**PRÁCTICA FINAL**

**MACHINE LEARNING**

# **Índice de Contenidos**

1. Índice de Contenidos · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - 1
2. Datos de la Práctica · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - 3
3. Origen del Dataset de Trabajo · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - 3
4. Visión de la Estructura del Dataset

y Selección Inicial de Columnas · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - 4

1. Partición de Conjuntos de Training y Test · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - 6
2. Análisis Individual del Conjunto de

Entrenamiento · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - 7

* 1. Columna Experiences Offered - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 7
  2. Columna Host Response Time - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 8
  3. Columna Host Response Rate - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 8
  4. Columna Host Acceptance Rate - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 9
  5. Columna Neighbourhood Group Cleansed - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 10
  6. Columna Neighbourhood Cleansed - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 12
  7. Columnas City, State, Zipcode, Market,

Country Code y Country - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 12

* 1. Columna Property Type - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 13
  2. Columna Room Type - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 13
  3. Columna Accommodates - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 14
  4. Columna Bathrooms - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 14
  5. Columna Bedrooms - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 15
  6. Columna Beds - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 16
  7. Columna Bed Type - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 17
  8. Columna Amenities - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 17
  9. Columna Square Feet - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 18
  10. Columna Price - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 19
  11. Columnas Weekly Price y Monthly Price - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 20
  12. Columna Security Deposit - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 20
  13. Columna Cleaning Fee - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 21
  14. Columna Guests Included - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 21
  15. Columna Extra People - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 22
  16. Columna Minimum Nights - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 23
  17. Columna Maximum Nights - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 23
  18. Columna Number of Reviews - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 24
  19. Columna Review Scores Rating - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 24
  20. Columna Review Scores Accuracy - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 25
  21. Columna Review Scores Cleanliness - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 26
  22. Columna Review Scores Checkin - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 26
  23. Columna Review Scores Communication - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 27
  24. Columna Review Scores Location - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 28
  25. Columna Review Scores Value - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 28
  26. Columna Cancellation Policy - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 29
  27. Columna Reviews per Month - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 30
  28. Columna Geolocation - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 30
  29. Columna Features - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 31

1. Análisis Exploratorio del Conjunto de

Entrenamiento · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - 32

1. Proceso del Conjunto de Test · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - 37
2. Reduciendo la Dimensionalidad · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - 39
   1. Modelo F-Regression - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 39
   2. Modelo Mutual Info - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 40
   3. Modelo Lasso - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 40
   4. Modelo Decission Tree - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 42
   5. Modelo Random Forest - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 43
   6. Modelo Gradient Boosting - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 44
   7. Comparativa de Modelos y Reducción de la

Dimensionalidad - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 47

* 1. Tratamiento de Outliers - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 48

1. Modelos para Predicción · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - 50
   1. Modelo de Regresión Lineal - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 50
   2. Modelo de Regresión Ridge - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 50
   3. Modelo de Regresión Lasso - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 51
   4. Modelo de Regresión Decission Tree - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 52
   5. Modelo de Regresión Random Forest - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 53
   6. Modelo de Regresión Gradient Boosting - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - · - 54
2. Conclusión Final · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - 56
3. Agradecimientos · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - 57

ANEXO · - ·· - · - ·· - · - ·· - · - ·· - · - ·· - 57

# **Datos de la Práctica**

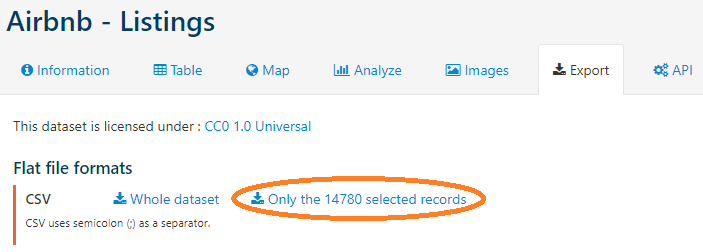
|  |  |
| --- | --- |
| **Bootcamp IV Big Data & Machine Learning - KeepCoding** | |
| **Asignatura** | Machine Learning |
| **Profesor** | Rubén Romero García |
| **Alumno** | Francisco Javier Gonzálvez Chico |
| **Título** | Modelo Predictivo de Precios de Alquiler de Inmuebles en Madrid |

***NOTA***: *En el párrafo ANEXO se incluyen los Notebooks conteniendo todas las celdas de código ejecutadas durante la práctica.*

# **Origen del Dataset de Trabajo**

Utilizaremos, como dataset de trabajo, un fichero de **AIRBNB**, con datos de inmuebles, que descargaremos manualmente desde este enlace a página web:

<https://public.opendatasoft.com/explore/dataset/airbnb-listings/export/?disjunctive.host_verifications&disjunctive.amenities&disjunctive.features&q=Madrid&dataChart=%3D&location=16,41.38377,2.15774&basemap=jawg.streets>

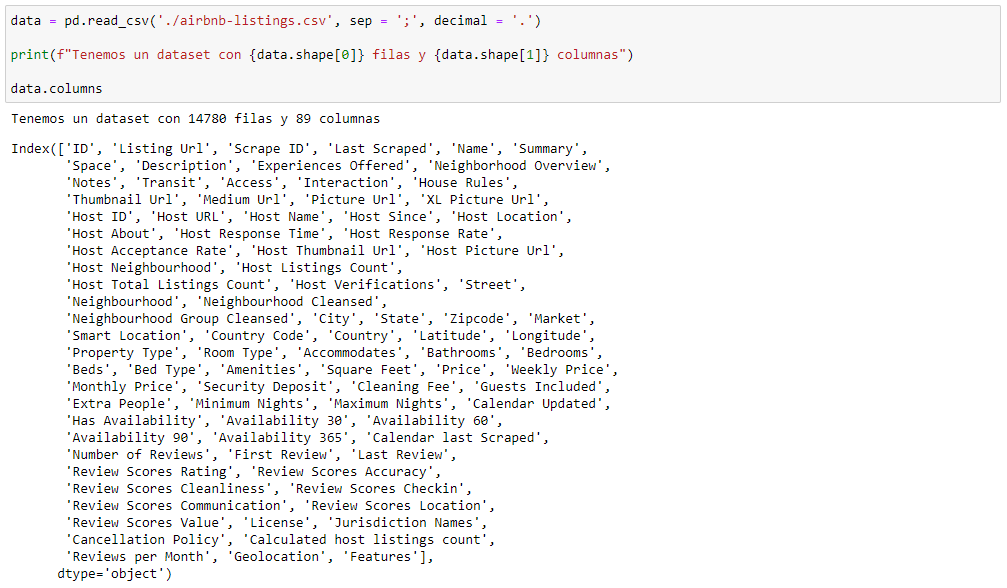
La opción de descarga del dataset a elegir tendrá formato plano CSV y sólo 14.780 registros.

La opción del dataset completo contiene 494.954 (casi medio millón de) filas, lo cual prevemos exigirá un coste computacional excesivo al trabajar localmente y en consecuencia elegimos la versión reducida en detrimento de poder contar con más datos a la hora de entrenar el modelo.

Trabajaremos este dataset en un **JupyterNotebook** paralelo en el que ejecutaremos las celdas con el código oportuno. Celdas que serán copiadas y aportadas en los distintos párrafos de este documento.

El cuaderno completo de ejecución también se anexará al final de este documento.

# **Visión de la Estructura del Dataset y Selección Inicial de Columnas**

Una vez tenemos el dataset de trabajo descargado en local, lo cargamos con **python**/**pandas**.

Y confirmamos que contiene los 14.780 registros estructurados en 89 columnas, número que nos parece demasiado elevado como cantidad de posibles variables predictoras.

Para tener una primera toma de contacto con los datos con los que trabajaremos, daremos un vistazo superficial al dataset y realizaremos una selección inicial de las columnas que nos parezcan adecuadas, conforme a su contenido, como elementos predictores para nuestro modelo, descartando las que creamos que no van a aportar valor.

| **COLUMNA** | **DESCRIPCIÓN DE CONTENIDO** | **TIPO** | **ACCIÓN** |
| --- | --- | --- | --- |
| ID | Código identificador del inmueble en AirBNB | Int | Drop |
| Listing Url | URL. Enlace al inmueble en AirBNB | String | Drop |
| Scrape ID | Timestamp de scrapeo del registro del inmueble | Int | Drop |
| Last Scraped | Fecha. Último scrapeo del registro del inmueble | Date | Drop |
| Name | Texto libre. Nombre del inmueble | String | Drop |
| Summary | Texto libre. Resumen descriptivo del inmueble | String | Drop |
| Space | Texto libre. Descripción del entorno del inmueble | String | Drop |
| Description | Texto libre. Descripción detallada del inmueble | String | Drop |
| Experiences Offered | Tipo de experiencia o adecuada para el inmueble | String | Maintain |
| Neighborhood Overview | Texto libre. Descripción del barrio de ubicación del inmueble | String | Drop |
| Notes | Texto libre. Notas adicionales sobre el inmueble | String | Drop |
| Transit | Texto libre. Medios de transporte cerca del inmueble | String | Drop |
| Access | Texto libre. Acceso al inmueble | String | Drop |
| Interaction | Texto libre. Interacción con el anfitrión | String | Drop |
| House Rules | Texto libre. Reglas del inmueble | String | Drop |
| Thumbnail Url | URL. Enlace a foto tamaño pequeño | String | Drop |
| Medium Url | URL. Enlace a foto tamaño mediano | String | Drop |
| Picture Url | URL. Enlace a foto tamaño grande | String | Drop |
| XL Picture Url | URL. Enlace a foto tamaño extra grande | String | Drop |
| Host ID | Código identificador del anfitrión en AirBNB | Int | Drop |
| Host URL | URL. Enlace al anfitrión en AirBNB | String | Drop |
| Host Name | Texto libre. Nombre del anfitrión | String | Drop |
| Host Since | Fecha. Registro del anfitrión en AirBNB | Date | Drop |
| Host Location | Texto libre. Ubicación del anfitrión | String | Drop |
| Host About | Texto libre. Acerca del anfitrión | String | Drop |
| Host Response Time | Tasa temporal de respuesta del anfitrión | String | Maintain |
| Host Response Rate | Tasa de respuesta del anfitrión | Int | Maintain |
| Host Acceptance Rate | Tasa de aceptación de huéspedes del anfitrión | String | Maintain |
| Host Thumbnail Url | URL. Enlace a fotografía tamaño pequeño del anfitrión | String | Drop |
| Host Picture Url | URL. Enlace a fotografía del anfitrión | String | Drop |
| Host Neighbourhood | Vecindario o barrio del anfitrión | String | Drop |
| Host Listings Count | Conteo de veces que se selecciona al anfitrión | Int | Drop |
| Host Total Listings Count | Cuenta total de veces que se selecciona al anfitrión | Int | Drop |
| Host Verifications | Verificaciones del anfitrión | String | Drop |
| Street | Texto libre. Dirección del inmueble | String | Drop |
| Neighbourhood | Texto libre. Vecindario o barrio en el que se ubica el inmueble | String | Drop |
| Neighbourhood Cleansed | Vecindario o barrio limpio en el que se ubica el inmueble | String | Maintain |
| Neighbourhood Group Cleansed | Distrito en el que se ubica el inmueble | String | Maintain |
| City | Ciudad en la que se ubica el inmueble | String | Maintain |
| State | Provincia o estado en el que se ubica el inmueble | String | Maintain |
| Zipcode | Código postal en el que se ubica el inmueble | String | Maintain |
| Market | Mercado de alquiler del inmueble | String | Maintain |
| Smart Location | Ubicación genérica del inmueble | String | Drop |
| Country Code | Código identificador del país en el que se ubica el inmueble | String | Maintain |
| Country | País en el que se ubica el inmueble | String | Maintain |
| Latitude | Coordenada geoposicional de latitud de ubicación del inmueble | String | Drop |
| Longitude | Coordenada geoposicional de longitud de ubicación del inmueble | String | Drop |
| Property Type | Tipo de inmueble | String | Maintain |
| Room Type | Tipo de alquiler del inmueble | String | Maintain |
| Accommodates | Cupo de inquilinos incluido en el alquiler del inmueble | Int | Maintain |
| Bathrooms | Número de baños en el inmueble | Float | Maintain |
| Bedrooms | Número de habitaciones en el inmueble | Int | Maintain |
| Beds | Número de camas en el inmueble | Int | Maintain |
| Bed Type | Tipo de camas en el inmueble | String | Maintain |
| Amenities | Comodidades del inmueble | String | Maintain |
| Square Feet | Superficie del inmueble en pies cuadrados | Int | Maintain |
| Price | Precio en euros de alquiler por día del inmueble | Int | Target |
| Weekly Price | Precio en euros de alquiler por semana del inmueble | Int | Maintain |
| Monthly Price | Precio en euros de alquiler por mes del inmueble | Int | Maintain |
| Security Deposit | Depósito o fianza en euros para el alquiler del inmueble | Int | Maintain |
| Cleaning Fee | Recargo o tasa en euros por la limpieza del inmueble | Int | Maintain |
| Guests Included | Número máximo de invitados permitido en el inmueble | Int | Maintain |
| Extra People | Recargo en euros por invitado | Int | Maintain |
| Minimum Nights | Número mínimo de noches a alquilar el inmueble | Int | Maintain |
| Maximum Nights | Número máximo de noches a alquilar el inmueble | Int | Maintain |
| Calendar Updated | Calendario de disponibilidad del inmueble actualizado en AirBNB | String | Drop |
| Has Availability | Disponibilidad actual del inmueble | String | Drop |
| Availability 30 | Disponibilidad a 30 días del inmueble | Int | Drop |
| Availability 60 | Disponibilidad a 60 días del inmueble | Int | Drop |
| Availability 90 | Disponibilidad a 90 días del inmueble | Int | Drop |
| Availability 365 | Disponibilidad a 365 días del inmueble | Int | Drop |
| Calendar last Scraped | Fecha. Último scrapeo del calendario de disponibilidad del inmueble | Date | Drop |
| Number of Reviews | Número de comentarios sobre el inmueble | Int | Maintain |
| First Review | Fecha. Primer comentario sobre el inmueble | Date | Drop |
| Last Review | Fecha. Último comentario sobre el inmueble | Date | Drop |
| Review Scores Rating | Tasa de puntuación de los comentarios del inmueble | Int | Maintain |
| Review Scores Accuracy | Puntuación sobre la veracidad de la información del inmueble | Int | Maintain |
| Review Scores Cleanliness | Puntuación sobre la limpieza de los comentarios del inmueble | Int | Maintain |
| Review Scores Checkin | Puntuación sobre el recibimiento de los comentarios del inmueble | Int | Maintain |
| Review Scores Communication | Puntuación sobre la comunicación de los comentarios del inmueble | Int | Maintain |
| Review Scores Location | Puntuación sobre la localización de los comentarios del inmueble | Int | Maintain |
| Review Scores Value | Puntuación general de los comentarios del inmueble | Int | Maintain |
| License | Código identificador de licencia | String | Drop |
| Jurisdiction Names | Jurisdicción bajo la que se ampara el alquiler del inmueble | String | Drop |
| Cancellation Policy | Política de cancelación de reservas del inmueble | String | Maintain |
| Calculated host listings count | Cuenta calculada de veces que se selecciona al anfitrión | Int | Drop |
| Reviews per Month | Comentarios sobre el inmueble por mes | Float | Maintain |
| Geolocation | Coordenadas de geolocalización de la ubicación del inmueble | String | Maintain |
| Features | Características del anfitrión y del inmueble | String | Maintain |

Después de este somero vistazo, en esta selección inicial hemos decidido mantener 42 columnas del dataset de trabajo y, por tanto, descartar 47.

Hemos descartado todas las columnas clasificadas como códigos identificadores, fechas, URLs o textos libres que entendemos que no tendrán valor predictor en nuestro modelo. También se han desestimado columnas referentes a información personal o del perfil del anfitrión porque no creemos que tengan relación alguna con precio de alquiler de los inmuebles, por ejemplo, el barrio en el que reside el anfitrión, las veces que aparece en las búsquedas o sus verificaciones de perfil. Asimismo, hemos desechado columnas con datos de ubicación del inmueble cuya información nos parece redundante, o su formato poco adecuado, habiendo otras columnas más útiles. Y por último decidimos prescindir de las columnas con información sobre la disponibilidad actual y a futuro del inmueble entendiendo que tampoco aportarán un valor predictor al modelo.

# **Partición de Conjuntos de Training y Test**

Nos enfrentamos a un problema de regresión (no logística), ya que nuestro fin es predecir el precio de alquiler de los inmuebles. Por tanto, nuestra variable o columna objetivo será **Price**.

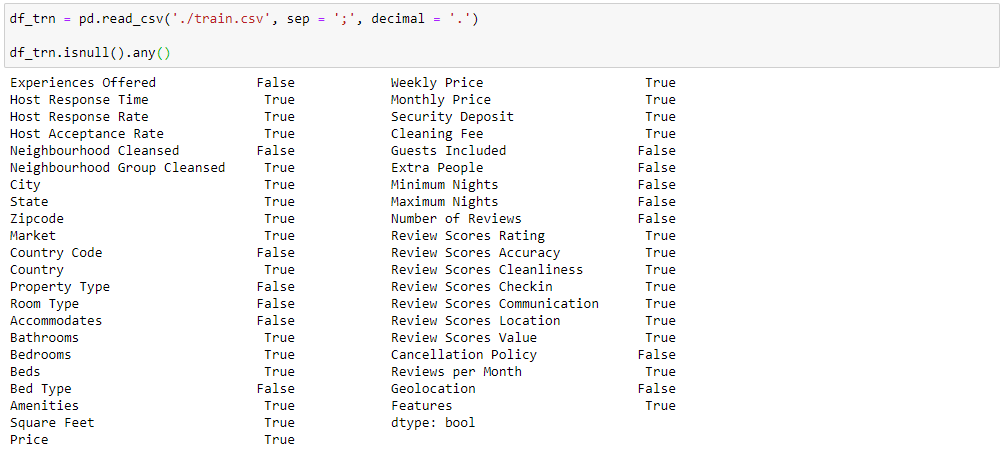
Como en la elaboración de cualquier modelo de machine learning, ya sea de clasificación o de regresión, lo primero –tras entender y acomodarnos a los datos con los que debemos trabajar– es realizar el particionado del dataset en dos conjuntos de registros, uno para el entrenamiento del modelo y otro para su testeo posterior.

Para ello utilizaremos el método **train\_test\_split** de la librería **model\_selection** de **scikit-learn**. Asignando un 75% de los registros al conjunto de entrenamiento y un 25% para el de testeo.

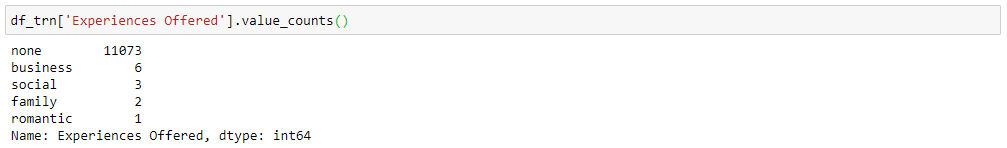
Una vez partido el dataset de trabajo y obtenidos los conjuntos de datos de entrenamiento y testeo los almacenamos en dos archivos CSV y la partición de test quedará apartada y aislada de nuestra tarea de análisis hasta que le llegue su turno de uso.

# **Análisis Individual del Conjunto de Entrenamiento**

Analizaremos detalladamente el conjunto de datos de entrenamiento, columna a columna, e iremos aplicando los tratamientos o procesos que vayamos decidiendo según el análisis.

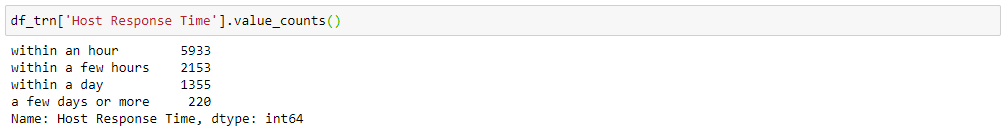
Leemos el archivo train.csv y como prioridad vemos qué columnas contienen valores nulos (missings), ya que esto condicionará también de cierta forma su análisis.

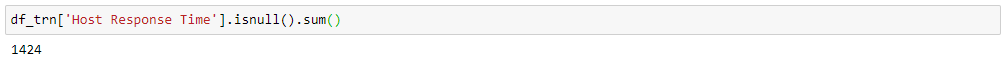
## **Columna Experiences Offered**

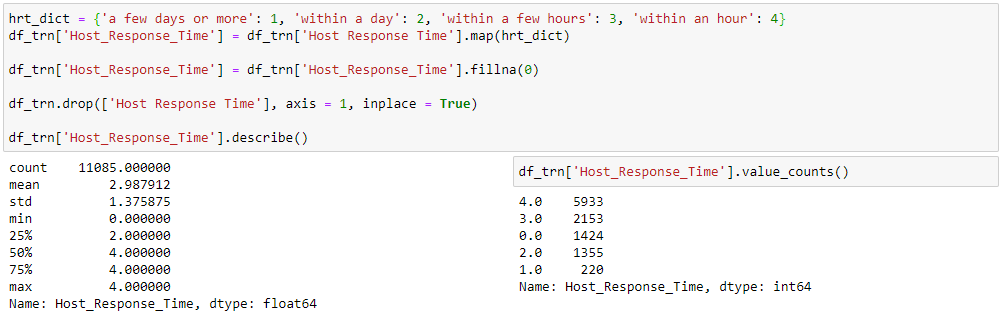
Esta columna no se descartó inicialmente por indicar el tipo de experiencia que ofrece cada inmueble, pensando que el precio de alquiler pudiera ser diferente si el inmueble está orientado a un entorno de negocios, romántico o de tipo familiar. La columna es numérica, por tanto, no necesitará categorización.

Como ya sabemos, esta columna no contiene valores nulos y además ahora podemos observar que únicamente 12 registros –de los 11.085 totales– tienen información significativa, 11.073 son indefinidos, es decir, sólo un 0,1% de los registros podrían aportar un valor diferenciador al modelo predictor. Este porcentaje nos parece ínfimo y decidimos descartar definitivamente esta columna.

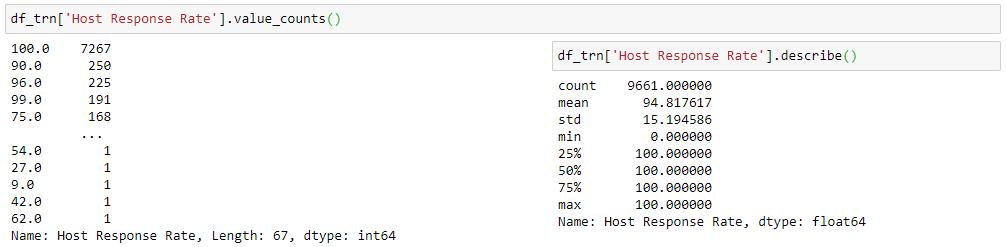
## **Columna Host Response Time**

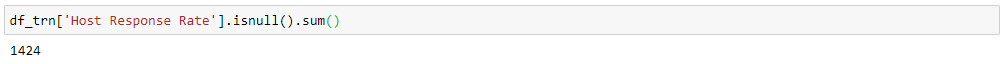
Esta columna no se descartó inicialmente, por marcar el tiempo de respuesta del anfitrión para aceptar/denegar una reserva su inmueble, pensando que anfitriones más dedicados o atentos pudieran ser empresas o profesionales y que esto pudiera influenciar el precio de sus inmuebles.

Sabemos que la columna contiene nulos. Confirmamos su número a continuación.

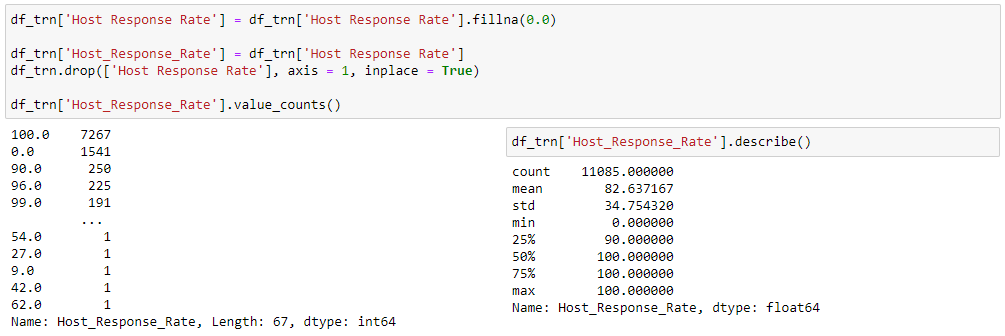
Nos parece justo pensar, que en los casos en los que esta columna no contiene información, la causa pudiera estar en que nunca nadie haya querido reservar el inmueble y en consecuencia el anfitrión no haya podido responder. Asignamos pesos a los valores de la columna –y la renombramos de forma adecuada– con mayor valor a los mejores tiempos de respuesta del anfitrión y 0 para los nulos.

## **Columna Host Response Rate**

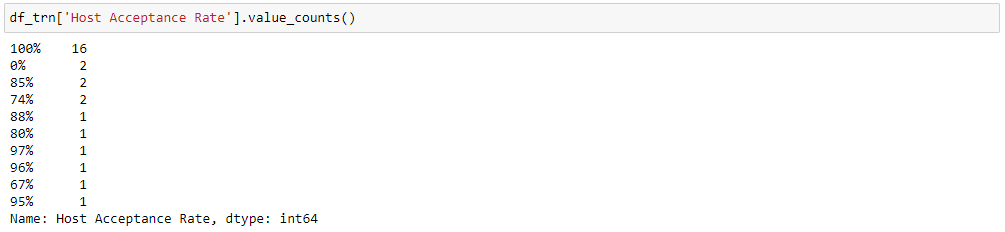
Esta columna no se descartó inicialmente por indicar el porcentaje de respuestas del anfitrión cuando le reservan su inmueble, pensando que anfitriones más dedicados o atentos pudieran ser empresas o profesionales y que esto pudiera influenciar el precio de sus inmuebles. La columna es numérica, por tanto, no necesitará categorización.

Sabemos que la columna contiene nulos. Confirmamos su número a continuación.

Tiene toda la lógica, al ver que el número de nulos coincide con el que tenía la columna anterior, que si un anfitrión no ha podido responder por no haber tenido nunca reservas de su inmueble, su tasa de respuestas sea nula en consecuencia.

Decidimos rellenar estos valores nulos con el valor 0.0, para que la columna quede total y consecuentemente informada. Y la renombramos de forma adecuada.

## **Columna Host Acceptance Rate**

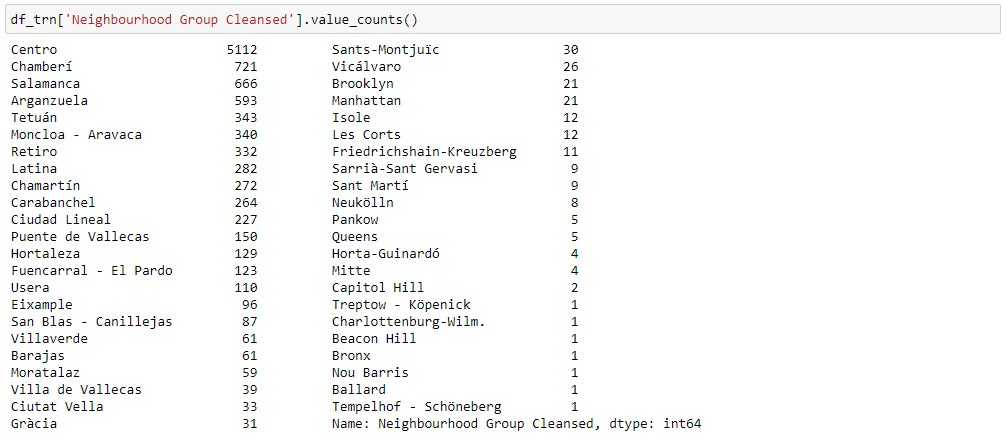
Esta columna no se descartó inicialmente por indicar el porcentaje de aceptación de reservas del anfitrión, pensando que los anfitriones más dedicados o atentos pudieran ser empresas o profesionales y que esto pudiera influenciar el precio de sus inmuebles.

Observamos un número muy bajo de registros informados –sólo 28 de los 11.085 totales – y sabemos que la columna contiene nulos. Confirmaremos su número a continuación.

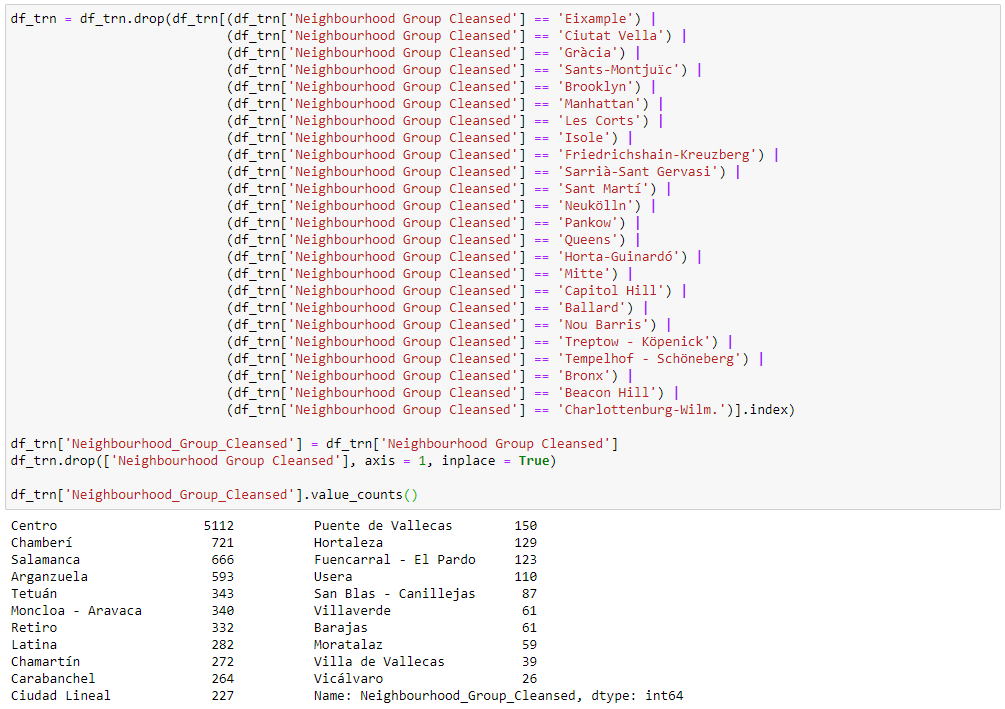
A la vista de lo anterior, sólo un 0,25% de los registros podrían dar un valor diferenciador al modelo predictor. Este porcentaje nos parece ínfimo y decidimos descartar definitivamente esta columna.

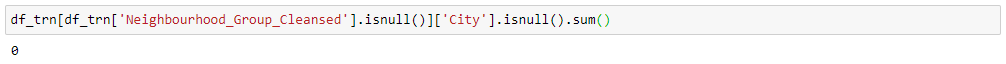
## **Columna Neighbourhood Group Cleansed**

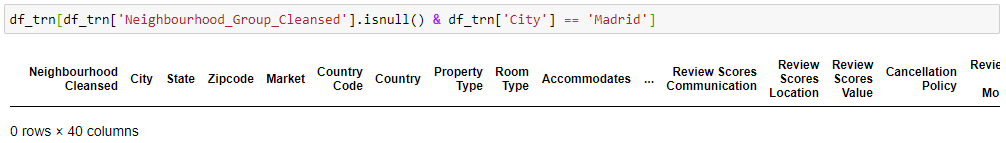
Vamos a anteponer el análisis de esta columna, al de **Neighbourhood Cleansed**, por ser su contenido un nivel jerárquico superior, lo cual nos puede ayudar o simplificar analizarla después.

Esta columna no se descartó inicialmente por indicar el distrito en el que se ubica cada inmueble, sabiendo que diferentes zonas de la misma ciudad infieren un valor más o menos alto a sus inmuebles e implícitamente a sus precios de alquiler.

Sabemos que esta columna contiene valores nulos. Confirmamos su número a continuación.

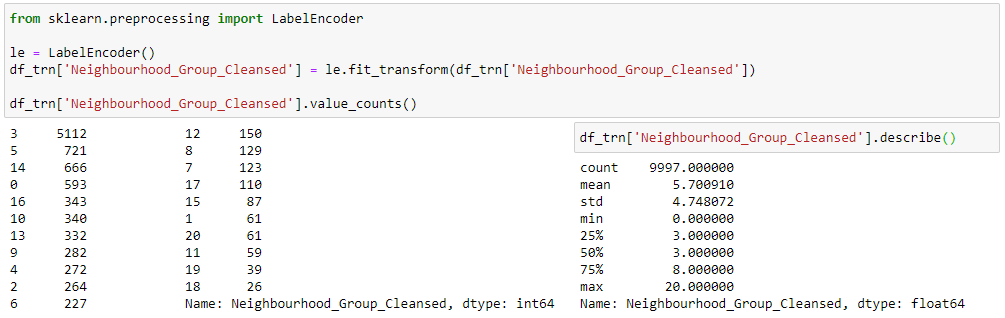
Entre sus diferentes valores podemos ver distritos que no pertenecen a la ciudad de Madrid. Estos valores no pueden considerarse outliers, puesto que realmente no son válidos, al menos atendiendo al scope de nuestro trabajo ***Modelo Predictivo de Precios de Alquiler de Inmuebles en Madrid***, por tanto, debemos considerarlos errores y decidimos descartar sus observaciones del conjunto de entrenamiento. Aprovechamos para renombrar de forma adecuada la columna.

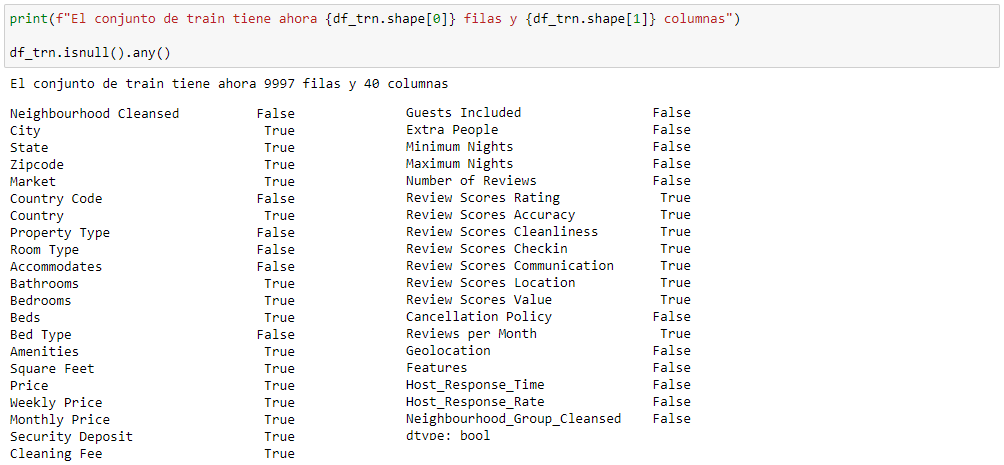
Seguimos teniendo 768 valores nulos en la columna. Veamos cuántos de ellos tienen también valor nulo en la columna **City** cuyo contenido es un nivel jerárquico superior.

Ninguno. Veamos entonces cuántos de ellos tienen el valor Madrid en la columna **City**.

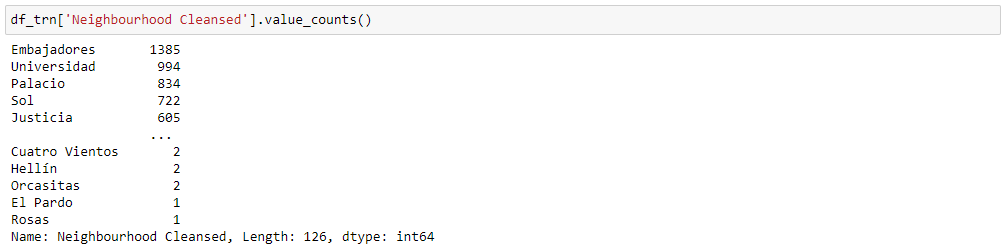
Ninguno(\*). En consecuencia, decidimos eliminar también sus observaciones del conjunto de entrenamiento puesto que debemos considerarlos errores por no entrar en el scope de nuestro trabajo.

(\*) Esta consulta también se ha comprobado visualizando en el propio fichero CSV los 768 registros.

Una vez tratados los valores erróneos de la columna es el momento de categorizarla y para ello utilizaremos el método **LabelEncoder** de la librería **preprocessing** de **scikit-learn**.

Hasta ahora hemos descartado 2 columnas y eliminado 1.088 registros de nuestro conjunto de entrenamiento. Por ello debemos ver qué aspecto tiene nuestro dataset y qué columnas tienen en este momento valores nulos.

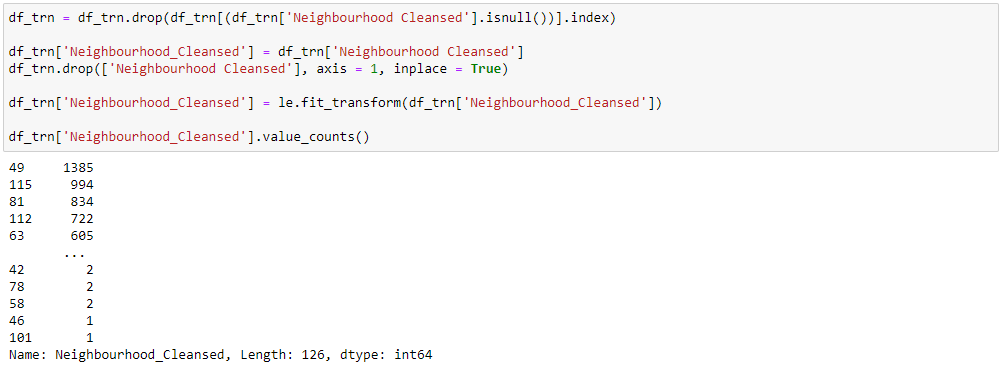
## **Columna Neighbourhood Cleansed**

Esta columna no se descartó inicialmente por indicar el barrio o vecindario en el que se ubica cada inmueble, sabiendo que diferentes zonas de la misma ciudad infieren un valor más o menos alto a sus inmuebles e implícitamente a sus precios de alquiler.

Sabemos que esta columna tiene 126 valores únicos(\*\*) y no contiene nulos.

(\*\*) Esta columna se ha comprobado visualizándola en el propio fichero CSV y todos sus valores son barrios de Madrid, correctos.

Pero debemos tener en cuenta que el hecho de que esta columna no contenga valores nulos en el conjunto de entrenamiento no significa que en otros conjuntos de datos no vaya a tenerlos.

Decidimos eliminar los posibles registros con valores nulos en esta columna y acto seguido la categorizamos utilizando el método **LabelEncoder** y la renombramos de forma adecuada.

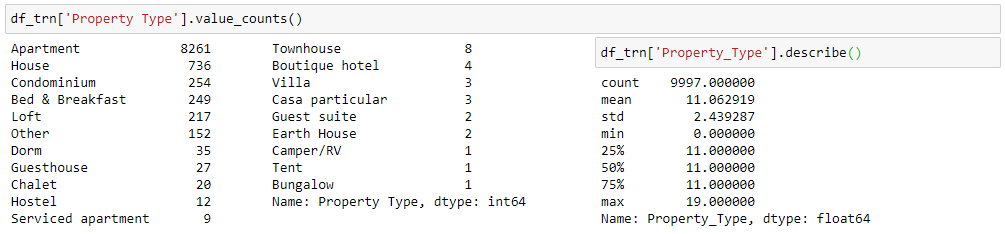
## **Columnas City, State, Zipcode, Market, CountryCode y Country**

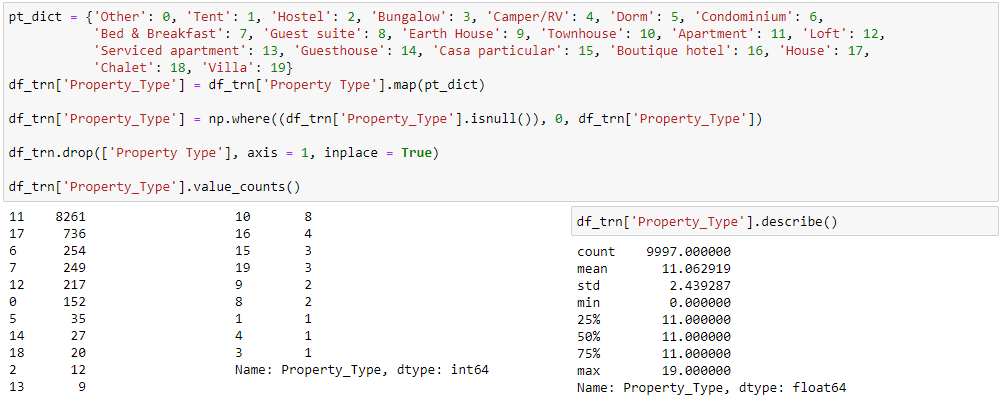
Estas columnas no se descartaron inicialmente porque se pensó que podrían ser útiles a la hora de sanear otras columnas con datos de ubicación de los inmuebles.

Así ha sido, al poder utilizar la columna **City** en el tratamiento de cleansing de la columna **Neighbourhood Group Cleansed**.

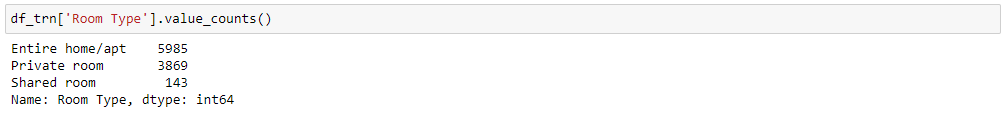
Una vez cumplida su función las descartamos definitivamente del conjunto de entrenamiento, porque nos parece que contienen valores redundantes, únicos –y por ende no diferenciadores para el modelo– o altamente correlacionados con el resto de columnas de ubicación.

## **Columna PropertyType**

Esta columna no se descartó inicialmente por indicar el tipo del inmueble, sabiendo que influirá en el precio de alquiler el hecho de ofrecer una casa, una villa o una tienda de campaña.

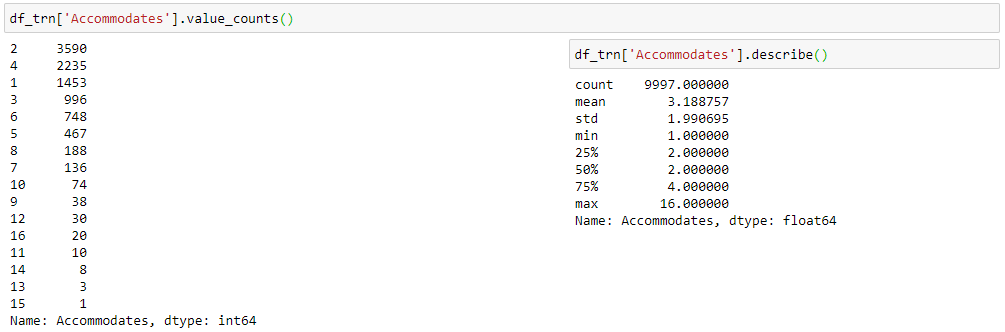
Sabemos que esta columna no contiene valores nulos en el conjunto de entrenamiento, pero prevendremos que puedan existir en otros conjuntos de datos asimilando sus valores al valor Other. Categorizamos la columna –y la renombramos adecuadamente– asignando pesos con mayor valor a los –que pensamos son– mejores tipos de inmuebles.

## **Columna RoomType**

Esta columna no se descartó inicialmente por indicar el tipo de alquiler que se ofrece de cada inmueble, sabiendo que influirá en el precio el hecho de alquilar el inmueble completo, una habitación privada o una habitación compartida.

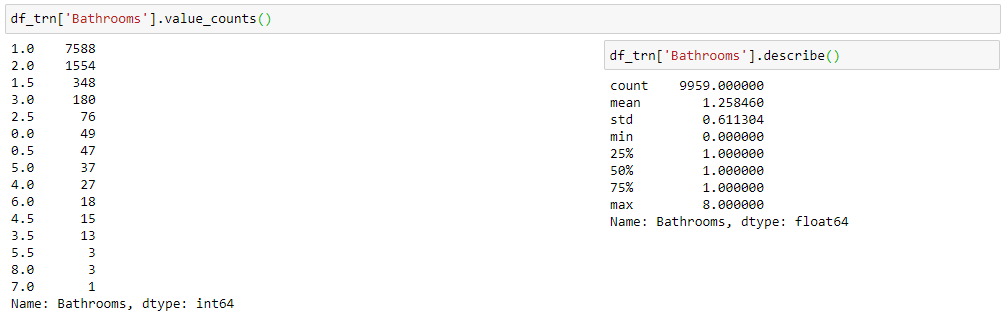
Sabemos que esta columna no contiene valores nulos en el conjunto de entrenamiento, pero prevendremos que puedan existir en otros conjuntos de datos asimilando sus valores al valor Entire home/apt por ser el mayoritario del conjunto. Acto seguido pasamos a categorizar la columna –y a renombrarla de forma adecuada– asignando pesos con mayor valor a los –que pensamos son– tipos de alquiler con más privacidad.

## **Columna Accommodates**

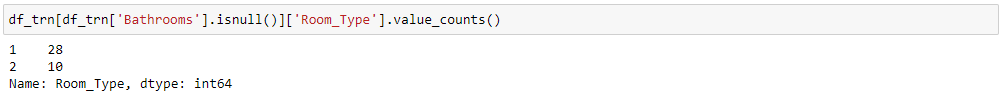
Esta columna no se descartó inicialmente por indicar el cupo de personas incluidas en el precio de alquiler de un inmueble, sabiendo que influirá en él que el inmueble pueda albergar más o menos personas. La columna es numérica, por tanto, no necesitará categorización.

Sabemos que esta columna no contiene valores nulos en el conjunto de entrenamiento, pero prevendremos que puedan existir en otros conjuntos de datos asimilando sus valores al valor 2 por ser el mayoritario del conjunto.

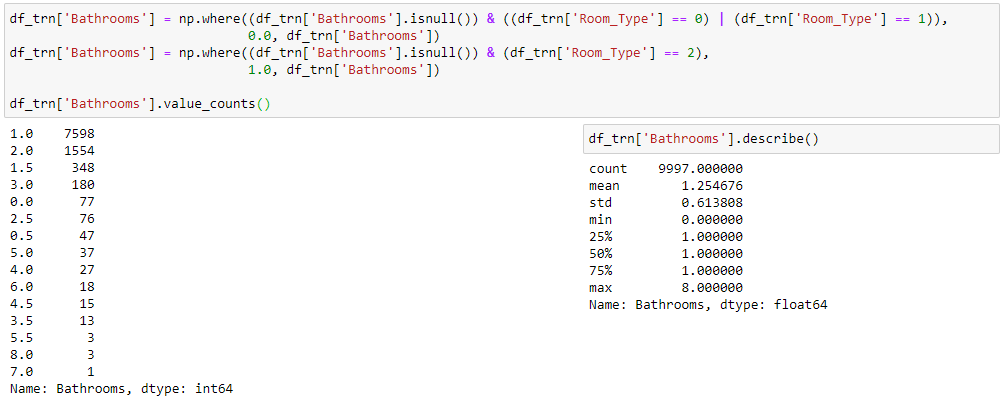
## **Columna Bathrooms**

Esta columna no se descartó inicialmente por indicar el número de cuartos de baño de cada inmueble, sabiendo que influirá en el precio de su alquiler que un inmueble cuente con más o menos cuartos de este tipo. La columna es numérica, por tanto, no necesitará categorización.

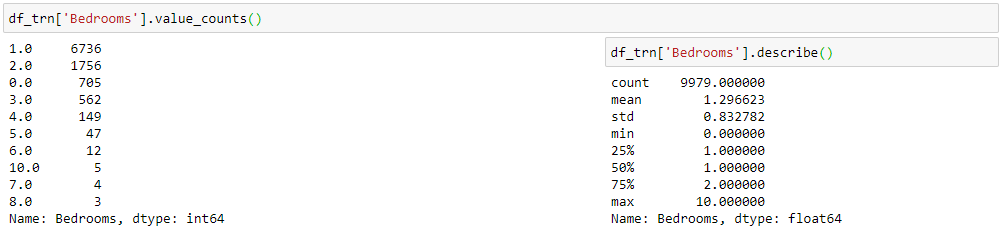
Sabemos que esta columna contiene valores nulos. Confirmamos su número a continuación.

Veamos qué tipo de alquiler, –qué valores de **Room\_Type**– tienen estos registros que contienen nulos en la columna **Bathrooms**.

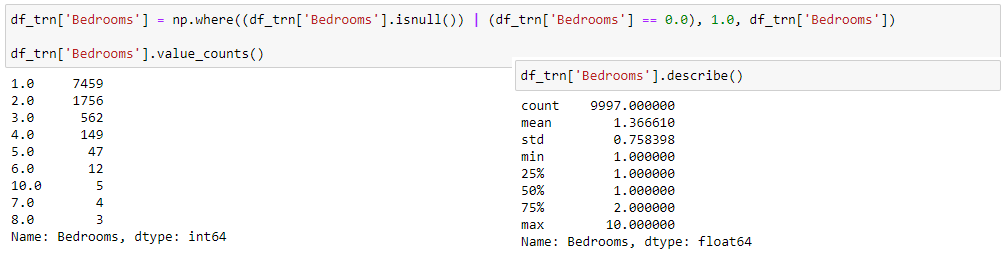
Según la consulta realizada, su tipo de alquiler es Privateroom o Entirehome/apt.

Decidimos que si el tipo de alquiler es Private room –o Shared room, para incluir también esa opción por si acaso existiera en el conjunto de testeo– los nulos de la columna **Bathrooms** tomarán el valor 0.0 y si el tipo de alquiler es Entire home/apt tomarán el valor 1.0, ya que entendemos que una habitación –sea privada o compartida– no tiene por qué tener cuarto de baño privado (o se indicaría), mientras que un inmueble completo tendrá al menos uno.

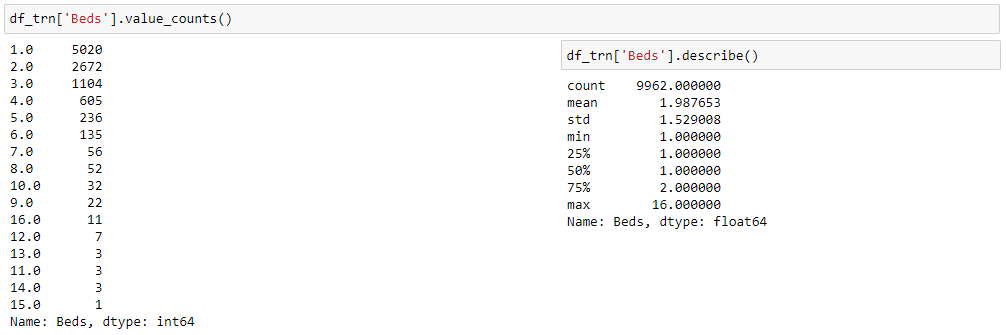
## **Columna Bedrooms**

Esta columna no se descartó inicialmente por indicar el número de habitaciones con las que cuenta cada inmueble, sabiendo que influirá en el precio de su alquiler que un inmueble tenga más o menos dormitorios. Esta columna es numérica, por lo que no necesitará categorización.

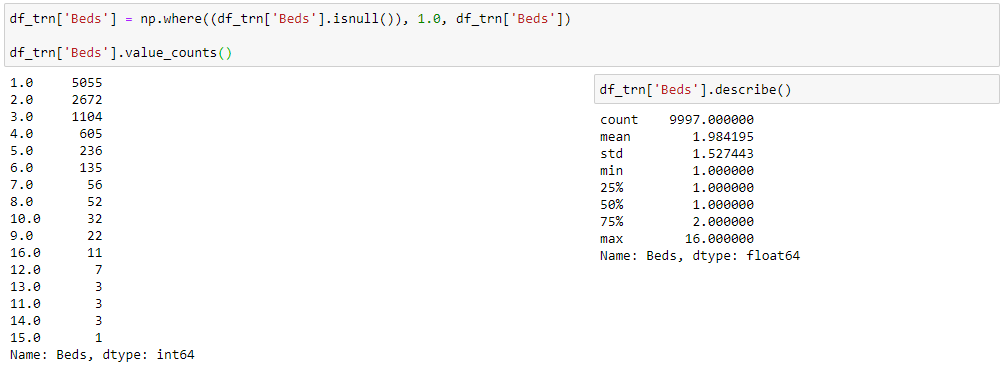
Sabemos que esta columna contiene valores nulos. Confirmamos su número a continuación.

Decidimos adjudicar el valor 1.0a la columna **Bedrooms**, tanto en los registros con valor nulo, como en los que tienen el valor 0.0, ya que entendemos que cualquier tipo de inmueble tendrá al menos un cuarto o un espacio –aunque no sea un dormitorio propiamente dicho– en el que dispondrá de algún tipo de cama.

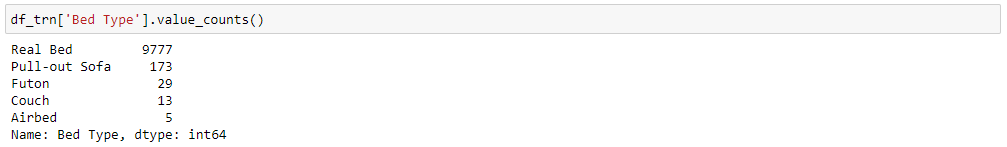
## **Columna Beds**

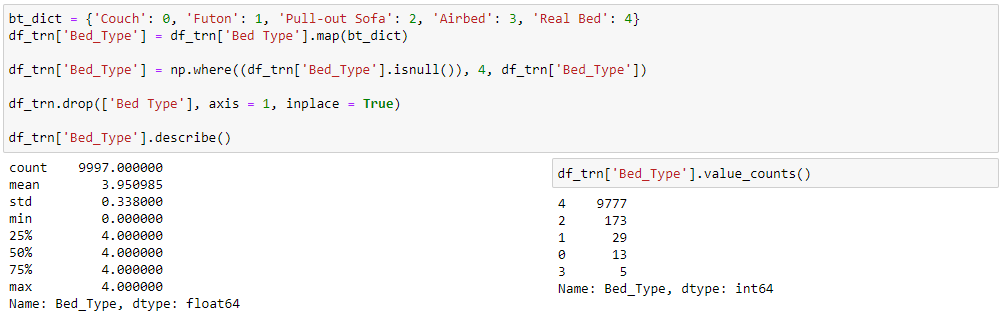
Esta columna no se descartó inicialmente por indicar el número de camas disponibles en cada inmueble, sabiendo que influirá en el precio de su alquiler que el inmueble cuente con más o con menos. Esta columna es numérica, por lo que no necesitará categorización.

Sabemos que esta columna contiene valores nulos. Confirmamos su número a continuación.

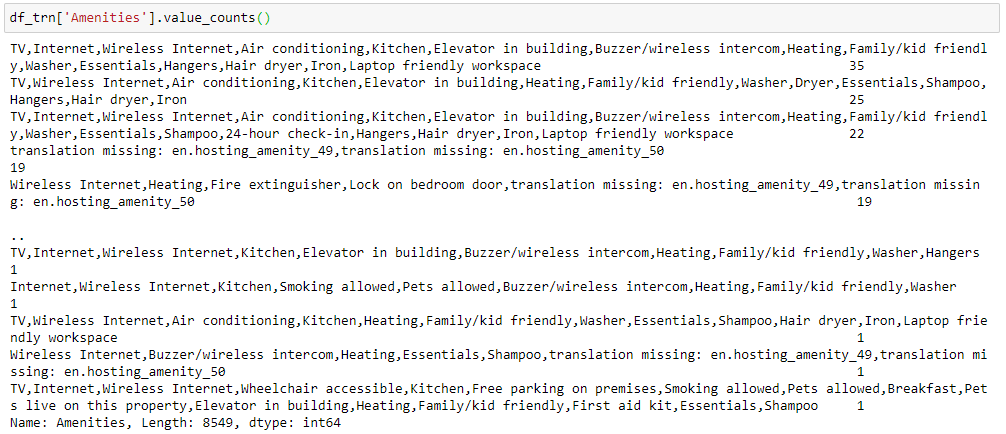
Decidimos adjudicar el valor 1.0 a los registros con valor nulo, ya que entendemos que cualquier tipo de inmueble debería tener al menos una cama.

## **Columna Bed Type**

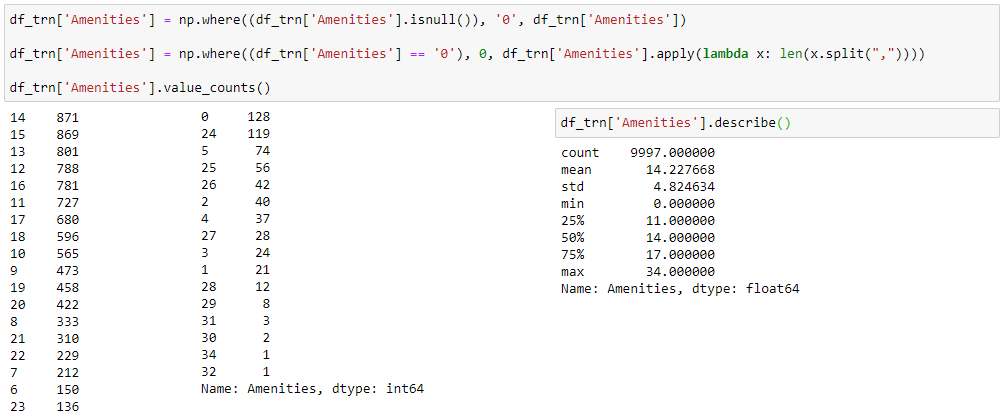
Esta columna no se descartó inicialmente por indicar el tipo de las camas disponibles en cada inmueble, sabiendo que influirá en el precio de su alquiler que un inmueble cuente con camas de mejor o peor calidad.

Sabemos que esta columna no contiene valores nulos en el conjunto de entrenamiento, pero prevendremos que puedan existir en otros conjuntos de datos asimilando sus valores al valor Real Bed ya que es, con diferencia, el mayoritario del conjunto. Acto seguido pasamos a categorizar la columna –y a renombrarla de forma adecuada– asignando pesos con mayor valor a los –que pensamos son– tipos de camas de mejor calidad.

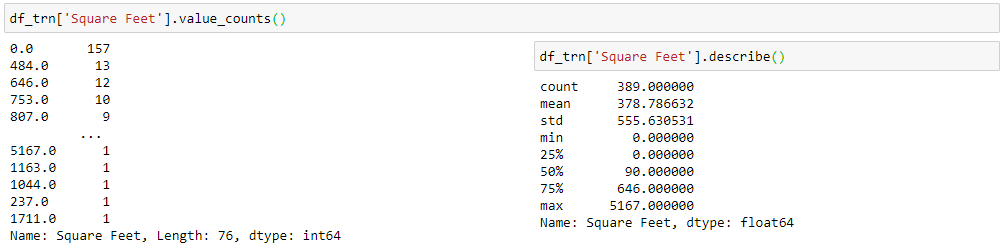
## **Columna Amenities**

Esta columna no se descartó inicialmente por indicar las comodidades disponibles en cada inmueble, sabiendo que influirá en el precio de su alquiler que el inmueble cuente con más o menos lujos o servicios.

Esta columna contiene, o valores nulos, o texto libre y se nos ocurre que podríamos categorizarla sumando las comodidades que contenga –y asignando valor cero a los registros con valores nulos–. Esto le daría un peso directamente proporcional a los lujos de cada inmueble.

Bien es cierto que se puede valorar más un inmueble con ascensor o con wireless internet frente a otro que provea champú o una televisión y que se podría realizar un estudio de pesos más detallado o ajustado, pero entendiendo que el coste sería alto para el posible valor futuro que aportaría, nos sirve una simple adición.

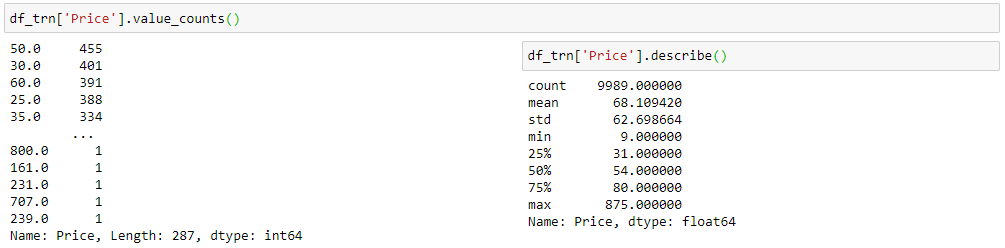
## **Columna Square Feet**

Esta columna no se descartó inicialmente por indicar la superficie de cada inmueble, sabiendo que influirá en el precio de su alquiler el hecho de alquilar inmuebles más o menos espaciosos. Esta columna es numérica, por lo que no necesitará categorización.

Sabemos que esta columna contiene valores nulos. Confirmamos su número a continuación.

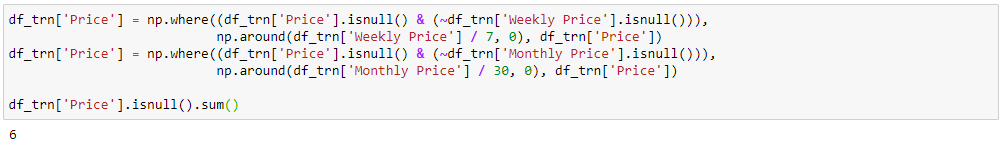
Observamos 9.608 registros con valor nulos y 157 registros con valor 0.0 –erróneos puesto que no puede existir un inmueble sin superficie–. Estos registros inválidos suman 9.765 de un total de 9.997, es decir, únicamente un 2,32% de los registros podrían aportar un valor diferenciador al modelo, lo cual nos parece ínfimo y decidimos descartar definitivamente esta columna.

## **Columna Price**

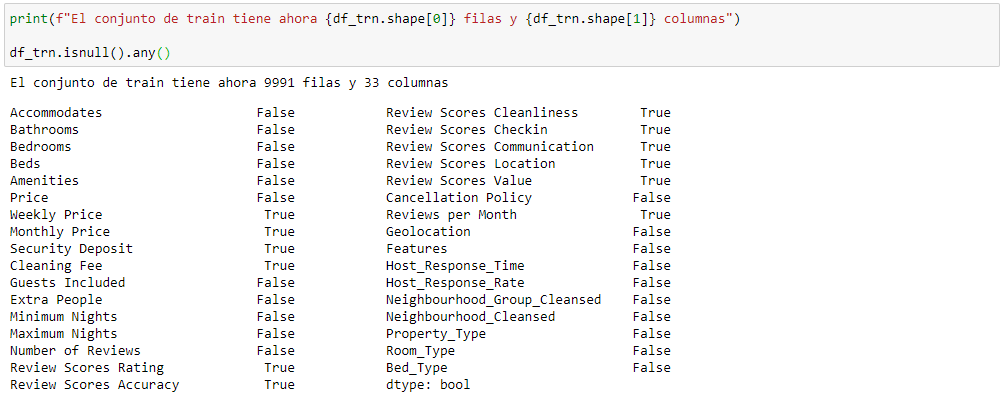
Esta columna no se descartó evidentemente por ser nuestro target: el precio en euros de alquiler de los inmuebles en Madrid. Esta columna es numérica, por lo que no necesitará categorización.

El comando *describe* nos muestra que la columna no contiene valores erróneos –no contiene valores 0.0–, pero sabemos que contiene nulos. Confirmemos su número.

El número de nulos es muy bajo, un 0,08% del conjunto de entrenamiento. Antes de descartarlos podemos mirar si otras columnas nos pueden ayudar a asimilar un valor para ellos.

Las columnas **Weekly Price** y **Monthly Price** –estando informadas– indican el precio semanal y mensual de alquiler cada inmueble. Por tanto, si tienen valor, en estos registros con valor nulo en **Price**, lo dividiremos entre 7 y entre 30, respectivamente, para asignarlo.

Tras la operación se han conseguido rescatar 2 registros nulos. El resto decidimos considerarlos irrecuperables y los descartamos definitivamente.

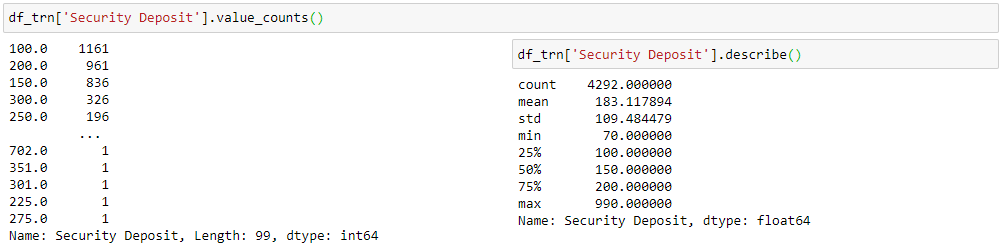
Desde el último descarte (en el párrafo 6.5) se han eliminado 7 columnas y 6 registros más de nuestro conjunto de entrenamiento. Por ello debemos ver de nuevo qué aspecto tiene nuestro dataset y qué columnas tienen ahora valores nulos.

## **Columnas Weekly Price y Monthly Price**

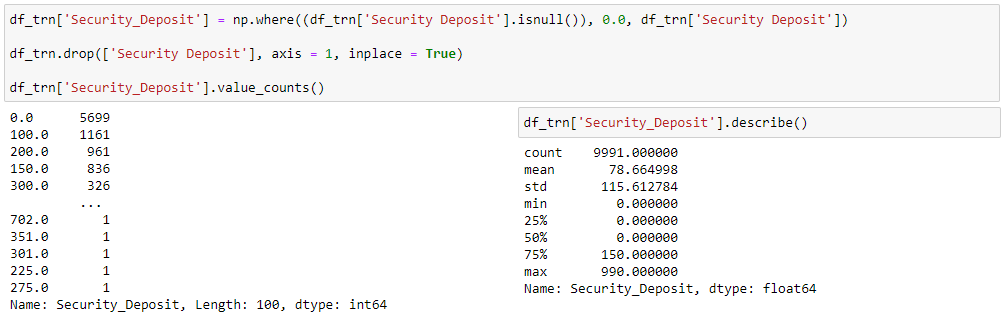
Estas columnas no se descartaron inicialmente porque se pensó que podrían ser útiles a la hora de tratar la columna **Price**, como así ha sido.

Una vez nos han servido para tal fin, no les otorgamos gran valor adicional puesto que seguro están totalmente correlacionadas con la variable objetivo y no aportarán significancia estadística en el modelo. En consecuencia, las descartamos definitivamente.

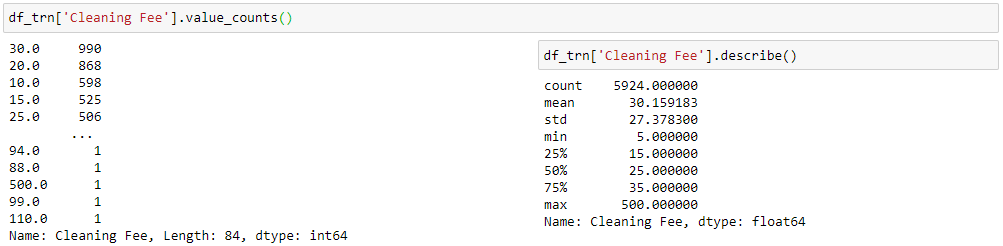
## **Columna Security Deposit**

Esta columna no se descartó inicialmente por indicar la cantidad en euros que se pide depositar como fianza si se desea alquilar un inmueble, pensando que esa cantidad fuera directamente proporcional al valor que cada propietario da a su inmueble y que esto influyera en su precio de alquiler. Esta columna es numérica, por lo que no necesitará categorización.

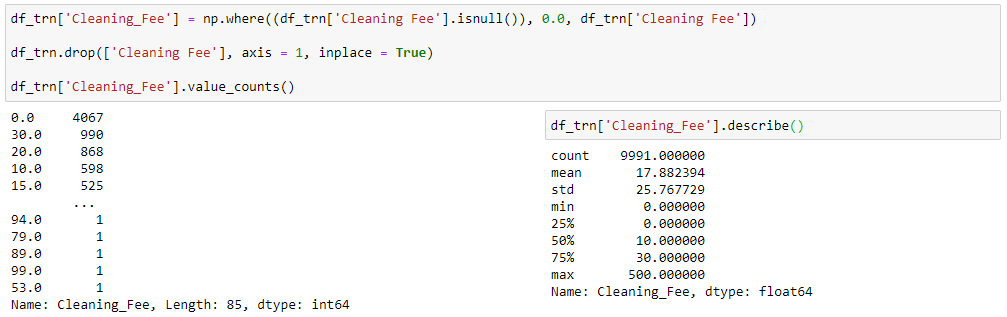
Sabemos que la columna contiene nulos. Confirmemos su número.

Decidimos adjudicar el valor 0.0 a los registros con valor nulo, aun teniendo en cuenta que existe un número bastante alto de ellos, ya que entendemos que si no se ha especificado una fianza es porque no se cobrará. Al mismo tiempo renombramos la columna de forma correcta.

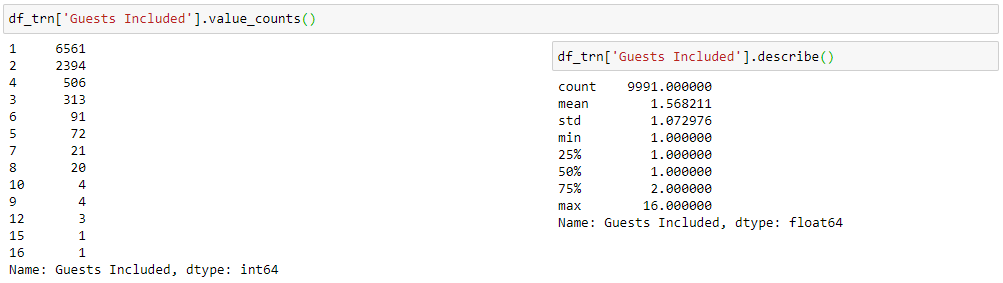
## **Columna Cleaning Fee**

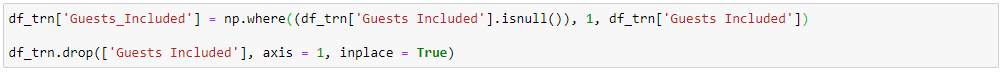
Esta columna no se descartó inicialmente por indicar la cantidad en euros que se cobrará como recargo por el servicio de limpieza de un inmueble, pensando que esa cantidad pudiera ser directamente proporcional a lo que cuesta limpiar el inmueble –siempre dependiendo de su superficie– y que, por tanto, a mayor superficie se incrementará también el precio de alquiler del inmueble. Esta columna es numérica, por lo que no necesitará categorización.

Sabemos que la columna contiene nulos. Confirmemos su número.

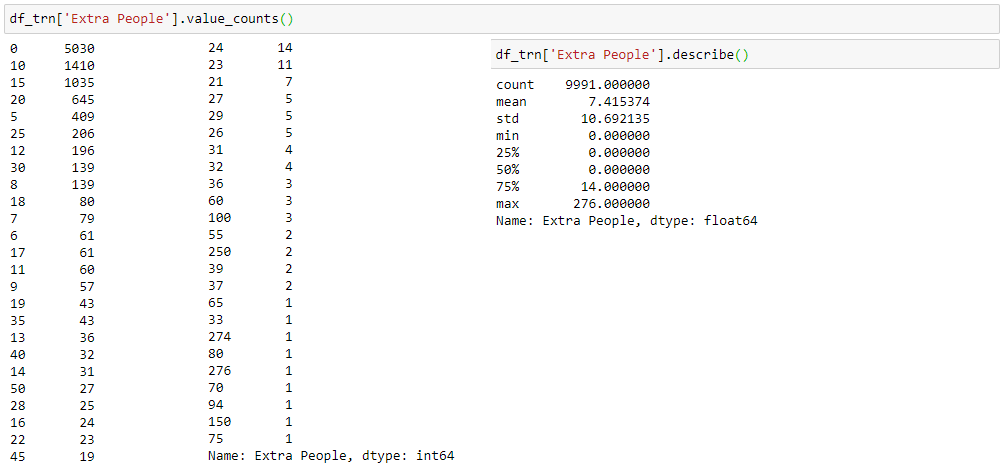
Decidimos adjudicar el valor 0.0 a los registros con valor nulo, aun teniendo en cuenta que existe un número bastante alto de ellos, ya que entendemos que si no se ha especificado un recargo por servicio de limpieza es porque no se cobrará. Al mismo tiempo renombramos la columna de forma correcta.

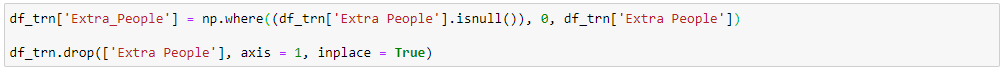
## **Columna Guests Included**

Esta columna no se descartó inicialmente por indicar la cantidad de invitados –por los que se deberá pagar un precio extra– permitidos en cada inmueble, pensando que influirá en el precio el hecho de que esta cantidad sea directamente proporcional al valor que cada propietario da a su inmueble. Esta columna es numérica, por lo que no necesitará categorización.

Sabemos que esta columna no contiene valores nulos en el conjunto de entrenamiento, pero prevendremos que puedan existir en otros conjuntos de datos asimilando sus valores al valor 1 ya que es, con diferencia, el mayoritario del conjunto. Acto seguido procederemos a nombrar la columna de forma adecuada.

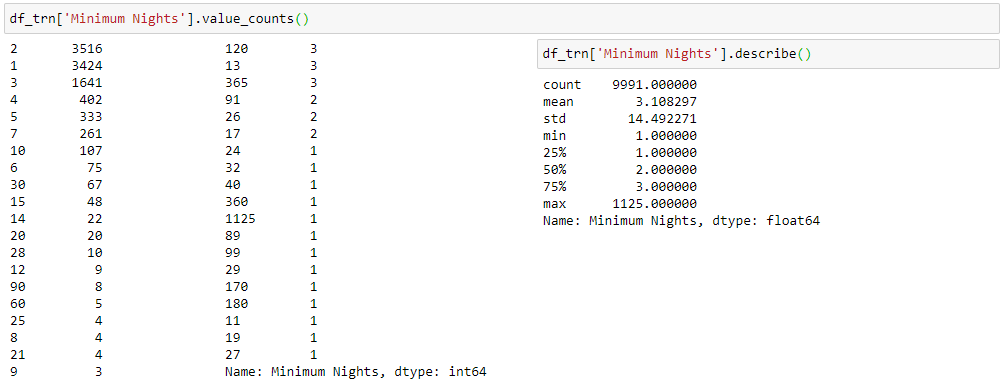
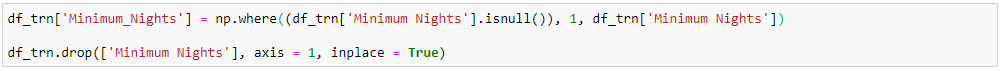
## **Columna Extra People**

Esta columna no se descartó inicialmente por indicar la cantidad en euros especificada como recargo por cada invitado –permitido– en cada inmueble, pensando que influirá en el precio el hecho de que esta cantidad sea directamente proporcional al valor que cada propietario da a su inmueble. Esta columna es numérica, por lo que no necesitará categorización.

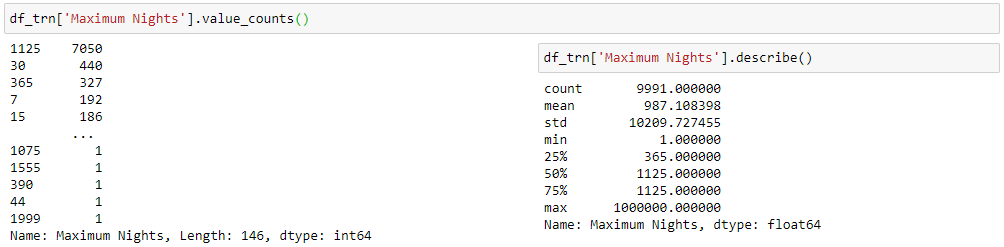
Sabemos que esta columna no contiene valores nulos en el conjunto de entrenamiento, pero prevendremos que puedan existir en otros conjuntos de datos asimilando sus valores al valor 0 ya que es, con diferencia, el mayoritario del conjunto. Acto seguido procederemos a nombrar la columna de forma adecuada.

## **Columna Minimum Nights**

Esta columna no se descartó inicialmente por indicar la mínima cantidad de noches aceptada para poder alquilar cada inmueble, pensando que este valor pueda influir en el precio de alquiler diario. Pensamos que si un propietario, por ejemplo, obliga a alquilar su inmueble como mínimo por una semana el precio por día será algo menor que si obliga a alquilarlo al menos un par de días. Esta columna es numérica, por lo que no necesitará categorización.

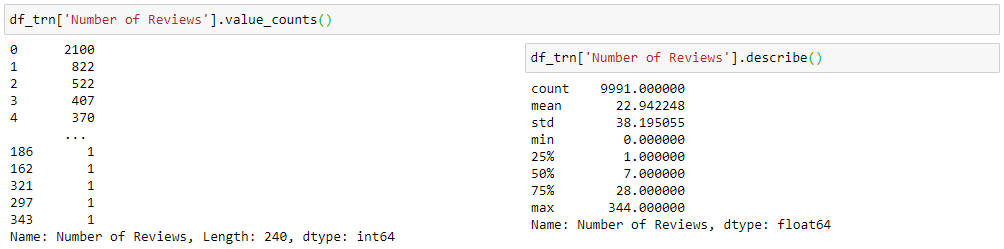
Sabemos que esta columna no contiene valores nulos en el conjunto de entrenamiento, pero prevendremos que puedan existir en otros conjuntos de datos asimilando sus valores al valor 1 ya que como mínimo habrá que alquilar un inmueble por un día. Acto seguido procederemos a nombrar la columna de forma adecuada.

## **Columna Maximum Nights**

Esta columna no se descartó inicialmente por indicar la máxima cantidad de noches aceptada para poder alquilar cada inmueble, pensando que este valor pueda influir en el precio de alquiler diario de cada inmueble. Pensamos que los inmuebles con valores más bajos en esta columna podrían estar considerados por sus propietarios como más exclusivos y que por ende su precio pudiera encarecerse. Esta columna es numérica, por lo que no necesitará categorización.

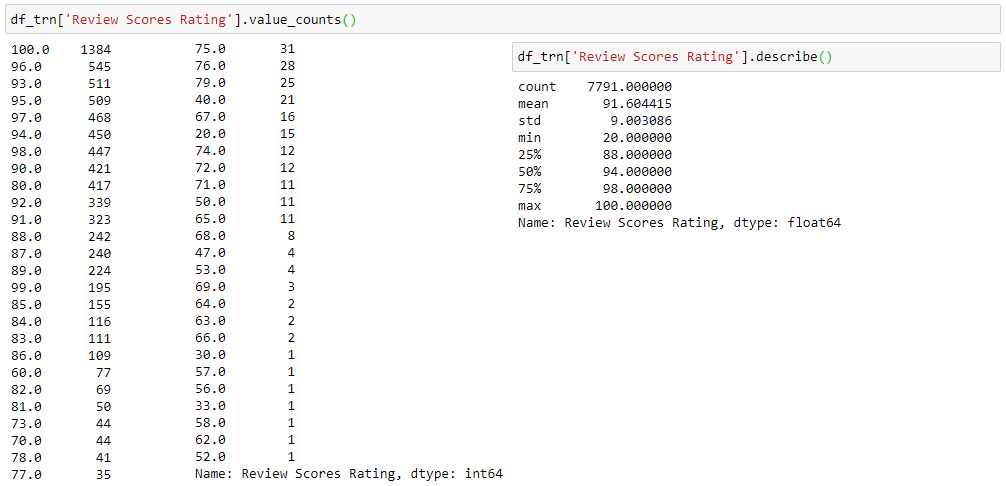
Sabemos que esta columna no contiene valores nulos en el conjunto de entrenamiento, pero prevendremos que puedan existir en otros conjuntos de datos asimilando sus valores al valor 1125 ya que es, con diferencia, el mayoritario del conjunto. Acto seguido procederemos a nombrar la columna de forma adecuada.

## **Columna Number of Reviews**

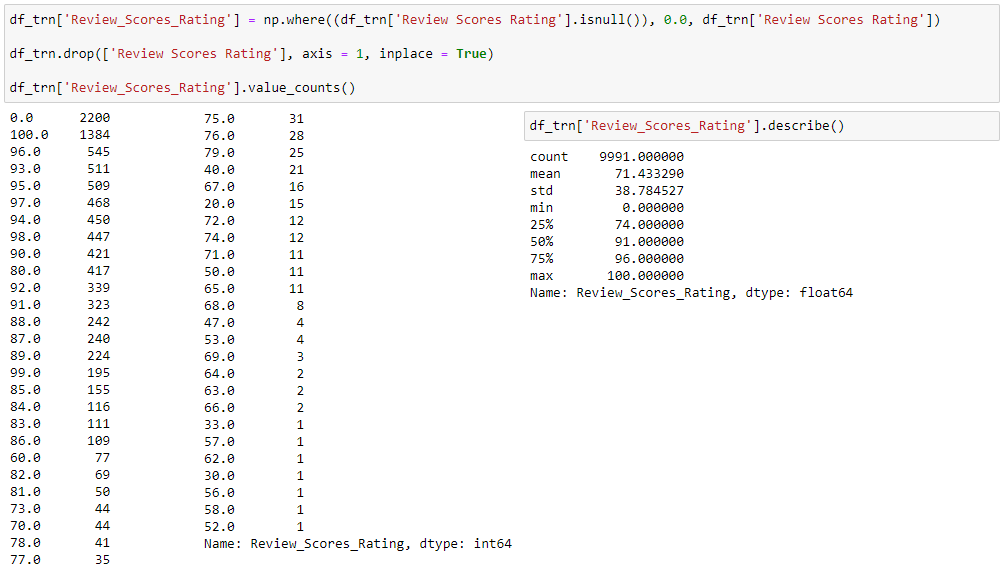
Esta columna no se descartó inicialmente por indicar el número de comentarios recibidos –de parte de sus clientes– por cada inmueble, pensando que cuantos más tuviera, mayor número de clientes se habrían involucrado comentando –para bien o para mal– su experiencia en él y que esto pudiera influenciar su popularidad y su precio de su alquiler. Esta columna es numérica, por lo que no necesitará categorización.

Sabemos que esta columna no contiene valores nulos en el conjunto de entrenamiento, pero prevendremos que puedan existir en otros conjuntos de datos asimilando sus valores al valor 0 ya que es, con diferencia, el mayoritario del conjunto. Acto seguido procederemos a nombrar la columna de forma adecuada.

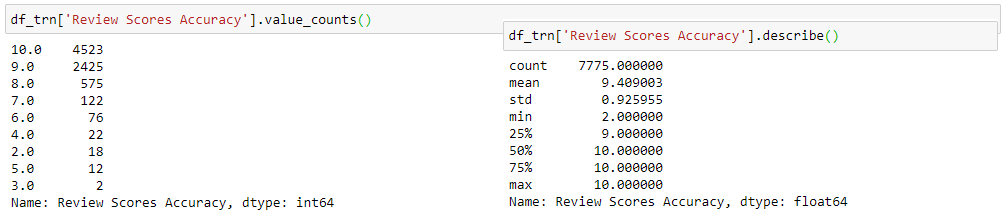
## **Columna Review Scores Rating**

Esta columna no se descartó inicialmente por indicar el porcentaje de comentarios recibidos por cada inmueble, pensando que cuanto más alto fuera, mayor número de clientes se habrían involucrado comentando –para bien o para mal– su experiencia en él y que esto pudiera influenciar su popularidad y su precio de alquiler. Esta columna es numérica, por lo que no necesitará categorización.

Sabemos que esta columna contiene valores nulos. Confirmamos su número a continuación.

Decidimos adjudicar el valor 0.0 a los registros con valor nulo, ya que entendemos que este sería el valor que debería aparecer. Al mismo tiempo renombramos la columna de forma correcta.

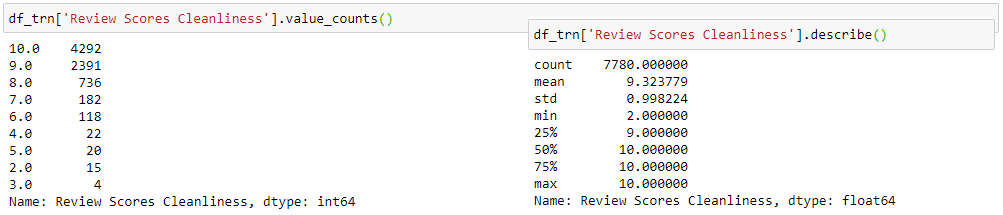
## **Columna Review Scores Accuracy**

Esta columna no se descartó inicialmente por indicar la puntuación recibida por un inmueble respecto a la veracidad de la información que se aporta sobre él, pensando que cuanto más alta fuera, más fiabilidad tendría el inmueble y que esto pudiera influenciar su popularidad y su precio de alquiler. Esta columna es numérica, por lo que no necesitará categorización.

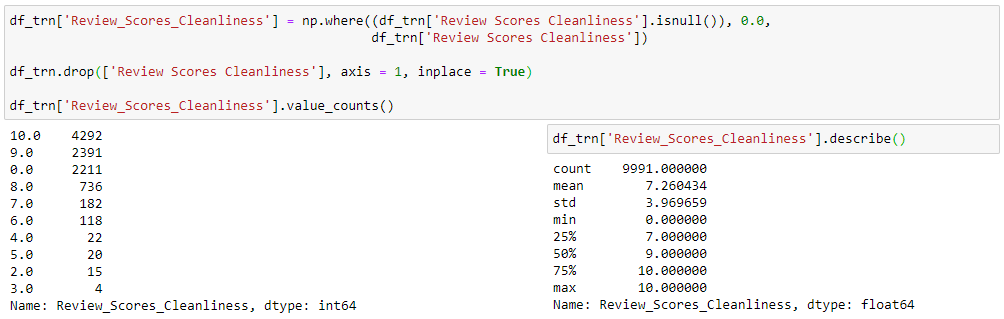
Sabemos que esta columna contiene valores nulos. Confirmamos su número a continuación.

Decidimos adjudicar el valor 0.0 a los registros con valor nulo, ya que entendemos que éste sería el valor que debería aparecer. Al mismo tiempo renombramos la columna de forma correcta.

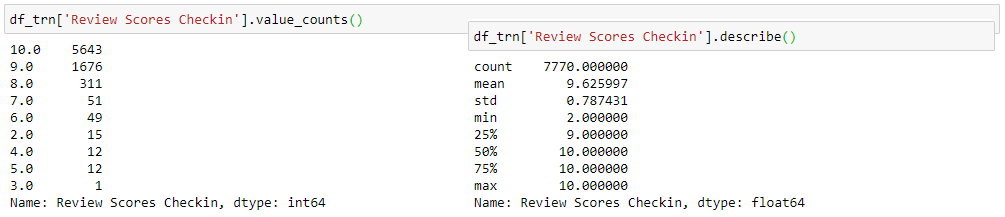
## **Columna Review Scores Cleanliness**

Esta columna no se descartó inicialmente por indicar la puntuación recibida por un inmueble respecto a su limpieza, pensando que cuanto más alta fuera, mejor prestigio tendría el inmueble y que esto pudiera influenciar su popularidad y su precio de alquiler. Esta columna es numérica, por lo que no necesitará categorización.

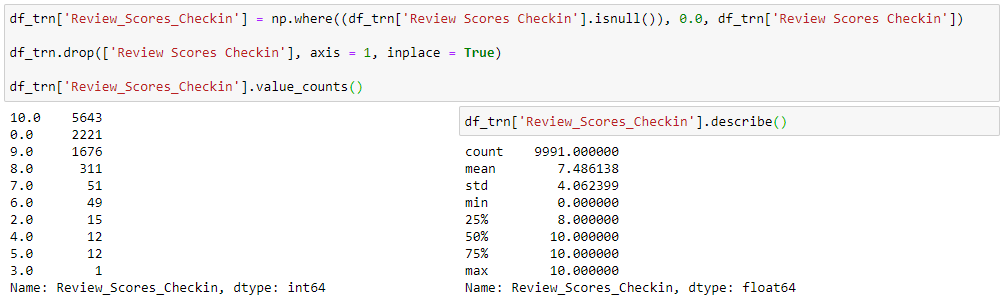
Sabemos que esta columna contiene valores nulos. Confirmamos su número a continuación.

Decidimos adjudicar el valor 0.0 a los registros con valor nulo, ya que entendemos que éste sería el valor que debería aparecer. Al mismo tiempo renombramos la columna de forma correcta.

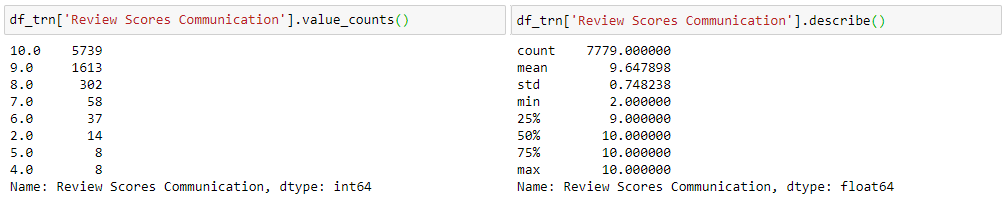
## **Columna Review Scores Checkin**

Esta columna no se descartó inicialmente por indicar la puntuación recibida por un inmueble respecto a la recepción que hace el propietario anfitrión al cliente, pensando que cuanto más alta fuera, mejor prestigio tendría el inmueble y que esto pudiera influenciar su popularidad y su precio de su alquiler. Esta columna es numérica, por lo que no necesitará categorización.

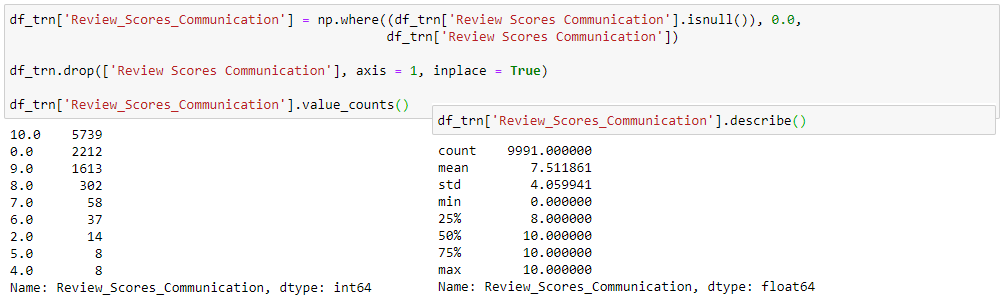
Sabemos que esta columna contiene valores nulos. Confirmamos su número a continuación.

Decidimos adjudicar el valor 0.0 a los registros con valor nulo, ya que entendemos que éste sería el valor que debería aparecer. Al mismo tiempo renombramos la columna de forma correcta.

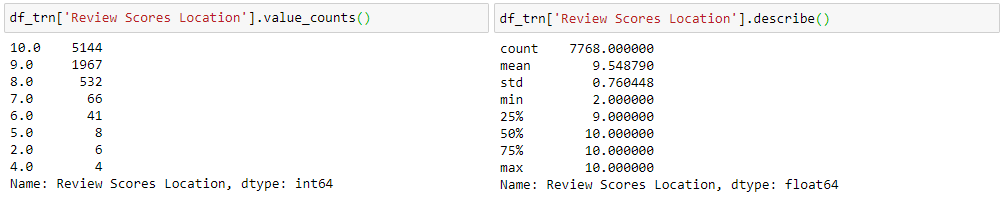
## **Columna Review Scores Communication**

Esta columna no se descartó inicialmente por indicar la puntuación recibida por un inmueble respecto a la comunicación del cliente con el propietario anfitrión, pensando que cuanto más alta fuera, mejor prestigio tendría el inmueble y que esto pudiera influenciar su popularidad y su precio de su alquiler. Esta columna es numérica, por lo que no necesitará categorización.

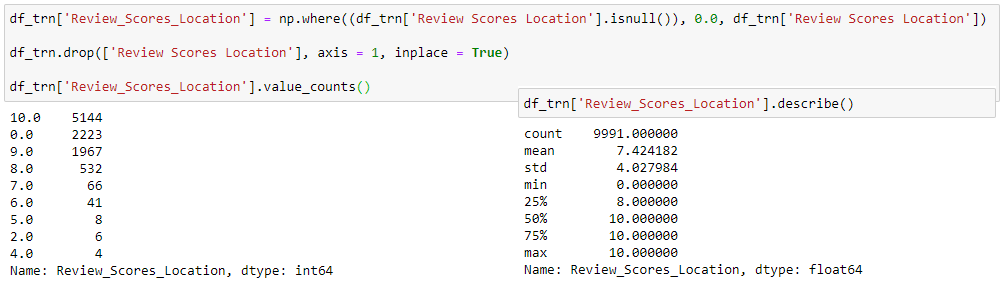
Sabemos que esta columna contiene valores nulos. Confirmamos su número a continuación.

Decidimos adjudicar el valor 0.0 a los registros con valor nulo, ya que entendemos que este sería el valor que debería aparecer. Al mismo tiempo renombramos la columna de forma correcta.

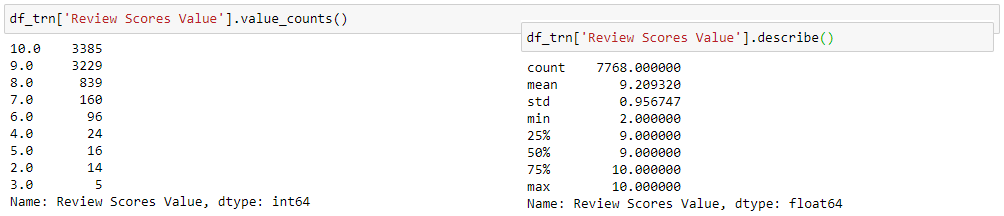
## **Columna Review Scores Location**

Esta columna no se descartó inicialmente por indicar la puntuación recibida por un inmueble respecto de su ubicación, pensando que cuanto más alta fuera, mejor prestigio tendría el inmueble y que esto pudiera influenciar su popularidad y su precio de su alquiler. Esta columna es numérica, por lo que no necesitará categorización.

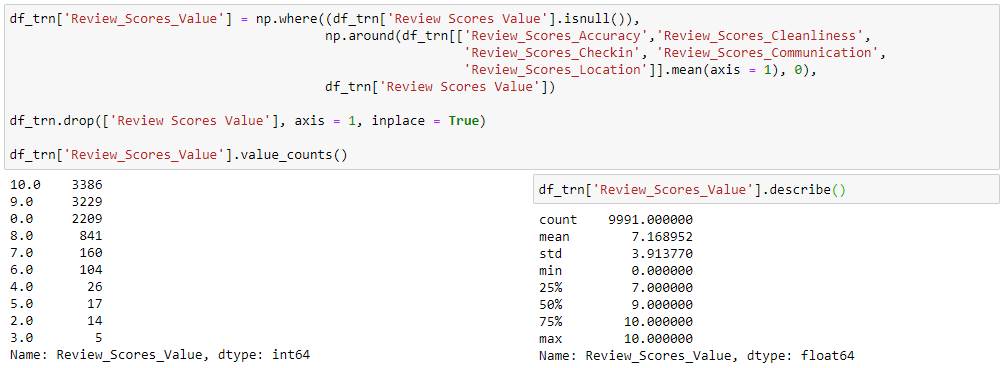
Sabemos que esta columna contiene valores nulos. Confirmamos su número a continuación.

Decidimos adjudicar el valor 0.0 a los registros con valor nulo, ya que entendemos que este sería el valor que debería aparecer. Al mismo tiempo renombramos la columna de forma correcta.

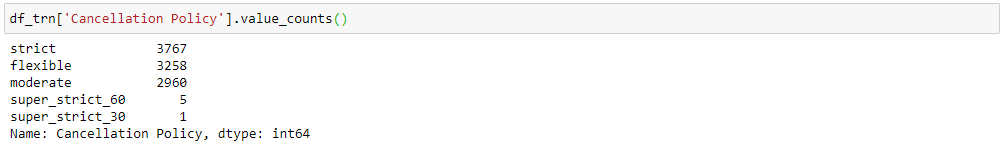
## **Columna Review Scores Value**

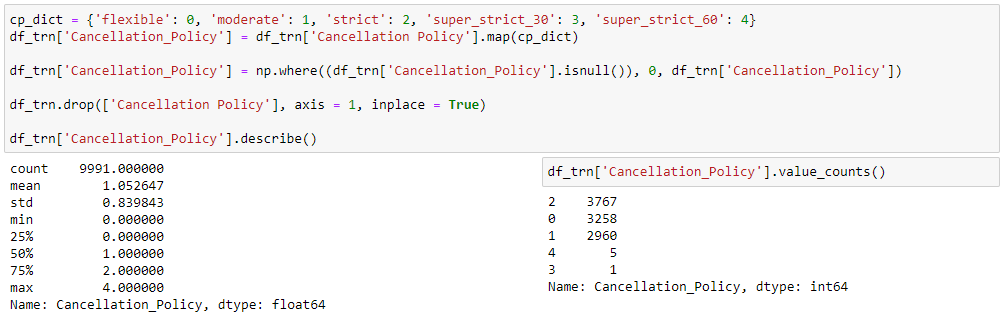
Esta columna no se descartó inicialmente por indicar la puntuación media total recibida por un inmueble, pensando que cuanto más alta fuera, mejor prestigio tendría el inmueble y que esto pudiera influenciar su popularidad y su precio de su alquiler. Esta columna es numérica, por lo que no necesitará categorización.

Sabemos que esta columna contiene valores nulos. Confirmamos su número a continuación.

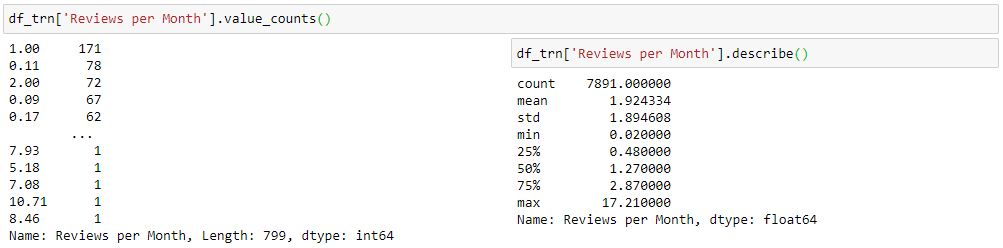
Es lógico pensar que el número de nulos en esta columna coincida con el máximo del resto de columnas de puntuación: 2.223, ya que esta columna es un promedio de ellas. Decidimos que, ahora que ya no existen nulos en dichas columnas podemos calcular la media de sus valores para completar esta, entendiendo que este sería el valor que debería aparecer. Al mismo tiempo renombramos la columna.

## **Columna Cancellation Policy**

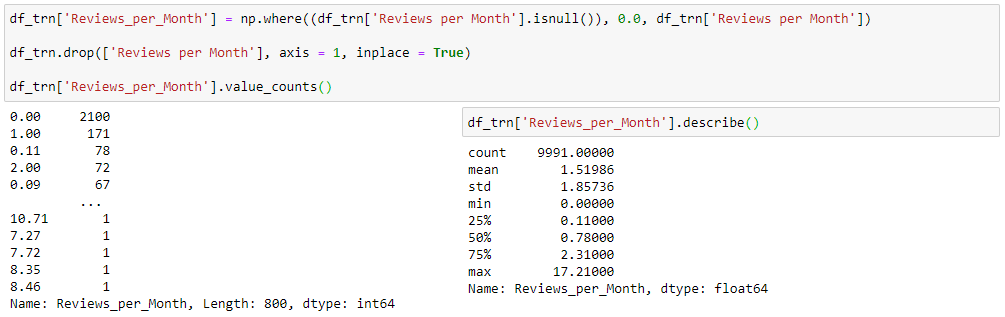
Esta columna no se descartó inicialmente por indicar el tipo de política de cancelación del alquiler de cada inmueble, pensando que cuanto más estricta fuera, podría significar un mejor prestigio del inmueble y que esto pudiera influenciar su popularidad y su precio de alquiler.

Sabemos que esta columna no contiene valores nulos en el conjunto de entrenamiento, pero prevendremos que puedan existir en otros conjuntos de datos asimilando sus valores al valor flexible ya que, si no se ha especificado, entendemos que la política de cancelación debe ser la menos estricta. Acto seguido procederemos a categorizarla –y a renombrarla de forma adecuada– asignando pesos con mayor valor a las políticas de cancelación más estrictas.

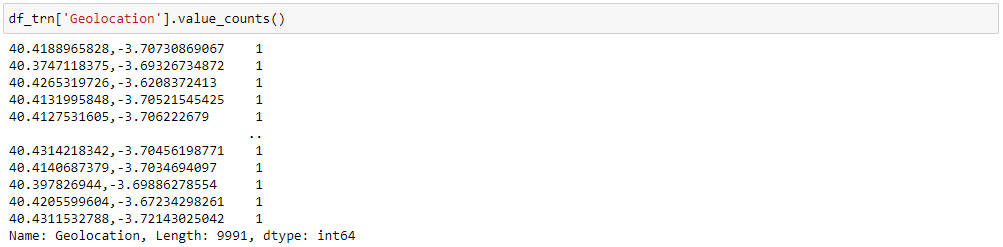
## **Columna** **Reviews per Month**

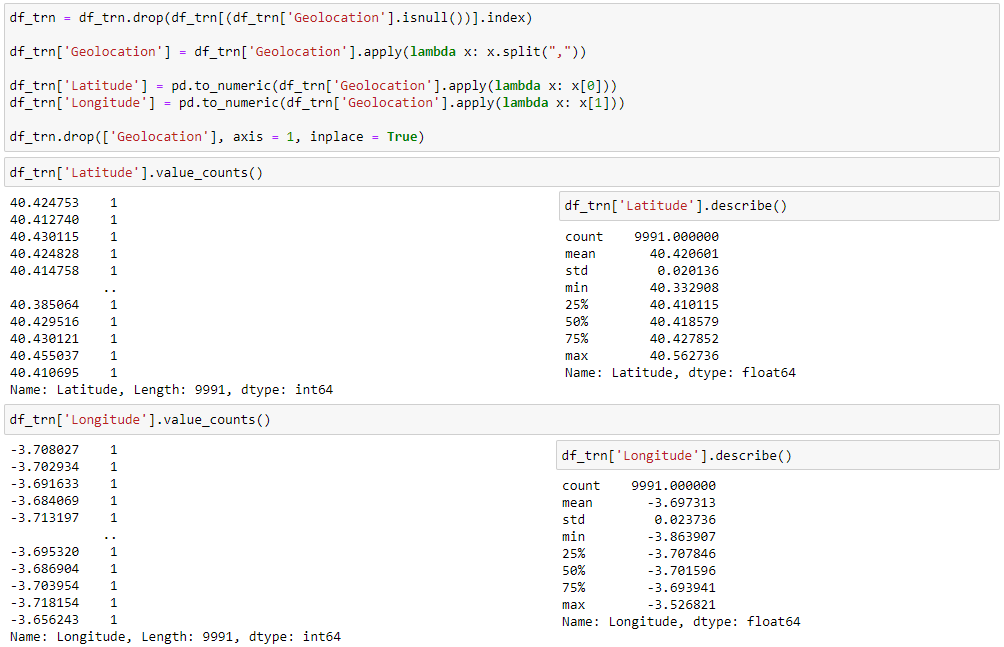
Esta columna no se descartó inicialmente por indicar el promedio mensual del número de comentarios recibidos por un inmueble, pensando que cuanto más alta fuera, más alquileres habría tenido y mejor prestigio tendría el inmueble y que esto pudiera influenciar su popularidad y su precio de su alquiler. Esta columna es numérica, por lo que no necesitará categorización.

Sabemos que esta columna contiene valores nulos. Confirmamos su número a continuación.

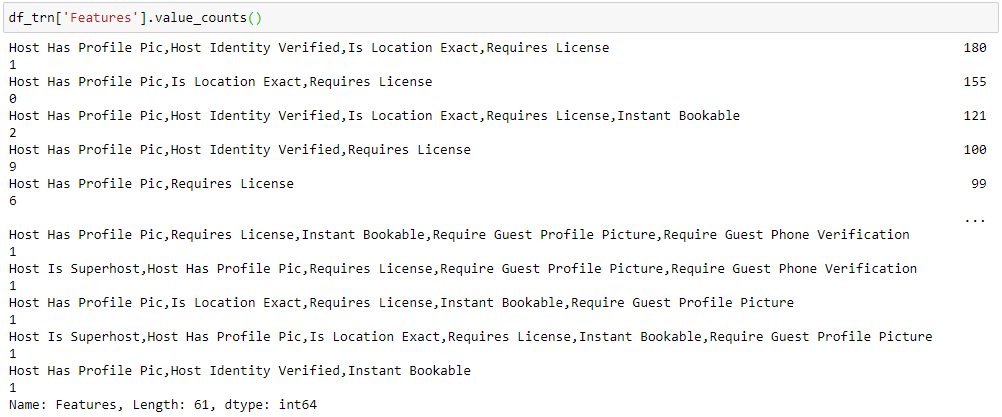
Decidimos adjudicar el valor 0.0 a los registros con valor nulo, ya que entendemos que este sería el valor que debería aparecer. Al mismo tiempo renombramos la columna.

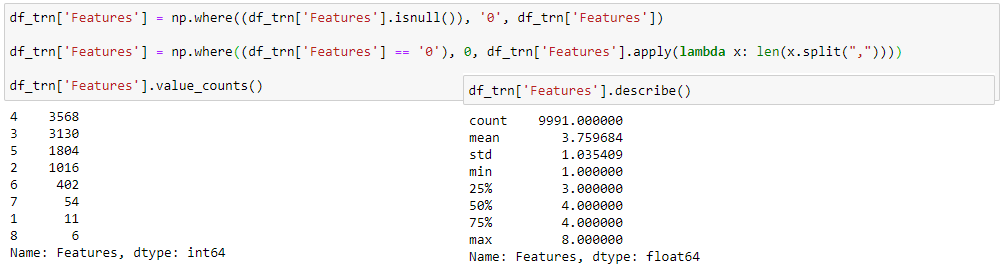
## **Columna Geolocation**

Esta columna no se descartó inicialmente por indicar las coordenadas de geolocalización de cada inmueble en un mejor formato que las columnas individuales **Latitude** y **Longitude** que sí se eliminaron al principio.

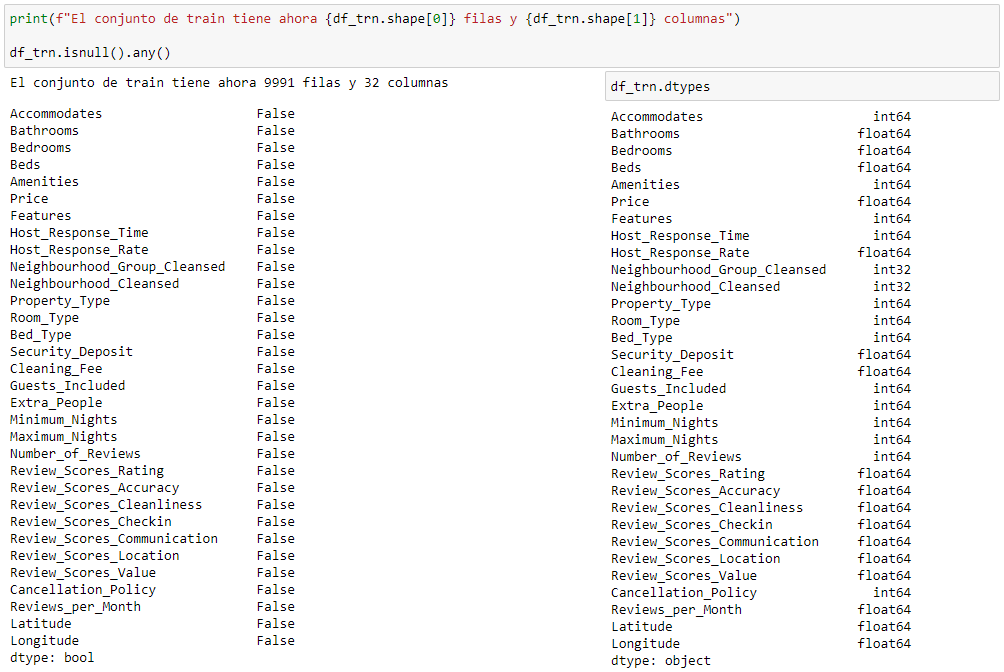
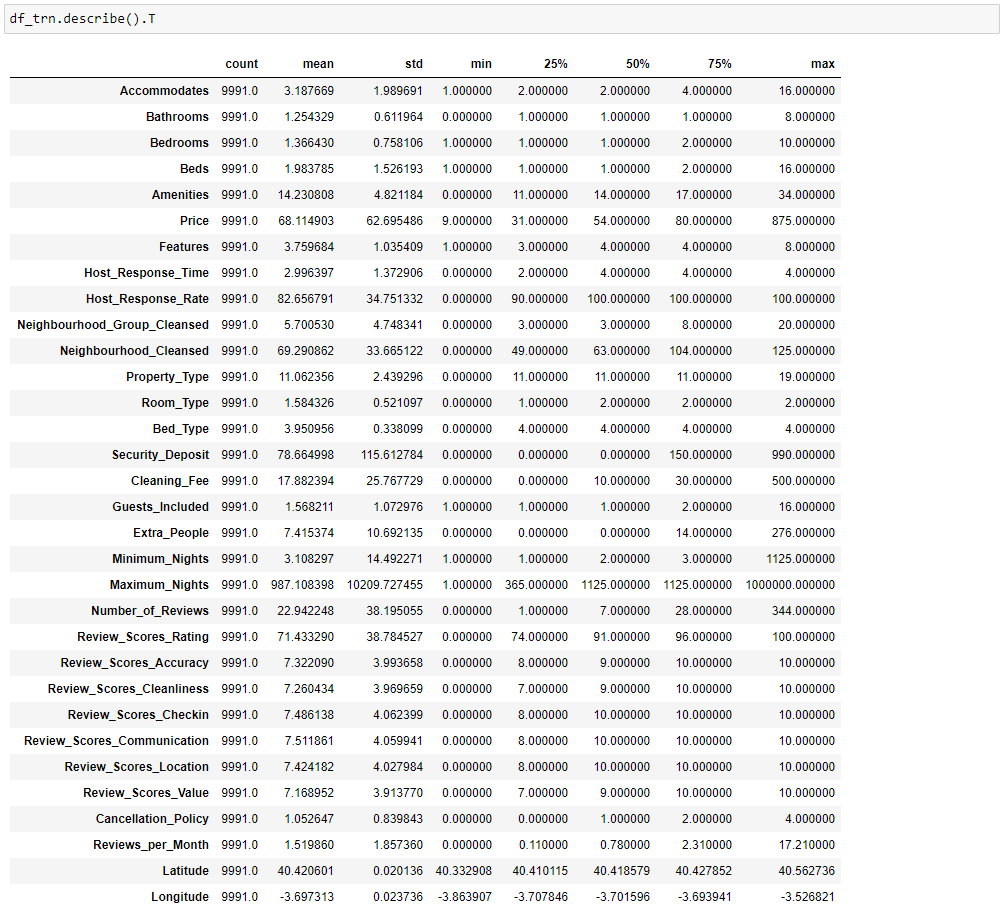
Sabemos que esta columna no contiene valores nulos en el conjunto de entrenamiento, pero prevendremos que puedan existir en otros conjuntos de datos eliminando sus registros. Esta columna contiene tantos valores únicos como inmuebles, en sí misma sobreajusta. La dividimos replicando **Latitude** y **Longitude**, que serán numéricas y no necesitarán categorización.

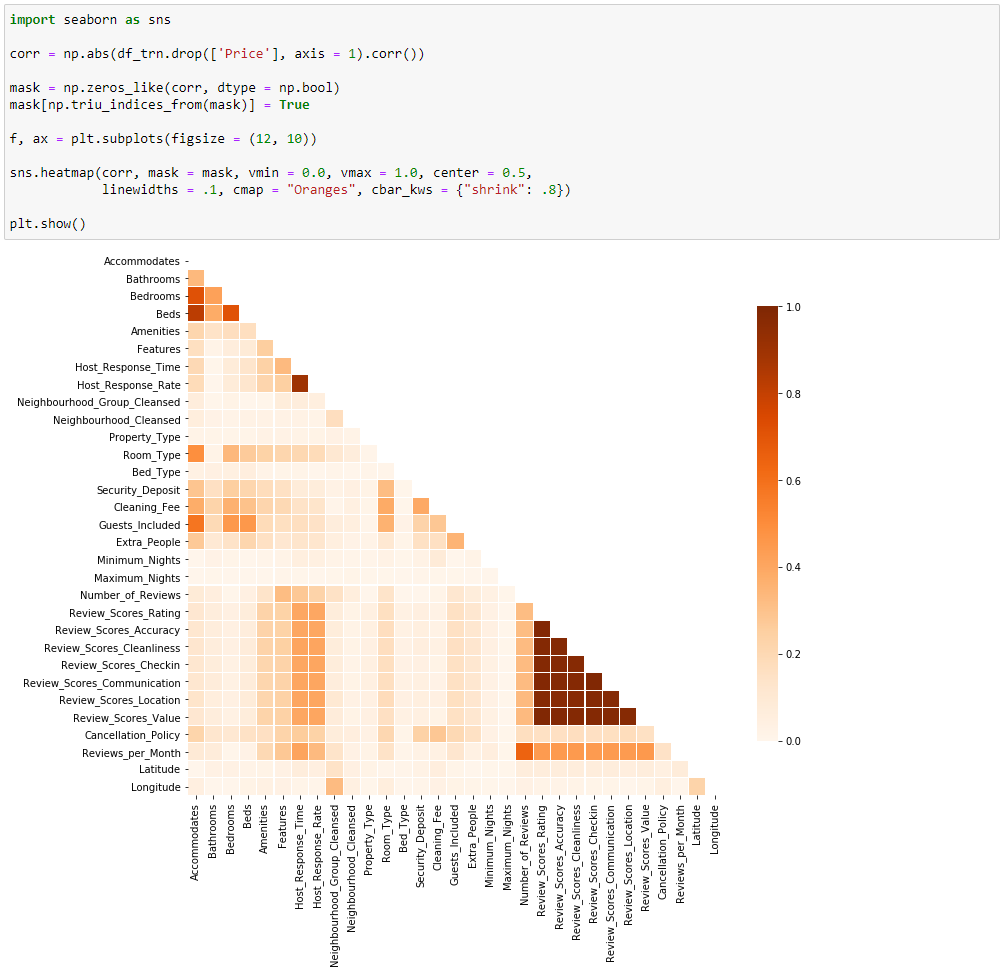
## **Columna Features**

Esta columna no se descartó inicialmente por enumerar características tanto del inmueble como de su propietario, pensando que su número podría influir en el precio de alquiler del inmueble.

Esta columna contiene texto libre y se nos ocurre que podríamos categorizarla sumando las características que contenga, lo que le daría un peso directamente proporcional a su número.

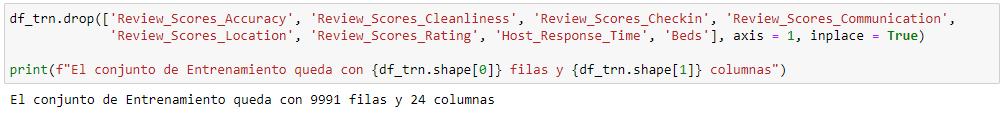
# **Análisis Exploratorio del Conjunto de Entrenamiento**

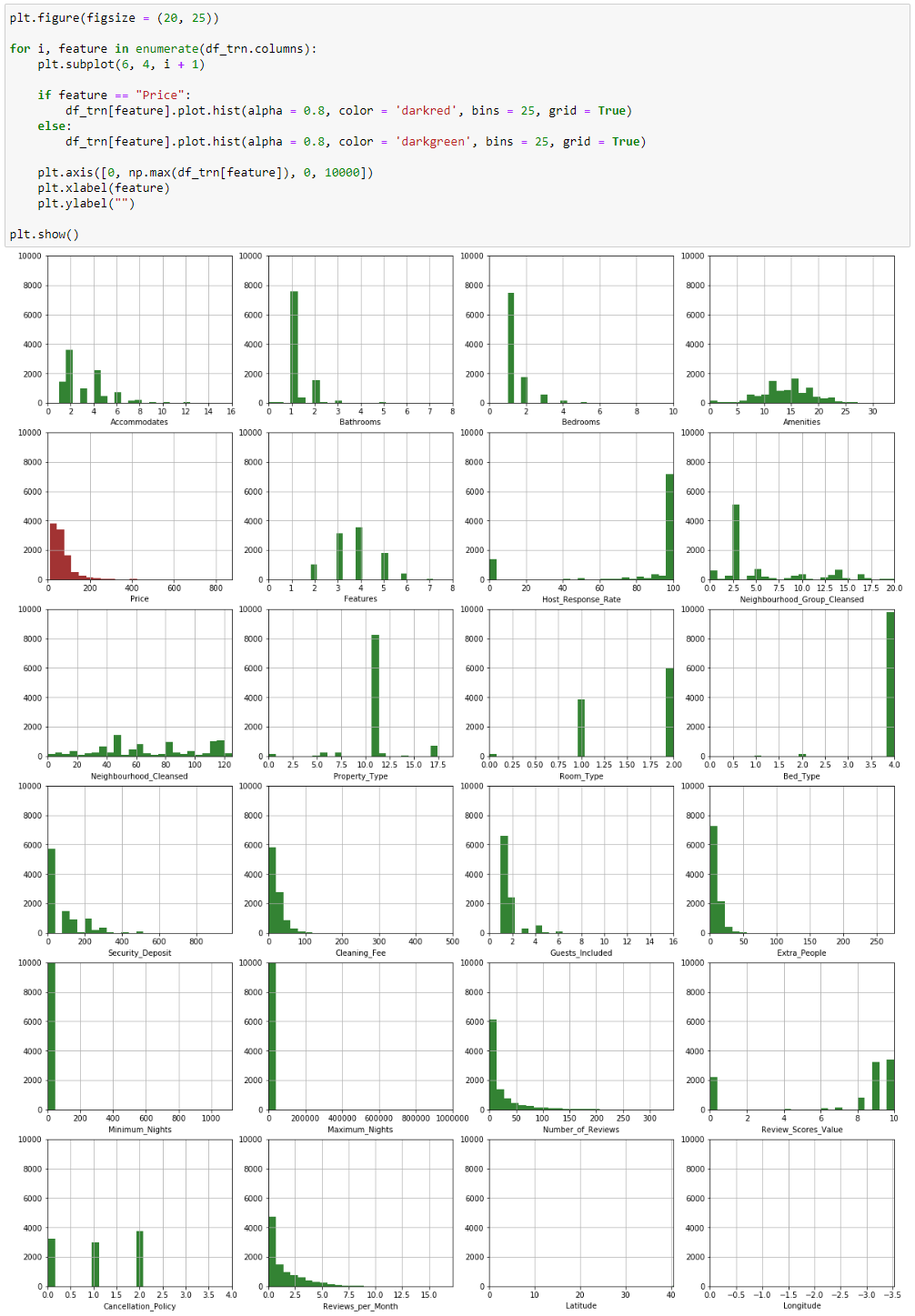
Una vez se han analizado todas las columnas seleccionadas inicialmente de forma individual y se ha procedido a su limpieza de nulos y/o categorización si ha sido necesario, veamos que aspecto tiene nuestro conjunto de entrenamiento.

Tenemos un conjunto de entrenamiento de 9.991 filas y 32 columnas, todas ellas numéricas y sin valores nulos. Veamos un análisis de correlación entre ellas

Observamos una correlación muy alta entre todas las columnas con valores de puntuación de los comentarios sobre cada inmueble. Esto nos sugiere que todas ellas aportarán al modelo la misma significación estadística. Siendo así, nuestra decisión es elegir una de ellas, la columna **Review\_Scores\_Value** frente al resto, ya que contiene un promedio de sus puntuaciones.

También observamos una correlación muy alta entre las columnas **Host\_Response\_Rate** y **Host\_Response\_Time**, lo cual tiene sentido porque ambas contienen información sobre cómo responde aceptando/denegando un anfitrión cuando un posible cliente le reserva su inmueble. Elegimos quedarnos con la columna **Host\_Response\_Rate** por ser un porcentaje frente a la categorización de pesos de la columna **Host\_Response\_Time**.

Por último, también observamos una correlación bastante alta entre las columnas **Beds** y **Accommodates**; con todo el sentido del mundo ya que es lógico pensar que, si un inmueble tiene un cupo de personas determinado, cuente con un número de camas afín. Elegimos quedarnos con la variable **Accommodates** por indicar el aforo incluido en el precio de alquiler de cada inmueble.

Veamos los histogramas de nuestras 24 columnas.

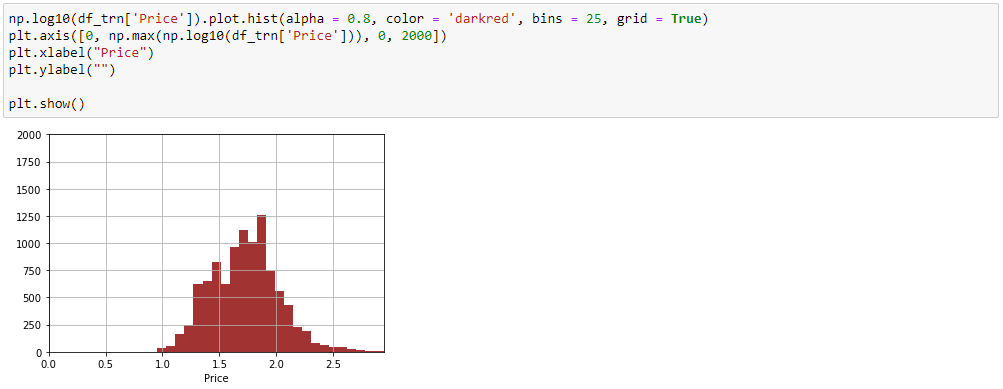
Observamos distribuciones totalmente dispares a la forma normal, salvo quizá en la variable **Amenities**, aunque esos picos nos hacen disentir de que realmente sea una gaussiana.

Lo primero que nos llama la atención es que las columnas **Latitude** y **Longitude** no presentan un histograma apreciable, lo cual pudiera explicarse por la práctica unicidad de sus valores.

También podemos ver que, variables como **Property\_Type**, **Bed\_Type**, **Minimum\_Nights** y **Maximum\_Nights**, tienen un valor único mayoritario. Esto pudiera derivar en no aportarnos significación estadística y en consecuencia las eliminaríamos del modelo.

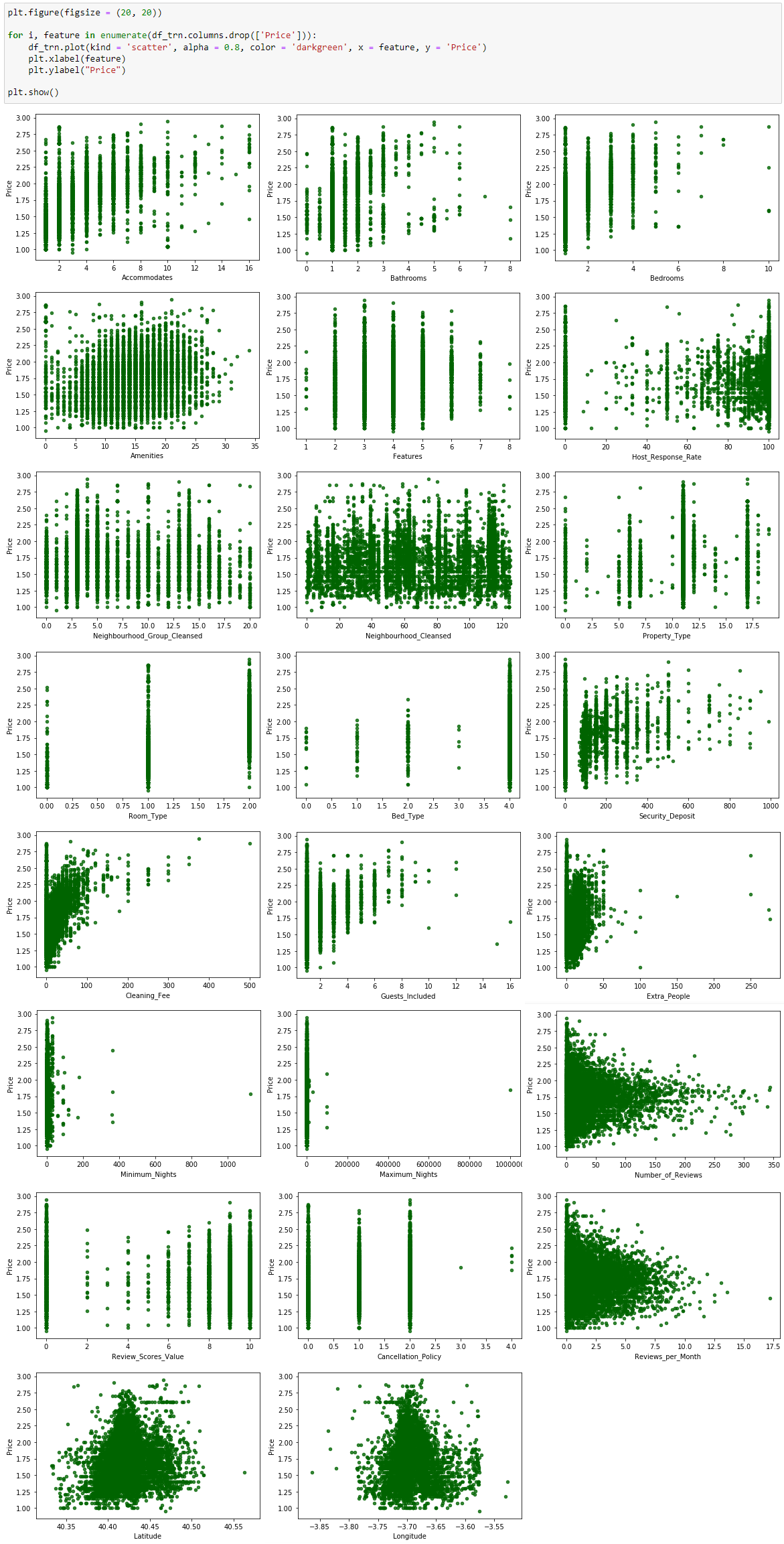
En principio estas gráficas no parecen indicarnos mucho más, pero junto al resultado estadístico básico, en el que podemos ver que tenemos 7 variables –**Security\_Deposit**, **Cleaning\_Fee**, **Extra\_People**, **Minimum\_Nights**, **Maximum\_Nights**, **Number\_of\_Reviews** y **Reviews\_per\_Month**– en las que su desviación típica supera a la media, pudieran estar mostrándonos la posible presencia de outliers en ellas.

Nos gustaría aplicar una transformación logarítmica a las variables **Price**, **Security\_Deposit**, **Cleaning\_Fee**, **Extra\_People**, **Number\_of\_Reviews** y **Reviews\_per\_Month** ya que vemos sus histogramas escorados hacia la izquierda. Pero solamente podemos hacerlo sobre **Price** ya que el resto tienen registros con valor cero –y la transformación logarítmica no puede aplicarse sobre rangos con valor cero o números negativos–.

Veamos ahora que aspecto tiene el histograma de nuestra variable objetivo.

Pues parece que ha tornado en un aspecto más gaussiano, más centrado, aunque dudamos que la distribución sea normal por los picos de frecuencia entre los valores 1.2 y 1.5.

Ejecutaremos a continuación comando **scatter-plot** para observar las relaciones de cada una de las 23 columnas explicativas frente a nuestro target.

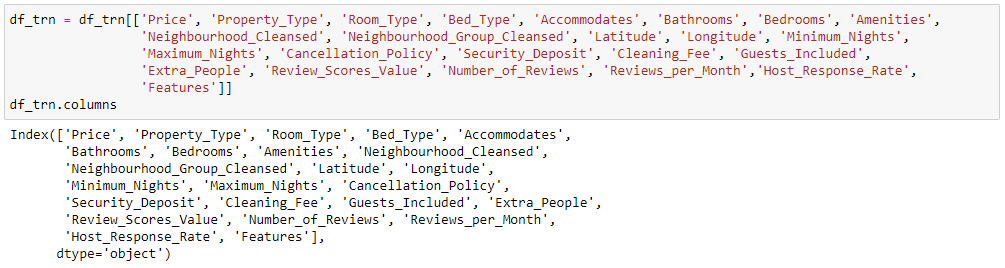


Efectivamente observamos que las dos variables con mayor desviación típica con respecto a su media **Minimum\_Nights** y **Maximum\_Nights** presentan outliers a la derecha de sus gráficas.

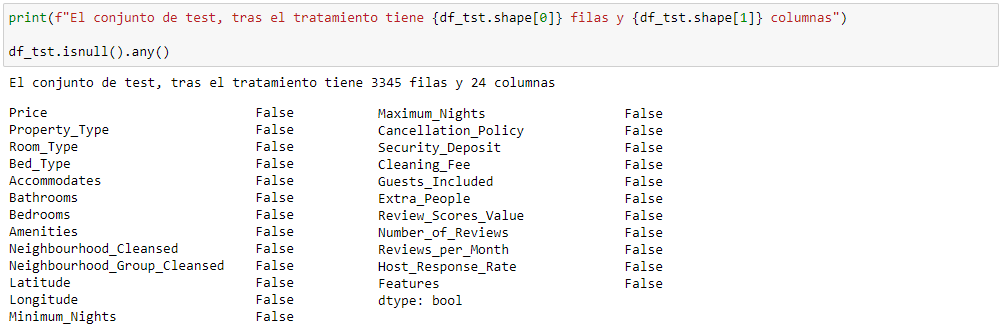
También vemos algún otro posible outlier en **Bathrooms**, **Bedrooms**, **Features**, **Cleaning\_Fee**, **Guests\_Included**, **Extra\_People**, **Cancelation\_Policy**, **Latitude** y **Longitude**.

Por ahora, no vamos a tomar la decisión de eliminar sus registros, puesto que en otras columnas pueden tener valores significativos para nuestro modelo. Antes seguiremos investigando como reducir la dimensionalidad porque 24 nos sigue pareciendo un número bastante elevado de variables explicativas.

# **Proceso del Conjunto de Test**

Reorganizamos el conjunto de entrenamiento, colocando las variables en su mejor acomodo.

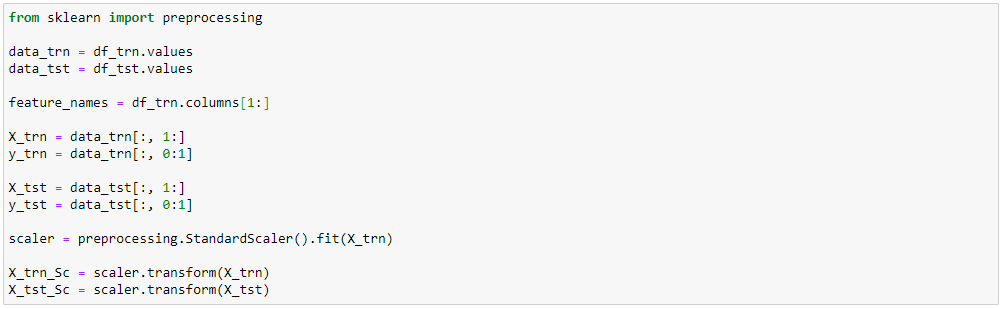
Ahora es el momento de aplicar y ejecutar sobre el conjunto de test todos los tratamientos que hemos ido realizando al conjunto de entrenamiento, para que ambos conjuntos queden alineados.

Observamos que el conjunto de datos de test ha quedado alineado con el de entrenamiento y sin valores nulos en sus variables, lo cual nos confirma la bondad del tratamiento de cleansing aplicado.

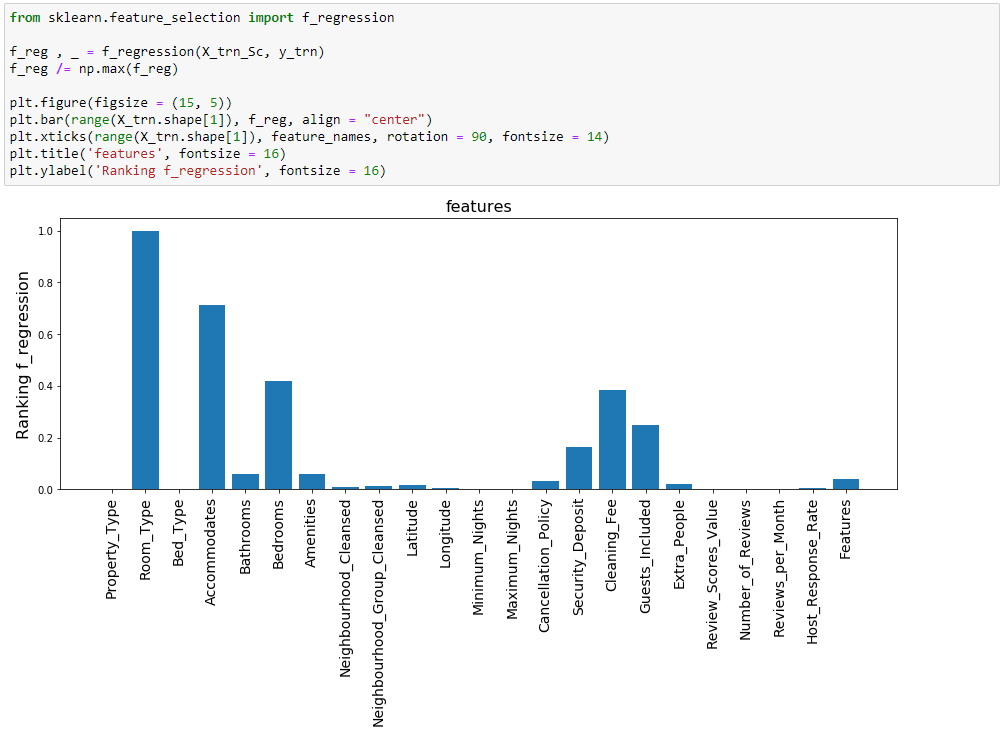
# **Reduciendo la Dimensionalidad**

Como decíamos en el párrafo 7, 24 aun nos parece un número bastante elevado de variables explicativas, por ello, vamos a ejecutar a continuación algunos modelos que nos ofrecen la posibilidad de ver la aportación estadística de las columnas.

A partir de sus ejecuciones tomaremos las decisiones oportunas para desechar las columnas que no aporten significado a nuestro modelo.

Pero antes, normalizaremos nuestros conjuntos de datos utilizando el método **StandardScaler** de la librería **preprocessing** de **scikit-learn**.

## **Modelo F-Regression**

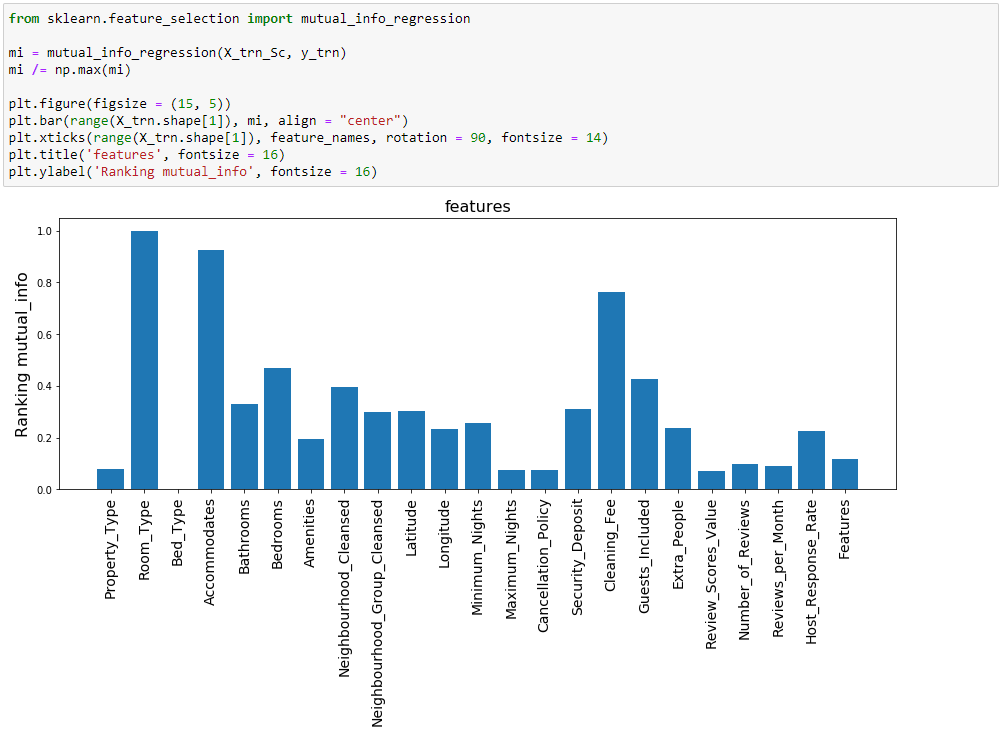
Ejecutaremos primero un modelo **f\_regression** de la librería **feature\_selection** de **scikit-learn**. Este algoritmo no es un modelo predictivo en sí, su utilidad radica en realizar selección de características en conjuntos de datos sobre los que se quiere aplicar un modelo de regresión.

El algoritmo desecha directamente las siguientes variables: **Property\_Type,** **Bed\_Type,** **Maximum\_Nights,** **Minimum\_Nights,** **Review\_Score\_Values,** **Numer\_of\_Reviews** y **Reviews\_per\_Month**.

Podríamos descartar también, dada la poca importancia que parecen tener: **Neighbourhood\_Cleansed**, **Neighbourhood\_Group\_Cleansed**, **Latitude**, **Longitude** y **Extra\_People**.

Esto significaría reducir nuestro número de variables explicativas de 23 a 11, menos de la mitad.

## **Modelo Mutual Info**

Ahora ejecutaremos otro modelo selector de características análogo al anterior, **mutual\_info** de la librería **feature\_selection** de **scikit-learn**.

El algoritmo desecha directamente la columna **Bed\_Type**.

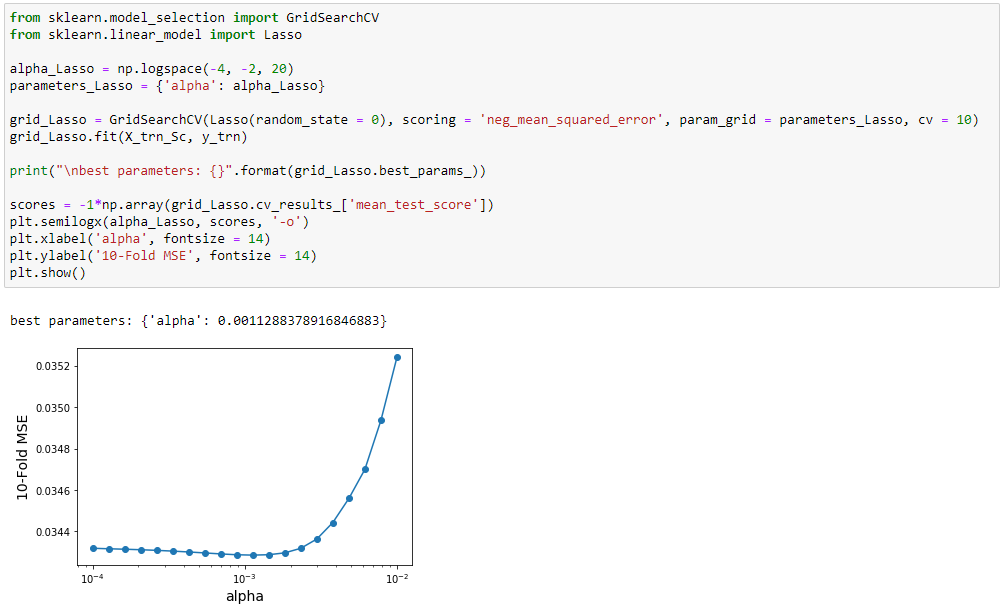
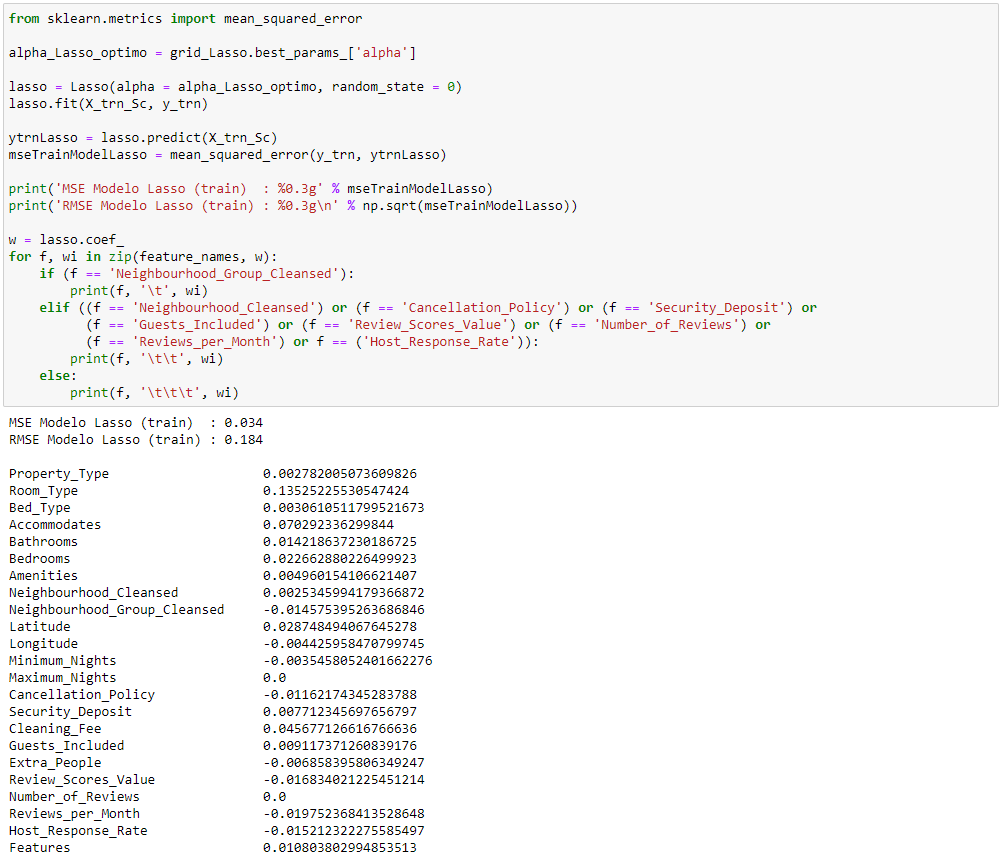
Comparando las alturas de las barras, nos atreveríamos a descartar **Property\_Type,** **Maximum\_Nights,** **Cancellation\_Policy,** **Review\_Score\_Values,** **Numer\_of\_Reviews,** **Reviews\_per\_Month** y **Features**.

Esto significaría reducir nuestro número de variables explicativas de 23 a 15.

## **Modelo Lasso**

Veamos ahora la ejecución de un algoritmo **Lasso** de la librería **linear\_model** de **scikit-learn**, cuya utilidad estriba en que regulariza (reduce) los coeficientes del modelo llegando incluso a anular variables explicativas no –o ínfimamente– relevantes.

Lo primero es buscar el hiper-parámetro (**alpha**) óptimo para que el algoritmo **Lasso** trabaje con nuestros datos. Entrenaremos un algoritmo **GridSearchCV** de la librería **model\_selection** de **scikit-learn**, orientado para**Lasso**.

Una vez tenemos el hiper-parámetro**alpha** óptimo, lo aplicamos sobre el algoritmo **Lasso** trabajando sobre la muestra de entrenamiento.

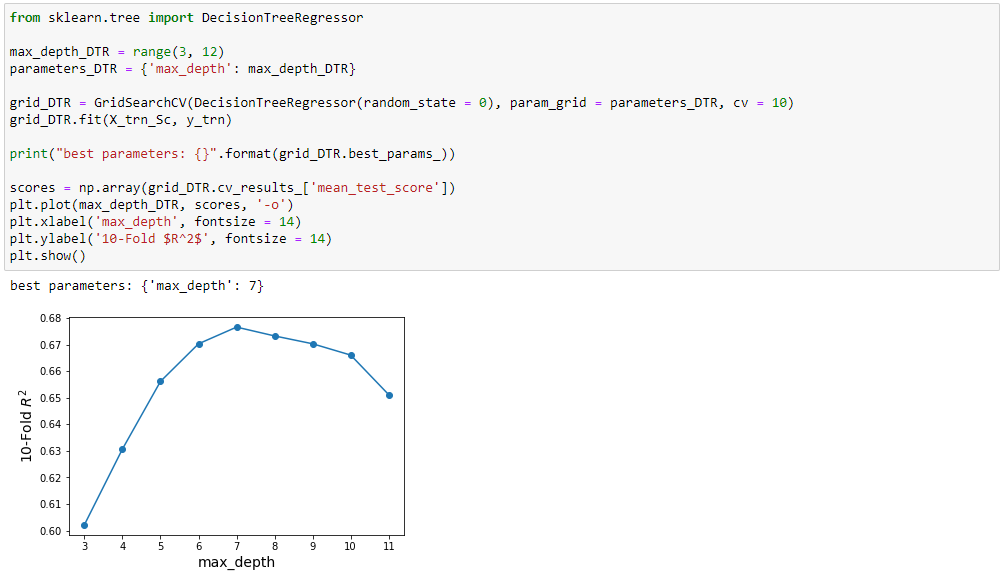
Este algoritmo nos devuelve un error cuadrático medio **MSE** =0.034 y a simple vista vemos que desecha las variables **Maximum\_Nights** y **Number\_of\_Reviews**.

Podríamos descartar las variables con coeficientes menores –valor absoluto– a 0.01, **Property\_Type**, **Bed\_Type**, **Amenities**, **Neighbourhood\_Cleansed**, **Longitude**, **Minimum\_Nights**, **Security\_Deposit**, **Guests\_Included** y **Extra\_People**.

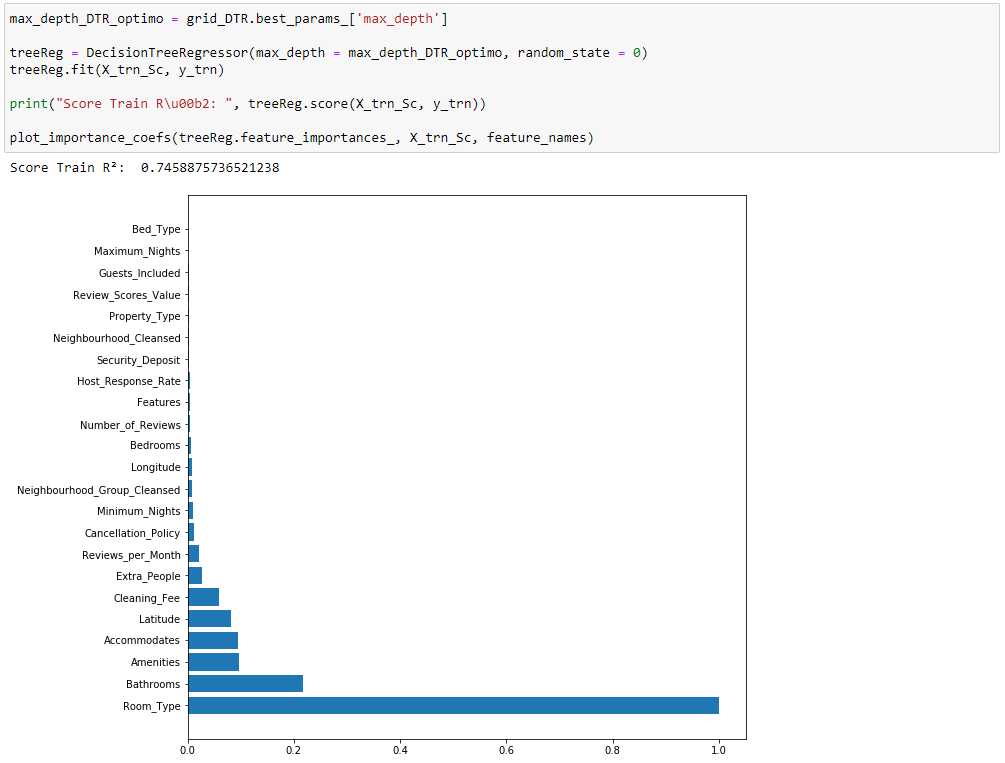
Esto significaría reducir nuestro número de variables explicativas de 23 a 12, casi a la mitad.

## **Modelo Decission Tree**

Ahora ejecutaremos un **DecissionTreeRegressor** de la librería **tree** de **scikit-learn**, cuya utilidad estriba en que aporta coeficientes de relevancia o importancia sobre las variables del dataset, lo cual nos puede dar una idea sobre qué columnas podríamos descartar.

Buscaremos el hiper-parámetro (**max\_depth**) óptimo para que el árbol de decisión trabaje con nuestros datos. Entrenaremos un algoritmo **GridSearchCV**, orientado a**DecissionTreeRegressor**.

Una vez tenemos el hiper-parámetro**max\_depth** óptimo, lo aplicamos sobre el algoritmo **DecissionTreeRegressor** trabajando sobre la muestra de entrenamiento.

Para visualizar gráficamente los coeficientes de importancia utilizaremos una función –ad-hoc– plot\_importance\_coefs.

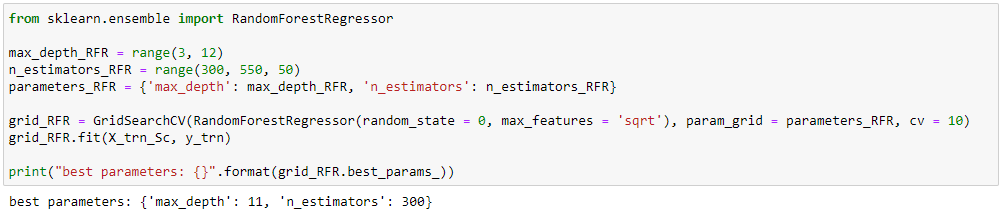
Este algoritmo nos devuelve un **R2** =0.74589 (bondad de ajuste o coeficiente de determinación predictora) y a simple vista vemos que considera de nula importancia las variables **Bed\_Type**, **Maximum\_Nights**, **Guests\_Included**, **Review\_Scores\_Value**, **Property\_Type**, **Neighbourhood\_Cleansed** y **Security\_Deposit**.

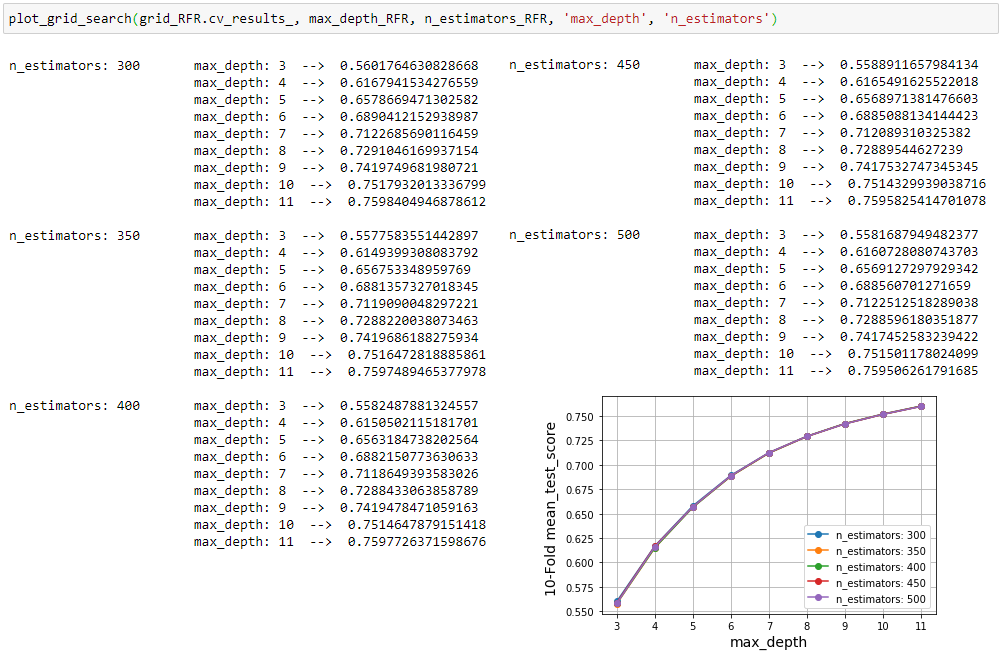
Siendo más estrictos, podríamos considerar prácticamente nada significativas las columnas **Host\_Response\_Rate**, **Features** y **Number\_of\_Reviews**.

