Advanced Data Analysis

DATA 71200

Class 2

Course Schedule

3-Jun	**Machine Learning Pipeline/ Representing Data**
4-Jun	*Async: DataCamp Modules*
5-Jun	Inspecting Data
6-Jun	*Async: DataCamp Modules*
10-Jun	Evaluation Methods
11-Jun	*Async: DataCamp Modules*

Question Set 1

Géron (p. 4–9)

- **How would you define Machine Learning?**
- **Can you name four types of problems where it shines?**
- **What is a labeled training set?**

Question Set 1

- **How would you define Machine Learning?**
  - "Machine Learning is about building systems that can learn from data. Learning means getting better at some task, given some performance measure."
- **Can you name four types of problems where it shines?**
  - "Machine Learning is great for complex problems for which we have no algorithmic solution, to replace long lists of hand-tuned rules, to build systems that adapt to fluctuating environments, and finally to help humans learn (e.g., data mining)."

Question Set 1

- **What is a labeled training set?**
  - “A labeled training set is a training set that contains the desired solution (a.k.a. a label) for each instance.”

Question Set 2

Géron (p. 22–30)

- **Can you name some of the main challenges in Machine Learning?**
- **If your model performs great on the training data but generalizes poorly to new instances, what is happening? Can you name three possible solutions?**
- **What is a test set and why would you want to use it?**
- **What is the purpose of a validation set?**
- **What can go wrong if you tune hyperparameters using the test set?**
- **What is cross-validation and why would you prefer it to a validation set?**

Question Set 2

- **Can you name some of the main challenges in Machine Learning?**
  - "Some of the main challenges in Machine Learning are the **lack of data**, **poor data quality**, **non-representative data**, **uninformative features**, excessively simple **models that underfit** the training data, and excessively complex **models that overfit** the data."

Question Set 2

- **If your model performs great on the training data but generalizes poorly to new instances, what is happening? Can you name three possible solutions?**
  - "If a model performs great on the training data but generalizes poorly to new instances, the model is likely overfitting the training data (or we got extremely lucky on the training data). Possible solutions to overfitting are getting more data, simplifying the model (selecting a simpler algorithm, reducing the number of parameters or features used, or regularizing the model), or reducing the noise in the training data."

Question Set 2

- **What is a test set and why would you want to use it?**
  - "A test set is used to estimate the generalization error that a model will make on new instances, before the model is launched in production."
- **What is the purpose of a validation set?**
  - "A validation set is used to compare models. It makes it possible to select the best model and tune the hyperparameters."

Question Set 2

- **What can go wrong if you tune hyperparameters using the test set?**
  - "If you tune hyperparameters using the test set, you risk overfitting the test set, and the generalization error you measure will be optimistic (you may launch a model that performs worse than you expect)."
- **What is cross-validation and why would you prefer it to a validation set?**
  - "Cross-validation is a technique that makes it possible to compare models (for model selection and hyperparameter tuning) without the need for a separate validation set. This saves precious training data."

Question Set 2

- **If your model performs great on the training data but generalizes poorly to new instances, what is happening? Can you name three possible solutions?**
  - "If a model performs great on the training data but generalizes poorly to new instances, the model is likely overfitting the training data (or we got extremely lucky on the training data). Possible solutions to overfitting are **getting more data**, **simplifying the model** (selecting a simpler algorithm, **reducing the number of parameters** or features used, or regularizing the model), or **reducing the noise in the training data**."

Question Set 2

- **What is a test set and why would you want to use it?**
  - "A test set is used to **estimate the generalization error** that a model will make **on new instances**, before the model is launched in production."
- **What is the purpose of a validation set?**
  - "A validation set is used to compare models. It makes it possible to **select the best model** and **tune the hyperparameters**."

Question Set 2

- **What can go wrong if you tune hyperparameters using the test set?**
  - "If you tune hyperparameters using the test set, you risk **overfitting** the test set, and the generalization error you measure will be optimistic (you may launch a model that performs worse than you expect)."
- **What is cross-validation and why would you prefer it to a validation set?**
  - "Cross-validation is a technique that makes it possible to **compare models** (for model selection and hyperparameter tuning) **without the need for a separate validation set**. This saves precious training

Terminology Review

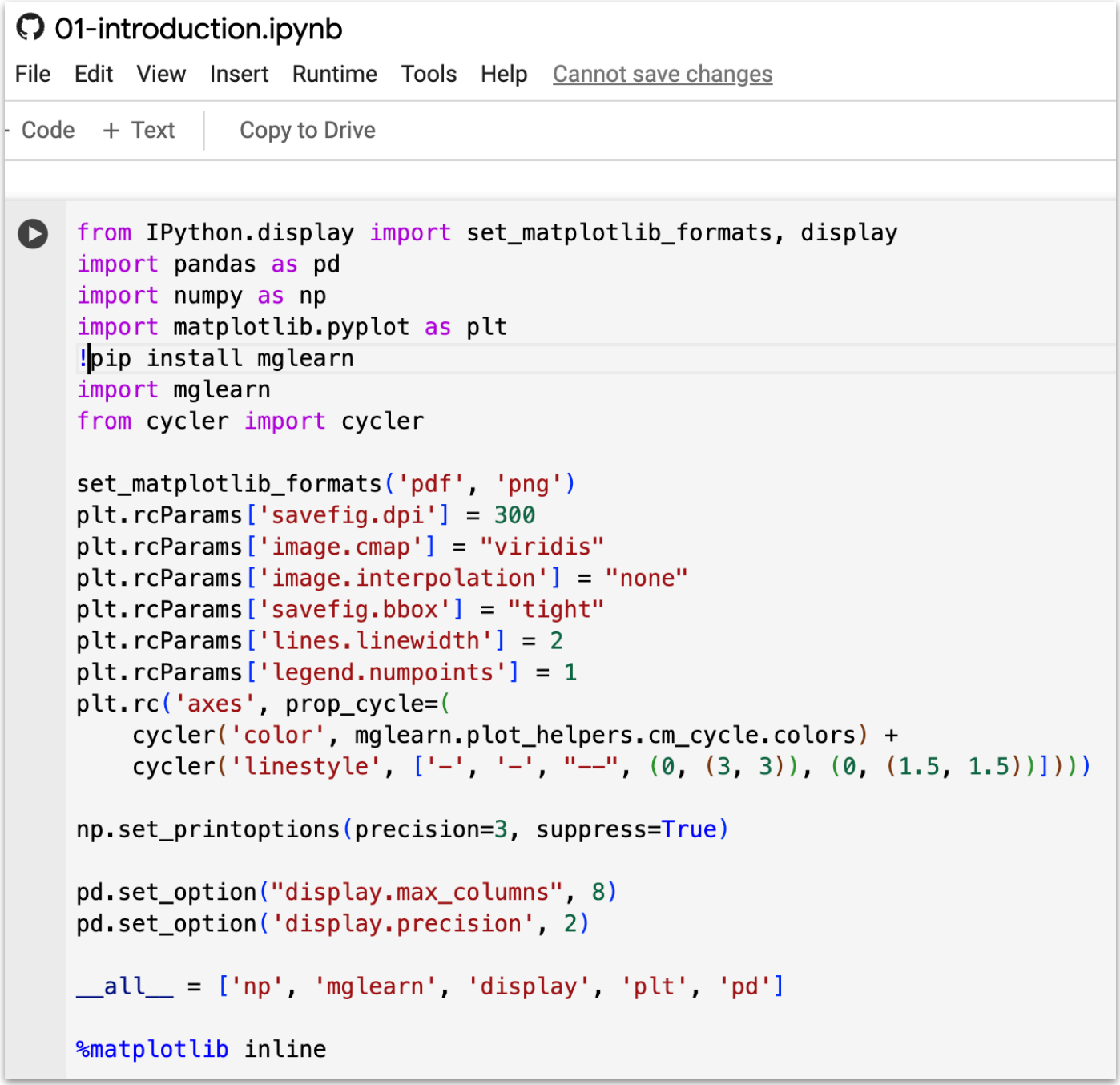
- **training set**
  - labeled training set
- **testing set**
- **validation set**
- **cross-validation**
- **class**
- **classification**
- **samples**
- **labels**
- **features**
- **regularization**
- **hyperparameter**

Reminder

Open 01-introduction.ipynb in Google Colab

Copy contents of preamble.py from repo into the first cell, replacing:

```
from preamble import *
```



Machine Learning Pipeline

- **“However simple or complex the Machine Learning problem at hand may be, it will always contain the following steps:**
  - Data loading, preparation and splitting into the train and test partitions
  - Model selection and training ("fitting")
  - Model performance assessment”

Machine Learning Pipeline

- **“However simple or complex the Machine Learning problem at hand may be, it will always contain the following steps:**
  - **Data loading, preparation and splitting into the train and test partitions**
  - Model selection and training ("fitting")
  - Model performance assessment”

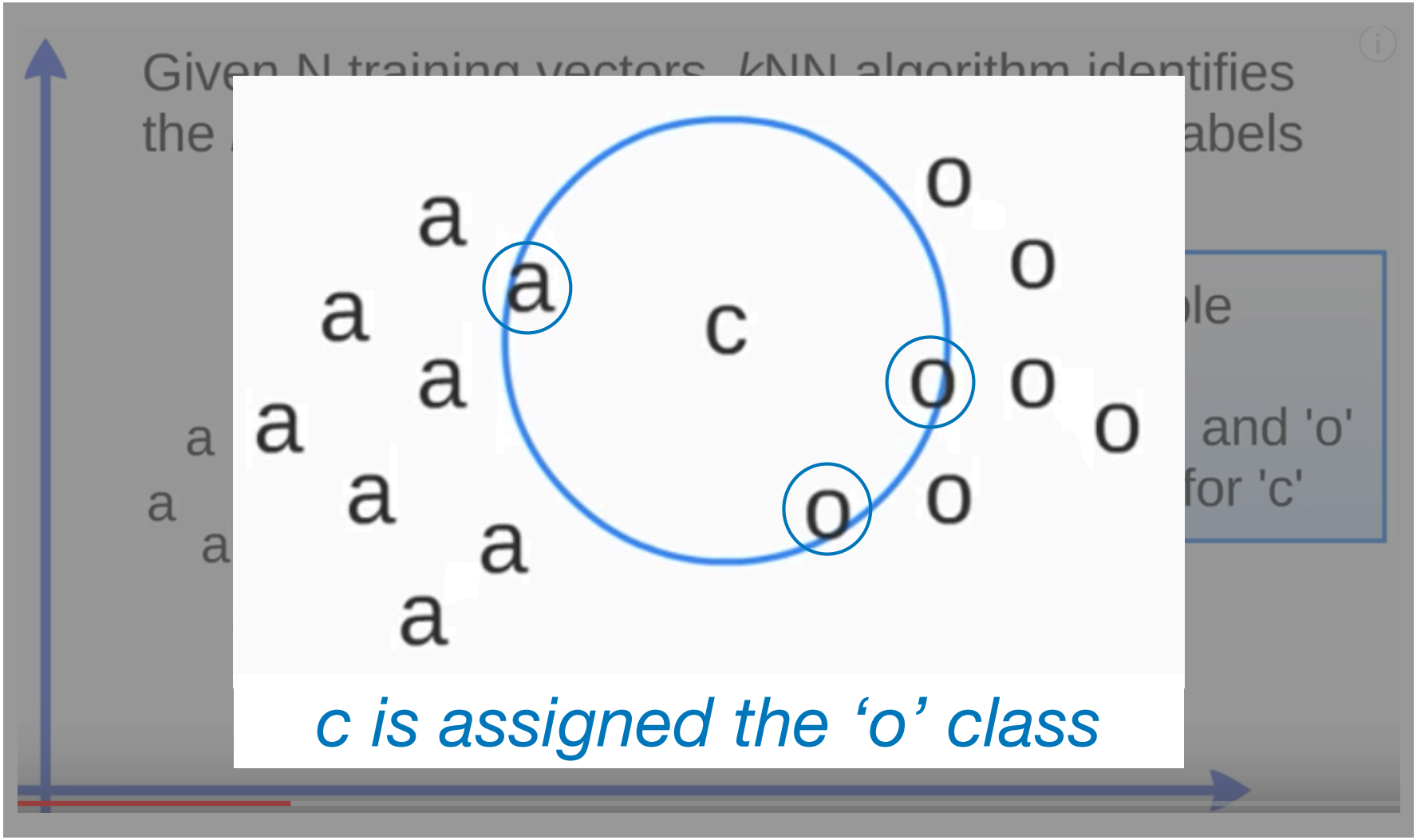
Notebook
01-introduction.ipynb
[10-24]

Machine Learning Pipeline

- **“However simple or complex the Machine Learning problem at hand may be, it will always contain the following steps:**
  - Data loading, preparation and splitting into the train and test partitions
  - **Model selection and training ("fitting")**
  - Model performance assessment”

Notebook
01-introduction.ipynb
[25-28]

k Nearest Neighbor (kNN)



https://www.youtube.com/watch?v=UqYde-LULfs

Machine Learning Pipeline

- **“However simple or complex the Machine Learning problem at hand may be, it will always contain the following steps:**
  - Data loading, preparation and splitting into the train and test partitions
  - Model selection and training ("fitting")
  - **Model performance assessment”**

Notebook
01-introduction.ipynb
[29-32]

Frame the Problem

- **“what exactly is the business (research) objective”**
  - “how does the company (researcher) expect to use and benefit from this model?”
  - will determine
    - “how you frame the problem”
    - “what algorithms you will select”
    - “what performance measure you will use to evaluate your model”
    - "how much effort you should spend tweaking it”

Géron, Aurélien. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* O’Reilly Media, Inc.

Group Question

- **What do you most want to learn to do with machine learning?**
  - What kind of data are you interested in working with?
  - What kind of questions do you want to be able to ask of your data?

Inspecting Data to Gain Insights

- **Data size and type**
- **Summary statistics**
- **Histograms**
- **Visualizing Geographic Data**

Géron, Aurélien. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* O'Reilly Media, Inc.

```
In [6]: housing.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 10 columns):
longitude             20640 non-null float64
latitude              20640 non-null float64
housing_median_age    20640 non-null float64
total_rooms           20640 non-null float64
total_bedrooms        20433 non-null float64
population            20640 non-null float64
households            20640 non-null float64
median_income         20640 non-null float64
median_house_value    20640 non-null float64
ocean_proximity       20640 non-null object
dtypes: float64(9), object(1)
memory usage: 1.6+ MB
```

Figure 2-6. Housing info

Géron, Aurélien. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* O'Reilly Media, Inc.

```
In [8]: housing.describe()
```

Out[8]:

	longitude	latitude	housing_median_age	total_rooms	total_bedr
count	20640.000000	20640.000000	20640.000000	20640.000000	20433.000(
mean	-119.569704	35.631861	28.639486	2635.763081	537.870553
std	2.003532	2.135952	12.585558	2181.615252	421.38507(
min	-124.350000	32.540000	1.000000	2.000000	1.000000
25%	-121.800000	33.930000	18.000000	1447.750000	296.00000(
50%	-118.490000	34.260000	29.000000	2127.000000	435.00000(
75%	-118.010000	37.710000	37.000000	3148.000000	647.00000(
max	-114.310000	41.950000	52.000000	39320.000000	6445.0000(

Figure 2-7. Summary of each numerical attribute

Géron, Aurélien. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* O'Reilly Media, Inc.

```
%matplotlib inline   # only in a Jupyter notebook
import matplotlib.pyplot as plt
housing.hist(bins=50, figsize=(20,15))
plt.show()
```

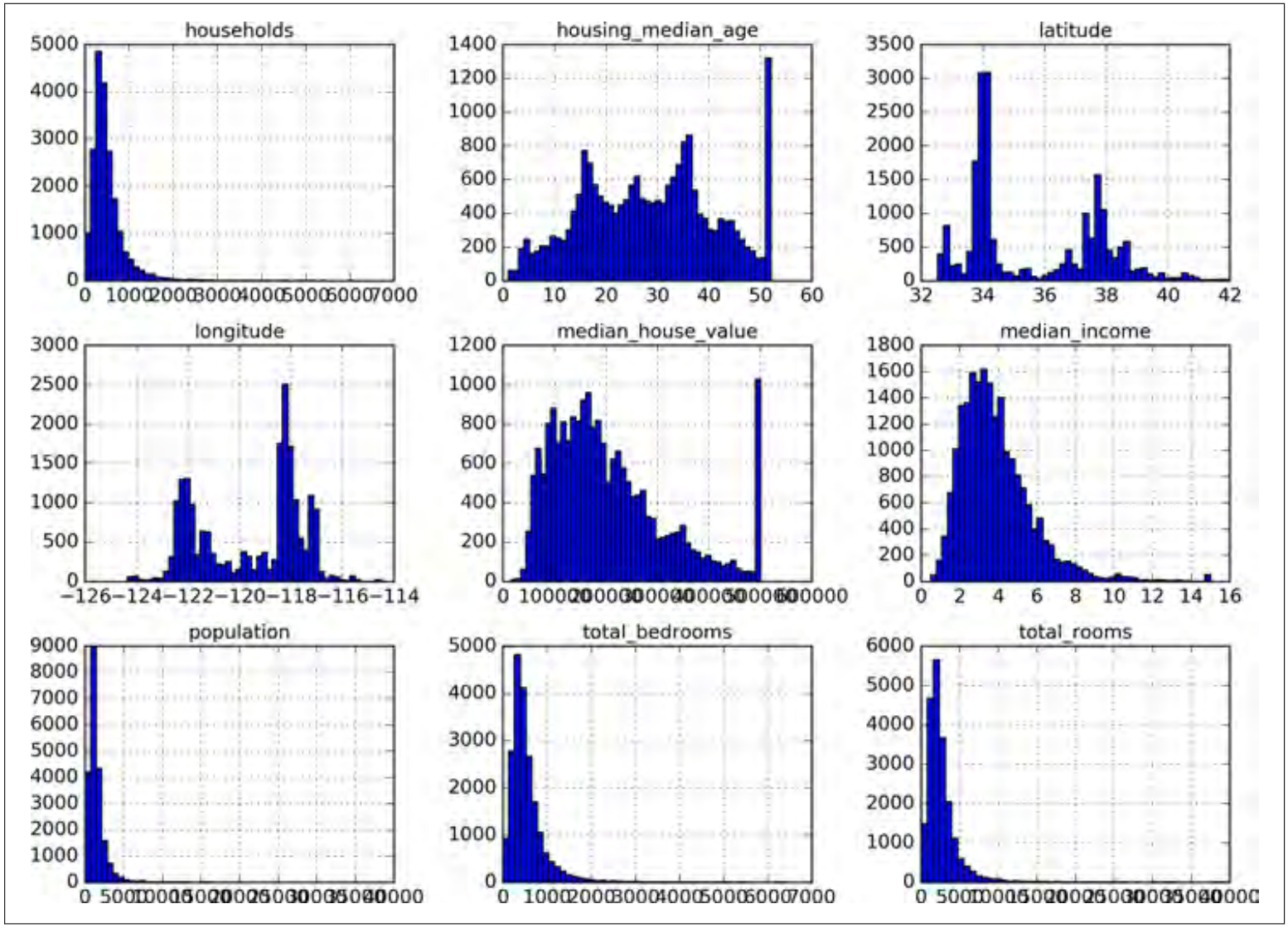


Figure 2-8. A histogram for each numerical attribute

```
%matplotlib inline   # only in a Jupyter notebook
import matplotlib.pyplot as plt
housing.hist(bins=50, figsize=(20,15))
plt.show()
```

First, the median income attribute does not look like it is expressed in US dollars (USD). After checking with the team that collected the data, you are told that the data has been scaled and capped at 15 (actually 15.0001) for higher median incomes, and at 0.5 (actually 0.4999) for lower median incomes. Working with preprocessed attributes is common in Machine Learning, and it is not necessarily a problem, but you should try to understand how the data was computed.

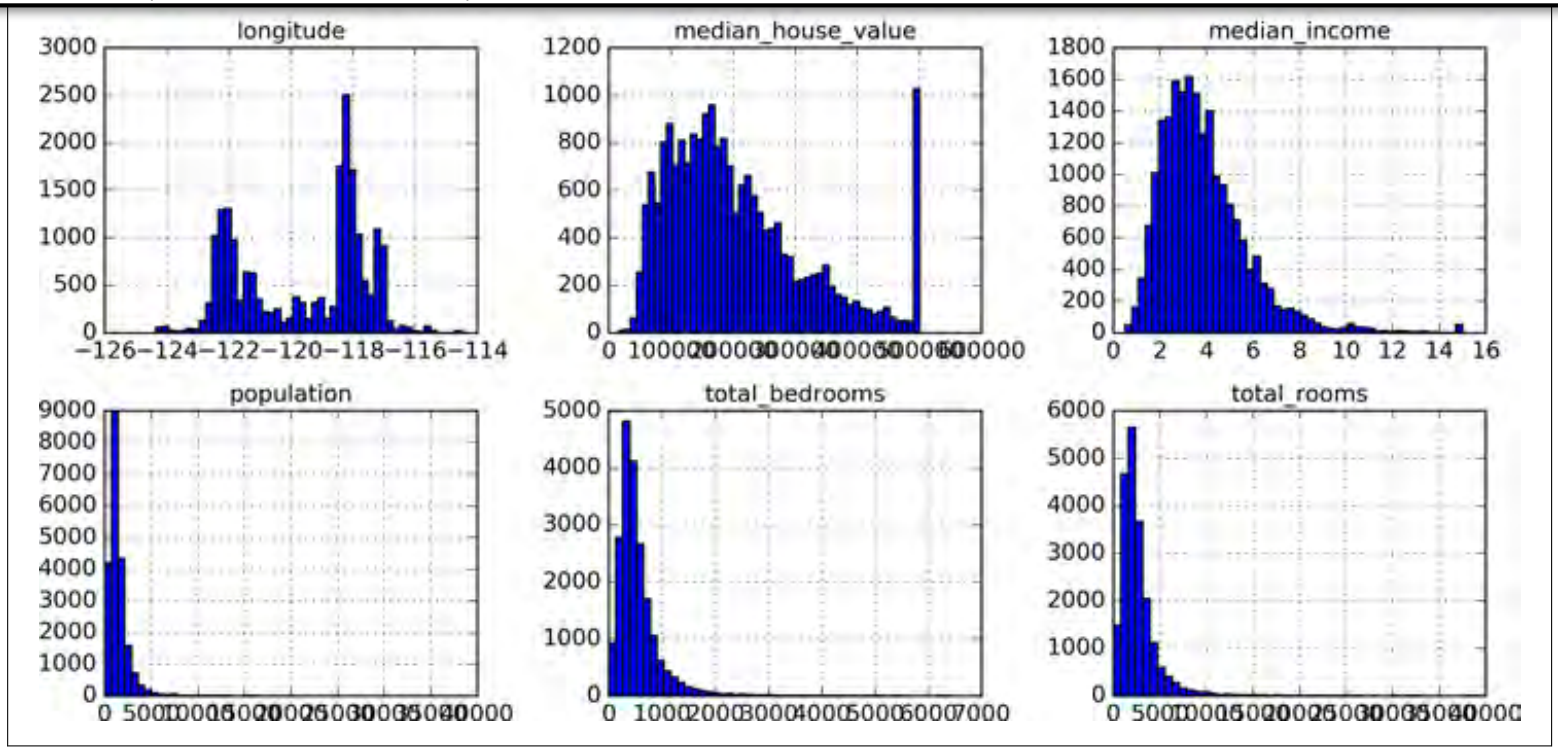
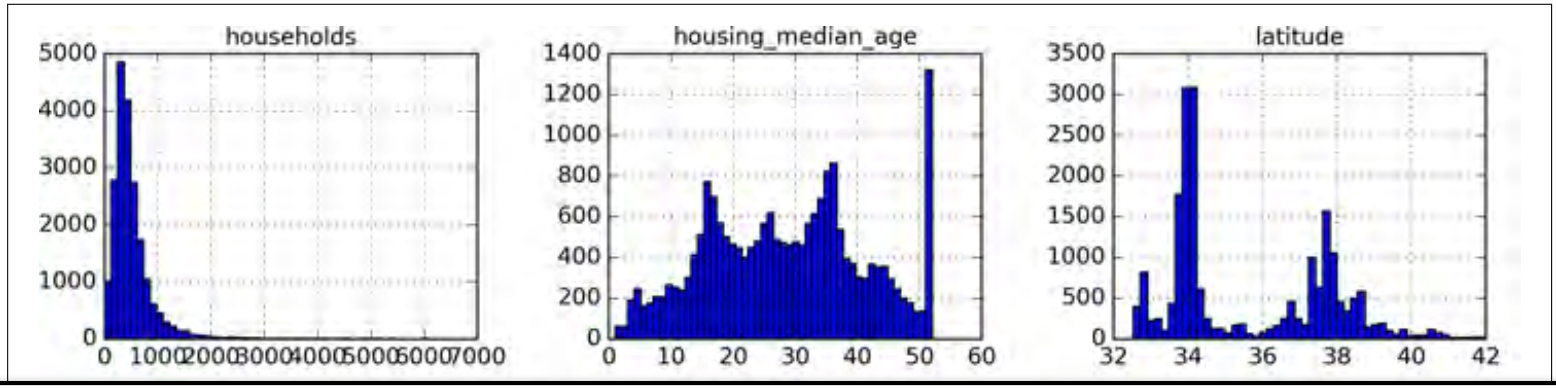


Figure 2-8. A histogram for each numerical attribute

```
%matplotlib inline   # only in a Jupyter notebook
import matplotlib.pyplot as plt
housing.hist(bins=50, figsize=(20,15))
plt.show()
```



The housing median age and the median house value were also capped. The latter may be a serious problem since it is your target attribute (your labels). Your Machine Learning algorithms may learn that prices never go beyond that limit. You need to check with your client team (the team that will use your system's output) to see if this is a problem or not. If they tell you that they need precise predictions even beyond $500,000, then you have mainly two options:

a. Collect proper labels for the districts whose labels were capped.

b. Remove those districts from the training set (and also from the test set, since your system should not be evaluated poorly if it predicts values beyond $500,000).

Finally, many histograms are *tail heavy*: they extend much farther to the right of the median than to the left. This may make it a bit harder for some Machine Learning algorithms to detect patterns. We will try transforming these attributes later on to have more bell-shaped distributions.

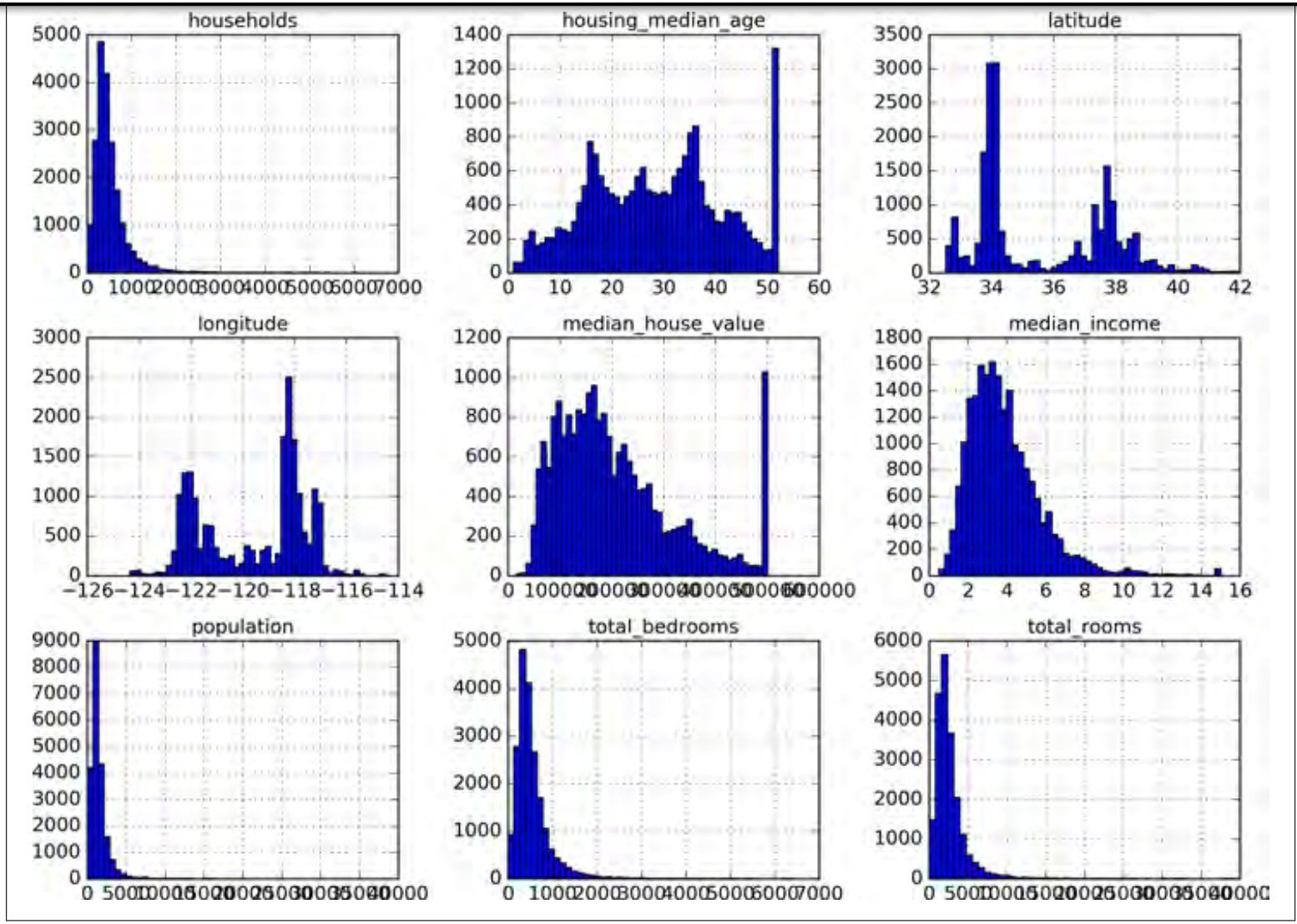


Figure 2-8. A histogram for each numerical attribute

Géron, Aurélien. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* O'Reilly Media, Inc.

```
housing.plot(kind="scatter", x="longitude", y="latitude")
```

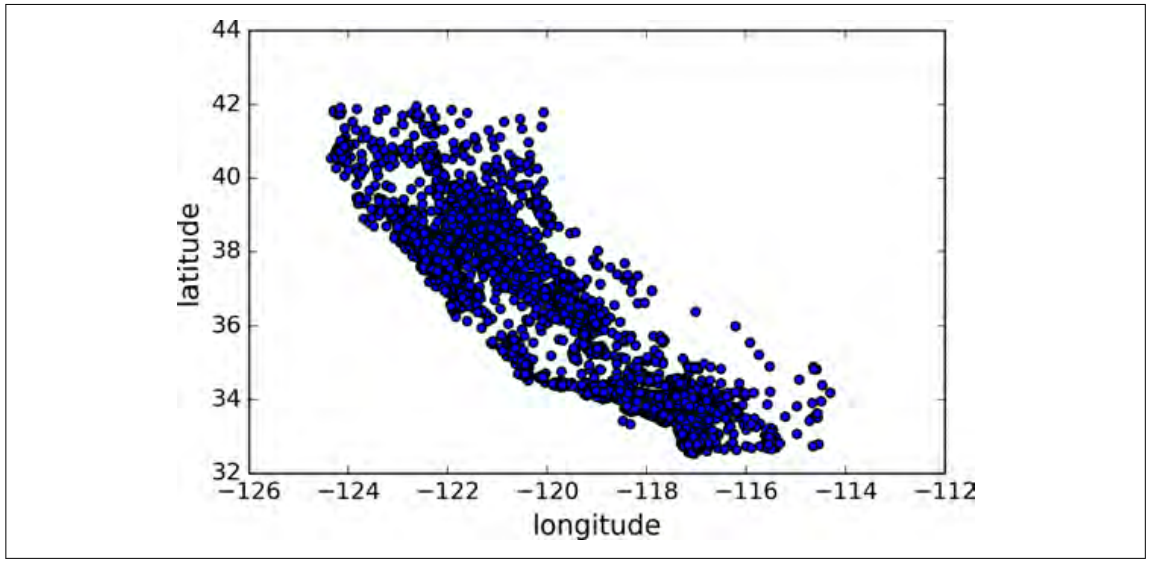


Figure 2-11. A geographical scatterplot of the data

```
housing.plot(kind="scatter", x="longitude", y="latitude", alpha=0.1)
```

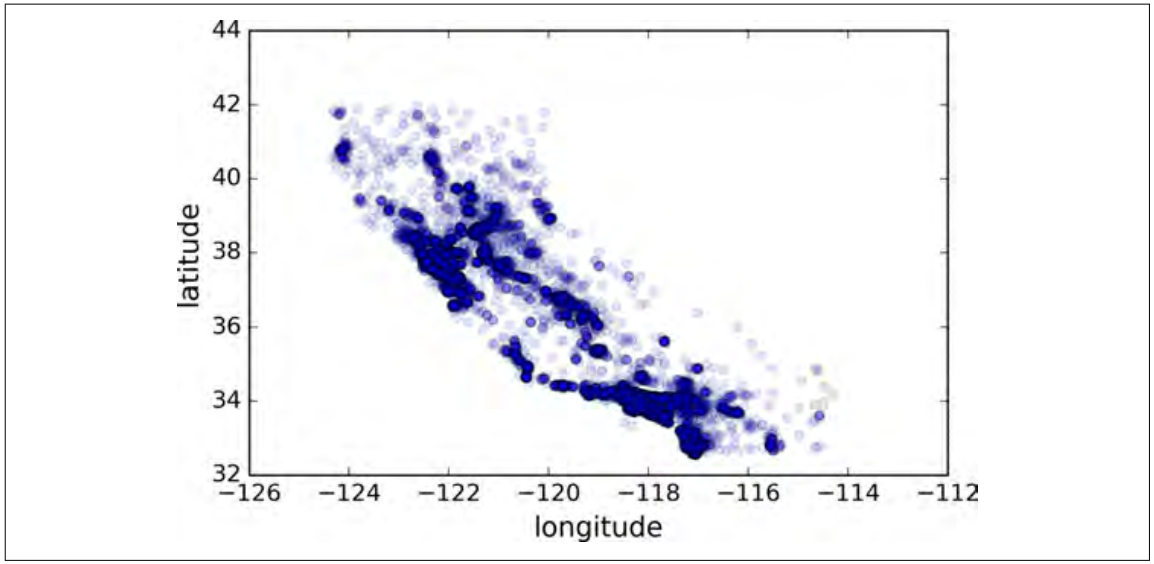


Figure 2-12. A better visualization highlighting high-density areas

Setting alpha to 0.1 emphasizes high density areas

Géron, Aurélien. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* O'Reilly Media, Inc.

```
housing.plot(kind="scatter", x="longitude", y="latitude", alpha=0.4,
    s=housing["population"]/100, label="population",
    c="median_house_value", cmap=plt.get_cmap("jet"), colorbar=True,
)
plt.legend()
```

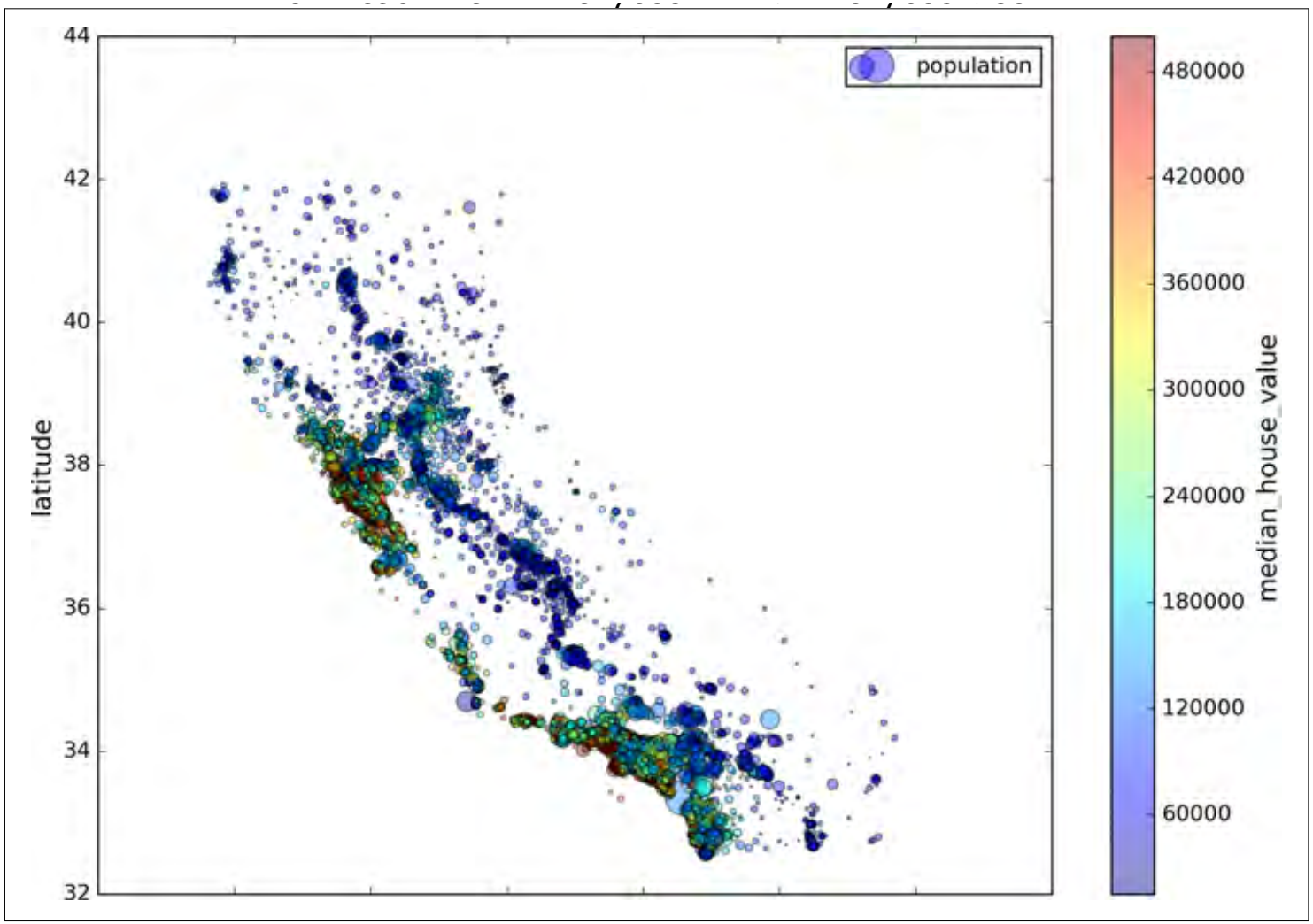


Figure 2-13. California housing prices

Géron, Aurélien. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* O'Reilly Media, Inc.

Look for Correlations

```
corr_matrix = housing.corr()
```

```
>>> corr_matrix["median_house_value"].sort_values(ascending=False)
median_house_value    1.000000
median_income         0.687170
total_rooms           0.135231
housing_median_age    0.114220
households            0.064702
total_bedrooms        0.047865
population           -0.026699
longitude            -0.047279
latitude             -0.142826
Name: median_house_value, dtype: float64
```

Géron, Aurélien. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* O'Reilly Media, Inc.

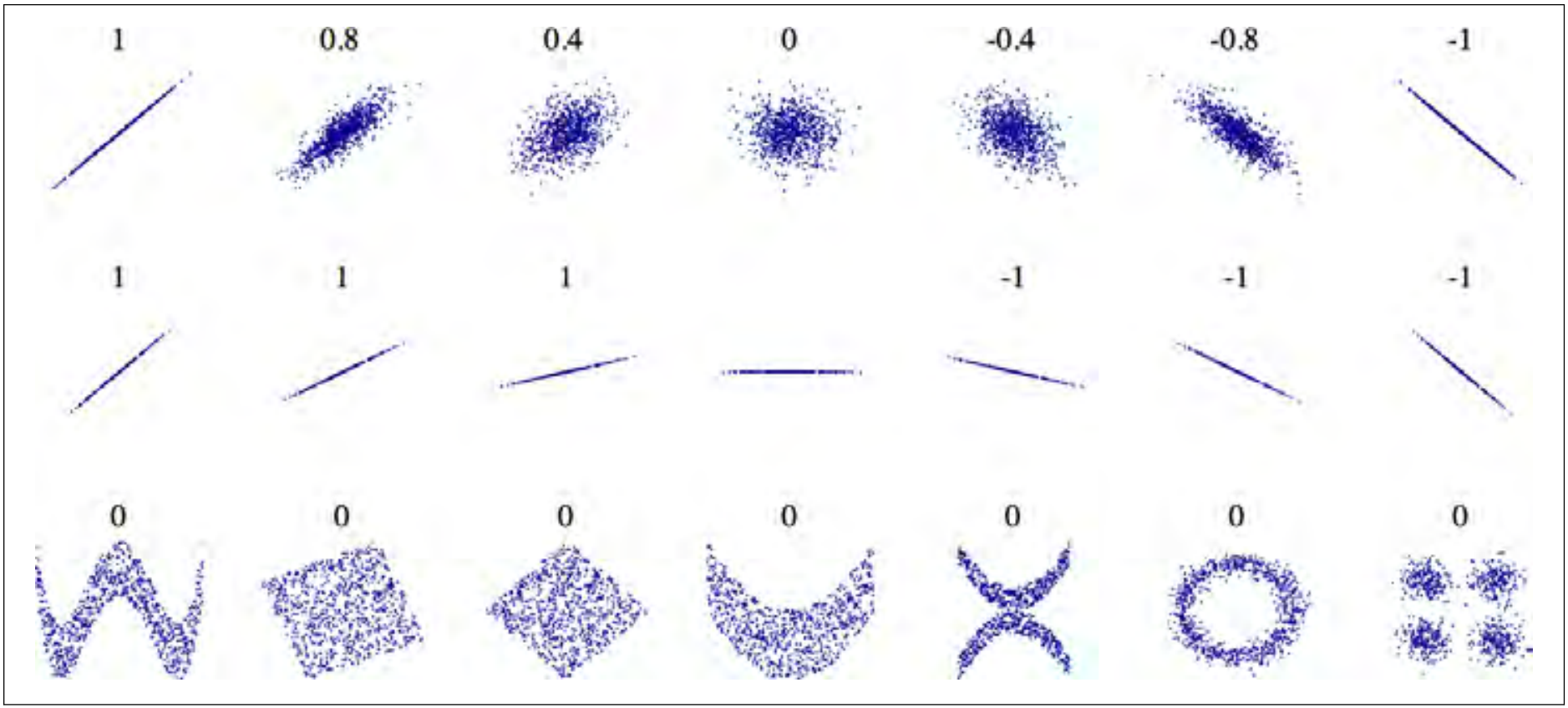


Figure 2-14. Standard correlation coefficient of various datasets (source: Wikipedia; public domain image)

Géron, Aurélien. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* O'Reilly Media, Inc.

```
from pandas.plotting import scatter_matrix

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

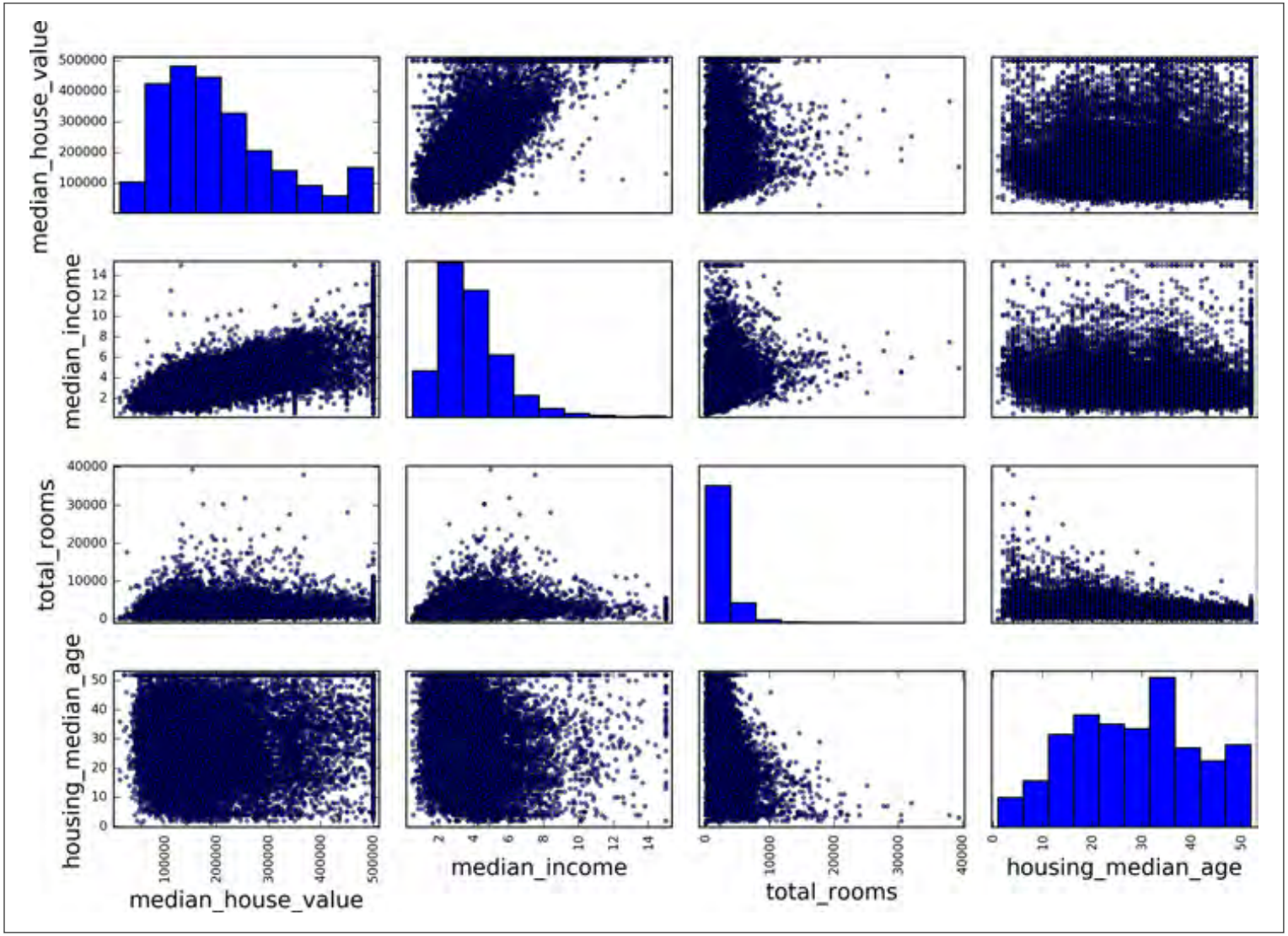


Figure 2-15. Scatter matrix

```
from pandas.plotting import scatter_matrix

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

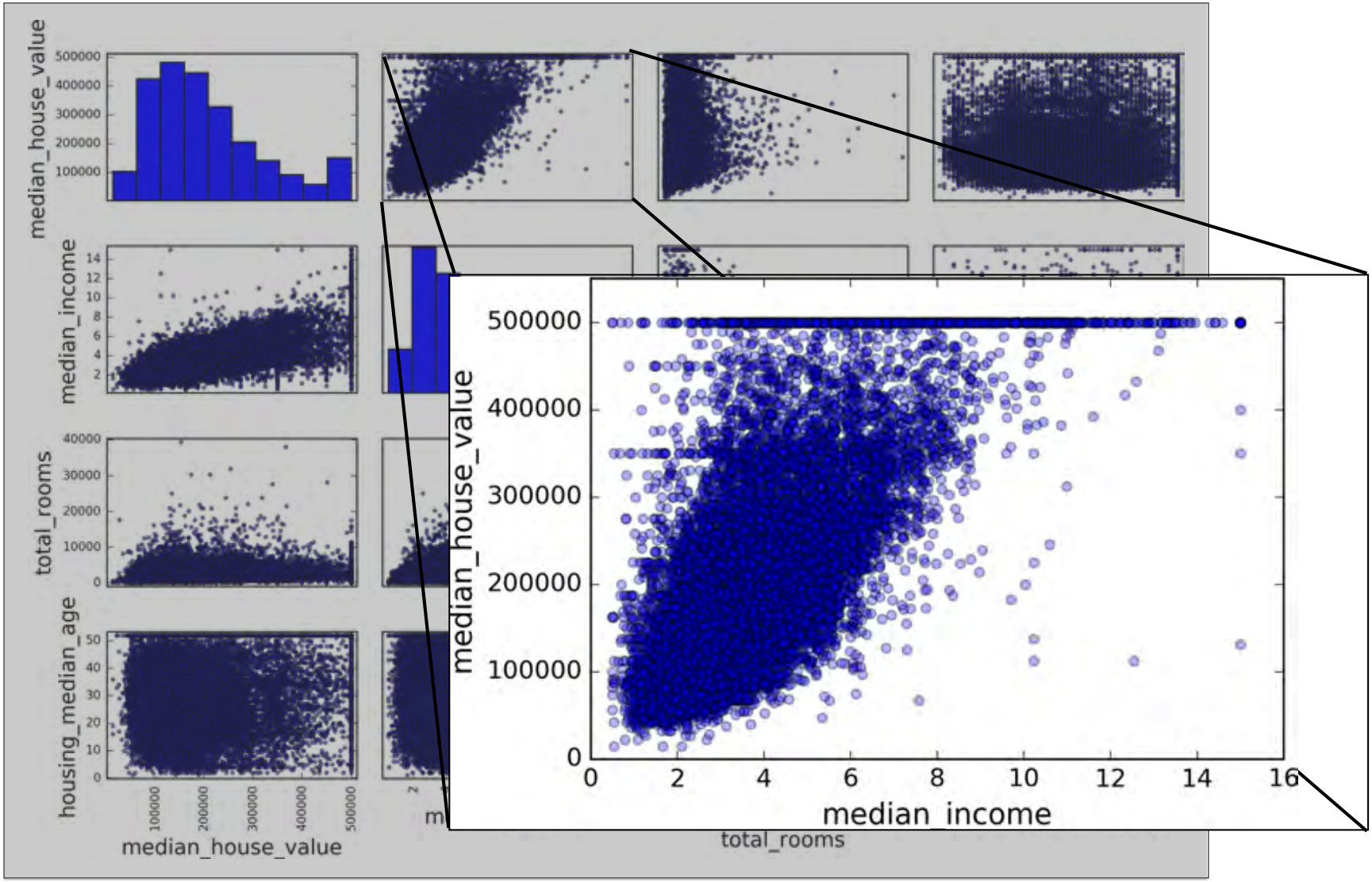


Figure 2-15. Scatter matrix

Activity

- **Replicate some of the plots from the California Housing Dataset with the Ames Housing Dataset**
  - https://www.kaggle.com/datasets/prevek18/ames-housing-dataset
  - ```
    import pandas as pd
    from sklearn.datasets import fetch_openml
    housing = fetch_openml(name="house_prices", as_frame=True)
    ```
- **Be sure to add Sale Price to the main dataframe**
  - ```
    housing_df['SalePrice'] = housing.target
    ```
- **Generate a histogram for all of the data columns**
- **Generate a scatter matrix for the attributes "SalePrice", "LotArea", “1stFlrSF", "YearBuilt"**

Also check out: https://pandas.pydata.org/pandas-docs/stable/user_guide/visualization.html

Project 1

- **Due June 13**
- **Start exploring potential datasets**
  - kaggle.com
  - archive.ics.uci.edu/ml/datasets.php
  - libguides.nypl.org/eresources
  - opendata.cityofnewyork.us/data/
- **The data set will need to be labeled as you are going to use it for both supervised and unsupervised learning tasks**

Reading for next class

- Ch 2: End-to-End Machine Learning Project. in Géron, Aurélien. (2019). Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow' O'Reilly Media, Inc. 33–66

DataCamp for next class

- Streamlined Data Ingestion with pandas (Required)