# Ejercicio 5 Análisis y Curación de Datos

Grupo 1 - 2022

15 de agosto de 2022

## 1. Introducción

Para comenzar la exploración, análisis y curación de datos, partimos de dos bases de datos distintas, la primera está relacionada al precio de viviendas en la ciudad de Melbourne, Australia, y la segunda base de datos fue realizada con información del valor de alquileres de viviendas, en distintas áreas del mismo país.

## 2. Desarrollo del Entregable 1

### 2.1. Ejercicio 1

Para comenzar, el primer entregable comienza con la creación de una base de datos en SQL, gracias a la librería SQLalchemy. Luego, se arma una segunda base de datos con información de alquileres de Airbnb y se hacen las siguientes consultas:

- Cantidad de registros totales por ciudad.
- Cantidad de registros totales por barrio y ciudad.

Dado que solo la base de Airbnb contiene datos de ciudad y barrio, es esta base la que se utilizó para realizar consultas por SQL. Estos se pueden ver desarrollados en su totalidad en el documento *"Grupo N°1 Ejercicio 1 Entregable Parte 1 EyCD"*

El primer ejercicio finaliza con la combinación de los dos dataset armados (Melbourne y Airbnb), gracias a la función *JOIN* de *SQL*. Ambas bases de datos, fueron unidas a través de *Postcode* de *Melbourne* y *Zipcode* de *Airbnb*, ya que representan los mismos datos.

### 2.2. Ejercicio 2

La segunda parte del entregable pide seleccionar aquellas columnas que al equipo le parezca más relevante para poder hacer una predicción del valor de las propiedades. Luego, se pide agregar información adicional, respectiva al entorno de la propiedad.

Para saber qué información procedente de las columnas del dataset recientemente creado, el equipo realizó un análisis general de cada una de las bases de datos previas, donde se presenta información como dirección, cantidad de habitaciones, baños, año del inmueble o su superficie, entre otras cosas. Para tener una idea más detallada del comportamiento del precio respecto a las distintas variables, se hizo un análisis de correlación, donde se obtuvieron los siguientes resultados:

- Existe una correlación positiva (0.496634) entre el valor de la propiedad y la cantidad de habitaciones (Rooms") que tiene.
- Hay una correlación negativa (-0.162522) entre el valor de la propiedad y la distancia que la separa del centro financiero/ de negocios de Melbourne ("Distance").
- Se puede ver una correlación positiva (0.475951) entre el valor de la propiedad y el número de dormitorios ("Bedroom2") que tiene.
- Existe una correlación positiva (0.467038) entre el valor de la propiedad y el número de baños ("Bathroom") que tiene.

- Encontramos una correlación positiva (0.238979) entre el valor de la propiedad y el número lugares para autos (Çar") que tiene.
- se muestra una correlación negativa (-0.323617) entre el valor de la propiedad y el año en el que fue construida ("YearBuilt").

Por razones de simplicidad, no se consideraron las correlaciones inferiores a 0.15 (en valor absoluto) ni las variables de latitud y longitud por tener que ser analizadas en conjunto.

Posteriormente, se utiliza el test de Pearson para contrastar la significancia de dichas correlaciones. En todos los casos, el p-value es inferior a 0.05 por lo que rechazaríamos la hipótesis nula de no correlación entre las variables y confirmaríamos las correlaciones son estadísticamente significativas al 5 %. Se presentaron casos donde no se pudo realizar el test, cuando se utilizaron columnas que contenían valores nulos, tales como *Car* y *Yearbuilt*, entre otras. En estos casos se procedió a calcular dichos coeficientes con dos bases alternativas, sin valores nulos en esas columnas, para analizar la significancia de la correlación. De su análisis se concluyó que también esas columnas mencionadas eran relevantes para predecir el valor de la propiedad. Para una mejor visualización de cómo puede ser la correlación entre dos variables, se gráfica en la figura 1 como ejemplo, cómo es la variación del precio de los inmuebles, respecto a la cantidad de habitaciones. Habiendo realizado un análisis similar con la

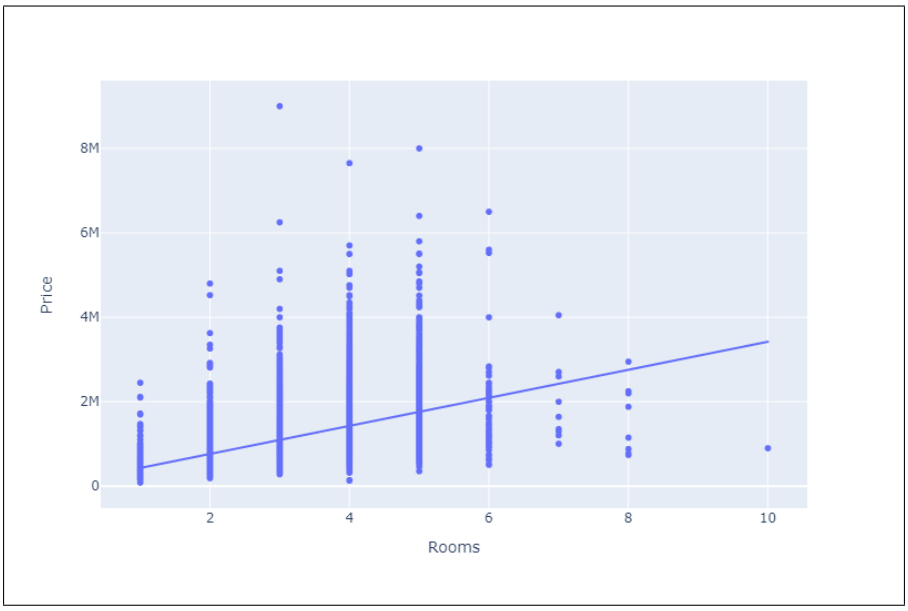


Figura 1: Gráfico de comportamiento del precio de inmuebles (Price), respecto a la cantidad de habitaciones (Rooms).

base de Airbnb, las columnas más relevantes para sumar a la base de Melbourne son: *Price, Weekly Price y Monthly Price.*

## 2.3. Ejercicio 3

Como punto final se pide crear y guardar un nuevo conjunto de datos, con todas las transformaciones realizadas a lo largo del primer entregable, para luego utilizarse en la segunda parte.

Otras columnas que se encuentran tanto en la base de Melbourne como en la de Airbnb que podrían haber servido para combinar los datasets son las de latitud y longitud. Pero, los porcentajes de los valores de intersección para ambas columnas son muy bajos. Esto puede deberse porque la forma en que están presentados, ya que se escriben con un gran numero de decimales y es difícil que coincidan exactamente. Por ende, se decidió no utilizarlos.

# 3. Desarrollo del entregable 2

El segundo entregable comienza cargando la base de datos trabajada en la primera parte. Entre los objetivos aquí, es poder trabajar tanto con variables numéricas como categóricas y con diferentes herramientas lograr solucionar algunos problemas como bases de datos con valores faltantes , hacer un análisis PCA sobre la base de datos, ente otros.

## 3.1. Ejercicio 1: Enconding

En la primera parte del ejercicio 1 lo que se pide es tomar el dataset del entregable anterior, quitarle las columnas de *BuildingArea* y *YearBuilt* y aplicarles una codificación *One Hot Encoding* a cada fila, tanto a las variables categóricas como numéricas.

Como opcionales se permite utilizar la función *OneHotEncoder (OHE)* o *Directvectorizer (DV)*. Ambas funciones lo que hacen es recorrer cada una de las columnas asignadas y analizar la cantidad de variables distintas que posee cada una de las columnas, posteriormente arma una matriz numérica con la misma cantidad de filas que el dataset original, pero desglosa las columnas según la cantidad de variables que esta tenga, donde para el caso de columnas categóricas, se le asigna un *1* a cada variable individual.

Para la aplicación de las funciones OHE y DV, se fueron quitando columnas que no se consideraron relevantes, ya sea porque tenía variables numéricas continuas (lo que aumentaría enormemente la matriz mencionada) o que tuviera un excesivo numero de variables categóricas, como los es *Address*. Un aumento del tamaño de la matriz implica que la misma comience a demandar mayores costos computacionales y se torne ineficiente su uso. Luego de haber aplicado OHE y DV, el equipo optó por quedarse con los resultados obtenidos de DV, ya que devolvían una matriz con una distribución más compacta de columnas, sin perder información.

## 3.2. Ejercicio 2: Imputación por KNN

En esta etapa lo que se busca es de manera eficiente, con la matriz previamente obtenida, imputar aquellas columnas que tuvieran datos faltantes, a través de la herramienta *IterativeImputer*, utilizando un estimador *KneighborsRegressor*.

Antes de realizar la imputación, se verificó que hubieran columnas con valores nulos (*nan*) en las columnas que pertenecían al dataset. La que más valores nulos poseían, eran *BuildingArea* y *YearBuilt*. En la figura 2 se muestran columnas de color azul representando los datos de cada columna del dataset que hemos trabajado, además se pueden ver espacios en blanco, que son los valores faltantes. Gráficamente la distribución de estos datos faltantes no están agrupados en un sector, sino que están distribuidos aleatoriamente en las columnas, por lo que sería un desperdicio de información eliminar las filas donde existen estos *nan*. Aquí lo más conveniente es imputarlos utilizando información de la base de datos, usando por ejemplo el método de KNRegressor.

Comenzamos por agregar nuevamente las columnas *YearBuilt* y *BuildingArea*, además de otras columnas con variables continuas como *Price*, que también son de interés. Es necesario tener en cuenta que el resultado del *DirectVectorizer* es una matriz, que es en sí un *array* y no un dataset, por lo que será necesaria convertirla, para agregarle las columnas mencionadas.

Como el estimador KNN trabaja midiendo la distancia euclídea que hay entre las variables, es muy susceptible a cometer un error, si los ordenes de magnitud entre las distintas variables es muy grande, logrando que algunas tengan más influencia sobre otras. Para solucionar esto, se escalaron todas las columnas de la base de datos, en un rango entre cero y uno, a fin de no cometer este error. Una vez realizado la imputación con KNRegressor, se debe reescalar todas las columnas para que regresen a sus valores correspondientes. A modo de ejemplo, se muestra en la figura 3 se muestra la densidad de datos de la columna *YearBuilt* antes y después de haber pasado por el KNRegressor, donde se puede ver que la densidad de información crece rotundamente al imputar los valores *nan* de la original.

## 3.3. Ejercicio 3: Reducción de dimensionalidad.

En este ejercicio lo que se busca es reducir la dimensionalidad de la matriz obtenida del punto anterior, utilizando la técnica de Análisis de Componentes Principales o en inglés PCA, que de forma

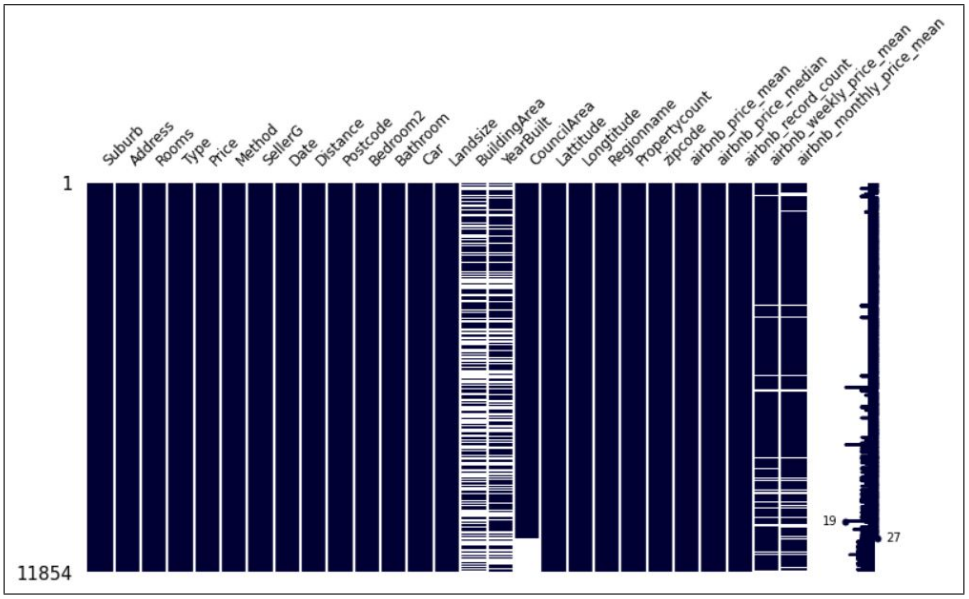


Figura 2: Distribución de datos faltantes en las columnas del dataset.

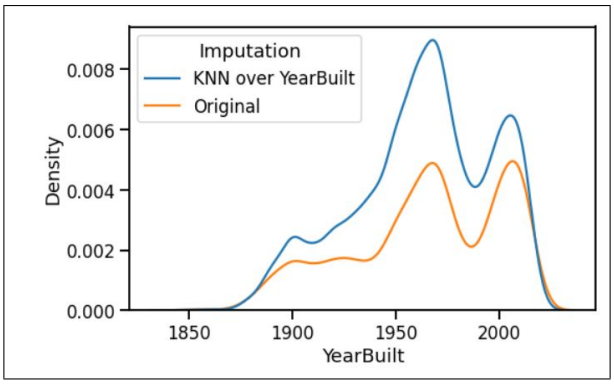


Figura 3: Comparación entre la densidad de datos de YearBuilt, antes y después de haber sido imputado.

resumida, lo que quiere es tener una descripción del conjunto de datos a través de un menor número de variables (componentes), tratando de conservar la mayor cantidad de información posible.

Para aplicar PCA siempre es necesario estandarizar previamente las variables de la matriz, ya que el procedimiento identifica las variables con mayor varianza y, si no están estandarizadas, las de mayor valor dominarán al resto. En el ejercicio anterior ya se estandarizó la matriz con la función MinMaxScaler. Al aplicarse el PCA, tal como se pide en este ejercicio, se comenzó utilizando 20 componentes. A través de la gráfica mostrada en la figura 4 se puede ver que la relación de la varianza cae fuertemente hasta los seis primeros componentes principales, luego la misma va decreciendo de forma suave, donde se llega a un Ratio de 0.03, conforme al numero de componentes aumenta. Es por ello que se decidió tomar finalmente 6 componentes principales, en lugar de 20.

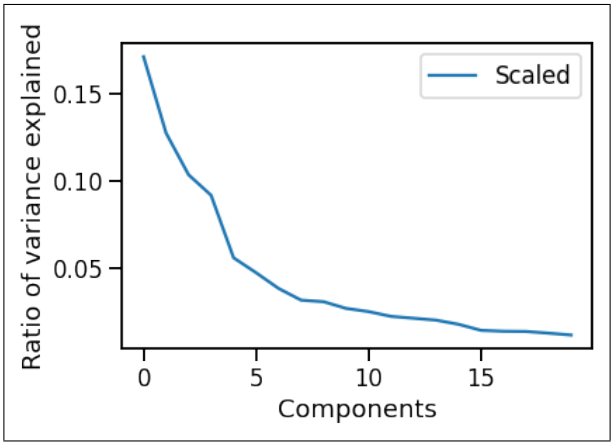


Figura 4: Curva de relación de varianza explicada en función del número de componentes.