Trabajo Práctico 1 - Grupo 08

Integrantes:
Avalos, Camila Lucia
Benitez Potochek, Tomás
Roldan Montes, Cristian
Sprenger, Roberta

EJ1: Análisis Exploratorio

Descripción:

Inicialmente el dataset está compuesto por 2.559.642 registros y 19 columnas.

Algunas columnas significativas:

- ➔ **VendorID**
- ➔ **tpep_pickup_datetime**
- ➔ **tpep_dropoff_datetime**
- ➔ **passenger_count**
- ➔ **trip_distance**
- ➔ **PULocationID**
- ➔ **DOLocationID**
- ➔ **payment_type**
- ➔ **tip_amount**
- ➔ **total_amount**
- ➔ **congestion_surcharge**

Features destacadas:

1. VendorID: código que indica el proveedor de TPEP que proporcionó el registro.
    a. Tipo: int
2. Passenger_count: cantidad de pasajeros en el vehículo *(ingresado por el conductor)*
    a. Tipo: float*
    b. Nos permite analizar los viajes en función a la cantidad de pasajeros del mismo
3. Trip_distance: la distancia en millas reportada por el taxímetro
    a. Tipo: float
4. Payment_type: código numérico que representa como pagó el viaje un pasajero.
    a. Tipo: int

Preprocesamiento de Datasets

Outliers (Valores Atípicos)

- **VendorID**: Se identificó un valor atípico con *VendorID == 6*, que no está documentado.
  El porcentaje sobre el total de los registros del dataset es del 0,01% por lo que lo descartamos al considerarlo erróneo y de poco peso.
- **Trip Distance**: Se encontraron observaciones con distancias extremadamente alejadas de la media, que no pueden ser representadas en un boxplot. La mayoría de los viajes registraron distancias entre 0 y 5 millas, con pocos valores superiores a 5 millas.
  Se decidió categorizar las distancias en tres grupos: Cortas (0-5 millas), Medias (6-20 millas) y Largas (más de 20 millas).
- **Payment Type:** Se observó un valor atípico con *payment_type == 0*, que no está documentado. A pesar de ser un outlier, no se modificó debido a la cantidad considerable de estos casos.

Datos Faltantes

- Passenger Count: La variable *passenger_count* presenta un 5% de datos faltantes.
  Se empleó la técnica de imputación de datos, asignando valores como el cero, la media o la mediana. Los viajes registrados con cero pasajeros se incluyeron dentro del grupo de viajes con varios ocupantes.
  Clasificamos los viajes en tres categorías: Solitario, Acompañado o Multitud

Observaciones

- Passenger Count: El tipo de dato asignado originalmente fue *float64*, lo cual se considera incorrecto, ya que no es lógico tener fracciones de pasajeros. Esta variable debe ser numérica, es decir, int32, estamos hablando de un conjunto numerable.

Nuevas Features:

Agregamos las siguientes variables a partir de las columnas existentes.

trip_duration_in_mins

Duración del viaje : calculado a partir de la hora de salida y llegada

hour_pu

Hora de PickUp : extraído de la hora de PickUp

trip_distance_km

Distancia en km : calculado a partir de la distancia en millas

trip_speed

Velocidad promedio km/h : calculado a partir de la distancia en km y la duración en minutos

base_fare

Tarifa base : calculado a partir del total y los extras cobrados

Además, agregamos una nueva feature a partir del dataset de ids de zonas:

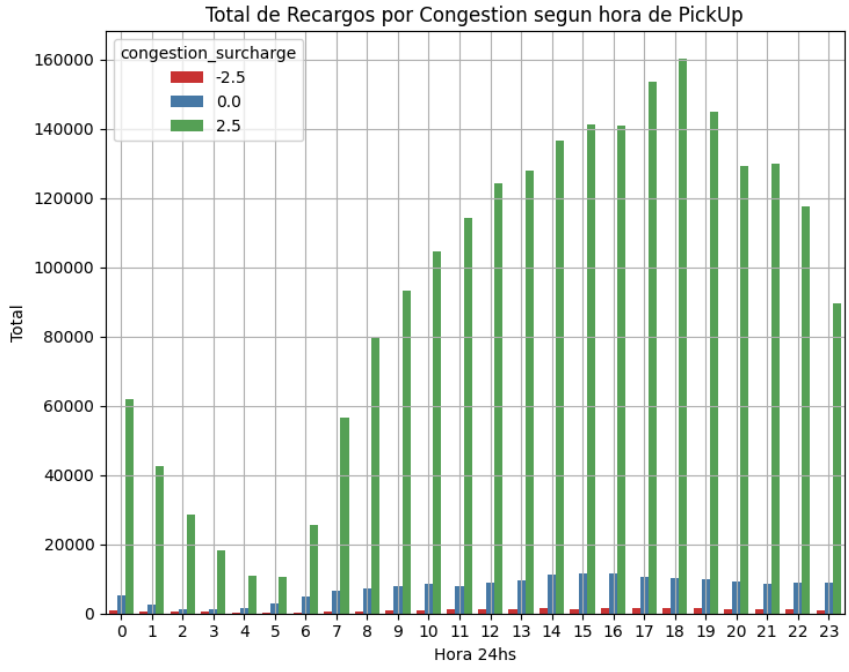
PU_borough y DO_borough

Distrito de Pick-Up/Drop-Off : calculado a partir del merge con dataset de ids de zonas

Visualizaciones

1. Cantidad de Recargos x Hora

Pregunta: ¿Cómo varía la cantidad de recargos por congestión a lo largo del día?



Este gráfico muestra cómo los recargos aumentan según la hora del día. Podemos ver claramente cuál es la hora pico (18hs) ya que la congestión de tráfico y la cantidad de recargos es máxima.

Se eligió este gráfico para analizar la dinámica temporal de los recargos y su posible relación con el tráfico y la demanda.

2. <u>Promedio y Cantidad de Propinas según Tipo de Viaje</u>

Pregunta: ¿Qué relación existe entre el tipo de viaje y las propinas recibidas?

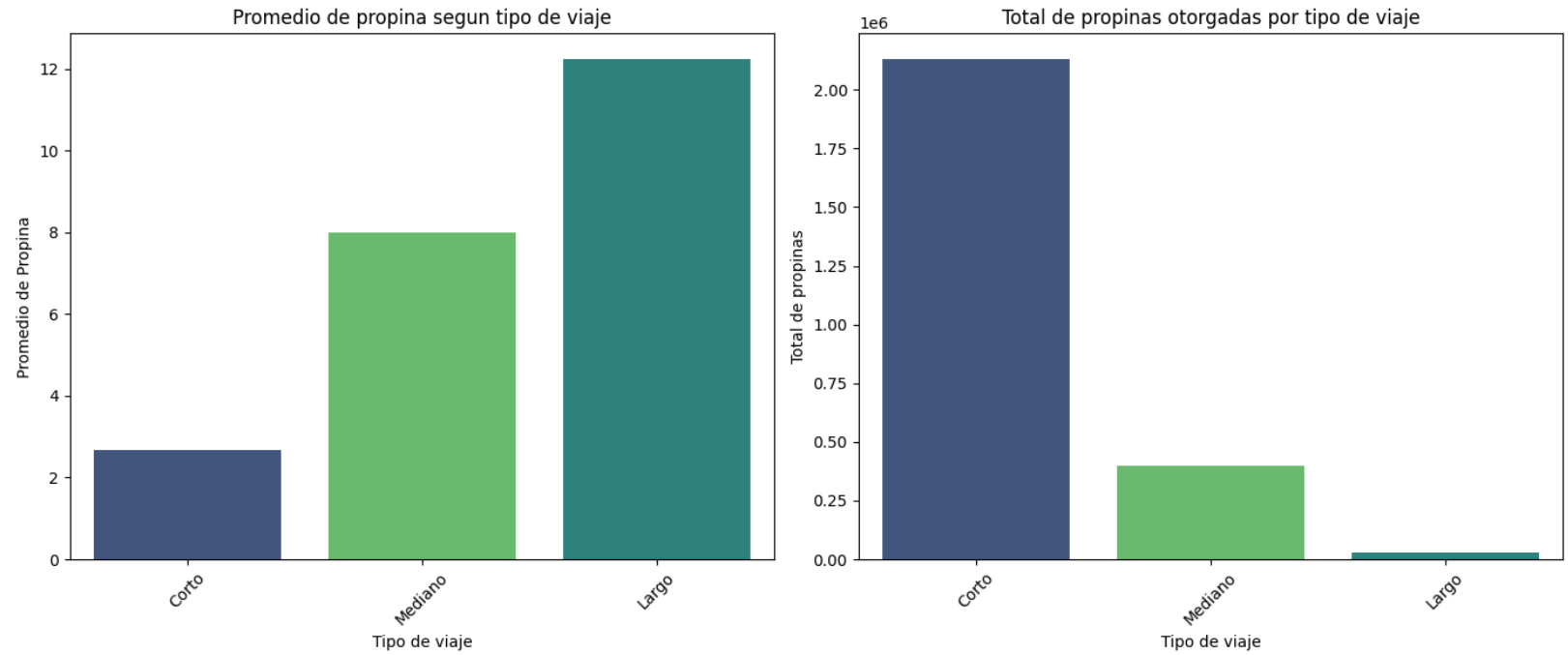


Figura 2: Promedio de propinas por tipo de viaje y total de propinas otorgadas por tipo de viaje

El gráfico de la izquierda se genera agrupando los viajes por distancia (corto, mediano, largo) y calculando el promedio de propinas para cada categoría.

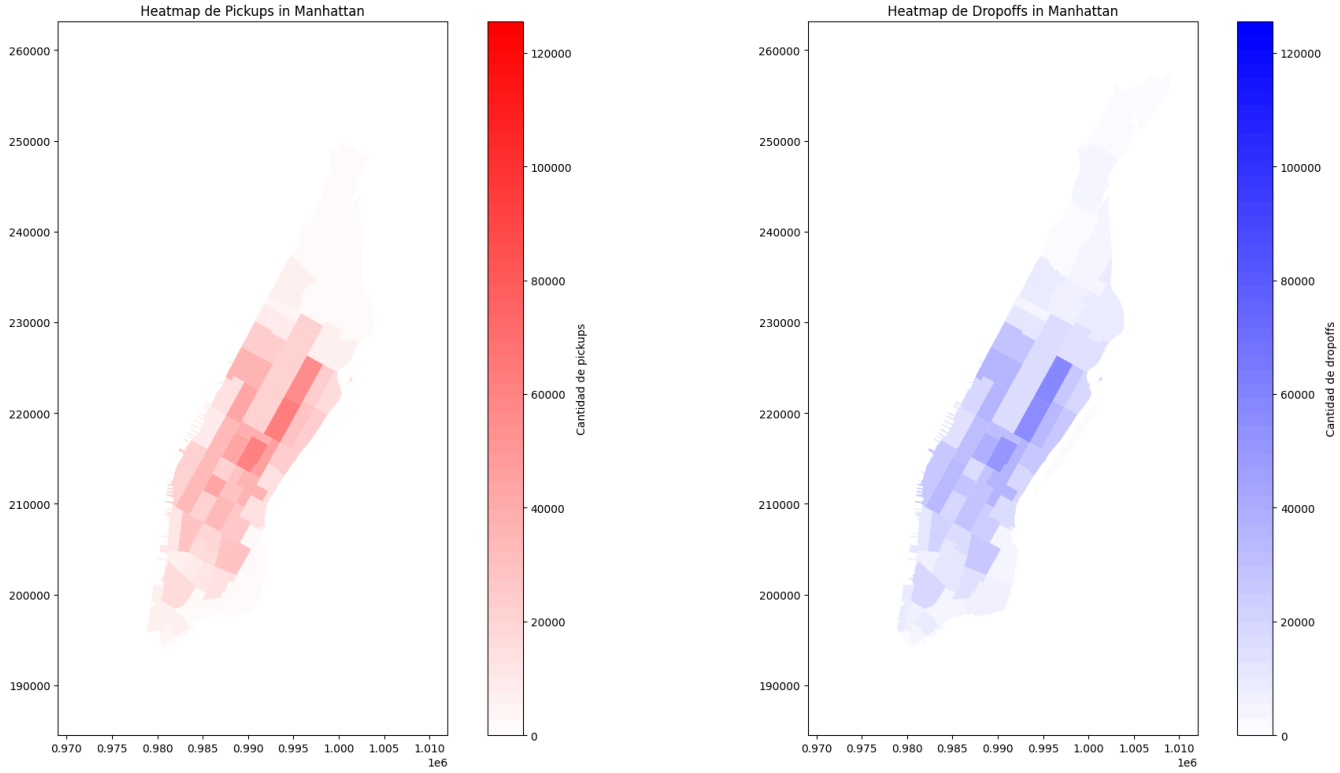
Por otro lado, el gráfico de la derecha se genera contando la cantidad total de propina.

Podemos apreciar que, aunque en promedio los viajes de larga distancia generan más propinas, los viajes cortos son considerablemente más frecuentes, generando así un total mayor de propinas.

Este gráfico fue elegido para entender mejor la relación entre el tipo de viaje y la propina, destacando aquellos tipos que tienden a generar más ganancia.

3. Visualización de Zonas: Cantidad de Pick-ups y Drop-Offs por Zona de Manhattan

Pregunta: ¿Cómo se distribuyen los pickups y drop offs en Manhattan?



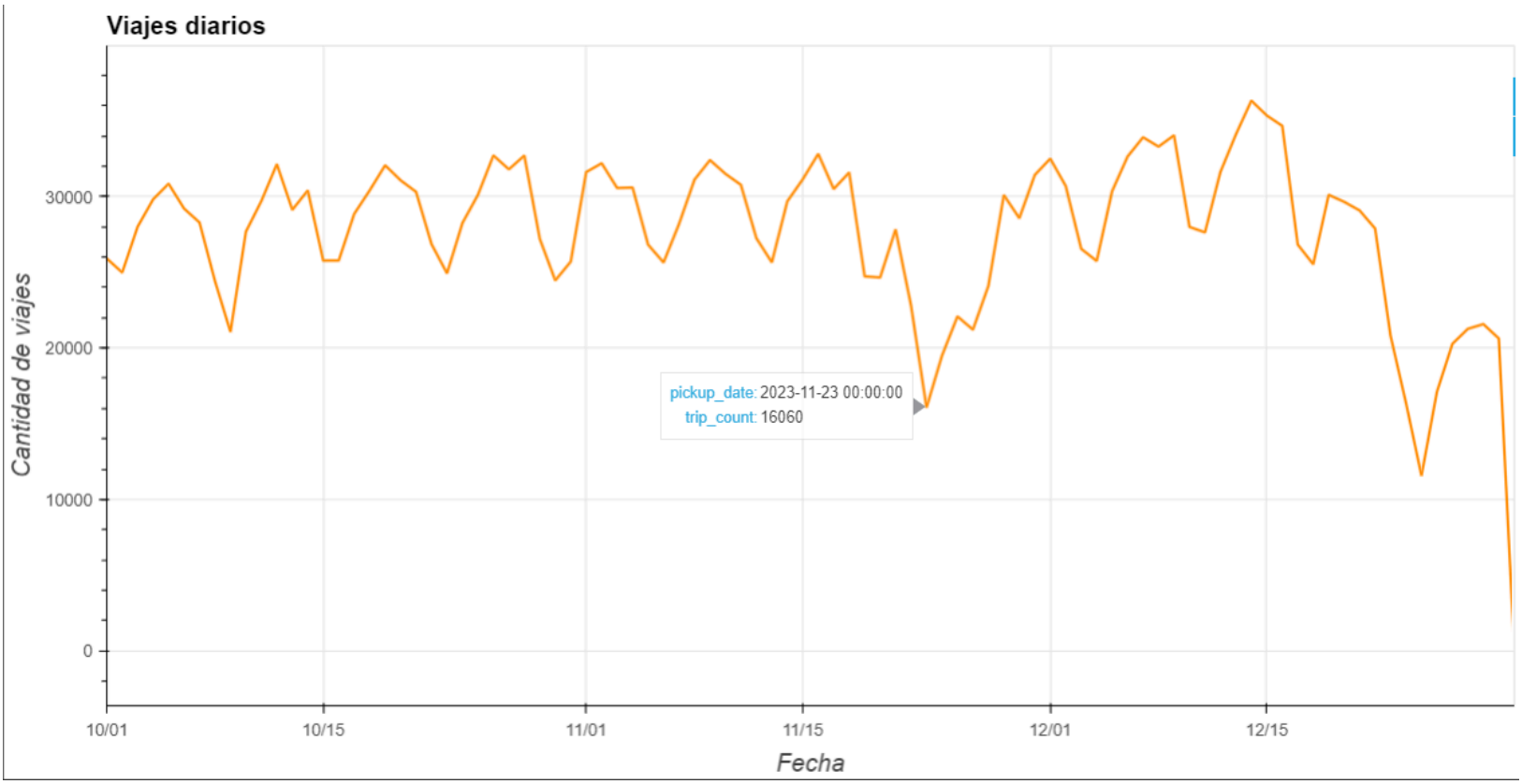
Este heatmap muestra la distribución espacial de los pickups y drop offs en diferentes ubicaciones de Manhattan. Las áreas más oscuras representan mayor densidad de actividad.

