MACHINE LEARNING PROJECT - 1 - BUILDING A MODEL USING DECISION TREE REGRESSOR TO PREDICT MEDIAN HOUSE VALUE BY ANALYZING CALIFORNIA HOUSING DATASET

```
import pandas as pd
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error


cal_housing_data = pd.read_csv("D:/6.Data Analytics/Machine learning - Kaggle/Project1/housing.csv")
```

```
print(cal_housing_data.shape)
```

(20640, 10)

This means there are 20640 rows and 10 columns.

```
print(cal_housing_data.head())
```

```
   longitude  latitude  ...  median_house_value  ocean_proximity
0    -122.23     37.88  ...            452600.0         NEAR BAY
1    -122.22     37.86  ...            358500.0         NEAR BAY
2    -122.24     37.85  ...            352100.0         NEAR BAY
3    -122.25     37.85  ...            341300.0         NEAR BAY
4    -122.25     37.85  ...            342200.0         NEAR BAY

[5 rows x 10 columns]
```

```
print(cal_housing_data.describe())
```

```
          longitude      latitude  ...  median_income  median_house_value
count  20640.000000  20640.000000  ...   20640.000000        20640.000000
mean    -119.569704     35.631861  ...       3.870671       206855.816909
std        2.003532      2.135952  ...       1.899822       115395.615874
min     -124.350000     32.540000  ...       0.499900        14999.000000
25%     -121.800000     33.930000  ...       2.563400       119600.000000
50%     -118.490000     34.260000  ...       3.534800       179700.000000
75%     -118.010000     37.710000  ...       4.743250       264725.000000
max     -114.310000     41.950000  ...      15.000100       500001.000000

[8 rows x 9 columns]
```

```
print(cal_housing_data.isna().any().any())
print("\n")
print(cal_housing_data.isna().sum())
```

```
True


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
```

```
print(cal_housing_data['total_bedrooms'].head(50))
```

```
0      129.0
1     1106.0
2      190.0
3      235.0
4      280.0
5      213.0
6      489.0
7      687.0
8      665.0
9      707.0
10     434.0
11     752.0
12     474.0
13     191.0
14     626.0
15     283.0
16     347.0
17     293.0
18     455.0
19     298.0
20     184.0
21     367.0
22     541.0
23     337.0
24     437.0
25     123.0
26     244.0
27     421.0
28     492.0
29     160.0
30     447.0
31     481.0
32     409.0
33     366.0
34     574.0
35     282.0
36     432.0
37     390.0
38     330.0
39     715.0
40     419.0
41     311.0
42     202.0
43     202.0
44     311.0
45     420.0
46     322.0
47     312.0
48     195.0
49     375.0
Name: total_bedrooms, dtype: float64
print(cal_housing_data['total_bedrooms'].tail
())
```

```
print(cal_housing_data['total_bedrooms'].tail(50))
```

```
20590      350.0
20591      376.0
20592      160.0
20593      197.0
20594      302.0
20595      642.0
20596      375.0
20597      378.0
20598      396.0
20599      441.0
20600      367.0
20601      103.0
20602      608.0
20603     1021.0
20604      899.0
20605      629.0
20606      534.0
20607      333.0
20608      261.0
20609      480.0
20610      484.0
20611      441.0
20612      289.0
20613      365.0
20614      460.0
20615      216.0
20616      441.0
20617      109.0
20618      247.0
20619      340.0
20620       41.0
20621      247.0
20622      147.0
20623      244.0
20624      300.0
20625       17.0
20626      184.0
20627       65.0
20628      421.0
20629     1856.0
20630      505.0
20631      493.0
20632      416.0
20633      412.0
20634      395.0
20635      374.0
20636      150.0
20637      485.0
20638      409.0
20639      616.0
Name: total_bedrooms, dtype: float64
```

```
cal_housing_data.info()
print("\n")
print(cal_housing_data['ocean_proximity'].head())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   longitude           20640 non-null  float64
 1   latitude            20640 non-null  float64
 2   housing_median_age  20640 non-null  float64
 3   total_rooms         20640 non-null  float64
 4   total_bedrooms      20433 non-null  float64
 5   population          20640 non-null  float64
 6   households          20640 non-null  float64
 7   median_income       20640 non-null  float64
 8   median_house_value  20640 non-null  float64
 9   ocean_proximity     20640 non-null  object
dtypes: float64(9), object(1)
memory usage: 1.6+ MB


0    NEAR BAY
1    NEAR BAY
2    NEAR BAY
3    NEAR BAY
4    NEAR BAY
Name: ocean_proximity, dtype: object
```

```
filtered_cal_housing_data = cal_housing_data.dropna(axis = 0) #remove all the rows that contain null values
print(filtered_cal_housing_data.isna().sum())
```

```
longitude             0
latitude              0
housing_median_age    0
total_rooms           0
total_bedrooms        0
population            0
households            0
median_income         0
median_house_value    0
ocean_proximity       0
dtype: int64
```

In here I removed the rows that contain NULL values rather than removing whole column. Because only 207 entries contains NULLs in the total_bedrooms column and according to my opinion, it is negligible compared to the total number of rows, that is 20640.
And also, total number of bedrooms are highly affects the value of the house. Therefore, removing whole column is misleading the model.

```
y = filtered_cal_housing_data.median_house_value
print(y.head(10))
print("\n")
print("Mininum Median House Value:",min(y))
print("Maximum Median House Value:",max(y))
```

```
0    452600.0
1    358500.0
2    352100.0
3    341300.0
4    342200.0
5    269700.0
6    299200.0
7    241400.0
8    226700.0
9    261100.0
Name: median_house_value, dtype: float64


Mininum Median House Value: 14999.0
Maximum Median House Value: 500001.0
```

```
cal_housing_features = ['longitude','latitude','housing_median_age','total_rooms','population','households','median_income','ocean_proximity','total_bedrooms']

X = filtered_cal_housing_data[cal_housing_features]
X = pd.get_dummies(X, columns = ['ocean_proximity'])
print(X.head())
```

```
   longitude  latitude  ...  ocean_proximity_NEAR BAY  ocean_proximity_NEAR OCEAN
0    -122.23     37.88  ...                      True                       False
1    -122.22     37.86  ...                      True                       False
2    -122.24     37.85  ...                      True                       False
3    -122.25     37.85  ...                      True                       False
4    -122.25     37.85  ...                      True                       False

[5 rows x 13 columns]
```

```
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size = 0.2)

cal_housing_model = DecisionTreeRegressor()
cal_housing_model.fit(X_train,y_train)
prediction = cal_housing_model.predict(X_test)
mae = mean_absolute_error(y_test, prediction)
print("The Mean Absolute Error: ",round(mae,2))
```

File Edit Shell Debug Options Window Help

```
Python 3.11.3 (tags/v3.11.3:f3909b8, Apr  4 2023, 23:49:59) [MSC v.1934 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: D:\coding files so far\python\ML1.py ==========
The Mean Absolute Error:  43708.66
>>>
========== RESTART: D:\coding files so far\python\ML1.py ==========
The Mean Absolute Error:  42145.45
>>>
========== RESTART: D:\coding files so far\python\ML1.py ==========
The Mean Absolute Error:  44399.24
>>>
========== RESTART: D:\coding files so far\python\ML1.py ==========
The Mean Absolute Error:  42503.27
>>>
========== RESTART: D:\coding files so far\python\ML1.py ==========
The Mean Absolute Error:  41899.09
>>>
========== RESTART: D:\coding files so far\python\ML1.py ==========
The Mean Absolute Error:  43483.04
>>>
========== RESTART: D:\coding files so far\python\ML1.py ==========
The Mean Absolute Error:  43798.16
>>>
========== RESTART: D:\coding files so far\python\ML1.py ==========
The Mean Absolute Error:  43482.57
>>>
========== RESTART: D:\coding files so far\python\ML1.py ==========
The Mean Absolute Error:  43593.24
>>>
========== RESTART: D:\coding files so far\python\ML1.py ==========
The Mean Absolute Error:  43931.48
>>>
========== RESTART: D:\coding files so far\python\ML1.py ==========
The Mean Absolute Error:  44204.12
>>>
```