Datasets

Prof.Dr. Bahadır AKTUĞ
Machine Learning with Python

**Compiled from sources given in the references.*

Datasets

- Sci-kit Learn provides built-in datasets called «toy datasets»
- Sci-kit Learn also provides utility functions to retrieve several popular datasets.

Toy Datasets

load_iris(*[, return_X_y, as_frame])	Load and return the iris dataset (classification).
load_diabetes(*[, return_X_y, as_frame])	Load and return the diabetes dataset (regression).
load_digits(*[, n_class, return_X_y, as_frame])	Load and return the digits dataset (classification).
load_linnerud(*[, return_X_y, as_frame])	Load and return the physical excercise linnerud dataset.
load_wine(*[, return_X_y, as_frame])	Load and return the wine dataset (classification).
load_breast_cancer(*[, return_X_y, as_frame])	Load and return the breast cancer wisconsin dataset (classification).

load_boston(*[, return_X_y])	Load and return the boston house-prices dataset (regression).

IRIS Dataset

- The Iris flower data set or Fisher's Iris data set is a multivariate data set.
- This data sets consist of the attributes of 3 different types of irises

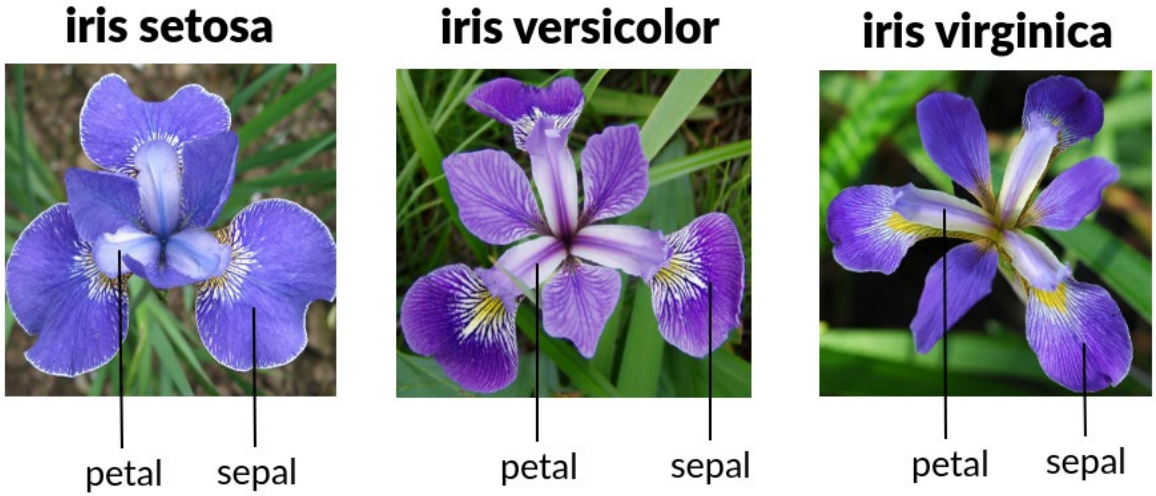
Iris setosa

Iris versicolor

Iris virginica

IRIS Dataset

- The data consists of 150x5 numpy.ndarray
- The rows being the samples and the columns being:
  - Sepal Length,
  - Sepal Width,
  - Petal Length
  - Petal Width
  - Species



Iris Plants

Number of Instances:	150 (50 in each of three classes)
Number of Attributes:	4 numeric, predictive attributes and the class

Attribute Information:	• sepal length in cm
	• sepal width in cm
	• petal length in cm
	• petal width in cm
	• **class:**
	  ◦ Iris-Setosa
	  ◦ Iris-Versicolour
	  ◦ Iris-Virginica

Summary Statistics:	

sepal length:	4.3	7.9	5.84	0.83	0.7826
sepal width:	2.0	4.4	3.05	0.43	-0.4194
petal length:	1.0	6.9	3.76	1.76	0.9490 (high!)
petal width:	0.1	2.5	1.20	0.76	0.9565 (high!)

Missing Attribute Values:	None
Class Distribution:	33.3% for each of 3 classes.
Creator:	R.A. Fisher
Donor:	Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
Date:	July, 1988

IRIS Dataset

- We can download and examine the data set from numerous sources.
- Since the dataset is highly popular for taxonomy problems, several python libraries (e.g. sklearn, seaborn) have it already built-in.
- We'll learn about sklearn later in this class. Seaborn can also be used to load iris dataset.

```
import seaborn as sns
iris = sns.load_dataset('iris')
iris.head()
```

```
from sklearn.datasets import load_iris
dataLoaded = load_iris()
```

