(PRIYA MORE 305C002)

```
import pandas as pd
import numpy as np
from sklearn import linear_model
from sklearn.model_selection import train_test_split
```

```
from sklearn.datasets import fetch_california_housing
housing = fetch_california_housing()
```

```
# Create a DataFrame from the dataset
df_x = pd.DataFrame(housing.data, columns=housing.feature_names)
```

```
print(housing)
```

{'data': array([[8.3252   ,   41.   ,   6.98412698, ...,   2.55555556,
           37.88   , -122.23   ],
        [8.3014   ,   21.   ,   6.23813708, ...,   2.10984183,
           37.86   , -122.22   ],
        [7.2574   ,   52.   ,   8.28813559, ...,   2.80225989,
           37.85   , -122.24   ],
        ...,
        [1.7   ,   17.   ,   5.20554273, ...,   2.3256351 ,
           39.43   , -121.22   ],
        [1.8672   ,   18.   ,   5.32951289, ...,   2.12320917,
           39.43   , -121.32   ],
        [2.3886   ,   16.   ,   5.25471698, ...,   2.61698113,
           39.37   , -121.24   ]]), 'target': array([4.526, 3.585, 3.521, ..., 0.923, 0.847, 0.894]), 'frame': None, 'target_na

```
df_y=pd.DataFrame(housing.target)
```

```
df_x
```

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude
0	8.3252	41.0	6.984127	1.023810	322.0	2.555556	37.88	-122.23
1	8.3014	21.0	6.238137	0.971880	2401.0	2.109842	37.86	-122.22
2	7.2574	52.0	8.288136	1.073446	496.0	2.802260	37.85	-122.24
3	5.6431	52.0	5.817352	1.073059	558.0	2.547945	37.85	-122.25
4	3.8462	52.0	6.281853	1.081081	565.0	2.181467	37.85	-122.25
...	...	...	...	...	...	...	...	...
20635	1.5603	25.0	5.045455	1.133333	845.0	2.560606	39.48	-121.09
20636	2.5568	18.0	6.114035	1.315789	356.0	3.122807	39.49	-121.21
20637	1.7000	17.0	5.205543	1.120092	1007.0	2.325635	39.43	-121.22
20638	1.8672	18.0	5.329513	1.171920	741.0	2.123209	39.43	-121.32
20639	2.3886	16.0	5.254717	1.162264	1387.0	2.616981	39.37	-121.24

20640 rows × 8 columns

```
df_y
```

	0
0	4.526
1	3.585
2	3.521
3	3.413
4	3.422
...	...
20635	0.781
20636	0.771
20637	0.923
20638	0.847
20639	0.894

20640 rows × 1 columns

```
df_x.describe()
```

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude
count	20640.000000	20640.000000	20640.000000	20640.000000	20640.000000	20640.000000	20640.000000	20640.000000
mean	3.870671	28.639486	5.429000	1.096675	1425.476744	3.070655	35.631861	-119.569704
std	1.899822	12.585558	2.474173	0.473911	1132.462122	10.386050	2.135952	2.003532
min	0.499900	1.000000	0.846154	0.333333	3.000000	0.692308	32.540000	-124.350000
25%	2.563400	18.000000	4.440716	1.006079	787.000000	2.429741	33.930000	-121.800000
50%	3.534800	29.000000	5.229129	1.048780	1166.000000	2.818116	34.260000	-118.490000
75%	4.743250	37.000000	6.052381	1.099526	1725.000000	3.282261	37.710000	-118.010000
max	15.000100	52.000000	141.909091	34.066667	35682.000000	1243.333333	41.950000	-114.310000

```
df_y.describe()
```

	0
count	20640.000000
mean	2.068558
std	1.153956
min	0.149990
25%	1.196000
50%	1.797000
75%	2.647250
max	5.000010

```
reg=linear_model.LinearRegression()
```

```
x_train,x_test,y_train,y_test=train_test_split(df_x,df_y,test_size=0.3,random_state=42)
```

```
reg.fit(x_train,y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
print(reg.coef_)
```

```
[[ 4.45822565e-01  9.68186799e-03 -1.22095112e-01  7.78599557e-01
  -7.75740400e-07 -3.37002667e-03 -4.18536747e-01 -4.33687976e-01]]
```

```
y_pred=reg.predict(x_test)
print(y_pred)
```

```
[[0.72604907]
 [1.76743383]
 [2.71092161]
 ...
 [2.07465531]
 [1.57371395]
 [1.82744133]]
```

```
print(y_test)
```

```
                0
20046  0.47700
3024   0.45800
15663  5.00001
20484  2.18600
9814   2.78000
...        ...
17505  2.37500
13512  0.67300
10842  2.18400
16559  1.19400
5786   2.09800

[6192 rows x 1 columns]
```

```
print(np.mean((y_pred-y_test)**2))
```

```
0.5305677824766758
```

```
from sklearn.metrics import mean_squared_error
print(mean_squared_error(y_test,y_pred))
```

```
0.5305677824766758
```

```
df_y.describe()
```

	0
count	20640.000000
mean	2.068558
std	1.153956
min	0.149990
25%	1.196000
50%	1.797000
75%	2.647250
max	5.000010