Organización de Datos - Ciencia de Datos
(9558 - TA047) Curso 02 - Rodriguez

Trabajo Práctico 1

2º Cuatrimestre 2024
Grupo Nº 1

Integrantes:

Beltrán Malbrán	**110036**
Florencia Dellisola	**109897**
Gian Luca Spagnolo	**108072**
Marcela Jazmín Cruz	**110066**

Índice

Ejercicio 1: Análisis Exploratorio

Se nos asignó trabajar con el conjunto de datos sobre el uso de taxis *Yellow Cab en USA* [1] específicamente con los meses de *Enero*, *Febrero* y *Marzo*. Al unificarlos en un dataset se obtuvo que el mismo cuenta con un total de 9.384.487 registros y 20 columnas. A continuación se muestran las columnas encontradas:

- `VendorID`: código que indica quién proporcionó el registro.
- `Tpep_pickup_datetime`: fecha y hora de inicialización del viaje.
- `Tpep_dropoff_datetime`: fecha y hora de finalización del viaje.
- `Passenger_count`: cantidad de pasajeros (ingresado por el conductor).
- `Trip_distance`: distancia del viaje en millas (reportado por el taxímetro).
- `PULocationID`: zona donde inicia el viaje.
- `DOLocationID`: zona donde finaliza el viaje.
- `RateCodeID`: corresponde al tipo de tarifa del viaje, definido al finalizar.
    - 1 = Viaje estándar
    - 2 = JFK
    - 3 = Newark
    - 4 = Nassau or Westchester
    - 5 = Tarifa negociada
    - 6 = Viaje en grupo
- `Store_and_fwd_flag`: indica si el viaje fue almacenado para ser reportado al finalizar la jornada (en caso de `True`), o si el viaje fue reportado al instante (en caso de `False`)
- `Payment_type`: código numérico, representa el tipo de pago utilizado.
    - 1 = Tarjeta de Crédito
    - 2 = Efectivo
    - 3 = Sin Cargo
    - 4 = Disputado
    - 5 = Desconocido
    - 6 = Viaje desechado
- `Fare_amount`: la tarifa por tiempo y distancia calculada por el taxímetro.
- `Extra`: extras y recargos varios. Actualmente, esto solo incluye los cargos de $0,50 y $1 por hora pico y durante la noche.
- `MTA_tax`: impuesto MTA de $0.50 que es cargado automáticamente en base a la tasa de uso medida.
- `Improvement_surcharge`: precio base de un viaje, en $0.30 que se realiza al momento de bajar la bandera. Esta sobrecarga comenzó a regir en 2015.

- `Tip_amount`: este campo corresponde a la propina realizada mediante tarjetas de crédito. No se incluyen las propinas en efectivo.
- `Tolls_amount`: importe total de todos los peajes pagados en el viaje.
- `Total_amount`: importe total cobrado a los pasajeros (no incluye propinas en efectivo).
- `Congestion_Surcharge`: monto total cobrado en el viaje por el recargo por congestión del Estado de Nueva York.
- `Airport_fee`: costo por comienzo de viaje en aeropuerto, de $1.25. Esto aplica únicamente para los siguientes aeropuertos: LaGuardia y John F. Kennedy.

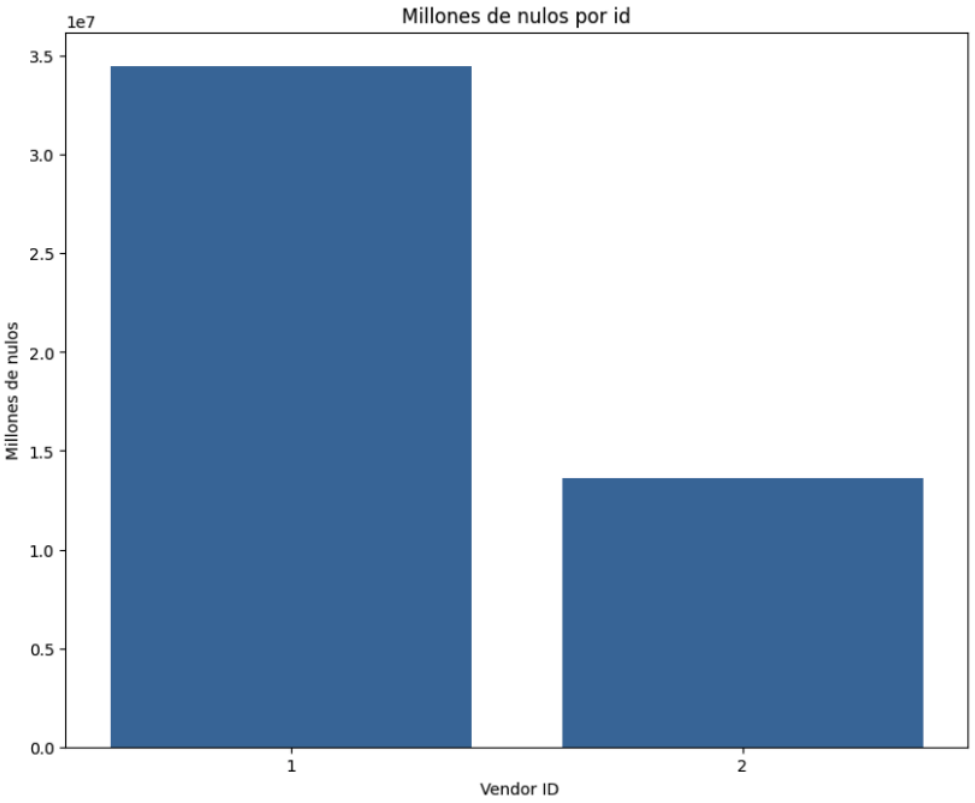
Se ha decidido, para trabajar de forma más eficiente con las variables a disposición, establecer una diferenciación entre aquellas *variables cualitativas* y *variables cuantitativas*, separándolas en dataframes distintos para manejarlas específicamente.

Limpieza de Datos

El preprocesamiento de los datos inició con la identificación de valores nulos en las columnas presentes. Se ha logrado identificar un total de 1.180.895 valores nulos los cuales corresponden al 2.5% de cada una de las siguientes columnas:

- `Passenger_count`
- `RatecodeID`
- `Store_and_fwd_flag`
- `Congestion_surcharge`
- `Airport_fee`

Dentro de este grupo se observó que el `VendorID` con valor 6 posee todos los datos con valores nulos, por lo cual se decidió eliminar los registros con dicho ID. Dentro de los IDs restantes, se logró identificar que aquellos con valor 1 presentan más registros nulos, tal como se puede observar en la *Figura 1*.



[Figura 1]

Por otro lado, en aquellos valores con `VendorID` igual a 6 se han identificado una totalidad de valores nulos para las features destacadas anteriormente, los cuales corresponden a una cantidad de 3.129 registros que hemos eliminado ya que, debido al tamaño del dataset, se consideran redundantes.

Para el resto de valores nulos, se ha decidido completarlos con el valor correspondiente a la mediana. Dentro de este reemplazo hay un caso que se trató de forma diferente, los registros nulos de la columna `store_and_fwd_flag` fueron reemplazaron por el valor *Desconocido*. Esta decisión se tomó al tener en cuenta que no se puede representar el 50% de una variable categórica.

Luego de realizar la limpieza de valores nulos se realizó el manejo de los datos mal ingresados. Cabe aclarar que se consideran como datos “mal ingresados” a los que no pertenecen a una de las categorías predefinidas arriba, o a datos con valores negativos. Dentro de este grupo se han identificado más de 40.000 registros con valores incorrectos en la columna de `RatecodeID`. Estos tenían como tipo de tarifa el número 99, debido a que corresponden a una cantidad inferior al 1% de todas las entradas. A estos registros y al resto con datos mal ingresados, se tomó la decisión de no considerarlos para el análisis.

Generación de Nuevas Features

En primer lugar se generaron nuevas columnas a partir de la fecha, decidimos analizar los días, horas, meses y duración de viaje por separado. Esto se hizo teniendo en cuenta sólo los datos de partida, ya que nuestro análisis va a estar enfocado en las zonas y fechas que más taxis se solicitaron. Respecto a la duración del viaje, se decidió categorizar en tres tipos posibles: *cortos*, *promedio* y *largos*. Considerando esta categorización se realizaron tres más: primero se clasificó la cantidad de pasajeros a bordo en *pocos, normal* y *muchos*. Luego la hora de partida se categorizó según su franja horaria, los días se agruparon y se consideraron por quincena en vez de individualmente. También se decidió trabajar complementariamente con un dataset externo [1] el cual nos brinda la información sobre qué *distrito*, *región* y *zona de servicio* pertenece cada `ID` de ubicación, ya sea de partida o fin de viaje. La distribución de estos datos se encuentra en la sección de **Visualización de los datos** de la notebook correspondiente a este ejercicio.

Teniendo en cuenta la creación de estas features se decidió eliminar las siguientes columnas: `tpep_pickup_datetime`, `diferencia_viaje`, `duracion_viaje`, `tpep_dropoff_datetime`, ya que por separado nos permiten visualizar de forma más clara y concisa las posibles relaciones entre la información dada.

Correlación Entre los Datos

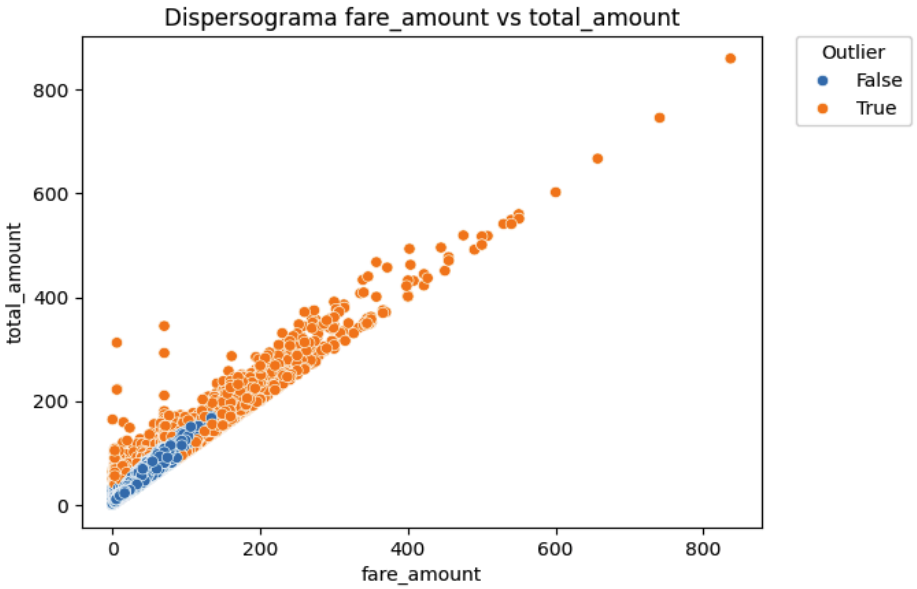
En principio, se ha descubierto que las *variables cualitativas* no presentan relación alguna entre sí, mientras que las *cuantitativas* si. Las features con mayor relación son:

- `fare_amount` con `total_amount` (0.98 de índice de correlación)
- `tip_amount` con `total_amount` (0.72 de índice de correlación)
- `tolls_amount` con `total_amount` (0.71 de índice de correlación)
- `tip_amount` con `tolls_amount` (0.48 de índice de correlación)

Análisis de Valores Atípicos

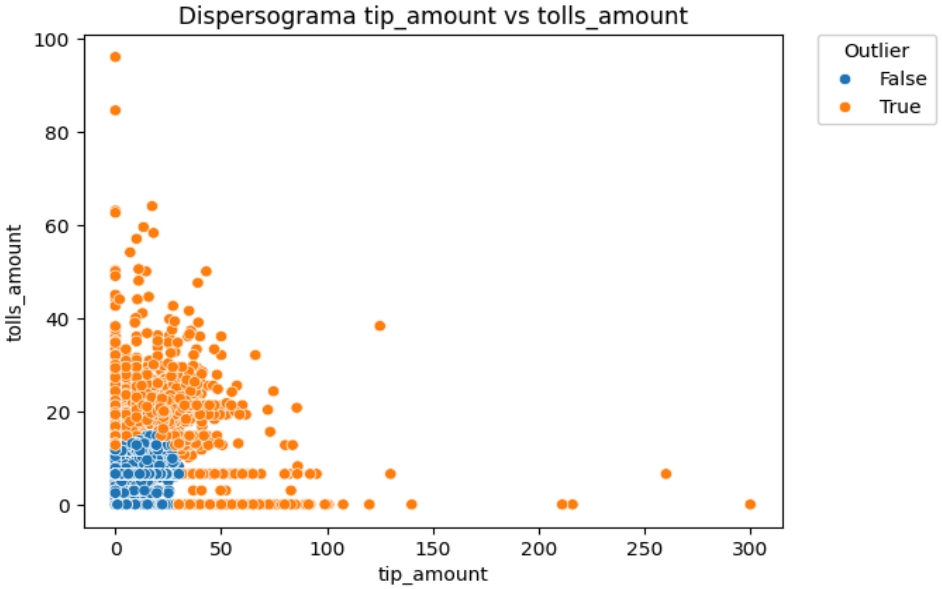
Por motivos de simplicidad para dicho análisis solamente se ha considerado el 10% del dataset, sabiendo que no se modifica la interpretación visual de los datos.

Considerando que `fare_amount` y `total_amount` es la correlación positiva más predominante en el dataset decidimos hacer un respectivo análisis de los valores atípicos.



[Figura 2]

También se realizó un análisis similar con `tip_amount` con `tolls_amount` ya que presentan el menor coeficiente de correlación.



[Figura 3]

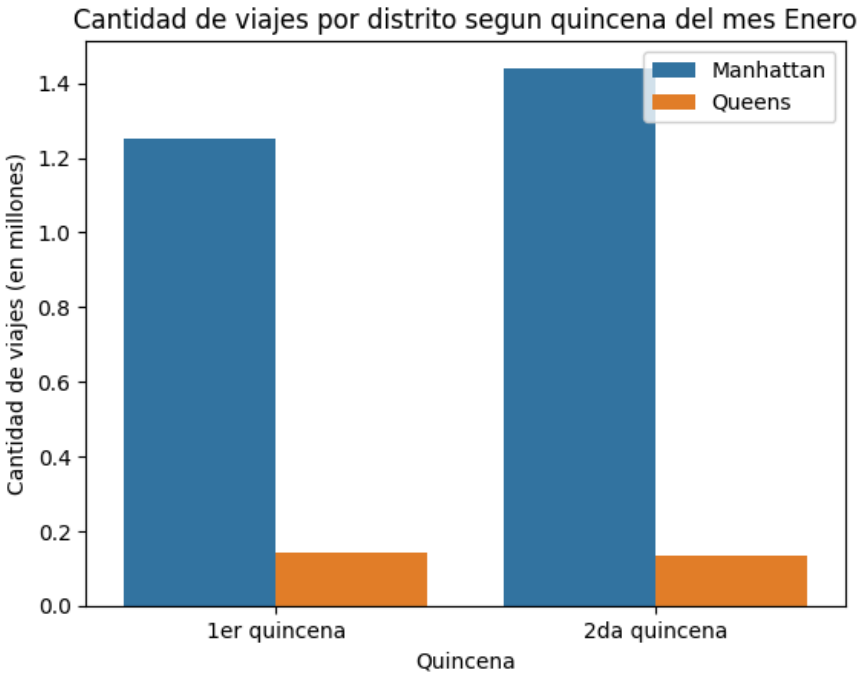
Estos análisis se realizaron al buscar la distancia de *Mahalanobis* [2] la cual busca la similitud entre cada una de las relaciones. Usamos este cálculo para la visualización de los valores atípicos entre ambas variables, en la *Figura 2* y *Figura 3* se evidencia esta relación.

En complemento al análisis multivariado realizado se hizo un análisis para cada una de las features, este se hizo al tomar como atípicos a todos los valores fuera del rango intercuartílico, es decir los que no pertenecen al rango comprendido entre el 25% y el 75% de los valores posibles.

Preguntas de Investigación y Visualizaciones

1. ¿Qué distritos solicitaron la mayor cantidad de viajes por quincena?

Decidimos investigar la relación de estas dos features ya que nos permite tener un mayor entendimiento de la distribución de taxis en cada uno de los distritos. Con el fin de ver el flujo de taxis en los distritos más solicitados, decidimos quedarnos con los dos distritos más solicitados de cada quincena.

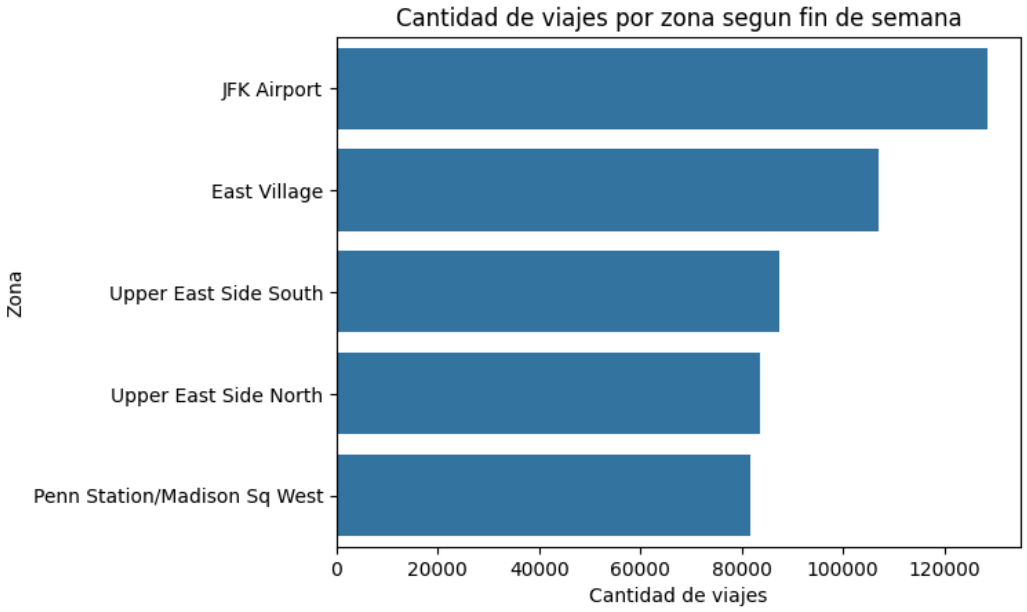


[Figura 4]

Este análisis nos permite observar que Manhattan es el distrito con mayor cantidad de viajes en ambas quincenas por una amplia diferencia, este comportamiento pasa indiferentemente del mes, como se puede ver en la *Figura 4*, lo cual es congruente con lo esperado ya que Manhattan es el distrito con mayor población de los analizados.

2. ¿Cuál es la zona con mayor cantidad de viajes en el fin de semana?

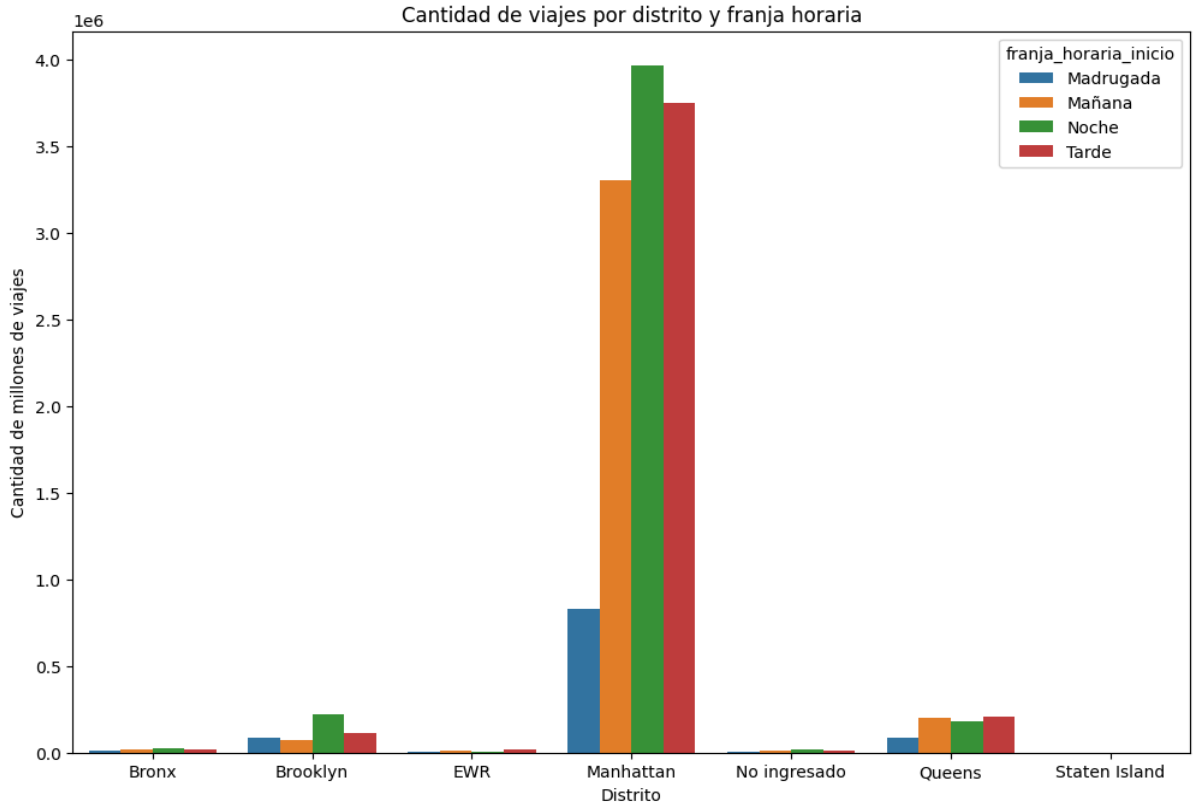
Teniendo en cuenta los resultados obtenidos de la primera hipótesis planteada, decidimos analizar desde qué zonas se solicitaron más viajes en los días no hábiles, esto nos permite ver si el uso de taxis está relacionado con algún tipo de actividades laborales o de ocio. Para lograr visualizar las zonas con mayor cantidad de taxis solicitados se tomó la decisión de quedarnos con las primeras cinco zonas. Tal como se puede ver en la *Figura 5* una zona del distrito de Queens (JFK Airport) es la que más viajes solicitó, seguido de una zona perteneciente al distrito de Manhattan (East Village). Esto nos permite concluir que, si bien Manhattan es el distrito con mayor pedido de taxis, posiblemente se deba a actividades laborales.