Vemos que las zonas más densas de pick up y drop off coinciden en cierto grado pudiendo identificar así aquellas en las que hay más movimiento de personas.

Como último detalle, se puede notar que, para los drop offs, hay más densidad a las afueras de Manhattan comparando con los pickups. Con esto podemos deducir que los viajes tienden a ir desde el centro hacia las afueras.

El heatmap fue seleccionado para analizar la distribución espacial de la demanda en Manhattan.

4. Cantidad de Viajes por Fecha



Pregunta: ¿Cómo varía la cantidad de viajes entre octubre y diciembre de 2023?
Figura 3: Cantidad de viajes por fecha

En esta línea de tiempo se puede ver la tendencia general de la cantidad de viajes a lo largo de los meses elegidos.

Podemos relacionar la caída notable de la cantidad de viajes alrededor del fin de diciembre y principio de enero, ya que en esas fechas están los festejos de navidad y año nuevo.

Al elegir este tipo de gráfico pudimos visualizar patrones de demanda temporal.

5. Correlación entre total de viajes y total de accidentes por día

Pregunta: ¿Existe relación entre la cantidad de viajes y la de accidentes?

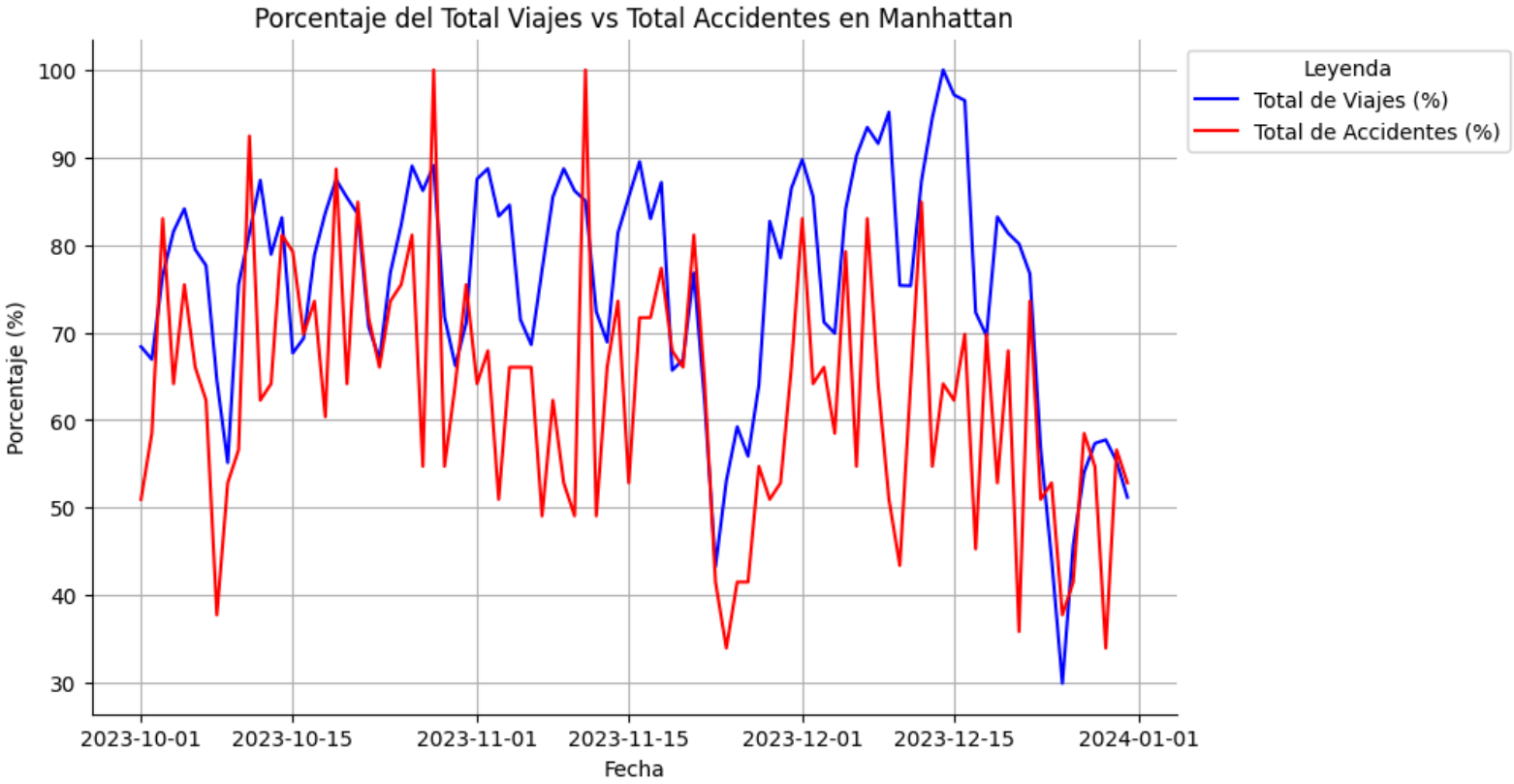


Figura 5: Porcentajes de cantidad de accidentes vs de viajes

Para este gráfico utilizamos un dataset extraído de la página de NYC Open Data que recopila los accidentes reportados por la policía de NYC.

El valor calculado para la correlación entre estas dos variables fué de: 0.483
Vemos que la correlación es moderada positiva entre el porcentaje de viajes totales y el porcentaje de accidentes totales en Manhattan.
Podemos observar que hay cierto nivel de asociación entre las dos variables: a medida que aumenta el número de viajes, también tiende a aumentar el número de accidentes.
Esta correlación podría implicar que, además del volumen de viajes, otros factores influyen en la cantidad de accidentes. Por ejemplo, las condiciones de tráfico o el clima.

EJ2: Clasificación

El dataset con el que trabajamos poseía una cantidad de 145459 registros y 23 columnas de las cuales nos quedamos con las siguientes:

- **Date**: Fecha del día
- **Location**: Nombre de la ciudad dentro del estado.
- **MinTemp**: Temperatura mínima en el día.
- **MaxTemp**: Temperatura máxima en el día.
- **Rainfall**: Se tomó como la cantidad de lluvia que cayó en el día.
- **WindGustSpeed**: Ráfaga de viento más fuerte del día.
- **Humidity3pm**: Humedad a las 3 de la tarde.
- **Pressure3pm**: Presión atmosférica a las 3 de la tarde.
- **RainToday**: Predicción de lluvia en el día.
- **RainTomorrow**: Predicción de si lloverá al día siguiente.
- Se analizo la posibilidad de quedarse con las columnas “Sunshine” y “Cloud3pm” debido a la correlatividad con la variable target “RainTomorrow” pero el porcentaje de datos nulos era de un 40%.

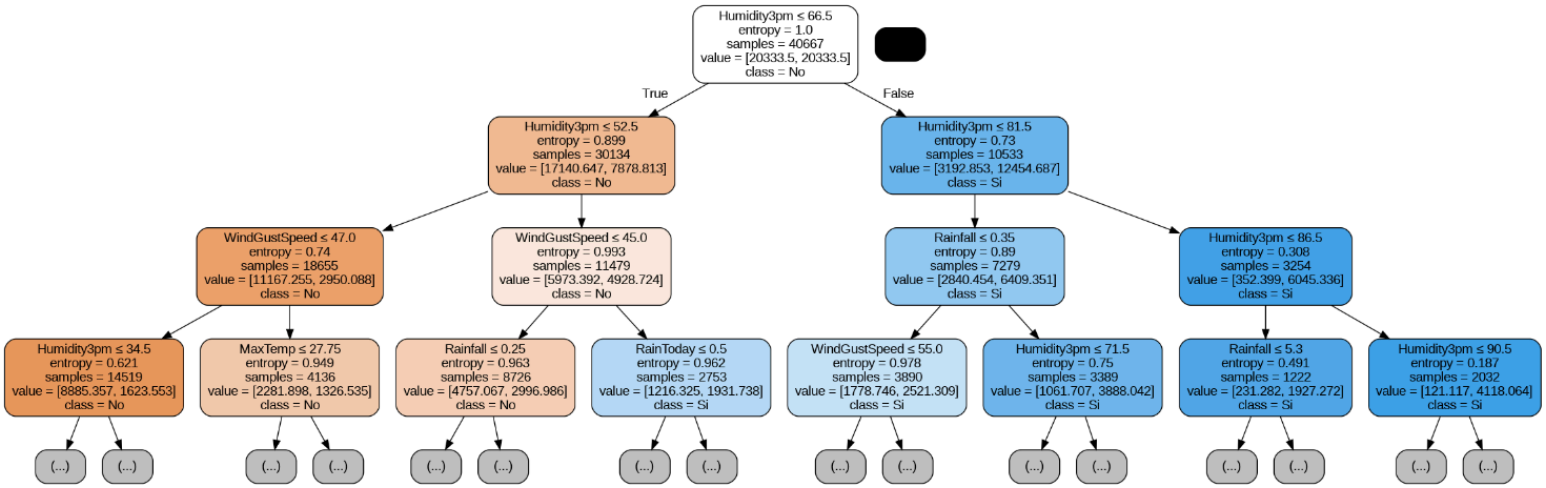
Se realizaron múltiples transformaciones en los datos:

- Se probaron las opciones de eliminar las filas o calcular mediante KKN. Pero finalmente se eliminaron debido a que como son algoritmos que toman decisiones en base a datos, se consideró mejor mantener los mismos lo más representativos a la realidad posible. Igualmente se realizaron pruebas para determinar el mejor curso de acción. Se vio que el score disminuyo, no de manera significativa, cuando se usaba el método de imputación de KKN.
- Se unió el dataset a otro que contenía los estados y las ciudades pertenecientes a los mismos mediante Inner Join para saber a qué estado pertenecían las ciudades y poder quedarnos con las que necesitábamos.
- Se convirtieron las columnas categóricas a cualitativas, que tenían más de una categoría para poder usarlas en los modelos de clasificación mediante el One-Hot Encoding.
- Se probó la idea de normalizar los datos, pero se descarto debido a que sin normalizar los mismos el score aumento casi un 20%.

Modelos

1. Árbol de decisión

- Si, se optimizaron los siguientes hiperparámetros:
    - criterion.
    - min_samples_split.
    - min_samples_leaf.
    - ccp_alpha.
    - max_depth.
    - max_features.
- Si se utilizó el K-fold Cross Validation. Se utilizaron 15 folds para compensar el hecho que de utilizar el randomSearch. Preferimos el randomSearch debido a la amplia cantidad de hiperparámetros que puede comprobar, cuando se trató con el GridSearch el coste computacional fue muy alto y después de 5hs de ejecución el algoritmo seguía sin conseguir un resultado. Para compensar este hecho también se realizó una iteración del algoritmo para finalmente quedarse con el mejor score.
- Utilizamos F1 como métrica debido a que el dataset estaba desbalanceado. Consideramos que usar una métrica que combine tanto la precisión como el recall lograría que la clase minoritaria también se tenga en cuenta en misma proporción que la mayoritaria, ya que el F1 se enfoca en el rendimiento de ambas clases.

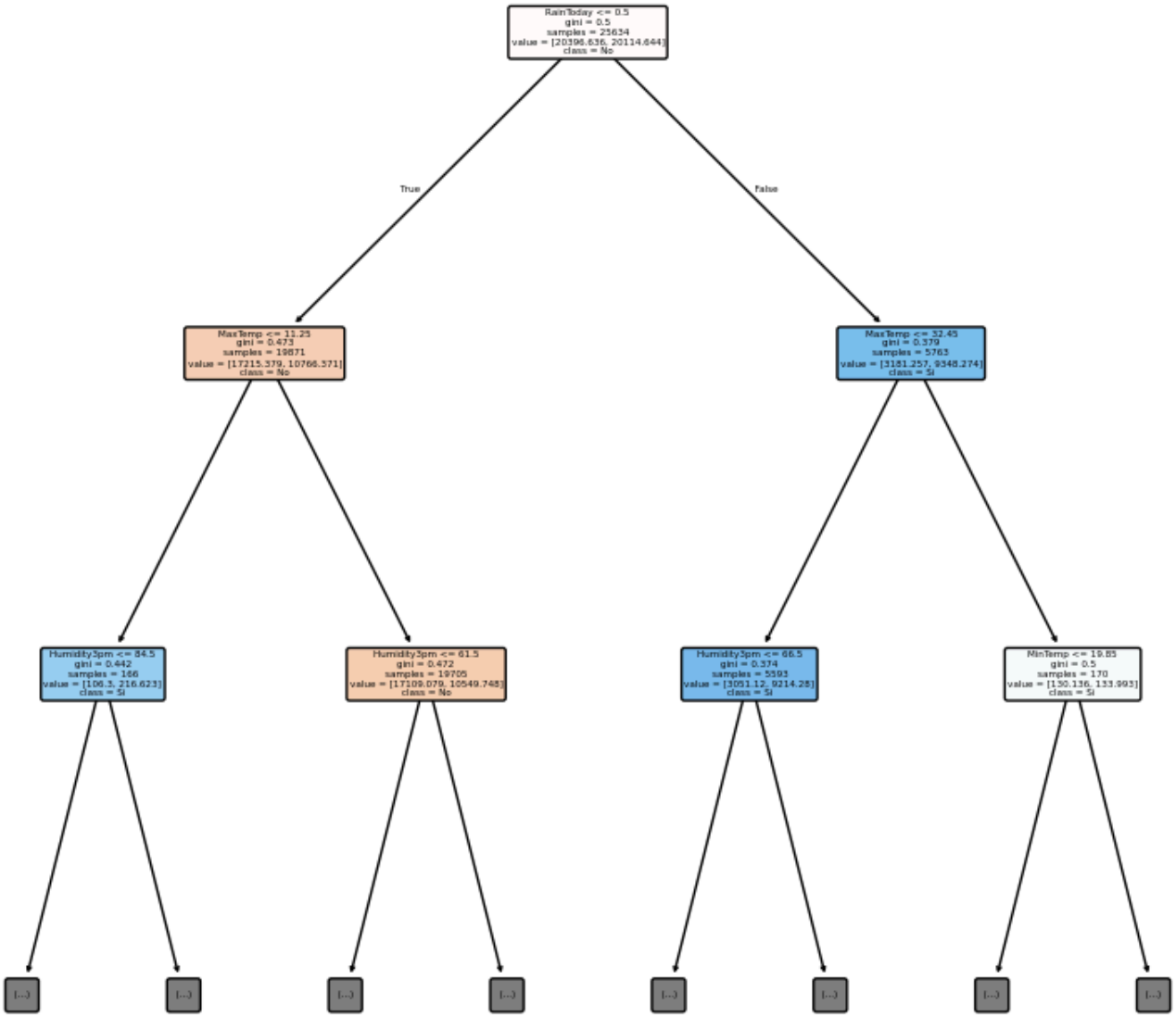


Como se puede ver, las variables que más se tuvieron en cuenta en la realización del árbol de decisión fueron:

- la humedad a las 3pm fue la característica más importante en las decisiones.
- la velocidad de la ráfaga de viento más grande del día.
- La probabilidad de lluvia.
- La cantidad de lluvia que cayó

2. Random Forest

- Si, se optimizaron los siguientes hiperparámetros:
    - criterion.
    - min_samples_split.
    - min_samples_leaf.
    - ccp_alpha.
    - max_depth.
    - n_estimators.
- Si se utilizó el K-fold Cross Validation. Se utilizaron 15 folds para compensar el hecho que de utilizar el randomSearch, no más porque los resultados no variaban mucho después de ese número.
- Utilizamos F1 como métrica para que el modelo sea mejor a la hora de identificar tanto la clase minoritaria como la mayoritaria y que el modelo sea más eficiente.



Reglas:

Si hablamos de la rama izquierda:

- Si la probabilidad de lluvia en el día es nula y la máxima temperatura que hizo en el mismo es menor o igual a 11.25 grados, es probable que al día siguiente no llueva.
    - Hay dos casos posibles después, si la humedad a las 3PM es menor o igual a 34.5 y la temperatura es menor o igual a 11.25 grados, es probable que "RainTomorrow" sea "Si".
    - Sin embargo, si la humedad es menor o igual a 61.5 y la máxima temperatura supera los 11.25° la probabilidad es que "RainTomorrow" sea "No".

Hablando de la rama derecha:

- Si hay una probabilidad de lluvia en el día y la temperatura máxima es menor o igual a 32.45° es probable que al día siguiente llueva.
    - Después si la temperatura máxima es mayor a 32.45° y la temperatura mínima es menor o igual a 19.53° la probabilidad es que "RainTomorrow" sea "Si"
    - Pero si la temperatura máxima es menor o igual a 32.45 grados se tiene en cuenta si la humedad que hizo a las tres de la tarde fue menor a 66.5 para decir que es probable que al día siguiente llueva.

3. SVM: Support Vector Machines

- Si, se optimizaron los siguientes hiperparámetros:
    - kernel
    - C
    - gamma
    - degree
    - coef0

- Si se utilizó el K-fold Cross Validation. Se utilizaron 5 folds para reducir los costes computacionales y el tiempo de entrenamiento, ya que con 10 folds el modelo estaba tardando más de 3hs.