Lo anterior significaría reducir nuestro número de variables explicativas de 23 a 13, poco más de la mitad.

## **Modelo Random Forest**

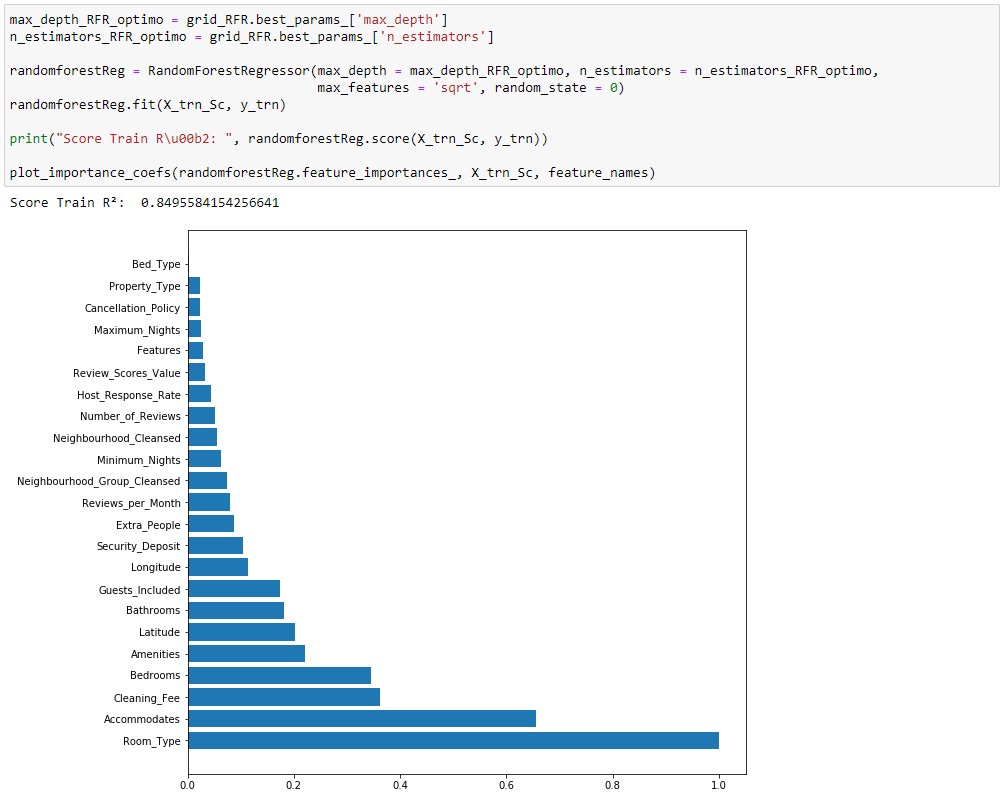
El siguiente modelo a ejecutar será un **RandomForestRegressor** de la librería **ensemble**de **scikit-learn**, que al igual que el árbol de decisión nos aportará coeficientes de relevancia o importancia sobre las variables del modelo.

Buscaremos los hiper-parámetros (**max\_depth** y **n\_estimators**) óptimos –fijando **max\_features** al valor sqrt– para que el bosque aleatorio trabaje con nuestros datos. Entrenaremos un algoritmo **GridSearchCV**, orientado para**RandomForestRegressor**.

Para visualizar gráficamente la comparativa de hiper-parámetros resultante utilizaremos una función –ad-hoc– plot\_grid\_search.

En la gráfica comparativa las curvas se superponen y es complicado apreciar diferencias.

Una vez tenemos los hiper-parámetros**max\_depth** y **n\_estimators** óptimos, los aplicamos sobre el algoritmo **RandomForestRegressor** trabajando sobre la muestra de entrenamiento.

Este algoritmo nos devuelve un **R2** =0.84956 y a simple vista vemos que considera de nula importancia la variable **Bed\_Type**.

A la vista de los coeficientes de la gráfica podríamos considerar poco significativas las columnas **Property\_Type**, **Cancellation\_Policy**, **Maximum\_Nights**, **Features**, **Review\_Scores\_Value**, **Host\_Response\_Rate**, **Minimum\_Nights**, **Number\_of\_Reviews** y **Neighbourhood\_Cleansed**.

Lo que significaría reducir nuestro número de variables explicativas de 23 a 13, poco más de la mitad.

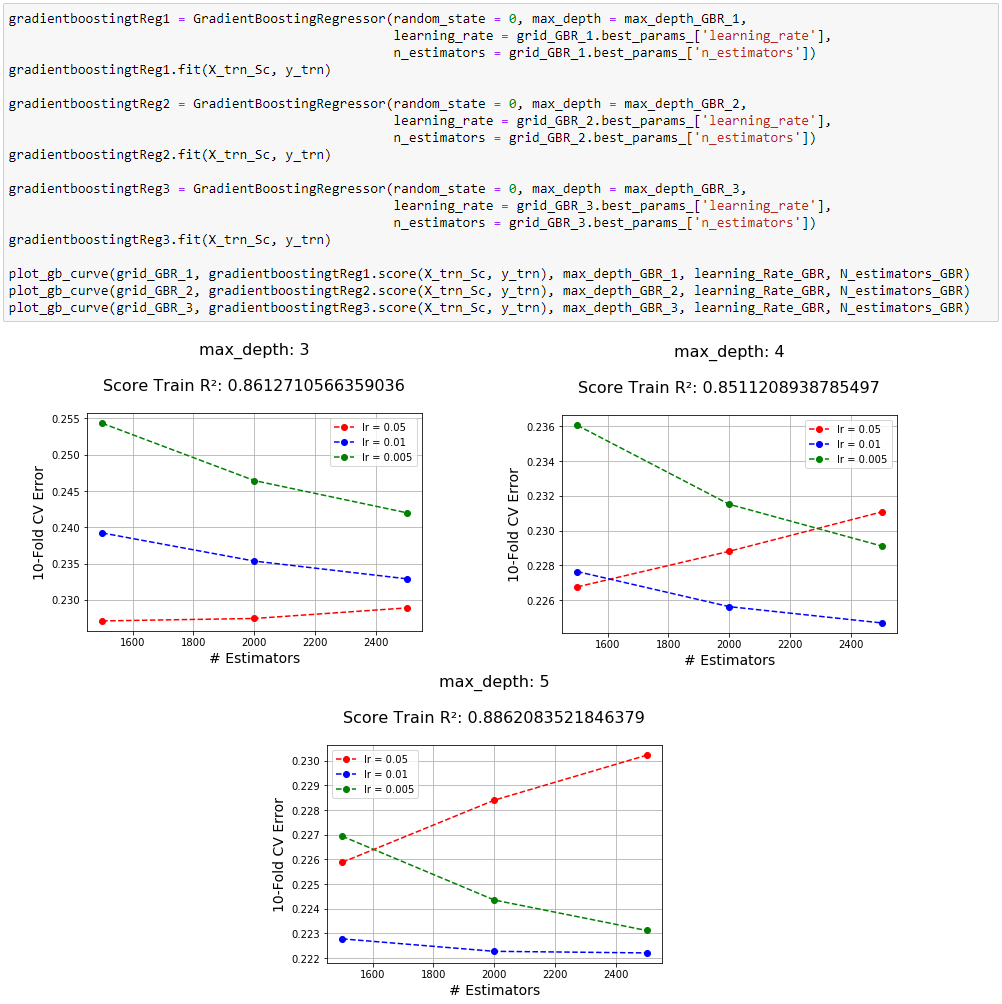
## **Modelo Gradient Boosting**

Ejecutaremos ahora un modelo **GradientBoostingRegressor** de la librería **ensemble**de **scikit-learn**, que al igual que el árbol de decisión y el bosque aleatorio nos aportará coeficientes de relevancia o importancia sobre las variables del modelo.

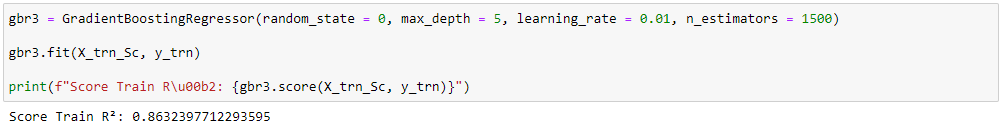
Buscaremos los hiper-parámetros (**max\_depth**, **n\_iterations** y **learning\_rate**) óptimos para que el potenciador de gradiente trabaje con nuestros datos. Entrenaremos un algoritmo **GridSearchCV**, orientado a**GradientBoostingRegressor**.

Este proceso de entrenamiento es muy costoso si se realiza buscando los tres hiper-parámetros al mismo tiempo. Para minorar o reducir su coste se lanza tres veces fijando el hiper-parámetro **max\_depth** con tres valores diferentes.

Como podemos ver, los resultados de los hiper-parámetros fijando **max\_depth** en 4 y en 5 coinciden y en ambos casos son más sobreajustados que fijando **max\_depth** en 3.

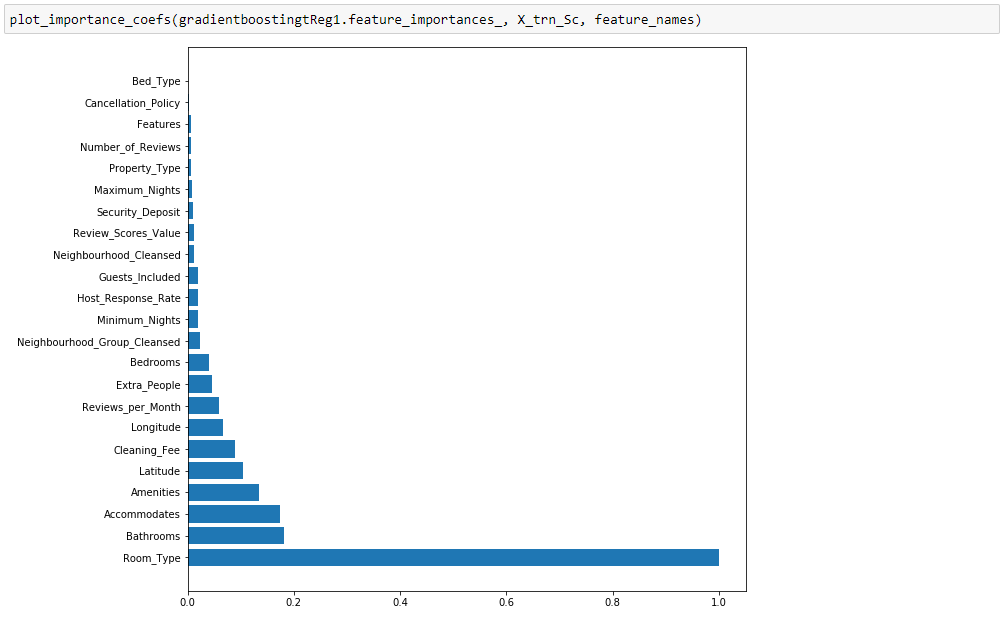
 Para visualizar gráficamente curvas y métricas, utilizaremos una función –ad-hoc– plot\_gb\_curve.

Las mejores bondades de ajuste, como podemos ver se obtienen con el hiper-parámetro **max\_depth** en valores 5 y 3, respectivamente.

Con **max\_depth** = 3, los hiper-parámetros óptimos coinciden con los mínimos de cada rango. Con **max\_depth** = 5, observamos que el número de estimadores, en la curva en la que **learning\_rate** = 0.01, es prácticamente irrelevante. Veamos que bondad de ajuste presenta con el menor valor de ellos.

La bondad de ajuste es 2 milésimas mejor que la del modelo con **max\_depth** = 3, lo cual nos parece inapreciable, por tanto elegimos como mejor modelo el primero de los entrenados, con bondad de ajuste **R2** =0.86127 e hiper-parámetros óptimos: **max\_depth** = 3, **n\_estimators** = 1500 y **learning\_rate** = 0.05.

Veamos entonces sus coeficientes de importancia sobre las variables del modelo.

El modelo considera de nula importancia las variables **Bed\_Type** y **Cancellation\_Policy**.

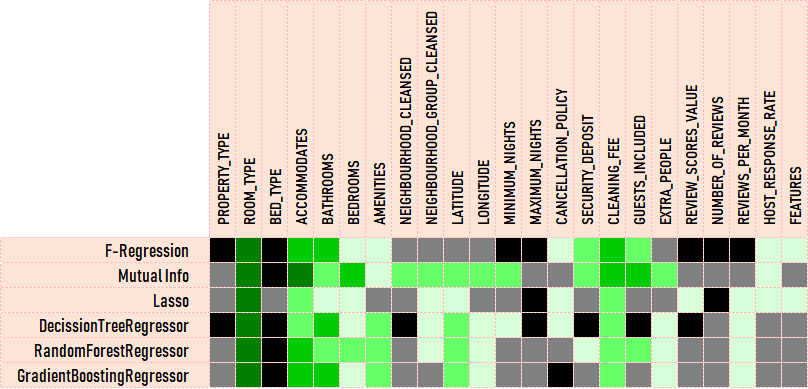
A la vista de los coeficientes de la gráfica podríamos considerar poco significativas las columnas **Features**, **Number\_of\_Reviews**, **Property\_Type**, **Maximum\_Nights**, **Security\_Deposit, Review\_Scores\_Value**, **Guests\_Included**, **Neighbourhood\_Cleansed**, **Host\_Response\_Rate**, **Minimum\_Nights** y **Neighbourhood\_Group\_Cleansed**.

Lo que significaría reducir nuestro número de variables explicativas de 23 a 10, menos de la mitad.

## **Comparativa de Modelos y Reducción de la Dimensionalidad**

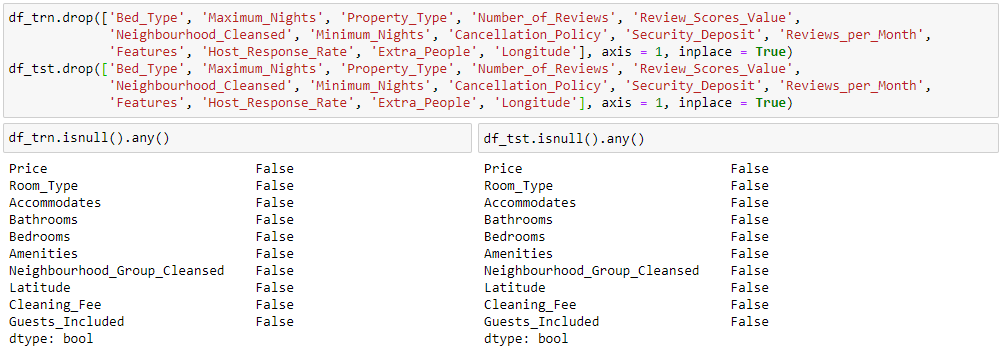
Hemos entrenado, en los párrafos anteriores, seis algoritmos para obtener qué columnas tenían más y menos importancia para cada uno de ellos, con el objetivo de poder descartar algunas de ellas y así paliar o reducir la *maldición de la dimensionalidad* que sufrimos con nuestro dataset.

Representemos un producto cartesiano con el resultado de nuestras observaciones.

En verdes se resaltan las variables importantes para cada modelo y en grises las descartables.

Parece claro que la variable más significativa, con diferencia en todos los modelos entrenados es **Room\_Type**, seguida de **Accommodates**, **Bathrooms**, **Cleaning\_Fee**, **Guests\_Included**, **Bedrooms**, **Latitude** y **Amenities**.

En cuanto a las menos significativas tendríamos: **Bed\_Type**, **Maximum\_Nights**, **Neighbourhood\_Cleansed**, **Property\_Type**, **Number\_of\_Reviews**, **Review\_Score\_Values**, **Minimum\_Nights**, **Cancellation\_Policy**, **Security\_Deposit**, **Features**, **Reviews\_per\_Month**, **Host\_Response\_Rate**, **Extra\_People** y **Longitude**.

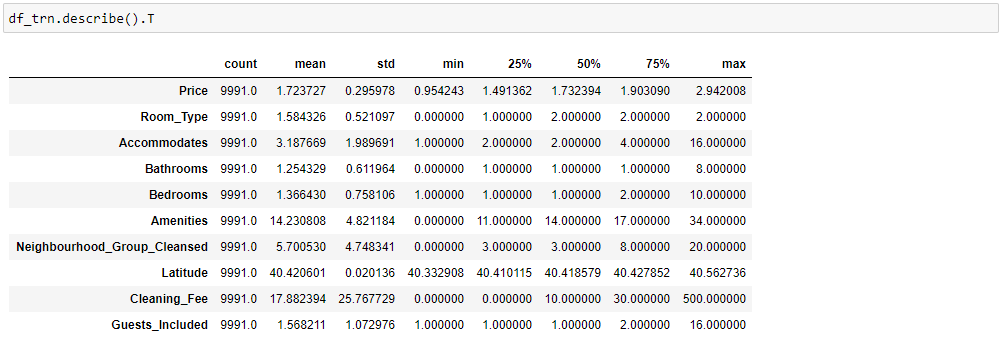
En base a lo anterior, decidimos eliminar estas catorce últimas columnas, reduciendo el dataset a un total de diez variables, nueve de ellas explicativas y nuestra columna target.

Es posible que, a futuro, entrenemos algún modelo de predicción para el cual falte/sobre alguna de las variables que hemos descartado/mantenido. Lo asumimos entendiéndolo como un daño colateral o un mal necesario en beneficio de un bien mayor, como lo es obtener un modelo que pueda llegar a ser implementable con unos costes de ejecución asumibles.

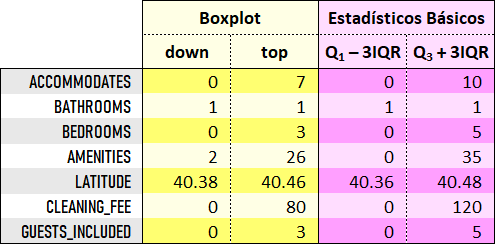
## **Tratamiento de Outliers**

Veamos de nuevo las relaciones de cada una de las 9 columnas explicativas frente a **Price**.

Observamos –de forma totalmente subjetiva y entendiendo que el scope de nuestros datos engloba la ciudad de Madrid– lo que nos parecen claros outliers en todas las columnas, excepto en **Neighbourhood\_Group\_Cleansed** y **Room\_Type** por ser estas variables categorizadas.

Veamos los estadísticos básicos y gráficas boxplot de nuestras variables (incluida **Price**).

Hemos excluido de las gráficas, las variables categorizadas ya que sus pesos son indicadores y no cuantificadores.

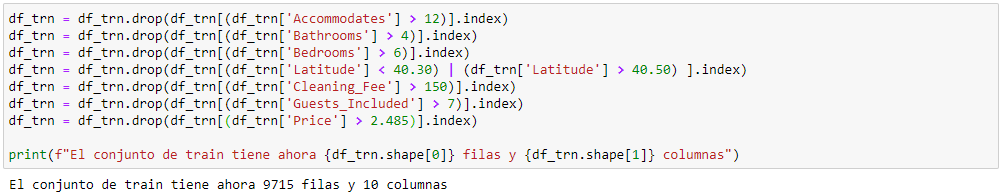
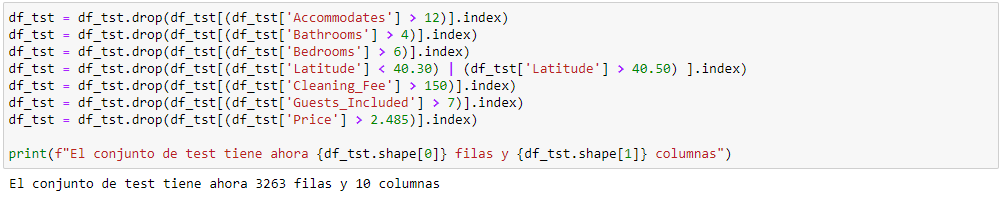
Dejaremos para el último lugar nuestra variable objetivo y primero compararemos la apreciación de los datos sobre los diagramas de cajas con los cálculos matemáticos sobre los datos estadísticos básicos.

A la vista de la comparativa anterior y teniendo en cuenta, que ninguna de las dos formas de definir o calcular los outliers es absolutamente correcta, que como outliers debemos considerar valores totalmente extremos y que no deberíamos eliminar un porcentaje excesivamente elevado de registros de nuestros conjuntos de datos, fijamos –con cierta cordura, pero de forma subjetiva– los siguientes umbrales para marcar los valores outliers de nuestras variables:

* **Accommodates** ≤ 12
* **Bathrooms** ≤ 4
* **Bedrooms** ≤ 6
* 0 < **Amenities** ≤ 35
* 40.30 < **Latitude** < 40.50
* **Cleaning\_Fee** ≤ 150
* **Guests\_Included** ≤ 7

Veamos ahora que pasa con nuestra variable objetivo **Price**.

Su boxplot nos marca outliers por encima de 2.6 –lo que equivale a unos 400€–, recordemos que esta variable esta transformada logarítmicamente. Según los cálculos estadísticos Q3 + 3IQR = 3.14 –unos 1375€–.

Nos encontramos con dos valores demasiado dispares y con poco sentido, por tanto, debemos decidir de nuevo subjetivamente. Consideraremos como outliers los valores de **Price** por encima de 2.845 –que son unos 700€–. Aplicamos este tratamiento sobre nuestros conjuntos de datos.

Hemos eliminado 276 observaciones del conjunto de entrenamiento, un 2,76%.

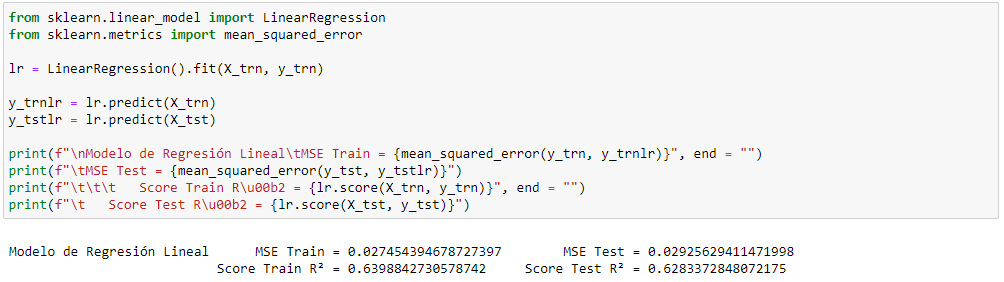
Y del conjunto de test se han borrado 82 registros, un 2,45%.

# **Modelos para Predicción**

Tenemos analizados, limpios y ajustados nuestros conjuntos de datos, por tanto, vamos a aplicar diferentes modelos de machine learning para nuestro ejercicio de predicción.

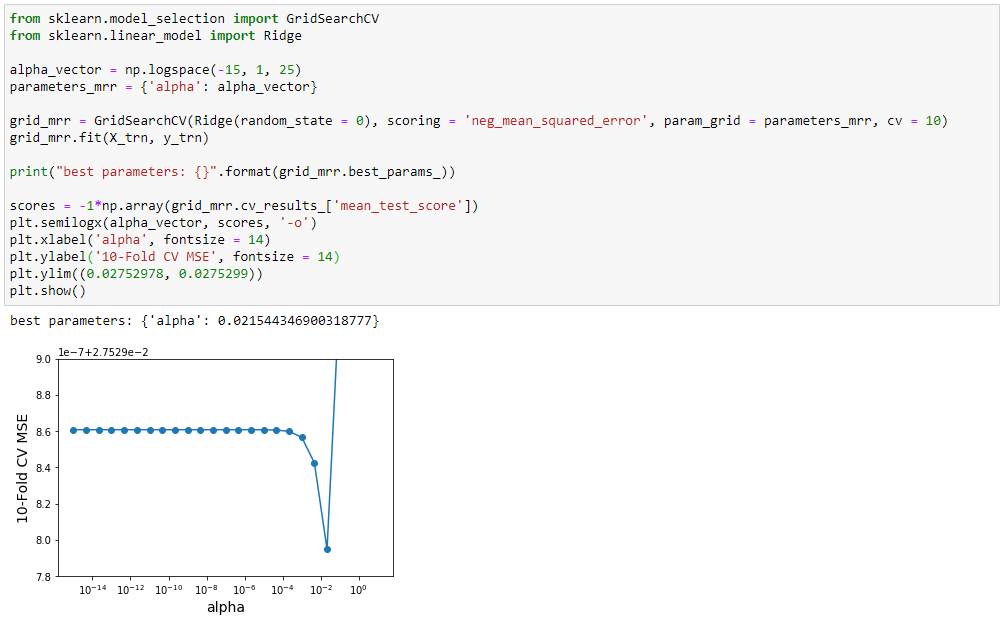
Siguiendo el principio de la *Navaja de Occam* comenzaremos probando modelos más simples e iremos aumentando su complejidad. Finalmente compararemos y veremos cual nos quedamos.

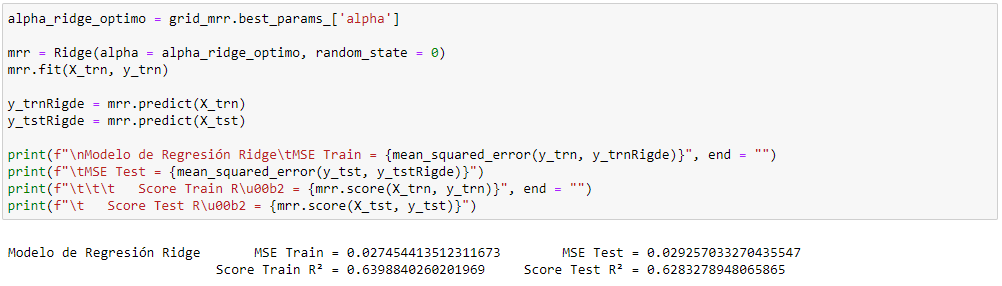
## **Modelo de Regresión Lineal**

El modelo más simple sería un **LinearRegression** de la librería **linear\_model** de **scikit-learn**. Este modelo no tiene hiper-parámetros, por tanto, lo aplicaremos directamente sobre los conjuntos de datos y mediremos sus resultados.

El modelo no se aprecia especialmente bueno, su bondad de ajuste no es siquiera alta, aunque su error cuadrático medio no es pésimo.

## **Modelo de Regresión Ridge**

Es el turno del modelo de regresión **Ridge** de la librería **linear\_model** de **scikit-learn**. Este modelo tiene un hiper-parámetro **alpha** que debemos obtener optimizado.

Una vez obtenido el valor óptimo de **alpha** lo aplicamos sobre el modelo de regresión **Ridge** trabajando sobre nuestros datos.

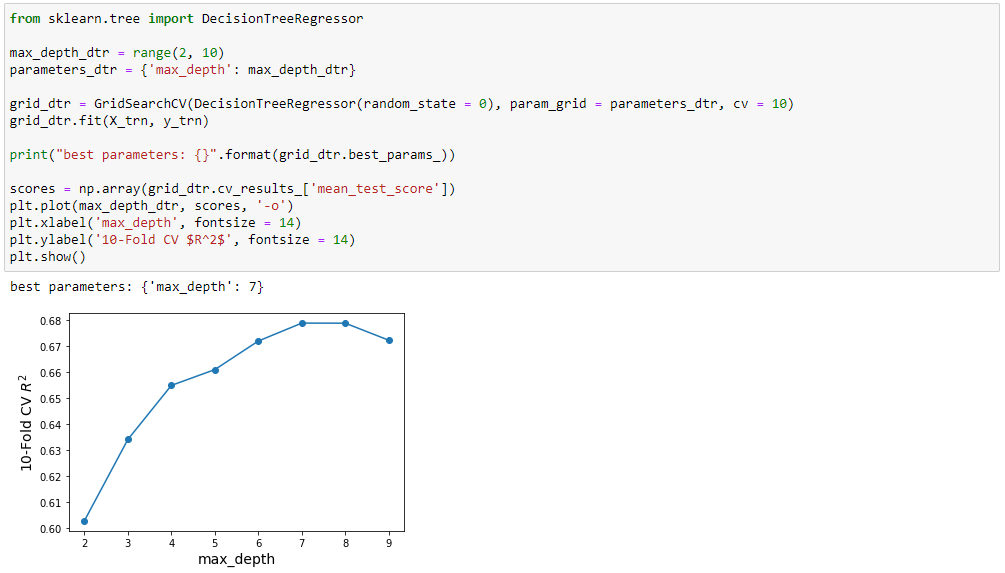
El valor óptimo de **alpha** es muy bajo, por eso el modelo se aproxima mucho al **LinearRegression** y los resultados son prácticamente calcados.

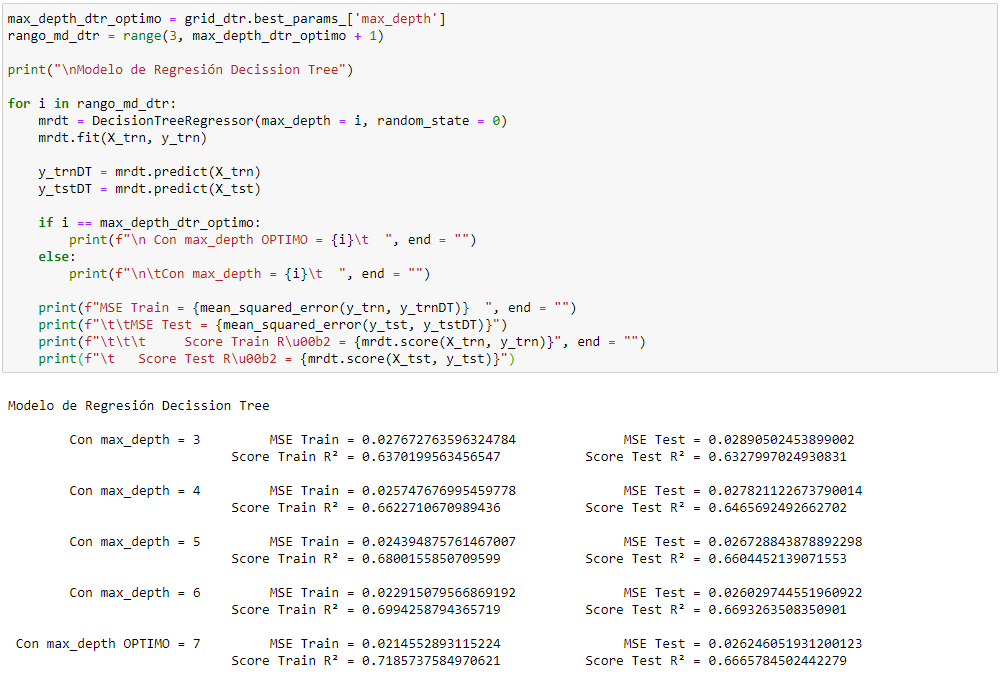
## **Modelo de Regresión Lasso**

Veamos ahora un modelo de regresión **Lasso**. Al igual que hicimos antes, en el párrafo 9.3, obtendremos su hiper-parámetro**alpha** óptimo y lo aplicaremos sobre el algoritmo.

De nuevo el valor de **alpha** es muy bajo y este modelo vuelve a aproximarse a los dos anteriores. En consecuencia, sus resultados son totalmente análogos, por no decir prácticamente iguales, a los de los modelos **LinearRegression** y **Ridge**.

## **Modelo de Regresión Decission Tree**

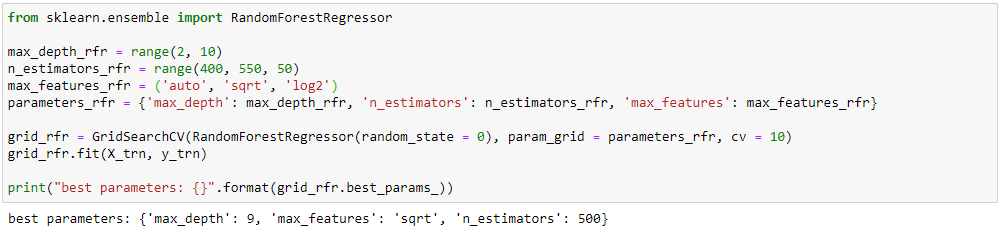
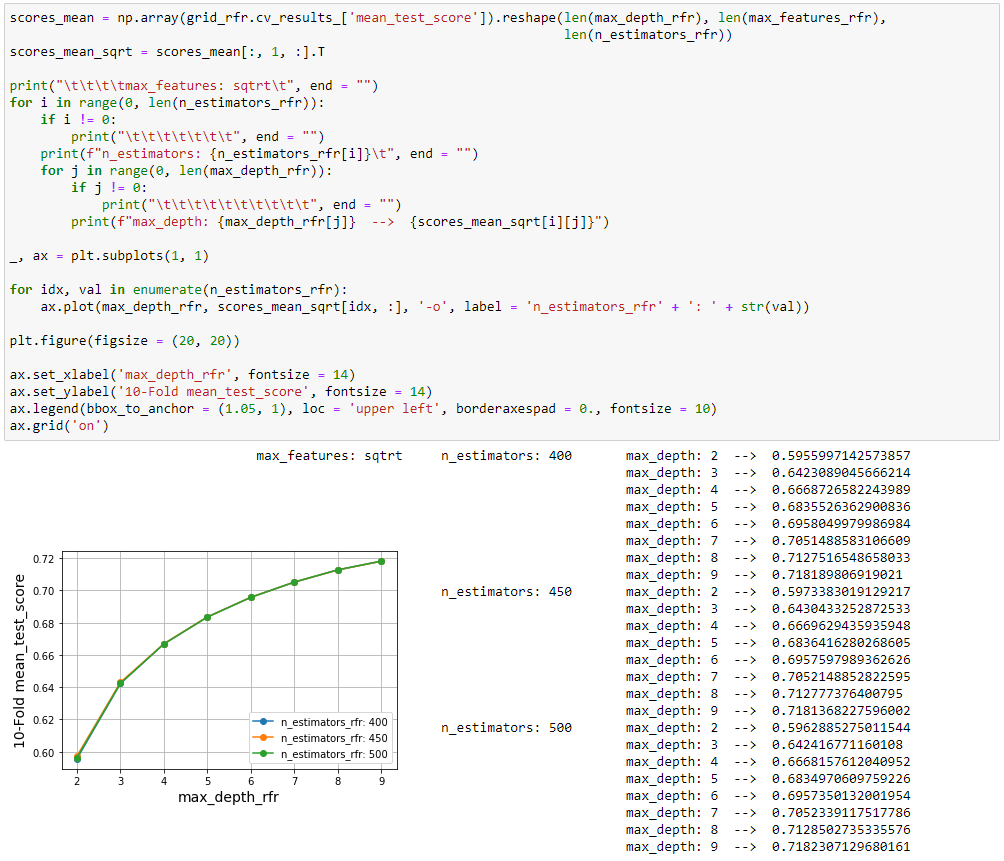
Veamos ahora un modelo de regresión **DecissionTreeRegressor**. Tal y como hicimos en el párrafo 9.4, primero obtendremos su hiper-parámetro**max\_depth** óptimo.

Hemos obtenido un valor óptimo de **max\_depth** = 7. Como nos parece una profundidad bastante alta, veamos que tal responde el modelo con varios valores menores.

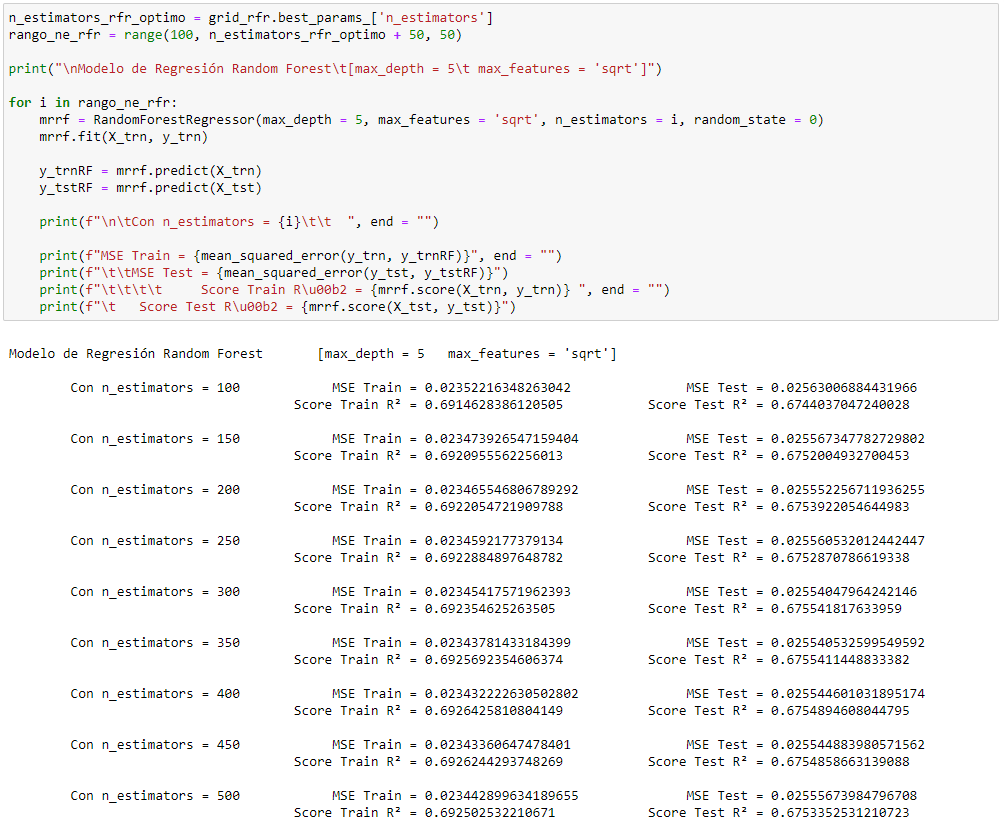
Los resultados mejoran los de los modelos anteriores sobre el conjunto de test a partir del valor 3 de **max\_depth**, pero tampoco podrían definirse como especialmente buenos.

Los valores tanto del error cuadrático medio, como de la bondad de ajuste mejoran a medida que aumenta la profundidad del árbol. Observamos que a partir del valor 5 de **max\_depth**, la mejora no es tanta en el conjunto de test. Elegiríamos ese valor para el hiper-parámetro en beneficio de obtener un modelo menos complejo que probablemente podría generalizar mejor.

## **Modelo de Regresión Random Forest**

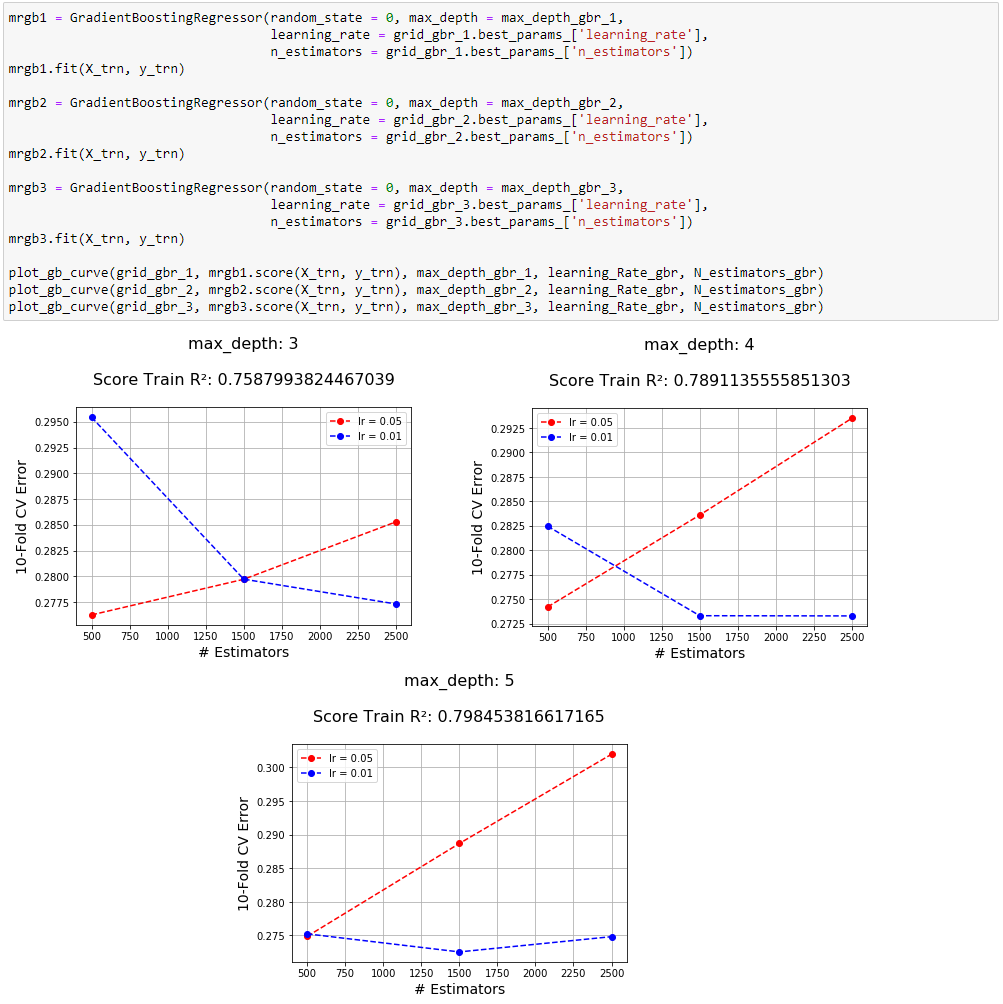
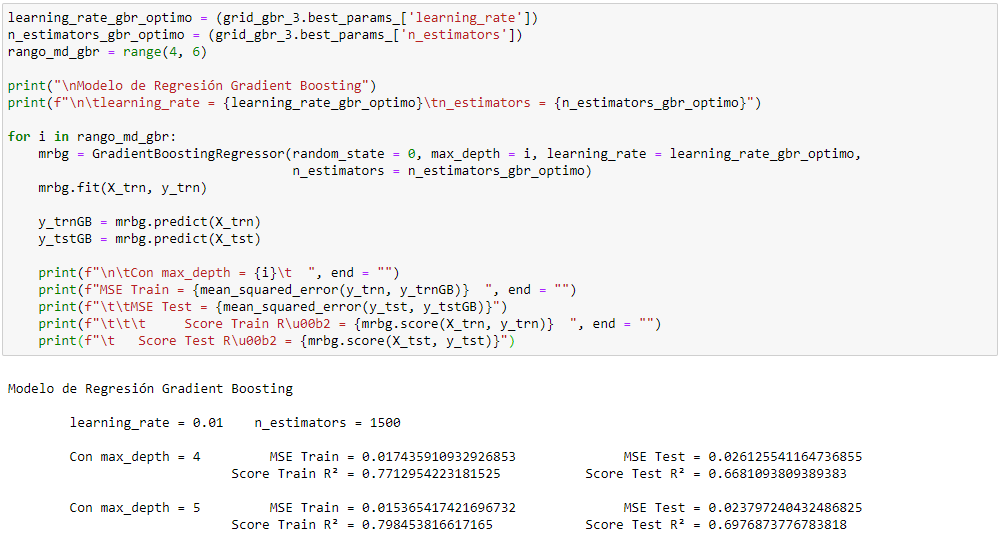
Toca ejecutar un modelo de regresión **RandomForestRegressor**. Tal y como hicimos en el párrafo 9.5, lo primero será buscar los valores óptimos de los hiper-parámetros (**max\_depth**, **n\_estimators** y **max\_features**).

Fijando **max\_features** al valor sqrt, las tres curvas se superponen y es muy complicado poder ver diferencias. Pero observamos que a partir del valor 5 de **max\_depth** la ganancia al aumentar **n\_estimators** no es demasiada. Fijamos por tanto ese hiper-parámetro también.

Los resultados mejoran un poco los de los modelos anteriores y habiendo fijado los valores de **max\_depth** y **max\_features**, **n\_estimators** parece no tener casi influencia, por lo que elegiríamos su mínimo valor por sostenibilidad de ejecución.

## **Modelo de Regresión Gradient Boosting**

Ejecutaremos ahora un modelo de regresión **GradientBoostingRegressor**. Tal y como hicimos en el párrafo 9.6, lo primero será buscar, en tres ejecuciones, los valores óptimos de los hiper-parámetros (**max\_depth**, **n\_iterations** y **learning\_rate**).

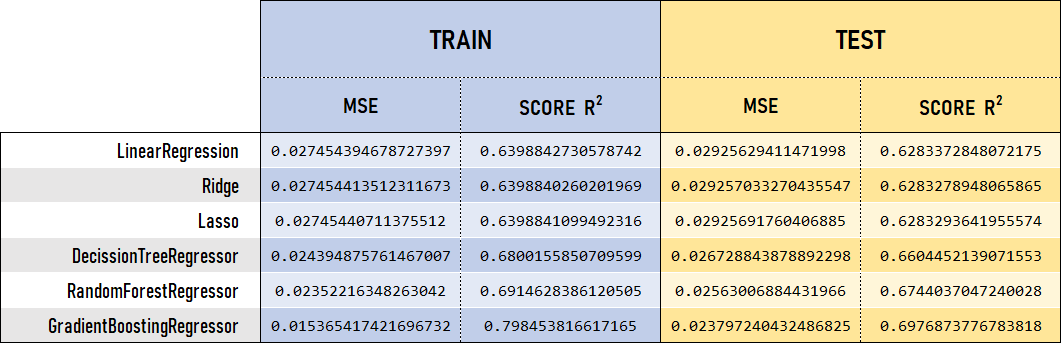
La mejor bondad de ajuste, –ya cercana 0.8– se obtiene con el hiper-parámetro **max\_depth** = 5, aunque el menor error nos lo da con **max\_depth** = 4.

Observamos mejores métricas con el hiper-parámetro **max\_depth** = 5 y por tanto lo elegimos.

Este modelo ya mejora de forma más ostensible los modelos de regresión anteriores.

# **Conclusión Final**

Una vez entrenados y medidos los principales algoritmos para regresión de machine learning debemos seleccionar uno como modelo final y solución al problema que se nos plantea.

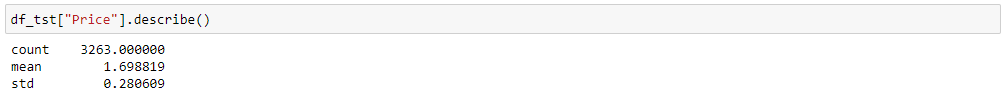
En el cuadro inferior se muestran las métricas de los seis modelos analizados.

Los tres primeros modelos, **LinearRegression**, **Ridge** y **Lasso** son prácticamente idénticos, sus métricas difieren a partir del quinto o sexto decimal. Al ser modelos más simples, proporcionan métricas mucho más discretas que sus sucesores.

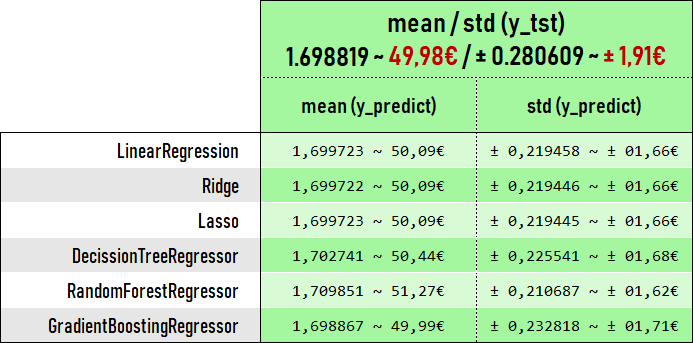
El modelo **DecissionTreeRegression** ya nos ofrece una mejora apreciable sobre los tres anteriores reduciendo el error cuadrático medio a las 25 milésimas y llevando la bondad de ajuste al 0.7.

El modelo **RandomForestRegression** no mejora en demasía al anterior, reduce una milésima el error y aumenta una centésima la capacidad de predicción. De estar nuestra decisión final dividida entre este modelo y el anterior, sin dudar elegiríamos el algoritmo del árbol simple.

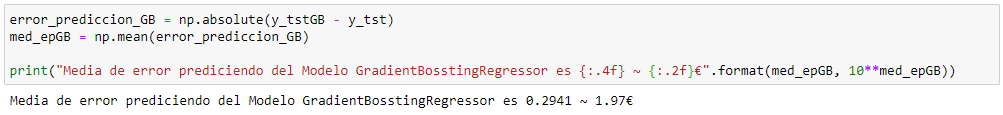
El último modelo visto **GradientBoostingRegressor** es el que mejores resultados ha ofrecido, reduciendo 1 centésima el error y aumentando prácticamente una décima la bondad de ajuste. Esta diferencia es la que le lleva a ser el elegido como solución final.

Durante las ejecuciones de código de los modelos hemos ido realizando una comprobación de la que no hemos querido hablar hasta ahora.

Nuestro target, en el conjunto de test –con 3.263 registros– tiene una media de 1.698819 y una desviación típica de 0.280609. Recordemos que la variable fue transformada logarítmicamente, por tanto, sus valores reales serían 49,98€ y 1,91€ respectivamente.

Hemos ido midiendo una a una las medias de los modelos prediciendo sobre el conjunto de test. El resultado se muestra a continuación.

Nuestro modelo elegido, **GradientBoostingRegressor** prediciendo, es el que más acerca los valores de media y de desviación típica a los del conjunto real de datos de test. Quizá podríamos decir que hasta sobreajusta, pero realmente no estamos hablando de los valores en sí, si no de sus estadísticos básicos calculados a partir de todos ellos.

De hecho, la media de su error prediciendo sería:

Es decir, falla en aproximadamente 2€ de media prediciendo.

Este nos parece un buen resultado y concluimos confirmando nuestra elección.

# **Agradecimientos**

Como nota final de este trabajo me gustaría dar las gracias al profesor **Rubén Romero**, por su gran atención despejando nuestras dudas y siguiendo nuestra progresión (tanto en las clases como en Slack) y por su generosidad preparando toda la documentación teórica y práctica para las clases.

Pienso que esta asignatura de Machine Learning es muy densa y ciertamente compleja. El profesor ha hecho un trabajo inmejorable ajustando el temario al número de clases, habiendo conseguido llegar a ver –creo– todo lo imprescindible y más importante. Personalmente me ha costado un poco llegar hasta aquí… no me considero siquiera un pobre estadístico y estoy aun bastante lejos de poder entender perfectamente y diseñar un buen modelo de Machine Learning, pero pasito a pasito –lo pienso y me enorgullezco un poco de ello– he conseguido llegar a poder elaborar este trabajo y en adelante espero poder seguir avanzando y aprendiendo en este campo de la ingeniería de los datos.

Sin más, mi más sincero agradecimiento por todas las cosas que he aprendido, que espero se vean reflejadas en esta práctica.

**F. Javier Gonzálvez**

# **ANEXO**

Se incluyen aquí los notebooks con las celdas de código ejecutadas durante la práctica.

**  **