

```
In [1]: !pip install pandas numpy matplotlib scikit-learn
```

```
Requirement already satisfied: pandas in /home/sargam/.conda/envs/myenv/lib/python3.11/site-packages (2.2.3)
Requirement already satisfied: numpy in /home/sargam/.conda/envs/myenv/lib/python3.11/site-packages (2.0.1)
Requirement already satisfied: matplotlib in /home/sargam/.conda/envs/myenv/lib/python3.11/site-packages (3.10.1)
Requirement already satisfied: scikit-learn in /home/sargam/.conda/envs/myenv/lib/python3.11/site-packages (1.6.1)
Requirement already satisfied: python-dateutil>=2.8.2 in /home/sargam/.conda/envs/myenv/lib/python3.11/site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /home/sargam/.conda/envs/myenv/lib/python3.11/site-packages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in /home/sargam/.conda/envs/myenv/lib/python3.11/site-packages (from pandas) (2025.2)
Requirement already satisfied: contourpy>=1.0.1 in /home/sargam/.conda/envs/myenv/lib/python3.11/site-packages (from matplotlib) (1.3.2)
Requirement already satisfied: cycler>=0.10 in /home/sargam/.conda/envs/myenv/lib/python3.11/site-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /home/sargam/.conda/envs/myenv/lib/python3.11/site-packages (from matplotlib) (4.57.0)
Requirement already satisfied: kiwisolver>=1.3.1 in /home/sargam/.conda/envs/myenv/lib/python3.11/site-packages (from matplotlib) (1.4.8)
Requirement already satisfied: packaging>=20.0 in /home/sargam/.conda/envs/myenv/lib/python3.11/site-packages (from matplotlib) (24.2)
Requirement already satisfied: pillow>=8 in /home/sargam/.conda/envs/myenv/lib/python3.11/site-packages (from matplotlib) (11.2.1)
Requirement already satisfied: pyparsing>=2.3.1 in /home/sargam/.conda/envs/myenv/lib/python3.11/site-packages (from matplotlib) (3.2.3)
Requirement already satisfied: scipy>=1.6.0 in /home/sargam/.conda/envs/myenv/lib/python3.11/site-packages (from scikit-learn) (1.15.2)
Requirement already satisfied: joblib>=1.2.0 in /home/sargam/.conda/envs/myenv/lib/python3.11/site-packages (from scikit-learn) (1.5.0)
Requirement already satisfied: threadpoolctl>=3.1.0 in /home/sargam/.conda/envs/myenv/lib/python3.11/site-packages (from scikit-learn) (3.6.0)
Requirement already satisfied: six>=1.5 in /home/sargam/.conda/envs/myenv/lib/python3.11/site-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
Note: you may need to restart the kernel to use updated packages.
```

```
In [2]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.datasets import fetch_california_housing # Changed to Calif
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

```
In [4]: california = fetch_california_housing()
data = pd.DataFrame(california.data)

# Add feature names to the dataframe
data.columns = california.feature_names
data['PRICE'] = california.target
```

```
In [5]: data.isnull().sum()
```

```
Out[5]: MedInc      0
        HouseAge    0
        AveRooms    0
        AveBedrms   0
        Population  0
        AveOccup    0
        Latitude    0
        Longitude   0
        PRICE       0
        dtype: int64
```

```
In [6]: x = data.drop(['PRICE'], axis=1)
        y = data['PRICE']

        # Split data into training and testing datasets
        xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.2, rand
```

```
In [7]: lm = LinearRegression()
        model = lm.fit(xtrain, ytrain)

        # Predict for training and testing data
        ytrain_pred = lm.predict(xtrain)
        ytest_pred = lm.predict(xtest)
```

```
In [8]: df_train = pd.DataFrame({'Actual': ytrain, 'Predicted': ytrain_pred})
        df_test = pd.DataFrame({'Actual': ytest, 'Predicted': ytest_pred})

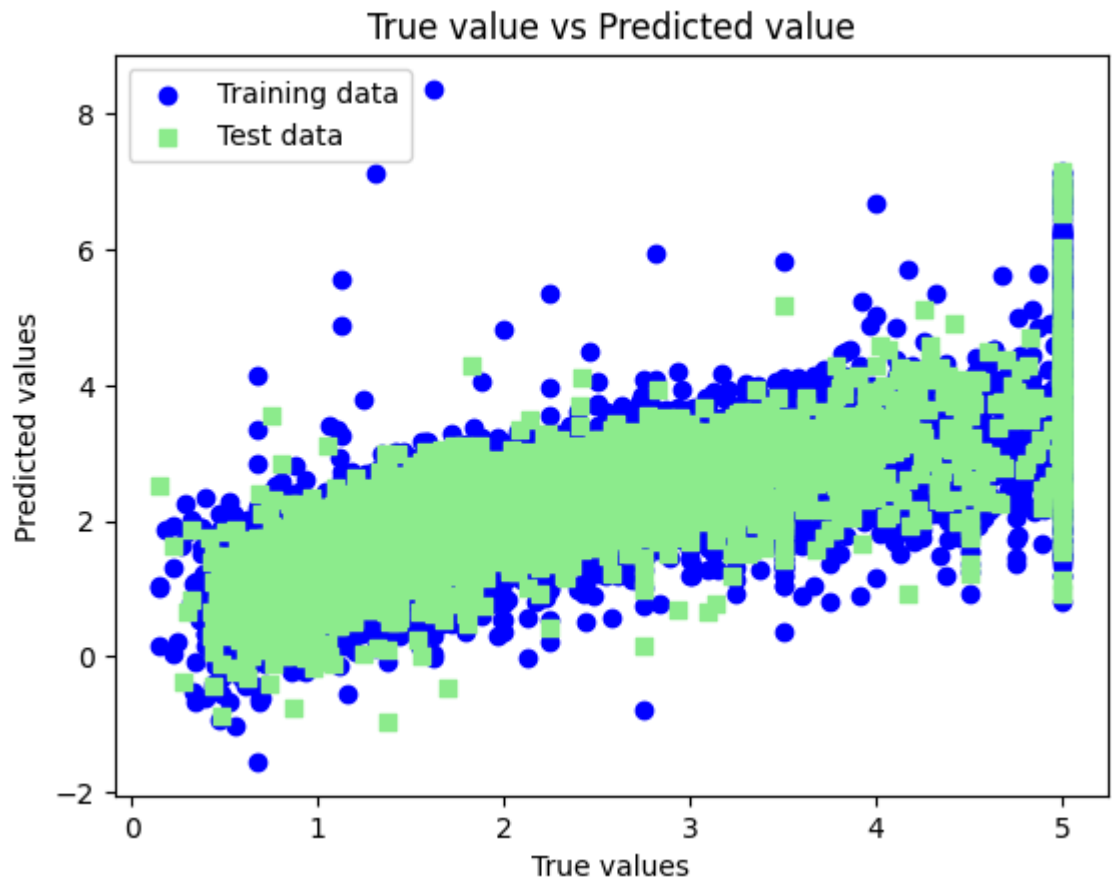
        # Calculate Mean Squared Error for both train and test sets
        mse_train = mean_squared_error(ytrain, ytrain_pred)
        mse_test = mean_squared_error(ytest, ytest_pred)
```

```
In [9]: print(f"Training Mean Squared Error: {mse_train}")
        print(f"Testing Mean Squared Error: {mse_test}")
```

Training Mean Squared Error: 0.5234413607125449

Testing Mean Squared Error: 0.5289841670367219

```
In [10]: plt.scatter(ytrain, ytrain_pred, c='blue', marker='o', label='Training da
plt.scatter(ytest, ytest_pred, c='lightgreen', marker='s', label='Test da
plt.xlabel('True values')
plt.ylabel('Predicted values')
plt.title("True value vs Predicted value")
plt.legend(loc='upper left')
plt.show()
```



```
In [11]: r2_train = r2_score(ytrain, ytrain_pred)
r2_test = r2_score(ytest, ytest_pred)
print(f"R2 Score for Training Data: {r2_train}")
print(f"R2 Score for Testing Data: {r2_test}")
```

R² Score for Training Data: 0.6088968118672868
R² Score for Testing Data: 0.5943232652466195

In []: