HOUSE PRICE PREDICTION PROJECT

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

importing data from local drive

I first downloaded the data from kaggle , than saved it in a folder , and created this project in the same folder . after that , i imported the data from there

In [2]:
```
df=pd.read_csv("housing.csv")
```

In [3]:
```
df
```

Out[3]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households
0	-122.23	37.88	41.0	880.0	129.0	322.0	126.(
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0	1138.(
2	-122.24	37.85	52.0	1467.0	190.0	496.0	177.(
3	-122.25	37.85	52.0	1274.0	235.0	558.0	219.(
4	-122.25	37.85	52.0	1627.0	280.0	565.0	259.(
...	...	...	...	...	...	...	..
20635	-121.09	39.48	25.0	1665.0	374.0	845.0	330.(
20636	-121.21	39.49	18.0	697.0	150.0	356.0	114.(
20637	-121.22	39.43	17.0	2254.0	485.0	1007.0	433.(
20638	-121.32	39.43	18.0	1860.0	409.0	741.0	349.(
20639	-121.24	39.37	16.0	2785.0	616.0	1387.0	530.(

20640 rows × 10 columns

In [4]:
```
df.head()
```

Out[4]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	m
0	-122.23	37.88	41.0	880.0	129.0	322.0	126.0	
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0	1138.0	
2	-122.24	37.85	52.0	1467.0	190.0	496.0	177.0	
3	-122.25	37.85	52.0	1274.0	235.0	558.0	219.0	
4	-122.25	37.85	52.0	1627.0	280.0	565.0	259.0	

In [5]:
```
df.tail()
```

Out[5]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	household
20635	-121.09	39.48	25.0	1665.0	374.0	845.0	330.(
20636	-121.21	39.49	18.0	697.0	150.0	356.0	114.(
20637	-121.22	39.43	17.0	2254.0	485.0	1007.0	433.(
20638	-121.32	39.43	18.0	1860.0	409.0	741.0	349.(
20639	-121.24	39.37	16.0	2785.0	616.0	1387.0	530.(

In [6]:
```
df.shape
```

Out[6]:
```
(20640, 10)
```

In [7]:
```
df.dtypes
```

Out[7]:
```
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
```

In [8]:
```
df.describe
```

Out[8]:
```
<bound method NDFrame.describe of       longitude  latitude  housing_median_age  tot
al_rooms  total_bedrooms  \
0       -122.23     37.88                41.0        880.0           129.0
1       -122.22     37.86                21.0       7099.0          1106.0
2       -122.24     37.85                52.0       1467.0           190.0
3       -122.25     37.85                52.0       1274.0           235.0
4       -122.25     37.85                52.0       1627.0           280.0
...         ...       ...                 ...          ...             ...
20635   -121.09     39.48                25.0       1665.0           374.0
20636   -121.21     39.49                18.0        697.0           150.0
20637   -121.22     39.43                17.0       2254.0           485.0
20638   -121.32     39.43                18.0       1860.0           409.0
20639   -121.24     39.37                16.0       2785.0           616.0

       population  households  median_income  median_house_value  \
0           322.0       126.0         8.3252            452600.0
1          2401.0      1138.0         8.3014            358500.0
2           496.0       177.0         7.2574            352100.0
3           558.0       219.0         5.6431            341300.0
4           565.0       259.0         3.8462            342200.0
...           ...         ...            ...                 ...
20635       845.0       330.0         1.5603             78100.0
20636       356.0       114.0         2.5568             77100.0
20637      1007.0       433.0         1.7000             92300.0
20638       741.0       349.0         1.8672             84700.0
20639      1387.0       530.0         2.3886             89400.0

      ocean_proximity
0            NEAR BAY
1            NEAR BAY
2            NEAR BAY
3            NEAR BAY
4            NEAR BAY
...               ...
20635          INLAND
20636          INLAND
20637          INLAND
20638          INLAND
20639          INLAND

[20640 rows x 10 columns]>
```

In [9]:
```
df.describe(include="all")
```

Out[9]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population
count	20640.000000	20640.000000	20640.000000	20640.000000	20433.000000	20640.000000
unique	NaN	NaN	NaN	NaN	NaN	NaN
top	NaN	NaN	NaN	NaN	NaN	NaN
freq	NaN	NaN	NaN	NaN	NaN	NaN
mean	-119.569704	35.631861	28.639486	2635.763081	537.870553	1425.476744
std	2.003532	2.135952	12.585558	2181.615252	421.385070	1132.462122
min	-124.350000	32.540000	1.000000	2.000000	1.000000	3.000000
25%	-121.800000	33.930000	18.000000	1447.750000	296.000000	787.000000
50%	-118.490000	34.260000	29.000000	2127.000000	435.000000	1166.000000
75%	-118.010000	37.710000	37.000000	3148.000000	647.000000	1725.000000
max	-114.310000	41.950000	52.000000	39320.000000	6445.000000	35682.000000

In [10]:
```
df.columns
```

Out[10]:
```
Index(['longitude', 'latitude', 'housing_median_age', 'total_rooms',
       'total_bedrooms', 'population', 'households', 'median_income',
       'median_house_value', 'ocean_proximity'],
      dtype='object')
```

In [11]:
```
df.info()
```

```
<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
```

In [12]:
```
df.dropna(inplace=True)
```

In [13]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 20433 entries, 0 to 20639
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   longitude           20433 non-null  float64
 1   latitude            20433 non-null  float64
 2   housing_median_age  20433 non-null  float64
 3   total_rooms         20433 non-null  float64
 4   total_bedrooms      20433 non-null  float64
 5   population          20433 non-null  float64
 6   households          20433 non-null  float64
 7   median_income       20433 non-null  float64
 8   median_house_value  20433 non-null  float64
 9   ocean_proximity     20433 non-null  object 
dtypes: float64(9), object(1)
memory usage: 1.7+ MB
```

In [14]:
```
df["median_house_value"].value_counts()
```

Out[14]:
```
median_house_value
500001.0    958
137500.0    119
162500.0    116
112500.0    103
187500.0     92
           ...
359200.0      1
51200.0       1
39800.0       1
377600.0      1
47000.0       1
Name: count, Length: 3833, dtype: int64
```

In [15]:
```
df["median_house_value"].mean()
```

Out[15]:
```
206864.41315519012
```

In [16]:
```
df.isnull()
```

Out[16]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households
0	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...
20635	False	False	False	False	False	False	False
20636	False	False	False	False	False	False	False
20637	False	False	False	False	False	False	False
20638	False	False	False	False	False	False	False
20639	False	False	False	False	False	False	False

20433 rows × 10 columns

In [17]:
```
df.isnull().sum()
```

Out[17]:
```
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 [18]:
```
df.duplicated
```

Out[18]:
```
<bound method DataFrame.duplicated of        longitude  latitude  housing_median_age
total_rooms  total_bedrooms  \
0         -122.23     37.88                41.0        880.0           129.0
1         -122.22     37.86                21.0       7099.0          1106.0
2         -122.24     37.85                52.0       1467.0           190.0
3         -122.25     37.85                52.0       1274.0           235.0
4         -122.25     37.85                52.0       1627.0           280.0
...           ...       ...                 ...          ...             ...
20635     -121.09     39.48                25.0       1665.0           374.0
20636     -121.21     39.49                18.0        697.0           150.0
20637     -121.22     39.43                17.0       2254.0           485.0
20638     -121.32     39.43                18.0       1860.0           409.0
20639     -121.24     39.37                16.0       2785.0           616.0

       population  households  median_income  median_house_value  \
0           322.0       126.0         8.3252            452600.0
1          2401.0      1138.0         8.3014            358500.0
2           496.0       177.0         7.2574            352100.0
3           558.0       219.0         5.6431            341300.0
4           565.0       259.0         3.8462            342200.0
...           ...         ...            ...                 ...
20635       845.0       330.0         1.5603             78100.0
20636       356.0       114.0         2.5568             77100.0
20637      1007.0       433.0         1.7000             92300.0
20638       741.0       349.0         1.8672             84700.0
20639      1387.0       530.0         2.3886             89400.0

      ocean_proximity
0            NEAR BAY
1            NEAR BAY
2            NEAR BAY
3            NEAR BAY
4            NEAR BAY
...               ...
20635          INLAND
20636          INLAND
20637          INLAND
20638          INLAND
20639          INLAND

[20433 rows x 10 columns]>
```

In [19]:
```
df.duplicated().sum()
```

Out[19]:
```
0
```

In [20]:
```
df["median_house_value"]
```

Out[20]:
```
0        452600.0
1        358500.0
2        352100.0
3        341300.0
4        342200.0
           ...
20635     78100.0
20636     77100.0
20637     92300.0
20638     84700.0
20639     89400.0
Name: median_house_value, Length: 20433, dtype: float64
```

In [21]:
```
df["ocean_proximity"]
```

Out[21]:
```
0        NEAR BAY
1        NEAR BAY
2        NEAR BAY
3        NEAR BAY
4        NEAR BAY
           ...
20635      INLAND
20636      INLAND
20637      INLAND
20638      INLAND
20639      INLAND
Name: ocean_proximity, Length: 20433, dtype: object
```

In [22]:
```
dummy_column=pd.get_dummies(df.ocean_proximity)
```

In [23]:
```
dummy_column=dummy_column.astype(int)
dummy_column
```

Out[23]:

	<1H OCEAN	INLAND	ISLAND	NEAR BAY	NEAR OCEAN
0	0	0	0	1	0
1	0	0	0	1	0
2	0	0	0	1	0
3	0	0	0	1	0
4	0	0	0	1	0
...	...	...	...	...	...
20635	0	1	0	0	0
20636	0	1	0	0	0
20637	0	1	0	0	0
20638	0	1	0	0	0
20639	0	1	0	0	0

20433 rows × 5 columns

In [24]:
```
dummy_column.isnull().sum()
```

Out[24]:
```
<1H OCEAN     0
INLAND        0
ISLAND        0
NEAR BAY      0
NEAR OCEAN    0
dtype: int64
```

In [25]:
```
dummy_column.shape
```

Out[25]:
```
(20433, 5)
```

In [26]:
```
df2=pd.DataFrame(dummy_column)
df2
```

Out[26]:

	<1H OCEAN	INLAND	ISLAND	NEAR BAY	NEAR OCEAN
0	0	0	0	1	0
1	0	0	0	1	0
2	0	0	0	1	0
3	0	0	0	1	0
4	0	0	0	1	0
...	...	...	...	...	...
20635	0	1	0	0	0
20636	0	1	0	0	0
20637	0	1	0	0	0
20638	0	1	0	0	0
20639	0	1	0	0	0

20433 rows × 5 columns

In [27]:
```
df=df.drop(["ocean_proximity"],axis=1)
df
```

Out[27]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households
0	-122.23	37.88	41.0	880.0	129.0	322.0	126.(
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0	1138.(
2	-122.24	37.85	52.0	1467.0	190.0	496.0	177.(
3	-122.25	37.85	52.0	1274.0	235.0	558.0	219.(
4	-122.25	37.85	52.0	1627.0	280.0	565.0	259.(
...	...	...	...	...	...	...	..
20635	-121.09	39.48	25.0	1665.0	374.0	845.0	330.(
20636	-121.21	39.49	18.0	697.0	150.0	356.0	114.(
20637	-121.22	39.43	17.0	2254.0	485.0	1007.0	433.(
20638	-121.32	39.43	18.0	1860.0	409.0	741.0	349.(
20639	-121.24	39.37	16.0	2785.0	616.0	1387.0	530.(

20433 rows × 9 columns

In [28]:
```
df=pd.concat([df,df2],axis=1 )
df
```

Out[28]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households
0	-122.23	37.88	41.0	880.0	129.0	322.0	126.(
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0	1138.(
2	-122.24	37.85	52.0	1467.0	190.0	496.0	177.(
3	-122.25	37.85	52.0	1274.0	235.0	558.0	219.(
4	-122.25	37.85	52.0	1627.0	280.0	565.0	259.(
...	...	...	...	...	...	...	..
20635	-121.09	39.48	25.0	1665.0	374.0	845.0	330.(
20636	-121.21	39.49	18.0	697.0	150.0	356.0	114.(
20637	-121.22	39.43	17.0	2254.0	485.0	1007.0	433.(
20638	-121.32	39.43	18.0	1860.0	409.0	741.0	349.(
20639	-121.24	39.37	16.0	2785.0	616.0	1387.0	530.(

20433 rows × 14 columns

In [29]:
```
df.dtypes
```

Out[29]:
```
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
<1H OCEAN               int32
INLAND                  int32
ISLAND                  int32
NEAR BAY                int32
NEAR OCEAN              int32
dtype: object
```

In [30]:
```
df["median_income"].value_counts()
```

Out[30]:
```
median_income
3.1250     49
15.0001    48
2.8750     46
4.1250     44
2.6250     44
           ..
6.0723      1
4.6992      1
5.4042      1
6.7744      1
2.0943      1
Name: count, Length: 12825, dtype: int64
```

In [31]:
```
df["population"]
```

Out[31]:
```
0         322.0
1        2401.0
2         496.0
3         558.0
4         565.0
          ...
20635     845.0
20636     356.0
20637    1007.0
20638     741.0
20639    1387.0
Name: population, Length: 20433, dtype: float64
```

In most cases , our data is clean here

Data Visualization

In [32]:
```
x=df.drop(["median_house_value"],axis=1)
y=df['median_house_value']
```

In [33]:
```
x
```

Out[33]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	household
0	-122.23	37.88	41.0	880.0	129.0	322.0	126.(
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0	1138.(
2	-122.24	37.85	52.0	1467.0	190.0	496.0	177.(
3	-122.25	37.85	52.0	1274.0	235.0	558.0	219.(
4	-122.25	37.85	52.0	1627.0	280.0	565.0	259.(
...	...	...	...	...	...	...	..
20635	-121.09	39.48	25.0	1665.0	374.0	845.0	330.(
20636	-121.21	39.49	18.0	697.0	150.0	356.0	114.(
20637	-121.22	39.43	17.0	2254.0	485.0	1007.0	433.(
20638	-121.32	39.43	18.0	1860.0	409.0	741.0	349.(
20639	-121.24	39.37	16.0	2785.0	616.0	1387.0	530.(

20433 rows × 13 columns

In [34]:
```
y
```

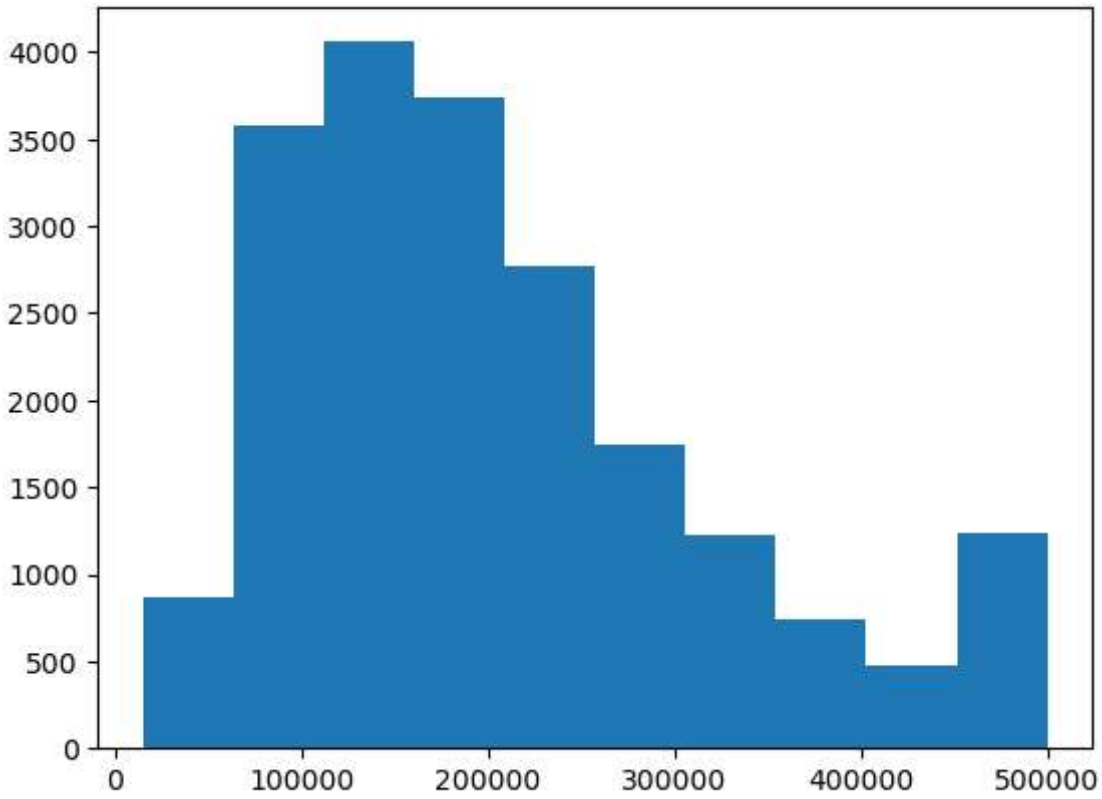
Out[34]:
```
0        452600.0
1        358500.0
2        352100.0
3        341300.0
4        342200.0
           ...
20635     78100.0
20636     77100.0
20637     92300.0
20638     84700.0
20639     89400.0
Name: median_house_value, Length: 20433, dtype: float64
```

In [35]:
```
x.columns
```

Out[35]:
```
Index(['longitude', 'latitude', 'housing_median_age', 'total_rooms',
       'total_bedrooms', 'population', 'households', 'median_income',
       '<1H OCEAN', 'INLAND', 'ISLAND', 'NEAR BAY', 'NEAR OCEAN'],
      dtype='object')
```

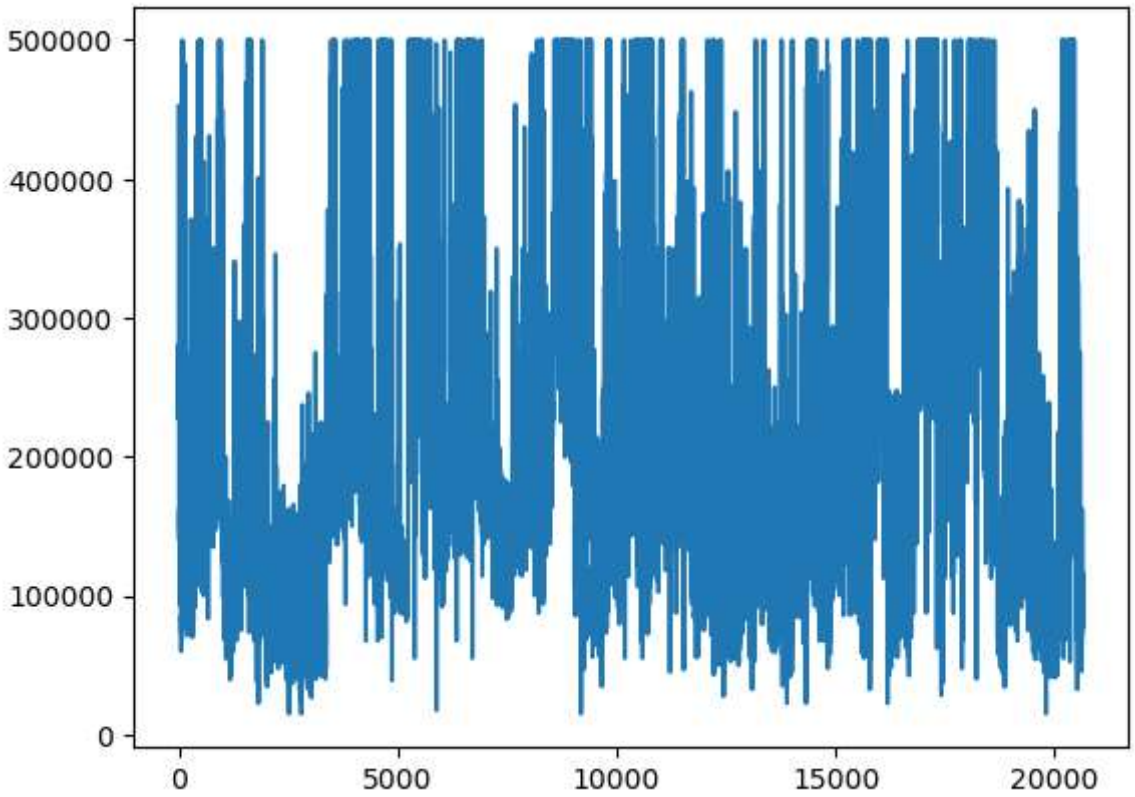
In [36]:
```
import matplotlib.pyplot as plt
%matplotlib inline
plt.hist(y)
```

Out[36]:
```
(array([ 868., 3577., 4058., 3733., 2772., 1747., 1228.,  746.,  471.,
        1233.]),
 array([ 14999. ,  63499.2, 111999.4, 160499.6, 208999.8, 257500. ,
        306000.2, 354500.4, 403000.6, 451500.8, 500001. ]),
 <BarContainer object of 10 artists>)
```



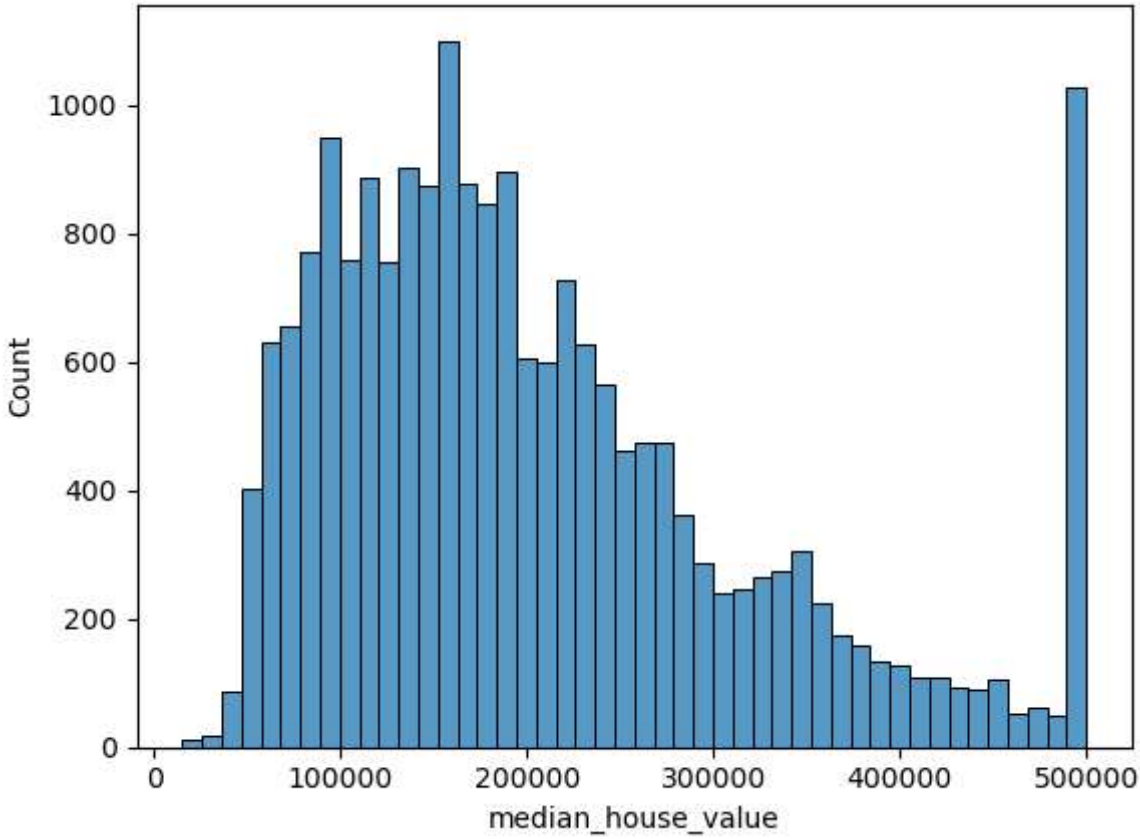
In [37]:
```
plt.plot(df["median_house_value"])
```

Out[37]:
```
[<matplotlib.lines.Line2D at 0x24aa0f1ff10>]
```



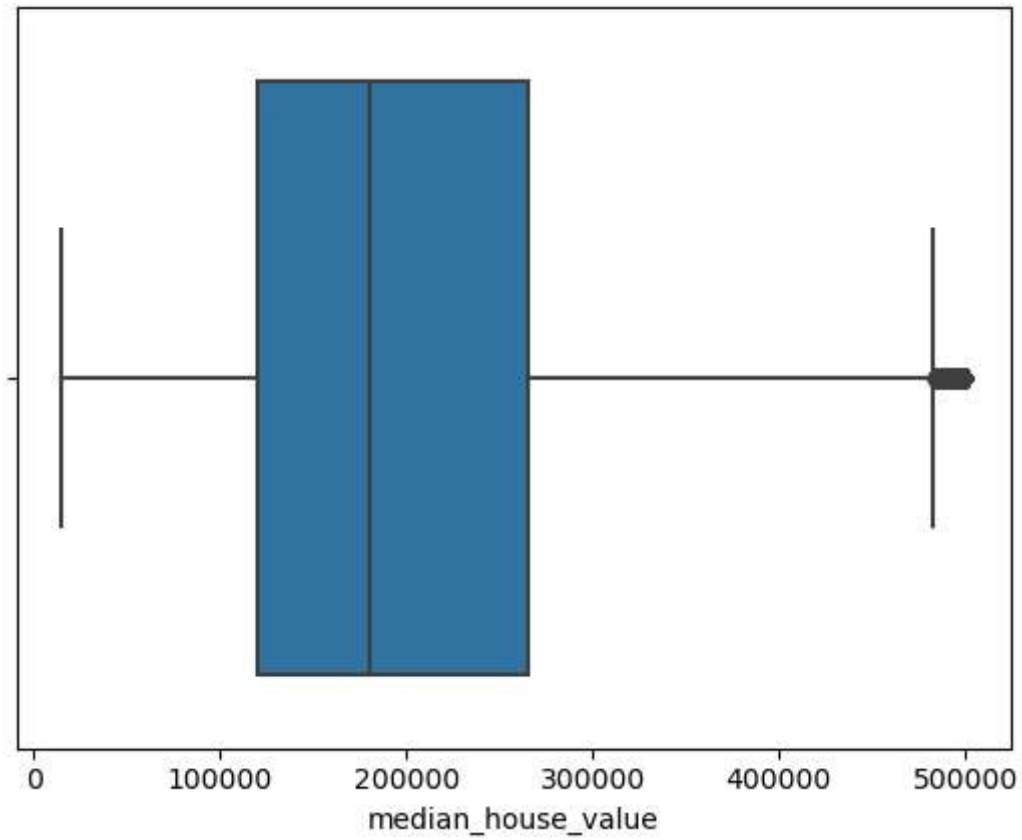
In [38]:
```
sns.histplot(df["median_house_value"])
```

Out[38]:
```
<Axes: xlabel='median_house_value', ylabel='Count'>
```



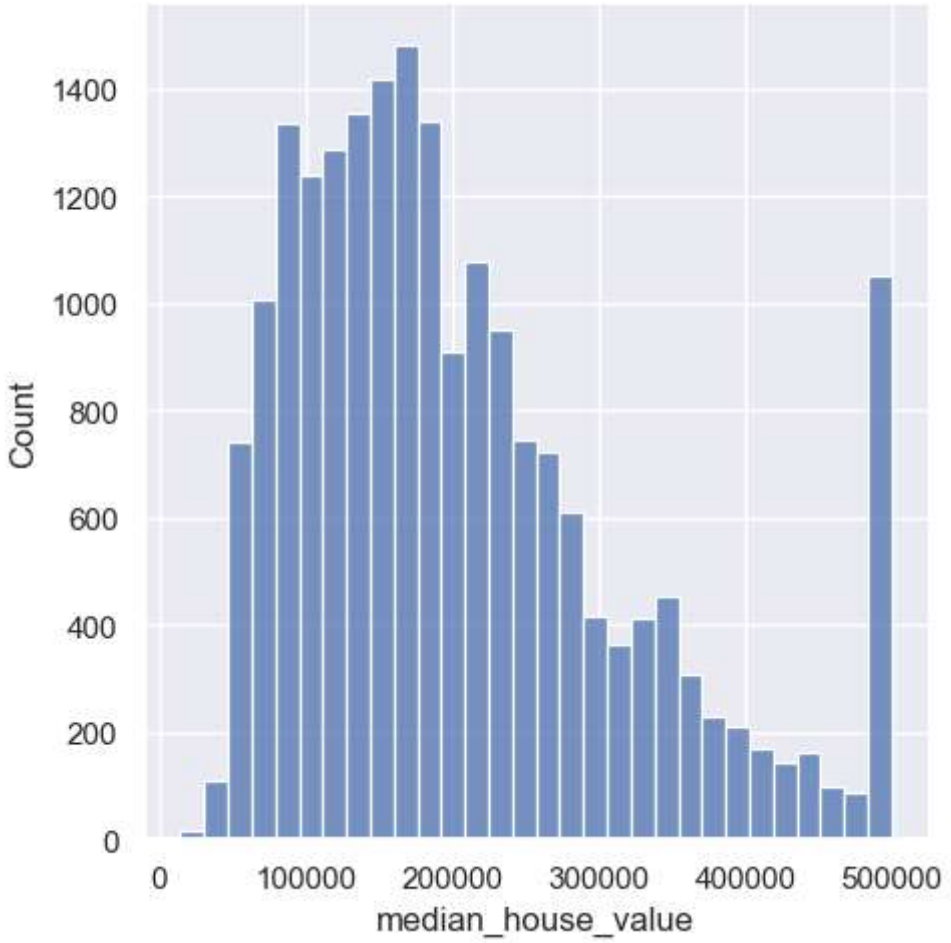
In [39]:
```
sns.boxplot(x="median_house_value",data=df)
```

Out[39]:
```
<Axes: xlabel='median_house_value'>
```



In [40]:
```
sns.set(rc={'figure.figsize':(11.7,8.27)})
sns.displot(df['median_house_value'],bins=30)
plt.show()
```

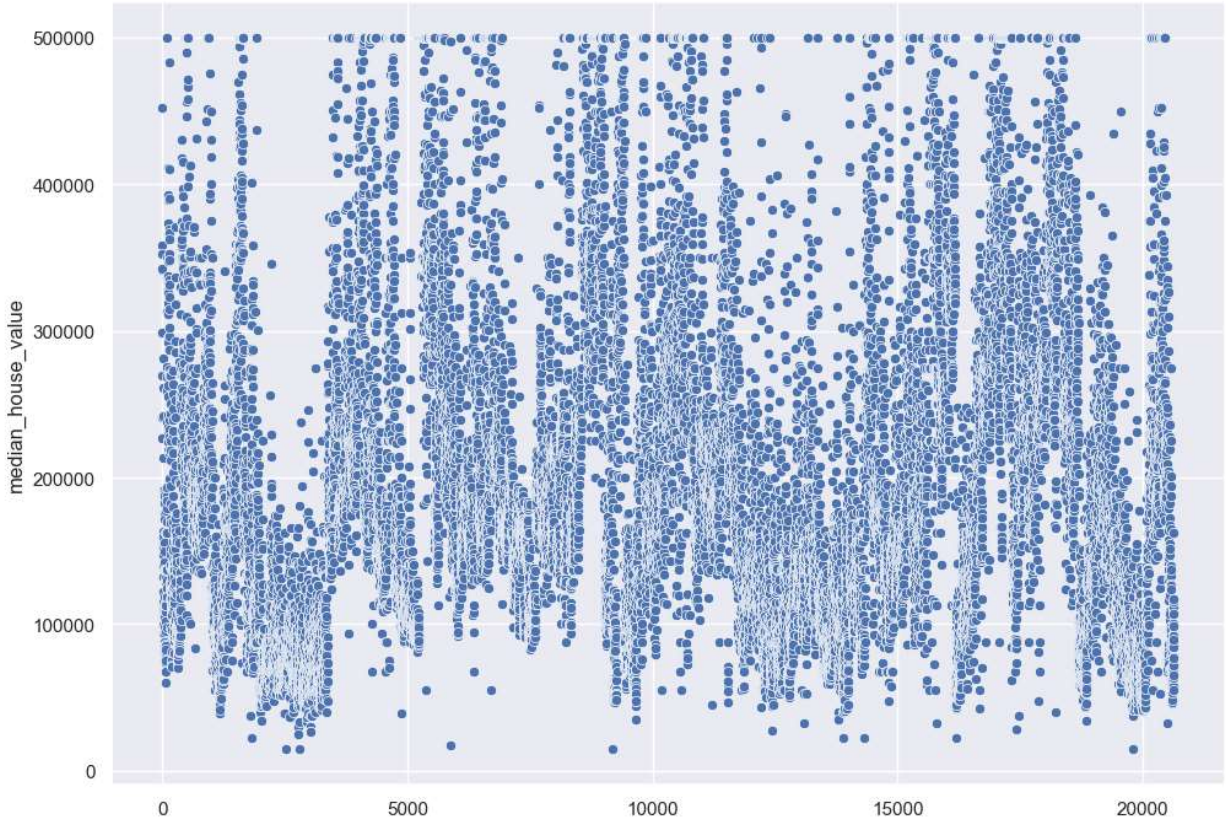
```
C:\Users\chand\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The
figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



Let's Findout the outlier in a dataset using Inter quartile Range(IQR

In [41]:
```
sns.scatterplot(data=df['median_house_value'])
```

Out[41]:
```
<Axes: ylabel='median_house_value'>
```



In [42]:
```
df["median_house_value"].value_counts()
```

Out[42]:
```
median_house_value
500001.0    958
137500.0    119
162500.0    116
112500.0    103
187500.0     92
           ...
359200.0      1
51200.0       1
39800.0       1
377600.0      1
47000.0       1
Name: count, Length: 3833, dtype: int64
```

In [43]:
```
df["median_house_value"].max()
```

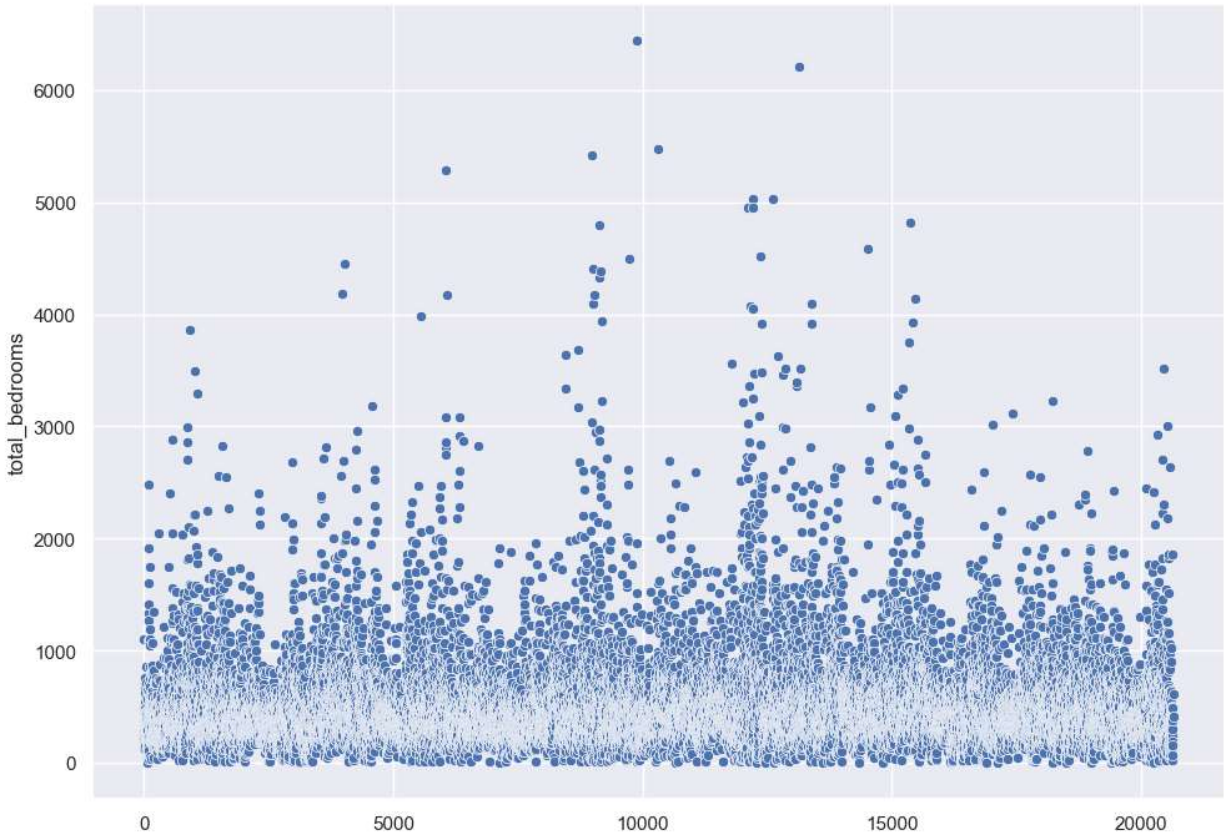
Out[43]:
```
500001.0
```

In [44]:
```
df["median_house_value"].describe()
```

Out[44]:
```
count     20433.000000
mean     206864.413155
std      115435.667099
min       14999.000000
25%      119500.000000
50%      179700.000000
75%      264700.000000
max      500001.000000
Name: median_house_value, dtype: float64
```

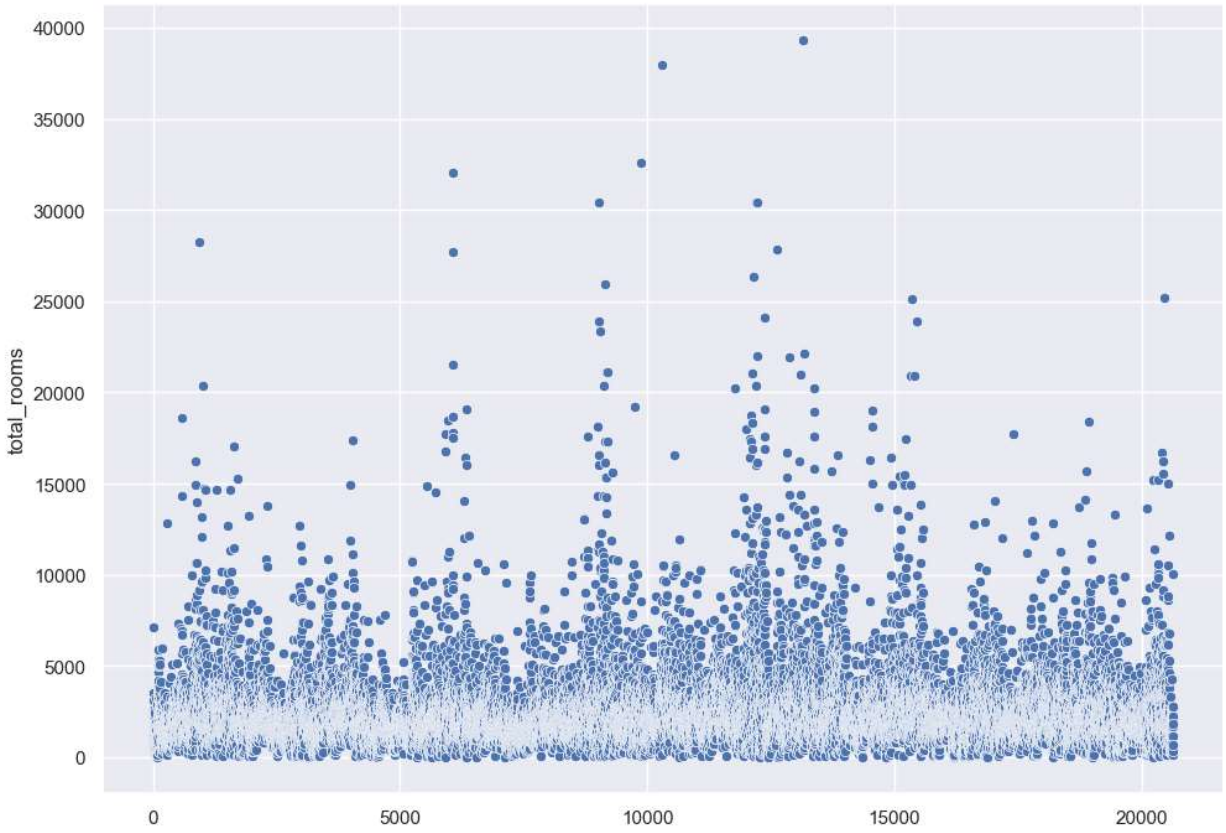
In [45]:
```
sns.scatterplot(data=df['total_bedrooms'])
```

Out[45]: <Axes: ylabel='total_bedrooms'>



In [46]:
```
sns.scatterplot(data=df['total_rooms'])
```

Out[46]: <Axes: ylabel='total_rooms'>



In [47]:
```
df["total_rooms"].value_counts()
```

Out[47]:
```
total_rooms
1527.0     18
1582.0     17
1613.0     17
2127.0     16
1471.0     15
           ..
539.0       1
6442.0      1
5367.0      1
5122.0      1
10035.0     1
Name: count, Length: 5911, dtype: int64
```

In [48]:
```
df["total_rooms"].max()
```

Out[48]:
```
39320.0
```

In [49]:
```
df["total_rooms"].describe()
```

Out[49]:
```
count    20433.000000
mean      2636.504233
std       2185.269567
min          2.000000
25%       1450.000000
50%       2127.000000
75%       3143.000000
max      39320.000000
Name: total_rooms, dtype: float64
```

so,we got

Q1=1450and Q3=3143

so,let's findout IQR

In [50]:
```
Q1=1450
Q3=3143
IQR=Q3-Q1
print("So,IQR of our total_rooms features is : " , IQR)
```

```
So,IQR of our total_rooms features is :  1693
```

In [51]:
```
High=Q3+1.5*IQR
Low=Q1-1.5*IQR
print("High value of our totalPrice Features is",High)
print("Low value of our totalPrice Features is",Low)
```

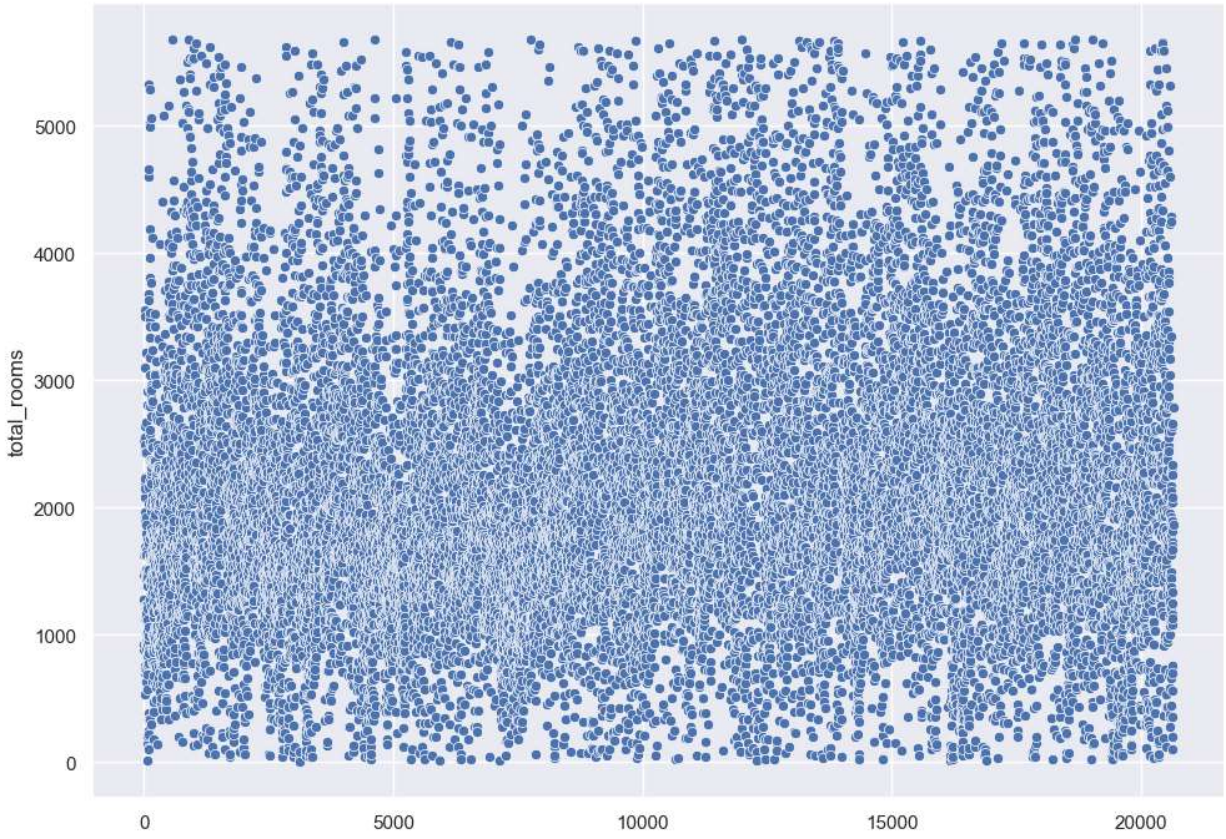
```
High value of our totalPrice Features is 5682.5
Low value of our totalPrice Features is -1089.5
```

In [52]:
```
df=df.copy()
```

In [53]:
```
df=df[df['total_rooms']<= High]
df=df[df['total_rooms']>=Low]
```

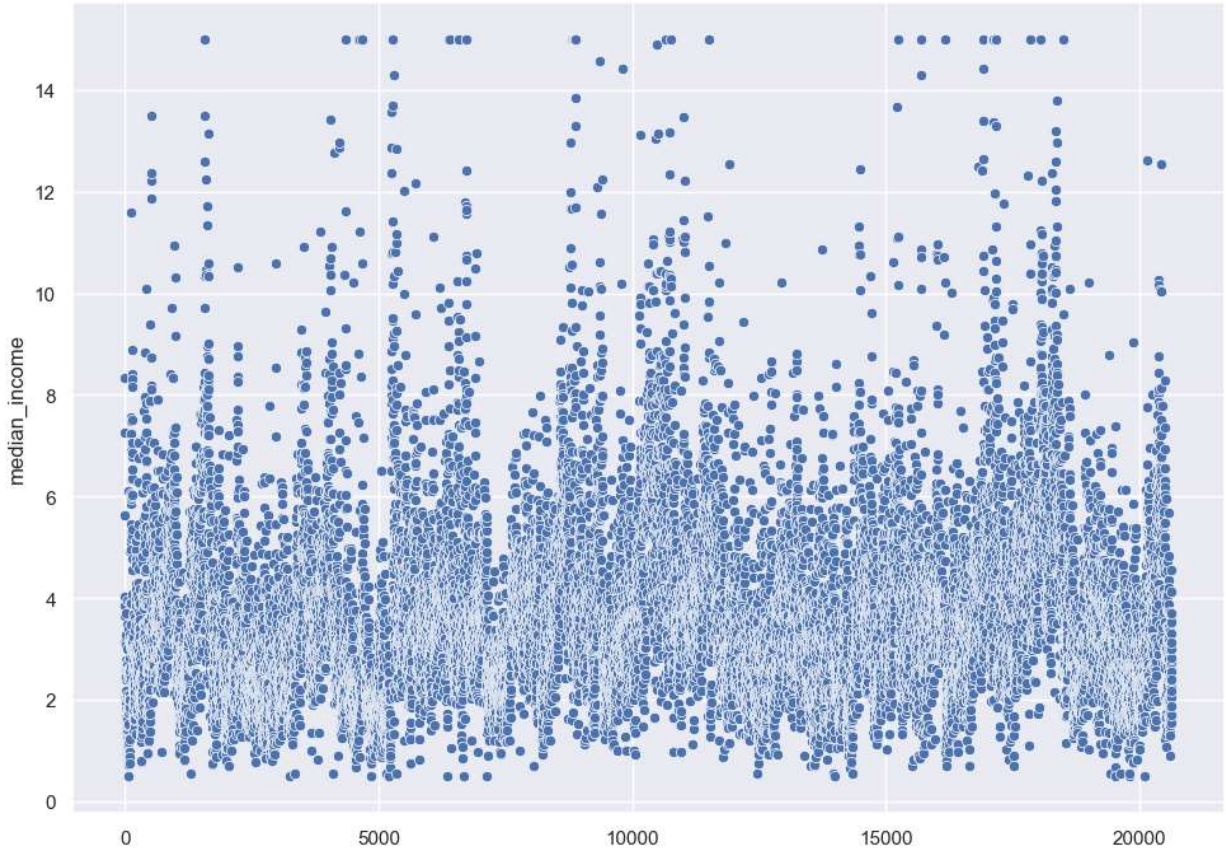
In [54]:
```
sns.scatterplot(data=df['total_rooms'])
```

Out[54]:
```
<Axes: ylabel='total_rooms'>
```



In [55]:
```
sns.scatterplot(data=df['median_income'])
```

Out[55]:
```
<Axes: ylabel='median_income'>
```



In [56]:
```
df
```

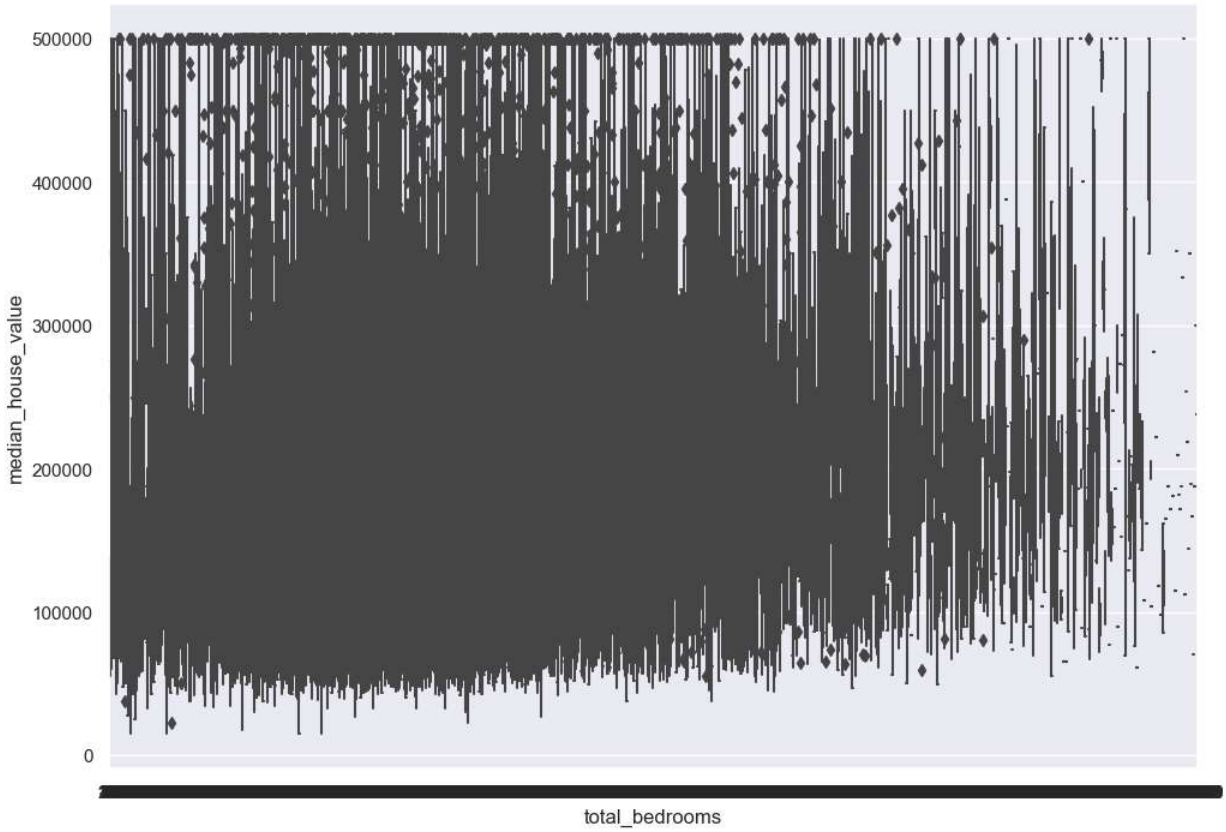
Out[56]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households
0	-122.23	37.88	41.0	880.0	129.0	322.0	126.(
2	-122.24	37.85	52.0	1467.0	190.0	496.0	177.(
3	-122.25	37.85	52.0	1274.0	235.0	558.0	219.(
4	-122.25	37.85	52.0	1627.0	280.0	565.0	259.(
5	-122.25	37.85	52.0	919.0	213.0	413.0	193.(
...	...	...	...	...	...	...	..
20635	-121.09	39.48	25.0	1665.0	374.0	845.0	330.(
20636	-121.21	39.49	18.0	697.0	150.0	356.0	114.(
20637	-121.22	39.43	17.0	2254.0	485.0	1007.0	433.(
20638	-121.32	39.43	18.0	1860.0	409.0	741.0	349.(
20639	-121.24	39.37	16.0	2785.0	616.0	1387.0	530.(

19143 rows × 14 columns

In [57]:
```
sns.boxplot(x=df["total_bedrooms"],y=df["median_house_value"])
```

Out[57]:
```
<Axes: xlabel='total_bedrooms', ylabel='median_house_value'>
```



Machine learning Model

In [58]:
```
from sklearn.model_selection import train_test_split
```

In [59]:
```
x=df.drop(["median_house_value"],axis=1)
y=df['median_house_value']
```

In [60]:
```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3)
```

In [61]:
```
x_train
```

Out[61]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	household
16824	-122.49	37.63	31.0	3109.0	621.0	1472.0	618.(
14480	-117.25	32.82	19.0	5255.0	762.0	1773.0	725.(
10257	-117.86	33.87	12.0	1600.0	251.0	685.0	256.(
13258	-117.69	34.10	17.0	3759.0	1035.0	1722.0	847.(
19318	-123.02	38.46	52.0	2154.0	499.0	524.0	259.(
...	...	...	...	...	...	...	..
15640	-122.42	37.80	52.0	3823.0	1040.0	1830.0	977.(
16158	-122.50	37.77	52.0	2433.0	454.0	1070.0	420.(
978	-121.87	37.57	13.0	5519.0	833.0	2444.0	825.(
15000	-117.04	32.74	33.0	3880.0	770.0	2288.0	805.(
19477	-120.97	37.67	31.0	1648.0	293.0	792.0	294.(

13400 rows × 13 columns

In [62]:
```
y_train
```

Out[62]:
```
16824    263900.0
14480    474000.0
10257    254000.0
13258    137500.0
19318    120000.0
           ...
