Documentación

1. Selección de columnas relevantes

En este ejercicio, se realizó una selección de columnas relevantes para el problema de predicción del valor de la propiedad. Primero se creó una matriz de correlación para identificar las columnas numéricas que tenían una correlación significativa con el precio de la propiedad.

Observamos que la cantidad de habitaciones (en sus variables rooms y bethrooms) y la cantidad de baños son a primera vista las variables que mayor correlación directa tienen con el precio. En segundo lugar la cantidad de años de la construcción y la ubicación geográfica de la misma tiene una correlación negativa y menor, es interesante creemos tener en cuenta estas variables ya que en este primer análisis nos proveen la intuición de que están relacionadas con el precio de las viviendas. Tomamos como argumento seleccionar las variables que tienen una correlación con un valor absoluto mayor a 0,1. La variable BuildingArea la tendremos en cuenta a pesar de tener baja correlación porque nos parece una variable relevante a priori.

Respecto a las variables categóricas definimos eliminar la dirección ya que es una variable que no puede medirse como sí sucede con latitud y longitud (además de que con la ubicación geográfica ya estamos analizando la influencia de la ubicación en el precio), además eliminamos las columnas 'Method', 'SellerG' y 'Date' ya que no aportan mayor información.

Las columnas seleccionadas fueron:

- Suburb: lo tomamos como barrio
- Rooms: la cantidad de espacios en la vivienda.
- Type: tipo de vivienda
- Price: precio de la vivienda
- Distance: la distancia al centro de la ciudad.
- Postcode: el código postal.
- Bedroom2: la cantidad de habitaciones
- Bathroom: la cantidad de baños.
- Car: la cantidad de plazas de estacionamiento.
- YearBuilt: Año de construcción
- CouncilArea: el área del consejo.
- Regionname: el nombre de la región. (lo tomamos como ciudad.)
- Lattitude: la latitud geográfica.
- Longtitude: la longitud geográfica.
- Building Area: superficie construída

2. Limpieza de outliers

En esta etapa, se realizó la limpieza de outliers en la columna de precio ('Price'). Se realizó una función que elimina los valores extremos basándose en un criterio de percentiles. La función eliminó los valores que se encontraban por encima de 2.5 veces el rango intercuartil (IQR) por encima del tercer cuartil (Q3). Esto ayudó a eliminar los valores atípicos que podrían afectar el análisis y los modelos posteriores.

Podemos observar que la función no se eliminaron gran cantidad de registros y que los valores superiores de Price tienen más sentido.

3. Agregado de información adicional de AirBnB

Se agregó información adicional sobre el entorno de las propiedades utilizando un conjunto de datos de AirBnB. Primero se revisaron las variables disponibles en AirBnB y se incorporaron al análisis variables relevantes, como la ciudad, el barrio, el estado, el código postal, el tipo de propiedad, la cantidad mínima y máxima de noches, y los precios mensuales y semanales.

Luego de incorporar se observaron los valores faltantes para ver sobre qué columnas era necesario aplicar transformaciones. Las columnas de codigo postal, estado, barrio y los precios mensuales y semanales. son las que tienen datos faltantes. Como los precios mensuales y semanales tienen demasiados datos faltantes procederemos a eliminarlas del análisis. Para las otras variables con datos faltantes, se procedió a imputar los faltantes.

Se realiza la imputación de valores faltantes utilizando la estrategia de 'valor más frecuente' (most frequent) en el conjunto de datos de AirBnB. Primero, se crea una copia del conjunto de datos de AirBnB para preservar los datos originales. Luego, se instancia la clase SimpleImputer y se configura la estrategia de imputación. A continuación, se aplica la imputación en el conjunto de datos utilizando el método fit_transform. Esto ajusta el imputador a los datos y realiza la imputación de los valores faltantes en todas las columnas.

Finalmente, se muestra un gráfico de barras para visualizar la distribución de los valores faltantes en el conjunto de datos imputado. Este proceso de imputación y visualización de valores faltantes es útil para comprender la cantidad y distribución de los valores faltantes en el conjunto de datos y evaluar la efectividad de la estrategia de imputación utilizada. Una vez aplicadas las transformaciones se observa que ya no hay datos nulos.

Finalmente se unieron los datos de AirBnB con el conjunto de datos de ventas utilizando el código postal como clave de unión (zipcode para la base de AirBnB y postcode para la base original). Primero se estandarizaron los zipcode y se eligió dejar solo aquellos zipcode que puedan agregar informacion, por lo que se establece un mínimo 10 registros. Y luego se unieron las bases con el método

merge. Esto permitió enriquecer el conjunto de datos con información adicional sobre el entorno de las propiedades.

4. Creación de un nuevo conjunto de datos

Se creó un nuevo conjunto de datos que incluye todas las transformaciones realizadas anteriormente. El conjunto de datos limpio y enriquecido se guardó en una base de datos SQLite y también se exportó como un archivo CSV para su uso en el siguiente ejercicio.

5. Encoding

Se realiza la codificación de variables utilizando la técnica de One-hot encoding. Se seleccionan todas las filas y columnas del conjunto de datos obtenido en la parte 1, excepto las columnas 'BuildingArea' y 'YearBuilt', que se imputarán más adelante.

Se codifican las variables categóricas y numéricas utilizando One-hot encoding. Para la variable categórica con una alta cantidad de categorías únicas, columna 'Suburb', se decidió no incluirla en el análisis debido a la cantidad exponencial de memoria requerida para el almacenamiento de la matriz densa. Aunque se pierde información relevante, se consideró una selección adecuada para este ejercicio.

Después de la codificación, se crea un nuevo conjunto de datos que combina el conjunto original con las características codificadas.

Imputación de valores faltantes por KNN

A continuación, se realizó la imputación de datos faltantes utilizando el algoritmo KNN en la matriz de datos codificada. Se seleccionan las columnas 'BuildingArea' y 'YearBuilt' del conjunto de datos original. En primer lugar se verificó si hay valores faltantes en las columnas a imputar.

Luego, se utiliza el algoritmo KNN (k vecinos más cercanos) para imputar los valores faltantes en las columnas 'BuildingArea' y 'YearBuilt' del conjunto de datos utilizando IterativeImputer y KNeighborsRegressor en las variables YearBuilt y BuildingArea del DataFrame df_final. Esto nos permite llenar los valores faltantes de manera más precisa utilizando información de los vecinos más cercanos.Al finalizar, las columnas 'YearBuilt' y 'BuildingArea' del conjunto de datos contendrán los valores imputados utilizando KNN.

Reducción de dimensionalidad

Para esto utilizamos la matriz de datos preprocesada obtenida para reducir la dimensionalidad del conjunto de datos mediante el método de Análisis de Componentes Principales (PCA). Para aplicar el PCA calculamos el número de componentes principales a obtener, siendo n el mínimo entre 20 y el número de filas de la matriz de datos.

Antes de aplicar PCA, nos preguntamos si es necesario estandarizar o escalar los datos. En este caso, optamos por estandarizar los datos utilizando la desviación estándar y la media de cada columna para asegurar que todas las variables estén en las mismas unidades y no tengan un peso desproporcionado.

Aplicamos PCA con el número de componentes especificado y ajustamos el modelo a los datos estandarizados. Obtenemos los componentes principales y la proporción de varianza explicada por cada componente.

Graficamos la varianza explicada acumulada por los primeros n componentes principales. Esto nos permite visualizar cuánta varianza del conjunto de datos se captura al considerar un número creciente de componentes principales. Basándonos en el gráfico de varianza explicada, seleccionamos las primeras m columnas de la matriz transformada obtenida a través de PCA. Estas columnas representan las nuevas características que añadiremos al conjunto de datos original.

Composición del resultado

Llevamos a cabo la transformación del conjunto de datos procesado en un objeto pandas.DataFrame y lo guardamos en un archivo. En primer lugar, se crea una lista llamada new_columns que contiene los nombres de las columnas del conjunto de datos original. Esto asegura que las nuevas columnas generadas a partir del PCA tengan los mismos nombres que las columnas originales, lo cual facilita la interpretación y el análisis de los resultados.

A continuación, se concatenan horizontalmente el conjunto de datos original (X) con la matriz transformada obtenida del PCA (X_reduced). Esto resulta en una nueva matriz X_pca que combina las columnas originales con las nuevas columnas generadas a partir del PCA. Después, se extiende la lista new_columns agregando los nombres de las nuevas columnas del PCA, en este caso, 'pca1', 'pca2' y 'pca3'.

Finalmente, se construye un nuevo DataFrame llamado processed_melb_df utilizando la matriz X_pca como datos y la lista new_columns como nombres de las columnas. Este DataFrame representa el conjunto de datos procesado, que incluye todas las columnas originales más las nuevas columnas generadas a partir del PCA.

En resumen, el permite agregar las nuevas columnas del PCA al conjunto de datos original, lo que nos brinda una representación resumida y más compacta de la información original. Esto puede ser útil para reducir la dimensionalidad de los datos y facilitar el análisis y la interpretación de los mismos.