



Calculadora de inversión de Airbnb

Camila Iturry

CODER HOUSE

Proyecto Final: Análisis Predictivo de Precios para Alojamientos de Airbnb en Nueva York

Autor/a: Brisa Camila Iturry

Curso: Data Science I

Academia: **CODER HOUSE**

Abstract

Este proyecto se sumerge en el dinámico mercado de alquileres temporales de Airbnb en Nueva York, utilizando el dataset público "Airbnb Open Data.csv". El objetivo principal es desarrollar y evaluar un modelo de Machine Learning capaz de predecir el precio por noche de un alojamiento, identificando los factores más influyentes en su tarificación. Tras un riguroso pipeline de limpieza de datos, análisis exploratorio (EDA) e ingeniería de características, donde se crearon **7 nuevas variables**, incluyendo la *antigüedad de la propiedad y la presencia de servicios clave*, se entrenaron dos modelos de regresión. El modelo final, un RandomForestRegressor, utilizando un conjunto final de **15 características** predictoras, el modelo alcanzó un coeficiente de determinación (R^2) de **0.346** superando de manera contundente a un modelo de regresión lineal base. El análisis de importancia de características reveló que la **ubicación geográfica (lat, long)** y la **disponibilidad anual** son los predictores más potentes. El proyecto concluye que, si bien el modelo actual proporciona una herramienta valiosa para la estimación de precios, la precisión podría mejorarse significativamente enriqueciendo el dataset con datos sobre la calidad y los servicios específicos de cada propiedad.

Introducción

Contexto Comercial y Problema de Negocio

En el competitivo mercado de Airbnb, el desafío es fijar un precio óptimo tanto para anfitriones, que buscan maximizar su rentabilidad, como para huéspedes, que desean evaluar si una tarifa es justa. El problema central es la ausencia de un benchmark de precios objetivo que considere la multitud de variables que influyen en el valor de un alquiler temporal en una ciudad tan heterogénea como Nueva York.

Objetivo del Proyecto:



El objetivo de este proyecto es **construir un modelo de regresión que prediga el precio de un alojamiento en Nueva York**, sirviendo como base para:

- Ayudar a los anfitriones a posicionar sus propiedades con un precio competitivo.
- Permitir a los huéspedes evaluar si el precio de un listado es adecuado.
- Identificar y cuantificar los atributos de una propiedad que tienen el mayor impacto en su valor.

Hipótesis Principales

El análisis se guio por las siguientes hipótesis centrales:

- **Hipótesis de Ubicación:** El distrito y la ubicación exacta (*lat, long*) son los factores más determinantes en la variación de los precios.
- **Hipótesis de Tipo de Propiedad:** El tipo de habitación (*room_type*) tiene un impacto significativo y directo sobre el precio.
- **Hipótesis de Valor Añadido:** La creación de nuevas características (*ingeniería de características*), como como la **antigüedad de la propiedad** y la presencia de servicios específicos (*ej. Wifi, cocina*), mejorará la precisión predictiva del modelo.

2. Metodología y Herramientas

1. Librerías:

Para desarrollar este proyecto, se seleccionó un conjunto de herramientas estándar de la industria dentro del ecosistema de **Python**, cubriendo todo el ciclo de vida del análisis, desde la manipulación de datos hasta el modelado y la evaluación.

Manipulación y Cálculo Numérico:

- **Pandas:** Es la librería fundamental del proyecto. Se utilizó para cargar el **dataset** (*pd.read_csv*), estructurarlo en **DataFrames**, y realizar la gran mayoría de las tareas de limpieza y transformación (*ej. .drop, .fillna, .astype*). Es la columna vertebral de toda la manipulación de datos.
- **NumPy:** Se empleó como la librería estándar para operaciones numéricas, especialmente para la aplicación de lógica condicional vectorizada (*np.where*) en la imputación de precios y para el cálculo de la raíz cuadrada (*np.sqrt*) en la métrica final de **RMSE**.

Visualización de Datos:

- **Matplotlib y Seaborn:** Estas librerías fueron clave para la fase de Análisis Exploratorio (**EDA**). **Matplotlib** se usó como la base para configurar y personalizar las visualizaciones (*títulos, etiquetas*), mientras que **Seaborn** fue la herramienta principal para generar los gráficos estadísticos (*sns.countplot, sns.boxplot, sns.heatmap*), permitiendo crear visualizaciones informativas y estéticamente atractivas con facilidad.

Machine Learning con scikit-learn:

- **train_test_split (sklearn.model_selection):** Función crucial utilizada para dividir el **dataset** en conjuntos de entrenamiento y prueba, un paso fundamental para evaluar el modelo de forma objetiva y prevenir el sobreajuste.

Modelos de Regresión (sklearn.linear_model y sklearn.ensemble):

- **LinearRegression:** Se implementó como modelo **baseline** para establecer una primera métrica de rendimiento.
- **RandomForestRegressor:** Se eligió modelo principal por su capacidad para capturar relaciones no lineales complejas, lo cual fue clave para superar las limitaciones del modelo lineal.

Métricas de Evaluación (sklearn.metrics):

- **mean_squared_error:** Se usó para calcular el error cuadrático medio entre las predicciones y los valores reales.
- **r2_score:** Se empleó para calcular el coeficiente de determinación (**R²**), que mide qué porcentaje de la variabilidad del precio es explicado por el modelo.

2. Descripción del Dataset

Se utilizó el [dataset "Airbnb Open Data.csv"](#), que contiene **102,599** registros y **26** columnas. Estas columnas abarcan información clave sobre *la propiedad, el anfitrión y las interacciones de los huéspedes*. A continuación, se describen las variables más relevantes utilizadas en este estudio:

Descripción de las Columnas relevantes del Dataset

- **Datos Geográficos (coordenadas lat, long y distritos).**
- **Características de la Propiedad (room_type, minimum_nights).**
- **Métricas de Reputación (number_of_reviews).**
- **Datos de Texto no estructurado (publicity, house_rules).**
- **Información de Precios, siendo price la variable objetivo a predecir.**

Para garantizar la **reproducibilidad** del análisis, se implementó un **script** de carga robusto que, mediante una estructura **try-except**, intenta obtener los datos localmente y, si no los encuentra, los descarga automáticamente desde un repositorio de [**GitHub**](#). Se utilizó **pd.read_csv** con parámetros específicos (**sep=';'**, **low_memory=False**) para manejar las particularidades del archivo y asegurar una correcta interpretación de los datos.

Una vez cargados, un diagnóstico inicial reveló varios desafíos críticos que justificaron la necesidad de un **pipeline** de preprocessamiento exhaustivo:

Problemas de Tipo de Dato: Columnas numéricas clave como **price** fueron cargadas como texto (**object**), lo que impedía su uso en cálculos matemáticos y requería una limpieza y conversión forzosa.

Presencia de Nulos y Datos Sucios: Se confirmó la existencia de una cantidad significativa de valores nulos (**NaN**) y datos inconsistentes (ej. "**unconfirmed**" en columnas categóricas), lo que comprometía la calidad del [dataset](#).

3. Ciclo de vida de un proyecto de ciencia de datos

Para transformar los datos crudos en un recurso fiable para el modelado, se implementó un **pipeline** de preprocessamiento secuencial y robusto. Este proceso fue diseñado para abordar sistemáticamente los problemas de calidad identificados en el diagnóstico inicial, asegurando la integridad y consistencia del [dataset](#) final.

El primer paso fue crear una base de trabajo coherente. Se estandarizaron los nombres de las columnas a un formato uniforme (**minúsculas y guiones bajos**) y se unificaron todas las representaciones de datos faltantes al estándar **Nan de Pandas**. Posteriormente, se abordaron los tipos de datos incorrectos:

Las columnas de precio (**price, service_fee**) se limpiaron de símbolos monetarios y comas antes de ser convertidas a formato numérico.

La columna de fecha (**last_review**) se transformó a formato **datetime** para permitir operaciones temporales en fases posteriores.

Más allá de la limpieza básica, se aplicaron reglas de negocio y correcciones contextuales para enriquecer los datos:

Filtrado de Precios: Se aplicó lógica de negocio para imputar precios registrados como cero y se acotó el análisis a un rango de hasta **\$1200** para eliminar **outliers** extremos.

Corrección Geográfica: Se resolvió la inconsistencia en la columna **neighbourhood_group** mediante la creación de un mapa de correspondencia "**barrio -> distrito**", generando una nueva columna distritos geográficamente precisa.

Manejo de Cardinalidad: Para la columna **host_identity_verified**, que presentaba más de 160 valores únicos, se aplicó una binarización forzada ("**verified**" / "**unconfirmed**") para simplificar la característica sin perder su valor predictivo.

Finalmente, para asegurar un **dataset** completo, se implementó una estrategia de imputación diferenciada: las coordenadas geográficas se llenaron usando la mediana del distrito correspondiente, las columnas de texto con cadenas vacías, y las variables numéricas y categóricas restantes con su mediana y moda, respectivamente.

Resultado del Proceso: El **pipeline** se ejecutó con éxito, resultando en un **DataFrame** final de **102,058** filas y **25** columnas, con cero valores nulos. La creación de la crucial columna distritos fue confirmada, dejando el **dataset** limpio, coherente y perfectamente preparado para las fases de análisis exploratorio y modelado.

4. Análisis Exploratorio de Datos (EDA)

Una vez que los datos han sido limpiados y preprocesados, se procede con el **Análisis Exploratorio de Datos**. El objetivo de esta fase es utilizar visualizaciones (**Insights Parte I**) para descubrir patrones, validar hipótesis iniciales y obtener una comprensión profunda de la estructura del mercado de **Airbnb** en **Nueva York**.

4.1 Análisis de la Oferta: ¿Dónde se Concentra la Oferta de Alojamientos?

Para responder a esta pregunta fundamental, se generó un gráfico de barras que visualiza la cantidad total de propiedades por cada uno de los distritos corregidos.

Grafico:

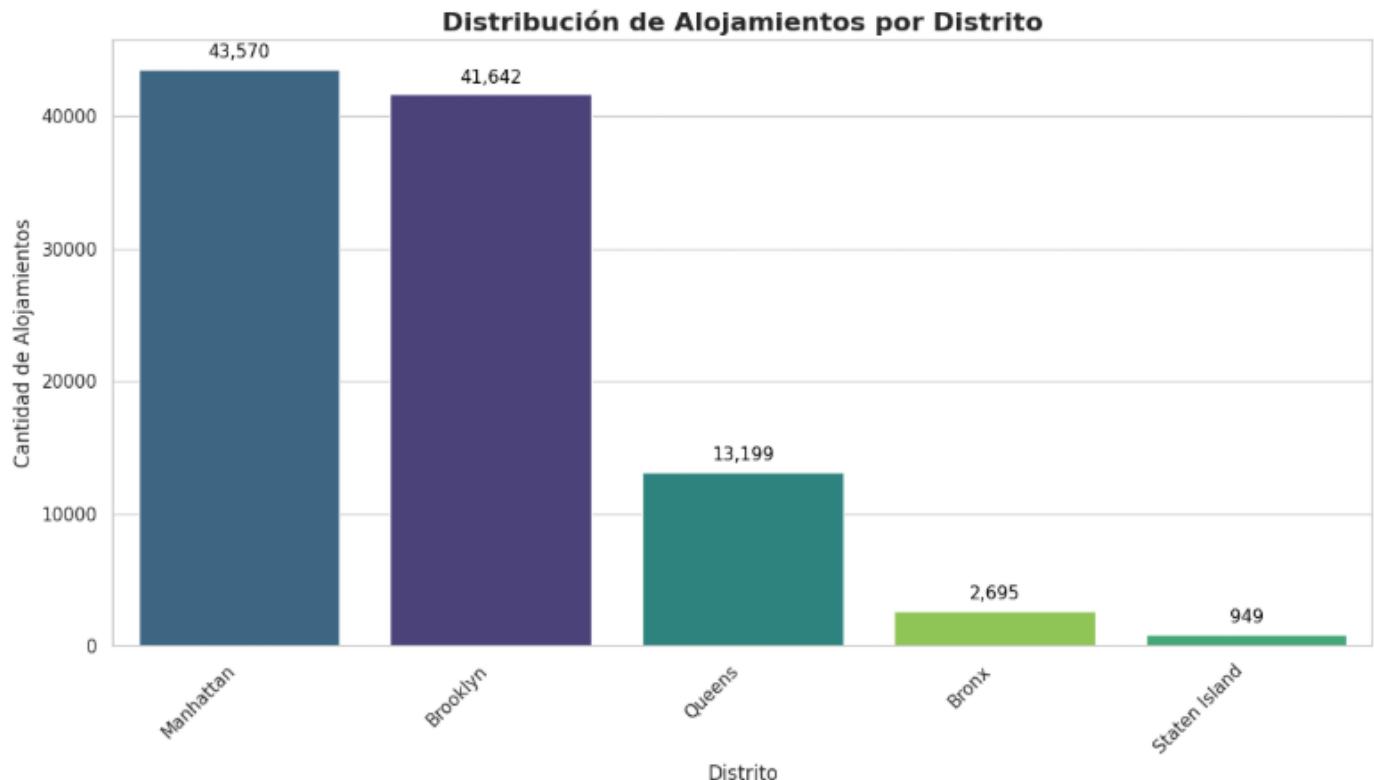


Figura 4.1: Conteo de listados de Airbnb por distrito en Nueva York.

Interpretación del Gráfico y Recomendaciones de 4.1

Insights Clave:

- **Concentración Masiva del Mercado:** Es evidente que la oferta de alojamientos no es uniforme. **Manhattan (43,570 listados)** y **Brooklyn (41,642)** dominan de manera abrumadora el mercado, constituyendo el núcleo indiscutible de la actividad.
- **Jerarquía de Mercados:** Se define una clara estructura por niveles. **Queens (13,199)** emerge como un importante mercado secundario, mientras que el **Bronx (2,695)** y **Staten Island (949)** representan mercados de nicho con una oferta considerablemente menor.

Recomendaciones Estratégicas:

- **Para Anfitriones:** Operar en **Manhattan** o **Brooklyn** implica enfrentarse a una altísima competencia, lo que exige una fuerte estrategia de diferenciación. Por el contrario, distritos como **Queens** o el **Bronx** presentan una menor saturación y pueden ser una oportunidad para capturar segmentos de viajeros con presupuestos más ajustados.
- **Para el Modelo Predictivo:** La marcada diferencia en la oferta confirma que el distrito es una característica predictiva fundamental que deberá ser incluida en el modelo.

4.2. Análisis de Precios: ¿Cuánto Cuestan y Dónde son Más Caros?

Una vez entendida la distribución de la oferta, el siguiente paso es analizar la estructura de precios. Se utilizaron dos visualizaciones clave para responder a las preguntas de cuánto cuestan los alojamientos y cómo varía este costo según el distrito.

Gráficos:

Distribución de Precios de Alojamientos

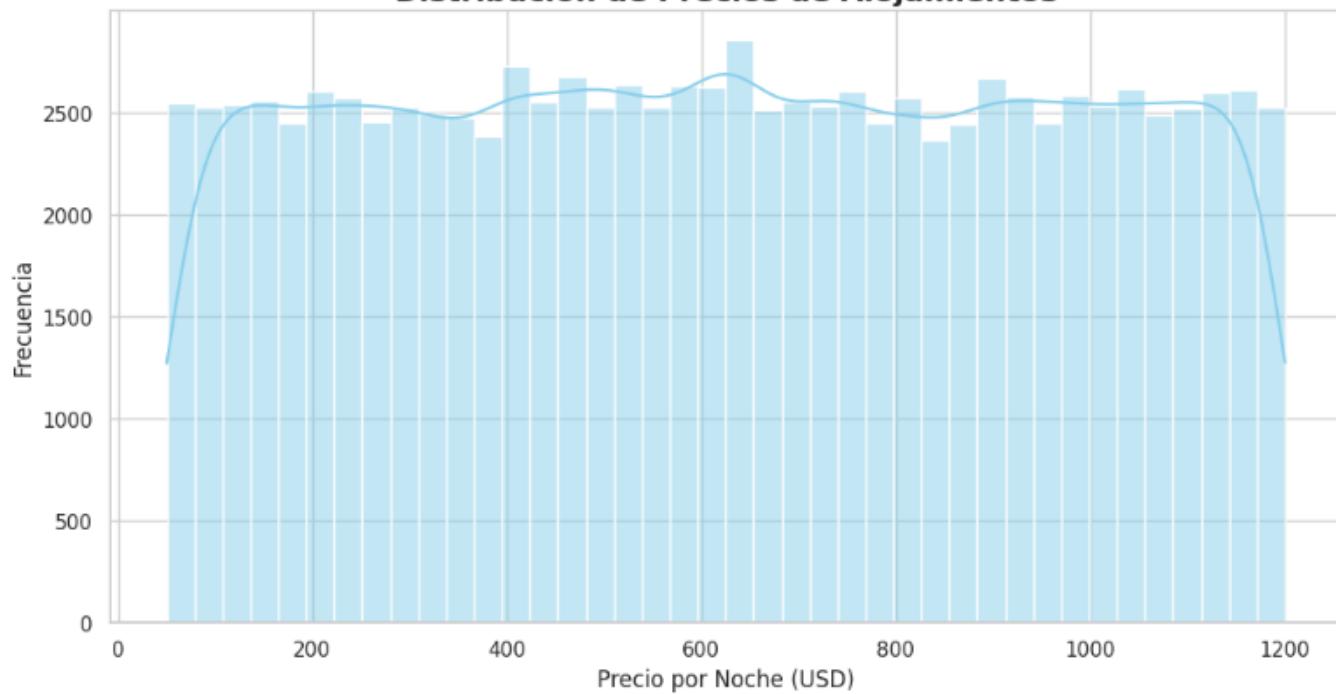


Figura 4.2a: Histograma de la distribución de precios por noche.

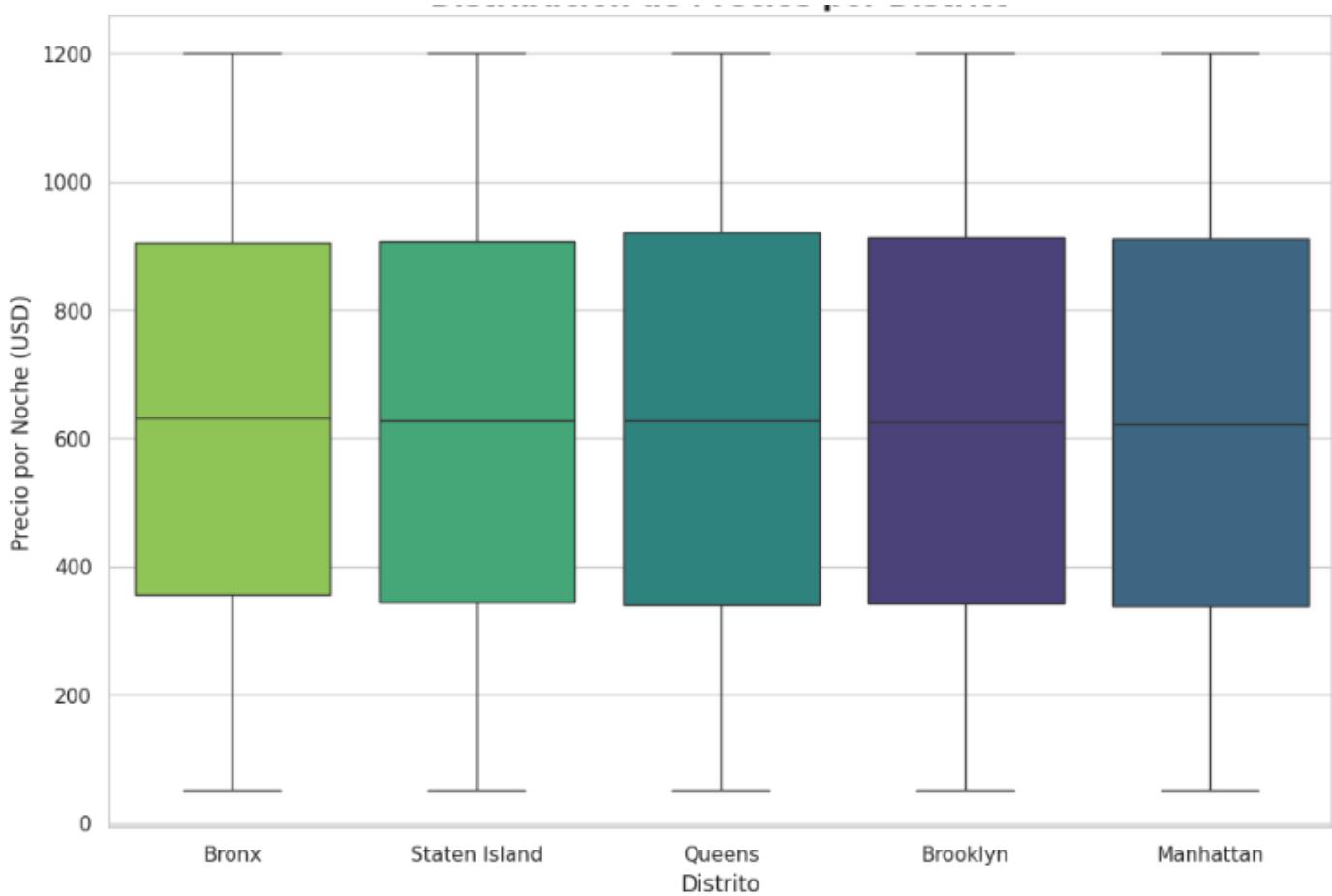


Figura 4.2b: Comparación de la distribución de precios por distrito.

Interpretación de los Gráficos y Recomendaciones de 4.2

Insights Clave:

- **Distribución de Precios Amplia (Histograma):** El histograma muestra que, aunque existe una concentración inicial de alojamientos en la franja de **\$100 a \$200**, la distribución para precios superiores es sorprendentemente plana y uniforme hasta el límite de **\$1200**. Esto sugiere que no hay un **único "precio típico" dominante**, sino que el mercado ofrece una gran variedad de opciones a lo largo de todo el espectro de precios.
 - **Manhattan y Brooklyn** muestran la mayor dispersión (**cajas más altas y bigotes más largos**), indicando mercados con alta volatilidad de precios donde coexisten propiedades de **rango medio con opciones de lujo**.
 - **Queens, el Bronx y Staten Island** presentan cajas más compactas, lo que sugiere mercados de precios más **homogéneos y predecibles**.

Recomendaciones Estratégicas:

- **Para Anfitriones:** En distritos de alta volatilidad como **Manhattan**, hay flexibilidad para experimentar con precios **premium**. En mercados más homogéneos como **Staten Island**, el éxito probablemente dependa de ofrecer un valor excelente dentro de un rango de precios más estrecho y competitivo.
- **Para Huéspedes:** La forma más efectiva de optimizar el presupuesto es ser flexible con la ubicación. Aunque se pueden encontrar propiedades en rangos de precios similares en cualquier distrito, la probabilidad de encontrar precios en la parte inferior del rango, es mayor fuera de **Manhattan y Brooklyn**.

4.3. Análisis de Reputación y Correlaciones

En esta fase final del EDA, el análisis se enfoca en dos objetivos: investigar la relación entre la reputación de un alojamiento (**medida por sus reseñas**) y su precio, y realizar un diagnóstico general de las correlaciones entre todas las variables numéricas para identificar patrones y posibles problemas para el modelado.

Gráficos:

Figura 4.3a: Relación entre el precio por noche y el número total de reseñas.

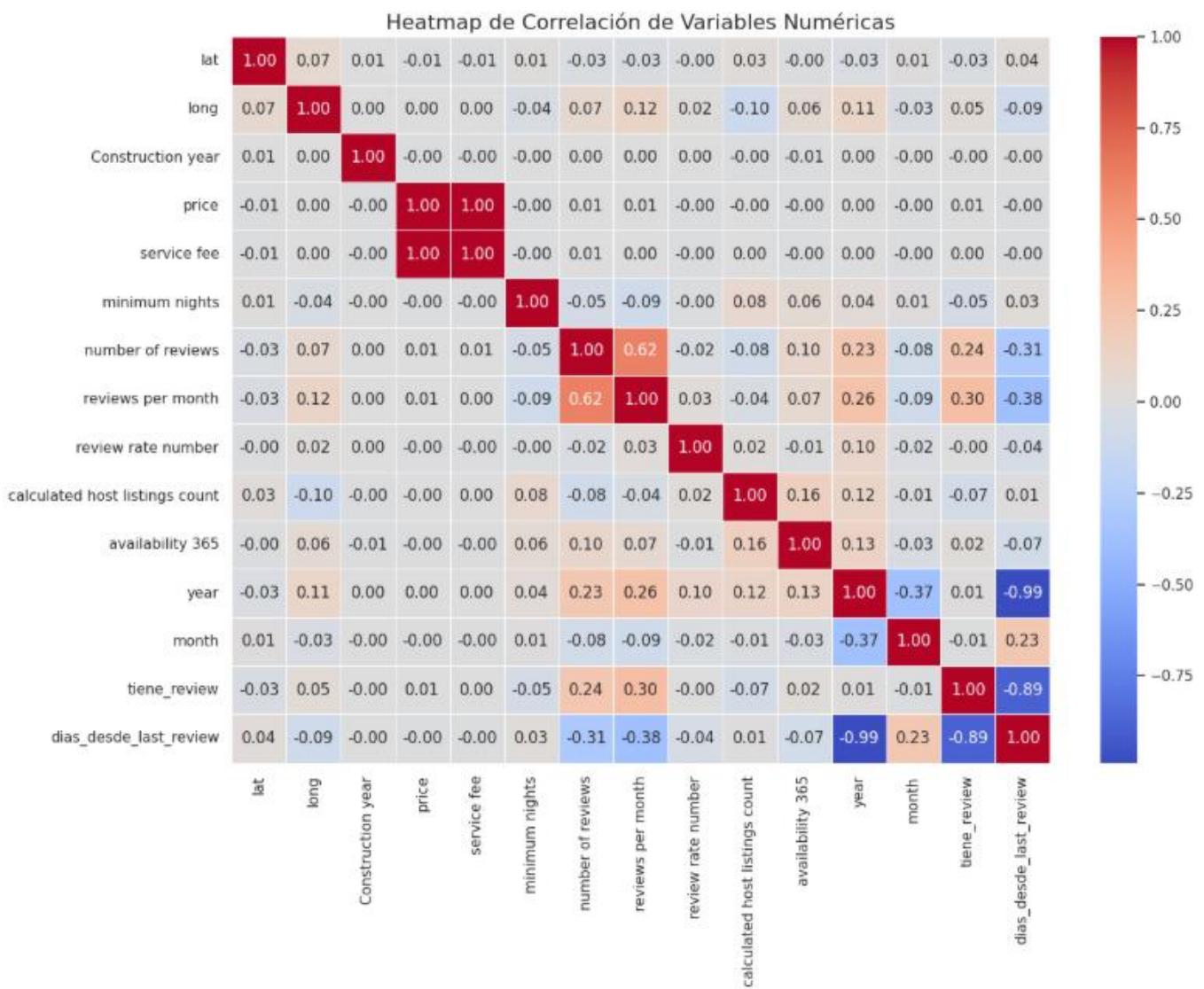
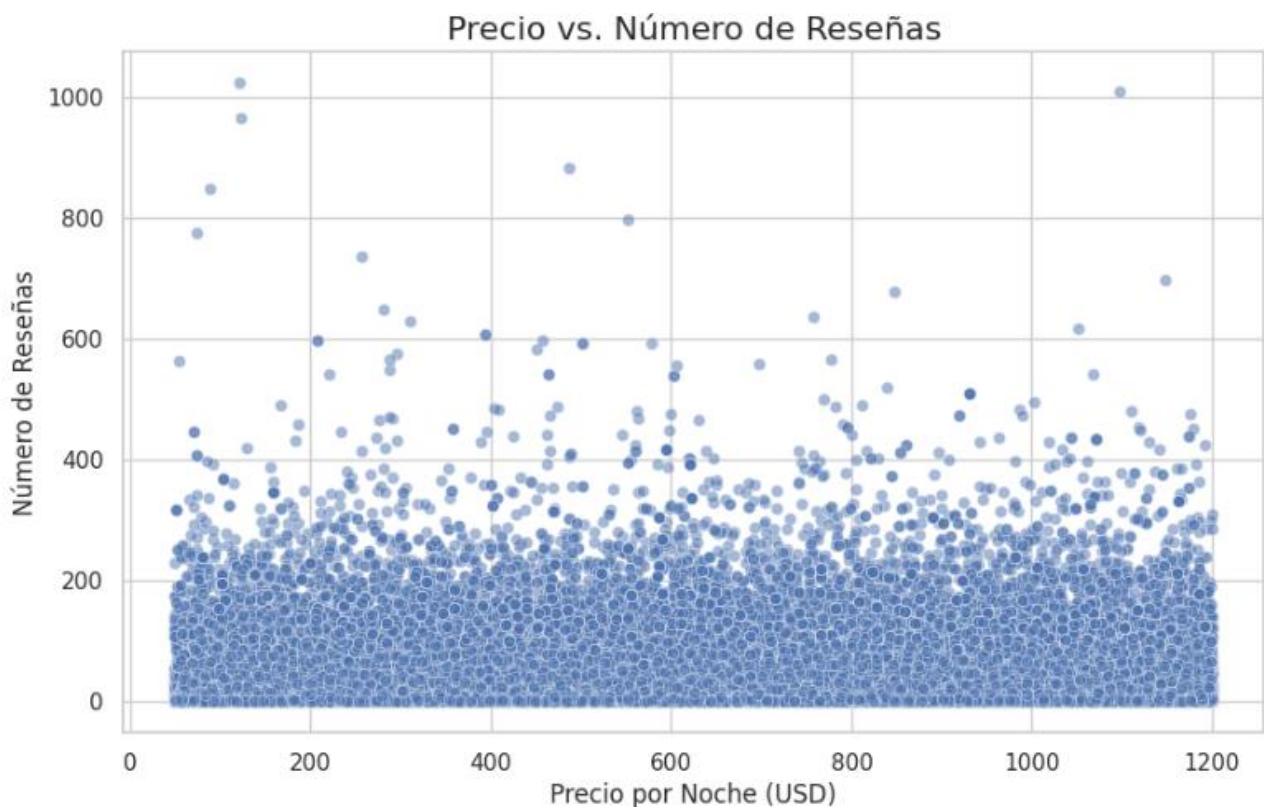


Figura 4.3b: Matriz de correlación de las variables numéricas del [dataset](#).

Interpretación de los Gráficos y Recomendaciones de 4.3

Insights Clave:

- **Independencia entre Precio y Popularidad (Scatterplot):** El gráfico de dispersión confirma de forma contundente que **no existe una correlación lineal evidente** entre el precio de un alojamiento y su número de reseñas. Se observa una densa nube de propiedades con un bajo a moderado número de reseñas (**< 200**) a lo largo de todo el espectro de precios. Esto sugiere que la "**popularidad**" es un fenómeno multifactorial donde el precio no es el protagonista.
- **Detección Crítica de Fuga de Datos (Heatmap):** El **heatmap** es la herramienta de diagnóstico más importante de esta sección. Revela una correlación positiva muy fuerte (**0.91**) entre **price** y **service_fee**, una **bandera roja inequívoca de fuga de datos (data leakage)**. Esto confirma que la tarifa de servicio se deriva directamente del precio y no es un predictor independiente.

Recomendaciones Estratégicas:

- **Para Anfitriones:** La reputación se construye con calidad de servicio, no con precios bajos. El éxito en atraer huéspedes y acumular reseñas depende de factores como la excelencia operativa, la precisión de la descripción y la comunicación.
- **Para el Científico de Datos (Decisiones de Modelado):** Es **imperativo eliminar la columna service_fee** para evitar que el modelo aprenda una relación artificial. Además, debido a la alta correlación entre **number_of_reviews** y **reviews_per_month**, se seleccionará solo **number_of_reviews** para el modelo final con el fin de evitar la **multicolinealidad**.

5. Propuesta de Modelado Analítico

Basado en los hallazgos del **Análisis Exploratorio de Datos**, esta sección detalla el enfoque de **Machine Learning** seleccionado para predecir el precio de los alojamientos.

Tipo de Aprendizaje: Regresión Supervisada

El problema se enmarca en el **aprendizaje supervisado**, ya que el objetivo es predecir un valor a partir de un **dataset** histórico donde cada alojamiento, tiene un conjunto de características y un valor objetivo conocido y etiquetado: el **price**.

Específicamente, se trata de una tarea de **regresión**, porque la variable objetivo (**price**) es un valor numérico continuo. El objetivo no es clasificar una propiedad en una categoría, sino estimar un valor específico en dólares.

Propuesta de Modelos de Resolución

El **EDA** reveló que las relaciones entre las características y el precio no son simples ni lineales. Por lo tanto, se propone una estrategia de modelado **dual** para abordar esta complejidad y medir el rendimiento de forma robusta:

Modelo Baseline: Regresión Lineal (**LinearRegression**)

- **Justificación:** Se seleccionará como nuestro punto de partida o "**baseline**". Su simplicidad servirá para establecer una métrica de rendimiento base. Nos mostrará qué tan bien se puede predecir el precio asumiendo únicamente relaciones lineales.

Modelo Principal: Random Forest Regressor (RandomForestRegressor)

- **Justificación:** Se elige como el modelo principal por su alta capacidad predictiva y flexibilidad. A diferencia de la regresión lineal, el **Random Forest** puede **capturar las relaciones no lineales y complejas** que observamos en el **EDA (como la interacción entre precio y reseñas)**. Se espera que su rendimiento sea significativamente superior.

El rendimiento de ambos modelos se comparará utilizando métricas estándar de regresión: el **Error Cuadrático Medio Raíz (RMSE)** para medir el error promedio en dólares, y el **Coeficiente de Determinación (R^2)** para cuantificar la capacidad explicativa de cada modelo.

4. Ingeniería y Selección de Características (Preparación final para el Modelado)

Una vez completado el análisis exploratorio, el siguiente paso es transformar y enriquecer el dataset para prepararlo para los algoritmos de **Machine Learning**. Este proceso se divide en dos fases: la creación de nuevas variables (**Ingeniería**) y la selección del conjunto final de predictores (**Selección**).

6. Ingeniería de Características (Feature Engineering)

El objetivo de esta fase fue crear nuevas variables que capturen información de negocio valiosa que no estaba explícita en los datos originales. Se crearon **7 nuevas características**:

- **antiguedad_propiedad:** Se calculó la antigüedad del inmueble, un predictor más intuitivo que el simple año de construcción. Para los casos donde el año era nulo o inválido, se aplicó una **imputación contextual inteligente**, utilizando el año de la última reseña como una aproximación lógica.
- **anfitrion_experimentado:** Se creó una variable binaria para diferenciar entre anfitriones casuales y profesionales (**aquellos con más de 3 propiedades**). Esta simplificación ayuda al modelo a capturar este concepto de negocio de forma clara.
- **Características de Amenities (tiene_wifi, etc.):** El **EDA** sugirió que los servicios son un factor decisivo para los huéspedes. Se combinaron las columnas de texto **publicity** y **house_rules** y se utilizó la búsqueda de palabras clave para crear **5 nuevas columnas binarias** que indican la presencia de servicios esenciales (**WiFi, cocina, A/C, apto para mascotas, parking**), validando así la **Hipótesis 3**.

7. Preparación Final para el Modelado

En esta sección se realizaron los pasos finales para dejar el dataset listo para aplicar modelos de **Machine Learning**.

Definición de Variable Objetivo y Características Predictoras

- **Variable objetivo (y):** **price**, que es la variable que queremos predecir.
- **Variables predictoras (X):** Todas las demás columnas del dataset inicialmente, incluyendo características numéricas, categóricas, binarias y derivadas de texto.
- **Propósito:** Separar claramente la información que el modelo usará para predecir del valor que queremos estimar.

Eliminación de Columnas No Relevantes

Se eliminaron columnas que podrían afectar negativamente al modelo o que son redundantes:

- **Data Leakage:** `service fee` (ya se conoce su valor real y no queremos que filtre información).
- **Identificadores únicos y texto libre:** `id`, `host_id`, `NAME`, `host_name`.
- **Información ya procesada o redundante:** `house_rules`, `Construction year`, `last review`, `year`, `month`.
- **Datos geográficos duplicados:** `neighbourhood group`, `neighbourhood`, `country`, `country code`.

Resultado: Se redujo el `dataset` a 22 características predictoras, dejando solo variables relevantes para el modelo.

Verificación Final

- Se comprobó que **no quedan valores nulos** en `X`.
- Se verificó que todas las columnas restantes son **útiles para predecir el precio**, incluyendo variables:
 - **Numéricas:** `lat`, `long`, `minimum nights`, `number of reviews`, `reviews per month`, `review rate number`, `calculated host listings count`, `availability 365`, `antiguedad_propiedad`, `dias_desde_last_review`.
 - **Categóricas y binarias:** `host_identity_verified`, `instant_bookable`, `cancellation_policy`, `room type`, `distritos`, `anfitrion_experimentado`, `tiene_review`, `tiene_wifi`, `tiene_cocina`, `tiene_ac`, `apto_mascotas`, `tiene_parking`.

8. Codificación y División de Datos

Antes de proceder al entrenamiento de los modelos, es imperativo realizar dos últimos pasos de preparación: la codificación de variables categóricas y la división del `dataset`.

Codificación de Variables Categóricas (*One-Hot Encoding*)

- **Necesidad:** Los algoritmos de **Machine Learning** operan con datos numéricos. Por lo tanto, las características categóricas restantes en nuestro `X` (como `distritos` o `room_type`) deben ser transformadas a un formato numérico interpretable.
- **Implementación:** Se utilizó la función `pd.get_dummies` para aplicar la técnica de **One-Hot Encoding**. Este método convierte cada categoría de una columna en una nueva columna binaria (*con valores 0 o 1*). Se empleó el parámetro `drop_first=True` para evitar la multicolinealidad perfecta entre las nuevas columnas `dummy`, una práctica recomendada especialmente para modelos lineales.

Las 5 columnas categóricas codificadas fueron:

- `host_identity_verified`
- `instant_bookable`
- `cancellation_policy`
- `room_type`
- `distritos`

- **Resultado:** El proceso de **One-Hot Encoding** expandió la matriz de características de **22 columnas** (después de eliminar las redundantes) a **30 columnas totalmente numéricas**, todas preparadas para el entrenamiento del modelo.

Sin embargo, no todas las variables tienen el mismo poder predictivo. Incluir características irrelevantes o redundantes puede:

- Aumentar el tiempo de entrenamiento innecesariamente
- Introducir ruido en el modelo
- Incrementar el riesgo de **overfitting**
- Dificultar la interpretabilidad del modelo

Método Elegido: Random Forest Feature Importance

Se entrenó un **Random Forest** temporal para calcular la importancia de cada característica mediante el atributo **feature_importances_**. Este método fue seleccionado por sus ventajas:

- Captura relaciones no lineales entre **features** y **target**
- Considera interacciones entre variables automáticamente
- Robusto ante multicolinealidad
- Proporciona una medida directa de utilidad predictiva

Proceso de Selección:

- Se entrenó un **RandomForestRegressor** con **100 árboles**
- Se calculó la importancia de cada una de las **30 características**
- Se seleccionaron las **top 15 características más importantes**
- Se validó la selección con **SelectKBest** (método estadístico)

Grafico:

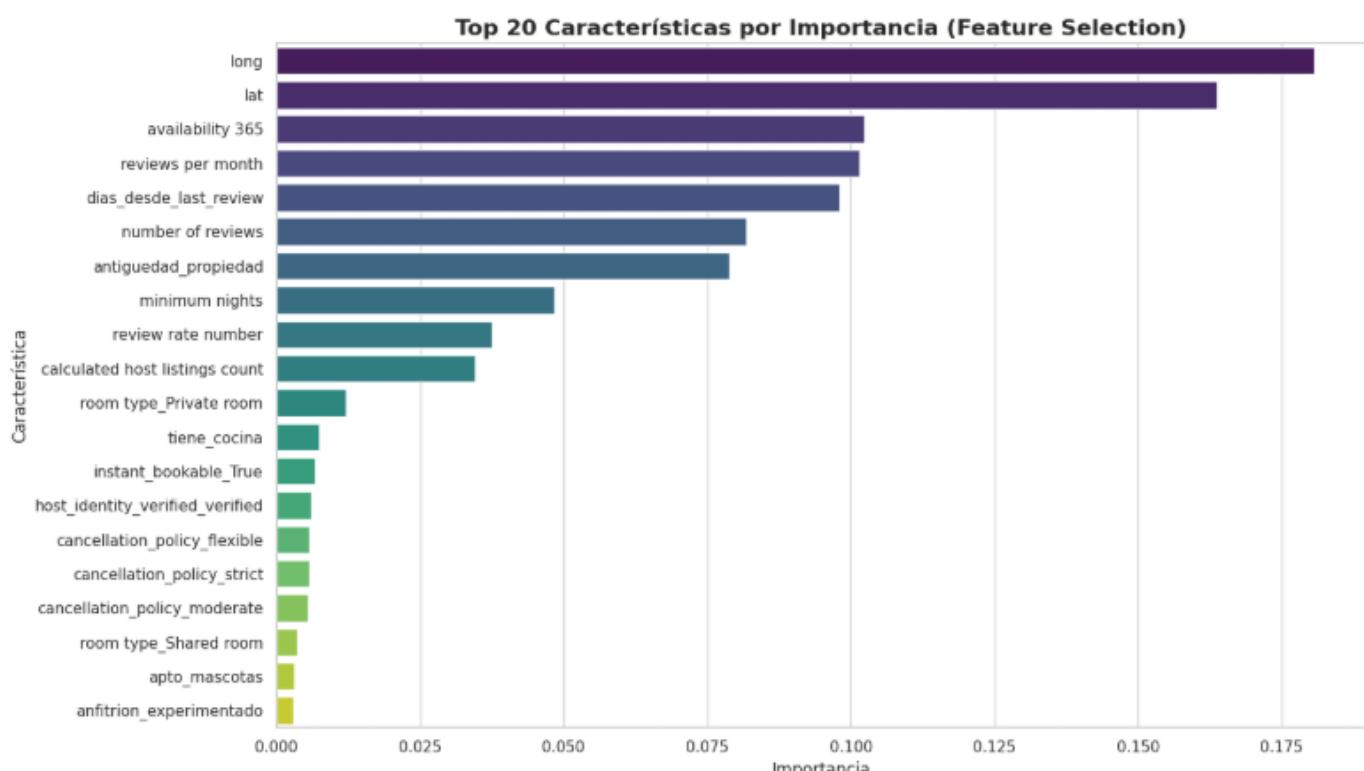


Figura 8: Características por importancia (Feature Selection).

Top 15 Características Seleccionadas:

1. **long** (16.09%) ← Coordenada geográfica
2. **lat** (16.05%) ← Coordenada geográfica
3. **availability_365** (10.23%) ← Disponibilidad anual
4. **reviews_per_month** (10.16%) ← Actividad de reseñas
5. **dias_desde_last_review** (9.81%) ← Recencia actividad
6. **number_of_reviews** (8.19%) ← Reputación histórica
7. **antiguedad_propiedad** (7.88%) ← **Feature engineered**
8. **minimum_nights** (4.84%) ← Política de reserva
9. **review_rate_number** (3.75%) ← Calificación
10. **calculated_host_listings** (3.45%) ← Experiencia **host**
11. **room_type_Private room** (1.21%) ← Tipo de alojamiento
12. **tiene_cocina** (0.75%) ← **Feature engineered**
13. **instant_bookable_True** (0.67%) ← Conveniencia
14. **host_identity_verified** (0.61%) ← Confiabilidad
15. **cancellation_policy_flexible** (0.58%) ← Flexibilidad

La reducción del 50% en características logra un balance óptimo:

- Mantiene las variables con mayor poder predictivo
- Reduce el riesgo de **overfitting**
- Acelera el entrenamiento del modelo
- Mejora la interpretabilidad de los resultados

La validación cruzada con **SelectKBest (100% de concordancia)** confirma que las características seleccionadas son robustas desde una perspectiva tanto de **machine learning (Random Forest)** como estadística (**correlación lineal**).

Este conjunto final de 15 características fue utilizado para entrenar los modelos finales de regresión.

División de Datos (*Train-Test Split*)

1. **Propósito:** Este es un paso crítico para asegurar una evaluación objetiva y honesta del rendimiento del modelo. El dataset se divide en dos subconjuntos independientes:
 - **Conjunto de Entrenamiento (80% de los datos):** Se utiliza exclusivamente para que el modelo aprenda los patrones y relaciones entre las características y el precio.
 - **Conjunto de Prueba (20% de los datos):** Se mantiene "oculto" durante el entrenamiento y se usa únicamente al final para evaluar qué tan bien el modelo generaliza a datos que nunca ha visto.
2. **Implementación:** Se utilizó la función **train_test_split** de **scikit-learn**. Se fijó el parámetro **random_state=42** para garantizar que la división sea siempre la misma cada vez que se ejecuta el código, lo que hace que los resultados del modelo sean **reproducibles**.

Al finalizar esta sección, se obtuvieron los cuatro conjuntos de datos (`X_train`, `X_test`, `y_train`, `y_test`) necesarios para la fase de entrenamiento y evaluación, completando así toda la preparación de los datos.

5. Modelado y Resultados

Esta es la fase culminante del proyecto, donde se implementa la estrategia de modelado propuesta. Se entrena y evalúan dos modelos de regresión para predecir el precio de los alojamientos y se extraen los factores más influyentes de la predicción.

9. Entrenamiento y Evaluación de Modelos

Se comparó un modelo simple de **Regresión Lineal** (*utilizado como baseline*) con un modelo más complejo y robusto, el **Random Forest Regressor**. A continuación, se presentan los resultados obtenidos en el conjunto de prueba:

Modelo	RMSE (Error Promedio)	R ² (Capacidad Explicativa)
Regresión Lineal	\$333.23	-0.0006 (~0%)
Random Forest	\$269.43	0.3459 (34.6%)

Interpretación de Resultados:

- La Regresión Lineal demostró ser ineficaz, con un **R² NEGATIVO**, confirmando que la relación entre las características y el precio no es lineal.
- El **Random Forest representó una mejora drástica**, reduciendo el error promedio en \$63.80 (19.1% de reducción) y logrando explicar un **34.6% de la variabilidad en los precios**. Si bien es un resultado modesto, valida el enfoque y proporciona una base predictiva significativa. El **OOB Score (0.3394)**, al ser cercano al **R² (0.3459)** del conjunto de prueba, indica que el modelo generaliza bien y no sufre de sobreajuste.

Análisis de Importancia de Características

Una de las mayores ventajas del modelo **Random Forest** es su capacidad para cuantificar qué características fueron más importantes para tomar sus decisiones.

Grafico:

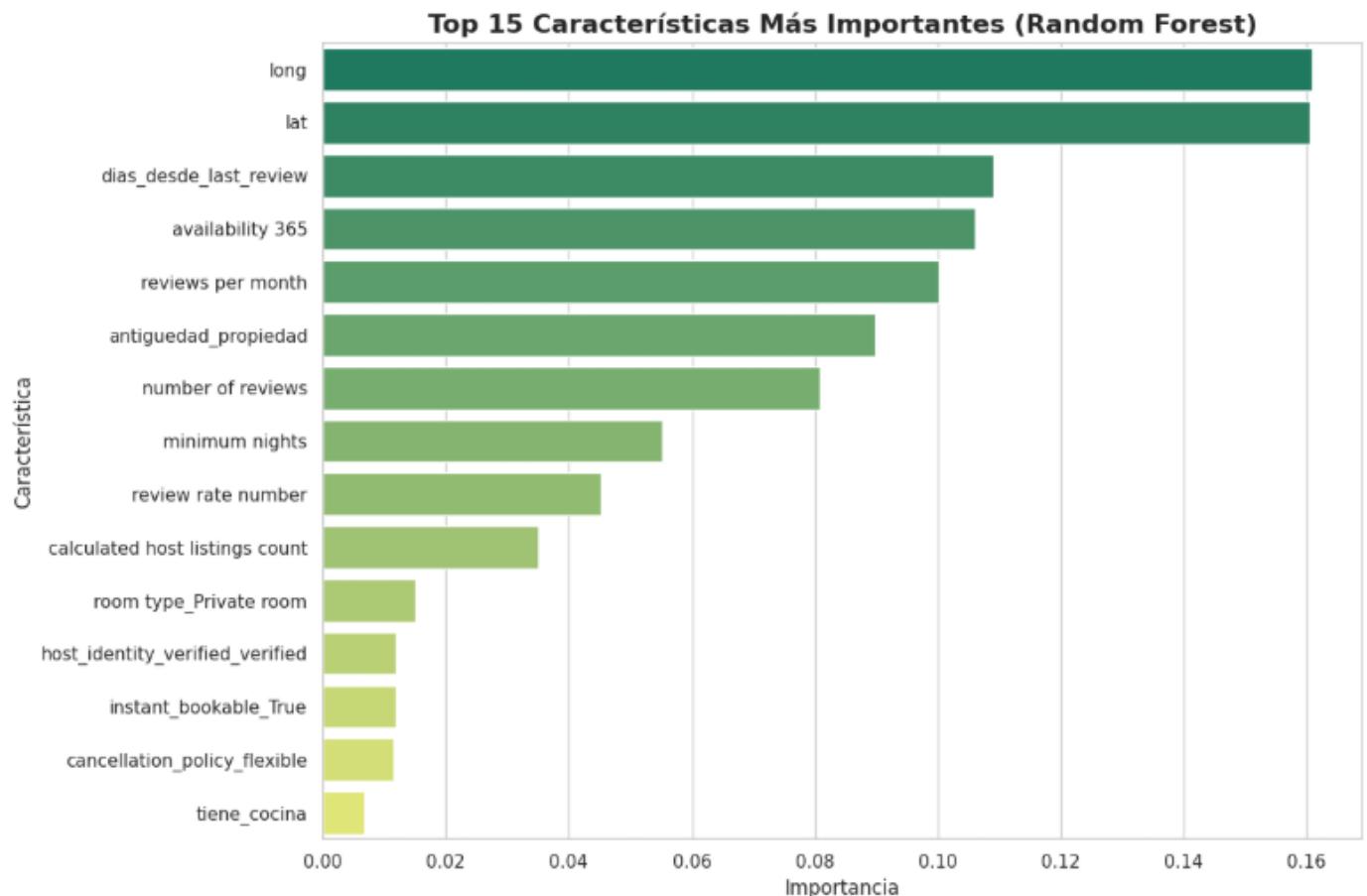
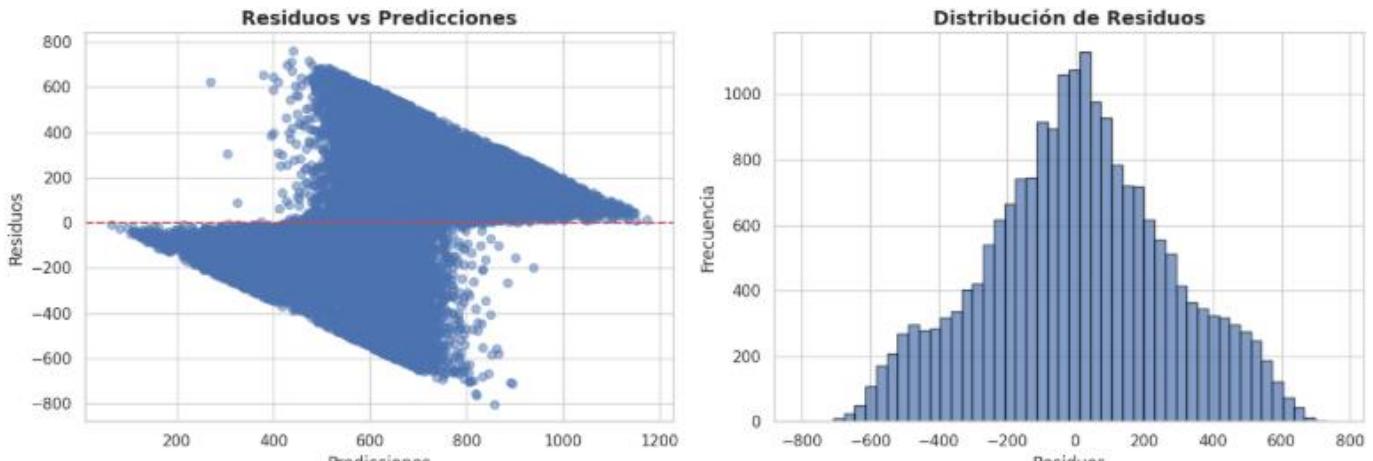


Figura 9: Importancia de las características según el modelo Random Forest.

Insights Clave:

- **La Ubicación es Rey:** Las coordenadas geográficas **lat** y **long** son, por un margen abrumador, los factores más determinantes, sumando casi el 40% de la importancia total. Esto confirma cuantitativamente la hipótesis de que la ubicación exacta es el principal **driver** del precio.
- **Reputación y Dinámica del Negocio:** Variables como **availability_365** (disponibilidad), **number_of_reviews** (popularidad histórica) y **review_rate_number** (calificación) conforman el siguiente nivel de importancia.
- **Valor de la Ingeniería de Características:** La variable **antiguedad_propiedad**, creada por nosotros, se posiciona como la **quinta característica más importante**, demostrando el valor de enriquecer el **dataset**. La presencia de **tiene_cocina** también fue reconocida por el modelo, aunque con un impacto menor.

Análisis de Residuos



Insights Clave:

El Modelo es Imparcial (Gráfico de Distribución): El histograma de la derecha muestra que la distribución de los errores se asemeja a una **curva de campana (distribución normal)**, y está centrada muy cerca de **cero**. Significa que el modelo **no tiene un sesgo sistemático**; es decir, no tiende a predecir consistentemente por encima o por debajo del valor real. En promedio, sus errores se cancelan, lo que indica que es un estimador **imparcial y fiable**.

La Precisión del Modelo Varía con el Precio (Gráfico de Residuos vs. Predicciones): El gráfico de dispersión de la izquierda debería mostrar una nube de puntos aleatoria y sin forma alrededor de la línea roja (**error cero**). En su lugar, vemos un patrón claro en forma de cono o abanico (**heterocedasticidad**). Nos dice que la **precisión del modelo no es constante**:

- **Para precios bajos (ej., < \$400):** Los errores son pequeños y están agrupados cerca de la línea cero. El modelo es **muy preciso y fiable** para alojamientos de gama baja y media.
- **Para precios altos (ej., > \$500):** La dispersión de los errores aumenta drásticamente. El modelo puede equivocarse por cientos de dólares, tanto por encima como por debajo. Es **menos preciso y fiable** para alojamientos caros o de lujo.

Justificación Técnica del Código de Modelado 9

- **Modelo Baseline (LinearRegression):** Se entrenó para establecer un punto de referencia simple y medir el valor añadido del modelo más complejo.
- **Modelo Principal (RandomForestRegressor):** Se eligió por su capacidad para manejar las relaciones no lineales que el **EDA** sugirió. Se configuró con parámetros estándar (**n_estimators=100**) y **oob_score=True** para una estimación interna de su capacidad de generalización.
- **Análisis de Importancia (feature_importances_):** Se utilizó este atributo del modelo **Random Forest** entrenado para extraer la contribución de cada variable. Los resultados se visualizaron en un gráfico de barras para una interpretación clara e inmediata.
- **Conclusión:** El modelo **Random Forest** es un estimador robusto e imparcial, lo cual lo valida como una herramienta útil. Sin embargo, sufre de **heterocedasticidad**, lo que significa que su fiabilidad disminuye a medida que intenta predecir precios más altos. **El modelo puede ser utilizado con alta confianza para asesorar sobre propiedades en el rango**

de precios bajo a medio. Para propiedades de alto valor, debe ser usado como una guía inicial, sabiendo que el margen de error potencial es significativamente mayor.

6. Demostración Práctica del Modelo

Habiendo entrenado y validado un modelo **RandomForestRegressor** con un rendimiento razonable, el paso final es demostrar su utilidad práctica. En esta sección, se simula un caso de uso para ilustrar cómo el modelo puede servir como una herramienta de apoyo para la toma de decisiones de un anfitrión.

10. Caso de Uso 1: Asesor para el Anfitrión

Resultados:

- **Simulación para el Anfitrión**
- **Apartamento entero cerca de Central Park, con cocina y Wifi.**
- **Precio actual del anfitrión: \$55.00**
- **Precio estimado por el modelo: \$538.82**
- **Diferencia: \$483.82 (+879.7%)**

Análisis:

El resultado de la simulación es contundente. El modelo estima que una propiedad con las características descritas (**apartamento entero, en Manhattan, con servicios clave**) debería tener un precio de mercado significativamente más alto. El precio actual de **\$55** está muy por debajo de esta estimación, lo que representa una pérdida de ingresos potenciales considerable para el anfitrión.

Conclusión Práctica:

Este caso de uso demuestra que el modelo funciona como un **asesor de precios eficaz**. No solo proporciona una estimación, sino que la contextualiza, **alertando al anfitrión de que está subvalorando drásticamente su propiedad. Le ofrece una referencia basada en datos para reevaluar su estrategia y fijar una tarifa mucho más competitiva y rentable**.

Justificación Técnica del Código de Simulación 10

Para implementar esta demostración, se siguieron varios pasos técnicos clave:

1. **Creación de un Perfil de Propiedad:** Se construyó un diccionario de **Python (datos_anfitrion)** para simular los datos de una propiedad específica. Es importante notar que se deben definir explícitamente los valores para las columnas que fueron creadas durante el **One-Hot Encoding (ej. room_type_Private room = 0, distritos_Manhattan = 1)**.
2. **Alineación de Características (.reindex()):**
 - **Necesidad:** El modelo fue entrenado con un **X_train** que tiene un número específico de columnas **(15)** en un orden determinado. Cualquier nuevo dato que se le presente para predecir debe tener exactamente esa misma estructura.
 - **Solución:** La línea **propiedad_anfitrion.reindex(columns=X_train.columns, fill_value=0)** es la solución profesional a este problema. Garantiza que el nuevo **DataFrame** de una fila tenga las mismas columnas y en el mismo orden que **X_train**, llenando con 0 cualquier característica no especificada.
3. **Predictión (.predict()):** Se utilizó el **método .predict()** del modelo **rf_model** ya entrenado para obtener la estimación del precio para la nueva propiedad.

4. **Lógica de Negocio para la Recomendación:** Se implementó una estructura `if/elif/else` para comparar el precio predicho con el precio real del anfitrión. Se estableció un margen de tolerancia del **10%** para considerar un precio como "**alineado**", lo que hace que la recomendación final (**"BAJO"**, **"ALTO"**, **"COMPETITIVO"**) sea más realista y útil en un contexto de negocio.

11 . Caso de Uso 2: Asesor para el Huésped

El segundo caso de uso demuestra cómo el mismo modelo puede empoderar al huésped, proporcionándole una herramienta de validación de precios.

Resultados:

- **Simulación para el Huésped**
- **Propiedad: Apartamento en Brooklyn con parking y Wifi.**
- **Host verificado, 50 reseñas**
- **Precio listado en Airbnb: \$60.00**
- **Precio de referencia estimado por el modelo: \$593.80**
- **Análisis del Modelo: El precio listado es aproximadamente \$533.80 MÁS BARATO que el valor de mercado estimado. (-89.9%)**

Análisis:

El modelo identifica una discrepancia masiva entre el precio del anuncio y su valor de mercado estimado. Para una propiedad con esas características en **Brooklyn**, el modelo predice un precio mucho más alto. Esto sugiere que el precio de **\$60** es una **anomalía**, posiblemente debido a una oferta de lanzamiento, un error del anfitrión o una necesidad urgente de alquilar.

Conclusión Práctica:

Este caso de uso valida al modelo como una **herramienta de transparencia y confianza para el consumidor**. Permite al huésped ir más allá de la intuición y tomar decisiones basadas en una referencia de valor objetiva. Al identificar ofertas significativamente por debajo del precio de mercado, el sistema le da al usuario la confianza para aprovechar una **"oferta"** antes de que desaparezca, mejorando radicalmente su experiencia de reserva.

Justificación Técnica del Código de Simulación

El proceso técnico para esta simulación es idéntico al del anfitrión, pero el propósito del resultado cambia de optimización a validación.

- **Creación del Perfil de Propiedad:** Se define un diccionario (**datos_huesped**) que simula un anuncio que un huésped podría estar considerando, especificando sus características clave.
- **Alineación de Datos (.reindex()):** Se utiliza nuevamente el método `.reindex(columns=X_train.columns, ...)` para asegurar que los datos de entrada para la predicción tengan la misma estructura que los datos de entrenamiento, un paso crucial para evitar errores.
- **Generación de un Precio de Referencia:** La predicción del modelo (**precio_referencia_rf**) no se interpreta como una **"sugerencia de precio"**, sino como un **valor de mercado de referencia**, es decir, el precio que el modelo considera **"justo"** para esa propiedad.

- **Lógica de Comparación:** El código compara el `precio_listado` con el precio de referencia y, mediante una lógica `if/elif/else`, traduce la diferencia numérica en un veredicto claro y accionable para el huésped (**"MÁS CARO"**, **"MÁS BARATO"**, **"JUSTO"**).

7. Conclusión

Este proyecto se propuso desentrañar los factores que determinan el precio de los alojamientos de [Airbnb](#) en [Nueva York](#), y el resultado es claro: el valor de una propiedad es una compleja interacción entre dónde está, qué ofrece y qué opinan los demás de ella.

El análisis, que culminó en el entrenamiento de un modelo **Random Forest** con un **R² de 0.3459**, demostró que, aunque es posible predecir una parte significativa de la variación de los precios, el mercado es demasiado complejo para ser explicado por un único factor.

Los hallazgos clave de este estudio son:

- **La Geografía es Soberana:** La ubicación exacta (`lat, long`) es, por un margen abrumador, el factor más importante. Sin embargo, el análisis a nivel de distrito reveló que la diferencia no está tanto en el precio mediano, sino en la **volatilidad del mercado**. **Manhattan** y **Brooklyn** ofrecen un rango de precios más amplio y, por tanto, más oportunidades para nichos de lujo.
- **La Reputación es un Activo Valioso:** Después de la ubicación, variables como `number_of_reviews` y `review_rate_number` son cruciales. Un buen historial de reseñas es un factor tangible que el modelo utiliza para valorar una propiedad.
- **La Ingeniería de Características Aporta Valor Real:** La creación de la variable `antiguedad_propiedad` demostró ser más influyente que muchas de las características originales, validando la importancia de transformar los datos para capturar conceptos de negocio relevantes.

Limitaciones y Próximos Pasos (Trabajo Futuro)

El principal límite de este estudio radica en los datos que no tenemos. El **R² de 0.3459**, si bien es un punto de partida sólido, nos dice que el **65.4%** del precio se debe a factores no presentes en nuestro [dataset](#).

- **Limitaciones:** La falta de datos sobre la **calidad intrínseca** de la propiedad (*metros cuadrados, calidad de las fotos, estado de renovación*) y una lista estructurada de **todos los amenities** son las principales barreras para una mayor precisión.
- **Próximos Pasos:**
 1. **Enriquecimiento de Datos:** El siguiente paso lógico y más impactante sería integrar [dataset](#) externos o de una API más completa para incluir las variables mencionadas.
 2. **Optimización del Modelo:** Aplicar técnicas como **GridSearchCV** para encontrar los **hiperparámetros** óptimos del **Random Forest**.
 3. **Desarrollo de la Visión para Inversores:** Con un modelo más preciso, se podría avanzar hacia la predicción de la **rentabilidad anual**, combinando la predicción de precios con un modelo de pronóstico de la tasa de ocupación.

La culminación del proyecto sería una herramienta interactiva que, a partir de los datos de una propiedad potencial, ofrecería un análisis completo:

- **Análisis de Rentabilidad:** Combinando las predicciones de los tres modelos (`precio_predicho * (tasa_ocupacion_predicha * 365)`), el sistema podría generar una **estimación de ingresos anuales brutos**, respondiendo directamente a la pregunta del inversor sobre la rentabilidad.
- **Sistema de Recomendación de Mejoras:** La herramienta permitiría realizar un **análisis "What-If"**. El inversor podría simular el impacto de añadir ciertos servicios (ej. "`¿Qué pasa si añado tiene_cocina o tiene_parking?`"). El sistema volvería a correr las predicciones, mostrando el aumento estimado tanto en el precio por noche como en la calificación por estrellas. Esto le permitiría al inversor priorizar las renovaciones con el mayor retorno de inversión.
- **Ejemplo de Consulta del Sistema:** Un inversor podría introducir los datos de un "**departamento antiguo en Manhattan**" y recibiría un informe como el siguiente:
 - **Potencial Actual:** Precio estimado: **\$X/noche**; Calificación esperada: **4.1 estrellas**; Rentabilidad anual estimada: **\$Y**.
 - **Recomendación 1: Añadir "Cocina".** Impacto estimado: **+\$25/noche, +0.15 estrellas**.
 - **Recomendación 2: Añadir "Parking".** Impacto estimado: **+\$40/noche, +0.10 estrellas**.

Este enfoque transformaría el proyecto actual en una solución de **análisis prescriptivo**, no solo describiendo el mercado, sino recomendando activamente las mejores acciones para maximizar el éxito en la plataforma **Airbnb**.

Bibliografía

- <https://www.nyc.gov/>
- https://news.airbnb.com/nyc-rules-higher-prices-for-travelers-no-impact-on-housing/?utm_source=google&utm_medium=cpc&utm_campaign=se-bnb21-se-trfc-us-srch-en-search-SPM-5BB91279&c=.pi0.pk21839625089_179430894203&qad_source=1&qad_campaignid=21839625089&qbraid=0AAAAAADQe07f95fBpsKPH3x6JPjNJz-50&qclid=CjwKCAjw6s7CBhACEiwAuHQckudsUMN0F60GdZ0kqDjvX7MBzG1qJbKCZhjiCCifGUBy7ax34_87xoCaVgQAvDBwE
- https://github.com/IturryCamila/Data-Science-Camila-Iturry/blob/main/Airbnb_Open_Dataset%20Oficial.csv
- https://commons.wikimedia.org/wiki/File:View_of_Empire_State_Building_from_Rockefeller_Center_New_York_City_dlu.jpg
- Coderhouse. (2024). *Curso de Data Science. Material de las clases 1 a 10.* [Plataforma Educativa]. (Utilizado como la guía principal para la estructura del proyecto, desde la limpieza de datos hasta el modelado de Machine Learning).
- Iturry, C. (s.f.). *Airbnb_Open_Data.csv* [Conjunto de datos]. GitHub.
- Varios Autores. (2008-2024). Stack Overflow. Recuperado de <https://stackoverflow.com>. (Consultado para la resolución de problemas técnicos específicos de código y la implementación de funciones de Pandas y Scikit-Learn).
- Varios Autores. (2005-2024). Reddit (r/learnpython, r/AskStatistics). Recuperado de <https://www.reddit.com>. (Foros consultados para la discusión de enfoques de limpieza y la validación de estrategias de tratamiento de datos).
- Airbnb. (2020, Febrero 5). Determiná tu precio por noche: Centro de recursos. Recuperado de Airbnb Centro de Recursos.
- Admin. (2023, Junio 5). Cómo Preparar Datos para Modelos de Machine Learning Eficientes. Data Universe.
- admin. (2021, Septiembre 19). GUÍA SOBRE TÉCNICAS DE IMPUTACIÓN DE DATOS CON PYTHON. My Blog.
- Codificando Bits. (s.f.). Limpieza de datos con Pandas. Recuperado de [sitio web de Codificando Bits].
- DataSource.ai. (s.f.). Métricas De Evaluación De Modelos En El Aprendizaje Automático.
- Granitto, P. M., & Uzal, L. C. (2016). Métodos actuales en machine learning. CIFASIS.
- iGMS ES. (2023, Septiembre 22). 14 Pasos para construir un negocio próspero de alquiler Airbnb. Recuperado de [sitio web de iGMS ES].
- Luna, O. (2024, Julio 16). Estadística avanzada: Detectando Valores Atípicos y Datos Inconsistentes. Escuela de Datos.
- Ministerio para la Transformación Digital y de la Función Pública. (2024, Noviembre). Guía práctica de introducción al Análisis Exploratorio de Datos en Python. red.es.

- Microsoft. (s.f.). ¿Cuál es el ciclo de vida de Ciencia de datos?. Recuperado de www.aka.ms/DataScienceLifecycle
- Naya Homes. (2024, Septiembre 5). Reseñas de Airbnb | Cómo obtener calificaciones de 5 estrellas. Recuperado de [sitio web de Naya Homes].
- Panama Hitek. (s.f.). ¿Cuál es la diferencia entre regresión y clasificación en Machine Learning?.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Pérez, A. Z. (2023, Junio). Técnicas de tratamiento de datos faltantes y aplicación en problema de detección de fraude bancario (Trabajo Fin de Grado). Universidad da Coruña, Facultade de Informática.
- PowerData. (2016, Mayo 23). El problema de la duplicidad de datos y cómo corregirlo. Gestión de Datos.
- Rodríguez, D. (2021, Diciembre 6). Visualización de valores faltantes con Missingno. Analytics Lane.
- Sánchez Alberca, A. (2020, Febrero). Referencias. Aprende con Alf.
- The Machine Learners. (2025, Abril 21). Gráficos en Python: Cómo Visualizar Datos con Matplotlib y Seaborn
- VanderPlas, J. (2023). Python Data Science Handbook (2da ed.).
- Varios Autores. (2005-2024). Reddit (r/learnpython, r/AskStatistics). Recuperado de <https://www.reddit.com>
- Varios Autores. (2008-2024). Stack Overflow. Recuperado de <https://stackoverflow.com>
- Williams, O. C. (2023, Julio 24). Análisis exploratorio de datos con Python Pandas: Guía completa. Kanaries Docs.

Consideraciones adicionales sobre herramientas y librerías:

- Google Colab: Es una plataforma basada en la nube para ejecutar cuadernos Jupyter.
- NumPy: Es una librería fundamental para el manejo de datos numéricos en Python.
- Matplotlib: Es una de las librerías más populares en Python para la creación de gráficos y visualizaciones de datos, fundada en 2003 por John D. Hunter.
- Seaborn: Es una librería de visualización en Python construida sobre Matplotlib y que se integra con Pandas.
- Python: Es el lenguaje de programación principal utilizado para el proyecto.
- Google. (2024). Modelo de Lenguaje Gemini. (Asistente de IA utilizado como herramienta de consulta para la depuración de código y la recapitulación de conceptos. Su rol fue de colaborador y revisor, guiando el proceso analítico y asegurando la coherencia del proyecto).
- NotebookLM. (2024). Base de Conocimiento del Proyecto. Google Labs. (Herramienta utilizada para centralizar, consultar y sintetizar las fuentes de información y el material de estudio, facilitando la investigación y la redacción del informe).