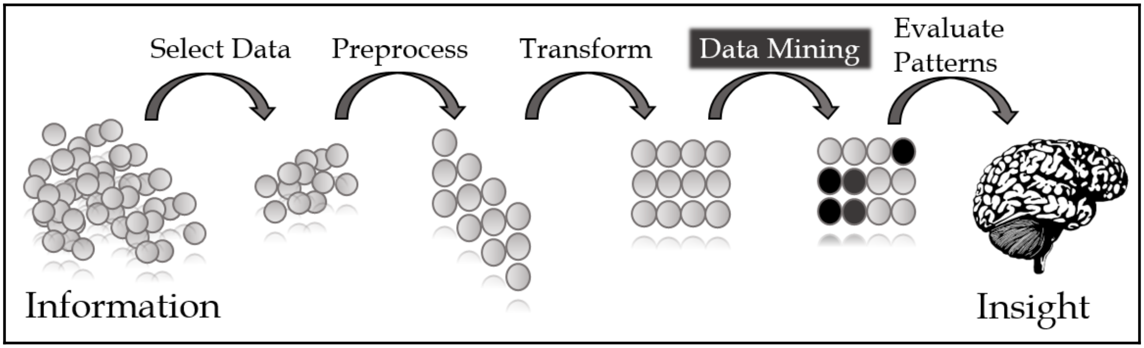
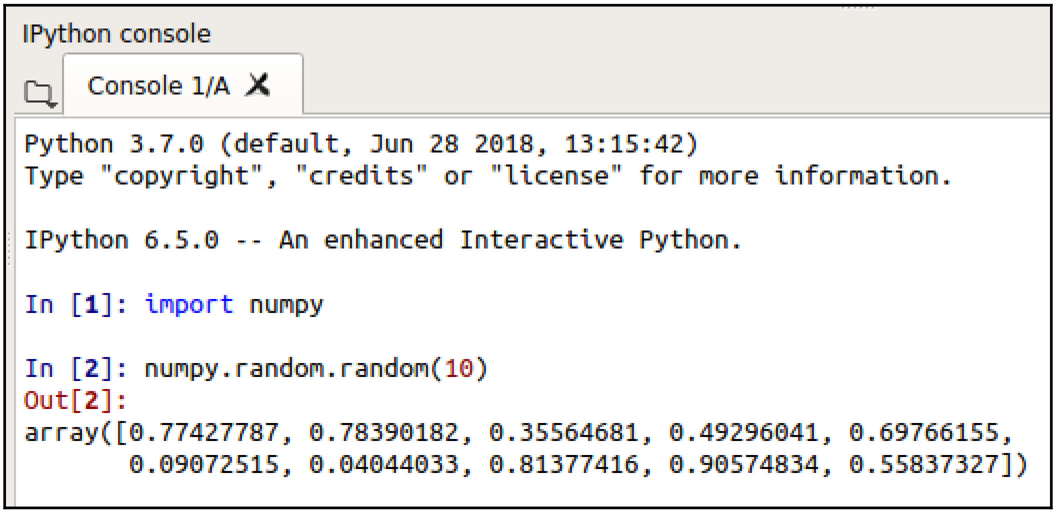
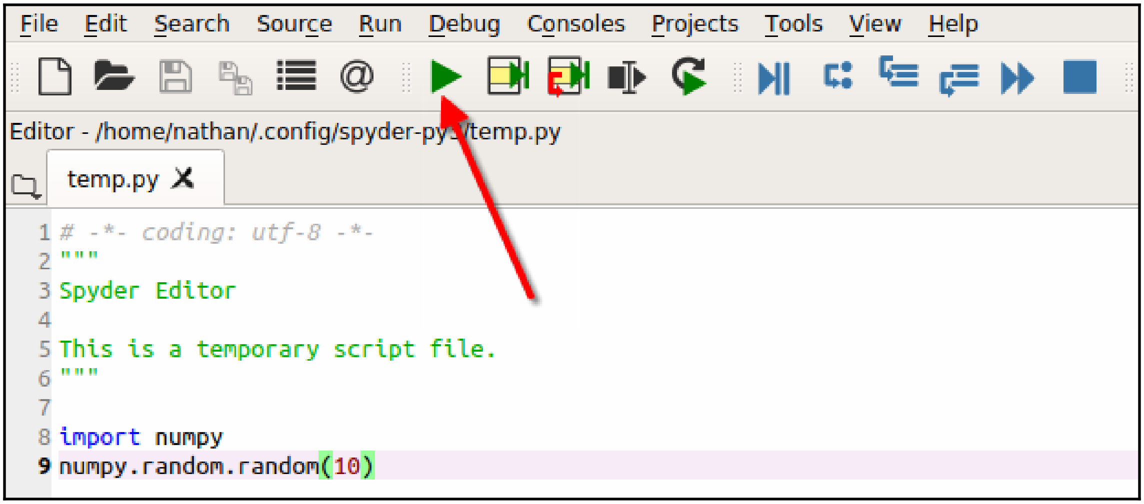
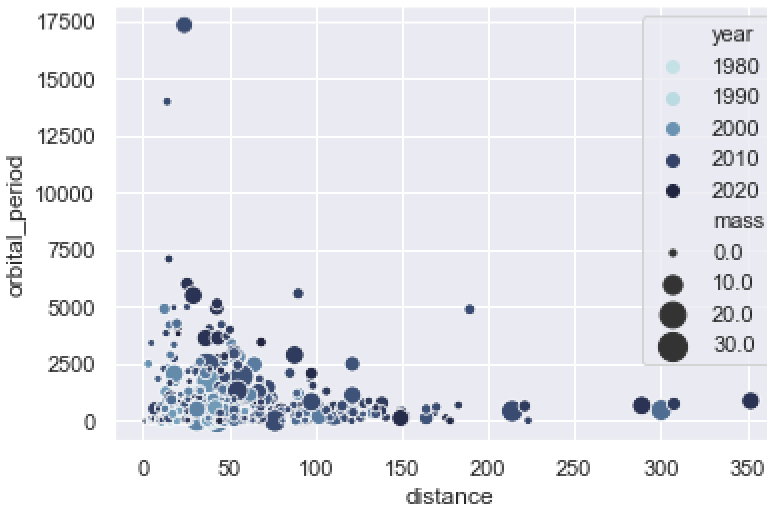
Chapter 1: Data Mining and Getting Started with Python Tools

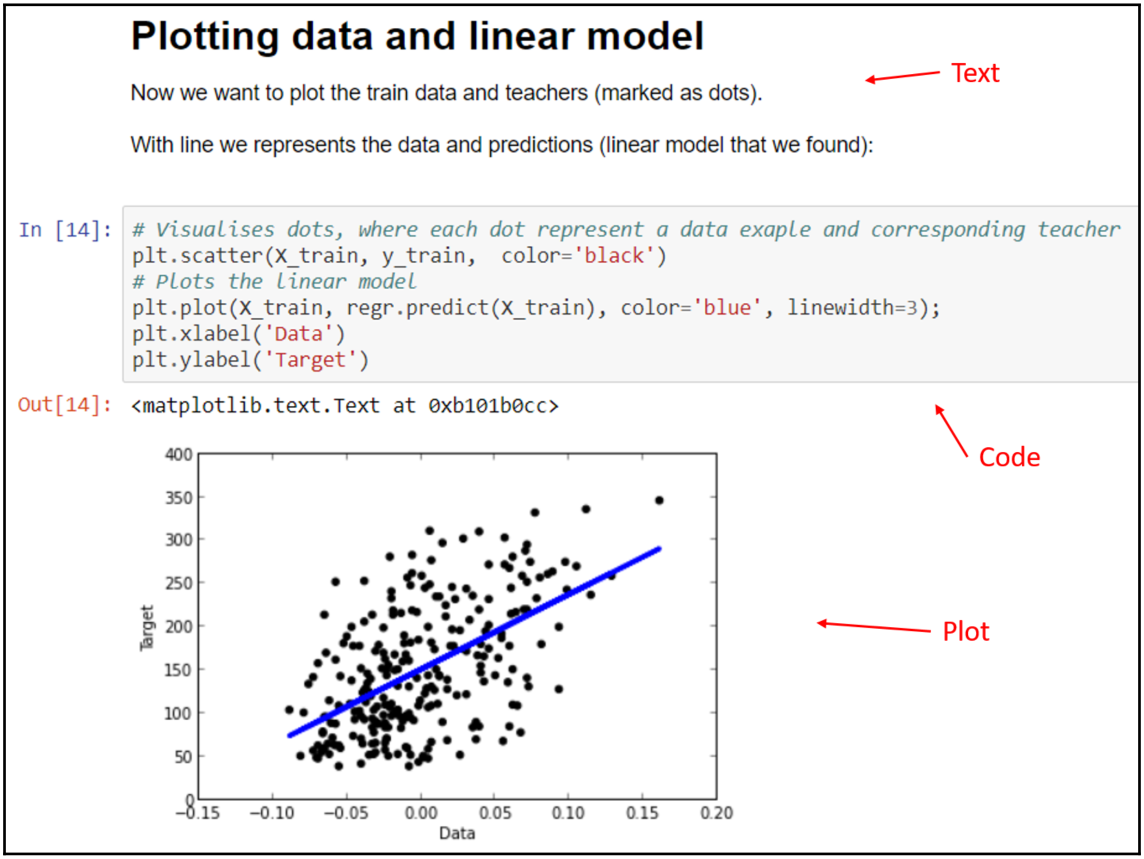


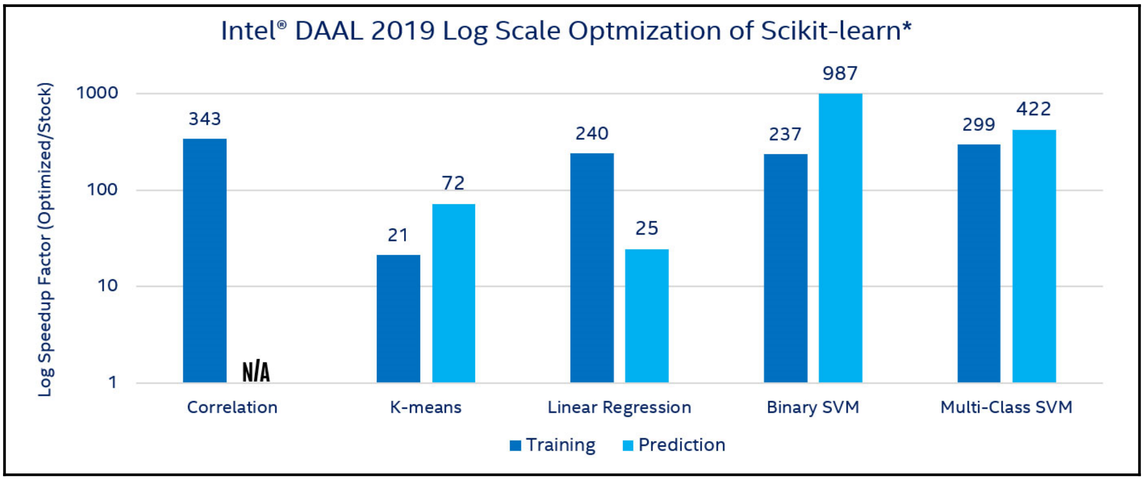



File Edit Search Source Run Debug Consoles Projects Tools View Help
Editor - /home/nathan/.config/spyder-py3/temp.py
temp.py
-*- coding: utf-8 -*-
"""
Spyder Editor
This is a temporary script file.
"""
import numpy
numpy.random.random(10)

```
In [5]: import seaborn as sns
   ...: sns.set()
   ...:
   ...: # Load the example iris dataset
   ...: planets = sns.load_dataset("planets")
   ...:
   ...: cmap = sns.cubehelix_palette(rot=-.2, as_cmap=True)
   ...: ax = sns.scatterplot(x="distance", y="orbital_period",
   ...:                      hue="year", size="mass",
   ...:                      palette=cmap, sizes=(10, 200),
   ...:                      data=planets)
   ...:
```




Plotting data and linear model
Now we want to plot the train data and teachers (marked as dots).
With line we represents the data and predictions (linear model that we found):
Text
In [14]:
# Visualises dots, where each dot represent a data exaple and corresponding teacher
plt.scatter(X_train, y_train, color='black')
# Plots the linear model
plt.plot(X_train, regr.predict(X_train), color='blue', linewidth=3);
plt.xlabel('Data')
plt.ylabel('Target')
Out[14]:
<matplotlib.text.Text at 0xb101b0cc>
Code
Target
Data
Plot


Intel® DAAL 2019 Log Scale Optmization of Scikit-learn*
Log Speedup Factor (Optimized/Stock)
1000
100
10
1
343
N/A
21
72
240
25
237
987
299
422
Correlation
K-means
Linear Regression
Binary SVM
Multi-Class SVM
Training
Prediction

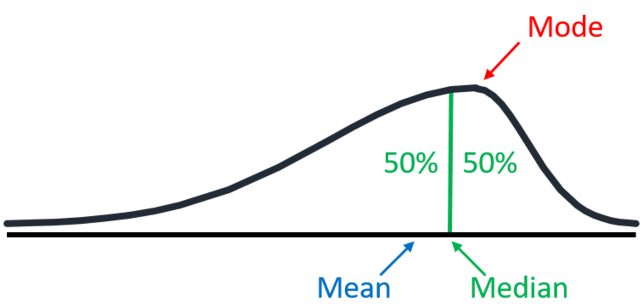
```
(base) nathan@nathan-ThinkPad-Twist:~$ conda info --envs
# conda environments:
#
base                  *  /home/nathan/anaconda3
idp                      /home/nathan/anaconda3/envs/idp
my_env                   /home/nathan/anaconda3/envs/my_env
```

```
(my_env) nathan@nathan-ThinkPad-Twist:~$ conda list
# packages in environment at /home/nathan/anaconda3/envs/my_env:
#
# Name                    Version                   Build  Channel
blas                      1.0                         mkl
ca-certificates           2018.03.07                    0
certifi                   2018.10.15               py37_0
intel-openmp              2019.0                      118
libedit                   3.1.20170329         h6b74fdf_2
libffi                    3.2.1                hd88cf55_4
libgcc-ng                 8.2.0                hdf63c60_1
libgfortran-ng            7.3.0                hdf63c60_0
libstdcxx-ng              8.2.0                hdf63c60_1
mkl                       2019.0                      118
mkl_fft                   1.0.6            py37h7dd41cf_0
mkl_random                1.0.1            py37h4414c95_1
ncurses                   6.1                  hf484d3e_0
numpy                     1.15.4           py37h1d66e8a_0
numpy-base                1.15.4           py37h81de0dd_0
openssl                   1.1.1                h7b6447c_0
pip                       18.1                     py37_0
python                    3.7.1                h0371630_3
readline                  7.0                  h7b6447c_5
setuptools                40.5.0                   py37_0
sqlite                    3.25.2               h7b6447c_0
tk                        8.6.8                hbc83047_0
wheel                     0.32.2                   py37_0
xz                        5.2.4                h14c3975_4
zlib                      1.2.11               ha838bed_2
```

Chapter 2: Basic Terminology and Our End-to-End Example

	X				Y
Person	**Age**	**Height**	**Weight**	**Training Hours/week**	**Long Jump**
Thomas	12	57.5	73.4	6.5	19.2
Jane	13	65.5	85.3	8.9	25.1
Vaughn	17	71.9	125.9	1.1	14.3
Vera	14	65.3	100.5	7.9	18.3
Vincent	18	70.1	110.7	10.5	21.1
Lei-Ann	12	52.3	70.4	0.5	10.6

Negative Skewed Distribution



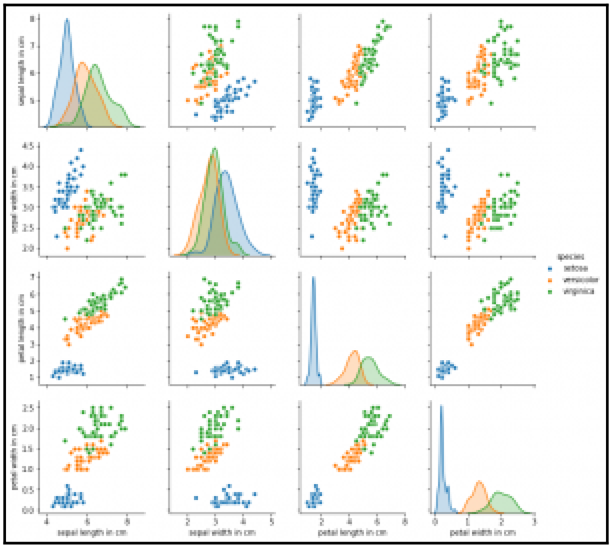
Mean

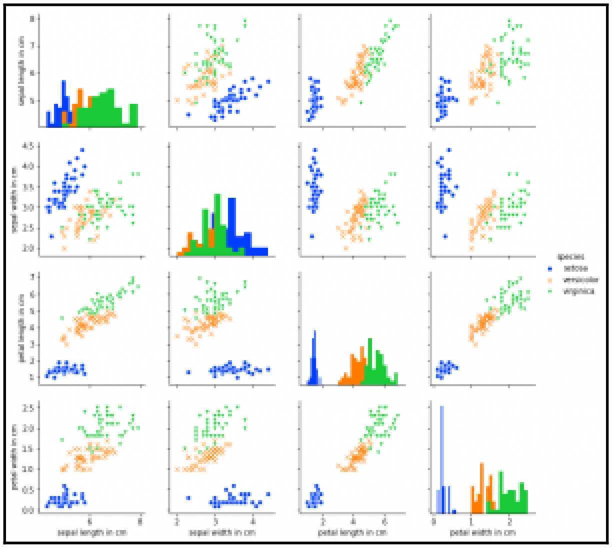
$$\bar{x} = \frac{x_1 + x_2 + \cdots + x_n}{n}$$

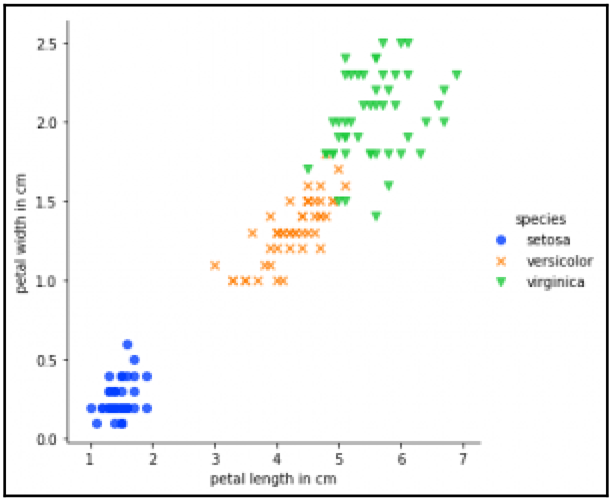
Standard Deviation

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}}$$

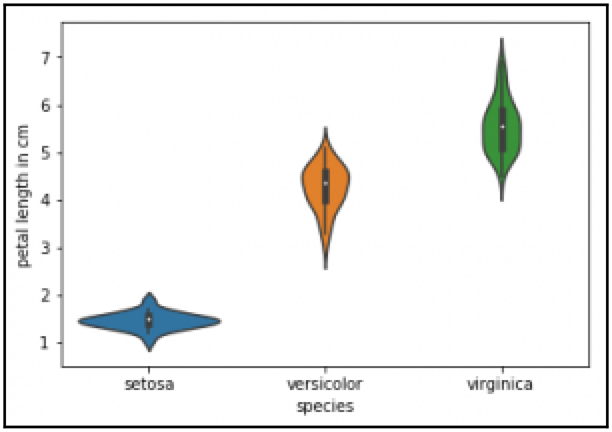
```
shape of data in (rows, columns) is (150, 5)
   sepal length in cm  sepal width in cm  petal length in cm  \
0                 5.1                3.5                 1.4
1                 4.9                3.0                 1.4
2                 4.7                3.2                 1.3
3                 4.6                3.1                 1.5
4                 5.0                3.6                 1.4

   petal width in cm species
0                0.2  setosa
1                0.2  setosa
2                0.2  setosa
3                0.2  setosa
4                0.2  setosa
                    count      mean       std  min  25%   50%  75%  max
sepal length in cm  150.0  5.843333  0.828066  4.3  5.1  5.80  6.4  7.9
sepal width in cm   150.0  3.054000  0.433594  2.0  2.8  3.00  3.3  4.4
petal length in cm  150.0  3.758667  1.764420  1.0  1.6  4.35  5.1  6.9
petal width in cm   150.0  1.198667  0.763161  0.1  0.3  1.30  1.8  2.5
```





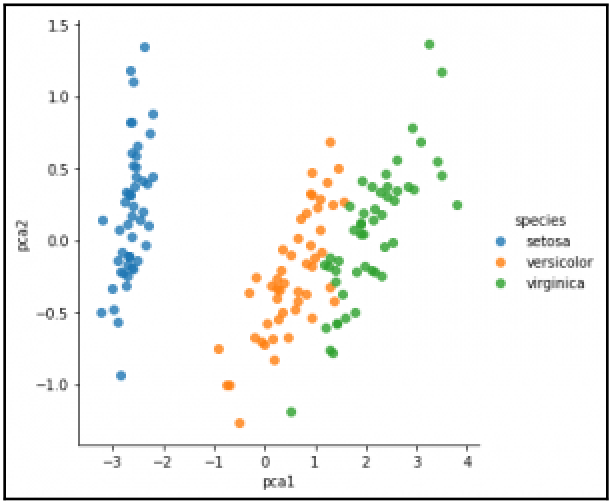


2.5
2.0
1.5
1.0
0.5
0.0
petal width in cm
1
2
3
4
5
6
7
petal length in cm
species
setosa
versicolor
virginica

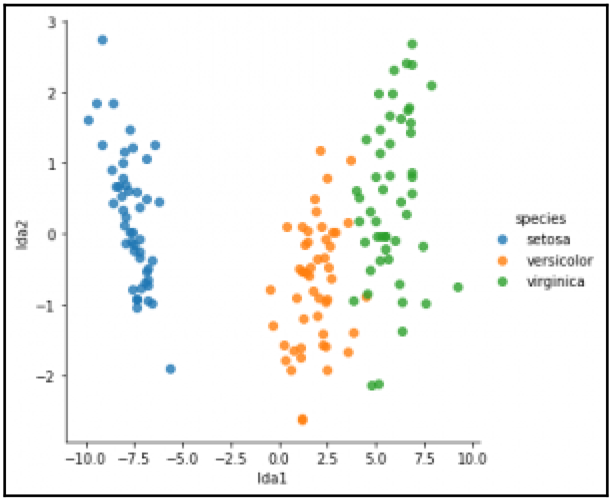


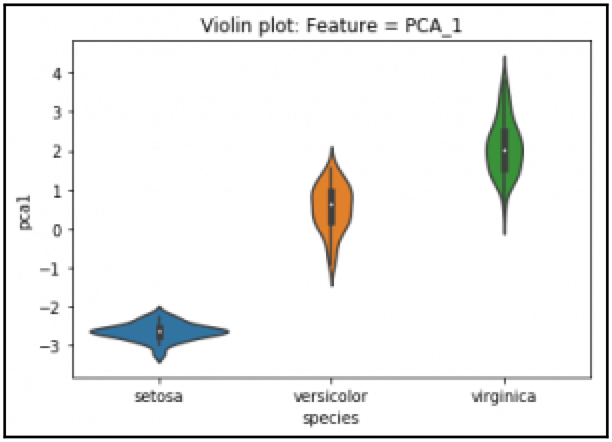
	pca1	pca2
0	-2.684207	0.326607
1	-2.715391	-0.169557
2	-2.889820	-0.137346
3	-2.746437	-0.311124
4	-2.728593	0.333925

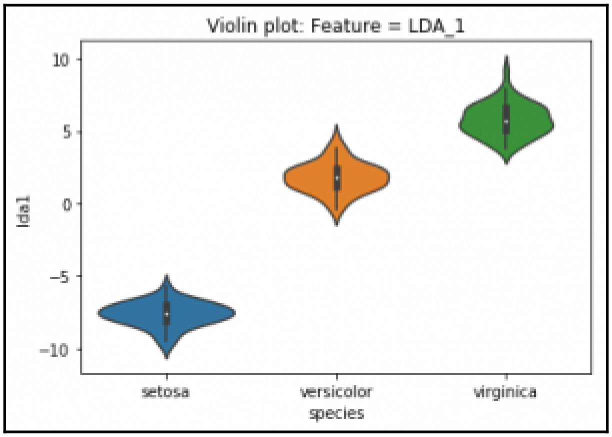
	pca1	pca2	species
0	-2.684207	0.326607	setosa
1	-2.715391	-0.169557	setosa
2	-2.889820	-0.137346	setosa
3	-2.746437	-0.311124	setosa
4	-2.728593	0.333925	setosa



	lda1	lda2	species
0	-8.084953	0.328454	setosa
1	-7.147163	-0.755473	setosa
2	-7.511378	-0.238078	setosa
3	-6.837676	-0.642885	setosa
4	-8.157814	0.540639	setosa








```
train set shape = (105, 3)
test set shape = (45, 3)
         lda1      lda2     species
81   0.598443 -1.923348  versicolor
133  3.809721 -0.934519   virginica
137  4.993563  0.184883   virginica
75   1.439522 -0.123147  versicolor
109  6.872871  2.694581   virginica
```

Chapter 3: Collecting, Exploring, and Visualizing Data

```
[(0, 0.00632, 18.0, 2.31, 0.0, 0.538, 6.575, 65.2, 4.09, 1.0, 296.0, 15.3, 396.9, 4.98, 24.0), (1, 0.02731, 0.0, 7.07, 0.0, 0.4
69, 6.421, 78.9, 4.9671, 2.0, 242.0, 17.8, 396.9, 9.14, 21.6), (2, 0.02729, 0.0, 7.07, 0.0, 0.469, 7.185, 61.1, 4.9671, 2.0, 24
2.0, 17.8, 392.83, 4.03, 34.7), (3, 0.03237, 0.0, 2.18, 0.0, 0.458, 6.998, 45.8, 6.0622, 3.0, 222.0, 18.7, 394.63, 2.94, 33.4),
(4, 0.06905, 0.0, 2.18, 0.0, 0.458, 7.147, 54.2, 6.0622, 3.0, 222.0, 18.7, 396.9, 5.33, 36.2)]
```

```
[(18.0,), (12.5,), (12.5,), (12.5,), (12.5,), (12.5,), (12.5,), (12.5,), (75.0,), (75.0,), (21.0,), (21.0,), (21.0,), (21.0,),
(75.0,), (90.0,), (85.0,), (100.0,), (25.0,), (25.0,), (25.0,), (25.0,), (25.0,), (25.0,), (17.5,), (80.0,), (80.0,), (12.5,),
(12.5,), (12.5,), (25.0,), (25.0,), (25.0,), (25.0,), (28.0,), (28.0,), (28.0,), (45.0,), (45.0,), (45.0,), (45.0,), (45.0,),
(45.0,), (60.0,), (60.0,), (80.0,), (80.0,), (80.0,), (80.0,), (95.0,), (95.0,), (82.5,), (82.5,), (95.0,), (95.0,), (30.0,),
(30.0,), (30.0,), (30.0,), (30.0,), (30.0,), (22.0,), (22.0,), (22.0,), (22.0,), (22.0,), (22.0,), (22.0,), (22.0,), (22.0,),
(22.0,), (80.0,), (80.0,), (90.0,), (20.0,), (20.0,), (20.0,), (20.0,), (20.0,), (20.0,), (20.0,), (20.0,), (20.0,), (20.0,),
(20.0,), (20.0,), (20.0,), (20.0,), (20.0,), (20.0,), (20.0,), (40.0,), (40.0,), (40.0,), (40.0,), (40.0,), (20.0,), (20.0,),
(20.0,), (20.0,), (90.0,), (90.0,), (55.0,), (80.0,), (52.5,), (52.5,), (52.5,), (80.0,), (80.0,), (80.0,), (70.0,), (70.0,),
(70.0,), (34.0,), (34.0,), (34.0,), (33.0,), (33.0,), (33.0,), (33.0,), (35.0,), (35.0,), (35.0,), (55.0,), (55.0,), (85.0,),
(80.0,), (40.0,), (40.0,), (60.0,), (60.0,), (90.0,), (80.0,), (80.0,)]
```

```
df.shape = (506, 15)
Sanity check with Pandas head():
   record     CRIM    ZN  INDUS  CHAS    NOX     RM   AGE     DIS  RAD    TAX  LSTAT  MEDV
0       0  0.00632  18.0   2.31   0.0  0.538  6.575  65.2  4.0900  1.0  296.0   4.98  24.0
1       1  0.02731   0.0   7.07   0.0  0.469  6.421  78.9  4.9671  2.0  242.0   9.14  21.6
2       2  0.02729   0.0   7.07   0.0  0.469  7.185  61.1  4.9671  2.0  242.0   4.03  34.7
3       3  0.03237   0.0   2.18   0.0  0.458  6.998  45.8  6.0622  3.0  222.0   2.94  33.4
4       4  0.06905   0.0   2.18   0.0  0.458  7.147  54.2  6.0622  3.0  222.0   5.33  36.2

Summarize with Pandas describe():
        count        mean         std        min         25%        50%         75%       max
record  506.0  252.500000  146.213884    0.00000  126.250000  252.50000  378.750000  505.0000
CRIM    506.0    3.593761    8.596783    0.00632    0.082045    0.25651    3.647423   88.9762
ZN      506.0   11.363636   23.322453    0.00000    0.000000    0.00000   12.500000  100.0000
INDUS   506.0   11.136779    6.860353    0.46000    5.190000    9.69000   18.100000   27.7400
CHAS    506.0    0.069170    0.253994    0.00000    0.000000    0.00000    0.000000    1.0000
NOX     506.0    0.554695    0.115878    0.38500    0.449000    0.53800    0.624000    0.8710
RM      506.0    6.284634    0.702617    3.56100    5.885500    6.20850    6.623500    8.7800
AGE     506.0   68.574901   28.148861    2.90000   45.025000   77.50000   94.075000  100.0000
DIS     506.0    3.795043    2.105710    1.12960    2.100175    3.20745    5.188425   12.1265
RAD     506.0    9.549407    8.707259    1.00000    4.000000    5.00000   24.000000   24.0000
TAX     506.0  408.237154  168.537116  187.00000  279.000000  330.00000  666.000000  711.0000
LSTAT   506.0   12.653063    7.141062    1.73000    6.950000   11.36000   16.955000   37.9700
MEDV    506.0   22.532806    9.197104    5.00000   17.025000   21.20000   25.000000   50.0000
```

	record	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	LSTAT	MEDV
0	0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	4.98	24.0
1	1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	9.14	21.6
2	2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	4.03	34.7
3	3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	2.94	33.4
4	4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	5.33	36.2

	count	mean	std	min	25%	50%	75%	max
record	506.0	252.500000	146.213884	0.00000	126.250000	252.50000	378.750000	505.0000
CRIM	506.0	3.593761	8.596783	0.00632	0.082045	0.25651	3.647423	88.9762
ZN	506.0	11.363636	23.322453	0.00000	0.000000	0.00000	12.500000	100.0000
INDUS	506.0	11.136779	6.860353	0.46000	5.190000	9.69000	18.100000	27.7400
CHAS	506.0	0.069170	0.253994	0.00000	0.000000	0.00000	0.000000	1.0000
NOX	506.0	0.554695	0.115878	0.38500	0.449000	0.53800	0.624000	0.8710
RM	506.0	6.284634	0.702617	3.56100	5.885500	6.20850	6.623500	8.7800
AGE	506.0	68.574901	28.148861	2.90000	45.025000	77.50000	94.075000	100.0000
DIS	506.0	3.795043	2.105710	1.12960	2.100175	3.20745	5.188425	12.1265
RAD	506.0	9.549407	8.707259	1.00000	4.000000	5.00000	24.000000	24.0000
TAX	506.0	408.237154	168.537116	187.00000	279.000000	330.00000	666.000000	711.0000
LSTAT	506.0	12.653063	7.141062	1.73000	6.950000	11.36000	16.955000	37.9700
MEDV	506.0	22.532806	9.197104	5.00000	17.025000	21.20000	25.000000	50.0000

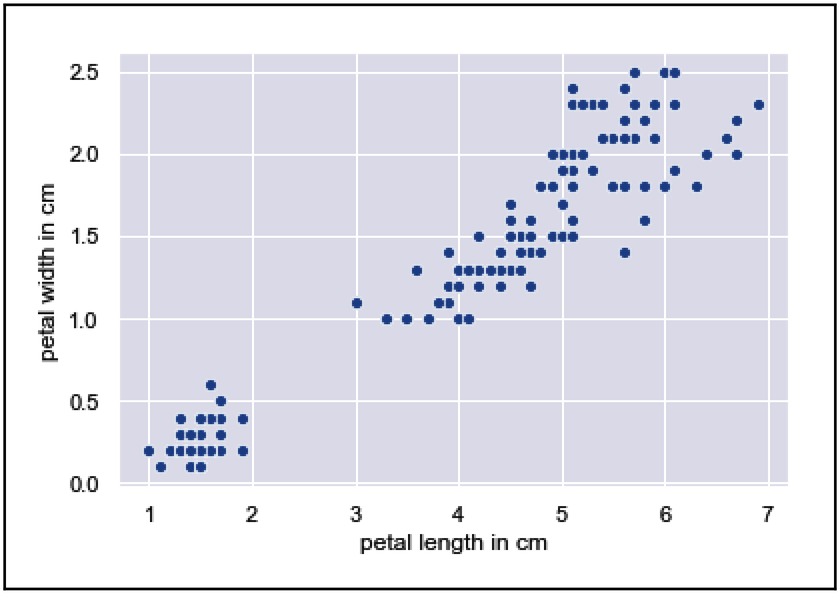
```
CRIM          0.25651
ZN            0.00000
INDUS         9.69000
CHAS          0.00000
NOX           0.53800
RM            6.20850
AGE          77.50000
DIS           3.20745
RAD           5.00000
TAX         330.00000
PTRATIO      19.05000
B           391.44000
LSTAT        11.36000
MEDV         21.20000
dtype: float64
```

```
CRIM         0
ZN           1
INDUS      195
CHAS         0
NOX        286
RM         365
AGE         41
DIS        372
RAD          0
TAX        353
PTRATIO    196
B          450
LSTAT      161
MEDV       398
dtype: int64
```

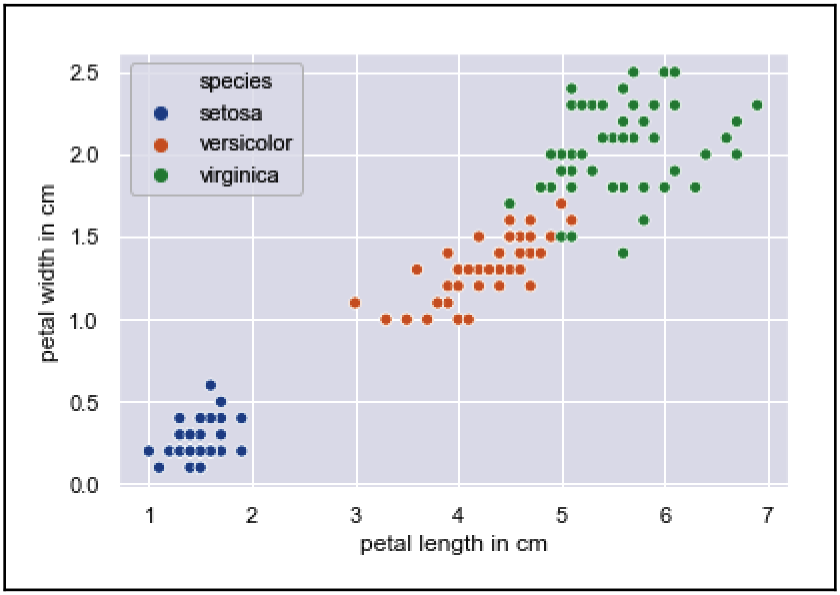
```
           CRIM     ZN  INDUS  CHAS     NOX     RM   AGE     DIS  RAD    TAX  LSTAT  MEDV
record
57      0.01432  100.0   1.32   0.0  0.4110  6.816  40.5  8.3248  5.0  256.0   3.95  31.6
204     0.02009   95.0   2.68   0.0  0.4161  8.034  31.9  5.1180  4.0  224.0   2.88  50.0
203     0.03510   95.0   2.68   0.0  0.4161  7.853  33.2  5.1180  4.0  224.0   3.81  48.5
200     0.01778   95.0   1.47   0.0  0.4030  7.135  13.9  7.6534  3.0  402.0   4.45  32.9
199     0.03150   95.0   1.47   0.0  0.4030  6.975  15.3  7.6534  3.0  402.0   4.56  34.9
```

```
           CRIM     ZN  INDUS  CHAS     NOX     RM   AGE     DIS  RAD    TAX  LSTAT  MEDV
record
57      0.01432  100.0   1.32   0.0  0.4110  6.816  40.5  8.3248  5.0  256.0   3.95  31.6
204     0.02009   95.0   2.68   0.0  0.4161  8.034  31.9  5.1180  4.0  224.0   2.88  50.0
203     0.03510   95.0   2.68   0.0  0.4161  7.853  33.2  5.1180  4.0  224.0   3.81  48.5
200     0.01778   95.0   1.47   0.0  0.4030  7.135  13.9  7.6534  3.0  402.0   4.45  32.9
199     0.03150   95.0   1.47   0.0  0.4030  6.975  15.3  7.6534  3.0  402.0   4.56  34.9
```

