```
from sklearn.datasets import fetch_california_housing
import pandas as pd
import matplotlib.pyplot as plt

# Загружаем датасет
data = fetch_california_housing(as_frame=True)
df = data.frame  # pandas DataFrame

# Выводим первые строки
df.head()
```

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude	MedHouseVal
0	8.3252	41.0	6.984127	1.023810	322.0	2.555556	37.88	-122.23	4.526
1	8.3014	21.0	6.238137	0.971880	2401.0	2.109842	37.86	-122.22	3.585
2	7.2574	52.0	8.288136	1.073446	496.0	2.802260	37.85	-122.24	3.521
3	5.6431	52.0	5.817352	1.073059	558.0	2.547945	37.85	-122.25	3.413
4	3.8462	52.0	6.281853	1.081081	565.0	2.181467	37.85	-122.25	3.422

Далее: New interactive sheet

```
# Разделяем признаки (X) и целевую переменную (y)
X = df.drop('MedHouseVal', axis=1)
y = df['MedHouseVal']

# Разделяем на обучающую (80%) и тестовую (20%) выборки
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

print("Размер обучающей выборки:", X_train.shape)
print("Размер тестовой выборки:", X_test.shape)
```

```
Размер обучающей выборки: (16512, 8)
Размер тестовой выборки: (4128, 8)
```

```
from sklearn.linear_model import LinearRegression

# Создаём и обучаем модель
model = LinearRegression()
model.fit(X_train, y_train)

# Делаем предсказания на тестовой выборке
y_pred = model.predict(X_test)
```

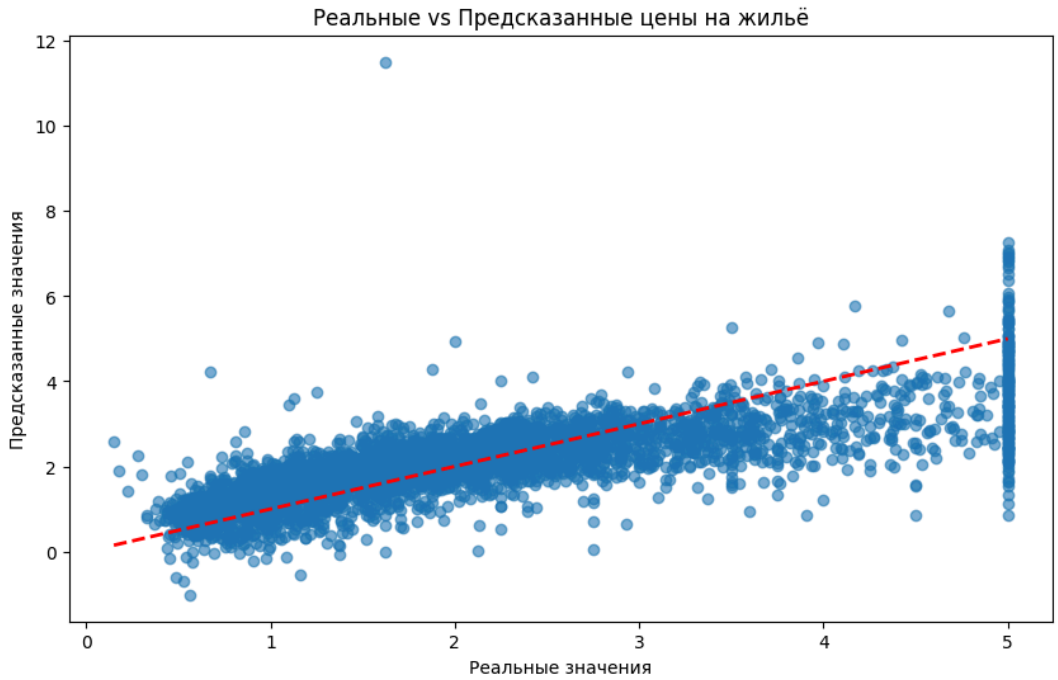
```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import numpy as np

mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

print(f"MAE (средняя абсолютная ошибка): {mae:.2f}")
print(f"RMSE (корень из MSE): {rmse:.2f}")
print(f"R² (коэффициент детерминации): {r2:.2f}")
```

```
MAE (средняя абсолютная ошибка): 0.53
RMSE (корень из MSE): 0.75
R² (коэффициент детерминации): 0.58
```

```
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred, alpha=0.6)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--', lw=2)
plt.xlabel('Реальные значения')
plt.ylabel('Предсказанные значения')
plt.title('Реальные vs Предсказанные цены на жильё')
plt.show()
```



Уровень 1: Практика Обучи модель на всех данных (без разбиения). Почему так делать нельзя в реальности? Посмотри, какие признаки сильнее всего влияют на предсказание:

```
coef_df = pd.DataFrame({'Признак': X.columns, 'Коэффициент': model.coef_})
print(coef_df.sort_values(by='Коэффициент', key=abs, ascending=False))
```

Уровень 2: Эксперименты Попробуй нормализовать признаки с помощью StandardScaler — улучшится ли R^2? Замени линейную регрессию на случайный лес (RandomForestRegressor) — сравни MAE и R^2.

Уровень 3: Творчество Возьми другой регрессионный датасет (например, boston — хотя он устарел, или diabetes из sklearn.datasets) и повтори весь пайплайн: загрузка → разбиение → обучение → оценка → визуализация.