IRIS Dataset

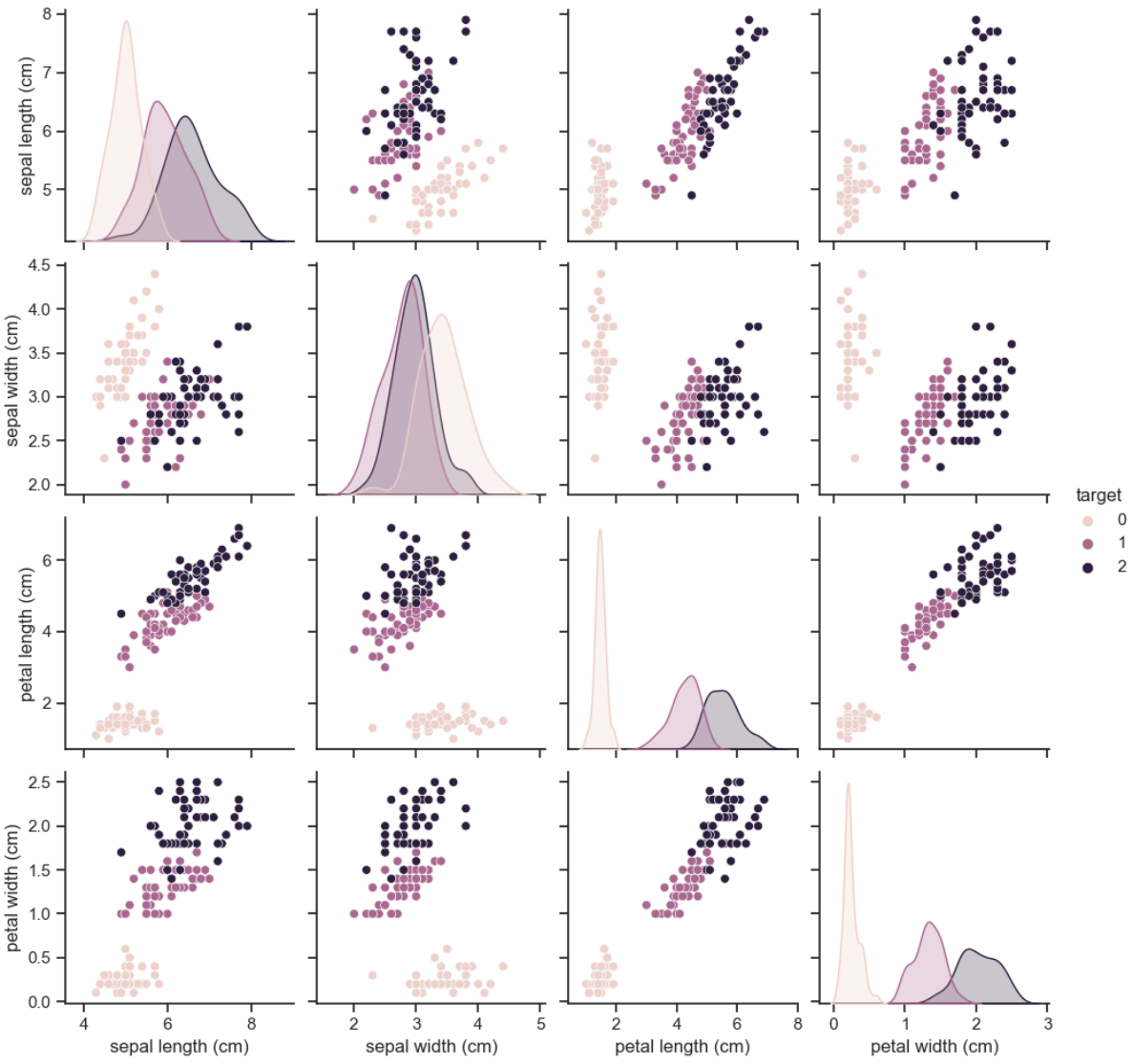
- The seaborn will load iris data as a Pandas dataframe.
- The dataframe consists of 150 rows x 5 columns
- We can check the data size and content by typing:

```
>>> iris.shape
(150, 5)
>>> type(iris)
<class 'pandas.core.frame.DataFrame'>
```

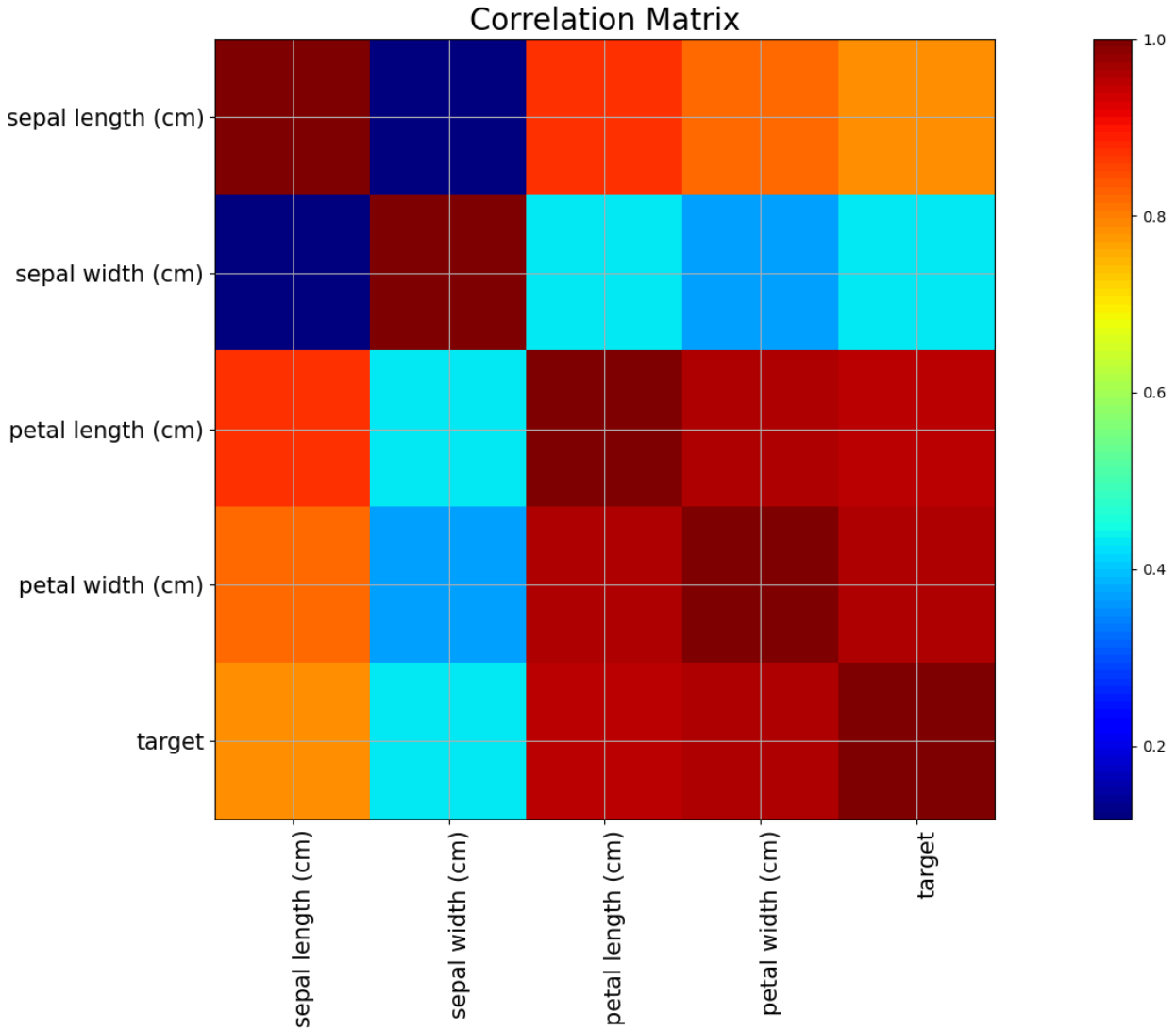
- Showing the first five rows:

```
>>> iris.head()
   sepal_length  sepal_width  petal_length  petal_width species
0           5.1          3.5           1.4          0.2  setosa
1           4.9          3.0           1.4          0.2  setosa
2           4.7          3.2           1.3          0.2  setosa
3           4.6          3.1           1.5          0.2  setosa
4           5.0          3.6           1.4          0.2  setosa
>>>
```

IRIS Dataset



IRIS Dataset

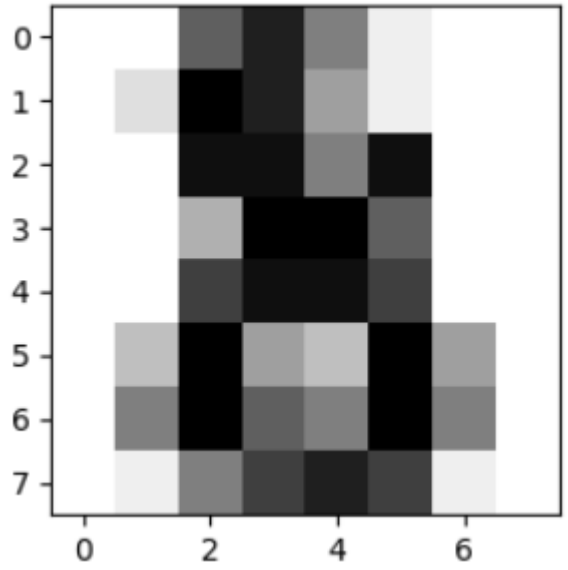


Handwritten Digits

Number of Instances:	1797
Number of Attributes:	64
Attribute Information:	8x8 image of integer pixels in the range 0..16.
Missing Attribute Values:	None
Creator:	5. Alpaydin (alpaydin '@' boun.edu.tr)
Date:	July; 1998

Handwritten Digits

This dataset is made up of 1797 8x8 images. Each image, like the one shown next, is of a hand-written digit.



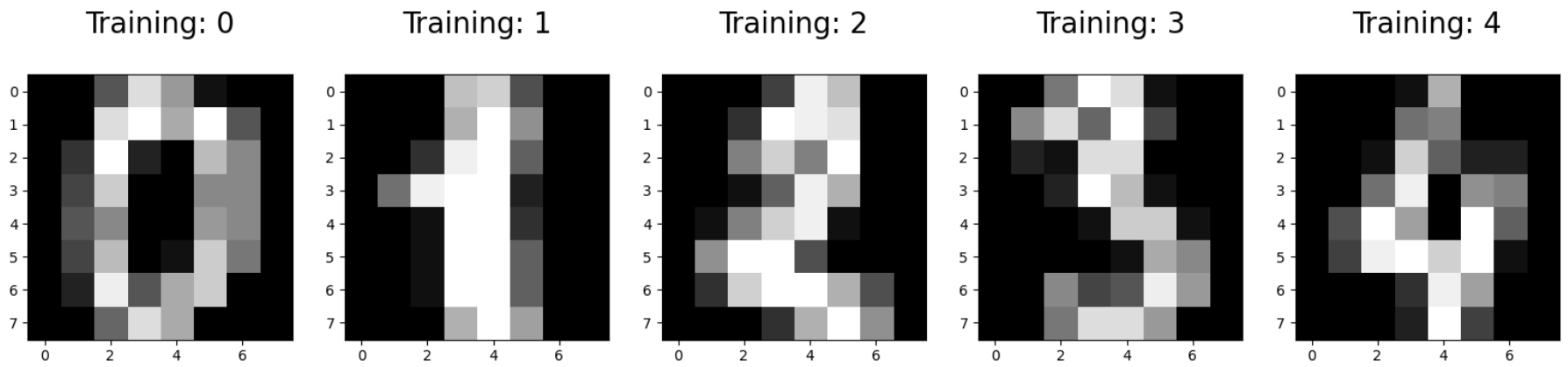

```
from sklearn import datasets

import matplotlib.pyplot as plt

# Load the digits dataset
digits = datasets.load_digits()

# Display the last digit
plt.figure(1, figsize=(3, 3))
plt.imshow(digits.images[-1], cmap=plt.cm.gray_r, interpolation="nearest")
plt.show()
```

Handwritten Digits




```
>>> print(digits.data)
[[ 0.   0.   5. ...   0.   0.   0.]
 [ 0.   0.   0. ...  10.   0.   0.]
 [ 0.   0.   0. ...  16.   9.   0.]
 ...
 [ 0.   0.   1. ...   6.   0.   0.]
 [ 0.   0.   2. ...  12.   0.   0.]
 [ 0.   0.  10. ...  12.   1.   0.]]
```

```
>>> digits.target
array([0, 1, 2, ..., 8, 9, 8])
```

Wine recognition dataset

Number of Instances:	178 (50 in each of three classes)
Number of Attributes:	13 numeric, predictive attributes and the class
Attribute Information:	• Alcohol • Malic acid • Ash • Alcalinity of ash • Magnesium • Total phenols • Flavanoids • Nonflavanoid phenols • Proanthocyanins • Color intensity • Hue • OD280/OD315 of diluted wines • Proline

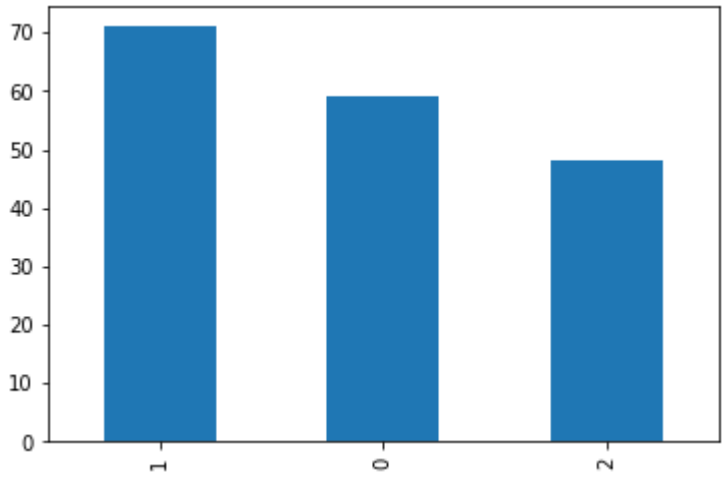
- **class:**
  - class_0
  - class_1
  - class_2

Alcohol:	11.0	14.8	13.0	0.8
Malic Acid:	0.74	5.80	2.34	1.12
Ash:	1.36	3.23	2.36	0.27
Alcalinity of Ash:	10.6	30.0	19.5	3.3
Magnesium:	70.0	162.0	99.7	14.3
Total Phenols:	0.98	3.88	2.29	0.63
Flavanoids:	0.34	5.08	2.03	1.00
Nonflavanoid Phenols:	0.13	0.66	0.36	0.12
Proanthocyanins:	0.41	3.58	1.59	0.57
Colour Intensity:	1.3	13.0	5.1	2.3
Hue:	0.48	1.71	0.96	0.23
OD280/OD315 of diluted wines:	1.27	4.00	2.61	0.71
Proline:	278	1680	746	315

Missing Attribute Values:	None
Class Distribution:	class_0 (59), class_1 (71), class_2 (48)
Creator:	R.A. Fisher
Donor:	Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
Date:	July, 1988

Wine recognition dataset

```
RangeIndex: 178 entries, 0 to 177
Data columns (total 14 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   alcohol                       178 non-null    float64
 1   malic_acid                    178 non-null    float64
 2   ash                           178 non-null    float64
 3   alcalinity_of_ash             178 non-null    float64
 4   magnesium                     178 non-null    float64
 5   total_phenols                 178 non-null    float64
 6   flavanoids                    178 non-null    float64
 7   nonflavanoid_phenols          178 non-null    float64
 8   proanthocyanins               178 non-null    float64
 9   color_intensity               178 non-null    float64
 10  hue                           178 non-null    float64
 11  od280/od315_of_diluted_wines  178 non-null    float64
 12  proline                       178 non-null    float64
 13  target                        178 non-null    float64
dtypes: float64(14)
memory usage: 19.6 KB
```



Breast cancer Wisconsin (diagnostic) dataset

Number of Instances:	569
Number of Attributes:	30 numeric, predictive attributes and the class
Attribute Information:	• radius (mean of distances from center to points on the perimeter) • texture (standard deviation of gray-scale values) • perimeter • area • smoothness (local variation in radius lengths) • compactness (perimeter^2 / area - 1.0) • concavity (severity of concave portions of the contour) • concave points (number of concave portions of the contour) • symmetry • fractal dimension ("coastline approximation" - 1)

radius (mean):	6.981	28.11
texture (mean):	9.71	39.28
perimeter (mean):	43.79	188.5
area (mean):	143.5	2501.0
smoothness (mean):	0.053	0.163
compactness (mean):	0.019	0.345
concavity (mean):	0.0	0.427
concave points (mean):	0.0	0.201
symmetry (mean):	0.106	0.304
fractal dimension (mean):	0.05	0.097
radius (standard error):	0.112	2.873
texture (standard error):	0.36	4.885
perimeter (standard error):	0.757	21.98
area (standard error):	6.802	542.2
smoothness (standard error):	0.002	0.031
compactness (standard error):	0.002	0.135
concavity (standard error):	0.0	0.396
concave points (standard error):	0.0	0.053
symmetry (standard error):	0.008	0.079
fractal dimension (standard error):	0.001	0.03
radius (worst):	7.93	36.04
texture (worst):	12.02	49.54
perimeter (worst):	50.41	251.2
area (worst):	185.2	4254.0
smoothness (worst):	0.071	0.223
compactness (worst):	0.027	1.058
concavity (worst):	0.0	1.252
concave points (worst):	0.0	0.291
symmetry (worst):	0.156	0.664
fractal dimension (worst):	0.055	0.208

- **class:**
  - WDBC-Malignant
  - WDBC-Benign

Missing Attribute Values:	None
Class Distribution:	212 - Malignant, 357 - Benign
Creator:	Dr. William H. Wolberg, W. Nick Street, Olvi L. Mangasarian
Donor:	Nick Street
Date:	November, 1995

Breast cancer Wisconsin (diagnostic) dataset

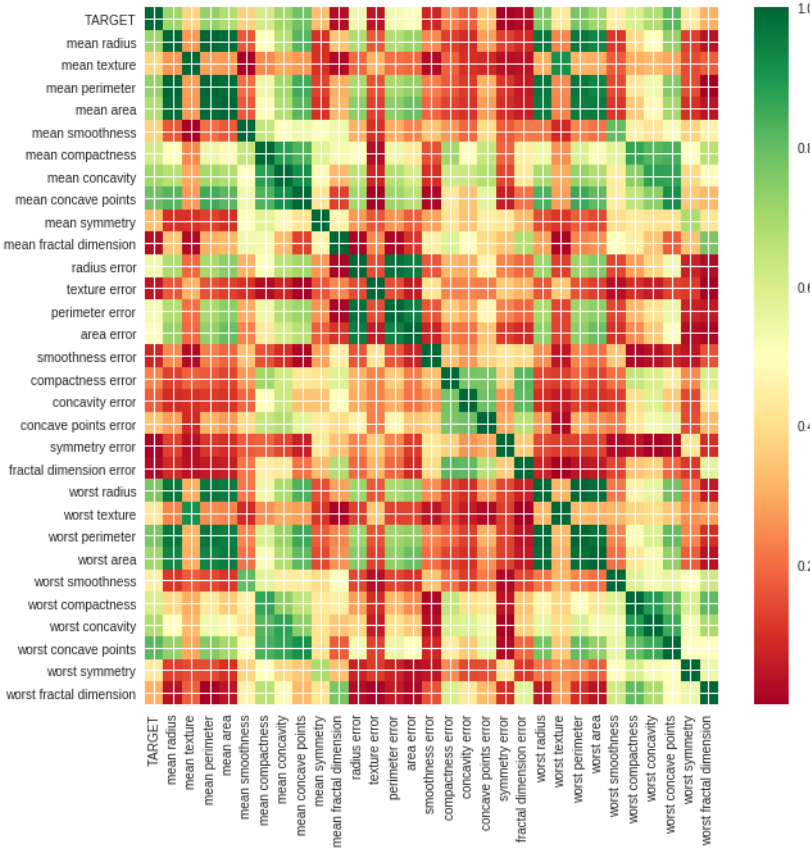
	COLUMN	INFORMATION
	1	ID
Mean	2	Radius
	3	Texture
	4	Perimeter
	5	Area
	6	Smoothness
	7	Compactness
	8	Concavity
	9	Concave_points
	10	Symmetry
	11	Fractal_dimension
Standard Error	12	Radius_SE
	13	Texture_SE
	14	Perimeter_SE
	15	Area_SE
	16	Smoothness_SE
	17	Compactness_SE
	18	Concavity_SE
	19	Concave_points_SE
	20	Symmetry_SE
	21	Fractal_dimension_SE
Worst or largest (mean of the three largest values)	22	W_Radius
	23	W_Texture
	24	W_Perimeter
	25	W_Area
	26	W_Smoothness
	27	W_Compactness
	28	W_Concavity
	29	W_Concave_points
	30	W_Symmetry
	31	W_fractal_dimension
	32	Diagnosis: Bening/Malignant

Breast cancer wisconsin (diagnostic) dataset

Describe dataframe, first 6 columns:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness
count	569.0000	569.0000	569.000	569.0000	569.0000	569.0000
mean	14.1273	19.2896	91.969	654.8891	0.0964	0.1043
std	3.5240	4.3010	24.299	351.9141	0.0141	0.0528
min	6.9810	9.7100	43.790	143.5000	0.0526	0.0194
25%	11.7000	16.1700	75.170	420.3000	0.0864	0.0649
50%	13.3700	18.8400	86.240	551.1000	0.095	
75%	15.7800	21.8000	104.100	782.7000	0.105	
max	28.1100	39.2800	188.500	2501.0000	0.163	

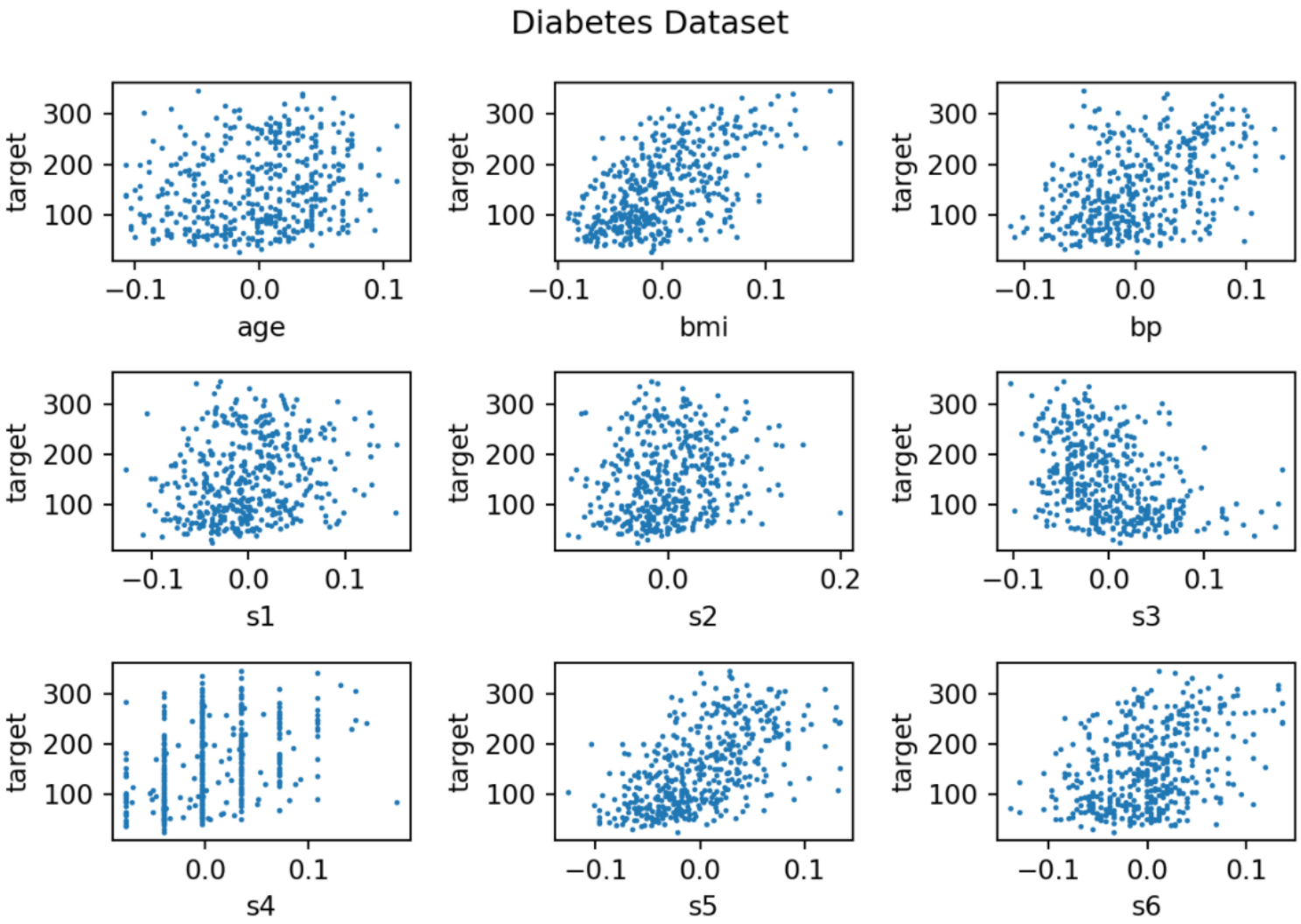
```
>>> from sklearn.datasets import load_breast_cancer
>>> data = load_breast_cancer()
>>> data.target[[10, 50, 85]]
array([0, 1, 0])
>>> list(data.target_names)
['malignant', 'benign']
```



Diabetes dataset

Number of Instances:	442
Number of Attributes:	First 10 columns are numeric predictive values
Target:	Column 11 is a quantitative measure of disease progression one year after baseline
Attribute Information:	• age age in years • sex • bmi body mass index • bp average blood pressure • s1 tc, T-Cells (a type of white blood cells) • s2 ldl, low-density lipoproteins • s3 hdl, high-density lipoproteins • s4 tch, thyroid stimulating hormone • s5 ltg, lamotrigine • s6 glu, blood sugar level

Diabetes dataset



Diabetes dataset

```
# Load the dataset
diabetes = datasets.load_diabetes(as_frame=True)
```

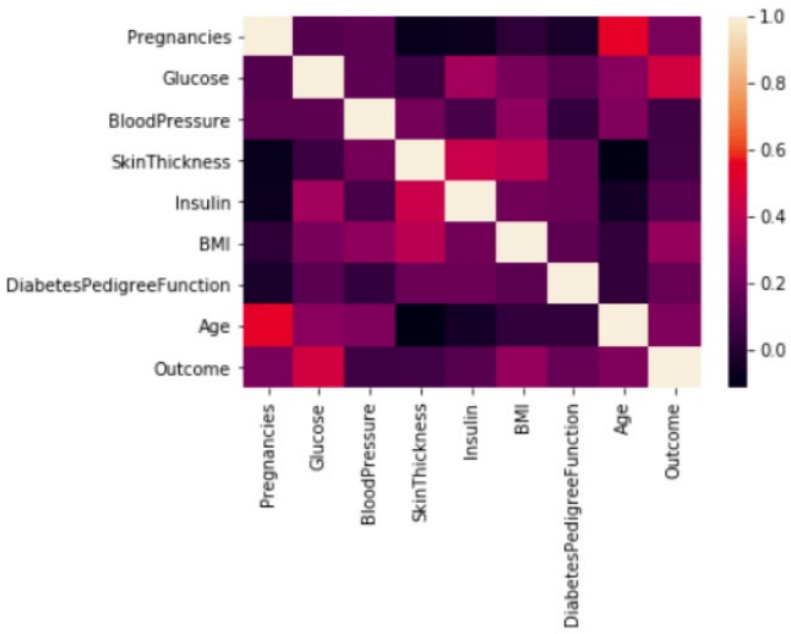
```
# Names of the 10 groups of data
print(diabetes['feature_names'])
```

```
# The 442 data points in each of the 10 groups of data, formatted as a 442x10 array
print(diabetes['data'])
```

```
##           age       sex       bmi  ...        s4        s5        s6
## 0    0.038076  0.050680  0.061696  ... -0.002592  0.019908 -0.017646
## 1   -0.001882 -0.044642 -0.051474  ... -0.039493 -0.068330 -0.092204
## 2    0.085299  0.050680  0.044451  ... -0.002592  0.002864 -0.025930
## 3   -0.089063 -0.044642 -0.011595  ...  0.034309  0.022692 -0.009362
## 4    0.005383 -0.044642 -0.036385  ... -0.002592 -0.031991 -0.046641
## ..        ...       ...       ...  ...       ...       ...       ...
## 437  0.041708  0.050680  0.019662  ... -0.002592  0.031193  0.007207
## 438 -0.005515  0.050680 -0.015906  ...  0.034309 -0.018118  0.044485
## 439  0.041708  0.050680 -0.015906  ... -0.011080 -0.046879  0.015491
## 440 -0.045472 -0.044642  0.039062  ...  0.026560  0.044528 -0.025930
## 441 -0.045472 -0.044642 -0.073030  ... -0.039493 -0.004220  0.003064
##
## [442 rows x 10 columns]
```

Diabetes Dataset

```
# The target data, namely a quantitative measure of disease progression one
# year after baseline
print(diabetes['target'][:20])
```




```
## 0     151.0
## 1      75.0
## 2     141.0
## 3     206.0
## 4     135.0
## 5      97.0
## 6     138.0
## 7      63.0
## 8     110.0
## 9     310.0
## 10    101.0
## 11     69.0
## 12    179.0
## 13    185.0
## 14    118.0
## 15    171.0
## 16    166.0
## 17    144.0
## 18     97.0
## 19    168.0
## Name: target, dtype: float64
```

Linnerrud Dataset

Number of Instances:	20
Number of Attributes:	3
Missing Attribute Values:	None

The Linnerud dataset is a multi-output regression dataset. It consists of three excercise (data) and three physiological (target) variables collected from twenty middle-aged men in a fitness club:

- ***physiological*** **- CSV containing 20 observations on 3 physiological variables:**
  Weight, Waist and Pulse.
- ***exercise*** **- CSV containing 20 observations on 3 exercise variables:**
  Chins, Situps and Jumps.

Boston House Prices

Data Set Characteristics:

Number of Instances:	506
Number of Attributes:	13 numeric/categorical predictive. Median Value (attribute 14) is usually the target.
Attribute Information (in order):	• CRIM per capita crime rate by town • ZN proportion of residential land zoned for lots over 25,000 sq.ft. • INDUS proportion of non-retail business acres per town • CHAS Charles River dummy variable (= 1 if tract bounds river; 0 otherwise) • NOX nitric oxides concentration (parts per 10 million) • RM average number of rooms per dwelling • AGE proportion of owner-occupied units built prior to 1940 • DIS weighted distances to five Boston employment centres • RAD index of accessibility to radial highways • TAX full-value property-tax rate per $10,000 • PTRATIO pupil-teacher ratio by town • B 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town • LSTAT % lower status of the population • MEDV Median value of owner-occupied homes in $1000's
Missing Attribute Values:	None
Creator:	Harrison, D. and Rubinfeld, D.L.

fetch_olivetti_faces(*[, data_home, …])	Load the Olivetti faces data-set from AT&T (classification).
fetch_20newsgroups(*[, data_home, subset, …])	Load the filenames and data from the 20 newsgroups dataset (classification).
fetch_20newsgroups_vectorized(*[, subset, …])	Load and vectorize the 20 newsgroups dataset (classification).
fetch_lfw_people(*[, data_home, funneled, …])	Load the Labeled Faces in the Wild (LFW) people dataset (classification).
fetch_lfw_pairs(*[, subset, data_home, …])	Load the Labeled Faces in the Wild (LFW) pairs dataset (classification).
fetch_covtype(*[, data_home, …])	Load the covertype dataset (classification).
fetch_rcv1(*[, data_home, subset, …])	Load the RCV1 multilabel dataset (classification).
fetch_kddcup99(*[, subset, data_home, …])	Load the kddcup99 dataset (classification).
fetch_california_housing(*[, data_home, …])	Load the California housing dataset (regression).

Inspecting Scikit-learn Datasets

- Information about data can be examined through several utility commands
- The utility «load» functions do not return data in tabular format as expected. Instead, they return data as a «bunch» (in sklearn terminology).
- Bunch object is basically a dictionary having keys as follows:
- dict_keys(['data', 'target', 'frame', 'target_names', 'DESCR', 'feature_names', 'filename'])

Inspecting Scikit-learn Datasets

- The most import ones are the «data» and «target»
- Data is the feature data which comprises the the attributes for each sample
- Target is dependent output variable consisting of values we want to predict
- Apart from these two, there are metadata which serve to describe the properties of the dataset
- «feature_names» are the names of the features (attributed)
- «target_names» is the name(s) of the target variable(s), in other words name(s) of the target column(s)
- «DESCR» is a description of the dataset
- «filename» is the path to the actual file of the data in CSV format. Loading data from For instance, the toy dataset Breast cancer Wisconsin can be inspected as:

Inspecting Scikit-learn Datasets

- Loading data from For instance, the toy dataset Breast cancer Wisconsin can be loaded and inspected as:

```
from sklearn import datasets
data = datasets.load_breast_cancer()
print(data.keys())
print(data.feature_names)
```

Output:

```
dict_keys(['data', 'target', 'frame', 'target_names', 'DESCR',
'feature_names', 'filename'])
['mean radius' 'mean texture' 'mean perimeter' 'mean area'
 'mean smoothness' 'mean compactness' 'mean concavity'
 'mean concave points' 'mean symmetry' 'mean fractal dimension'
 'radius error' 'texture error' 'perimeter error' 'area error'
 'smoothness error' 'compactness error' 'concavity error'
 'concave points error' 'symmetry error' 'fractal dimension error'
 'worst radius' 'worst texture' 'worst perimeter' 'worst area'
 'worst smoothness' 'worst compactness' 'worst concavity'
 'worst concave points' 'worst symmetry' 'worst fractal dimension']
```

Inspecting Scikit-learn Datasets

- Dataset Breast cancer Wisconsin

:Number of Instances: 569
:Number of Attributes: 30 numeric, predictive attributes and the class
:Attribute Information:
- radius (mean of distances from center to points on the perimeter)
- texture (standard deviation of gray-scale values)
- perimeter
- area
- smoothness (local variation in radius lengths)
- compactness (perimeter^2 / area - 1.0)
- concavity (severity of concave portions of the contour)
- concave points (number of concave portions of the contour)
- symmetry
- fractal dimension ("coastline approximation" - 1)

- class:
    - WDBC-Malignant
    - WDBC-Benign

Inspecting Scikit-learn Datasets

- If we prefer to convert sklearn bunch object into a dataframe (we usually do), a Pandas dataframe can be formed from a bunch as follows:

```
from sklearn import datasets
data = datasets.load_breast_cancer()
print(data.keys())
print(data.feature_names)
import pandas as pd
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = data.target
```

- This approach still requires original load data utility function

References

1 https://scikit-learn.org/
2 https://towardsdatascience.com/
3 McKinney, W. (2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython 2nd Edition.
4 Albon, C. (2018). Machine Learning with Python Cookbook: Practical Solutions from Preprocessing to Deep Learning
5 Géron, A. (2017). Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems 1st Edition
6 Müller, A. C., Guido, S. (2016). Introduction to Machine Learning with Python: A Guide for Data Scientists
7 Burkov, A. (2019). The Hundred-Page Machine Learning Book.
8 Burkov, A. (2020). Machine Learning Engineering.
9 Goodrich, M.T., Tamassia, R., Goldwasser, M.H. (2013). Data Structures and Algorithms in Python, Wiley.
10 https://towardsdatascience.com
11 https://docs.python.org/3/tutorial/
12 http://www.python-course.eu
13 https://developers.google.com/edu/python/
14 http://learnpythonthehardway.org/book/