

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.datasets import fetch_california_housing
```

```
# Load the dataset
data = fetch_california_housing()

# Convert it to a DataFrame for easier manipulation
df = pd.DataFrame(data.data, columns=data.feature_names)

# Add target (price) to the dataframe
df['Price'] = data.target

# Display the first few rows
print(df.head())
```

```

MedInc  HouseAge  AveRooms  AveBedrms  Population  AveOccup  Latitude  \
0  8.3252    41.0    6.984127   1.023810      322.0    2.555556    37.88
1  8.3014    21.0    6.238137   0.971880     2401.0    2.109842    37.86
2  7.2574    52.0    8.288136   1.073446      496.0    2.802260    37.85
3  5.6431    52.0    5.817352   1.073059      558.0    2.547945    37.85
4  3.8462    52.0    6.281853   1.081081      565.0    2.181467    37.85

Longitude  Price
0   -122.23   4.526
1   -122.22   3.585
2   -122.24   3.521
3   -122.25   3.413
4   -122.25   3.422
```

```
# Features (independent variables)
X = df[['MedInc', 'HouseAge', 'AveRooms', 'AveBedrms', 'Population', 'AveOccup']]

# Target (dependent variable)
y = df['Price']
```

```
# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
# Initialize and train the linear regression model
model = LinearRegression()
model.fit(X_train, y_train)
```

```

LinearRegression
LinearRegression()
```

```
# Make predictions on the test set
y_pred = model.predict(X_test)
```

```
# Calculate the Mean Squared Error (MSE)
mse = mean_squared_error(y_test, y_pred)
```

```
# Calculate the R2 score
r2 = r2_score(y_test, y_pred)
```

```
# Print the results
print(f'Mean Squared Error: {mse}')
print(f'R2 Score: {r2}')
```

```

Mean Squared Error: 0.642187231453487
R2 Score: 0.5099337366296416
```

```
# Model coefficients and intercept
print('Model Coefficients:')
for feature, coef in zip(X.columns, model.coef_):
    print(f'{feature}: {coef}')

# Intercept (bias term)
print(f'Intercept: {model.intercept_}')
```

```
↗ Model Coefficients:
MedInc: 0.5461607791074247
HouseAge: 0.016787909062568093
AveRooms: -0.2239199440047988
AveBedrms: 1.1154926114808392
Population: 2.3167197368202663e-05
AveOccup: -0.004618231345406933
Intercept: -0.5528727644615126
```

```
# Example: Predict the price for a house with specific features
new_house = np.array([[5.0, 30.0, 6.0, 1.0, 1000.0, 2.5]])
```

```
# Predict the price
predicted_price = model.predict(new_house)
```

```
print(f'Predicted Price for the new house: {f"${predicted_price[0] * 100000:,.2f}"'}')
```

```
↗ Predicted Price for the new house: $246,516.30
/usr/local/lib/python3.11/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but LinearRe
warnings.warn(
```

Start coding or [generate](#) with AI.