

LAB-2 1BM23CS134

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split, StratifiedShuffleSplit
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.pipeline import Pipeline
from sklearn.compose import ColumnTransformer
from sklearn.base import BaseEstimator, TransformerMixin

# Load dataset
housing = pd.read_csv("/content/sample_data/california_housing_train.csv")

housing.head()
```

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
0	-114.31	34.19	15.0	5612.0	1283.0	1015.0	472.0	1.4936	66900.0
1	-114.47	34.40	19.0	7650.0	1901.0	1129.0	463.0	1.8200	80100.0
2	-114.56	33.69	17.0	720.0	174.0	333.0	117.0	1.6509	85700.0
3	-114.57	33.64	14.0	1501.0	337.0	515.0	226.0	3.1917	73400.0
4	-114.57	33.57	20.0	1454.0	326.0	624.0	262.0	1.9250	65500.0

Next steps: [Generate code with housing](#) [New interactive sheet](#)

1. Perform the describe and info steps:

```
import pandas as pd

# Load dataset
housing = pd.read_csv("/content/sample_data/california_housing_train.csv")

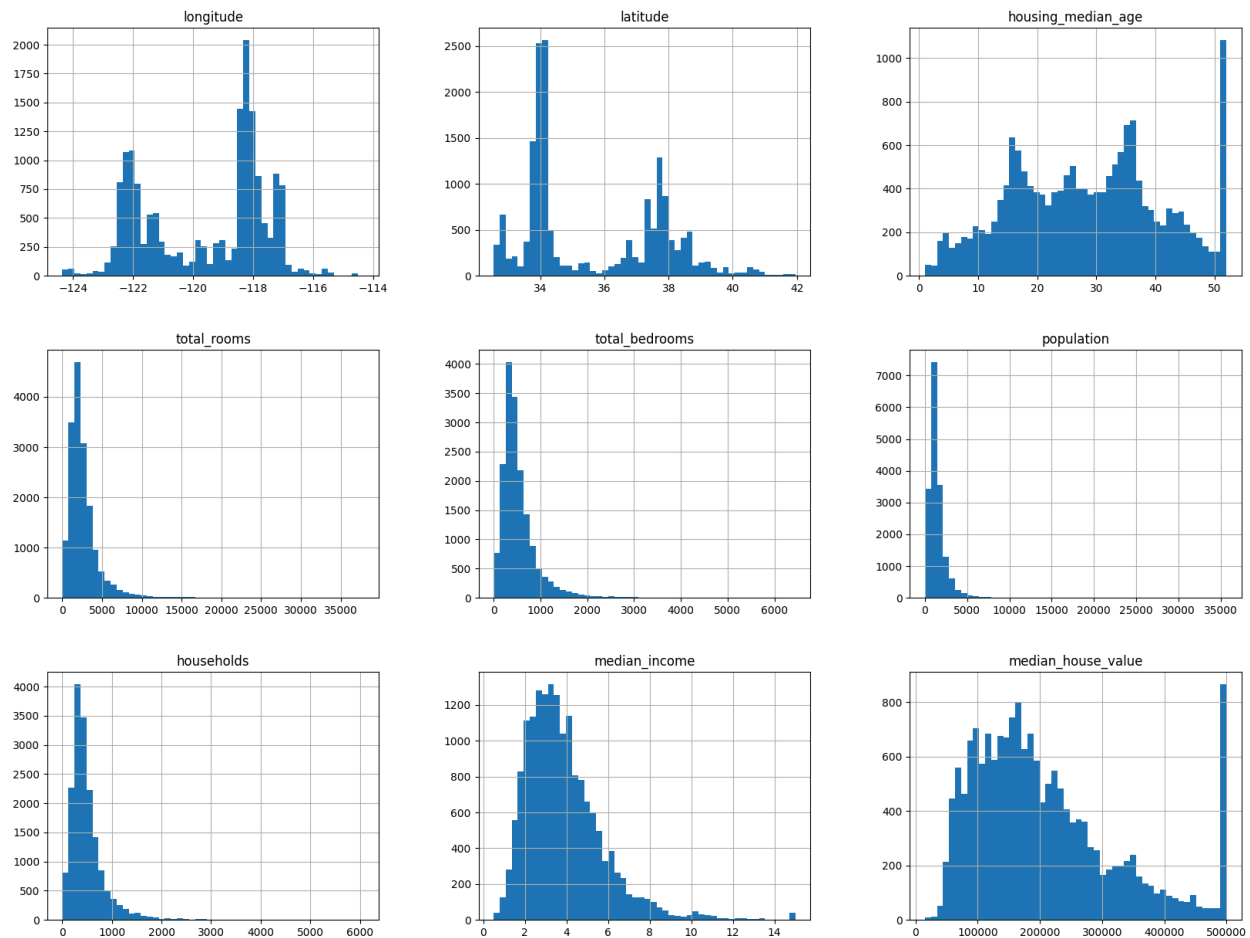
# Info
housing.info()

# Describe
housing.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17000 entries, 0 to 16999
Data columns (total 9 columns):
#   Column                Non-Null Count  Dtype  
---  -
0   longitude              17000 non-null float64
1   latitude               17000 non-null float64
2   housing_median_age     17000 non-null float64
3   total_rooms            17000 non-null float64
4   total_bedrooms        17000 non-null float64
5   population             17000 non-null float64
6   households             17000 non-null float64
7   median_income          17000 non-null float64
8   median_house_value     17000 non-null float64
dtypes: float64(9)
memory usage: 1.2 MB
```

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
count	17000.000000	17000.000000	17000.000000	17000.000000	17000.000000	17000.000000	17000.000000	17000.000000	17000.000000
mean	-119.562108	35.625225	28.589353	2643.664412	539.410824	1429.573941	501.221941	3.883578	207300.912353
std	2.005166	2.137340	12.586937	2179.947071	421.499452	1147.852959	384.520841	1.908157	115983.764387
min	-124.350000	32.540000	1.000000	2.000000	1.000000	3.000000	1.000000	0.499900	14999.000000
25%	-121.790000	33.930000	18.000000	1462.000000	297.000000	790.000000	282.000000	2.566375	119400.000000
50%	-118.490000	34.250000	29.000000	2127.000000	434.000000	1167.000000	409.000000	3.544600	180400.000000

2. Plot the histogram of each feature(Indicate what does histogram indicate on median_income and house_median_age)



3. Demonstrate the process of creating a test set(write the difference between random and stratified test set)

```
from sklearn.model_selection import train_test_split
import pandas as pd

# Load dataset
housing = pd.read_csv("/content/sample_data/california_housing_train.csv")

# Random split
train_set, test_set = train_test_split(housing, test_size=0.2, random_state=42)

print("Training set size:", len(train_set))
print("Test set size:", len(test_set))
```

... Training set size: 13600
Test set size: 3400

```
import numpy as np
from sklearn.model_selection import StratifiedShuffleSplit

# Create income category for stratification
housing["income_cat"] = pd.cut(
    housing["median_income"],
    bins=[0., 1.5, 3.0, 4.5, 6., np.inf],
    labels=[1,2,3,4,5]
)

