Linear Regression II: Homework

Question 1:

- Load the California Housing Dataset
- Find the feature with highest correlation with the Median House Value
- Predict the Median House Value with the highest correlated feature
- How does the prediction of the this model (in terms of R^2) compare with the prediction of the model used in the exercise in class that used all the features instead of just the most correlated feature?

In [1]:

```
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error,mean_squared_error, r2_scor

from sklearn.datasets import fetch_california_housing
california_housing = fetch_california_housing()
```

In [2]:

```
california = pd.DataFrame(california_housing.data, columns=california_housin
california['Median House Value'] = california_housing.target

corr = california.corr()['Median House Value']
corr
```

Out[2]:

```
MedInc                0.688075
HouseAge              0.105623
AveRooms              0.151948
AveBedrms            -0.046701
Population           -0.024650
AveOccup             -0.023737
Latitude             -0.144160
Longitude            -0.045967
Median House Value    1.000000
Name: Median House Value, dtype: float64
```

In [3]:

```
MedInc_corr = corr[0]

print(f"MedInc has the highest correlation with median housing value of {Me
```

```
MedInc has the highest correlation with median housing value of 0.69.
```

In [4]:

```
# predict

# get dependent and independent variables from the data set
X = california["MedInc"] #MedInc
y = california["Median House Value"]

X = X.values.reshape(-1, 1)
print(X.shape, y.shape)
```

```
(20640, 1) (20640,)
```

In [5]:

```
# Training the linear regression model
model = LinearRegression()
```

```
model.fit(X, y)

y_pred = model.predict(X)
```

In [6]:
```
y_data_array = np.array(y).reshape(-1, 1)
y_pred_array = np.array(y_pred).reshape(-1, 1)
```

In [45]:
```
r2 = r2_score(y_true = y_data_array, y_pred = y_pred_array)

print(f"The Coefficient of Determination (R²) is {r2:.2f}.")
```

```
The Coefficient of Determination (R²) is 0.47.
```

This model's prediction is less accurate than the model used in the exercise because it has a R^2 that is less than the R^2 of the model in the exercise. The model from the exercise used all of the features, allowing for a more accurate prediction than using only one feature, regardless of whether it is the most correlated feature.

Question 2:

- Load the California Housing Dataset
- Find the feature with lowest correlation with the Median House Value
- Predict the Median House Value with all the features *except* the one with lowest correlation
- How does the prediction of the this model (in terms of R^2) compare with the prediction of the full model model used in the exercise in class?

In [8]:
```
# find the feature with lowest correlation with median housing value
latitude_corr = corr.iloc[-2]

print(f"Latitude has the lowest correlation with median housing value of {la
```

```
Latitude has the lowest correlation with median housing value of -0.05.
```

In [39]:
```
X1 = california.drop("Latitude", axis = 1)
y1 = california.drop("Latitude", axis = 1)["Median House Value"]

print(X1.shape, y1.shape)
```

```
(20640, 8) (20640,)
```

In [46]:
```
# Create and fit the Linear Regression model
model = LinearRegression()
model.fit(X1, y1)

# Predict Median House Value using all features except "Latitude"
y_pred1 = model.predict(X1)

y_data_array1 = np.array(y1).reshape(-1, 1)
y_pred_array1 = np.array(y_pred1).reshape(-1, 1)

r2_lat = r2_score(y_true = y_data_array, y_pred = y_pred_array)

print(f"The Coefficient of Determination (R²) is {r2_lat:.2f}.")
```

```
The Coefficient of Determination (R²) is 0.47.
```

This model's prediction, in terms of R^2, is less accurate than the full model used in the class exercise. This indicates that the model makes better predictions with more features, even if the feature is the least correlated with the target.