- Utilizamos F1 como métrica debido al desbalance entre clases y las razones mencionadas tanto en el árbol de decisión como en el Random Forest

Cuadro de Resultados

Modelo	F1-Test	Precision Test	Recall Test	Accuracy Test
Árbol de Decisión	0.59	0.52	0.69	0.79
Random Forest	0.60	0.51	0.72	0.78
SVM	1	1	1	1

Respecto al set de entrenamiento

- Árbol de decisión: Fue uno de los que más variación mostró respecto a los resultados de las métricas.
- Random Forest: No se ve mucha variación, el único ligeramente diferente es el recall que en el set de entrenamiento estos dos puntos por arriba.
- SVMt: Mostró los mismos resultados tanto para el sert de entrenamiento como para el de evaluación.

Los modelos utilizados se crearon mediante los siguientes parámetros:

- Árbol de decisión: {'min_samples_split': 15, 'min_samples_leaf': 8, 'max_features': 0.7, 'max_depth': 10, 'criterion': 'entropy', 'ccp_alpha': 0.0}
- Random Forest: {'n_estimators': 30, 'min_samples_split': 11, 'min_samples_leaf': 11, 'max_depth': 7, 'criterion': 'gini', 'ccp_alpha': 0.0}
- SVM: {'kernel': 'linear', 'gamma': 'auto', 'degree': 2, 'coef0': -0.11111111111111116, 'C': 7.857357142857144}

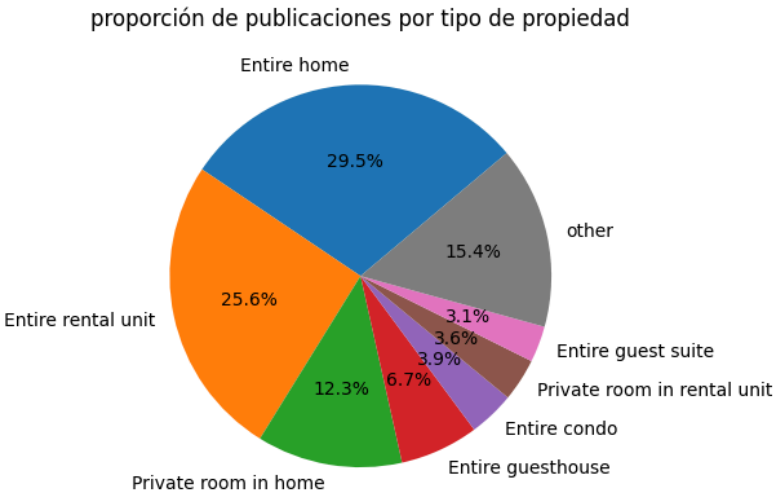
Elección del modelo

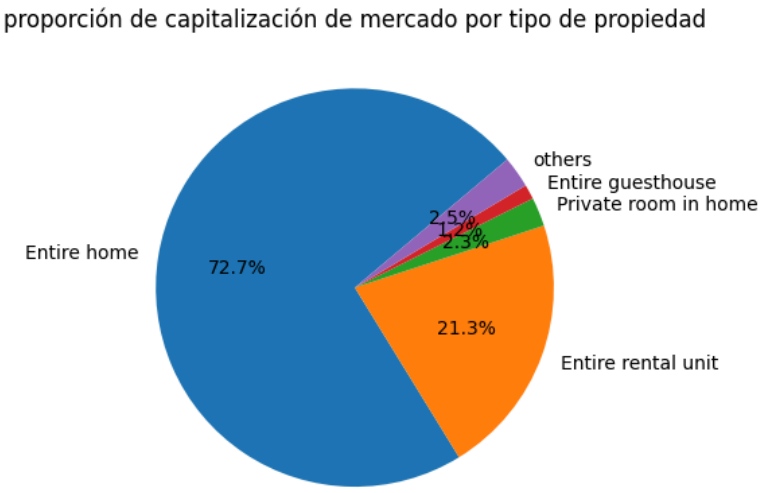
Elegimos el SMV, ya que, fue el que mejores métricas demostró tanto en los datos de entrenamiento como con los de evaluación. Por lo que consideramos que es un mejor modelo de predicción. Teniendo en cuenta las métricas, fue también el que mejor pudo manejar el desbalanceo de clases que estaban en una proporción de casi 70-30, siendo la clase minoritaria “RainTomorrow = Si”

EJ3: Regresión

Este dataset contiene la información de más de 37 mil publicaciones de AirBnB en la región de Los Ángeles, California. Es un dataset extenso con más de 70 columnas, algunas irrelevantes para un análisis de machine learning como el id de cada publicación, la url de la imágen de la publicación, url de imagen del host, fecha en que fueron obtenidos los datos, descripción de la publicación en lenguaje natural, etc. Algunas columnas contienen listas de información (categórica) también demasiado extensas y difíciles de usar con nuestros algoritmos (relativamente) simples de machine learning o con pocos recursos computacionales.

Una columna interesante es el "tipo de propiedad" o "property type". Las publicaciones pueden variar desde casas, hoteles, departamentos y es de esperarse que esté fuertemente correlacionada al precio.





En esto vemos que la gran mayoria de las publicaciones son para alquileres de casas enteras y “unidades en alquiler” (entire home y entire rental unit) pero aún asi el mercado esta dominado por las casa enteras, ya que no solo son muy numerosas sino que los precios de estas suelen ser relativamente altos.

luego de investigar las distintas columnas y descartar todas aquellas que eran completamente irrelevantes para nosotros, descartar todas las filas con la variable precio en falta y las columnas con más del 30% de sus datos en falta, continuamos teniendo demasiadas columnas y filas.

Entrenamos un random forests con valores por defecto con el fin de hacer un análisis simple de las importancias de cada columna y asi elegir un subconjunto relativamente pequeño de features que aun asi mantengan significancia. Asi elegimos las primeras 10 features más importantes para obtener:

- bathrooms: numerica. cantidad de baños
- host_neighbourhood: categorica. nombre de la zona en que vive el host
- neighbourhood_cleansed: categorica. nombre de la zona de la publicación
- longitude : numerica. coordenada de longitud
- host_acceptance_rate: numerica. porcentaje de aceptacion
- latitude: numerica. coordenada latitud
- property_type: categórica. indica que tipo de propiedad es la publicación (casa, hotel, etc)
- bedrooms: numerica. cantidad de habitaciones
- availability_365: numerica. dias al año en que está disponible
- reviews_per_month: numerica. numero de reseñas al mes.

Para imputar, decidimos imputar los datos faltantes por la mediana de la columna, y los datos categóricos por la variante más frecuente. La mediana y no la media ya que tiende a representar mejor la tendencia central y no es tan sensible a datos extremos/outliers.

Para encodear las variables categoricas, usamos one hot encoding (3 features).

Modelos

1. Regresión Lineal
   Usamos las 10 features seleccionadas. Es un modelo simple pero solo produce buenos resultados y generaliza la información si las correlaciones entre los datos y la variable a predecir es lineal.

   Evaluamos el modelo bajo métricas r^2, mean squared error y root mean squared error, contra el conjunto de entrenamiento y contra el conjunto de testeo.
   Entrenamiento:

MSE	RMSE	R^2
238540.2994	488.4058756534051	0.548281298551347

   A simple vista parece razonable, dado que la variación standard en el precio es de ~727. Sin embargo, en testing:

MSE	RMSE	R^2
6.5e+19	8050345017	-145183166089395

   Donde vemos que los valores son inmensos y el R^2 incluse es negativo. Esto implica que no encontramos ninguna correlación real entre las variables y nuestro target.

2. XGBoost

   Usamos un modelo de XGBoost y optimizamos sus hyperparametros mediante random search y k-folds cross validation.

   hyperparámetros optimizados:
   - max_depth
   - n_estimators
   - learning_rate
   - subsample

   Utilizamos 5 folds para k-folds cross validation. En una primera instancia, no encontramos diferencias significativas aumentando el numero de folds a costo de más tiempo de procesamiento, por lo que mantuvimos el estandar de 5.

   La métrica utilizada para la busqueda de hyperparámetros fue la versión negada de MSE, ya que xgboost tiende a funcionar mejor con esta forma de medir el error medio.

   Performance en train:

MSE	RMSE	R^2

9997.1	99.985	0.9811

Performance en test

MSE	RMSE	R^2
121161.8	348.0831	0.7285730

Estos son valores mucho mejores que los de la regresión lineal, y buenos en términos absolutos. Si nos enfocamos en RMSE, podemos ver que el error medio es de $348 cuando la desviación estándar es más del doble. Podemos decir que el modelo logró generalizar considerablemente.

3. Modelo a elección: LightGBM
   Elegimos como tercer modelo Lightgbm. Optimizamos sus hyperparámetros con una búsqueda randomizada y también 5 folds de cross validation.
   Los hyperparametros optimizados fueron:
   - num_leaves
   - learning_rate
   - n_estimators
   - subsample
   - colsample_bytree

   Usando como métrica el error cuadrado medio negativo.

   Performance en train:

MSE	RMSE	R^2
200922.65052243715	448.243963174561	0.6195170416727597

   Performance en test

MSE	RMSE	R^2
186033.40944120946	431.3159044612307	0.5832474924878055

Cuadro de Resultados

Medidas de rendimiento en el conjunto de TEST:

- MSE
- RMSE
- R^2

Confeccionar el siguiente cuadro con esta información:

Modelo	MSE	RMSE	R^2
Regresión Lineal	6.5e+19	8e+9	-1.45e+14
XGBoost	121161.80	348.08	0.729
LightGBM	186033.41	431.32	0.5832

Ambos XGBoost y LightGBM mantuvieron buenos resultados al pasar del set de entrenamiento al set de testeo, pero XGBoost resulto mejor bajo las 3 métricas consideradas.
Nota: indicar brevemente en qué consiste cada modelo de la tabla

Elección del modelo

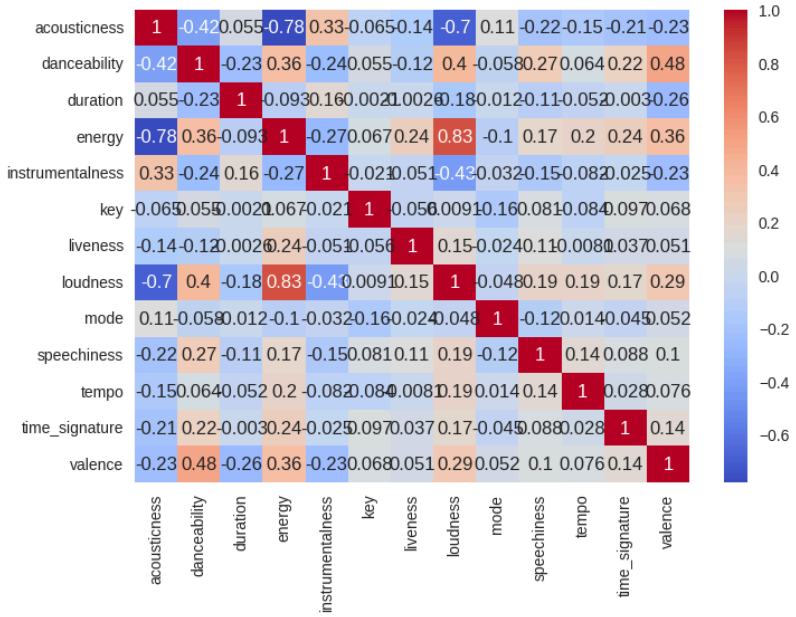
El modelo elegido por nosotros es entonces XGBoost. Logró una buena generalización de los datos para las propiedades en alquiler en los ángeles con variaciones menores a la mitad de la desviación estándar en los datos estudiados sobre la variable de precio.

EJ4: Clustering

Se realizó un análisis utilizando un conjunto de datos de Spotify, destacando las siguientes características clave para la clusterización:

- Danceability
- Energy
- Acousticness
- Valence
- Duration
- Instrumentalness
- loudness
- tempo
- speechiness

Cobertura de diferentes dimensiones: Estas columnas abarcan una variedad de características musicales, como ritmo, energía, acústica, valencia, duración, instrumentalidad, volumen, tempo y habla. Son características relevantes para la agrupación de canciones basadas en estilos musicales o preferencias de escucha, son métricas son útiles para identificar clusters interesantes, permitiendo ubicar las canciones en grupos que reflejan similitudes y patrones significativos.

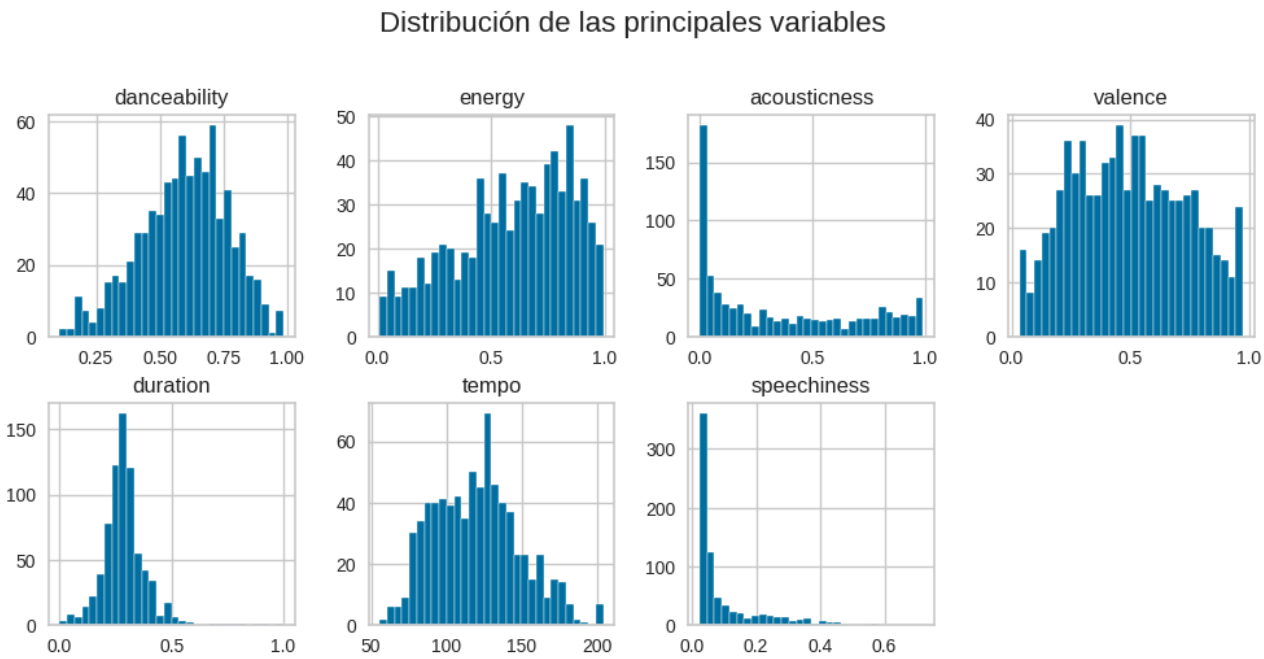


Análisis de la Matriz de Correlación

Observaciones:

- Alta correlación negativa entre acousticness y energy lo indica que las canciones con alta acústica tienden a tener baja energía, y viceversa.
- Alta correlación positiva entre energy y loudness: Las canciones con alta energía suelen tener alto volumen.
- Moderada correlación positiva entre danceability y tempo: Las canciones con alto ritmo tienden a tener un tempo más rápido.

Debido a la alta correlación negativa, entre acousticness y energy consideramos quedarnos con ambas variables lo que nos permitirá capturar variabilidad, por otro lado de energy y loudnes, optamos por quedarnos únicamente con energy, ya que correlacionan bastante bien y esto evitara redundancia.

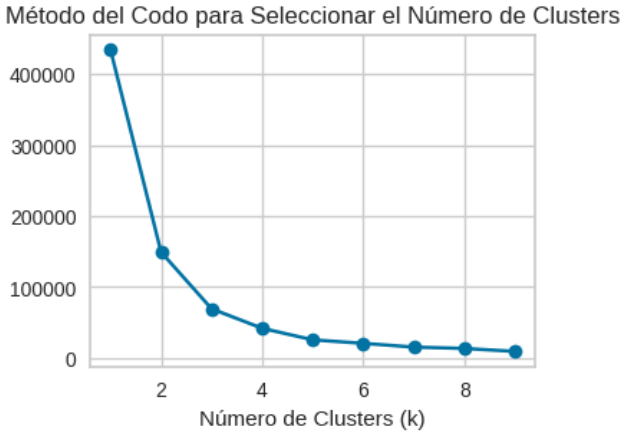


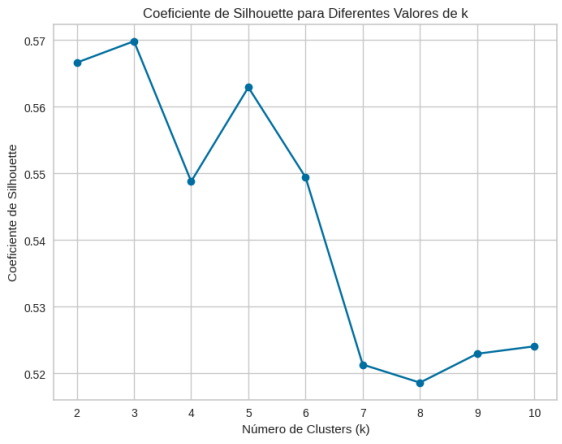
Para speechiness, luego de un análisis estadístico obtenemos como valor máximo de 0.22 quedando claro que la mayoría de las canciones son musicales. Las pistas habladas aparecen solo en casos puntuales, lo cual se evidencia en el tercer cuartil (Q3), donde los valores no superan 0.11. Esto pone de manifiesto que hay una notable cantidad de valores atípicos en este dato, nuestro objetivo es una clusterización de canciones musicales por lo que no vamos a contar con estos outliers.

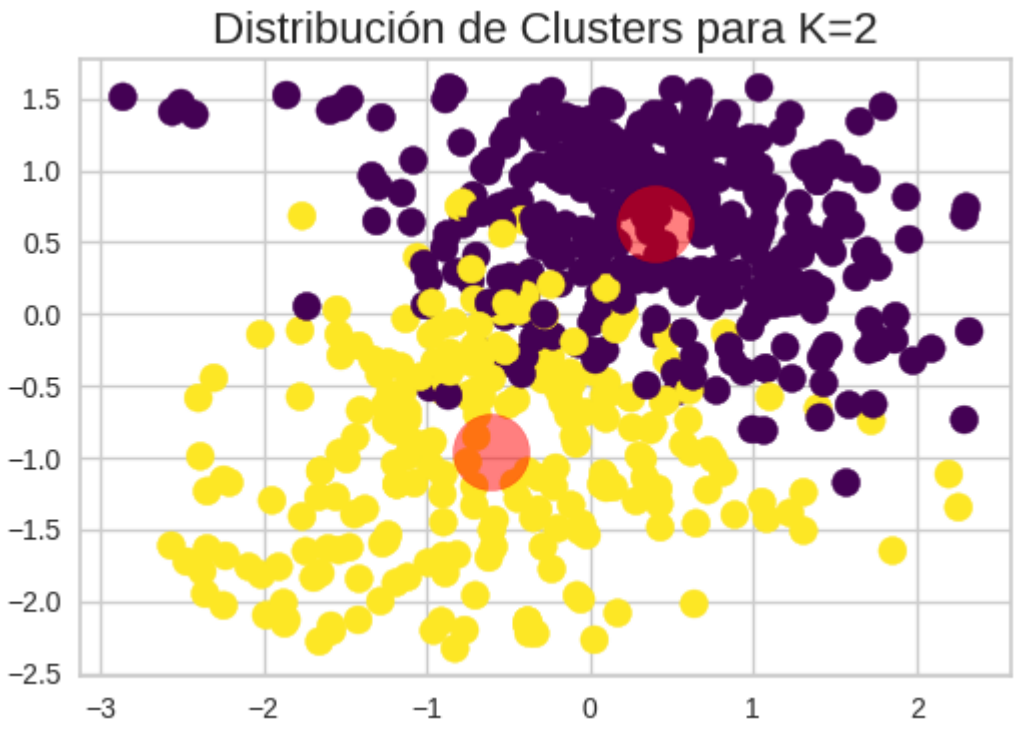
- Acousticness: La mayoría de las canciones no presentan un carácter acústico destacado. Aunque algunas pocas canciones son completamente acústicas, la mayoría tienen niveles bajos o moderados de esta característica.
- Valence: Las canciones presentan una diversidad emocional bastante equilibrada. El conjunto incluye tanto canciones con tonos más tristes como otras más alegres, sin un sesgo claro hacia una emoción predominante.
- Duration: Las duraciones de las canciones varían considerablemente, desde pistas muy cortas hasta algunas más extensas. Aunque la mayoría tiene una duración moderada, hay algunas canciones significativamente más largas que aportan variedad al conjunto.
- Danceability: La capacidad de las canciones para ser bailables es predominantemente moderada a alta. Esto sugiere que el catálogo incluye un buen equilibrio de música que tienen un ritmo favorable para el baile.
- Energy: La energía de las canciones varía, con una leve tendencia hacia niveles moderados y altos. Aunque algunas canciones son muy enérgicas, otras presentan niveles considerablemente más bajos.

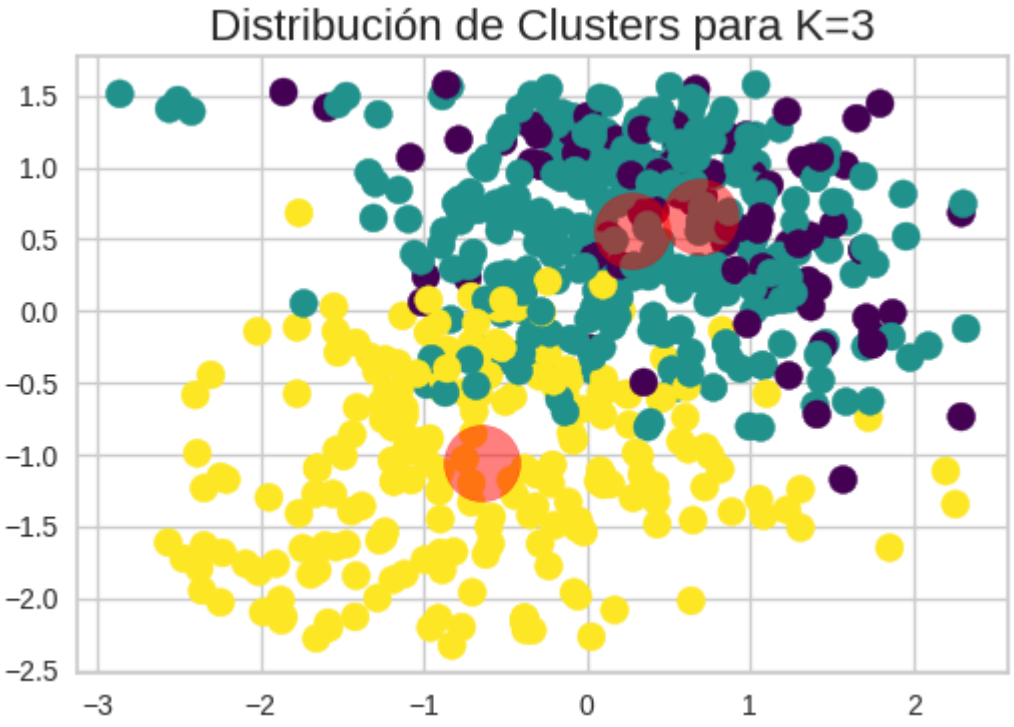
Durante el análisis, identificamos que “duration” presenta una cantidad significativa de valores atípicos. Por sus características particulares, podrían formar parte de un cluster bien diferenciado. Además, debido a la sensibilidad de los datos, realizamos una transformación min-max en la duración de las canciones para que el proceso de clusterización fuera más homogéneo y efectivo.

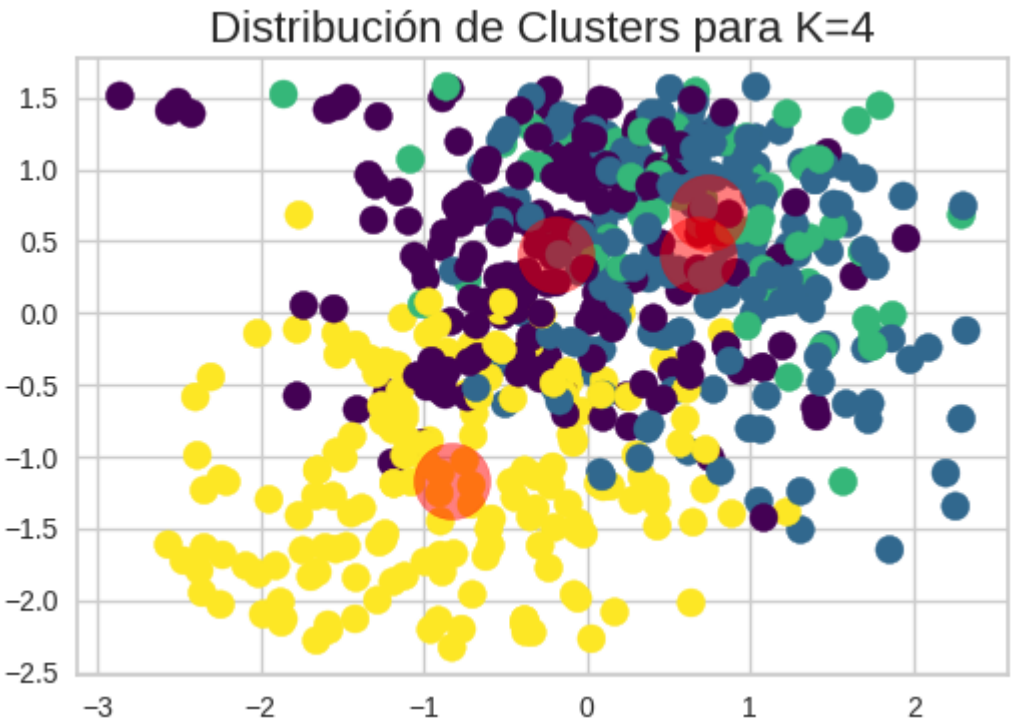
Utilizamos el algoritmo K-means con el soporte de la biblioteca sklearn para determinar el número óptimo de clusters. Para ello, aplicamos dos técnicas: la regla del codo y el coeficiente de silhouette, para identificar el valor adecuado de K. La regla del codo nos permitió visualizar en qué punto la variación dentro de los clusters deja de disminuir significativamente, mientras que el coeficiente de silueta nos indicó qué tan bien definidos y separados estaban los clusters formados.

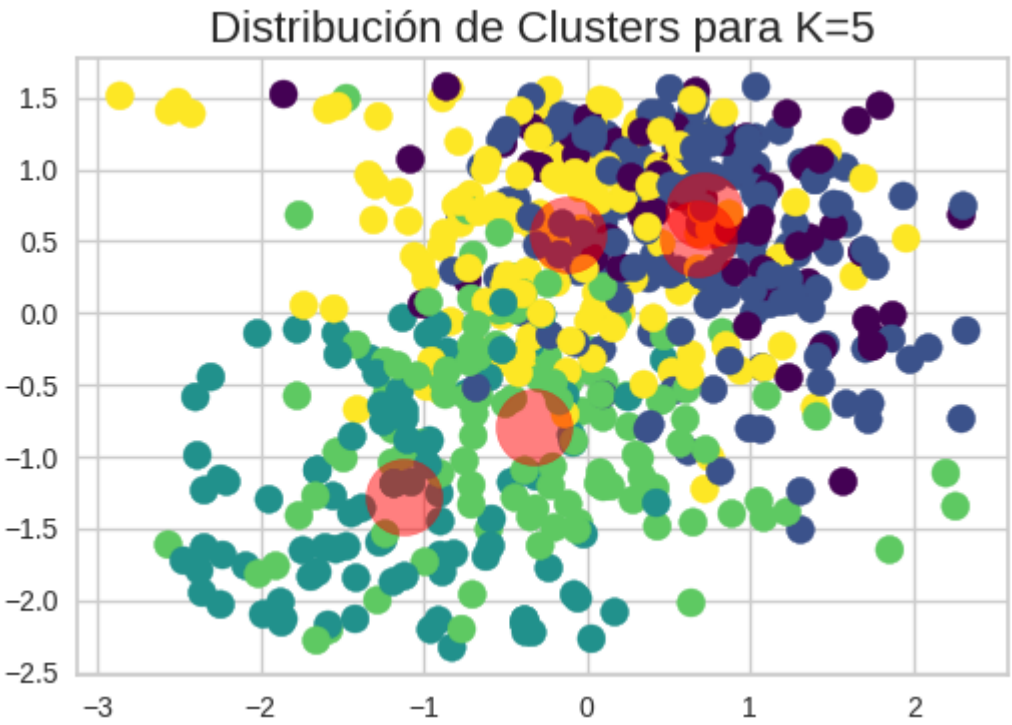





Distribución de Clusters para K=2
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−3
−2
−1
0
1
2


Distribución de Clusters para K=3
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−3
−2
−1
0
1
2


Distribución de Clusters para K=4
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−3
−2
−1
0
1
2



Al analizar los gráficos para diferentes valores de K, se observa que una categoría se clasifica de manera consistente. Sin embargo, al aumentar el número de clusters, la variabilidad de las agrupaciones crece significativamente. Por lo tanto, se considera que K=2 es el número óptimo de clusters para este conjunto de datos.

Los clusters agrupan canciones con características similares. Por ejemplo, un cluster puede reunir canciones enérgicas y poco acústicas, mientras que otro agrupa canciones más tranquilas y acústicas.

CLUSTER	DANCEABILITY	ENERGY	ACOUSTICNESS	VALANCE	DURATION	TEMPO	SPEECHINESS
0	0.657049	0.758081	0.141035	0.582931	0.265095	122.627238	0.077566
1	0.486112	0.357507	0.694011	0.373510	0.273128	113.781727	0.043940

Cluster 0: Bailabilidad (danceability): Tiene un valor promedio relativamente alto (0.657), lo que sugiere que las canciones en este cluster son generalmente más fáciles de bailar.

Cluster 1: Bailabilidad (danceability): Más baja que el Cluster 0 (0.486), lo que indica que las canciones en este cluster son menos bailables.

Cluster 0: Energía (energy): También es alta (0.758), lo que indica que estas canciones tienden a ser más dinámicas y potentes.

Cluster 1: Energía (energy): Mucho más baja (0.357), lo que sugiere que las canciones tienden a ser más suaves, tranquilas o menos intensas.

Cluster 0: Acústica (acousticness): Muy baja (0.141), lo que significa que estas canciones no son predominantemente acústicas, sino que están más orientadas hacia sonidos.

Cluster 1: Acústica (acousticness): Bastante alta (0.694), lo que implica que estas canciones tienden a ser más acústicas, con sonidos más naturales y menos producidos.

Cluster 0: Valencia emocional (valence): Moderadamente alta (0.582), lo que sugiere que las canciones en este cluster tienden a ser más alegres o positivas.

Cluster 1: Valencia emocional (valence): Moderadamente baja (0.373), lo que sugiere que las canciones de este cluster tienden a tener un tono emocional más triste o neutral.

Cluster 0: Duración (duration): Promedio de duración relativamente estándar (0.265), lo que sugiere que las canciones no son ni muy cortas ni muy largas.

Cluster 1: Duración (duration): Similar al Cluster 0 (0.273), lo que indica una duración promedio estándar.

Cluster 0: Tempo: Moderadamente rápido (122.6), lo cual es un ritmo bastante adecuado para canciones bailables.

Cluster 1: Tempo: Un poco más lento (113.7), lo que refuerza la idea de que las canciones de este cluster son más calmadas.

Cluster 0: Habla (speechiness): Muy baja (0.077), lo que indica que hay pocas o ninguna canción en este cluster con predominancia de habla (como podría ser en el rap o el spoken word).

Cluster 1: Habla (speechiness): Muy baja (0.043), incluso más baja que en el Cluster 0, lo que indica que estas canciones tienen aún menos componentes hablados.

El Cluster 0 agrupa canciones más enérgicas, bailables y alegres, con una fuerte producción electrónica o no acústica.

El Cluster 1 contiene canciones más tranquilas, con mayor componente acústico, un tempo más lento y una valencia emocional más baja, sugiriendo una inclinación hacia canciones más suaves o tristes.

Tiempo dedicado

Indicar brevemente en qué tarea trabajó cada integrante del equipo durante estas semanas. Si trabajaron en las mismas tareas lo detallan en cada caso (como en el ejemplo el armado de reporte). Deben indicar el promedio de horas semanales que dedicaron al trabajo práctico.

Integrante	Tarea	Prom. Hs Semana
Cristin Roldan + Roberta Sprenger	Detección de Outliers Armado de Reporte - Ej 1	3 hs
Roberta Sprenger	Reporte - Ej 1	
Cristian Roldan	Análisis de Valores Faltantes Imputación de Datos - Ej 1	1 hs
Cristian Roldan	Analisis exploratprio - Ej 4	3 hs
Cristian Roldan	Manejpo de outlies - Ej 4	0.5 Hs
Cristian Roldan	Clusterizacin - Ej - 4	4 hs
Camila Avalos	Ejercicio 2 - Análisis y pre-procesamiento de datos	2hs
Camila Avalos	Ejercicio 2 - Entrenamiento de modelos de clasificación	8hs
Camila Avalos	Ejercicio 2 - Armado del reporte	1hs

Tomás Benitez Potochek	Ejercicio 1 - Visualizaciones geográficas sobre viajes en Manhattan	1 hs
Tomás Benitez Potochek	Ejercicio 3 - Análisis preliminar del set, procesamiento de columnas, datos faltantes, feature selection, visualizaciones	5 hs
Tomás Benitez Potochek	Ejercicio 3 - Imputacion y encodeo de datos. Entrenamiento y busqueda de hyperparametros.	4 hs