[Figura 5]

3. ¿Cuál es la cantidad de viajes en cada franja horaria, en cada distrito?

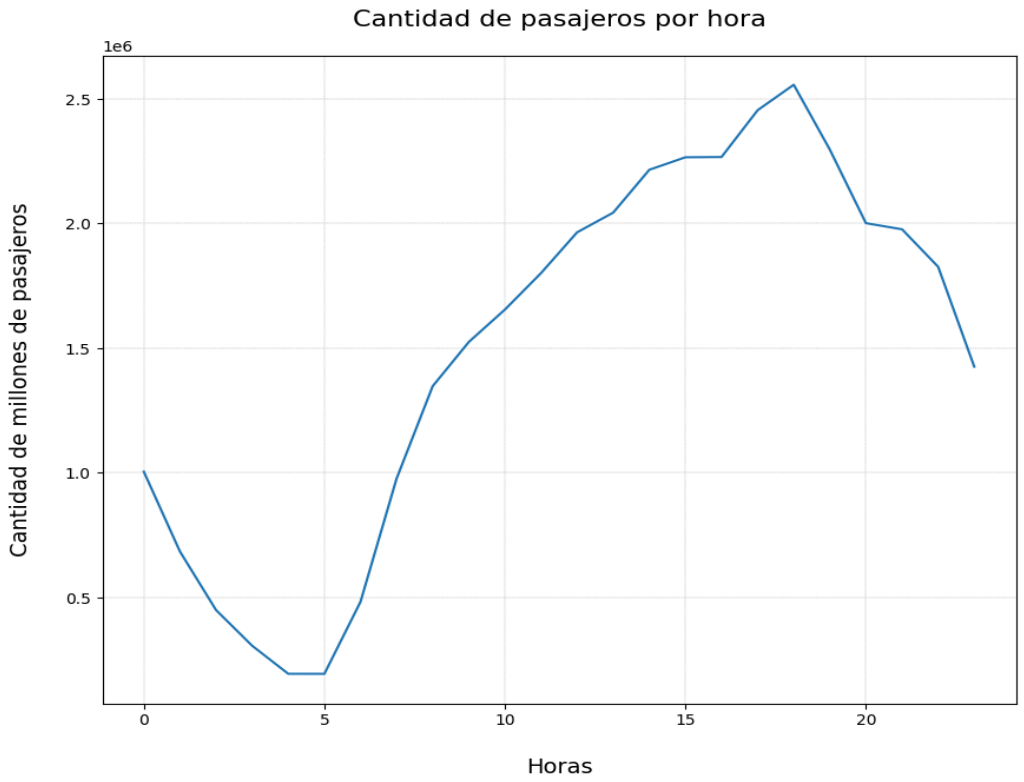
Considerando los resultados obtenidos en las anteriores investigaciones, analizamos en qué franja horaria se solicitaron más viajes en cada uno de los distritos. Esto nos permite lograr comprender el flujo de los taxis y lograr afirmar si el consumo de estos está relacionado con el ámbito laboral o no. La *Figura 6* nos permite deducir que, si bien hay una fuerte demanda en horarios relacionados al fin del dia laboral, considerando la finalización del dia laboral entre las horas de la tarde y noche, la demanda no está asociada exclusivamente a esto ya que hay una considerable cantidad de taxis solicitados en las horas de la mañana.



[Figura 6]

4. ¿Cuál es la distribución de la cantidad de pasajeros en un día?

En este análisis buscamos observar cual es el comportamiento general de la cantidad de viajes iniciados para así lograr identificar si existen horas con mayor cantidad de inicios de viajes o si es un promedio constante, lo que también nos permite identificar si la mayor cantidad de viajes ocurren cerca del fin del día laboral o no, corroborando la información obtenida anteriormente. Como se evidencia en la *Figura 7*, la mayor cantidad de viajes sucede entre las 15 y las 20 horas, fortaleciendo aún más la hipótesis de la relación del consumo de taxis con actividades laborales y/o académicas.



[Figura 7]

5. ¿Se deja más propina, en promedio, en los días hábiles o los fines de semana?

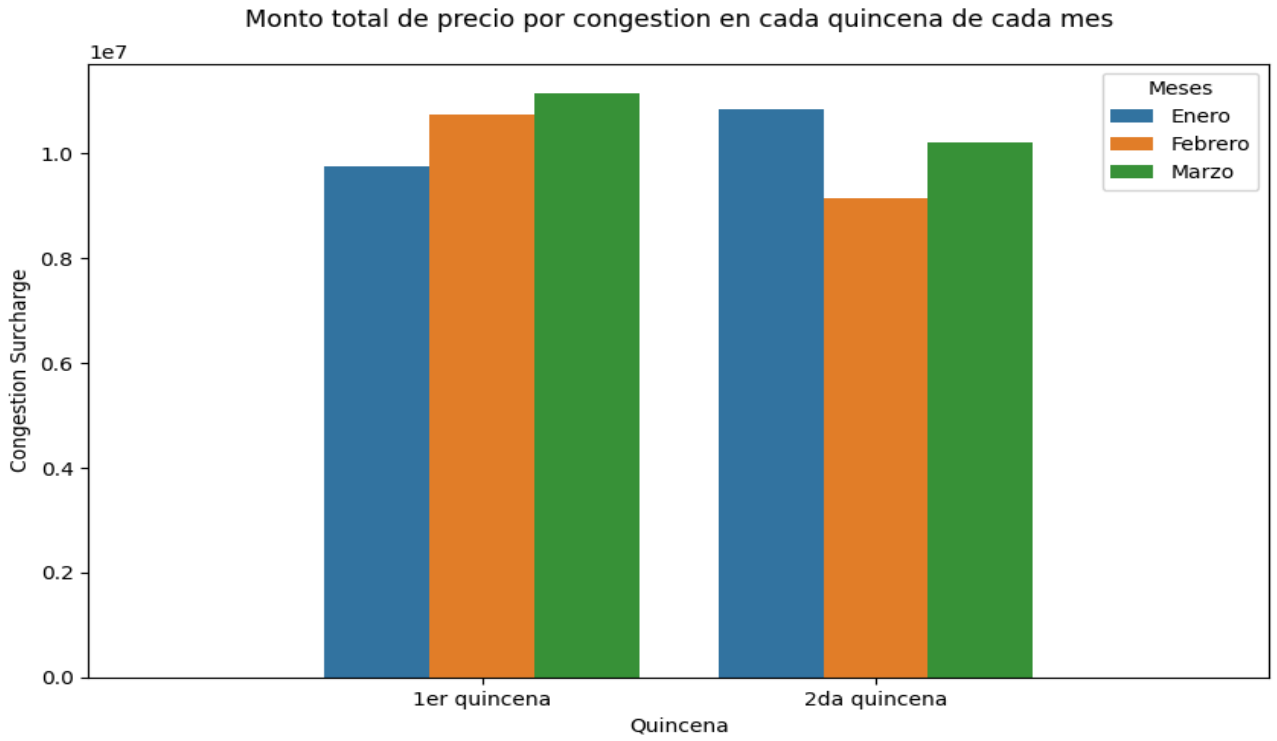
[Figura 8]

En esta investigación decidimos analizar la relación del promedio de propina con los días hábiles, para poder visualizar si en los fines de semana hay un incremento en la

cantidad de propinas. En este análisis, viendo la *Figura 8*, se puede deducir que es indiferente si se trata de un día hábil o no, analizando en base a la cantidad de propinas.

6. ¿Cuál es la quincena de cada mes en la que hay más congestión?

En este análisis se busca visualizar cuál quincena en cada mes presenta mayor demanda de taxis. La *Figura 9* nos permite visualizar qué febrero tuvo una gran bajada en cuanto a la congestión, mientras que enero ha visto un incremento pronunciado.

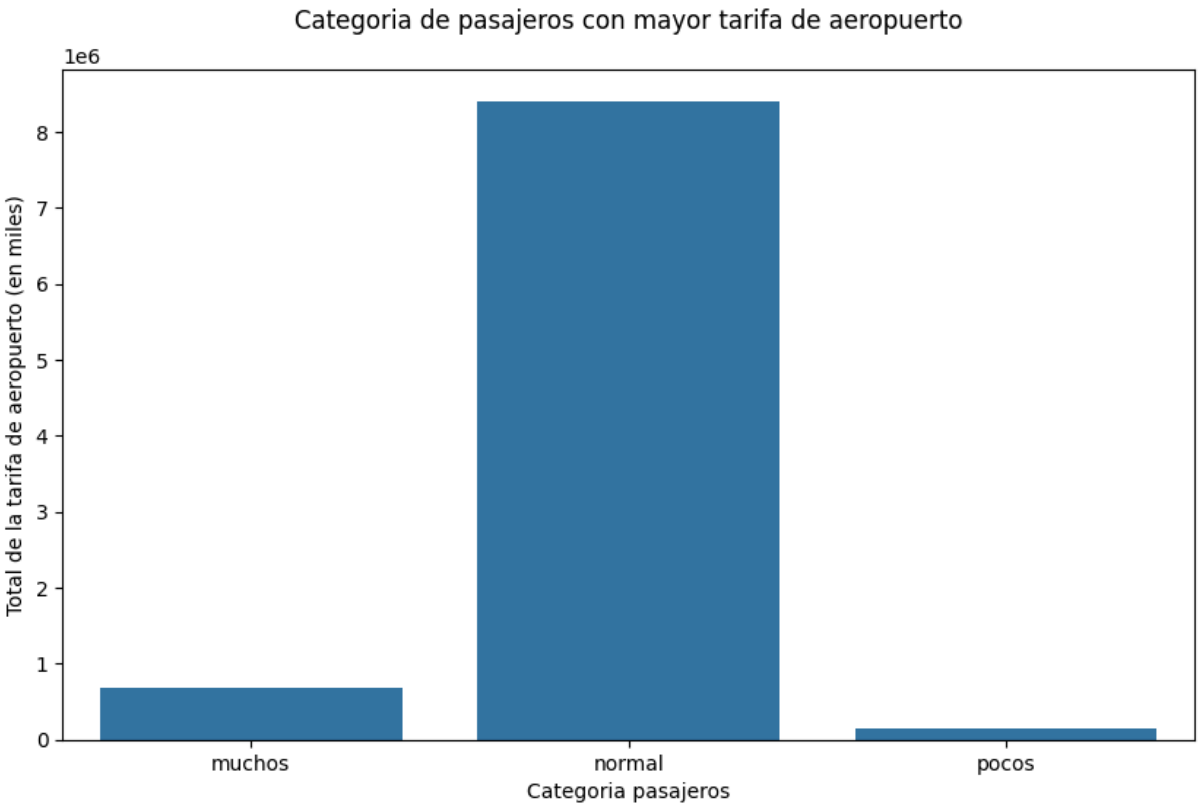


[Figura 9]

7. ¿Cuántos pasajeros suelen abordar un taxi cuando parte del aeropuerto?

