Attribution

Derived from:

- version two of book by Aurelien Geron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition (https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/)

There is a later version of this book (Version 3 in year 2022)

- can't verify that this example is still present

Original file name `02_end_to_end_machine_learning_project.ipynb`

Modified by Ken Perry

Changes

- drop non-numeric attribute before taking correlation
- updated deprecated `sklearn` function arguments

Warning:

Chapter 2 – End-to-end Machine Learning project

Welcome to Machine Learning Housing Corp.! Your task is to predict median house values in Californian districts, given a number of features from these districts.

This notebook contains all the sample code and solutions to the exercices in chapter 2.

Open in Colab

(https://colab.research.google.com/github/ageron/handson-ml2/blob/master/02_end_to_end_machine_learning_project.ipynb)

Open in Kaggle

(https://kaggle.com/kernels/welcome?src=https://github.com/ageron/handson-ml2/blob/master/02_end_to_end_machine_learning_project.ipynb)

Setup

First, let's import a few common modules, ensure MatplotLib plots figures inline and prepare a function to save the figures. We also check that Python 3.5 or later is installed (although Python 2.x may work, it is deprecated so we strongly recommend you use Python 3 instead), as well as Scikit-Learn ≥0.20.

In [1]:

```
# Python ≥3.5 is required
import sys
assert sys.version_info >= (3, 5)

# Scikit-Learn ≥0.20 is required
import sklearn
assert sklearn.__version__ >= "0.20"

# Common imports
import numpy as np
import os

# To plot pretty figures
%matplotlib inline
import matplotlib as mpl
import matplotlib.pyplot as plt
mpl.rc('axes', labelsize=14)
mpl.rc('xtick', labelsize=12)
mpl.rc('ytick', labelsize=12)

# Where to save the figures
PROJECT_ROOT_DIR = "."
CHAPTER_ID = "end_to_end_project"
IMAGES_PATH = os.path.join(PROJECT_ROOT_DIR, "images", CHAPTER_ID)
os.makedirs(IMAGES_PATH, exist_ok=True)

def save_fig(fig_id, tight_layout=True, fig_extension="png", resolution=300):
    path = os.path.join(IMAGES_PATH, fig_id + "." + fig_extension)
    print("Saving figure", fig_id)
    if tight_layout:
        plt.tight_layout()
    plt.savefig(path, format=fig_extension, dpi=resolution)
```

Get the Data

Download the Data

In [2]:

```
import os
import tarfile
import urllib.request

DOWNLOAD_ROOT = "https://raw.githubusercontent.com/ageron/handson-ml2/master/"
HOUSING_PATH = os.path.join("datasets", "housing")
HOUSING_URL = DOWNLOAD_ROOT + "datasets/housing/housing.tgz"

def fetch_housing_data(housing_url=HOUSING_URL, housing_path=HOUSING_PATH):
    if not os.path.isdir(housing_path):
        os.makedirs(housing_path)
    tgz_path = os.path.join(housing_path, "housing.tgz")
    urllib.request.urlretrieve(housing_url, tgz_path)
    housing_tgz = tarfile.open(tgz_path)
    housing_tgz.extractall(path=housing_path)
    housing_tgz.close()
```

In [3]:

```
fetch_housing_data()
```

```
/tmp/ipython-input-1387014147.py:15: DeprecationWarning: Python 3.14 will, by
default, filter extracted tar archives and reject files or modify their metada
ta. Use the filter argument to control this behavior.
  housing_tgz.extractall(path=housing_path)
```

In [4]:
```python
import pandas as pd

def load_housing_data(housing_path=HOUSING_PATH):
    csv_path = os.path.join(housing_path, "housing.csv")
    return pd.read_csv(csv_path)
```

Take a Quick Look at the Data Structure

In [5]:
```python
housing = load_housing_data()
housing.head()
```

Out[5]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
0	-122.23	37.88	41.0	880.0	129.0	322.0	126.0	8.3252	452600.0
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0	1138.0	8.3014	358500.0
2	-122.24	37.85	52.0	1467.0	190.0	496.0	177.0	7.2574	352100.0
3	-122.25	37.85	52.0	1274.0	235.0	558.0	219.0	5.6431	341300.0
4	-122.25	37.85	52.0	1627.0	280.0	565.0	259.0	3.8462	342200.0

In [6]:
```
housing.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 [7]:
```
housing["ocean_proximity"].value_counts()
```

Out[7]:

	count
ocean_proximity	
<1H OCEAN	9136
INLAND	6551
NEAR OCEAN	2658
NEAR BAY	2290
ISLAND	5

dtype: int64

Statistical characterization of attributes

- added by Ken Perry

Observe the **units** of

- `total_rooms`
- `total_bedrooms`

These are rooms *for the entire geographic area*

- **not** average per home !

Probably want to apply a "normalization" transformation to them

In [8]:

```
housing.describe()
```

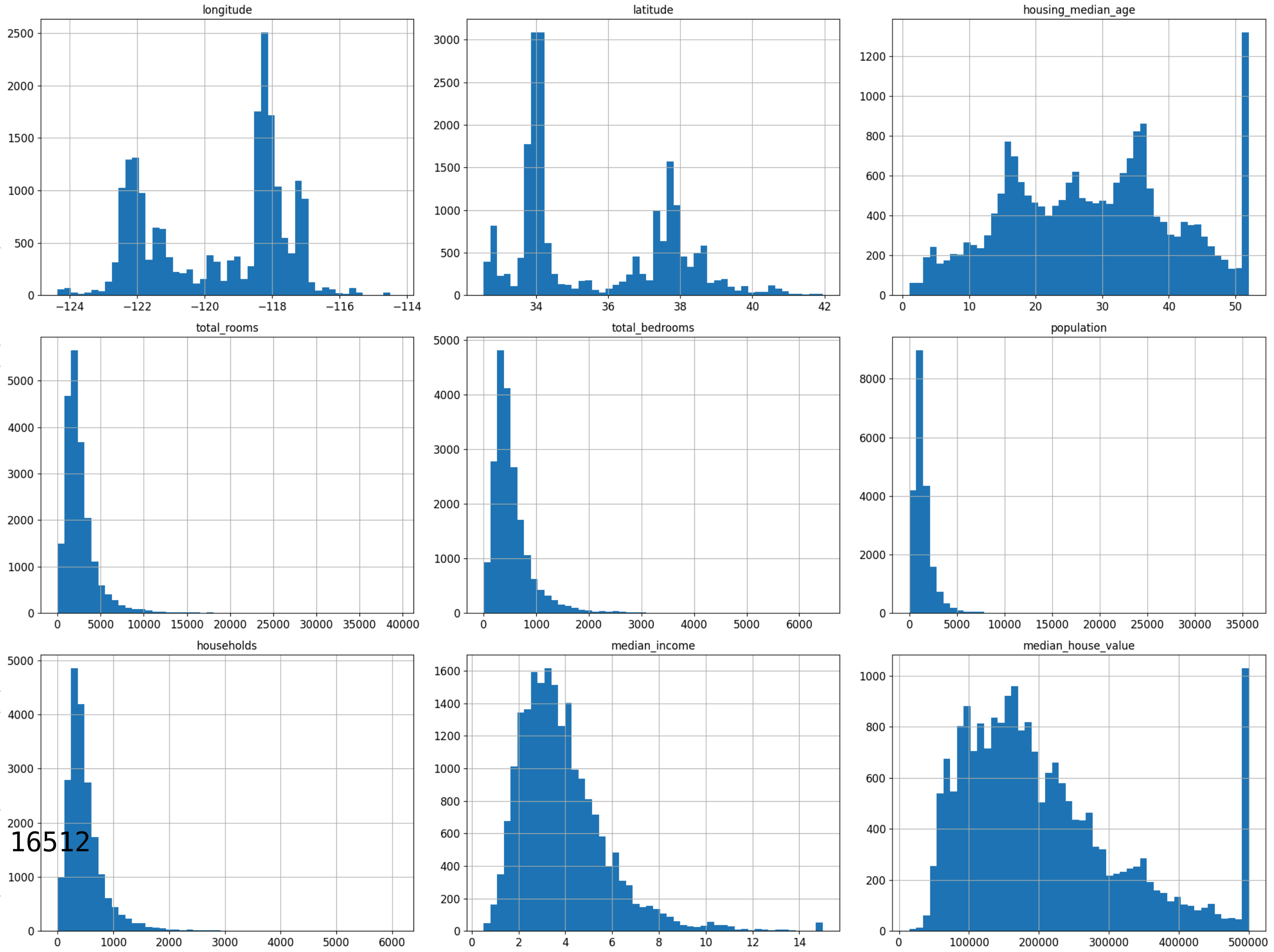
Out[8]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_incor
count	20640.000000	20640.000000	20640.000000	20640.000000	20433.000000	20640.000000	20640.000000	20640.00000
mean	-119.569704	35.631861	28.639486	2635.763081	537.870553	1425.476744	499.539680	3.870671
std	2.003532	2.135952	12.585558	2181.615252	421.385070	1132.462122	382.329753	1.899822
min	-124.350000	32.540000	1.000000	2.000000	1.000000	3.000000	1.000000	0.499900
25%	-121.800000	33.930000	18.000000	1447.750000	296.000000	787.000000	280.000000	2.563400
50%	-118.490000	34.260000	29.000000	2127.000000	435.000000	1166.000000	409.000000	3.534800
75%	-118.010000	37.710000	37.000000	3148.000000	647.000000	1725.000000	605.000000	4.743250
max	-114.310000	41.950000	52.000000	39320.000000	6445.000000	35682.000000	6082.000000	15.000100

In [9]:
```
%matplotlib inline
import matplotlib.pyplot as plt
housing.hist(bins=50, figsize=(20,15))
save_fig("attribute_histogram_plots")
plt.show()
```

```
Saving figure attribute_histogram_plots
```

In [10]:



In [11]:

In [12]:

Out[12]: 16512

In [13]:

Out[13]: 4128

In [14]:
```python
from zlib import crc32

def test_set_check(identifier, test_ratio):
    return crc32(np.int64(identifier)) & 0xffffffff < test_ratio * 2**32

def split_train_test_by_id(data, test_ratio, id_column):
    ids = data[id_column]
    in_test_set = ids.apply(lambda id_: test_set_check(id_, test_ratio))
    return data.loc[~in_test_set], data.loc[in_test_set]
```

The implementation of `test_set_check()` above works fine in both Python 2 and Python 3. In earlier releases, the following implementation was proposed, which supported any hash function, but was much slower and did not support Python 2:

In [15]:
```python
import hashlib

def test_set_check(identifier, test_ratio, hash=hashlib.md5):
    return hash(np.int64(identifier)).digest()[-1] < 256 * test_ratio
```

If you want an implementation that supports any hash function and is compatible with both Python 2 and Python 3, here is one:

In [16]:
```python
def test_set_check(identifier, test_ratio, hash=hashlib.md5):
    return bytearray(hash(np.int64(identifier)).digest())[-1] < 256 * test_ratio
```

In [17]:

```python
housing_with_id = housing.reset_index()   # adds an `index` column
train_set, test_set = split_train_test_by_id(housing_with_id, 0.2, "index")
```

In [18]:

```python
housing_with_id["id"] = housing["longitude"] * 1000 + housing["latitude"]
train_set, test_set = split_train_test_by_id(housing_with_id, 0.2, "id")
```

In [19]:

```python
test_set.head()
```

Out[19]:

	index	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_hous
8	8	-122.26	37.84	42.0	2555.0	665.0	1206.0	595.0	2.0804	226700.0
10	10	-122.26	37.85	52.0	2202.0	434.0	910.0	402.0	3.2031	281500.0
11	11	-122.26	37.85	52.0	3503.0	752.0	1504.0	734.0	3.2705	241800.0
12	12	-122.26	37.85	52.0	2491.0	474.0	1098.0	468.0	3.0750	213500.0
13	13	-122.26	37.84	52.0	696.0	191.0	345.0	174.0	2.6736	191300.0

In [20]:

```python
from sklearn.model_selection import train_test_split

train_set, test_set = train_test_split(housing, test_size=0.2, random_state=42)
```

In [21]:

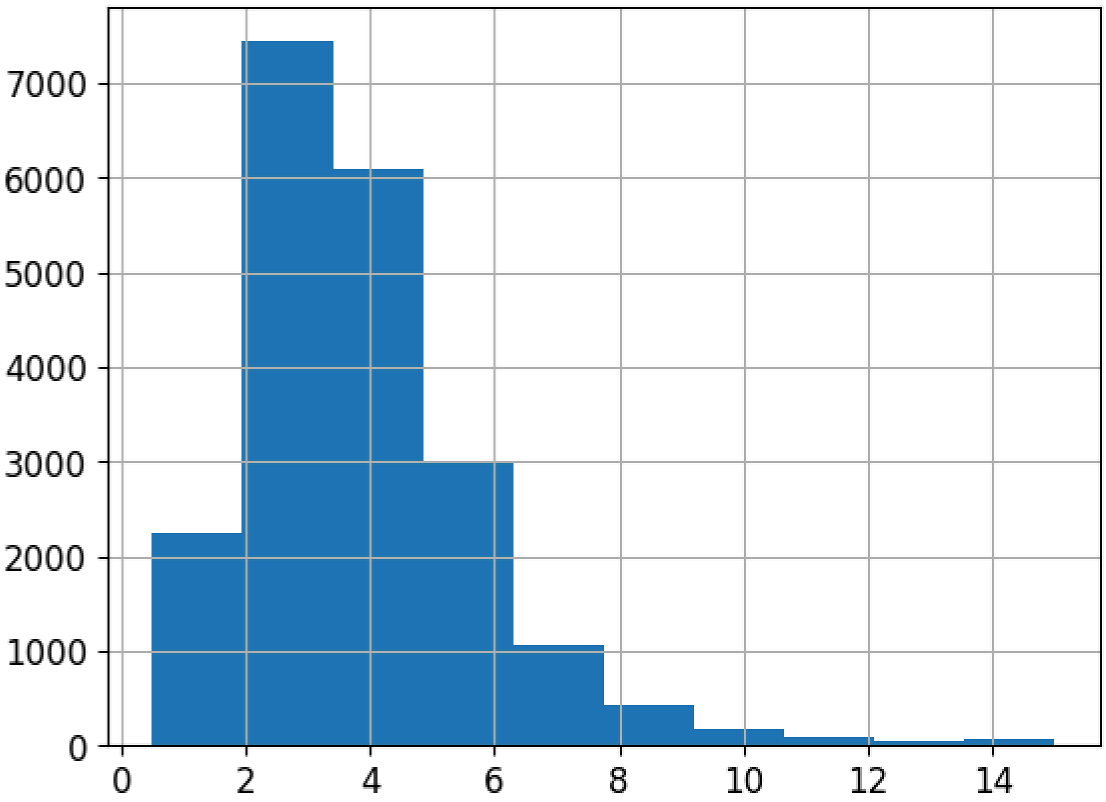
```python
test_set.head()
```

Out[21]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_va
20046	-119.01	36.06	25.0	1505.0	NaN	1392.0	359.0	1.6812	47700.0
3024	-119.46	35.14	30.0	2943.0	NaN	1565.0	584.0	2.5313	45800.0
15663	-122.44	37.80	52.0	3830.0	NaN	1310.0	963.0	3.4801	500001.0
20484	-118.72	34.28	17.0	3051.0	NaN	1705.0	495.0	5.7376	218600.0
9814	-121.93	36.62	34.0	2351.0	NaN	1063.0	428.0	3.7250	278000.0

In [22]:
```
housing["median_income"].hist()
```

Out[22]: <Axes: >



In [23]:
```
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])
```

In [24]:

```
housing["income_cat"].value_counts()
```

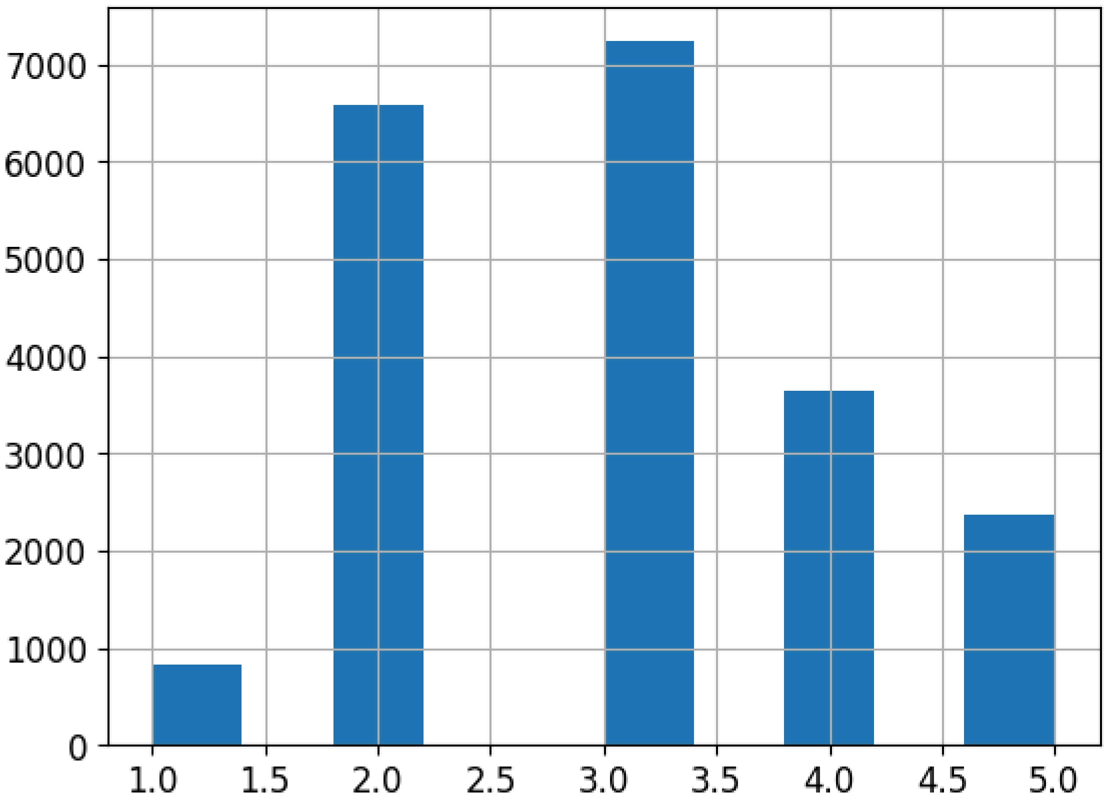
Out[24]:

	count
income_cat	
3	7236
2	6581
4	3639
5	2362
1	822

dtype: int64

In [25]:
```python
housing["income_cat"].hist()
```

Out[25]: <Axes: >



In [26]:
```python
from sklearn.model_selection import StratifiedShuffleSplit

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]
```

In [27]:
```
strat_test_set["income_cat"].value_counts() / len(strat_test_set)
```

Out[27]:

	count
income_cat	
3	0.350533
2	0.318798
4	0.176357
5	0.114341
1	0.039971

dtype: float64

In [28]:
```
housing["income_cat"].value_counts() / len(housing)
```

Out[28]:

	count
income_cat	
3	0.350581
2	0.318847
4	0.176308
5	0.114438
1	0.039826

dtype: float64

In [29]:
```
def income_cat_proportions(data):
    return data["income_cat"].value_counts() / len(data)

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

compare_props = pd.DataFrame({
    "Overall": income_cat_proportions(housing),
    "Stratified": income_cat_proportions(strat_test_set),
    "Random": income_cat_proportions(test_set),
}).sort_index()
compare_props["Rand. %error"] = 100 * compare_props["Random"] / compare_props["Overall"] - 100
compare_props["Strat. %error"] = 100 * compare_props["Stratified"] / compare_props["Overall"] - 100
```

In [30]:
```
compare_props
```

Out[30]:

	Overall	Stratified	Random	Rand. %error	Strat. %error
income_cat					
1	0.039826	0.039971	0.040213	0.973236	0.364964
2	0.318847	0.318798	0.324370	1.732260	-0.015195
3	0.350581	0.350533	0.358527	2.266446	-0.013820
4	0.176308	0.176357	0.167393	-5.056334	0.027480
5	0.114438	0.114341	0.109496	-4.318374	-0.084674

In [31]:
```
for set_ in (strat_train_set, strat_test_set):
    set_.drop("income_cat", axis=1, inplace=True)
```

Discover and Visualize the Data to Gain Insights

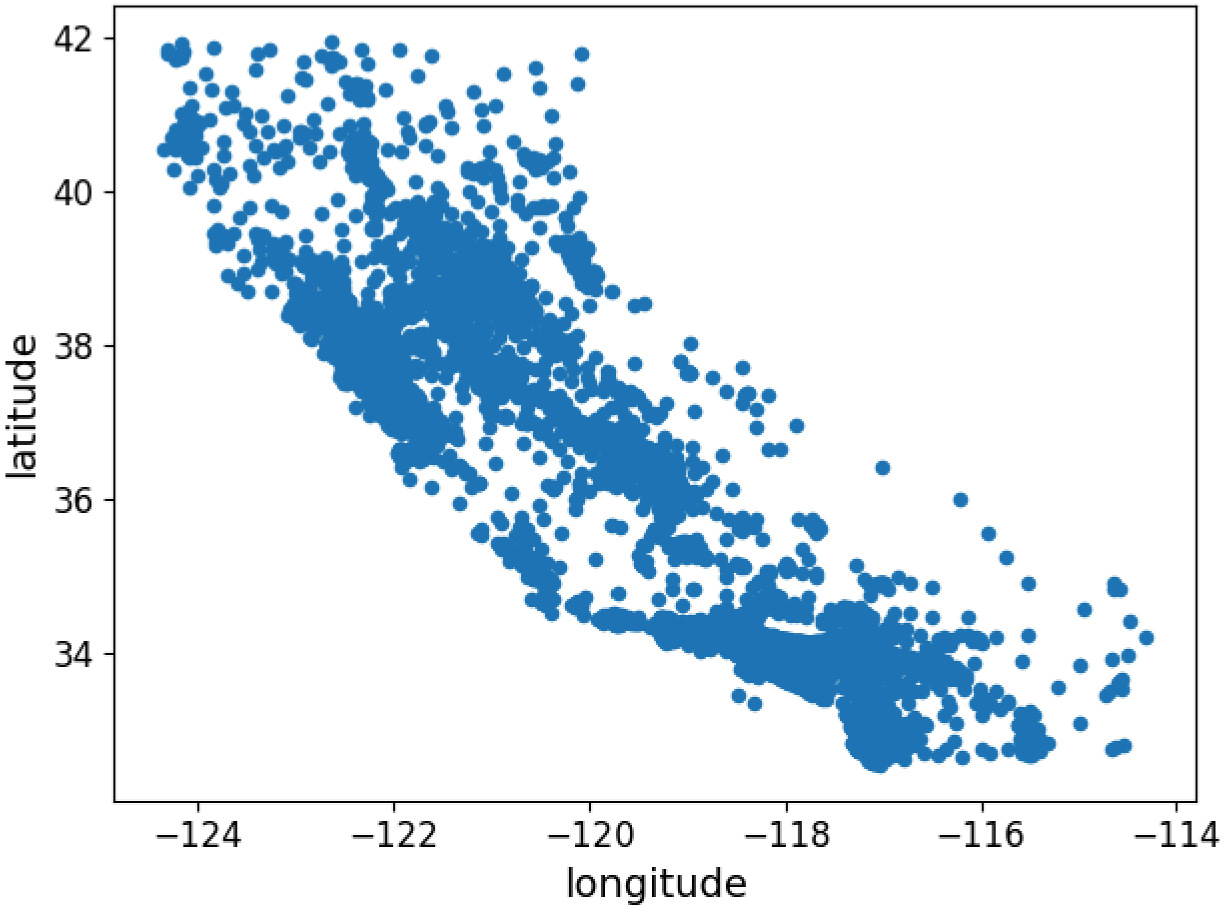
In [32]:

```
housing = strat_train_set.copy()
```

Visualizing Geographical Data

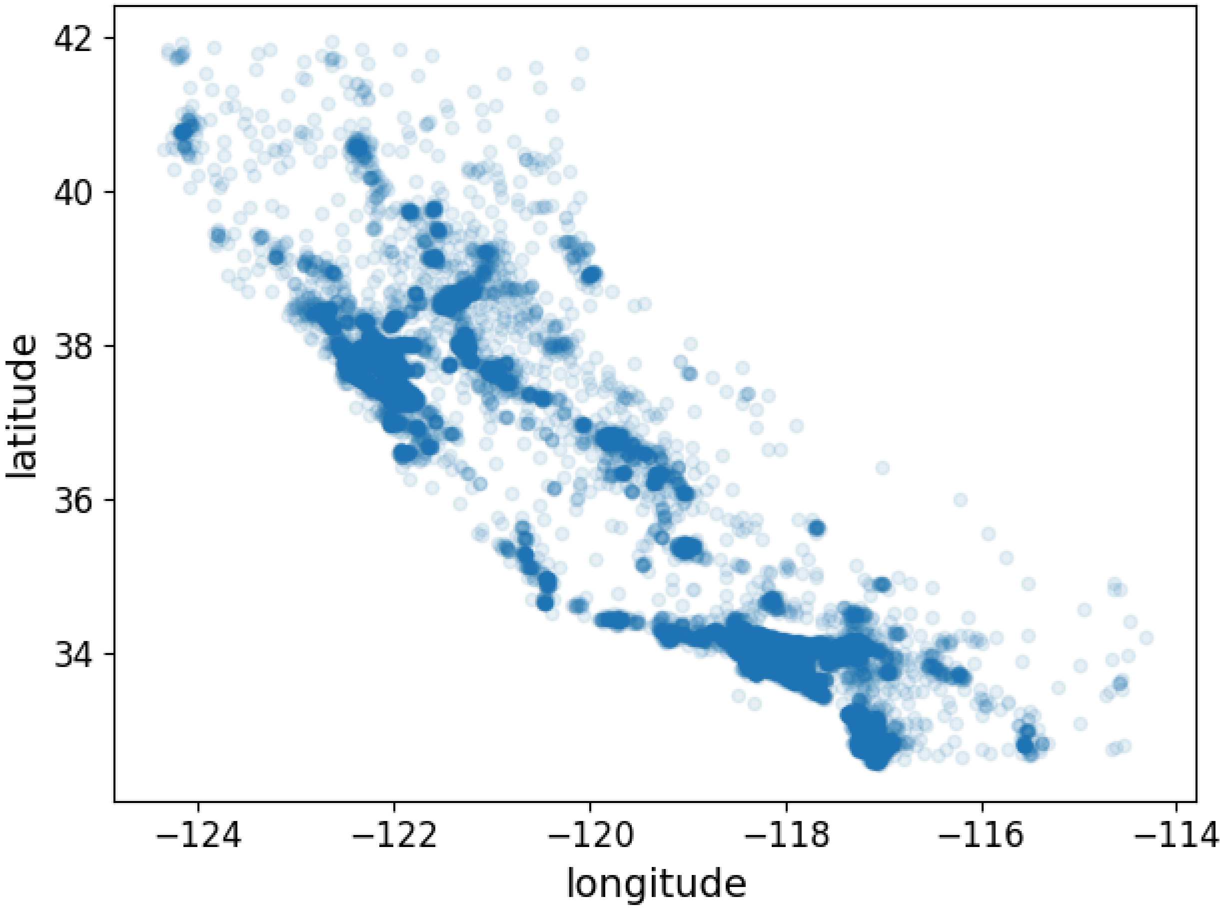
In [33]:
```
housing.plot(kind="scatter", x="longitude", y="latitude")
save_fig("bad_visualization_plot")
```

```
Saving figure bad_visualization_plot
```



In [34]:
```
housing.plot(kind="scatter", x="longitude", y="latitude", alpha=0.1)
save_fig("better_visualization_plot")
```

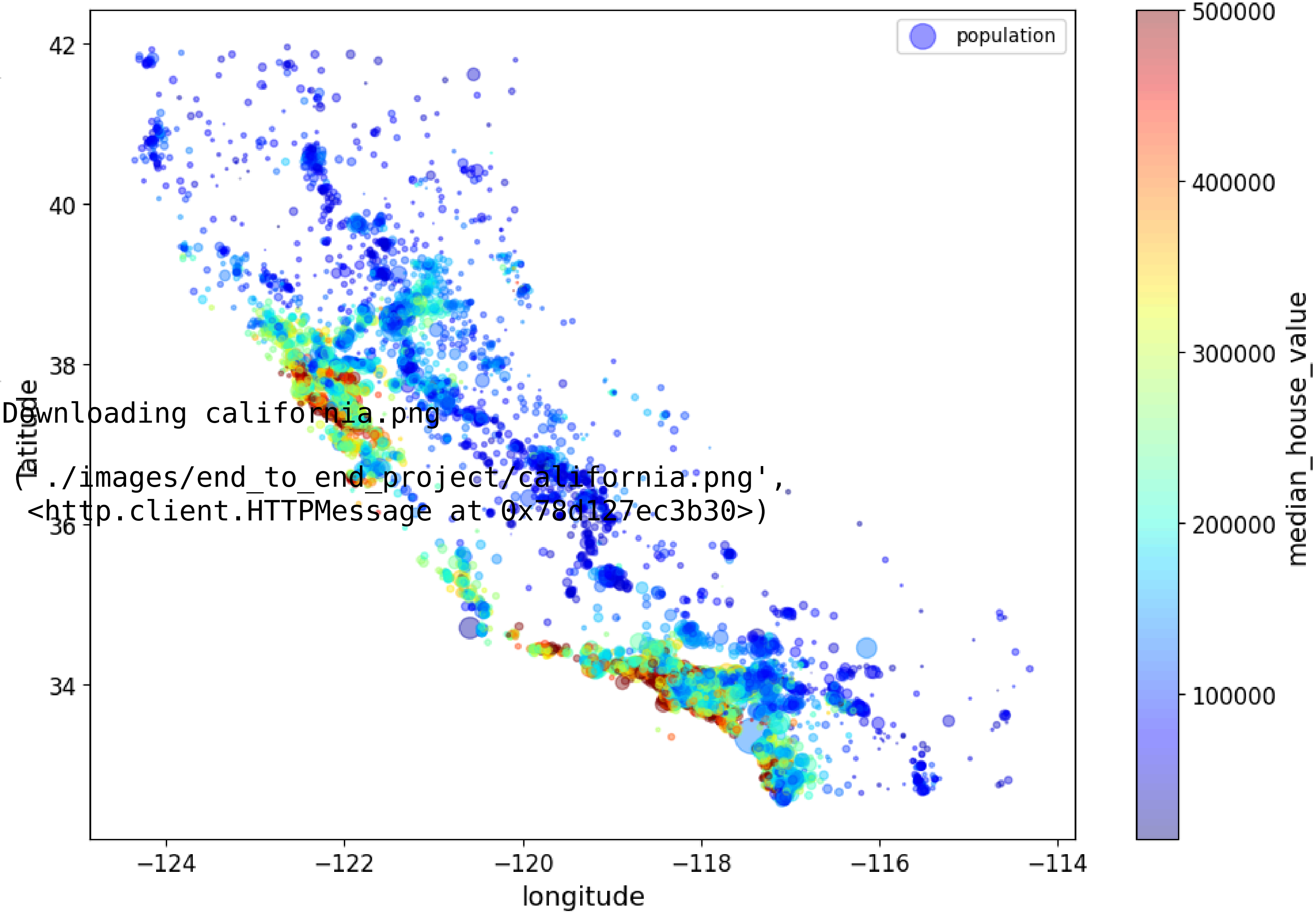
```
Saving figure better_visualization_plot
```



The argument `sharex=False` fixes a display bug (the x-axis values and legend were not displayed). This is a temporary fix (see: https://github.com/pandas-dev/pandas/issues/10611 (https://github.com/pandas-dev/pandas/issues/10611)). Thanks to Wilmer Arellano for pointing it out.

In [35]:
```
housing.plot(kind="scatter", x="longitude", y="latitude", alpha=0.4,
             s=housing["population"]/100, label="population", figsize=(10,7),
             c="median_house_value", cmap=plt.get_cmap("jet"), colorbar=True,
             sharex=False)
plt.legend()
save_fig("housing_prices_scatterplot")
```

```
Saving figure housing_prices_scatterplot
```



In [36]:

```
Downloading california.png
```

Out[36]:

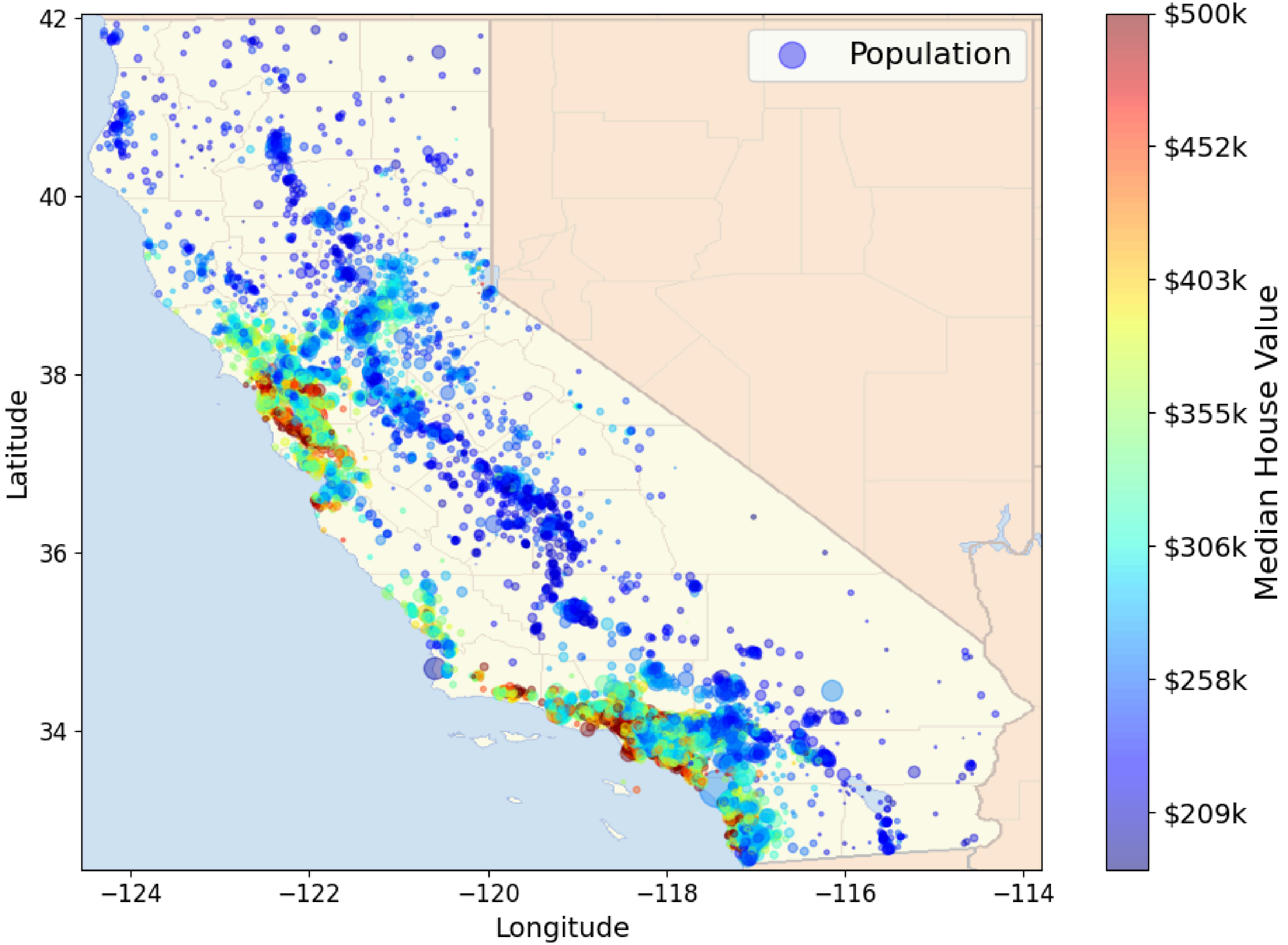
```
('./images/end_to_end_project/california.png',
 <http.client.HTTPMessage at 0x78d127ec3b30>)
```

```
In [37]: import matplotlib.image as mpimg
california_img=mpimg.imread(os.path.join(images_path, filename))
ax = housing.plot(kind="scatter", x="longitude", y="latitude", figsize=(10,7),
                  s=housing['population']/100, label="Population",
                  c="median_house_value", cmap=plt.get_cmap("jet"),
                  colorbar=False, alpha=0.4)
plt.imshow(california_img, extent=[-124.55, -113.80, 32.45, 42.05], alpha=0.5,
           cmap=plt.get_cmap("jet"))
plt.ylabel("Latitude", fontsize=14)
plt.xlabel("Longitude", fontsize=14)

prices = housing["median_house_value"]
tick_values = np.linspace(prices.min(), prices.max(), 11)
cbar = plt.colorbar(ticks=tick_values/prices.max())
cbar.ax.set_yticklabels(["$%dk"%(round(v/1000)) for v in tick_values], fontsize
=14)
cbar.set_label('Median House Value', fontsize=16)

plt.legend(fontsize=16)
save_fig("california_housing_prices_plot")
plt.show()
```

```
Saving figure california_housing_prices_plot
```



Population
Latitude
Longitude
Median House Value
42
40
38
36
34
−124
−122
−120
−118
−116
−114
$500k
$452k
$403k
$355k
$306k
$258k
$209k

In [40]:

In [41]:
```
corr_matrix["median_house_value"].sort_values(ascending=False)
```

Out[41]:

	median_house_value
median_house_value	1.000000
median_income	0.687151
total_rooms	0.135140
housing_median_age	0.114146
households	0.064590
total_bedrooms	0.047781
population	-0.026882
longitude	-0.047466
latitude	-0.142673

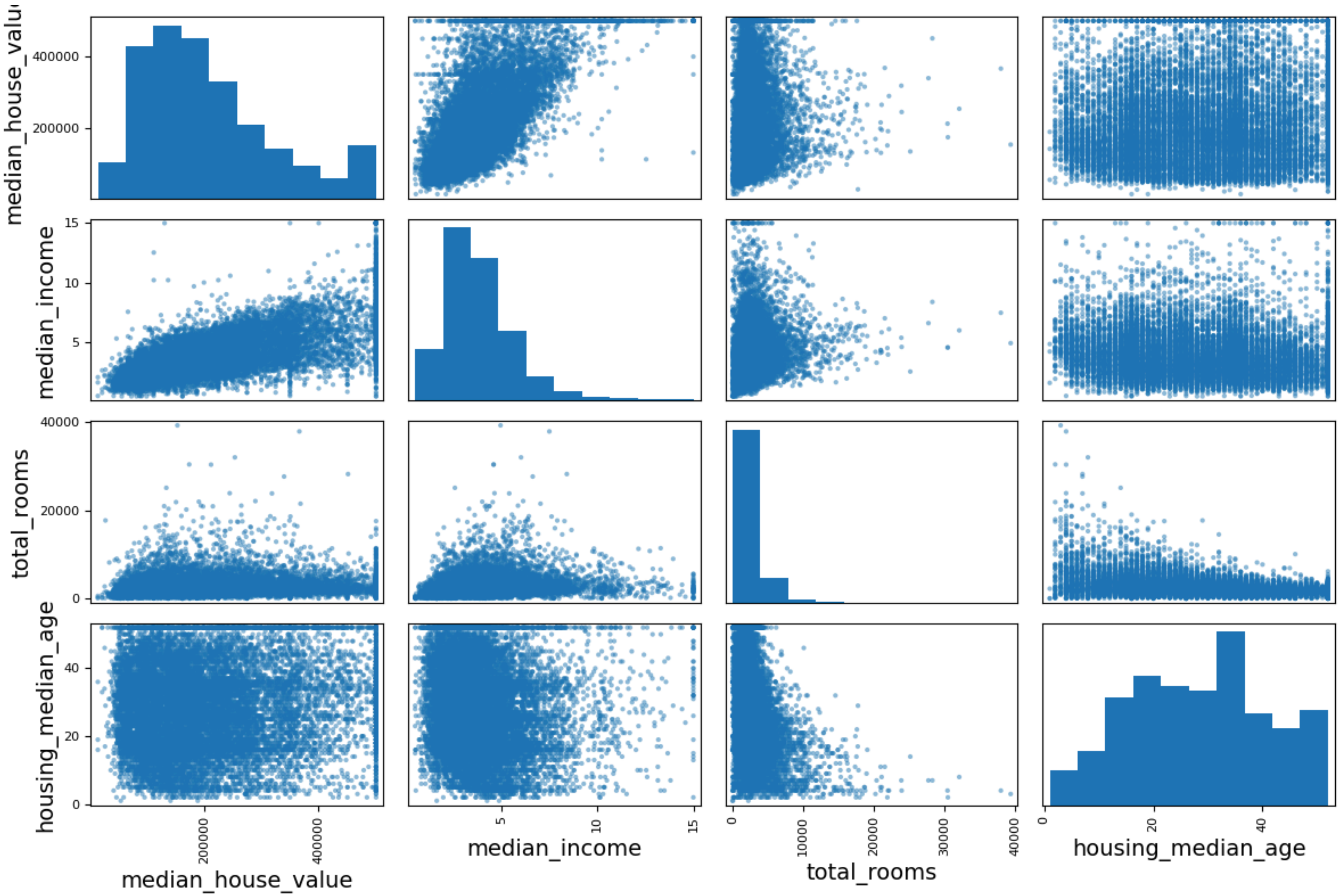
dtype: float64

In [42]:

```
# from pandas.tools.plotting import scatter_matrix # For older versions of Panda
s
from pandas.plotting import scatter_matrix

attributes = ["median_house_value", "median_income", "total_rooms",
              "housing_median_age"]
scatter_matrix(housing[attributes], figsize=(12, 8))
save_fig("scatter_matrix_plot")
```

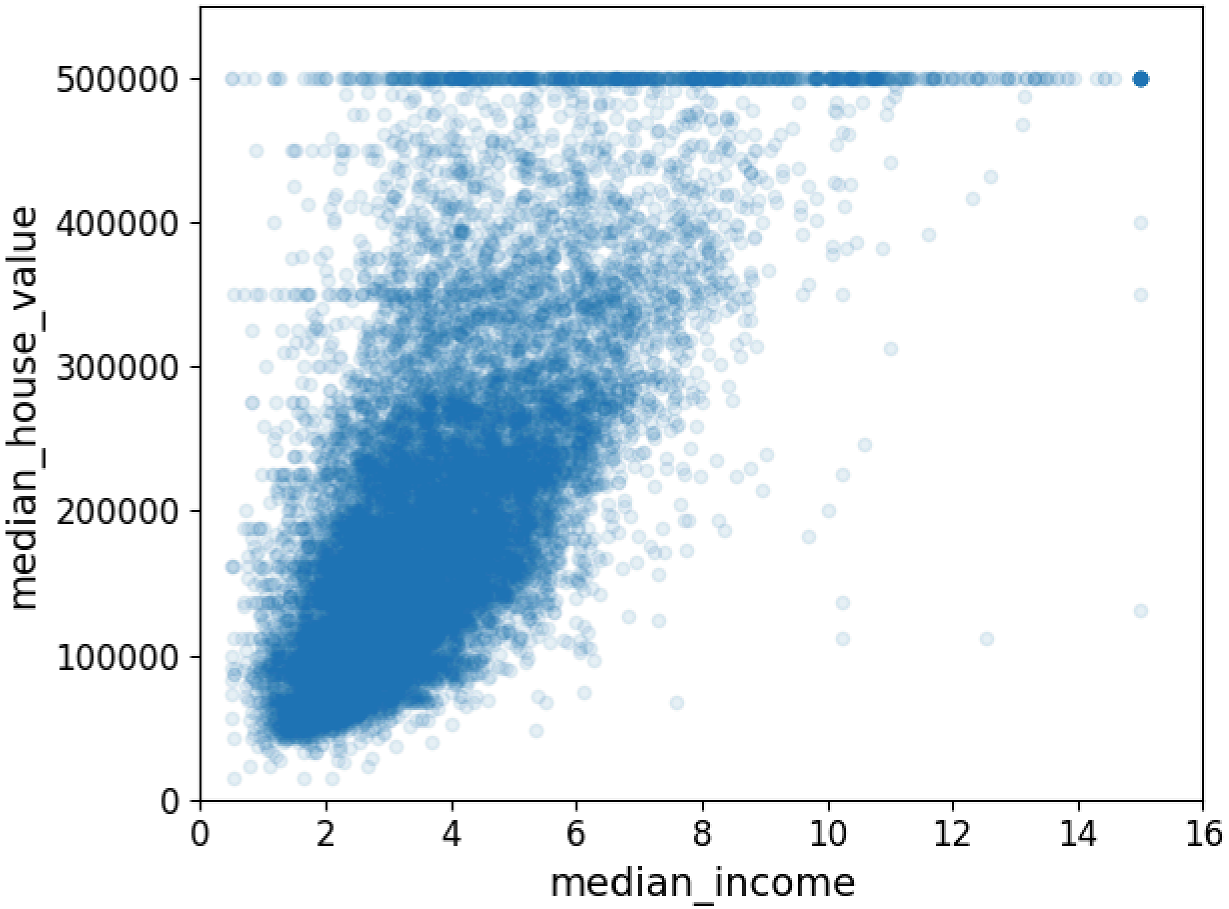
```
Saving figure scatter_matrix_plot
```



median_house_valu
median_income
total_rooms
housing_median_age
400000
200000
15
10
5
40000
20000
0
40
20
0
200000
400000
5
10
15
0
10000
20000
30000
40000
0
20
40
median_house_value
median_income
total_rooms
housing_median_age

In [43]:
```
housing.plot(kind="scatter", x="median_income", y="median_house_value",
             alpha=0.1)
plt.axis([0, 16, 0, 550000])
save_fig("income_vs_house_value_scatterplot")
```

```
Saving figure income_vs_house_value_scatterplot
```



Experimenting with Attribute Combinations

Normalization transformation

- added by Ken Perry

Convert attributes

- that are **total** within a geographic area
- to units that *normalize* for "size" of the area. Some size proxies
    - `households`
    - `total_rooms`

In [44]:
```
housing["rooms_per_household"] = housing["total_rooms"]/housing["households"]
housing["bedrooms_per_room"] = housing["total_bedrooms"]/housing["total_rooms"]
housing["population_per_household"]=housing["population"]/housing["households"]
```

In [47]:

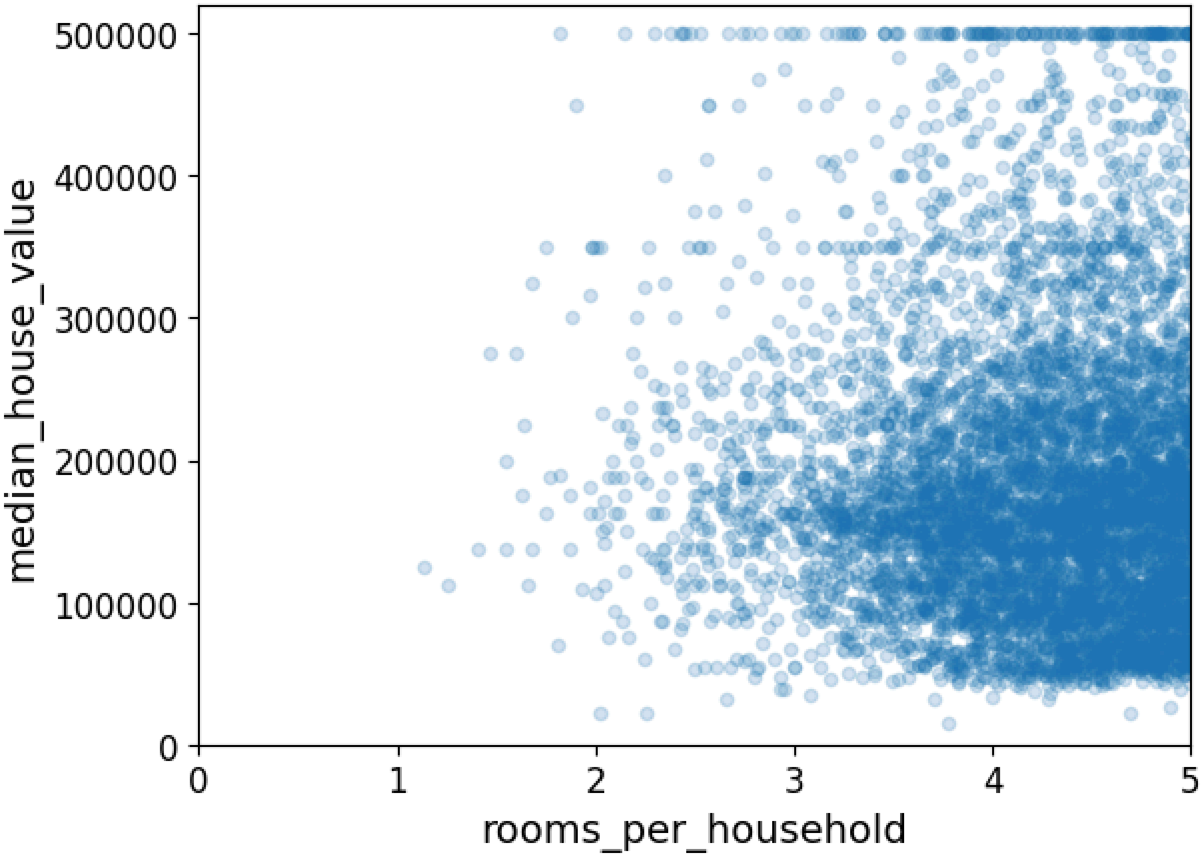
```
corr_matrix = housing.drop("ocean_proximity", axis=1).corr()
corr_matrix["median_house_value"].sort_values(ascending=False)
```

Out[47]:

	median_house_value
median_house_value	1.000000
median_income	0.687151
rooms_per_household	0.146255
total_rooms	0.135140
housing_median_age	0.114146
households	0.064590
total_bedrooms	0.047781
population_per_household	-0.021991
population	-0.026882
longitude	-0.047466
latitude	-0.142673
bedrooms_per_room	-0.259952

dtype: float64

In [48]:
```
housing.plot(kind="scatter", x="rooms_per_household", y="median_house_value",
             alpha=0.2)
plt.axis([0, 5, 0, 520000])
plt.show()
```



In [49]:

```
housing.describe()
```

Out[49]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_incor
count	16512.000000	16512.000000	16512.000000	16512.000000	16354.000000	16512.000000	16512.000000	16512.00000
mean	-119.575635	35.639314	28.653404	2622.539789	534.914639	1419.687379	497.011810	3.875884
std	2.001828	2.137963	12.574819	2138.417080	412.665649	1115.663036	375.696156	1.904931
min	-124.350000	32.540000	1.000000	6.000000	2.000000	3.000000	2.000000	0.499900
25%	-121.800000	33.940000	18.000000	1443.000000	295.000000	784.000000	279.000000	2.566950
50%	-118.510000	34.260000	29.000000	2119.000000	433.000000	1164.000000	408.000000	3.541550
75%	-118.010000	37.720000	37.000000	3141.000000	644.000000	1719.000000	602.000000	4.745325
max	-114.310000	41.950000	52.000000	39320.000000	6210.000000	35682.000000	5358.000000	15.000100

Prepare the Data for Machine Learning Algorithms

In [50]:

```
housing = strat_train_set.drop("median_house_value", axis=1) # drop labels for t
raining set
housing_labels = strat_train_set["median_house_value"].copy()
```

Data Cleaning

In the book 3 options are listed:

```
housing.dropna(subset=["total_bedrooms"])    # option 1
housing.drop("total_bedrooms", axis=1)       # option 2
median = housing["total_bedrooms"].median()  # option 3
housing["total_bedrooms"].fillna(median, inplace=True)
```

To demonstrate each of them, let's create a copy of the housing dataset, but keeping only the rows that contain at least one null. Then it will be easier to visualize exactly what each option does:

In [51]:
```
sample_incomplete_rows = housing[housing.isnull().any(axis=1)].head()
sample_incomplete_rows
```

Out[51]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	ocean_proximity
1606	-122.08	37.88	26.0	2947.0	NaN	825.0	626.0	2.9330	NEAR BAY
10915	-117.87	33.73	45.0	2264.0	NaN	1970.0	499.0	3.4193	<1H OCEAN
19150	-122.70	38.35	14.0	2313.0	NaN	954.0	397.0	3.7813	<1H OCEAN
4186	-118.23	34.13	48.0	1308.0	NaN	835.0	294.0	4.2891	<1H OCEAN
16885	-122.40	37.58	26.0	3281.0	NaN	1145.0	480.0	6.3580	NEAR OCEAN

In [52]:
```
sample_incomplete_rows.dropna(subset=["total_bedrooms"])    # option 1
```

Out[52]:

longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	ocean_proximity

In [53]:
```
sample_incomplete_rows.drop("total_bedrooms", axis=1)       # option 2
```

Out[53]:

	longitude	latitude	housing_median_age	total_rooms	population	households	median_income	ocean_proximity
1606	-122.08	37.88	26.0	2947.0	825.0	626.0	2.9330	NEAR BAY
10915	-117.87	33.73	45.0	2264.0	1970.0	499.0	3.4193	<1H OCEAN
19150	-122.70	38.35	14.0	2313.0	954.0	397.0	3.7813	<1H OCEAN
4186	-118.23	34.13	48.0	1308.0	835.0	294.0	4.2891	<1H OCEAN
16885	-122.40	37.58	26.0	3281.0	1145.0	480.0	6.3580	NEAR OCEAN

In [54]:
```python
median = housing["total_bedrooms"].median()
sample_incomplete_rows["total_bedrooms"].fillna(median, inplace=True) # option 3
```

```
/tmp/ipython-input-760120979.py:2: FutureWarning: A value is trying to be set
on a copy of a DataFrame or Series through chained assignment using an inplace
method.
The behavior will change in pandas 3.0. This inplace method will never work be
cause the intermediate object on which we are setting values always behaves as
a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.m
ethod({col: value}, inplace=True)' or df[col] = df[col].method(value) instead,
to perform the operation inplace on the original object.


  sample_incomplete_rows["total_bedrooms"].fillna(median, inplace=True) # opti
on 3
```

In [55]:
```python
sample_incomplete_rows
```

Out[55]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	ocean_proximity
1606	-122.08	37.88	26.0	2947.0	433.0	825.0	626.0	2.9330	NEAR BAY
10915	-117.87	33.73	45.0	2264.0	433.0	1970.0	499.0	3.4193	<1H OCEAN
19150	-122.70	38.35	14.0	2313.0	433.0	954.0	397.0	3.7813	<1H OCEAN
4186	-118.23	34.13	48.0	1308.0	433.0	835.0	294.0	4.2891	<1H OCEAN
16885	-122.40	37.58	26.0	3281.0	433.0	1145.0	480.0	6.3580	NEAR OCEAN

In [56]:
```python
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(strategy="median")
```

Remove the text attribute because median can only be calculated on numerical attributes:

In [57]:
```
housing_num = housing.drop("ocean_proximity", axis=1)
# alternatively: housing_num = housing.select_dtypes(include=[np.number])
```

In [58]:
```
imputer.fit(housing_num)
```

Out[58]:
```
▾ SimpleImputer
SimpleImputer(strategy='median')
```
(https://scikit-learn.org/1.6/modules/generated/sklearn.impute.

In [59]:
```
imputer.statistics_
```

Out[59]:
```
array([-118.51    ,   34.26    ,   29.     , 2119.      ,  433.      ,
        1164.      ,  408.      ,    3.54155])
```

Check that this is the same as manually computing the median of each attribute:

In [60]:
```
housing_num.median().values
```

Out[60]:
```
array([-118.51    ,   34.26    ,   29.     , 2119.      ,  433.      ,
        1164.      ,  408.      ,    3.54155])
```

Transform the training set:

In [61]:
```
X = imputer.transform(housing_num)
```

In [62]:
```
housing_tr = pd.DataFrame(X, columns=housing_num.columns,
                          index=housing.index)
```

In [63]:
```
housing_tr.loc[sample_incomplete_rows.index.values]
```

Out[63]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income
1606	-122.08	37.88	26.0	2947.0	433.0	825.0	626.0	2.9330
10915	-117.87	33.73	45.0	2264.0	433.0	1970.0	499.0	3.4193
19150	-122.70	38.35	14.0	2313.0	433.0	954.0	397.0	3.7813
4186	-118.23	34.13	48.0	1308.0	433.0	835.0	294.0	4.2891
16885	-122.40	37.58	26.0	3281.0	433.0	1145.0	480.0	6.3580

In [64]:
```
imputer.strategy
```

Out[64]:
```
'median'
```

In [65]:
```
housing_tr = pd.DataFrame(X, columns=housing_num.columns,
                          index=housing_num.index)
```

In [66]:
```
housing_tr.head()
```

Out[66]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income
12655	-121.46	38.52	29.0	3873.0	797.0	2237.0	706.0	2.1736
15502	-117.23	33.09	7.0	5320.0	855.0	2015.0	768.0	6.3373
2908	-119.04	35.37	44.0	1618.0	310.0	667.0	300.0	2.8750
14053	-117.13	32.75	24.0	1877.0	519.0	898.0	483.0	2.2264
20496	-118.70	34.28	27.0	3536.0	646.0	1837.0	580.0	4.4964

Handling Text and Categorical Attributes

Now let's preprocess the categorical input feature, `ocean_proximity`:

In [67]:
```
housing_cat = housing[["ocean_proximity"]]
housing_cat.head(10)
```

Out[67]:

	ocean_proximity
12655	INLAND
15502	NEAR OCEAN
2908	INLAND
14053	NEAR OCEAN
20496	<1H OCEAN
1481	NEAR BAY
18125	<1H OCEAN
5830	<1H OCEAN
17989	<1H OCEAN
4861	<1H OCEAN

In [68]:
```
from sklearn.preprocessing import OrdinalEncoder

ordinal_encoder = OrdinalEncoder()
housing_cat_encoded = ordinal_encoder.fit_transform(housing_cat)
housing_cat_encoded[:10]
```

Out[68]:
```
array([[1.],
       [4.],
       [1.],
       [4.],
       [0.],
       [3.],
       [0.],
       [0.],
       [0.],
       [0.]])
```

In [69]:
```
ordinal_encoder.categories_
```

Out[69]:
```
[array(['<1H OCEAN', 'INLAND', 'ISLAND', 'NEAR BAY', 'NEAR OCEAN'],
       dtype=object)]
```

In [70]:
```
from sklearn.preprocessing import OneHotEncoder

cat_encoder = OneHotEncoder()
housing_cat_1hot = cat_encoder.fit_transform(housing_cat)
housing_cat_1hot
```

Out[70]:
```
<Compressed Sparse Row sparse matrix of dtype 'float64'
	with 16512 stored elements and shape (16512, 5)>
```

By default, the `OneHotEncoder` class returns a sparse array, but we can convert it to a dense array if needed by calling the `toarray()` method:

In [71]:
```
housing_cat_1hot.toarray()
```

Out[71]:
```
array([[0., 1., 0., 0., 0.],
       [0., 0., 0., 0., 1.],
       [0., 1., 0., 0., 0.],
       ...,
       [1., 0., 0., 0., 0.],
       [1., 0., 0., 0., 0.],
       [0., 1., 0., 0., 0.]])
```

Alternatively, you can set `sparse=False` when creating the `OneHotEncoder`:

In [74]:
```
cat_encoder = OneHotEncoder(sparse_output=False)
housing_cat_1hot = cat_encoder.fit_transform(housing_cat)
housing_cat_1hot
```

Out[74]:
```
array([[0., 1., 0., 0., 0.],
       [0., 0., 0., 0., 1.],
       [0., 1., 0., 0., 0.],
       ...,
       [1., 0., 0., 0., 0.],
       [1., 0., 0., 0., 0.],
       [0., 1., 0., 0., 0.]])
```

In [75]:
```
cat_encoder.categories_
```

Out[75]:
```
[array(['<1H OCEAN', 'INLAND', 'ISLAND', 'NEAR BAY', 'NEAR OCEAN'],
       dtype=object)]
```

Custom Transformers

Let's create a custom transformer to add extra attributes:

In [76]:
```python
from sklearn.base import BaseEstimator, TransformerMixin

# column index
rooms_ix, bedrooms_ix, population_ix, households_ix = 3, 4, 5, 6

class CombinedAttributesAdder(BaseEstimator, TransformerMixin):
    def __init__(self, add_bedrooms_per_room=True): # no *args or **kargs
        self.add_bedrooms_per_room = add_bedrooms_per_room
    def fit(self, X, y=None):
        return self  # nothing else to do
    def transform(self, X):
        rooms_per_household = X[:, rooms_ix] / X[:, households_ix]
        population_per_household = X[:, population_ix] / X[:, households_ix]
        if self.add_bedrooms_per_room:
            bedrooms_per_room = X[:, bedrooms_ix] / X[:, rooms_ix]
            return np.c_[X, rooms_per_household, population_per_household,
                         bedrooms_per_room]
        else:
            return np.c_[X, rooms_per_household, population_per_household]

attr_adder = CombinedAttributesAdder(add_bedrooms_per_room=False)
housing_extra_attribs = attr_adder.transform(housing.values)
```

Note that I hard coded the indices (3, 4, 5, 6) for concision and clarity in the book, but it would be much cleaner to get them dynamically, like this:

In [77]:
```python
col_names = "total_rooms", "total_bedrooms", "population", "households"
rooms_ix, bedrooms_ix, population_ix, households_ix = [
    housing.columns.get_loc(c) for c in col_names] # get the column indices
```

Also, `housing_extra_attribs` is a NumPy array, we've lost the column names (unfortunately, that's a problem with Scikit-Learn). To recover a `DataFrame`, you could run this:

In [78]:
```
housing_extra_attribs = pd.DataFrame(
    housing_extra_attribs,
    columns=list(housing.columns)+["rooms_per_household", "population_per_househ
old"],
    index=housing.index)
housing_extra_attribs.head()
```

Out[78]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	ocean_proximity
12655	-121.46	38.52	29.0	3873.0	797.0	2237.0	706.0	2.1736	INLAND
15502	-117.23	33.09	7.0	5320.0	855.0	2015.0	768.0	6.3373	NEAR OCEAN
2908	-119.04	35.37	44.0	1618.0	310.0	667.0	300.0	2.875	INLAND
14053	-117.13	32.75	24.0	1877.0	519.0	898.0	483.0	2.2264	NEAR OCEAN
20496	-118.7	34.28	27.0	3536.0	646.0	1837.0	580.0	4.4964	<1H OCEAN

Transformation Pipelines

Now let's build a pipeline for preprocessing the numerical attributes:

In [79]:
```python
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler

num_pipeline = Pipeline([
        ('imputer', SimpleImputer(strategy="median")),
        ('attribs_adder', CombinedAttributesAdder()),
        ('std_scaler', StandardScaler()),
    ])

housing_num_tr = num_pipeline.fit_transform(housing_num)
```

In [80]:
```python
housing_num_tr
```

Out[80]:
```
array([[-0.94135046,  1.34743822,  0.02756357, ...,  0.01739526,
         0.00622264, -0.12112176],
       [ 1.17178212, -1.19243966, -1.72201763, ...,  0.56925554,
        -0.04081077, -0.81086696],
       [ 0.26758118, -0.1259716 ,  1.22045984, ..., -0.01802432,
        -0.07537122, -0.33827252],
       ...,
       [-1.5707942 ,  1.31001828,  1.53856552, ..., -0.5092404 ,
        -0.03743619,  0.32286937],
       [-1.56080303,  1.2492109 , -1.1653327 , ...,  0.32814891,
        -0.05915604, -0.45702273],
       [-1.28105026,  2.02567448, -0.13148926, ...,  0.01407228,
         0.00657083, -0.12169672]])
```

In [81]:
```python
from sklearn.compose import ColumnTransformer

num_attribs = list(housing_num)
cat_attribs = ["ocean_proximity"]

full_pipeline = ColumnTransformer([
        ("num", num_pipeline, num_attribs),
        ("cat", OneHotEncoder(), cat_attribs),
    ])

housing_prepared = full_pipeline.fit_transform(housing)
```

In [82]:
```python
housing_prepared
```

Out[82]:
```
array([[-0.94135046,  1.34743822,  0.02756357, ...,  0.        ,
         0.        ,  0.        ],
       [ 1.17178212, -1.19243966, -1.72201763, ...,  0.        ,
         0.        ,  1.        ],
       [ 0.26758118, -0.1259716 ,  1.22045984, ...,  0.        ,
         0.        ,  0.        ],
       ...,
       [-1.5707942 ,  1.31001828,  1.53856552, ...,  0.        ,
         0.        ,  0.        ],
       [-1.56080303,  1.2492109 , -1.1653327 , ...,  0.        ,
         0.        ,  0.        ],
       [-1.28105026,  2.02567448, -0.13148926, ...,  0.        ,
         0.        ,  0.        ]])
```

In [83]:
```python
housing_prepared.shape
```

Out[83]:
```
(16512, 16)
```

For reference, here is the old solution based on a `DataFrameSelector` transformer (to just select a subset of the Pandas `DataFrame` columns), and a `FeatureUnion`:

In [84]:

```
from sklearn.base import BaseEstimator, TransformerMixin

# Create a class to select numerical or categorical columns
class OldDataFrameSelector(BaseEstimator, TransformerMixin):
    def __init__(self, attribute_names):
        self.attribute_names = attribute_names
    def fit(self, X, y=None):
        return self
    def transform(self, X):
        return X[self.attribute_names].values
```

Now let's join all these components into a big pipeline that will preprocess both the numerical and the categorical features:

In [87]:

```
num_attribs = list(housing_num)
cat_attribs = ["ocean_proximity"]

old_num_pipeline = Pipeline([
        ('selector', OldDataFrameSelector(num_attribs)),
        ('imputer', SimpleImputer(strategy="median")),
        ('attribs_adder', CombinedAttributesAdder()),
        ('std_scaler', StandardScaler()),
    ])

old_cat_pipeline = Pipeline([
        ('selector', OldDataFrameSelector(cat_attribs)),
        ('cat_encoder', OneHotEncoder(sparse_output=False)),
    ])
```

In [88]:

```
from sklearn.pipeline import FeatureUnion

old_full_pipeline = FeatureUnion(transformer_list=[
        ("num_pipeline", old_num_pipeline),
        ("cat_pipeline", old_cat_pipeline),
    ])
```

In [89]:
```
old_housing_prepared = old_full_pipeline.fit_transform(housing)
old_housing_prepared
```

Out[89]:
```
array([[-0.94135046,  1.34743822,  0.02756357, ...,  0.        ,
         0.        ,  0.        ],
       [ 1.17178212, -1.19243966, -1.72201763, ...,  0.        ,
         0.        ,  1.        ],
       [ 0.26758118, -0.1259716 ,  1.22045984, ...,  0.        ,
         0.        ,  0.        ],
       ...,
       [-1.5707942 ,  1.31001828,  1.53856552, ...,  0.        ,
         0.        ,  0.        ],
       [-1.56080303,  1.2492109 , -1.1653327 , ...,  0.        ,
         0.        ,  0.        ],
       [-1.28105026,  2.02567448, -0.13148926, ...,  0.        ,
         0.        ,  0.        ]])
```

The result is the same as with the `ColumnTransformer`:

In [90]:
```
np.allclose(housing_prepared, old_housing_prepared)
```

Out[90]:
```
True
```

Select and Train a Model

Training and Evaluating on the Training Set

In [91]:
```python
from sklearn.linear_model import LinearRegression

lin_reg = LinearRegression()
lin_reg.fit(housing_prepared, housing_labels)
```

Out[91]:
```
▾ LinearRegression (https://scikit-learn.org/1.6/modules/generated/sklearn.linear_mo

LinearRegression()
```

In [92]:
```python
# let's try the full preprocessing pipeline on a few training instances
some_data = housing.iloc[:5]
some_labels = housing_labels.iloc[:5]
some_data_prepared = full_pipeline.transform(some_data)

print("Predictions:", lin_reg.predict(some_data_prepared))
```

```
Predictions: [ 85657.90192014 305492.60737488 152056.46122456 186095.70946094
 244550.67966089]
```

Compare against the actual values:

In [93]:
```python
print("Labels:", list(some_labels))
```

```
Labels: [72100.0, 279600.0, 82700.0, 112500.0, 238300.0]
```

In [94]:
```python
some_data_prepared
```

Out[94]:
```
array([[-0.94135046,  1.34743822,  0.02756357,  0.58477745,  0.64037127,
         0.73260236,  0.55628602, -0.8936472 ,  0.01739526,  0.00622264,
        -0.12112176,  0.        ,  1.        ,  0.        ,  0.        ,
         0.        ],
       [ 1.17178212, -1.19243966, -1.72201763,  1.26146668,  0.78156132,
         0.53361152,  0.72131799,  1.292168  ,  0.56925554, -0.04081077,
        -0.81086696,  0.        ,  0.        ,  0.        ,  0.        ,
         1.        ],
       [ 0.26758118, -0.1259716 ,  1.22045984, -0.46977281, -0.54513828,
        -0.67467519, -0.52440722, -0.52543365, -0.01802432, -0.07537122,
        -0.33827252,  0.        ,  1.        ,  0.        ,  0.        ,
         0.        ],
       [ 1.22173797, -1.35147437, -0.37006852, -0.34865152, -0.03636724,
        -0.46761716, -0.03729672, -0.86592882, -0.59513997, -0.10680295,
         0.96120521,  0.        ,  0.        ,  0.        ,  0.        ,
         1.        ],
       [ 0.43743108, -0.63581817, -0.13148926,  0.42717947,  0.27279028,
         0.37406031,  0.22089846,  0.32575178,  0.2512412 ,  0.00610923,
        -0.47451338,  1.        ,  0.        ,  0.        ,  0.        ,
         0.        ]])
```

In [95]:
```python
from sklearn.metrics import mean_squared_error

housing_predictions = lin_reg.predict(housing_prepared)
lin_mse = mean_squared_error(housing_labels, housing_predictions)
lin_rmse = np.sqrt(lin_mse)
lin_rmse
```

Out[95]:
```
np.float64(68627.87390018745)
```

Note: since Scikit-Learn 0.22, you can get the RMSE directly by calling the `mean_squared_error()` function with `squared=False`.

In [96]:
```
from sklearn.metrics import mean_absolute_error

lin_mae = mean_absolute_error(housing_labels, housing_predictions)
lin_mae
```

Out[96]:
```
49438.66860915802
```

In [97]:
```
from sklearn.tree import DecisionTreeRegressor

tree_reg = DecisionTreeRegressor(random_state=42)
tree_reg.fit(housing_prepared, housing_labels)
```

Out[97]:

▾ DecisionTreeRegressor (https://scikit-learn.org/1.6/modules/generated/sklearn.tr

```
DecisionTreeRegressor(random_state=42)
```

In [98]:
```
housing_predictions = tree_reg.predict(housing_prepared)
tree_mse = mean_squared_error(housing_labels, housing_predictions)
tree_rmse = np.sqrt(tree_mse)
tree_rmse
```

Out[98]:
```
np.float64(0.0)
```

Better Evaluation Using Cross-Validation

In [99]:

```
from sklearn.model_selection import cross_val_score

scores = cross_val_score(tree_reg, housing_prepared, housing_labels,
                         scoring="neg_mean_squared_error", cv=10)
tree_rmse_scores = np.sqrt(-scores)
```

In [100]:

```
def display_scores(scores):
    print("Scores:", scores)
    print("Mean:", scores.mean())
    print("Standard deviation:", scores.std())

display_scores(tree_rmse_scores)
```

```
Scores: [72831.45749112 69973.18438322 69528.56551415 72517.78229792
 69145.50006909 79094.74123727 68960.045444   73344.50225684
 69826.02473916 71077.09753998]
Mean: 71629.89009727491
Standard deviation: 2914.035468468928
```

In [101]:

```
lin_scores = cross_val_score(lin_reg, housing_prepared, housing_labels,
                             scoring="neg_mean_squared_error", cv=10)
lin_rmse_scores = np.sqrt(-lin_scores)
display_scores(lin_rmse_scores)
```

```
Scores: [71762.76364394 64114.99166359 67771.17124356 68635.19072082
 66846.14089488 72528.03725385 73997.08050233 68802.33629334
 66443.28836884 70139.79923956]
Mean: 69104.07998247063
Standard deviation: 2880.3282098180694
```

Note: we specify `n_estimators=100` to be future-proof since the default value is going to change to 100 in Scikit-Learn 0.22 (for simplicity, this is not shown in the book).

In [102]:
```
from sklearn.ensemble import RandomForestRegressor

forest_reg = RandomForestRegressor(n_estimators=100, random_state=42)
forest_reg.fit(housing_prepared, housing_labels)
```

Out[102]:
```
▾ RandomForestRegressor
RandomForestRegressor(random_state=42)
```
(https://scikit-learn.org/1.6/modules/generated/sklearn.en

In [103]:
```
housing_predictions = forest_reg.predict(housing_prepared)
forest_mse = mean_squared_error(housing_labels, housing_predictions)
forest_rmse = np.sqrt(forest_mse)
forest_rmse
```

Out[103]:
```
np.float64(18650.698705770003)
```

In [104]:
```
from sklearn.model_selection import cross_val_score

forest_scores = cross_val_score(forest_reg, housing_prepared, housing_labels,
                                scoring="neg_mean_squared_error", cv=10)
forest_rmse_scores = np.sqrt(-forest_scores)
display_scores(forest_rmse_scores)
```

```
Scores: [51559.63379638 48737.57100062 47210.51269766 51875.21247297
 47577.50470123 51863.27467888 52746.34645573 50065.1762751
 48664.66818196 54055.90894609]
Mean: 50435.58092066179
Standard deviation: 2203.3381412764606
```

In [105]:
```
scores = cross_val_score(lin_reg, housing_prepared, housing_labels, scoring="neg
_mean_squared_error", cv=10)
pd.Series(np.sqrt(-scores)).describe()
```

Out[105]:

	0
count	10.000000
mean	69104.079982
std	3036.132517
min	64114.991664
25%	67077.398482
50%	68718.763507
75%	71357.022543
max	73997.080502

dtype: float64

In [106]:
```python
from sklearn.svm import SVR

svm_reg = SVR(kernel="linear")
svm_reg.fit(housing_prepared, housing_labels)
housing_predictions = svm_reg.predict(housing_prepared)
svm_mse = mean_squared_error(housing_labels, housing_predictions)
svm_rmse = np.sqrt(svm_mse)
svm_rmse
```

Out[106]:
```
np.float64(111095.06635291968)
```

Fine-Tune Your Model

Grid Search

In [107]:

```
from sklearn.model_selection import GridSearchCV

param_grid = [
    # try 12 (3×4) combinations of hyperparameters
    {'n_estimators': [3, 10, 30], 'max_features': [2, 4, 6, 8]},
    # then try 6 (2×3) combinations with bootstrap set as False
    {'bootstrap': [False], 'n_estimators': [3, 10], 'max_features': [2, 3, 4]},
  ]

forest_reg = RandomForestRegressor(random_state=42)
# train across 5 folds, that's a total of (12+6)*5=90 rounds of training
grid_search = GridSearchCV(forest_reg, param_grid, cv=5,
                           scoring='neg_mean_squared_error',
                           return_train_score=True)
grid_search.fit(housing_prepared, housing_labels)
```

Out[107]:

```
▸ GridSearchCV
(https://scikit-learn.org/1.6/modules/generate

▸ best_estimator_: RandomForestRegressor

  ▸ RandomForestRegressor
  (https://scikit-learn.org/1.6/modules/generated/sklea
```

The best hyperparameter combination found:

In [108]:
```
grid_search.best_params_
```

Out[108]:
```
{'max_features': 8, 'n_estimators': 30}
```

In [109]:
```
grid_search.best_estimator_
```

Out[109]:
```
RandomForestRegressor
RandomForestRegressor(max_features=8, n_estimators=30, random_state=42)
```

(https://scikit-learn.org/

Let's look at the score of each hyperparameter combination tested during the grid search:

In [110]:

```
cvres = grid_search.cv_results_
for mean_score, params in zip(cvres["mean_test_score"], cvres["params"]):
    print(np.sqrt(-mean_score), params)
```

```
63895.161577951665 {'max_features': 2, 'n_estimators': 3}
54916.32386349543 {'max_features': 2, 'n_estimators': 10}
52885.86715332332 {'max_features': 2, 'n_estimators': 30}
60075.3680329983 {'max_features': 4, 'n_estimators': 3}
52495.01284985185 {'max_features': 4, 'n_estimators': 10}
50187.24324926565 {'max_features': 4, 'n_estimators': 30}
58064.73529982314 {'max_features': 6, 'n_estimators': 3}
51519.32062366315 {'max_features': 6, 'n_estimators': 10}
49969.80441627874 {'max_features': 6, 'n_estimators': 30}
58895.824998155826 {'max_features': 8, 'n_estimators': 3}
52459.79624724529 {'max_features': 8, 'n_estimators': 10}
49898.98913455217 {'max_features': 8, 'n_estimators': 30}
62381.765106921855 {'bootstrap': False, 'max_features': 2, 'n_estimators': 3}
54476.57050944266 {'bootstrap': False, 'max_features': 2, 'n_estimators': 10}
59974.60028085155 {'bootstrap': False, 'max_features': 3, 'n_estimators': 3}
52754.5632813202 {'bootstrap': False, 'max_features': 3, 'n_estimators': 10}
57831.13606121427 4 {'bootstrap': False, 'max_features': 4, 'n_estimators': 3}
51278.3787714025 3 {'bootstrap': False, 'max_features': 4, 'n_estimators': 10}
```

In [111]:

```
pd.DataFrame(grid_search.cv_results_)
```

Out[111]:

	mean_fit_time	std_fit_time	mean_score_time	std_score_time	param_max_features	param_n_estimators	param_bootstrap	pa
0	0.110077	0.003531	0.005191	0.000222	2	3	NaN	{'max_featu 2, 'n_estimato 3}
1	0.363616	0.010778	0.013031	0.000205	2	10	NaN	{'max_featu 2, 'n_estimato 10}
2	1.220374	0.168491	0.037442	0.004462	2	30	NaN	{'max_featu 2, 'n_estimato 30}
3	0.187690	0.005442	0.005392	0.000243	4	3	NaN	{'max_featu 4, 'n_estimato 3}
4	0.624381	0.015184	0.013445	0.000942	4	10	NaN	{'max_featu 4, 'n_estimato 10}
5	3.012486	0.867238	0.050869	0.022758	4	30	NaN	{'max_featu 4, 'n_estimato 30}
6	0.423193	0.040724	0.009847	0.003883	6	3	NaN	{'max_featu 6, 'n_estimato 3}
7	1.120588	0.272176	0.016621	0.005206	6	10	NaN	{'max_featu 6, 'n_estimato 10}
8	2.786753	0.207748	0.042939	0.012513	6	30	NaN	{'max_featu 6, 'n_estimato 30}
9	0.481541	0.065177	0.007467	0.000711	8	3	NaN	{'max_featu 8, 'n_estimato 3}

	mean_fit_time	std_fit_time	mean_score_time	std_score_time	param_max_features	param_n_estimators	param_bootstrap	pa
10	1.411470	0.162194	0.015355	0.001987	8	10	NaN	{'max_featu 8, 'n_estimato 10}
11	3.861695	0.807494	0.037716	0.004085	8	30	NaN	{'max_featu 8, 'n_estimato 30}
12	0.163109	0.007506	0.006005	0.000183	2	3	False	{'bootstrap False, 'max_featu 2, 'n_est...
13	0.711979	0.135041	0.017516	0.001889	2	10	False	{'bootstrap False, 'max_featu 2, 'n_est...
14	0.220098	0.003218	0.006009	0.000223	3	3	False	{'bootstrap False, 'max_featu 3, 'n_est...
15	0.922887	0.096323	0.019085	0.001845	3	10	False	{'bootstrap False, 'max_featu 3, 'n_est...
16	0.278346	0.007092	0.006117	0.000349	4	3	False	{'bootstrap False, 'max_featu 4, 'n_est...
17	1.396295	0.350659	0.021608	0.005539	4	10	False	{'bootstrap False, 'max_featu 4, 'n_est...

18 rows × 23 columns

Randomized Search

In [112]:
```
from sklearn.model_selection import RandomizedSearchCV
from scipy.stats import randint

param_distribs = {
        'n_estimators': randint(low=1, high=200),
        'max_features': randint(low=1, high=8),
    }

forest_reg = RandomForestRegressor(random_state=42)
rnd_search = RandomizedSearchCV(forest_reg, param_distributions=param_distribs,
                                n_iter=10, cv=5, scoring='neg_mean_squared_erro
r', random_state=42)
rnd_search.fit(housing_prepared, housing_labels)
```

Out[112]:

▸ **RandomizedSearchCV**

(https://scikit-learn.org/1.6/modules/generate

▸ **best_estimator_: RandomForestRegressor**

▸ RandomForestRegressor

(https://scikit-learn.org/1.6/modules/generated/sklea

In [113]:
```
cvres = rnd_search.cv_results_
for mean_score, params in zip(cvres["mean_test_score"], cvres["params"]):
    print(np.sqrt(-mean_score), params)
```

```
49117.55344336652 {'max_features': 7, 'n_estimators': 180}
51450.63202856348 {'max_features': 5, 'n_estimators': 15}
50692.53588182537 {'max_features': 3, 'n_estimators': 72}
50783.614493515 {'max_features': 5, 'n_estimators': 21}
49162.89877456354 {'max_features': 7, 'n_estimators': 122}
50655.798471042704 {'max_features': 3, 'n_estimators': 75}
50513.856319990606 {'max_features': 3, 'n_estimators': 88}
49521.17201976928 {'max_features': 5, 'n_estimators': 100}
50302.90440763418 {'max_features': 3, 'n_estimators': 150}
65167.02018649492 {'max_features': 5, 'n_estimators': 2}
```

Analyze the Best Models and Their Errors

In [114]:
```
feature_importances = grid_search.best_estimator_.feature_importances_
feature_importances
```

Out[114]:
```
array([6.96542523e-02, 6.04213840e-02, 4.21882202e-02, 1.52450557e-02,
       1.55545295e-02, 1.58491147e-02, 1.49346552e-02, 3.79009225e-01,
       5.47789150e-02, 1.07031322e-01, 4.82031213e-02, 6.79266007e-03,
       1.65706303e-01, 7.83480660e-05, 1.52473276e-03, 3.02816106e-03])
```

In [115]:

```python
extra_attribs = ["rooms_per_hhold", "pop_per_hhold", "bedrooms_per_room"]
#cat_encoder = cat_pipeline.named_steps["cat_encoder"] # old solution
cat_encoder = full_pipeline.named_transformers_["cat"]
cat_one_hot_attribs = list(cat_encoder.categories_[0])
attributes = num_attribs + extra_attribs + cat_one_hot_attribs
sorted(zip(feature_importances, attributes), reverse=True)
```

Out[115]:

```
[(np.float64(0.3790092248170967), 'median_income'),
 (np.float64(0.16570630316895876), 'INLAND'),
 (np.float64(0.10703132208204354), 'pop_per_hhold'),
 (np.float64(0.06965425227942929), 'longitude'),
 (np.float64(0.0604213840080722), 'latitude'),
 (np.float64(0.054778915018283726), 'rooms_per_hhold'),
 (np.float64(0.048203121338269206), 'bedrooms_per_room'),
 (np.float64(0.04218822024391753), 'housing_median_age'),
 (np.float64(0.015849114744428634), 'population'),
 (np.float64(0.015554529490469328), 'total_bedrooms'),
 (np.float64(0.01524505568840977), 'total_rooms'),
 (np.float64(0.014934655161887776), 'households'),
 (np.float64(0.006792660074259966), '<1H OCEAN'),
 (np.float64(0.0030281610628962747), 'NEAR OCEAN'),
 (np.float64(0.0015247327555504937), 'NEAR BAY'),
 (np.float64(7.834806602687504e-05), 'ISLAND')]
```

Evaluate Your System on the Test Set

In [116]:
```
final_model = grid_search.best_estimator_

X_test = strat_test_set.drop("median_house_value", axis=1)
y_test = strat_test_set["median_house_value"].copy()

X_test_prepared = full_pipeline.transform(X_test)
final_predictions = final_model.predict(X_test_prepared)

final_mse = mean_squared_error(y_test, final_predictions)
final_rmse = np.sqrt(final_mse)
```

In [117]:
```
final_rmse
```

Out[117]:
```
np.float64(47873.26095812988)
```

We can compute a 95% confidence interval for the test RMSE:

In [118]:
```
from scipy import stats

confidence = 0.95
squared_errors = (final_predictions - y_test) ** 2
np.sqrt(stats.t.interval(confidence, len(squared_errors) - 1,
                         loc=squared_errors.mean(),
                         scale=stats.sem(squared_errors)))
```

Out[118]:
```
array([45893.36082829, 49774.46796717])
```

We could compute the interval manually like this:

In [119]:
```
m = len(squared_errors)
mean = squared_errors.mean()
tscore = stats.t.ppf((1 + confidence) / 2, df=m - 1)
tmargin = tscore * squared_errors.std(ddof=1) / np.sqrt(m)
np.sqrt(mean - tmargin), np.sqrt(mean + tmargin)
```

Out[119]:
```
(np.float64(45893.360828285535), np.float64(49774.46796717361))
```

Alternatively, we could use a z-scores rather than t-scores:

In [120]:
```
zscore = stats.norm.ppf((1 + confidence) / 2)
zmargin = zscore * squared_errors.std(ddof=1) / np.sqrt(m)
np.sqrt(mean - zmargin), np.sqrt(mean + zmargin)
```

Out[120]:
```
(np.float64(45893.9540110131), np.float64(49773.92103065038))
```

Extra material

A full pipeline with both preparation and prediction

In [121]:
```
full_pipeline_with_predictor = Pipeline([
        ("preparation", full_pipeline),
        ("linear", LinearRegression())
    ])

full_pipeline_with_predictor.fit(housing, housing_labels)
full_pipeline_with_predictor.predict(some_data)
```

Out[121]:
```
array([ 85657.90192014, 305492.60737488, 152056.46122456, 186095.70946094,
       244550.67966089])
```

Model persistence using joblib

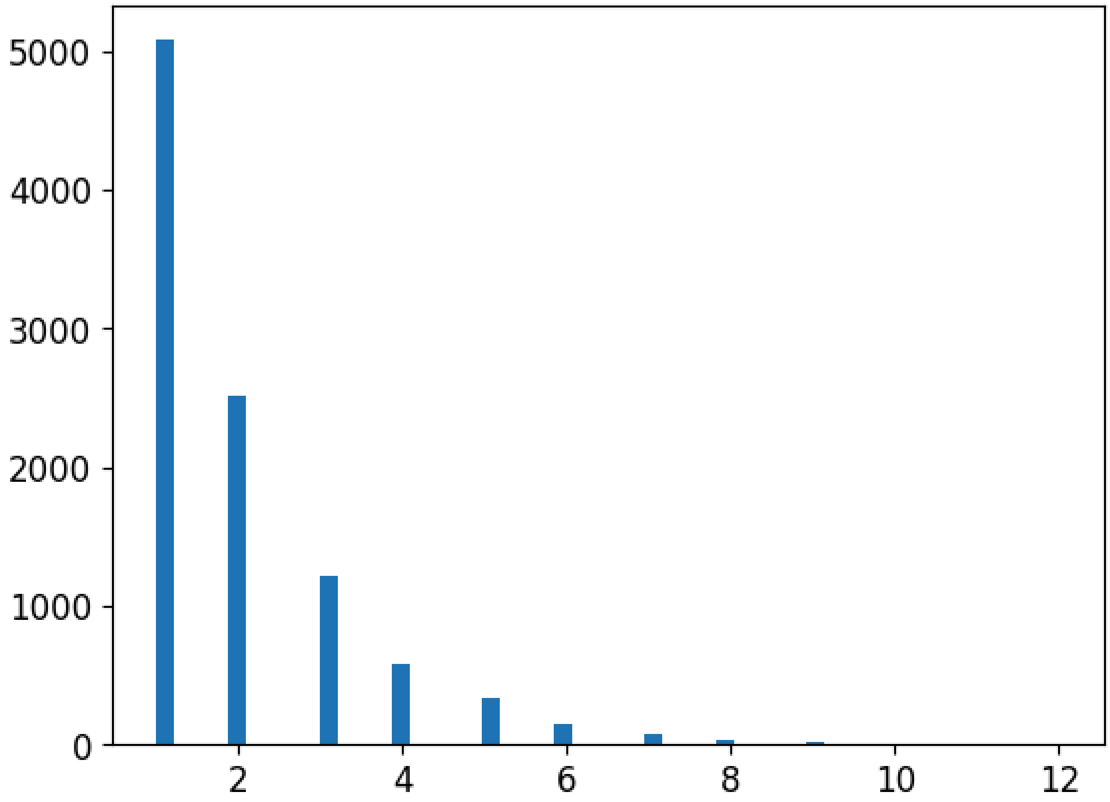
In [122]:
```
my_model = full_pipeline_with_predictor
```

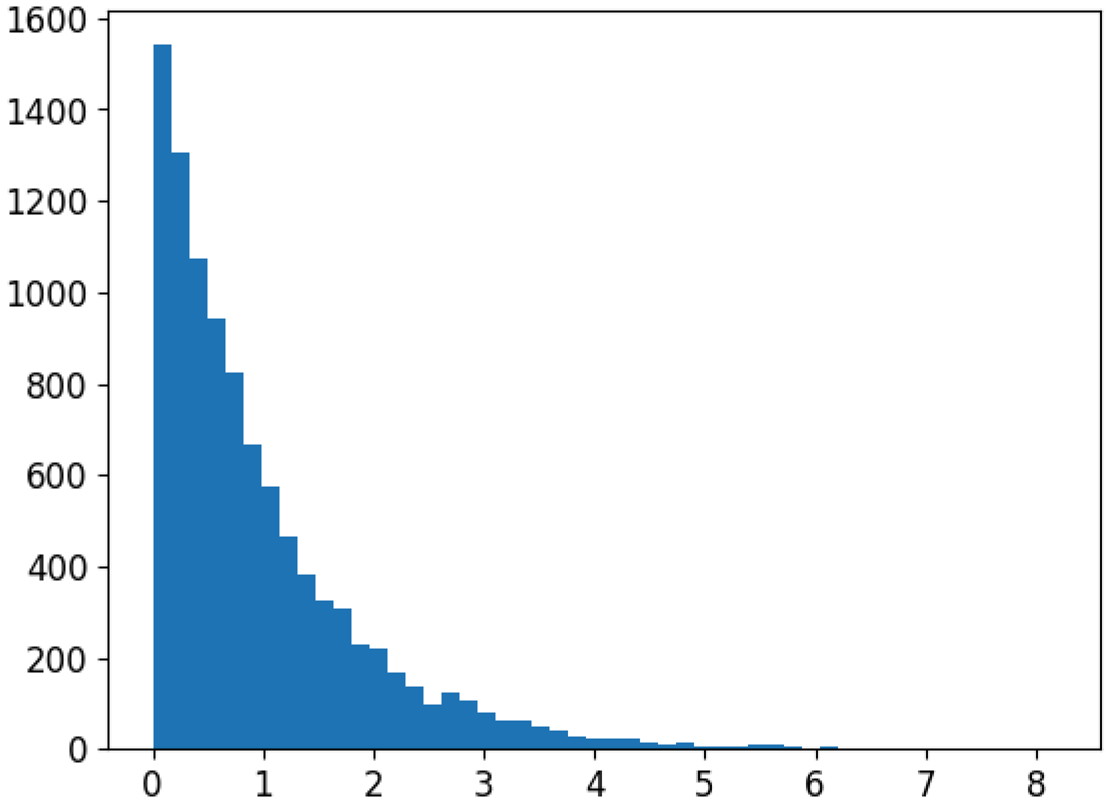
In [123]:
```
import joblib
joblib.dump(my_model, "my_model.pkl") # DIFF
#...
my_model_loaded = joblib.load("my_model.pkl") # DIFF
```

Example SciPy distributions for `RandomizedSearchCV`

In [124]:

```
from scipy.stats import geom, expon
geom_distrib=geom(0.5).rvs(10000, random_state=42)
expon_distrib=expon(scale=1).rvs(10000, random_state=42)
plt.hist(geom_distrib, bins=50)
plt.show()
plt.hist(expon_distrib, bins=50)
plt.show()
```





Exercise solutions

1.

Question: Try a Support Vector Machine regressor (`sklearn.svm.SVR`), with various hyperparameters such as `kernel="linear"` (with various values for the `C` hyperparameter) or `kernel="rbf"` (with various values for the `C` and `gamma` hyperparameters). Don't worry about what these hyperparameters mean for now. How does the best `SVR` predictor perform?

Warning: the following cell may take close to 30 minutes to run, or more depending on your hardware.

In [125]:

```
from sklearn.model_selection import GridSearchCV

param_grid = [
        {'kernel': ['linear'], 'C': [10., 30., 100., 300., 1000., 3000., 10000., 30000.0]},
        {'kernel': ['rbf'], 'C': [1.0, 3.0, 10., 30., 100., 300., 1000.0],
         'gamma': [0.01, 0.03, 0.1, 0.3, 1.0, 3.0]},
    ]

svm_reg = SVR()
grid_search = GridSearchCV(svm_reg, param_grid, cv=5, scoring='neg_mean_squared_error', verbose=2)
grid_search.fit(housing_prepared, housing_labels)
```

```
Fitting 5 folds for each of 50 candidates, totalling 250 fits
[CV] END ..............................C=10.0, kernel=linear; total time=   7.
1s
[CV] END ..............................C=10.0, kernel=linear; total time=  10.
3s
[CV] END ..............................C=10.0, kernel=linear; total time=  12.
3s
[CV] END ..............................C=10.0, kernel=linear; total time=  10.
0s
[CV] END ..............................C=10.0, kernel=linear; total time=   7.
2s
[CV] END ..............................C=30.0, kernel=linear; total time=   8.
4s
[CV] END ..............................C=30.0, kernel=linear; total time=   8.
0s
[CV] END ..............................C=30.0, kernel=linear; total time=   7.
5s
[CV] END ..............................C=30.0, kernel=linear; total time=   8.
2s
[CV] END ..............................C=30.0, kernel=linear; total time=   7.
2s
[CV] END .............................C=100.0, kernel=linear; total time=   8.
4s
[CV] END .............................C=100.0, kernel=linear; total time=  10.
4s
[CV] END .............................C=100.0, kernel=linear; total time=   7.
0s
[CV] END .............................C=100.0, kernel=linear; total time=   8.
9s
[CV] END .............................C=100.0, kernel=linear; total time=   8.
4s
[CV] END .............................C=300.0, kernel=linear; total time=   7.
4s
[CV] END .............................C=300.0, kernel=linear; total time=   8.
3s
```

```
[CV] END .............................C=300.0, kernel=linear; total time=   8.
1s
[CV] END .............................C=300.0, kernel=linear; total time=   7.
3s
[CV] END .............................C=300.0, kernel=linear; total time=   8.
3s
[CV] END ............................C=1000.0, kernel=linear; total time=   7.
4s
[CV] END ............................C=1000.0, kernel=linear; total time=   8.
4s
[CV] END ............................C=1000.0, kernel=linear; total time=   8.
5s
[CV] END ............................C=1000.0, kernel=linear; total time=   7.
9s
[CV] END ............................C=1000.0, kernel=linear; total time=   9.
5s
[CV] END ............................C=3000.0, kernel=linear; total time=   9.
5s
[CV] END ............................C=3000.0, kernel=linear; total time=   9.
0s
[CV] END ............................C=3000.0, kernel=linear; total time=   8.
3s
[CV] END ............................C=3000.0, kernel=linear; total time=   9.
3s
[CV] END ............................C=3000.0, kernel=linear; total time=   9.
1s
[CV] END ...........................C=10000.0, kernel=linear; total time=  11.
3s
[CV] END ...........................C=10000.0, kernel=linear; total time=  10.
6s
[CV] END ...........................C=10000.0, kernel=linear; total time=  11.
2s
[CV] END ...........................C=10000.0, kernel=linear; total time=  11.
0s
[CV] END ...........................C=10000.0, kernel=linear; total time=  11.
3s
```

```
[CV] END ...........................C=30000.0, kernel=linear; total time=  17.
2s
[CV] END ...........................C=30000.0, kernel=linear; total time=  17.
5s
[CV] END ...........................C=30000.0, kernel=linear; total time=  15.
2s
[CV] END ...........................C=30000.0, kernel=linear; total time=  17.
5s
[CV] END ...........................C=30000.0, kernel=linear; total time=  18.
3s
[CV] END ......................C=1.0, gamma=0.01, kernel=rbf; total time=  12.
1s
[CV] END ......................C=1.0, gamma=0.01, kernel=rbf; total time=  12.
2s
[CV] END ......................C=1.0, gamma=0.01, kernel=rbf; total time=  12.
2s
[CV] END ......................C=1.0, gamma=0.01, kernel=rbf; total time=  12.
3s
[CV] END ......................C=1.0, gamma=0.01, kernel=rbf; total time=  12.
5s
[CV] END ......................C=1.0, gamma=0.03, kernel=rbf; total time=  12.
9s
[CV] END ......................C=1.0, gamma=0.03, kernel=rbf; total time=  18.
9s
[CV] END ......................C=1.0, gamma=0.03, kernel=rbf; total time=  15.
3s
[CV] END ......................C=1.0, gamma=0.03, kernel=rbf; total time=  17.
1s
[CV] END ......................C=1.0, gamma=0.03, kernel=rbf; total time=  16.
4s
[CV] END .......................C=1.0, gamma=0.1, kernel=rbf; total time=  15.
1s
[CV] END .......................C=1.0, gamma=0.1, kernel=rbf; total time=  17.
4s
[CV] END .......................C=1.0, gamma=0.1, kernel=rbf; total time=  16.
2s
```

```
[CV] END .......................C=1.0, gamma=0.1, kernel=rbf; total time=  12.
8s
[CV] END .......................C=1.0, gamma=0.1, kernel=rbf; total time=  14.
4s
[CV] END .......................C=1.0, gamma=0.3, kernel=rbf; total time=  13.
0s
[CV] END .......................C=1.0, gamma=0.3, kernel=rbf; total time=  11.
6s
[CV] END .......................C=1.0, gamma=0.3, kernel=rbf; total time=  20.
1s
[CV] END .......................C=1.0, gamma=0.3, kernel=rbf; total time=  13.
5s
[CV] END .......................C=1.0, gamma=0.3, kernel=rbf; total time=  17.
2s
[CV] END .......................C=1.0, gamma=1.0, kernel=rbf; total time=  13.
0s
[CV] END .......................C=1.0, gamma=1.0, kernel=rbf; total time=  14.
9s
[CV] END .......................C=1.0, gamma=1.0, kernel=rbf; total time=  13.
6s
[CV] END .......................C=1.0, gamma=1.0, kernel=rbf; total time=  14.
4s
[CV] END .......................C=1.0, gamma=1.0, kernel=rbf; total time=  14.
4s
[CV] END .......................C=1.0, gamma=3.0, kernel=rbf; total time=  13.
0s
[CV] END .......................C=1.0, gamma=3.0, kernel=rbf; total time=  12.
3s
[CV] END .......................C=1.0, gamma=3.0, kernel=rbf; total time=  12.
3s
[CV] END .......................C=1.0, gamma=3.0, kernel=rbf; total time=  12.
3s
[CV] END .......................C=1.0, gamma=3.0, kernel=rbf; total time=  12.
3s
[CV] END ......................C=3.0, gamma=0.01, kernel=rbf; total time=  12.
1s
```

```
[CV] END .......................C=3.0, gamma=0.01, kernel=rbf; total time=  12.
1s
[CV] END .......................C=3.0, gamma=0.01, kernel=rbf; total time=  12.
0s
[CV] END .......................C=3.0, gamma=0.01, kernel=rbf; total time=  12.
6s
[CV] END .......................C=3.0, gamma=0.01, kernel=rbf; total time=  12.
0s
[CV] END .......................C=3.0, gamma=0.03, kernel=rbf; total time=  12.
0s
[CV] END .......................C=3.0, gamma=0.03, kernel=rbf; total time=  11.
9s
[CV] END .......................C=3.0, gamma=0.03, kernel=rbf; total time=  11.
9s
[CV] END .......................C=3.0, gamma=0.03, kernel=rbf; total time=  13.
1s
[CV] END .......................C=3.0, gamma=0.03, kernel=rbf; total time=  11.
9s
[CV] END ........................C=3.0, gamma=0.1, kernel=rbf; total time=  12.
0s
[CV] END ........................C=3.0, gamma=0.1, kernel=rbf; total time=  12.
0s
[CV] END ........................C=3.0, gamma=0.1, kernel=rbf; total time=  12.
0s
[CV] END ........................C=3.0, gamma=0.1, kernel=rbf; total time=  11.
8s
[CV] END ........................C=3.0, gamma=0.1, kernel=rbf; total time=  11.
4s
[CV] END ........................C=3.0, gamma=0.3, kernel=rbf; total time=  11.
1s
[CV] END ........................C=3.0, gamma=0.3, kernel=rbf; total time=  11.
6s
[CV] END ........................C=3.0, gamma=0.3, kernel=rbf; total time=  11.
6s
[CV] END ........................C=3.0, gamma=0.3, kernel=rbf; total time=  11.
6s
```

```
[CV] END ......................C=3.0, gamma=0.3, kernel=rbf; total time=  11.
6s
[CV] END ......................C=3.0, gamma=1.0, kernel=rbf; total time=  11.
3s
[CV] END ......................C=3.0, gamma=1.0, kernel=rbf; total time=  11.
3s
[CV] END ......................C=3.0, gamma=1.0, kernel=rbf; total time=  11.
3s
[CV] END ......................C=3.0, gamma=1.0, kernel=rbf; total time=  11.
5s
[CV] END ......................C=3.0, gamma=1.0, kernel=rbf; total time=  11.
0s
[CV] END ......................C=3.0, gamma=3.0, kernel=rbf; total time=  12.
3s
[CV] END ......................C=3.0, gamma=3.0, kernel=rbf; total time=  12.
2s
[CV] END ......................C=3.0, gamma=3.0, kernel=rbf; total time=  12.
2s
[CV] END ......................C=3.0, gamma=3.0, kernel=rbf; total time=  12.
2s
[CV] END ......................C=3.0, gamma=3.0, kernel=rbf; total time=  12.
1s
[CV] END ....................C=10.0, gamma=0.01, kernel=rbf; total time=  12.
0s
[CV] END ....................C=10.0, gamma=0.01, kernel=rbf; total time=  13.
1s
[CV] END ....................C=10.0, gamma=0.01, kernel=rbf; total time=  12.
0s
[CV] END ....................C=10.0, gamma=0.01, kernel=rbf; total time=  12.
0s
[CV] END ....................C=10.0, gamma=0.01, kernel=rbf; total time=  12.
0s
[CV] END ....................C=10.0, gamma=0.03, kernel=rbf; total time=  12.
0s
[CV] END ....................C=10.0, gamma=0.03, kernel=rbf; total time=  12.
0s
```

```
[CV] END .....................C=10.0, gamma=0.03, kernel=rbf; total time=  11.
9s
[CV] END .....................C=10.0, gamma=0.03, kernel=rbf; total time=  11.
9s
[CV] END .....................C=10.0, gamma=0.03, kernel=rbf; total time=  11.
9s
[CV] END ......................C=10.0, gamma=0.1, kernel=rbf; total time=  11.
8s
[CV] END ......................C=10.0, gamma=0.1, kernel=rbf; total time=  11.
8s
[CV] END ......................C=10.0, gamma=0.1, kernel=rbf; total time=  11.
8s
[CV] END ......................C=10.0, gamma=0.1, kernel=rbf; total time=  11.
9s
[CV] END ......................C=10.0, gamma=0.1, kernel=rbf; total time=  11.
9s
[CV] END ......................C=10.0, gamma=0.3, kernel=rbf; total time=  11.
7s
[CV] END ......................C=10.0, gamma=0.3, kernel=rbf; total time=  11.
3s
[CV] END ......................C=10.0, gamma=0.3, kernel=rbf; total time=  10.
9s
[CV] END ......................C=10.0, gamma=0.3, kernel=rbf; total time=  11.
6s
[CV] END ......................C=10.0, gamma=0.3, kernel=rbf; total time=  11.
6s
[CV] END ......................C=10.0, gamma=1.0, kernel=rbf; total time=  11.
3s
[CV] END ......................C=10.0, gamma=1.0, kernel=rbf; total time=  11.
3s
[CV] END ......................C=10.0, gamma=1.0, kernel=rbf; total time=  11.
2s
[CV] END ......................C=10.0, gamma=1.0, kernel=rbf; total time=  11.
3s
[CV] END ......................C=10.0, gamma=1.0, kernel=rbf; total time=  12.
7s
```

```
[CV] END ......................C=10.0, gamma=3.0, kernel=rbf; total time=  12.
4s
[CV] END ......................C=10.0, gamma=3.0, kernel=rbf; total time=  12.
5s
[CV] END ......................C=10.0, gamma=3.0, kernel=rbf; total time=  12.
4s
[CV] END ......................C=10.0, gamma=3.0, kernel=rbf; total time=  12.
5s
[CV] END ......................C=10.0, gamma=3.0, kernel=rbf; total time=  12.
5s
[CV] END .....................C=30.0, gamma=0.01, kernel=rbf; total time=  12.
1s
[CV] END .....................C=30.0, gamma=0.01, kernel=rbf; total time=  12.
1s
[CV] END .....................C=30.0, gamma=0.01, kernel=rbf; total time=  11.
9s
[CV] END .....................C=30.0, gamma=0.01, kernel=rbf; total time=  11.
8s
[CV] END .....................C=30.0, gamma=0.01, kernel=rbf; total time=  11.
6s
[CV] END .....................C=30.0, gamma=0.03, kernel=rbf; total time=  11.
8s
[CV] END .....................C=30.0, gamma=0.03, kernel=rbf; total time=  12.
0s
[CV] END .....................C=30.0, gamma=0.03, kernel=rbf; total time=  11.
9s
[CV] END .....................C=30.0, gamma=0.03, kernel=rbf; total time=  11.
9s
[CV] END .....................C=30.0, gamma=0.03, kernel=rbf; total time=  11.
9s
[CV] END ......................C=30.0, gamma=0.1, kernel=rbf; total time=  11.
8s
[CV] END ......................C=30.0, gamma=0.1, kernel=rbf; total time=  11.
8s
[CV] END ......................C=30.0, gamma=0.1, kernel=rbf; total time=  11.
7s
```