15640    450000.0
16158    359500.0
978      393200.0
15000    140700.0
19477    121500.0
Name: median_house_value, Length: 13400, dtype: float64
```

In [63]:
```
x_test
```

Out[63]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	household
84	-122.28	37.81	35.0	948.0	184.0	467.0	169.(
3720	-118.42	34.20	27.0	3201.0	970.0	3403.0	948.(
17092	-122.23	37.47	39.0	5264.0	1259.0	3057.0	1265.(
20577	-121.94	38.89	15.0	1462.0	314.0	774.0	271.(
8313	-118.13	33.77	37.0	4365.0	926.0	1661.0	868.(
...	...	...	...	...	...	...	..
6105	-117.89	34.12	35.0	1566.0	321.0	1396.0	317.(
11990	-117.53	33.97	34.0	1293.0	215.0	774.0	217.(
5932	-117.83	34.15	20.0	2421.0	306.0	1023.0	298.(
3776	-118.42	34.16	25.0	2769.0	566.0	1201.0	545.(
5373	-118.38	34.04	36.0	3005.0	771.0	2054.0	758.(

5743 rows × 13 columns

In [64]:
```
train_data=x_train.join(y_train)
```

In [65]:
```
train_data
```

Out[65]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	household
16824	-122.49	37.63	31.0	3109.0	621.0	1472.0	618.(
14480	-117.25	32.82	19.0	5255.0	762.0	1773.0	725.(
10257	-117.86	33.87	12.0	1600.0	251.0	685.0	256.(
13258	-117.69	34.10	17.0	3759.0	1035.0	1722.0	847.(
19318	-123.02	38.46	52.0	2154.0	499.0	524.0	259.(
...	...	...	...	...	...	...	..
15640	-122.42	37.80	52.0	3823.0	1040.0	1830.0	977.(
16158	-122.50	37.77	52.0	2433.0	454.0	1070.0	420.(
978	-121.87	37.57	13.0	5519.0	833.0	2444.0	825.(
15000	-117.04	32.74	33.0	3880.0	770.0	2288.0	805.(
19477	-120.97	37.67	31.0	1648.0	293.0	792.0	294.(

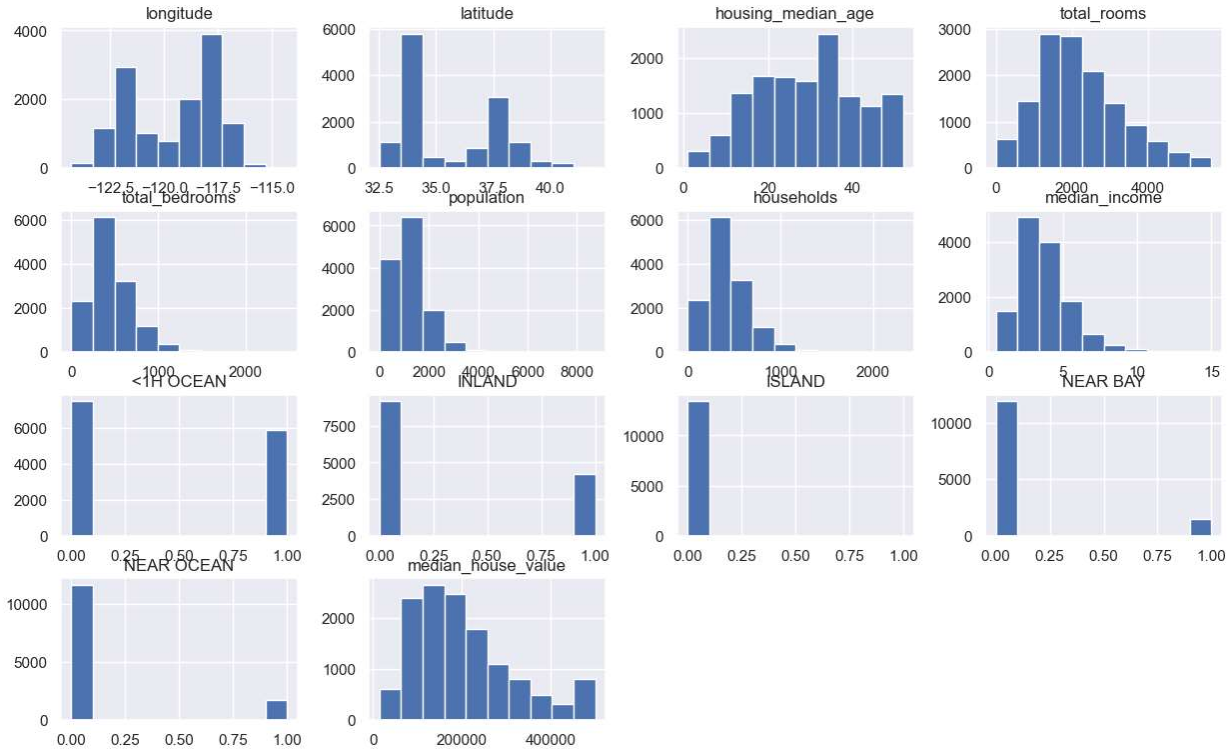
13400 rows × 14 columns

In [66]:
```
train_data.shape
```

Out[66]: (13400, 14)

In [67]:
```
train_data.hist(figsize=(15,9))
```

Out[67]:
```
array([[<Axes: title={'center': 'longitude'}>,
        <Axes: title={'center': 'latitude'}>,
        <Axes: title={'center': 'housing_median_age'}>,
        <Axes: title={'center': 'total_rooms'}>],
       [<Axes: title={'center': 'total_bedrooms'}>,
        <Axes: title={'center': 'population'}>,
        <Axes: title={'center': 'households'}>,
        <Axes: title={'center': 'median_income'}>],
       [<Axes: title={'center': '<1H OCEAN'}>,
        <Axes: title={'center': 'INLAND'}>,
        <Axes: title={'center': 'ISLAND'}>,
        <Axes: title={'center': 'NEAR BAY'}>],
       [<Axes: title={'center': 'NEAR OCEAN'}>,
        <Axes: title={'center': 'median_house_value'}>, <Axes: >,
        <Axes: >]], dtype=object)
```



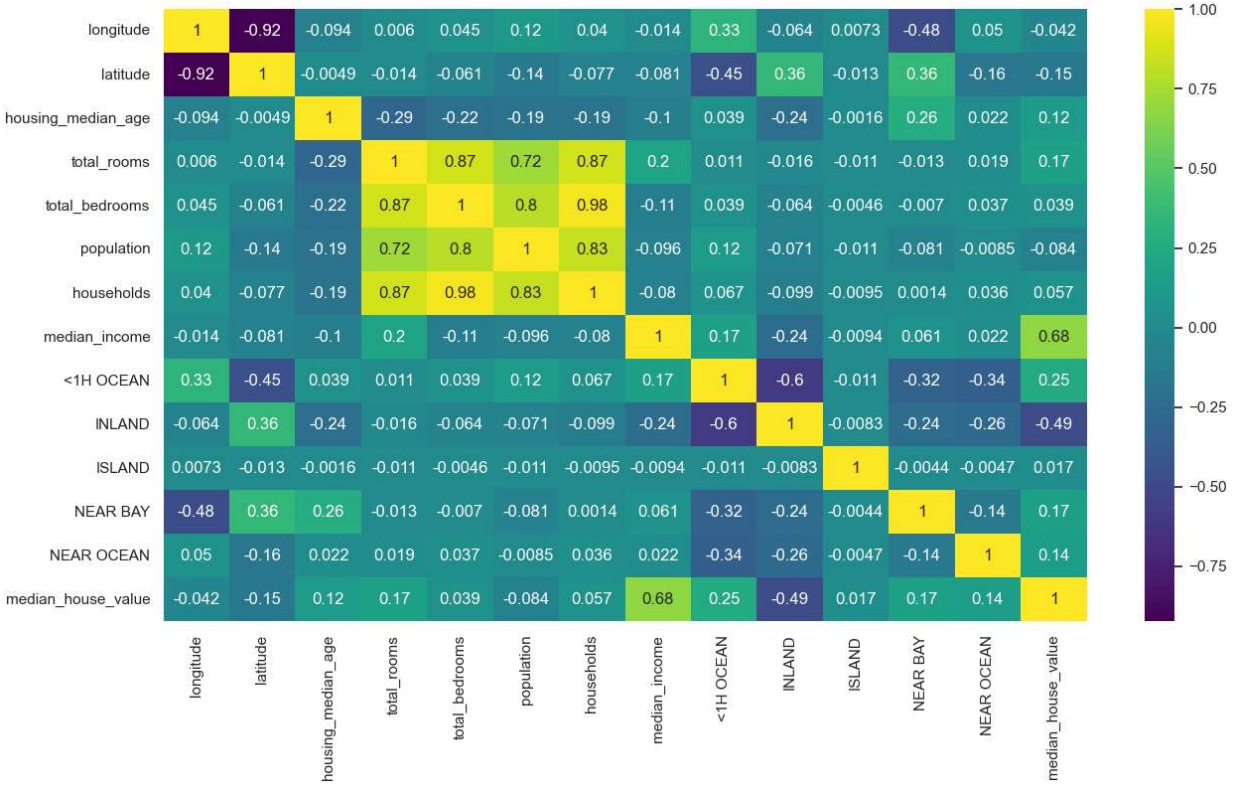
In [68]:
```
train_data.corr()
```

Out[68]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	populat
longitude	1.000000	-0.924432	-0.093777	0.005977	0.044518	0.116
latitude	-0.924432	1.000000	-0.004871	-0.013684	-0.060534	-0.141
housing_median_age	-0.093777	-0.004871	1.000000	-0.288954	-0.220263	-0.191
total_rooms	0.005977	-0.013684	-0.288954	1.000000	0.868814	0.724
total_bedrooms	0.044518	-0.060534	-0.220263	0.868814	1.000000	0.798
population	0.116353	-0.141475	-0.191953	0.724342	0.798597	1.000
households	0.040324	-0.076745	-0.194944	0.865120	0.976673	0.833
median_income	-0.014118	-0.081340	-0.099539	0.201949	-0.111591	-0.096
<1H OCEAN	0.329089	-0.452177	0.039051	0.010659	0.039258	0.124
INLAND	-0.064097	0.359291	-0.236858	-0.016398	-0.063505	-0.071
ISLAND	0.007340	-0.012934	-0.001569	-0.010958	-0.004600	-0.011
NEAR BAY	-0.475458	0.356514	0.263417	-0.012819	-0.007037	-0.081
NEAR OCEAN	0.050334	-0.164436	0.021725	0.019399	0.036620	-0.008
median_house_value	-0.042204	-0.149052	0.121417	0.167878	0.038536	-0.083

In [69]:
```
plt.figure(figsize=(15,8))
sns.heatmap(train_data.corr(),annot=True, cmap="viridis")
```

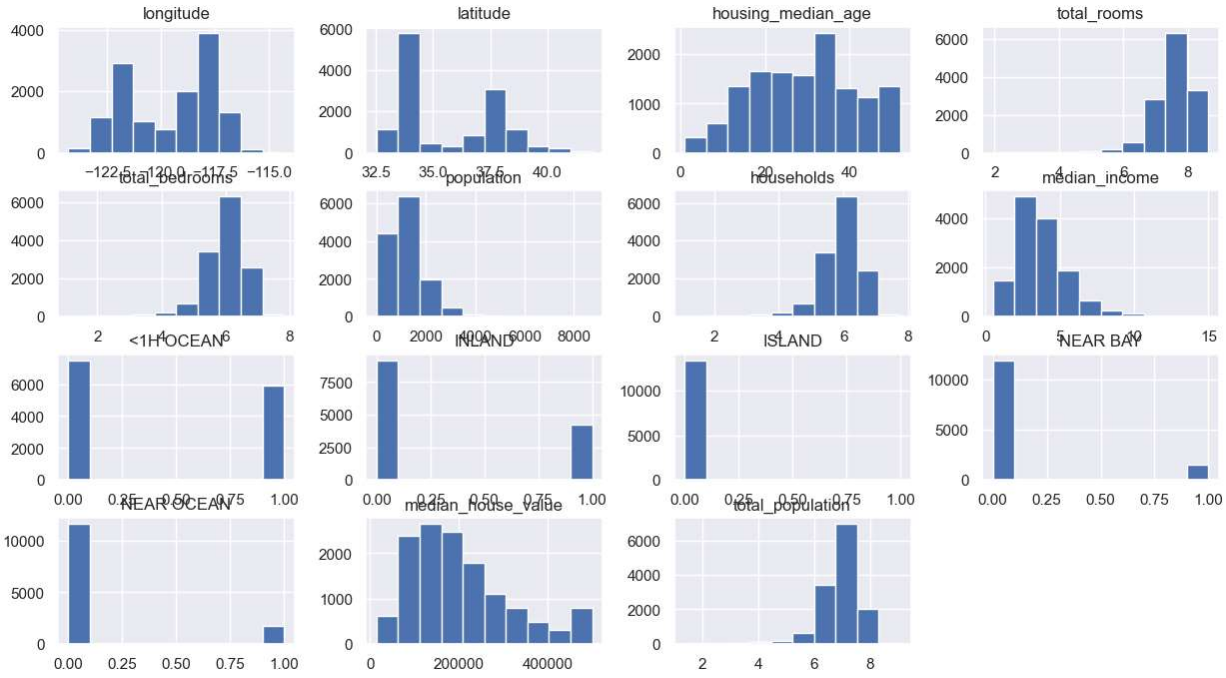
Out[69]: <Axes: >



In [70]:
```
train_data["total_rooms"] = np.log(train_data['total_rooms']+1)
train_data["total_bedrooms"] = np.log(train_data['total_bedrooms']+1)
train_data["total_population"] = np.log(train_data['population']+1)
train_data["households"] = np.log(train_data['households']+1)
```

In [71]:
```
train_data.hist(figsize=(15,8))
```

Out[71]:
```
array([[<Axes: title={'center': 'longitude'}>,
        <Axes: title={'center': 'latitude'}>,
        <Axes: title={'center': 'housing_median_age'}>,
        <Axes: title={'center': 'total_rooms'}>],
       [<Axes: title={'center': 'total_bedrooms'}>,
        <Axes: title={'center': 'population'}>,
        <Axes: title={'center': 'households'}>,
        <Axes: title={'center': 'median_income'}>],
       [<Axes: title={'center': '<1H OCEAN'}>,
        <Axes: title={'center': 'INLAND'}>,
        <Axes: title={'center': 'ISLAND'}>,
        <Axes: title={'center': 'NEAR BAY'}>],
       [<Axes: title={'center': 'NEAR OCEAN'}>,
        <Axes: title={'center': 'median_house_value'}>,
        <Axes: title={'center': 'total_population'}>, <Axes: >]],
      dtype=object)
```

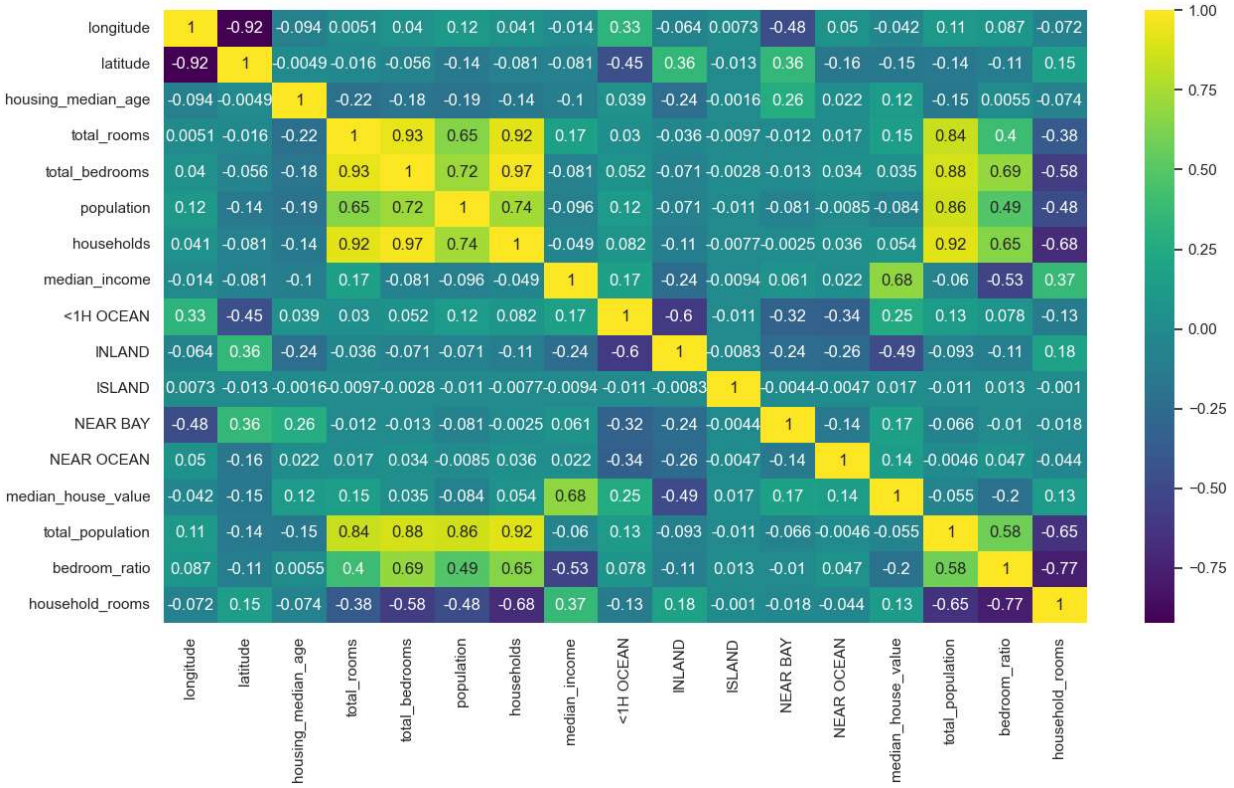


In [72]:
```
test_data=x_test.join(y_test)
```

In [73]:
```
train_data["bedroom_ratio"] =train_data["total_bedrooms"]/train_data['total_rooms']
train_data["household_rooms"]=train_data["total_rooms"]/train_data["households"]
```

In [74]:
```
plt.figure(figsize=(15,8))
sns.heatmap(train_data.corr(),annot=True, cmap="viridis")
```

Out[74]:
```
<Axes: >
```



In [75]:
```
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()



x_train,y_train = train_data.drop(["median_house_value"],axis=1),train_data["median_ho
x_train_s =scaler.fit_transform(x_train)

reg=LinearRegression()
reg.fit(x_train_s,y_train)
```

Out[75]:
```
▾ LinearRegression

LinearRegression()
```

In [76]:
```
test_data=x_test.join(y_test)

test_data["total_rooms"] = np.log(test_data['total_rooms']+1)
test_data["total_bedrooms"] = np.log(test_data['total_bedrooms']+1)
test_data["total_population"] = np.log(test_data['population']+1)
test_data["households"] = np.log(test_data['households']+1)

test_data["bedroom_ratio"] =test_data["total_bedrooms"]/test_data['total_rooms']
test_data["household_rooms"]=test_data["total_rooms"]/test_data["households"]

x_test,y_test = test_data.drop(["median_house_value"],axis=1),test_data["median_house_

x_test_s = scaler.transform(x_test)
```

In [77]:
```
test_data
```

Out[77]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households
84	-122.28	37.81	35.0	6.855409	5.220356	467.0	5.13579
3720	-118.42	34.20	27.0	8.071531	6.878326	3403.0	6.85540
17092	-122.23	37.47	39.0	8.568836	7.138867	3057.0	7.14361
20577	-121.94	38.89	15.0	7.288244	5.752573	774.0	5.60580
8313	-118.13	33.77	37.0	8.381603	6.831954	1661.0	6.76734
...	...	...	...	...	...	...	...
6105	-117.89	34.12	35.0	7.356918	5.774552	1396.0	5.76205
11990	-117.53	33.97	34.0	7.165493	5.375278	774.0	5.38449
5932	-117.83	34.15	20.0	7.792349	5.726848	1023.0	5.70044
3776	-118.42	34.16	25.0	7.926603	6.340359	1201.0	6.30261
5373	-118.38	34.04	36.0	8.008366	6.648985	2054.0	6.63200

5743 rows × 17 columns

In [78]:
```
train_data
```

Out[78]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households
16824	-122.49	37.63	31.0	8.042378	6.432940	1472.0	6.42810
14480	-117.25	32.82	19.0	8.567126	6.637258	1773.0	6.58755
10257	-117.86	33.87	12.0	7.378384	5.529429	685.0	5.54907
13258	-117.69	34.10	17.0	8.232174	6.943122	1722.0	6.74288
19318	-123.02	38.46	52.0	7.675546	6.214608	524.0	5.56068
...	...	...	...	...	...	...	...
15640	-122.42	37.80	52.0	8.249052	6.947937	1830.0	6.88551
16158	-122.50	37.77	52.0	7.797291	6.120297	1070.0	6.04263
978	-121.87	37.57	13.0	8.616133	6.726233	2444.0	6.71659
15000	-117.04	32.74	33.0	8.263848	6.647688	2288.0	6.69208
19477	-120.97	37.67	31.0	7.407924	5.683580	792.0	5.68697

13400 rows × 17 columns

In [79]:
```
reg.score(x_test_s,y_test)
```

Out[79]:
```
0.6601736611801438
```

In [80]:
```
from sklearn.ensemble import RandomForestRegressor
forest = RandomForestRegressor()
forest.fit(x_train_s,y_train)
```

Out[80]:
```
▾ RandomForestRegressor

RandomForestRegressor()
```

In [81]:
```
forest.score(x_test_s,y_test)
```

Out[81]:
```
0.8127711643626598
```

more than 0.7 score is strong correlation

In [82]:
```
predicted=forest.predict(x_test_s)
expected=y_test
```

In [83]:
```
predicted
```

Out[83]:
```
array([114260.  , 184198.  , 324650.01, ..., 366012.03, 322653.01,
       221970.01])
```

In [84]:
```
expected
```

Out[84]:
```
84       118800.0
3720     231700.0
17092    276600.0
20577     91700.0
8313     360700.0
           ...
6105     141300.0
11990    141000.0
5932     451500.0
3776     386100.0
5373     309100.0
Name: median_house_value, Length: 5743, dtype: float64
```

In [85]:
```
print("RMS: %s " % np.sqrt(np.mean((predicted - expected) ** 2)))
```

```
RMS: 49701.2521248034
```

In [ ]: