Trabajo Práctico N°1 - Grupo 03

EJERCICIO 1 - Análisis Exploratorio de Datos

Análisis Exploratorio

El dataset contiene 8578039 registros y 19 columnas que representan datos de viajes en taxi. Cada fila corresponde a un viaje individual, y las columnas incluyen diversas características asociadas a cada viaje.

Características Destacables

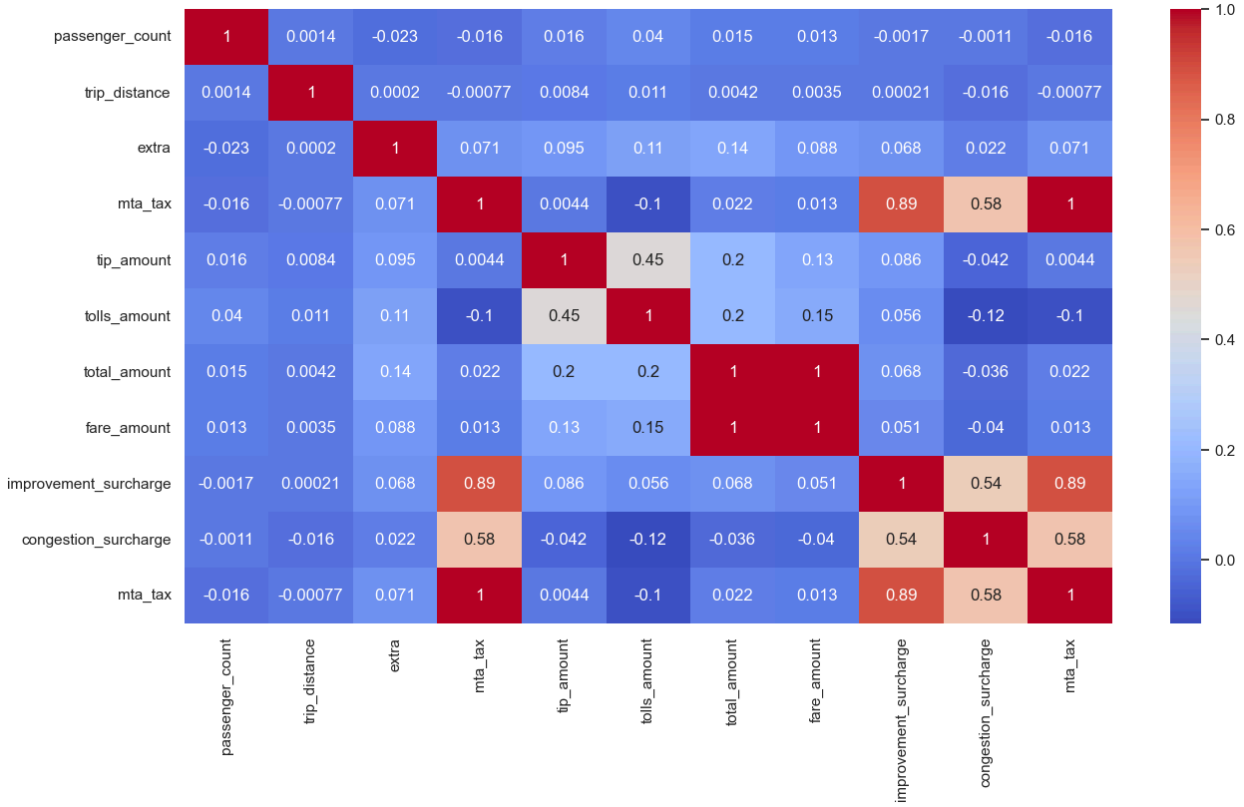
1. Passenger_count:
   - ➔ Tipo de dato: Entero
   - ➔ Descripción: Número de pasajeros en el viaje.
   - ➔ Relevancia: Es fundamental para calcular las tarifas por pasajero y para evaluar la capacidad del taxi.
2. trip_distance:
   - ➔ Tipo de dato: Flotante
   - ➔ Descripción: Distancia total del viaje en millas.
   - ➔ Relevancia: Es crucial para calcular el costo del viaje y evaluar la eficiencia del servicio.
3. RatecodeID:
   - ➔ Tipo de dato: Entero
   - ➔ Descripción: Código de tarifa utilizado para el viaje.
   - ➔ Relevancia: Indica el tipo de tarifa aplicable y puede afectar el costo total del viaje.
4. fare_amount:
   - ➔ Tipo de dato: Flotante
   - ➔ Descripción: Monto de la tarifa por el viaje.
   - ➔ Relevancia: Permite analizar la estructura de precios y compararla con las distancias recorridas.
5. total_amount:
   - ➔ Tipo de dato: Flotante
   - ➔ Descripción: Monto total a pagar por el viaje.

Preprocesamiento de Datos

1. ¿Se eliminaron columnas (Nombre de la columna y motivo de eliminación?

   No, no se han eliminado columnas.

2. ¿Detectaron correlaciones interesantes (entre qué variables y qué coeficiente)?



3. ¿Generaron nuevos features?
   Si
   - Tiempo del viaje (Haciendo la resta entre fecha y hora de llegada menos Fecha y hora de partida)
   - Precio neto por milla (Precio Neto medido/Millas Recorridas)
   - Precio total por milla (Precio Total/Millas recorridas)
   - Tipo de congestión
   - Precio por tipo de congestión

4. ¿Encontraron valores atípicos?¿Cuáles?¿Qué técnicas utilizaron y qué decisiones tomaron?

