EAE Business School

Data Science & Advanced Analytics

Propuesta de Trabajo Final

Autores:

Karla Rojas Vélez
Dennier Ágreda López
María José Velandia Zambrano

Abril 2020

Contents

1 Resumen

1.1 Motivación

Los desarrollos tecnológicos han sido de gran ayuda para poder entender las preferencias de las personas. Las nuevas generaciones están acostumbradas a ver y conocer el mundo a través de la web; con la ayuda de estas tecnologías, viajar es cada vez más fácil y enriquecedor. Por otro parte, la dinámica tecnológica también ha permitido generar nuevas opciones para ofrecer servicios, como lo es el de hospedaje, siendo una oportunidad para más personas que quieren generar ingresos adicionales, dando en alquiler habitaciones del lugar donde viven.

En general, este nuevo escenario ha modificado la forma en la que se hace turismo y como se gestionan los servicios relacionados a ello, tantos oferentes como demandantes de espacios de hospedaje se han visto en la necesidad de adaptarse a este nuevo enfoque.

De ahí que, nuestro objetivo es atacar los dos frentes, tanto como el de los usuarios como de los que ofrecen el servicio de hospedaje. Por el lado del usuario, desarrollaremos un sistema de recomendación híbrido (usando información de los viajes anteriores de los usuarios, la información de otros usuarios y la descripción de los sitios), que ayude a los viajeros a tomar una decisión de una forma más rápida, simple e inteligente sobre su hospedaje, considerando variables como la localización, descripción de los lugares ofertados, gustos y preferencias de cada persona. Por otro lado, desde una perspectiva del oferente, vamos a realizar un predictor de precios, que se adecue a las características de sitio que quiere ofrecer.

El impacto de este cambio de paradigma para las empresas hoteleras ha sido tan significativo, que muchas de ellas han decidido agregar dentro de la carta de sus servicios, como una opción adicional el participar dentro de estas aplicaciones.

1.2 Problema

Al realizar un viaje, uno de los limitantes es encontrar un hospedaje que se adapte a nuestras necesidades. La creación de nuevos modelos de negocio, como lo es Airbnb, han ayudado a encontrar un hospedaje en ciudades o sitios que son desconocidos y también ha permitido a las personas ofrecer dichos lugares como un medio de ingresos adicionales.

Sin embargo, a pesar de todas estas ayudas tecnológicas, aún sigue siendo hasta cierto punto engorroso encontrar algo que se acople a nuestras necesidades, esa labor toma tiempo al buscar página a página lo que se desea y muchas veces se elige lo primero que se encuentra por el tiempo que demanda esta actividad. Por esto, la creación de un sistema de recomendación es importante para personalizar aun más estos productos de una forma eficaz. De esta forma viajar, descubrir el mundo y concebir la vida desde una perspectiva más global resulta más sencillo.

En el otro lado, tenemos a las personas que quieren iniciar en el alquiler de habitaciones[1], y que no tienen una logística detrás que permita optimizar un *pricing* ideal para ofrecer este servicio, pensando en ello es que resulta indispensable tener una herramienta que facilite la forma de generar un tablero de precios acorde a las características del lugar.

1.3 Revisión

La iniciativa de la creación de una API de Airbnb[2], muchas personas utilizan estos datos para diversos propósitos, tanto para análisis enteramente descriptivos, así como el desarrollo y planteamiento de modelos recomendación, agrupamiento y regresión. Favoreciéndose de la cuantiosa información valiosa que se almacena (tanto del lugar y sus características, como los comentarios que recibe de parte de los usuarios).

Para nuestro análisis en relación al sistema de recomendación hemos uti-

[1] **Tanto familias como empresas utilizan el canal de Airbnb para alquiler**
[2] **API Airbnb:** http://insideairbnb.com/get-the-data.html

lizado 5 diferentes fuentes, en la cuales, realizaron análisis de un sistema de recomendación para Airbnb en distintas ciudades, diferentes técnicas, diferentes atributos y objetivos y también en otras fuentes.

- Royan (2018) fue una de nuestras fuentes bases para el sistema de recomendación. Utiliza una análisis de *TF-IDF (Term Frequency - Inverse Document Frequency)* sobre el nombre y la descripción del lugar, usando como métrica en su análisis el *Coseno Similitud*, para encontrar la semejanza de cada sitio.

- Usando el informe Olivares pudimos apreciar que utiliza el aprendizaje supervisado para crear distintos sistemas de recomendación con varias técnicas, su base de datos se basa en posiciones de trabajos, posiciones de interés entre otros.De la misma forma como lo usa Royan, utiliza el *Coseno Similitud*, como métrica de análisis y la aplica en 4 diferentes sistemas de recomendación: TF-IDF (Term Frequency - Inverse Document Frequency), CountVectorizer, Spacy (pre-entrena con vectores de palabras) y KNN (K Nearest Neighbors).

- Prateek, nos enseña una nueva metodología de redes neuronales para PLN, utilizando el *word2vec* predice las palabras cercanas para cada palabra de una oración, para medir los pesos de cada palabra.

- Aisulu (2019) en su informe *Alternative Ways to Recommend Airbnb Listings Using Natural Language Processing*, realiza un análisis de sentimientos y aplica la métrica del *Coseno Similitud*, para evaluar los comentarios de los Airbnb y de esta forma predecir una recomendación para los usuarios.

- Grbovic et al. (2018) usa una técnica llamada *Embedding* para codificar las distintas variables como ubicación, precio, tipo de habitación, arquitectura y estilo. Utiliza un algoritmo de Kmedias para clusterizar al los sitios con todas las características antes mencionadas y con el *Coseno Similitud* encuentra la semejanza en tiempo real de los sitios de alquiler.

- En referencia para el modelo de predicción de precios, Sharma, nos muestra distintos análisis: Regresiones lineales, árboles de desición, regresión usando vectores de soporte y un booter del gradiente.

- Lahiri, realiza una modelo predictivo de precios en base a la distancia de *Haversine* (calcula el círculo de distancia entre un punto de la tierra y el punto (0, 0)) y de esta forma con el resto de atributos del sitio modela el predictor de precios.

1.4 Propuesta

Conociendo la problemática ya descrita, hemos desarrollo un sistema de recomendación para los viajeros que deseen encontrar hospedaje en las ciudades de España.[3] Así como también, un algoritmo de soporte de precios para personas que quieran empezar a ofrecer un espacio es casa, y que no tengan idea qué precio cobrar.

Por el lado del sistema de recomendación, partiendo que se conoce de donde se ha hospedado el usuario con anterioridad, el objetivo del modelo es recomendarle de acuerdo a su itinerario de viaje, un hospedaje que se alinee con sus expectativas.

En el otro lado, considerando las características de los lugares que se han alquilado se propone cuál debería ser el precio adecuado a cobrar en función a lo que el potencial oferente plantea materializar.

1.5 Evaluación

Para el sistema de recomendación se utiliza como medida de evaluación a la *similitud coseno*, la cual es una medida de la similitud existente entre dos vectores en un espacio que posee un producto interior con el que se evalúa el valor del coseno del ángulo comprendido entre ellos. Jurafsky and Martin (2019).

Esta aplicación nos permite determinar cuán de similares son dos documentos en función a la distancia que subyace entre ambos, siendo una de las

[3] **Todas las que están en la API InsideAirbnb**

más usadas para estos fines.

Mientras que, por el lado del algoritmo de precios se utiliza diversas medida de bondad de ajuste, como lo son el coeficiente de determinación (R^2). Éste, es un estadístico alternativo para medir la bondad de ajuste, toma la forma de una proporción: la proporción de la varianza explicado, por lo que siempre toma un valor entre 0 y 1, y es independiente de la escala de Y. James et al. (2013).

Mientras más se acerca a 1, implica que mejor ajuste respecto a la serie original tiene el modelo. Adicionalmente, se utilizan otras medidas como *RMSE* y *MSE*, que al contrario del primero estos miden el error no explicado entre dos conjuntos de datos, para nuestro caso son los valores originales respecto a los estimados.

1.6 Resultados del proceso de análisis

Los resultados del proceso del sistema de recomendación, se utilizaron distintas fuentes de evaluación, tanto como el descriptivo del sitio, un análisis de sentimientos de los comentarios correspondientes a cada sitio, su variabilidad a lo largo de tiempo y los adjetivos que más describen el sitio para los usuarios. El algoritmo que se implementó es *TF-IDF* basado en el *similitud coseno*, al imputar 3 valores distintos: id del sitio anterior en el que estuviste, el número de recomendaciones y la ciudad de destino.

Entretanto, el modelo de referencia para los precios que se deberían cobrar por parte de los oferentes, a las cuales se les aplicó diversos tratamientos y cálculo de estadísticos. Con ello, se obtuvo un conjunto de datos con variables potenciales que permitiesen estimar el precio en base a características conocidas. Los algoritmos que se han probado, han sido diversos tales como: *Decision Tree Regressor*, *Random Forest Regressor*, *Gradient Boosting* y *LightGBM*, siendo este último con el cual se obtuvo mejores resultados.

2 Descripción del conjunto de datos

El conjunto de datos que utilizamos es proporcionado por el el sitio web de la API de Airbnb, www.insideairbnb.com. De muchas ciudades que figuran en el conjunto de datos de Airbnb, seleccionamos los datos de las siguientes ciudades para España.

Ciudades: Barcelona, Euskadi, Girona, Madrid, Málaga, Mallorca, Menorca, Sevilla y Valencia.

Descargamos el formato *csv.gz*, si bien existe otro que está en formato csv, este último no tiene la variedad de variables como el primero, por ello resulta idóneo usarlo.

La cantidad de registros del conjunto de datos de las 9 ciudades a analizar es: 109137, para la tabla *listings*, el cual almacena información desde los inicios de este formato, que va desde el 2011, vale mencionar que esta tabla tiene información a modo de portafolio total, es decir tiene *host*[4]que fueron en algún momento, pero ya no y los que continúan siendo. Este datataset cuenta con 106 columnas de las cuales consideramos para nuestro análisis y creación del modelo, las más intuitivas y que aporten más para nuestro interés, luego de procesos de selección y e ingeniería de características que se realiza.

Además, se tiene dos bases de datos adicionales, las cuales son: *reviews* (con 5 variables inicialmente) con un total de 3.088.002 de comentarios y *calendars*, de éstas tomamos la de *reviews* y también la de *listings* para hacer el sistema de recomendación mediante el uso de *NLP*, el cual se detallará más adelante. Mientras que para el estimador de precios se utilizó la mencionada en el párrafo anterior.

La diversidad de campos que hay entre ambos conjunto de datos van desde variables con textos de descripción y comentarios de los usuarios, así como también del vecindario. También se encuentra las puntuaciones del lugar tanto en diversos puntos de análisis, como servicio, limpieza, comunicación,

[4]**Así se le denomina a las personas que ponen en alquiler hospedaje.**

trato y entre otros. Adicionalmente, nos encontramos con variables del tipo de especificación del espacio en alquiler propiamente, como el número de habitaciones, camas, baños, tipo de habitación, propiedad, características si es que hay o no WiFi, TV, Radio, acceso a cocina, etc.

Table 1: Matriz de Pagos

País	Ciudad
España	Madrid
España	Sevilla
Francia	París

3 Descripción del procesamiento de datos

Vamos a detallar cada punto en las siguientes líneas ergo, en general se utilizó un flujo de trabajo como sigue:

Extracción $\Rightarrow$ *Transformación* $\Rightarrow$ *AED* $\Rightarrow$ *Ing. de Características* $\Rightarrow$ *Modelado*

3.1 Extracción

Este suele ser el primer punto en todo proceso de manejo de datos, pues resulta de la necesidad de una fuente de datos, dependiendo el origen, muchas veces este procedimiento se puede realizar por la misma persona, en otros casos la información llega ya pre-elaborada. Para nuestro caso, básicamente es acceder a la página web www.insideairbnb.com y descargar las bases (listings y reviews) en el formato csv.gz, que es más completo, y partiendo de ello se lee y se convierte en un *DataFrame* de **Python**.

```
####################################
#----- Carga de datos Listings -----#
####################################


ciudades = ['Barcelona', 'Euskadi', 'Girona', 'Madrid', 'Malaga', 'Mallorca', 'Menorca',
            'Sevilla', 'Valencia']
tmp = []
for i in range(len(ciudades)):
    a = 'listings_'
    b = ciudades[i]
    c = '.csv.gz'
    d = a+b+c
    res = pd.read_csv("DB_listings/"+d,compression='gzip', error_bad_lines=False, parse_dates=["last_review","first_review"],
                      dtype = {"id":"str", "host_id":"str", "name":"str", "description":"str",
                                                                          "space":"str", "summary": "str","neighborhood_overv
    res['city']= b
    tmp.append(res)

listings_complete = pd.DataFrame([])
for i in range(len(ciudades)):
    listings_complete = listings_complete.append(tmp[i])
```

```
###################################
#-----  Carga de datos Reviews -----#
###################################

tmp = []
for i in range(len(ciudades)):
    a = 'reviews_'
    b = ciudades[i]
    c = '.csv.gz'
    d = a+b+c
    res = pd.read_csv(d,compression='gzip', error_bad_lines=False, parse_dates=["date"])
    tmp.append(res)

reviews_complete = pd.DataFrame([])
for i in range(len(ciudades)):
    reviews_complete = reviews_complete.append(tmp[i])
```

3.2 Transformación

También conocida como **Wrangling** es el proceso de transformar y mapear datos de un formulario de datos "en bruto" a otro formato con la intención de hacerlo más apropiado y valioso para una variedad de propósitos posteriores, como el análisis. Kazil and Jarmul (2016) .

Después de cargada la información a un Dataframe, lo siguiente es realizar una descripción de tipología de datos y la coherencia de éstos, se identificó que muchas variables necesitan una reestructuración de su tipado, variables como precio, depósito de seguridad, tarifa de limpieza, personas adicionales, tasa de respuesta y aceptación, estaban en formato **string** y requerían estar en **int or float**. Asimismo se hizo limpieza de variables que tenían bien el tipado pero tenían caracteres extraños como es el caso de amenities, que dentro de cada observación había una lista insertada, lo cual dificultada su análisis, casos como ese también se dieron con verificación y tiempo de respuesta de parte del **host**.
En este punto también se hizo un filtrado de información desde el año 2018 en adelante, para considerar información mucho más fresca y que en términos de relevancia para fines prácticos tenga mayor impacto y también se filtra los valores los campos que van a ser necesarios para *reviews*.

```
# Cambiar el X's a float.

listings_complete[['price','security_deposit','cleaning_fee','extra_people',
                   'host_response_rate','host_acceptance_rate']] = listings_complete[['price','security_deposit',
                                                                                      'cleaning_fee','extra_people',
listings_complete['host_is_superhost']=listings_complete['host_is_superhost'].apply(lambda x: 1 if x=='t' else 0).astype(int)

#Transformar en string

listings_complete["amenities"]=listings_complete["amenities"].replace('[\\"\\{\\}]',"",regex=True)
listings_complete["host_verifications"]=listings_complete["host_verifications"].replace("[\\'\\[\\]]","",regex=True)
listings_complete[['name','description']] = listings_complete[['name','description']].astype(str)

#Extraer el año: Haciendo más amigable la Lectura del campo de la fecha de última estancia

listings_complete['year'] = listings_complete['last_review'].apply(lambda x: x.year)
listings_complete['year_month'] = listings_complete['last_review'].dt.to_period('M')

# Reasignando etiquetas

listings_complete['cancellation_policy']=listings_complete['cancellation_policy'].apply(lambda x : "strict_14_with_grace_period"
listings_complete['cancellation_policy']=listings_complete['cancellation_policy'].apply(lambda x : "super_strict_30_60" if (x==

listings_complete['host_response_time']=listings_complete['host_response_time'].replace( "[\\s]","_",regex=True)
```

3.3 Análisis Exploratorio de Datos (AED)

Este apartado también forma parte del pre-procesamiento de información enfocado al análisis estadístico propiamente, tanto a nivel descriptivo como relacional.
El AED formalmente se define como el tratamiento estadístico al que se someten las muestras recogidas durante un proceso de investigación en cualquier campo científico. Para mayor rapidez y precisión, todo el proceso suele realizarse por medios informáticos, con aplicaciones específicas para el tratamiento estadístico. Tukey (1977).

1. Vista General: Aquí se plantea identificar la proporción de valores vacíos por variables, para luego determinar qué hacer con ello, en el apartado de *ingeniería de características*, la tabla imagen que sigue muestra las variables con mayor presencia de nulos.

   La imagen muestra que el campo *neighborhood_overview* tiene mayor presencia de valores vacíos con cerca de un tercio del total, siendo del tipo *object*, esta variable está enfocada a dar una descripción acerca del vecindario de donde se encuentra el predio, le sigue *security_deposit* que es el monto que requiere el *host* como garante de pago antes de efectuarse la instalación en el lugar, tiene cerca del 22% de ausencia de valores. En total, 17 de las variables preprocesadas tienen valores vacíos.

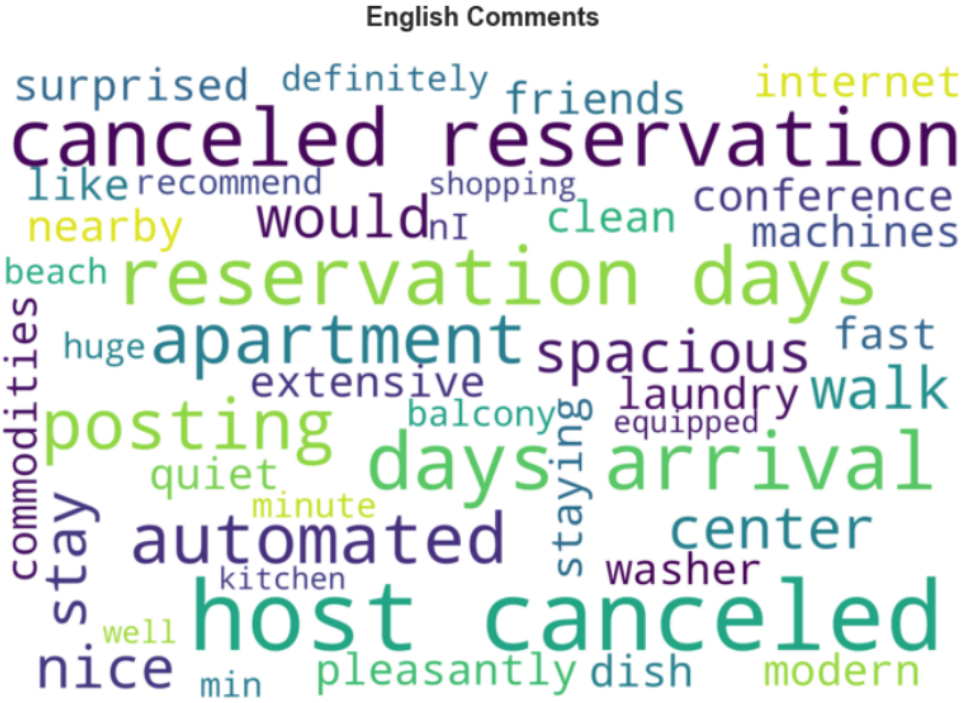
	variable	type_of_var	missing_percent
0	neighborhood_overview	object	32.47
1	security_deposit	float64	21.58
2	space	object	21.38
3	cleaning_fee	float64	16.11
4	host_response_time	object	9.61
5	host_response_rate	float64	9.61
6	summary	object	3.29
7	host_acceptance_rate	float64	2.37
8	review_scores_accuracy	float64	1.36
9	review_scores_value	float64	1.35
10	review_scores_cleanliness	float64	1.35
11	review_scores_location	float64	1.35
12	review_scores_communication	float64	1.35
13	review_scores_checkin	float64	1.35
14	review_scores_rating	float64	1.34
15	beds	float64	0.27
16	bedrooms	float64	0.03
17	bathrooms	float64	0.01

2. Variables de Texto: El análisis de las variables de texto para cada base. Para la base *reviews*, evaluamos operaciones de detección de idioma de los comentarios, creamos un campo nuevo con los comentarios transformados a minúsculas. Ya con estas operaciones efectuadas realizamos una evaluación descriptiva según cada idioma y podemos ver:

Idioma	%
Inglés	52%
Español	25%
Francés	9.8%

Realizamos un análisis de nube de palabras para identificar las palabras

más repetidas por cada idioma.

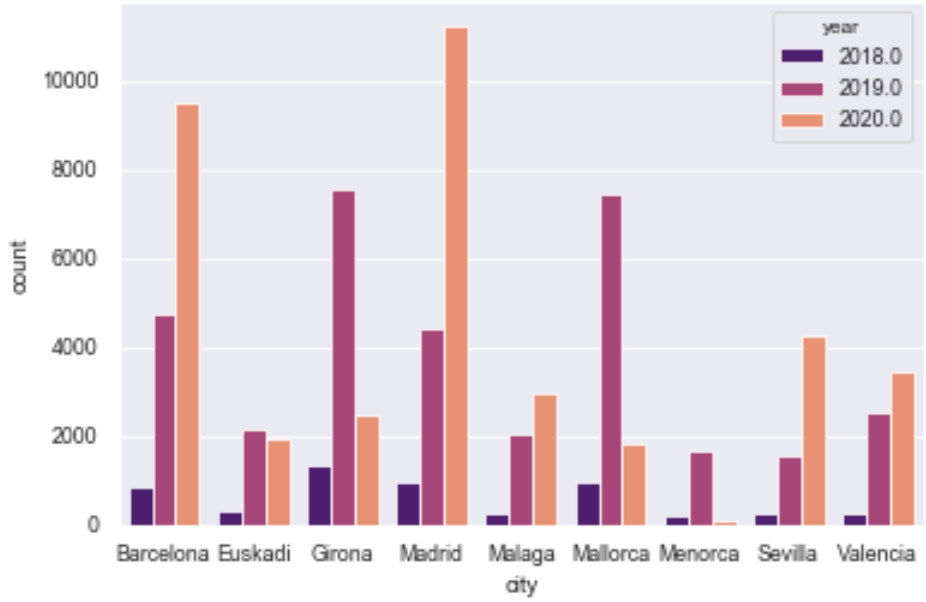


Las palabras más repetidas en inglés son *canceled, reservation, host, arrival, days, entre otras.*

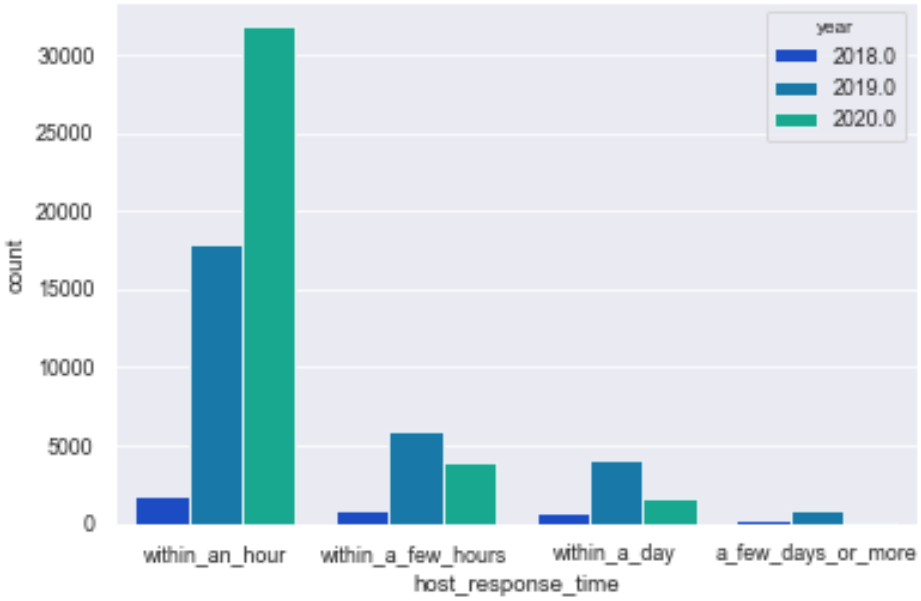
Las palabras más repetidas en español son *duda, apartamento, necesario, Barcelona,limpio, entre otras.*

Las palabras más repetidas en francés son *très, bien, apartement, bien, Merci, plus, Pedro.*

3. Variables Categóricas: Dentro del set de variables categóricas hay algunas que nos resultaron fuertemente relevantes e intuitivas considerarlas para el modelado final, por cual se consideran dentro del análisis en este punto. Las variables en mención son las que siguen: *ciudad*,*tiempo de respuesta* (por agrupación), *política de cancelación* y *tipo de habitación.* Todas están completadas en su totalidad excepto, tiempo de respuesta, para este podría imputarse sin ningún problema con el valor modal, ergo podría acarrear que se sobredimensionara la presencia de una de las categorías existencias generando perturbación innecesaria, para ello un opción razonable es utilizar el valor mediano.



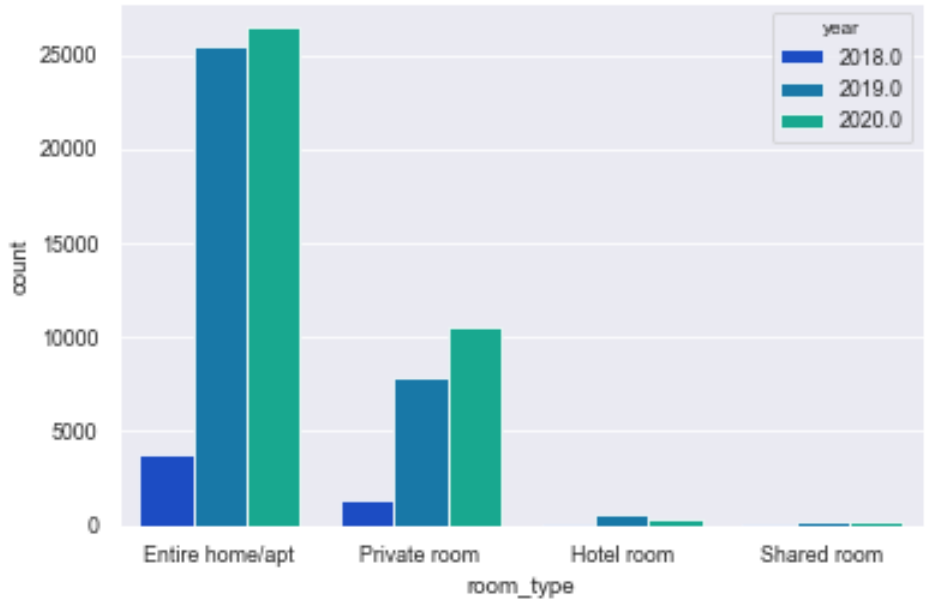
Las 4 principales ciudades son en orden de importancia: Madrid, Barcelona, Girona y Mallorca; entre ellas suman cerca del 70% del total de 76.821 usuarios-host de Airbnb en España. La que menos presencia tiene es Menorca con 2.5%. Adicionalmente como un dato relevante es mencionar que, si bien Madrid tiene la mayor participación los precios más altos suelen encontrarse tanto en Barcelona como Mallorca, entiéndase por su cercanía la zona costera-playa



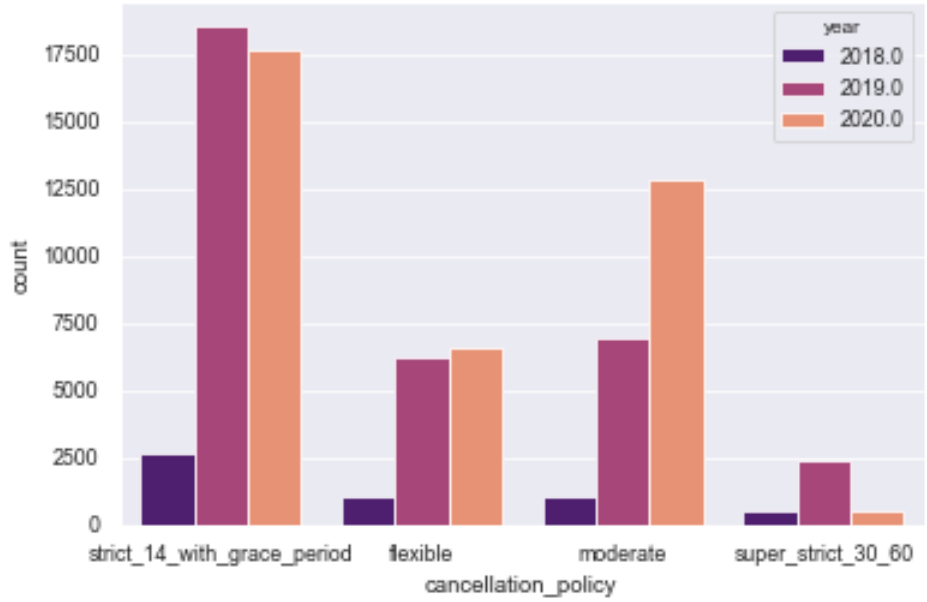
Con relación a la variable tiempo de respuesta que demora el *host* en ponerse en contacto con el potencial usuario, la proporcionalidad es como se ve en el gráfico previo, donde 2/3 del total lo hacen dentro de la primera hora de reacción, lo cual da un indicio que en efecto las personas que se dedican a esto tienden a darle la prioridad debida. Dicha hipótesis se valida, observando que los que demoran más de un día en responder llega a tan solo 1.3%. Vale decir si vemos la figura a como se presenta, abierta por años, la presencia de los que suelen dar respuesta ha ido en incremento paulatino en comparación de merma de los otros 3 casos, esto nuevamente nos da pie a precisar que este modelo de negocio funciona, y para que rentabilice beneficios, como cualquier otro negocio, necesitar dedicar tiempo, y es por ello que las personas tienden a atender pronto para que el usuario potencial no se desanime y busque otro lugar.

Con respecto a las variable tipo de habitación, salta a la vista claramente la presencia sustancial de casas y departamentos completos en alquiler, siendo más del 70% del total, es decir más de 50 mil sitios. El 25% se lleva las habitaciones personales y poco o nada las compartidas y habitaciones de hotel, sumando menos del 2% entre ambos. Vale decir que la presencia hotelera se ha visto en el último periodo, notándose la adaptación de este rubro hacia es este formato de alquiler rentable

y fácil de acceder.



Finalmente, con relación a las políticas de cancelación el grueso se encuentra concentrado en la posobilidad de cancelar a los más 14 días antes de efectuarse la estancia con el 50% del total, esto podría estar alineado con el depósito de seguridad que suelen pedir algunos recintos, en el otro extremo están los súper estrictos que solo permiten cancelar con al menos 30 días de anticipación, y es el 4.3% esto implicaría, de querer cancelar que puedan recibir una penalidad, dependiendo de las condiciones que hayan puesto los host.



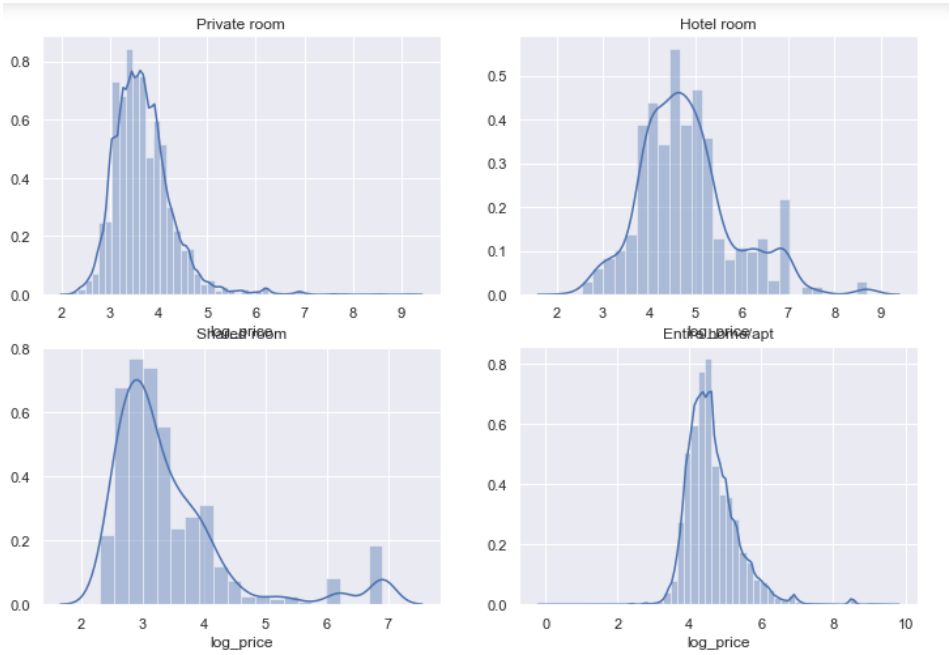
4. Variables Continuas: Dentro de la diversidad de variables continuas destacan, el **precio** que es un insumo para el modelo de precios referencia para los futuros *host*, además se tiene *acommodates*, *beds*, *bedrooms*, *bathrooms*, *security_deposit*, *cleaning_fee* y *host_listings_count*

	variable	count	mean	std	min	0.5%	1%	5%	10%	50%	90%	95%	99%	99.5%	max
0	price	76821.0	126.378503	336.204372	0.0	13.0	15.0	22.0	30.0	75.0	206.0	310.0	999.0	1169.000	15000.0
1	security_deposit	60246.0	174.567556	265.479882	0.0	0.0	0.0	0.0	0.0	150.0	400.0	500.0	1000.0	1500.000	4533.0
2	bedrooms	76795.0	1.873338	1.250491	0.0	0.0	0.0	1.0	1.0	2.0	3.0	4.0	6.0	7.000	32.0
3	beds	76617.0	2.885809	2.236640	0.0	0.0	0.0	1.0	1.0	2.0	6.0	7.0	10.0	12.000	50.0
4	bathrooms	76810.0	1.491219	0.881131	0.0	1.0	1.0	1.0	1.0	1.0	2.0	3.0	5.0	6.000	32.0
5	cleaning_fee	64444.0	39.058888	44.996948	0.0	0.0	0.0	0.0	0.0	30.0	90.0	120.0	200.0	257.355	980.0
6	host_response_rate	69439.0	95.504371	13.180563	0.0	0.0	25.0	75.0	87.0	100.0	100.0	100.0	100.0	100.000	100.0
7	host_acceptance_rate	75004.0	91.488961	17.144102	0.0	0.0	13.0	53.0	71.0	99.0	100.0	100.0	100.0	100.000	100.0
8	accommodates	76821.0	4.255165	2.521659	1.0	1.0	1.0	1.0	2.0	4.0	8.0	8.0	13.0	16.000	40.0
9	review_scores_cleanliness	75781.0	9.333606	1.014458	2.0	4.0	6.0	8.0	8.0	10.0	10.0	10.0	10.0	10.000	10.0
10	review_scores_rating	75791.0	92.187476	9.393773	20.0	40.0	60.0	78.0	80.0	95.0	100.0	100.0	100.0	100.000	100.0

De esta tabla podemos sacar algunas primeras aproximaciones relevantes, en primer lugar encontramos que la diferencia entre la mediana y el promedio es amplia, lo cual nos da indicio de presencia de una cola pesada a la derecha, evidenciado por valores extremos con valores muy altos, véase que el valor $\mathbf{max = 15000}$ euros, y la diferencia con respecto al al percentil 99 y 99.5 respectivamente es de 10 veces

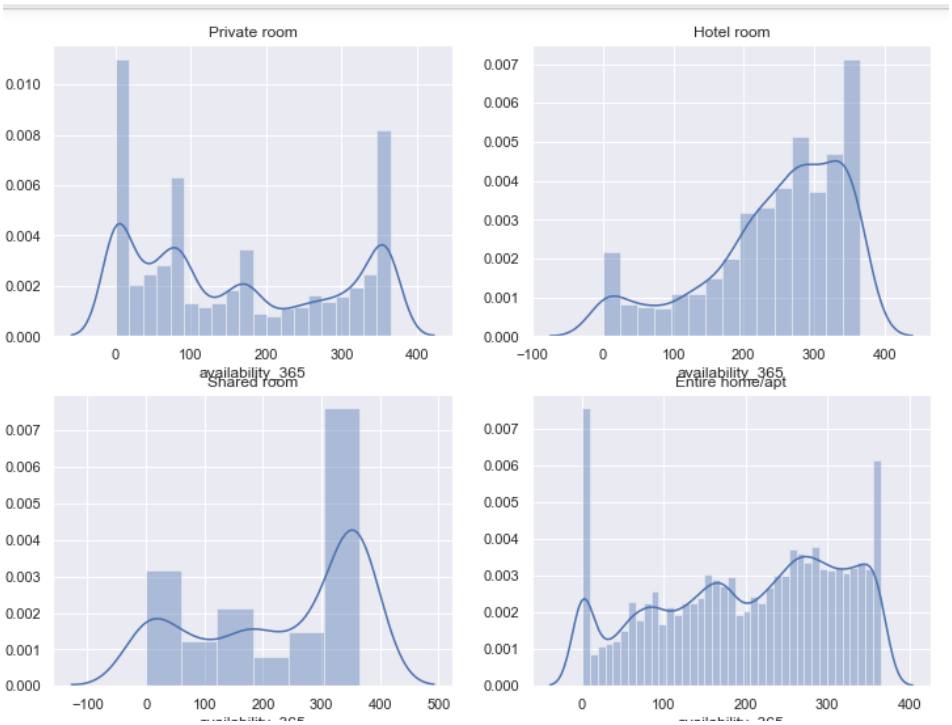
a más. Por el otro extremo encontramos un valor **min = 0**, lo cual no es lógico, asumimos que es un error de carga de la información, ello implicaría por lo tanto realizar un tratamiento en ambos frentes, es decir valores atípicos tanto a la derecha como izquierda de la distribución.

Entretanto como relación a las otras variables mostradas no hay algo que significativamente nos pueda llamar la atención, podría decirse que **beds = 0** podría resultar atípicos, sin embargo resulta hasta cierto punto razonable, pues dentro de las políticas de alquiler, hay lugares que alquilan el espacio y no necesariamente una cama, a veces es un sofá un sofá-cama, además la proporción que tienen esas características no resulta muy material (menos del 5%). En cualquier caso, los valores atípicos serán tratan bajo una misma lógica en las variables cuantitativas.

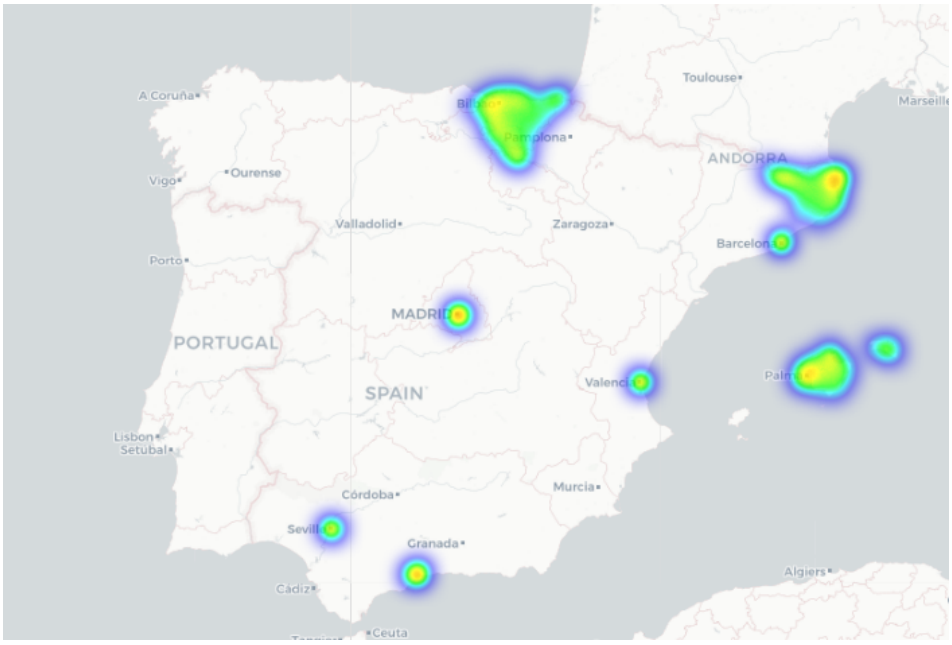


Los histogramas muestran la distribución de precios por tipos de habitación, para una mejor visualización se hizo con un precio en términos logarítmicos, y aún se así se puede apreciar valores atípicos presentes a la derecha de la distribución que son los que finalmente influyen en la diferencia entre media y mediana.

nuevamente se muestra un gráfico de la distribución de la disponibilidad durante el año por tipo de habitación, la imagen nos aproxima que las habitaciones de hotel suelen estar disponibles en mayor cuantía disponibles todo el año, mostrándose una asimetría negativa, lo cual ocurre cuando la media está por debajo del valor mediano, otra imagen interesante de mencionar es lo uniforme que se distribuye la disponibilidad de casas y departamentos completos (salvo por unos picos a los extremos)

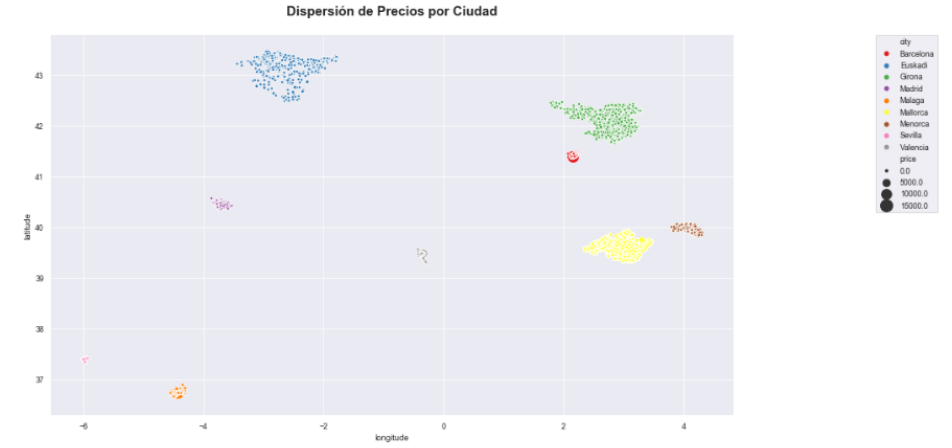


5. Variables Geográficas: Dentro de este grupo, encajan perfectamente las variables longitud y latitud, que a pesar de un primer vistazo no podrían ser útiles, cuando se analiza con relación al precio, que es nuestro interés estimar, resulta que si tiene un efecto determinante debido a que las coordenadas diferencian ciudades y en cada ciudad los precios son diferenciados por el hecho de que en esas ciudades tienen características específicas que lo permite, y es ahí donde estas variables de georeferencia, nos ayudan.



Este primer mapa, me ayuda a visualizar que de concentradas se encuentran las ciudades con relación a los lugares dónde se alquilan habitaciones mediante este mecanismo, Madrid salta a la vista por su alta concentración en el centro de la ciudad.

En este siguiente mapa se referencia la idea detrás del primer párrafo de este punto, y además se aprecia la dispersión que existe de precios dentro de cada ciudad, puede visualizarse por ejemplo que en Barcelona, los precios suelen ser homogéneos salvo por algunos *outliers* que cobran precios exorbitantes.



3.4 Ingeniería de Características

1. El precio, la principal variable del algoritmo que busca estimar un tablero de precios para los nuevos host de Airbnb, tenía una distribución asimétrica, lo cual impide modelar adecuadamente, ergo, esto no quiere decir que no es bueno tener presencia de valores extremos, es más estos podrían aportar porque explican el comportamiento de un pequeño nicho, sin embargo, buscamos mitigar de cierta manera lo extremo que puedan llegar a ser, por ello reducimos el dataset el percentil 99.5 y 0.05, que nos permita despejar la colas muy extremas por tipo de habitación. Con esto logramos desechar el valor cero que encontramos como mínimo erróneo y el de 15 mil euros, en el otro extremo. El código que nos permite realizar esto es el que sigue:

```
listings_price_filt=listings_price[listings_price.groupby("room_type")["price"].\
                                   transform(lambda x : (x<=x.quantile(0.995))&(x>=(x.quantile(0.005)))).eq(1)]
```

```
print(len(listings_price))
print(len(listings_price_filt))
print("Porcentaje de exclusión: "+str(round((1-(len(listings_price_filt)/len(listings_price)))*100,3))+"%")
```

```
76821
76104
Porcentaje de exclusión: 0.933%
```

el porcentaje que se pierde por esta operación es minúsculo, 0.93% del total, quedando finalmente con 76.104 observaciones con las cuales realizar los modelos propuestos inicialmente.

2. Reestructura de Categoría: En este lo que se planteó fue convertir en numérico discreto los valores de las etiquetas de algunas variables, como política de cancelación, tipo de habitación, ciudad y tiempo de respuesta. Esta operación fue posible con el siguiente código:

```
room_type_mapping={'Hotel room':4,'Entire home/apt':3,'Private room':2,'Shared room':1}
city_mapping={'Mallorca':9,'Euskadi':8,'Menorca':7,'Barcelona':6,'Madrid':5,'Girona':4,'Sevilla':3,'Malaga':2,'Valencia':1}
canc_policy_mapping={'super_strict_30_60':4,'strict_14_with_grace_period':3,'moderate':2,'flexible':1}
response_time={'within_a_few_hours':4,'within_a_day':3,'within_an_hour':2,'a_few_days_or_more':1}
listings_price_filt['room_type'] = listings_price_filt['room_type'].map(room_type_mapping)
listings_price_filt['city'] = listings_price_filt['city'].map(city_mapping)
listings_price_filt['cancellation_policy'] = listings_price_filt['cancellation_policy'].map(canc_policy_mapping) # convertir en
listings_price_filt['host_response_time']=listings_price_filt['host_response_time'].map(response_time)
```

3. Imputación de Valores Vacíos: Los missing values se abordaron desde diferentes frentes dependiendo el tipo de variable a imputar, por ejemplo en el caso de variables discretas como número de camas, baños o habitaciones se optó por imputar con el valor modal, en el caso cantidad de usuarios que han sido atendidos host_listings_count_rating_sentimiento, esa variable tenía 5 valores NaN y generaba problemas al momento de querer imputar por lo cual se decidió imputarle un valor cero.

```
room_type_mapping={'Hotel room':4,'Entire home/apt':3,'Private room':2,'Shared room':1}
city_mapping={'Mallorca':9,'Euskadi':8,'Menorca':7,'Barcelona':6,'Madrid':5,'Girona':4,'Sevilla':3,'Malaga':2,'Valencia':1}
canc_policy_mapping={'super_strict_30_60':4,'strict_14_with_grace_period':3,'moderate':2,'flexible':1}
response_time={'within_a_few_hours':4,'within_a_day':3,'within_an_hour':2,'a_few_days_or_more':1}
listings_price_filt['room_type'] = listings_price_filt['room_type'].map(room_type_mapping)
listings_price_filt['city'] = listings_price_filt['city'].map(city_mapping)
listings_price_filt['cancellation_policy'] = listings_price_filt['cancellation_policy'].map(canc_policy_mapping) # convertir en
listings_price_filt['host_response_time']=listings_price_filt['host_response_time'].map(response_time)
```

```
from sklearn.impute import SimpleImputer

imr_mode = SimpleImputer(missing_values=np.nan, strategy='constant')

imr = imr_mode.fit(listings_recomen[['review_scores_rating','sentimiento']])
listings_recomen[['review_scores_rating','sentimiento']] = imr.transform(listings_recomen[['review_scores_rating','senti
```

Por otra parte variables continuas como depósito de seguridad se imputó con el valor mediano, dado que este estadístico no se ve contaminado por la presencia de valores atípicos y nos resulto razonable hacerlo, en caso existan valores extremos.

4. Tratamiento de Valores extremos: Tal como se planteó con el precio, se trabajó con los percentiles 99.5 y 0.05, a los valores superiores o inferiores a esos límites se les imputó el valor correspondiente a cada percentil, si menos menores al 0.005% entonces se le asigna el valor de dicho percentil y en el otro caso el del 99.5 si es que superior a éste.

```
listings_x=listings_x.clip(listings_x.quantile(0.005), listings_x.quantile(0.995), axis=1)
```

```
listings_price_filt=listings_price_filt.drop(columns=["log_price"])
listings_price_filt["log_price"]=listings_price_filt["price"].apply(lambda x: round(np.log(x),2))
```

```
list_log_price=listings_price_filt["log_price"]
list_price=listings_price_filt["price"]
```

5. Para *listings*, se crearon nuevas columnas donde se transformaban a: *name*, *description* y *space*. Estas transformaciones implican eliminar dígitos, signos de puntuación, caracteres especiales y stopwords[5], con el fin de preparar los datos para el sistema de recomendación.

```
stopword = stopwords.words('english')

def digitos(x):
    try:
        return ''.join(c for c in x if not c.isdigit())
    except:
        return x

def puntuacion(x):
    try:
        return ''.join(c for c in x if c not in punctuation)
    except:
        return x

def stop(x):
    try:
        word_tokens = nltk.word_tokenize(x)
        return ' '.join(word for word in word_tokens if word not in stopword)
    except:
        return x

def especiales(x):
    try:
        for k in x.split("\n"):
            y = re.sub(r"[^a-zA-Z0-9]+", ' ', k)
        return y
    except:
        return x

listings_recon['name_low'] = listings_recon.name_low.apply(lambda x: digitos(x))
listings_recon['name_low'] = listings_recon.name_low.apply(lambda x: puntuacion(x))
listings_recon['name_low'] = listings_recon.name_low.apply(lambda x: stop(x))
listings_recon['name_low'] = listings_recon.name_low.apply(lambda x: especiales(x))

listings_recon['description_low'] = listings_recon.description_low.apply(lambda x: digitos(x))
listings_recon['description_low'] = listings_recon.description_low.apply(lambda x: puntuacion(x))
listings_recon['description_low'] = listings_recon.description_low.apply(lambda x: stop(x))
listings_recon['description_low'] = listings_recon.description_low.apply(lambda x: especiales(x))

listings_recon['space_low'] = listings_recon.space_low.apply(lambda x: digitos(x))
listings_recon['space_low'] = listings_recon.space_low.apply(lambda x: puntuacion(x))
listings_recon['space_low'] = listings_recon.space_low.apply(lambda x: stop(x))
listings_recon['space_low'] = listings_recon.space_low.apply(lambda x: especiales(x))
```

Por otro lado creamos una columna más para el procesamiento del algoritmo de recomendación, donde une los textos de nombre y descripción y de esta forma se analiza con la ayuda de ngramas[6].

```
listings_recomen['content'] = listings_recomen[['name_low','description_low']].astype(str).apply(lambda x: ' // '.join(
```

[5] **stopwords:** Lista de artículos, pronombres, preposiciones, adverbios e incluso algunos verbos, para cada idioma que tienen poca relevancia

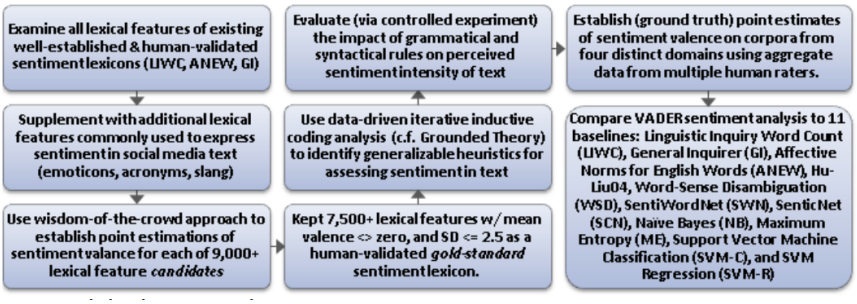
[6] **ngramas:** Es una subsecuencia de n elementos de un texto o oración.

4 Descripción del modelo

En relación a la descripción de los modelos usados, el análisis dentro de trabajo contiene 2 distintos análisis:

1. Modelo de recomendación, forma parte de varios análisis para pode procesar toda la información:

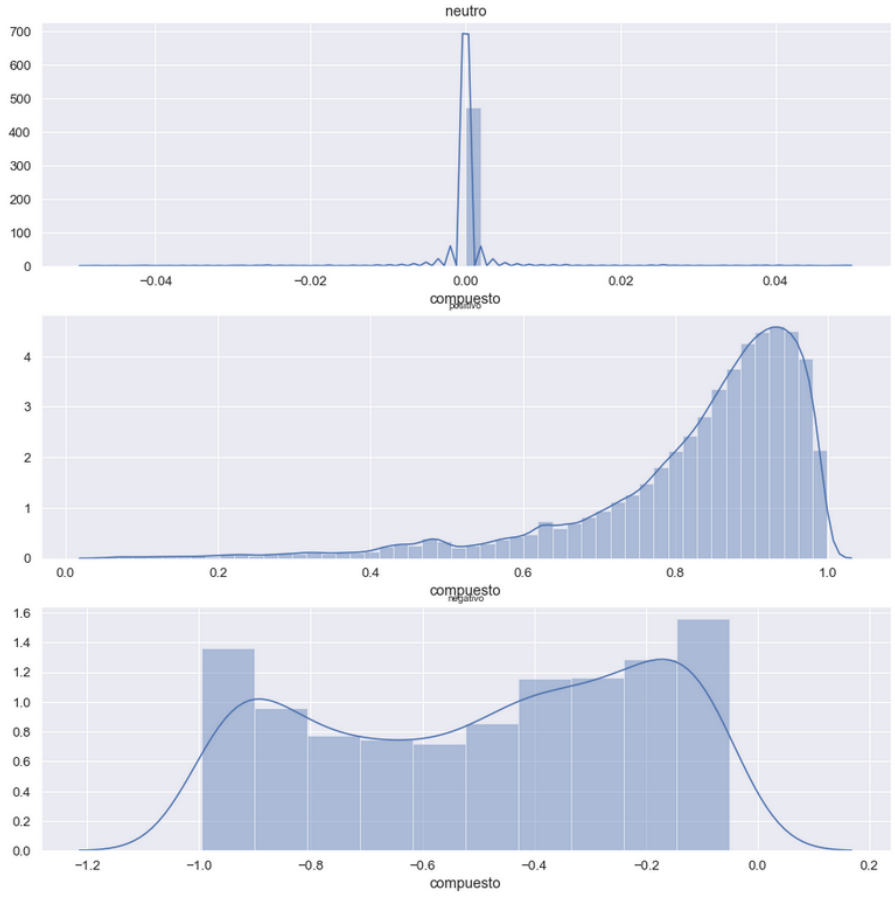
   - **Análisis de Sentimiento:** Existen muchas librerías que ayudan a entender de forma analítica cómo un usuario piensa o siente con respecto a un tema específico. Para nuestro trabajo realizamos un proceso de análisis de sentimiento con la ayuda de una librería llamada ***VADER***. Desarrollada con la ayuda del MIT Hutto and Gilbert (2015). Esta librería métodos para datos cualitativos y cuantitativos y evaluar con la ayuda de un procesador de palabras el sentimiento enfocado para textos con un contexto de microblog-like. Utiliza 5 reglas de análisis para analizar e intensificar el sentimiento y es ideal para textos que contienen emoticones. Evaluaron ***VADER*** *Precisión* = 0.96 y 0.84.



   De esta forma utilizamos su función para identificar los sentimientos. Lo interesante de esta función es los 4 valores resultantes, positivo, negativo, neutro y compuesto:

	listing_id	date	comments	language_description	comments_lower	negativo	neutro	positivo	compuesto	year	adjetivos
0	23197	2011-03-15	We (5 friends) were staying for a conference i...	en	we (5 friends) were staying for a conference i...	0.000	0.625	0.375	0.9617	2011.0	[spacious, nice, quiet, nearby, extensive, fas...
1	23197	2012-01-11	The host canceled my reservation 138 days befo...	en	the host canceled my reservation 138 days befo...	0.000	1.000	0.000	0.0000	2012.0	[]
2	23197	2015-07-27	We had a nice stay at the apartment. It is a 1...	en	we had a nice stay at the apartment. it is a 1...	0.000	0.844	0.156	0.7351	2015.0	[spacious, huge, nice]
3	23197	2015-09-28	Great apartment! It was plenty spacious for us...	en	great apartment! it was plenty spacious for us...	0.045	0.713	0.242	0.9656	2015.0	[easy, oriented, spacious, great, helpful, short]
4	23197	2015-10-11	I stayed in Marnie's apartment with two collea...	en	i stayed in marnie's apartment with two collea...	0.000	0.854	0.146	0.9072	2015.0	[spanish, pleased, lovely, clean, convenient, ...

Parámetro	**Valor**
Positivo	$compoundscore >= 0.05$
Negativo	$compoundscore <= -0.05$
Neutro	$(compoundscore > -0.05)$ y $(compoundscore < 0.05)$



Este gráfico de distribución de los sentimientos en los comentarios (reviews), la imagen nos delimita el comportamiento de los del sentimiento donde se ve claramente su agrupación. Para el sentimiento positivo es claro como su distribución es sesgada a la derecha, lo que nos indica que cuando una persona comenta a un sitio, los comentarios son muy positivos. Por otro lado, para el sentimiento negativo se ve que los bins de la distribución abarcan más datos y su rango de variación es pequeño y por último, el sentimiento neutro al tener un rango tan pequeño los datos dentro de esta región son pocos.

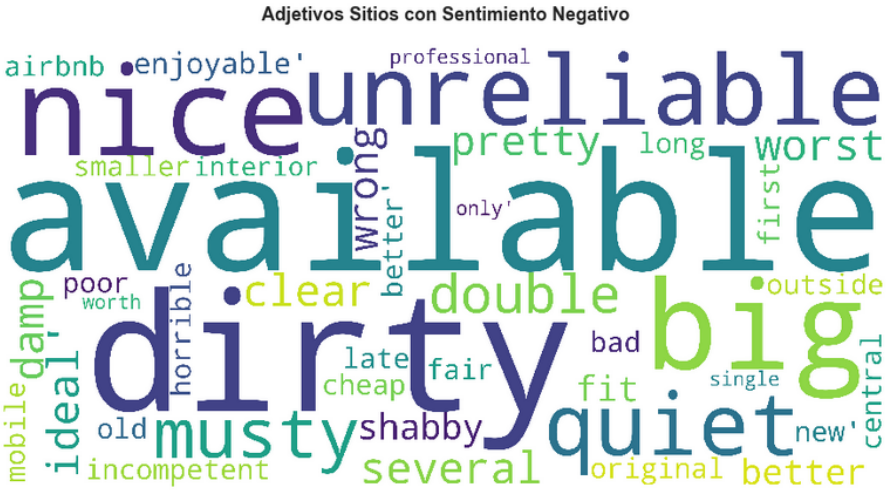
- **Análisis Adjetivos:** El objetivo de este análisis es para verificar si el sentimiento en los comentarios están siendo evaluado bien y también crear una base donde especifica por cada sitio, los adjetivos que más detallan al lugar. La librería en la que nos basamos es ***SPACY***. Esta librería se especializa en análisis de entidades en distintos idiomas y otras funciones más. Es un modelo que predice los *tags* en un texto según su contexto.

	listing_id	year	adjetivos
0	4620	2012.0	[many, great, nearby, pure, beautiful, worth, ...
1	4620	2013.0	[close, old, kind, young, good, possible, grea...
2	4620	2014.0	[spanish, happy, wonderful, lovely, traditiona...
3	4620	2015.0	[]
4	4620	2016.0	[great, kind, clean, historic, nice, prolific,...
5	4620	2017.0	[historic, hot, disappointing, great, most, ki...
6	4620	2018.0	[happy, welcome, perfect, pleased, local, long...
7	4620	2019.0	[unclean, nice, desperate, same, little, overw...
8	6369	2010.0	[flat, ultimate, appointed, clean, fantastic, ...
9	6369	2011.0	[largest, gorgeous, helpful, clean, comfortabl...
10	6369	2016.0	[normal, possible, lovely, kind, welcoming, ot...
11	6369	2019.0	[easy, big, light, nice, little, excellent, di...
12	6369	2020.0	[highest, exciting, lovely, unwell, kind, okay...
13	11547	2012.0	[private, nice, close, great, small, whole, pe...
14	11547	2013.0	[pleasant, private, serviceminded, quiet, welc...
15	11547	2014.0	[nice, close, lovely, fantastic, good, great, ...
16	11547	2015.0	[spanish, private, shingle, main, prime, authe...
17	11547	2016.0	[greatest, clean, able, perfect, beautiful, pl...
18	11547	2017.0	[relaxing, many, more, lush, cute, easy, minde...
19	11547	2018.0	[unique, full, amazing, cozy, perfect, positiv...

Con la ayuda de este modelo, podemos identificar los adjetivos de los comentarios según su sentimiento.

Las palabras más repetidas con sentimiento positivo son *nice, great, clean, helpful, good, perfect, confortable, entre otras.*

Las palabras más repetidas con sentimiento netro son *nice, cold, little, ok, great, begining, enough, entre otras.*

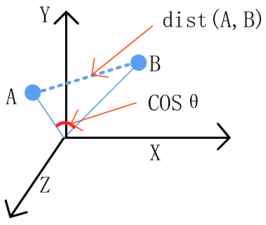


Las palabras más repetidas con sentimiento negativo son *nice, available, dirty, unrealiable, big, wrong, damp, incompetent, entre otras.*

- **TF-IDF y Coseno de Semejanza:** es una medida que expresa que tan relevante es una palabra para un documento, utilizando ngramas.
  $TF : \frac{f(t,d)}{max(f(t,d):\in d)}$

  $IDF : \log \frac{|D|}{|(d \in D : t \in d)|}$

  Por otro lado otra medida que utilizamos es la de coseno de similitud. Calcula la similitud entre los textos, en este caso o los ngramas, que mide el ángulo del coseno entre 2 vectores, que en este caso son los vectores del TF-IDF.



```
tf = TfidfVectorizer(analyzer = 'word', ngram_range = (1, 2), min_df = 0.1, stop_words = 'english')
tfidf_matrix = tf.fit_transform(listings_recomen['content'])
```

```
cosine_similarities = linear_kernel(tfidf_matrix, tfidf_matrix)
cosine_similarities
```

```
array([[1.        , 0.32667986, 0.32480185, ..., 0.01054849, 0.21307069,
        0.02255963],
       [0.32667986, 1.        , 0.15608752, ..., 0.        , 0.2804671 ,
        0.08754449],
       [0.32480185, 0.15608752, 1.        , ..., 0.        , 0.25922326,
        0.02418105],
       ...,
       [0.01054849, 0.        , 0.        , ..., 1.        , 0.        ,
        0.53038485],
       [0.21307069, 0.2804671 , 0.25922326, ..., 0.        , 1.        ,
        0.04706026],
       [0.02255963, 0.08754449, 0.02418105, ..., 0.53038485, 0.04706026,
        1.        ]])
```

```
results = {}
for idx, row in listings_recomen.iterrows():
    similar_indices = cosine_similarities[idx].argsort()[:-100:-1]
    similar_items = [(cosine_similarities[idx][i], listings_recomen['id'][i]) for i in similar_indices]
    results[row['id']] = similar_items[1:]

results
```

```
{'23197': [(0.709592083180209, '36618612'),
  (0.6968469547251619, '16848332'),
  (0.6945101790747081, '5769043'),
  (0.684073955319661, '21776861'),
  (0.6739717423024972, '4785074'),
  (0.6704356785241403, '14840219'),
  (0.667197771590086, '27270164'),
  (0.6635462591586797, '27484956'),
  (0.6634670276262118, '18323993'),
  (0.663396830415367, '2713720'),
  (0.6632908061143552, '25095438'),
  (0.6595292752414614, '35557291'),
  (0.6560941598529396, '49968'),
  (0.6560941598529396, '32711'),
  (0.653916172528757, '33292312'),
  (0.6417270092222412, '4146455'),
  (0.6399998164897166, '32992640'),
  (0.6393967555051833, '24217669'),
  (0.6386731639587815, '31184437'),
```

Lo que logramos como resultado un diccionario en la cual tiene los valores de todos los id como key, y dentro tiene una lista de los id con los que tiene similitud y su correspondiente valor.

- **Sistema de Recomendación:** Dentro del sistema evaluamos el resultado del TF-IDF Y del coseno de similitud, con el fin de recomendar n numero de de lugares, en base a un id que el usuario ha estado, dada una ciudad a la que se dirige.

```
def item(id):
    name   = listings_recomen.loc[listings_recomen['id'] == id]['content'].tolist()[0].split(' // ')[0]
    desc   = ' Descripción: ' + listings_recomen.loc[listings_recomen['id'] == id]['content'].tolist()[0].s
    rat    = ' Rating: ' + listings_recomen.loc[listings_recomen['id'] == id]['review_scores_rating'].astyp
    var    = ' Variación: ' + listings_recomen.loc[listings_recomen['id'] == id]['compuesto_var'].astype(st
    sent   = ' Sentimiento: ' + listings_recomen.loc[listings_recomen['id'] == id]['sentimiento']
    ciud   = 'Ciudad: ' + listings_recomen.loc[listings_recomen['id'] == id]['city']
    prediction = str(name)  + str(desc) + str(ciud) + str(sent)+ str(rat) + str(var)
    return prediction

def recommend(item_id, num, ciudad):
    print('Sitio de inicio ' + str(num) + ' producto similar a ' + item(item_id))
    print('---')
    recs = results[item_id]
    tmp = []
    for rec in recs:
        if (listings_recomen[listings_recomen['id'] == (rec[1])].city.values != listings_recomen[listings_r
            print(" Recommended: " + item(rec[1]) + "  (score:" + str(rec[0]) + ')')
            tmp.append(rec[1])
        if len(tmp)>= num:
            break
```

```
recommend(item_id = '31184437' , num = 4, ciudad = 'Barcelona')
```

```
Sitio de inicio 4 producto similar a entire stfl flat brbalconyterpool view Descripción: licence tasteful
bright fresh apartment villa alexia nd floor apt balcony terrace overlooking swimming pool set complex ap
artments charming puerto pollen a property ...60484    Ciudad: Mallorca
Name: city, dtype: object60484     Sentimiento: positivo
Name: sentimiento, dtype: object60484     Rating: 100.0
Name: review_scores_rating, dtype: object60484     Variación: nan
Name: compuesto_var, dtype: object
---
 Recommended: bcn sagrada familia penthouse Descripción: large bright apartment superb terrace view emble
matic sagrada familia completely renovated early apartment ideally located steps sagrada familia avenue g
audi large b...4651    Ciudad: Barcelona
Name: city, dtype: object4651     Sentimiento: positivo
Name: sentimiento, dtype: object4651     Rating: 85.0
Name: review_scores_rating, dtype: object4651     Variación: 0.2037368421052632
Name: compuesto_var, dtype: object  (score:0.6627231111712387)
 Recommended: forum deluxe mins walk ccib center sea  Descripción: accept groups young people apartment s
uitable ideal families quiet people beautiful apartment large terrace min walk ccib center sea port forum
great location combin...0    Ciudad: Barcelona
Name: city, dtype: object0     Sentimiento: positivo
Name: sentimiento, dtype: object0     Rating: 95.0
Name: review_scores_rating, dtype: object0     Variación: 0.07289615384615378
Name: compuesto_var, dtype: object  (score:0.6386731639587815)
 Recommended: ab luxury palace ref gi Descripción: apartment forms part stately palauet tres torres house
modern styled house built early th century luxury apartment made five apartments joined large terrace con
necte...13127    Ciudad: Barcelona
Name: city, dtype: object13127     Sentimiento: positivo
Name: sentimiento, dtype: object13127     Rating: 90.0
Name: review_scores_rating, dtype: object13127     Variación: nan
Name: compuesto_var, dtype: object  (score:0.6304493736214841)
 Recommended: stylish eixample luxury center big terrace Descripción: elegant quiet apartment private ter
race next paseo de gracia ac rooms staying stylish eixample ensures perfect unforgettable holidays accept
ed youth groups stylish a...3195    Ciudad: Barcelona
Name: city, dtype: object3195     Sentimiento: positivo
Name: sentimiento, dtype: object3195     Rating: 93.0
Name: review_scores_rating, dtype: object3195     Variación: -0.17501666666666682
Name: compuesto_var, dtype: object  (score:0.6274842296374196)
```

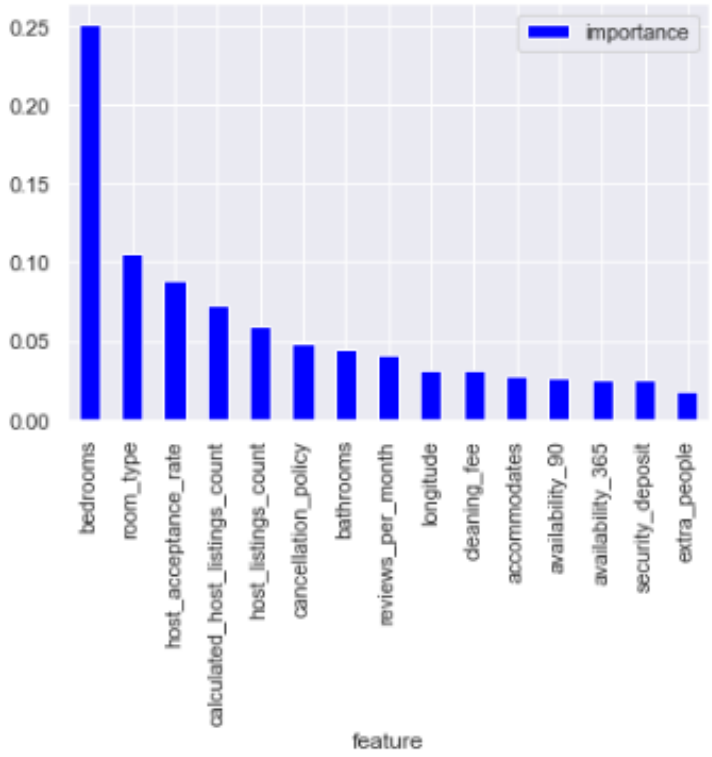
2. **Modelo de Estimación de Precios:** Como se ha venido mencionando la motivación y la propuesta del trabajo en su conjunto es adentrarse en el modelo de negocio que ha surgido a raíz del formato de alquiler propiciado por Airbnb y en el cual es partícipe un sinnúmero de personas, para ese fin, por el lado de la estimación de precios se plantean diversos modelos, primero un grupo alineado con los modelos basados en árboles y que son no paramétricos por definición, y por el otro frente los paramétricos, que en este caso está soportado únicamente por la Regresión Lineal.
   en todos los casos se utilizó las cerca de 40 variables finales, esto como un primer de entrenamiento, siempre que, se podría volver a entrenar el modelo con una menor cantidad de variables considerando el nivel de importancia que éstas tienen en la definición del modelo.

	index	Total	Total_abs
13	accommodates	0.435358	0.435358
15	bedrooms	0.418054	0.418054
16	beds	0.392665	0.392665
4	host_listings_count	0.387434	0.387434
14	bathrooms	0.387434	0.387434
10	cleaning_fee	0.283135	0.283135
8	room_type	0.282780	0.282780
18	guests_included	0.248531	0.248531
9	security_deposit	0.206716	0.206716
33	city	0.158623	0.158623
11	reviews_per_month	-0.151154	0.151154
7	longitude	0.132892	0.132892
22	number_of_reviews_ltm	-0.124447	0.124447
21	number_of_reviews	-0.112474	0.112474
5	calculated_host_listings_count	0.106281	0.106281
35	cancellation_policy	0.104069	0.104069
20	availability_365	0.102387	0.102387
34	year	-0.099375	0.099375
28	availability_90	0.061694	0.061694
27	availability_60	0.060894	0.060894
17	extra_people	0.050247	0.050247
26	availability_30	0.046731	0.046731

Entre las principales variables que se encuentran correladas con respecto al precio son: accommodates con cerca de 43%, que es en esencia el número de personas que pueden entrar en la casa, el número de habitaciones del recinto, con 41.8% y número de camas con alrededor de 39.0%.

Una de las razones por las que una regresión lineal no funciona en esta situación es por la naturaleza que, se ve afectada por la presencia de multicolinealidad, tiende a generar no significancia en los parámetros estimados, por lo cual resultaría lógico si se desea plantear como una solución posible este tipo de modelo, hacer un análisis exhaustivo de correlaciones parciales entre las variables explicativas.

- **Decision Tree Regressor:** Los árboles de decisión en general son un método de aprendizaje supervisado no paramétrico utilizado para la clasificación y la regresión. El objetivo es crear un modelo que prediga el valor de una variable objetivo mediante el aprendizaje de reglas de decisión simples inferidas de las características de los datos.
  Los árboles de regresión aprenden de los datos para aproximar una curva sinusoidal con un conjunto de reglas de decisión si-entonces-otro. Cuanto más profundo es el árbol, más complejas son las reglas de decisión y más se ajusta el modelo, lo cual genera el riesgo de generar sobreajuste y por lo cual poca o nula replicabilidad para otros escenarios fuera el de la unidad de entrenamiento. Se dice que es no parámetrico debido a que de antemano no se determina una forma distribución, como en el caso de la regresión lineal donde se asume ciertos supuestos respecto a los residuos y variables.

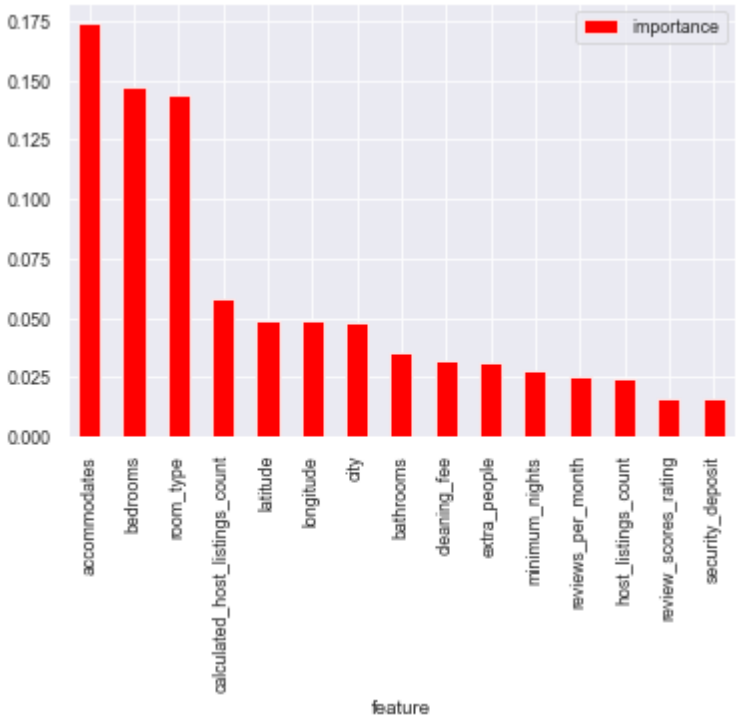


El gráfico de importancia de características refleja que en el caso del árbol de regresión, las variables principales que han aportado en la ramas superiores del árbol, las cuales son: número de camas, tipo de habitación, la tasa de aceptación del host y el número de huéspedes que ha tenido hasta la fecha.

```
print("Precisión:",metrics.r2_score(y_test, y_pred))
print("Precisión:",metrics.mean_absolute_error(y_test, y_pred))
print("Precisión:",np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
print("Precisión:",(metrics.mean_squared_error(y_test, y_pred)))
```

```
Precisión: 0.4171993099418345
Precisión: 42.906431939098766
Precisión: 105.3194307102286
Precisión: 11092.182485126645
```

- **Gradient Boosting Regressor:** Graident Boosting 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. En esencia lo que realiza es optimizar en cada iteración y reentrena en función a ese punto.

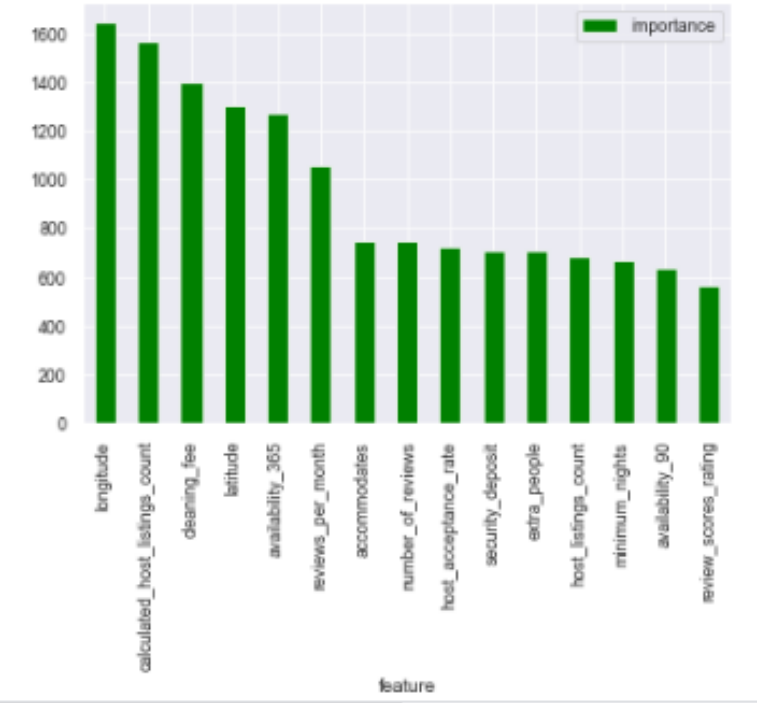


Del mismo modo que en el caso previo, nuevamente las principales variables son: accommodates, número de camas, el tipo de habitación el número de huespedes que ha tenido a la fecha el usuario, la latitud entre otras, la gráfica muestra las 15 primeras mejores.

```
print("Precisión:",metrics.r2_score(y_test, y_pred))
print("Precisión:",metrics.mean_absolute_error(y_test, y_pred))
print("Precisión:",np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
print("Precisión:",metrics.mean_squared_error(y_test, y_pred))
```

```
Precisión: 0.3724825463174831
Precisión: 35.865155732671724
Precisión: 109.28520697766177
Precisión: 11943.256464150374
```

- **Light GBM Regressor:** Es un marco de aumento de gradiente que utiliza algoritmos de aprendizaje basados en árboles. Está diseñado para ser distribuido y eficiente con las siguientes ventajas: mayor velocidad de entrenamiento y mayor eficiencia, así como también menor uso de memoria y mejor precisión. Por lo cual dentro del marco de los modelos de boosting, tiende a ser mejor en algunas ocasiones.



Con el mismo enfoque que el previo las principales variables son: longitude, el número de huéspedes, la tarifa de limpieza, la latitud y la disponibilidad durante el año.

```
print(r2_score(y_test, lgbm.predict(X_test), multioutput='variance_weighted'))
```

```
0.7262810562517121
```

Una primera reflexión que se puede plantear como interesante es que en el caso de los dos primeros modelos, las variables resultan ser muy similares entre sí, en general se debe porque la naturaleza es parecida, parten de árboles, sin embargo lo que también se acentúa es que salvo las 3 primeras variables el resto resultan con un peso mucho menor, sin embargo en el caso del modelo de LightGBM, el peso se ve mejor distribuido y con variables relavantes diferentes a las mencionadas atrás y que un tanto con las correlaciones mostradas previamente, esto quiere decir que el entrenamiento ha generado un cambio en la dirección.
Como segunda conclusión es que, en efecto el modelo de LigthGBM, resulta tener mejores resultados, en términos generales analizado por el coeficiente de determinación.

3. **Modelo de Regresión Lineal:** En estadística la regresión lineal o ajuste lineal es un modelo matemático usado para aproximar la relación de dependencia entre una variable dependiente Y, las variables independientes X's.
   El modelo de regresión al estar condicionado bajo ciertos supuestos paramétricos no performa bien con estos datos, debido a la presencia de correlación alta entre las variables explicativas.[7]. Esto viene alineado con las variables que encontró como más relevantes en cada caso, mientras que el árbol de regresión alcanzó el nivel de 42% de coeficiente de determinación el gradient boosting llegó solo a 37% mientras que el modelo entrenado con LightGBM alcanzó el 73% aproximadamente.

   Las características sobre las cuales fueron entrenados los modelos fueron muy similares, tanto el número de iteraciones (n=1000), el mínimo de muestras en cada nodo para que se genera una nueva partición (n=60), así como el número de ramas mínimas (n=20) necesarias.

   La imagen, permite visualizar una pequeña muestra de los valores reales v.s. los estimados del modelo de LightGBM.

	index	price	price_est
5420	2427	85.0	128.830913
11989	1256	24.0	26.554877
21403	272	70.0	76.119129
6385	11462	1169.0	968.893909
6768	4140	109.0	75.827177
2274	2352	270.0	176.502684
5719	11058	40.0	43.254541
18674	4074	102.0	67.475000
14170	11117	58.0	55.033743
7744	6060	81.0	56.085505
8543	2409	70.0	92.835002
12941	2428	38.0	72.190590
10889	11207	234.0	385.691295
3552	3525	185.0	185.233002
12933	3970	45.0	46.493828

[7]**No se logró conocer mediante la documentación la forma de cálculo del RMSE y MSE para el caso de LightGBM**

5 Evaluación de resultados

Los algoritmos hoy en día se están convirtiendo en un producto muy preciado y útil para las personas, debido a que han logrado optimizar procesos y ejecutar tareas de una manera más sencilla. Y el mundo de hotelería y turismo no es ajeno a ello, Airbnb, es un claro ejemplo, el cual ha logrado captar una gran parte de usuarios y darles un sistema que genera confianza y se adecua con lo que necesitan los usuarios. Sin embargo, se puede recomendar mejoras para que el sistema trabaje de una forma mucho más personalizada. Tomemos como referencia este proyecto para trazar una hoja de ruta, para que en el futuro se desarrollen mejoras al mismo y tenga gran aceptación en el mercado. Nuestro algoritmo es un gran paso para Airbnb, para poder identificar no solo las preferencias del usuario, sino también generarle una recomendación de potenciales lugares donde hospedarse que se ajustan al cliente en base a las experiencias de los otros usuarios.

En un sistema de recomendación, la mejor forma de comprobar si el sistema lo esta realizando bien, es por medio del mismo usuario. Por otro lado, el uso de métricas como es el *coseno de similitud*, ayuda mucha para comprobar los resultados. Lo más importante para este tipo de modelos, es seguir alimentando los datos con el fin de que sean lo más completos posibles. Dentro de nuestros objetivos en este trabajo fue crear un modelo de recomendación híbrido que pueda ajustarse a las preferencias del usuario y predecir lugares de una ciudad de destino. Aunque el objetivo fue alcanzado, se podrían realizar mejoras al modelo, como utilizar más variables en consideración dentro del modelo TF-IDF u otr; pero debemos tener en cuenta que al realizar estos cambios implica que el procesador de la PC tenga mayor capacidad.

Para predecir el precio, resulta intuitivo tener algunas otras consideraciones propias de los lugares, como la estación del año, lo cual influye directamente en los precios y en la disponibilidad de espacios, es decir habitaciones, así como esta podría resultar sobre la discusión otras más, que de hecho podría aportar significativamente en la precisión y también en la razonabilidad del modelo, el cual se espera mejorar tanto a nivel de performance como de tratamiento variables. Sin embargo, se deja marcado un sendero por el cual tomar pie para seguir emprendiendo mejoras. Considerando la utili-

dad del mismo no solo para fines académicos sino que también como ya se ha mencionado, para personas interesadas en poner en marcha el alquiler de habitaciones dentro de su hogar e incluso , hoteles que también han empezado a incursionar dentro de este mundo de Airbnb.

References

Aisulu, Omar, "Alternative Ways to Recommend Airbnb Listings Using Natural Language Processing," 2019.

Grbovic, Mihajlo, Haibin Cheng, Qing Zhang, Lynn Yang, Phillippe Siclait, and Matt Jones, "Listing Embeddings in Search Ranking," 2018.

Hutto, C.J. and Eric Gilbert, "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text," in "in" 01 2015.

James, Gareth, Daniel Witten, Trevor Hastie, and Robert Tibshiriani, *An Introduction to Statistical Learning*, Springer, 2013.

Jurafsky, Daniel and James H. Martin, *Speech and Language Processing An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, Prentice Hall, 2019.

Kazil, Jacqueline and Katharine Jarmul, *Data Wrangling with Python*, O'Reilly, 2016.

Lahiri, Rupa, "NY City Airbnb: price prediction," 2019.

Olivares, Armand, "Building NLP Content-Based Recommender Systems," 2019.

Prateek, Joshi, "Building a Recommendation System using Word2vec: A Unique Tutorial with Case Study in Python," 2019.

Royan, Aldian, "Listing Embeddings in Search Ranking," 2018.

Sharma, Girish, "AirBnB Price Prediction Model," 2020.

Tukey, John W., *Exploratory Data Analysis*, Addison-Wesley Publishing Company, 1977.