Este análisis se hace con el fin de lograr ver cual es la cantidad de pasajeros que deciden subirse a un taxi cuando provienen del aeropuerto. La *Figura 10* evidencia que suele subirse una cantidad promedio de pasajeros, por lo que podemos asumir que no varía la cantidad de pasajeros que abordan el taxi cuando se pide desde un aeropuerto que cuando no.

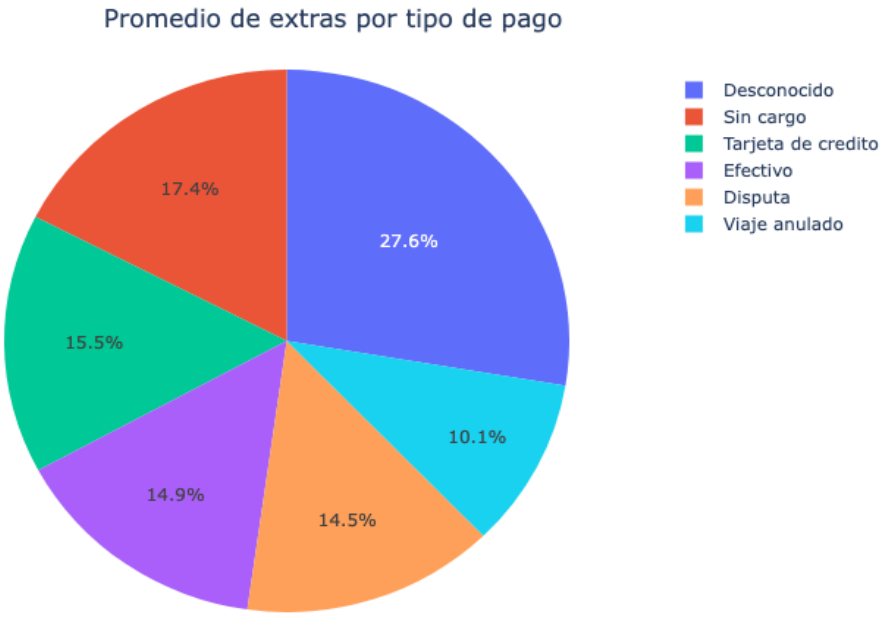
Se recuerda que la diferenciación entre *pocos*, *promedio* y *muchos* pasajeros está realizada en la sección de **Visualización de los datos** de la notebook correspondiente a este ejercicio.



[Figura 10]

8. ¿Cuál es el promedio de `extra` según el tipo de pago?

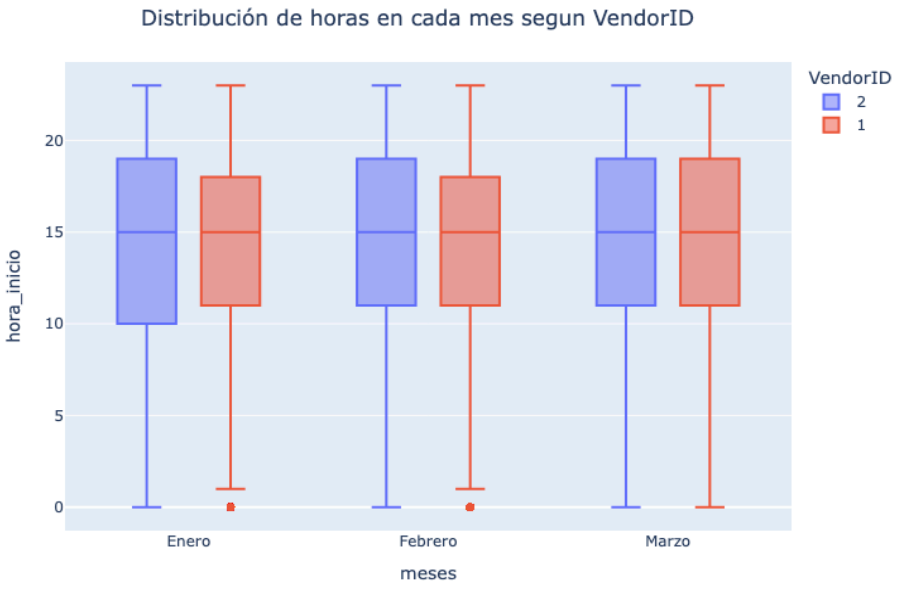
Dado que se considera como un `extra` a todo tipo de recargo, ya sea por hora pico y/o por solicitar el viaje de noche, buscamos saber qué medio de pago suele preferir las personas que toman la decisión de pagar dicho extra, en caso de que deban de pagarlo. La *Figura 11* nos permite visualizar que, de los cargos conocidos, cuando se aplica un recargo los pasajeros prefieren pagar con tarjeta de crédito.



[Figura 11]

9. ¿En qué horas presenta más viajes cada `VendorID` y en qué mes?

Este análisis se realizó con el fin de saber la distribución de `VendorID` en cada hora y ver si varía mes a mes. La *Figura 12* nos permite concluir que el `VendorID` 2 suele tener más inicios de viajes en distintas horas que el `VendorID` 1, con la excepción del mes marzo, en el cual ambos presentan una distribución muy pareja.



[Figura 12]

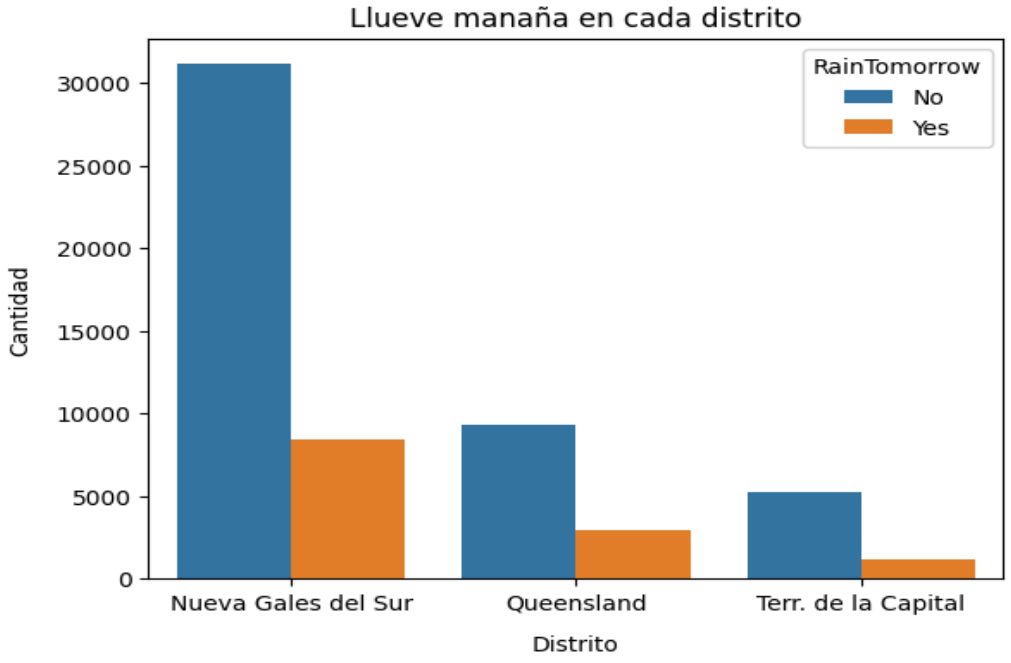
Ejercicio 2: Clasificación

Se nos asignó trabajar con datos de estaciones meteorológicas de Australia. Específicamente con los distritos de *Queensland*, *Nueva Gales del Sur* y *Terr. de la Capital*, con el objetivo de predecir si lloverá o no al día siguiente [3].

Nos encontramos con un dataset de 58.357 registros y 23 columnas, al realizar el preprocesamiento se encontró que el mismo no contaba con datos duplicados, pero sí con datos nulos, dado la gran variabilidad de nulos en cada columna, yendo del 0.86% al 60.85%, se optó por imputarlos por el valor más frecuente según el caso de cada columna.

Modificaciones Realizadas

Entre las 23 columnas se encontró la columna `Date`, se decidió reemplazarla por las nuevas columnas `día`, `mes`, `año`, además se agregó la columna `estación` para indicar en qué temporada del año se obtuvo ese registro. Como el dataset cuenta con la columna `Location`, pero no indica a qué distrito pertenece esa zona, decidimos agregar la columna `distrito`. Luego de generar estas nuevas columnas decidimos ver el comportamiento de la variable a predecir en cada distrito, tal como se muestra en la *Figura 13*. Esto nos indica que en el distrito de Nueva Gales del Sur es en el que mayor cantidad de veces ocurrió este evento.



[Figura 13]

Para la predicción pedida se realizó un encoding con *one hot encoder* de las columnas categóricas, es decir `Location`, `WindGustDir`, `WindDir9am`, `WindDir3pm`, `RainToday`, `Estación`, `Distrito` y el target `Rain_Tomorrow`. Luego se separó un 80% de los datos para el entrenamiento y el 20% restante para el testeo.

Modelos

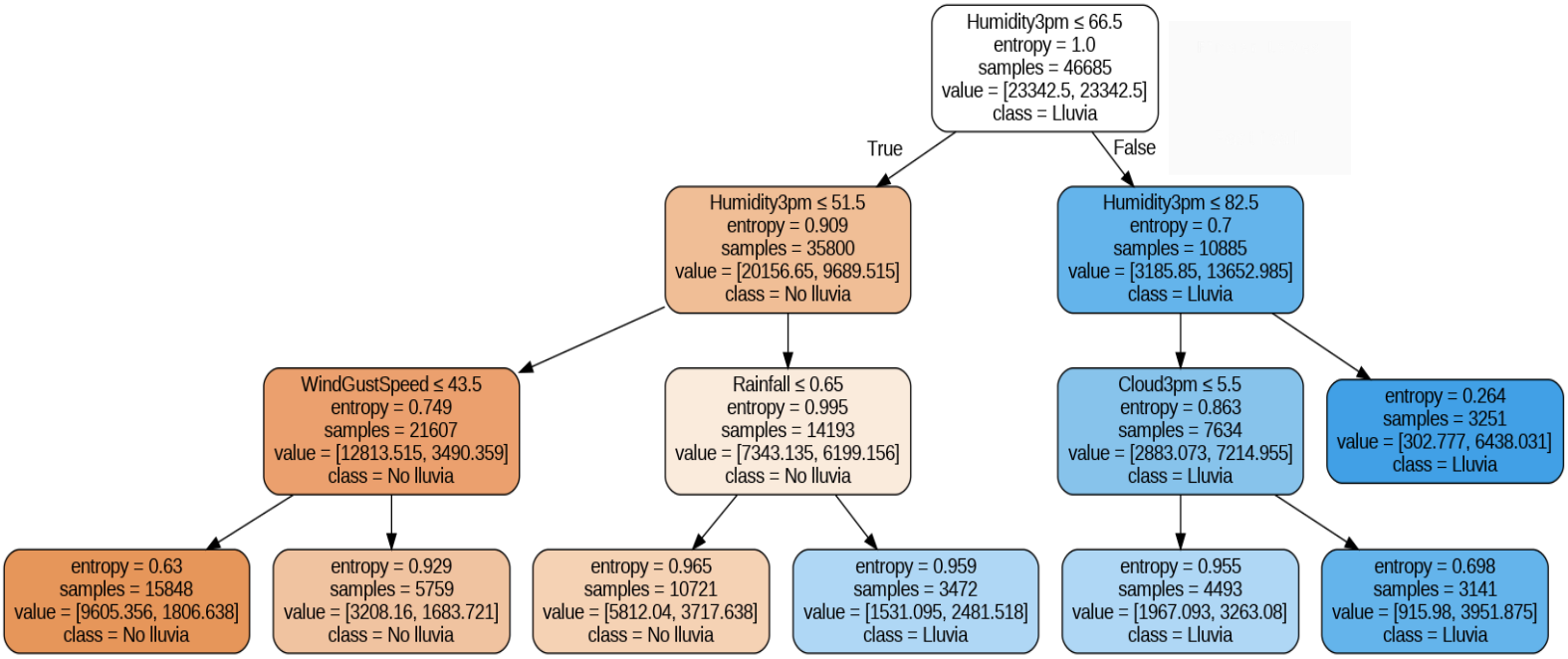
1. Arbol de Decision

Para este modelo se utilizó **K-fold Cross Validation** [4] para, en primer lugar encontrar la cantidad de fold más adecuada, la cual resultó ser 10, mientras que para optimizar los hiperparametros se utilizó **Random Search Cross Validation** con los folds obtenidos anteriormente y aplicando la métrica **F1**. Esta métrica fue elegida ya que permite trabajar de forma consistente con clases desbalanceadas, lo cual se ve evidenciado en la *Figura 13*, y nos permite lograr obtener una media entre el recall y la precisión. Los hiperparametros optimizados fueron:

- `min_sample_split`
- `min_samples_leaf`
- `max_depth`
- `criterion`
- `ccp_alpha`

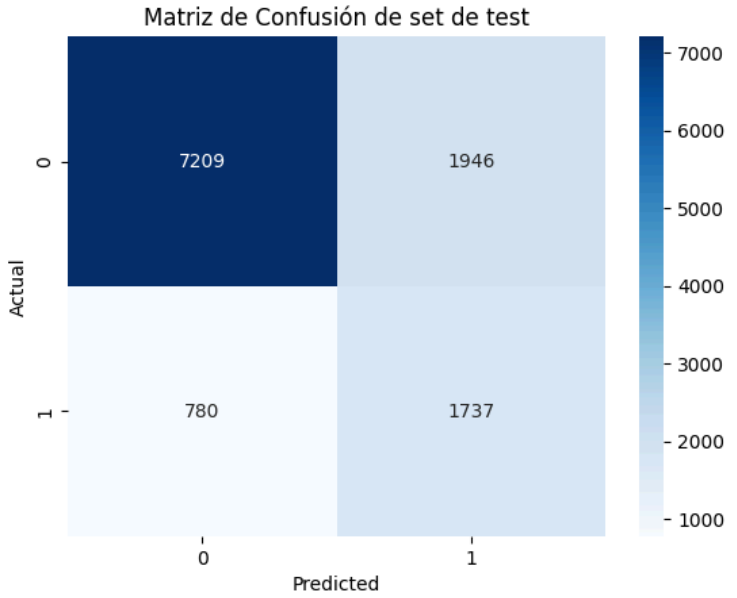
A partir de esta optimización se obtuvieron el conjunto de reglas con la cual se puede generar el árbol de decisiones. Estas reglas nos permiten predecir si va a llover mañana o no. En primer lugar, nuestro árbol compara los índices de humedad, en base a este resultado verifica la cantidad de viento o de nubes y las precipitaciones presentes. Todo esto queda evidenciado en la *Figura 14*. El arbol en cuestion quedo conformado en la siguiente manera:

```
{'min_samples_split': 7, 'min_samples_leaf': 8, 'max_depth': 3,
'criterion': 'entropy', 'ccp_alpha': 0.005555555555555556}
```

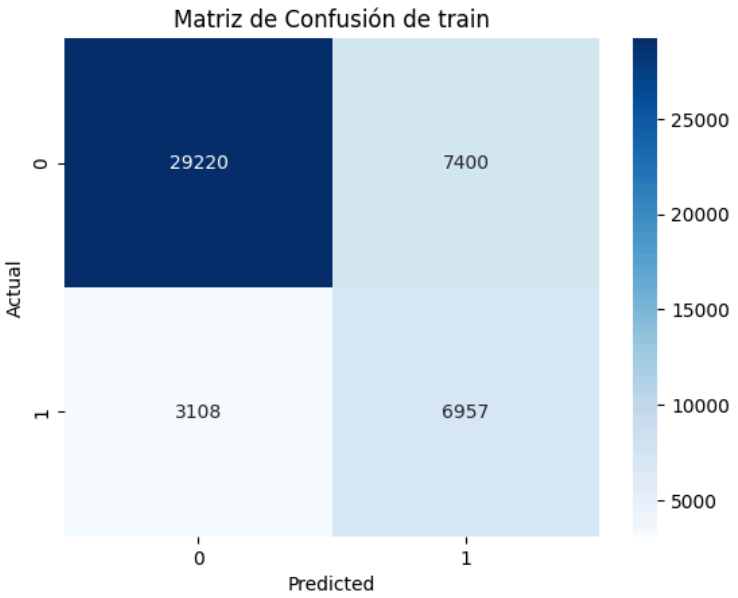


[Figura 14]

La performance de este árbol en el conjunto de test, mostrada en la *Figura 15*, nos permite ver que la cantidad de predicciones correctas es buena. Esto es consistente con los datos obtenidos en el conjunto de train, mostrada en la *Figura 16*, ya que en ambas matrices de confusión presentan un buen valor de recall, sin embargo en el set de train hay una ligera mejora en la métrica f1.



[Figura 15]



[Figura 16]

2. Random Forest

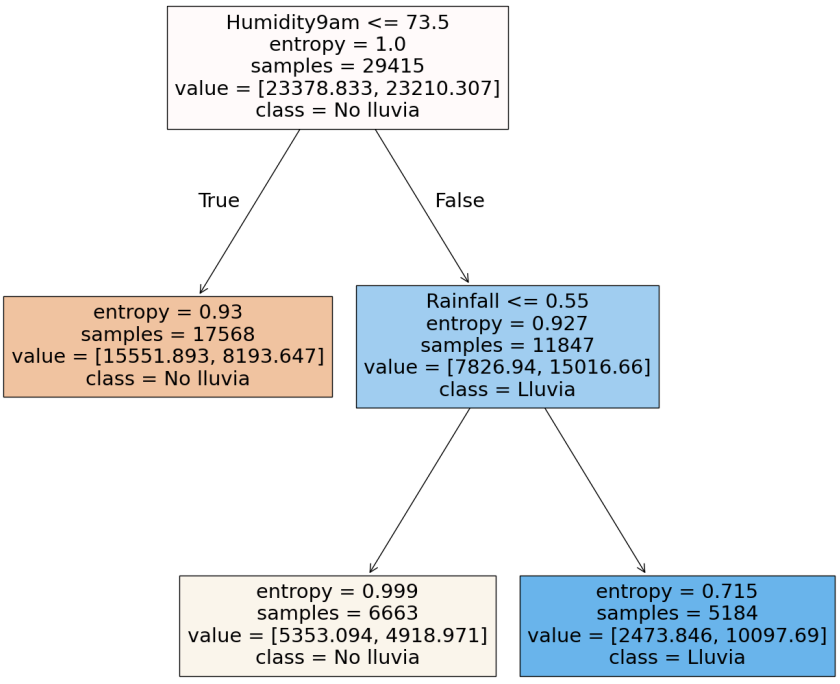
Para este modelo se utilizó **Random Search Cross Validation** con 10 folds y aplicando la métrica **F1 Weighted**, la cual se decidió balancear por lo evidenciado en la *Figura 13*. Los hiperparametros optimizados fueron:

- `min_sample_split`
- `min_samples_leaf`
- `max_depth`
- `criterion`
- `ccp_alpha`
- `n_estimators`

A partir de esta optimización se obtuvo el modelo final, este modelo se desempeña de forma similar en los sets de test y train, con la diferencia de que en este último presenta una ligera mejora en el recall. El modelo final quedó conformado por de la siguiente manera:

```
{'n_estimators': 40, 'min_samples_split': 2, 'min_samples_leaf': 10, 'max_depth': 2, 'criterion': 'entropy', 'ccp_alpha': 0.02222222
```

Un ejemplo de uno de estos árboles es la *Figura 17*, donde en este se compara los índices de humedad y en base a ese resultado verifica si hay precipitaciones.



[Figura 17]

3. Regresión Logística

Para este modelo se utilizó **Score Cross Validation** con 10 folds y aplicando la métrica **F1 Weighted**, lo que nos permite comparar los distintos puntajes y lograr una estimación más confiable. Cabe aclarar que se realizó un escalado de los valores con el fin de mantener una distribución menos dispersa de los datos.

Este modelo en el set de test logró ser bastante preciso para los casos en que el evento analizado no ocurre, donde también presenta un recall del 80%. Para los casos en los que sí ocurre, presenta un recall del 75%. Para el set de *train*, el modelo presenta una precisión y f1 ligeramente más altas aunque mantiene el recall.

Cuadro de Resultados

Modelo	F1-Test	Precision Test	Recall Test	Accuracy Test
Arbol de Decision	0.78	0.81	0.77	0.77
Random Forest	0.78	0.81	0.76	0.76
Regresión Logística	0.80	0.83	0.79	0.79

Elección del Modelo

En base a los resultados obtenidos elegimos el modelo de **Regresión Logística** ya que presenta un mayor balance entre la precisión y el recall, evidenciado por su alto valor en la métrica **F1**.

Ejercicio 3: Regresión

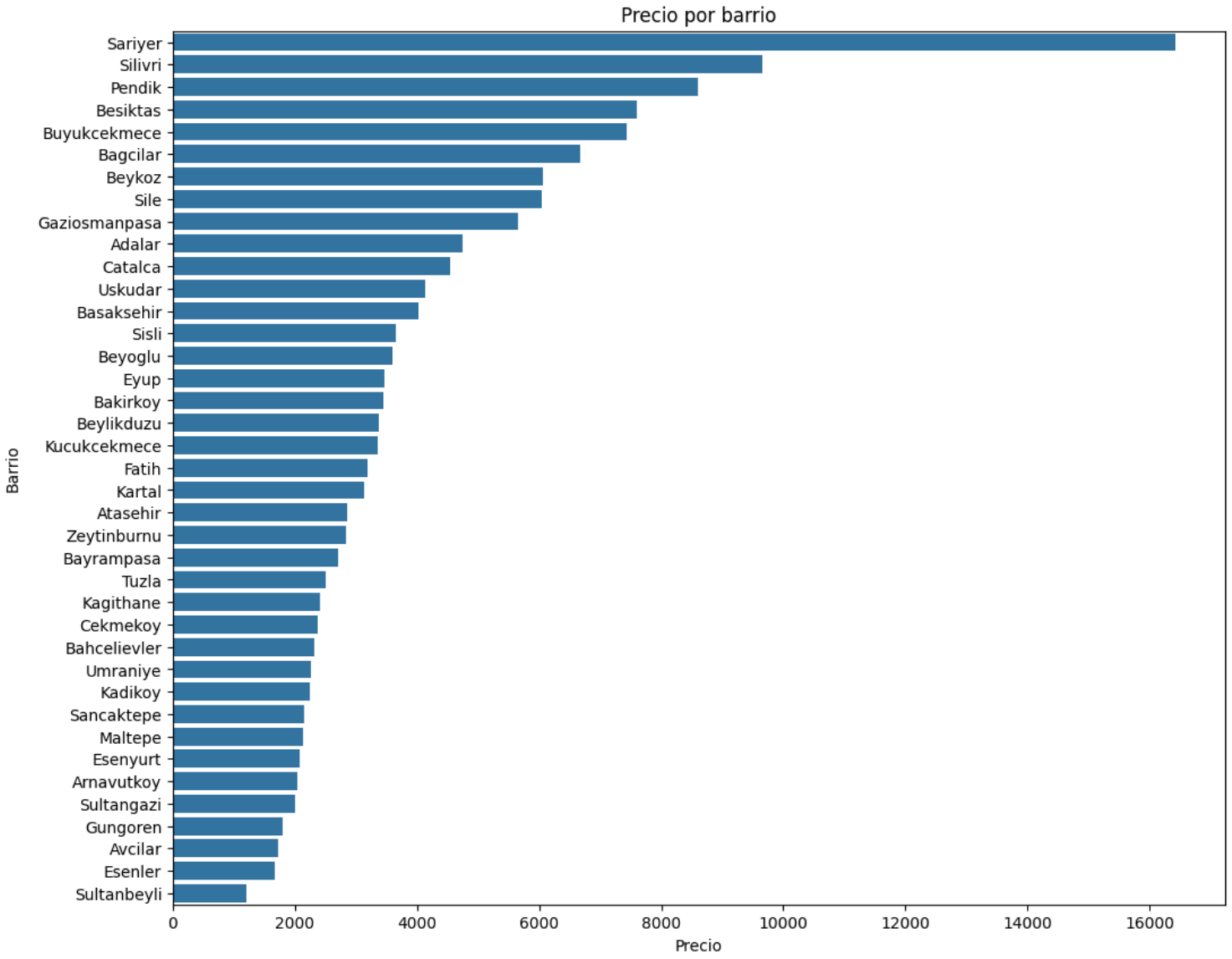
En este ejercicio debemos trabajar con datos de alojamientos de la plataforma *AirBnB*, específicamente los datos provenientes de *Estambul* [5].

Procesamiento de Datos

Nos encontramos con un dataset de 31.758 registros y 18 columnas. En primer lugar, verificamos si presentaba datos duplicados y el porcentaje de datos nulos. Este no presentaba duplicados pero si valores nulos, en particular la columna `neighbourhood_group` la cual tiene un 100% de valores nulos, por lo que se decidió no tenerla en cuenta para el análisis. El resto de datos en las columnas con valores nulos, se imputó con la mediana en los casos de valores numéricos, para los casos de las fechas se decidió usar una interpolación, es decir estimamos el valor correspondiente para esa columna teniendo en cuenta los valores existentes, si luego de realizar esto quedan valores nulos con este tipo de dato se reemplazan con el siguiente valor no nulo en la columna. Por último, a los valores nulos con otros tipos de datos, se los relleno con la categoría *desconocido*.

Luego de realizar la limpieza del dataset, se generaron nuevas columnas a partir de la fecha de la última reseñas, es decir de la columna `last_review`, esta información se descompuso en tres, `mes_last_review`, `year last review` y `dia_last_review`. A partir de esto observamos la fecha de las últimas reseñas y decidimos que, si la última reseña de un lugar fue hace más de tres años de la última actualización del dataset (la cual fue el 30 de junio de 2024), no se va a tener en cuenta para el análisis, esta decisión se tomó ya que aproximadamente 30.000 de los registros pertenecen al periodo comprendido por los años 2021, 2022, 2023 y 2024, siendo este último el que mayor cantidad de últimas reseñas posee. Por lo explicado anteriormente se eliminó la columna `last_review` ya que su información se fragmentó en las columnas previamente mencionadas.

Para lograr un mayor entendimiento del problema a analizar, decidimos ver cómo varían los precios según el barrio donde se encuentran. Este análisis, el cual se puede observar en la *Figura 18*, nos permite concluir que el precio por barrio es más caro en Sariyer, el cual se ubica al oeste del país, sin embargo los altos precios no son exclusivos de esta zona ya que tanto el centro como el este presentan altos niveles de precios también.



[Figura 18]

También se analizó qué tipo de alojamiento alquilado posee más reseñas por mes, lo cual nos permite entender el patrón de consumo y que prefieren los clientes. Las conclusiones que sacamos de este análisis fueron que los departamentos y/o casas en su totalidad presentan un alto nivel de reseñas en promedio, seguido por cuartos privados y habitaciones de hoteles. En último lugar están los cuartos compartidos.

Análisis de Valores Atípicos

A través del uso del **Z Score**, se pudo identificar aquellos valores atípicos para cada una de las features que se disponen. Esto fue realizado tomando como valores atípicos a todos aquellos mayores a 2 o menores a -2 del **Z Score** de cada una de las columnas. Luego de realizar la detección de estos outliers, se decidió imputar logarítmicamente, con el fin de normalizar la distribución de los datos.

Modelos

Antes de empezar a describir los distintos modelos de predicción utilizados hay que tener en cuenta que para poder evaluar de forma precisa y concisa cada uno de los barrios y licencias nos quedamos con los primeros 10 de cada uno.

1. Regresión Lineal

Para construir este modelo no se tuvieron en cuenta las columnas de `host_name`, `name`, `host_id` e `id`. Esta selección de features se realizó con el fin de armar el modelo solo con columnas que puedan llegar a inferir en la variable a predecir. Luego de hacer la separación de los sets de train y test se decidió encodear los datos con **one hot encoder** y luego normalizarlos con el método **StandardScaler**. Por último se realizó una validación cruzada entre los puntajes, con el fin de tener una estimación más certera. La performance en el conjunto de entrenamiento fue la siguiente:

- MSE: 0.48
- RMSE: 0.70
- R2: 0.27

Mientras que, en el conjunto de prueba, la performance fue la siguiente:

- MSE: 0.45
- RMSE: 0.67
- R2: 0.30

2. XGBoost

Para este modelo se decidió hacer la búsqueda de los hiperparámetros más óptimos a través de **Grid Search** con 5 folds y con la métrica de la raíz cuadrática media. Esto dio como resultado los siguientes hiperparametros: `alpha=10`, `colsample_bytree=0.8`, `learning_rate=0.1`, `max_depth=20`,`min_child_weight=5`, `n_estimators=200`.

La performance de este modelo en el conjunto de entrenamiento resultó óptima, con un 0.73 de valoración para la métrica R2 la cual está acompañada de un 0.18 en MSE y un

0.42 en RMSE. Mientras que en el conjunto de pruebas también se muestran resultados positivos, con un 0.50 de valoración para la métrica R2 la cual está acompañada de un 0.32 en MSE y un 0.57 en RMSE.

3. LightGBM

En este modelo se utilizó **Randomized Search** para la búsqueda de hiperparametros. Este modelo utiliza 5 folds y la métrica *raíz cuadrática media*, lo cual concluyó en que los hiperparametros más óptimos para este modelo son: `max_depth=30`, `min_child_weight=10`, `n_estimators=200`.

La performance de este modelo en el conjunto de entrenamiento presenta las siguientes características.

- RMSE = 0.51
- MSE = 0.26
- R2 = 0.60

Mientras que la performance en el conjunto de prueba queda definida por los siguientes puntajes

- RMSE = 0.57
- MSE = 0.32
- R2 = 0.50

Cuadro de Resultados

Modelo	**MSE**	**RMSE**	**R2**
Regresión Lineal	0.54	0.45	0.30
XGBoost	0.32	0.57	0.50
LightGBM	0.32	0.57	0.50

Elección del Modelo

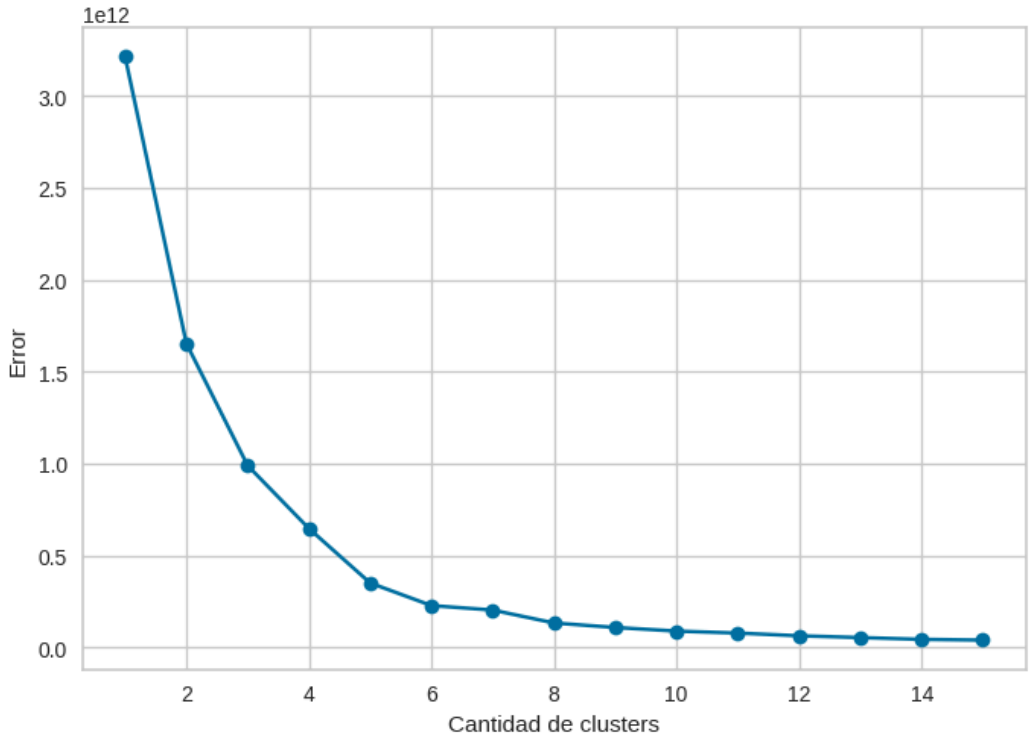
En base a los resultados obtenidos el modelo que elegimos para predecir el precio de una propiedad de *AirBnB* en *Estambul* es **XGBoost** ya que, como se indica en el cuadro, explica un 50% de la variabilidad de los datos y presenta una mejor performance en train respecto a **LightGBM**.

Ejercicio 4: Clustering

En este ejercicio debemos trabajar con un dataset [6] con estadísticas de las playlist de spotify, este tiene 736 registros y 13 columnas. En primer lugar se eliminaron los valores duplicados y se observaron la cantidad de valores nulos, la cual resultó ser 0. Para analizar la tendencia al clustering se decidió calcular la **Estadística de Hopkins** del dataset. Se observó que al calcularlo 10 veces, los valores oscilan entre 0.98 y 0.99, indicándonos que existe una fuerte tendencia al clustering.

Análisis de Tendencia al Clustering

Para lograr obtener la cantidad apropiada de clusters que se deben formar para trabajar con el dataset utilizamos la *Regla del Codo*, la cual se basa en ir aumentando la cantidad de clústeres hasta que la variabilidad intra-cluster disminuya de forma significativa, En el momento donde la variabilidad pasa a ser ínfima es la cantidad de clusters óptimos. Esto se realiza a través de la utilización del algoritmo **KMeans**. Como podemos observar en la *Figura 19*, la cantidad ideal de clústeres está entre los valores de 4 y 6.

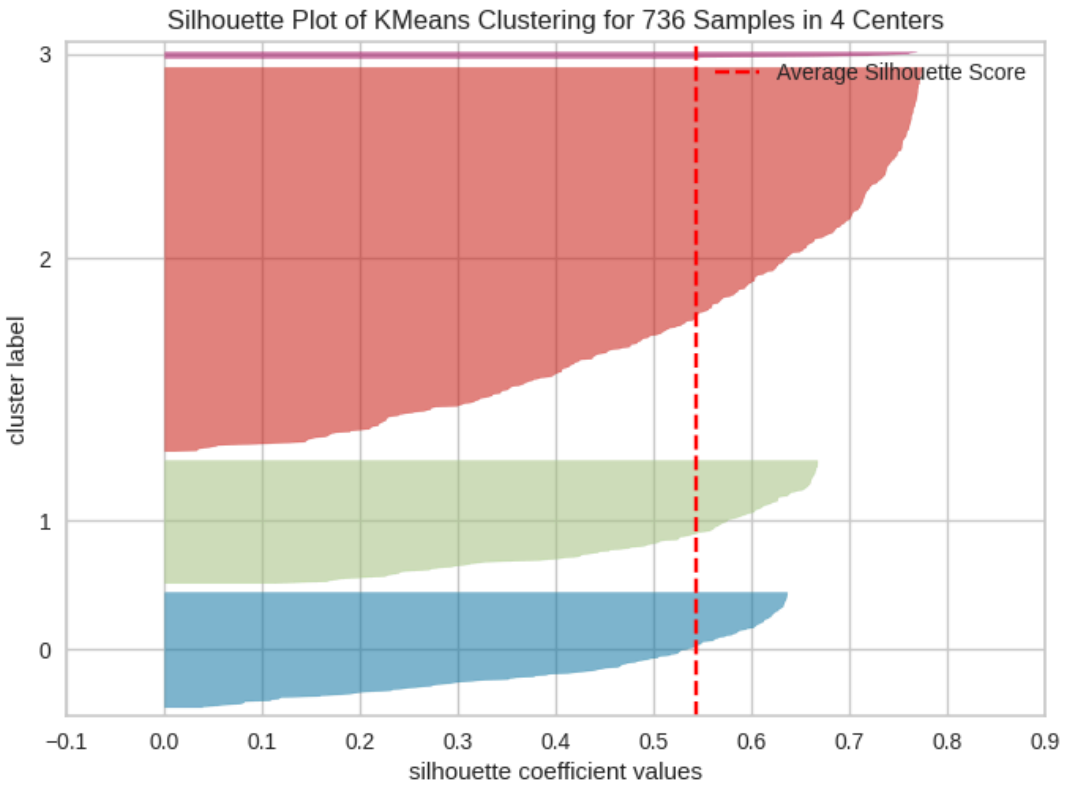


[Figura 19]

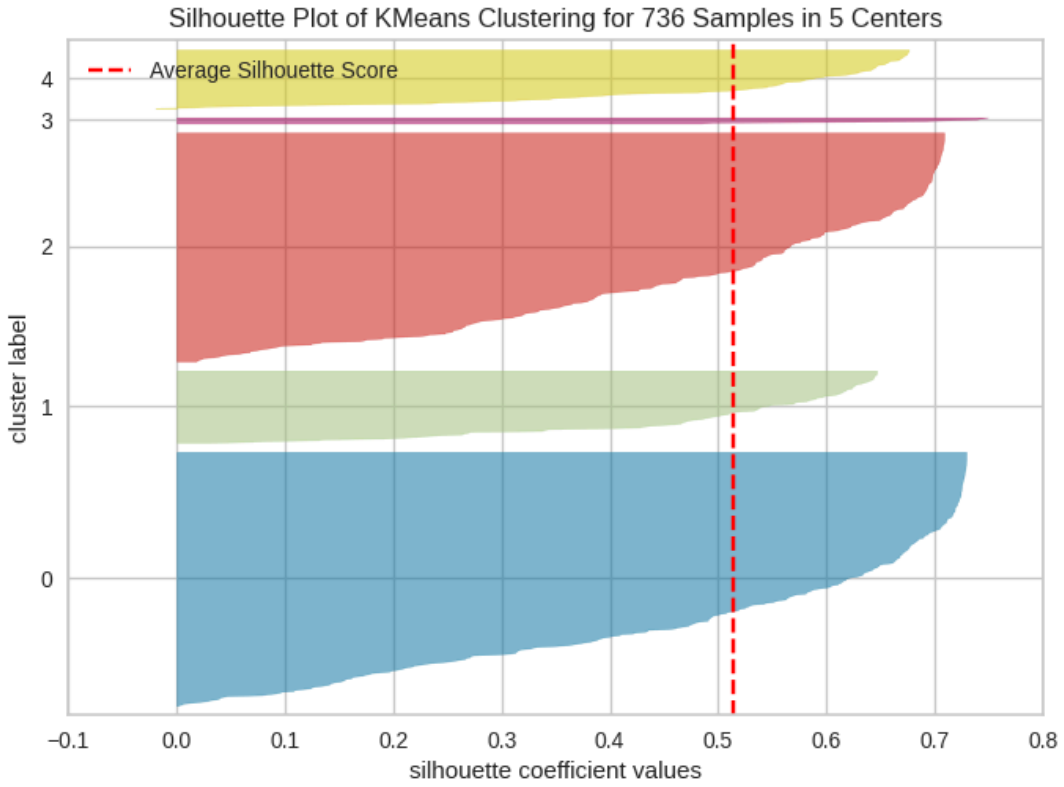
Para confirmar dicho hallazgo se utilizó el **Coeficiente de Silhouette** para esas cantidades de clústeres , con lo cual obtuvimos:

- For `n_clusters = 4`, *Silhouette Score* es 0.5429290236660737
- For `n_clusters = 5`, *Silhouette Score* es 0.5136451702953135
- For `n_clusters = 6`, *Silhouette Score* es 0.5200732935060439

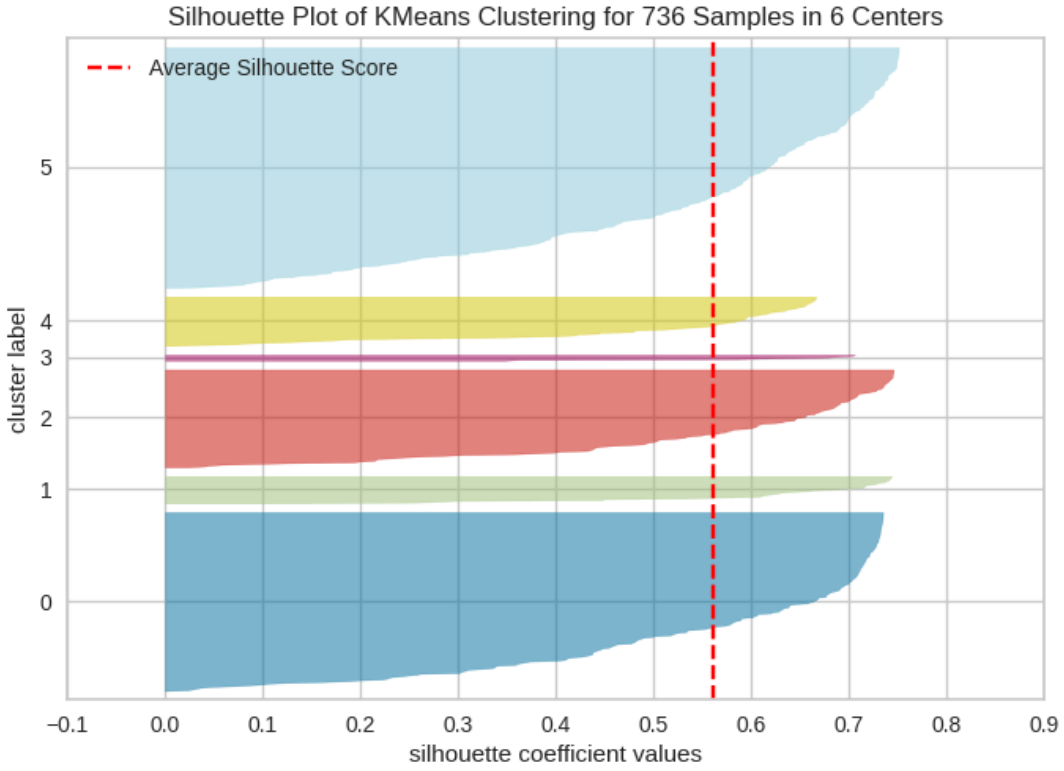
Dicha cantidades generan los siguientes *Silhouettes Plots*:



[Figura 20]

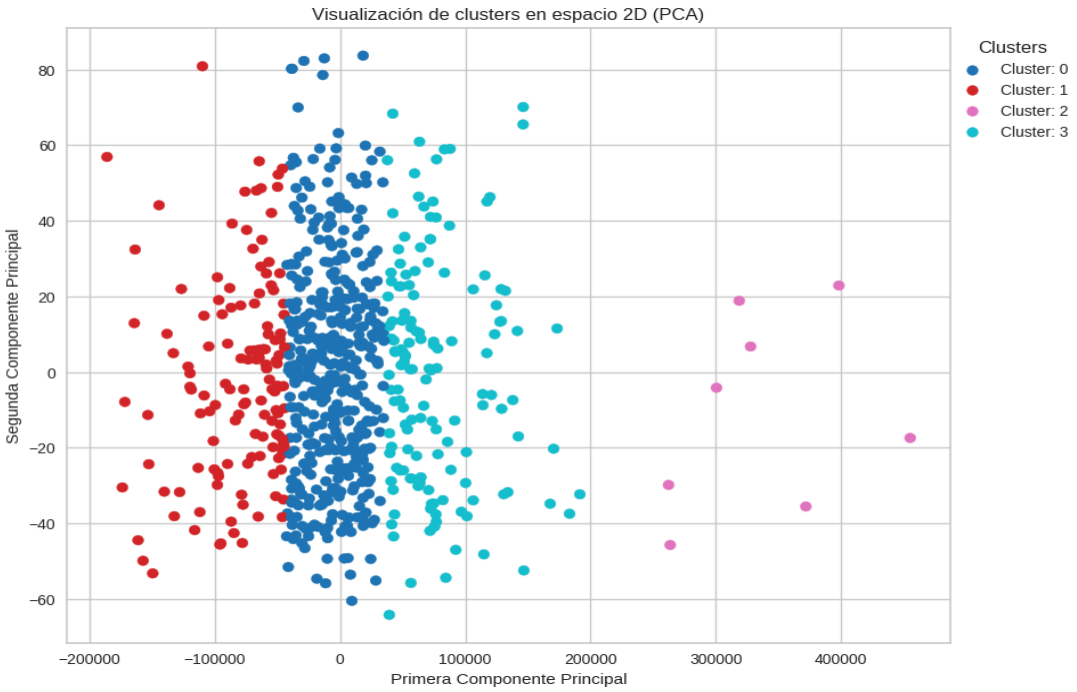


[Figura 21]



[Figura 22]

Teniendo en cuenta los **Coeficientes de Silhouette**, decidimos que el número de clusters óptimos eran 4, por lo que la distribución de los grupos queda de la siguiente manera:

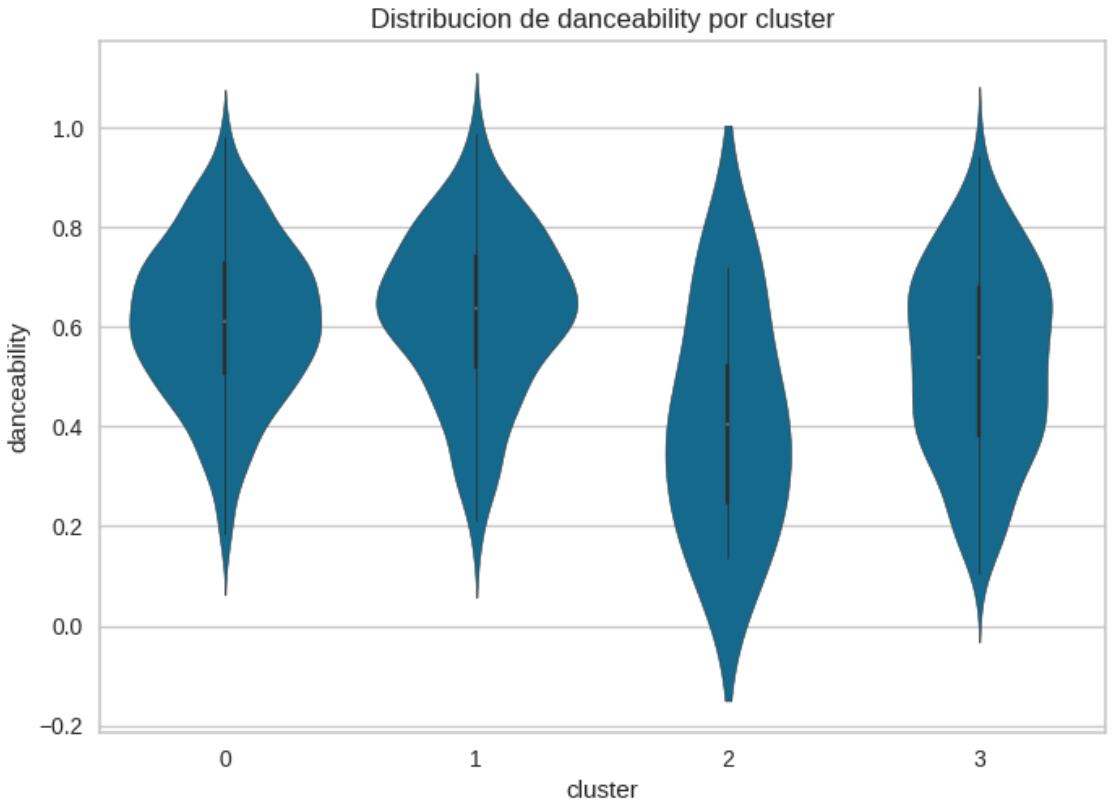


[Figura 23]

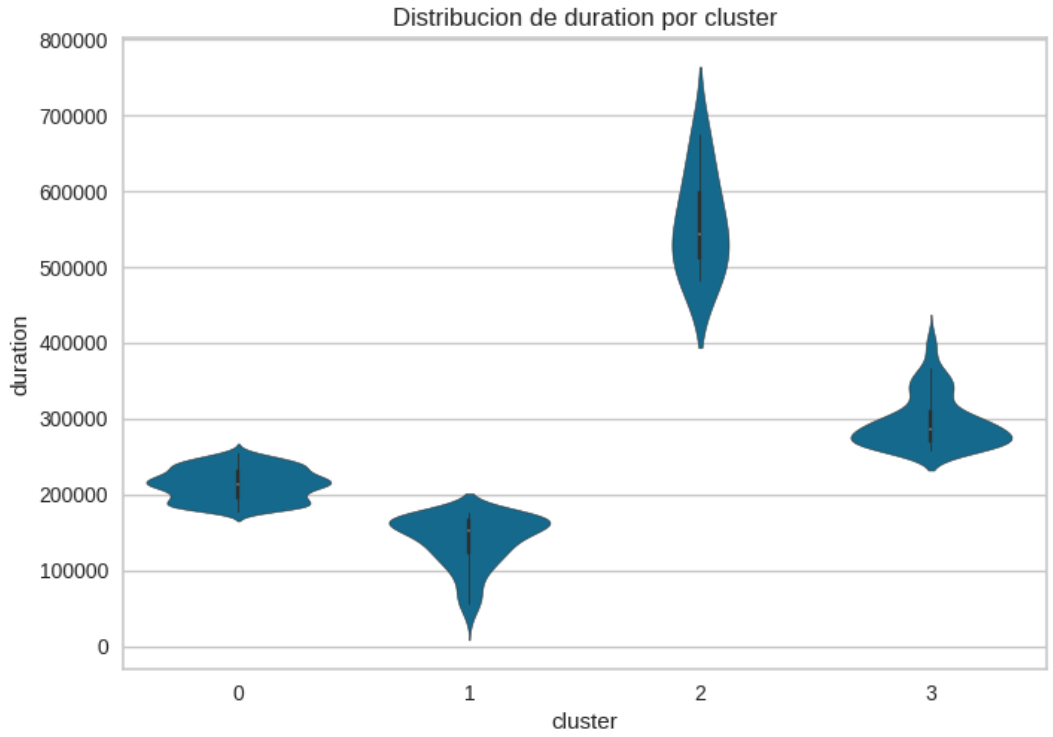
Conclusiones del Análisis

Luego de un análisis de las estadísticas de resumen de estos clusters, se concluyó que en el cluster 0 se encuentran canciones energéticas y bailables de duración media, esto queda evidenciado en que presenta un promedio de 0.61 de `danceability` y una mediana de 0.667 de `energy`. Para el cluster 1 se continúa el patrón de canciones energéticas y bailables con la principal diferencia que son canciones mucho más cortas, un promedio de 2 minutos por canción. Mientras que el cluster 2 presenta canciones largas, acústicas y poco bailables, esto se puede ver reflejado en un promedio de 9 minutos por canción y un 0.4 de `danceability`. Por último el cluster 3 sigue con la tendencia de canciones largas pero con un alto promedio de instrumentalidad, lo que permite pensar que son canciones melancólicas, esto se refuerza con un promedio de 0.4 en la columna de `valence`.

Todo lo anteriormente mencionado se puede observar en la distribución de cada cluster en dichas columnas presentadas a continuación.



[Figura 24]



[Figura 25]

Tiempo Dedicado

Integrante	Tarea	Prom. Hs Semana
Beltrán Malbrán	Detección de Outliers Armado de Reporte Imputación datos Predicción	12
Florencia Dellisola	Detección de Outliers Armado de Reporte Imputación datos Predicción	12
Gian Luca Spagnolo	Detección de Outliers Armado de Reporte Imputación datos Predicción	12
Marcela Jazmín Cruz	Detección de Outliers Armado de Reporte Imputación datos Predicción	12

Referencias y Recursos

[1] Página de *New York City Taxi and Limousine Comission*, que provee el set de datos para el ejercicio 1. [TLC Trip Record Data - TLC (nyc.gov)]

[2] *Mahalanobis Distance and its Application for Detecting Multivariate Outliers*. Hamid Ghorbani. Octubre 2019.

[3] Página de *Rain in Australia*, que provee el set de datos para el ejercicio 2. [Rain in Australia (kaggle.com)]

[4] *Model averaging prediction by K-fold cross-validation*. Xinyu Zhang, Chu-An Liu. Julio de 2023.

[5] Página de *Inside Airbnb*, que provee el set de datos para el ejercicio 3. [Get the Data | Inside Airbnb]

[6] Página de *Spotify for Developers*, que provee el set de datos para el ejercicio 4. [Web API Reference | Spotify for Developers]