In [24]:

```
import pandas as pd
import numpy as np
from sklearn.datasets import fetch_california_housing
housing = fetch_california_housing()
import statsmodels.api as sm
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

In [25]:

```
df = pd.DataFrame(housing.data, columns = housing.feature_names)
```

In [6]:

```
df
```

Out[6]:

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude
0	8.3252	41.0	6.984127	1.023810	322.0	2.555556	37.88	-122.23
1	8.3014	21.0	6.238137	0.971880	2401.0	2.109842	37.86	-122.22
2	7.2574	52.0	8.288136	1.073446	496.0	2.802260	37.85	-122.24
3	5.6431	52.0	5.817352	1.073059	558.0	2.547945	37.85	-122.25
4	3.8462	52.0	6.281853	1.081081	565.0	2.181467	37.85	-122.25
...	...	...	...	...	...	...	...	...
20635	1.5603	25.0	5.045455	1.133333	845.0	2.560606	39.48	-121.09
20636	2.5568	18.0	6.114035	1.315789	356.0	3.122807	39.49	-121.21
20637	1.7000	17.0	5.205543	1.120092	1007.0	2.325635	39.43	-121.22
20638	1.8672	18.0	5.329513	1.171920	741.0	2.123209	39.43	-121.32
20639	2.3886	16.0	5.254717	1.162264	1387.0	2.616981	39.37	-121.24

20640 rows × 8 columns

In [12]:

```
#splitting the dependant and independant variables
X = df.drop("MedInc",axis=1)
y = df.MedInc
```

In [13]:

```
X
```

Out[13]:

	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude
0	41.0	6.984127	1.023810	322.0	2.555556	37.88	-122.23
1	21.0	6.238137	0.971880	2401.0	2.109842	37.86	-122.22
2	52.0	8.288136	1.073446	496.0	2.802260	37.85	-122.24
3	52.0	5.817352	1.073059	558.0	2.547945	37.85	-122.25
4	52.0	6.281853	1.081081	565.0	2.181467	37.85	-122.25

	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude
...	...	...	...	...	...	...	...
20635	25.0	5.045455	1.133333	845.0	2.560606	39.48	-121.09
20636	18.0	6.114035	1.315789	356.0	3.122807	39.49	-121.21
20637	17.0	5.205543	1.120092	1007.0	2.325635	39.43	-121.22
20638	18.0	5.329513	1.171920	741.0	2.123209	39.43	-121.32
20639	16.0	5.254717	1.162264	1387.0	2.616981	39.37	-121.24

20640 rows × 7 columns

In [14]:

```
y
```

Out[14]:

```
0        8.3252
1        8.3014
2        7.2574
3        5.6431
4        3.8462
          ...
20635    1.5603
20636    2.5568
20637    1.7000
20638    1.8672
20639    2.3886
Name: MedInc, Length: 20640, dtype: float64
```

In [15]:

```
# Add a constant to the model (for the intercept term)
X = sm.add_constant(X)
```

In [16]:

```
# Step 2: Split the Data 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
```

In [17]:

```
# Step 3: Fit the Regression Model
model = sm.OLS(y_train, X_train)  # Ordinary Least Squares
results = model.fit()
```

In [18]:

```
results
```

Out[18]:

```
<statsmodels.regression.linear_model.RegressionResultsWrapper at 0x23fd1ec7350>
```

In [19]:

```
results.summary
```

Out[19]:

```
<bound method RegressionResults.summary of <statsmodels.regression.linear_model.OLSResul
ts object at 0x0000023FD1EC64D0>>
```

In [20]:

```python
# Step 5: Make Predictions
y_pred = results.predict(X_test)
```

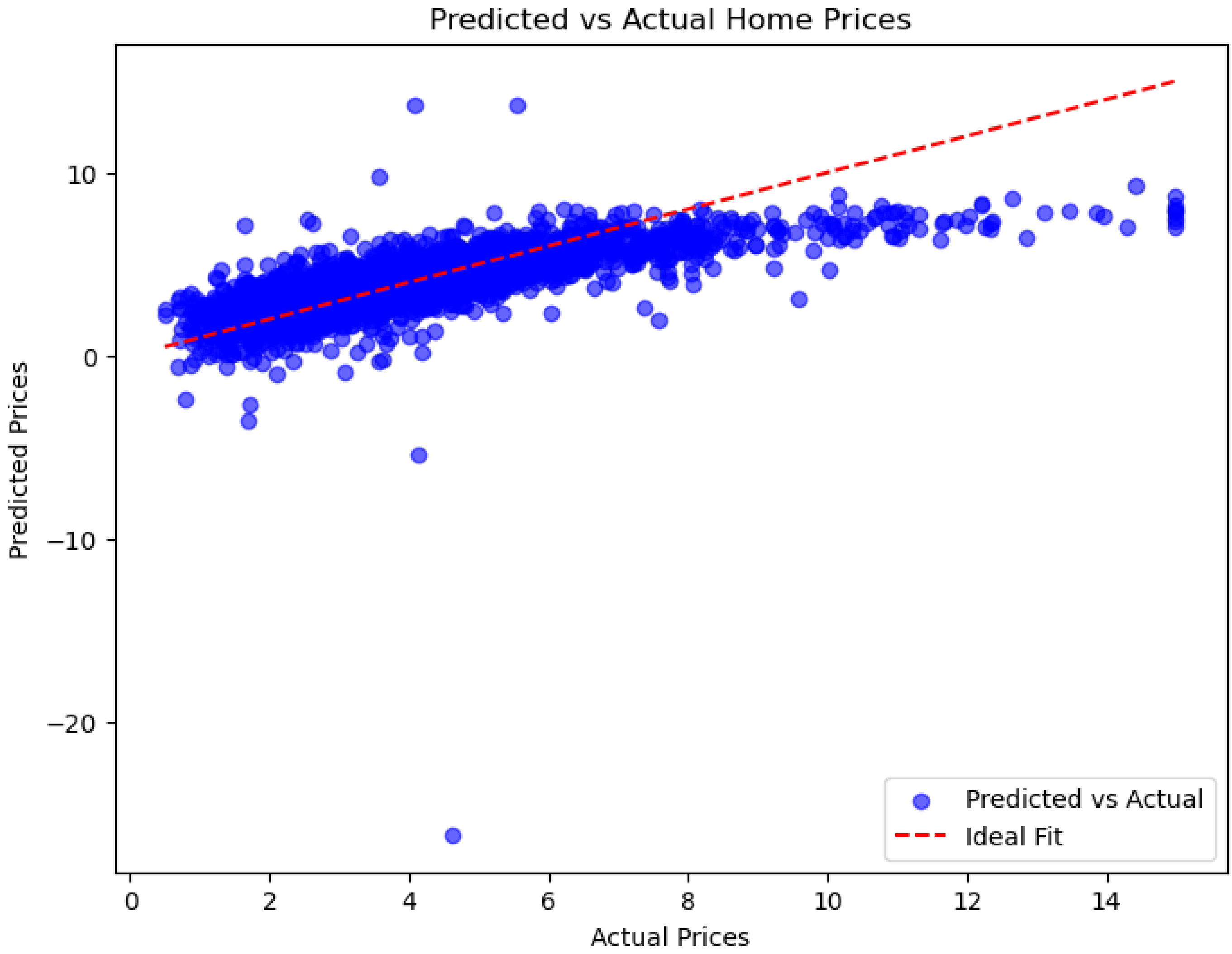
In [22]:

```python
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("\nModel Evaluation Metrics:")
print(f"Mean Squared Error (MSE): {mse}")
print(f"Mean Absolute Error (MAE): {mae}")
print(f"R-squared: {r2}")
```

```
Model Evaluation Metrics:
Mean Squared Error (MSE): 1.5662417233389965
Mean Absolute Error (MAE): 0.8032908533877124
R-squared: 0.5574640908701838
```

In [26]:

```python
# Step 7: Visualization - Predicted vs Actual Values
import matplotlib.pyplot as plt
plt.figure(figsize=(8, 6))
plt.scatter(y_test, y_pred, color='blue', alpha=0.6, label='Predicted vs Actual')
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], '--', color='red',
plt.title("Predicted vs Actual Home Prices")
plt.xlabel("Actual Prices")
plt.ylabel("Predicted Prices")
plt.legend()
plt.show()
```


Predicted vs Actual Home Prices
10
0
−10
−20
Predicted Prices
0
2
4
6
8
10
12
14
Actual Prices
Predicted vs Actual
Ideal Fit

In [ ]: