ClusterAI reporte

Grupo 06

Integrantes : Guillermo Lopez Dovigo y Emilie Gadrat

INDICE

0. Introduccion

La plataforma Airbnb solicitó el desarrollo de un modelo predictivo capaz de estimar el precio de los alojamientos publicados en distintas ciudades de Estados Unidos. Para ello, se proporciona un conjunto de datos compuesto por 19.309 publicaciones y 29 variables, que incluyen información estructural del inmueble, características del anfitrión, políticas de reserva y disponibilidad de amenities.

El objetivo principal de este trabajo es construir, mediante técnicas de aprendizaje automático implementadas en Python, un modelo que permita predecir el precio de cada alojamiento en función de sus atributos. Dado que la variable objetivo es numérica y continua, el problema se enmarca dentro del ámbito de la regresión supervisada.
A lo largo del proyecto se realizará la limpieza y transformación del dataset, un análisis exploratorio, la reducción de dimensionalidad mediante PCA y la comparación final entre distintos algoritmos de regresión. Esto permitirá evaluar qué modelo se ajusta mejor a la complejidad del problema y ofrece la mayor precisión predictiva.

1.Descripción del dataset y análisis exploratorio de datos

Para el desarrollo del modelo se trabajó con un dataset compuesto por 19.309 registros y 29 variables, que describen características de alojamientos publicados en la plataforma Airbnb dentro de Estados Unidos. Previamente al modelado, se realizó un proceso completo de limpieza, transformación y exploración de los datos.

1.1 Limpieza inicial del dataset

Se eliminaron todas las filas que contienen valores faltantes con el fin de evitar problemas durante el entrenamiento y garantizar la consistencia de las observaciones.

Asimismo, se removieron diversas columnas que no aportan información relevante para predecir el precio o que duplicaban información presente en otras variables :

- **Columnas irrelevantes o poco informativas :** id, thumbnail_url, name, first_review, host_since, last_review, zipcode.

- **Columnas redundantes :** neighbourhood, city y description, cuyos contenidos se superponen con información presente en latitude, longitude y la lista de amenities.

Esta depuración permitió reducir ruido y simplificar la estructura del dataset.

1.2. Transformación de variables y preparación para el aprendizaje automático

Dado que los modelos de regresión requieren variables numéricas, se aplicaron distintas transformaciones :

- Variables categóricas : Las columnas property_type, room_type, bed_type y cancellation_policy fueron convertidas mediante one-hot encoding utilizando get_dummies.

- Variables binarias con formato de texto ('t' / 'f') : Las columnas host_has_profile_pic, host_identity_verified e instant_bookable fueron transformadas a tipo booleano.

- Conversión de porcentajes : host_response_rate, originalmente expresada como texto con un símbolo “%”, fue limpiada y convertida a valores numéricos (float).

1.3 Procesamiento de la columna “amenities”

La columna amenities contenía una lista textual con múltiples comodidades por alojamiento.
Para aprovechar esta información, se aplicaron expresiones regulares (regex) y se seleccionaron algunos amenities considerados especialmente relevantes para el precio :

→ *Air conditioning, Kitchen, Washer, TV, Gym, Free parking on premises, Pool*

Para cada uno de ellos se creó una columna binaria (0/1) indicando su presencia. Finalmente, la columna original amenities fue eliminada al quedar representada en estas nuevas variables.

1.4 Análisis exploratorio de la variable objetivo

Se examinó la distribución de la variable price mediante histogramas y boxplots.
Estos gráficos permitieron identificar valores atípicos significativos (outliers) que podían distorsionar el entrenamiento de los modelos.
Para corregirlo, se aplicó el método IQR (Interquartile Range) y se eliminaron aquellas observaciones que excedían los límites establecidos, obteniendo así una distribución más representativa y robusta.

2. Materiales y métodos (algoritmos utilizados)

Con el fin de construir modelos predictivos robustos, se preparó el dataset mediante una serie de transformaciones previas y posteriormente se evaluaron distintos algoritmos de regresión. A continuación, se describen las etapas del procesamiento y los métodos utilizados.

2.1. Preparación del dataset para Machine Learning

Una vez completada la limpieza y transformación del dataset, se procedió a la preparación para el modelado.

2.1.1 Separación de variables

- X : conjunto de variables predictoras (todas las columnas excepto “price”).
- y : variable objetivo (precio del alojamiento).

2.1.2 División en entrenamiento y prueba

Se separaron los datos en : 80% para entrenamiento y 20% para prueba. Esto permitió evaluar el rendimiento final de los modelos sobre datos no utilizados durante el entrenamiento.

2.1.3 Estandarización

Las variables predictoras se estandarizaron mediante StandardScaler, transformándolas para que tengan media 0 y desviación estándar 1.
Este paso es esencial para : calcular correctamente la matriz de covarianza, aplicar PCA, y mejorar el rendimiento de modelos basados en distancia (SVR y MLP).

2.2. Reducción de dimensionalidad : PCA

Se aplicó un Análisis de Componentes Principales (PCA) con 10 componentes, seleccionados a partir de la curva de varianza explicada.
Se calcularon : los autovalores (eigenvalues), los autovectores (eigenvectors), y la varianza acumulada, que permitió justificar la elección de mantener 10 componentes.

El PCA se utilizó como paso previo para : reducir la dimensionalidad, eliminar redundancias, mejorar estabilidad de los modelos, y disminuir tiempos de entrenamiento.

2.3. Algoritmos de regresión utilizados

Se implementaron tres técnicas de regresión con características y capacidades predictivas diferentes, para comparar modelos lineales y no lineales.

2.3.1. Regresión Lineal Múltiple

Funciona como modelo base.
Su objetivo es servir como referencia para evaluar la mejora obtenida mediante algoritmos no lineales.

2.3.2. Support Vector Regression (SVR)

Se utilizó un kernel RBF (Radial Basis Function) para capturar relaciones no lineales entre las variables.
Los hiper parámetros empleados fueron : kernel='rbf', C = 30, epsilon = 0.2, gamma = 0.03

2.3.3. Justificación de la elección de hiper parámetros (sin Grid Search completo)

No se realizó un Grid Search exhaustivo debido a que el objetivo principal de este trabajo es comparar métodos de regresión, y no optimizar al máximo cada uno.
Los valores seleccionados provienen de configuraciones frecuentemente recomendadas en la literatura, conocidas por ofrecer resultados estables sin incurrir en un costo computacional elevado.

Realizar un Grid Search completo hubiese implicado entrenar un gran número de combinaciones posibles, aumentando considerablemente el tiempo de cómputo sin aportar un beneficio central al objetivo del TP.

2.3.4 Mini–búsqueda de hiper parámetros (exploración reducida)

Se llevó a cabo una pequeña búsqueda manual alrededor de los valores principales, probando rangos acotados como :

- $C \in \{10, 30, 50\}$
- $gamma \in \{0.01, 0.03, 0.05\}$

Esto permitió seleccionar configuraciones estables y evitar sobreajuste utilizando un costo computacional razonable.

2.3.5. MLPRegressor (Red Neuronal Artificial)

Se implementó un perceptrón multicapa (MLP) con la siguiente arquitectura :

- Capa oculta 1 : 64 neuronas
- Capa oculta 2 : 32 neuronas
- Función de activación : ReLU
- Optimizador : Adam
- Regularización L2 : $\alpha = 0.0005$
- Máximo de iteraciones : 1000

Esta configuración representa una red neuronal pequeña, adecuada para datasets tabulares como Airbnb y suficientemente compleja para modelar relaciones no lineales.

La arquitectura se definió siguiendo buenas prácticas de la literatura, priorizando un balance entre capacidad de aprendizaje y prevención del sobreajuste.

3. Experimentos y resultados

Para analizar la estructura interna del dataset y reducir la dimensionalidad, se aplicó un Análisis de Componentes Principales (PCA) sobre las variables estandarizadas. El objetivo fue identificar qué proporción de la variabilidad total puede explicarse con un número reducido de componentes, y evaluar si esta transformación mejora la estabilidad y el rendimiento de los modelos.

En primer lugar, se calcularon los autovalores (eigenvalues) asociados a cada componente principal, los cuales indican la cantidad de varianza explicada individualmente por cada una. Luego se obtuvo la varianza acumulada, lo que permitió determinar cuántas componentes son necesarias para capturar la mayor parte de la información contenida en los datos originales.

El análisis mostró que las primeras componentes concentran una proporción significativa de la variabilidad del dataset. En función de la curva de varianza acumulada y del balance entre pérdida de información y simplicidad del modelo, se decidió conservar 10 componentes principales. Esta cantidad permitió mantener una alta representación de la estructura original reduciendo al mismo tiempo el número de dimensiones y el riesgo de sobreajuste.

Una vez aplicado el PCA, se continuó con el entrenamiento y evaluación de los modelos de regresión, comparando su desempeño sobre el conjunto de prueba.

4. Discusión y conclusiones

Los modelos fueron evaluados utilizando RMSE y R^2, lo que permitió comparar su capacidad de predicción sobre el conjunto de prueba sin y con PCA. Los resultados obtenidos fueron los siguientes :

Modelo sin PCA	**RMSE (USD)**	**R²**
MLPRegressor (NN)	49.800	0.544
SVR (RBF)	51.718	0.508
Regresión Lineal	59.731	0.344

Modelo con PCA	**RMSE (USD)**	**R²**
MLPRegressor (NN)	**48.783**	**0.508**
SVR (RBF)	48.868	0.506
Regresión Lineal	51.912	0.443

El modelo que presentó el mejor desempeño global fue el MLPRegressor con PCA (red neuronal), con un RMSE de 48,78 USD y un R² de 0,508, lo que indica que explica más de la mitad de la variabilidad del precio. En comparación :

- La Regresión Lineal con PCA obtuvo un RMSE mayor (51,91 USD) y un R² menor (0,443), mostrando que no captura adecuadamente la relación entre las variables.

- El SVR con kernel RBF con PCA alcanzó un rendimiento muy cercano al MLP, con RMSE de 48,87 USD y R² de 0,506, confirmando que la relación entre las variables es no lineal.

- La red neuronal MLP logró el menor error promedio y la mejor capacidad explicativa.

Estos resultados permiten concluir que los modelos no lineales superan claramente a la regresión lineal clásica, ya que logran representar mejor las interacciones y patrones complejos del conjunto de datos. Entre ellos, el MLPRegressor se destaca como la alternativa más precisa y robusta para predecir precios de alojamientos de Airbnb, constituyéndose en la mejor opción entre los modelos evaluados.

También podemos ver que los resultados son mejores con el PCA, los errores son menores. Al reducir el número de variables :

- los modelos se vuelven más rápidos,
- menos sensibles al ruido,
- menos propensos al sobreaprendizaje (reducción del sobreaprendizaje)