

```
pip install pandas numpy scikit-learn flask
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (2.2.2)
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (2.0.2)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.11/dist-packages (1.6.1)
Requirement already satisfied: flask in /usr/local/lib/python3.11/dist-packages (3.1.0)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.1)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.1)
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (1.14.1)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (3.6.0)
Requirement already satisfied: Werkzeug>=3.1 in /usr/local/lib/python3.11/dist-packages (from flask) (3.1.3)
Requirement already satisfied: Jinja2>=3.1.2 in /usr/local/lib/python3.11/dist-packages (from flask) (3.1.6)
Requirement already satisfied: itsdangerous>=2.2 in /usr/local/lib/python3.11/dist-packages (from flask) (2.2.0)
Requirement already satisfied: click>=8.1.3 in /usr/local/lib/python3.11/dist-packages (from flask) (8.1.8)
Requirement already satisfied: blinker>=1.9 in /usr/local/lib/python3.11/dist-packages (from flask) (1.9.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.11/dist-packages (from Jinja2>=3.1.2->flask) (3.0.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
```

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error
import pickle

# Load dataset
from sklearn.datasets import load_boston
boston = load_boston("C:\\Users\\kkata\\Downloads")
df = pd.DataFrame(boston.data, columns=boston.feature_names)
df['PRICE'] = boston.target

# Show first few rows of the dataset
print(df.head())

# Split data into features (X) and target variable (y)
X = df.drop(columns=['PRICE'])
y = df['PRICE']

# Split 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)

# Train a Linear Regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Evaluate the model
y_pred = model.predict(X_test)
error = mean_absolute_error(y_test, y_pred)
print(f'Mean Absolute Error: {error}')

# Save the trained model
with open("house_price_model.pkl", "wb") as f:
    pickle.dump(model, f)
```



```

-----
ImportError                                Traceback (most recent call last)
<ipython-input-4-3096476512aa> in <cell line: 0>()
      7
      8 # Load dataset
----> 9 from sklearn.datasets import load_boston
     10 boston = load_boston("C:\\Users\\kkata\\Downloads")
     11 df = pd.DataFrame(boston.data, columns=boston.feature_names)

/usr/local/lib/python3.11/dist-packages/sklearn/datasets/_init_.py in __getattr__(name)
     159         """
     160     )
--> 161         raise ImportError(msg)
     162     try:
     163         return globals()[name]

```

ImportError:
`load_boston` has been removed from scikit-learn since version 1.2.

The Boston housing prices dataset has an ethical problem: as investigated in [1], the authors of this dataset engineered a non-invertible variable "B" assuming that racial self-segregation had a positive impact on house prices [2]. Furthermore the goal of the research that led to the creation of this dataset was to study the impact of air quality but it did not give adequate demonstration of the validity of this assumption.

The scikit-learn maintainers therefore strongly discourage the use of this dataset unless the purpose of the code is to study and educate about ethical issues in data science and machine learning.

In this special case, you can fetch the dataset from the original source::

```

import pandas as pd
import numpy as np

data_url = "http://lib.stat.cmu.edu/datasets/boston"
raw_df = pd.read_csv(data_url, sep="\s+", skiprows=22, header=None)
data = np.hstack([raw_df.values[::2, :], raw_df.values[1::2, :2]])
target = raw_df.values[1::2, 2]

```

Alternative datasets include the California housing dataset and the Ames housing dataset. You can load the datasets as follows::

```

from sklearn.datasets import fetch_california_housing
housing = fetch_california_housing()

```

for the California housing dataset and::

```

from sklearn.datasets import fetch_openml
housing = fetch_openml(name="house_prices", as_frame=True)

```

for the Ames housing dataset.

[1] M Carlisle.
"Racist data destruction?"
<https://medium.com/@docintangible/racist-data-destruction-113e3eff54a8>>

[2] Harrison Jr, David, and Daniel L. Rubinfeld.
"Hedonic housing prices and the demand for clean air."
Journal of environmental economics and management 5.1 (1978): 81-102.
https://www.researchgate.net/publication/4974606_Hedonic_housing_prices_and_the_demand_for_clean_air>

NOTE: If your import is failing due to a missing package, you can manually install dependencies using either `!pip` or `!apt`.

To view examples of installing some common dependencies, click the "Open Examples" button below.

OPEN EXAMPLES

Next steps: [Explain error](#)

```

import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

```

```

from sklearn.metrics import mean_absolute_error
import pickle

# Load dataset using fetch_california_housing instead of load_boston
from sklearn.datasets import fetch_california_housing
housing = fetch_california_housing()
df = pd.DataFrame(housing.data, columns=housing.feature_names)
df['PRICE'] = housing.target # Assuming 'PRICE' is the target variable name

# Show first few rows of the dataset
print(df.head())

# Split data into features (X) and target variable (y)
X = df.drop(columns=['PRICE'])
y = df['PRICE']

# Split 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)

# Train a Linear Regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Evaluate the model
y_pred = model.predict(X_test)
error = mean_absolute_error(y_test, y_pred)
print(f'Mean Absolute Error: {error}')

# Save the trained model
with open("house_price_model.pkl", "wb") as f:
    pickle.dump(model, f)

```

```

↩
0  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
Mean Absolute Error: 0.5332001304956553

```

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