Import necessary libraries

```
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, mean_absolute_error, r2_score
```

Load your dataset

Replace 'your_dataset.csv' with the actual path to your dataset

	<pre>data = pd.read_csv("predicting_house_prices.csv") data</pre>						
	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	Addres
	0 79545.45857	5.682861	7.009188	4.09	23086.80050	1.059034e+06	208 Michael Ferry Ap 674\nLaurabury, NE 3701.
	1 79248.64245	6.002900	6.730821	3.09	40173.07217	1.505891e+06	188 Johnson Views Sui 079\nLake Kathleen, CA
:	2 61287.06718	5.865890	8.512727	5.13	36882.15940	1.058988e+06	9127 Elizabe Stravenue\nDanieltown, \ 06482
	3 63345.24005	7.188236	5.586729	3.26	34310.24283	1.260617e+06	USS Barnett\nFPO AP 4482
	4 59982.19723	5.040555	7.839388	4.23	26354.10947	6.309435e+05	USNS Raymond\nFPO A 0938
499	5 60567.94414	7.830362	6.137356	3.46	22837.36103	1.060194e+06	USNS Williams\nFPO A 30153-76
499	6 78491.27543	6.999135	6.576763	4.02	25616.11549	1.482618e+06	PSC 9258, Box 8489\nAPO A 42991-33
499	7 63390.68689	7.250591	4.805081	2.13	33266.14549	1.030730e+06	4215 Tracy Garden Su 076\nJoshualand, VA 01
499	8 68001.33124	5.534388	7.130144	5.44	42625.62016	1.198657e+06	USS Wallace\nFPO AE 733
499	9 65510.58180	5.992305	6.792336	4.07	46501.28380	1.298950e+06	37778 George Ridges A 509\nEast Holly, NV 2

Define the features (independent variables) and the target variable (house prices)

```
In [24]: X = data[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms', 'Avg. Area Number of Bedrooms
         y = data['Price']
         У
            1.059034e+06
Out[24]:
                1.505891e+06
                1.058988e+06
         2
         3
                1.260617e+06
                6.309435e+05
               1.060194e+06
         4995
         4996
                1.482618e+06
         4997
               1.030730e+06
         4998
                1.198657e+06
         4999
                 1.298950e+06
         Name: Price, Length: 5000, dtype: float64
```

Split the data into training and testing sets

```
In [25]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Create and train a Linear Regression model

```
Tn [261: model = LinearRegression()
```

```
model.fit(X_train, y_train)

Out[26]: 

✓ LinearRegression

LinearRegression()
```

Make predictions

```
In [27]: y_pred = model.predict(X_test)
```

Evaluate the model

R-squared: 0.917997170698532

```
In [29]: mse = mean_squared_error(y_test, y_pred)
    mae = mean_absolute_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)
```

Print the evaluation metrics

```
In [30]: print("Mean Squared Error:", mse)
print("Mean Absolute Error:", mae)
print("R-squared:", r2)

Mean Squared Error: 10089009299.499422
Mean Absolute Error: 80879.09722218302
```

You can also examine the model coefficients to see feature importance

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js