Trabajo Fin de Máster

Data Science & Machine Learning:
Airbnb New York City

Autor:

Cristóbal Bascuñán Larenas

Máster en Big Data & Business Analytics

IMF Business School

2020

RESUMEN

En la última década la capacidad de almacenar y procesar datos ha tenido un boom con respecto a todo lo visto anteriormente gracias a la generalización del uso de internet, la reducción de costes de almacenamiento y la aparición de nuevas tecnologías capaces de procesar estos mismos. A pesar de esto, el tratamiento de los datos sigue siendo un proceso clave debido a que es necesario enfocar los datos a casos de uso reales que acaben derivando en conocimiento y valor añadido. Para esto es necesario aplicar los procesos claves de la ciencia de datos: ETL (extracción, transformación y carga), el análisis exploratorio de los datos, el uso de modelos de aprendizaje automático y una visualización eficaz de los datos. En este caso se utilizarán datos abiertos de la plataforma Airbnb para poder hacer predicciones y clasificaciones, y descubrir que información valiosa se encuentra escondida.

PALABRAS CLAVE: ciencia de datos, aprendizaje automático, ETL, Python, Sklearn.

Contenido

ABSTRACT:

In the last decade, the capacity to store and process data has boomed realated to everything seen previously, mainly because the general use of the internet, the reduction of storage costs and the appearance of new technologies capable of processing data. Despite this, data processing remains a key process because it is necessary to focus the data on real use cases that end up resulting in knowledge and added value. For this it is necessary to apply the key processes of data science: ETL (extraction, transformation and loading), exploratory analysis of data, the use of machine learning models and effective visualization of data. In this case, open data from the Airbnb platform will be used to make predictions and classifications, and discover what valuable information is hidden.

KEY WORDS: data science, machine learning, ETL, Python, Sklearn.

1. Presentación y justificación del trabajo.

1.1 Ciencia de datos

Para poner en situación lo que es la ciencia de datos empezaré con la definición más genérica que corresponde a la puesta en Wikipedia: *La ciencia de datos es un "concepto para unificar estadísticas, análisis de datos, Machine Learning y sus métodos relacionados", para "entender y analizar fenómenos reales" con datos.* (Liu, 2015) Y aunque puede ser acertada, puede ser más correcta en la aplicación la de la estadística Cassie Kozyrkov: *"La ciencia de datos es la disciplina de hacer que los datos sean útiles"* (Kozyrkov, 2018).

Hay que tener en cuenta que el término se acuña a una realidad y entorno Big Data, por lo que hay que tener las siguientes consideraciones con respecto a la migración existente que hay en la analítica tradicional y los sistemas de BI (Inteligencia de negocio) que quieren pasar a la tecnología Big Data:

- Los sistemas de BI tardan mucho en comparación con los de Big Data y puede conllevar que los resultados no estén disponibles cuando se necesiten.
- Los beneficios que conllevan la transformación son los siguientes: agilidad, ahorro de costes, facilidad de acceso a los datos y escalabilidad.

Los proyectos de ciencia de datos en los cuales uno de los objetivos es la creación de un sistema predictivo, tiene que ser de carácter táctico o estratégicos y ser liderados por el área de negocio o por la alta dirección. Las búsquedas de mejoras pueden ser orientadas a:

- Mejora en la toma de decisiones: queremos saber qué es lo más probable para poder tomar mejores decisiones. Este tipo de proyectos son normalmente regresiones.
- Mejorar un proceso: identificar un proceso de negocio en el que cree que puede mejorar usando el aprendizaje automático.

El proyecto de ciencia de datos se compone principalmente de los siguientes pasos:

- Construcción del set de datos: es el punto de partida y el cual puede variar en dificultad dependiendo si es obtención de los datos almacenados en un a base de datos o datalake, llegando a mayor complejidad si los datos vienen de fuentes

distintas, y se tienen que lidiar con diferentes tipos de datos tanto estructurados como no estructurados.

- Limpieza del set de datos: se define como se van a unir las diferentes fuentes, como lidiar con las variables categóricas, definir la política de datos faltantes y nulos y eliminar variables innecesarias.
- Estudio del set de datos: en esta fase se estudian los datos, correlaciones con la variable objetivo, distribuciones, poder predictor de las variables (cuáles son más importantes). Si tenemos un conjunto de datos suficiente, si es un conjunto de datos balanceado o no balanceado. En el caso de un conjunto de datos no balanceado, debemos construir un conjunto de datos de entrenamiento que lo sea, es decir, que proporcione suficientes ejemplos de ambas clases para que el algoritmo pueda aprender (este problema se da en clasificación).
- Prueba de modelos sobre el set de datos: en función del problema que tengamos, debemos identificar qué algoritmos son los más idóneos para resolverlo. Siempre tenemos, como referencia para identificar si hemos tenido éxito, las métricas identificadas por negocio. Normalmente se comienza por los algoritmos más sencillos, como regresión lineal, y se van probando diferentes algoritmos hasta obtener los resultados esperados por negocio. No es muy aconsejable comenzar por algoritmos de caja negra como redes neuronales, ya que estos algoritmos no nos permiten saber qué variables son las más importantes.
- Evaluación de modelos: en esta fase aplicamos siempre validación cruzada, ya que el objetivo de cualquier algoritmo es funcionar de manera óptima con ejemplos que el algoritmo no ha visto. Es decir, que el algoritmo no tenga bajo aprendizaje o sobreaprendizaje.
- Ingeniería de características: en las primeras iteraciones no se suelen obtener los resultados esperados por negocio, por ello es necesario explorar los datos, inferir nuevas variables y hablar con negocio para identificar más variables que son importantes para resolver el problema.
- Necesidad de un set de datos más grande: en ocasiones, el problema que tenemos no es que necesitemos más variables, sino que no tenemos un conjunto de datos lo suficientemente grande como para que el algoritmo pueda aprender de manera correcta. En este escenario, debemos obtener más datos.

1.2 Aprendizaje automático

El aprendizaje automático se puede definir de la siguiente forma: "Un subconjunto de la inteligencia artificial (IA), el aprendizaje automático (en adelante Machine Learning) es el área de la ciencia computacional que se centra en el análisis y la interpretación de patrones y estructuras de datos que hacen posible el aprendizaje, el razonamiento y la toma de decisiones sin interacción humana."

Es decir, permite que, mediante el uso de algoritmos informáticos con una gran cantidad de datos, se pueda analizar, tomar decisiones y hacer recomendaciones o predicciones basándose solo en los datos introducidos. También puede entrenarse para mejorar las tomas de decisiones futuras. Esto se consigue partiendo de unos datos de entrada y una serie de características facilitadas por el analista de datos para aplicar al modelo.

Esta es una tecnología cada vez más utilizada ya que tanto como el coste de almacenamiento como la capacidad de computo cada vez es menor, pudiendo hacer de esta forma análisis que antes eran impensables por la cantidad de datos o por el coste que este proceso pudiera tener. Los beneficios que derivan de este uso son mucho mayores al analizar grandes conjuntos ya que puede llegar a identificar patrones, modelos y oportunidades imperceptibles con métodos tradicionales.

Los algoritmos de Machine Learning se suelen categorizar como supervisados o sin supervisar, habiendo también modelos híbridos.

1.2.1 Aprendizaje supervisado

Son algoritmos que pueden aplicar lo aprendido en el pasado a nuevos para poder predecir eventos futuros (Simeone, 2018). Se trabaja con datos 'etiquetados', que mediante os datos de entrada se asigna una etiqueta de salida correcta. Sus principales usos son los problemas de regresión (variable continua) y de clasificación (variable discreta). En ambos casos los datos deben estar definidos como vectores numéricos. Algunos ejemplos de algoritmos serían:

- Árboles de decisión.
- Clasificación de Naïve Bayes.
- Regresión por mínimos cuadrados.
- Regresión Logística.
- Support Vector Machines (SVM).

- Métodos “Ensemble” (Conjuntos de clasificadores).

1.2.2. Aprendizaje no Supervisado

Por el contrario, el aprendizaje no supervisado tiene lugar cuando la información usada para el entrenamiento no está ni etiquetada ni clasificada (Simeone, 2018). Por lo tanto, el aprendizaje no supervisado estudia en como el sistema puede encontrar alguna función que describa una estructura ocultada desde los datos sin etiquetar. Tiene carácter exploratorio.

El aprendizaje no supervisado se suele usar en problemas de clustering, agrupamientos de co-ocurrencias o en profiling. Sin embargo, los problemas que implican tareas de encontrar similitud, predicción de enlaces o reducción de datos, pueden ser supervisados o no.

Los tipos de algoritmo más habituales en aprendizaje no supervisado son:

1. Algoritmos de clustering
2. Análisis de componentes principales
3. Descomposición en valores singulares (singular value decomposition)
4. Análisis de componentes principales (Independent Component Analysis)

1.2.3 Otros tipos de aprendizaje automático

- Aprendizaje semi-supervisado: es el uso de datos etiquetados y sin etiquetar para el entrenamiento. Normalmente una pequeña porción del total son datos etiquetados y el resto datos sin etiquetar.

- Aprendizaje por refuerzo: es un método de aprendizaje que interactúa con su entorno produciendo acciones y descubriendo errores o recompensas. Usando búsqueda mediante prueba y error y las recompensas con retraso son las características más relevantes.

2. Hipótesis de trabajo y objetivos.

Puesta en escena la definición y las fases de un proyecto de Data Science se desarrollarán las hipótesis y objetivos buscados en este proyecto.

El proyecto se basará en los datasets distribuidos por Inside Airbnb, página la cual recoge los datos de los alojamientos disponibles en la web de Airbnb para la ciudad de

Nueva York. Con estos mismos se realizarán todos los pasos descritos anteriormente, desde la obtención y carga de datos hasta la aplicación de modelos de aprendizaje automático.

Para este estudio se utilizará un dataset publicado en la página Kaggle.com, una de las comunidades de Data Science más grande del mundo, en la cual se publican distintos datasets para competiciones y resoluciones comunes a proyectos planteados por los propios usuarios o por entidades que ofrecen recompensa. En los primeros ensayos con los datos contenidos en el CSV proporcionado, al no obtener resultados concluyentes ni de gran peso se han complementado con un dataset completo ofrecido por la página Inside Airbnb.

El objetivo principal del proyecto será, por lo tanto, hacer un análisis exploratorio de los datos para poder inferir en como son los datos y que relación e inferencia tienen en el precio de cada una de las ofertas de alojamiento ofrecidas con vistas a poder crear un modelo en el cual el algoritmo propuesto pueda predecir los precios atendiendo a las distintas variables de nuevos alojamientos.

2.2 Especificaciones en el caso de uso

2.2.1 Contexto

Airbnb es una plataforma online que conecta a personas que buscan alquilar sus casas con personas que buscan alojamiento. Actualmente cubre 191 países alrededor del mundo. Desde 2008, *guests* y *hosts* han usado Airbnb para expandir las posibilidades y de esta forma presentar una forma más personalizada de experimentar el mundo.

Debido a estas premisas se generan cantidades ingentes de datos para cada una de las ciudades cada año. En este caso de uso se utilizarán los datos de Airbnb en la ciudad de Nueva York en los periodos de junio de 2019 a junio de 2020.

Se pueden hacer preguntas a priori que serán contestadas mediante los datos como: ¿Qué diferencias hay entre barrios?, ¿el precio varía dependiendo de la antigüedad del host?, ¿qué podemos aprender de las predicciones?...

2.2.2 Leyes aplicables

Hay que tener en cuenta que existen exigencias legales para evitar la creación de 'hoteles ilegales', entornos inmobiliarios que aprovechen una distinta legislación a favor

en el caso de utilizar la plataforma para hacer negocios para pagar menos impuestos… etc. Esto no será caso de estudio en este proyecto, pero sí que tendrá repercusión en los casos en que se encuentren valores extremos (outliers) que difieran de la realidad.

Las leyes aplicables al alquiler de vivienda se pueden encontrar en https://www.nolo.com/legal-encyclopedia/overview-airbnb-law-new-york-city.html

También existen otras leyes que podrían aplicarse a un host de Airbnb, incluidos los impuestos, la regulación de alquileres, los códigos de zonificación, las licencias comerciales y los contratos, incluidos los arrendamientos. Airbnb se vio obligado en un acuerdo extrajudicial a proporcionar esta información a sus anfitriones (Airbnb, 2018).

3. Material y métodos

Para este proyecto se usará la versión de Python 3.7.6 en el entorno de herramientas de Anaconda. Precisamente se usará Spyder 4.1.3 para el control y gestión de las distintas variables creadas y Jupyter 6.0.3 como notebook en el que presentar los resultados finales.

Las librerías y módulos de Python que se utilizarán son los siguientes:

- NumPy: biblioteca de funciones matemáticas de alto nivel para operar con esos vectores o matrices.
- Pandas: biblioteca que ofrece estructuras de datos y operaciones para manipular tablas numéricas y series temporales.
- Matplotlib: biblioteca para la generación de gráficos a partir de datos contenidos en listas o arrays en el lenguaje de programación Python.
- Seaborn: biblioteca para la visualización de datos, basada en Matplotlib. Provee una interfaz de alto nivel para diseñar gráficos estadísticos.
- Datetime: módulo que proporciona formas de manipulación de fechas y de tiempos.
- Folium: librería que permite crear mapas interactivos usando Leaflet.js mediante la creación de código javascript.
- Plotly: librería para la creación de gráficos interactivos.
- Wordcloud: módulo para la creación de visualizaciones con palabras.
- Sklearn: librería para el uso de Machine Learning.

- Geopandas: extensión de la librería de pandas para el uso de archivos y operaciones geoespaciales.

El dataset del que se dispone se compone de los siguientes elementos:

Nombre de archivo	**Descripción**
calendar.csv	Contiene datos relativos al precio de alojamineto por noche de cada uno de los días divido por id del host. 746 MB
listings.csv	Contiene información resumida relativa a cada uno de los alojamientos. 6,84 MB
listings_detailed.csv	Contiene todo el despliegue de información detallada para cada uno de los alojamientos. 178 MB
neighbourhoods.csv	Listado de los barrios de Nueva York y a que grupo de barrios pertenecen. 4,84Kb
neighbourhoods.geojson	Contiene los atributos geoespaciales para cada uno de los barrios.
New_York_City.png	Imagen de la ciudad de Nueva York
reviews.csv	Fechas en las que se ha producido una reseña. 21,8MB

De estos, no se utilizarán los archivos listings.csv, neighbourhoods.csv y reviews.csv debido a que esa información ya está contenida en el archivo listings_detailed.

4. Solución desarrollada.

4.1 Descripción de los datos

El dataset principal con el que se trabajará será listings_detailed.csv. Este será convertido a un Dataframe de Pandas en su lectura. Al ver su contenido nos podemos percatar que se compone de 49.530 filas y 106 filas.

Tras revisar las columnas de este primer Dataframe nos podemos percatar que hay columnas que no aportarán valor al estudio, como pueden ser IDs o partes de formularios que no añaden profundidad.

Por lo tanto, tras el primer cribado, nos quedamos con 41 columnas las cuales son:

Nombre	Descripción	Tipo
name	Nombre de la oferta de alojamiento	String
description	Descripción de la oferta	String
transit	Información acerca de cómo transitar por la ciudad	String
house_rules	Reglas en el alojamiento	String
host_id	ID del host	String
host_since	Tiempo que lleva en la plataforma siendo host	Datetime
avilability_365	Tiempo disponible al año	int64
host_is_superhost	El host tiene el reconocimiento de superhost	Bool
host_listings_count	Número de anuncios publicados que tiene el mismo host	float64
host_identity_verified	El host ha verificado su identidad	Bool
neighbourhood	Barrio al que pertenece el sitio de alojamiento	String
neighbourhood_group_cleansed	Agrupación en el que se clasifican los barrios	String
latitude	Latitud	float64
longitude	Longitud	float64
property_type	Tipo de alojamiento: casa, apartamento, hotel, barco...	String
room_type	Espacio para el guest: casa entera, habitación privada…	String
accommodates	Número de habitaciones	int64

bathrooms	Número de baños	float64
bedrooms	Número de dormitorios	float64
beds	Número de camas	float64
bed_type	Tipo de cama	String
square_feet	Metros cuadrados de la propiedad	float64
price	Precio por noche	float64
security_deposit	Precio de la fianza	float64
cleaning_fee	Gastos de limpieza	float64
guests_incuded	Número de huéspedes que incluye el precio	int64
extra_people	Precio por huéspedes extra	float64
minimum_nights	Noches mínimas para la contratación del servicio	int64
máximum_nights	Noches máximas de estadía	int64
number_of_reviews	Número de opiniones recibidas	int64
review_scores_rating	Valoración total de las opiniones	float64
review_scores_accuracy	Valoración sobre la precisión del anuncio con la realidad	float64
review_scores_cleanlines	Valoración la limpieza del sitio	float64
review_scores_checkin	Valoración de la facilidad en el check-in	float64
review_scores_communication	Valoración sobre la facilidad de comunicación con el host	float64
review_scores_location	Valoración sobre la ubicación del alojamiento	float64
instant_bookable	Es posible reservar instantáneamente	Bool

cancellation_policy	Tipo de penalización por cancelación	String
amenities	Listado de comodidades/extras que incluye el alojamiento	Lista anidada

Aparte, tenemos el fichero 'calendar.csv' el cual se compone de 18.078.753 de filas y 7 columnas. Se seleccionan solo 3 debido a que posteriormente estos datos se juntarán con el otro dataframe. Estos datos son los siguientes:

Nombre	Descripción	Tipo
listing_id	ID de la oferta de alojamiento	String
date	Día entre el 08/06/2019 y 09/06/2020	Datetime
price	Precio por noche	float64

4.1.1 Conversión de datos

Por lo tanto, al tener definido que son cada una de las columnas y que tipo de datos deberían tener, se procede a hacer cambios en las columnas que no han podido ser convertidas automáticamente en la lectura al tupo de dato que debería ser. Esto pasa con las columnas:

- "price", "security_deposit", "cleaning_fee", "extra_people": son columnas que, aparte de aparecer como cadena de texto, tienen el símbolo '$' el cual se tiene que retirar antes de convertirla en float.
- "transit", "house_rules", "host_is_superhost", "host_has_profile_pic", "host_identity_verified", "instant_bookable": a estos se los convierte en tipo booleano. Si tienen valores 0 o f (de falso) se considerará que no hay/no cumple, y en caso contrario que sí que hay / existe.
- "host_since": viene como fecha, pero se convertirá en un integer, debido a que para los algoritmos posteriormente usados será más eficiente tener una escala númerica que marque el número de días entre 'hoy' y el día en que empezó a ser host.
- "date": en 'calendar' se convierte correctamente a fecha.

4.1.2 Política de datos faltantes o nulos

Tras convertir los datos en el tipo de datos que les corresponde toca lidiar con los datos faltantes o datos nulos. En este caso se tomarán tres tipos de políticas:

1. Para los datos que pueden o no tener contenido, en el caso de que no contengan se les dará un valor de 0.
2. Para los datos que tienen que tener estrictamente un valor, pero se mantenga dentro de un rango numérico discreto, se usará la media.
3. Para los datos que debido a su significado no se da un valor, se pondrá una cadena de texto. Ej.: Si falta un valor en "neighbourhood" se pondrá "Others".

4.1.3 Política de outliers

Es normal que en los conjuntos de datos numéricos haya datos extremos (u outliers) que desvirtúen los objetivos conseguidos, ya sea porque aparecen de forma natural o por errores.

Por lo tanto, hay que decidir que se hace con los valores extremos. Para encontrarlos una de las formas más eficaces es con una simple llamada describe() en la cual se expresan las principales estadísticas como son el mínimo, máximo, media, mediana, desviación estándar o los cuartiles.

Tras visualizar y comprobar los outliers se toman las siguientes decisiones:

- El pedir más de 365 días como mínimo de estadía, se consideraría un alquiler a medio plazo y por lo tanto quedaría fuera del estudio y del caso de uso de Airbnb. Por lo tanto, se eliminan del estudio los valores superiores a 365 días (15 filas).
- Se puede comprobar que hay valores en las que el precio es 0, y cerca de 10000 precio por noche. Esto no tiene sentido alguno, por lo tanto, se considerará que menos de 10 dólares por noche, y mayores que 1000$ son outliers que no representan el supuesto sistema de "sharing economy" que se supone que es Airbnb. Aún se podría reducir aún más, pero hay que tener en cuenta el caso de uso en el que se esta utilizando: Nueva York es una de las ciudades más caras del mundo, puede haber estancias en el punto más céntrico de Manhattan y que

cuenten con todo tipo de lujos, por lo tanto para tener en consideración este tipo de escenarios los cuales si pueden ser plausibles, se dejará ese límite. Al calcular el porcentaje de datos que supone el eliminar estos datos, no llegan ni a un 1% del total. Por lo cual para una mayor precisión en el análisis y en la creación de los modelos, se prescindirá de estos.

4.1.4 Adición de nuevas columnas

Para enriquecer más los datos que tenemos, se añadirán nuevas columnas de carácter booleano al dataframe existente, con objeto de crear características más específicas. Estas serán:

- Clasificar los pisos en función si son mayores o menores de 120 m2. Se eliminará la columna original que contiene los metros cuadrados.
- En relación con la disponibilidad de un alojamiento se añadirán dos columnas, dependiendo si son de alta disponibilidad o baja, si los días que tienen disponibles son mayores a 351 días (solo dos semanas no disponibles al año) o disponibilidad menos a 14 días (dos semanas al año), respectivamente.
- Las estancias que no hayan recibido ninguna valoración (review).
- Además, extrayendo los datos de la columna de listas anidadas 'amenities' (en este caso se dejará con formato string), se clasificarán las características que se crean que pueden tener un mayor impacto en el precio de una estancia. Estas serán las siguientes.
    - Breakfast
    - Patio or balcony
    - Garden or backyard
    - Bathtub
    - Paid parking off premises
    - Pool
    - Suitable for events
    - Air conditioning
    - Laptop-friendly workspace
    - Indoor fireplace

4.1.5 Operaciónes de join

Para el uso en el análisis exploratorios de los datos se usará la mezcla de los archivos ‘calendar.csv’ y ‘listing_detailed.csv’ mediante una función de merge como se ve en el siguiente código:

calendar_neigh = pd.merge(calendar, df_neigh, how='left', left_on="listing_id", right_on="id")

El nuevo dataset será un left join entre estos datasets unidos por las id’s correspondientes a la oferta de alojamiento, en la cual solo se sumará a los datos contenidos en el dataframe ‘calendar’ la columna de ‘neighborhood’ ya que el objetivo de este estudio no es la predicción de los precios con la variable tiempo debido a que al ser un trabajo académico se necesitarían mayores recursos para poder usar Machine Learning en un dataset de más de 18 millones de filas y más de 40 variables. Pero si que se usará para tener una visión en la evolución de los precios en la ciudad como en la subdivisión por barrios.

4.2 EDA (Exploratory Data Analysis)

El análisis exploratorio de datos (EDA) se refiere al proceso crítico de realizar investigaciones iniciales sobre los datos para descubrir patrones, detectar anomalías, probar hipótesis y verificar supuestos con la ayuda de estadísticas resumidas y representaciones gráficas. (Patil, 2018)

En este estudio se verán diferentes puntos de vista dependiendo de las diferentes variables, preferiblemente para ver cómo estas inciden en el precio por noche, pero también encontrando información que puede ser valiosa para diferentes estudios u objetivos.

4.2.1 Neighborhoods

4.1.1 Agrupación de barrios

Una de las grandes variables que tienen gran peso y protagonismo en el caso de estudio de Airbnb, que podría ser extrapolable a no solo Nueva York sino a cualquier ciudad del mundo la división de barrios, en las cuales sus características varían, así como los precios serán mayores o menores.

Por lo tanto, como punto de partida hay que dejar claro donde se sitúan estas agrupaciones de barrios (que contienen los barrios individuales) y su distribución geográfica.

Para la realización de este gráfico se utiliza tanto la información de polígonos contenida en "neighbourhoods.geojson" como la imagen de la silueta de Nueva York.

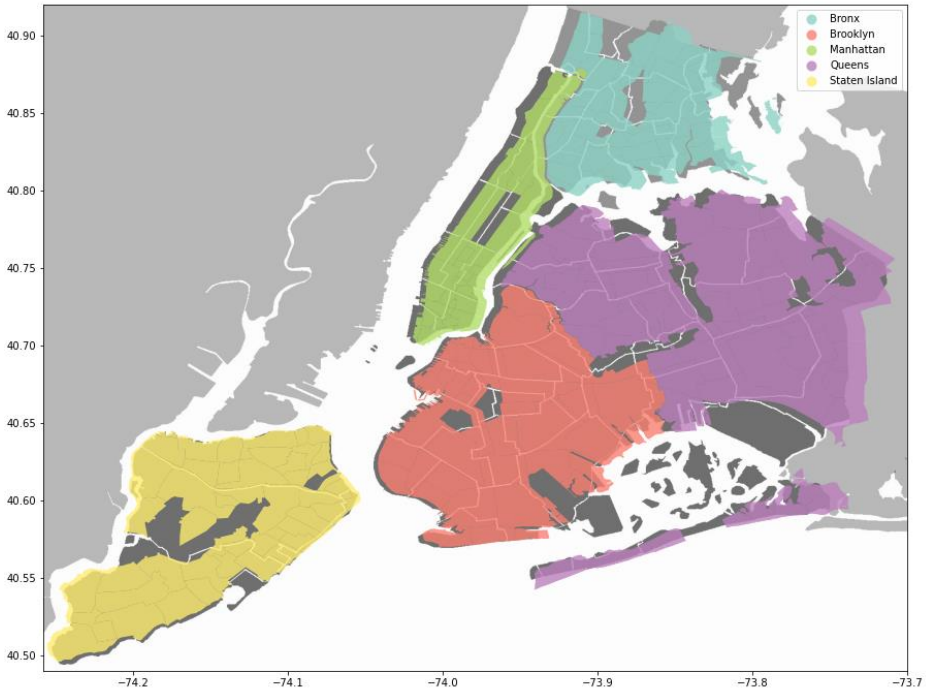

Figura 1. Localización de los grandes barrios de Nueva York.
Creación propia mediante Plotly

Las divisiones por grupos de barrios son claros, en la parte superior se encuentra el Bronx, en la isla intermedia Manhattan, en la península en la parte derecha se encuentra Queens, en la parte izquierda Brooklyn y pasado el puente de Verrazano-Narrows nos encontramos con Staten Island en la izquierda del todo.

Se puede ver la distribución de cuáles son las zonas con más ofertas de alojamiento mediante un mapa de calor interactivo en el cuales es posible acercarse o alejarse dependiendo de que zona nos interese más, y además de que nos indicará por colores la densidad. Esto se logra mediante el uso de la librería *folium*. En gris están marcadas las zonas con menos densidad hasta el amarillo que es la zona con mayor densidad.

Tras visualizar los datos se puede ver que las zonas con mayor número de alojamientos publicados son en Manhattan y en Brooklyn. En donde menos es cuanto más alejado del centro, como es el caso de Staten Island donde el azul escasea. Esto, aunque no sea

parte del estudio, se puede relacionar con la problemática y la controversia que existe en la actualidad ante la problemática que trae el convertir los centros de las ciudades en alojamientos para turistas, provocando la subida de los alquileres para la gente que vive en la ciudad y el éxodo a zonas a las afueras (Guttentag, 2018).

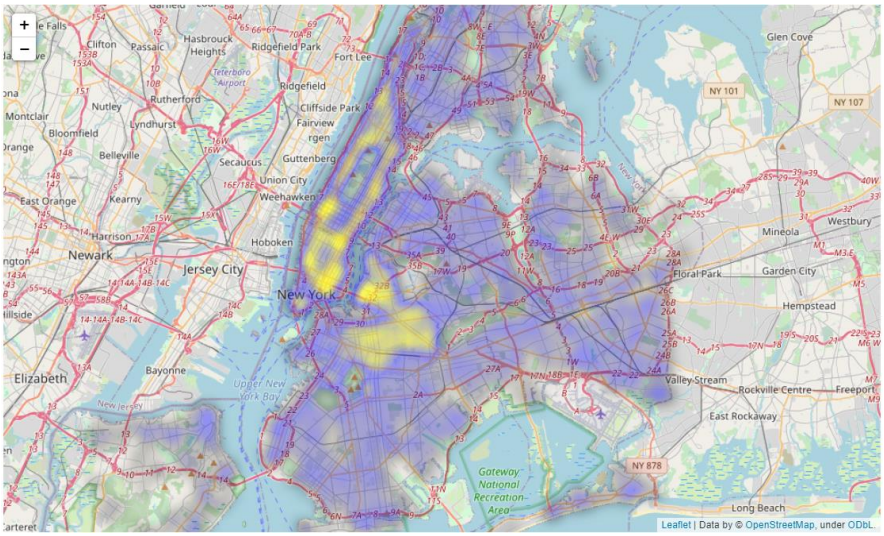

Figura 2. Densidad de ofertas de alojamiento en Nueva York.
Creación propia mediante Folium.

También mirando el mapa en cual se representen cada uno de los alojamientos en una escala de precios desde el más claro al más oscuro por precio nos podemos hacer una idea de cómo está distribuido los precios a nivel espacial.

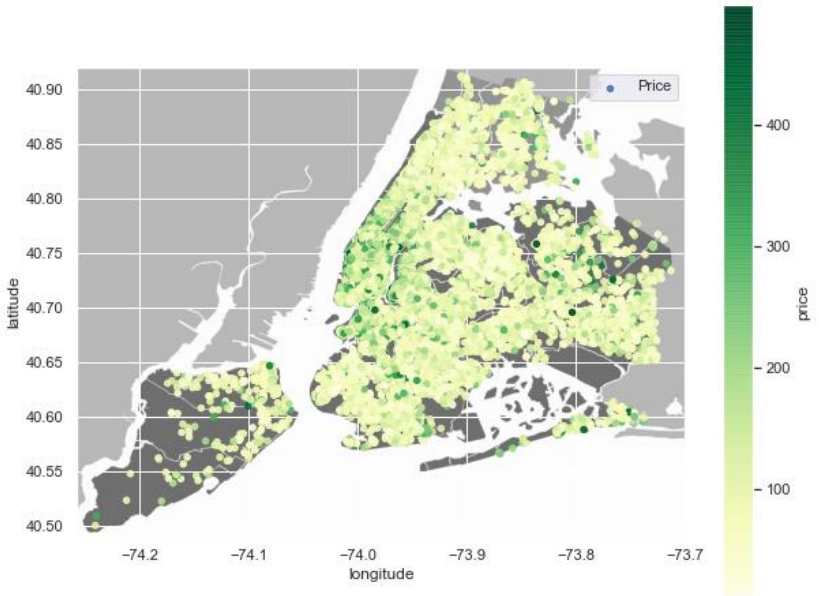

Figura 3. Distribución de los precios en el mapa de Nueva York.
Creación propia mediante Plotly.

En el gráfico anterior se puede estimar que donde los precios están más altos son en la zona de Manhattan y alrededores, y por la parte más del norte de de Brooklyn y por lo tanto cercana a Manhattan. También nos podemos dar cuenta que hay precios altos en las zonas más alejadas del centro en las que hay menos densidad de alojamientos.

Para poder concretar cuál es la situación de precios por agrupaciones de barrios, lo mejor es recurrir a las medidas estadísticas principales desglosadas por cada uno de ellos.

Stats	Brooklyn	Manhattan	Queens	Staten Island	Bronx
mean	114.404267	164.276028	91.431152	93.697802	85.567477
std	81.559778	108.146673	65.726231	72.067208	65.329346
min	10.000000	10.000000	10.000000	20.000000	11.000000
25%	60.000000	90.000000	50.000000	50.000000	49.000000
50%	90.000000	139.000000	72.000000	75.000000	69.000000
75%	149.000000	200.000000	110.000000	110.000000	100.000000
max	745.000000	749.000000	700.000000	700.000000	680.000000

Por lo tanto, se puede crear un ranking de grupos de barrios dependiendo del precio medio por noche. Desde el más caro al más barato: Manhattan, Brooklyn, Staten Island, Queens y Bronx.

Para poder revisar de una forma más visual el resto de estadísticas se usa un gráfico boxplot. Se usa un límite de precios de 500$ para una mejor visualización en la cual los valores más extremos no desvirtúen la escala de precios.

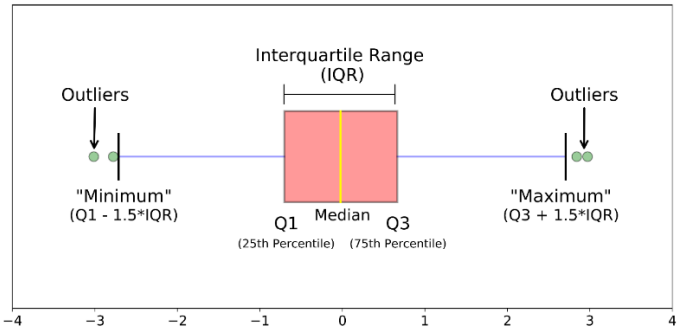

Figura 4. Understanding boxplots.

Tomado de: https://towardsdatascience.com/understanding-boxplots-5e2df7bcbd51

Para entender el boxplot es necesario saber que la caja central contiene desde el 25% del percentil hasta el 75% con la línea central marcando la mediana. Las barras horizontales en las colas (líneas de desviación) marcan el mínimo y el máximo valor. Los puntos

fuera de estos son los considerados outliers: valores extremos con una densidad que comparado con el tota no son significantes.

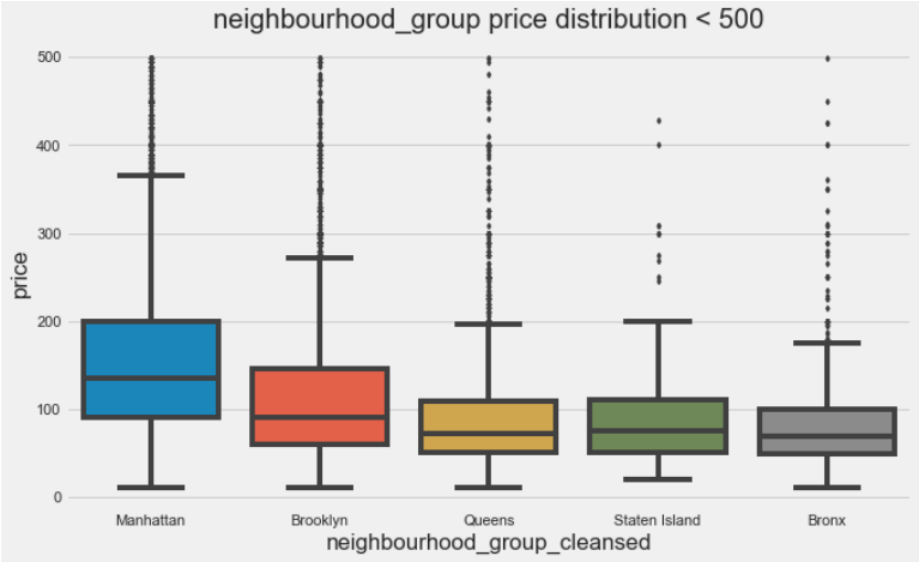

Figura 5. Boxplots de la variable precio por conjunto de barrios.
Creación propia mediante Seaborn

Como se puede ver en el gráfico se mantienen los datos de mayor precio en Manhattan y Brooklyn con una gran varianza. En el caso de Queens se puede ver una gran cantidad de outliers. En todos los casos se solapan bastante los outliers por lo cual, se presenta otro gráfico con el fin de evitar esto y ver también las densidades.

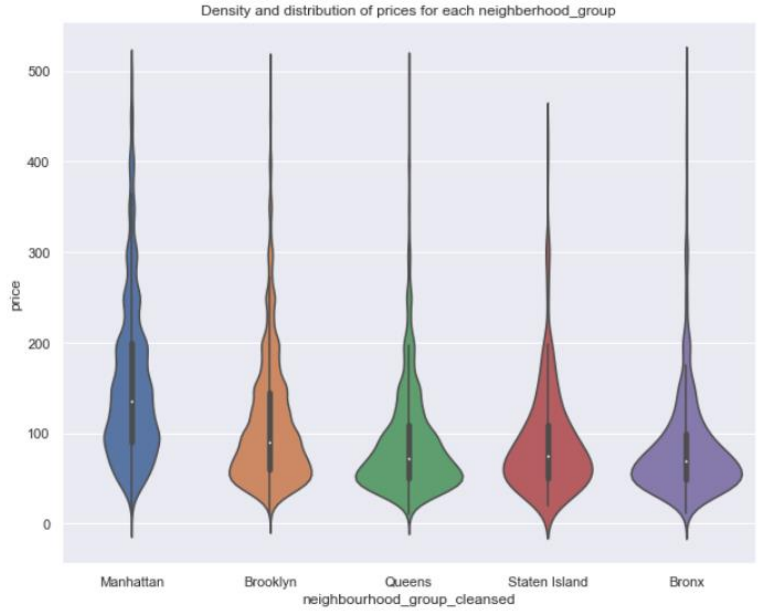

Figura 6. Violin plot de la variable precio por conjunto de barrios
Creación propia mediante Seaborn.

Con este tipo de visualización podemos ver de una forma más exacta como se distribuyen los precios y que densidad tienen los outliers.

Si se quiere hacer un análisis más detallado en uno de los conjuntos de barrios, se puede utilizar un gráfico de distribución como el siguiente que se hace para Brooklyn:

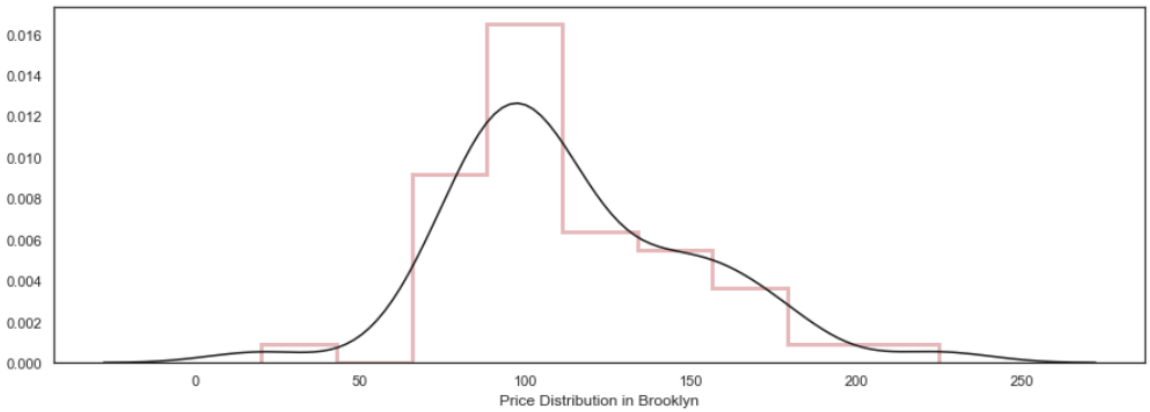

Figura 7. Gráfico de distribución de los precios en Brooklyn.

Creación propia mediante Seaborn

Pero ya que el objetivo no es ahondar en solo un conjunto de barrios, sino una visión general, no se procederá a hacer un gráfico para cada uno.

4.2.1.1 Disgregación por barrio

Para tener una visión más clara del contenido en cuanto a precio de cada uno de los barrios, se utilizará un gráfico interactivo mediante plotly.

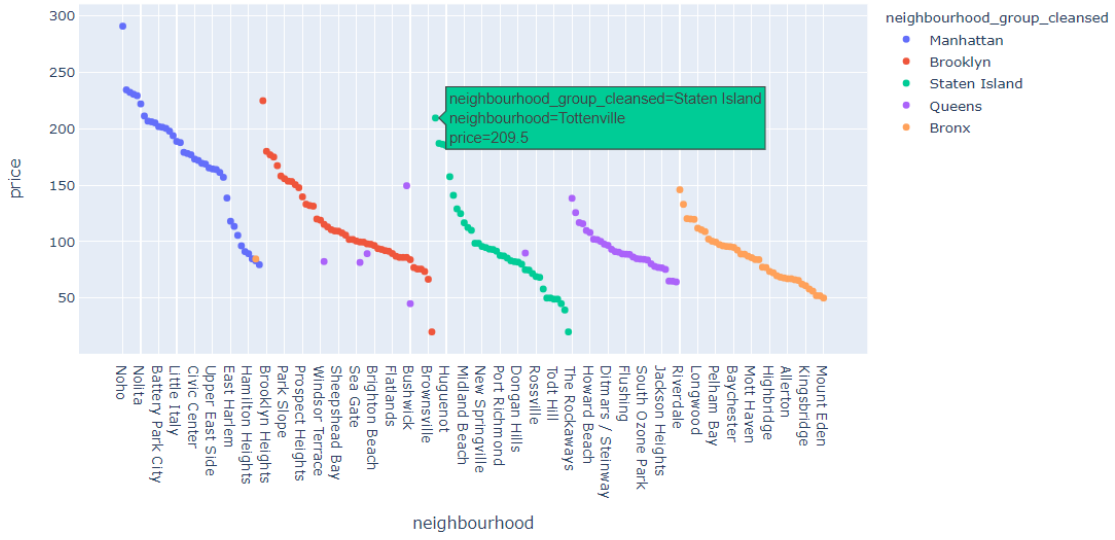

Figura 8. Scatterplot de precios medios de cada barrio.

Creación propia mediante Plotly.

De esta rápida forma podemos comparar de forma rápida la escala de los precios medios de cada barrio, ver en forma de ranking y poder compararlo con el resto de agrupaciones de barrios que hay. Por ejemplo, se puede ver que el barrio más caro es Noho en Manhattan, y los más baratos son Mill Basin en Brooklyn junto a Graniteville en Staten Island. Hay barrios más baratos en Staten Island que el más barato de Bronx… etc.

4.2.2. Características del alojamiento

Dejando de lado la situación posición geográfica del alojamiento, se pueden encontrar distintas características importantes en las cuales poder agrupar y estudiar cómo estas afectan al precio.

4.2.2.1 Tipo de alojamiento

Como tipo de alojamiento podemos encontrar 4 modalidades: casa/departamento entero, habitación privada, habitación compartida o habitación de hotel.

Por lo tanto, primero será ver qué tipo de estancia es la que más se ofrece y en qué proporción. Para esto es posible utilizar un clásico bar plot o un pie plot.

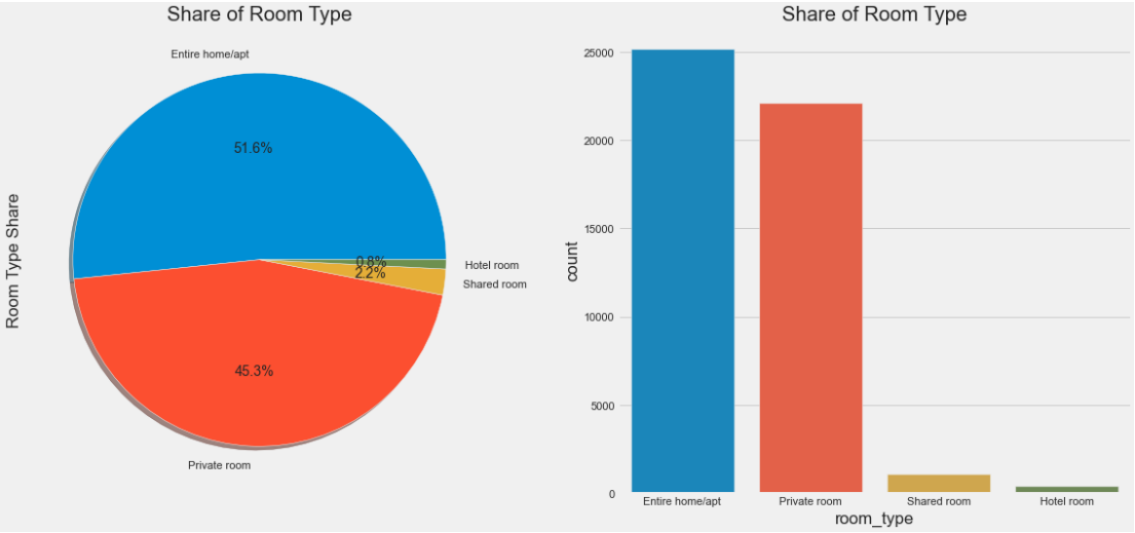

Figura 9. Distribución del tipo de vivienda.

Creación propia mediante Pyplot.

De esta forma comprobamos que las opciones de habitación compartida, como podría pasar en un hostel o la opción de habitación de hotel propia de otro tipo de negocio y que no tendría que formar parte de esta plataforma, serían de un mínimo 3% del total.

Si nos referimos al precio de cada una de estas formas de estadía se puede comprobar fácilmente con un cat plot.

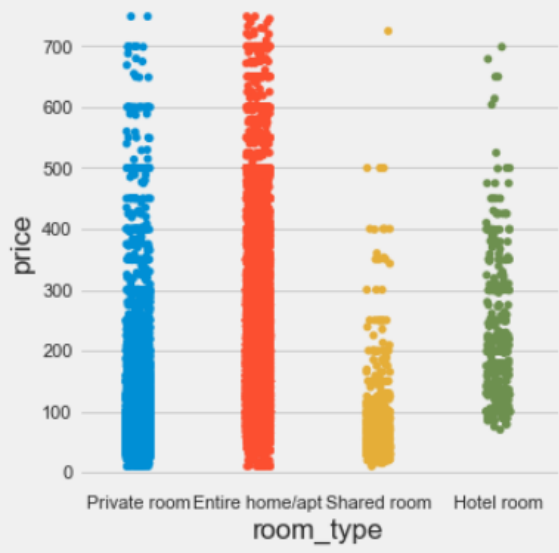

Figura 10. Distribución del tipo de vivienda.

Creación propia mediante Pyplot.

Existe un orden natural en cada una de los tipos, siendo lo más caro una habitación de hotel (el precio mínimo ronda los 80$) seguido por casa/apartamento entero, habitación privada y por último las habitaciones compartidas.

También se puede ver la distribución de los tipos por barrios, en el cual una conclusión que se puede sacar es que en Manhattan hay mayor número de casas/pisos enteros en oferta que habitaciones privadas, mientras que en Brooklyn pasa lo contrario.

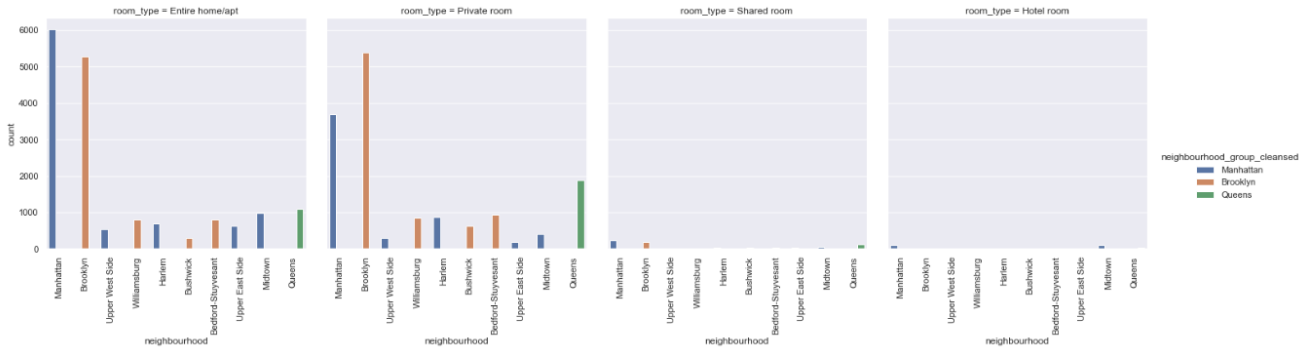

Figura 11. Distribución del tipo de vivienda.

Creación propia mediante Pyplot.

4.2.2.2 Valoración del alojamiento.

Al final de cada una de las estancias, el usuario puede dejar feedback valorando su estadía en diferentes puntos: limpieza, veracidad, localización, comunicación… Estos datos se resumen en la medida "review_scores_rating". En el siguiente gráfico se puede

ver como la valoración se relaciona con el precio, del cual se puede sacar alguna conclusión a priori.

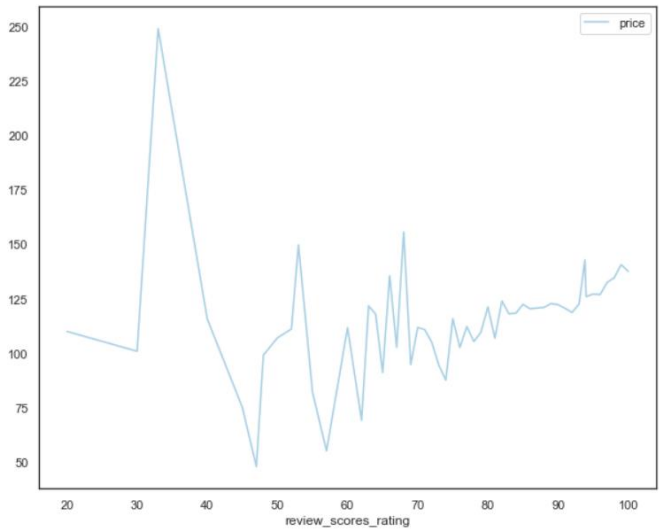

Figura 12. Evolución del precio medio dependiendo de la valoración. Creación propia mediante Pyplot.

La media de las buenas valoraciones (mayores a 50 puntos) tienden a ser mejores cuanto mayor sea el precio. Por el contrario, en las malas valoraciones, se ve un pico entre los 30 y 40 puntos y cuyo precio medio es muy superior al precio medio en Nueva York por una noche. Esto podría explicar que cuando en un alojamiento caro no se cumplen las expectativas, es cuando peores valoraciones se reciben y los usuarios se preocupan de dejar constancia de ello.

4.2.2.3 Tipo de vivienda.

Una de las variables que pueden tener gran variación de precio es el tipo de vivienda en el cual se va a alojar. No es un indicador per se ya que cada una tendrá sus cualidades y propiedades que podrán hacer una estancia más básica o más lujosa.

En el siguiente gráfico de barras se puede ver de forma ordenada, el precio medio de un tipo de vivienda más caro hasta el más barato.

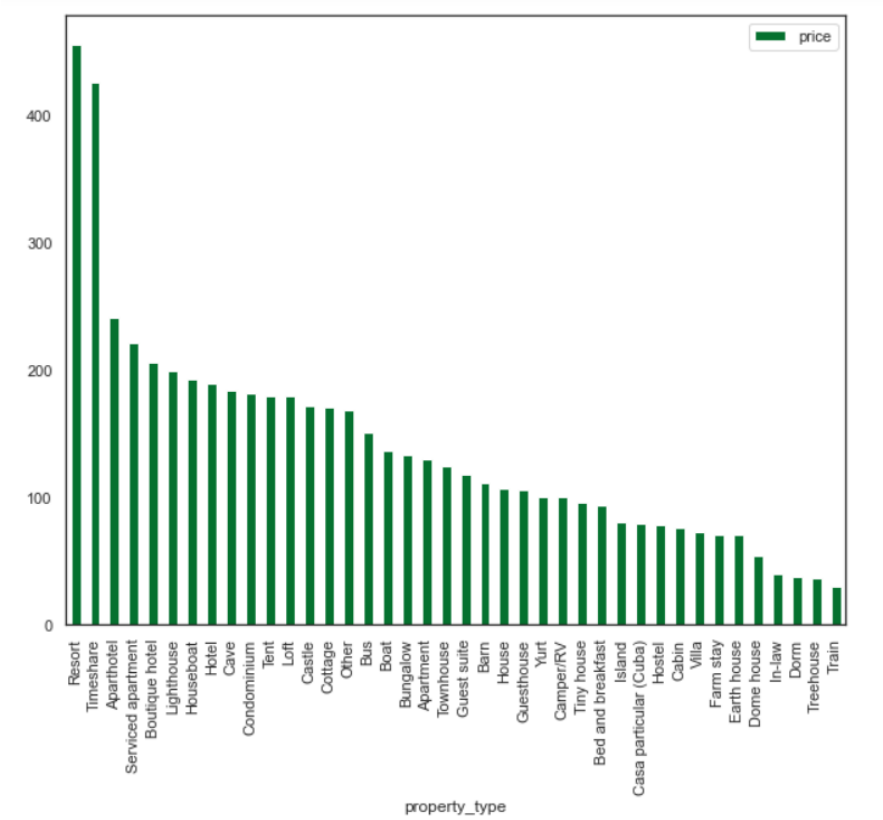

Figura 13. Precio medio por tipo de vivienda.

Creación propia mediante Pyplot.

Como manda a pensar el sentido común, hay grandes diferencias entre alojarse en un resort, en un loft o en un hostel. Con diferencias que son de media de más de 250$ el quedarse en un complejo turístico que en un apartamento.

4.2.3. Evolución temporal de precios

También se puede mirar cómo han ido evolucionando los precios en el último año (desde junio de 2019 a junio de 2020). Como se ha explicado anteriormente esta serie no se utilizará en el modelado de Machine Learning, o encontrar alguna posible situación de estacionalidad.

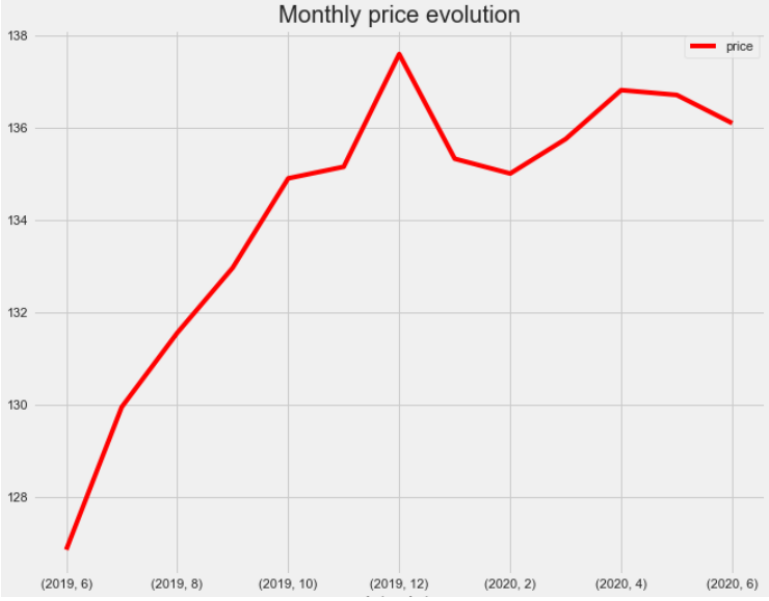

Figura 14. Evolución de los precios medios mensuales.

Creación propia mediante Pyplot.

Viendo este gráfico se pueden hacer algunas observaciones:

- Desde junio de 2019 a junio de 2020 se ve una subida del 7,08% del precio en el alquiler en Airbnb.
- En la primera mitad de la serie se ve una pronunciada tendencia de crecimiento de precios, llegando a su situación más alta en diciembre debido a que coincide con la época navideña. Luego se relaja la tendencia y empieza a asentarse en febrero, pero vuelve a subir en marzo. Esto podría plantear dos preguntas: ¿esto pasa por la tendencia al alza en los precios del alquiler? ¿es debido a la situación sanitaria del Covid-19 el cual se inicia a finales de marzo de 2020 a expandir en Nueva York como foco principal de afección?

En cualquier caso, una conclusión clara que se puede hacer de la serie de datos es que en ninguna época del año es más barato o caro el precio medio de un barrio con respecto al resto. Esto se refleja claramente en la siguiente serie temporal dividido por agrupaciones de barrios.

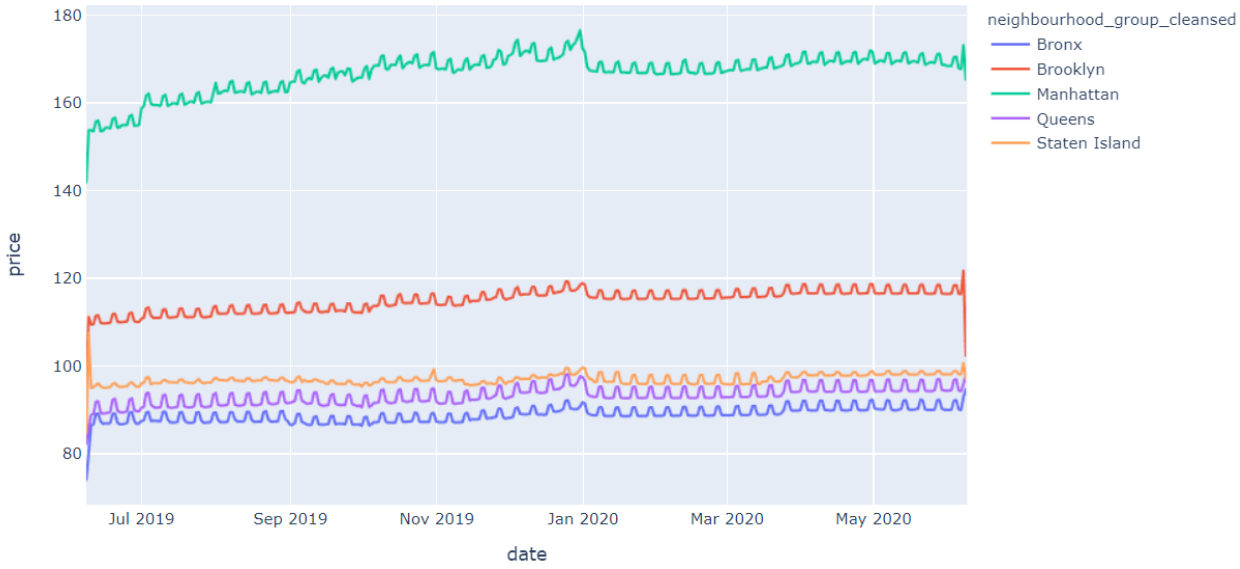

Figura 15. Evolución de los precios medios mensuales por agrupaciones de barrios. Creación propia mediante Plotly.

4.2.4. Otros análisis

4.2.4.1. Análisis de textos

Se puede utilizar la columna de títulos para ver cuáles son las palabras y reclamos más frecuentes en los anuncios. Mediante el módulo Wordcloud y una previa transformación y separación de las palabras contenidos en los títulos de los anuncios, se pueden retirar las palabras comunes en inglés que no aportan información al analizar texto (is, of, in, that…) mediante stopwords, y de esta forma luego poder crear una visualización con las palabras más usadas diferenciadas por el tamaño.

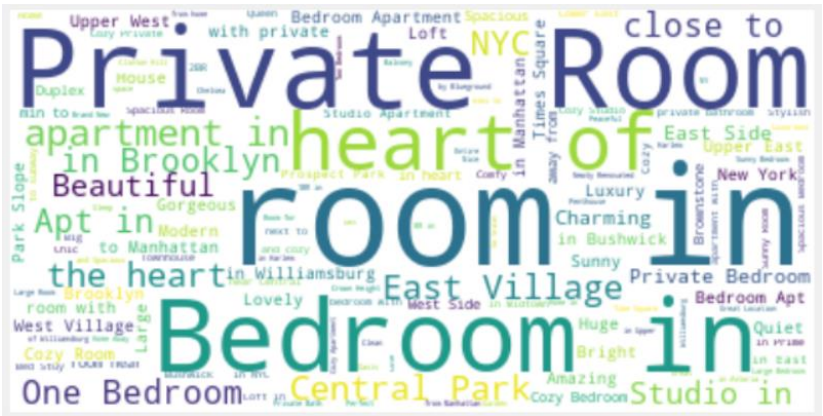

Figura 16. Palabras más usadas en los anuncios. Creación propia mediante Wordcloud.

Lo interesante de esto es diferenciar como se anuncian los sitios más económicos, y los sitios más caros:

Figura 16. Palabras más usadas en los anuncios de menos de 50$.

Creación propia mediante Wordcloud.

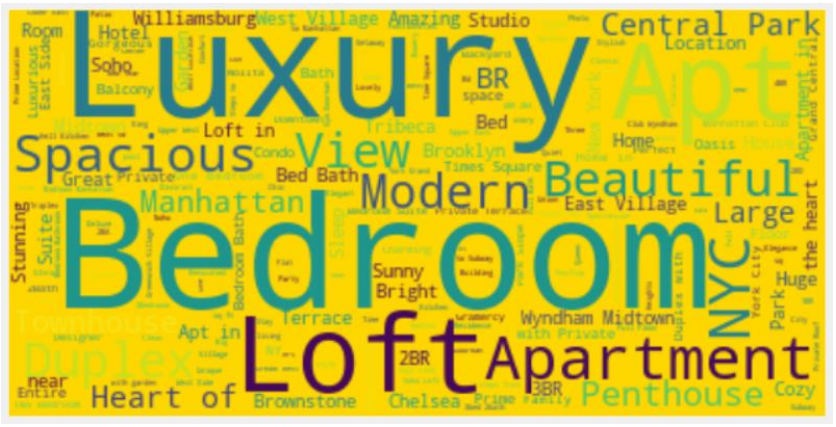

Figura 17. Palabras más usadas en los anuncios de más de 400$.

Creación propia mediante Wordcloud.

Mientras que en los sitios más económicos se hace referencia especialmente a que es una habitación privada, que la habitación está situada en tal sitio, que es un sitio acogedor o referencias a Brooklyn… para los sitios con los precios más altos se hace referencia a palabras como lujo, loft, habitación (características, no localización), las vistas y características del sitio.

4.2.4.2. Abusos en el uso de la plataforma

Se dan casos en los cuales, utilizando la plataforma como mediador de contactos, se utiliza de forma incorrecta el servicio. Mirando la columna de host_id, se puede ver que ID's tienen más ofertas diferentes de alojamiento en Airbnb, y poder de esta forma ver quién podría estar utilizando y la plataforma para hacer negocio inmobiliario sin el

consecuente pago de impuestos que vendría de una actividad de tal tipo. Se ha mencionado anteriormente: no es objetivo el hacer una investigación de la legalidad de estos alojamientos. Solo el uso de las herramientas y los datos disponibles para ver de una manera visual cuales serían estas ID.

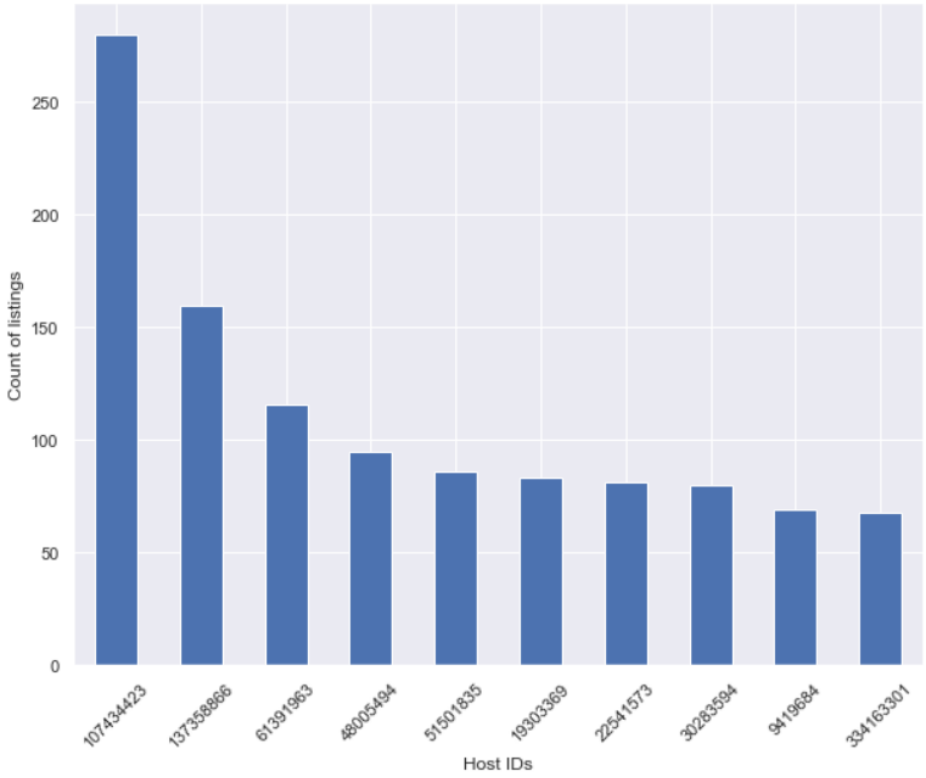

Figura 18. Top 10 de anuncios distintos por ID

Creación propia mediante pyplot.

Como se ve, hay algunas ID de host que tienen más de 150 ofertas de alojamiento y si fuera necesario para un departamento antifraude, se podrían obtener estos datos.

4.2.4.3 Correlaciones

En este caso, resultará más fácil comprobar en el notebook las relaciones que se quieran buscar debido a la gran cantidad de columnas. En cualquier caso, las correlaciones fuertes serán más claras en sentido positivo u oscuras en las de sentido negativo. Las que se mantengan cerca del color rojo/morado serán las que tienen correlación muy débil.

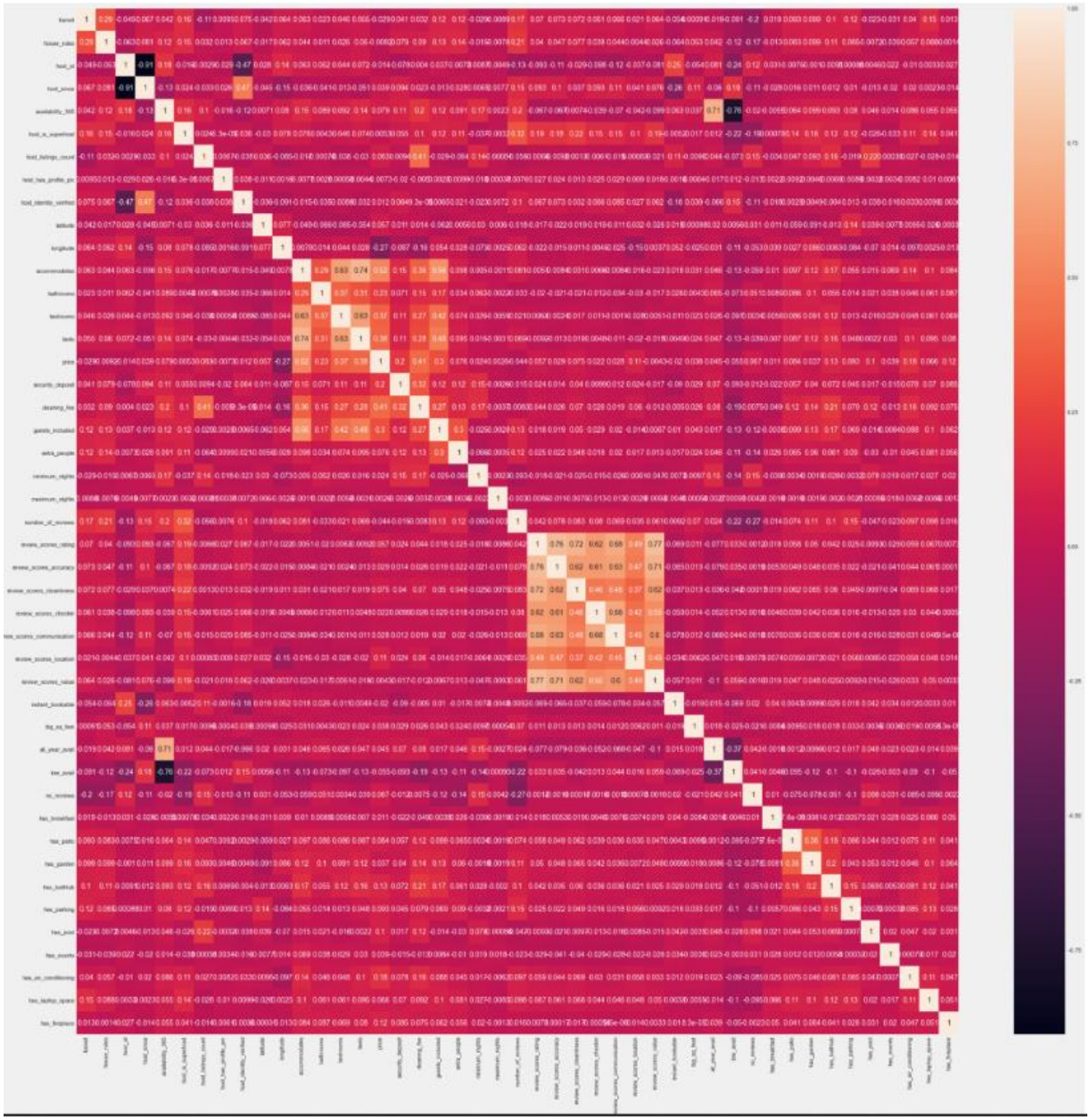

Figura 19. Tabla de correlaciones

Creación propia mediante Seaborn.

Se pueden ver dos grupos a distinguir que serían:

- Las reviews entre si: al obtener buena puntuación en uno de los aspectos, se puede sobreentender que también se dará buenas puntuaciones en el resto de aspectos.
- Se ve una correlación del precio con respecto a: la longitud, el número de habitaciones, baños, dormitorios, camas, tarifa de limpieza y los guests incluidos.

4.3 Aplicación de Machine Learning

4.3.1 Preprocesado

Antes de plantear los objetivos y aplicar aprendizaje supervisado y no supervisado es necesario hacer un preprocesado de la información que tenemos para eliminar las variables categóricas que no sirvan o transformar las que ya tenemos.

Por lo tanto para empezar, se eliminarán las variables categóricas: "name", "description", "host_id", "amenities" (tener en cuenta que los datos que interesaban de amenities se mantenien como valores binarios en las columnas dummies).

Se categorizarán manualmente de forma ordinal las columnas “cancellation_policy”, “bed_type”, “room_type” y “neighbourhood_group_cleansed” en forma numérica.

De la misma forma, se ordenan los barrios del más caro al más barato en precio medio y se les asigna un número dependiendo de su posición en el ranking.

De esta forma solo quedarían el resto de variables que se mantienen en forma booleana (Verdadero o falso) y que mediante la función LabelEnconder() dentro de sklearn, que codifica en forma de 1 y 0.

Tras realizar todas las transformaciones descritas solo quedarían caracteres numéricos en el dataframe transformado.

Con esto se podrá dividir los datos en X e Y, es decir, las variables independientes y la variable dependiente. Todas las variables excepto ‘price’ irán a X y esta última a Y.

Además, se dividirán mediante la función train_test_split(), también de sklearn, en conjuntos de entrenamiento y test con proporción 80% -20% como suelen marcar las buenas prácticas que se asocian al principio de Pareto (Wikipedia, 2015).

De esta forma ya tendríamos los 4 conjuntos que se utilizarán en los distintos modelos de aprendizaje automático: X_train, X_test, Y_train e Y_test.

4.3.2 Aprendizaje supervisado

En este estudio se utilizarán modelos de regresión con los cuales predecir cuáles serán los precios de alquiler por noche atendiendo a las distintas variables que se puedan presentar.

Hay que tener en cuenta que el análisis predictivo se utiliza para identificar tendencias, correlaciones y causalidad. Este tipo de análisis va un paso adelante de los análisis descriptivos y diagnósticos. El análisis predictivo utiliza los datos que hemos resumido para hacer predicciones lógicas de los resultados de los eventos.

Para poder, entonces, evaluar la fiabilidad o desempeño del modelo se utilizarán las siguientes medidas de calidad de modelos:

- Error cuadrático medio (MSE, *Mean Squared Error*): se define como el valor medio de la diferencia entre la variable independiente y la predicción al cuadrado. Este estimador mide los errores al cuadrado, penalizando los puntos en los que la discrepancia entre la variable independiente y la predicción es mayor. Su fórmula es la siguiente:

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

- Coeficiente de determinación o R^2: se puede interpretar como el porcentaje de la variabilidad total de la variable dependiente respecto a la media que se puede explicar con el modelo. Idealmente su valor debería ser 1, lo que indicaría que el modelo reproduce perfectamente los datos del conjunto de entrenamiento. Su fórmula es la siguiente:

$$R^2 = \frac{\sum_{t=1}^{T}(\hat{Y}_t - \bar{Y})^2}{\sum_{t=1}^{T}(Y_t - \bar{Y})^2}$$

Definido como se evaluarán los modelos, es hora de definir que modelos de regresión se utilizarán.

4.3.2.1 Regresión lineal (Linear regression)

4.3.2.1.1. Regresión lineal simple

La regresión lineal es usada para aproximar la relación de dependencia entre una variable dependiente Y, las variables independientes Xi y un término aleatorio o error ε.

$$\gamma = \alpha \pm \beta\chi + \varepsilon_i$$

Aplicando el modelo a los conjuntos de entrenamiento y test mediante la librería de sklearn podemos obtener los siguientes coeficientes para las variables más importantes según este modelo:

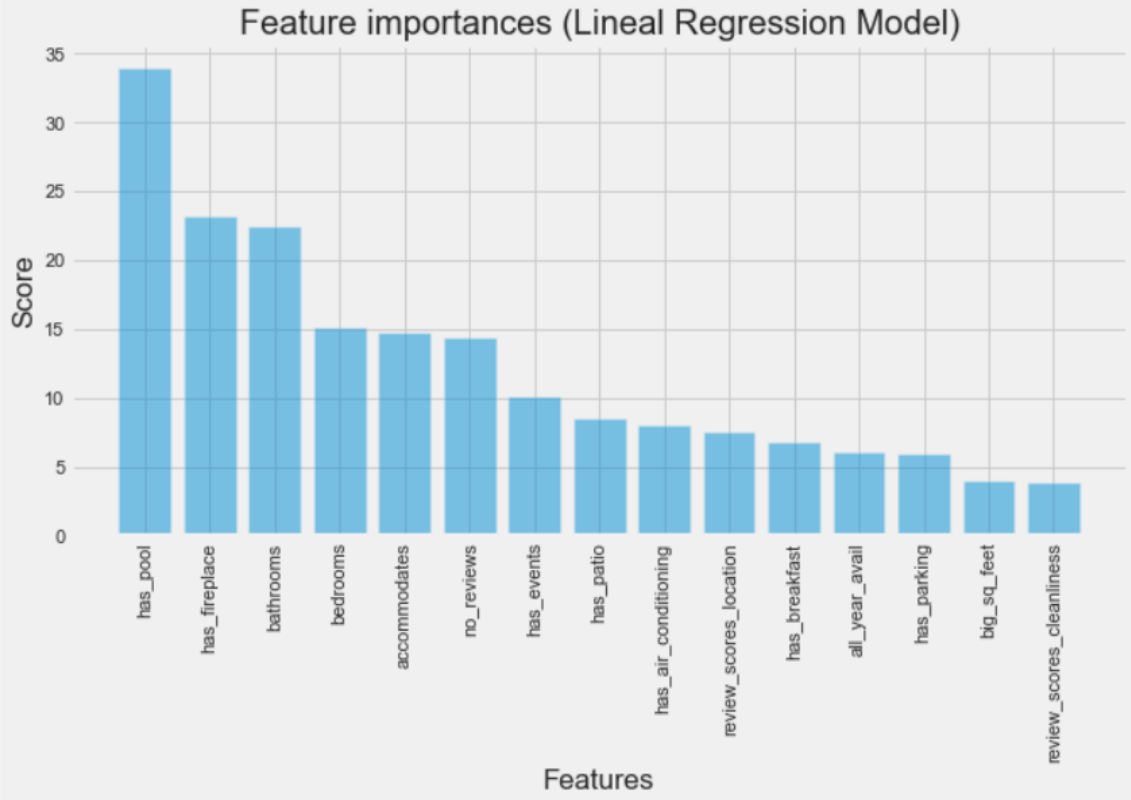

Figura 20. Coeficientes en el modelo de regresión lineal
Creación propia mediante Pyplot.

Aquí se puede ver que afecta de forma muy grande el que el alojamiento tenga piscina o chimenea, el número de baños, dormitorios y habitaciones o que no tenga reviews anteriores la oferta.

Si se quisiera más detalles sobre el mismo tipo de regresión se puede recurrir a otras funciones para hallar, por ejemplo, los mínimos cuadrados ordinarios como se deja marcado en el notebook.

Las métricas de este modelo son:

	algorithm	CV error	CV std	training error	test error	training_r2_score	test_r2_score
0	Linear Regression	4628.695353	251.450541	4609.771522	4599.067258	0.50757	0.507279

4.3.2.1.2. Regresión lineal con selección de características

Además se puede hacer una selección de características mediante regularización en las funciones de esfuerzo, para reducir el sobreajuste mediante las regresiones Ridge y LASSO (least absolute shrinkage and selection operator),.

En la regresión Ridge el valor natural de los coeficientes es 0, penalizando la adjudicación de valor. En este modelo no se obtienen coeficientes finales nulos, aunque sean muy pequeños. Es, por tanto, un modelo que regulariza de manera continua todos los coeficientes.

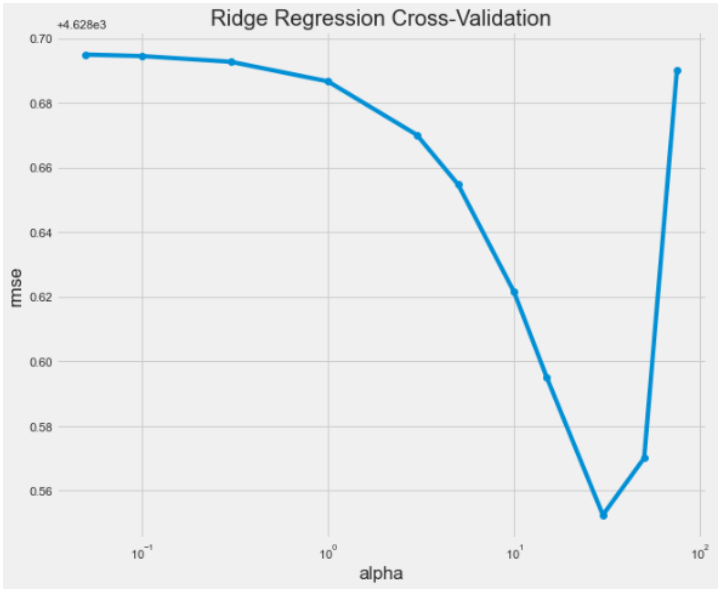

Figura 21. Cross-validation en la regresión Ridge

Creación propia mediante Pyplot..

Las métricas de este modelo son:

	algorithm	CV error	CV std	training error	test error	training_r2_score	test_r2_score
0	Ridge Regression	4628.651124	251.476949	4609.773763	0.0	0.50757	0.507292

De forma parecida la regresión LASSO, penaliza los parámetros que requieren mayor peso, permite seleccionar las características eliminado aquellas que necesitan valores de los parámetros más grandes para poder aportar información al modelo.

Para ambos modelados se busca el valor óptimo de alpha, en el cual el rmse sea el más bajo. Hallando el mejor valor de alpha se dará con la mejor versión del modelo para los datos proporcionados.

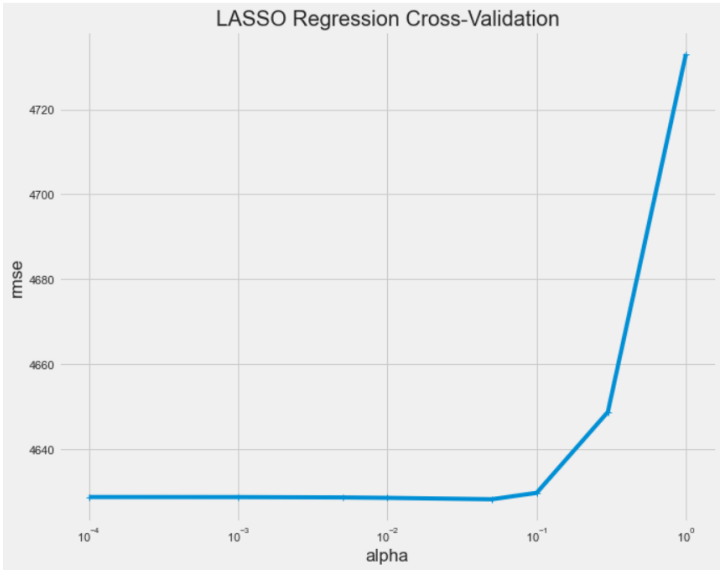

Figura 22. Cross-validation en la regresión Lasso

Creación propia mediante Pyplot..

Con respecto a la regresión linear simple, de las 47 variables LASSO elimina 5. Los coeficientes acaban distribuidos de esta forma.

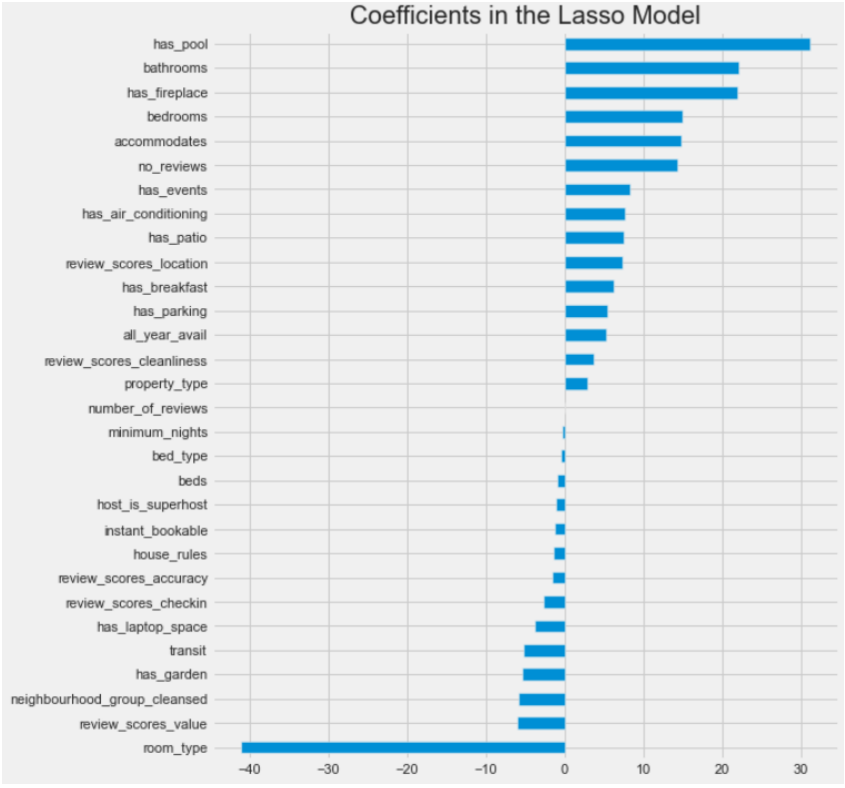

Figura 23. Coeficientes en la regresión Lasso.

Creación propia mediante Pyplot..

Las métricas de este modelo son:

	algorithm	CV error	CV std	training error	test error	training_r2_score	test_r2_score
0	LASSO Regression	4628.164668	251.463255	4610.7372	4600.222652	0.507467	0.507156

4.3.2.2 Random Forest Regressor

Es una combinación de árboles predictores tal que cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos. Se ajusta a varios árboles de decisión de clasificación en varias submuestras del conjunto de datos y utiliza el promedio para mejorar la precisión predictiva y controlar el sobreajuste.

En cuanto a sus desventajas, los árboles de decisión pueden crear modelos complejos que no generalicen bien los resultados, es decir, se llega fácilmente a modelos que sobreajustan el conjunto de entrenamiento. Esto es debido a que, si el entrenamiento no se realiza cuidadosamente, los modelos obtenidos pueden llegar a generar una regla específica para cada uno de los casos en el conjunto del entrenamiento. Este problema se puede mitigar fijando la profundidad a la que se puede llegar.

Mediante RandomizedSearchCV se itera en el modelo para encontrar los mejores parámetros que se usarán posteriormente en el best model. Además, usando el módulo eli5 se puede ver los pesos que tienen las variables en el modelo final.

Weight	Feature
0.1377 ± 0.1828	room_type
0.1130 ± 0.1761	accommodates
0.0901 ± 0.0803	neighbourhood
0.0863 ± 0.1278	cleaning_fee
0.0821 ± 0.1403	bedrooms
0.0548 ± 0.0453	bathrooms
0.0424 ± 0.0570	neighbourhood_group_cleansed
0.0412 ± 0.0948	beds
0.0312 ± 0.0058	host_since
0.0295 ± 0.0166	property_type
0.0246 ± 0.0064	minimum_nights
0.0245 ± 0.0460	guests_included
0.0235 ± 0.0282	security_deposit
0.0214 ± 0.0112	extra_people
0.0209 ± 0.0084	host_listings_count
0.0191 ± 0.0042	availability_365
0.0181 ± 0.0046	maximum_nights
0.0166 ± 0.0048	number_of_reviews
0.0116 ± 0.0105	review_scores_location
0.0115 ± 0.0035	review_scores_rating
	… 27 more …

Figura 24. Pesos de las variables en el modelo de Random Forest.

Creación propia mediante Eli5..

Las métricas de este modelo son:

	algorithm	CV error	CV std	training error	test error	training_r2_score	test_r2_score
0	Random Forest Regressor	3517.82673	192.007065	795.553294	3353.04855	0.915017	0.640772

4.3.2.3 Gradient Boosting

- Gradient Boosting Regressor

Construye un modelo aditivo de manera progresiva por etapas; Permite la optimización de funciones arbitrarias de pérdida diferenciable. En cada etapa se ajusta un árbol de regresión en el gradiente negativo de la función de pérdida dada.

Las métricas de este modelo son:

	algorithm	training error	test error	training_r2_score	test_r2_score
0	Gradient Boosting Regressor	3272.506193	3477.341323	0.650421	0.627455

- Extreme Gradient Boosting (XGBoost)

XGBoost es una implementación específica del método Gradient Boosting que ofrece aproximaciones más precisas al usar las fortalezas de la derivada de segundo orden de la función de pérdida, la regularización L1 y L2 y la computación paralela mejorando las capacidades de generalización del modelo.

Las métricas de este modelo son:

	algorithm	CV error	CV std	training error	test error	training_r2_score	test_r2_score
0	XGBoost	3355.462523	152.593475	1422.217177	3175.920854	0.848074	0.659748

4.3.2.4 Selección del mejor modelo

Atendiendo a las métricas obtenidas, y habiendo decidido que el mejor modelo se iba a basar en el mejor R2 o en el menor MSE: para ambas medidas seleccionaríamos el XGBoost. Hay que destacar que, aunque en el conjunto de entrenamiento el modelo de Random Forest explica mejor los datos, será imperante que el desempeño sea el mayor en el conjunto de test.

Por lo cual, ya podríamos utilizar este modelo para predecir los precios según las características de una nueva oferta de alojamiento.

	algorithm	CV error	CV std	training error	test error	training_r2_score	test_r2_score
0	Linear Regression	4628.695353	251.450541	4609.771522	4599.067258	0.507570	0.507279
1	Ridge Regression	4628.651124	251.476949	4609.773763	4598.954488	0.507570	0.507292
2	LASSO Regression	4628.164668	251.463255	4610.737200	4600.222652	0.507467	0.507156
3	Random Forest Regressor	3517.826730	192.007065	795.553294	3353.048550	0.915017	0.640772
4	Gradient Boosting Regressor	NaN	NaN	3272.506193	3477.341323	0.650421	0.627455
5	XGBoost	3355.462523	152.593475	1422.217177	3175.920854	0.848074	0.659748

En los siguientes gráficos se pueden ver visualmente la dispersión de las predicciones del modelo, las barras de error (las primeras 20 predicciones) asi como las 15 variables con más peso en el modelo.

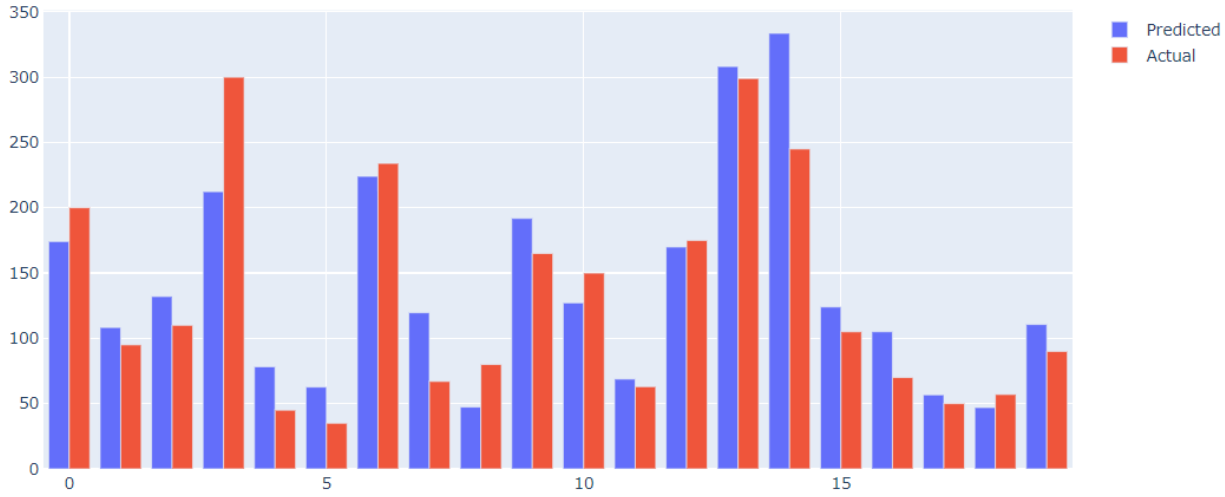

Figura 24. Gráfico de precios reales y predichos.

Creación propia mediante Plotly.

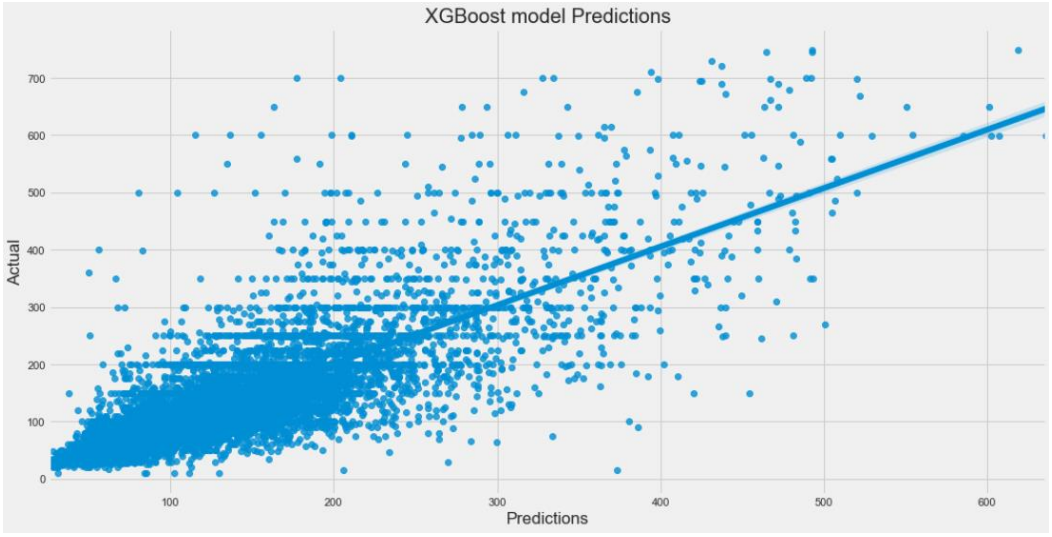

Figura 25. Gráfico de dispersión entre precios reales y predichos.

Creación propia mediante Pyplot.

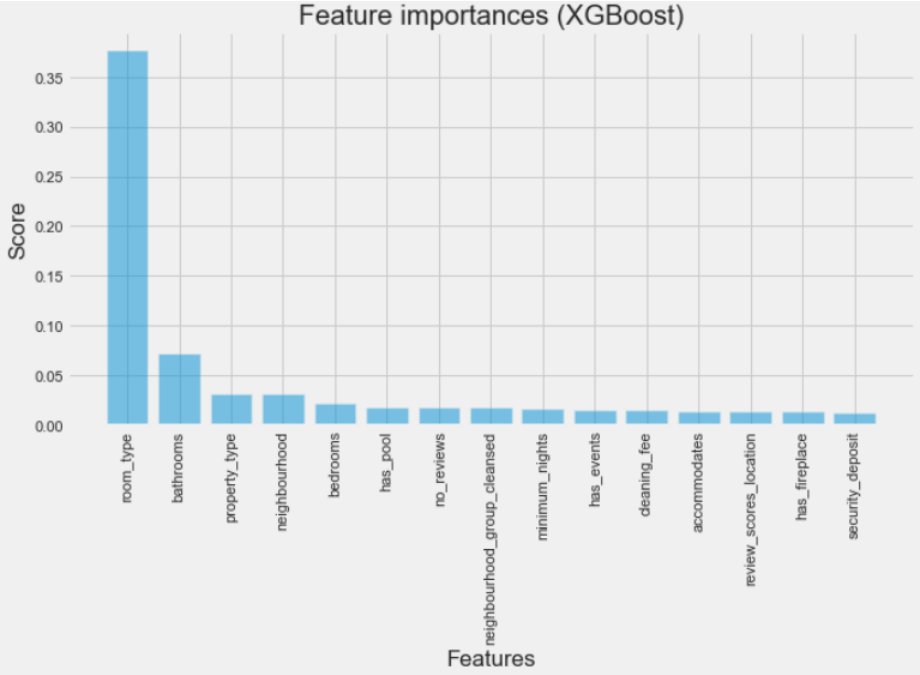

Figura 26. Importancia de las variables en el modelo XGBoost.
Creación propia mediante Pyplot.

Interpretando las importancias de las variables, lo que más va a ayudar a predecir el precio de un alojamiento será el tipo de estadía que se ofrezca, el número de baños, el tipo de propiedad y el barrio al que pertenezca, seguidos de que si tiene piscina… etc

4.3.3. Aprendizaje no supervisado

Con objetivo de clasificar los datos y agruparlos en clústeres se usará la herramienta K-means de Sklearn. K-means es un algoritmo de clasificación no supervisada que agrupa objetos en k grupos basándose en sus características. El agrupamiento se realiza minimizando la suma de distancias entre cada objeto y el centroide de su grupo o cluster. El algoritmo consta de tres pasos: inicialización, asignación objetos a los centroides y actualización centroides.

Las variables que se utilizarán son las siguientes: "property_type", "room_type", "accommodates", "bathrooms", "bedrooms", "price", "has_pool", "has_garden", "big_sq_feet", "has_fireplace". Buscando de esta forma que la clasificación sea solo dependiendo de las características físicas que tiene la vivienda.

Para poder obtener el número óptimo de clústeres, se utiliza la función silhouette y una representación gráfica de la puntuación en distintos números de clústeres. Nos

quedaremos con el cual el score sea el mayor. En este caso el número óptimo es de 2 clústeres.

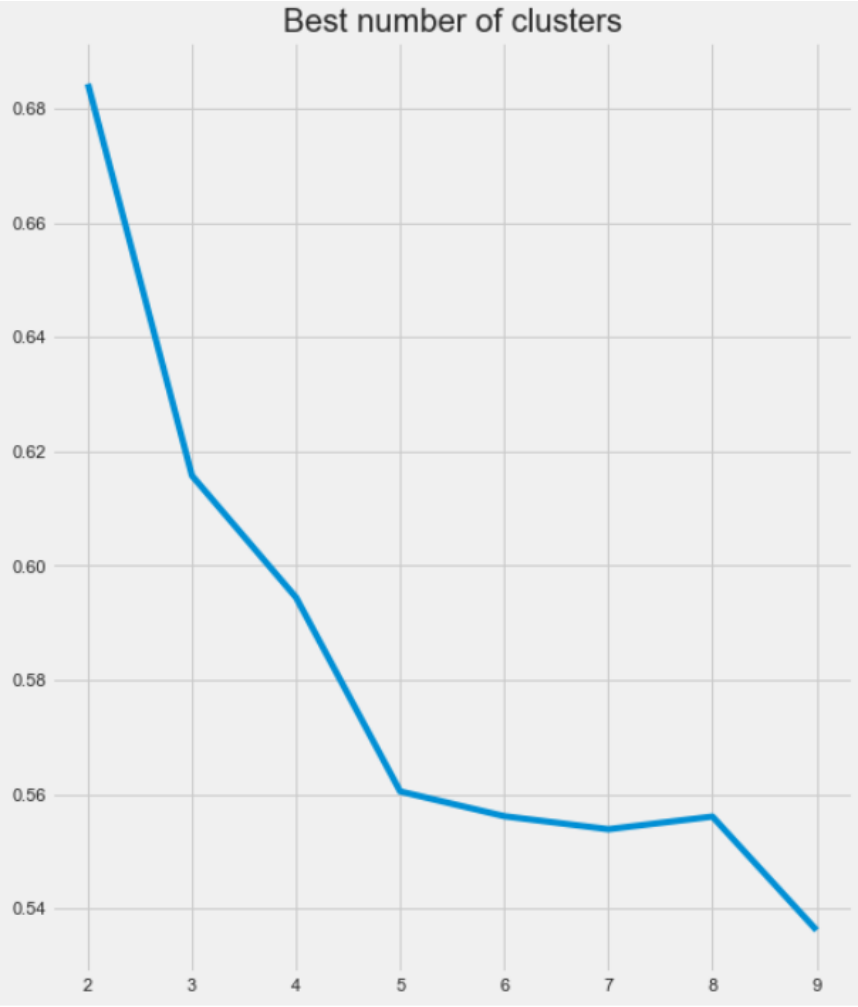

*Figura 27. Gráfico del método S*ilhouette

Creación propia mediante Pyplot.

Tras la aplicación del algoritmo y de la obtención de los centroides, ya se puede acceder a un listado los cuales son contenidos en el clúster 0 o en el 1. Para verlo de forma gráfica, se utiliza la latitud y la longitud en conjunto de un mapeado de polígonos hecho con los datos contenidos en el geojson.

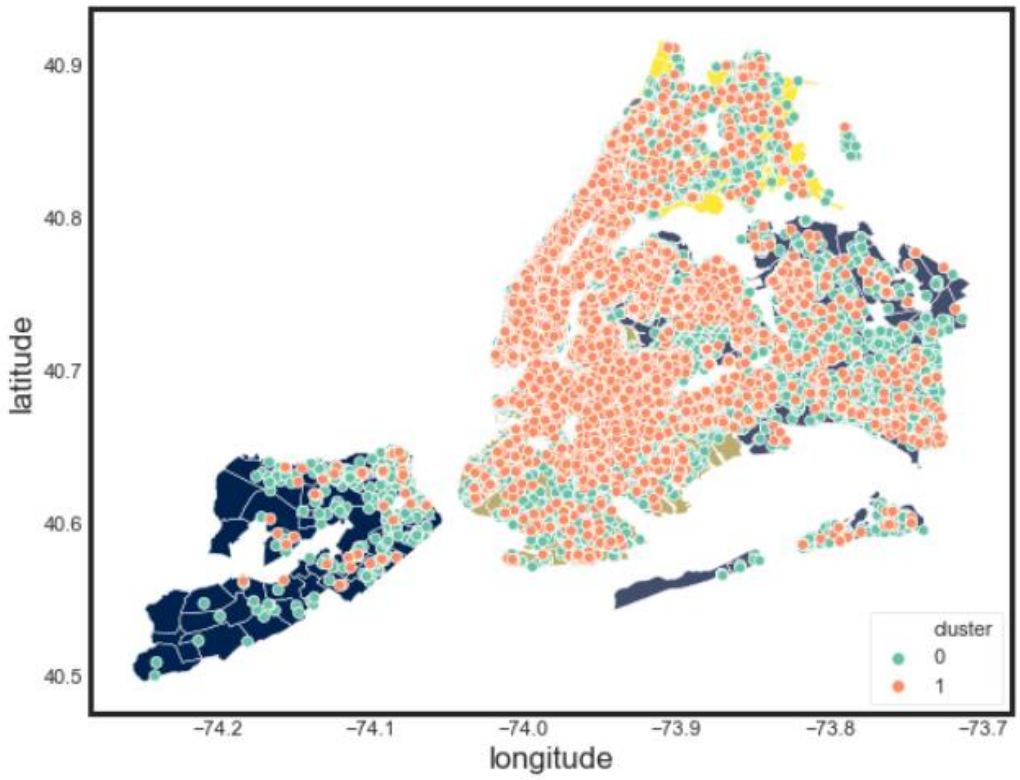

Figura 27. Representación en el mapa de Nueva York del clustering
Creación propia mediante Seaborn.

Como puede se puede ver, el algoritmo divide dependiendo de las características de la casa, y coincide con que el clúster 0 se encuentra mayormente en las zonas más exteriores de Nueva York, mientras que del clúster uno serían prácticamente todo Manhattan y Brooklyn.

5. Conclusión

Tras el extensivo análisis del dataset del que se dispone, se ha podido encontrar interesantes relaciones las cuales se han cubierto de forma estadística y visual, se ha visto de forma global la distribución de alojamientos por barrio y las distintas variables más determinantes en relación a determinar el precio por noche en los alojamientos de Airbnb.

También se ha creado un modelo para predicción de precios que podría ser usado internamente por la plataforma para recomendar a los nuevos hosts un precio dependiendo de las características que tenga el alojamiento o usarlo de forma externa para dependiendo de nuestras expectativas, buscar que precio sería el justo dependiendo de las distintas características buscadas.

Por último, se ha podido utilizar la función de clustering para dividir cada uno de los registros en uno de los dos clústeres atendiendo a las características físicas del alojamiento.

Referencias

Airbnb. (2018). *Airbnb Local law 146*. Obtenido de https://www.airbnb.es/help/article/868/nueva-york-nueva-york

Guttentag, D. (30 de Agosto de 2018). *Qué impacto tiene en las ciudades Airbnb, la controvertida plataforma de alquiler temporal para turistas*. Obtenido de https://www.bbc.com/mundo/noticias-45355426

Kozyrkov, C. (22 de Diciembre de 2018). *Medium*. Obtenido de https://medium.com/datos-y-ciencia/qué-diablos-es-ciencia-de-datos-f1c8c7add107

Liu, A. (2015). *Wikipedia*. Obtenido de https://es.wikipedia.org/wiki/Ciencia_de_datos#cite_ref-IBM_1-0

Patil, P. (23 de Marzo de 2018). *What is Exploratory Data Analysis?* Obtenido de https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15

Simeone, O. (5 de Noviembre de 2018). Obtenido de https://arxiv.org/pdf/1808.02342.pdf

Wikipedia. (13 de Junio de 2015). *Principio de Pareto*. Obtenido de https://es.wikipedia.org/wiki/Principio_de_Pareto