```
[CV] END .......................C=30.0, gamma=0.1, kernel=rbf; total time=  11.
8s
[CV] END .......................C=30.0, gamma=0.1, kernel=rbf; total time=  11.
7s
[CV] END .......................C=30.0, gamma=0.3, kernel=rbf; total time=  11.
6s
[CV] END .......................C=30.0, gamma=0.3, kernel=rbf; total time=  12.
1s
[CV] END .......................C=30.0, gamma=0.3, kernel=rbf; total time=  12.
4s
[CV] END .......................C=30.0, gamma=0.3, kernel=rbf; total time=  11.
7s
[CV] END .......................C=30.0, gamma=0.3, kernel=rbf; total time=  11.
3s
[CV] END .......................C=30.0, gamma=1.0, kernel=rbf; total time=  10.
6s
[CV] END .......................C=30.0, gamma=1.0, kernel=rbf; total time=  11.
4s
[CV] END .......................C=30.0, gamma=1.0, kernel=rbf; total time=  11.
4s
[CV] END .......................C=30.0, gamma=1.0, kernel=rbf; total time=  11.
3s
[CV] END .......................C=30.0, gamma=1.0, kernel=rbf; total time=  11.
4s
[CV] END .......................C=30.0, gamma=3.0, kernel=rbf; total time=  12.
4s
[CV] END .......................C=30.0, gamma=3.0, kernel=rbf; total time=  12.
4s
[CV] END .......................C=30.0, gamma=3.0, kernel=rbf; total time=  12.
4s
[CV] END .......................C=30.0, gamma=3.0, kernel=rbf; total time=  12.
4s
[CV] END .......................C=30.0, gamma=3.0, kernel=rbf; total time=  12.
3s
[CV] END .....................C=100.0, gamma=0.01, kernel=rbf; total time=  11.
9s
```

```
[CV] END ...................C=100.0, gamma=0.01, kernel=rbf; total time=  12.
0s
[CV] END ...................C=100.0, gamma=0.01, kernel=rbf; total time=  12.
0s
[CV] END ...................C=100.0, gamma=0.01, kernel=rbf; total time=  12.
1s
[CV] END ...................C=100.0, gamma=0.01, kernel=rbf; total time=  12.
1s
[CV] END ...................C=100.0, gamma=0.03, kernel=rbf; total time=  11.
9s
[CV] END ...................C=100.0, gamma=0.03, kernel=rbf; total time=  11.
8s
[CV] END ...................C=100.0, gamma=0.03, kernel=rbf; total time=  11.
3s
[CV] END ...................C=100.0, gamma=0.03, kernel=rbf; total time=  11.
4s
[CV] END ...................C=100.0, gamma=0.03, kernel=rbf; total time=  13.
3s
[CV] END ....................C=100.0, gamma=0.1, kernel=rbf; total time=  11.
6s
[CV] END ....................C=100.0, gamma=0.1, kernel=rbf; total time=  13.
9s
[CV] END ....................C=100.0, gamma=0.1, kernel=rbf; total time=  11.
6s
[CV] END ....................C=100.0, gamma=0.1, kernel=rbf; total time=  11.
7s
[CV] END ....................C=100.0, gamma=0.1, kernel=rbf; total time=  11.
7s
[CV] END ....................C=100.0, gamma=0.3, kernel=rbf; total time=  11.
5s
[CV] END ....................C=100.0, gamma=0.3, kernel=rbf; total time=  13.
0s
[CV] END ....................C=100.0, gamma=0.3, kernel=rbf; total time=  11.
5s
[CV] END ....................C=100.0, gamma=0.3, kernel=rbf; total time=  11.
4s
```

```
[CV] END .....................C=100.0, gamma=0.3, kernel=rbf; total time=  11.
5s
[CV] END .....................C=100.0, gamma=1.0, kernel=rbf; total time=  11.
3s
[CV] END .....................C=100.0, gamma=1.0, kernel=rbf; total time=  11.
5s
[CV] END .....................C=100.0, gamma=1.0, kernel=rbf; total time=  11.
1s
[CV] END .....................C=100.0, gamma=1.0, kernel=rbf; total time=  10.
5s
[CV] END .....................C=100.0, gamma=1.0, kernel=rbf; total time=  11.
3s
[CV] END .....................C=100.0, gamma=3.0, kernel=rbf; total time=  12.
3s
[CV] END .....................C=100.0, gamma=3.0, kernel=rbf; total time=  12.
3s
[CV] END .....................C=100.0, gamma=3.0, kernel=rbf; total time=  12.
3s
[CV] END .....................C=100.0, gamma=3.0, kernel=rbf; total time=  12.
2s
[CV] END .....................C=100.0, gamma=3.0, kernel=rbf; total time=  12.
3s
[CV] END ....................C=300.0, gamma=0.01, kernel=rbf; total time=  11.
8s
[CV] END ....................C=300.0, gamma=0.01, kernel=rbf; total time=  11.
8s
[CV] END ....................C=300.0, gamma=0.01, kernel=rbf; total time=  13.
0s
[CV] END ....................C=300.0, gamma=0.01, kernel=rbf; total time=  11.
8s
[CV] END ....................C=300.0, gamma=0.01, kernel=rbf; total time=  11.
8s
[CV] END ....................C=300.0, gamma=0.03, kernel=rbf; total time=  11.
5s
[CV] END ....................C=300.0, gamma=0.03, kernel=rbf; total time=  11.
7s
```

```
[CV] END ....................C=300.0, gamma=0.03, kernel=rbf; total time=  11.
7s
[CV] END ....................C=300.0, gamma=0.03, kernel=rbf; total time=  11.
4s
[CV] END ....................C=300.0, gamma=0.03, kernel=rbf; total time=  11.
0s
[CV] END .....................C=300.0, gamma=0.1, kernel=rbf; total time=  11.
4s
[CV] END .....................C=300.0, gamma=0.1, kernel=rbf; total time=  11.
4s
[CV] END .....................C=300.0, gamma=0.1, kernel=rbf; total time=  11.
5s
[CV] END .....................C=300.0, gamma=0.1, kernel=rbf; total time=  11.
5s
[CV] END .....................C=300.0, gamma=0.1, kernel=rbf; total time=  11.
5s
[CV] END .....................C=300.0, gamma=0.3, kernel=rbf; total time=  11.
4s
[CV] END .....................C=300.0, gamma=0.3, kernel=rbf; total time=  11.
4s
[CV] END .....................C=300.0, gamma=0.3, kernel=rbf; total time=  11.
6s
[CV] END .....................C=300.0, gamma=0.3, kernel=rbf; total time=  11.
2s
[CV] END .....................C=300.0, gamma=0.3, kernel=rbf; total time=  10.
5s
[CV] END .....................C=300.0, gamma=1.0, kernel=rbf; total time=  11.
3s
[CV] END .....................C=300.0, gamma=1.0, kernel=rbf; total time=  11.
3s
[CV] END .....................C=300.0, gamma=1.0, kernel=rbf; total time=  11.
3s
[CV] END .....................C=300.0, gamma=1.0, kernel=rbf; total time=  11.
3s
[CV] END .....................C=300.0, gamma=1.0, kernel=rbf; total time=  11.
3s
```

```
[CV] END .....................C=300.0, gamma=3.0, kernel=rbf; total time=  13.
5s
[CV] END .....................C=300.0, gamma=3.0, kernel=rbf; total time=  12.
3s
[CV] END .....................C=300.0, gamma=3.0, kernel=rbf; total time=  12.
4s
[CV] END .....................C=300.0, gamma=3.0, kernel=rbf; total time=  12.
5s
[CV] END .....................C=300.0, gamma=3.0, kernel=rbf; total time=  12.
4s
[CV] END ...................C=1000.0, gamma=0.01, kernel=rbf; total time=  11.
7s
[CV] END ...................C=1000.0, gamma=0.01, kernel=rbf; total time=  11.
2s
[CV] END ...................C=1000.0, gamma=0.01, kernel=rbf; total time=  10.
8s
[CV] END ...................C=1000.0, gamma=0.01, kernel=rbf; total time=  11.
6s
[CV] END ...................C=1000.0, gamma=0.01, kernel=rbf; total time=  11.
5s
[CV] END ...................C=1000.0, gamma=0.03, kernel=rbf; total time=  11.
4s
[CV] END ...................C=1000.0, gamma=0.03, kernel=rbf; total time=  11.
3s
[CV] END ...................C=1000.0, gamma=0.03, kernel=rbf; total time=  11.
3s
[CV] END ...................C=1000.0, gamma=0.03, kernel=rbf; total time=  11.
4s
[CV] END ...................C=1000.0, gamma=0.03, kernel=rbf; total time=  11.
4s
[CV] END ....................C=1000.0, gamma=0.1, kernel=rbf; total time=  11.
4s
[CV] END ....................C=1000.0, gamma=0.1, kernel=rbf; total time=  10.
6s
[CV] END ....................C=1000.0, gamma=0.1, kernel=rbf; total time=  10.
9s
```

```
[CV] END ....................C=1000.0, gamma=0.1, kernel=rbf; total time=  11.
3s
[CV] END ....................C=1000.0, gamma=0.1, kernel=rbf; total time=  11.
3s
[CV] END ....................C=1000.0, gamma=0.3, kernel=rbf; total time=  11.
2s
[CV] END ....................C=1000.0, gamma=0.3, kernel=rbf; total time=  11.
2s
[CV] END ....................C=1000.0, gamma=0.3, kernel=rbf; total time=  11.
2s
[CV] END ....................C=1000.0, gamma=0.3, kernel=rbf; total time=  12.
3s
[CV] END ....................C=1000.0, gamma=0.3, kernel=rbf; total time=  11.
4s
[CV] END ....................C=1000.0, gamma=1.0, kernel=rbf; total time=  10.
6s
[CV] END ....................C=1000.0, gamma=1.0, kernel=rbf; total time=  10.
7s
[CV] END ....................C=1000.0, gamma=1.0, kernel=rbf; total time=  11.
3s
[CV] END ....................C=1000.0, gamma=1.0, kernel=rbf; total time=  11.
2s
[CV] END ....................C=1000.0, gamma=1.0, kernel=rbf; total time=  11.
3s
[CV] END ....................C=1000.0, gamma=3.0, kernel=rbf; total time=  12.
3s
[CV] END ....................C=1000.0, gamma=3.0, kernel=rbf; total time=  12.
2s
[CV] END ....................C=1000.0, gamma=3.0, kernel=rbf; total time=  12.
3s
[CV] END ....................C=1000.0, gamma=3.0, kernel=rbf; total time=  12.
3s
[CV] END ....................C=1000.0, gamma=3.0, kernel=rbf; total time=  12.
3s
```

Out[125]:

```
▸ GridSearchCV
(https://scikit-learn.org/1.6/modules/generated/sklearn.model_select
▸ best_estimator_: SVR
    ▸ SVR
    (https://scikit-learn.org/1.6/modules/generated/sklearn.svm.SVR.html)
```

The best model achieves the following score (evaluated using 5-fold cross validation):

In [126]:

```
negative_mse = grid_search.best_score_
rmse = np.sqrt(-negative_mse)
rmse
```

Out[126]:

```
np.float64(70286.61835383571)
```

That's much worse than the `RandomForestRegressor`. Let's check the best hyperparameters found:

In [127]:

```
grid_search.best_params_
```

Out[127]:

```
{'C': 30000.0, 'kernel': 'linear'}
```

The linear kernel seems better than the RBF kernel. Notice that the value of `C` is the maximum tested value. When this happens you definitely want to launch the grid search again with higher values for `C` (removing the smallest values), because it is likely that higher values of `C` will be better.

2.

Question: Try replacing `GridSearchCV` with `RandomizedSearchCV`.

Warning: the following cell may take close to 45 minutes to run, or more depending on your hardware.

In [128]:

```
from sklearn.model_selection import RandomizedSearchCV
from scipy.stats import expon, reciprocal

# see https://docs.scipy.org/doc/scipy/reference/stats.html
# for `expon()` and `reciprocal()` documentation and more probability distributi
on functions.

# Note: gamma is ignored when kernel is "linear"
param_distribs = {
        'kernel': ['linear', 'rbf'],
        'C': reciprocal(20, 200000),
        'gamma': expon(scale=1.0),
    }

svm_reg = SVR()
rnd_search = RandomizedSearchCV(svm_reg, param_distributions=param_distribs,
                                n_iter=50, cv=5, scoring='neg_mean_squared_erro
r',
                                verbose=2, random_state=42)
rnd_search.fit(housing_prepared, housing_labels)
```

```
Fitting 5 folds for each of 50 candidates, totalling 250 fits
[CV] END C=629.7823295913721, gamma=3.010121430917521, kernel=linear; total ti
me=   8.3s
[CV] END C=629.7823295913721, gamma=3.010121430917521, kernel=linear; total ti
me=   7.1s
[CV] END C=629.7823295913721, gamma=3.010121430917521, kernel=linear; total ti
me=   8.2s
[CV] END C=629.7823295913721, gamma=3.010121430917521, kernel=linear; total ti
me=   7.2s
[CV] END C=629.7823295913721, gamma=3.010121430917521, kernel=linear; total ti
me=   8.4s
[CV] END C=26290.20646430022, gamma=0.9084469696321253, kernel=rbf; total time
=  13.8s
[CV] END C=26290.20646430022, gamma=0.9084469696321253, kernel=rbf; total time
=  13.9s
[CV] END C=26290.20646430022, gamma=0.9084469696321253, kernel=rbf; total time
=  14.3s
[CV] END C=26290.20646430022, gamma=0.9084469696321253, kernel=rbf; total time
=  15.5s
[CV] END C=26290.20646430022, gamma=0.9084469696321253, kernel=rbf; total time
=  14.7s
[CV] END C=84.14107900575871, gamma=0.059838768608680676, kernel=rbf; total ti
me=  11.8s
[CV] END C=84.14107900575871, gamma=0.059838768608680676, kernel=rbf; total ti
me=  11.9s
[CV] END C=84.14107900575871, gamma=0.059838768608680676, kernel=rbf; total ti
me=  11.9s
[CV] END C=84.14107900575871, gamma=0.059838768608680676, kernel=rbf; total ti
me=  11.9s
[CV] END C=84.14107900575871, gamma=0.059838768608680676, kernel=rbf; total ti
me=  11.1s
[CV] END C=432.37884813148844, gamma=0.15416196746656105, kernel=linear; total
time=   8.0s
[CV] END C=432.37884813148844, gamma=0.15416196746656105, kernel=linear; total
time=   8.2s
```

```
[CV] END C=432.37884813148844, gamma=0.1541619674656105, kernel=linear; total
time=   7.1s
[CV] END C=432.37884813148844, gamma=0.1541619674656105, kernel=linear; total
time=   8.3s
[CV] END C=432.37884813148844, gamma=0.1541619674656105, kernel=linear; total
time=   8.2s
[CV] END C=24.175082946113903, gamma=3.503557475158312, kernel=rbf; total time
=  13.1s
[CV] END C=24.175082946113903, gamma=3.503557475158312, kernel=rbf; total time
=  12.9s
[CV] END C=24.175082946113903, gamma=3.503557475158312, kernel=rbf; total time
=  13.0s
[CV] END C=24.175082946113903, gamma=3.503557475158312, kernel=rbf; total time
=  12.9s
[CV] END C=24.175082946113903, gamma=3.503557475158312, kernel=rbf; total time
=  12.9s
[CV] END C=113564.03940586244, gamma=0.0007790692366582295, kernel=rbf; total
time=  11.6s
[CV] END C=113564.03940586244, gamma=0.0007790692366582295, kernel=rbf; total
time=  11.3s
[CV] END C=113564.03940586244, gamma=0.0007790692366582295, kernel=rbf; total
time=  10.8s
[CV] END C=113564.03940586244, gamma=0.0007790692366582295, kernel=rbf; total
time=  11.5s
[CV] END C=113564.03940586244, gamma=0.0007790692366582295, kernel=rbf; total
time=  11.5s
[CV] END C=108.30488238805071, gamma=0.3627537294604771, kernel=rbf; total tim
e=  11.5s
[CV] END C=108.30488238805071, gamma=0.3627537294604771, kernel=rbf; total tim
e=  11.4s
[CV] END C=108.30488238805071, gamma=0.3627537294604771, kernel=rbf; total tim
e=  11.5s
[CV] END C=108.30488238805071, gamma=0.3627537294604771, kernel=rbf; total tim
e=  11.4s
[CV] END C=108.30488238805071, gamma=0.3627537294604771, kernel=rbf; total tim
e=  11.4s
```

```
[CV] END C=21.34495367264743, gamma=0.023332523598323388, kernel=linear; total
time=   7.1s
[CV] END C=21.34495367264743, gamma=0.023332523598323388, kernel=linear; total
time=   8.2s
[CV] END C=21.34495367264743, gamma=0.023332523598323388, kernel=linear; total
time=   8.3s
[CV] END C=21.34495367264743, gamma=0.023332523598323388, kernel=linear; total
time=   7.0s
[CV] END C=21.34495367264743, gamma=0.023332523598323388, kernel=linear; total
time=   8.3s
[CV] END C=5603.270317432522, gamma=0.15023452872733867, kernel=rbf; total tim
e=  11.3s
[CV] END C=5603.270317432522, gamma=0.15023452872733867, kernel=rbf; total tim
e=  11.4s
[CV] END C=5603.270317432522, gamma=0.15023452872733867, kernel=rbf; total tim
e=  10.6s
[CV] END C=5603.270317432522, gamma=0.15023452872733867, kernel=rbf; total tim
e=  10.8s
[CV] END C=5603.270317432522, gamma=0.15023452872733867, kernel=rbf; total tim
e=  11.2s
[CV] END C=157055.10989448498, gamma=0.26497040005002437, kernel=rbf; total ti
me=  30.7s
[CV] END C=157055.10989448498, gamma=0.26497040005002437, kernel=rbf; total ti
me=  30.0s
[CV] END C=157055.10989448498, gamma=0.26497040005002437, kernel=rbf; total ti
me=  32.9s
[CV] END C=157055.10989448498, gamma=0.26497040005002437, kernel=rbf; total ti
me=  28.1s
[CV] END C=157055.10989448498, gamma=0.26497040005002437, kernel=rbf; total ti
me=  32.1s
[CV] END C=27652.46435873972, gamma=0.2227358621286903, kernel=linear; total t
ime=  16.6s
[CV] END C=27652.46435873972, gamma=0.2227358621286903, kernel=linear; total t
ime=  15.2s
[CV] END C=27652.46435873972, gamma=0.2227358621286903, kernel=linear; total t
ime=  14.9s
```

```
[CV] END C=27652.46435873972, gamma=0.2227358621286903, kernel=linear; total t
ime=  17.8s
[CV] END C=27652.46435873972, gamma=0.2227358621286903, kernel=linear; total t
ime=  16.4s
[CV] END C=171377.39570377997, gamma=0.628789100540856, kernel=linear; total t
ime=  52.3s
[CV] END C=171377.39570377997, gamma=0.628789100540856, kernel=linear; total t
ime=  56.1s
[CV] END C=171377.39570377997, gamma=0.628789100540856, kernel=linear; total t
ime=  43.7s
[CV] END C=171377.39570377997, gamma=0.628789100540856, kernel=linear; total t
ime=  54.1s
[CV] END C=171377.39570377997, gamma=0.628789100540856, kernel=linear; total t
ime=  59.8s
[CV] END C=5385.293820172355, gamma=0.18696125197741642, kernel=linear; total
time=  10.1s
[CV] END C=5385.293820172355, gamma=0.18696125197741642, kernel=linear; total
time=   9.5s
[CV] END C=5385.293820172355, gamma=0.18696125197741642, kernel=linear; total
time=   8.7s
[CV] END C=5385.293820172355, gamma=0.18696125197741642, kernel=linear; total
time=   9.6s
[CV] END C=5385.293820172355, gamma=0.18696125197741642, kernel=linear; total
time=   9.6s
[CV] END C=22.599032166213227, gamma=2.850796878935603, kernel=rbf; total time
=  12.1s
[CV] END C=22.599032166213227, gamma=2.850796878935603, kernel=rbf; total time
=  12.2s
[CV] END C=22.599032166213227, gamma=2.850796878935603, kernel=rbf; total time
=  12.2s
[CV] END C=22.599032166213227, gamma=2.850796878935603, kernel=rbf; total time
=  12.4s
[CV] END C=22.599032166213227, gamma=2.850796878935603, kernel=rbf; total time
=  12.4s
[CV] END C=34246.751946327975, gamma=0.3632878599687583, kernel=linear; total
time=  18.5s
```

```
[CV] END C=34246.751946327975, gamma=0.3632878599687583, kernel=linear; total
time=  19.8s
[CV] END C=34246.751946327975, gamma=0.3632878599687583, kernel=linear; total
time=  16.2s
[CV] END C=34246.751946327975, gamma=0.3632878599687583, kernel=linear; total
time=  17.3s
[CV] END C=34246.751946327975, gamma=0.3632878599687583, kernel=linear; total
time=  20.0s
[CV] END C=167.7278956080511, gamma=0.2757870542258224, kernel=rbf; total time
=  11.3s
[CV] END C=167.7278956080511, gamma=0.2757870542258224, kernel=rbf; total time
=  11.4s
[CV] END C=167.7278956080511, gamma=0.2757870542258224, kernel=rbf; total time
=  11.5s
[CV] END C=167.7278956080511, gamma=0.2757870542258224, kernel=rbf; total time
=  11.5s
[CV] END C=167.7278956080511, gamma=0.2757870542258224, kernel=rbf; total time
=  11.0s
[CV] END C=61.54360542501372, gamma=0.6835472281341501, kernel=linear; total t
ime=   7.7s
[CV] END C=61.54360542501372, gamma=0.6835472281341501, kernel=linear; total t
ime=   8.0s
[CV] END C=61.54360542501372, gamma=0.6835472281341501, kernel=linear; total t
ime=   6.9s
[CV] END C=61.54360542501372, gamma=0.6835472281341501, kernel=linear; total t
ime=   8.2s
[CV] END C=61.54360542501372, gamma=0.6835472281341501, kernel=linear; total t
ime=   7.8s
[CV] END C=98.73897389920917, gamma=0.4960365360493639, kernel=rbf; total time
=  10.6s
[CV] END C=98.73897389920917, gamma=0.4960365360493639, kernel=rbf; total time
=  11.5s
[CV] END C=98.73897389920917, gamma=0.4960365360493639, kernel=rbf; total time
=  11.3s
[CV] END C=98.73897389920917, gamma=0.4960365360493639, kernel=rbf; total time
=  11.4s
```

```
[CV] END C=98.73897389920917, gamma=0.4960365360493639, kernel=rbf; total time
=  11.4s
[CV] END C=8935.505635947806, gamma=0.37354658165762367, kernel=rbf; total tim
e=  11.3s
[CV] END C=8935.505635947806, gamma=0.37354658165762367, kernel=rbf; total tim
e=  11.4s
[CV] END C=8935.505635947806, gamma=0.37354658165762367, kernel=rbf; total tim
e=  11.4s
[CV] END C=8935.505635947806, gamma=0.37354658165762367, kernel=rbf; total tim
e=  11.4s
[CV] END C=8935.505635947806, gamma=0.37354658165762367, kernel=rbf; total tim
e=  10.9s
[CV] END C=135.7677582484244, gamma=0.838636245624803, kernel=linear; total ti
me=   7.7s
[CV] END C=135.7677582484244, gamma=0.838636245624803, kernel=linear; total ti
me=   8.0s
[CV] END C=135.7677582484244, gamma=0.838636245624803, kernel=linear; total ti
me=   6.9s
[CV] END C=135.7677582484244, gamma=0.838636245624803, kernel=linear; total ti
me=   8.2s
[CV] END C=135.7677582484244, gamma=0.838636245624803, kernel=linear; total ti
me=   8.0s
[CV] END C=151136.20282548846, gamma=1.4922453771381408, kernel=rbf; total tim
e= 2.3min
[CV] END C=151136.20282548846, gamma=1.4922453771381408, kernel=rbf; total tim
e= 2.2min
[CV] END C=151136.20282548846, gamma=1.4922453771381408, kernel=rbf; total tim
e= 2.3min
[CV] END C=151136.20282548846, gamma=1.4922453771381408, kernel=rbf; total tim
e= 2.1min
[CV] END C=151136.20282548846, gamma=1.4922453771381408, kernel=rbf; total tim
e= 2.0min
[CV] END C=761.4316758498787, gamma=2.6126336514161914, kernel=linear; total t
ime=   8.4s
[CV] END C=761.4316758498787, gamma=2.6126336514161914, kernel=linear; total t
ime=   7.1s
```

```
[CV] END C=761.4316758498787, gamma=2.6126336514161914, kernel=linear; total t
ime=   8.2s
[CV] END C=761.4316758498787, gamma=2.6126336514161914, kernel=linear; total t
ime=   8.3s
[CV] END C=761.4316758498787, gamma=2.6126336514161914, kernel=linear; total t
ime=   7.3s
[CV] END C=97392.8188304179, gamma=0.09265545895311562, kernel=linear; total t
ime=  34.4s
[CV] END C=97392.8188304179, gamma=0.09265545895311562, kernel=linear; total t
ime=  37.7s
[CV] END C=97392.8188304179, gamma=0.09265545895311562, kernel=linear; total t
ime=  34.0s
[CV] END C=97392.8188304179, gamma=0.09265545895311562, kernel=linear; total t
ime=  37.2s
[CV] END C=97392.8188304179, gamma=0.09265545895311562, kernel=linear; total t
ime=  30.2s
[CV] END C=2423.0759984939154, gamma=3.248614270240346, kernel=linear; total t
ime=   8.8s
[CV] END C=2423.0759984939154, gamma=3.248614270240346, kernel=linear; total t
ime=   7.7s
[CV] END C=2423.0759984939154, gamma=3.248614270240346, kernel=linear; total t
ime=   8.8s
[CV] END C=2423.0759984939154, gamma=3.248614270240346, kernel=linear; total t
ime=   9.0s
[CV] END C=2423.0759984939154, gamma=3.248614270240346, kernel=linear; total t
ime=   7.9s
[CV] END C=717.3632997255093, gamma=0.3165604432088257, kernel=linear; total t
ime=   8.3s
[CV] END C=717.3632997255093, gamma=0.3165604432088257, kernel=linear; total t
ime=   8.3s
[CV] END C=717.3632997255093, gamma=0.3165604432088257, kernel=linear; total t
ime=   7.1s
[CV] END C=717.3632997255093, gamma=0.3165604432088257, kernel=linear; total t
ime=   8.3s
[CV] END C=717.3632997255093, gamma=0.3165604432088257, kernel=linear; total t
ime=   8.3s
```

```
[CV] END C=4446.66752118407, gamma=3.3597284456608496, kernel=rbf; total time=
13.0s
[CV] END C=4446.66752118407, gamma=3.3597284456608496, kernel=rbf; total time=
12.9s
[CV] END C=4446.66752118407, gamma=3.3597284456608496, kernel=rbf; total time=
13.0s
[CV] END C=4446.66752118407, gamma=3.3597284456608496, kernel=rbf; total time=
12.8s
[CV] END C=4446.66752118407, gamma=3.3597284456608496, kernel=rbf; total time=
12.9s
[CV] END C=2963.564121207816, gamma=0.15189814782062885, kernel=linear; total
time=   7.8s
[CV] END C=2963.564121207816, gamma=0.15189814782062885, kernel=linear; total
time=   9.4s
[CV] END C=2963.564121207816, gamma=0.15189814782062885, kernel=linear; total
time=   9.2s
[CV] END C=2963.564121207816, gamma=0.15189814782062885, kernel=linear; total
time=   8.8s
[CV] END C=2963.564121207816, gamma=0.15189814782062885, kernel=linear; total
time=   8.1s
[CV] END C=91.64267381686706, gamma=0.01575994483585621, kernel=linear; total
time=   8.2s
[CV] END C=91.64267381686706, gamma=0.01575994483585621, kernel=linear; total
time=   6.9s
[CV] END C=91.64267381686706, gamma=0.01575994483585621, kernel=linear; total
time=   8.1s
[CV] END C=91.64267381686706, gamma=0.01575994483585621, kernel=linear; total
time=   8.1s
[CV] END C=91.64267381686706, gamma=0.01575994483585621, kernel=linear; total
time=   6.9s
[CV] END C=24547.601975705937, gamma=0.22153944050588595, kernel=rbf; total ti
me=  11.8s
[CV] END C=24547.601975705937, gamma=0.22153944050588595, kernel=rbf; total ti
me=  11.8s
[CV] END C=24547.601975705937, gamma=0.22153944050588595, kernel=rbf; total ti
me=  11.8s
```

```
[CV] END C=24547.601975705937, gamma=0.22153944050588595, kernel=rbf; total ti
me=  11.8s
[CV] END C=24547.601975705937, gamma=0.22153944050588595, kernel=rbf; total ti
me=  11.9s
[CV] END C=22.76927941060928, gamma=0.22169760231351215, kernel=rbf; total tim
e=  11.6s
[CV] END C=22.76927941060928, gamma=0.22169760231351215, kernel=rbf; total tim
e=  11.6s
[CV] END C=22.76927941060928, gamma=0.22169760231351215, kernel=rbf; total tim
e=  11.7s
[CV] END C=22.76927941060928, gamma=0.22169760231351215, kernel=rbf; total tim
e=  11.7s
[CV] END C=22.76927941060928, gamma=0.22169760231351215, kernel=rbf; total tim
e=  11.2s
[CV] END C=16483.85052975289, gamma=1.4752145260435134, kernel=linear; total t
ime=  13.8s
[CV] END C=16483.85052975289, gamma=1.4752145260435134, kernel=linear; total t
ime=  12.9s
[CV] END C=16483.85052975289, gamma=1.4752145260435134, kernel=linear; total t
ime=  12.5s
[CV] END C=16483.85052975289, gamma=1.4752145260435134, kernel=linear; total t
ime=  13.7s
[CV] END C=16483.85052975289, gamma=1.4752145260435134, kernel=linear; total t
ime=  12.9s
[CV] END C=101445.66881340076, gamma=1.052904084582266, kernel=rbf; total time
=  54.5s
[CV] END C=101445.66881340076, gamma=1.052904084582266, kernel=rbf; total time
= 1.0min
[CV] END C=101445.66881340076, gamma=1.052904084582266, kernel=rbf; total time
=  50.6s
[CV] END C=101445.66881340076, gamma=1.052904084582266, kernel=rbf; total time
=  56.1s
[CV] END C=101445.66881340076, gamma=1.052904084582266, kernel=rbf; total time
= 1.1min
[CV] END C=56681.8085902955, gamma=0.9763011917123741, kernel=rbf; total time=
21.5s
```

```
[CV] END C=56681.8085902955, gamma=0.9763011917123741, kernel=rbf; total time=
23.6s
[CV] END C=56681.8085902955, gamma=0.9763011917123741, kernel=rbf; total time=
22.6s
[CV] END C=56681.8085902955, gamma=0.9763011917123741, kernel=rbf; total time=
26.7s
[CV] END C=56681.8085902955, gamma=0.9763011917123741, kernel=rbf; total time=
23.9s
[CV] END C=48.15822390928913, gamma=0.4633351167983427, kernel=rbf; total time
=  11.3s
[CV] END C=48.15822390928913, gamma=0.4633351167983427, kernel=rbf; total time
=  11.4s
[CV] END C=48.15822390928913, gamma=0.4633351167983427, kernel=rbf; total time
=  11.4s
[CV] END C=48.15822390928913, gamma=0.4633351167983427, kernel=rbf; total time
=  11.5s
[CV] END C=48.15822390928913, gamma=0.4633351167983427, kernel=rbf; total time
=  11.5s
[CV] END C=399.7268155705776, gamma=1.3078757839577408, kernel=rbf; total time
=  10.8s
[CV] END C=399.7268155705776, gamma=1.3078757839577408, kernel=rbf; total time
=  10.8s
[CV] END C=399.7268155705776, gamma=1.3078757839577408, kernel=rbf; total time
=  11.3s
[CV] END C=399.7268155705776, gamma=1.3078757839577408, kernel=rbf; total time
=  11.3s
[CV] END C=399.7268155705776, gamma=1.3078757839577408, kernel=rbf; total time
=  11.3s
[CV] END C=251.1407388628136, gamma=0.8238105204914145, kernel=linear; total t
ime=   8.2s
[CV] END C=251.1407388628136, gamma=0.8238105204914145, kernel=linear; total t
ime=   6.9s
[CV] END C=251.1407388628136, gamma=0.8238105204914145, kernel=linear; total t
ime=   8.1s
[CV] END C=251.1407388628136, gamma=0.8238105204914145, kernel=linear; total t
ime=   7.7s
```

```
[CV] END C=251.1407388628136, gamma=0.8238105204914145, kernel=linear; total t
ime=   7.7s
[CV] END C=60.17373642891686, gamma=1.2491263443165994, kernel=linear; total t
ime=   8.1s
[CV] END C=60.17373642891686, gamma=1.2491263443165994, kernel=linear; total t
ime=   6.9s
[CV] END C=60.17373642891686, gamma=1.2491263443165994, kernel=linear; total t
ime=   8.0s
[CV] END C=60.17373642891686, gamma=1.2491263443165994, kernel=linear; total t
ime=   8.0s
[CV] END C=60.17373642891686, gamma=1.2491263443165994, kernel=linear; total t
ime=   7.2s
[CV] END C=15415.161544891862, gamma=0.2691677514619319, kernel=rbf; total tim
e=  11.5s
[CV] END C=15415.161544891862, gamma=0.2691677514619319, kernel=rbf; total tim
e=  11.6s
[CV] END C=15415.161544891862, gamma=0.2691677514619319, kernel=rbf; total tim
e=  11.5s
[CV] END C=15415.161544891862, gamma=0.2691677514619319, kernel=rbf; total tim
e=  11.6s
[CV] END C=15415.161544891862, gamma=0.2691677514619319, kernel=rbf; total tim
e=  11.7s
[CV] END C=1888.9148509967115, gamma=0.739678838777267, kernel=linear; total t
ime=   7.5s
[CV] END C=1888.9148509967115, gamma=0.739678838777267, kernel=linear; total t
ime=   8.8s
[CV] END C=1888.9148509967115, gamma=0.739678838777267, kernel=linear; total t
ime=   8.5s
[CV] END C=1888.9148509967115, gamma=0.739678838777267, kernel=linear; total t
ime=   7.5s
[CV] END C=1888.9148509967115, gamma=0.739678838777267, kernel=linear; total t
ime=   8.7s
[CV] END C=55.53838911232771, gamma=0.578634378499143, kernel=linear; total ti
me=   8.3s
[CV] END C=55.53838911232771, gamma=0.578634378499143, kernel=linear; total ti
me=   7.0s
```

```
[CV] END C=55.53838911232771, gamma=0.578634378499143, kernel=linear; total ti
me=   8.1s
[CV] END C=55.53838911232771, gamma=0.578634378499143, kernel=linear; total ti
me=   7.1s
[CV] END C=55.53838911232771, gamma=0.578634378499143, kernel=linear; total ti
me=   8.2s
[CV] END C=26.714480823948186, gamma=1.0117295509275495, kernel=rbf; total tim
e=  11.3s
[CV] END C=26.714480823948186, gamma=1.0117295509275495, kernel=rbf; total tim
e=  11.3s
[CV] END C=26.714480823948186, gamma=1.0117295509275495, kernel=rbf; total tim
e=  11.3s
[CV] END C=26.714480823948186, gamma=1.0117295509275495, kernel=rbf; total tim
e=  11.4s
[CV] END C=26.714480823948186, gamma=1.0117295509275495, kernel=rbf; total tim
e=  10.9s
[CV] END C=3582.05527804896, gamma=1.189137022133257, kernel=linear; total ti
me=   8.7s
[CV] END C=3582.05527804896, gamma=1.189137022133257, kernel=linear; total ti
me=   9.2s
[CV] END C=3582.05527804896, gamma=1.189137022133257, kernel=linear; total ti
me=   9.0s
[CV] END C=3582.05527804896, gamma=1.189137022133257, kernel=linear; total ti
me=   9.1s
[CV] END C=3582.05527804896, gamma=1.189137022133257, kernel=linear; total ti
me=   9.4s
[CV] END C=198.70047818127367, gamma=0.5282819748826726, kernel=linear; total
time=   8.3s
[CV] END C=198.70047818127367, gamma=0.5282819748826726, kernel=linear; total
time=   7.0s
[CV] END C=198.70047818127367, gamma=0.5282819748826726, kernel=linear; total
time=   8.2s
[CV] END C=198.70047818127367, gamma=0.5282819748826726, kernel=linear; total
time=   8.3s
[CV] END C=198.70047818127367, gamma=0.5282819748826726, kernel=linear; total
time=   7.1s
```

```
[CV] END C=129.80006041433077, gamma=2.8621383676481322, kernel=linear; total
time=   8.2s
[CV] END C=129.80006041433077, gamma=2.8621383676481322, kernel=linear; total
time=   6.9s
[CV] END C=129.80006041433077, gamma=2.8621383676481322, kernel=linear; total
time=   8.1s
[CV] END C=129.80006041433077, gamma=2.8621383676481322, kernel=linear; total
time=   8.1s
[CV] END C=129.80006041433077, gamma=2.8621383676481322, kernel=linear; total
time=   7.1s
[CV] END C=288.4269299593897, gamma=0.17580835850006285, kernel=rbf; total tim
e=  11.5s
[CV] END C=288.4269299593897, gamma=0.17580835850006285, kernel=rbf; total tim
e=  11.5s
[CV] END C=288.4269299593897, gamma=0.17580835850006285, kernel=rbf; total tim
e=  11.4s
[CV] END C=288.4269299593897, gamma=0.17580835850006285, kernel=rbf; total tim
e=  11.4s
[CV] END C=288.4269299593897, gamma=0.17580835850006285, kernel=rbf; total tim
e=  11.4s
[CV] END C=6287.039489427173, gamma=0.3504567255332862, kernel=linear; total t
ime=   9.9s
[CV] END C=6287.039489427173, gamma=0.3504567255332862, kernel=linear; total t
ime=   9.5s
[CV] END C=6287.039489427173, gamma=0.3504567255332862, kernel=linear; total t
ime=   9.6s
[CV] END C=6287.039489427173, gamma=0.3504567255332862, kernel=linear; total t
ime=  10.2s
[CV] END C=6287.039489427173, gamma=0.3504567255332862, kernel=linear; total t
ime=   9.9s
[CV] END C=61217.04421344495, gamma=1.6279689407405564, kernel=rbf; total time
=  40.7s
[CV] END C=61217.04421344495, gamma=1.6279689407405564, kernel=rbf; total time
=  48.3s
[CV] END C=61217.04421344495, gamma=1.6279689407405564, kernel=rbf; total time
=  33.6s
```

```
[CV] END C=61217.04421344495, gamma=1.6279689407405564, kernel=rbf; total time
=  35.7s
[CV] END C=61217.04421344495, gamma=1.6279689407405564, kernel=rbf; total time
=  43.6s
[CV] END C=926.9787684096652, gamma=2.147979593060577, kernel=rbf; total time=
11.8s
[CV] END C=926.9787684096652, gamma=2.147979593060577, kernel=rbf; total time=
11.2s
[CV] END C=926.9787684096652, gamma=2.147979593060577, kernel=rbf; total time=
11.2s
[CV] END C=926.9787684096652, gamma=2.147979593060577, kernel=rbf; total time=
11.7s
[CV] END C=926.9787684096652, gamma=2.147979593060577, kernel=rbf; total time=
11.7s
[CV] END C=33946.15706493403, gamma=2.2642426492862313, kernel=linear; total t
ime=  18.1s
[CV] END C=33946.15706493403, gamma=2.2642426492862313, kernel=linear; total t
ime=  18.5s
[CV] END C=33946.15706493403, gamma=2.2642426492862313, kernel=linear; total t
ime=  18.1s
[CV] END C=33946.15706493403, gamma=2.2642426492862313, kernel=linear; total t
ime=  18.1s
[CV] END C=33946.15706493403, gamma=2.2642426492862313, kernel=linear; total t
ime=  20.3s
[CV] END C=84789.82947739528, gamma=0.3176359085304841, kernel=linear; total t
ime=  31.1s
[CV] END C=84789.82947739528, gamma=0.3176359085304841, kernel=linear; total t
ime=  34.6s
[CV] END C=84789.82947739528, gamma=0.3176359085304841, kernel=linear; total t
ime=  29.4s
[CV] END C=84789.82947739528, gamma=0.3176359085304841, kernel=linear; total t
ime=  34.3s
[CV] END C=84789.82947739528, gamma=0.3176359085304841, kernel=linear; total t
ime=  35.7s
```

Out[128]:

▸ **RandomizedSearchCV**

(https://scikit-learn.org/1.6/modules/generated/sklearn.model_select

▸ **best_estimator_: SVR**

▸ SVR

(https://scikit-learn.org/1.6/modules/generated/sklearn.svm.SVR.html)

The best model achieves the following score (evaluated using 5-fold cross validation):

In [129]:

```
negative_mse = rnd_search.best_score_
rmse = np.sqrt(-negative_mse)
rmse
```

Out[129]:

```
np.float64(54751.69009488048)
```

Now this is much closer to the performance of the `RandomForestRegressor` (but not quite there yet). Let's check the best hyperparameters found:

In [130]:

```python
rnd_search.best_params_
```

Out[130]:

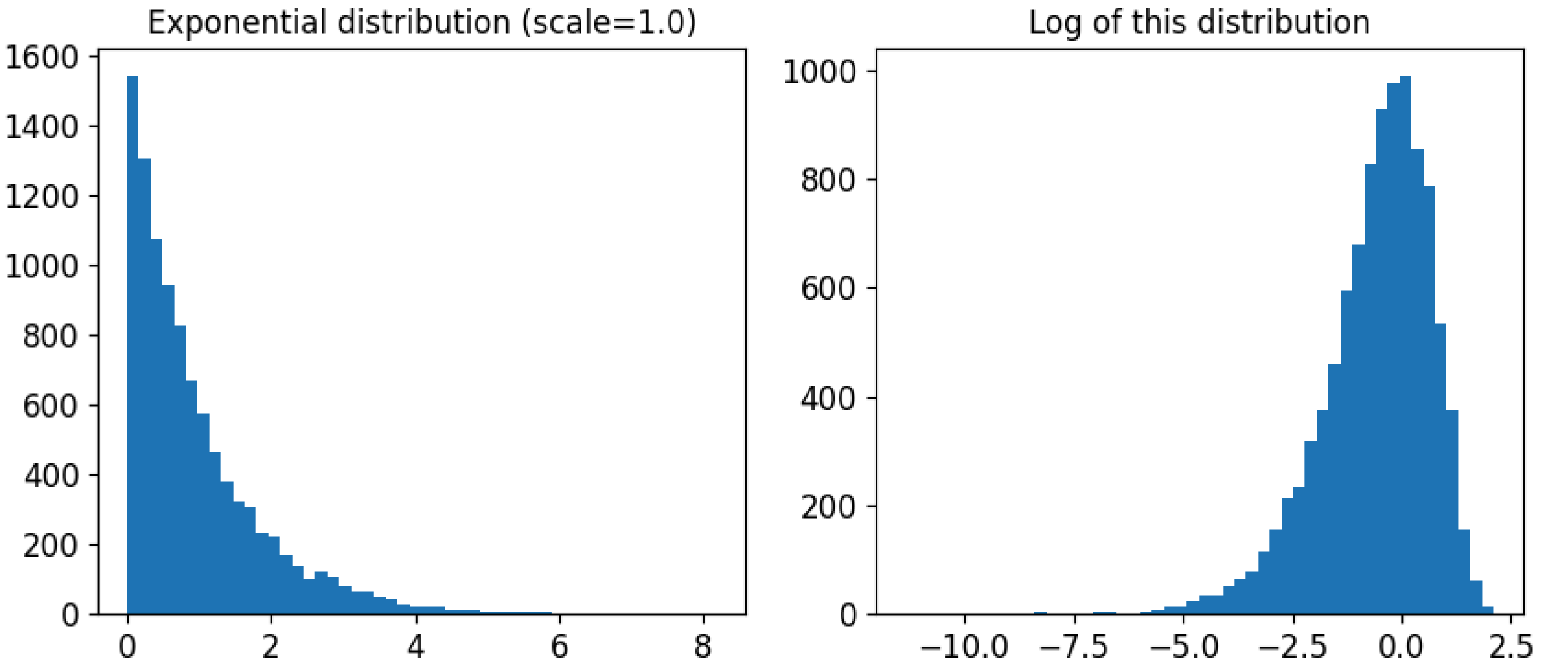
```
{'C': np.float64(157055.10989448498),
 'gamma': np.float64(0.26497040005002437),
 'kernel': 'rbf'}
```

This time the search found a good set of hyperparameters for the RBF kernel. Randomized search tends to find better hyperparameters than grid search in the same amount of time.

Let's look at the exponential distribution we used, with `scale=1.0`. Note that some samples are much larger or smaller than 1.0, but when you look at the log of the distribution, you can see that most values are actually concentrated roughly in the range of exp(-2) to exp(+2), which is about 0.1 to 7.4.

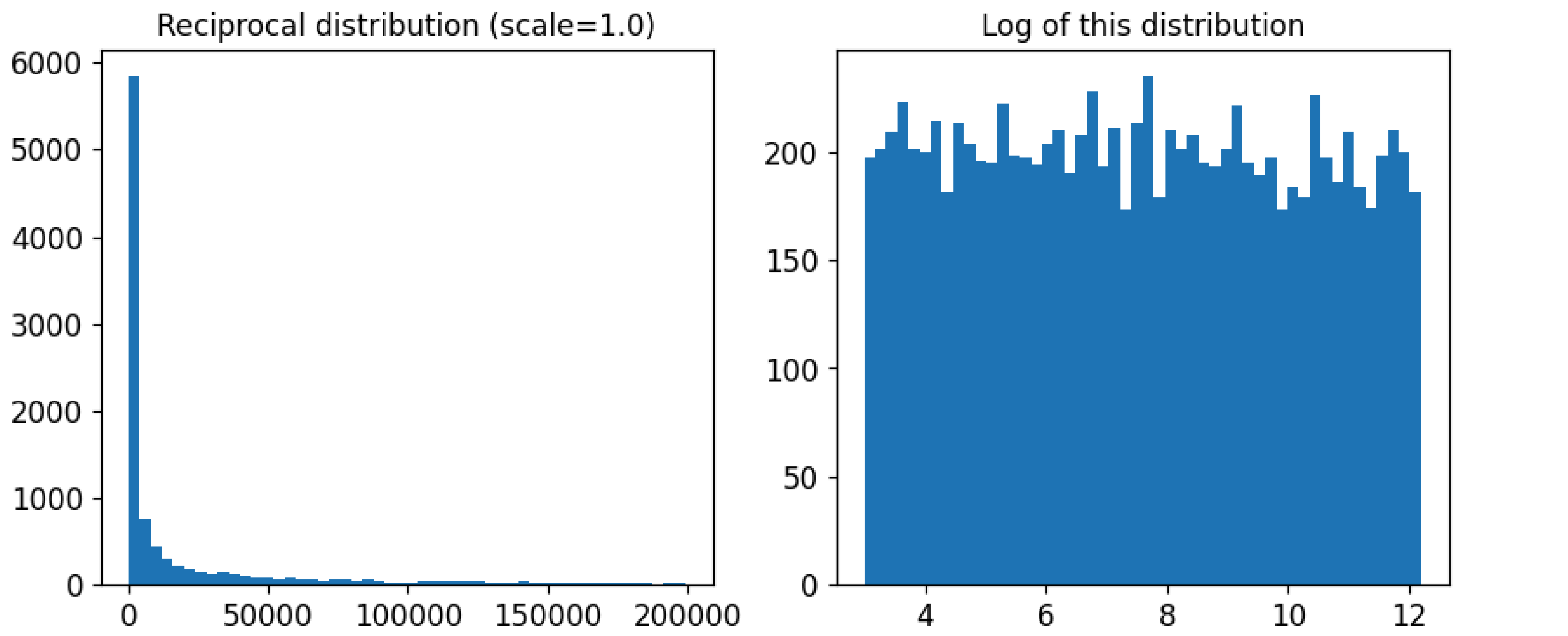
In [131]:

```
expon_distrib = expon(scale=1.)
samples = expon_distrib.rvs(10000, random_state=42)
plt.figure(figsize=(10, 4))
plt.subplot(121)
plt.title("Exponential distribution (scale=1.0)")
plt.hist(samples, bins=50)
plt.subplot(122)
plt.title("Log of this distribution")
plt.hist(np.log(samples), bins=50)
plt.show()
```



The distribution we used for `C` looks quite different: the scale of the samples is picked from a uniform distribution within a given range, which is why the right graph, which represents the log of the samples, looks roughly constant. This distribution is useful when you don't have a clue of what the target scale is:

In [132]:
```
reciprocal_distrib = reciprocal(20, 200000)
samples = reciprocal_distrib.rvs(10000, random_state=42)
plt.figure(figsize=(10, 4))
plt.subplot(121)
plt.title("Reciprocal distribution (scale=1.0)")
plt.hist(samples, bins=50)
plt.subplot(122)
plt.title("Log of this distribution")
plt.hist(np.log(samples), bins=50)
plt.show()
```



The reciprocal distribution is useful when you have no idea what the scale of the hyperparameter should be (indeed, as you can see on the figure on the right, all scales are equally likely, within the given range), whereas the exponential distribution is best when you know (more or less) what the scale of the hyperparameter should be.

3.

Question: Try adding a transformer in the preparation pipeline to select only the most important attributes.

In [133]:
```
from sklearn.base import BaseEstimator, TransformerMixin

def indices_of_top_k(arr, k):
    return np.sort(np.argpartition(np.array(arr), -k)[-k:])

class TopFeatureSelector(BaseEstimator, TransformerMixin):
    def __init__(self, feature_importances, k):
        self.feature_importances = feature_importances
        self.k = k
    def fit(self, X, y=None):
        self.feature_indices_ = indices_of_top_k(self.feature_importances, self.
k)
        return self
    def transform(self, X):
        return X[:, self.feature_indices_]
```

Note: this feature selector assumes that you have already computed the feature importances somehow (for example using a `RandomForestRegressor`). You may be tempted to compute them directly in the `TopFeatureSelector`'s `fit()` method, however this would likely slow down grid/randomized search since the feature importances would have to be computed for every hyperparameter combination (unless you implement some sort of cache).

Let's define the number of top features we want to keep:

In [134]:
```
k = 5
```

Now let's look for the indices of the top k features:

In [135]:
```
top_k_feature_indices = indices_of_top_k(feature_importances, k)
top_k_feature_indices
```

Out[135]:
```
array([ 0,  1,  7,  9, 12])
```

In [136]:
```
np.array(attributes)[top_k_feature_indices]
```

Out[136]:
```
array(['longitude', 'latitude', 'median_income', 'pop_per_hhold',
       'INLAND'], dtype='<U18')
```

Let's double check that these are indeed the top k features:

In [137]:
```
sorted(zip(feature_importances, attributes), reverse=True)[:k]
```

Out[137]:
```
[(np.float64(0.3790092248170967), 'median_income'),
 (np.float64(0.16570630316895876), 'INLAND'),
 (np.float64(0.10703132208204354), 'pop_per_hhold'),
 (np.float64(0.06965425227942929), 'longitude'),
 (np.float64(0.0604213840080722), 'latitude')]
```

Looking good... Now let's create a new pipeline that runs the previously defined preparation pipeline, and adds top k feature selection:

In [138]:
```
preparation_and_feature_selection_pipeline = Pipeline([
    ('preparation', full_pipeline),
    ('feature_selection', TopFeatureSelector(feature_importances, k))
])
```

In [139]:
```
housing_prepared_top_k_features = preparation_and_feature_selection_pipeline.fit
_transform(housing)
```

Let's look at the features of the first 3 instances:

In [140]:
```python
housing_prepared_top_k_features[0:3]
```

Out[140]:
```
array([[-0.94135046,  1.34743822, -0.8936472 ,  0.00622264,  1.        ],
       [ 1.17178212, -1.19243966,  1.292168  , -0.04081077,  0.        ],
       [ 0.26758118, -0.1259716 , -0.52543365, -0.07537122,  1.        ]])
```

Now let's double check that these are indeed the top k features:

In [141]:
```python
housing_prepared[0:3, top_k_feature_indices]
```

Out[141]:
```
array([[-0.94135046,  1.34743822, -0.8936472 ,  0.00622264,  1.        ],
       [ 1.17178212, -1.19243966,  1.292168  , -0.04081077,  0.        ],
       [ 0.26758118, -0.1259716 , -0.52543365, -0.07537122,  1.        ]])
```

Works great! :)

4.

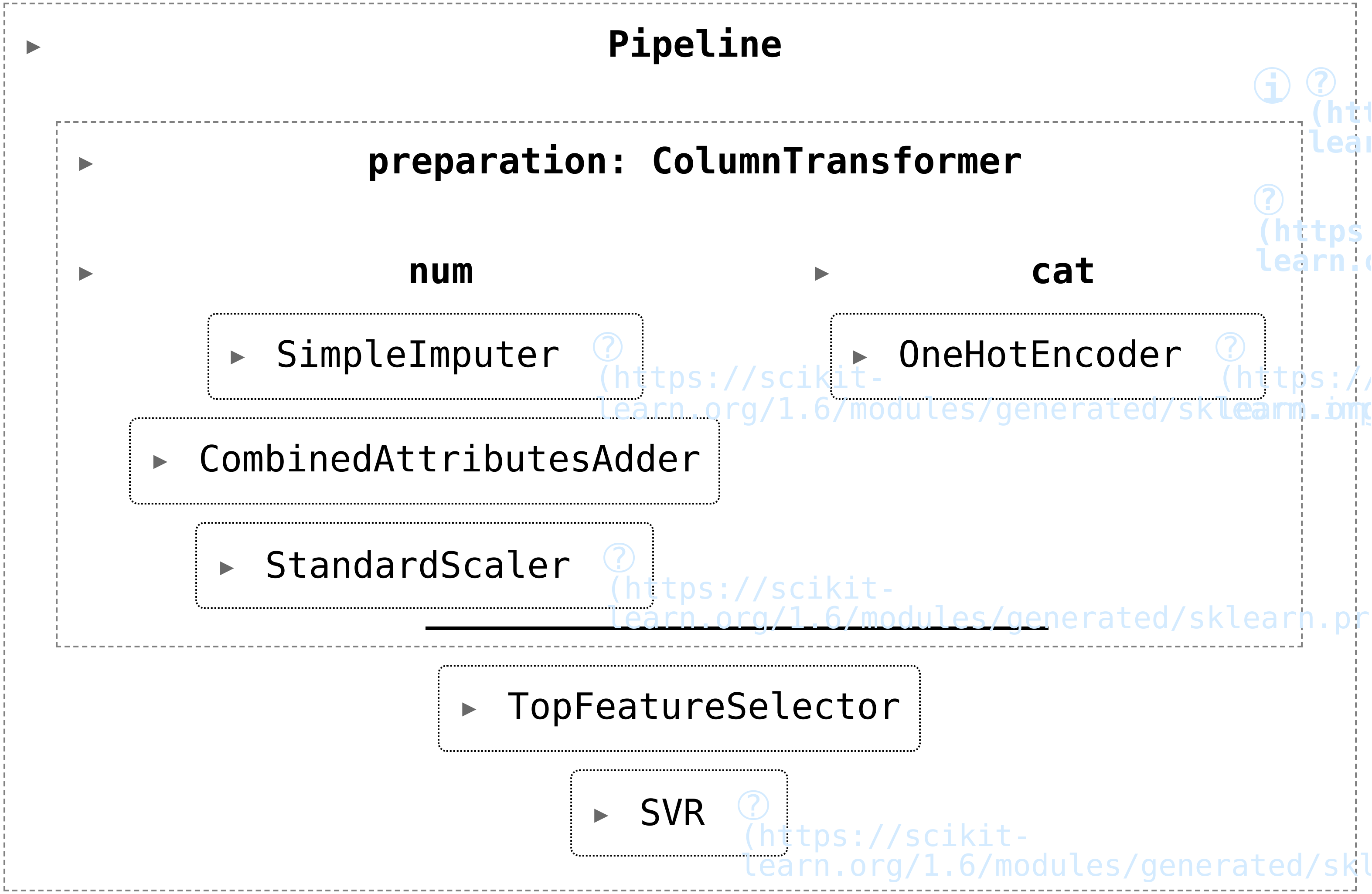
Question: Try creating a single pipeline that does the full data preparation plus the final prediction.

In [142]:

```
prepare_select_and_predict_pipeline = Pipeline([
    ('preparation', full_pipeline),
    ('feature_selection', TopFeatureSelector(feature_importances, k)),
    ('svm_reg', SVR(**rnd_search.best_params_))
])
```

In [143]:
```
prepare_select_and_predict_pipeline.fit(housing, housing_labels)
```

Out[143]:



Let's try the full pipeline on a few instances:

In [144]:
```python
some_data = housing.iloc[:4]
some_labels = housing_labels.iloc[:4]

print("Predictions:\t", prepare_select_and_predict_pipeline.predict(some_data))
print("Labels:\t\t", list(some_labels))
```

```
Predictions:     [ 83384.49158095 299407.90439234  92272.03345144 150173.16199
041]
Labels:          [72100.0, 279600.0, 82700.0, 112500.0]
```

Well, the full pipeline seems to work fine. Of course, the predictions are not fantastic: they would be better if we used the best `RandomForestRegressor` that we found earlier, rather than the best `SVR`.

5.

Question: Automatically explore some preparation options using `GridSearchCV`.

Warning: the following cell may take close to 45 minutes to run, or more depending on your hardware.

Note: In the code below, I've set the `OneHotEncoder`'s `handle_unknown` hyperparameter to `'ignore'`, to avoid warnings during training. Without this, the `OneHotEncoder` would default to `handle_unknown='error'`, meaning that it would raise an error when transforming any data containing a category it didn't see during training. If we kept the default, then the `GridSearchCV` would run into errors during training when evaluating the folds in which not all the categories are in the training set. This is likely to happen since there's only one sample in the `'ISLAND'` category, and it may end up in the test set in some of the folds. So some folds would just be dropped by the `GridSearchCV`, and it's best to avoid that.

In [145]:

```
full_pipeline.named_transformers_["cat"].handle_unknown = 'ignore'

param_grid = [{
    'preparation__num__imputer__strategy': ['mean', 'median', 'most_frequent'],
    'feature_selection__k': list(range(1, len(feature_importances) + 1))
}]

grid_search_prep = GridSearchCV(prepare_select_and_predict_pipeline, param_grid,
cv=5,
                                scoring='neg_mean_squared_error', verbose=2)
grid_search_prep.fit(housing, housing_labels)
```

```
Fitting 5 folds for each of 48 candidates, totalling 240 fits
```

```
/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_validation.p
y:960: UserWarning: Scoring failed. The score on this train-test partition for
these parameters will be set to nan. Details:
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_valid
ation.py", line 949, in _score
    scores = scorer(estimator, X_test, y_test, **score_params)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 288, in __call__
    return self._score(partial(_cached_call, None), estimator, X, y_true, **_k
wargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 380, in _score
    y_pred = method_caller(
             ^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 90, in _cached_call
    result, _ = _get_response_values(
                ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_response.py", l
ine 242, in _get_response_values
    y_pred, pos_label = prediction_method(X), None
                        ^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 78
7, in predict
    Xt = transform.transform(Xt)
         ^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
```

```
ormer.py", line 1101, in transform
    Xs = self._call_func_on_transformers(
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
ormer.py", line 910, in _call_func_on_transformers
    return Parallel(n_jobs=self.n_jobs)(jobs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 77, in __call__
    return super().__call__(iterable_with_config)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 198
6, in __call__
    return output if self.return_generator else list(output)
                                                ^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 191
4, in _get_sequential_output
    res = func(*args, **kwargs)
          ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 139, in __call__
    return self.function(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 153
1, in _transform_one
    res = transformer.transform(X, **params.transform)
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
s.py", line 1043, in transform
    X_int, X_mask = self._transform(
                    ^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
```

```
s.py", line 218, in _transform
    raise ValueError(msg)
ValueError: Found unknown categories ['ISLAND'] in column 0 during transform

  warnings.warn(

[CV] END feature_selection__k=1, preparation__num__imputer__strategy=mean; tot
al time=   7.4s
[CV] END feature_selection__k=1, preparation__num__imputer__strategy=mean; tot
al time=  10.0s
[CV] END feature_selection__k=1, preparation__num__imputer__strategy=mean; tot
al time=  10.0s
[CV] END feature_selection__k=1, preparation__num__imputer__strategy=mean; tot
al time=  10.1s
[CV] END feature_selection__k=1, preparation__num__imputer__strategy=mean; tot
al time=   8.9s
```

```
/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_validation.p
y:960: UserWarning: Scoring failed. The score on this train-test partition for
these parameters will be set to nan. Details:
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_valid
ation.py", line 949, in _score
    scores = scorer(estimator, X_test, y_test, **score_params)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 288, in __call__
    return self._score(partial(_cached_call, None), estimator, X, y_true, **_k
wargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 380, in _score
    y_pred = method_caller(
             ^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 90, in _cached_call
    result, _ = _get_response_values(
                ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_response.py", l
ine 242, in _get_response_values
    y_pred, pos_label = prediction_method(X), None
                        ^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 78
7, in predict
    Xt = transform.transform(Xt)
         ^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
```

```
ormer.py", line 1101, in transform
    Xs = self._call_func_on_transformers(
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
ormer.py", line 910, in _call_func_on_transformers
    return Parallel(n_jobs=self.n_jobs)(jobs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 77, in __call__
    return super().__call__(iterable_with_config)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 198
6, in __call__
    return output if self.return_generator else list(output)
                                                ^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 191
4, in _get_sequential_output
    res = func(*args, **kwargs)
          ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 139, in __call__
    return self.function(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 153
1, in _transform_one
    res = transformer.transform(X, **params.transform)
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
s.py", line 1043, in transform
    X_int, X_mask = self._transform(
                    ^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
```

```
s.py", line 218, in _transform
    raise ValueError(msg)
ValueError: Found unknown categories ['ISLAND'] in column 0 during transform

  warnings.warn(

[CV] END feature_selection__k=1, preparation__num__imputer__strategy=median; t
otal time=   8.6s
[CV] END feature_selection__k=1, preparation__num__imputer__strategy=median; t
otal time=  10.0s
[CV] END feature_selection__k=1, preparation__num__imputer__strategy=median; t
otal time=  10.1s
[CV] END feature_selection__k=1, preparation__num__imputer__strategy=median; t
otal time=   8.9s
[CV] END feature_selection__k=1, preparation__num__imputer__strategy=median; t
otal time=  10.1s
```

```
/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_validation.p
y:960: UserWarning: Scoring failed. The score on this train-test partition for
these parameters will be set to nan. Details:
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_valid
ation.py", line 949, in _score
    scores = scorer(estimator, X_test, y_test, **score_params)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 288, in __call__
    return self._score(partial(_cached_call, None), estimator, X, y_true, **_k
wargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 380, in _score
    y_pred = method_caller(
             ^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 90, in _cached_call
    result, _ = _get_response_values(
                ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_response.py", l
ine 242, in _get_response_values
    y_pred, pos_label = prediction_method(X), None
                        ^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 78
7, in predict
    Xt = transform.transform(Xt)
         ^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
```

```
ormer.py", line 1101, in transform
    Xs = self._call_func_on_transformers(
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
ormer.py", line 910, in _call_func_on_transformers
    return Parallel(n_jobs=self.n_jobs)(jobs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 77, in __call__
    return super().__call__(iterable_with_config)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 198
6, in __call__
    return output if self.return_generator else list(output)
                                                ^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 191
4, in _get_sequential_output
    res = func(*args, **kwargs)
          ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 139, in __call__
    return self.function(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 153
1, in _transform_one
    res = transformer.transform(X, **params.transform)
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
s.py", line 1043, in transform
    X_int, X_mask = self._transform(
                    ^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
```

```
s.py", line 218, in _transform
    raise ValueError(msg)
ValueError: Found unknown categories ['ISLAND'] in column 0 during transform

  warnings.warn(

[CV] END feature_selection__k=1, preparation__num__imputer__strategy=most_freq
uent; total time=   8.5s
[CV] END feature_selection__k=1, preparation__num__imputer__strategy=most_freq
uent; total time=  10.1s
[CV] END feature_selection__k=1, preparation__num__imputer__strategy=most_freq
uent; total time=   8.9s
[CV] END feature_selection__k=1, preparation__num__imputer__strategy=most_freq
uent; total time=  10.1s
[CV] END feature_selection__k=1, preparation__num__imputer__strategy=most_freq
uent; total time=  10.2s
```

```
/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_validation.p
y:960: UserWarning: Scoring failed. The score on this train-test partition for
these parameters will be set to nan. Details:
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_valid
ation.py", line 949, in _score
    scores = scorer(estimator, X_test, y_test, **score_params)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 288, in __call__
    return self._score(partial(_cached_call, None), estimator, X, y_true, **_k
wargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 380, in _score
    y_pred = method_caller(
             ^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 90, in _cached_call
    result, _ = _get_response_values(
                ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_response.py", l
ine 242, in _get_response_values
    y_pred, pos_label = prediction_method(X), None
                        ^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 78
7, in predict
    Xt = transform.transform(Xt)
         ^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
```

```
ormer.py", line 1101, in transform
    Xs = self._call_func_on_transformers(
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
ormer.py", line 910, in _call_func_on_transformers
    return Parallel(n_jobs=self.n_jobs)(jobs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 77, in __call__
    return super().__call__(iterable_with_config)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 198
6, in __call__
    return output if self.return_generator else list(output)
                                                ^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 191
4, in _get_sequential_output
    res = func(*args, **kwargs)
          ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 139, in __call__
    return self.function(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 153
1, in _transform_one
    res = transformer.transform(X, **params.transform)
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
s.py", line 1043, in transform
    X_int, X_mask = self._transform(
                    ^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
```

```
s.py", line 218, in _transform
    raise ValueError(msg)
ValueError: Found unknown categories ['ISLAND'] in column 0 during transform

  warnings.warn(

[CV] END feature_selection__k=2, preparation__num__imputer__strategy=mean; tot
al time=   8.7s
[CV] END feature_selection__k=2, preparation__num__imputer__strategy=mean; tot
al time=   9.8s
[CV] END feature_selection__k=2, preparation__num__imputer__strategy=mean; tot
al time=  10.4s
[CV] END feature_selection__k=2, preparation__num__imputer__strategy=mean; tot
al time=  10.8s
[CV] END feature_selection__k=2, preparation__num__imputer__strategy=mean; tot
al time=  10.7s
```

```
/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_validation.p
y:960: UserWarning: Scoring failed. The score on this train-test partition for
these parameters will be set to nan. Details:
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_valid
ation.py", line 949, in _score
    scores = scorer(estimator, X_test, y_test, **score_params)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 288, in __call__
    return self._score(partial(_cached_call, None), estimator, X, y_true, **_k
wargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 380, in _score
    y_pred = method_caller(
             ^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 90, in _cached_call
    result, _ = _get_response_values(
                ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_response.py", l
ine 242, in _get_response_values
    y_pred, pos_label = prediction_method(X), None
                        ^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 78
7, in predict
    Xt = transform.transform(Xt)
         ^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
```

```
ormer.py", line 1101, in transform
    Xs = self._call_func_on_transformers(
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
ormer.py", line 910, in _call_func_on_transformers
    return Parallel(n_jobs=self.n_jobs)(jobs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 77, in __call__
    return super().__call__(iterable_with_config)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 198
6, in __call__
    return output if self.return_generator else list(output)
                                                ^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 191
4, in _get_sequential_output
    res = func(*args, **kwargs)
          ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 139, in __call__
    return self.function(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 153
1, in _transform_one
    res = transformer.transform(X, **params.transform)
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
s.py", line 1043, in transform
    X_int, X_mask = self._transform(
                    ^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
```

```
s.py", line 218, in _transform
    raise ValueError(msg)
ValueError: Found unknown categories ['ISLAND'] in column 0 during transform

  warnings.warn(

[CV] END feature_selection__k=2, preparation__num__imputer__strategy=median; t
otal time=   8.7s
[CV] END feature_selection__k=2, preparation__num__imputer__strategy=median; t
otal time=   9.3s
[CV] END feature_selection__k=2, preparation__num__imputer__strategy=median; t
otal time=  10.5s
[CV] END feature_selection__k=2, preparation__num__imputer__strategy=median; t
otal time=  10.5s
[CV] END feature_selection__k=2, preparation__num__imputer__strategy=median; t
otal time=  10.5s
```

```
/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_validation.p
y:960: UserWarning: Scoring failed. The score on this train-test partition for
these parameters will be set to nan. Details:
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_valid
ation.py", line 949, in _score
    scores = scorer(estimator, X_test, y_test, **score_params)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 288, in __call__
    return self._score(partial(_cached_call, None), estimator, X, y_true, **_k
wargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 380, in _score
    y_pred = method_caller(
             ^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 90, in _cached_call
    result, _ = _get_response_values(
                ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_response.py", l
ine 242, in _get_response_values
    y_pred, pos_label = prediction_method(X), None
                        ^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 78
7, in predict
    Xt = transform.transform(Xt)
         ^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
```

```
ormer.py", line 1101, in transform
    Xs = self._call_func_on_transformers(
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
ormer.py", line 910, in _call_func_on_transformers
    return Parallel(n_jobs=self.n_jobs)(jobs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 77, in __call__
    return super().__call__(iterable_with_config)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 198
6, in __call__
    return output if self.return_generator else list(output)
                                                ^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 191
4, in _get_sequential_output
    res = func(*args, **kwargs)
          ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 139, in __call__
    return self.function(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 153
1, in _transform_one
    res = transformer.transform(X, **params.transform)
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
s.py", line 1043, in transform
    X_int, X_mask = self._transform(
                    ^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
```

```
s.py", line 218, in _transform
    raise ValueError(msg)
ValueError: Found unknown categories ['ISLAND'] in column 0 during transform

  warnings.warn(

[CV] END feature_selection__k=2, preparation__num__imputer__strategy=most_freq
uent; total time=   8.4s
[CV] END feature_selection__k=2, preparation__num__imputer__strategy=most_freq
uent; total time=   9.7s
[CV] END feature_selection__k=2, preparation__num__imputer__strategy=most_freq
uent; total time=  10.5s
[CV] END feature_selection__k=2, preparation__num__imputer__strategy=most_freq
uent; total time=  10.5s
[CV] END feature_selection__k=2, preparation__num__imputer__strategy=most_freq
uent; total time=  10.5s
```

```
/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_validation.p
y:960: UserWarning: Scoring failed. The score on this train-test partition for
these parameters will be set to nan. Details:
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_valid
ation.py", line 949, in _score
    scores = scorer(estimator, X_test, y_test, **score_params)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 288, in __call__
    return self._score(partial(_cached_call, None), estimator, X, y_true, **_k
wargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 380, in _score
    y_pred = method_caller(
             ^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 90, in _cached_call
    result, _ = _get_response_values(
                ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_response.py", l
ine 242, in _get_response_values
    y_pred, pos_label = prediction_method(X), None
                        ^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 78
7, in predict
    Xt = transform.transform(Xt)
         ^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
```

```
ormer.py", line 1101, in transform
    Xs = self._call_func_on_transformers(
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
ormer.py", line 910, in _call_func_on_transformers
    return Parallel(n_jobs=self.n_jobs)(jobs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 77, in __call__
    return super().__call__(iterable_with_config)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 198
6, in __call__
    return output if self.return_generator else list(output)
                                                ^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 191
4, in _get_sequential_output
    res = func(*args, **kwargs)
          ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 139, in __call__
    return self.function(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 153
1, in _transform_one
    res = transformer.transform(X, **params.transform)
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
s.py", line 1043, in transform
    X_int, X_mask = self._transform(
                    ^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
```

```
s.py", line 218, in _transform
    raise ValueError(msg)
ValueError: Found unknown categories ['ISLAND'] in column 0 during transform

  warnings.warn(

[CV] END feature_selection__k=3, preparation__num__imputer__strategy=mean; tot
al time=   7.7s
[CV] END feature_selection__k=3, preparation__num__imputer__strategy=mean; tot
al time=  10.6s
[CV] END feature_selection__k=3, preparation__num__imputer__strategy=mean; tot
al time=  10.7s
[CV] END feature_selection__k=3, preparation__num__imputer__strategy=mean; tot
al time=  10.6s
[CV] END feature_selection__k=3, preparation__num__imputer__strategy=mean; tot
al time=  10.7s
```

```
/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_validation.p
y:960: UserWarning: Scoring failed. The score on this train-test partition for
these parameters will be set to nan. Details:
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_valid
ation.py", line 949, in _score
    scores = scorer(estimator, X_test, y_test, **score_params)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 288, in __call__
    return self._score(partial(_cached_call, None), estimator, X, y_true, **_k
wargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 380, in _score
    y_pred = method_caller(
             ^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 90, in _cached_call
    result, _ = _get_response_values(
                ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_response.py", l
ine 242, in _get_response_values
    y_pred, pos_label = prediction_method(X), None
                        ^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 78
7, in predict
    Xt = transform.transform(Xt)
         ^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
```

```
ormer.py", line 1101, in transform
    Xs = self._call_func_on_transformers(
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
ormer.py", line 910, in _call_func_on_transformers
    return Parallel(n_jobs=self.n_jobs)(jobs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 77, in __call__
    return super().__call__(iterable_with_config)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 198
6, in __call__
    return output if self.return_generator else list(output)
                                                ^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 191
4, in _get_sequential_output
    res = func(*args, **kwargs)
          ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 139, in __call__
    return self.function(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 153
1, in _transform_one
    res = transformer.transform(X, **params.transform)
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
s.py", line 1043, in transform
    X_int, X_mask = self._transform(
                    ^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
```

```
s.py", line 218, in _transform
    raise ValueError(msg)
ValueError: Found unknown categories ['ISLAND'] in column 0 during transform

  warnings.warn(

[CV] END feature_selection__k=3, preparation__num__imputer__strategy=median; t
otal time=   7.7s
[CV] END feature_selection__k=3, preparation__num__imputer__strategy=median; t
otal time=  10.6s
[CV] END feature_selection__k=3, preparation__num__imputer__strategy=median; t
otal time=  10.6s
[CV] END feature_selection__k=3, preparation__num__imputer__strategy=median; t
otal time=  10.6s
[CV] END feature_selection__k=3, preparation__num__imputer__strategy=median; t
otal time=  10.6s
```

```
/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_validation.p
y:960: UserWarning: Scoring failed. The score on this train-test partition for
these parameters will be set to nan. Details:
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_valid
ation.py", line 949, in _score
    scores = scorer(estimator, X_test, y_test, **score_params)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 288, in __call__
    return self._score(partial(_cached_call, None), estimator, X, y_true, **_k
wargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 380, in _score
    y_pred = method_caller(
             ^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 90, in _cached_call
    result, _ = _get_response_values(
                ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_response.py", l
ine 242, in _get_response_values
    y_pred, pos_label = prediction_method(X), None
                        ^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 78
7, in predict
    Xt = transform.transform(Xt)
         ^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
```

```
ormer.py", line 1101, in transform
    Xs = self._call_func_on_transformers(
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
ormer.py", line 910, in _call_func_on_transformers
    return Parallel(n_jobs=self.n_jobs)(jobs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 77, in __call__
    return super().__call__(iterable_with_config)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 198
6, in __call__
    return output if self.return_generator else list(output)
                                                ^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 191
4, in _get_sequential_output
    res = func(*args, **kwargs)
          ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 139, in __call__
    return self.function(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 153
1, in _transform_one
    res = transformer.transform(X, **params.transform)
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
s.py", line 1043, in transform
    X_int, X_mask = self._transform(
                    ^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
```

```
s.py", line 218, in _transform
    raise ValueError(msg)
ValueError: Found unknown categories ['ISLAND'] in column 0 during transform

  warnings.warn(

[CV] END feature_selection__k=3, preparation__num__imputer__strategy=most_freq
uent; total time=   7.7s
[CV] END feature_selection__k=3, preparation__num__imputer__strategy=most_freq
uent; total time=  10.8s
[CV] END feature_selection__k=3, preparation__num__imputer__strategy=most_freq
uent; total time=  10.7s
[CV] END feature_selection__k=3, preparation__num__imputer__strategy=most_freq
uent; total time=  10.6s
[CV] END feature_selection__k=3, preparation__num__imputer__strategy=most_freq
uent; total time=  10.6s
```

```
/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_validation.p
y:960: UserWarning: Scoring failed. The score on this train-test partition for
these parameters will be set to nan. Details:
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_valid
ation.py", line 949, in _score
    scores = scorer(estimator, X_test, y_test, **score_params)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 288, in __call__
    return self._score(partial(_cached_call, None), estimator, X, y_true, **_k
wargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 380, in _score
    y_pred = method_caller(
             ^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 90, in _cached_call
    result, _ = _get_response_values(
                ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_response.py", l
ine 242, in _get_response_values
    y_pred, pos_label = prediction_method(X), None
                        ^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 78
7, in predict
    Xt = transform.transform(Xt)
         ^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
```

```
ormer.py", line 1101, in transform
    Xs = self._call_func_on_transformers(
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
ormer.py", line 910, in _call_func_on_transformers
    return Parallel(n_jobs=self.n_jobs)(jobs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 77, in __call__
    return super().__call__(iterable_with_config)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 198
6, in __call__
    return output if self.return_generator else list(output)
                                                ^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 191
4, in _get_sequential_output
    res = func(*args, **kwargs)
          ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 139, in __call__
    return self.function(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 153
1, in _transform_one
    res = transformer.transform(X, **params.transform)
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
s.py", line 1043, in transform
    X_int, X_mask = self._transform(
                    ^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
```

```
s.py", line 218, in _transform
    raise ValueError(msg)
ValueError: Found unknown categories ['ISLAND'] in column 0 during transform

  warnings.warn(

[CV] END feature_selection__k=4, preparation__num__imputer__strategy=mean; tot
al time=   8.7s
[CV] END feature_selection__k=4, preparation__num__imputer__strategy=mean; tot
al time=  11.1s
[CV] END feature_selection__k=4, preparation__num__imputer__strategy=mean; tot
al time=  11.4s
[CV] END feature_selection__k=4, preparation__num__imputer__strategy=mean; tot
al time=  11.3s
[CV] END feature_selection__k=4, preparation__num__imputer__strategy=mean; tot
al time=  11.3s
```

```
/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_validation.p
y:960: UserWarning: Scoring failed. The score on this train-test partition for
these parameters will be set to nan. Details:
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_valid
ation.py", line 949, in _score
    scores = scorer(estimator, X_test, y_test, **score_params)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 288, in __call__
    return self._score(partial(_cached_call, None), estimator, X, y_true, **_k
wargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 380, in _score
    y_pred = method_caller(
             ^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 90, in _cached_call
    result, _ = _get_response_values(
                ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_response.py", l
ine 242, in _get_response_values
    y_pred, pos_label = prediction_method(X), None
                        ^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 78
7, in predict
    Xt = transform.transform(Xt)
         ^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
```

```
ormer.py", line 1101, in transform
    Xs = self._call_func_on_transformers(
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
ormer.py", line 910, in _call_func_on_transformers
    return Parallel(n_jobs=self.n_jobs)(jobs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 77, in __call__
    return super().__call__(iterable_with_config)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 198
6, in __call__
    return output if self.return_generator else list(output)
                                                ^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 191
4, in _get_sequential_output
    res = func(*args, **kwargs)
          ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 139, in __call__
    return self.function(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 153
1, in _transform_one
    res = transformer.transform(X, **params.transform)
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
s.py", line 1043, in transform
    X_int, X_mask = self._transform(
                    ^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
```

```
s.py", line 218, in _transform
    raise ValueError(msg)
ValueError: Found unknown categories ['ISLAND'] in column 0 during transform

  warnings.warn(

[CV] END feature_selection__k=4, preparation__num__imputer__strategy=median; t
otal time=   9.5s
[CV] END feature_selection__k=4, preparation__num__imputer__strategy=median; t
otal time=  11.4s
[CV] END feature_selection__k=4, preparation__num__imputer__strategy=median; t
otal time=  11.5s
[CV] END feature_selection__k=4, preparation__num__imputer__strategy=median; t
otal time=  10.4s
[CV] END feature_selection__k=4, preparation__num__imputer__strategy=median; t
otal time=  11.1s
```

```
/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_validation.p
y:960: UserWarning: Scoring failed. The score on this train-test partition for
these parameters will be set to nan. Details:
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_valid
ation.py", line 949, in _score
    scores = scorer(estimator, X_test, y_test, **score_params)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 288, in __call__
    return self._score(partial(_cached_call, None), estimator, X, y_true, **_k
wargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 380, in _score
    y_pred = method_caller(
             ^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 90, in _cached_call
    result, _ = _get_response_values(
                ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_response.py", l
ine 242, in _get_response_values
    y_pred, pos_label = prediction_method(X), None
                        ^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 78
7, in predict
    Xt = transform.transform(Xt)
         ^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
```

```
ormer.py", line 1101, in transform
    Xs = self._call_func_on_transformers(
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
ormer.py", line 910, in _call_func_on_transformers
    return Parallel(n_jobs=self.n_jobs)(jobs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 77, in __call__
    return super().__call__(iterable_with_config)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 198
6, in __call__
    return output if self.return_generator else list(output)
                                                ^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 191
4, in _get_sequential_output
    res = func(*args, **kwargs)
          ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 139, in __call__
    return self.function(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 153
1, in _transform_one
    res = transformer.transform(X, **params.transform)
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
s.py", line 1043, in transform
    X_int, X_mask = self._transform(
                    ^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
```

```
s.py", line 218, in _transform
    raise ValueError(msg)
ValueError: Found unknown categories ['ISLAND'] in column 0 during transform

  warnings.warn(

[CV] END feature_selection__k=4, preparation__num__imputer__strategy=most_freq
uent; total time=   9.6s
[CV] END feature_selection__k=4, preparation__num__imputer__strategy=most_freq
uent; total time=  11.4s
[CV] END feature_selection__k=4, preparation__num__imputer__strategy=most_freq
uent; total time=  11.3s
[CV] END feature_selection__k=4, preparation__num__imputer__strategy=most_freq
uent; total time=  11.2s
[CV] END feature_selection__k=4, preparation__num__imputer__strategy=most_freq
uent; total time=  11.3s
```

```
/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_validation.p
y:960: UserWarning: Scoring failed. The score on this train-test partition for
these parameters will be set to nan. Details:
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_valid
ation.py", line 949, in _score
    scores = scorer(estimator, X_test, y_test, **score_params)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 288, in __call__
    return self._score(partial(_cached_call, None), estimator, X, y_true, **_k
wargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 380, in _score
    y_pred = method_caller(
             ^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 90, in _cached_call
    result, _ = _get_response_values(
                ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_response.py", l
ine 242, in _get_response_values
    y_pred, pos_label = prediction_method(X), None
                        ^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 78
7, in predict
    Xt = transform.transform(Xt)
         ^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
```

```
ormer.py", line 1101, in transform
    Xs = self._call_func_on_transformers(
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
ormer.py", line 910, in _call_func_on_transformers
    return Parallel(n_jobs=self.n_jobs)(jobs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 77, in __call__
    return super().__call__(iterable_with_config)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 198
6, in __call__
    return output if self.return_generator else list(output)
                                                ^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 191
4, in _get_sequential_output
    res = func(*args, **kwargs)
          ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 139, in __call__
    return self.function(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 153
1, in _transform_one
    res = transformer.transform(X, **params.transform)
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
s.py", line 1043, in transform
    X_int, X_mask = self._transform(
                    ^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
```

```
s.py", line 218, in _transform
    raise ValueError(msg)
ValueError: Found unknown categories ['ISLAND'] in column 0 during transform

  warnings.warn(

[CV] END feature_selection__k=5, preparation__num__imputer__strategy=mean; tot
al time=   8.5s
[CV] END feature_selection__k=5, preparation__num__imputer__strategy=mean; tot
al time=  11.8s
[CV] END feature_selection__k=5, preparation__num__imputer__strategy=mean; tot
al time=  11.6s
[CV] END feature_selection__k=5, preparation__num__imputer__strategy=mean; tot
al time=  11.5s
[CV] END feature_selection__k=5, preparation__num__imputer__strategy=mean; tot
al time=  11.8s
```

```
/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_validation.p
y:960: UserWarning: Scoring failed. The score on this train-test partition for
these parameters will be set to nan. Details:
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_valid
ation.py", line 949, in _score
    scores = scorer(estimator, X_test, y_test, **score_params)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 288, in __call__
    return self._score(partial(_cached_call, None), estimator, X, y_true, **_k
wargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 380, in _score
    y_pred = method_caller(
             ^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 90, in _cached_call
    result, _ = _get_response_values(
                ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_response.py", l
ine 242, in _get_response_values
    y_pred, pos_label = prediction_method(X), None
                        ^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 78
7, in predict
    Xt = transform.transform(Xt)
         ^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
```

```
ormer.py", line 1101, in transform
    Xs = self._call_func_on_transformers(
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
ormer.py", line 910, in _call_func_on_transformers
    return Parallel(n_jobs=self.n_jobs)(jobs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 77, in __call__
    return super().__call__(iterable_with_config)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 198
6, in __call__
    return output if self.return_generator else list(output)
                                                ^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 191
4, in _get_sequential_output
    res = func(*args, **kwargs)
          ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 139, in __call__
    return self.function(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 153
1, in _transform_one
    res = transformer.transform(X, **params.transform)
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
s.py", line 1043, in transform
    X_int, X_mask = self._transform(
                    ^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
```

```
s.py", line 218, in _transform
    raise ValueError(msg)
ValueError: Found unknown categories ['ISLAND'] in column 0 during transform

  warnings.warn(

[CV] END feature_selection__k=5, preparation__num__imputer__strategy=median; t
otal time=   9.6s
[CV] END feature_selection__k=5, preparation__num__imputer__strategy=median; t
otal time=  11.7s
[CV] END feature_selection__k=5, preparation__num__imputer__strategy=median; t
otal time=  11.6s
[CV] END feature_selection__k=5, preparation__num__imputer__strategy=median; t
otal time=  11.7s
[CV] END feature_selection__k=5, preparation__num__imputer__strategy=median; t
otal time=  11.4s
```

```
/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_validation.p
y:960: UserWarning: Scoring failed. The score on this train-test partition for
these parameters will be set to nan. Details:
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_valid
ation.py", line 949, in _score
    scores = scorer(estimator, X_test, y_test, **score_params)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 288, in __call__
    return self._score(partial(_cached_call, None), estimator, X, y_true, **_k
wargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 380, in _score
    y_pred = method_caller(
             ^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 90, in _cached_call
    result, _ = _get_response_values(
                ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_response.py", l
ine 242, in _get_response_values
    y_pred, pos_label = prediction_method(X), None
                        ^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 78
7, in predict
    Xt = transform.transform(Xt)
         ^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
```

```
ormer.py", line 1101, in transform
    Xs = self._call_func_on_transformers(
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
ormer.py", line 910, in _call_func_on_transformers
    return Parallel(n_jobs=self.n_jobs)(jobs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 77, in __call__
    return super().__call__(iterable_with_config)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 198
6, in __call__
    return output if self.return_generator else list(output)
                                                ^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 191
4, in _get_sequential_output
    res = func(*args, **kwargs)
          ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 139, in __call__
    return self.function(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 153
1, in _transform_one
    res = transformer.transform(X, **params.transform)
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
s.py", line 1043, in transform
    X_int, X_mask = self._transform(
                    ^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
```

```
s.py", line 218, in _transform
    raise ValueError(msg)
ValueError: Found unknown categories ['ISLAND'] in column 0 during transform

  warnings.warn(

[CV] END feature_selection__k=5, preparation__num__imputer__strategy=most_freq
uent; total time=   9.1s
[CV] END feature_selection__k=5, preparation__num__imputer__strategy=most_freq
uent; total time=  11.8s
[CV] END feature_selection__k=5, preparation__num__imputer__strategy=most_freq
uent; total time=  11.5s
[CV] END feature_selection__k=5, preparation__num__imputer__strategy=most_freq
uent; total time=  11.5s
[CV] END feature_selection__k=5, preparation__num__imputer__strategy=most_freq
uent; total time=  11.8s
```

```
/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_validation.p
y:960: UserWarning: Scoring failed. The score on this train-test partition for
these parameters will be set to nan. Details:
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_valid
ation.py", line 949, in _score
    scores = scorer(estimator, X_test, y_test, **score_params)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 288, in __call__
    return self._score(partial(_cached_call, None), estimator, X, y_true, **_k
wargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 380, in _score
    y_pred = method_caller(
             ^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 90, in _cached_call
    result, _ = _get_response_values(
                ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_response.py", l
ine 242, in _get_response_values
    y_pred, pos_label = prediction_method(X), None
                        ^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 78
7, in predict
    Xt = transform.transform(Xt)
         ^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
```

```
ormer.py", line 1101, in transform
    Xs = self._call_func_on_transformers(
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
ormer.py", line 910, in _call_func_on_transformers
    return Parallel(n_jobs=self.n_jobs)(jobs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 77, in __call__
    return super().__call__(iterable_with_config)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 198
6, in __call__
    return output if self.return_generator else list(output)
                                                ^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 191
4, in _get_sequential_output
    res = func(*args, **kwargs)
          ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 139, in __call__
    return self.function(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 153
1, in _transform_one
    res = transformer.transform(X, **params.transform)
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
s.py", line 1043, in transform
    X_int, X_mask = self._transform(
                    ^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
```

```
s.py", line 218, in _transform
    raise ValueError(msg)
ValueError: Found unknown categories ['ISLAND'] in column 0 during transform

  warnings.warn(

[CV] END feature_selection__k=6, preparation__num__imputer__strategy=mean; tot
al time=  10.0s
[CV] END feature_selection__k=6, preparation__num__imputer__strategy=mean; tot
al time=  12.0s
[CV] END feature_selection__k=6, preparation__num__imputer__strategy=mean; tot
al time=  12.5s
[CV] END feature_selection__k=6, preparation__num__imputer__strategy=mean; tot
al time=  12.3s
[CV] END feature_selection__k=6, preparation__num__imputer__strategy=mean; tot
al time=  12.3s
```

```
/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_validation.p
y:960: UserWarning: Scoring failed. The score on this train-test partition for
these parameters will be set to nan. Details:
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_valid
ation.py", line 949, in _score
    scores = scorer(estimator, X_test, y_test, **score_params)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 288, in __call__
    return self._score(partial(_cached_call, None), estimator, X, y_true, **_k
wargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 380, in _score
    y_pred = method_caller(
             ^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 90, in _cached_call
    result, _ = _get_response_values(
                ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_response.py", l
ine 242, in _get_response_values
    y_pred, pos_label = prediction_method(X), None
                        ^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 78
7, in predict
    Xt = transform.transform(Xt)
         ^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
```

```
ormer.py", line 1101, in transform
    Xs = self._call_func_on_transformers(
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
ormer.py", line 910, in _call_func_on_transformers
    return Parallel(n_jobs=self.n_jobs)(jobs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 77, in __call__
    return super().__call__(iterable_with_config)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 198
6, in __call__
    return output if self.return_generator else list(output)
                                                ^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 191
4, in _get_sequential_output
    res = func(*args, **kwargs)
          ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 139, in __call__
    return self.function(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 153
1, in _transform_one
    res = transformer.transform(X, **params.transform)
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
s.py", line 1043, in transform
    X_int, X_mask = self._transform(
                    ^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
```

```
s.py", line 218, in _transform
    raise ValueError(msg)
ValueError: Found unknown categories ['ISLAND'] in column 0 during transform

  warnings.warn(

[CV] END feature_selection__k=6, preparation__num__imputer__strategy=median; t
otal time=   9.0s
[CV] END feature_selection__k=6, preparation__num__imputer__strategy=median; t
otal time=  12.0s
[CV] END feature_selection__k=6, preparation__num__imputer__strategy=median; t
otal time=  12.4s
[CV] END feature_selection__k=6, preparation__num__imputer__strategy=median; t
otal time=  12.1s
[CV] END feature_selection__k=6, preparation__num__imputer__strategy=median; t
otal time=  12.1s
```

```
/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_validation.p
y:960: UserWarning: Scoring failed. The score on this train-test partition for
these parameters will be set to nan. Details:
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_valid
ation.py", line 949, in _score
    scores = scorer(estimator, X_test, y_test, **score_params)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 288, in __call__
    return self._score(partial(_cached_call, None), estimator, X, y_true, **_k
wargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 380, in _score
    y_pred = method_caller(
             ^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 90, in _cached_call
    result, _ = _get_response_values(
                ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_response.py", l
ine 242, in _get_response_values
    y_pred, pos_label = prediction_method(X), None
                        ^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 78
7, in predict
    Xt = transform.transform(Xt)
         ^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
```

```
ormer.py", line 1101, in transform
    Xs = self._call_func_on_transformers(
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
ormer.py", line 910, in _call_func_on_transformers
    return Parallel(n_jobs=self.n_jobs)(jobs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 77, in __call__
    return super().__call__(iterable_with_config)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 198
6, in __call__
    return output if self.return_generator else list(output)
                                                ^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 191
4, in _get_sequential_output
    res = func(*args, **kwargs)
          ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 139, in __call__
    return self.function(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 153
1, in _transform_one
    res = transformer.transform(X, **params.transform)
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
s.py", line 1043, in transform
    X_int, X_mask = self._transform(
                    ^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
```

```
s.py", line 218, in _transform
    raise ValueError(msg)
ValueError: Found unknown categories ['ISLAND'] in column 0 during transform

  warnings.warn(

[CV] END feature_selection__k=6, preparation__num__imputer__strategy=most_freq
uent; total time=  10.1s
[CV] END feature_selection__k=6, preparation__num__imputer__strategy=most_freq
uent; total time=  11.9s
[CV] END feature_selection__k=6, preparation__num__imputer__strategy=most_freq
uent; total time=  12.3s
[CV] END feature_selection__k=6, preparation__num__imputer__strategy=most_freq
uent; total time=  12.0s
[CV] END feature_selection__k=6, preparation__num__imputer__strategy=most_freq
uent; total time=  12.0s
```

```
/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_validation.p
y:960: UserWarning: Scoring failed. The score on this train-test partition for
these parameters will be set to nan. Details:
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_valid
ation.py", line 949, in _score
    scores = scorer(estimator, X_test, y_test, **score_params)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 288, in __call__
    return self._score(partial(_cached_call, None), estimator, X, y_true, **_k
wargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 380, in _score
    y_pred = method_caller(
             ^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 90, in _cached_call
    result, _ = _get_response_values(
                ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_response.py", l
ine 242, in _get_response_values
    y_pred, pos_label = prediction_method(X), None
                        ^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 78
7, in predict
    Xt = transform.transform(Xt)
         ^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
```

```
ormer.py", line 1101, in transform
    Xs = self._call_func_on_transformers(
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
ormer.py", line 910, in _call_func_on_transformers
    return Parallel(n_jobs=self.n_jobs)(jobs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 77, in __call__
    return super().__call__(iterable_with_config)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 198
6, in __call__
    return output if self.return_generator else list(output)
                                                ^^^^^^^^^^^^

  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 191
4, in _get_sequential_output
    res = func(*args, **kwargs)
          ^^^^^^^^^^^^^^^^^^^^^

  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 139, in __call__
    return self.function(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 153
1, in _transform_one
    res = transformer.transform(X, **params.transform)
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^

  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
s.py", line 1043, in transform
    X_int, X_mask = self._transform(
                    ^^^^^^^^^^^^^^^^

  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
```

```
s.py", line 218, in _transform
    raise ValueError(msg)
ValueError: Found unknown categories ['ISLAND'] in column 0 during transform

  warnings.warn(

[CV] END feature_selection__k=7, preparation__num__imputer__strategy=mean; tot
al time=  10.5s
[CV] END feature_selection__k=7, preparation__num__imputer__strategy=mean; tot
al time=  13.4s
[CV] END feature_selection__k=7, preparation__num__imputer__strategy=mean; tot
al time=  13.1s
[CV] END feature_selection__k=7, preparation__num__imputer__strategy=mean; tot
al time=  12.3s
[CV] END feature_selection__k=7, preparation__num__imputer__strategy=mean; tot
al time=  13.4s
```

```
/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_validation.p
y:960: UserWarning: Scoring failed. The score on this train-test partition for
these parameters will be set to nan. Details:
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_valid
ation.py", line 949, in _score
    scores = scorer(estimator, X_test, y_test, **score_params)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 288, in __call__
    return self._score(partial(_cached_call, None), estimator, X, y_true, **_k
wargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 380, in _score
    y_pred = method_caller(
             ^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 90, in _cached_call
    result, _ = _get_response_values(
                ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_response.py", l
ine 242, in _get_response_values
    y_pred, pos_label = prediction_method(X), None
                        ^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 78
7, in predict
    Xt = transform.transform(Xt)
         ^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
```

```
ormer.py", line 1101, in transform
    Xs = self._call_func_on_transformers(
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
ormer.py", line 910, in _call_func_on_transformers
    return Parallel(n_jobs=self.n_jobs)(jobs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 77, in __call__
    return super().__call__(iterable_with_config)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 198
6, in __call__
    return output if self.return_generator else list(output)
                                                ^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 191
4, in _get_sequential_output
    res = func(*args, **kwargs)
          ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 139, in __call__
    return self.function(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 153
1, in _transform_one
    res = transformer.transform(X, **params.transform)
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
s.py", line 1043, in transform
    X_int, X_mask = self._transform(
                    ^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
```

```
s.py", line 218, in _transform
    raise ValueError(msg)
ValueError: Found unknown categories ['ISLAND'] in column 0 during transform

  warnings.warn(

[CV] END feature_selection__k=7, preparation__num__imputer__strategy=median; t
otal time=  10.9s
[CV] END feature_selection__k=7, preparation__num__imputer__strategy=median; t
otal time=  13.2s
[CV] END feature_selection__k=7, preparation__num__imputer__strategy=median; t
otal time=  12.9s
[CV] END feature_selection__k=7, preparation__num__imputer__strategy=median; t
otal time=  13.0s
[CV] END feature_selection__k=7, preparation__num__imputer__strategy=median; t
otal time=  12.6s
```

```
/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_validation.p
y:960: UserWarning: Scoring failed. The score on this train-test partition for
these parameters will be set to nan. Details:
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_valid
ation.py", line 949, in _score
    scores = scorer(estimator, X_test, y_test, **score_params)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 288, in __call__
    return self._score(partial(_cached_call, None), estimator, X, y_true, **_k
wargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 380, in _score
    y_pred = method_caller(
             ^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 90, in _cached_call
    result, _ = _get_response_values(
                ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_response.py", l
ine 242, in _get_response_values
    y_pred, pos_label = prediction_method(X), None
                        ^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 78
7, in predict
    Xt = transform.transform(Xt)
         ^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
```

```
ormer.py", line 1101, in transform
    Xs = self._call_func_on_transformers(
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
ormer.py", line 910, in _call_func_on_transformers
    return Parallel(n_jobs=self.n_jobs)(jobs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 77, in __call__
    return super().__call__(iterable_with_config)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 198
6, in __call__
    return output if self.return_generator else list(output)
                                                ^^^^^^^^^^^^

  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 191
4, in _get_sequential_output
    res = func(*args, **kwargs)
          ^^^^^^^^^^^^^^^^^^^^^

  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 139, in __call__
    return self.function(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 153
1, in _transform_one
    res = transformer.transform(X, **params.transform)
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^

  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
s.py", line 1043, in transform
    X_int, X_mask = self._transform(
                    ^^^^^^^^^^^^^^^^

  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
```

```
s.py", line 218, in _transform
    raise ValueError(msg)
ValueError: Found unknown categories ['ISLAND'] in column 0 during transform

  warnings.warn(

[CV] END feature_selection__k=7, preparation__num__imputer__strategy=most_freq
uent; total time=  10.6s
[CV] END feature_selection__k=7, preparation__num__imputer__strategy=most_freq
uent; total time=  13.0s
[CV] END feature_selection__k=7, preparation__num__imputer__strategy=most_freq
uent; total time=  12.7s
[CV] END feature_selection__k=7, preparation__num__imputer__strategy=most_freq
uent; total time=  12.8s
[CV] END feature_selection__k=7, preparation__num__imputer__strategy=most_freq
uent; total time=  12.9s
```

```
/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_validation.p
y:960: UserWarning: Scoring failed. The score on this train-test partition for
these parameters will be set to nan. Details:
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_valid
ation.py", line 949, in _score
    scores = scorer(estimator, X_test, y_test, **score_params)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 288, in __call__
    return self._score(partial(_cached_call, None), estimator, X, y_true, **_k
wargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 380, in _score
    y_pred = method_caller(
             ^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 90, in _cached_call
    result, _ = _get_response_values(
                ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_response.py", l
ine 242, in _get_response_values
    y_pred, pos_label = prediction_method(X), None
                        ^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 78
7, in predict
    Xt = transform.transform(Xt)
         ^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
```

```
ormer.py", line 1101, in transform
    Xs = self._call_func_on_transformers(
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
ormer.py", line 910, in _call_func_on_transformers
    return Parallel(n_jobs=self.n_jobs)(jobs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 77, in __call__
    return super().__call__(iterable_with_config)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 198
6, in __call__
    return output if self.return_generator else list(output)
                                                ^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 191
4, in _get_sequential_output
    res = func(*args, **kwargs)
          ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 139, in __call__
    return self.function(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 153
1, in _transform_one
    res = transformer.transform(X, **params.transform)
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
s.py", line 1043, in transform
    X_int, X_mask = self._transform(
                    ^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
```

```
s.py", line 218, in _transform
    raise ValueError(msg)
ValueError: Found unknown categories ['ISLAND'] in column 0 during transform

  warnings.warn(

[CV] END feature_selection__k=8, preparation__num__imputer__strategy=mean; tot
al time=  14.6s
[CV] END feature_selection__k=8, preparation__num__imputer__strategy=mean; tot
al time=  14.9s
[CV] END feature_selection__k=8, preparation__num__imputer__strategy=mean; tot
al time=  15.4s
[CV] END feature_selection__k=8, preparation__num__imputer__strategy=mean; tot
al time=  16.9s
[CV] END feature_selection__k=8, preparation__num__imputer__strategy=mean; tot
al time=  15.2s
```

```
/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_validation.p
y:960: UserWarning: Scoring failed. The score on this train-test partition for
these parameters will be set to nan. Details:
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_valid
ation.py", line 949, in _score
    scores = scorer(estimator, X_test, y_test, **score_params)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 288, in __call__
    return self._score(partial(_cached_call, None), estimator, X, y_true, **_k
wargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 380, in _score
    y_pred = method_caller(
             ^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 90, in _cached_call
    result, _ = _get_response_values(
                ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_response.py", l
ine 242, in _get_response_values
    y_pred, pos_label = prediction_method(X), None
                        ^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 78
7, in predict
    Xt = transform.transform(Xt)
         ^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
```

```
ormer.py", line 1101, in transform
    Xs = self._call_func_on_transformers(
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
ormer.py", line 910, in _call_func_on_transformers
    return Parallel(n_jobs=self.n_jobs)(jobs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 77, in __call__
    return super().__call__(iterable_with_config)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 198
6, in __call__
    return output if self.return_generator else list(output)
                                                ^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 191
4, in _get_sequential_output
    res = func(*args, **kwargs)
          ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 139, in __call__
    return self.function(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 153
1, in _transform_one
    res = transformer.transform(X, **params.transform)
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
s.py", line 1043, in transform
    X_int, X_mask = self._transform(
                    ^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
```

```
s.py", line 218, in _transform
    raise ValueError(msg)
ValueError: Found unknown categories ['ISLAND'] in column 0 during transform

  warnings.warn(

[CV] END feature_selection__k=8, preparation__num__imputer__strategy=median; t
otal time=  13.7s
[CV] END feature_selection__k=8, preparation__num__imputer__strategy=median; t
otal time=  14.9s
[CV] END feature_selection__k=8, preparation__num__imputer__strategy=median; t
otal time=  16.4s
[CV] END feature_selection__k=8, preparation__num__imputer__strategy=median; t
otal time=  16.6s
[CV] END feature_selection__k=8, preparation__num__imputer__strategy=median; t
otal time=  16.4s
```

```
/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_validation.p
y:960: UserWarning: Scoring failed. The score on this train-test partition for
these parameters will be set to nan. Details:
Traceback (most recent call last):
  File "/usr/local/lib/python3.12/dist-packages/sklearn/model_selection/_valid
ation.py", line 949, in _score
    scores = scorer(estimator, X_test, y_test, **score_params)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 288, in __call__
    return self._score(partial(_cached_call, None), estimator, X, y_true, **_k
wargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 380, in _score
    y_pred = method_caller(
             ^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_scorer.py", l
ine 90, in _cached_call
    result, _ = _get_response_values(
                ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_response.py", l
ine 242, in _get_response_values
    y_pred, pos_label = prediction_method(X), None
                        ^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 78
7, in predict
    Xt = transform.transform(Xt)
         ^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
```

```
ormer.py", line 1101, in transform
    Xs = self._call_func_on_transformers(
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/compose/_column_transf
ormer.py", line 910, in _call_func_on_transformers
    return Parallel(n_jobs=self.n_jobs)(jobs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 77, in __call__
    return super().__call__(iterable_with_config)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 198
6, in __call__
    return output if self.return_generator else list(output)
                                                ^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/joblib/parallel.py", line 191
4, in _get_sequential_output
    res = func(*args, **kwargs)
          ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/parallel.py", li
ne 139, in __call__
    return self.function(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/pipeline.py", line 153
1, in _transform_one
    res = transformer.transform(X, **params.transform)
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/utils/_set_output.py",
line 319, in wrapped
    data_to_wrap = f(self, X, *args, **kwargs)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
s.py", line 1043, in transform
    X_int, X_mask = self._transform(
                    ^^^^^^^^^^^^^^^^
  File "/usr/local/lib/python3.12/dist-packages/sklearn/preprocessing/_encoder
```

```
s.py", line 218, in _transform
    raise ValueError(msg)
ValueError: Found unknown categories ['ISLAND'] in column 0 during transform

  warnings.warn(

[CV] END feature_selection__k=8, preparation__num__imputer__strategy=most_freq
uent; total time=  13.4s
[CV] END feature_selection__k=8, preparation__num__imputer__strategy=most_freq
uent; total time=  16.4s
[CV] END feature_selection__k=8, preparation__num__imputer__strategy=most_freq
uent; total time=  15.8s
[CV] END feature_selection__k=8, preparation__num__imputer__strategy=most_freq
uent; total time=  17.6s
[CV] END feature_selection__k=8, preparation__num__imputer__strategy=most_freq
uent; total time=  16.9s
```