Tutorial: Predicción de Precios de Vivienda en California

1. Configuración del Entorno

Primero, instalamos y cargamos las bibliotecas necesarias para trabajar con los datos y realizar análisis:

- pandas y numpy para la manipulación de datos.
- matplotlib y seaborn para visualización.
- scikit-learn para el modelado.
- xgboost para el uso de modelos avanzados de machine learning.

```
pip install numpy pandas seaborn matplotlib sklearn xgboost
```

2. Carga y Preprocesamiento de Datos

2.1 Cargando los Datos

Los datos provienen del conjunto California Housing Data. Este dataset incluye información sobre características de las viviendas y su ubicación geográfica. Se cargan los datos desde un archivo CSV:

```
import pandas as pd
data = pd.read_csv('california_housing_data.csv')
```

2.2 Tratamiento de Datos

Pasos realizados:

- Manejo de valores faltantes: Se identifican y eliminan valores NaN.
- Eliminación de duplicados: Se eliminan registros duplicados.

- Manejo de outliers: Utilizamos el método del rango intercuartílico (IQR) para identificar y eliminar valores atípicos en las variables continuas como el precio o el tamaño de las viviendas.

3. Visualización de los Datos

Generamos gráficos como:

- Diagramas de dispersión (scatter plots) para visualizar la relación entre características como el tamaño de la vivienda y su precio.
- Gráficos de correlación para ver cómo las variables numéricas están correlacionadas entre sí.

4. Modelado

4.1 División del conjunto de datos

Dividimos los datos en conjuntos de entrenamiento y prueba.

```
from sklearn.model_selection import train_test_split
X = data.drop('price', axis=1)
y = data['price']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

4.2 Entrenamiento de Modelos

Entrenamos varios modelos de machine learning, incluyendo:

- Regresión Lineal: Para establecer una línea base.
- XGBoost: Un modelo más avanzado basado en boosting de gradiente.

```
from xgboost import XGBRegressor
model = XGBRegressor()
```

```
model.fit(X_train, y_train)
```

4.3 Evaluación del Modelo

Se usan las métricas RMSE, MAE y R^2 para evaluar el desempeño:

```
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
y_pred = model.predict(X_test)

# Calcular las métricas
rmse = mean_squared_error(y_test, y_pred, squared=False)
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f'RMSE: {rmse}, MAE: {mae}, R²: {r2}')
```

5. Conclusiones

Este tutorial guía paso a paso la predicción de precios de vivienda en California, desde la carga y limpieza de los datos hasta la visualización, modelado y evaluación de los resultados.