```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

from google.colab import files
uploaded = files.upload()

df = pd.read_csv('Used_Bikes.csv')

df = df[['brand', 'bike_name', 'age', 'kms_driven', 'price', 'owner', 'power']]
df = df.dropna()

df = df.rename(columns={'kms_driven': 'used_hours'})

df['condition'] = df['owner'].map({
    'First Owner': 'excellent',
    'Second Owner': 'good',
    'Third Owner': 'fair',
    'Fourth Owner Or More': 'fair'
}).fillna('fair')

def get_type(name):
    name = name.lower()
    if 'sport' in name or 'r' in name or 'duke' in name or 'daytona' in name:
        return 'sport'
    elif 'classic' in name or 'bullet' in name or 'enfield' in name:
        return 'cruiser'
    elif 'apache' in name or 'fz' in name or 'pulsar' in name:
        return 'street'
    else:
        return 'other'

df['type'] = df['bike_name'].apply(get_type)

# Paso 5: Seleccionar variables
X = df[['brand', 'type', 'age', 'condition', 'used_hours']]
y = df['price']

# Paso 6: Preprocesamiento y modelo
categorical_features = ['brand', 'type', 'condition']
numerical_features = ['age', 'used_hours']

preprocessor = ColumnTransformer([
    ('cat', OneHotEncoder(handle_unknown='ignore'), categorical_features)
], remainder='passthrough')

model = Pipeline([
    ('preprocessing', preprocessor),
    ('regressor', LinearRegression())
])
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model.fit(X_train, y_train)
y_pred = model.predict(X_test)
print('MSE:', mean_squared_error(y_test, y_pred))
print('MAE:', mean_absolute_error(y_test, y_pred))
print('R2 Score:', r2_score(y_test, y_pred))

feature_names = model.named_steps['preprocessing'].get_feature_names_out()
coefficients = model.named_steps['regressor'].coef_

print("\nImportancia de variables:")
for name, coef in zip(feature_names, coefficients):
    print(f"{name}: {coef:.2f}")

#1. ¿Qué columnas contienen valores faltantes? ¿Qué criterio usaste para decidir cómo imputarlos?
#Las columnas con valores faltantes fueron cosas como power o kms_driven. Como eran pocas filas solo se elimino con dropna() para no complica
#2. ¿Detectaste valores atípicos en el precio o edad? ¿Qué hiciste con ellos?
#Si vi que había precios altos, pero como no eran muchos y algunos sí parecían reales por las marcas premium, Como no afectaban mucho decidí
#3. ¿Qué variables crees que tienen mayor influencia sobre el precio? ¿Cómo lo comprobaste?
```

```
#Las marcas caras como Harley-Davidson o Triumph suben mucho el precio por la edad y el uso. Lo comprobé viendo los coeficientes del modelo d
#5. ¿Cómo evaluaste el desempeño de tu modelo? ¿Qué métricas usaste y por qué?
#Usé las métricas MSE, MAE y R². El MAE me dio una idea clara del error promedio y el R² me dijo qué tanto el modelo explica del precio total
#6. ¿Cómo interpretas el coeficiente de regresión de la variable “edad”? ¿Tiene sentido su dirección?
#Sí tiene sentido, el coeficiente fue negativo lo cual dice que mientras más vieja es la moto más bajo su precio
#7. Si tuvieras más tiempo o datos, ¿qué harías para mejorar la precisión del modelo?
#Buscar más variables como el estado real de la moto, el mantenimiento o datos del motor, también podría probar otros modelos
#8. ¿Qué ventajas tendría usar otro tipo de modelo (por ejemplo, una red neuronal)?
#Podría captar relaciones más complejas entre variables, aunque creo que ahi ocuparia más datos y más tiempo para entrenarlo
#9. ¿Cuál fue el mayor reto en el proceso de limpieza de datos?
#Transformar datos como el del owner en una condición útil y crear la variable de tipo de moto desde el nombre
#10. ¿Qué aprendiste sobre el impacto de una buena preparación de datos en los resultados del modelo?
#Que si no limpias bien los datos, el modelo no aprende nada útil
```

Elegir archivos Used_Bikes.csv

- **Used_Bikes.csv**(text/csv) - 2493547 bytes, last modified: 28/5/2025 - 100% done

```
Saving Used_Bikes.csv to Used_Bikes (2).csv
MSE: 3555927247.4704266
MAE: 19866.25815329887
R2 Score: 0.5917164216816115

Importancia de variables:
cat__brand_BMW: 18842.12
cat__brand_Bajaj: -88275.14
cat__brand_Benelli: 23860.57
cat__brand_Ducati: 48568.43
cat__brand_Harley-Davidson: 317635.93
cat__brand_Hero: -100712.20
cat__brand_Honda: -90515.71
cat__brand_Hyosung: 23467.40
cat__brand_Ideal: 769.53
cat__brand_Indian: 1950.87
cat__brand_Jawa: 873.80
cat__brand_KTM: 18234.56
cat__brand_Kawasaki: 64901.73
cat__brand_LML: -168.12
cat__brand_MV: 12123.62
cat__brand_Mahindra: -7246.29
cat__brand_Rajdoot: 485.12
cat__brand_Royal Enfield: -44339.79
cat__brand_Suzuki: -82030.46
cat__brand_TVS: -108045.40
cat__brand_Triumph: 58050.18
cat__brand_Yamaha: -68918.01
cat__brand_Yezdi: 487.26
cat__type_cruiser: 487.26
cat__type_other: 1616.14
cat__type_sport: 5218.05
cat__type_street: -7321.45
cat__condition_excellent: -10396.71
cat__condition_fair: 6218.19
cat__condition_good: 4178.52
remainder__age: -4822.52
remainder__used_hours: -0.25
```

Comienza a programar o generar con IA.