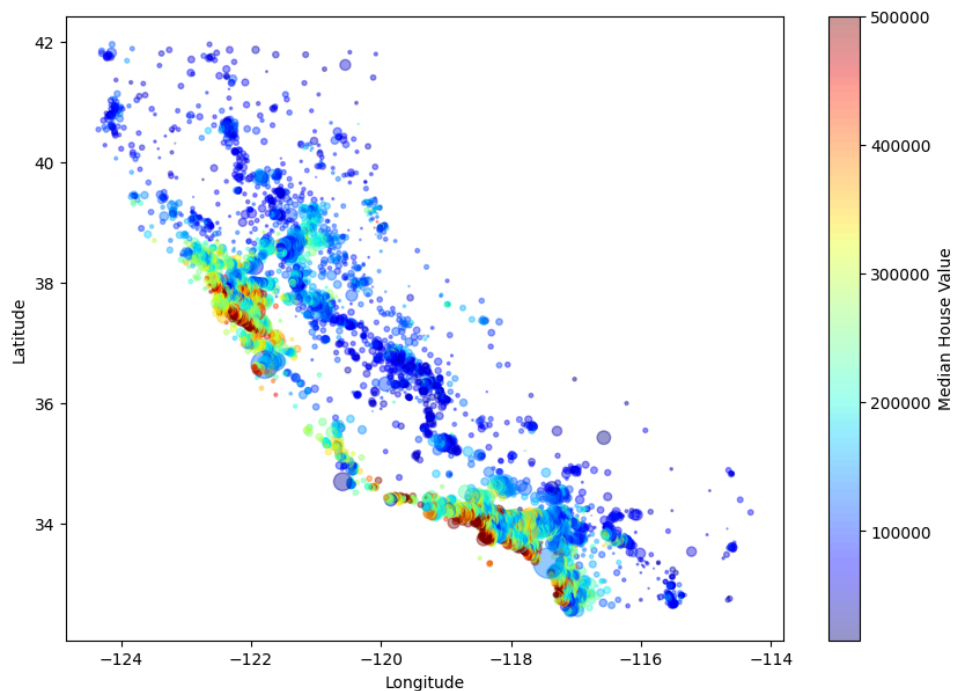
# Stratified split
split = StratifiedShuffleSplit(n_splits=1, test_size=0.2, random_state=42)

for train_index, test_index in split.split(housing, housing["income_cat"]):
    strat_train_set = housing.loc[train_index]
    strat_test_set = housing.loc[test_index]

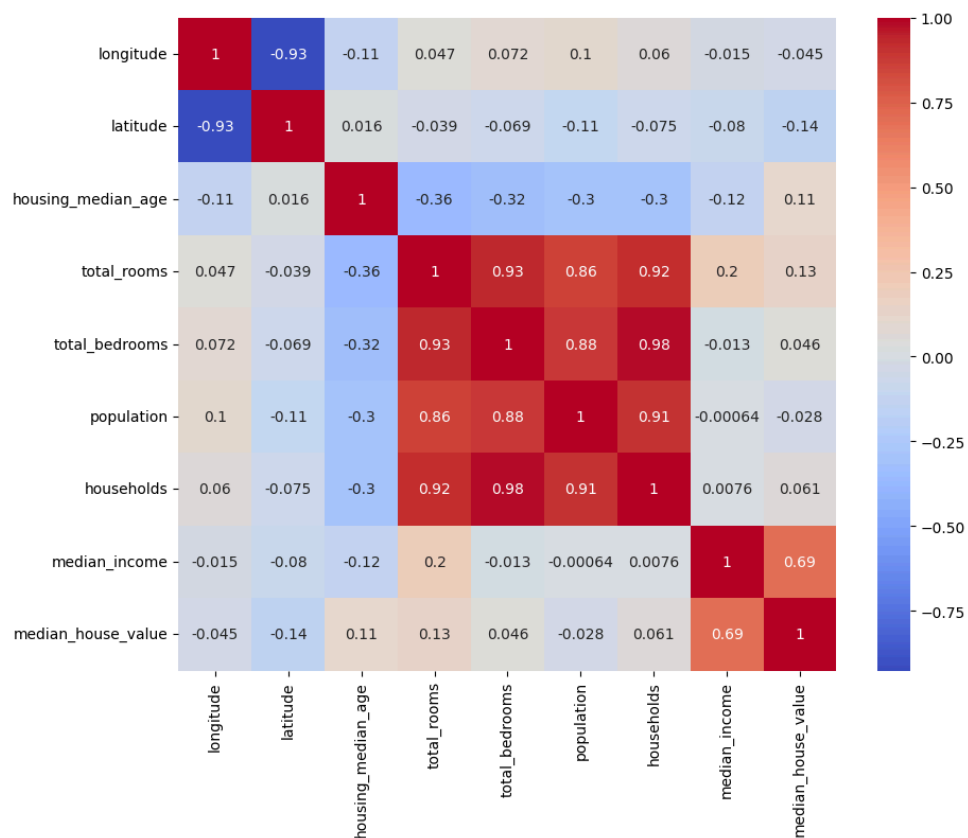
print("Stratified Training set size:", len(strat_train_set))
print("Stratified Test set size:", len(strat_test_set))
```

```
Stratified Training set size: 13600
Stratified Test set size: 3400
```

4. List the geographical features from the dataset and plot a graph to Visualize Geographical Data(what does the graph indicate w.r.t housing prices and location)



5. Plot a graph to show features correlation with housing price. Which feature correlates to the maximum. Plot the graph for that with housing price and analyze what the graph indicate



6. List the features that could be combined to improve correlation and plot again to see if correlation has improved

	median_house_value
median_house_value	1.000000
median_income	0.691871
rooms_per_household	0.150081
total_rooms	0.130991

housing_median_age	0.106758
households	0.061031
total_bedrooms	0.045783
population	-0.027850
population_per_household	-0.042764
longitude	-0.044982
latitude	-0.144917
bedrooms_per_room	-0.258190

dtype: float64

7. List the features that needs to be cleaned and demonstrate the process of cleaning

```

[[-114.31    34.19    15.    ... 11.88983051
   0.22861725  2.15042373]
 [-114.47    34.4     19.    ... 16.52267819
   0.24849673  2.43844492]
 [-114.56    33.69    17.    ...  6.15384615
   0.24166667  2.84615385]
 ...
 [-124.3     41.84    17.    ...  5.87061404
   0.19835637  2.72807018]
 [-124.3     41.8     19.    ...  5.58995816
   0.20658683  2.71548117]
 [-124.35    40.54    52.    ...  6.74074074
   0.16483516  2.98518519]]

```

8. Is there any categorical data that needs to be converted to numerical? If so explain the method used to convert and code the same and show the output.

NO

9. Discuss the importance of feature scaling

```
[[ 2.61936500e+00 -6.71520235e-01 -1.07967114e+00 ... 2.54055909e+00
 2.74245966e-01 -2.04549373e-01]
 [ 2.53956878e+00 -5.73264367e-01 -7.61872011e-01 ... 4.36514639e+00
 6.18054917e-01 -1.33216731e-01]
 [ 2.49468340e+00 -9.05462777e-01 -9.20771577e-01 ... 2.81515626e-01
 4.99931243e-01 -3.22416800e-02]
 ...
 [-2.36291168e+00  2.90780067e+00 -9.20771577e-01 ... 1.69968309e-01
 -2.49105729e-01 -6.14868203e-02]
 [-2.36291168e+00  2.88908527e+00 -7.61872011e-01 ... 5.94356115e-02
 -1.06762750e-01 -6.46046704e-02]
 [-2.38784800e+00  2.29955006e+00  1.85997083e+00 ... 5.12656490e-01
 -8.28843637e-01  2.19145536e-03]]
```

10. Design a pipeline inculcating (Custom transform, feature scaling and encoding). Explain how it works

```
[[11.88983051  2.15042373  0.22861725]
 [16.52267819  2.43844492  0.24849673]
 [ 6.15384615  2.84615385  0.24166667]
 [ 6.64159292  2.27876106  0.22451699]
 [ 5.54961832  2.38167939  0.22420908]]
```

```
Pipeline(steps=[('imputer', SimpleImputer(strategy='median')),
                 ('attrs_adder', CombinedAttributesAdder()),
                 ('scaler', StandardScaler())])
```