Decision Tree

In [7]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import OneHotEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from joblib import dump, load
```

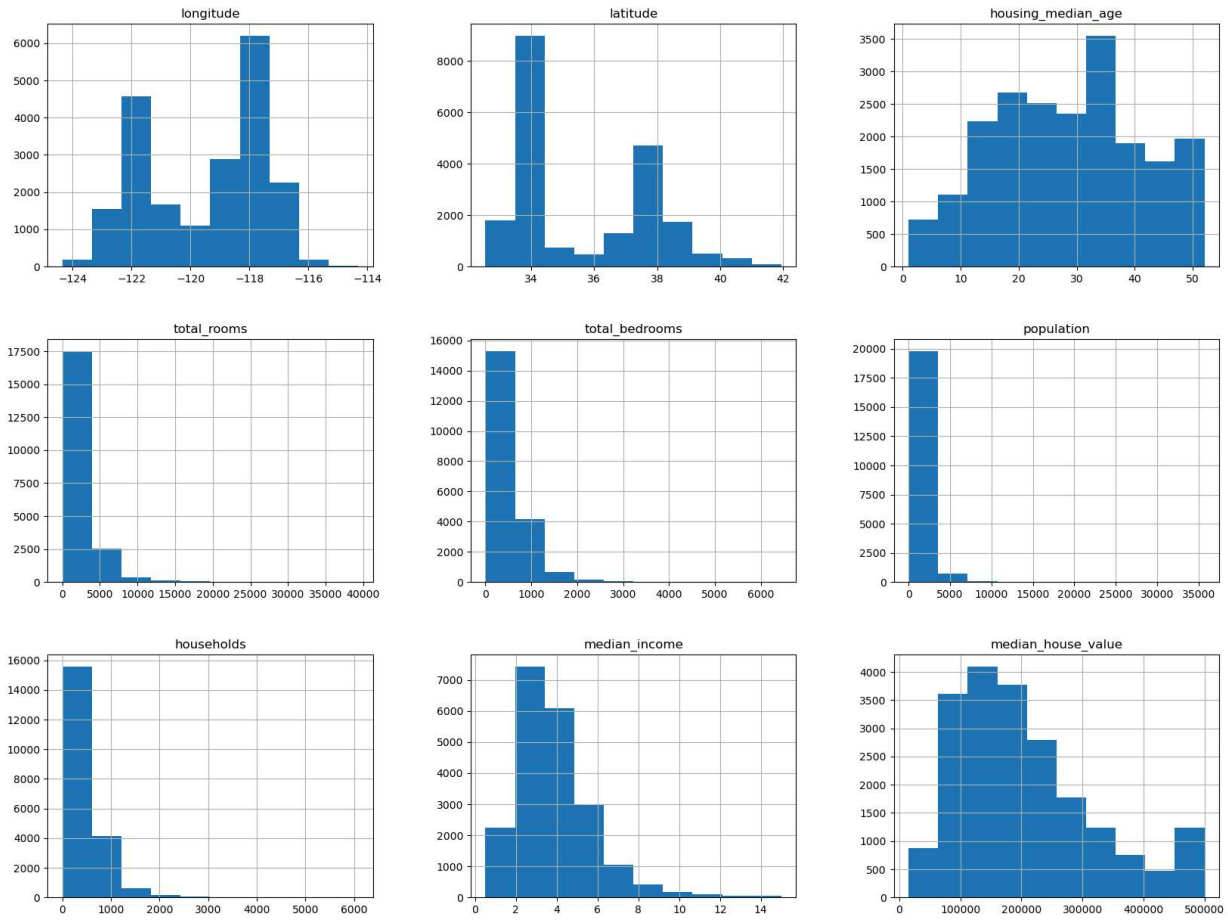
In [8]:
```
data = pd.read_csv(r"D:\New folder\dowenload\physics\machine learning\housing.csv
```

In [9]:
```
data
```

Out[9]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	househ
0	-122.23	37.88	41.0	880.0	129.0	322.0	1
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0	11
2	-122.24	37.85	52.0	1467.0	190.0	496.0	1
3	-122.25	37.85	52.0	1274.0	235.0	558.0	2
4	-122.25	37.85	52.0	1627.0	280.0	565.0	2
...	...	...	...	...	...	...	
20635	-121.09	39.48	25.0	1665.0	374.0	845.0	3
20636	-121.21	39.49	18.0	697.0	150.0	356.0	1
20637	-121.22	39.43	17.0	2254.0	485.0	1007.0	4
20638	-121.32	39.43	18.0	1860.0	409.0	741.0	3

In [10]:
```
data.hist(figsize=(20,15))
plt.show()
```



In [11]:
```
data_cat=data[["ocean_proximity"]]
encoder = OneHotEncoder()
encoded_data = encoder.fit_transform(data_cat)
```

In [ ]:
```

```

In [14]:
```
encoded_feature_names = encoder.get_feature_names_out(["ocean_proximity"])

encoded_df = pd.DataFrame(encoded_data.toarray(), columns=encoded_feature_names)

data_encoded = pd.concat([data, encoded_df], axis=1)

data_encoded.drop("ocean_proximity", axis=1, inplace=True)
```

In [15]:
```
data_encoded.isnull().sum()
```

Out[15]:
```
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_<1H OCEAN     0
ocean_proximity_INLAND        0
ocean_proximity_ISLAND        0
ocean_proximity_NEAR BAY      0
ocean_proximity_NEAR OCEAN    0
dtype: int64
```

In [26]:
```
median=data_encoded["total_bedrooms"].median()
data_encoded["total_bedrooms"].fillna(median,inplace=True)
```

In [17]:
```
data_encoded.duplicated().sum()
```

Out[17]:
```
0
```

In [18]:
```
X=data_encoded.drop('median_house_value', axis=1)
Y=data_encoded["median_house_value"]
```

In [19]:
```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2,random_st
```

In [20]:
```
model = RandomForestRegressor()
```

In [21]:
```
model.fit(X_train,Y_train)
```

Out[21]:
```
▾ RandomForestRegressor

RandomForestRegressor()
```

In [22]:
```
Y_pred = model.predict(X_test)
```

In [23]:
```
mse = mean_squared_error(Y_test, Y_pred)
rmse=np.sqrt(mse)
print("Root Mean Squared Error:",rmse)
```

```
Root Mean Squared Error: 48692.588455430385
```

In [28]:
```
dump(model, r"C:\Users\abohamam\Desktop\model.h5")
```

Out[28]:
```
['D:\\New folder\\dowenload\\physics\\machine learning\\New Text Document.h5']
```

In [ ]:

In [ ]:

In [ ]: