**Cell 1**

python

CopyEdit

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

**Problem Step**: Import required libraries.

**Definition**:

* pandas: For loading and manipulating data.
* train\_test\_split: Splits data into training and testing sets.
* LinearRegression: A machine learning model for regression analysis.
* mean\_squared\_error, r2\_score: Metrics to evaluate model performance.

**Why**:

* These libraries provide tools to clean data, build a regression model, and evaluate its accuracy.

**Cell 2**

python

CopyEdit

df = pd.read\_csv('datasets/HousingData.csv')

**Problem Step**: Load the housing dataset.

**Definition**:

* pd.read\_csv(): Loads CSV data into a DataFrame.

**Why**:

* Required to begin data analysis and modeling.

**Cell 3**

python

CopyEdit

df.head()

**Problem Step**: Preview the first few rows of data.

**Definition**:

* df.head(): Displays the first 5 rows.

**Why**:

* Quick check to understand what data looks like.

**Cell 4**

python

CopyEdit

df.info()

**Problem Step**: Check data types and missing values.

**Definition**:

* df.info(): Shows column types and non-null counts.

**Why**:

* Helps detect missing data and understand column types.

**Cell 5**

python

CopyEdit

def handle\_missing(name):

df[name].fillna(df[name].mean(), inplace=True)

**Problem Step**: Define a function to fill missing values.

**Definition**:

* fillna(mean): Fills missing values with column's mean.

**Why**:

* Replaces missing data to maintain model integrity.

**Cell 6**

python

CopyEdit

lis = ['CRIM', 'ZN', 'INDUS','CHAS','AGE','LSTAT']

for i in lis:

handle\_missing(i)

**Problem Step**: Apply missing value handler to specific columns.

**Definition**:

* Loops over selected columns and fills their missing values.

**Why**:

* Prepares the dataset for modeling by ensuring no null values in critical features.

**Cell 7**

python

CopyEdit

df.info()

**Problem Step**: Verify missing data is handled.

**Definition**:

* Checks updated data summary.

**Why**:

* Confirms that columns are now complete with no missing values.

**Cell 8**

python

CopyEdit

X = df.iloc[:,0:-1]

y = df.iloc[:,-1]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y,test\_size=0.2, random\_state=42)

**Problem Step**: Split data into features and target; then into training and testing sets.

**Definition**:

* iloc[:,0:-1]: All columns except last as features.
* iloc[:,-1]: Last column as target.
* train\_test\_split(): Randomly splits data into train (80%) and test (20%) sets.

**Why**:

* Prepares data for training and validating the model.

**Cell 9**

python

CopyEdit

model = LinearRegression()

**Problem Step**: Create regression model object.

**Definition**:

* Initializes a linear regression model.

**Why**:

* The model will learn from training data to make predictions.

**Cell 10**

python

CopyEdit

model.fit(X\_train, y\_train)

**Problem Step**: Train the regression model.

**Definition**:

* fit(): Trains the model on input (X) and output (y) data.

**Why**:

* Allows the model to learn relationships in the training data.

**Cell 11**

python

CopyEdit

y\_pred = model.predict(X\_test)

**Problem Step**: Make predictions on test data.

**Definition**:

* predict(): Uses trained model to predict outputs for new inputs.

**Why**:

* Evaluates model performance on unseen data.

**Cell 12**

python

CopyEdit

mean\_squared\_error(y\_test, y\_pred)

**Problem Step**: Evaluate prediction error.

**Definition**:

* Calculates average squared difference between predicted and actual values.

**Why**:

* Measures how close predictions are to actual results (lower is better).

**Cell 13**

python

CopyEdit

r2\_score(y\_test, y\_pred)

**Problem Step**: Measure prediction accuracy.

**Definition**:

* Returns R² score (coefficient of determination), where 1.0 is perfect fit.

**Why**:

* Indicates how well the model explains the variance in the target variable.