california-housing

June 20, 2024

CALIFORNIA HOUSING

[1]:
```
# Importing all the necesaary libraries
from sklearn.datasets import fetch_california_housing
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
```

[2]:
```
# Load dataset
housing = fetch_california_housing()
X, y = housing.data, housing.target
```

[7]:
```
print("Feature names:", housing.feature_names)
```

```
Feature names: ['MedInc', 'HouseAge', 'AveRooms', 'AveBedrms', 'Population',
'AveOccup', 'Latitude', 'Longitude']
```

[4]:
```
import pandas as pd
import numpy as np
```

[9]:
```
df = pd.DataFrame(X, columns = housing.feature_names)
```

[11]:
```
target = pd.DataFrame(y, columns = ['MedianHouseValue'])
target.head()
```

[11]:
```
   MedianHouseValue
0             4.526
1             3.585
2             3.521
3             3.413
4             3.422
```

[10]:
```
df.head()
```

[10]:
```
   MedInc  HouseAge  AveRooms  AveBedrms  Population  AveOccup  Latitude  \
0  8.3252      41.0  6.984127   1.023810       322.0  2.555556     37.88
1  8.3014      21.0  6.238137   0.971880      2401.0  2.109842     37.86
2  7.2574      52.0  8.288136   1.073446       496.0  2.802260     37.85
```

```
3  5.6431      52.0  5.817352   1.073059       558.0  2.547945     37.85
4  3.8462      52.0  6.281853   1.081081       565.0  2.181467     37.85

   Longitude
0    -122.23
1    -122.22
2    -122.24
3    -122.25
4    -122.25
```

[13]:

```
df.isnull().sum()
```

[13]:

```
MedInc        0
HouseAge      0
AveRooms      0
AveBedrms     0
Population    0
AveOccup      0
Latitude      0
Longitude     0
dtype: int64
```

[14]:

```
target.isnull().sum()
```

[14]:

```
MedianHouseValue    0
dtype: int64
```

[15]:

```
df.corr()
```

[15]:

```
              MedInc  HouseAge  AveRooms  AveBedrms  Population  AveOccup  \
MedInc      1.000000 -0.119034  0.326895  -0.062040    0.004834  0.018766
HouseAge   -0.119034  1.000000 -0.153277  -0.077747   -0.296244  0.013191
AveRooms    0.326895 -0.153277  1.000000   0.847621   -0.072213 -0.004852
AveBedrms  -0.062040 -0.077747  0.847621   1.000000   -0.066197 -0.006181
Population  0.004834 -0.296244 -0.072213  -0.066197    1.000000  0.069863
AveOccup    0.018766  0.013191 -0.004852  -0.006181    0.069863  1.000000
Latitude   -0.079809  0.011173  0.106389   0.069721   -0.108785  0.002366
Longitude  -0.015176 -0.108197 -0.027540   0.013344    0.099773  0.002476

            Latitude  Longitude
MedInc     -0.079809  -0.015176
HouseAge    0.011173  -0.108197
AveRooms    0.106389  -0.027540
AveBedrms   0.069721   0.013344
Population -0.108785   0.099773
AveOccup    0.002366   0.002476
Latitude    1.000000  -0.924664
```

```
 Longitude  -0.924664   1.000000
```

[19]:

```
# Calculate the correlation between each feature and the target variable
correlations = df.apply(lambda x: x.corr(pd.Series(y)), axis=0)
print("\nCorrelation with MedianHouseValue:\n", correlations.
 ↪sort_values(ascending=False))
```

```
Correlation with MedianHouseValue:
 MedInc        0.688075
AveRooms      0.151948
HouseAge      0.105623
AveOccup     -0.023737
Population   -0.024650
Longitude    -0.045967
AveBedrms    -0.046701
Latitude     -0.144160
dtype: float64
```

[20]:

```
# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

# Train model
rf_reg = RandomForestRegressor(random_state=42)
rf_reg.fit(X_train, y_train)

# Predict and evaluate
y_pred = rf_reg.predict(X_test)
print(f"\nMean Squared Error: {mean_squared_error(y_test, y_pred)}")
```

```
Mean Squared Error: 0.2553684927247781
```