1 Practical Lab 1: Univariate Linear Regression on California Housing Prices

Author: Shiru **Course:** Machine Learning Foundations
Repository: https://github.com/Jasminekite/ml-practical-labs.git

2 Problem Statement

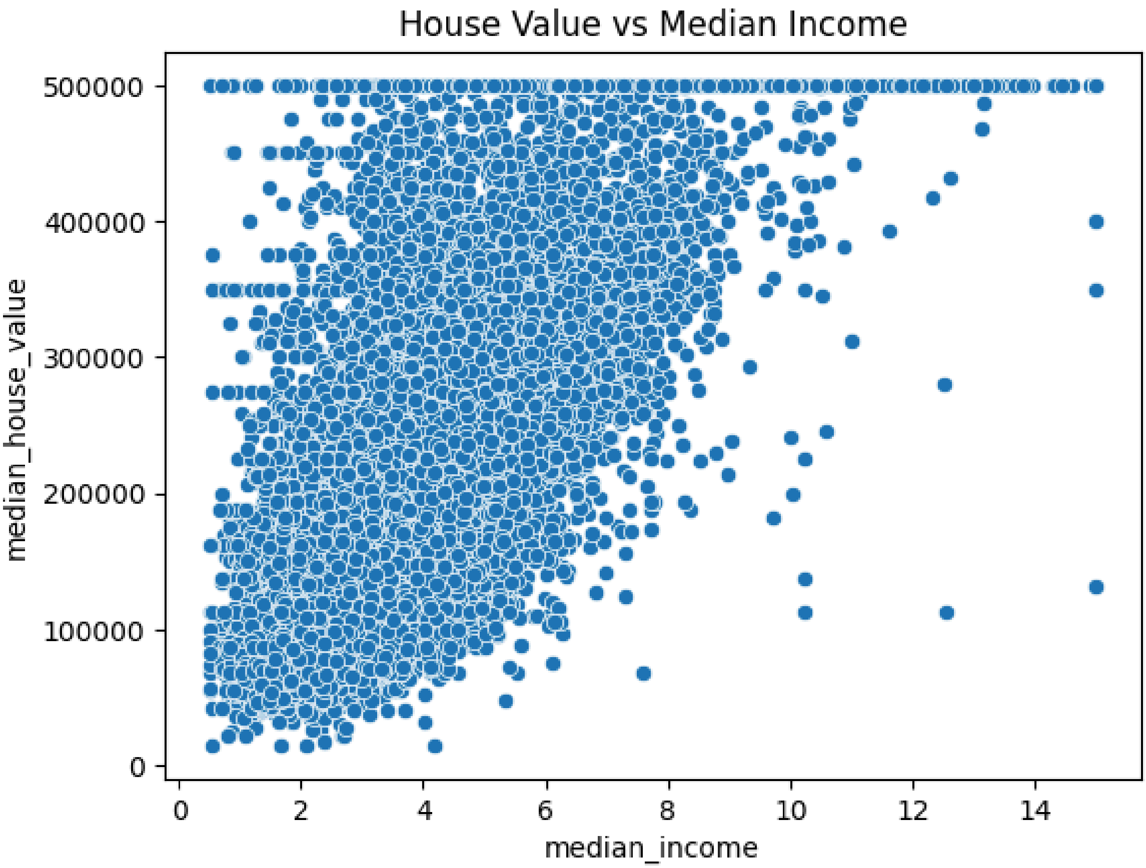
The goal of this lab is to train three univariate linear regression models to predict the median house value in California based on one independent variable at a time: median income, population, and number of households.

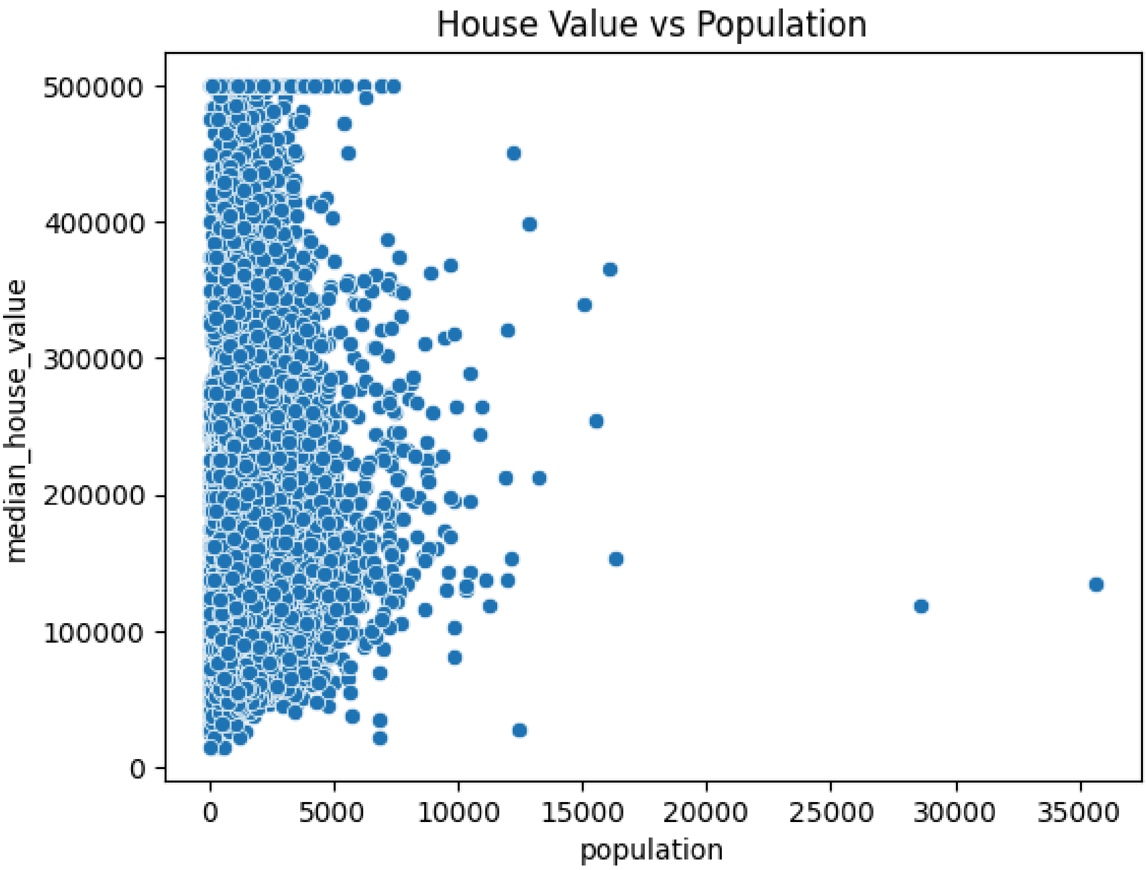
In [24]:
```
## Getting the data
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Link to data
url = "https://raw.githubusercontent.com/ageron/handson-ml/master/datasets/housi
df = pd.read_csv(url)
print(df)
```

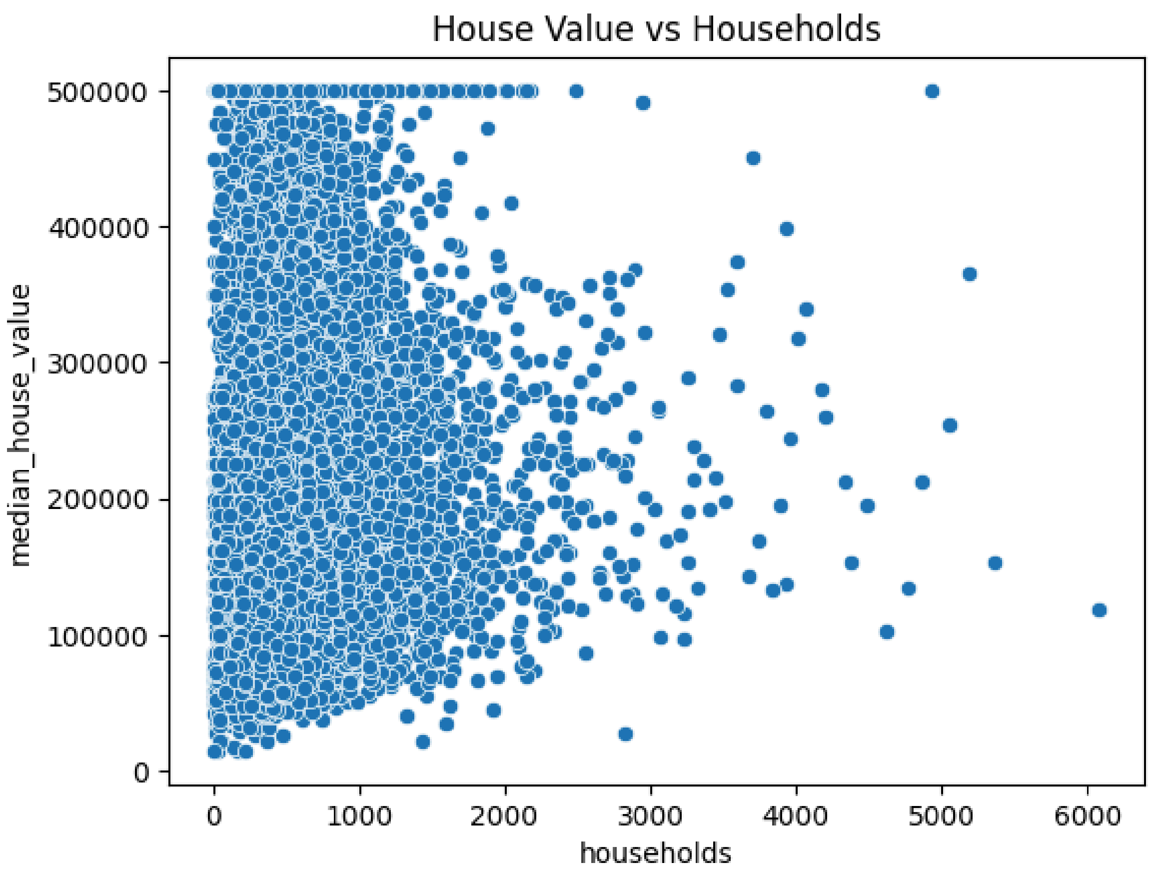
```
       longitude  latitude  housing_median_age  total_rooms  total_bedrooms  \
0        -122.23     37.88                41.0        880.0           129.0   
1        -122.22     37.86                21.0       7099.0          1106.0   
2        -122.24     37.85                52.0       1467.0           190.0   
3        -122.25     37.85                52.0       1274.0           235.0   
4        -122.25     37.85                52.0       1627.0           280.0   
...          ...       ...                 ...          ...             ...   
20635    -121.09     39.48                25.0       1665.0           374.0   
20636    -121.21     39.49                18.0        697.0           150.0   
20637    -121.22     39.43                17.0       2254.0           485.0   
20638    -121.32     39.43                18.0       1860.0           409.0   
20639    -121.24     39.37                16.0       2785.0           616.0   

       population  households  median_income  median_house_value  \
0           322.0       126.0         8.3252            452600.0   
1          2401.0      1138.0         8.3014            358500.0   
2           496.0       177.0         7.2574            352100.0   
3           558.0       219.0         5.6431            341300.0   
4           565.0       259.0         3.8462            342200.0   
...           ...         ...            ...                 ...   
20635       845.0       330.0         1.5603             78100.0   
20636       356.0       114.0         2.5568             77100.0   
20637      1007.0       433.0         1.7000             92300.0   
20638       741.0       349.0         1.8672             84700.0   
20639      1387.0       530.0         2.3886             89400.0   

      ocean_proximity  
0            NEAR BAY  
1            NEAR BAY  
2            NEAR BAY  
3            NEAR BAY  
4            NEAR BAY  
...               ...  
20635          INLAND  
20636          INLAND  
20637          INLAND  
20638          INLAND  
20639          INLAND  

[20640 rows x 10 columns]
```

In [25]:

```python
#EDA (Exploratory Data Analysis)
# Summary Statistics
df.describe()

# Scatter Plots
sns.scatterplot(data=df, x='median_income', y='median_house_value').set_title('H
plt.show()

sns.scatterplot(data=df, x='population', y='median_house_value').set_title('Hous
plt.show()

sns.scatterplot(data=df, x='households', y='median_house_value').set_title('Hous
plt.show()
```


House Value vs Median Income
median_house_value
median_income
0
100000
200000
300000
400000
500000
0
2
4
6
8
10
12
14


House Value vs Population
median_house_value
population
0
100000
200000
300000
400000
500000
0
5000
10000
15000
20000
25000
30000
35000



Text based Insights

- There is a strong positive correlation between `median_income` and `median_house_value`.
- Population and households show weaker correlations.

In [16]:
```
# 5. Linear Regression Models
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error

def run_regression(X_col):
    X = df[[X_col]]
    y = df["median_house_value"]
    model = LinearRegression()
    model.fit(X, y)
    y_pred = model.predict(X)
    mse = mean_squared_error(y, y_pred)
    mae = mean_absolute_error(y, y_pred)
    intercept = model.intercept_
    slope = model.coef_[0]
    return intercept, slope, mse, mae, y_pred

results = {}
for feature in ['median_income', 'population', 'households']:
    results[feature] = run_regression(feature)
```

In [17]:
```
results = {}
predictions = {}
```

```python
for feature in ['median_income', 'population', 'households']:
    intercept, slope, mse, mae, y_pred = run_regression(feature)
    results[feature] = [intercept, slope, mse, mae]  # only 4 values
    predictions[feature] = y_pred  # store predictions separately if needed late
```

In [18]:

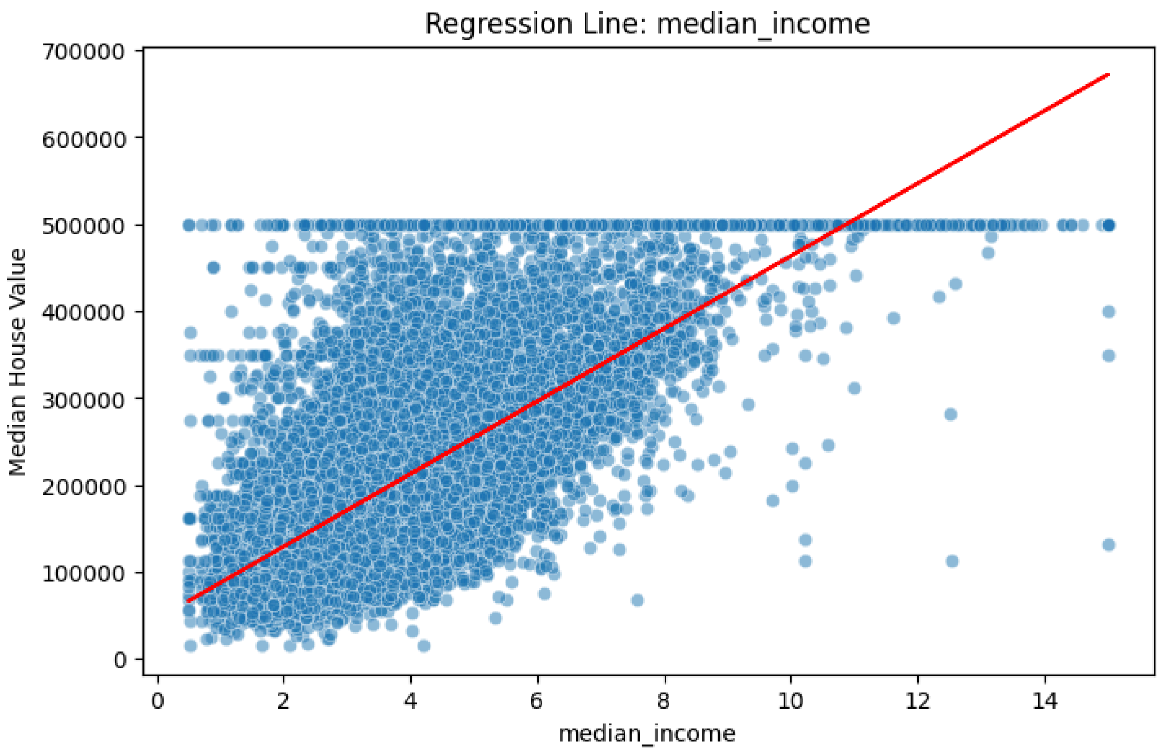
```python
# 6. Regression Table
table = pd.DataFrame(results, index=["Intercept", "Slope", "MSE", "MAE"]).T
table
```

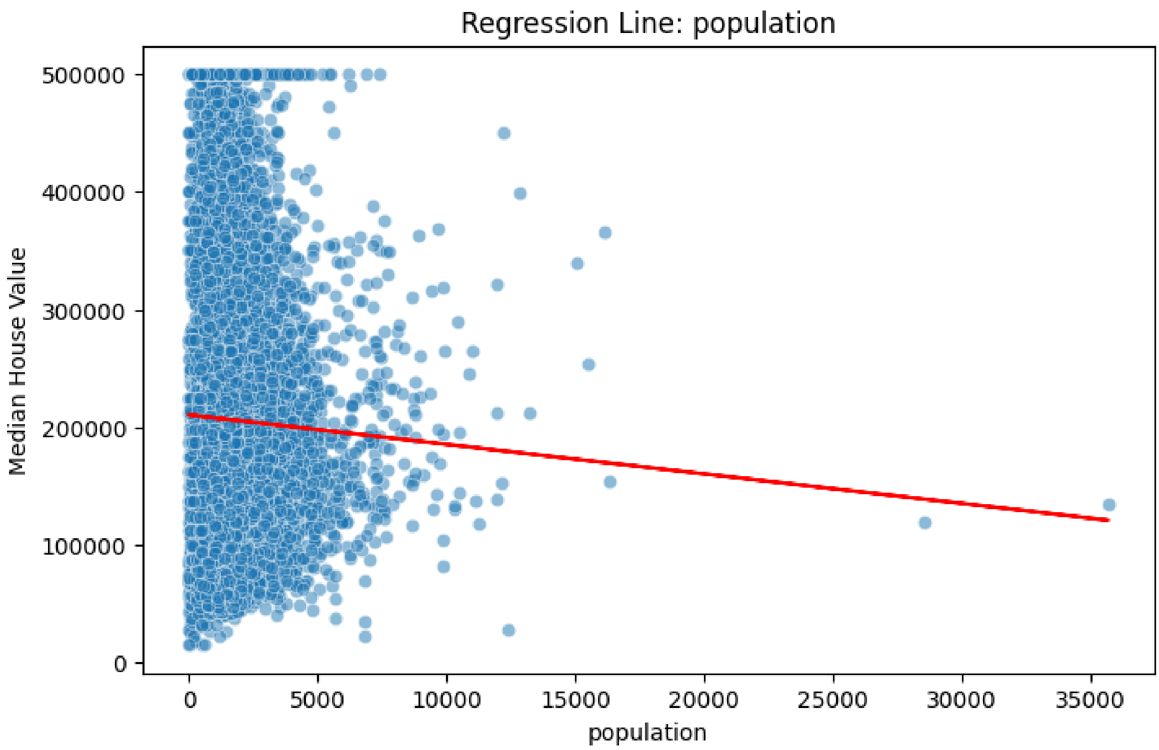
Out[18]:

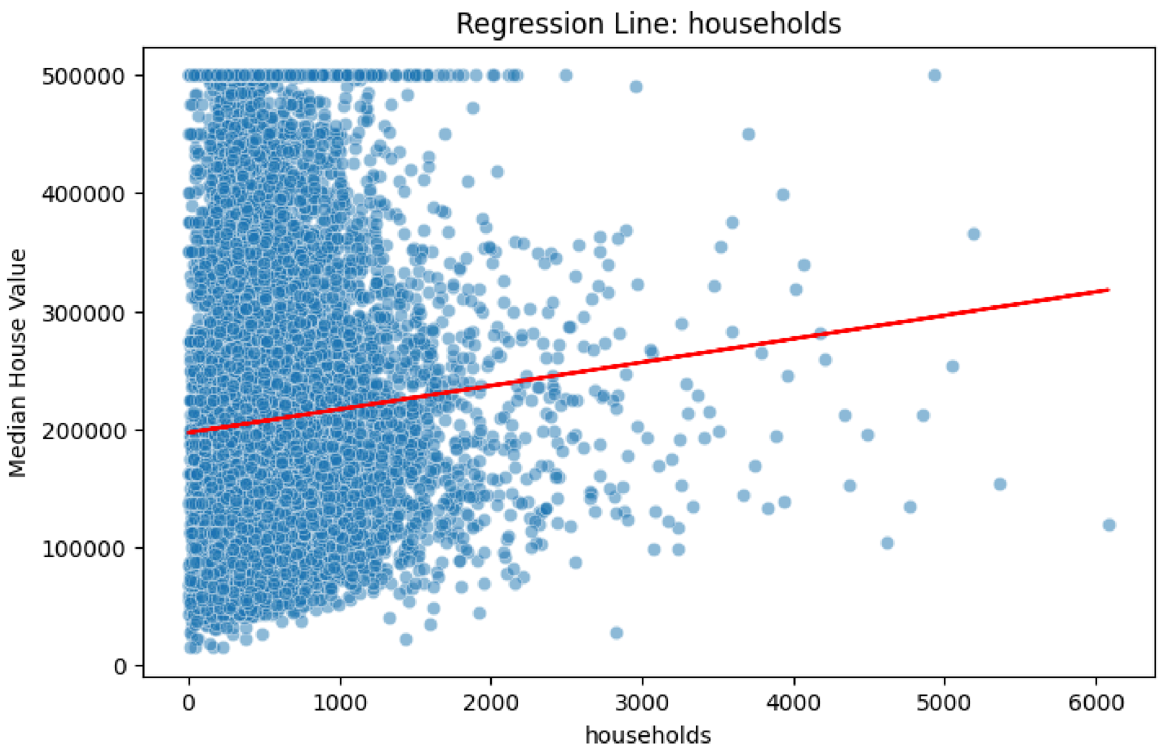
	Intercept	Slope	MSE	MAE
median_income	45085.576703	41793.849202	7.011312e+09	62625.933791
population	210436.262076	-2.511753	1.330741e+10	91153.820095
households	196928.577162	19.872775	1.325778e+10	90802.743243

In [19]:

```python
# Plot Regression Lines
import matplotlib.pyplot as plt
import seaborn as sns

for feature in ['median_income', 'population', 'households']:
    plt.figure(figsize=(8, 5))
    sns.scatterplot(x=df[feature], y=df["median_house_value"], alpha=0.5)
    plt.plot(df[feature], predictions[feature], color='red')
    plt.title(f"Regression Line: {feature}")
    plt.xlabel(feature)
    plt.ylabel("Median House Value")
    plt.show()
```







Summary & Recommendations

- **Median Income** is the best predictor among the three variables, with the lowest MSE and MAE.
- **Population** and **Households** have weak predictive power.
- We recommend using **median income** as a key feature in future models.

The strong positive trend between income and house values confirms expected economic behavior: higher income regions have higher property values.

In [ ]:

```
```