Step-by-step

1. Adquisición de datos:

Se obtuvieron los datos de la plataforma **Properati**, que recopila información sobre propiedades inmobiliarias en Bogotá, Colombia. El conjunto de datos incluye 38,644 registros en formato CSV con características físicas, geográficas y descriptivas de las propiedades.

2. Limpieza de datos:

- Se identificaron valores faltantes en variables clave como surface_total, rooms y bathrooms.
- Los valores faltantes fueron imputados utilizando la media en las variables numéricas.
- Variables con alta proporción de valores nulos, como barrio (95% faltantes), fueron descartadas.
- Se revisaron y eliminaron registros duplicados para garantizar la calidad de los datos.

3. Exploración de los datos:

- Se realizaron análisis descriptivos y gráficos preliminares para entender la distribución de variables como price y surface_total.
- Se calcularon correlaciones entre las variables predictoras y el precio (price), identificando aquellas con mayor impacto potencial.
- Se exploraron las distribuciones geográficas de las propiedades y su relación con puntos clave como el centro financiero.

4. Construcción de nuevas variables:

- **Distancias geográficas:** Se calcularon las distancias de cada propiedad a hospitales, parques, estaciones de transporte y el centro financiero utilizando la función geopy.distance.geodesic.
- **Variables basadas en texto:** Se extrajeron menciones de palabras clave en las descripciones (description) para generar variables como num_parqueaderos y num_baños.
- **Variables dummy:** Se crearon indicadores binarios, como cerca_virrey, que señala propiedades cercanas al Virrey.

5. Selección y partición de datos:

- Se seleccionaron las variables relevantes para la modelación, excluyendo aquellas con baja correlación o alta cantidad de valores faltantes.
- También para seleccionar variables se usaron modelos de random forest para ver cuales eran las más importantes, complementando los análisis de las correlaciones
- Los datos se dividieron en un conjunto de entrenamiento (80%) y un conjunto de prueba (20%) para garantizar evaluaciones consistentes.

6. Entrenamiento de modelos:

- Se entrenaron los siguientes modelos:
    1. **Regresión Lineal:** Como referencia inicial.
    2. **Elastic Net:** Para manejar colinealidad y selección de características.
    3. **Random Forest:** Para capturar relaciones no lineales mediante árboles de decisión.
    4. **XGBoost:** Para ajustar errores secuenciales en las predicciones.
    5. **Redes Neuronales:** Para capturar relaciones más complejas entre las variables.

7. Optimización de hiperparámetros:

- Para Elastic Net, se utilizó validación cruzada para ajustar alpha y lambda.
- En Random Forest, se ajustaron el número de árboles (ntree) y variables por división (mtry).
- Para XGBoost, se realizó un Grid Search explorando combinaciones de n_estimators, learning_rate, y max_depth.
- Se uso adam para nn. Demás especificaciones en el codigo de nn.

8. Generación de predicciones:

- Cada modelo generó predicciones en el conjunto de prueba.
- Las predicciones se exportaron en archivos CSV con el formato requerido: columnas property_id y price.