Entregable Parte 1

EJERCICIO 1.1 Y 1.2

Criterios de Inclusión y Exclusión de Variables.

Las conclusiones fueron extraídas a partir de la inspección de las variables con la función *profiling*.

Para tomar esta decisión tomamos dos criterios complementarios. Por un lado, un criterio de dominio, tratando de comprender en términos conceptuales cuáles son las variables que pueden ser relevantes para predecir el precio de una propiedad. Por otro lado, usamos un criterio técnico estadístico, en función de la calidad de los datos y la colinealidad entre las variables

Variables Redundantes:

Tenemos varias variables que trabajan con la ubicación de la propiedad. Si bien las variables **Latitud** y **Longitud** son las más exactas, también son las más difíciles de interpretar y utilizar. Lo mismo sucede con **Address.** También quitaremos **Region Name.**.

Vamos a quedarnos con la variable **Suburb** que es más específica y la variable **Council Area** que es relativamente más general. Además, tenemos la variable **Distance** que contiene la distancia desde el centro y tiene una relación negativa moderada con el Precio. También conservamos **Postcode** pero porque la necesitaremos para otros ejercicios. Si este dataset fuese solo para el objetivo de predecir nuestra VD, posiblemente conservaríamos un solo indicador de posición/locación.

También, tenemos varias variables que dan cuenta del tamaño y la distribución del espacio de la propiedad. Las variables **Rooms** y **Bedroom2** tienen una correlación casi perfecta, por lo que se vuelven redundantes. Decidimos quedarnos con la variable **Rooms** dado que es más confiable (la variable **Bedroom2** fue tomada de una fuente diferente). Exploramos también la relación de la variable Rooms con **Bathroom, Car, Landsize y BuildingArea** y, a su vez, de estas con el precio. Todas tienen algún nivel de correlación, pero no suficiente como para hablar de colinealidad y todas se relacionan con el precio, por lo que podemos asumir que serán relevantes en un modelo predictivo.

Variables Irrelevantes:

Las variables **Method** y **SellerG** no parecen ser relevantes para nuestro problema, por lo que fueron removidas. Además, esta última tiene 268 valores distintos, lo que escalaría el tamaño de nuestro dataset. Del mismo modo **PropertyCount** que contiene la cantidad de propiedades disponibles en la zona.

Resultado:

Variables Incluidas	Variables Excluidas
Price	Address
Type	Method
Rooms	SellerG
Bathroom	Bedroom2

Car	Lattitude
Date	Longitude
YearBuilt	RegionName
BuldingArea	PropertyCount
Landsize	
Suburb	
Distance	
CouncilArea	
PostCode	

Recategorización de Variables

Las variables Bathrooms, Rooms y Car fueron recategorizadas, generando un número menor de categorías y garantizando que todas tengan un número no menor a 600 casos. El criterio de categorización fue de dominio antes que estadístico, considerando cuál era la información relevante a conservar.

Baños: se asumió que –dado que ninguna es un terreno baldío- todas las propiedades debían tener al menos 1 baño. Así, todos los ceros (menos del 0,25% de la muestra) fueron reemplazados por uno (que es la mediana de la variable). Por otra parte, observando la distribución de los datos y, teniendo en cuenta la baja frecuencia de más de 3 baños, se generaron las siguientes categorías: 1, 2 y 3 o más.

Rooms: en este caso, no teníamos ceros. Había muchos 1, lo cual es razonable, se trataría de monoambientes. Atendiendo a la distribución de los datos y a la necesidad de conservar la mayor cantidad de información posible, solo se agruparon propiedades con 5 o más ambientes. Conservando las categorías individuales previas.

Cars: en este caso, el cero es un dato en sí mismo, ya que son propiedades que no tienen cochera, por lo que no se reemplaza ese dato. Atendiendo a la distribución de los datos, se conservan individualmente las categorías 0, 1 y 2 y se agrupan aquellas que tienen 3 o más espacios.

Detección y tratamiento de Outliers

Se analizaron los casos atípicos en las variables numéricas: Price y Landsize. En ambos casos se utilizó un método conservador para la identificación de casos atípicos basado en el rango intercuartílico estableciendo el límite en 3 veces ese rango (por encima y por debajo). Aplicando este método y eliminando estos casos atípicos se perdieron en total 349 casos (2,5% del total). Se consideró que era pertinente su eliminación teniendo en cuenta que el objetivo era predecir el valor de una propiedad, para lo cual los casos muy atípicos pueden ser irrelevantes e, inclusive, dificultar la tarea.

Además, se identificaron casos atípicos en las variables año de construcción y área construida. Sin embargo, eliminar los casos atípicos también nos dejaría sin las filas nulas, que luego necesitaremos para imputar.

EJERCICIO 1.3

a y b) se decidió agregar las siguientes variables del set de datos de Airbnb: media del precio diario de las propiedades en la zona (en función del Zipcode), discriminadas en habitación o propiedad completa y promedio del puntaje de la ubicación dado en las reviews

(review_scores_location). Se agregaron estas variables al dataset de Melbourne utilizando las variables Zipcode (airbnb) y Postcode (Melbourne).

c) En primer lugar, para obtener una mayor precisión de la ubicación de la propiedad, se podrían usar las variables latitud y longitud que están presentes en ambos datasets. Sin embargo, este dato podría servir para generar áreas más pequeñas que un suburbio o un zipcode y matchear los resultados de ambos datasets dentro de esas áreas. De cualquier modo, esto requería de cierta expertise en el manejo de mapas y variables de posición que posiblemente no justificarían la ganancia en precisión en la ubicación. De modo similar se podría proceder localizando la propiedad utilizando la variable Street de Airbnb y Address de Melbourne. Otra variable que podría aportar para obtener más información sobre la zona es el número de reviews por mes (reviews_per_month). Así, se podría conocer más sobre cuán demandada es una propiedad y, por ende, la zona en la que se ubica un conjunto de propiedades (calculando el puntaje medio por zona como hicimos con el precio).

EJERCICIO 2

Durante la exploración de las variables *Suburb* y *CouncilArea* se identificó que algunos suburbios estaban asignados a más de una CouncilArea. Para poder hacer la imputación de la variable CouncilArea con la información presente en Suburb se tuvo que tomar una decisión sobre esta situación: ¿a qué CouncilArea va a corresponder un determinado suburbio cuando hay dos o más posibilidades? En estos casos se tomó la CouncilArea a la que correspondía el suburbio con mayor frecuencia. Se genera una nueva tabla que contiene sólo los datos relevantes: una CouncilArea para cada suburbio y después se hace un leftjoin por la variable suburbio con el dataset original para rellenar la variable ciudad faltante. Aún restan rellenar las CouncilArea correspondientes a 8 suburbios que no tienen datos en este dataset. Para lograr esto se recuperan los datos de las variables *City* (equivalente a CouncilArea según nuestras búsquedas sobre la división geográfica de Melbourne y su área metropolitana) y *Suburb* del dataset de Airbnb. Se consigue rellenar 7 de estos suburbios, quedando únicamente una fila que corresponde al suburbio de Wallan para el cual desconocemos su CouncilArea porque está ausente en ambos datasets.

Para imputar los valores faltantes de las columnas que se agregaron a partir del conjunto de datos de AirBnB se decidió imputar por el promedio agrupado por CouncilArea (ya que se puede ver en el reporte que las nuevas features están correlacionadas con CouncilArea). Se calculan entonces estos promedios agrupados y se realiza un merge de estos con el dataframe completo a través de la feature CouncilArea. Se completan los valores faltantes con los correspondientes en las columnas de promedios por CouncilArea y luego se eliminan estas columnas. Finalmente se chequean cuantos valores nulos quedan y se eliminan las filas con valores nulos en Car_group, CouncilArea y las agregadas de airbnb ya que son solo 67 (0.5% del total). De esta forma queda listo el dataset para el próximo entregable con 13164 filas, 16 columnas y valores nulos solo en YearBuilt y BuildingArea. Guardamos este dataset en la carpeta donde estamos trabajando en formato csv con el nombre 'dataset_new.csv'

Entregable Parte 2

Con el dataset obtenido en la parte1 del entregable, al mismo se le eliminó la variable Postcode, la cual habíamos conservado para hacer uno de los ejercicios de la parte 1.
Esta variable se eliminó por ser redundante, ya que se puede ver que cada suburb tiene un único postcode, y cada postcode puede incluir a varios suburbs. Así que descartamos Postcode y nos quedamos solo con Suburb.

Ejercicios 1: Encoding
Antes de aplicar OneHotEncoder se tiene que reducir el número de valores posibles para las variables categóricas Suburb y CouncilArea, agrupando las menos comunes en la categoría 'otros'. Para esto definimos una función que toma el número de categorías con las que nos queremos quedar. Para que corran el resto de los algoritmos ponemos un n bastante bajo, así no se generan tantas columnas al hacer el OneHotEncoder. Pero si se dispone de una computadora más potente se puede aumentar n y probablemente obtener mejores resultados en las predicciones de los valores faltantes.

Se utiliza la función DicVectorizer el cual aplica el OneHotEncoder sólo a variables que son categóricas, y a las variables numéricas las concatena a la matriz resultante.Por esta razón en este punto no se omitieron las columnas `BuildingArea` y `YearBuilt', el cual era una sugerencia de este punto, para luego el punto siguiente agregarlas nuevamente. Para aplicar esta función primero se debe hacer un pre-procesamiento de los datos , el cual consiste en transformar el dataset en un diccionario.

Ejercicios 2: Imputación por KNN
El método KNN que se utilizara para imputar los valores faltantes, requiere que las variables esten escaladas, ya que el mismo trabaja con distancias euclídeas que son sensibles a la magnitud de los vectores.

Para el escaleo de las variables numéricas se utiliza el método MinMaxScaler. Se emplea este método porque conserva la forma de la distribución original de los datos, además de no cambiar significativamente la información incrustada en los datos originales. El rango predeterminado para la función devuelta por MinMaxScaler es de 0 a 1. Las variables a escalar son Price, YearBuilt, BuildingArea, Landsize, Distance, airbnb_entire_home_price_mean, airbnb_one_room_price_mean, airbnb_location_score_mean.

Este imputador se utilizará para comparar dos situaciones, la primera consta de aplicar el método utilizando sólo las columnas ['YearBuilt','BuildingArea'] y la segunda utilizando la matriz obtenida en el punto anterior. Se realizan gráficos comparando las distribuciones de datos obtenidas con cada método de imputación.

Por último, transformamos nuevamente el conjunto de datos procesado en un pandas.DataFrame, utilizando los valores de la matriz densa imputada y las columnas resultante de haber aplicado el one-hot encoding

Ejercicio 3: Reducción de dimensionalidad.

Documentación del proceso Principal Component Analysis (PCA) realizado.

Para el procesamiento de los datos utilizando la función PCA, debemos realizar una sucesión de pasos. La primera acción que realizamos es setear que el número n de componentes principales de la matriz sea el mínimo entre 20 y la cantidad de columnas de la matriz. Este paso es requisito necesario solicitado por la cátedra.

Como segundo paso tomamos el dataset ya imputado (el cual no contiene valores NaN) y ya pre procesado y lo copiamos como una matriz densa. A continuación chequeamos la cantidad de valores y columnas existentes.

Importamos la librería scikitlearn que contiene la clase sklearn.decomposition.PCA que implementa la mayoría de las funcionalidades necesarias para crear y utilizar modelos PCA y seleccionamos una cantidad de componentes adecuados para el costo computacional que disponemos, el n que definimos antes, y procedemos a transformar la matriz. Finalizamos este paso chequeando nuevamente la cantidad de componentes con la función shape.

Finalmente debemos observar el ratio de varianza que posee cada columna, y así podemos argumentar que porcentaje de varianza representa cada componente y confeccionamos un ploteo que nos permita identificar cuántos componentes debemos seleccionar de la matriz para agregar como las nuevas características al conjunto de datos. Se optó por agregar las primeras 5 columnas de la matriz transformada.

Para responder a la pregunta si es necesario estandarizar o escalar los datos, debemos responder que en este caso en particular no es necesario ya que los datos ya han sido procesados para realizarles KNN.

Como ya tenemos el conjunto de datos procesados en un pandas.DataFrame, finalmente lo guardamos en la carpeta donde estamos trabajando en formato csv con el nombre 'dataset_procesado.csv'