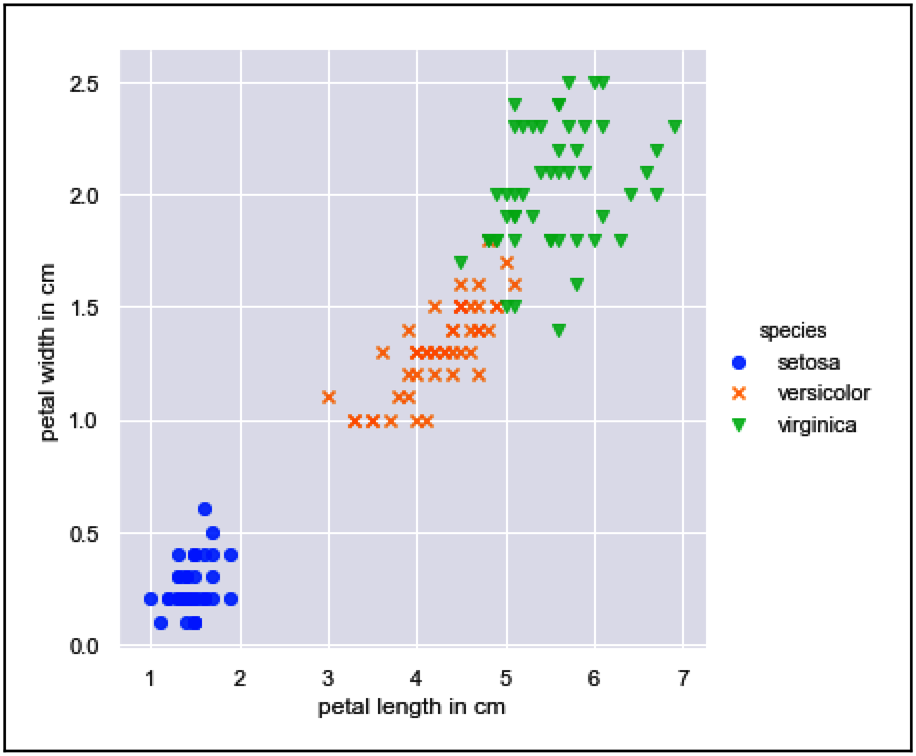
```
   record     CRIM    ZN  INDUS  CHAS    NOX     RM   AGE     DIS  RAD    TAX  LSTAT  MEDV
0       0  0.00632  18.0   2.31   0.0  0.538  6.575  65.2  4.0900  1.0  296.0   4.98  24.0
1       1  0.02731   0.0   7.07   0.0  0.469  6.421  78.9  4.9671  2.0  242.0   9.14  21.6
2       2  0.02729   0.0   7.07   0.0  0.469  7.185  61.1  4.9671  2.0  242.0   4.03  34.7
3       3  0.03237   0.0   2.18   0.0  0.458  6.998  45.8  6.0622  3.0  222.0   2.94  33.4
4       4  0.06905   0.0   2.18   0.0  0.458  7.147  54.2  6.0622  3.0  222.0   5.33  36.2
```

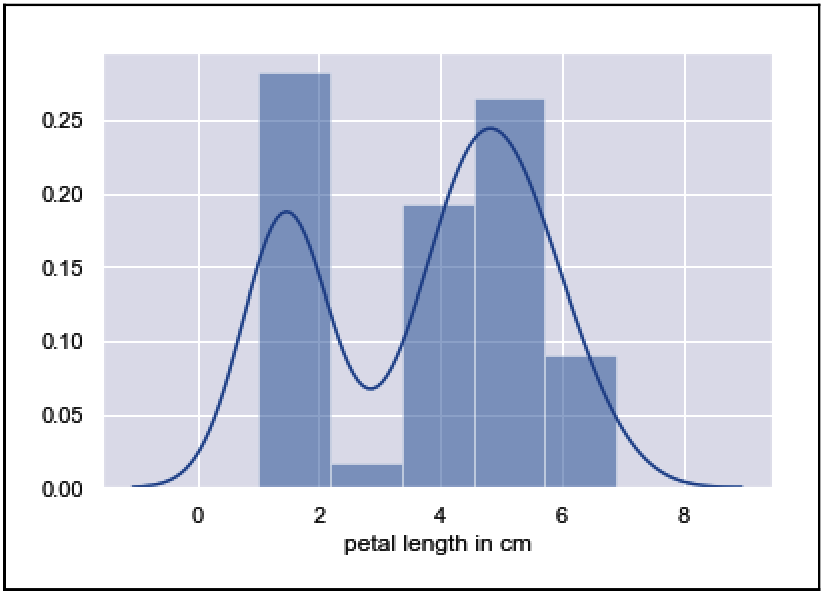


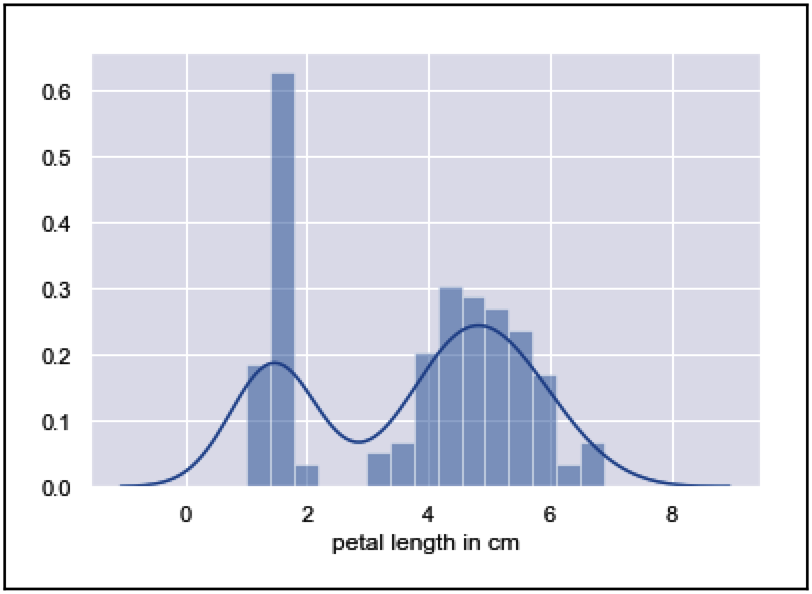
petal width in cm
petal length in cm
0.0
0.5
1.0
1.5
2.0
2.5
1
2
3
4
5
6
7

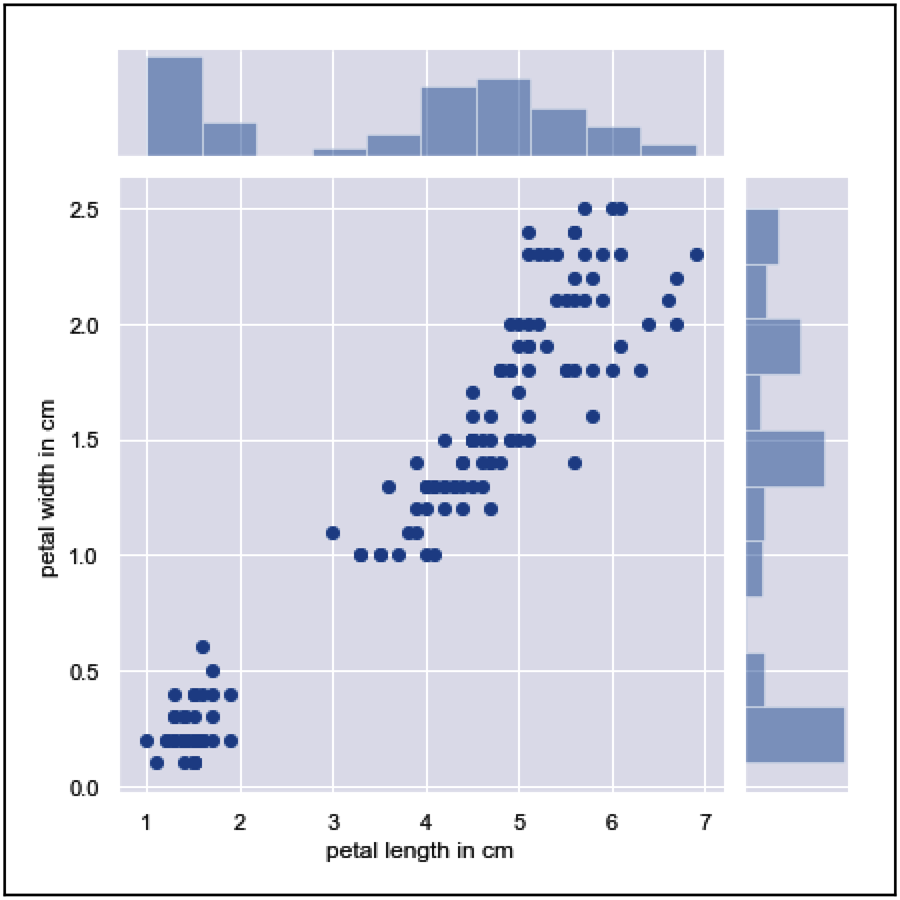


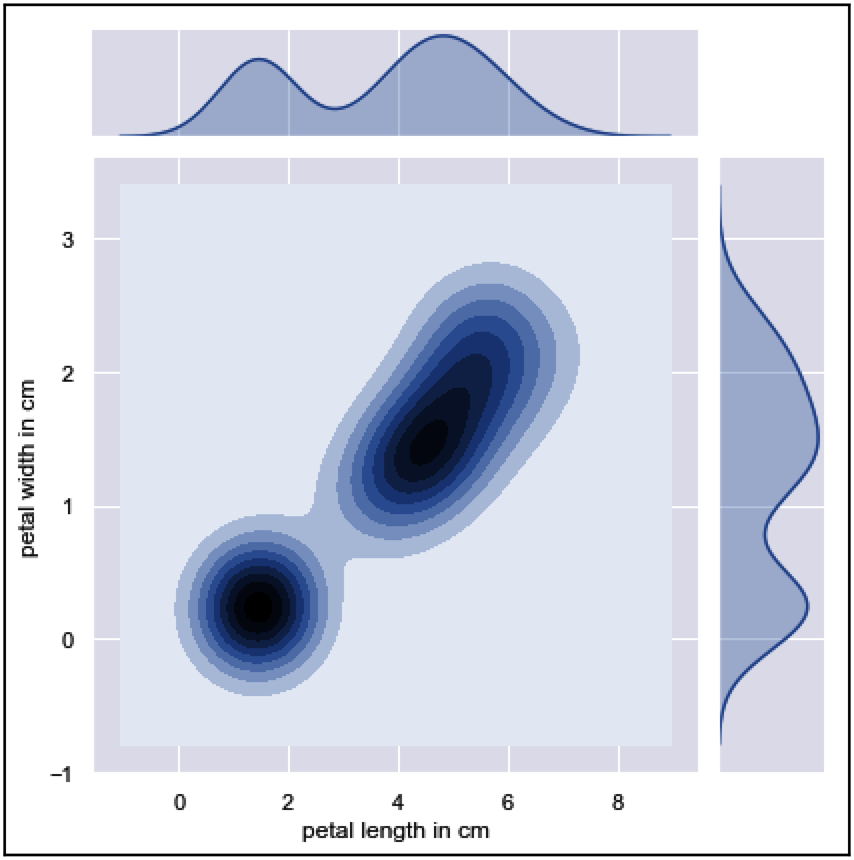
species
setosa
versicolor
virginica
petal width in cm
petal length in cm
0.0
0.5
1.0
1.5
2.0
2.5
1
2
3
4
5
6
7


2.5
2.0
1.5
1.0
0.5
0.0
1
2
3
4
5
6
7
petal length in cm
petal width in cm
species
setosa
versicolor
virginica

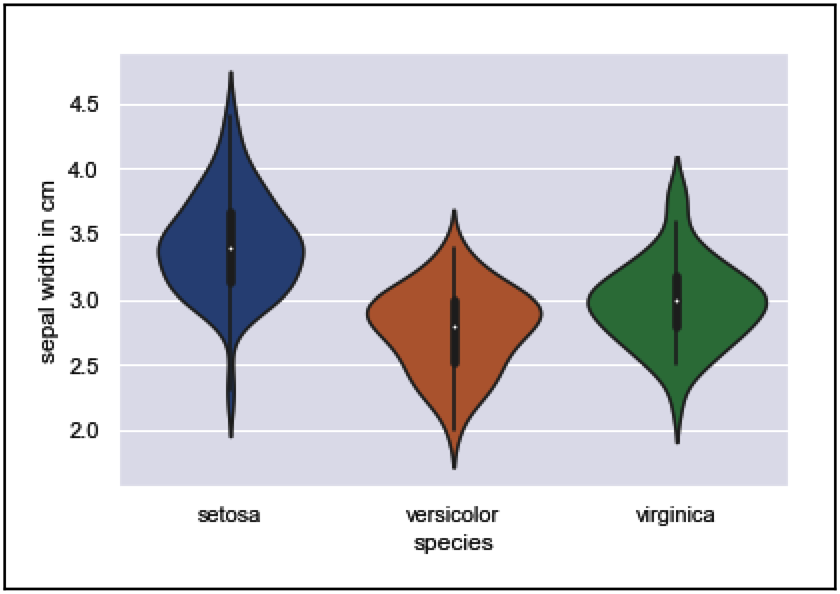

0.25
0.20
0.15
0.10
0.05
0.00
0
2
4
6
8
petal length in cm

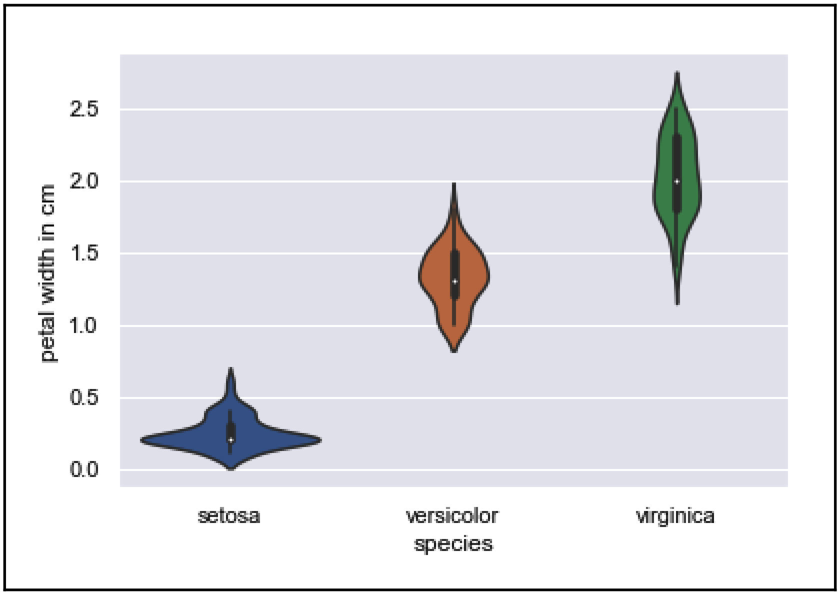

0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
2
4
6
8
petal length in cm

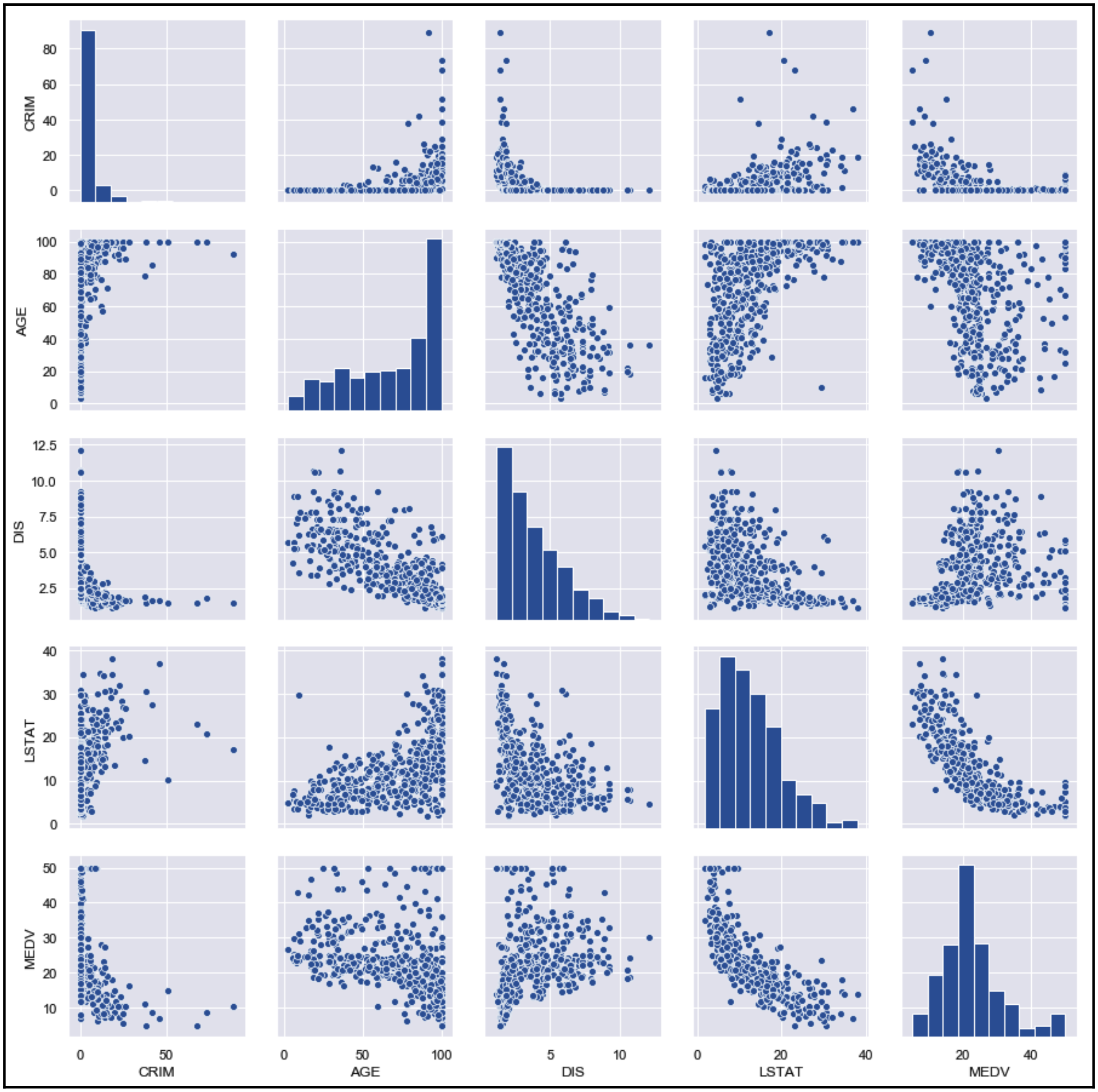

petal width in cm
2.5
2.0
1.5
1.0
0.5
0.0
1
2
3
4
5
6
7
petal length in cm



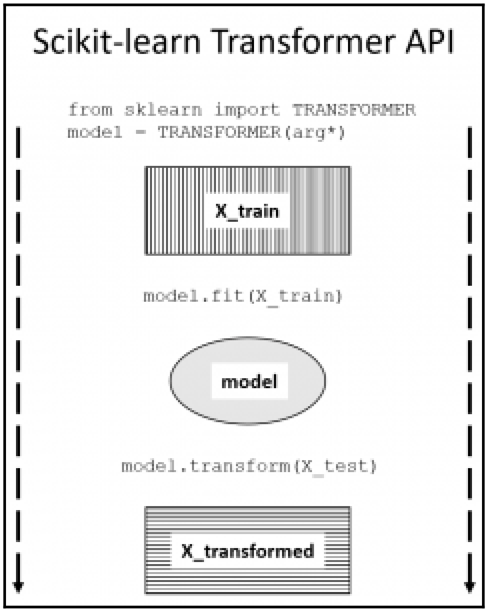
3
2
1
0
−1
petal width in cm
0
2
4
6
8
petal length in cm


sepal width in cm
4.5
4.0
3.5
3.0
2.5
2.0
setosa
versicolor
virginica
species


petal width in cm
2.5
2.0
1.5
1.0
0.5
0.0
setosa
versicolor
virginica
species


CRIM
AGE
DIS
LSTAT
MEDV
0
20
40
60
80
100
2.5
5.0
7.5
10.0
12.5
10
30
50

Chapter 4: Cleaning and Readying Data for Analysis



1	sepal length i	sepal width	petal length	petal width	species
2		3.5	1.4	0.2	setosa
3	4.9	3	1.4	0.2	setosa
4		3.2	1.3	0.2	setosa
5	4.6	3.1	1.5	0.2	setosa
6	5	3.6	1.4	0.2	setosa
7		3.9	1.7	0.4	setosa
8	4.6	3.4	1.4	0.3	setosa
9	5	3.4	1.5	0.2	setosa
10	4.4	2.9	1.4	0.2	setosa
11					
12	5.4	3.7	1.5	0.2	setosa
13	4.8	3.4	1.6	0.2	setosa
14	4.8	3	1.4	0.1	setosa
15	4.3	3	1.1	0.1	setosa
16	5.8	4	1.2	0.2	setosa
17					
18	5.4	3.9	1.3	0.4	setosa

```
        sepal length in cm  sepal width in cm
record
0                      NaN                3.5
1                      4.9                3.0
2                      NaN                3.2
3                      4.6                3.1
4                      5.0                3.6
```

```
record
0    example
1        4.9
2    example
3        4.6
4          5
Name: sepal length in cm, dtype: object
```

```
        sepal length in cm  sepal width in cm
record
1                      4.9                3.0
3                      4.6                3.1
4                      5.0                3.6
6                      4.6                3.4
7                      5.0                3.4
```

	sepal length in cm	sepal width in cm
0	5.870139	3.5
1	4.900000	3.0
2	5.870139	3.2
3	4.600000	3.1
4	5.000000	3.6

Min-Max Normalization

$$x_{i,scaled} = \frac{x_{i,original} - min_Y}{max_Y - min_Y}$$

Where:

x_i = datapoint

Y = column where x resides

	Jersey Size	Shoe Size
Person		
Thomas	small	7
Jane	medium	10
Vaughn	large	12
Vera	medium	9
Vincent	large	12
Lei-Ann	small	7

```
identified categories:
[array(['large', 'medium', 'small'], dtype=object), array([7, 9, 10, 12], dtype=object)]
encoded data:
[[2. 0.]
 [1. 2.]
 [0. 3.]
 [1. 1.]
 [0. 3.]
 [2. 0.]]
```

Person	Age	Height	Weight	Jersey Color	Jersey Size	Shoe Size	Long Jump
Thomas	12	57.5	73.4	blue	2.0	0.0	19.2
Jane	13	65.5	85.3	green	1.0	2.0	25.1
Vaughn	17	71.9	125.9	green	0.0	3.0	14.3
Vera	14	65.3	100.5	red	1.0	1.0	18.3
Vincent	18	70.1	110.7	blue	0.0	3.0	21.1

One-hot Encoding Example

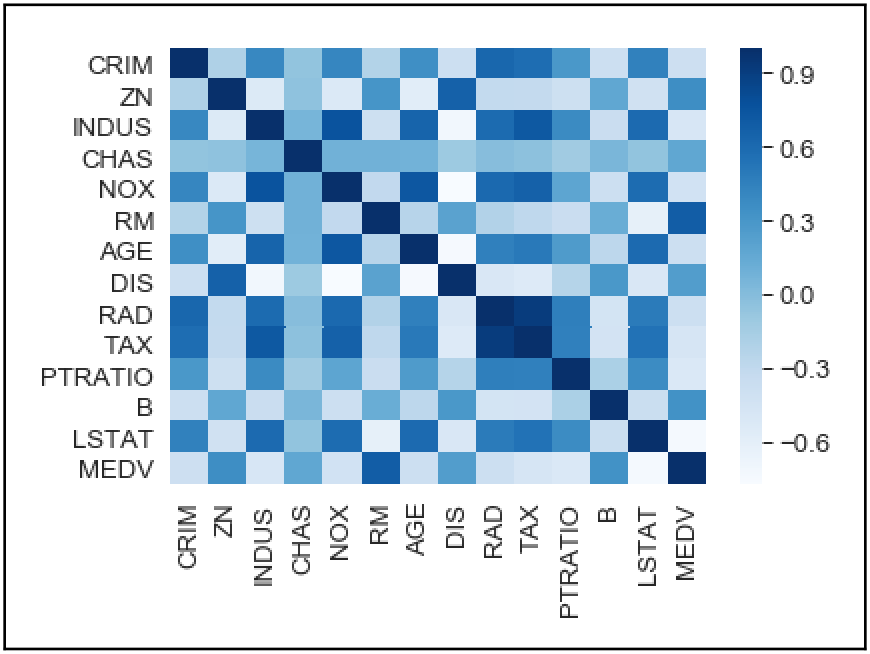
Source

Person	Shoe Size
Thomas	7
Jane	10
Vaughn	12
Vera	9
Vincent	12
Lei-Ann	7

Encoded

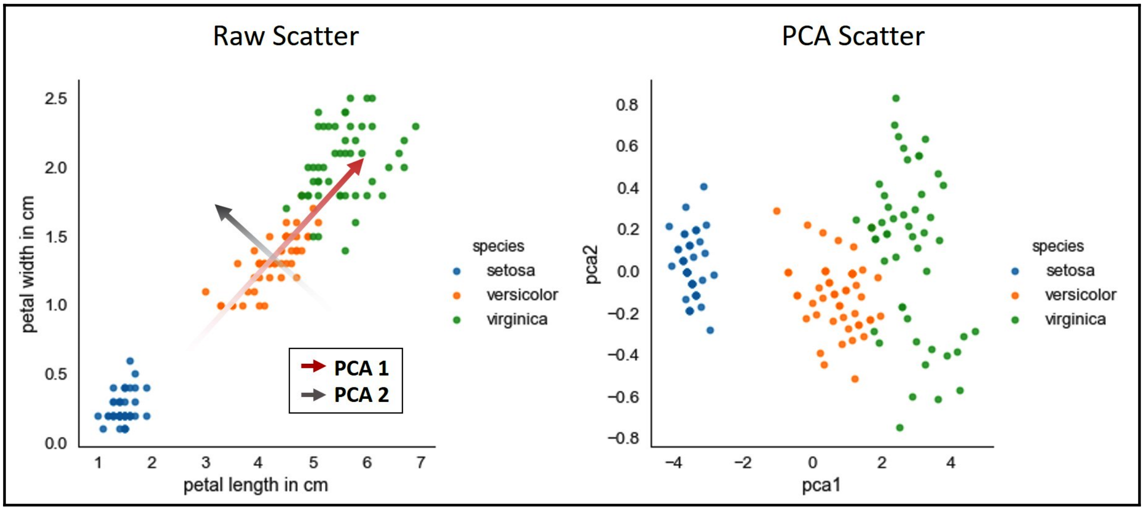
Person	Shoe Size_7	Shoe Size_9	Shoe Size_10	Shoe Size_12
Thomas	1	0	0	0
Jane	0	0	1	0
Vaughn	0	0	0	1
Vera	0	1	0	0
Vincent	0	0	0	1
Lei-Ann	1	0	0	0

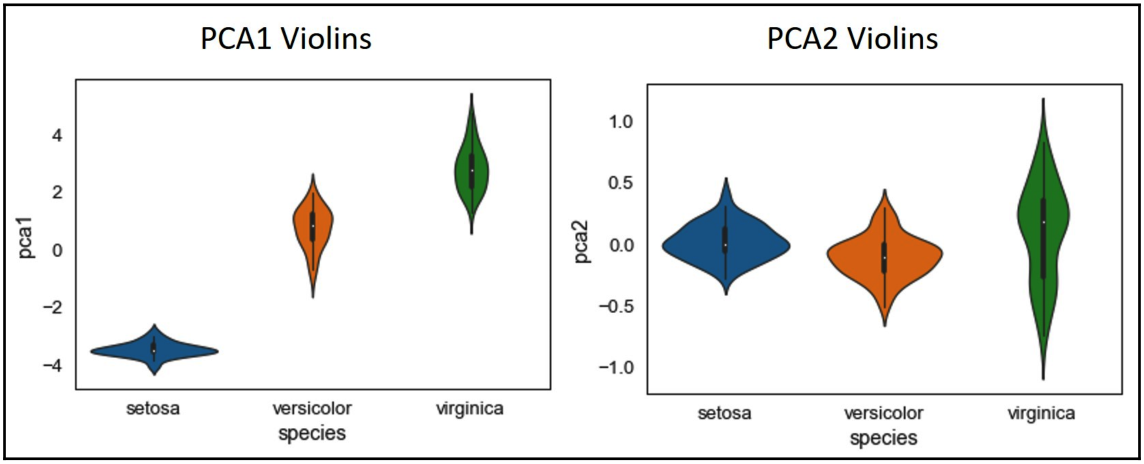
record	sepal length in cm	petal length in cm	species
0	5.1	1.4	setosa
1	4.9	1.4	setosa
2	4.7	1.3	setosa
3	4.6	1.5	setosa
4	5.0	1.4	setosa

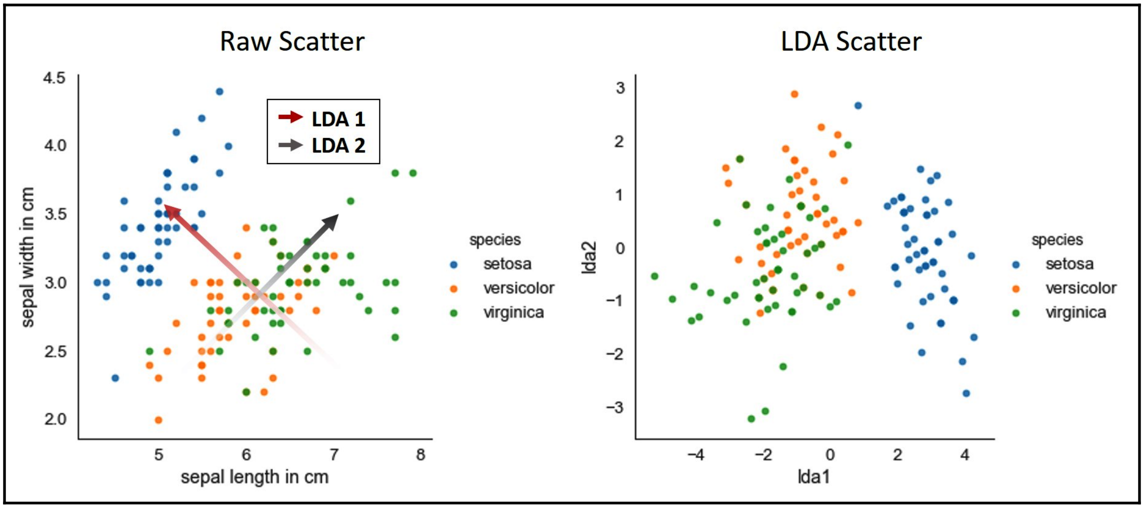



```
CRIM       0.385832
ZN         0.360445
INDUS      0.483725
CHAS       0.175260
NOX        0.427321
RM         0.695360
AGE        0.376955
DIS        0.249929
RAD        0.381626
TAX        0.468536
PTRATIO    0.507787
B          0.333461
LSTAT      0.737663
MEDV       1.000000
Name: MEDV, dtype: float64
```

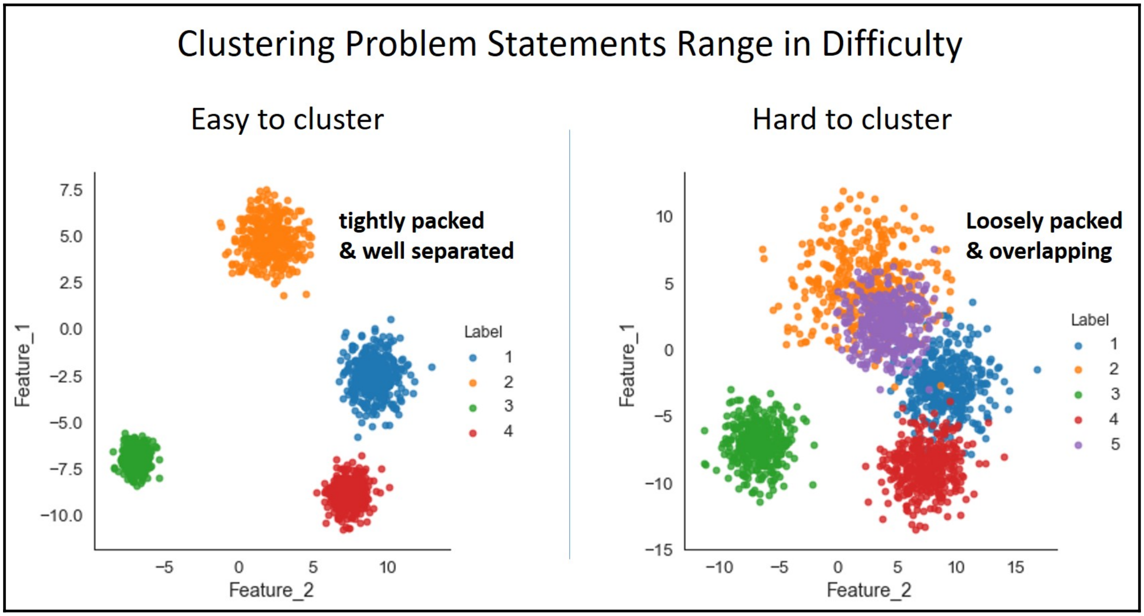
```
selected columns, correlation with target > 0.6
RM        0.695360
LSTAT     0.737663
MEDV      1.000000
Name: MEDV, dtype: float64
            RM  LSTAT  MEDV
record
0        6.575   4.98  24.0
1        6.421   9.14  21.6
2        7.185   4.03  34.7
3        6.998   2.94  33.4
4        7.147   5.33  36.2
```

