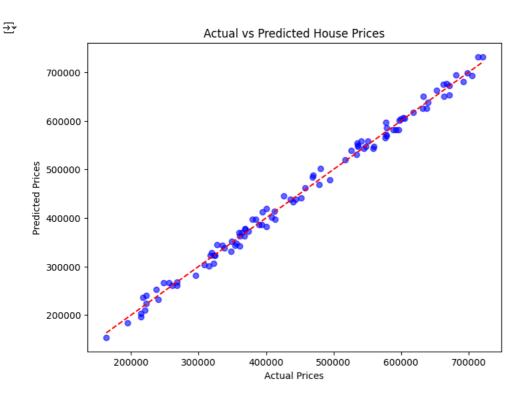
!pip -q install numpy pandas scikit-learn matplotlib joblib import numpy as np import pandas as pd import matplotlib.pyplot as plt from sklearn.model selection import train test split $from \ sklearn.preprocessing \ import \ StandardScaler$ from sklearn.linear_model import LinearRegression from sklearn.pipeline import Pipeline $from \ sklearn.metrics \ import \ r2_score, \ mean_absolute_error, \ mean_squared_error$ import joblib, json, os $\quad \hbox{from date time import date time} \\$ from google.colab import files uploaded = files.upload() CSV = list(uploaded.keys())[0] df = pd.read csv(CSV) Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable. rename_map = {} for c in df.columns: lc = c.strip().lower() if lc in ('sqft','sq.ft','sq_feet','square_feet','size'): rename_map[c]='square_feet' if lc in ('bed','beds','bedroom','bedrooms'): rename_map[c]='bedrooms' if lc in ('bath','baths','bathroom','bathrooms'): rename_map[c]='bathrooms' if lc in ('price','saleprice','sale_price','target'): rename_map[c]='price' if rename_map: df = df.rename(columns=rename_map) print('columns:', list(df.columns)) → columns: ['square_feet', 'bedrooms', 'bathrooms', 'price'] required = ['square_feet','bedrooms','bathrooms','price'] missing = [c for c in required if c not in df.columns] if missing: raise ValueError('Missing columns: ' + ', '.join(missing)) data = df[required].copy() data = data.dropna() data = data[(data.square_feet>0) & (data.price>0) & (data.bedrooms>=0) & (data.bathrooms>=0)] for col in required: low, high = data[col].quantile([0.01,0.99]) data[col] = data[col].clip(lower=low, upper=high) data = data.reset_index(drop=True) print('rows after clean:', data.shape[0]) → rows after clean: 500 from sklearn.model_selection import train_test_split X = df[['square_feet', 'bedrooms', 'bathrooms']] y = df['price'] X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42) from sklearn.linear model import LinearRegression model = LinearRegression() model.fit(X_train, y_train) y_pred = model.predict(X_test) from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score import numpy as np r2 = r2_score(y_test, y_pred) mae = mean_absolute_error(y_test, y_pred)



```
import joblib
joblib.dump(model, "linear_regression_house_price.joblib")
print("Model saved successfully!")
→ Model saved successfully!
import json
import pandas as pd
metrics = {
    "R2": round(r2, 4),
    "MAE": round(mae, 2),
    "RMSE": round(rmse, 2)
with open("metrics.json", "w") as f:
    json.dump(metrics, f)
cleaned_data = pd.concat([X, y], axis=1)
{\tt cleaned\_data.to\_csv("cleaned\_house\_prices.csv", index=False)}
print("Metrics and cleaned dataset saved!")
→ Metrics and cleaned dataset saved!
report = f"""
# House Price Prediction - Linear Regression
## Model Performance
- R<sup>2</sup> Score: {r2:.4f}
```

```
- Mean Absolute Error (MAE): {mae:.2f}
- Root Mean Squared Error (RMSE): {rmse:.2f}

## Notes
- Features used: square footage, bedrooms, bathrooms
- Target: House Price
"""

with open("REPORT.md", "w") as f:
    f.write(report)

print("Report saved successfully!")

→ Report saved successfully!
```