   Si,
   - Total_amount menores a 0 y mayores a 200
   - Trip_distance mayor a 100 millas
   - Tpep_pickup_datetime menores a 0
   - Hora de subida mayor a la hora de bajada
   - Viajes que tengan fecha fuera de nuestro rango

   Utilizamos análisis univariado para detectar outliers.

5. ¿Qué columnas tenían datos faltantes?
   ¿En qué proporción? ¿Qué se hizo con estos registros?

   Null

   - ➔ Passenger_count → 3.65%
   - ➔ RatecodeID → 3.65%
   - ➔ Store_and_fwd_flag → 3.65%
   - ➔ Congestion_surcharge → 3.65%
   - ➔ Airport_fee → 3.65%
   - ➔ En este caso se analizó que todas aquellas filas que contaban con 1 null, contaban con 5, por lo tanto todas tenían null en las 5 columnas anteriores. Se optó por eliminarlas.

Iguales a cero que no deberían serlo

Las columnas analizadas que no deberian ser 0 son passenger_count, trip_distance, fare_amount, mta_tax, improvement_surcharge, total_amount.

- ➔ Passenger_count → 1.45%
- ➔ trip_distance → 2.40%
- ➔ fare_amount → 0.03%
- ➔ mta_tax → 1.07%
- ➔ improvement_surcharge → 0.04%
- ➔ total_amount → 0.02%

Se optó por reemplazar estos valor con la media de cada una de las columnas

Valores que no corresponden a los valores pre-establecidos

- ➔ RateCodeId → 0.61%
- ➔ payment_type → 3.65%

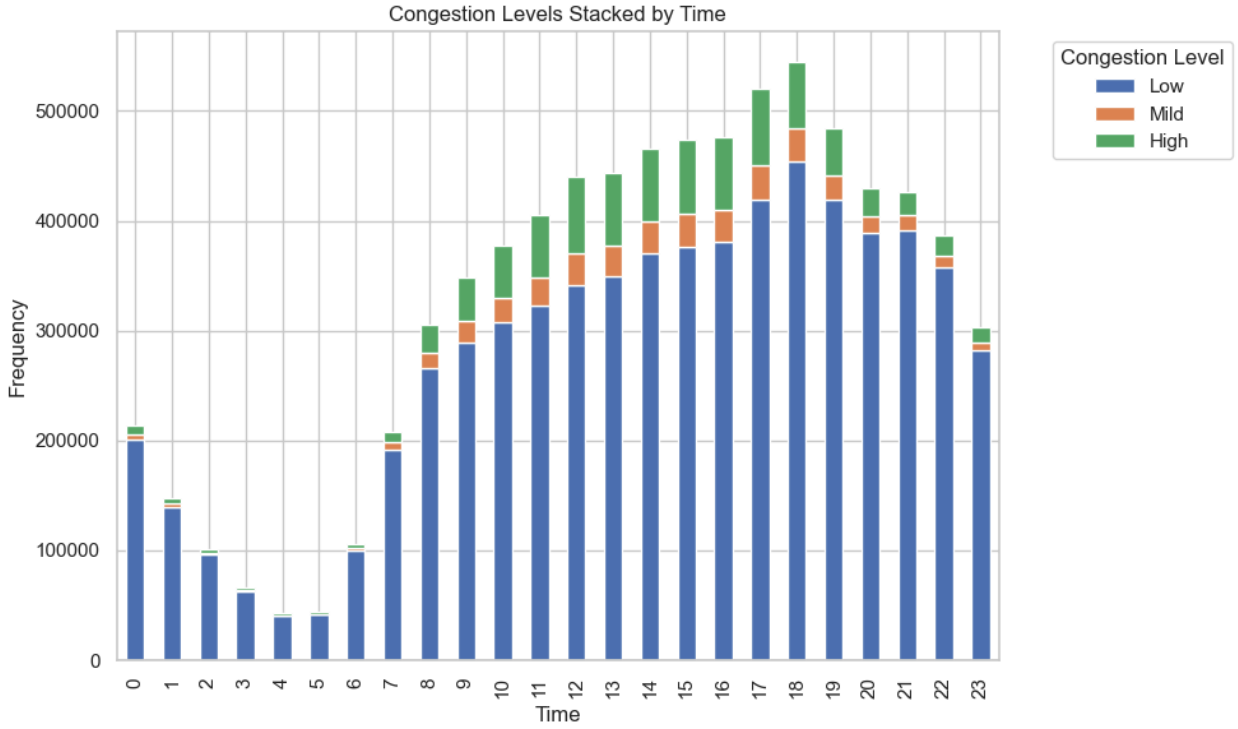
Se utiliza la moda para arreglar estos valores.

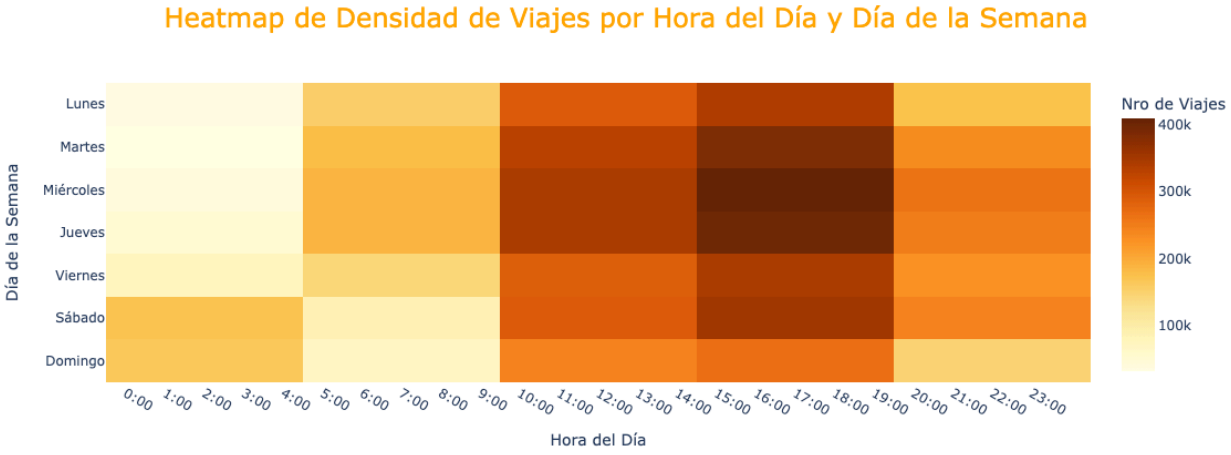
Menores a 0

- ➔ fare_amount → 1.07%
- ➔ extra → 0.53%
- ➔ mta_tax → 1.03%
- ➔ Tolls_amount → 0.08%
- ➔ improvement_surcharge → 1.06%
- ➔ Total_amount → 1.06%
- ➔ Congestion_surcharge → 0.84%
- ➔ Airport_fee → 0.17%

Suponemos que fueron un error de tipeo asi que los convertimos a positivos con un abs()

Visualizaciones





EJERCICIO 2 - Modelos de Clasificación Binaria

Descripción del Dataset

El dataset contiene 145460 registros y 23 columnas que representan datos del clima de Australia en distintas localizaciones. Cada fila corresponde a un día de 24hs, y las columnas incluyen diversas características asociadas a la situación climática.

Características Destacables

1. Rainfall:
   - ➔ Tipo de dato: float
   - ➔ Descripción: Cantidad de mm llovidos en el día.
   - ➔ Relevancia: Importante saber cuánto está lloviendo en una fecha para saber si puede llover al día siguiente.
2. Humidity3pm/Humidity9am:
   - ➔ Tipo de dato: Flotante
   - ➔ Descripción: Porcentaje de humedad registrado a las 3pm/9m.
   - ➔ Relevancia: Una alta humedad puede indicar una atmósfera cargada en agua, lo que puede influir en la predicción de lluvia.
3. Cloud3pm/Cloud9am:
   - ➔ Tipo de dato: float
   - ➔ Descripción: Cantidad de nubes en el cielo a las 3pm/9am.
   - ➔ Relevancia: Puede indicar condiciones propicias de lluvia.
4. Pressure3pm/Pressure9am:
   - ➔ Tipo de dato: float
   - ➔ Descripción: Presión atmosférica en hPa a las 3pm/9am.
   - ➔ Relevancia: Una baja presión puede dar signos de lluvia.
5. WindGuestSpeed:
   - ➔ Tipo de dato: float
   - ➔ Descripción: Velocidad máxima de viento registrada en el día.
   - ➔ Relevancia: Asociadas con posibles tormentas, que pueden indicar lluvias.
6. MaxTemp/MinTemp:
   - ➔ Tipo de dato: float
   - ➔ Descripción: Temperatura máxima/mínima registrada en el día.
   - ➔ Relevancia: Los valores de temperatura pueden causar inestabilidad atmosférica generando tormentas.

7. RainToday/RainTomorrow:
   - ➔ Tipo de dato: bool (string)
   - ➔ Descripción: Indica si llovió hoy/mañana/
   - ➔ Relevancia: Claramente usados para verificar si los modelos predicen correctamente.

Nuestro grupo tiene la hipótesis de que no debe ser fácil conseguir un modelo que pueda predecir qué días lloverá, dado que incluso las empresas que se dedican a la predicción meteorológica, teniendo miles de sensores y variables para usar, yerran bastante en la predicción de lluvia.

Preprocesamiento de Datos

Nuestro grupo tuvo que analizar las regiones de Victoria / Terr. del Norte / Aus. Meridional / Aus. Occidental / Tasmania, por lo que se agregó una columna tipo String que identifica de que área es cada localización.
La columna date contenía un tipo string que se pasó a tipo dateTime para poder manipular mejor. Para la dirección del viento, la localización, la región, el rain Today y rain Tomorrow, todos los tipo object/string o categóricos, se los redefinió usando one hot encoding, ya que algunos árboles de decisión sólo pueden trabajar con tipo de datos numéricos.
Se detectó una fuerte correlación entre tempMax y la temp3pm (del 0.99), entre la tempMin y la temp9am(0.88), entre la tempMin y tempMax (0.74), entre la temp9am y temp3pm(0.88), la pressure9am y pressure3pm(0.96), la windSpeed3pm y windGustSpeed(0.62). También se observa una fuerte correlación inversa entre cloud3pm y sunshine (-0.70), entre la humidity3pm y tempMax(-0.63), humidity3pm con temp3pm(-0.68).

Para limpiar outliers se realizó lo siguiente:

- Se reemplazaron los NaN de tempMax y temp3pm por los valores de la otra, ya que tienen una correlación casi perfecta.
- Dado que cloud3pm y sunshine tienen una correlación inversa, y que solo hay 10 valores posibles de cloud que van del 0 al 9, ordenamos los valores de sunshine y los dividimos en 10 grupos para reemplazar los NaNs de uno o del otro.

- Calculamos la diferencia media entre MaxTemp y MinTemp para ante la ausencia de una poder calcular la otra (es decir, si tenemos la MaxTemp podemos calcular la MinTemp faltante como MaxTemp - meanDifferenceTemp).

- Posteriormente reemplazamos MinTemp, MaxTemp y RainFall por las medias de estos.
- Luego filtramos los MinTemp y MaxTemp con Z-score.
- Realizamos un boxPlot de cada variable univariada y borramos los outliers de las variables de temperaturas, presión, humedad, viento, etc.
- Borramos unos outliers por lo investigado en internet respecto al clima de Australia (como por ejemplo que no llueve mas de 100mm, que los vientos máximos normales son de 110km/h, que las presiones atmosféricas más altas son de 1030 hPa por zonas altas).
- Utilizamos isolation forest y local outlier factor para eliminar outliers de temperatura, presión, humedad, nubes, Sunshine.

- Solo utilizamos las columnas de temperaturas, humedades, presiones, nubes, vientos, sunshine, y regiones para entrenar los árboles de decisión. Creemos que la fecha y la localidad no son relevantes para el modelo.
- Como hay muy pocos datos de días que llueve en comparación con los que no llovió, uso Smote para generar filas artificiales y equilibrar esa diferencia.

Modelos

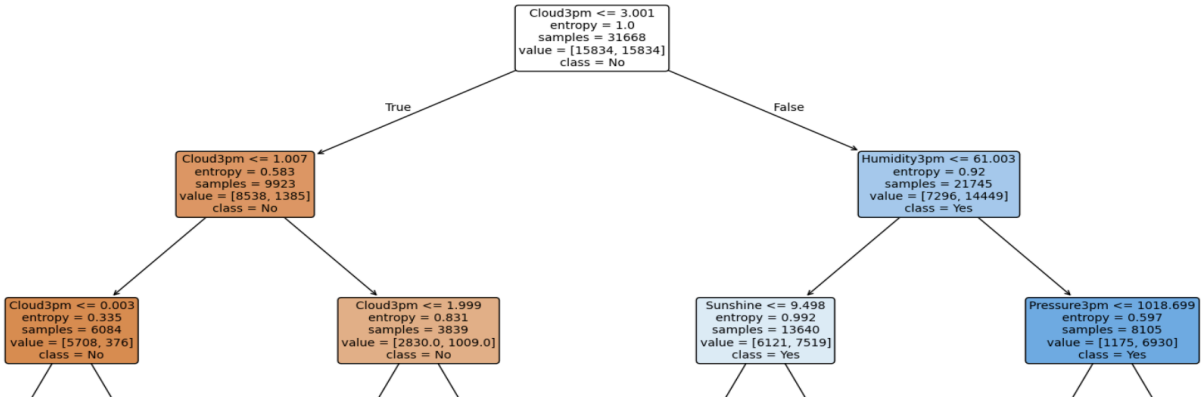
1. Arbol de Decision

- ¿Optimizaron hiperparámetros? ¿Cuáles?
  Si. Usamos Grid Search para optimizar los hiperparametros a mencionar.
  - max_depth: La profundidad del árbol.
  - min_samples_split: cantidad minima para dividir un nodo
  - min_sample_leaf: cantidad minima de muestras en una hoja
  - max_features: cantidad maxima de caracteristicas consideradas en cada nodo.
  - criterion: criterio de division.
  - splitter: metodo de division.
  - class_weight: ajuste de desbalances.

- ¿Utilizaron K-fold Cross Validation?¿Cuántos folds utilizaron?
  Se utilizaron 10-fold en el modelo.

- ¿Qué métrica utilizaron para buscar los hiperparámetros?
  Se usó F1 como métrica de scoring, para poder conseguir un equilibrio entre recall y precisión.

  ● Añadir imagen del árbol generado e incluir descripciones que consideren adecuadas para entender el mismo. Si es muy extenso mostrar una porción representativa.



Se destaca cluod3pm como la variable más importante de la raíz de nuestro árbol. Posteriormente se analiza tanto la humedad3pm y nuevamente la cloud3pm.

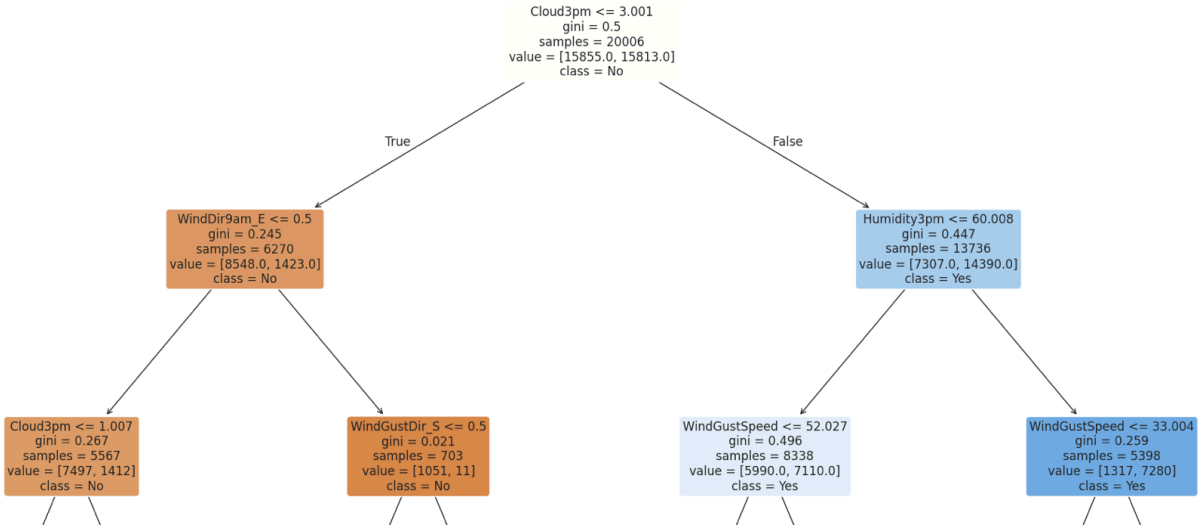
2. Random Forest

- ¿Optimizaron hiperparámetros? ¿Cuáles?
  Si. Usamos Grid Search para optimizar los hiperparametros a mencionar.
  - n_estimators: cantidad de árboles.
  - max_depth: profundidad máxima de un arbol.
  - max_features: cantidad máxima de caracteristicas consideradas en cada nodo.
  - min_samples_split: cantidad minima para dividir un nodo.
  - min_sample_leaf: cantidad minima de muestras en una hoja.
  - class_weight: ajuste de desbalances.

- ¿Utilizaron K-fold Cross Validation?¿Cuántos folds utilizaron?
  Se utilizaron 5-fold en el modelo.

- ¿Qué métrica utilizaron para buscar los hiperparámetros?
  Se usó F1 como métrica de scoring, para poder conseguir un equilibrio entre recall y precisión.

- Mostrar la conformación final de uno de los árboles generados. Si es muy extenso mostrar una porción representativa y explicar las primeras reglas.



En este caso se ve como las nubes de las 3pm es la variable más importante de la raíz, posteriormente se analiza la dirección del viento a la mañana y la si se ubica en la región meridional.

3. Modelo a Elección

- ¿Optimizaron hiperparámetros? ¿Cuáles?

  Si. Usamos Grid Search para optimizar los hiperparametros a mencionar.
  - n_estimators: cantidad de árboles
  - learning_rate: cuanto se ajusta el modelo al error de predicción en las iteraciones.

- max_depth: La profundidad del árbol.
- min_samples_split: cantidad mínima para dividir un nodo.
- min_sample_leaf: cantidad mínima de muestras en una hoja.
- subsample: fracción usada en cada árbol para entrenarlo.

- ¿Utilizaron K-fold Cross Validation?¿Cuántos folds utilizaron?
  Se utilizaron 15-fold en el modelo.

- ¿Qué métrica utilizaron para buscar los hiperparámetros?
  Se usó F1 como métrica de scoring, para poder conseguir un equilibrio entre recall y precisión.

Cuadro de Resultados

Modelo	F1	Precisión	Recall	Accuracy
Arbol de decision	0.44	40.63%	48.62%	84.30%
Random forest	0.56	56.30%	56.21%	88.79%
Gradient boosting	0.55	56.44%	53.62%	88.74%

Arbol de decision: {'class_weight': None, 'criterion': 'entropy', 'max_depth': None, 'max_features': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'splitter': 'best'}

Random Forest: {'class_weight': 'balanced', 'max_depth': 20, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}

Gradient Boosting: {'subsample': 1.0, 'n_estimators': 25, 'min_samples_split': 5, 'min_samples_leaf': 3, 'max_depth': 15, 'learning_rate': 0.1}

La precisión en el conjunto de prueba resultó ser del 84.30%, 88.79% y 88.74% siguiendo el orden anterior de los modelos.
Analizando estos resultados y conociendo teóricamente los modelos usados, creemos que el modelo que mejor se adapta a nuestro problema es el random forest, seguido muy de cerca por Gradient Boosting.

EJERCICIO 3 - Regresión

Descripción del Dataset

El dataset utilizado para este ejercicio de predicción de precios de alquiler en AirBnB cuenta con 34664 registros y 75 columnas, donde se incluyen tanto variables categóricas como numéricas. Entre las columnas destacadas se encuentran características relacionadas con la propiedad, como el número de habitaciones, baños, camas, y una variable objetivo que es el precio de alquiler.

- **Cantidad de registros**: 34664
- **Cantidad de columnas**: 75

Algunas de las transformaciones realizadas en los datos incluyen:

- **One-Hot Encoding** para las variables categóricas como ***property_type*** y ***room_type***.
- **Normalizació**n de las variables numéricas para mejorar el rendimiento de los modelos de regresión.
- **Imputación de valores faltantes** en variables como bathrooms, bedrooms, y beds usando la mediana.
- Creación de **nuevas variables derivadas** como ***price_per_bedroom, beds_per_bedroom, y accommodates_per_bathroom***.

Modelos

1. Regresión Lineal

- **Features seleccionadas**: Para este modelo, se seleccionaron variables que mostraron una fuerte correlación con el precio, tales como bedrooms, bathrooms, accommodates, beds, además de las variables codificadas como ***property_type, room_type, y neighbourhood_encoded.***
- **Transformaciones realizadas**: Las características categóricas fueron convertidas en variables dummies mediante One-Hot Encoding.
- **Performance del Modelo**:
  - **R^2 en entrenamiento**: 0.8594
  - **R^2 ajustado en entrenamiento**: 0.8593
  - **MSE en evaluación**: 7410.30
  - **RMSE en evaluación**: 86.08

La métrica principal utilizada fue el **RMSE** (Root Mean Squared Error) debido a su interpretación sencilla como el promedio del error en las predicciones, en unidades del precio. En este caso, el modelo de regresión lineal mostró una ligera pérdida de precisión al pasar del conjunto de entrenamiento al de evaluación, lo cual es esperado en un modelo que no esté sobreajustado.

2. XGBoost

- **K-Fold Cross Validation**: Se utilizó validación cruzada con **5 folds** para optimizar los hiperparámetros del modelo y evitar el sobreajuste.
- **Métrica para optimización**: La métrica utilizada para ajustar los hiperparámetros fue el **MSE**.
- **Performance del Modelo**:
  - **MSE en entrenamiento**: 548.23
  - **R^2 en entrenamiento**: 0.9903
  - **MSE en evaluación**: 651.11
  - **R^2 en evaluación**: 0.9884

XGBoost mostró un desempeño superior al modelo de regresión lineal, con un **MSE** y un **RMSE** mucho más bajos. Esto indica que el modelo de XGBoost tiene una capacidad notable para capturar las relaciones no lineales en los datos.

3. Support Vector Machine (Modelo a Elección)

- **K-Fold Cross Validation**: También se utilizó **5 folds** para este modelo.
- **Métrica para optimización**: Se utilizó el **MSE** para optimizar los hiperparámetros.
- **Performance del Modelo**:
  - **MSE en entrenamiento**: 15658.62
  - **R^2 en entrenamiento**: 0.7242
  - **MSE en evaluación**: 15544.95
  - **R^2 en evaluación**: 0.7240

El modelo SVM presentó un rendimiento inferior en comparación con XGBoost y la regresión lineal, lo que sugiere que este enfoque no es el más adecuado para este conjunto de datos en particular. El modelo tiene un alto error y no generaliza bien los datos.

Cuadro Comparativo de Resultados

Modelo	MSE (Entrenamiento)	MSE (Evaluación)	RMSE (Evaluación)	RMSE (Entrenamiento)	R^2 (Entrenamiento)	R^2 (Evaluación)
Regresión Lineal	7410.30	7410.30	89.27	86.08	0.8594	0.8593
XGBoost	548.23	651.11	25.52	23.41	0.9903	0.9884
Support Vector Machine	15658.62	15544.95	124.68	125.13	0.7242	0.7240

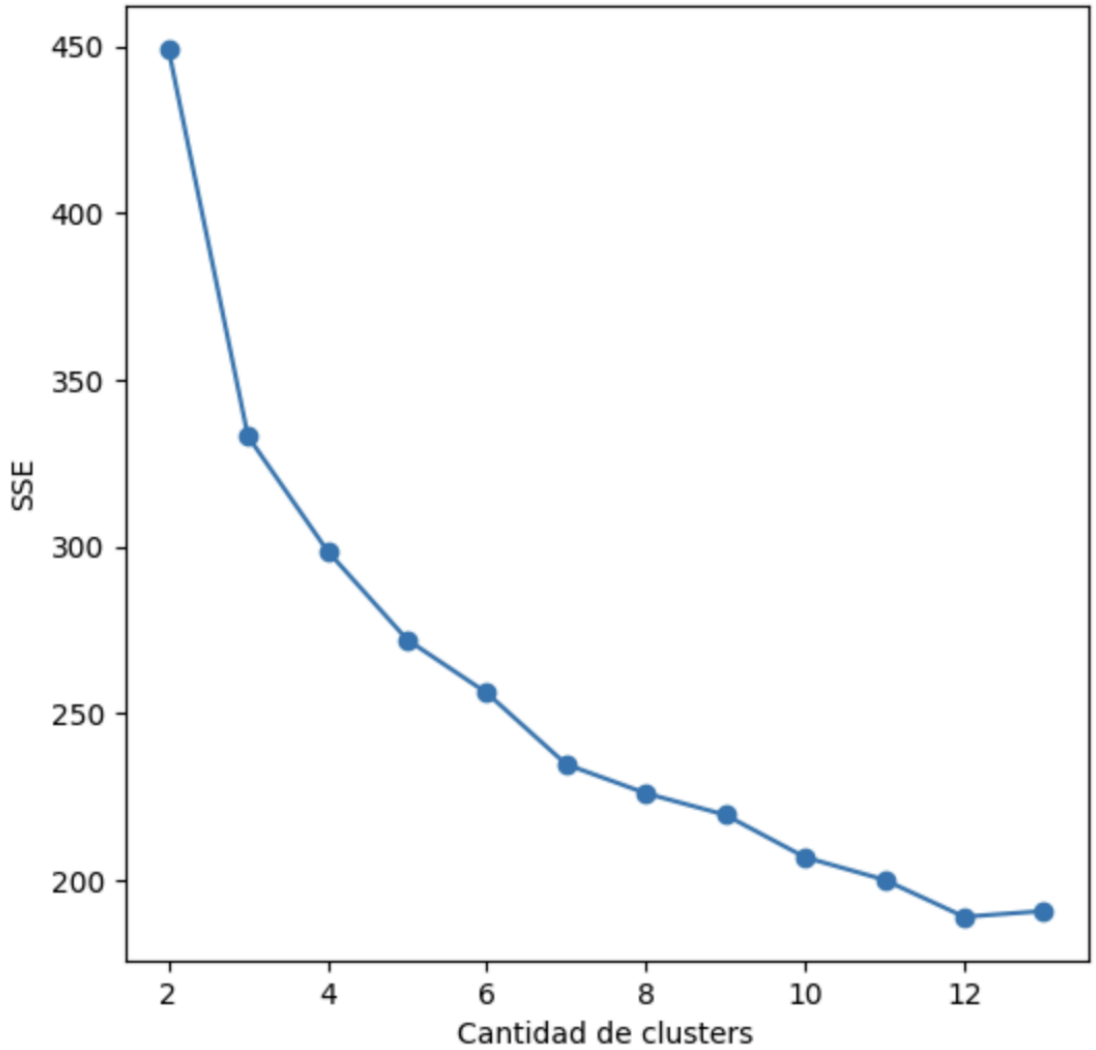
Elección del Modelo

El modelo XGBoost es el elegido como el mejor para la predicción de precios de propiedades en AirBnB debido a su desempeño tanto en el conjunto de entrenamiento como en el de evaluación. Presenta los menores valores de MSE y RMSE, lo que indica que es capaz de hacer predicciones más precisas. Su capacidad para capturar relaciones complejas entre las variables lo convierte en la opción más adecuada para este tipo de problema de regresión.

EJERCICIO 4 - Agrupamiento (Clustering)

Visualmente no se pueden ver tendencias de clustering, pero suponemos que haya una, dado que es un dataset de música, en los ritmos y estilos de canciones.

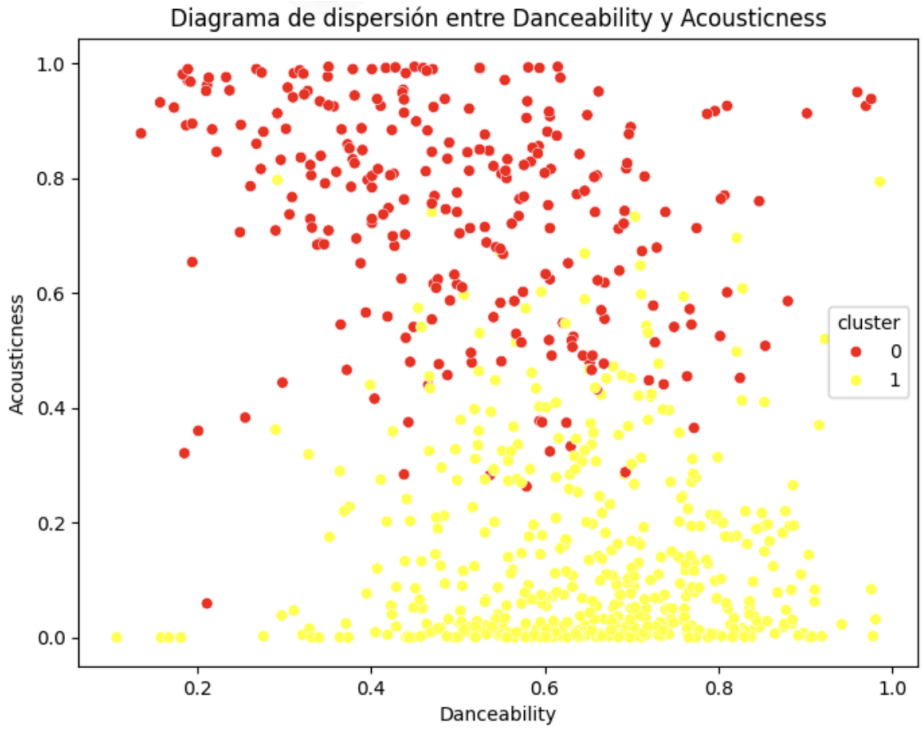
Usando el método de codo donde se ve una ligera tendencia a que se puedan dividir los grupos en 2 o 3.

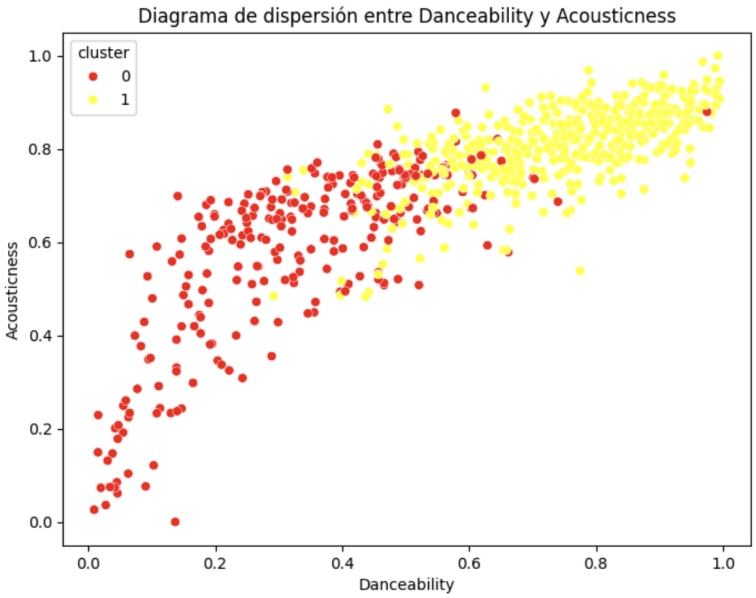


Para confirmar mejor cual es el mejor K usamos silhouette, que nos termina indicando que K = 2 es el coeficiente que tiene mejor separación de clusters. por lo que nos terminamos inclinando a este.

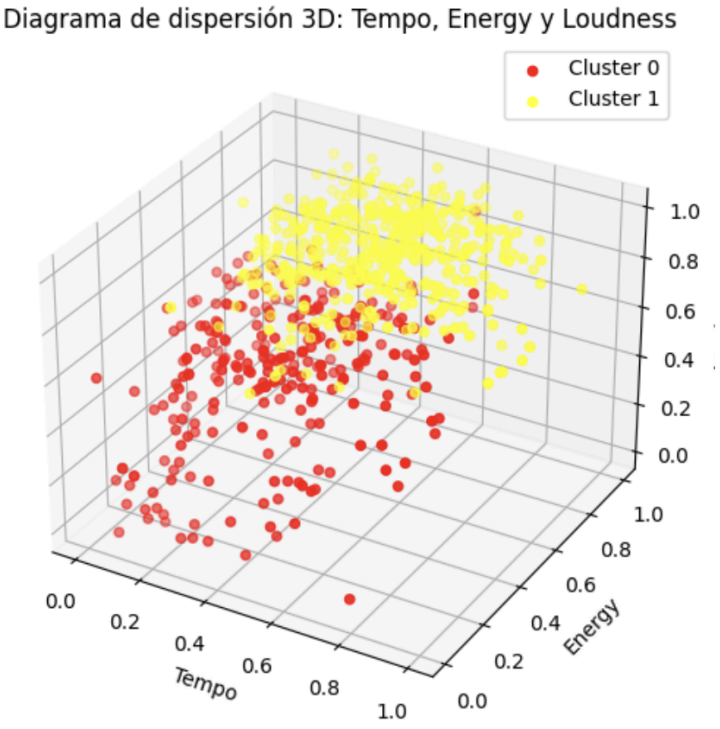
```
para K = 2 silhouette score es 0.3073751220809698
para K = 3 silhouette score es 0.2818500174749647
para K = 4 silhouette score es 0.21912428541173304
para K = 5 silhouette score es 0.22910702216581066
para K = 6 silhouette score es 0.23825796369118157
para K = 7 silhouette score es 0.22731018751591447
```

Dado que los grupos se calculan usando 9 features distintas, es imposible graficarlos en un gráfico de dispersión con todas las características. Sin embargo, al hacer los gráficos de dispersión entre dos de estas, probando todas las combinaciones posibles se ve un peso importante en acousticness, danceability y energy. Lógico con nuestra hipótesis inicial de que el ritmo iba a marcar la división de grupos.
Se ve la tendencia en siguientes gráficos:





En un gráfico 3D de tempo, energy y loudness también se ve una división clara de clusters.



Conclusiones Finales

En este trabajo hemos aplicado técnicas de análisis exploratorio y modelos de machine learning a diferentes datasets, logrando resultados satisfactorios en tareas de predicción y clasificación. En particular, se destacan los siguientes puntos:

- **Análisis exploratorio y preprocesamiento**: A lo largo del proyecto, se realizaron tareas como la detección y tratamiento de valores atípicos y datos faltantes. Este proceso fue fundamental para garantizar que los modelos de predicción trabajen con datos de alta calidad.

- **Modelos de predicción y clasificación**: Para la predicción de lluvia, el modelo Random Forest fue el que mejor se ajustó a los datos, resaltando variables clave como la nubosidad a las 3 p.m. Para la predicción de precios en AirBnB, XGBoost se destacó, capturando relaciones no lineales de manera más efectiva que otros modelos como la regresión lineal y SVM.

- **Resultados producto de la experimentación**: La experimentación con diversos modelos mostró la importancia de la selección y optimización de hiperparámetros. Random Forest y XGBoost demostraron ser las mejores opciones debido a su capacidad de manejar datos complejos y desbalanceados, mientras que otros enfoques, como SVM, no lograron capturar las relaciones presentes en los datos de manera efectiva.

- **Técnicas no exploradas**: Si bien logramos buenos resultados, algunas técnicas quedaron fuera del alcance de este trabajo. Hubiese sido interesante explorar modelos más avanzados como redes neuronales profundas, que podrían haber mejorado las predicciones en los datasets más complejos. La limitación principal fue el tiempo disponible y la complejidad de implementar estas técnicas.

En resumen, el trabajo logró cumplir con los objetivos planteados, pero futuras investigaciones podrían enfocarse en explorar métodos más avanzados y experimentar con técnicas adicionales para mejorar aún más la precisión de los modelos.

Tiempo dedicado

Integrante	Tarea	Prom. Hs Semana
Francisco Manuel Zimbimbakis	Modelos de clasificación Clustering Limpieza de datos Armado de reportes	10
Celeste De Benedetto	Visualizaciones Regresión Presentacion (ppt)	5
Morena Sandroni	Armado de reporte Detección de datos faltantes Revision de tareas	3
Adriana Tripodi Iglesias	Limpieza de datos	1