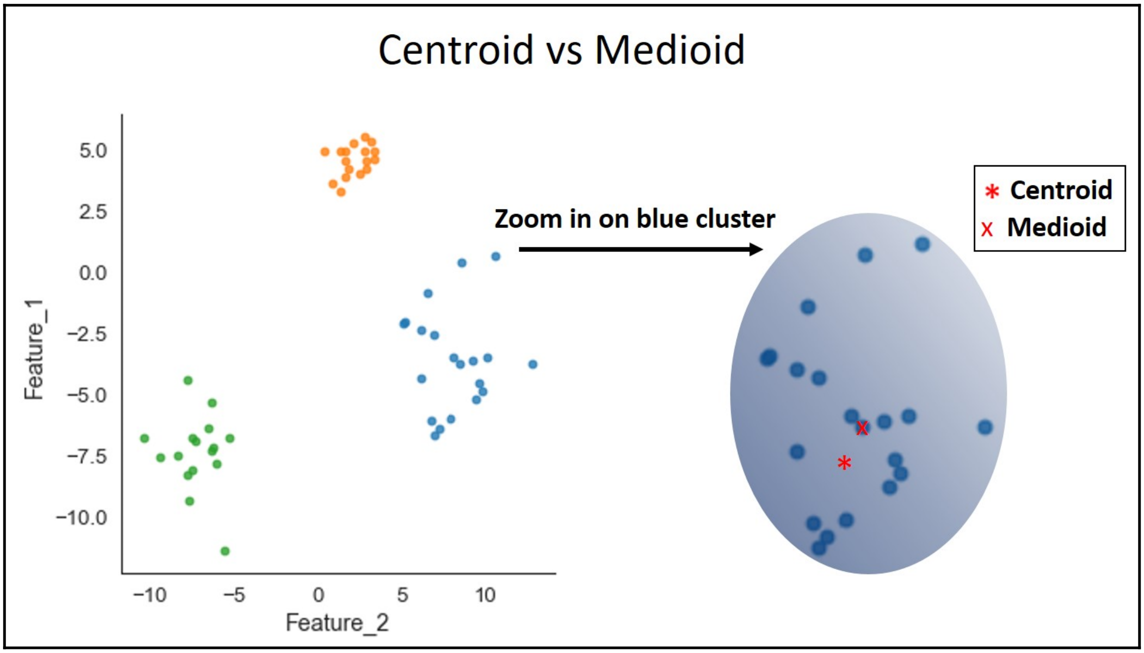


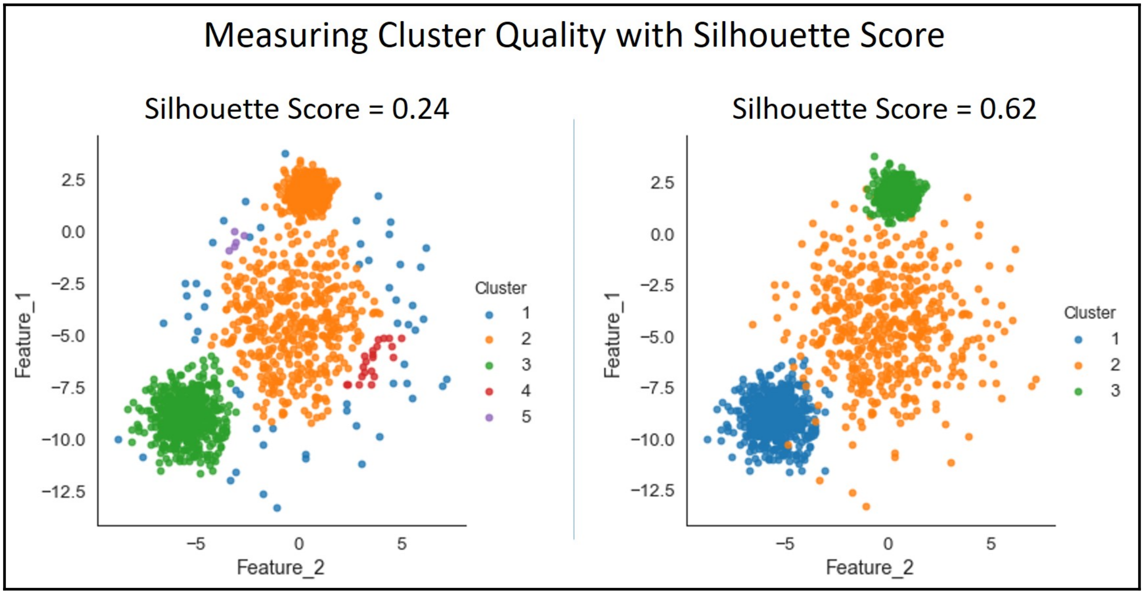

PCA1 Violins
PCA2 Violins
pca1
pca2
setosa
versicolor
virginica
species

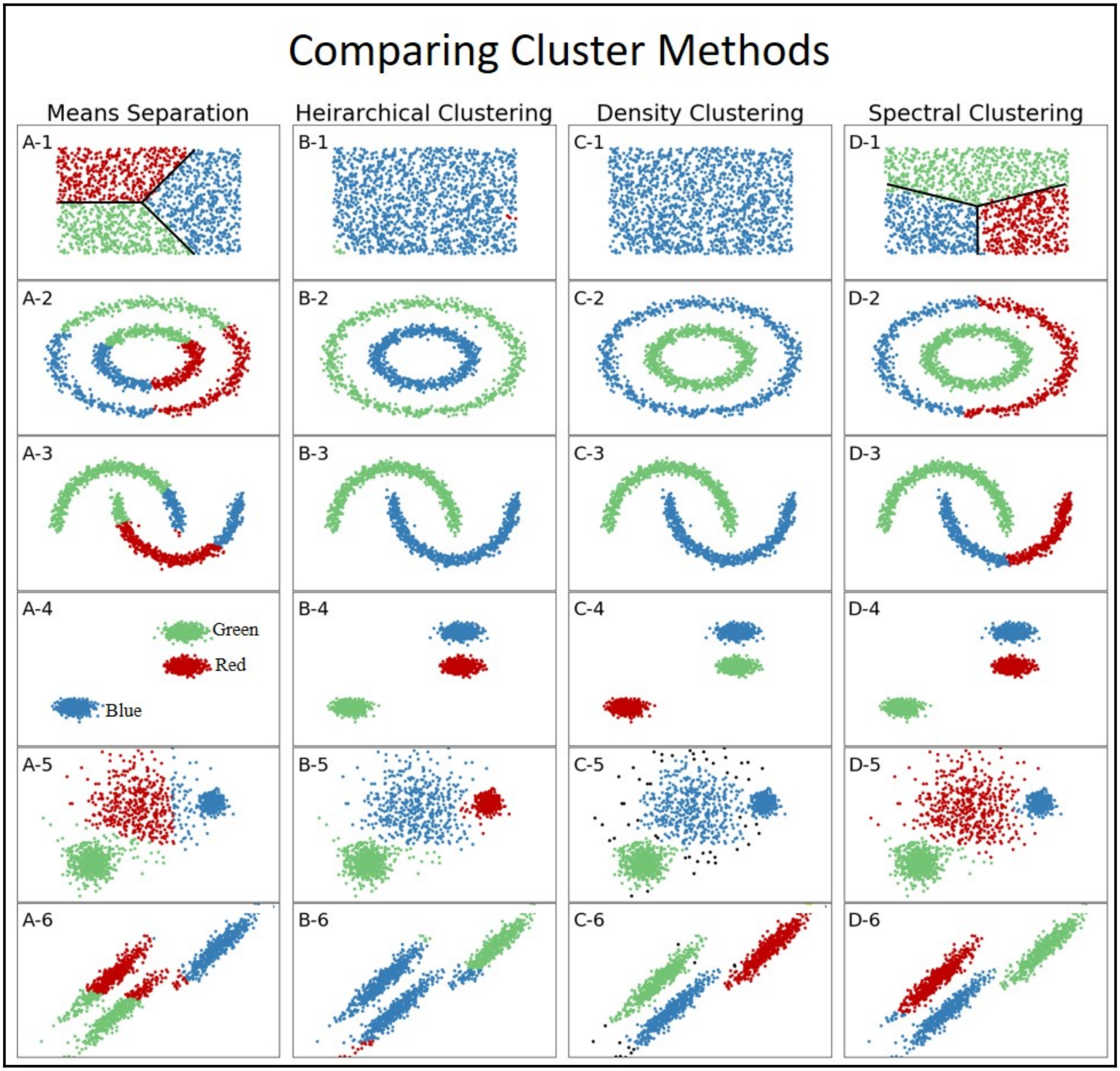

Raw Scatter
LDA Scatter
LDA 1
LDA 2
sepal width in cm
sepal length in cm
lda2
lda1
species
setosa
versicolor
virginica

Chapter 5: Grouping and Clustering Data

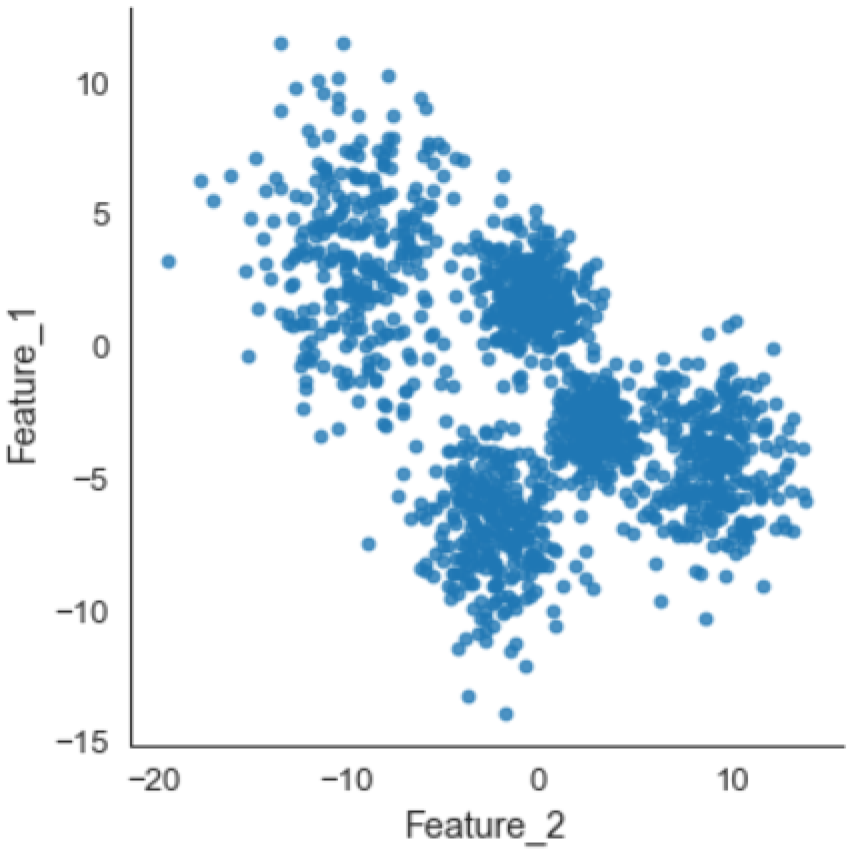


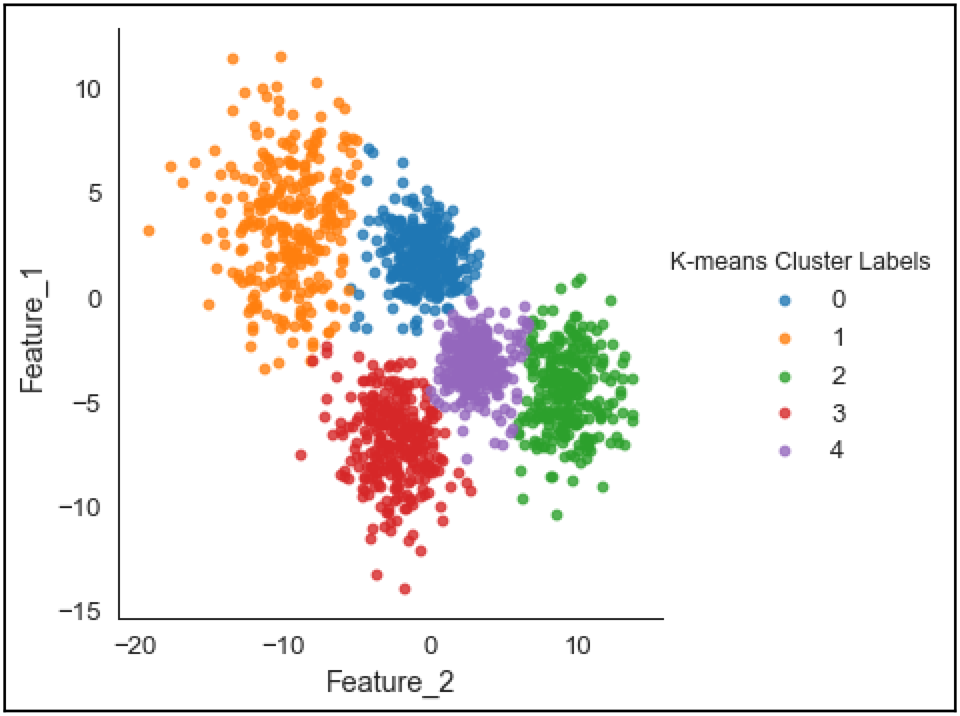


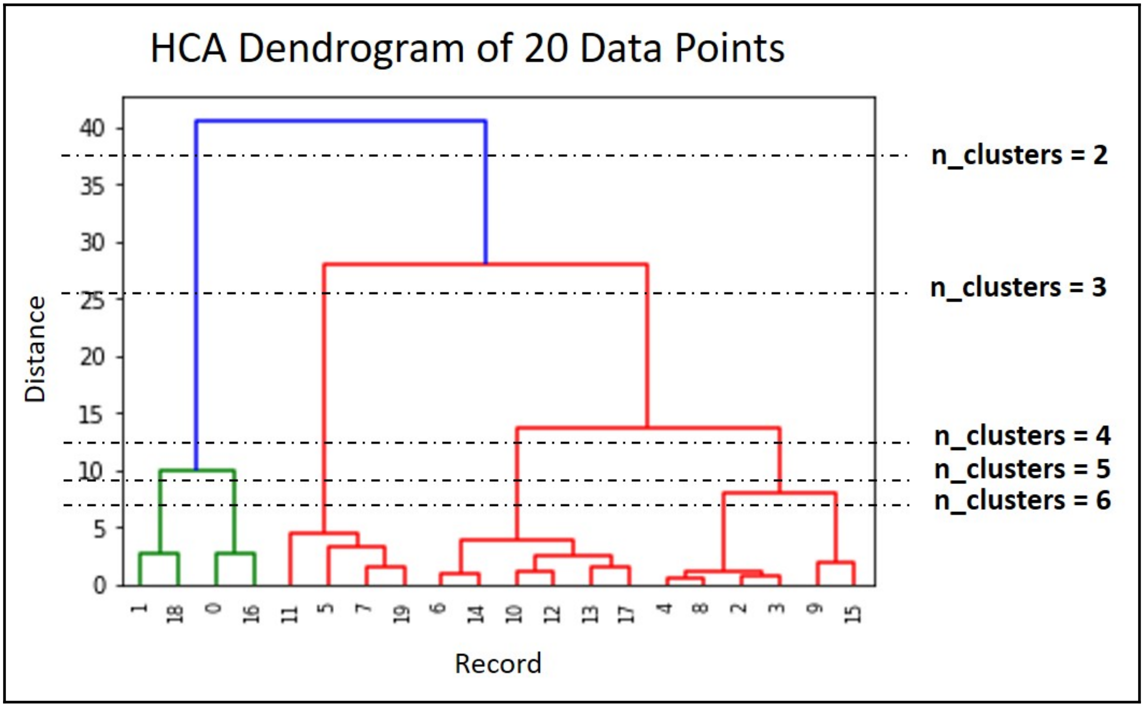

Measuring Cluster Quality with Silhouette Score
Silhouette Score = 0.24
Silhouette Score = 0.62
Feature_1
Feature_2
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−12.5
−5
0
5
Cluster
1
2
3
4
5


Comparing Cluster Methods
Means Separation
Heirarchical Clustering
Density Clustering
Spectral Clustering
A-1
B-1
C-1
D-1
A-2
B-2
C-2
D-2
A-3
B-3
C-3
D-3
A-4
Green
Red
Blue
B-4
C-4
D-4
A-5
B-5
C-5
D-5
A-6
B-6
C-6
D-6

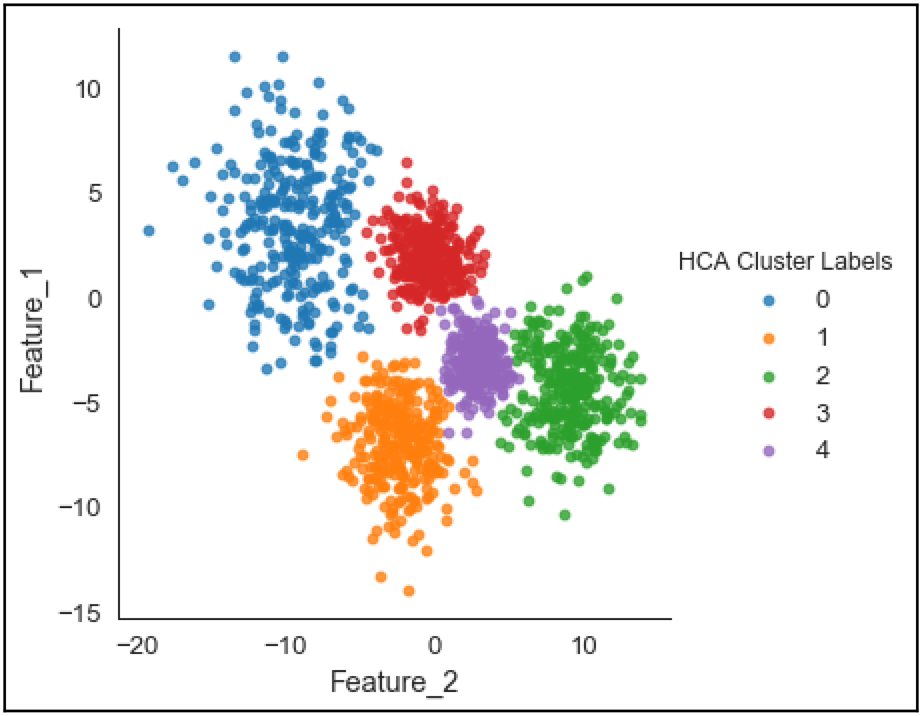
```
        Feature_1  Feature_2
record
0       11.492294 -10.236187
1        4.376245  -9.152790
2       -2.193675   3.212265
3       -2.976039   3.037043
4       -2.963703   2.336960

<seaborn.axisgrid.FacetGrid at 0x24c3da12e48>
```

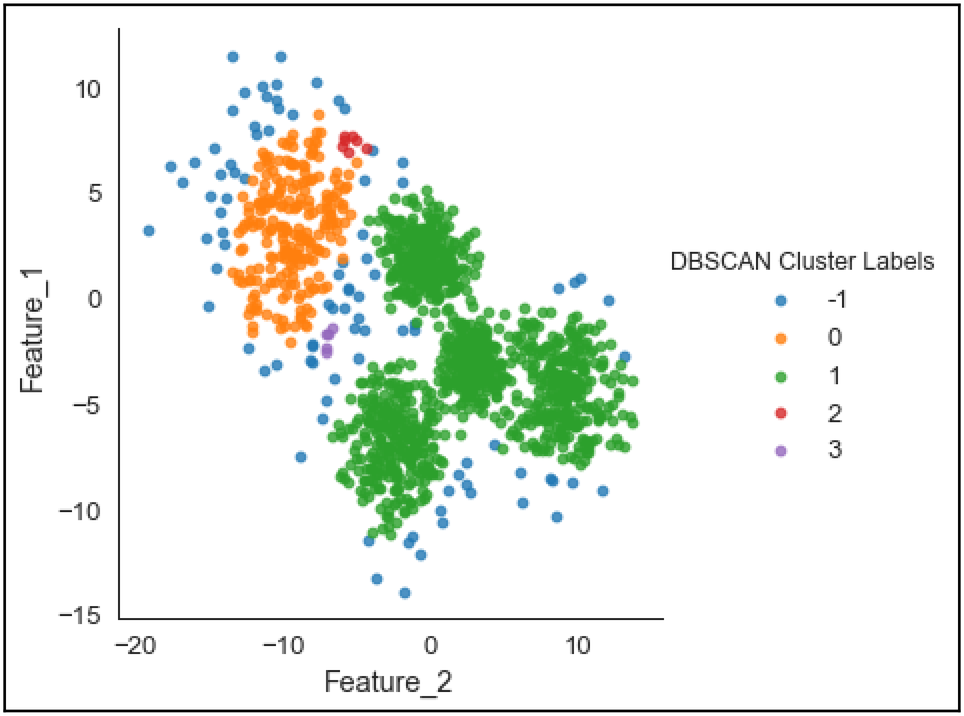



Feature_1
Feature_2
K-means Cluster Labels
0
1
2
3
4
10
5
0
−5
−10
−15
−20
−10
0
10


HCA Dendrogram of 20 Data Points
n_clusters = 2
n_clusters = 3
n_clusters = 4
n_clusters = 5
n_clusters = 6
Distance
40
35
30
25
20
15
10
5
0
1 18 0 16 11 5 7 19 6 14 10 12 13 17 4 8 2 3 9 15
Record

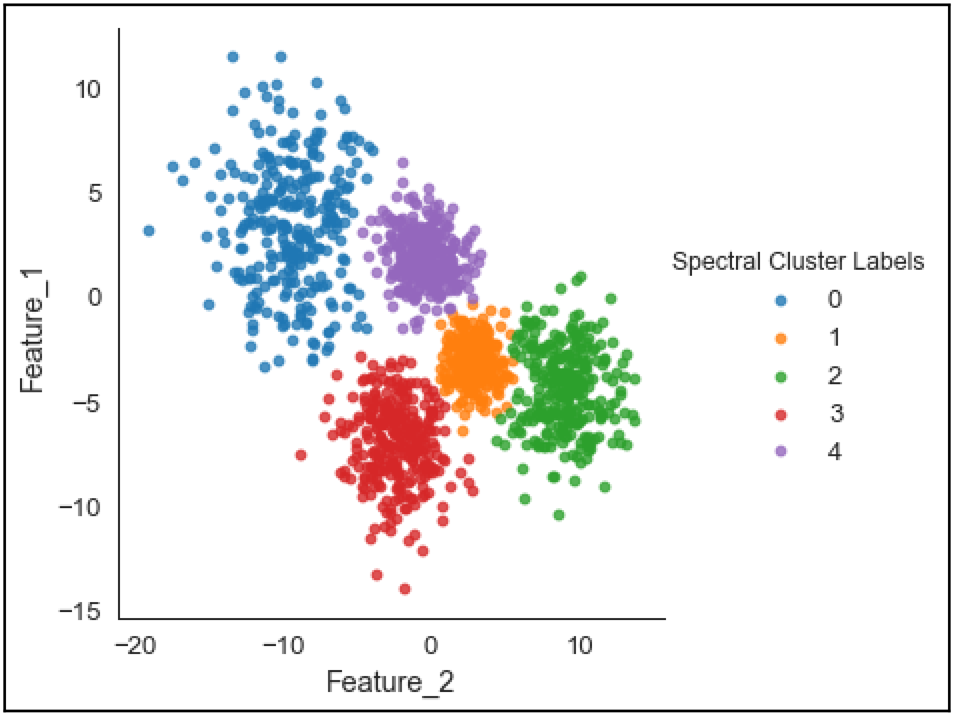
$$D = \begin{pmatrix} l_{11} & l_{12} & \ldots & l_{1n} \\ l_{21} & l_{22} & \ldots & l_{2n} \\ . & . & \ldots & . \\ . & . & \ldots & . \\ . & . & \ldots & . \\ l_{n1} & l_{n2} & \ldots & l_{nn} \end{pmatrix}$$


HCA Cluster Labels
0
1
2
3
4
Feature_1
Feature_2
10
5
0
−5
−10
−15
−20
−10
0
10



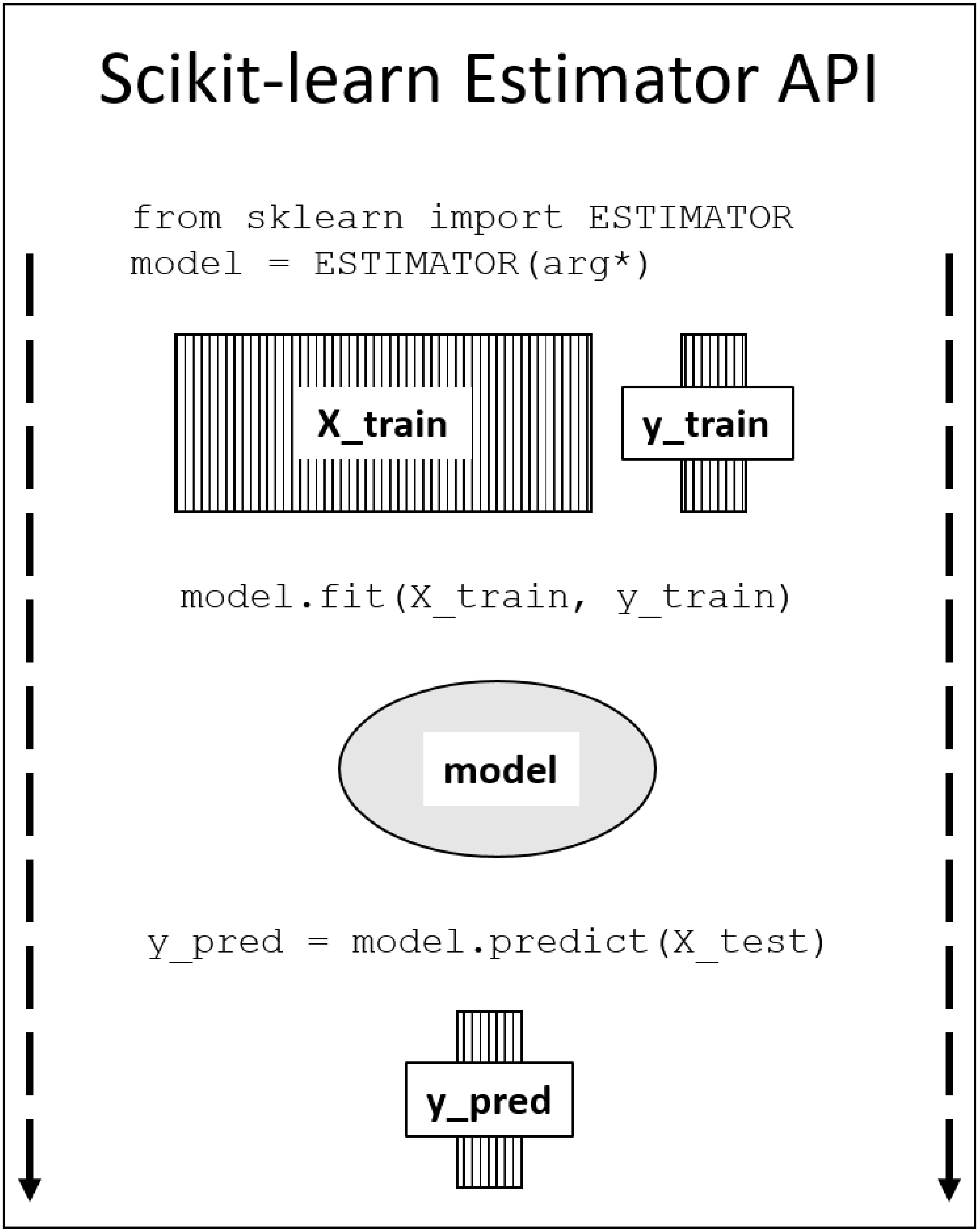
DBSCAN Cluster Labels
-1
0
1
2
3
Feature_1
Feature_2
10
5
0
−5
−10
−15
−20
−10
0
10

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$$



Feature_1
10
5
0
−5
−10
−15
−20
−10
0
10
Feature_2
Spectral Cluster Labels
0
1
2
3
4

Chapter 6: Prediction with Regression and Classification


Scikit-learn Estimator API
from sklearn import ESTIMATOR
model = ESTIMATOR(arg*)
X_train
y_train
model.fit(X_train, y_train)
model
y_pred = model.predict(X_test)
y_pred

Size of X

$m = 6$

$n = 4$

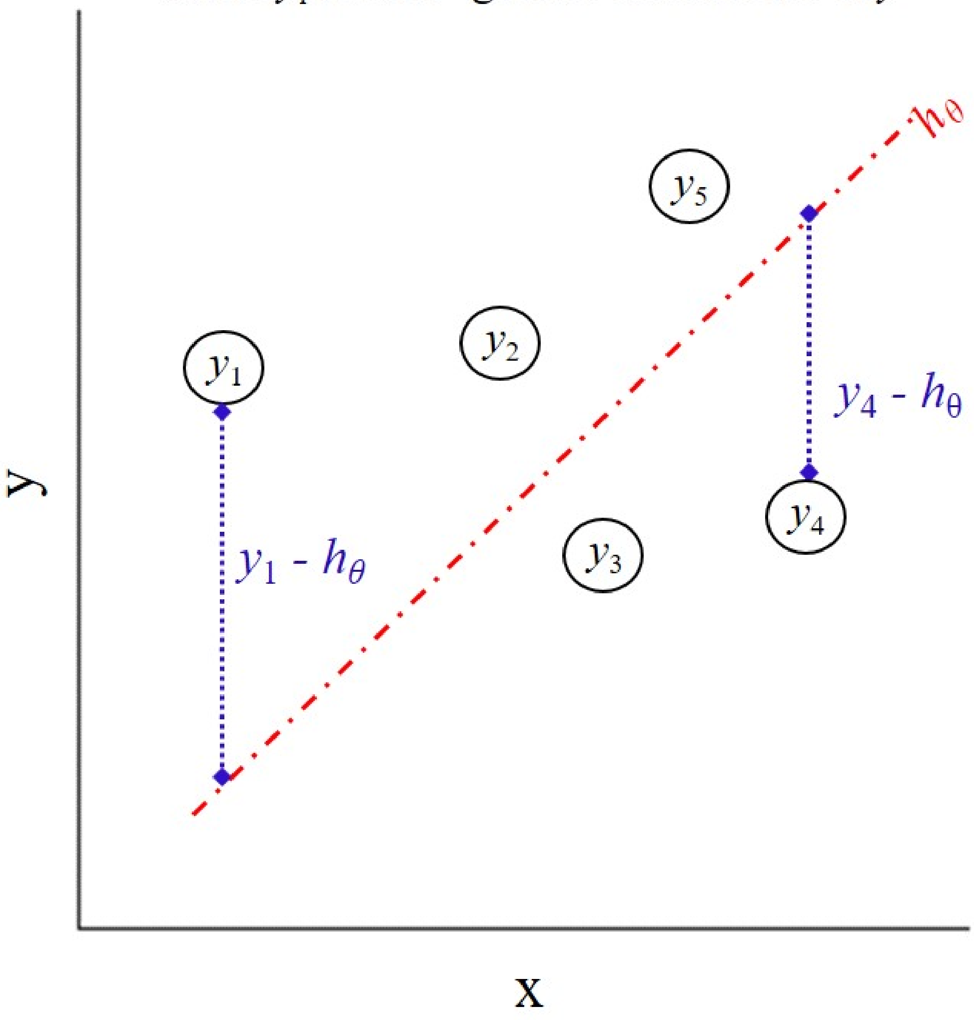
n = # of features

m = # of records

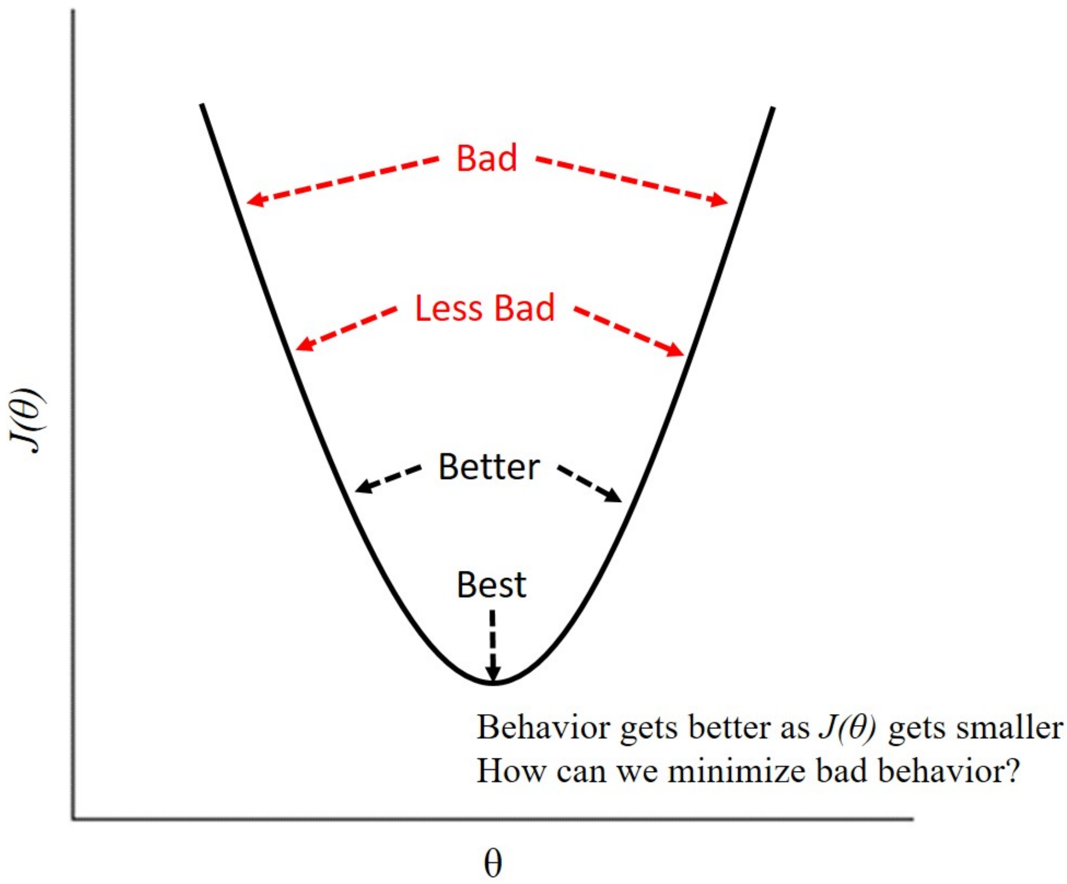
	X				Y
Person	**Age**	**Height**	**Weight**	**Training Hours/week**	**Long Jump**
Thomas	12	57.5	73.4	6.5	19.2
Charlize	13	65.5	85.3	8.9	25.1
Vaughn	17	71.9	125.9	1.1	14.3
Vera	14	65.3	100.5	7.9	18.3
Vincent	18	70.1	110.7	10.5	21.1
Lei-Ann	12	52.3	70.4	0.5	10.6

Hypothesis and Loss for Linear Regression

where y_i is the i^{th} ground truth record of y

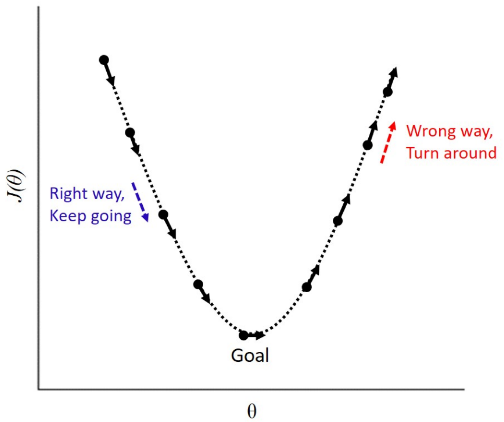


Visualizing Prediction Behavior on the Loss Function $J(\theta)$



Minimizing the Loss Function $J(\theta)$: Reasoning for use of Derivative Descent

Traverse Loss Function $J(\theta)$ from Left to Right
How do we reach the bottom of the bowl?

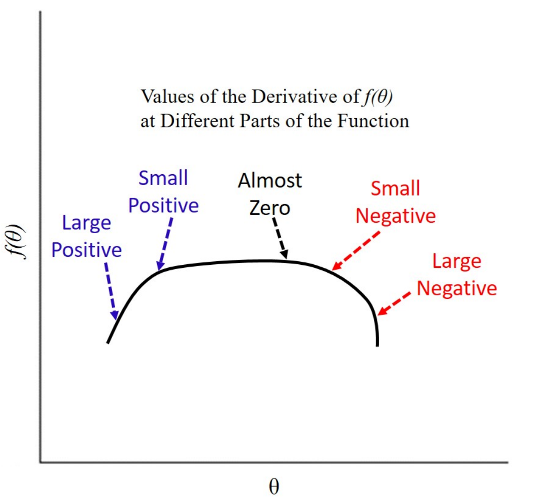


Observe Trends to Define Rules for Mathematical Machinery
How do we reach the bottom of the bowl?

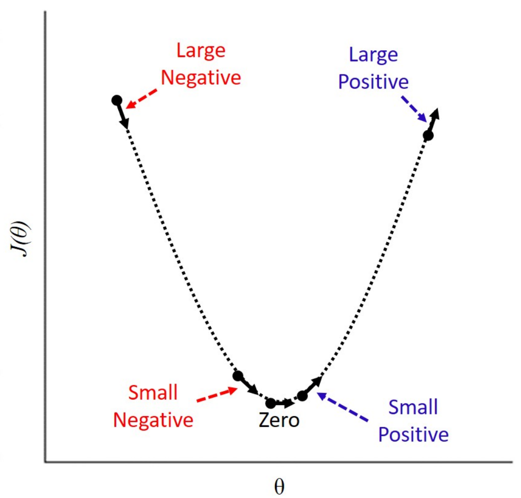
Observation	If ↘ If ↗	Right way, Keep going Wrong way, Turn around
Observation restated	If ● larger than ➔ If ● smaller than ➔	Right way, Keep going Wrong way, Turn around
What is this?	Is ● larger than ➔?	This happens to be (in part), the definition of the derivative
Observation with the derivative	If ● larger than ➔ If ● smaller than ➔	Derivative is negative, Keep going Derivate is positive, Turn around

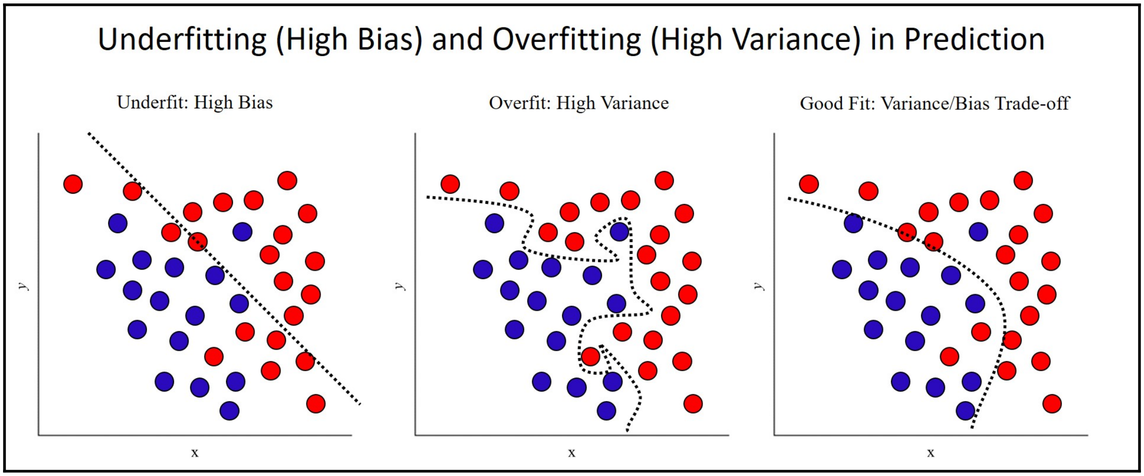
Visualizing Value of the Derivative

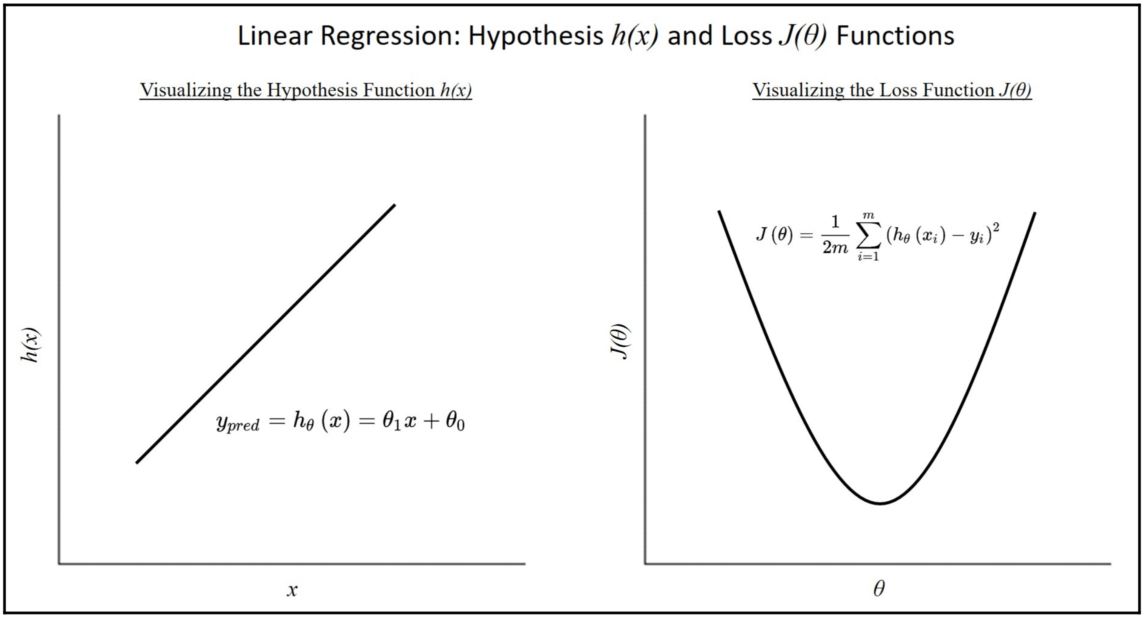
Visualizing Derivative Values of the Example Function $f(\theta)$



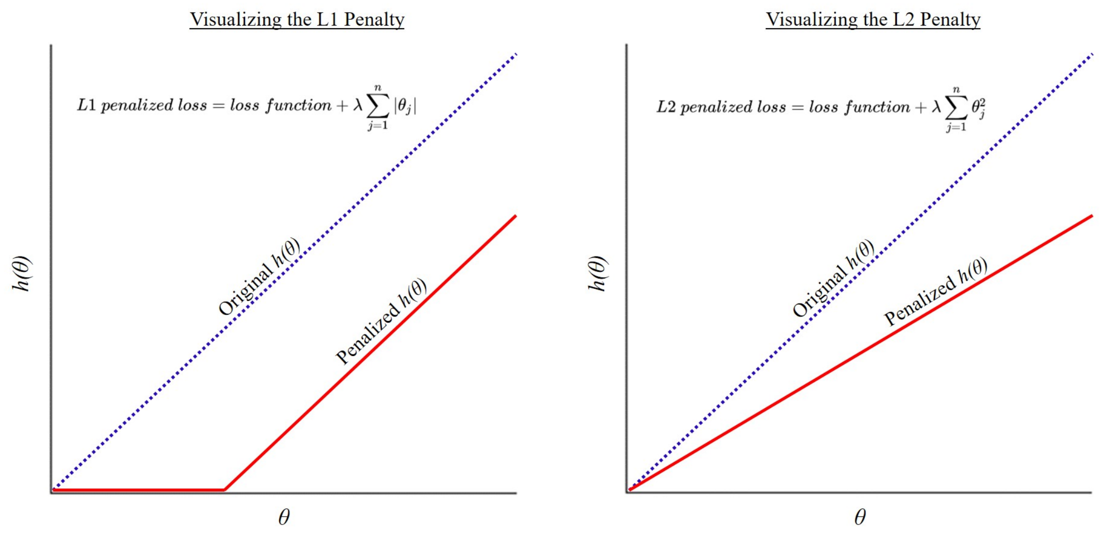
Visualizing Derivative Values of the Loss Function $J(\theta)$



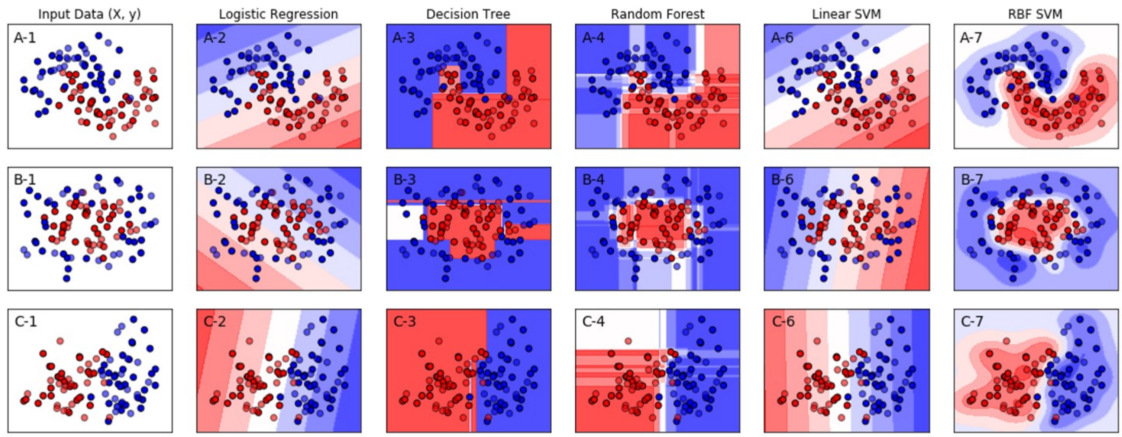

Underfitting (High Bias) and Overfitting (High Variance) in Prediction
Underfit: High Bias
Overfit: High Variance
Good Fit: Variance/Bias Trade-off
y
x
y
x
y
x


Linear Regression: Hypothesis $h(x)$ and Loss $J(\theta)$ Functions
Visualizing the Hypothesis Function $h(x)$
Visualizing the Loss Function $J(\theta)$
$h(x)$
$y_{pred} = h_\theta(x) = \theta_1 x + \theta_0$
x
$J(\theta)$
$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)^2$
θ

Regularization: L1 and L2 Penalties



Comparing Classification Methods



Example adapted from Scikit-learn open-source developer guide, Code source: Gaël Varoquaux, Andreas Müller; 2018
https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

Confusion Matrix

Actual Class

Prediction

	Positive	Negative
Positive	True Positive	False Positive
Negative	False Negative	True Negative

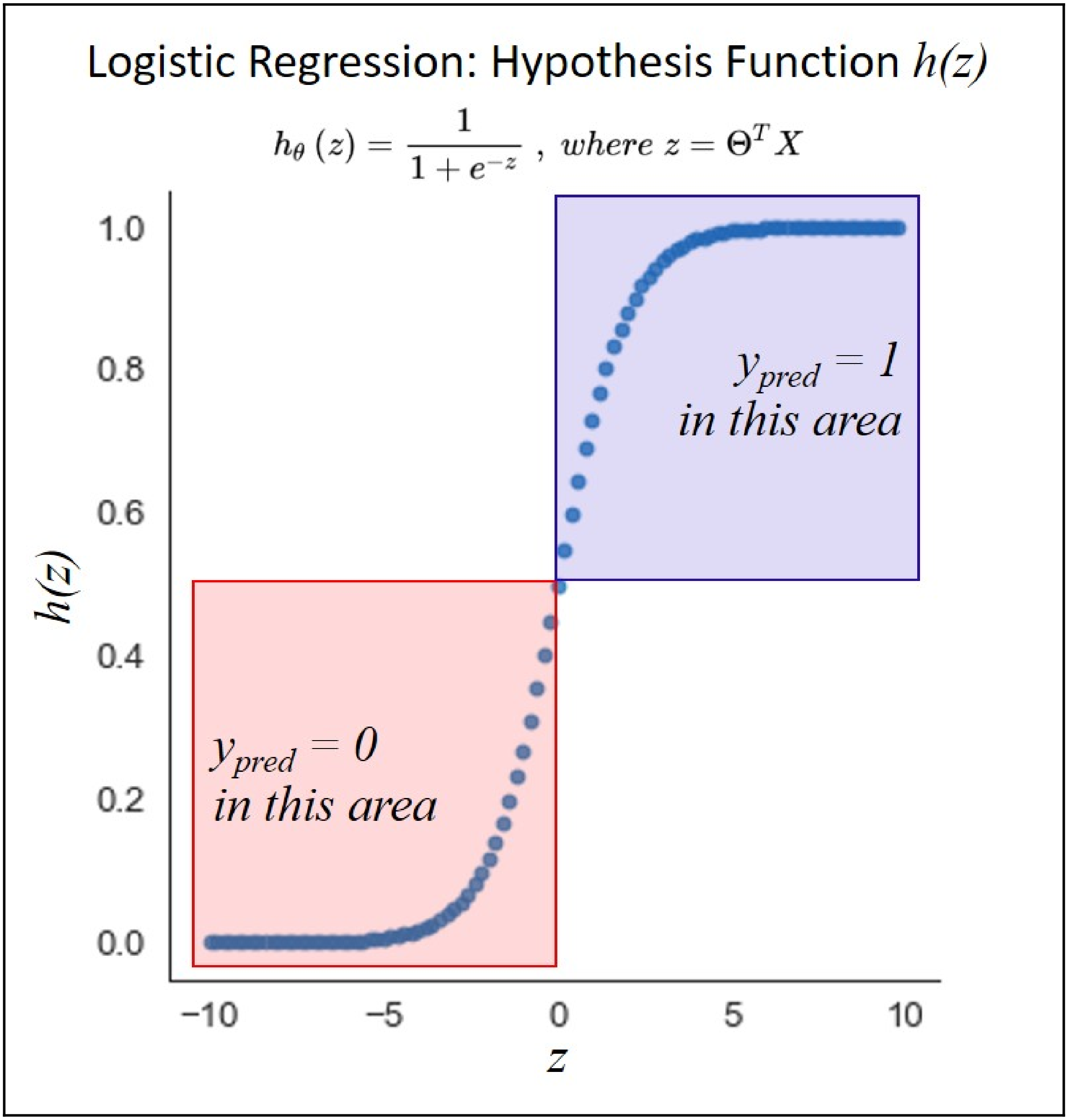
Metric Scores

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

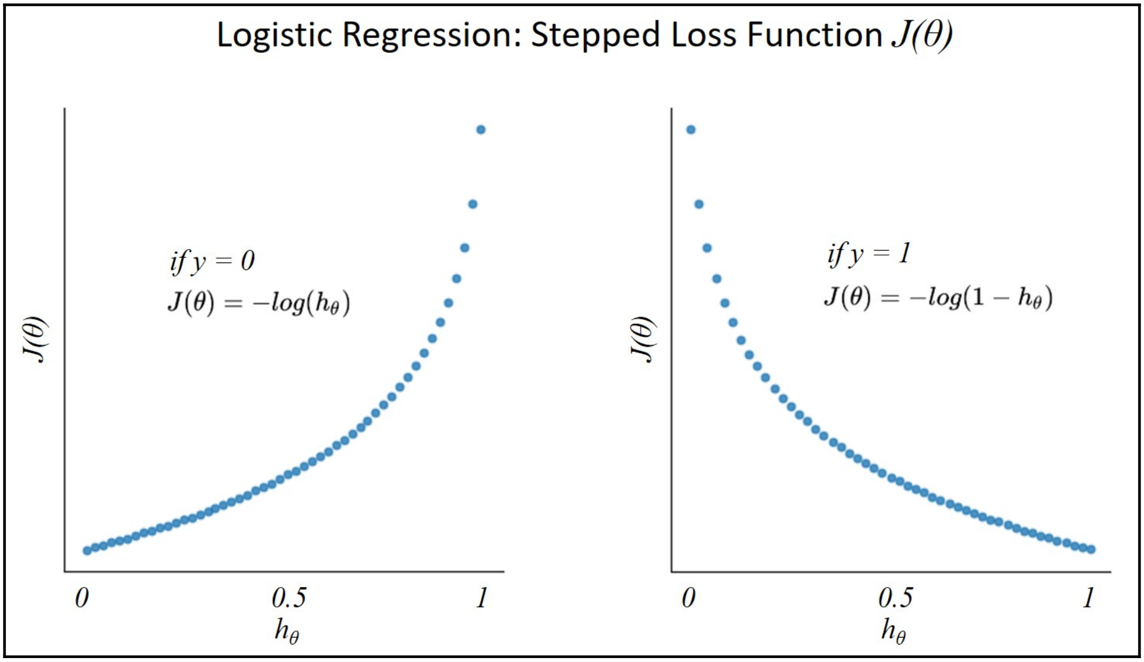
$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

$$F_1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

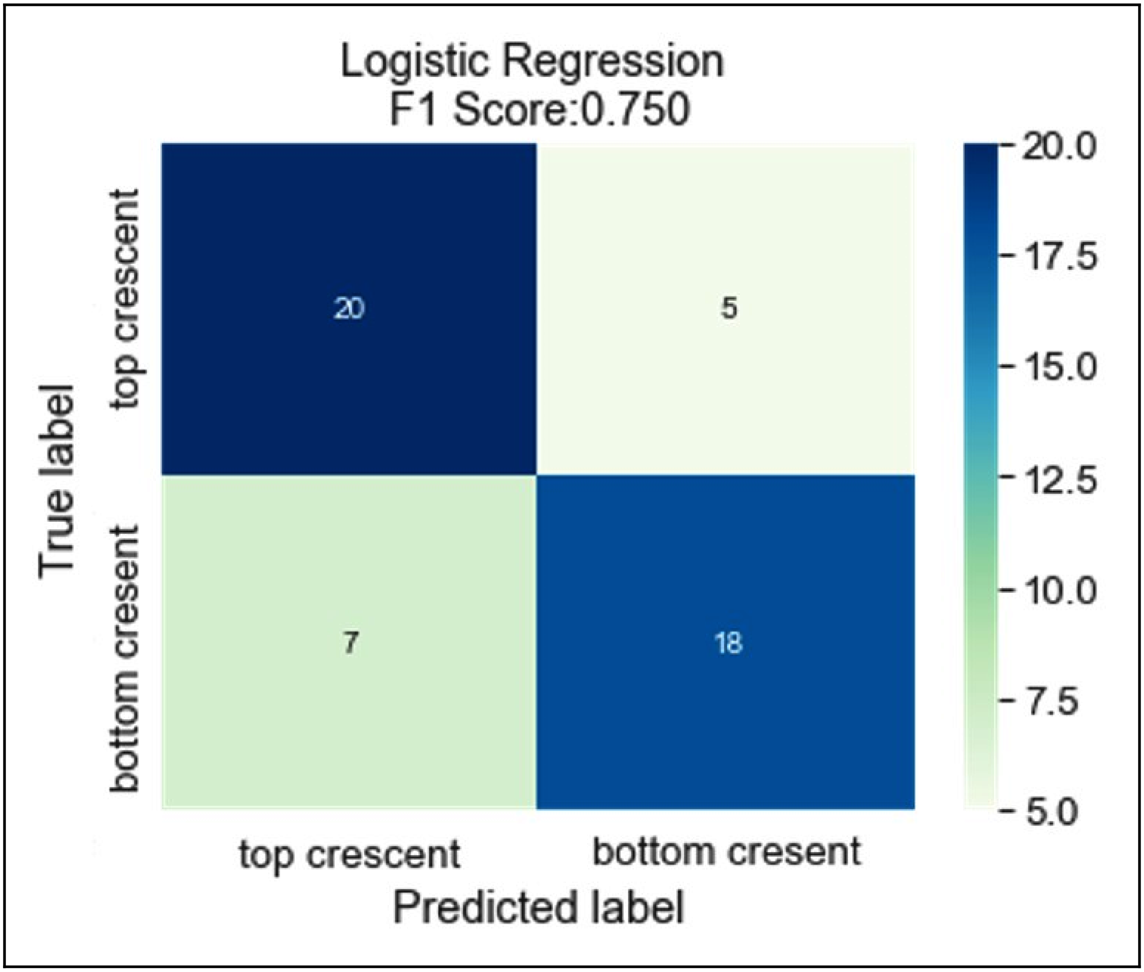
<u>Input Labels</u> 4 total	[A,B,C,D]
<u>One-vs-rest</u> Classifiers to be built = 4	[A] vs [B,C,D] [B] vs [A,C,D] [C] vs [A,B,D] [D] vs [A,B,C]
<u>One-vs-one</u> Classifiers to be built = 6	[A] vs [B] [A] vs [C] [A] vs [D] [B] vs [C] [B] vs [D] [C] vs [D]

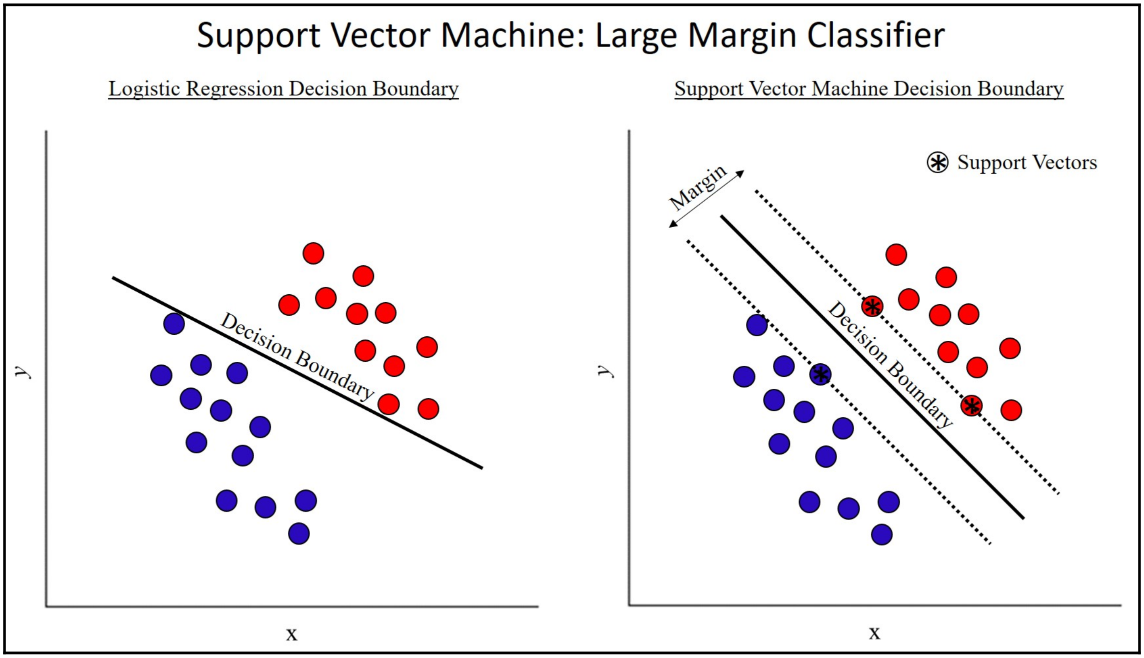


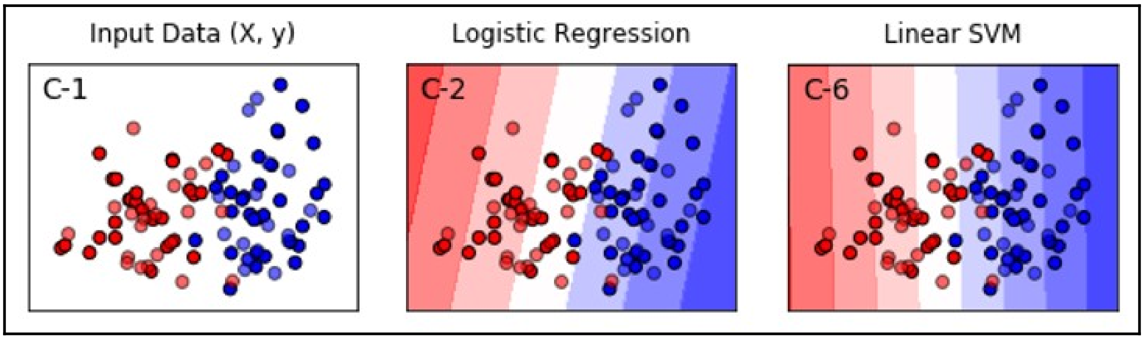
Logistic Regression: Hypothesis Function h(z)
$h_\theta(z) = \frac{1}{1+e^{-z}}$, where $z = \Theta^T X$
1.0
0.8
0.6
0.4
0.2
0.0
h(z)
$y_{pred} = 1$
in this area
$y_{pred} = 0$
in this area
−10
−5
0
5
10
z

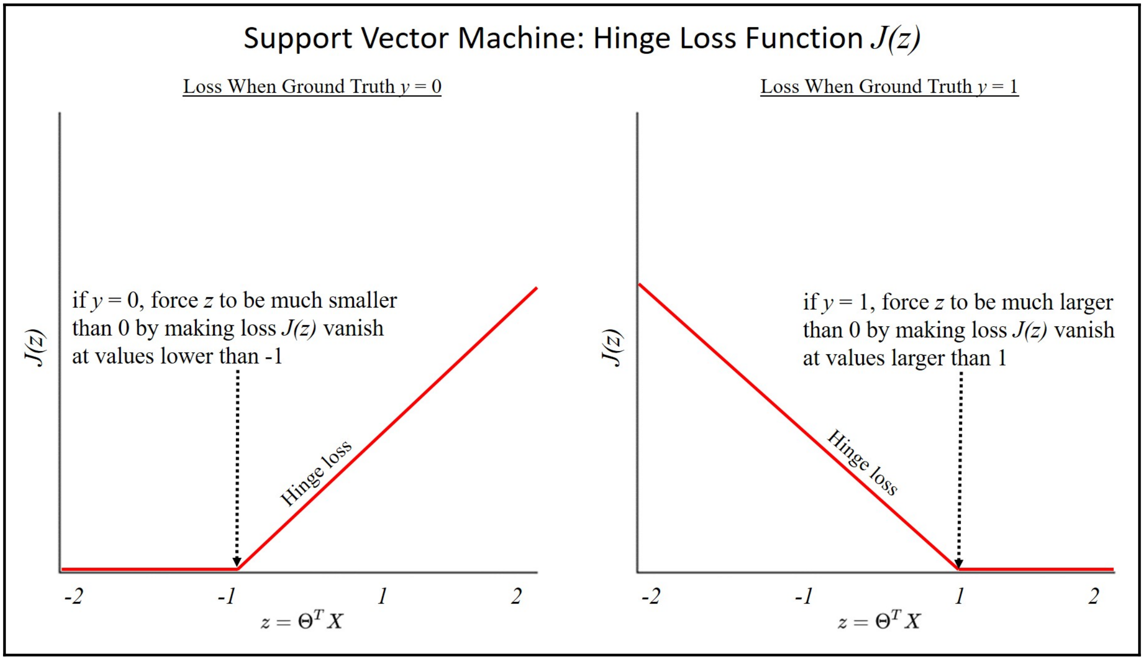


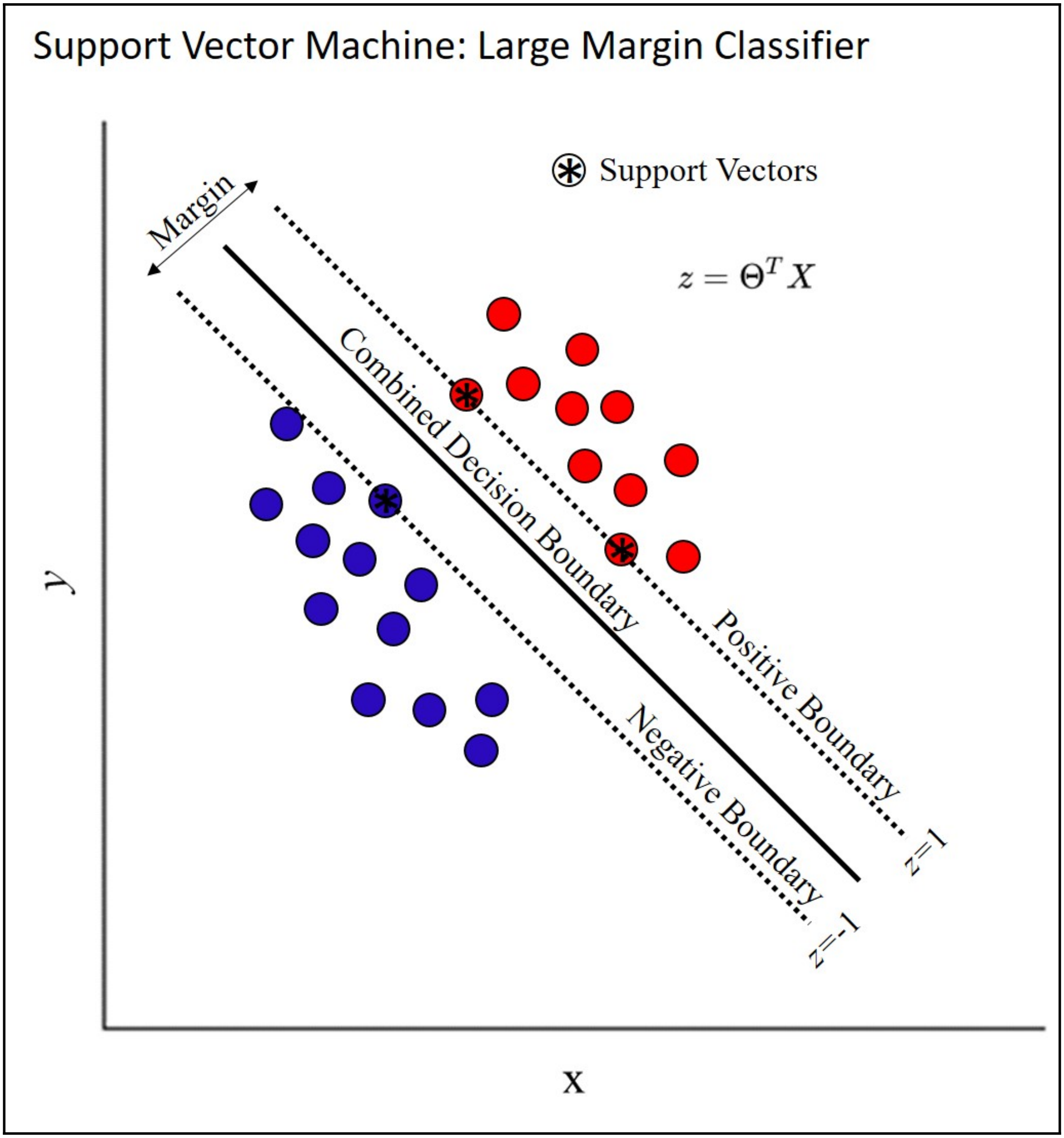
Logistic Regression: Stepped Loss Function $J(\theta)$
$J(\theta)$
$if\ y = 0$
$J(\theta) = -log(h_\theta)$
0
0.5
1
h_θ
$J(\theta)$
$if\ y = 1$
$J(\theta) = -log(1 - h_\theta)$
0
0.5
1
h_θ

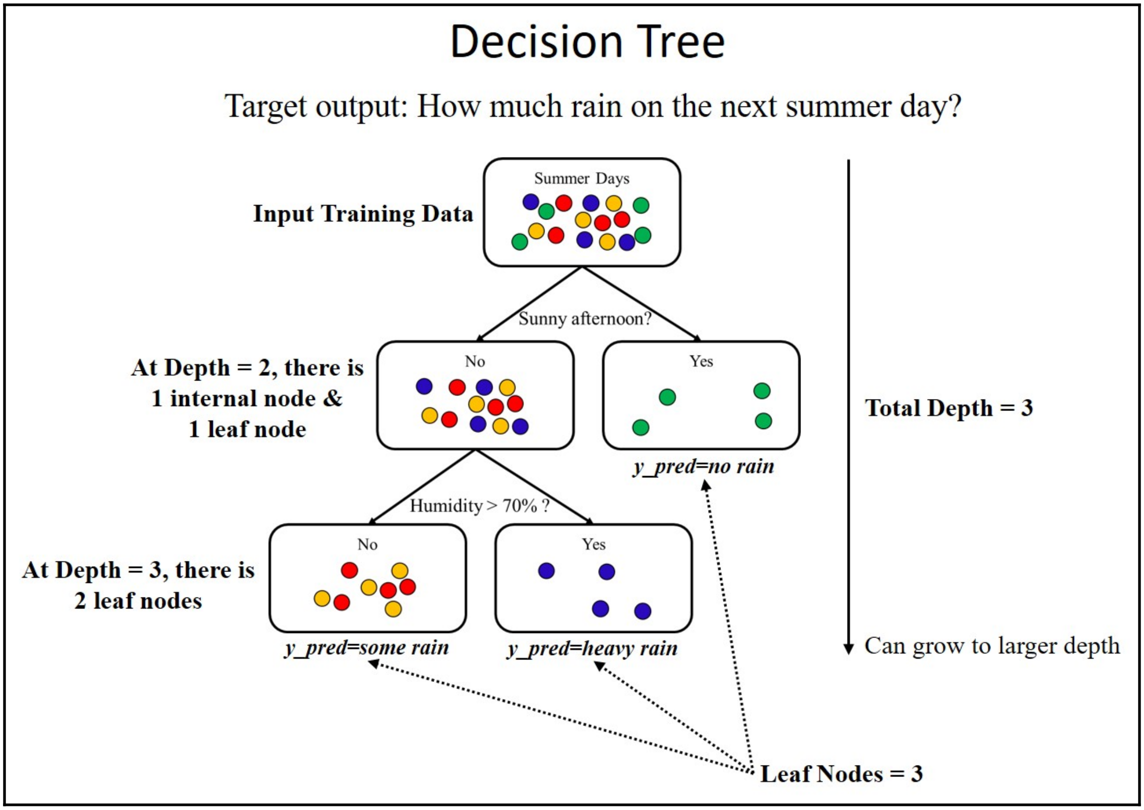

Logistic Regression
F1 Score:0.750
20
5
7
18
top crescent
bottom cresent
True label
top crescent
bottom cresent
Predicted label
20.0
17.5
15.0
12.5
10.0
7.5
5.0

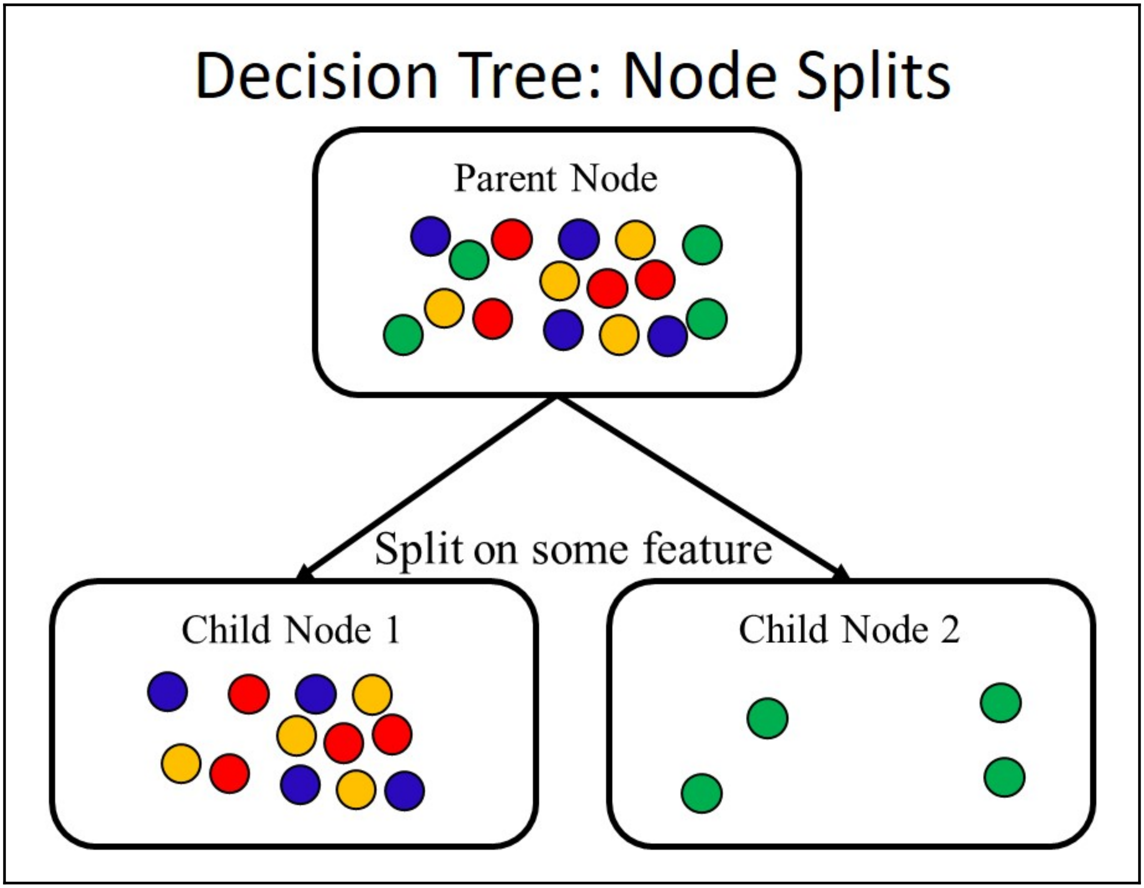

Support Vector Machine: Large Margin Classifier
Logistic Regression Decision Boundary
Support Vector Machine Decision Boundary
Decision Boundary
Margin
Support Vectors
y
x

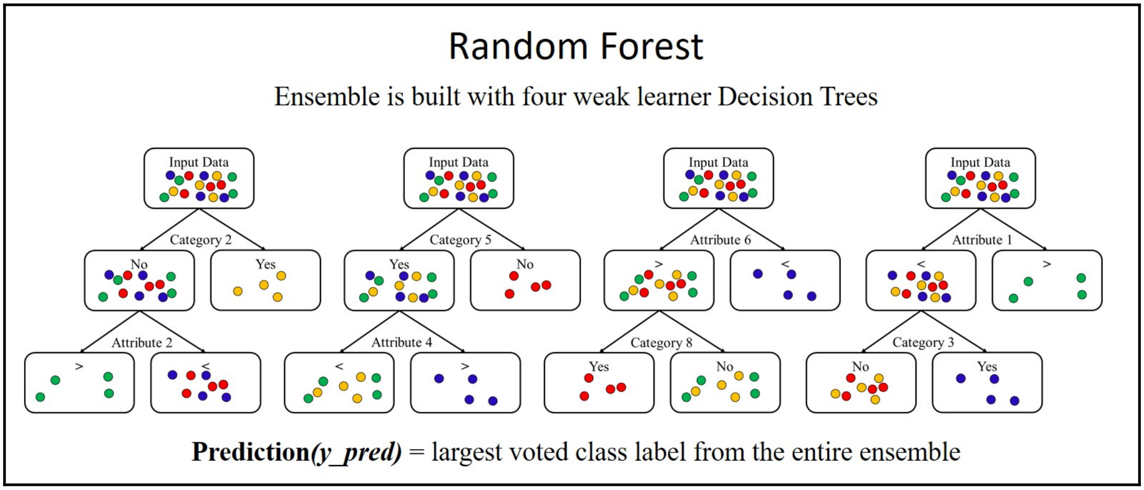

Input Data (X, y)
Logistic Regression
Linear SVM
C-1
C-2
C-6

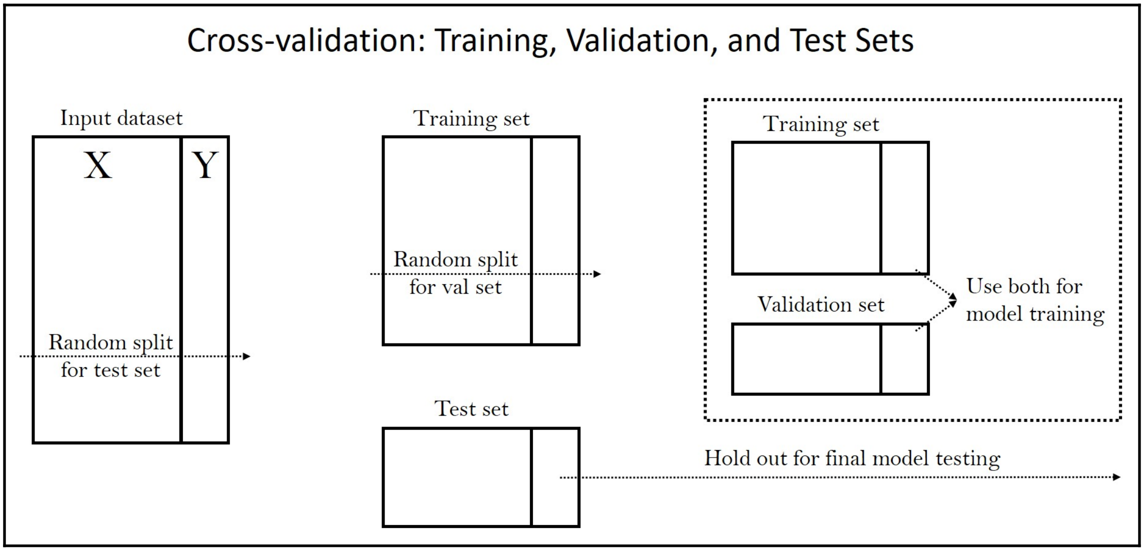

Support Vector Machine: Hinge Loss Function $J(z)$
Loss When Ground Truth $y = 0$
if $y = 0$, force z to be much smaller than 0 by making loss $J(z)$ vanish at values lower than -1
Hinge loss
$J(z)$
-2
-1
1
2
$z = \Theta^T X$
Loss When Ground Truth $y = 1$
if $y = 1$, force z to be much larger than 0 by making loss $J(z)$ vanish at values larger than 1
Hinge loss
$J(z)$
-2
-1
1
2
$z = \Theta^T X$

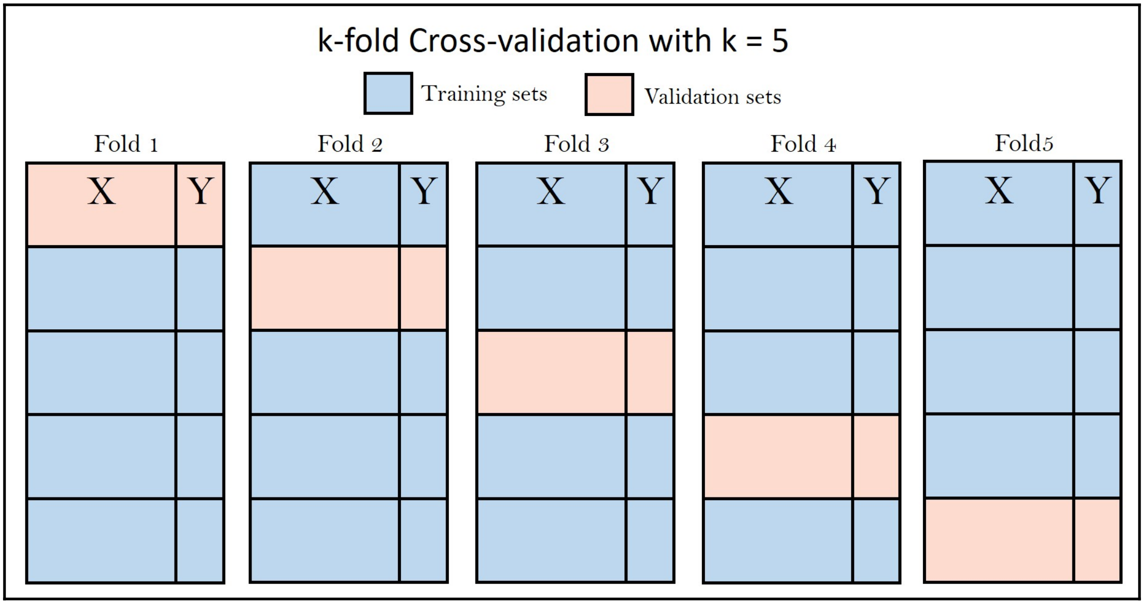

Support Vector Machine: Large Margin Classifier
Support Vectors
$z = \Theta^T X$
Margin
Combined Decision Boundary
Positive Boundary
Negative Boundary
z=1
z=-1
y
x


Decision Tree
Target output: How much rain on the next summer day?
Input Training Data
Summer Days
Sunny afternoon?
No
Yes
At Depth = 2, there is 1 internal node & 1 leaf node
y_pred=no rain
Humidity > 70% ?
No
Yes
At Depth = 3, there is 2 leaf nodes
y_pred=some rain
y_pred=heavy rain
Total Depth = 3
Can grow to larger depth
Leaf Nodes = 3


Decision Tree: Node Splits
Parent Node
Split on some feature
Child Node 1
Child Node 2


Random Forest
Ensemble is built with four weak learner Decision Trees
Input Data
Category 2
No
Yes
Attribute 2
Category 5
Attribute 4
Attribute 6
Category 8
Attribute 1
Category 3
Prediction(y_pred) = largest voted class label from the entire ensemble


Cross-validation: Training, Validation, and Test Sets
Input dataset
X
Y
Random split
for test set
Training set
Random split
for val set
Test set
Training set
Validation set
Use both for
model training
Hold out for final model testing


k-fold Cross-validation with k = 5
Training sets
Validation sets
Fold 1
Fold 2
Fold 3
Fold 4
Fold5
X
Y