validación cruzada estratificada con 8 folds, lo que permitió mantener la proporción de clases en cada partición, asegurando así una evaluación más robusta y representativa del desempeño general del modelo.

Como métrica principal para guiar el proceso de optimización se utilizó el F1-score macro, una medida que toma en cuenta tanto la precisión como el recall para cada clase por separado y luego promedia los resultados. Una vez identificados los mejores hiperparámetros, se procedió a entrenar el modelo final utilizando todo el conjunto de entrenamiento con la configuración óptima encontrada.

Imagen Del árbol llegado:



El nodo raíz, es decir, la primera división, se basa en el valor de la variable Humidity3pm (humedad a las 15:00 horas). Si el valor de humedad es menor o igual a 66.5, el modelo tiende a predecir que no lloverá; mientras que si supera ese umbral, la predicción es que sí lloverá. Esta variable aparece como el principal factor discriminante en el modelo, lo que tiene sentido desde el punto de vista meteorológico, ya que altos niveles de humedad por la tarde suelen asociarse a condiciones de lluvia.

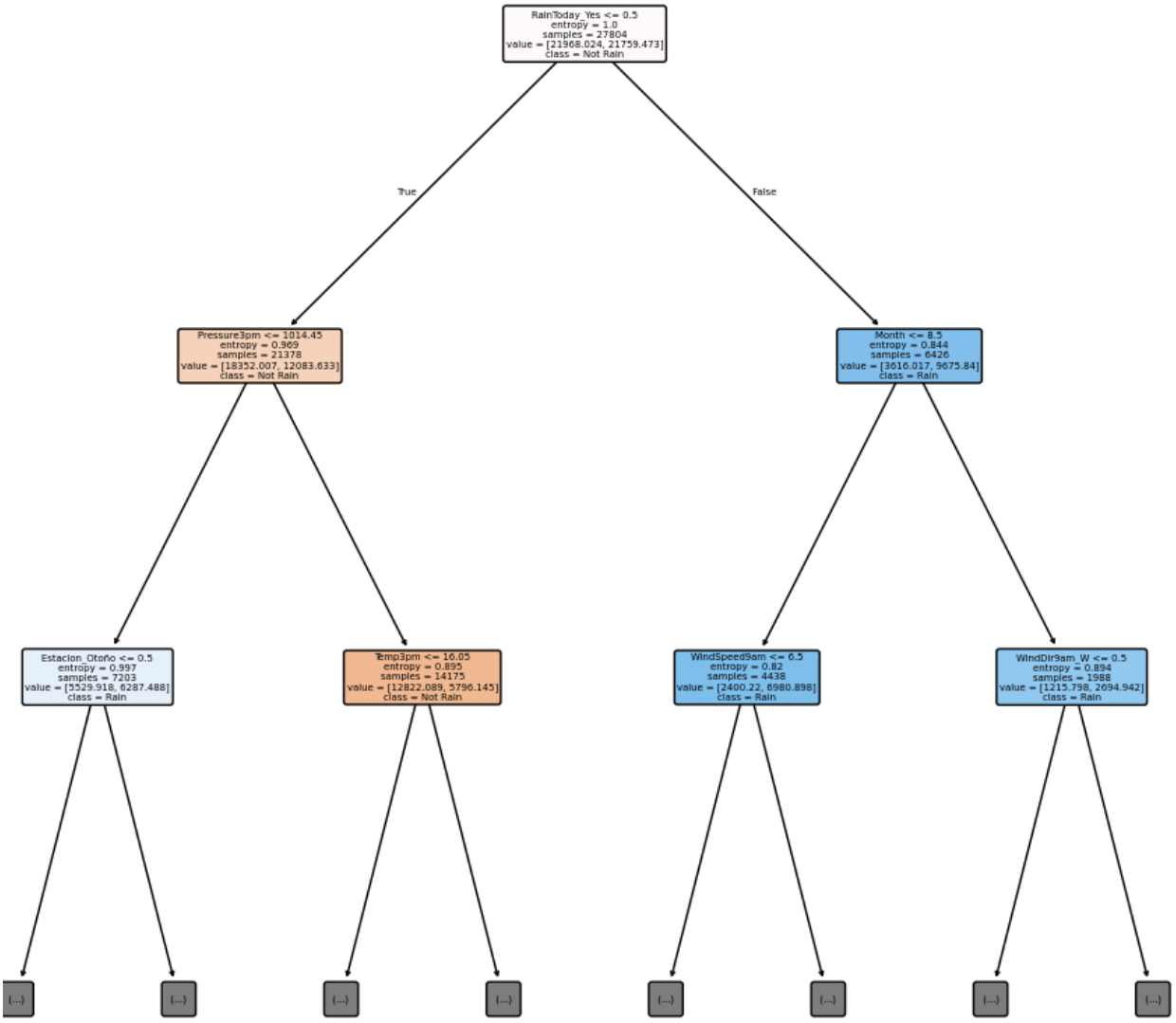
En el segundo nivel, se observa una nueva división para cada una de las ramas anteriores. Para los casos en los que la humedad es baja (≤ 66.5), la siguiente variable clave es la velocidad de ráfagas de viento (WindGustSpeed). Si esta es inferior o igual a 47 km/h, el modelo sigue orientado hacia la predicción de “No Rain”. Esto sugiere que condiciones de viento más suaves están asociadas a ausencia de precipitaciones.

Por otro lado, para los casos en que la humedad a las 15:00 h es mayor a 66.5, el segundo nivel de decisión considera nuevamente la variable Humidity3pm, pero ahora con un umbral superior: si la humedad es menor o igual a 81.5, el modelo refuerza la predicción de lluvia, mientras que valores aún más altos abren nuevas ramas que profundizan en otras variables como el viento y la lluvia acumulada.

- ## Random Forest

En el modelo Random Forest implementado, se llevó a cabo un proceso de optimización de hiperparámetros utilizando una búsqueda aleatoria (RandomizedSearchCV). Durante este proceso se exploraron distintas combinaciones de parámetros relevantes del modelo, como el número de árboles en el bosque (n_estimators), el criterio de calidad de la división (gini o entropy), la profundidad máxima de los árboles (max_depth), así como los mínimos requeridos para dividir un nodo (min_samples_split) y para considerarlo hoja (min_samples_leaf). Para evaluar el desempeño de cada configuración, se aplicó validación cruzada estratificada con 5 particiones (StratifiedKFold con 5 folds), asegurando una representación equilibrada de las clases en cada subdivisión. La métrica empleada para guiar la búsqueda de los mejores hiperparámetros fue el F1-score, dado que permite equilibrar precisión y exhaustividad, lo cual es especialmente útil en contextos con clases desbalanceadas.

Imagen de uno de los árboles generados



En este árbol de decisión, la primera regla de decisión se basa en la variable RainToday: si su valor es menor o igual a 0.5 (lo que puede interpretarse como "No llovió hoy"), el modelo sigue el camino de la izquierda; de lo contrario (es decir, si llovió), va hacia la derecha. Esta división inicial ya genera una separación notable entre los casos de "Lluvia" y "No Lluvia" para el día siguiente.

Si se sigue el camino de la izquierda (RainToday ≤ 0.5), la siguiente variable que se analiza es Pressure3pm. Si esta presión es menor o igual a 1014.45, el modelo continúa evaluando con base en Estación_Otoño, mientras que si es mayor, se analiza la variable Evaporation.

Por otro lado, si se sigue el camino derecho (RainToday > 0.5), la siguiente división se realiza en función de la variable Roth. Dependiendo de si esta es menor o igual a 1.5 o mayor, el modelo evalúa luego la WindSpeed3pm o WindDir3pm_V para afinar la predicción

- AdaBoost

Se realizó una optimización de hiperparámetros para el modelo AdaBoost, ajustando el número de estimadores, la tasa de aprendizaje y la profundidad máxima de los árboles base. Para esta búsqueda se utilizó validación cruzada con 5 particiones (Stratified K-Fold), lo que garantiza una evaluación equilibrada entre las clases en cada subdivisión. La métrica empleada para seleccionar los mejores hiperparámetros fue el puntaje F1, ya que ofrece un buen equilibrio entre precisión y exhaustividad en problemas de clasificación binaria desbalanceada.

Cuadro de Resultados:

Modelo	F1-test	Precision test	Recall test	Accuracy test
Arbol de decision	0.5706	0.5730	0.5681	0.8088
Random Forest	0.6196	0.5589	0.6951	0.8092
AdaBoost	0.5592	0.7006	0.4653	0.8360

Aunque AdaBoost presenta una mayor precisión en el conjunto de prueba (0.7006), el modelo más adecuado para predecir si lloverá o no al día siguiente es Random Forest, y esto se justifica considerando el objetivo del problema y el equilibrio de métricas.

En problemas de predicción de lluvia, es especialmente importante identificar correctamente los días en los que sí lloverá. Para eso, la métrica más relevante es el recall, ya que mide qué proporción de los días con lluvia fueron efectivamente detectados por el modelo. En este sentido, Random Forest tiene el mejor recall (0.6951), lo que significa que es el que menos días lluviosos deja pasar desapercibidos, algo clave si pensamos en prevenir o prepararse ante condiciones climáticas adversas.

Además, el modelo también alcanza el mejor F1-score (0.6196), que equilibra tanto la precisión como el recall, y una accuracy competitiva (0.8092). En cambio, aunque AdaBoost es más preciso, tiene un recall bastante bajo (0.4653), lo que indica que falla más frecuentemente al detectar días de lluvia, un riesgo mayor si nuestra prioridad es evitar falsos negativos (es decir, decir que no lloverá cuando en realidad sí lo hará).

Por tanto, Random Forest representa el mejor compromiso entre detectar días lluviosos correctamente y mantener un buen rendimiento general, lo cual lo convierte en la opción más confiable para este tipo de predicción.

Ejercicio 3: Regresión

Para este ejercicio, utilizamos un archivo .csv que contenía información del precio de los alquileres en AirBnb para la Ciudad de Nueva York. Comenzamos haciendo un preprocesamiento de los datos en algunas de las variables del dataset, aplicando por ejemplo limpieza de datos, transformación de los mismos e ingeniería de features. Dentro de lo que preprocesamos se encuentra, entre otras cosas:

- Conversión de price a numérico: era un string que además contenía comas y signos peso ($), lo convertimos a float quitando esos caracteres, pasando de $1,230.00 a 1230.00.

- Eliminación de valores faltantes en el target: eliminamos los valores faltantes de la variable price y filtramos sus valores outliers

- Aplicamos Log-transform a price para estabilizar la varianza

- Transformamos el campo hot_since de una fecha a un contador de años de experiencia

- Creamos nuevos atributos numéricos como por ejemplo la longitud de la descripción

- Contamos la cantidad de amenices aplicando flags binarios a cosas como wifi, cocina, calefacción y aire acondicionado.

Modelos

Regresión Lineal

- ¿Qué features seleccionaron para construir el modelo?

De features numéricas seleccionamos a ‘host_experience_years’, ,’description_length’, ‘num_amenities’,,’bathrooms’,,'bedrooms' y ‘beds’. De features categóricas utilizamos ‘neighbourhood_group_cleansed’ y ‘room_type'.

● Evaluar la performance del modelo en el conjunto de evaluación, explicar todas las métricas. Comparar con la performance de entrenamiento.

Las métricas en ambos conjuntos dieron los siguientes resultados:

Dataset	R^2	MAE	RMSE
Train	0.305	$78.1	$132.9
Test	0.288	$79.0	$135.9

Coeficiente de determinación:

Haber obtenido un 0.305 nos indica que el modelo explica un 30.5% de la varianza del precio en los datos de entrenamiento y el 28.8 % en los datos de test. La caída de solo 0.017 entre uno y otro demuestra que el modelo generaliza bien y no está memorizando el ruido de los datos de entrenamiento.

Error absoluto medio:

En promedio, las predicciones difieren del precio real por $79. Que se mantengan tan similares en ambos modelos nos dice que el preprocesamiento, la selección de features y la penalización dieron lugar a un modelo estable y robusto.

Raíz del error cuadrático medio:

Penaliza más las desviaciones grandes. Un RMSE de $136 indica que hay casos en los que el modelo se equivoca por aproximadamente $136 de media cuadrática, algo que ocurre especialmente en la parte alta de precios. Además de las características recién mencionadas que nos indican los números tan cercanos entre train y test, también podríamos decir que trabajamos con un overfitting bajo.

XGBoost

● ¿Utilizaron K-fold Cross Validation?¿Cuántos folds utilizaron?

Sí, utilizamos K-Fold Cross Validation (CV). Utilizamos 10 folds y unas 40 iteraciones, dando un total de 400 fits para intentar obtener un entrenamiento más preciso.

● ¿Qué métrica utilizaron para buscar los hiperparámetros?

Elegimos RMSE como criterio principal, implementado en scikit-learn con el parámetro scoring='neg_root_mean_squared_error'. La justificación es que, en problemas de precios de alquiler, creímos correcto penalizar especialmente los errores grandes (por ejemplo, una predicción errónea de cientos de dólares), y RMSE aumenta de manera cuadrática ante desviaciones importantes. Los mejores hiperparámetros obtenidos fueron:

learning_rate=0.1, max_depth=6, n_estimators=300, subsample=1.0, colsample_bytree=0.8, reg_alpha=0 y reg_lambda=5

● Evaluar la performance del modelo en el conjunto de evaluación, explicar todas las métricas. Comparar con la performance de entrenamiento.

Dataset	R^2	MAE	RMSE
Train	0.842	$33.9	$63.1
Test	0.551	$59.6	$108.8

R^2: En el conjunto de evaluación, el XGBoost obtuvo un R^2 de 0.551, lo que significa que explica el 55.1 % de la varianza del precio.

MAE: Se obtuvo un error absoluto medio de $59.6, es decir, la predicción media se desvía en torno a $60 del valor real.

RMSE: Alcanzamos una raíz del error cuadrático medio de $108.8, que penaliza con fuerza las desviaciones grandes; y un MSE de 11852.58, que refleja el cuadrado medio de esos errores.

En entrenamiento, estas métricas fueron $R^2 = 0.842$, MAE = $33.9 y RMSE = $63.1, por lo que existe un gap moderado entre un modelo y el otro.

Cuadro de Resultados

Modelo	MSE (RMSE^2)	MAE	RMSE
Regresión Lineal	18400	79.0	135.9
XGBoost	11852	59.6	108.8

Elección del modelo

Si tuviera que elegir uno de los dos modelos para predecir los precios de los alquileres de AirBnb en la Ciudad de Nueva York, elegiría XGBoost. ¿Por qué? Porque ofrece una ganancia clara en precisión frente a la regresión lineal. El R^2 obtenido explica más de la mitad de la varianza (0,55 frente a 0,29), reduce el error medio de $79 a casi $60 y el RMSE de $136 a casi $109, demuestrando que captura relaciones importantes y no lineales que el otro modelo, debido a su naturaleza lineal, no puede. El ahorro en error absoluto y cuadrático hace de XGBoost la mejor opción cuando el objetivo principal es maximizar la exactitud en la predicción de precios variables, ya que un modelo con menor margen de error absoluto significa que, en promedio, la desviación entre el precio real y el predicho es pequeña, obteniendo predicciones más fiables.

EJ4: Clustering:

-Tendencia al clustering:

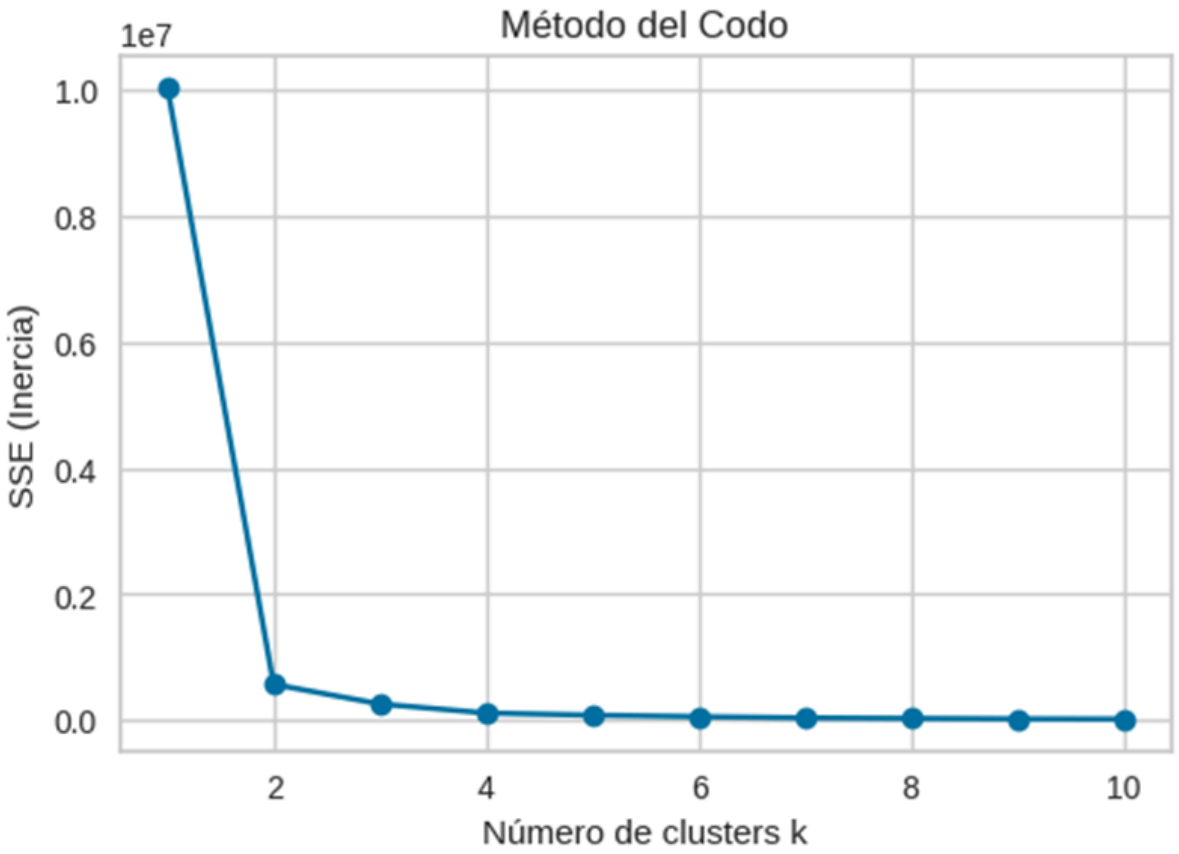
La tendencia al clustering fue medida a través de la **Estadística de Hopkins** ,para ver si los datos están distribuidos aleatoriamente (**H cercano a 0.5**) o con estructura de agrupamiento (**H cercano a 1**).

El resultado obtenido fue **0.8223** lo que indica una muy buena tendencia a la formación de clusters en el dataset, ya que es muy cercano a **1**. Por lo tanto, es apropiado aplicar técnicas de clustering sobre estos datos.

-Cantidad de grupos:

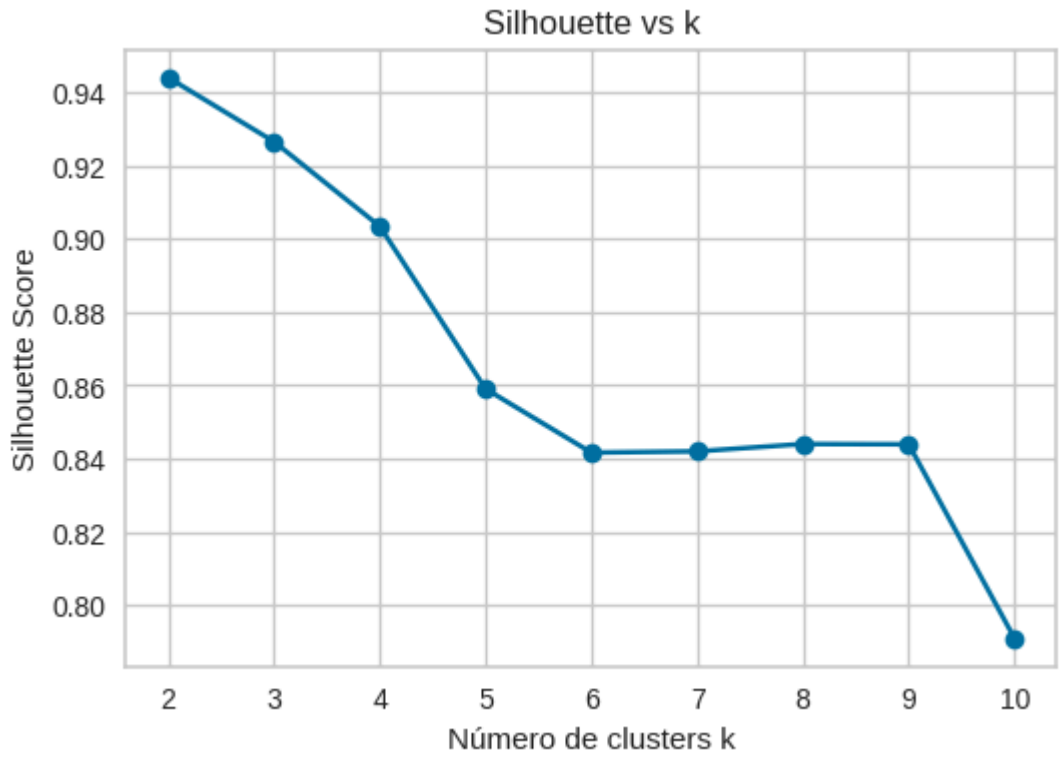
Para obtener la cantidad apropiada de grupos se utilizó **Regla del codo (Elbow Method).** Esta técnica consiste en graficar la suma de los errores cuadráticos dentro de los clusters **(SSE)** en función del número de clusters y buscar el punto donde la disminución de **SSE** comienza a estabilizarse, lo que indica el número adecuado de clusters.

El resultado obtenido fue **k=2** , donde se observa el punto donde se forma el "codo".



Para evaluar qué tan bien se han formado los grupos usamos el análisis de Silhouette.

Este método nos permite medir la cohesión interna y la separación entre los clusters. Un valor cercano a 1 indica que los puntos están bien agrupados, mientras que valores cercanos a 0 o negativos sugieren que los clusters podrían estar mal definidos.



-Visualización gráfica de los grupos:

Para graficar los grupos se utilizó **PCA** ya que permite reducir las dimensiones.

Gráfico en 2D:

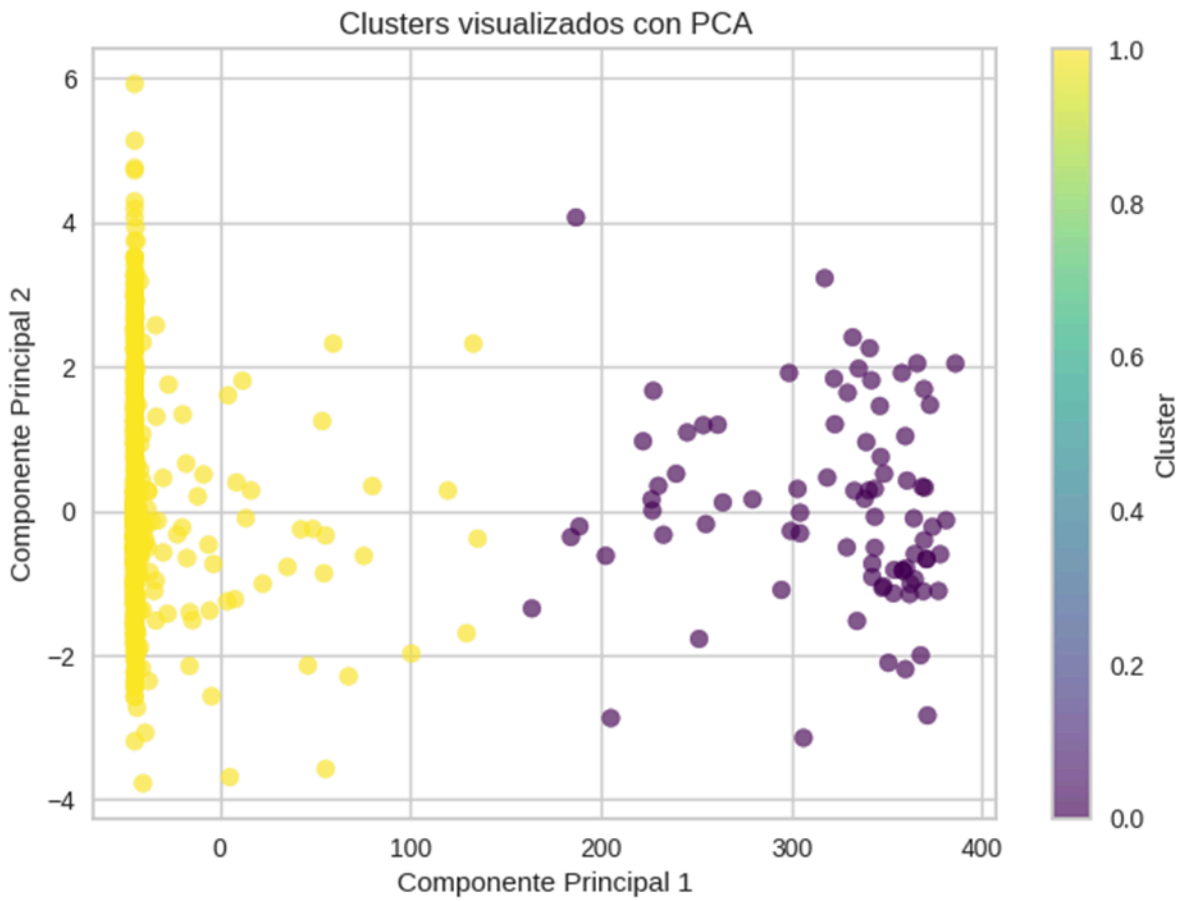
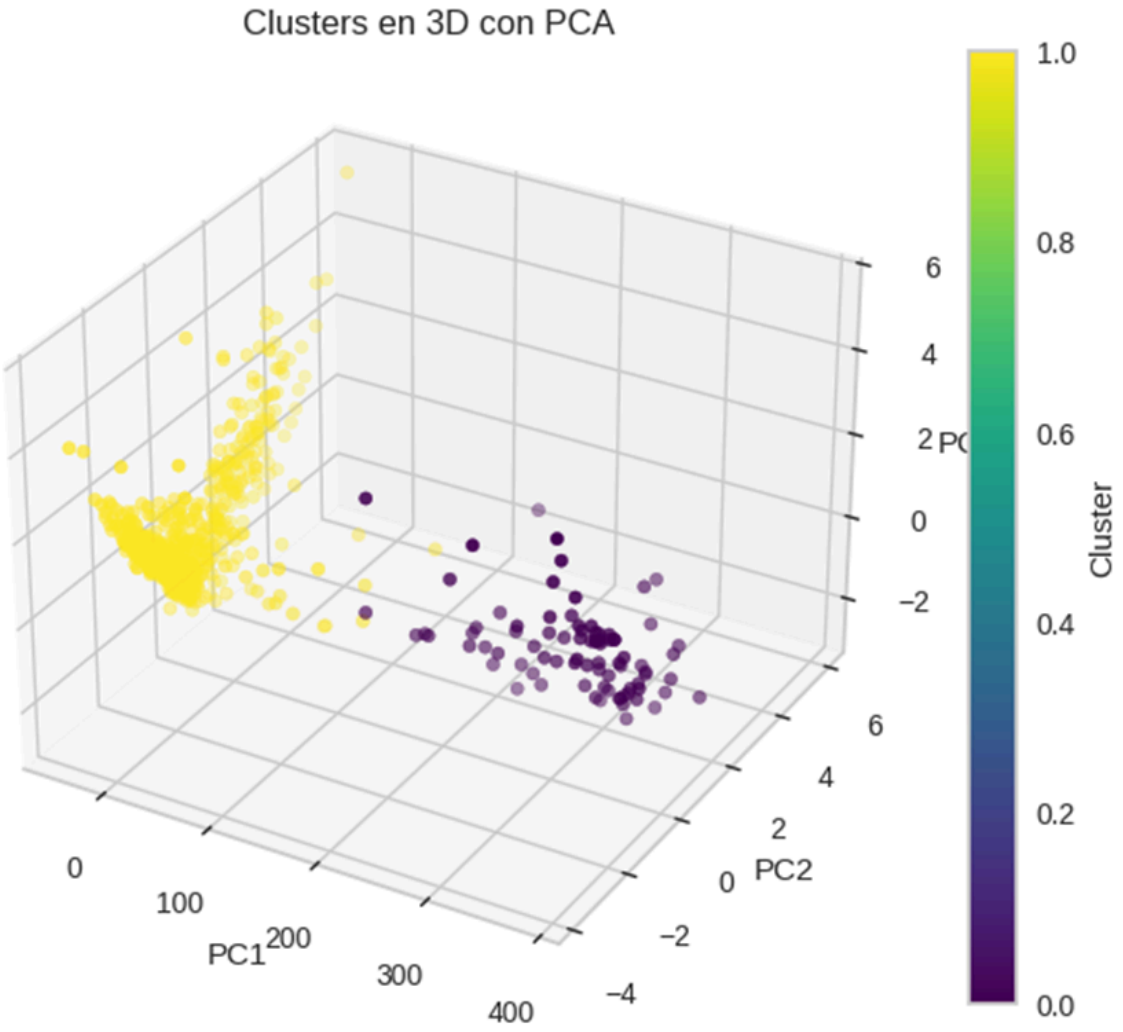

Clusters visualizados con PCA
Componente Principal 2
Componente Principal 1
Cluster
6
4
2
0
−2
−4
0
100
200
300
400
1.0
0.8
0.6
0.4
0.2
0.0

Gráfico en 3D:

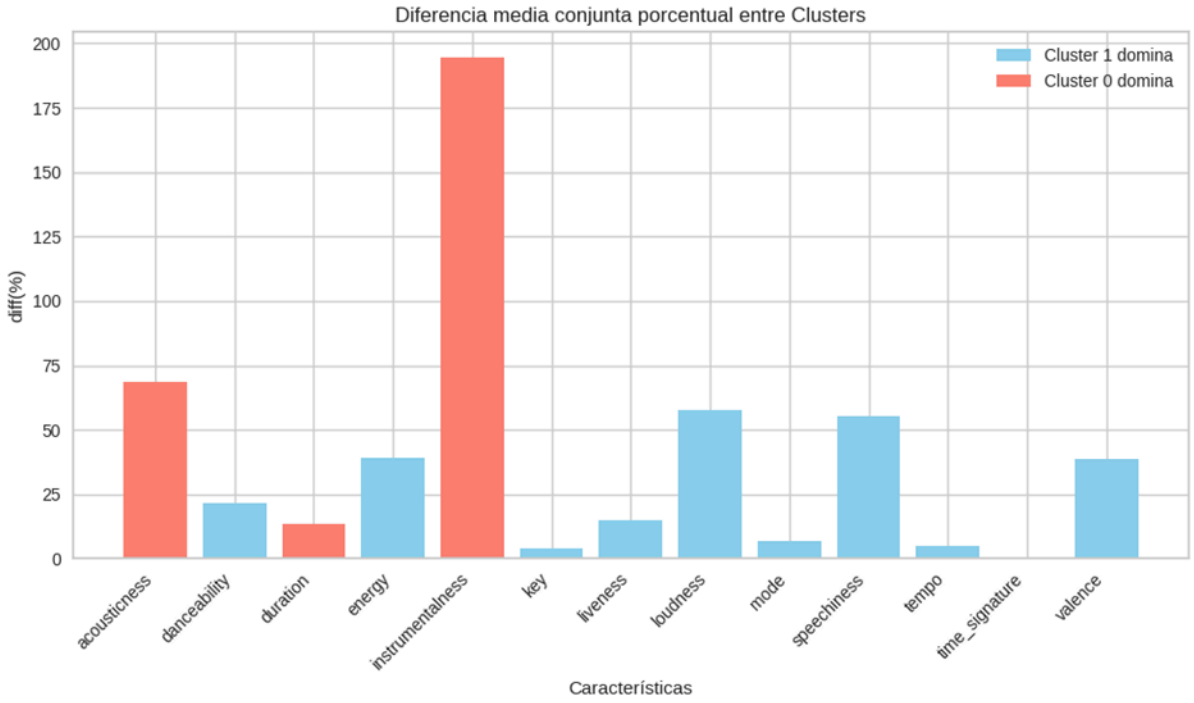


-Características de los grupos:

Para ello se utilizo 2 métodos:

- Comparación entre Clusters (cluster vs cluster), usando sus **medias**.
- Comparación entre Clusters y el dataset (cluster vs global), usando **Z-Score**.

Cluster vs Cluster:



Cluster 1 (comparado con Cluster 0)

-Es mayor que el Cluster 0 en:

- **Loudness**: –7,81 dB → **+44,8%**
  Pasa de un volumen suave (–14,14 dB) a uno intenso (–7,81 dB): música más "potente".
- **Speechiness**: 0,10 → **+76,1%**
  Aumenta la presencia de voces cantadas de 0,059 a 0,10, reflejando más densidad lírica.
- **Energy**: 0,62 → **+48,6%**
  Sube de 0,41 a 0,62, de un estilo moderado a uno dinámico y vibrante.
- **Valence**: 0,52 → **+48,3%**
  Pasa de tonos melancólicos (0,35) a alegres o estimulantes (0,52).
- **Danceability**: 0,61 → **+24,4%**
  De mediano (0,49) a alto (0,61) en ritmos marcados y regulares.
- **Liveness**: 0,21 → **+15,9%**
  Ligera subida de 0,18 a 0,21, sugiriendo algo de ambiente en vivo.

-Es menor que el Cluster 0 en:

- -**Instrumentalness**: 0.01 → **–7182,1%** , casi sin partes instrumentales.
- -**Acousticness**: 0.32 → **–104,3%** , mucho menos acústico.
- -**Duration**: 3.61 min → **–14,4%** , canciones más cortas (0.60 min ≈ 36 segundos).

-Conclusión para Cluster 1:

Agrupa canciones con sonido potente, con voces cantadas, enérgicas y vibrantes, muy rítmicas y bailables, con tonos alegres ,de corta duración, pueden tener algo de sonido de público. ideales para bailar, entrenar o ambientar una fiesta.

Ejemplos de géneros: Pop , Electrónica (Dance/House/Techno), Hip-Hop, Rock Alternativo.

Cluster 0 (comparado con Cluster 1):

-Tiene valores considerablemente mayores que el Cluster 1 en:

• **Instrumentalness**: 0,82 → +7.182,1%
Pasa de 0,01 a 0,82: prácticamente sin voces, enfoque casi puramente instrumental.

• **Acousticness**: 0,65 → +104,3%
Pasa de 0,32 a 0,65: fuerte presencia de instrumentos orgánicos, sonido acústico sobresaliente.

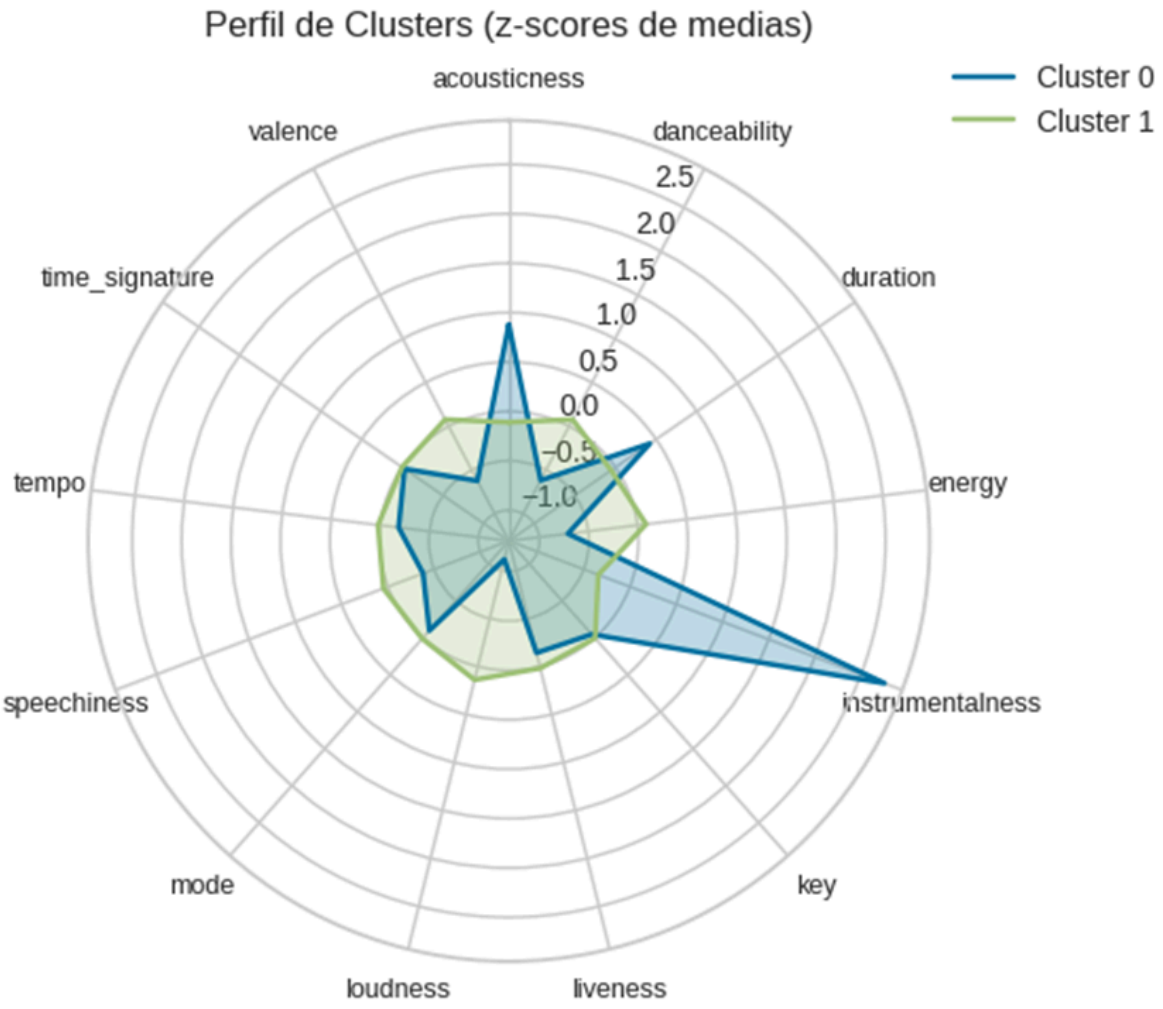
• **Duration**: 4,13 min → +14,4%
Pasa de 3,61 min (≈ 217 s) a 4,13 min (≈ 248 s): pistas más extensas.

-Conclusión para Cluster 0:

Agrupa canciones casi totalmente instrumentales y acústicas,sin voces , con gran presencia de instrumentos organicos, de duracion media-larga, con un volumen y energía suave,con un ambiente relajante y calmo ,con tonos tranquilos o melancolicos, siendo en su mayoria canciones de estudio(sin publico) .Ideales para relajarse , concentrarse o estudiar.

****Ejemplos de géneros:**** Música clásica, Ambient/New Age, Jazz Suave, Folk Acústico, Musica Cinematográfica instrumental.

Cluster vs Global:



Resumen de análisis de clusters (usando z-scores):

- **Cluster 0**
  - Muy alto en **instrumentalness** (≈ +2,75 σ): casi puramente instrumental.
  - Elevado en **acousticness** (+0,88 σ) y **duration** (+0,42 σ): más acústico y ligeramente más largo que el promedio.
  - Muy bajo en: **energy** (–0,71 σ), **loudness** (–1,12 σ), **valence** (–0,62 σ) y **danceability** (–0,62 σ): sugerente de canciones más suaves, melancólicas y menos bailable.
  - El resto de variables (**speechiness**, **liveness**, **mode**, **tempo**, etc.) se sitúan ligeramente por debajo de la media (entre –0,40 σ y –0,03 σ), sin marcadas diferencias.

-**Conclusión para Cluster 0:** Es el grupo de canciones largas, predominantemente instrumentales y acústicos, con bajo volumen y energía, que no están pensados para bailar ni buscan un tono alegre o vocal.

-Ejemplos de Generos: Música clásica, instrumental ,Ambient ,New Age ,Jazz suave,Folk, música de cine.

- **Cluster 1**
    - Destaca por tener valores positivos en casi todas las variables, salvo en ***instrumentalness*** (**-0.34 σ**), ***acousticness*** (**-0.11 σ**) y ***duration*** (**-0.05 σ**),oscilan cerca de **0 σ** (entre **−0.34 σ** y **+0.14 σ**), señalando que contienen menos partes instrumentales y menos elementos acústicos , con una duracion levemente menor.
    - Los pocos valores positivos más notables son ***energy*** (**+0.09 σ**) , ***loudness*** (**+0.14 σ**) y ***valence*** (**+0.08 σ**), lo cual indica canciones ligeramente más enérgicas, ligeramente con mas volumen y con un tono más positivo que el promedio.

-Conclusión para Cluster 1: Es un grupo muy cercano al perfil medio del dataset. Sus pistas son fundamentalmente vocales o electrónicas (poco contenido instrumental), de duración estándar, con energía y volumen apenas superiores al promedio y un tinte emocional algo más positivo.

Ejemplos de Generos: Pop , electronica (Dance/House/Techno), Hip-hop , Rock alternativo, Indie rock ,Punk.

Conclusión General

- El **Cluster 0** agrupa música pensada para escuchar con **calma**: piezas instrumentales o acústicas, ideales para momentos de concentración, relax o lectura.

- El **Cluster 1** reúne temas diseñados para **moverse y disfrutar**: canciones con voces, ritmos marcados y energía alta, perfectas para bailar, hacer ejercicio o animar el ambiente.

Integrantes

Integrante	Tareas	Promedio hs semanales
Nicolás Cardone	Ejercicio 1 y 2 Armado de reporte	5
Jonatan Gomez Godoy	Ejercicio 4 Armado de reporte	5
Nicolas García	Ejercicio 1 y 3 Armado de reporte	5