KEEPCODING
Tech School

Informe Proyecto Final

Análisis del mercado de alojamientos de Madrid a partir de datos extraídos de Airbnb

Nombre del grupo:
Distrito 13

Integrantes:
Marina González
Valeria Gonzalez
María Ojie
Sarasuadi Vargas
María Lucía Vicentin

Este informe tiene el objetivo de detallar cada uno de los pasos que llevó a cabo el equipo para crear un modelo de regresión lineal a partir de un dataset que tuvo pasar una serie de transformaciones con el fin de extraer información valiosa. La fuente de datos consiste en un archivo .csv que recoge información sobre una serie de inmuebles publicados en la conocida plataforma digital de alojamientos Airbnb. Para la realización del proyecto, se planteó un caso hipotético de negocio con la intención de crear una serie de suposiciones iniciales y preguntas que se intentaron responder con las diversas herramientas de análisis y visualización de datos a lo largo del proyecto.

Caso de empresa

Averiguar cuáles son las variables más relevantes para desarrollar una herramienta predictiva mediante la cual un nuevo anfitrión puede introducir las características de su inmueble y obtener un precio pertinente.

Para ello, se plantearon las siguientes preguntas:

- ¿Cómo influye el barrio en el precio de los inmuebles?
- ¿Hay alguna relación entre el vecindario y la política de cancelación?
- ¿Cómo influye la cantidad de inmuebles acumulados por host en los precios?
- ¿Los diferentes tipos de inmueble son igual de caros en cualquier barrio de Madrid?
- ¿Hay barrios donde los tipos de inmuebles típicamente más caros sean más baratos que en el resto de barrios?

Suposiciones iniciales

Durante las primeras fases de la exploración de los datos, los miembros del equipo fueron aportando diversas suposiciones iniciales. Las cinco consideradas más apropiadas fueron:

A. El barrio influye considerablemente en el precio del inmueble; los barrios más céntricos son más caros.
B. Los inmuebles con espacios abiertos (casas, chalets, etc.) serán en promedio más caros que los inmuebles que no lo tienen (apartamentos, dormitorios, etc.)
C. A mayor cantidad de huéspedes permitidos, mayor será el precio del inmueble.
D. Los inmuebles que ofrezcan camas normales serán más caros.
E. Los barrios, en promedio, más caros tendrán mejores reseñas.

Definición del dataset

El primer paso del proyecto consistió en definir el dataset con el se iba a trabajar. Se llevó a cabo una reunión para decidir cuáles columnas del dataset original se iban a conservar, después de que cada integrante del equipo llevase a cabo una revisión de los datos por su cuenta. Comenzamos analizando cada variable tratando de definir qué información se mostraba con cada una de ellas y evaluando la influencia que tendría en cada una de las etapas siguientes a la limpieza.

Los criterios que utilizamos fueron:

- Que una columna tuviera un gran porcentaje de nulos que no tuviera sentido reemplazar por otros valores como la media, mediana o 0 y 1.
- Que se tratara de campos de texto libre o URLs.
- Que la información de las variables estuviese duplicada.
- Que las filas pertenecieran a otras ciudades

Al final nos quedamos con 36 columnas de las 89 iniciales.

Exploración y preprocesamiento de los datos

En esta primera fase de exploración se realizó la descripción de todas las variables dataset para contextualizar los datos. Posteriormente, se emplearon Jupyter notebooks en Google Colab para trabajar colaborativamente.

El primer paso consistió en explorar los datos a través de operaciones sencillas como el conteo de nulos por columna, listado de tipos de datos por columna, datos únicos por columnas con información categórica (como el tipo de propiedad) y cálculos de estadísticos descriptivos como la media, mediana y los cuartiles. Esta fase fue fundamental porque permitió detectar irregularidades en el dataset, como columnas con una gran cantidad de nulos, tipos de datos inadecuados y detección de valores atípicos.

El proceso de limpieza de datos fue el siguiente:

1. Se eliminaron todas las columnas consideradas irrelevantes para nuestro análisis, lo que resultó en un dataset con sólo 36 columnas.
2. Se eliminaron los valores nulos con eliminación, imputación o interpolación según la variable. Por ejemplo, los nulos de las variables Weekly Price, Monthly Price, Security Deposit y Cleaning Fee se rellenaron con ceros, pues son valores opcionales.
3. Se subsanaron los valores atípicos que pudieran afectar el análisis.
4. Se eliminaron los registros duplicados.

5. Se exploraron las variables categóricas para identificar los niveles y estandarizarlas.
6. Se normalizaron los datos para asegurar que estuvieran en la misma escala y formato.
7. Se reasignó el tipo de dato en las variables que fuera necesario.
8. Finalmente se generó un nuevo dataset con 36 columnas y 13244 filas.

Definición e implementación del Datawarehouse

Para esta fase del proyecto, se utilizó DBeaver como herramienta para gestionar la base de datos. Los pasos que se siguieron en esta fase fueron:

1. Se creó nueva conexión en Postgres a una nueva base de datos local y se cargaron los datos del CSV en la herramienta de administración de bases de datos elegida: DBeaver.

2. Se experimentaron algunos problemas en la importación de datos:

    a. El tamaño asignado a cada variable no era suficiente para almacenar la información; por tanto, se tuvo se tuvo que ampliar el tipo de dato “varchar” a 100 caracteres.

    b. Los nombres de las columnas tenían espacios, lo que dificultaba la escritura del script para definir y consultar la base de datos, así que se reemplazaron los espacios por barras bajas utilizando la librería pandas en un Jupyter Notebook y exportamos un nuevo csv.

3. Se diseñó el esquema de entidad-relación (E-R) normalizado, lo que dio lugar a la creación de dos tablas para representar las dos entidades principales del dataset: los inmuebles y los anfitriones.

4. Se creó un esquema en la base de datos de la nueva conexión y se ejecutó un script de definición donde imitamos la estructura de las tablas del modelo E-R normalizado. Esto derivó en la creación de las tablas “host” y “property”.

    a. Como claves primarias (PK) se emplearon las columnas de “id_property” para identificar cada inmueble en la tabla “property” y, en el caso de los anfitriones, se asignó como clave primaria el “id_host” y el “id_property”. Esto último se debe a que se entiende que un mismo host puede tener varios inmuebles publicados.

b. La clave foránea (FK) de la tabla de anfitriones es “id_property”, la cual nos permitió hacer las uniones en las queries posteriores.

5. Creación de un script de consultas donde seleccionamos información de ambas tablas tanto de forma independiente como conjunta utilizando inners y outers joins.

Análisis exploratorio de los datos

Para este paso se realizó en un notebook empleando diversas librerías de Python como Pandas, Numpy, Seaborn, Matplotlib y Scipy. A continuación se detalla el paso a paso de esta fase:

1. Se cargaron los dataset original y final para monitorear los cambios en cada transformación. Después del análisis se removieron los outliers de la variable precio para ver si mejoraba la simetría de la distribución.

2. Se añadió la columna Total_Cost como la suma de las variables: Cleaning Fee, Security_Deposit y Precio (sin outliers).

3. Se visualizaron el resto de las variables para identificar valores atípicos y se corrigieron algunos.

4. Se realizó el análisis univariado de las variables Precio y Total Cost:

    4.1. En la primera inspección de precio se observaron valores altos en la asimetría y la curtosis que mejoraron significativamente cuando se removieron los outliers:

        4.1.1.1. Asimetría (grado de simetría de una distribución):

            4.1.1.1.1. Con outliers: 4.09056

            4.1.1.1.2. Sin outliers: 0.85602

        4.1.1.2. Curtosis (concentración de una distribución alrededor de la media):

            4.1.1.2.1. Con outliers: 29.987226298891287

            4.1.1.2.2. Sin outliers: 0.22138

    4.2. Se transformó la variable a escala logarítmica, pero el ajuste no fue significativo. Lo mismo ocurrió con la variable Total_Cost, cuyos valores de asimetría y curtosis fueron los siguientes antes y después de las transformaciones:

4.2.1. Asimetría**:**

4.2.1.1. Con outliers: 1.54407

4.2.1.2. Sin outliers: 0.92926

4.2.2. Curtosis**:**

4.2.2.1. Con outliers: 3.56980

4.2.2.2. Sin outliers: -0.09001

4.3. Este análisis se complementó con las siguientes visualizaciones: diagrama de caja, histograma con las medidas centrales, comparación de histograma con distribución normal y un gráfico Q-Q.

5. Finalmente se calculó el cociente de correlación de Pearson de ambas variables para determinar cuáles podrían ser relevantes en el entrenamiento del modelo.

6. Las transformaciones y pérdidas de información en cada transformación fueron las siguientes:

Dataset	**Filas**	**Columnas**	**% de registros en cada transformación**
Dataset original	14780	89	100%
Dataset final	13244	36	89,60%
Dataset sin outliers de Precio	12581	36	85,13%
Dataset sin outliers de Cleaning_Fee & Security_Deposit	12171	36	82,35%

Visualización de las métricas

La visualización de métricas se llevó a cabo con la herramienta de Tableau. El primer paso fue cargar la fuente de datos, en este caso, el archivo .csv preprocesado. Los campos calculados que se crearon fueron:

- Latitud float y Longitud float: se crearon a partir de dos campos del mismo nombre, aplicando las transformaciones a flotantes y asignándoles la función geográfica correspondiente.

- Precio total: Se calculó a partir de la suma del Precio + Tarifa de limpieza + Fianza.

Luego se diseñaron varias hojas de trabajo con gráficos centrados en la métrica principal: el precio total. Se generaron siete hojas de trabajo con las siguientes temáticas:

A. Mapa por barrio y precio**:** Consiste en un mapa divido por las diferentes zonas de la ciudad de Madrid donde se indica con una escala de color las zonas con precios totales promedios mas caros a los mas baratos.

    a. *Precio total por propiedad (tooltip)*: Permite comparar los precios totales según el tipo de propiedad.

B. Mapa de inmuebles por host**:** Similar al anterior, denota por color la mediana de inmuebles en propiedad de un mismo host en un mapa por barrios.

C. Política de cancelación por barrio: Un gráfico de barras donde se representan los diversos barrios según el promedio del precio total y con el detalle por colores del tipo de cancelación.

    a. *Precio total por tipo de cama (tooltip)*: Muestra los precios totales por tipo de cama.

D. Precio total y puntuación de reseñas: Otro gráfico de barras que representa en tamaño el precio total y por color la puntuación de las mismas. Incluye un parámetro “Top N” que el usuario puede actualizar al número que desee.

E. Precio total por número de huéspedes: Un gráfico de puntos con una recta de tendencia donde se representa como varían los precios en función del número de huéspedes con un nivel de detalle por barrios.

Una vez obtenidos los gráficos, se confeccionó el dashboard utilizando una paleta de colores ofrecido por Tableau y se incluyeron dos filtros por *Tipo de propiedad* y *Barrio*, los cuales afectan a tres de los cinco gráficos principales. El usuario puede interactuar y ajustar el tipo de propiedad y barrio para ver como varían las demás variables en función de estos dos filtros.

Modelo de regresión lineal

Para la elaboración de este paso final, utilizamos el lenguaje de programación R.

1. Se generó el dataset a partir del .csv generado en la tarea “Análisis exploratorio de los datos” y visualizamos el head para asegurarnos de que se haya creado correctamente.

2. Se obtuvieron con la librería GGally las tablas de correlación para confirmar los valores de relación entre variables que obtuvimos durante la exploración de los datos. Así comprobamos que las variables que más se relacionan con el precio son:

    a. Total_Cost (0.61)
    b. Accommodates (0.60)
    c. Cleaning_Fee (0.47)
    d. Guests_Included (0.44)
    e. Beds (0.40)
    f. Bedrooms (0.40)
    g. Security_Deposit (0.34)
    h. Has_TV (0.29)

3. Generamos 3 modelos a modo de prueba:

    a. Modelo para la variable “Price”: Obtuvimos un R^2 de 0.4024.

    b. Modelo para la variable “Total_Cost”: Decidimos no utilizarlo ya que es un campo calculado a partir de la suma de otros valores entre los que se encuentra el precio.

    c. Modelo para la variable “Precio” en escala logarítmica: Lo probamos porque detectamos que de esta manera la distribución de precios podría acercarse mas a una forma gaussiana. Los resultados no fueron los esperados, el R^2 conseguido fue de 0.3792.

4. Nos decidimos por utilizar el modelo generado para la variable “Price” porque es del que podíamos esperar mejores resultados con la información disponible.

5. Decidimos dejar fuera del modelo la variable Total_Cost ya que es un campo calculado a partir de la suma de otros valores entre los que se encuentra el precio que queremos predecir.

6. Quitamos también la variable “Has_TV” porque es una columna de booleanos. En caso de que no podamos conseguir un modelo aceptable consideraríamos manipular esa información para poder utilizarla.

7. Dividimos el dataset original en dos: un 70% de datos se destinaron al training, con el que entrenaremos el modelo, y el otro 30% fue al grupo del testing que utilizaremos para evaluar la calidad de las predicciones que realicemos.

8. Creamos el modelo de regresión lineal con el dataset de training y el resto de las variables. Obtuvimos los siguientes resultados de correlación y R^2 :

```
Call:
lm(formula = Price ~ Accommodates + Guests_Included + Beds +
    Bedrooms, data = df_train)

Residuals:
     Min       1Q   Median       3Q      Max
-176.447  -15.478   -4.478   12.628  115.522

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)      18.1399     0.5945   30.51   <2e-16 ***
Accommodates     10.4646     0.2629   39.80   <2e-16 ***
Guests_Included   4.8409     0.3285   14.74   <2e-16 ***
Beds             -3.9240     0.3351  -11.71   <2e-16 ***
Bedrooms          4.9567     0.4767   10.40   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 23.83 on 8514 degrees of freedom
Multiple R-squared:  0.4024,    Adjusted R-squared:  0.4022
F-statistic:  1434 on 4 and 8514 DF,  p-value: < 2.2e-16
```

9. Calculamos las figuras de calidad para evaluar nuestra predicción y obtuvimos lo siguiente:

```{r}
df_train$est <- predict(model, df_train)
caret::postResample(pred = df_train$est, obs=df_train$Price)
```

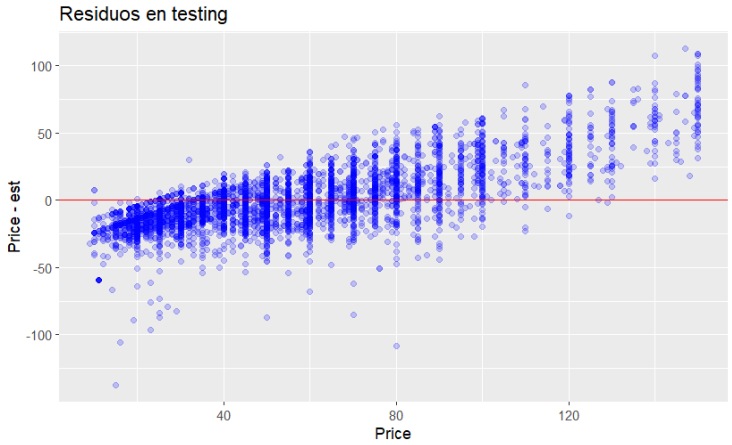
```
      RMSE   Rsquared        MAE
23.8272554  0.4024484 18.0540690
```

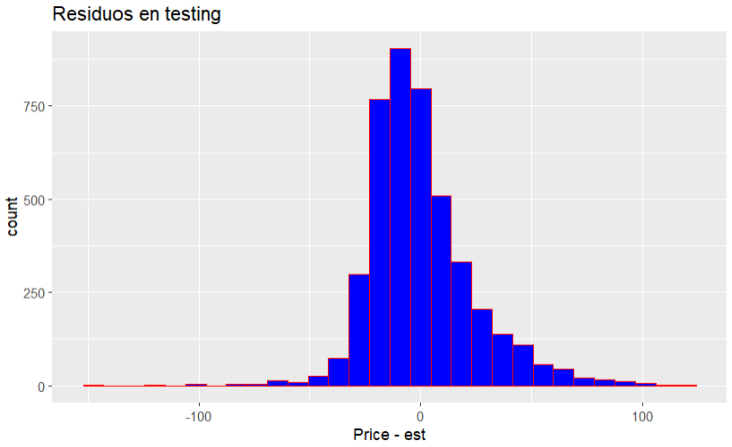
```{r}
df_test$est <- predict(model, df_test)
caret::postResample(pred = df_test$est, obs=df_test$Price)
```

```
      RMSE   Rsquared        MAE
24.2237875  0.3725053 18.0846108
```

Como se puede observar, RMSE_test ~= TMSE_train y R_train^2 ~= R_test^2, lo cual significa que los valores de train y test son parecidos, por tanto nuestro modelo cuenta con la cantidad de datos suficientes para poder predecir.

10. Para terminar, graficamos los residuos obtenidos de la diferencia entre el precio estimado y el precio real en test.





Los residuos deberían tener una distribución normal y estar centrados en cero, en nuestro caso eso no se cumple y queda demostrado que el modelo podría mejorarse.

Conclusiones y aprendizajes

Conseguimos generar un modelo aceptable de predicción considerando las herramientas y el tiempo que hemos tenido para trabajar los datos.

Algunos de los problemas que nos encontramos durante la realización de la práctica fueron:

- La falta de descripción de las variables y la forma de obtención de cada una de ellas.
- El poco tiempo que tuvimos para realizar la práctica: para obtener una predicción aceptable tuvimos que invertir la mayor parte del tiempo en analizar los datos y realizar la limpieza.
- La organización al principio no fue del todo óptima: la comunicación fue mayoritariamente escrita lo cual generó poca claridad y confusión para coordinar las primeras fases del proyecto.

- Falta de foco en las suposiciones iniciales: en la elaboración de fases como la visualización de datos no se siguieron del todo las suposiciones iniciales lo que dio lugar a la falta de respuesta a algunas de dichas suposiciones.

Algunos puntos fuertes de nuestro proyecto fueron:

- Utilizamos metodologías Ágiles y herramientas como Trello para organizar el trabajo estableciendo fechas y comunicación.

- Se realizaron varias reuniones para informar a todo el equipo del progreso de cada parte del proyecto de manera que todas conocieran los problemas y avances de cada etapa y dando la posibilidad de aportar nuevos puntos de vista y mejoras.

- Claridad y asertividad desde el primer momento con nuestras preferencias respecto a qué partes del proyecto nos gustaría dedicar más tiempo, teniendo en cuenta los gustos y habilidades adquiridas por cada una durante el Bootcamp.

Podríamos mejorar el modelo y obtener mejores resultados utilizando alguno de los siguientes recursos:

- Modificación de las variables categóricas por numéricas, separando las variables “Features” y “Amenities” para obtener más información.

- Codificación de variables categóricas como el Neighborhood, Zipcode y Tipo de propiedad pues se observó en el dashboard de Tableau que el precio, el barrio y el barrio están relacionados.

- Evaluación de la posibilidad de generar otro tipo de regresión que no sea lineal.

Respecto a las suposiciones iniciales, los resultados obtenidos fueron:

- Relación entre los barrios y el precio: A nivel de visualización se puede observar que algunos barrios más céntricos son más caros, sin embargo sería necesario realizar cálculos estadísticos para poder afirmar una relación entre ambas variables.

- Relación entre tipo de inmueble y el precio: En el dashboard de Tableau hemos podido observar que en diversos barrios propiedades como los chalets, las casas y las villas son en promedio más caras que el resto de inmuebles.

- Relación entre huéspedes y el precio: Tanto en la visualización como en el análisis exploratorio de datos, se pudo observar una correlación superior a las demás entre el precio y el número de huéspedes permitidos.

- Relación entre tipo de cama y el precio: En el dashboard se puede observar en diferentes barrios que los inmuebles con camas normales son más caros en promedio. Sin embargo, durante el análisis y exploración de los datos, nos hemos dado cuenta de que el dato de *Tipo de cama* es de mala calidad ya que no se puede saber si el tipo de cama es el mismo en aquellos inmuebles donde hay más de una cama.

- Relación entre la calidad de las reseñas y el precio: A nivel visual se puede ver que los barrios más caros tienen buenas reseñas pero es pertinente hacer un análisis estadístico para comprobar esta suposición.

En líneas generales, este proyecto final nos ha permitido poner en práctica los conocimientos adquiridos durante el bootcamp con un conjunto de datos reales, hemos demostrado autonomía a la hora de resolver las dificultades que han ido surgiendo en el camino y nos ha servido para tener una primera toma de contacto con las tareas típicas de un profesional en el área del big data.