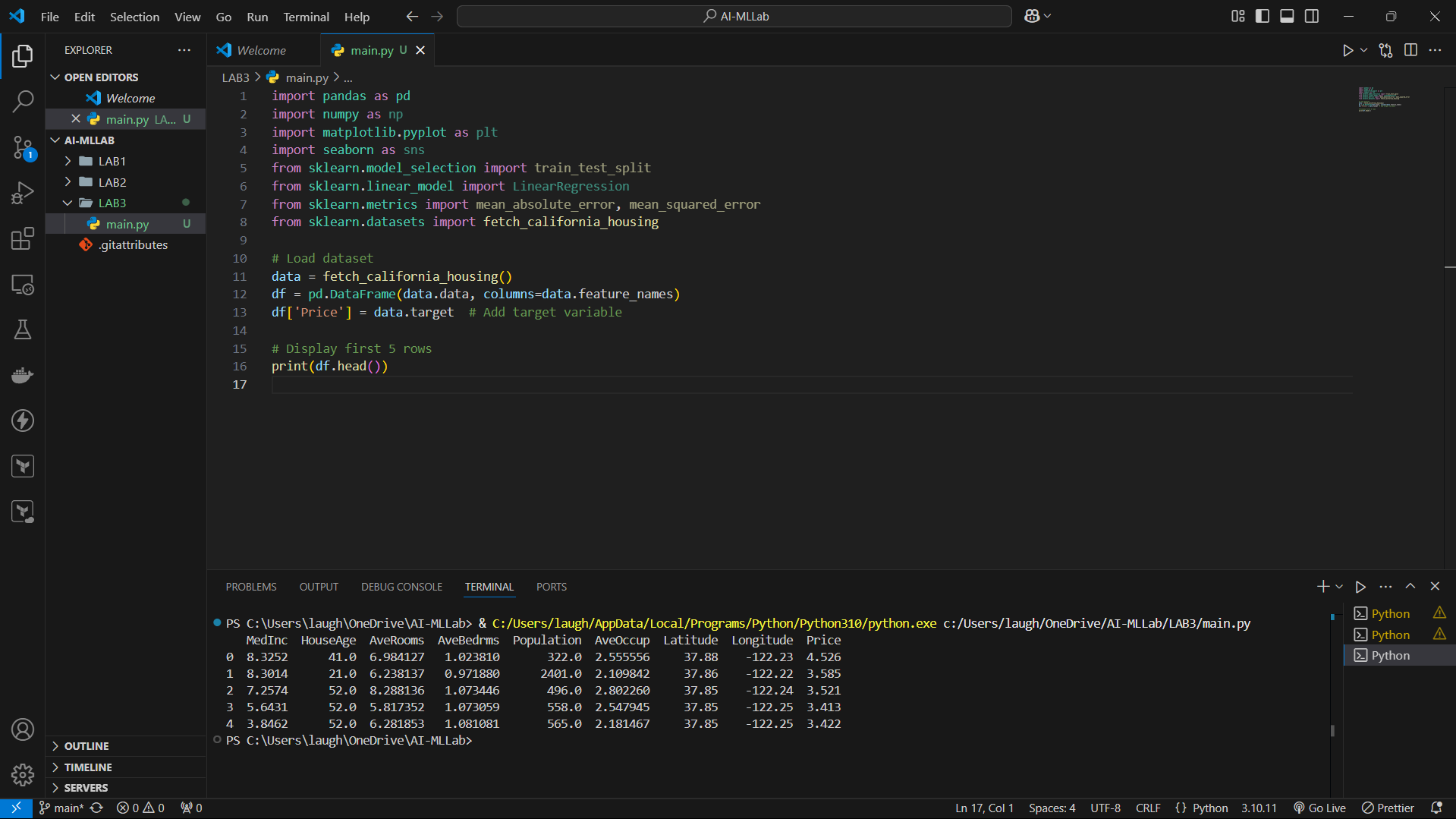
First install necessary libraries  
pip install pandas numpy scikit-learn matplotlib seaborn  
Ctrl shift P in VSCode and choose the interpreter with sci-kit learn  
To check where: python -m pip show scikit-learn

  
  
Using **California Housing Data** from sklearn.datasets

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error

from sklearn.datasets import fetch\_california\_housing

# Load dataset

data = fetch\_california\_housing()

df = pd.DataFrame(data.data, columns=data.feature\_names)

df['Price'] = data.target  # Add target variable

# Display first 5 rows

print(df.head())

# Data Preprocessing

print(df.isnull().sum())  # Check for missing values

print(df.describe())      # Summary statistics

df = df.dropna()  # Drop missing values

# Exploratory Data Analysis

sns.pairplot(df)

plt.show()

# Correlation heatmap

plt.figure(figsize=(10,6))

sns.heatmap(df.corr(), annot=True, cmap='coolwarm')

plt.show()

# Splitting data into training and testing Sets

X = df.drop(columns=['Price'])  # Features

y = df['Price']  # Target variable

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

# Train a Machine Learning Model (Linear Regression)

model = LinearRegression()

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

# Model Evaluation

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

mae = mean\_absolute\_error(y\_test, y\_pred)

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

rmse = np.sqrt(mse)

print(f"Mean Absolute Error (MAE): {mae}")

print(f"Mean Squared Error (MSE): {mse}")

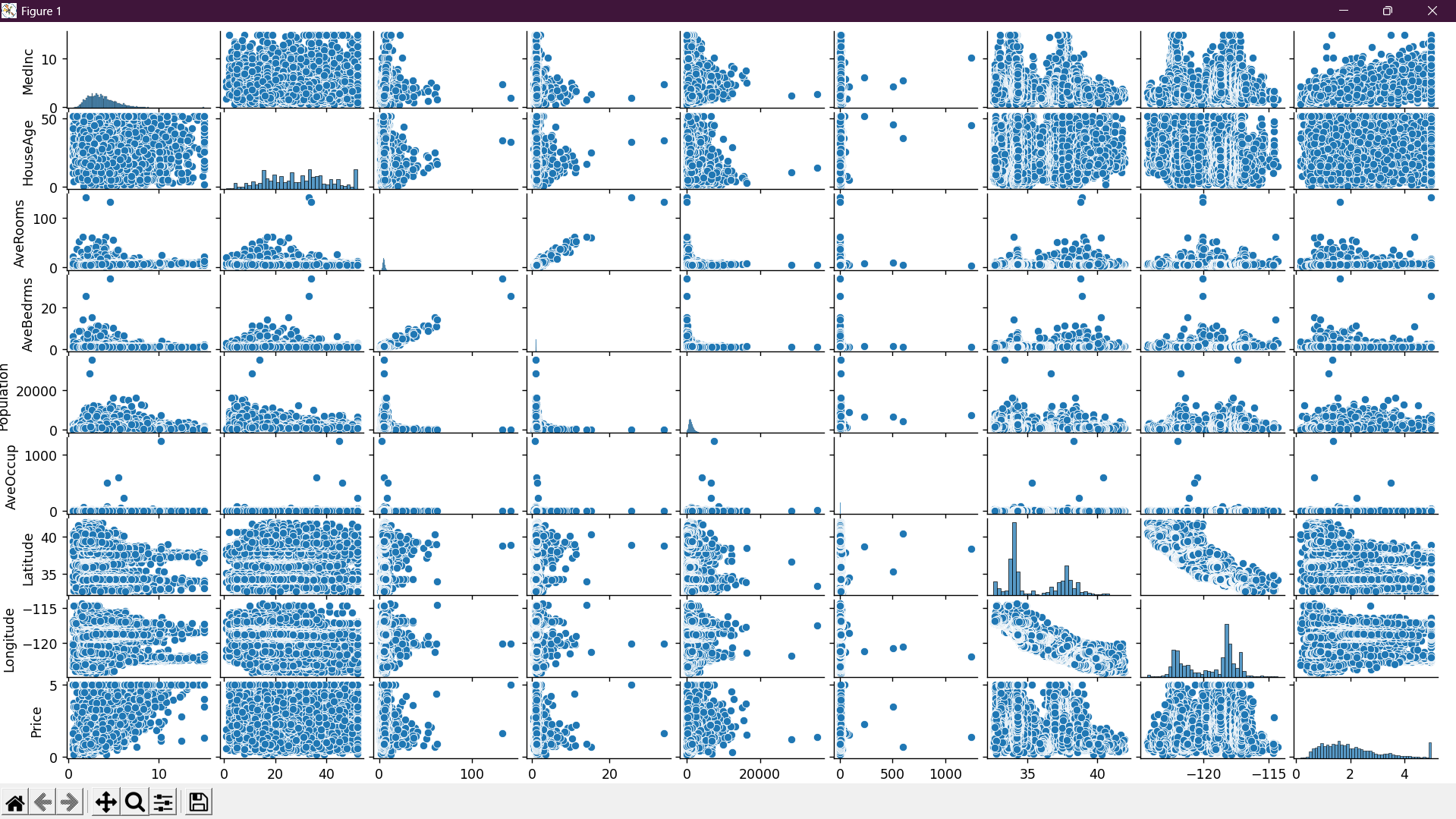
print(f"Root Mean Squared Error (RMSE): {rmse}")

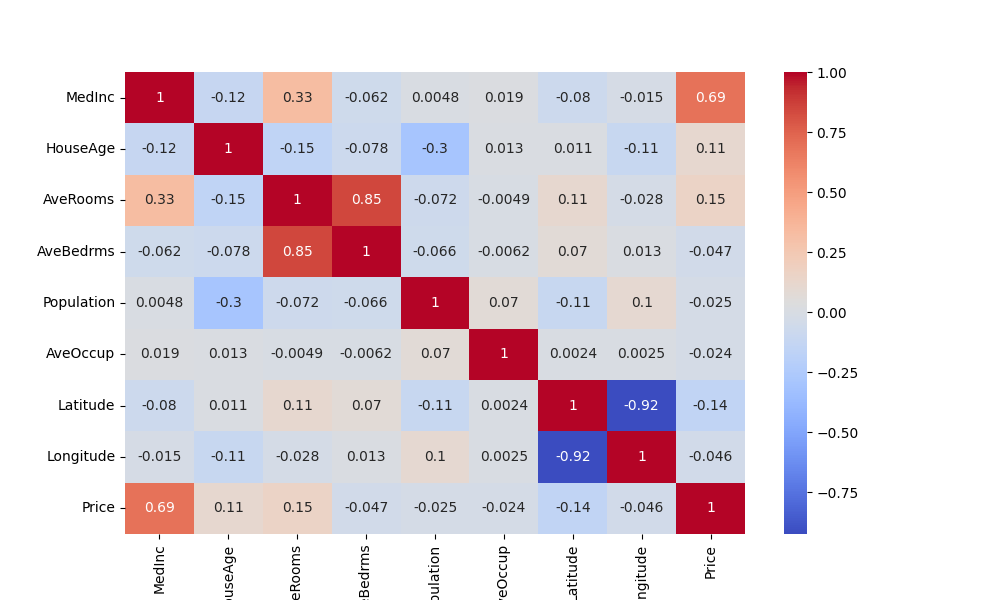
# Make Predictions

sample\_house = X\_test.iloc[0:1]  # Select a sample house

predicted\_price = model.predict(sample\_house)

print(f"Predicted House Price: {predicted\_price[0]}")





This is a **correlation heatmap** that shows how strongly each feature is related to the **house price (Price)** and to other features in the dataset

**Color Meaning**

* **Red (Closer to 1.0)** → Strong **positive correlation** (as one feature increases, the other also increases).
* **Blue (Closer to -1.0)** → Strong **negative correlation** (as one feature increases, the other decreases).
* **Near 0** → No significant correlation

**Key Observations for Price**

* **MedInc (0.69)** → **Strongest positive correlation**
  + Higher median income is strongly linked to higher house prices.
* **AveRooms (0.15)** → Weak positive correlation.
  + More rooms might lead to higher prices, but the effect is small.
* **HouseAge (0.11)** → Very weak positive correlation.
  + Older houses do not significantly affect prices.
* **Latitude (-0.14) & Longitude (-0.046)** → Weak correlation.
  + Location has some effect, but it’s not a dominant factor

**Feature Relationships**

* **AveRooms & AveBedrms (0.85)** → High multicollinearity
  + This means the number of rooms and bedrooms are highly related, so one can be removed to avoid redundancy.
* **Latitude & Longitude (-0.92)** → Strong inverse correlation
  + This likely reflects California’s geographic layout

