welcome-to-colaboratory-4

March 6, 2024

[14]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler

# a) Read the data with pandas and describe the data
data = pd.read_csv('/content/housing.csv')
description = data.describe()
print(description)

# b) Find data type and shape of each column
data_types = data.dtypes
shape = data.shape
print("Data Types:\n", data_types)
print("\nShape of Data:", shape)

# c) Find the null values (if yes fill the null values with '0' or mean of that␣
 ↪column)
null_values = data.isnull().sum()
print("\nNull Values:\n", null_values)

# Filling null values with mean
data.fillna(data.mean(), inplace=True)

# d) Find features and target variables
# Assuming the target variable is in the last column
features = data.iloc[:, :-1]
target = data.iloc[:, -1]

# e) Split the data into train and test
X_train, X_test, y_train, y_test = train_test_split(features, target,␣
 ↪test_size=0.2, random_state=42)

# f) Normalize the data with min-max scaling
scaler = MinMaxScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

```
          longitude      latitude  housing_median_age   total_rooms  \
```

```
count  20640.000000  20640.000000        20640.000000  20640.000000
mean    -119.569704     35.631861           28.639486   2635.763081
std        2.003532      2.135952           12.585558   2181.615252
min     -124.350000     32.540000            1.000000      2.000000
25%     -121.800000     33.930000           18.000000   1447.750000
50%     -118.490000     34.260000           29.000000   2127.000000
75%     -118.010000     37.710000           37.000000   3148.000000
max     -114.310000     41.950000           52.000000  39320.000000

       total_bedrooms    population    households  median_income  \
count    20433.000000  20640.000000  20640.000000   20640.000000
mean       537.870553   1425.476744    499.539680       3.870671
std        421.385070   1132.462122    382.329753       1.899822
min          1.000000      3.000000      1.000000       0.499900
25%        296.000000    787.000000    280.000000       2.563400
50%        435.000000   1166.000000    409.000000       3.534800
75%        647.000000   1725.000000    605.000000       4.743250
max       6445.000000  35682.000000   6082.000000      15.000100

       median_house_value
count        20640.000000
mean        206855.816909
std         115395.615874
min          14999.000000
25%         119600.000000
50%         179700.000000
75%         264725.000000
max         500001.000000
Data Types:
 longitude             float64
latitude              float64
housing_median_age    float64
total_rooms           float64
total_bedrooms        float64
population            float64
households            float64
median_income         float64
median_house_value    float64
ocean_proximity        object
dtype: object

Shape of Data: (20640, 10)

Null Values:
 longitude               0
latitude                0
housing_median_age      0
total_rooms             0
```

```
total_bedrooms        207
population              0
households              0
median_income           0
median_house_value      0
ocean_proximity         0
dtype: int64
```

```
<ipython-input-14-e3ab72bd7331>:21: FutureWarning: The default value of
numeric_only in DataFrame.mean is deprecated. In a future version, it will
default to False. In addition, specifying 'numeric_only=None' is deprecated.
Select only valid columns or specify the value of numeric_only to silence this
warning.
  data.fillna(data.mean(), inplace=True)
```