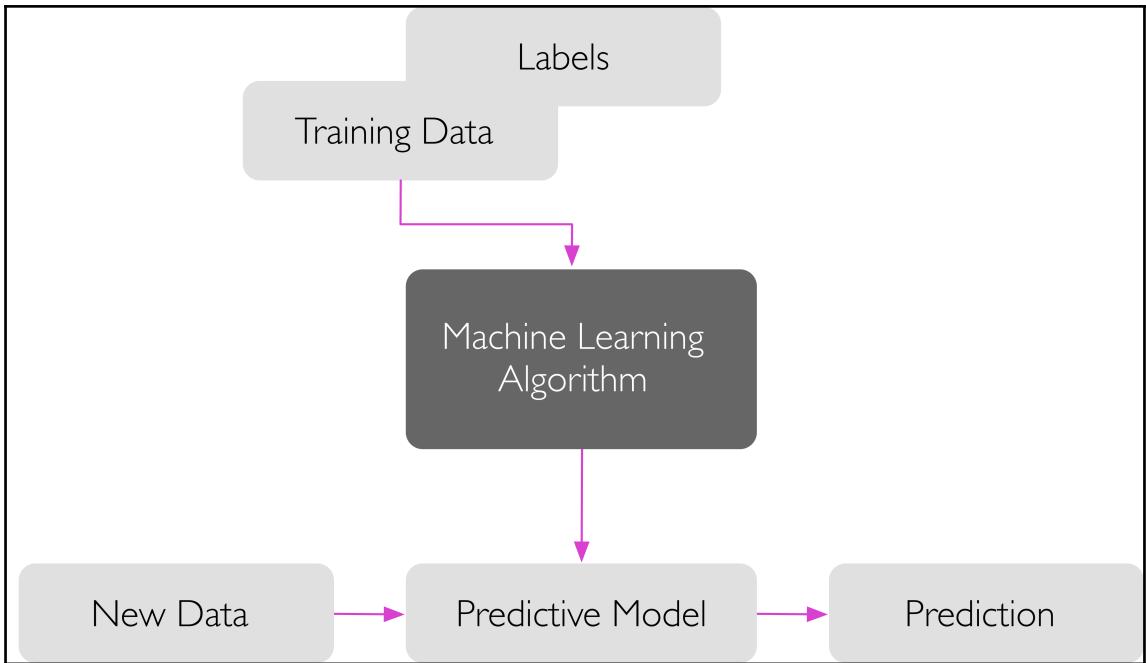
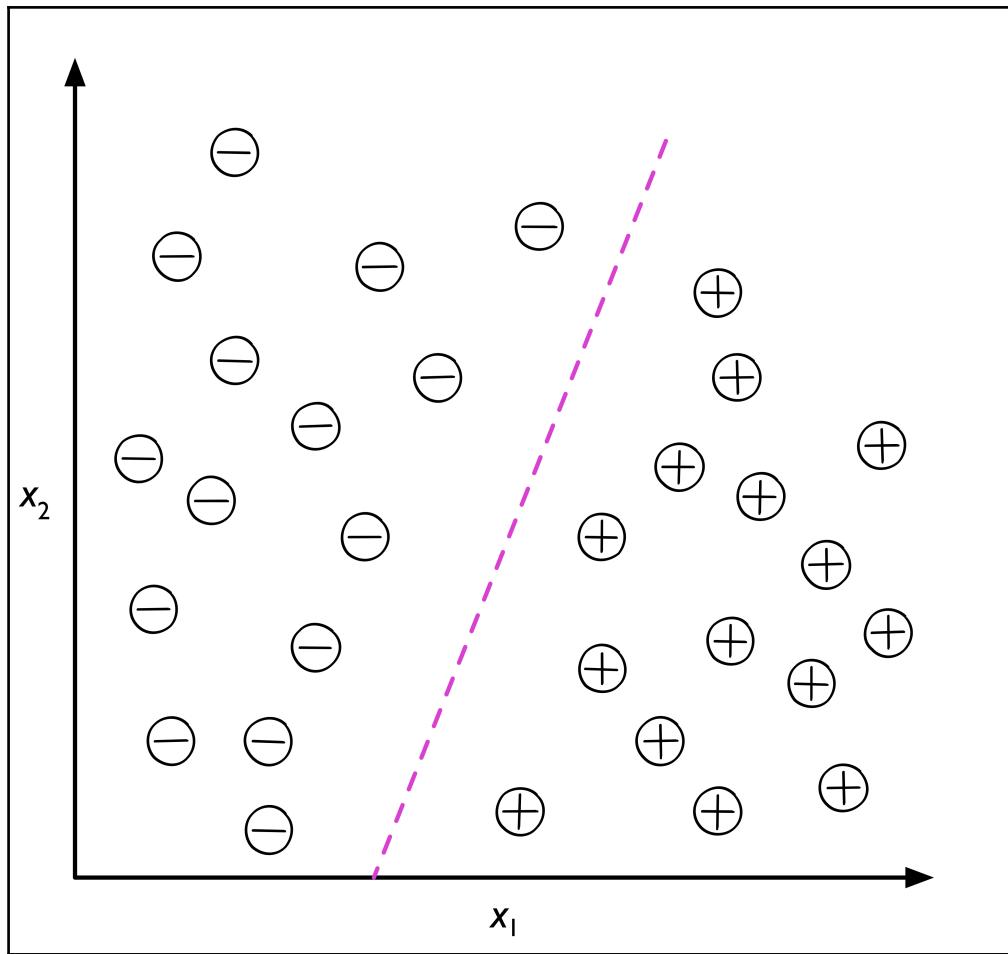
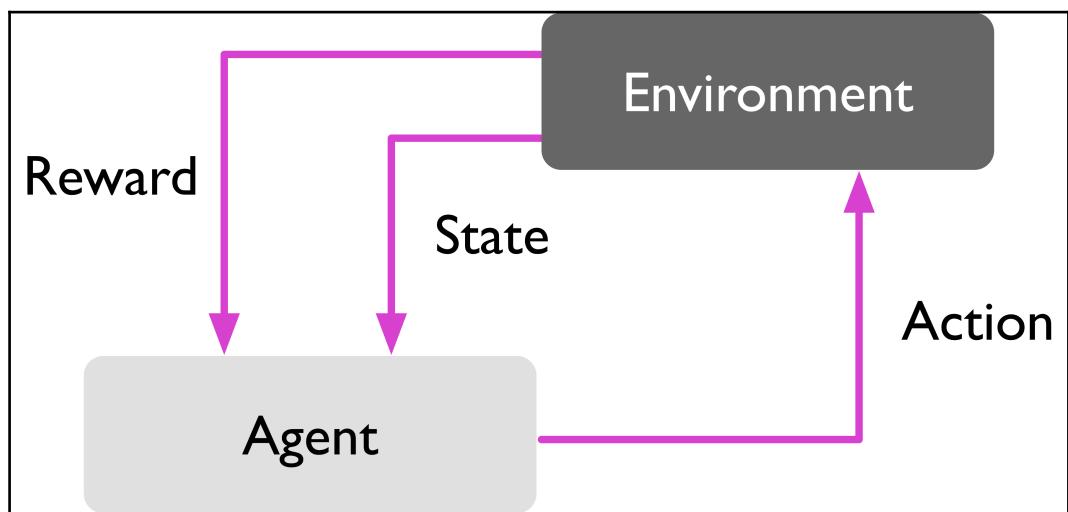
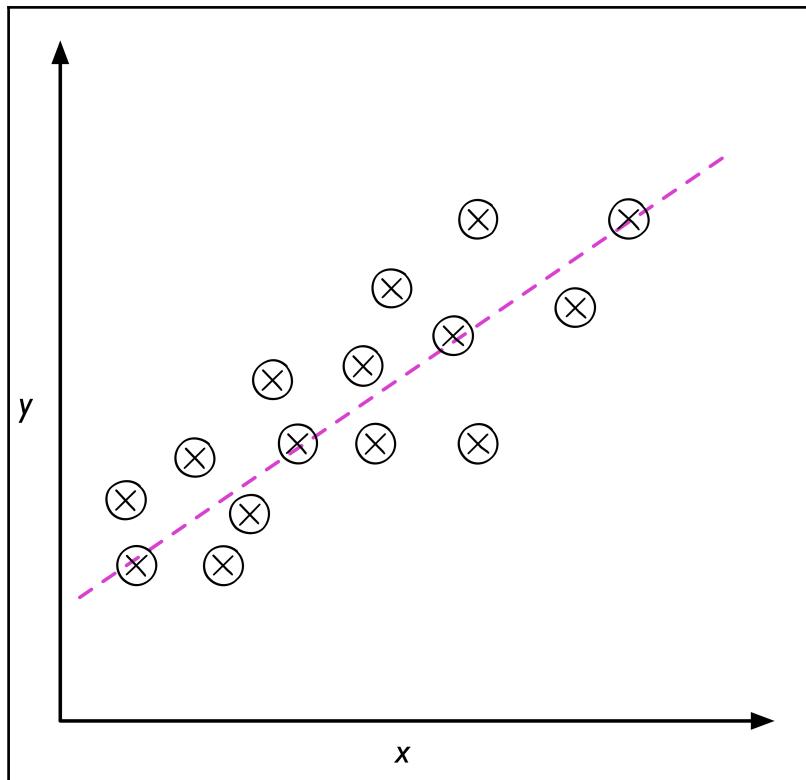


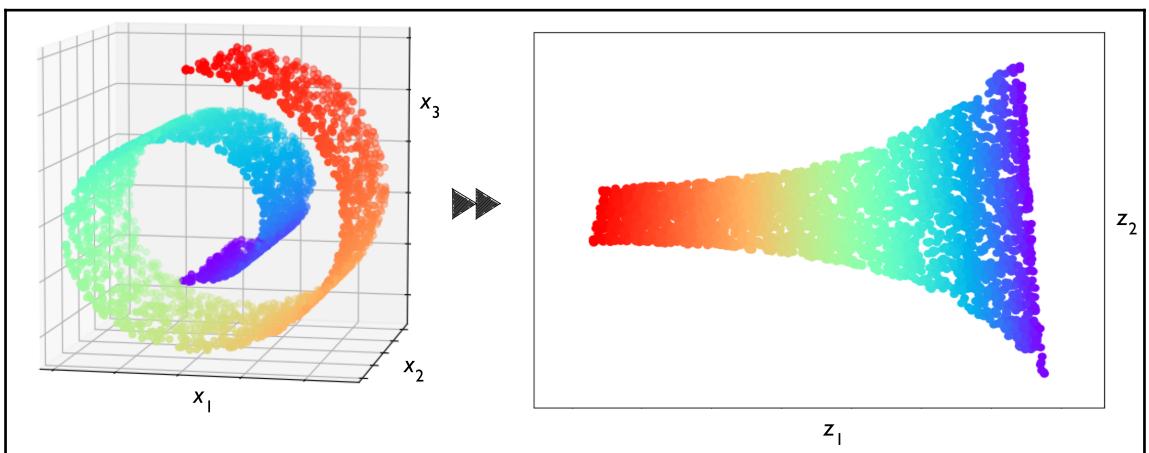
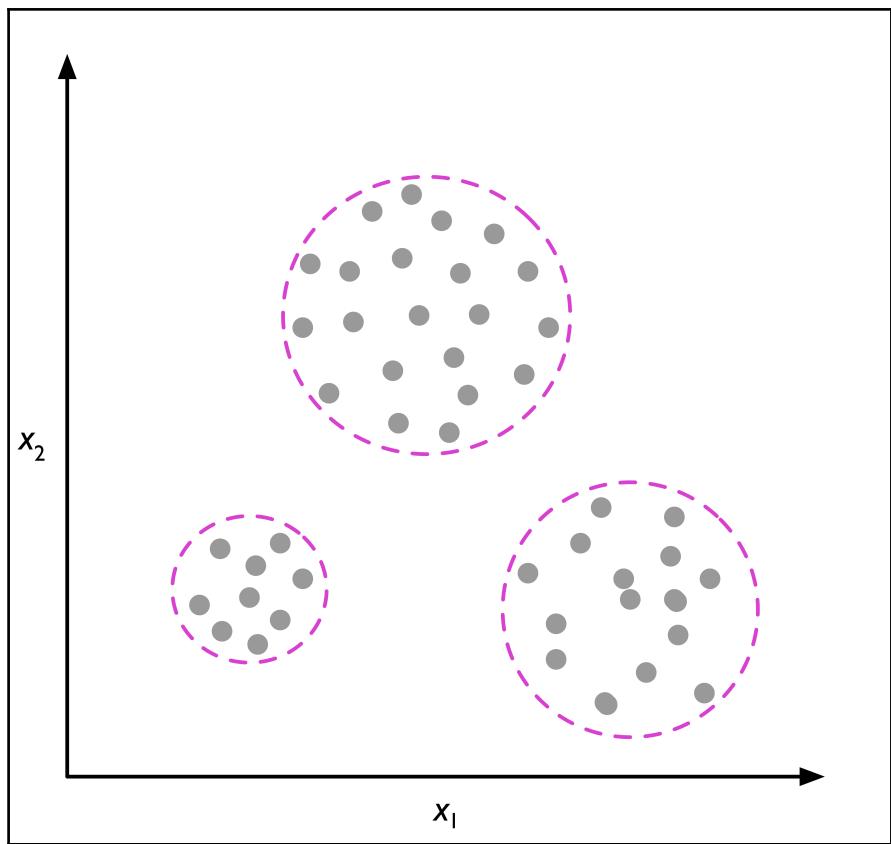
# Chapter 1: Giving Computers the Ability to Learn from Data

<b>Supervised Learning</b>	<ul style="list-style-type: none"><li>➤ Labeled data</li><li>➤ Direct feedback</li><li>➤ Predict outcome/future</li></ul>
<b>Unsupervised Learning</b>	<ul style="list-style-type: none"><li>➤ No labels</li><li>➤ No feedback</li><li>➤ Find hidden structure in data</li></ul>
<b>Reinforcement Learning</b>	<ul style="list-style-type: none"><li>➤ Decision process</li><li>➤ Reward system</li><li>➤ Learn series of actions</li></ul>







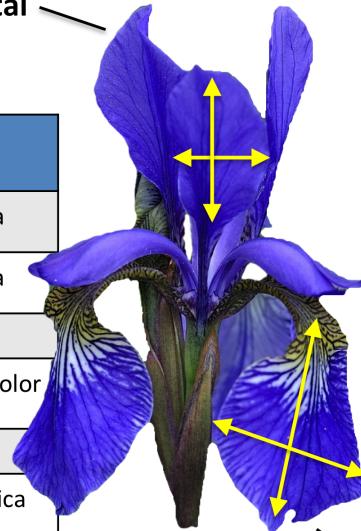


**Samples**

(instances, observations)

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

Petal

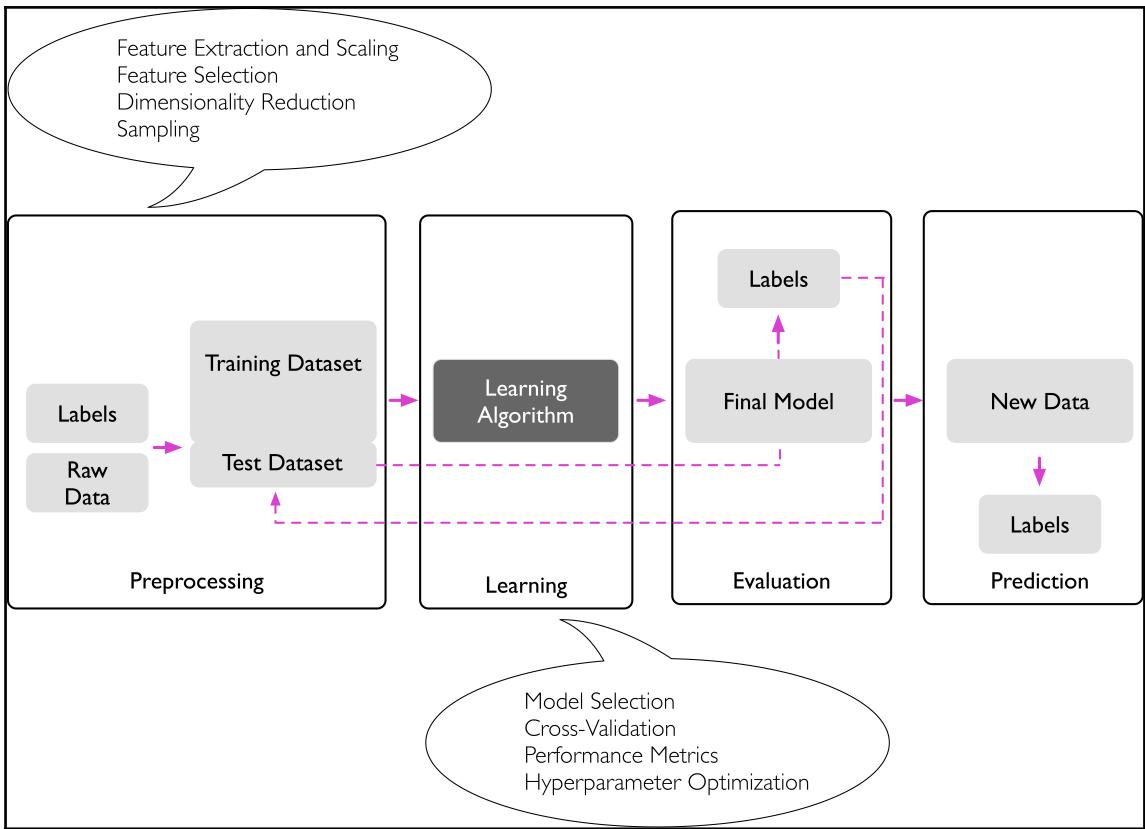


Sepal

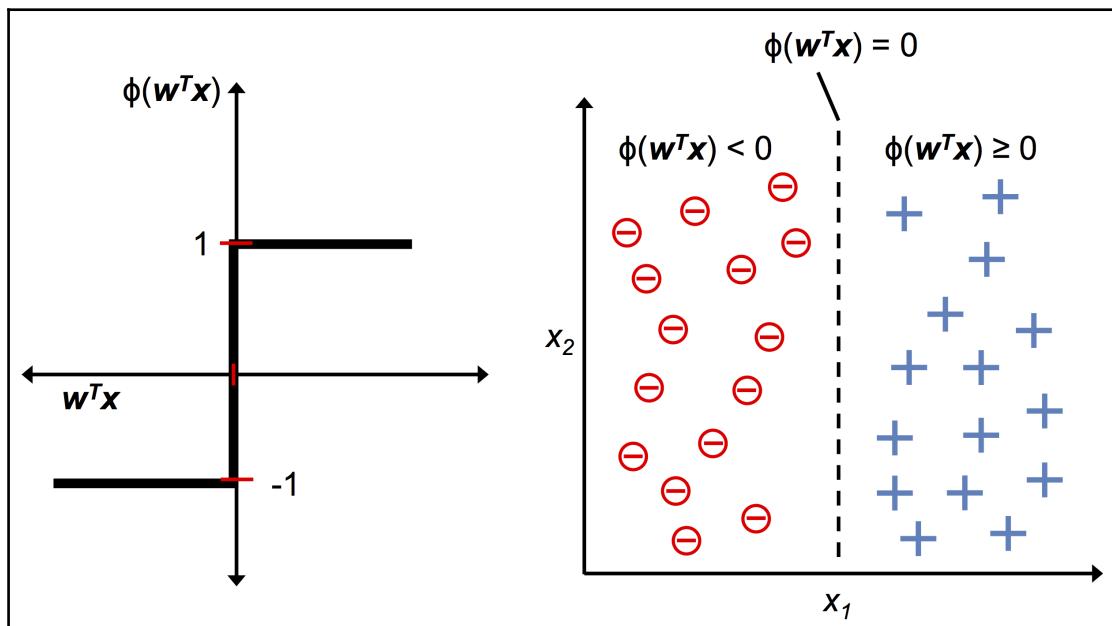
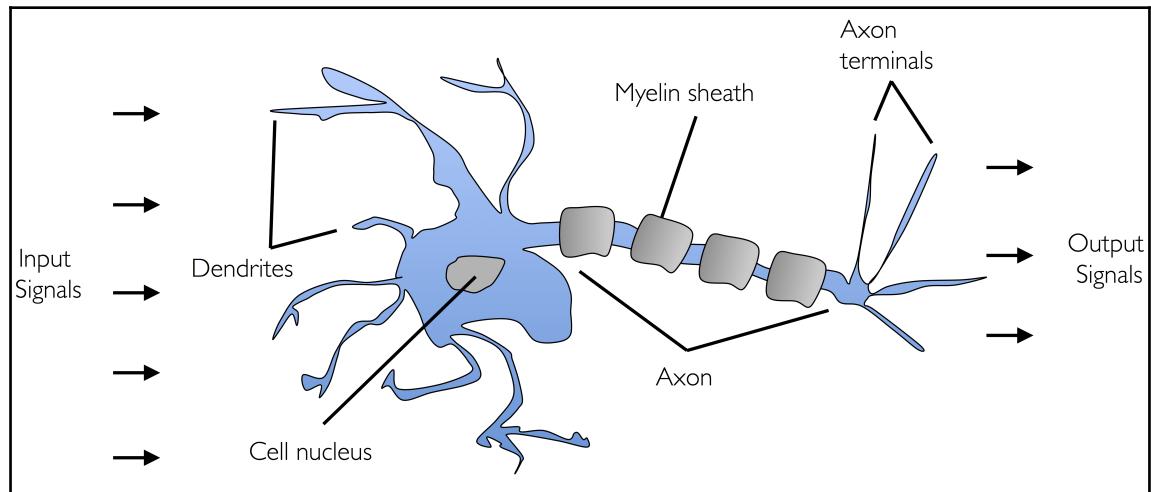
**Features**

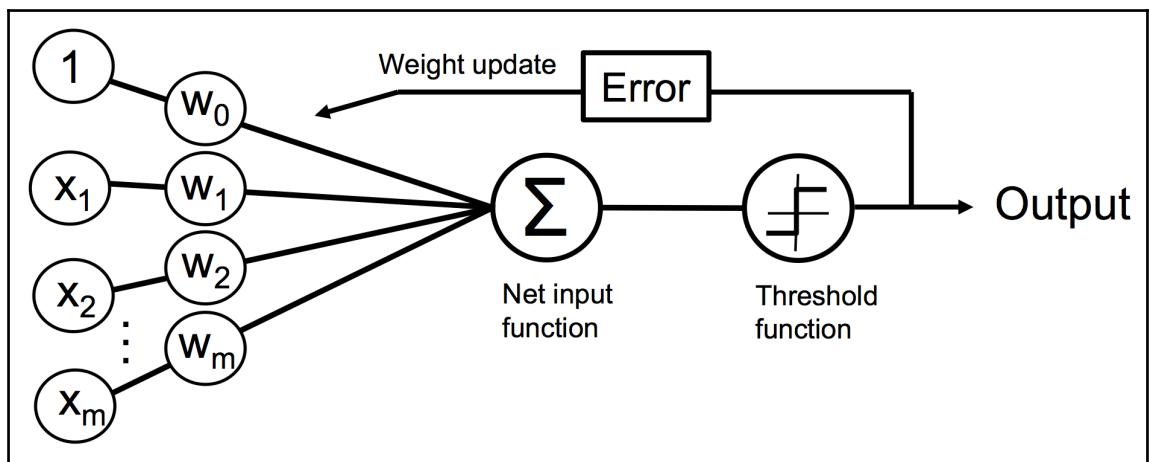
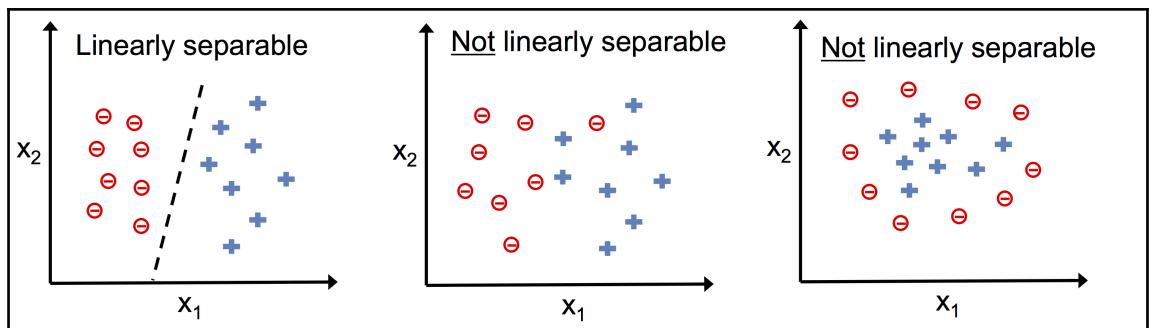
(attributes, measurements, dimensions)

Class labels  
(targets)

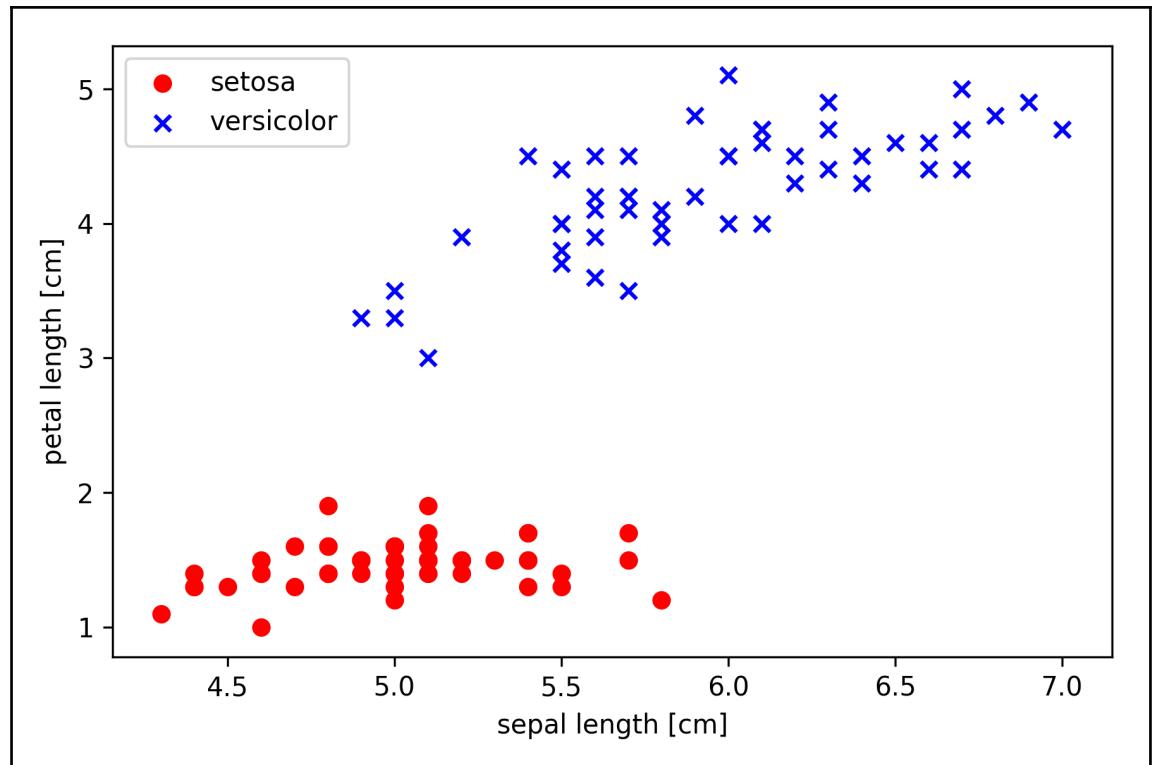


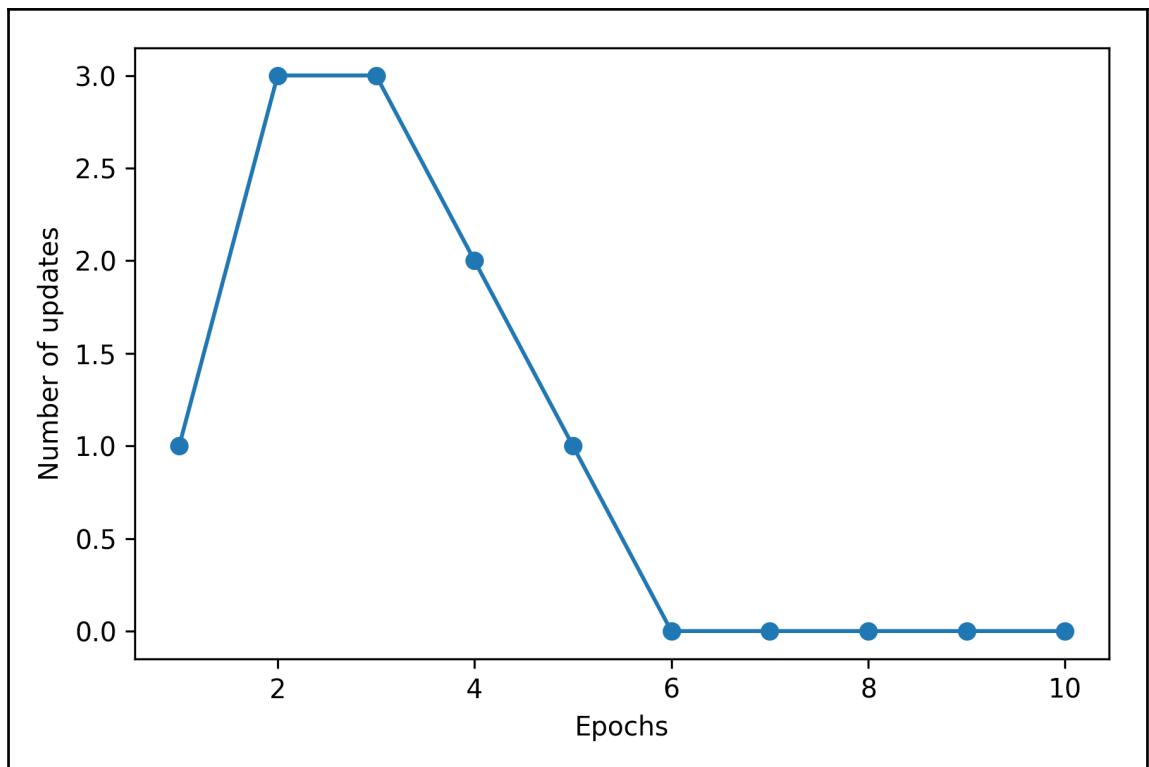
# Chapter 2: Training Simple Machine Learning Algorithms for Classification

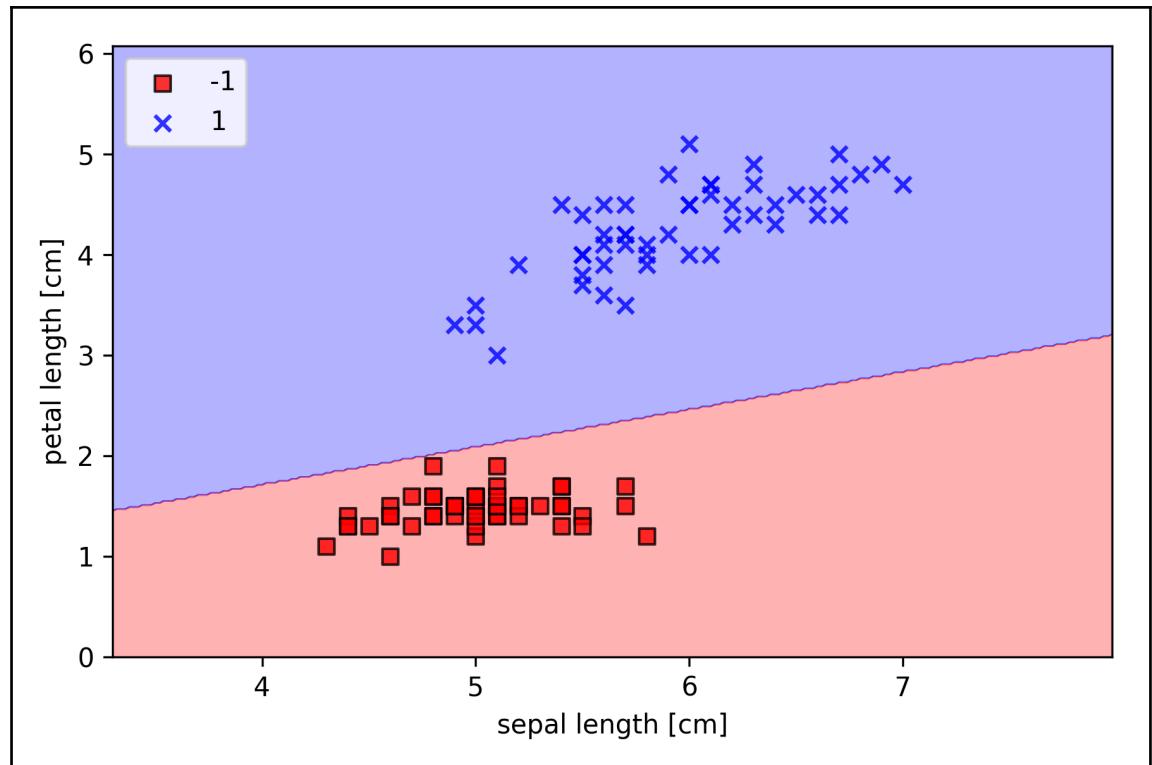


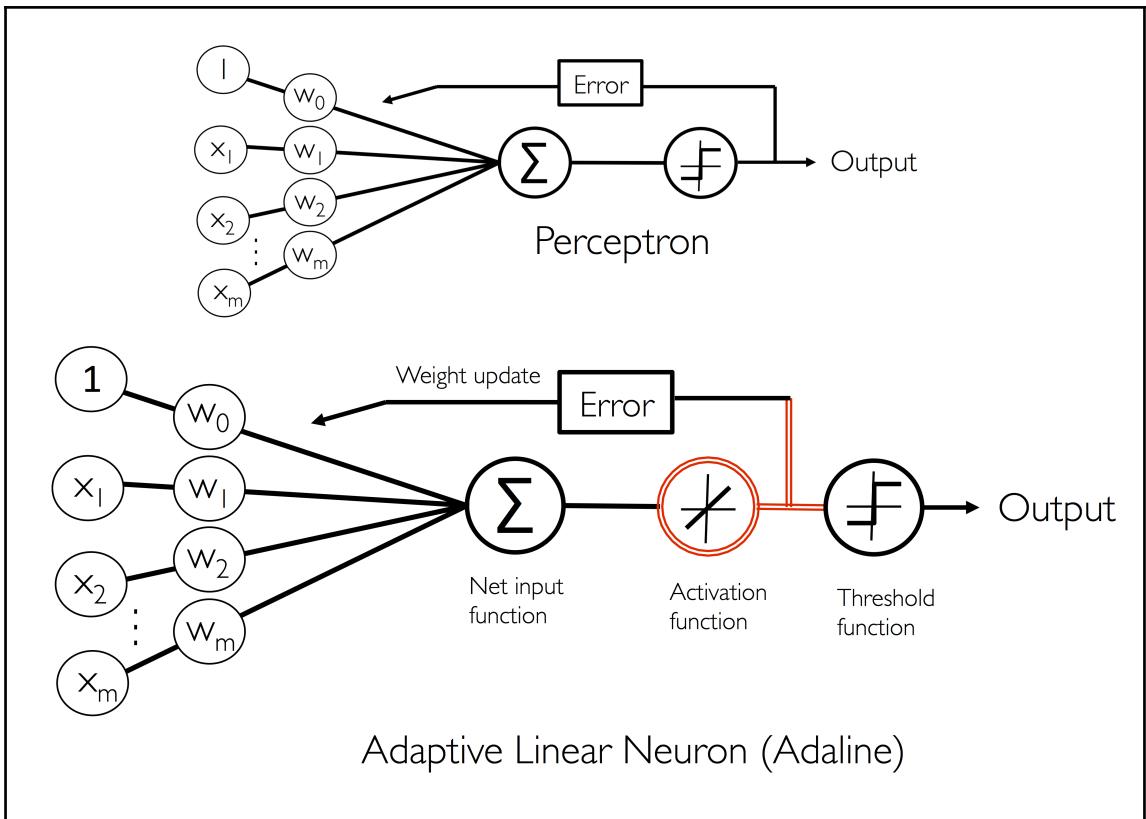


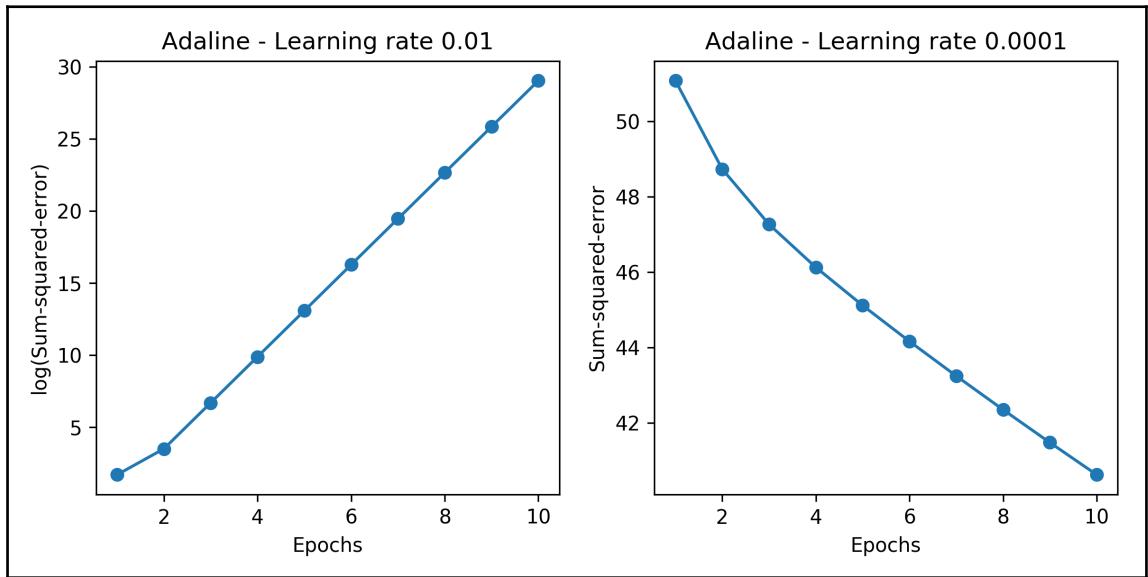
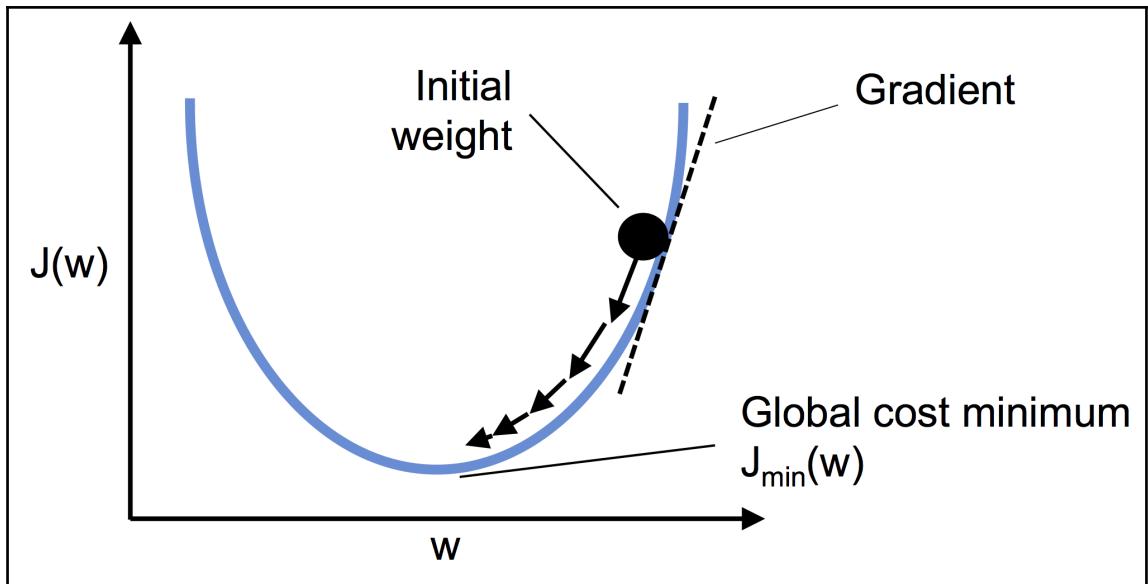
	0	1	2	3	4
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

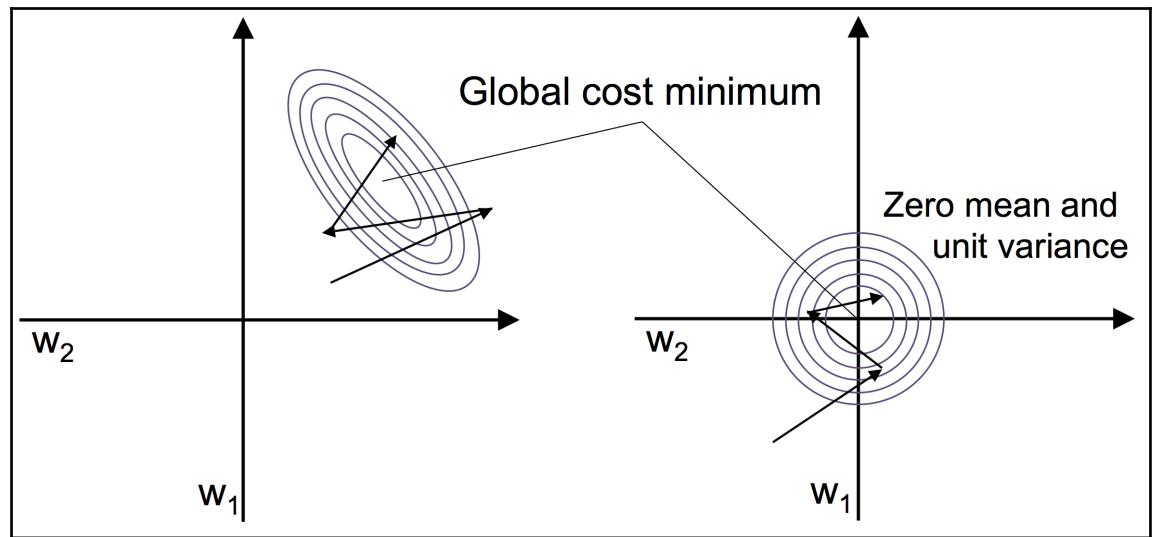
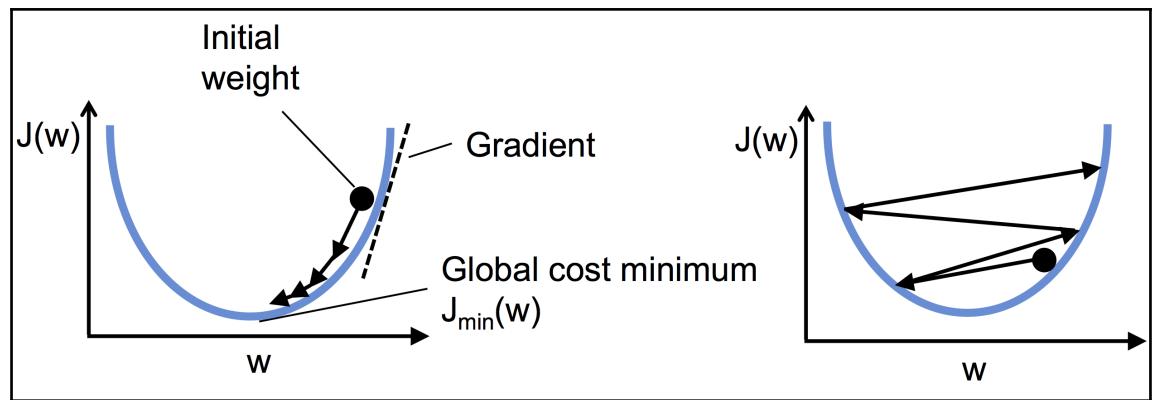


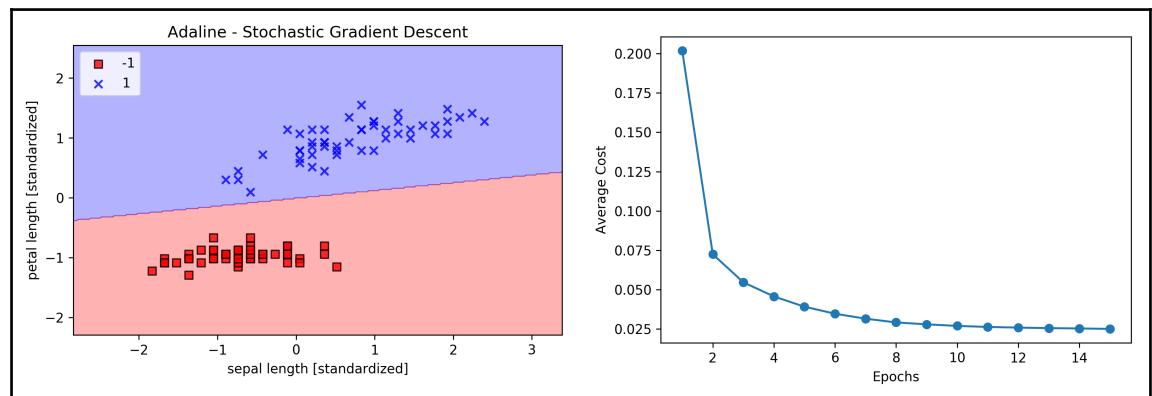
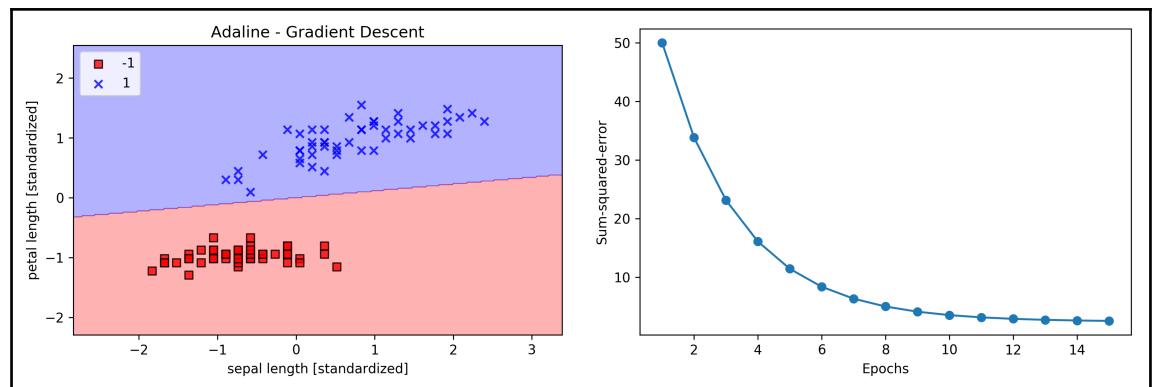




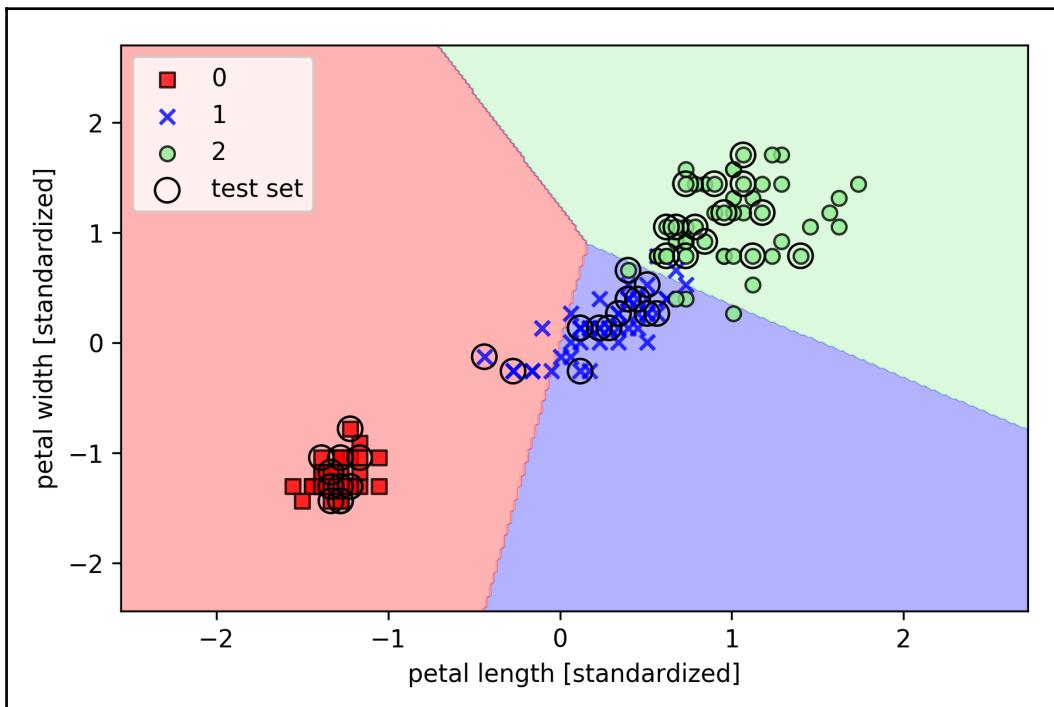


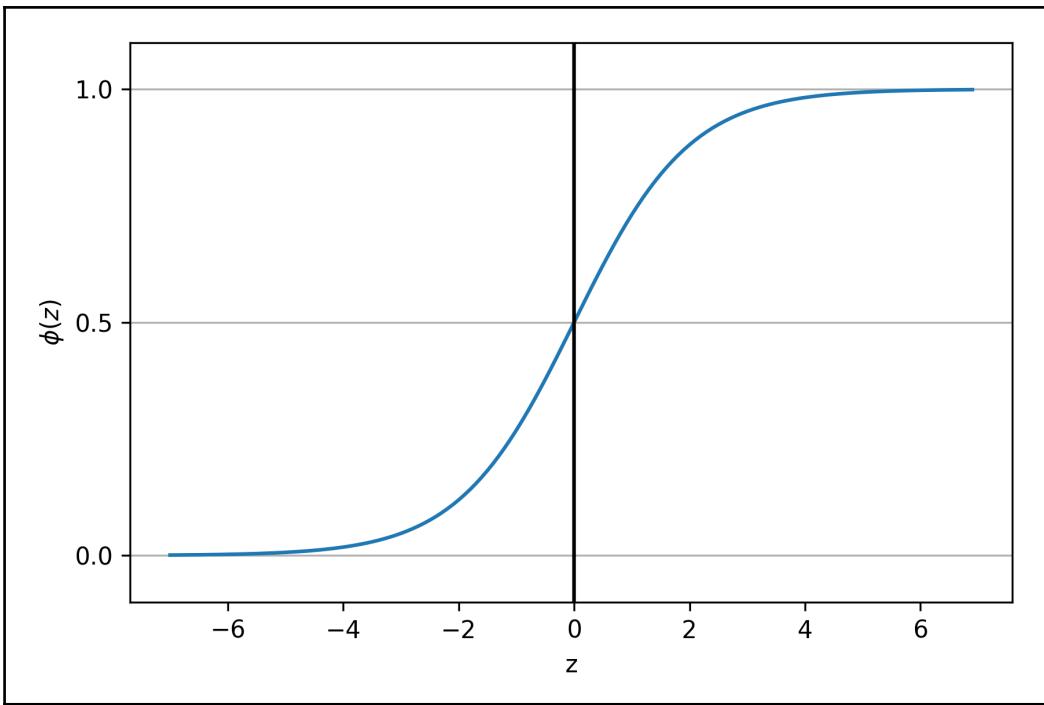


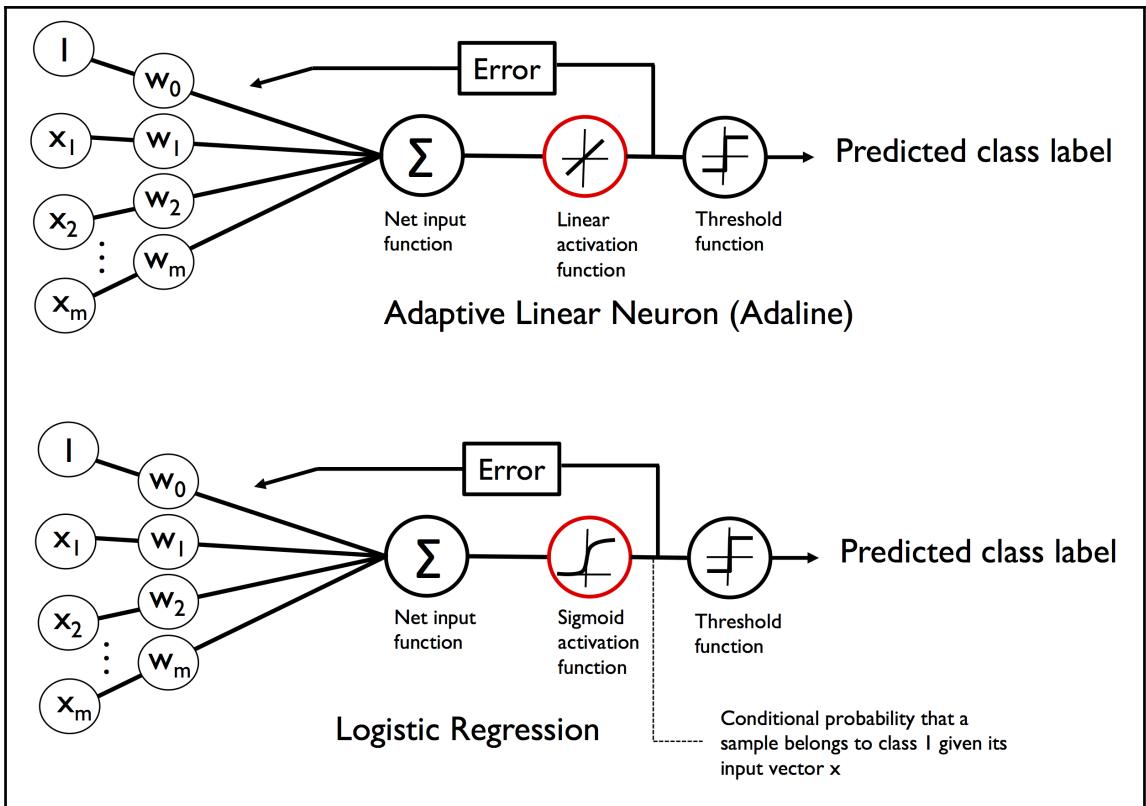


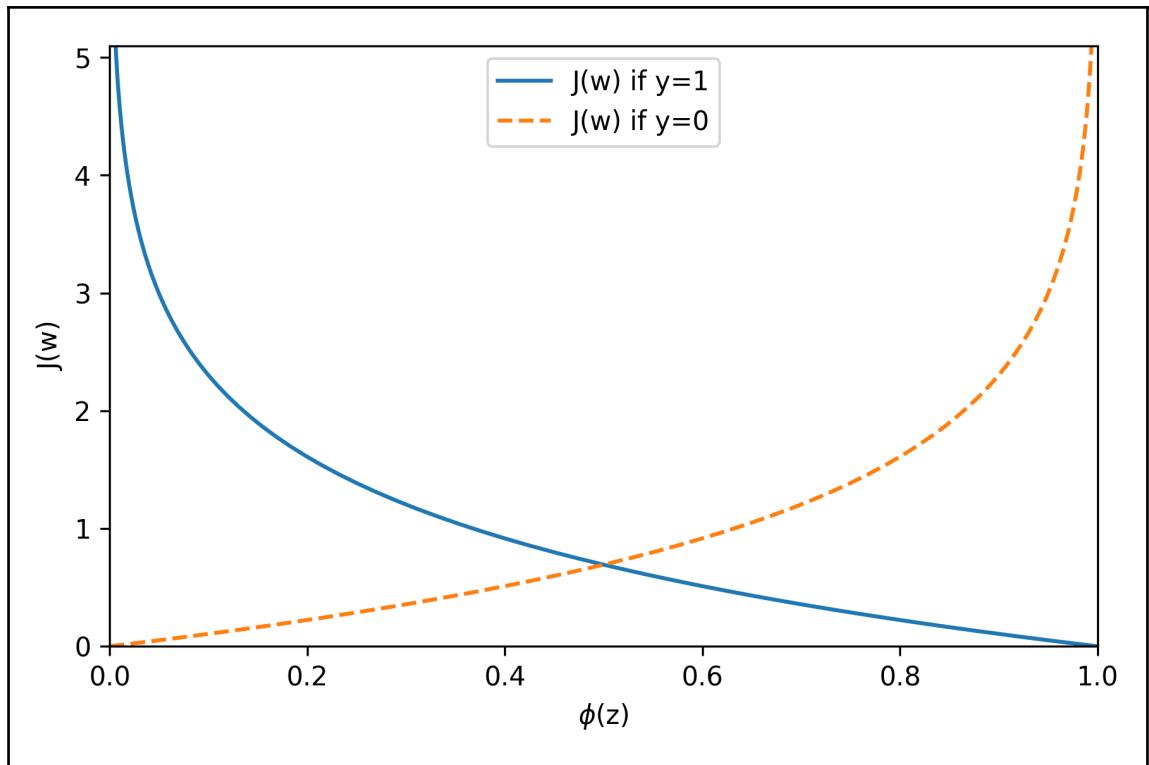


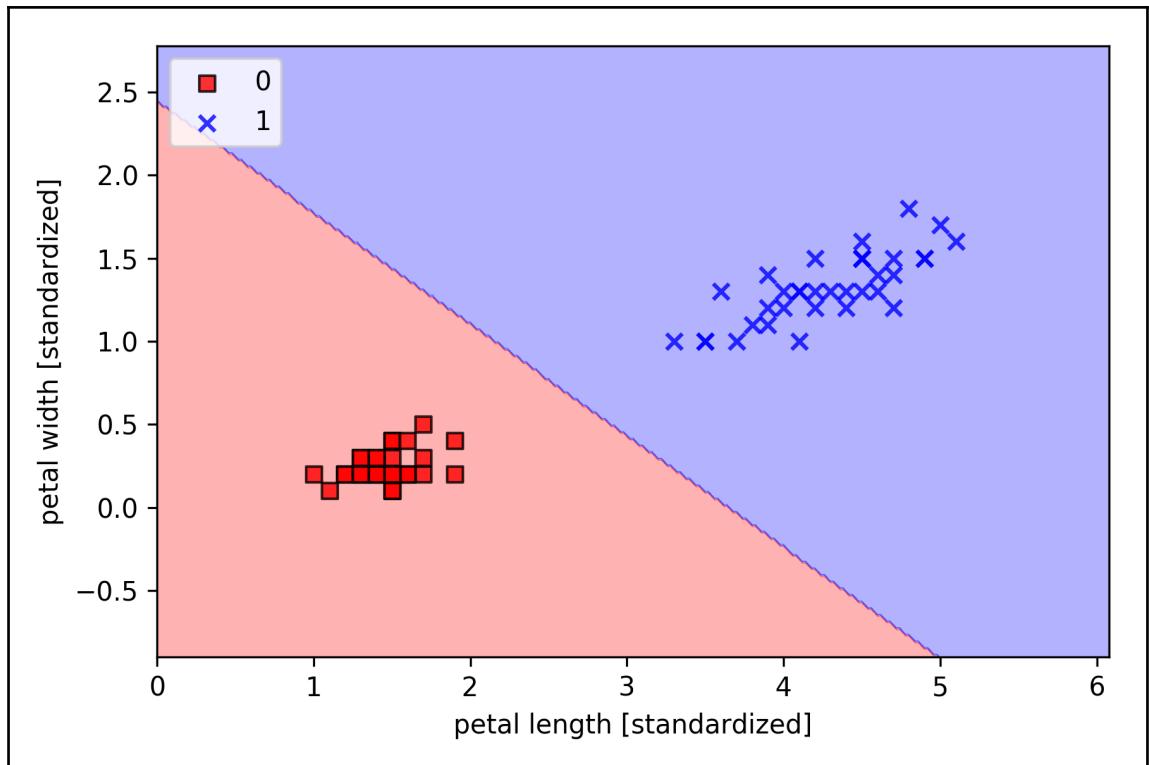
# Chapter 3: A Tour of Machine Learning Classifiers Using scikit-learn



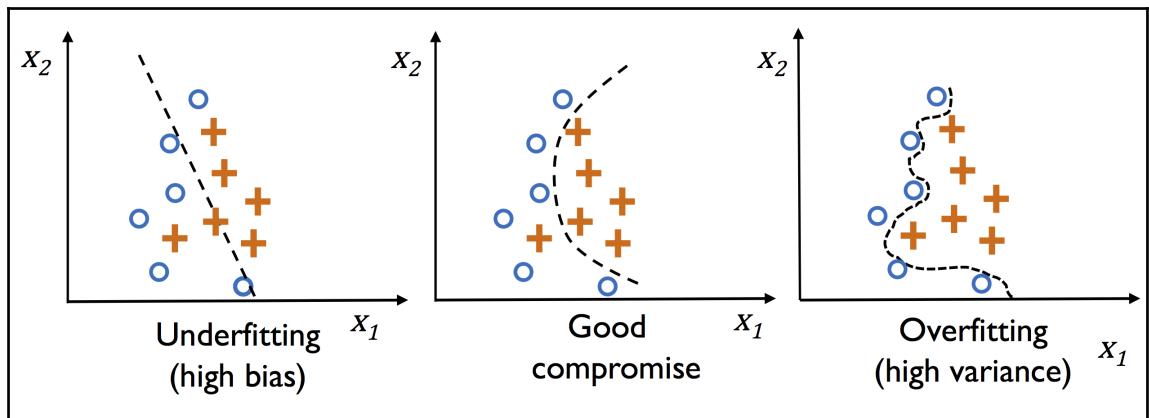
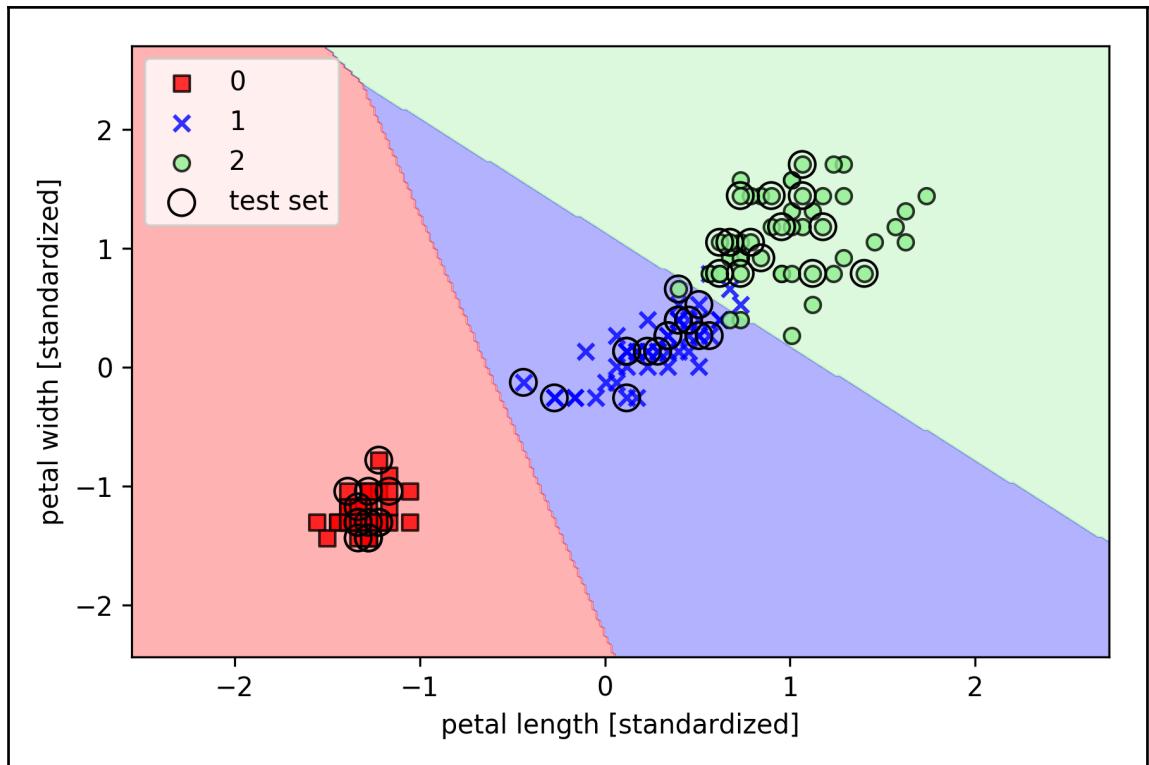


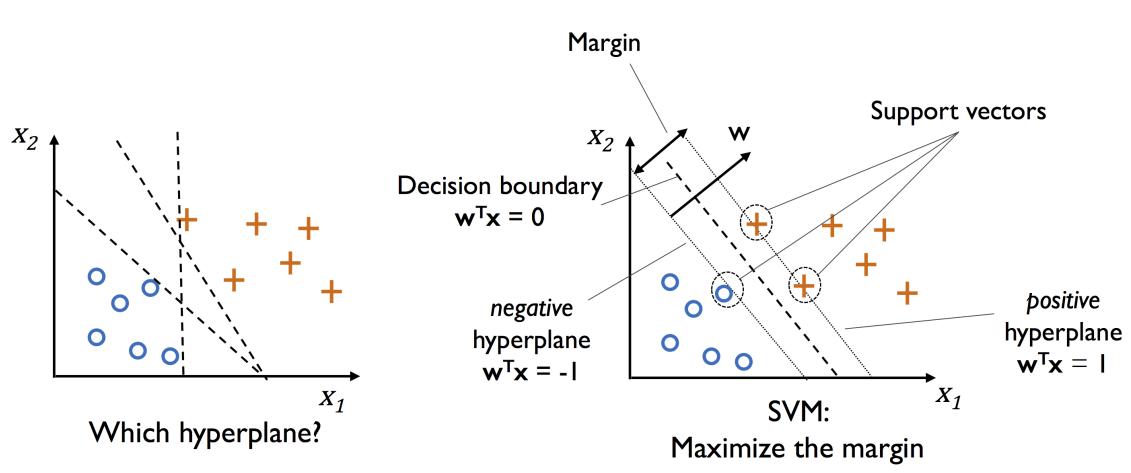
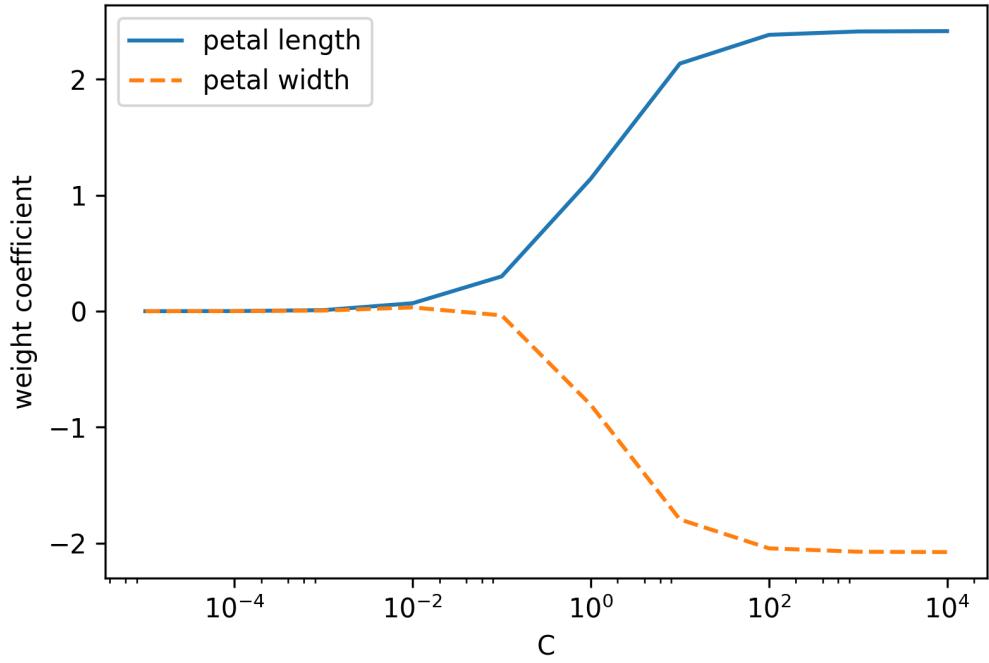


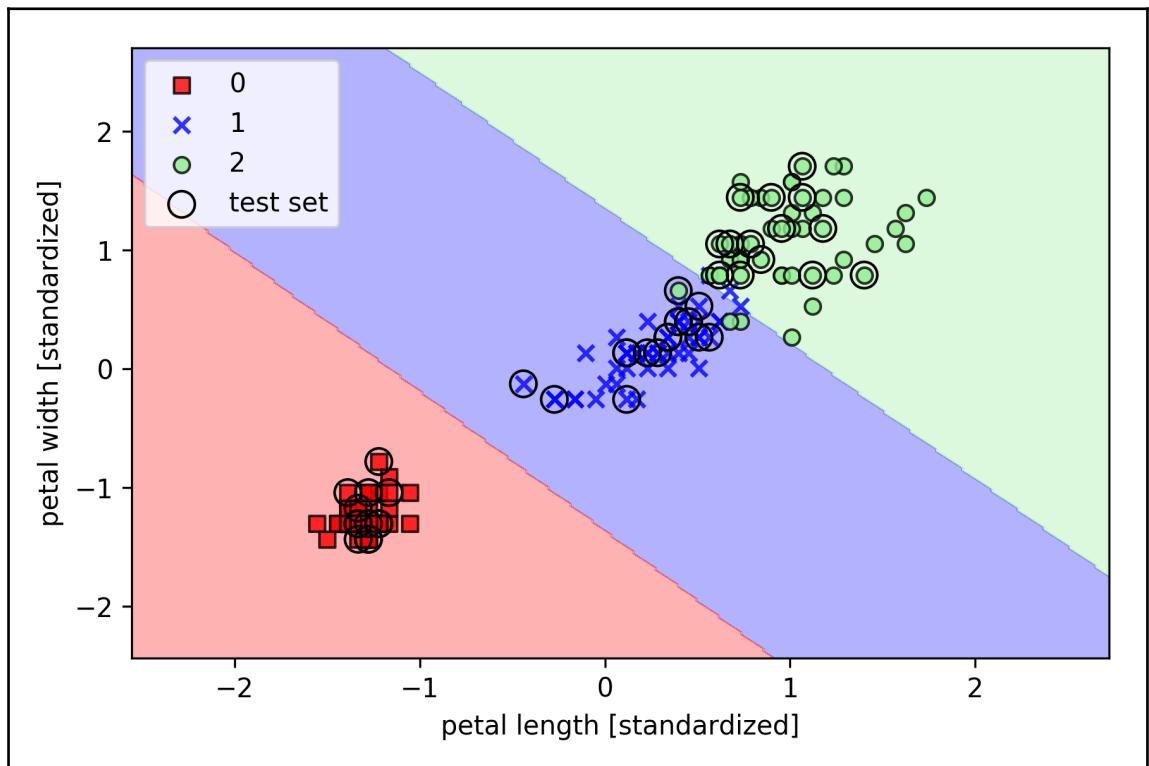
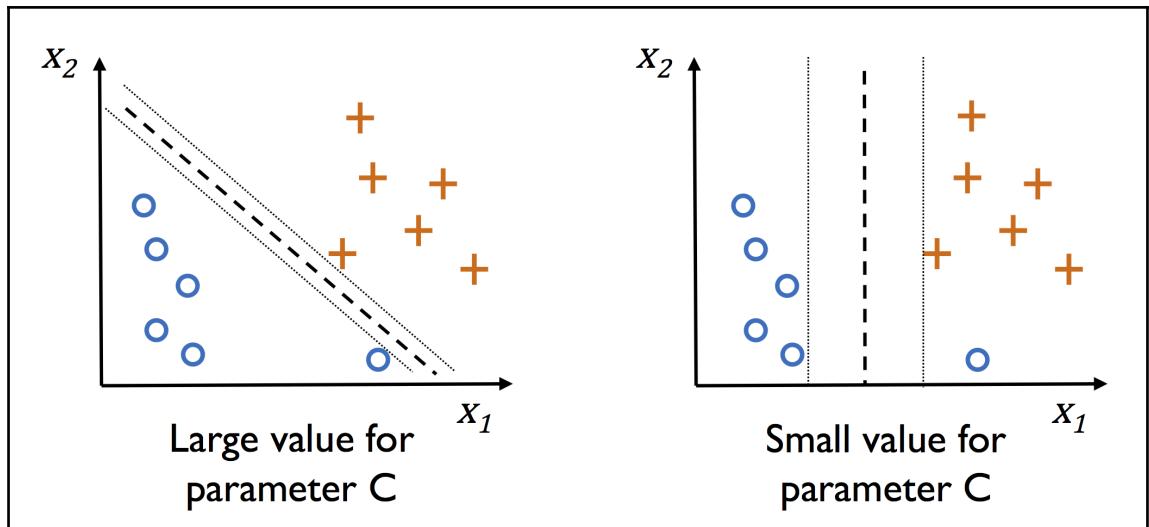


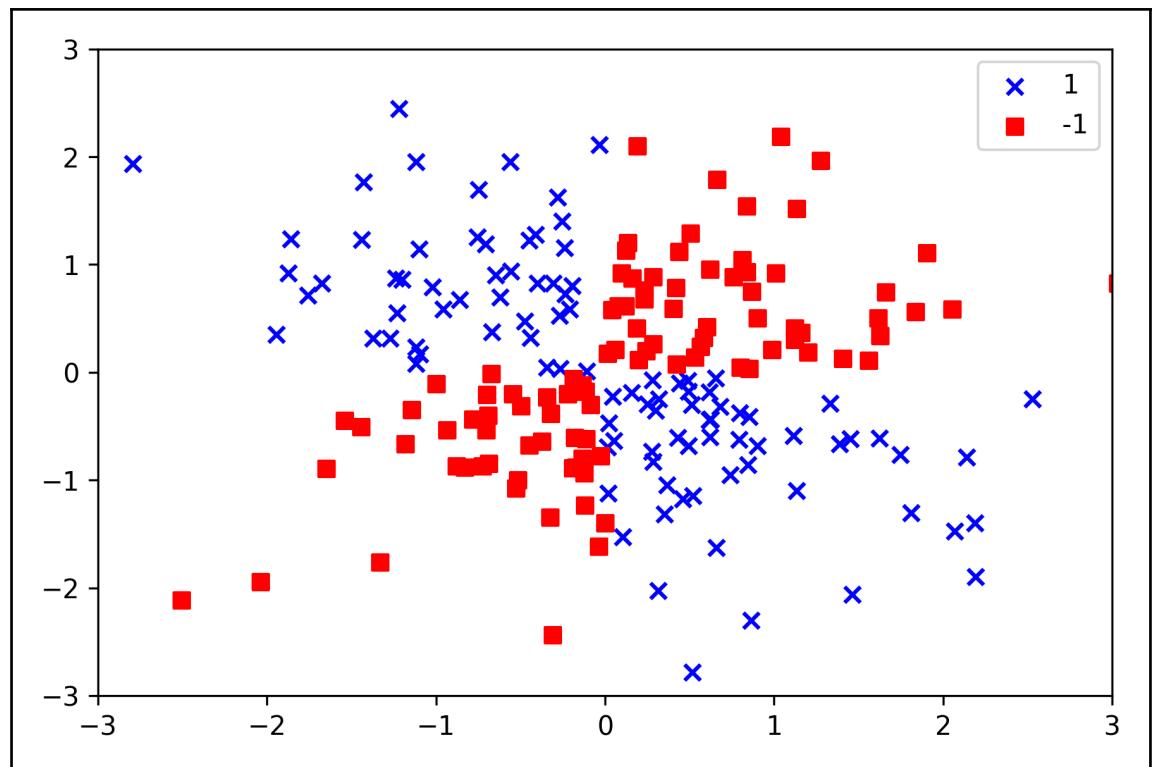


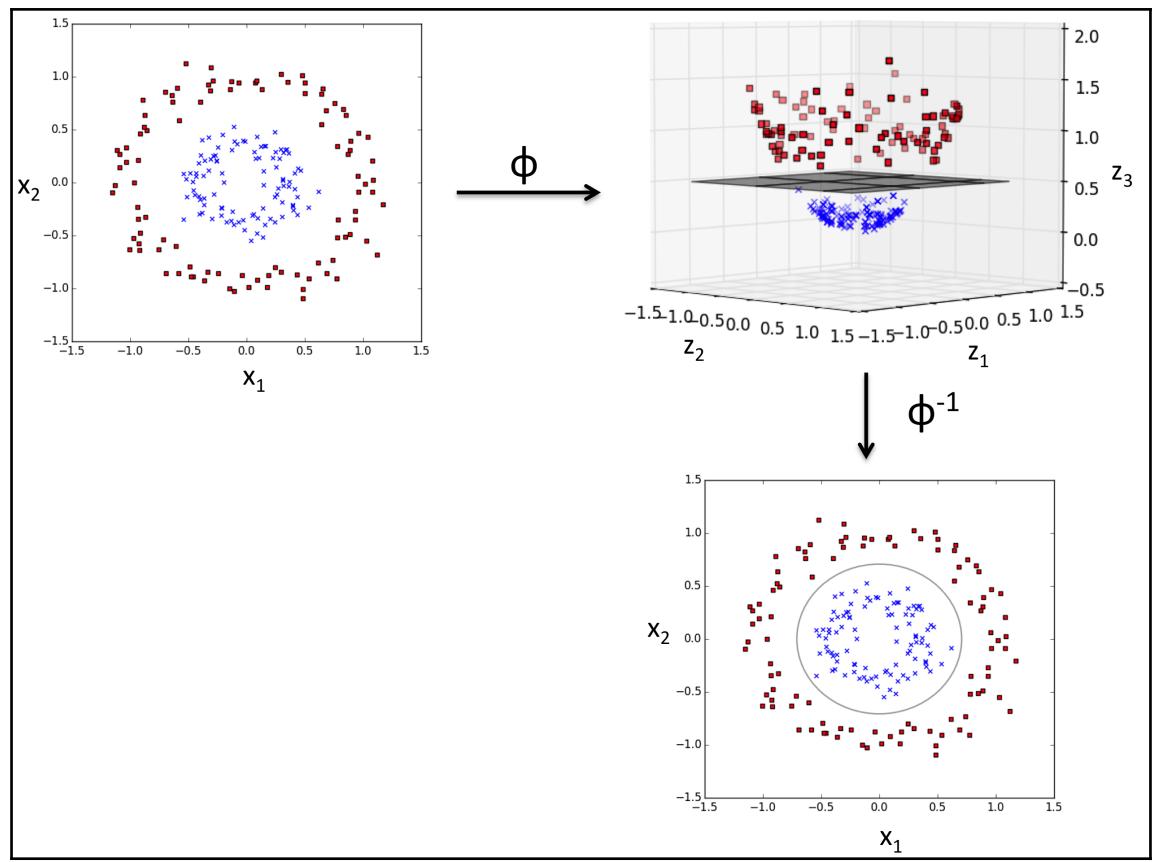
>

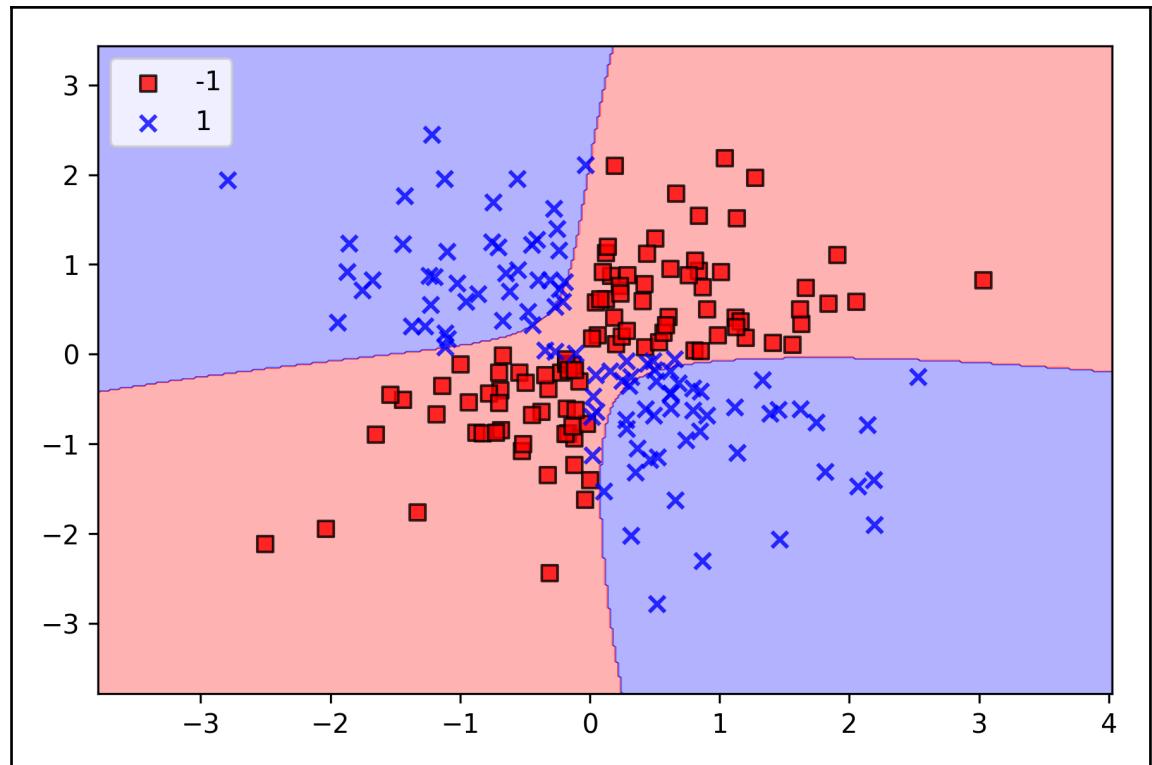


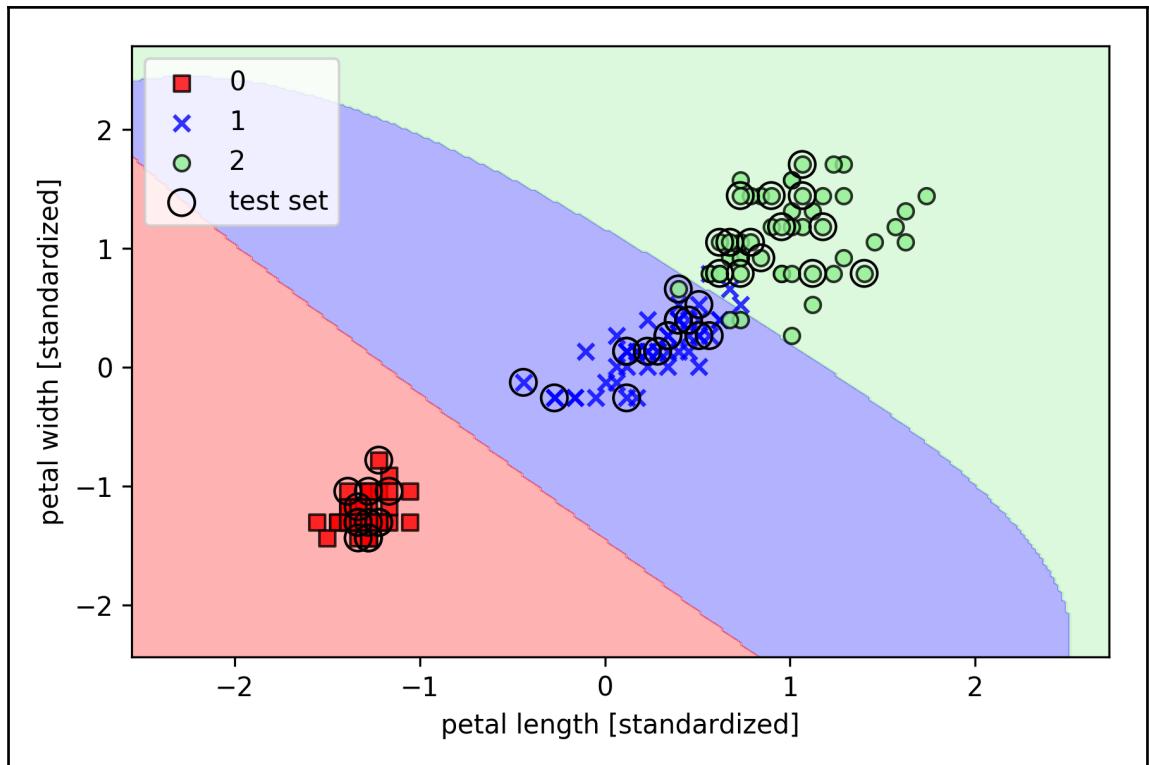


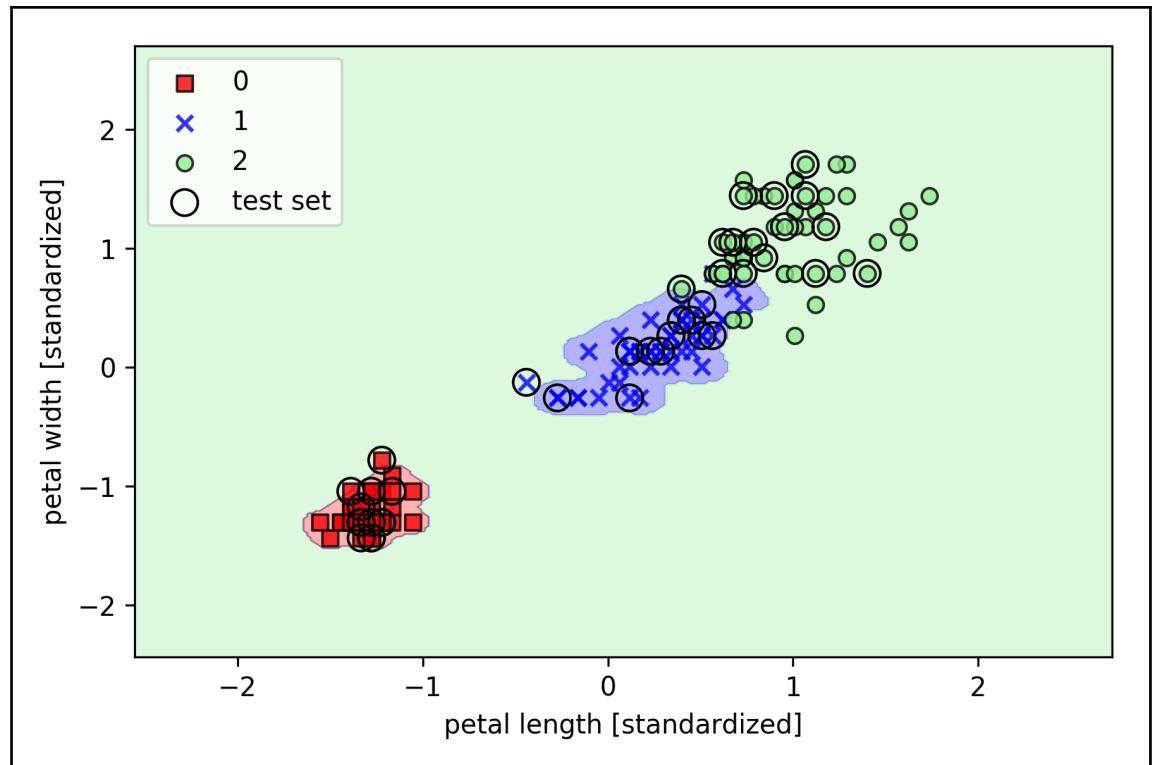


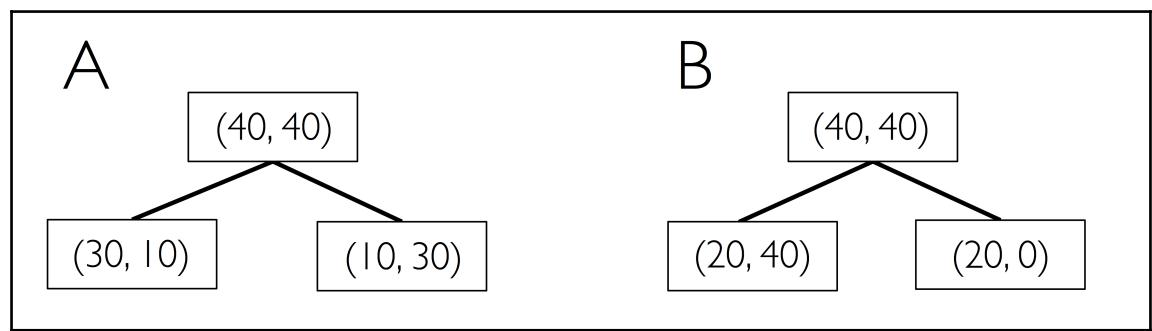
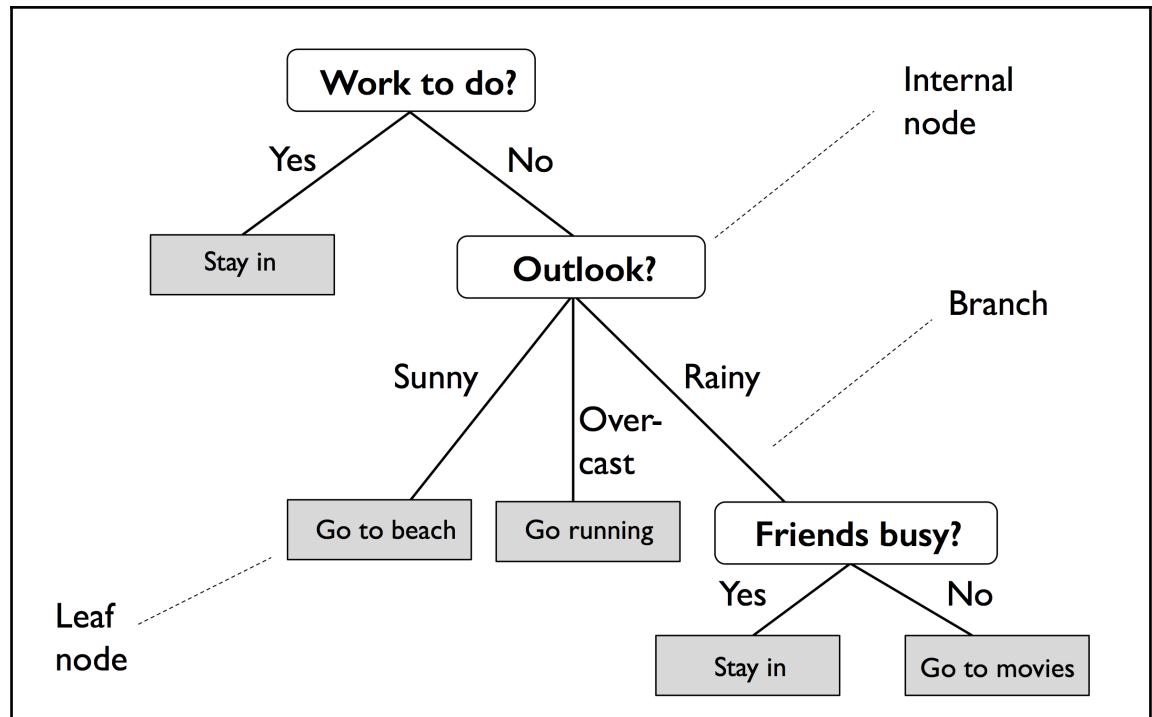


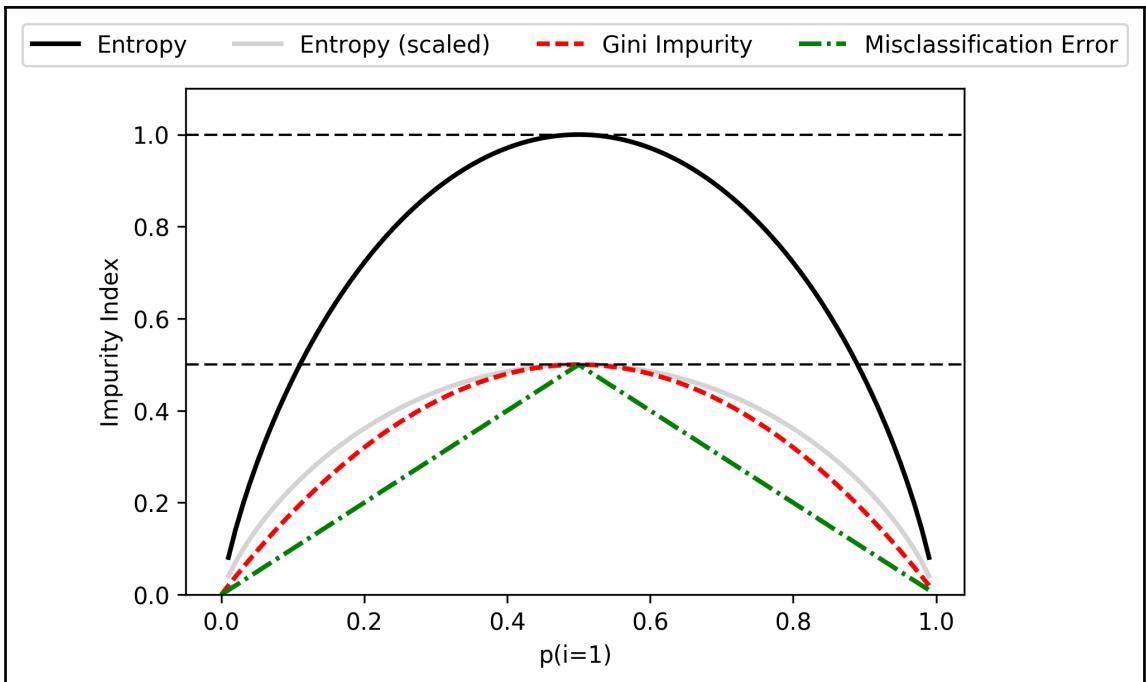


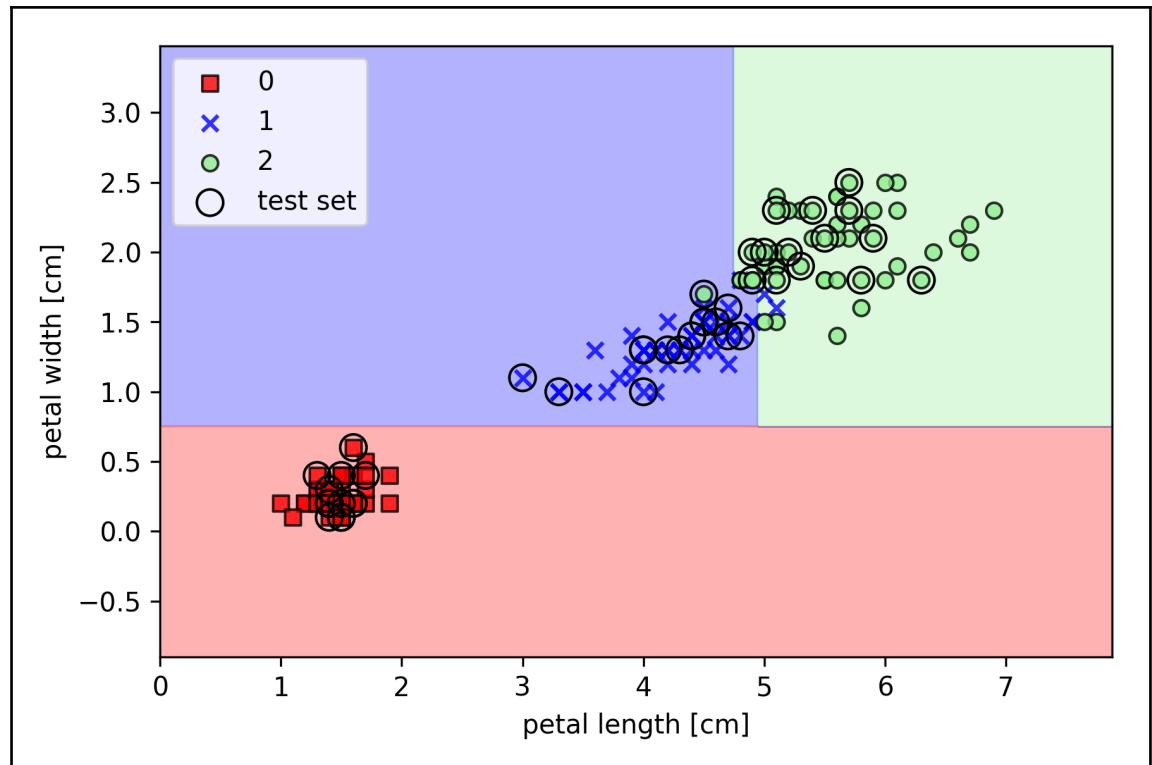


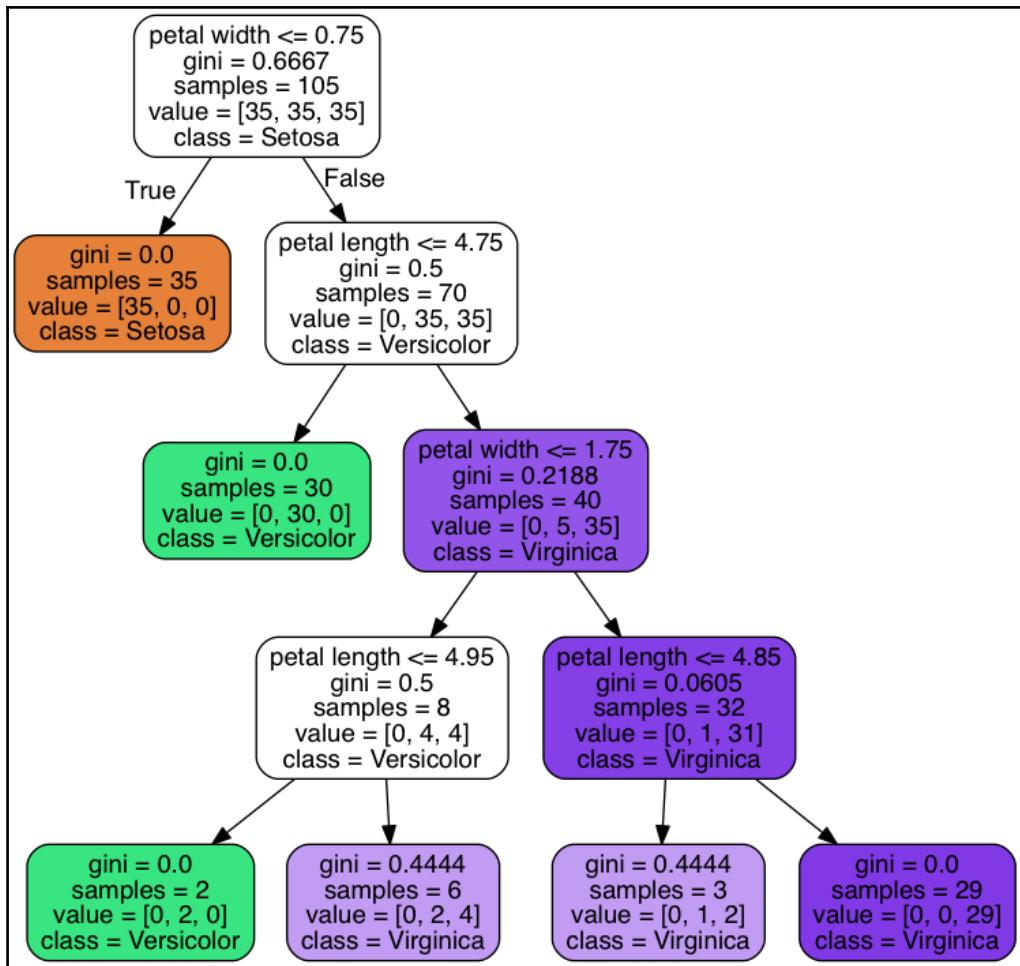


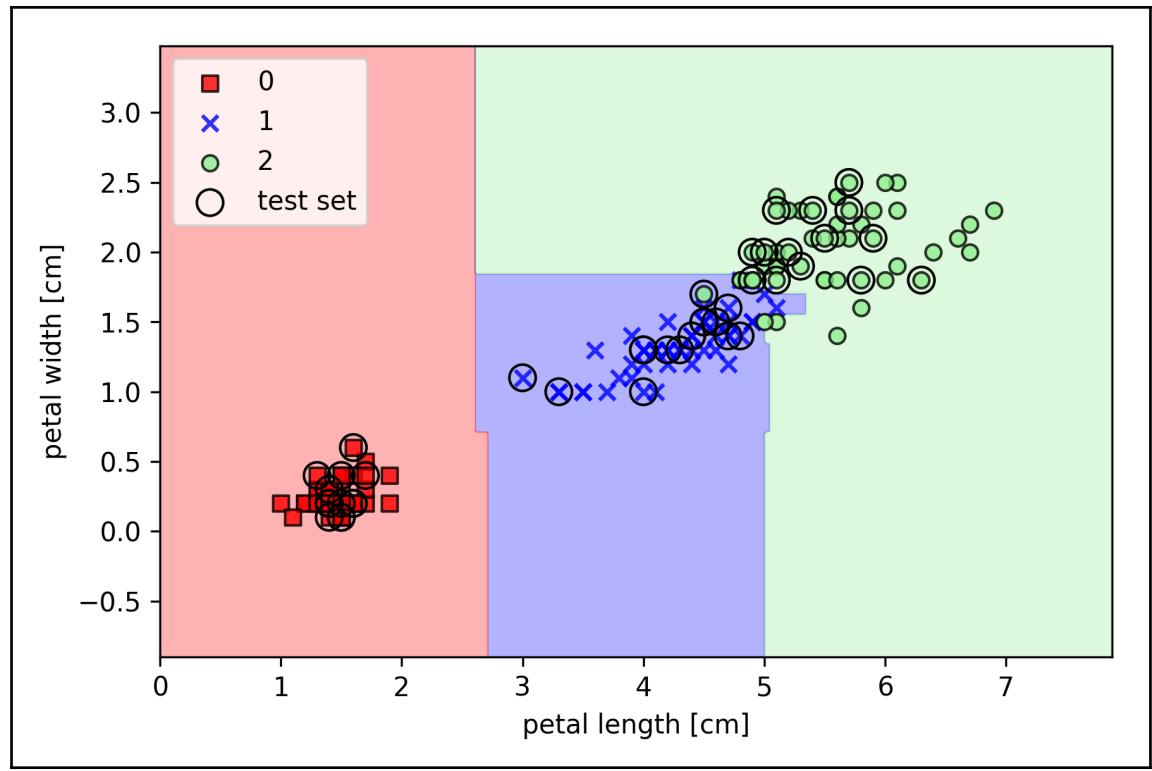


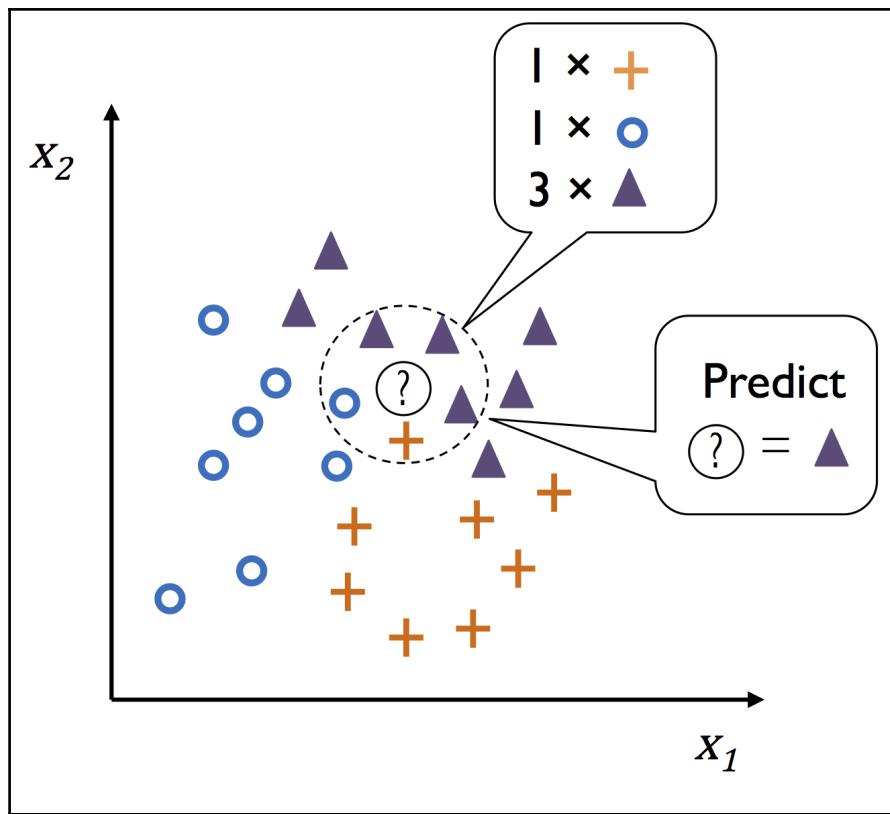


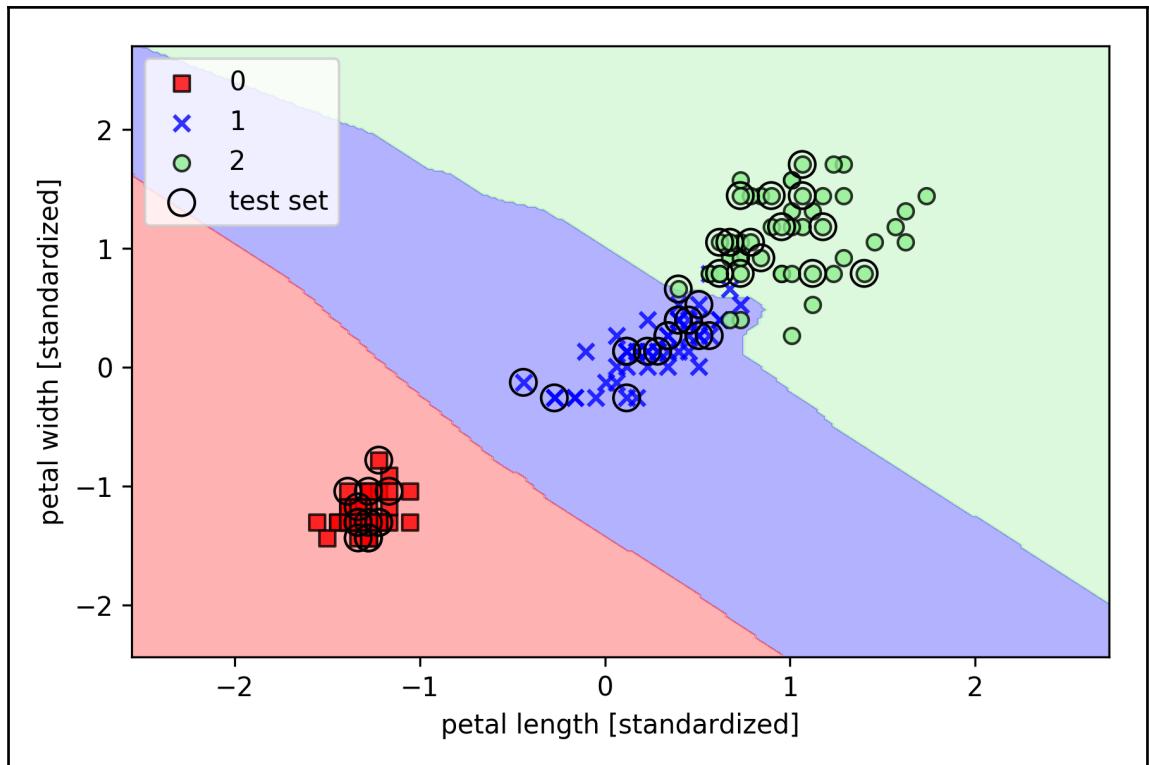




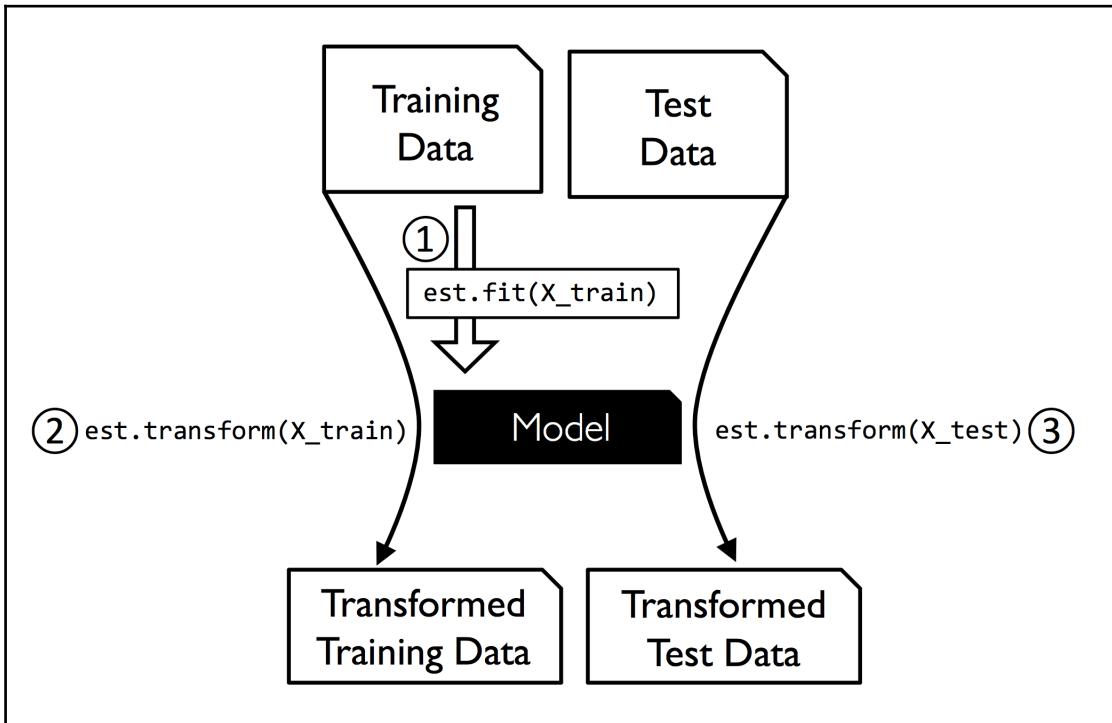


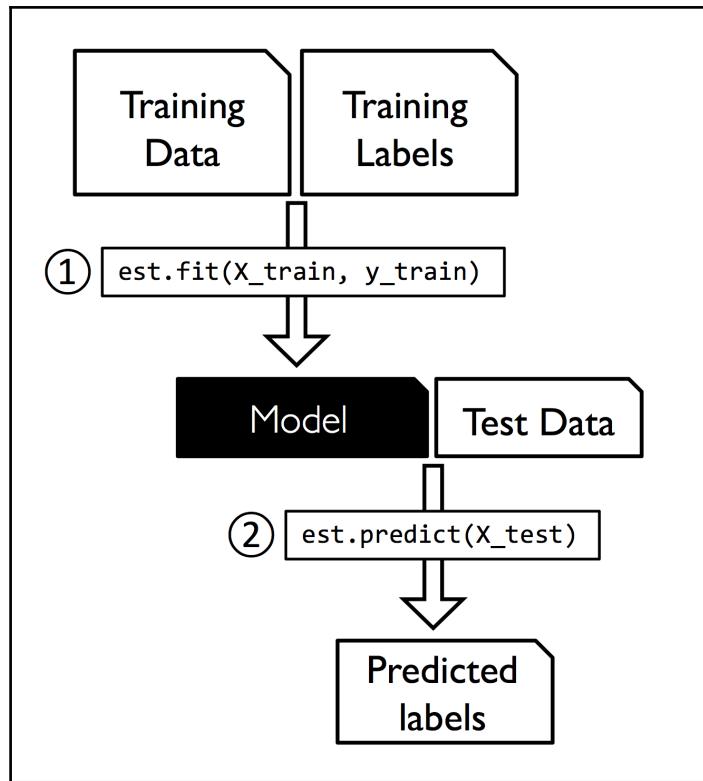




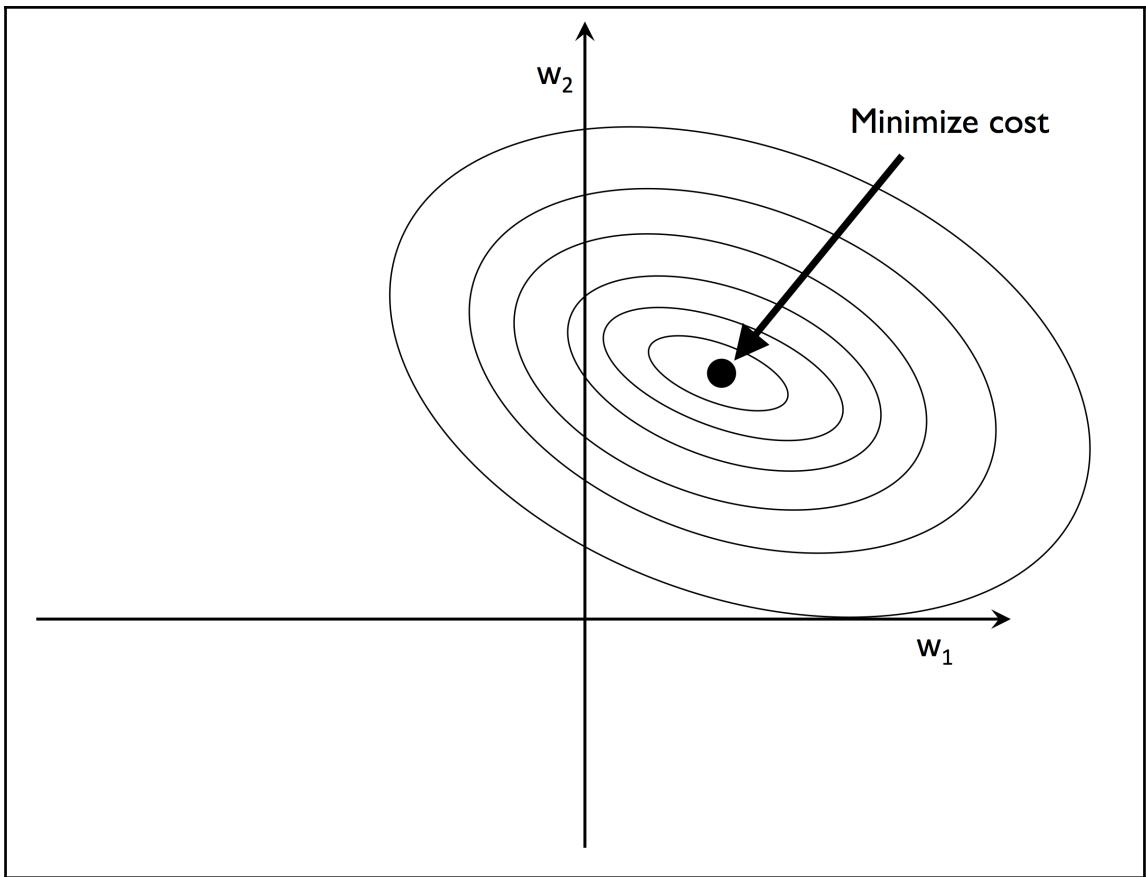


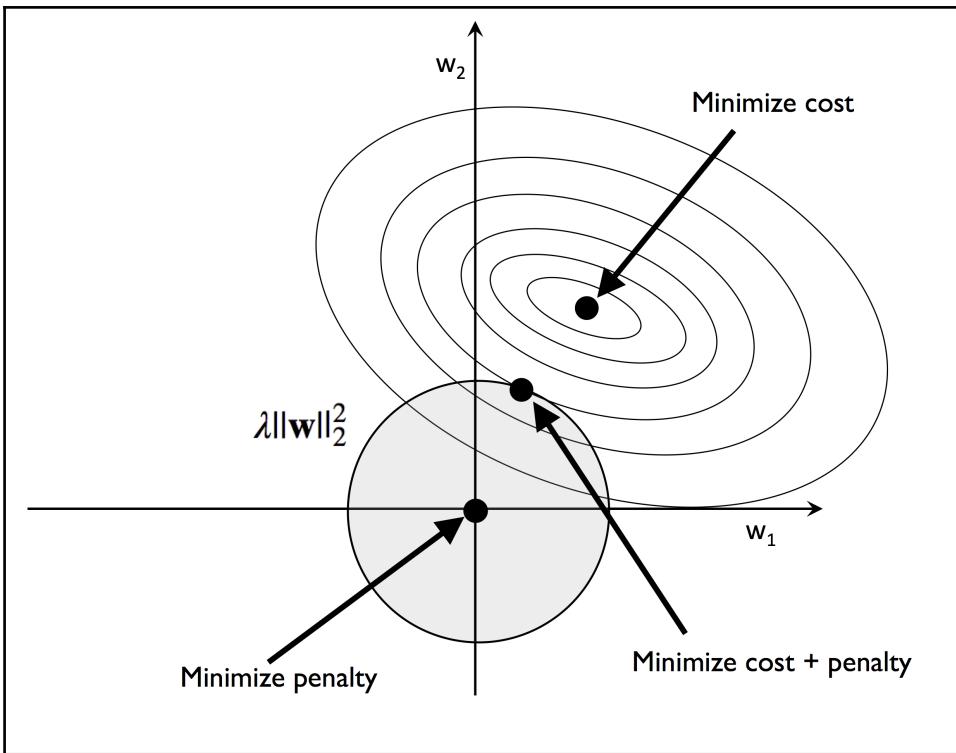
# Chapter 4: Building Good Training Sets – Data Preprocessing

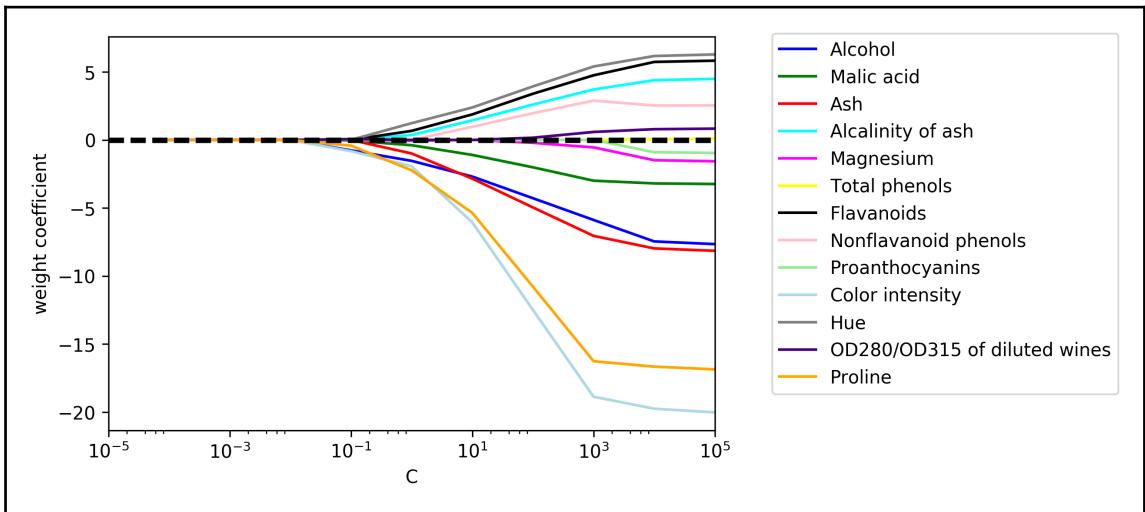
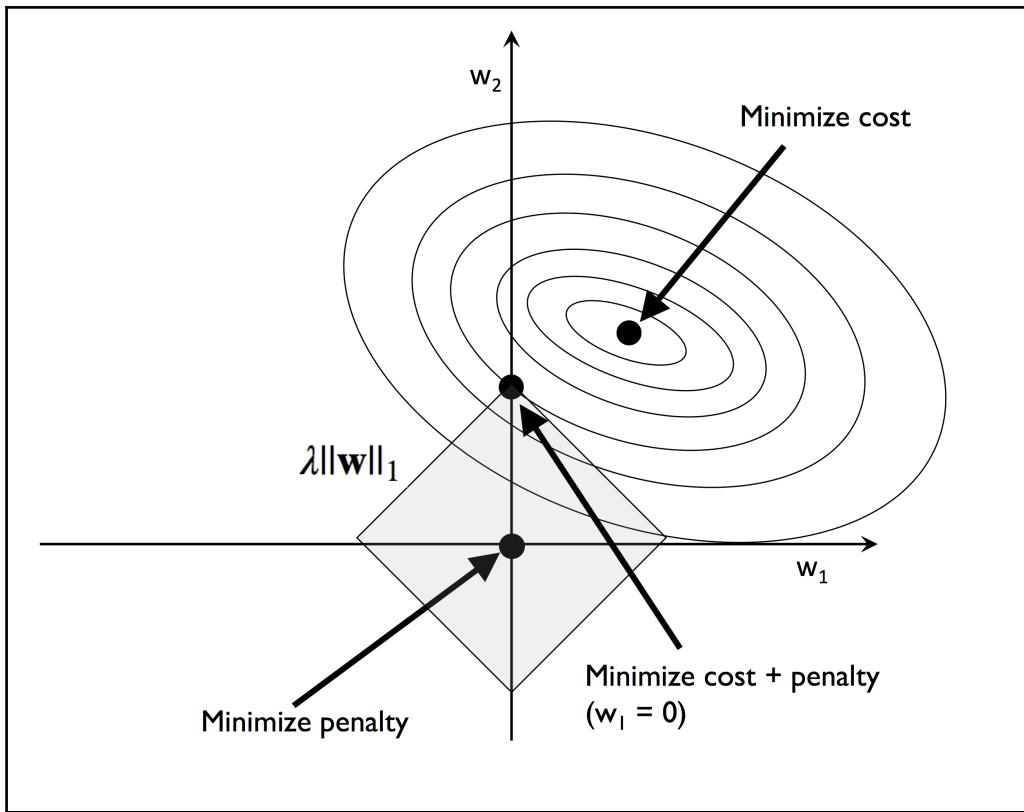


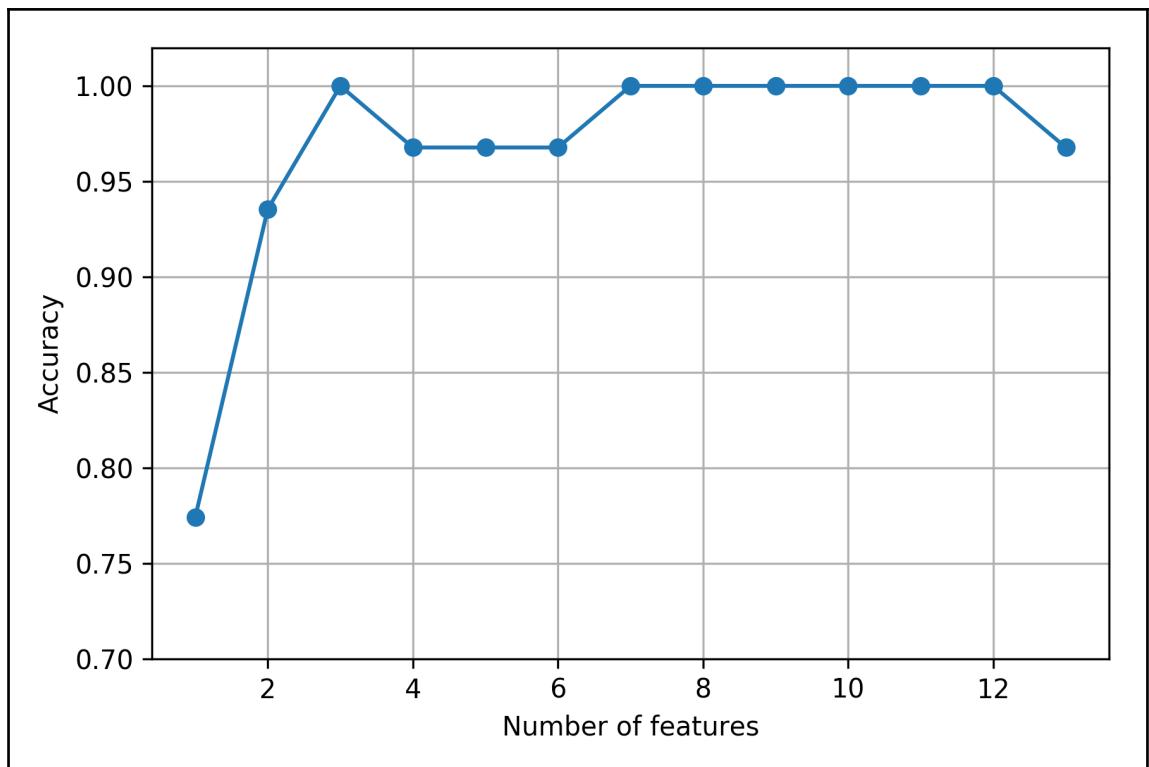


	Class label	Alcohol	Malic acid	Ash	Alcalinity of ash	Magnesium	Total phenols	Flavanoids	Nonflavanoid phenols	Proanthocyanins	Color intensity	Hue	OD280/OD315 of diluted wines	Proline
0	1	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	2.29	5.64	1.04	3.92	1065
1	1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.38	1.05	3.40	1050
2	1	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.81	5.68	1.03	3.17	1185
3	1	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.18	7.80	0.86	3.45	1480
4	1	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.82	4.32	1.04	2.93	735

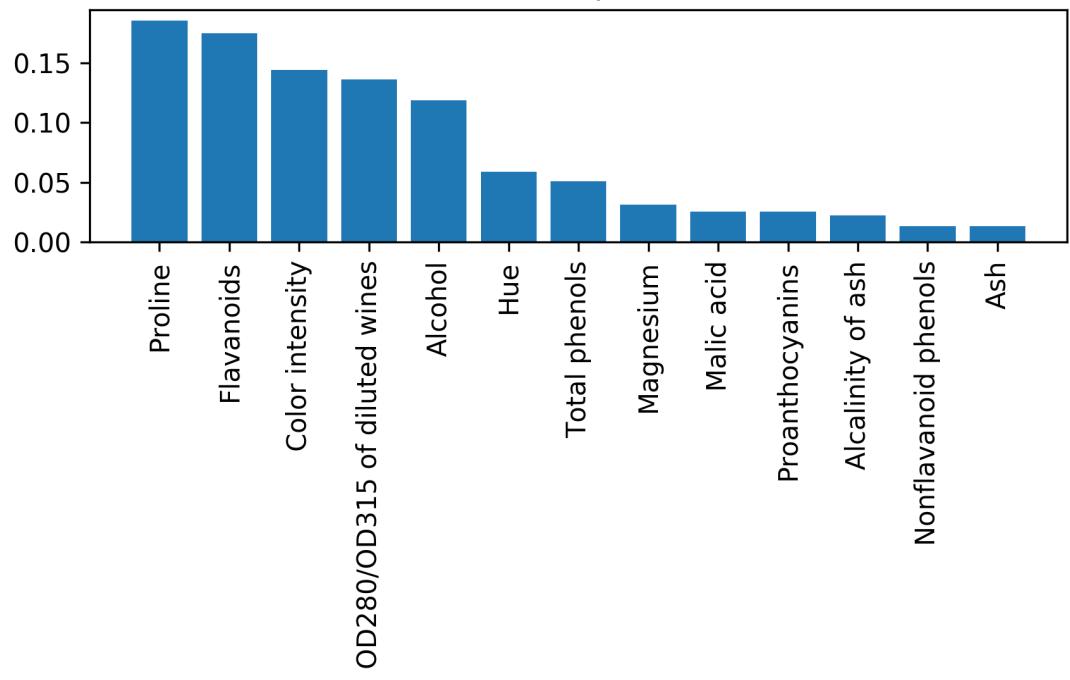




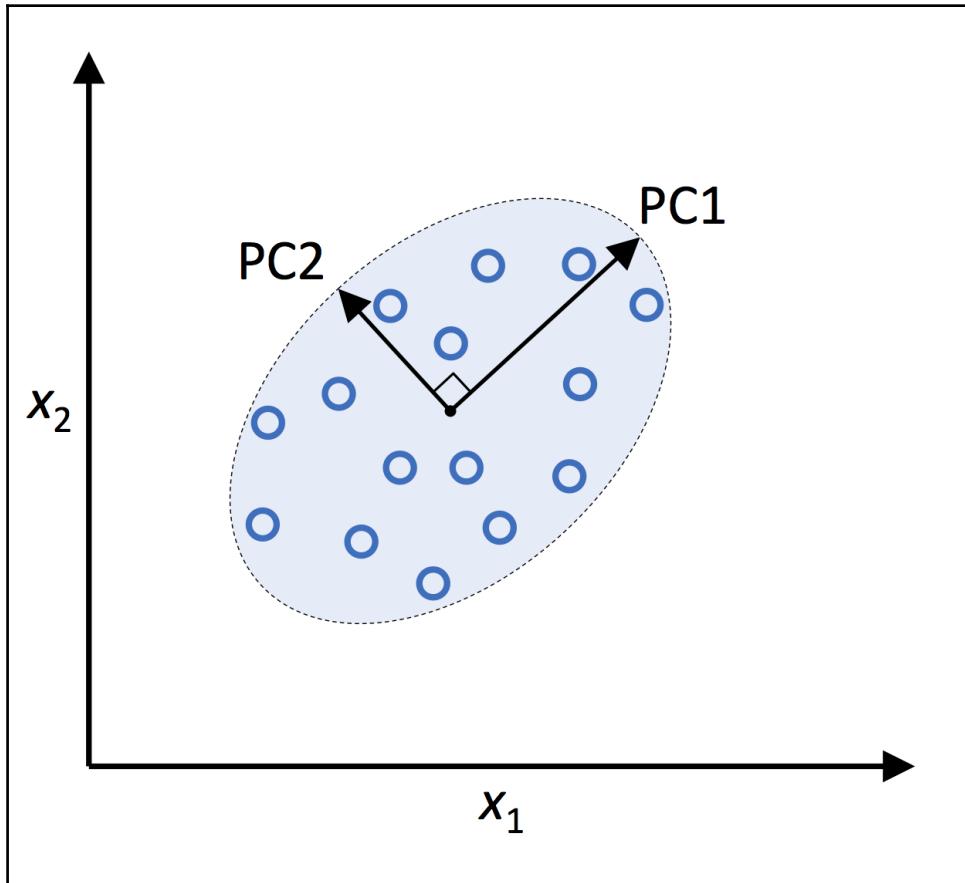


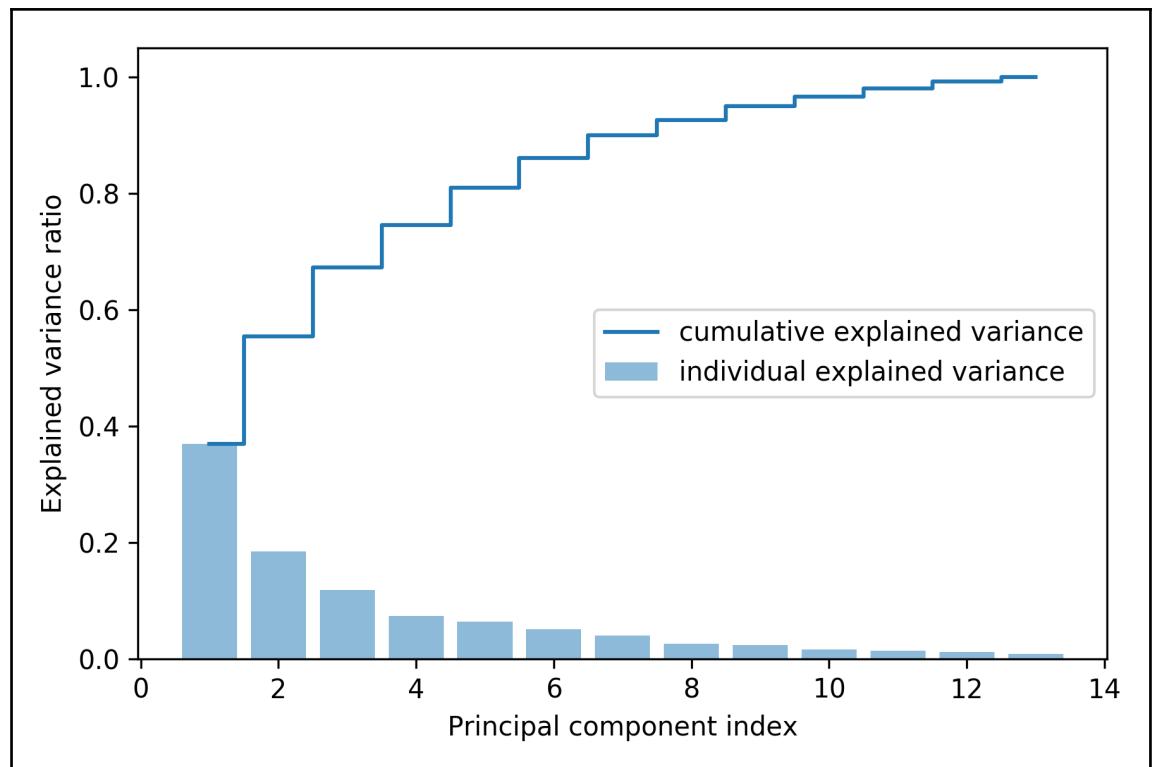


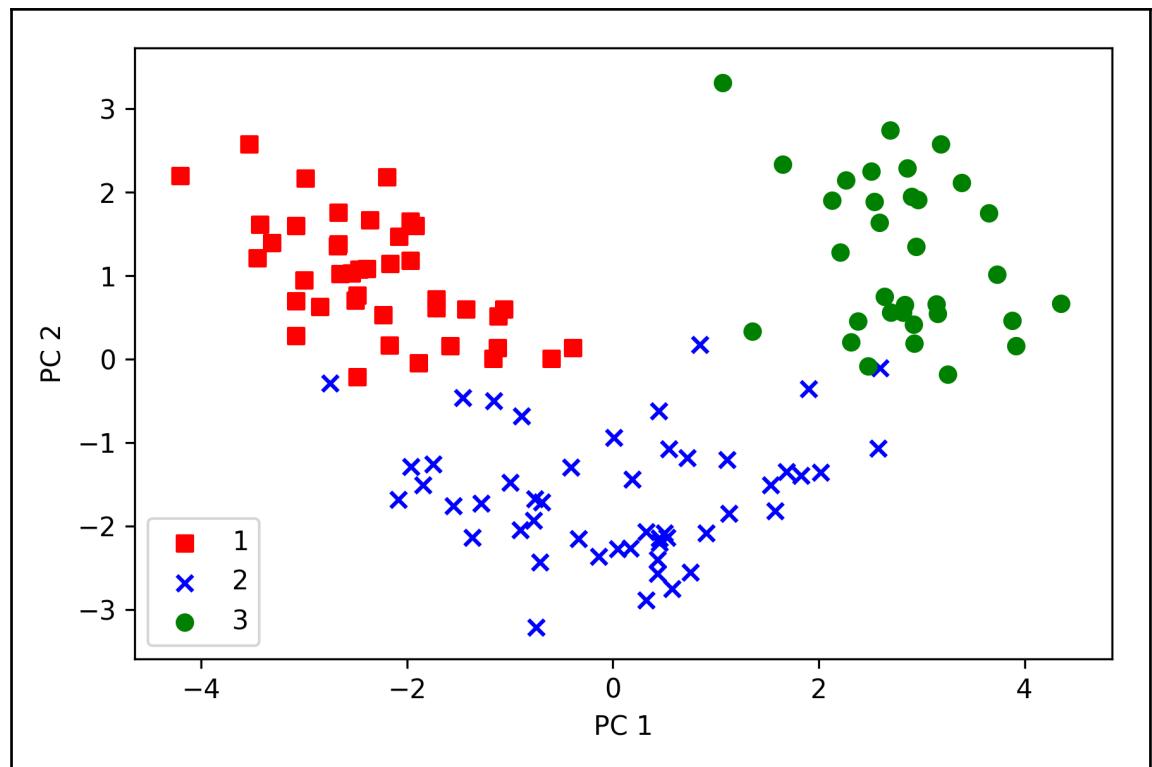
Feature Importance

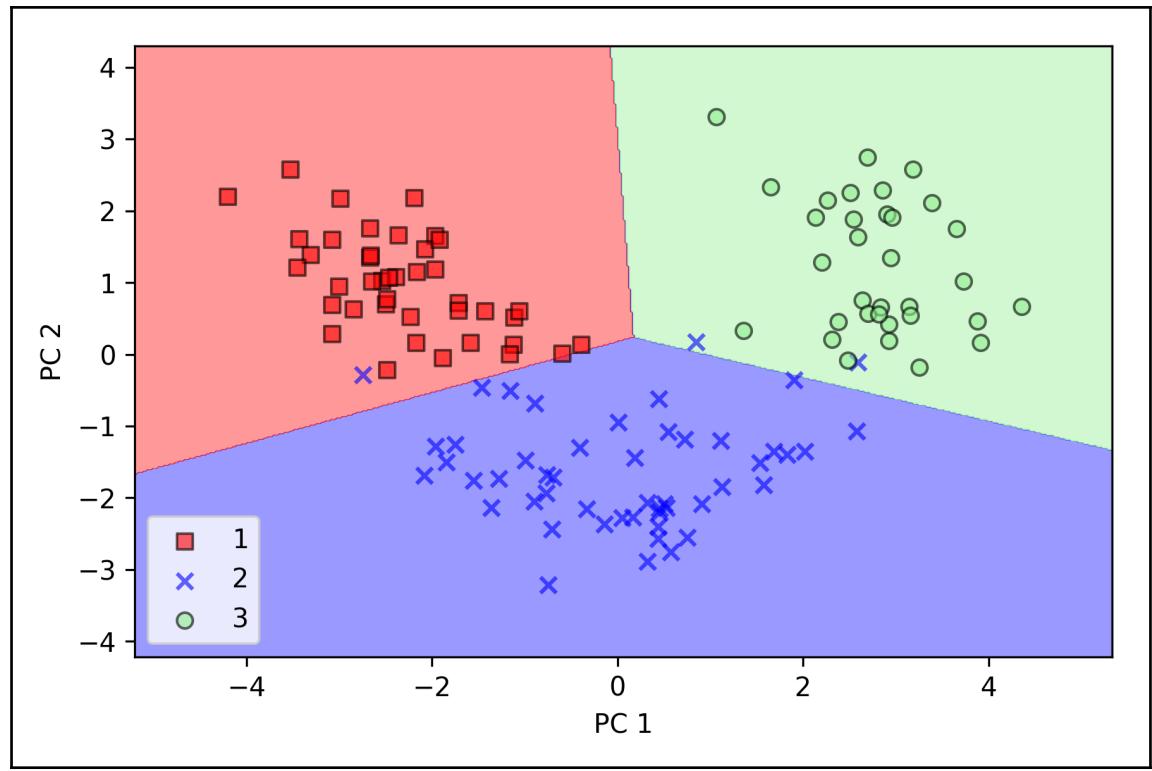


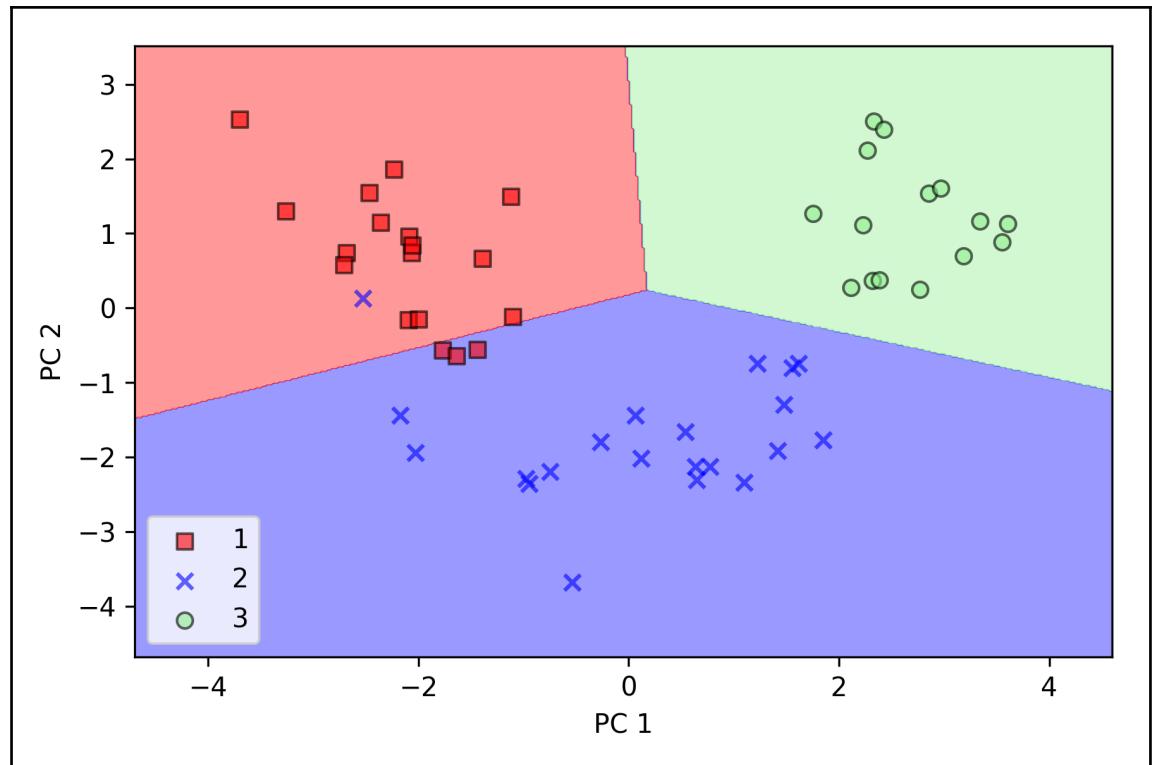
# Chapter 5: Compressing Data via Dimensionality Reduction

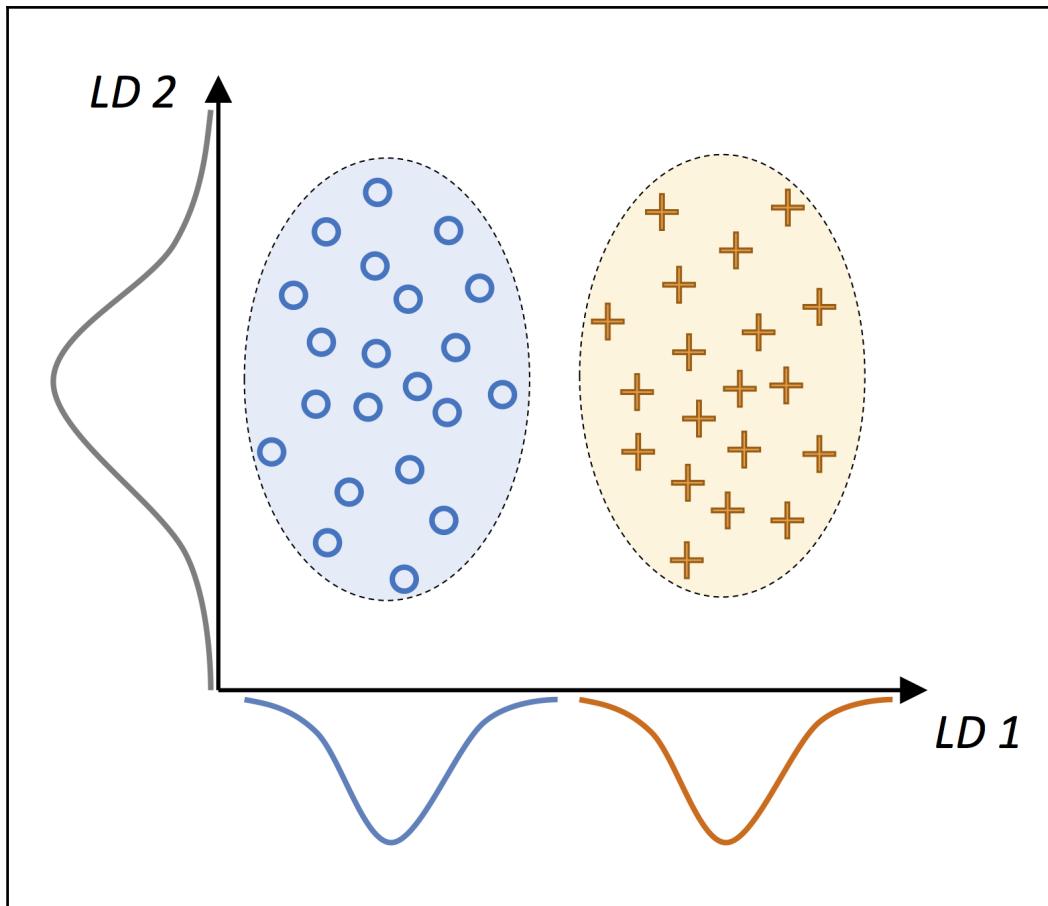


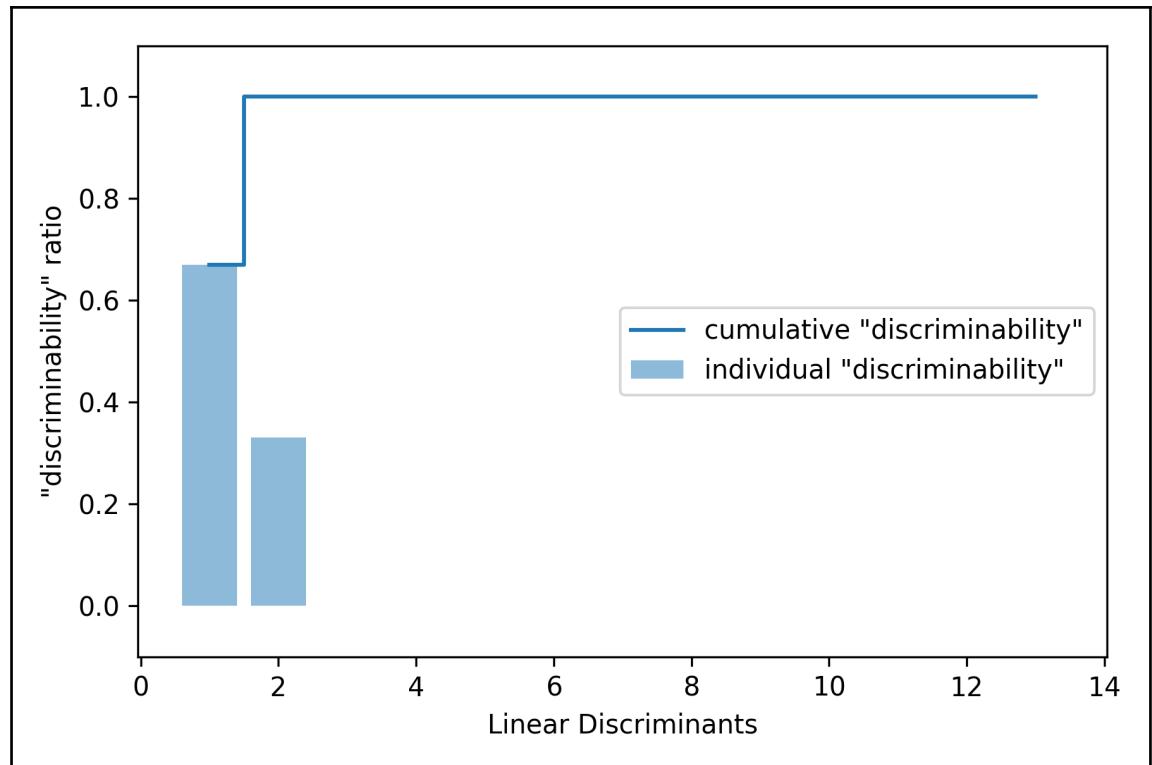


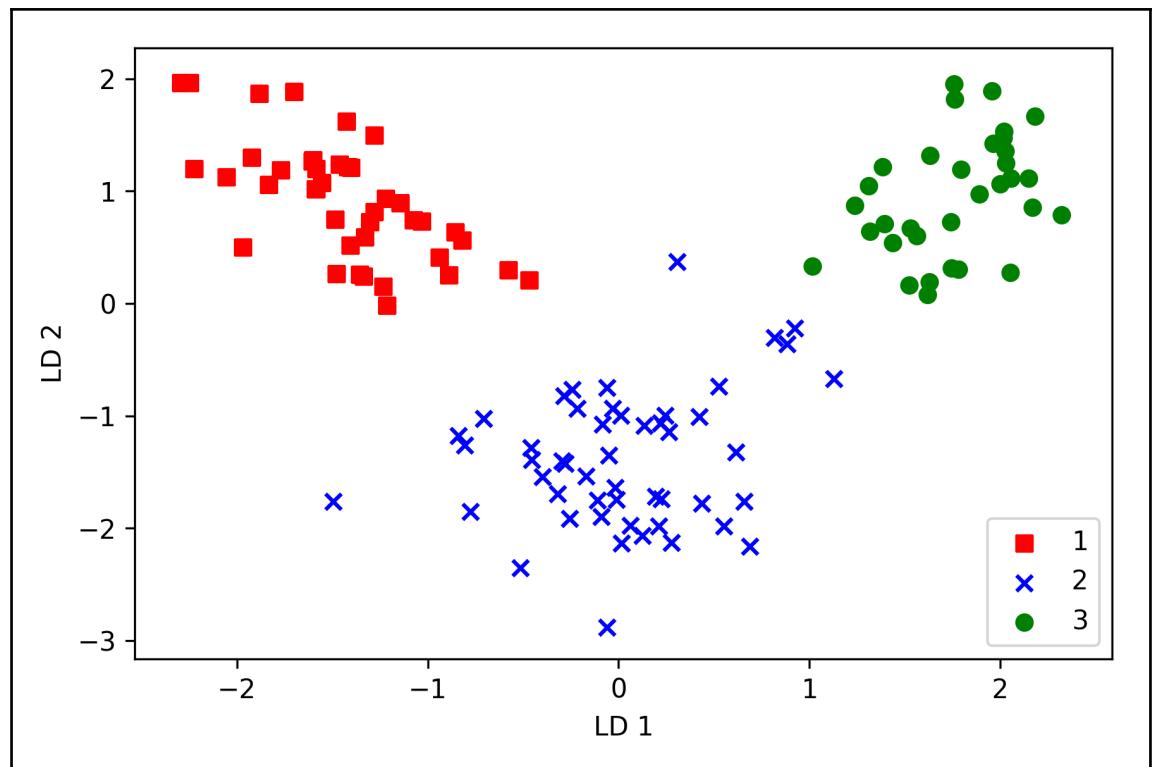


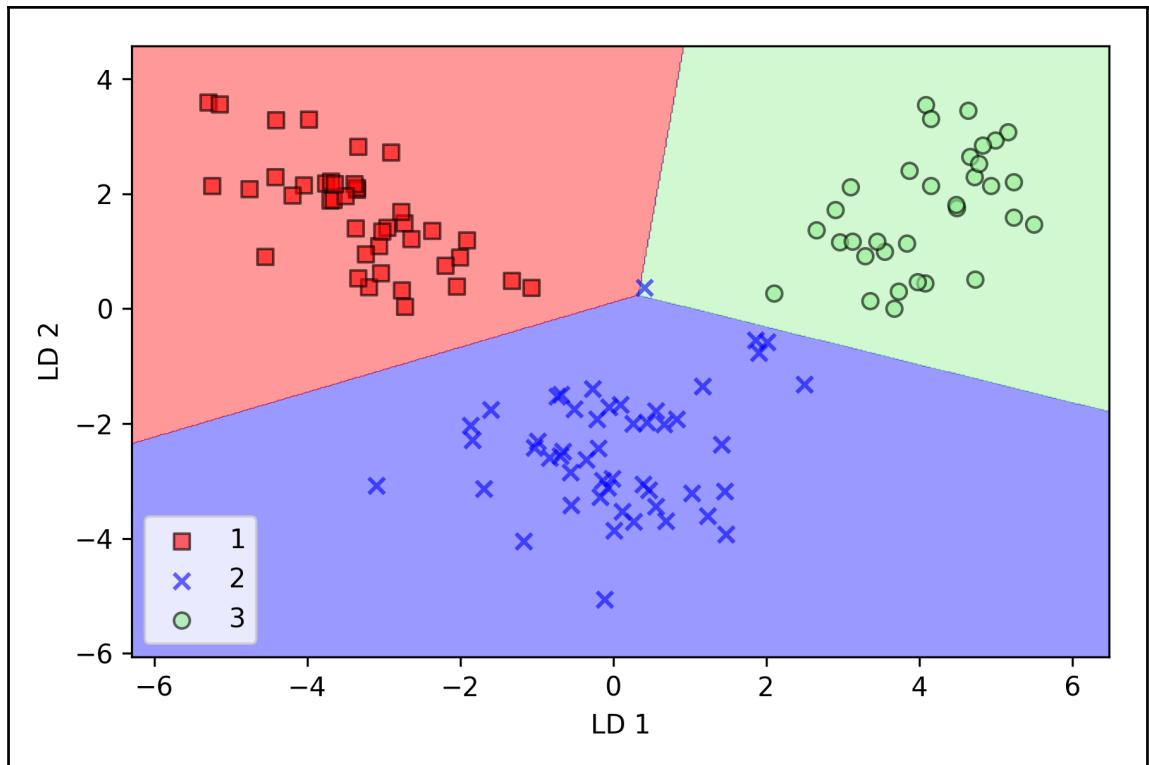


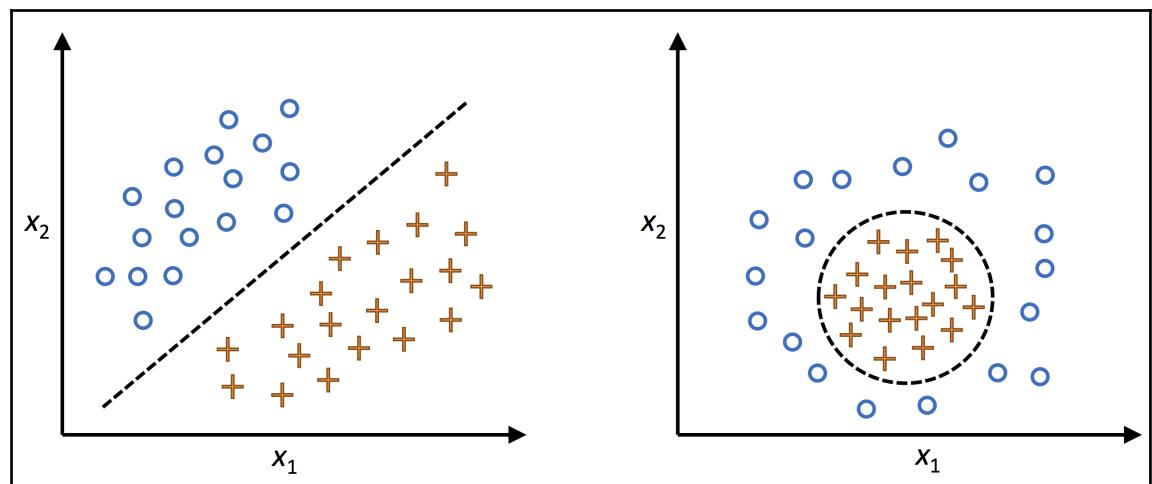
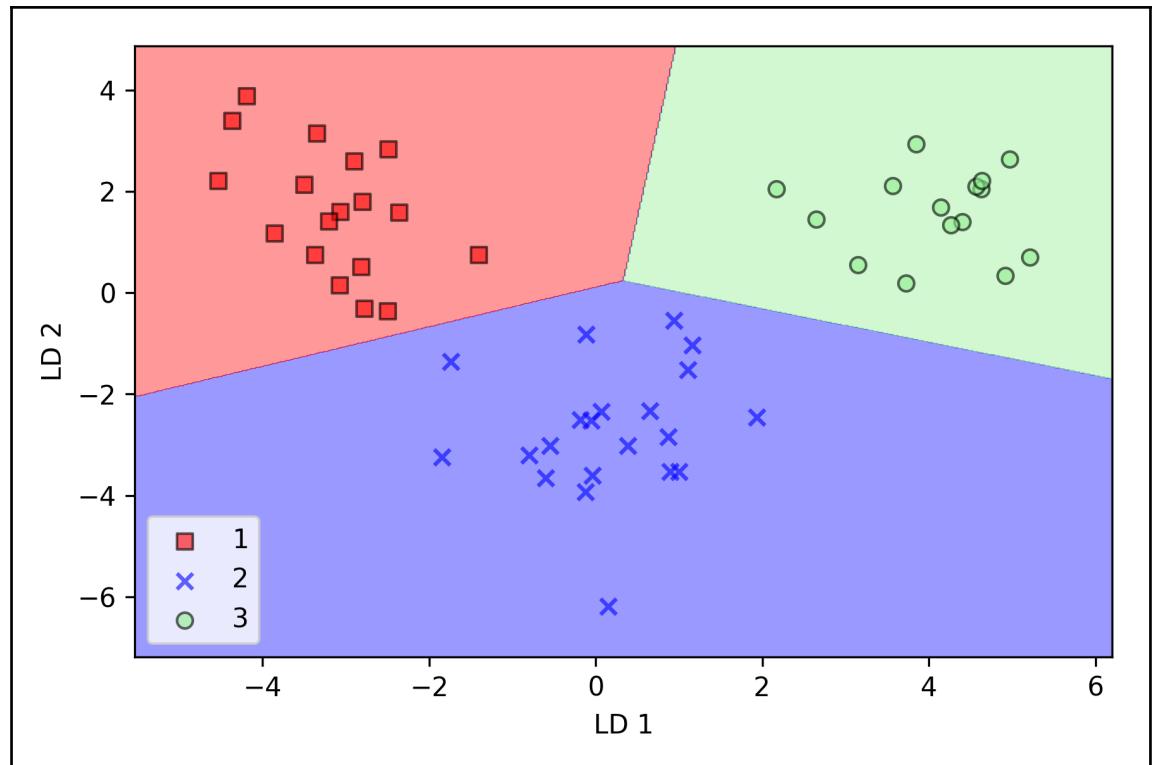


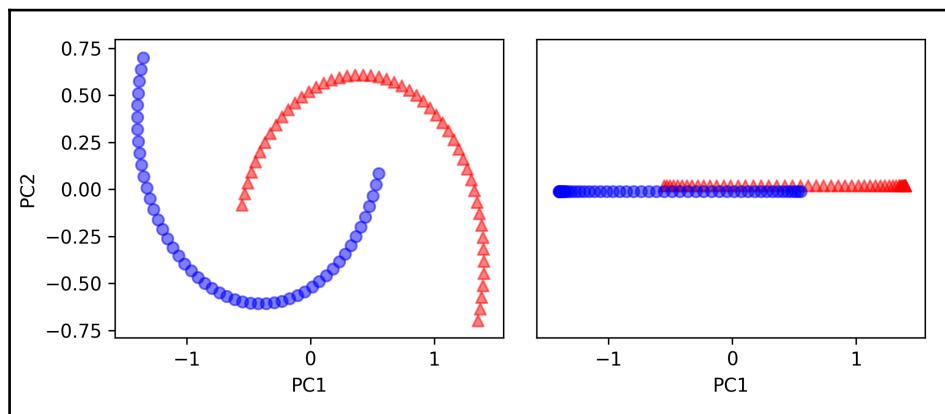
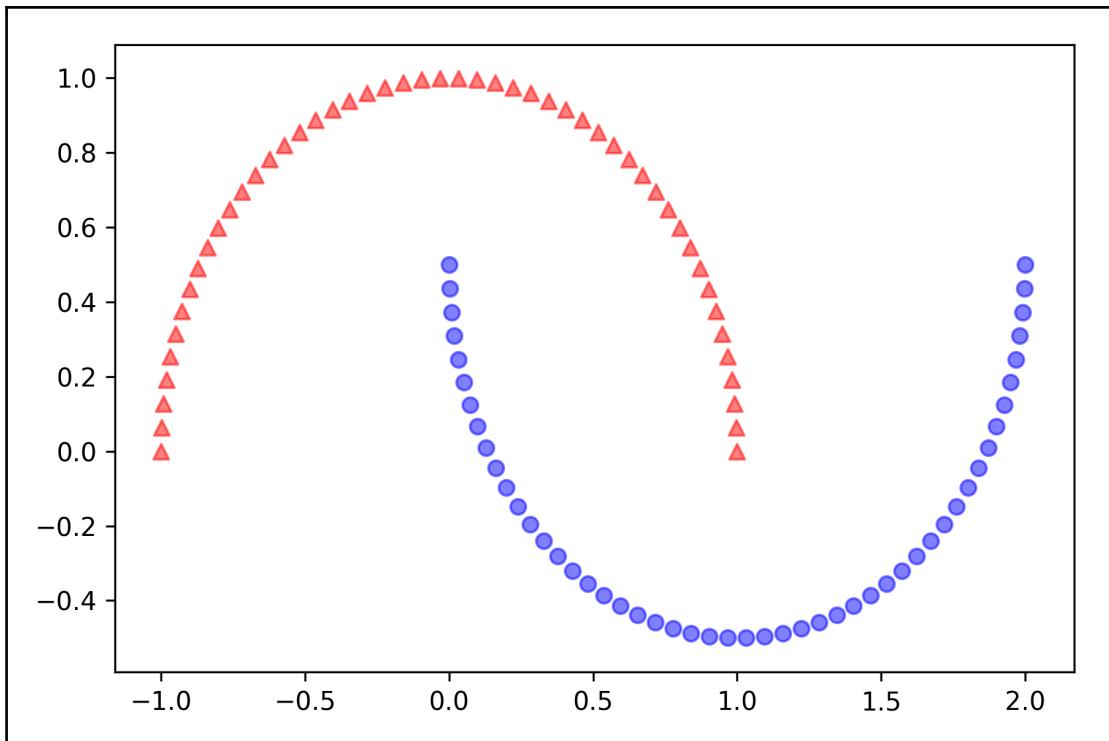


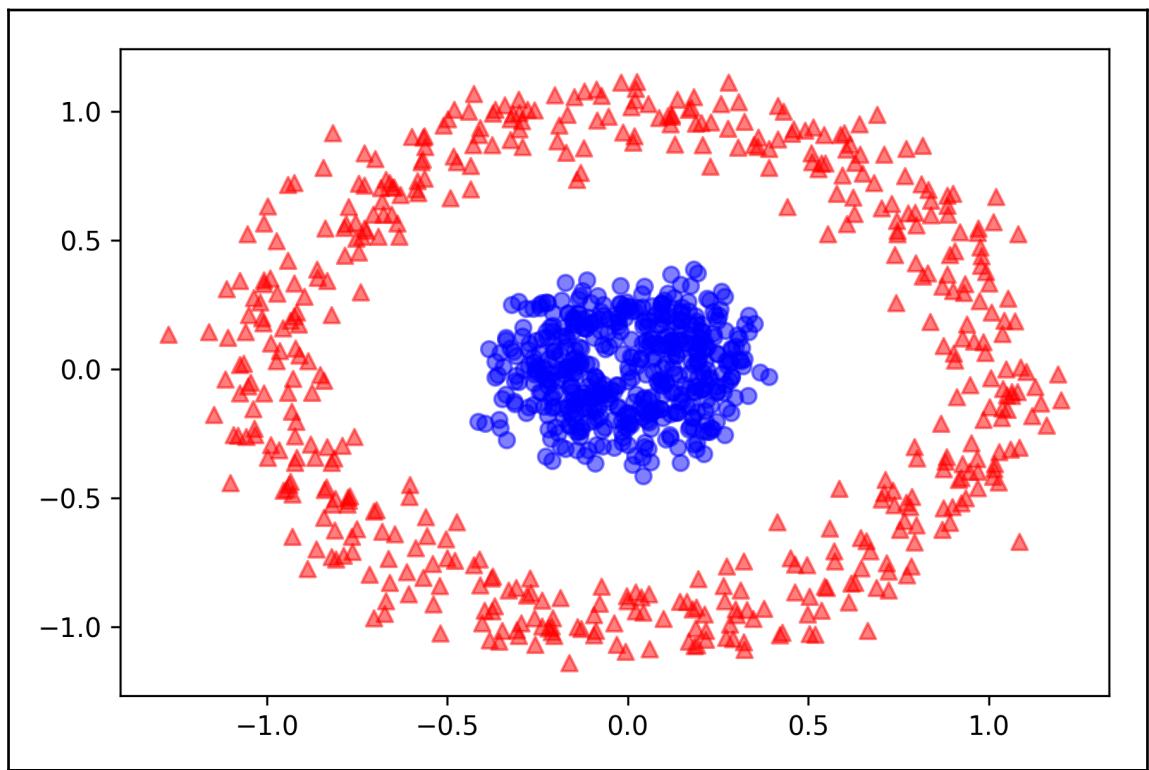
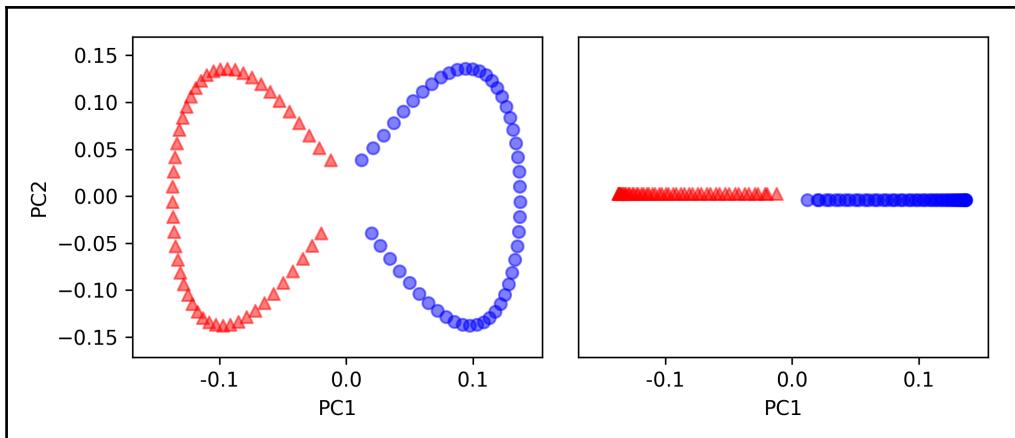


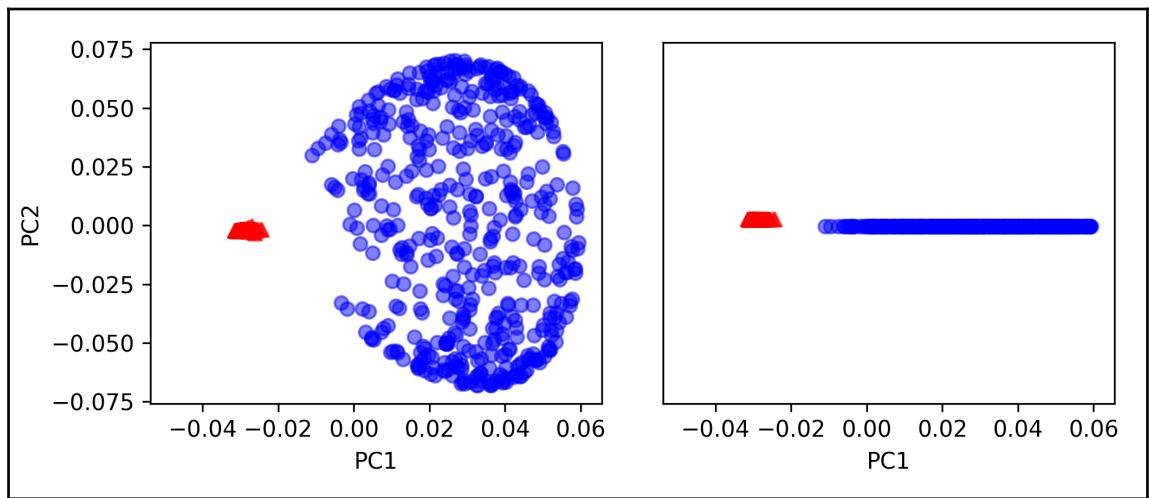
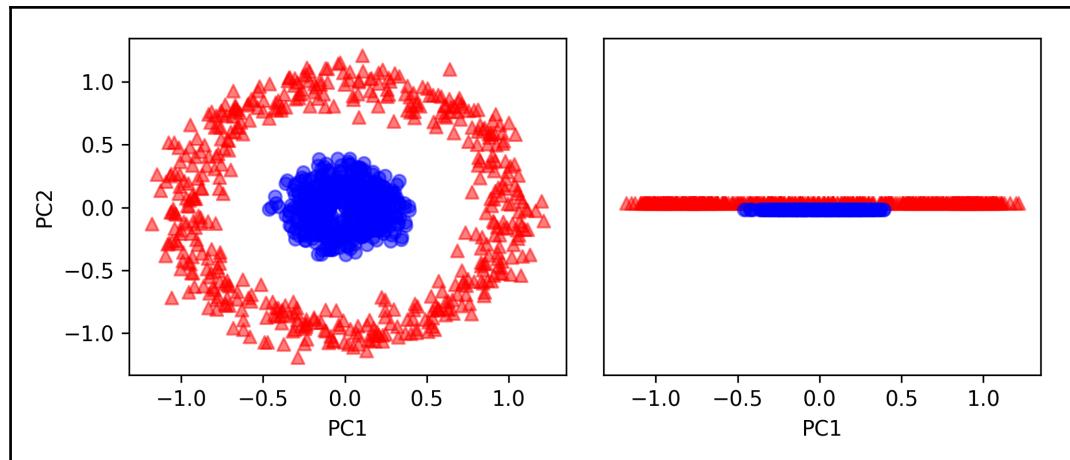


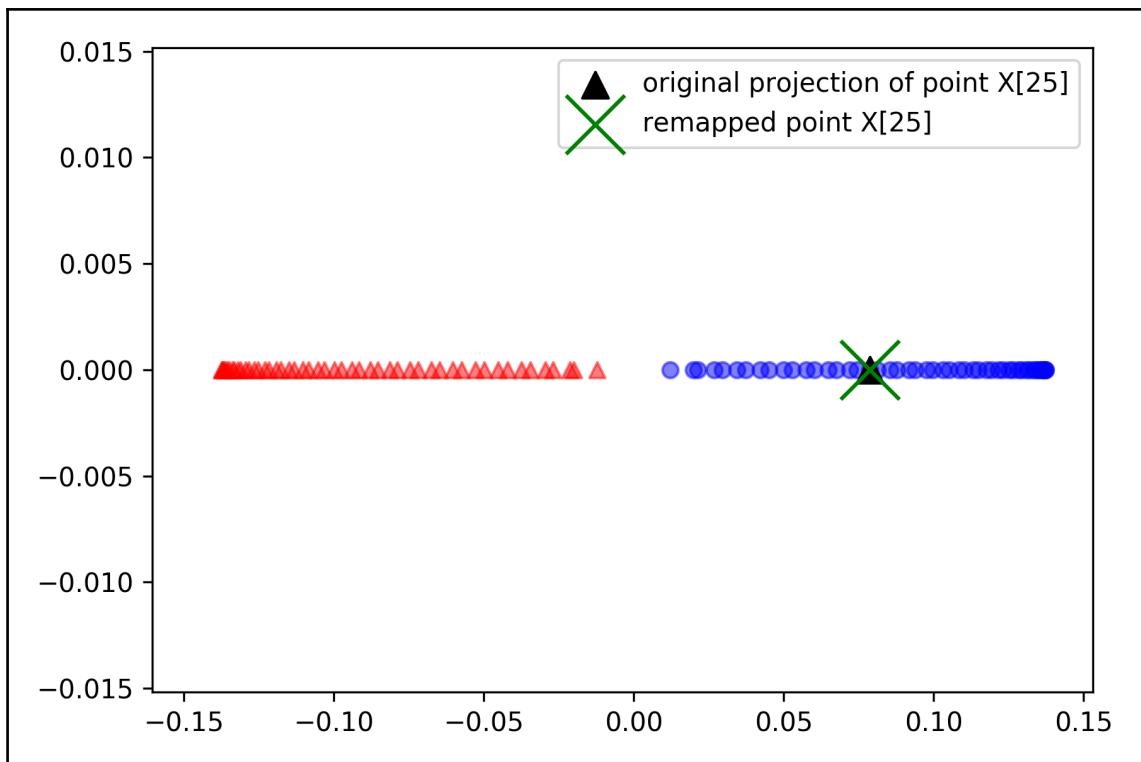


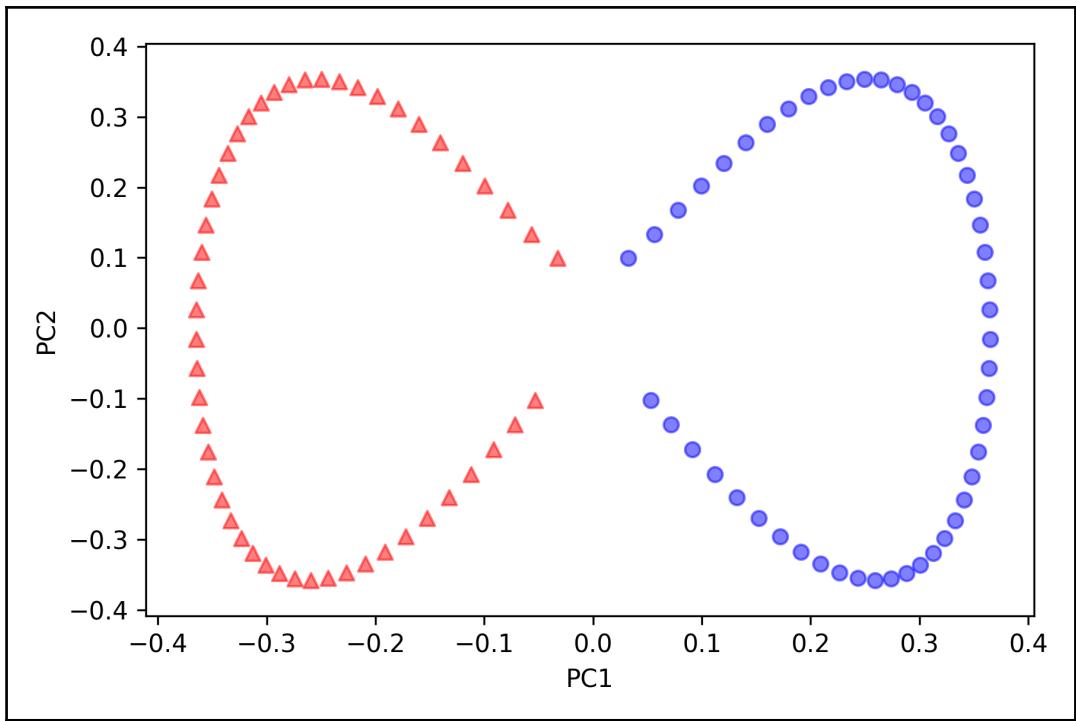




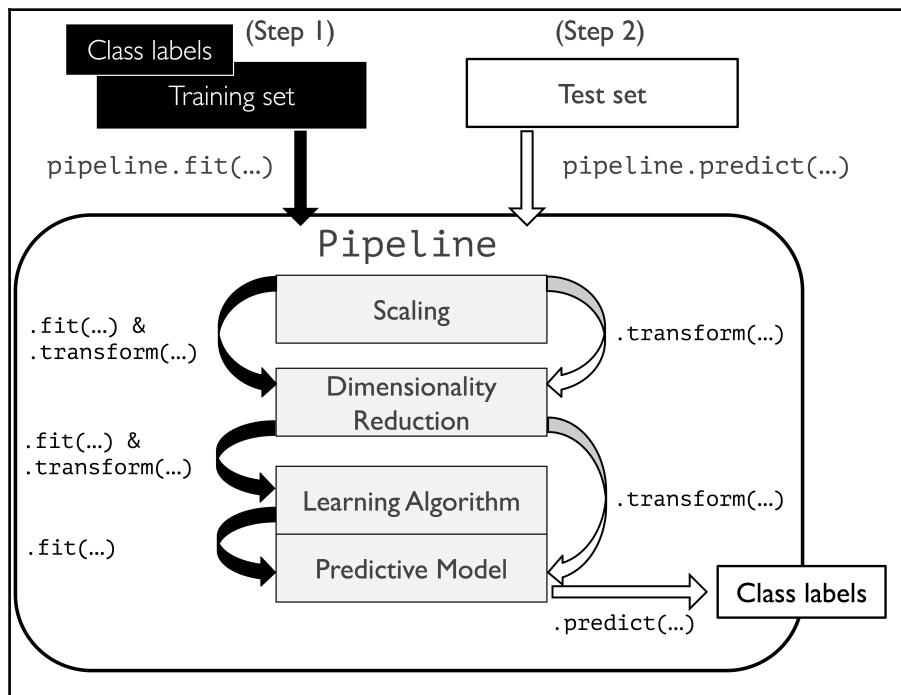


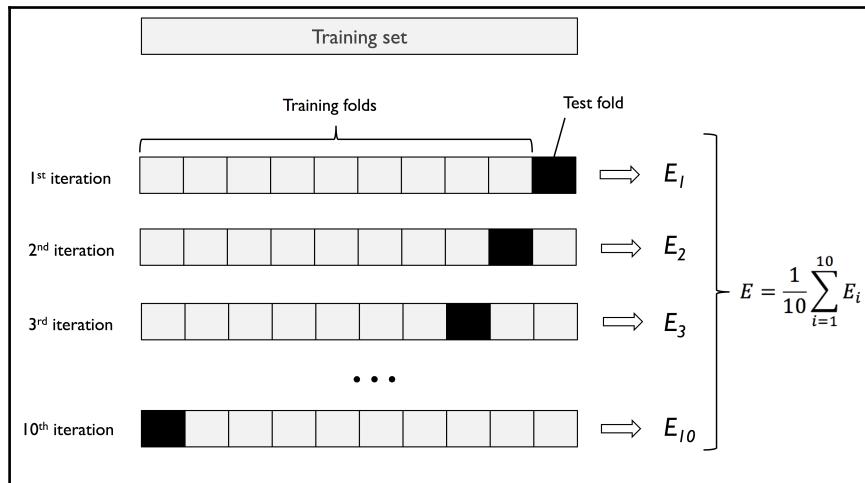
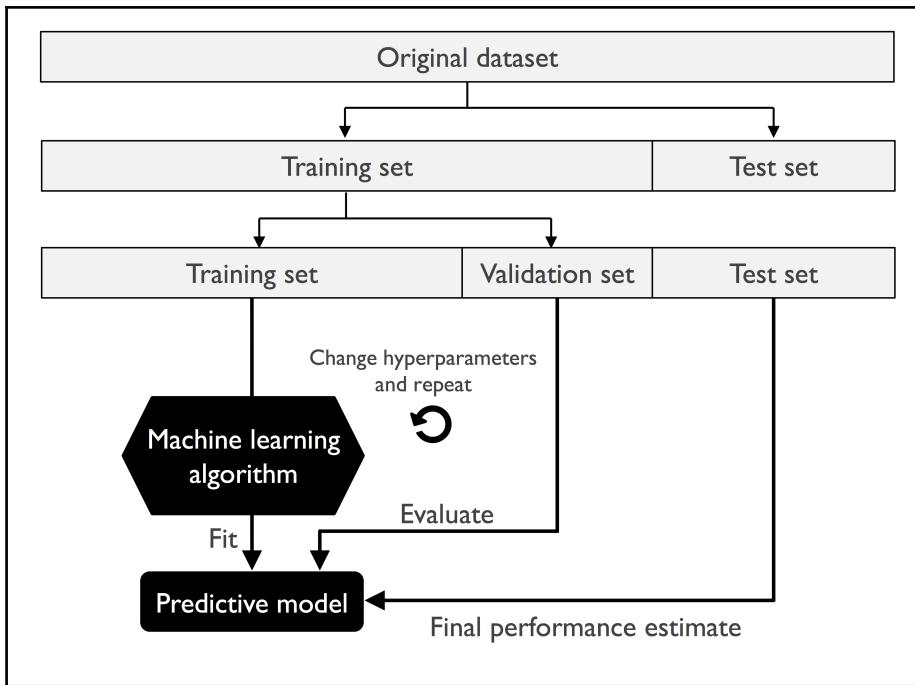


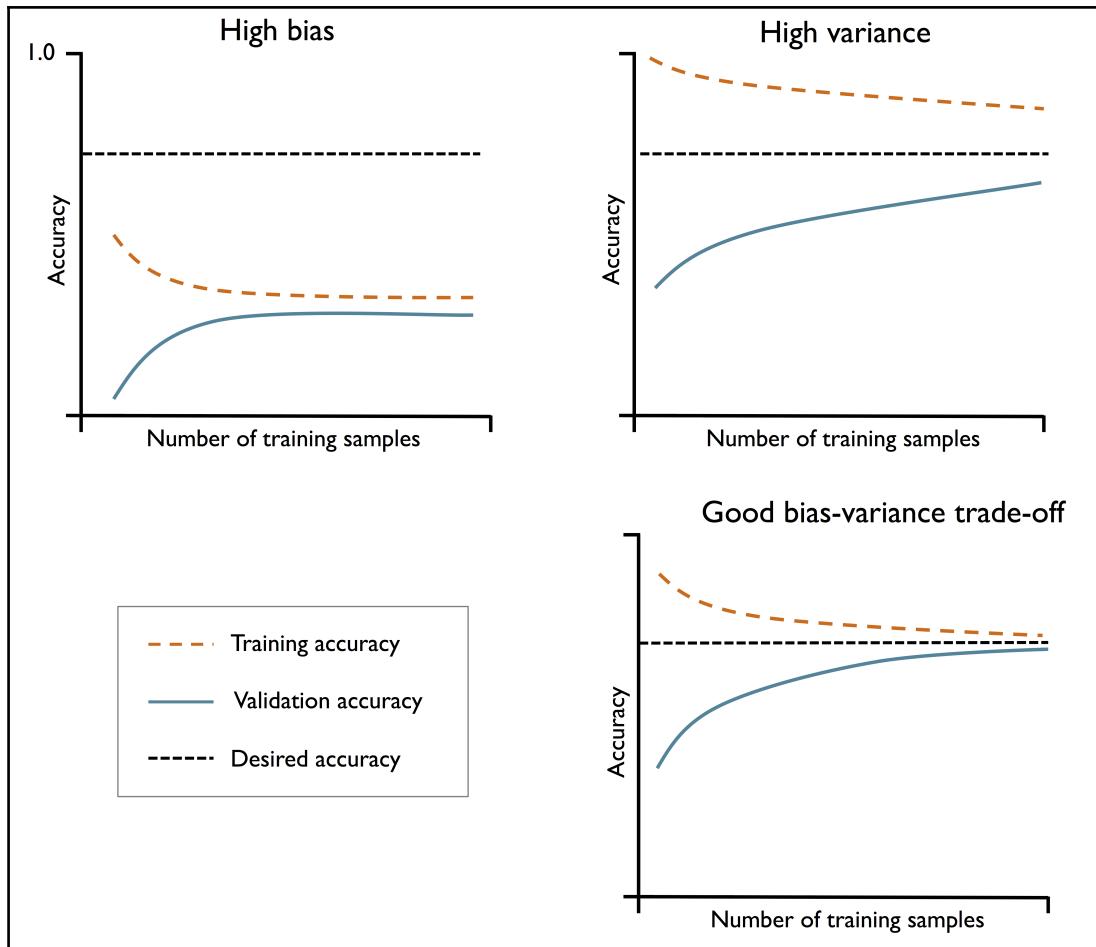


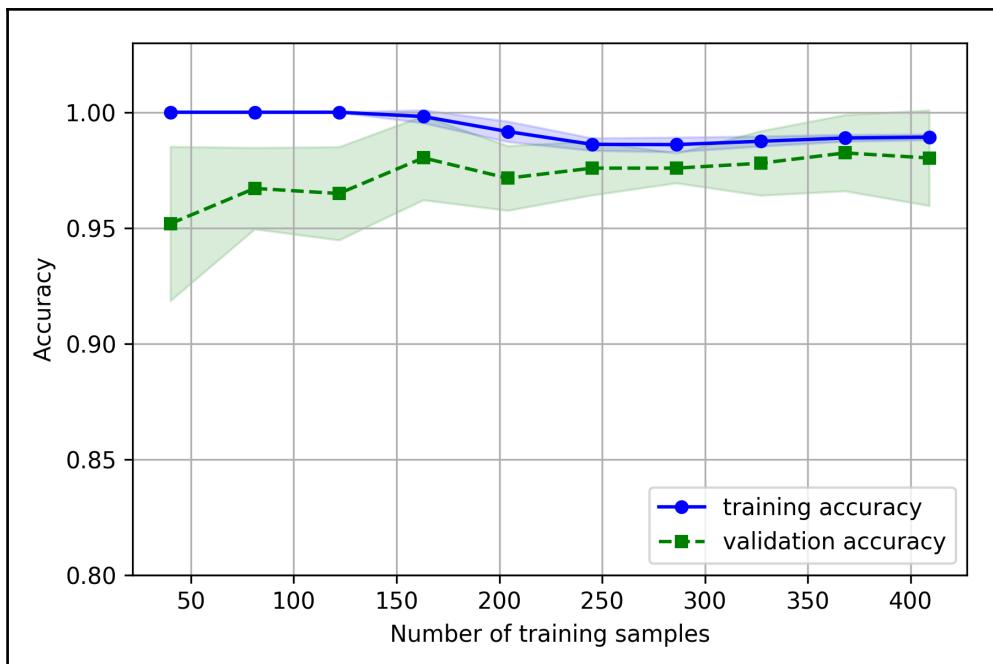


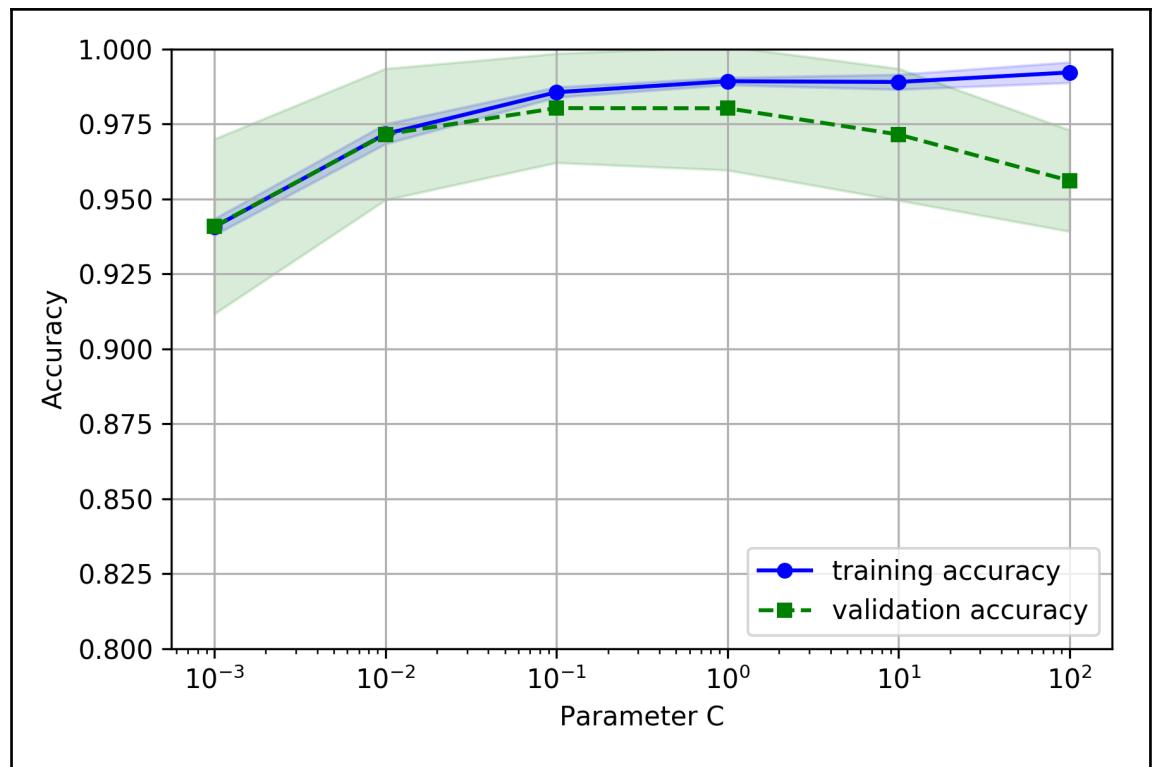
# Chapter 6: Learning Best Practices for Model Evaluation and Hyperparameter Tuning

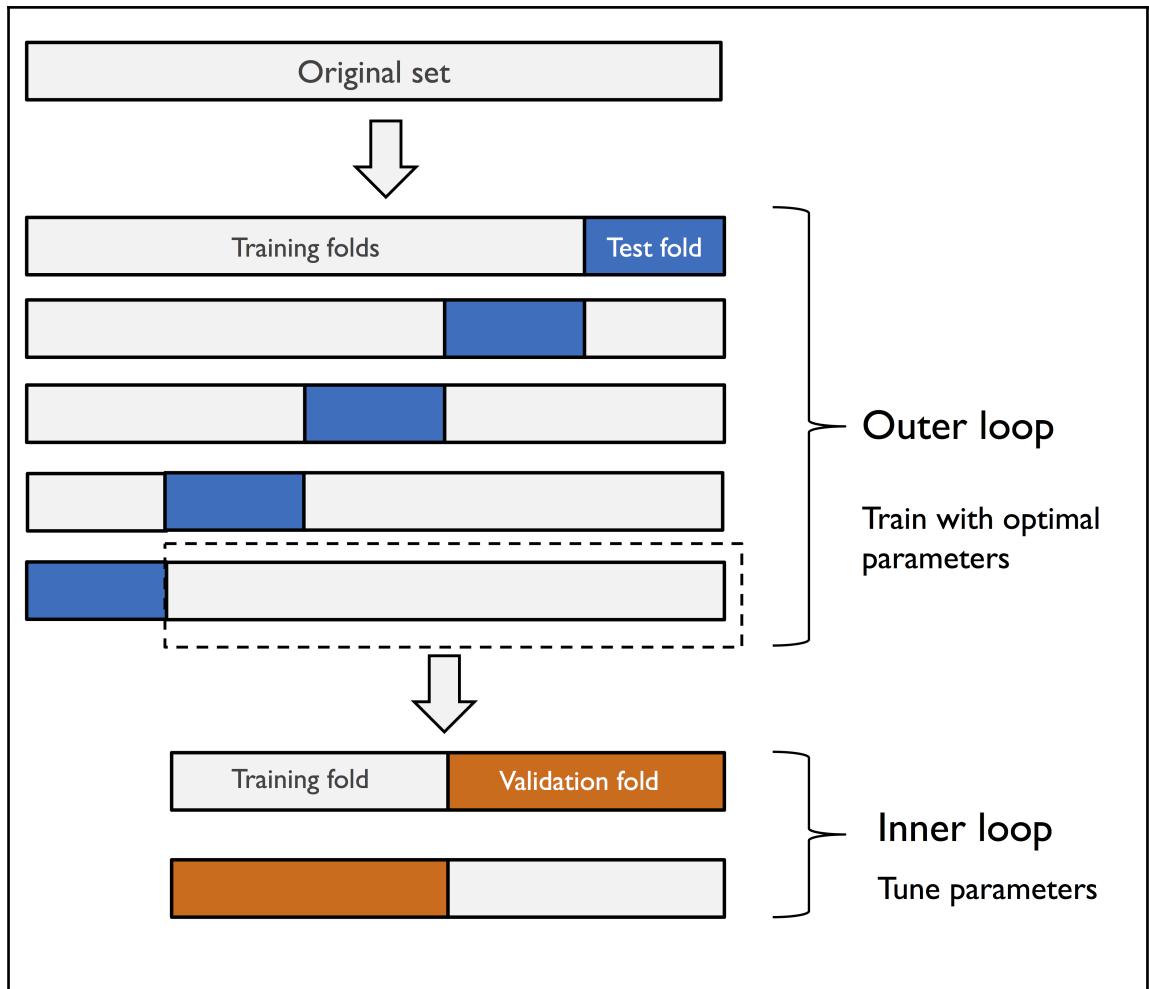




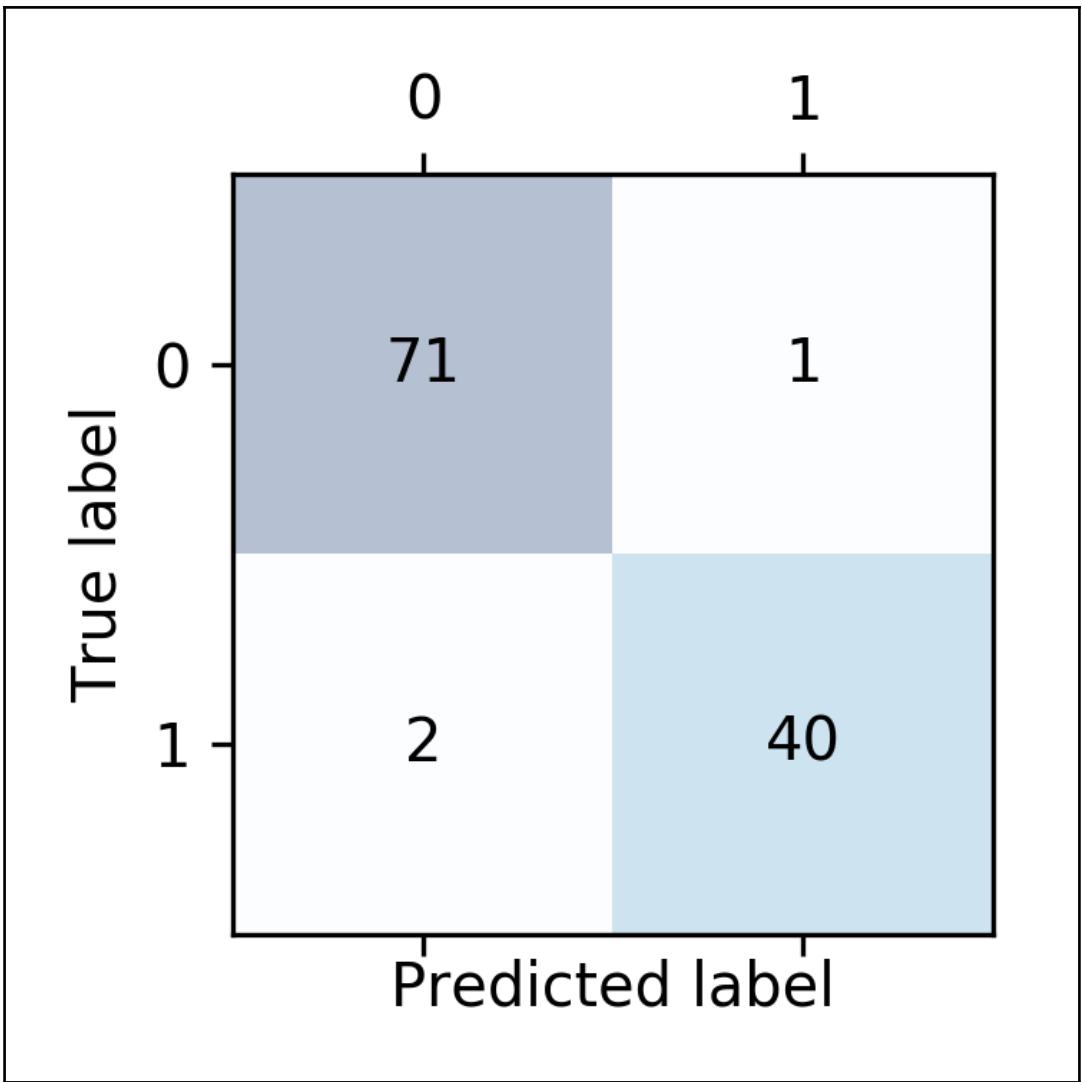


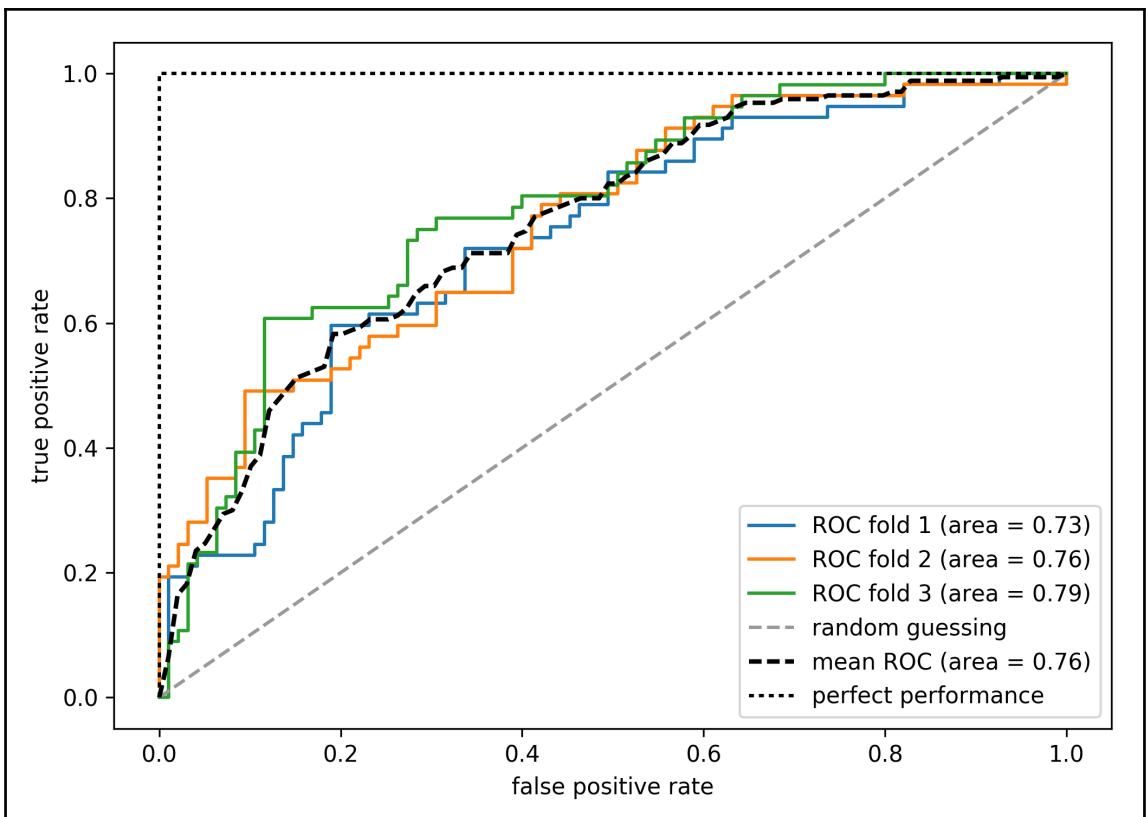




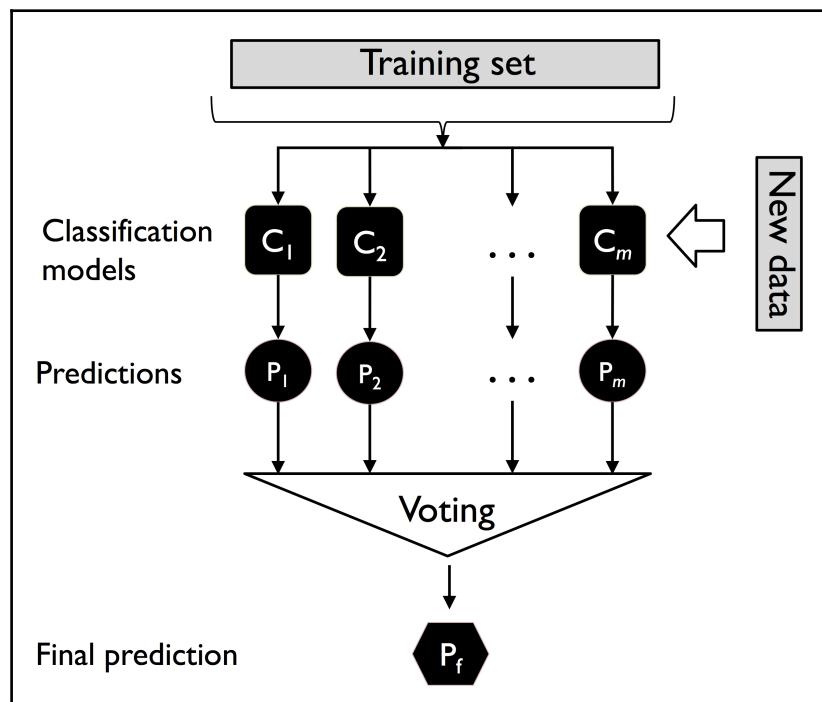
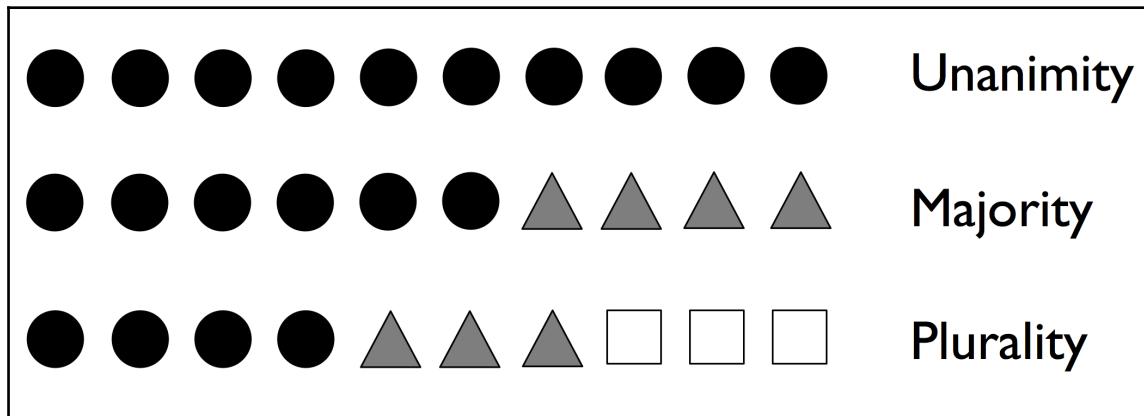


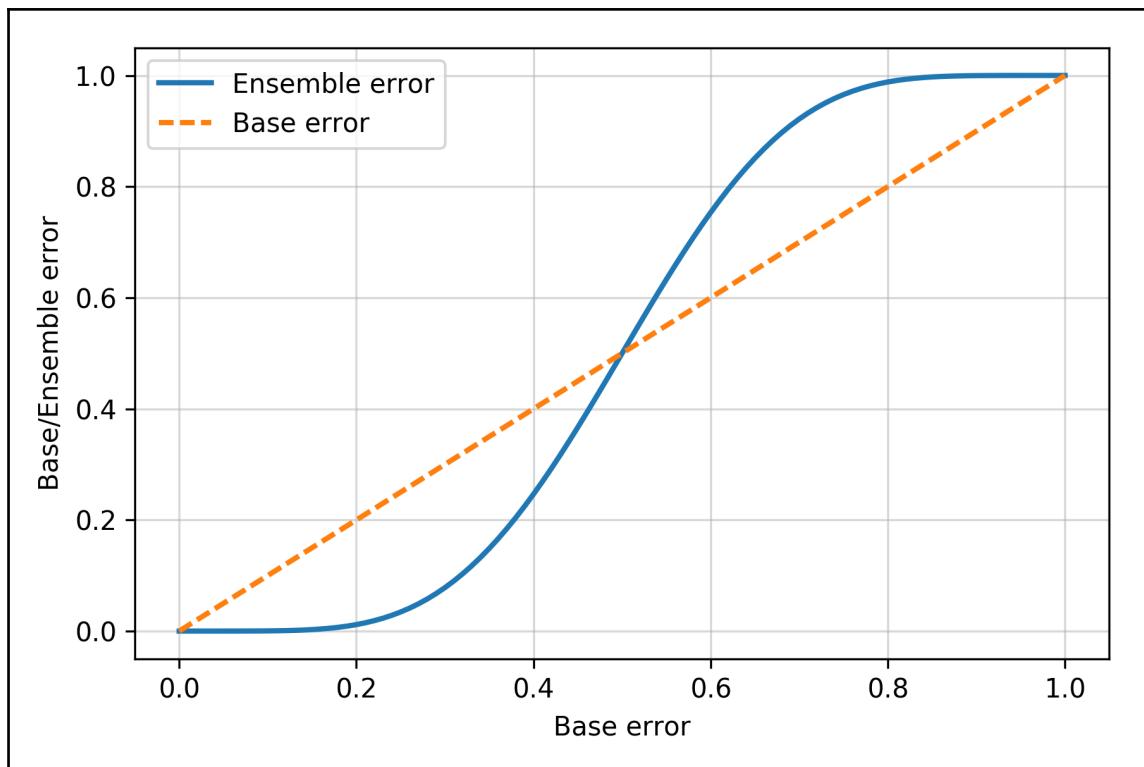
		Predicted class	
		$P$	$N$
$P$	$P$	True positives (TP)	False negatives (FN)
	$N$	False positives (FP)	True negatives (TN)

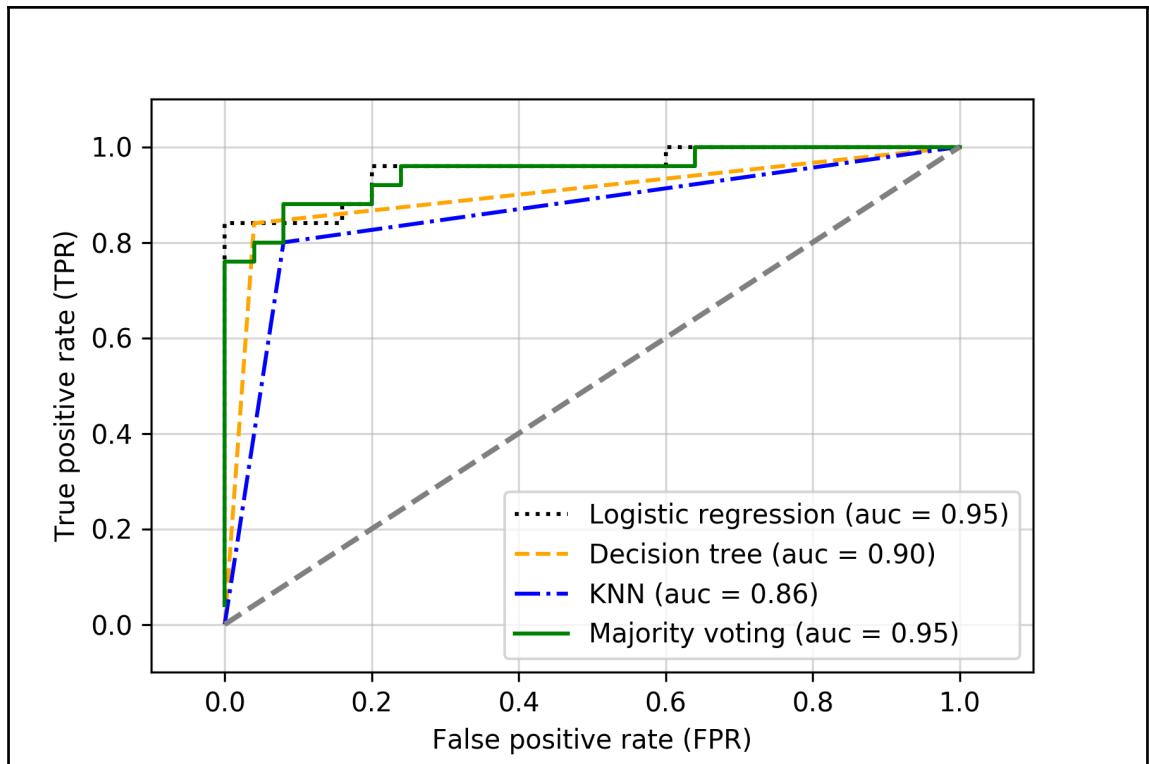


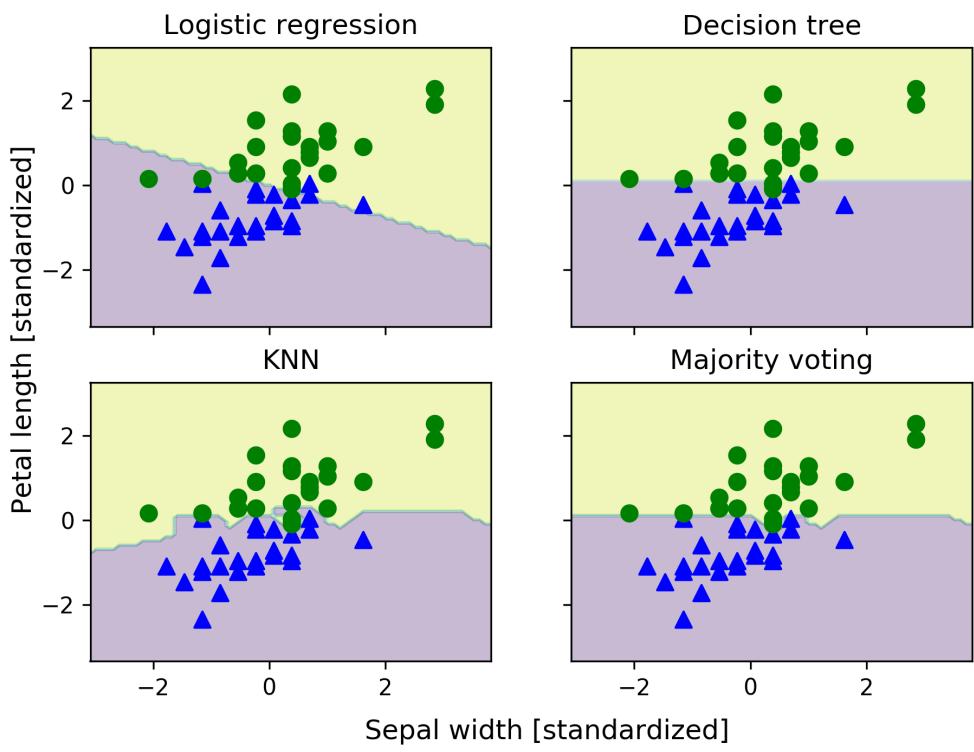


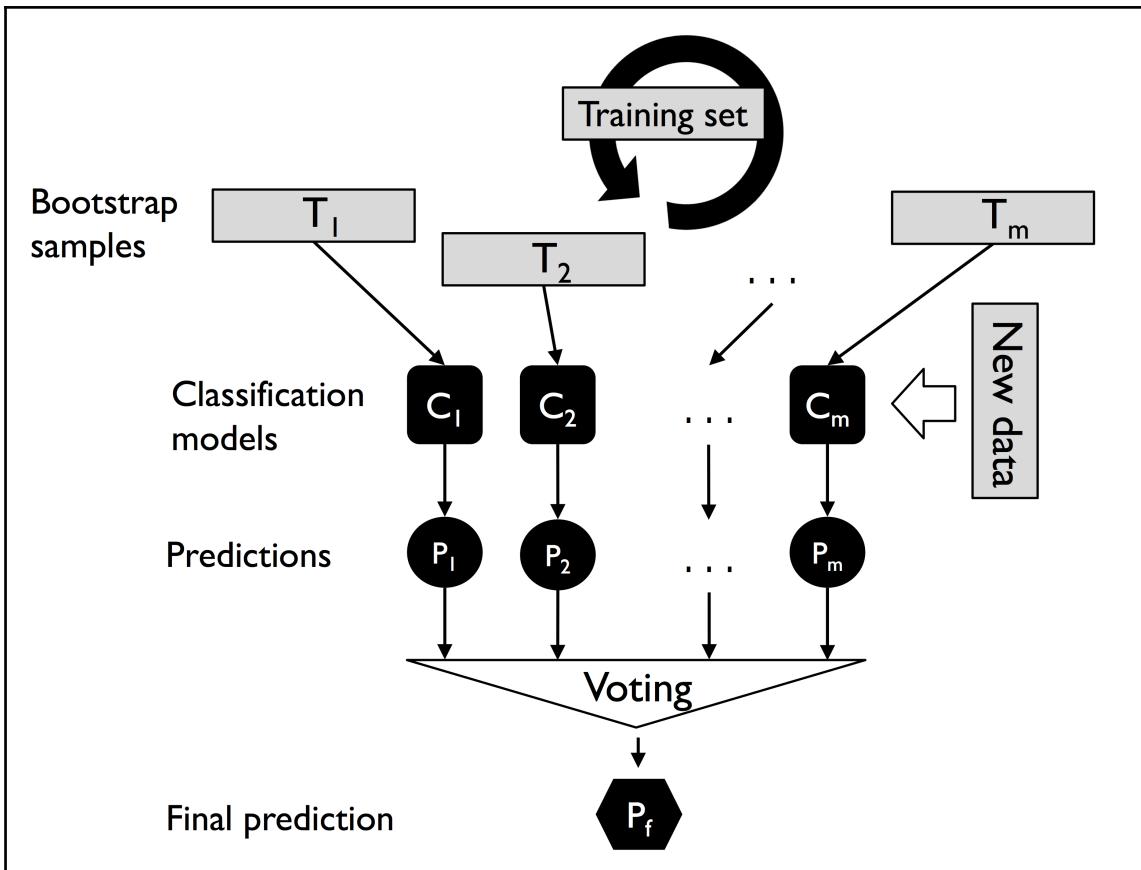
# Chapter 7: Combining Different Models for Ensemble Learning





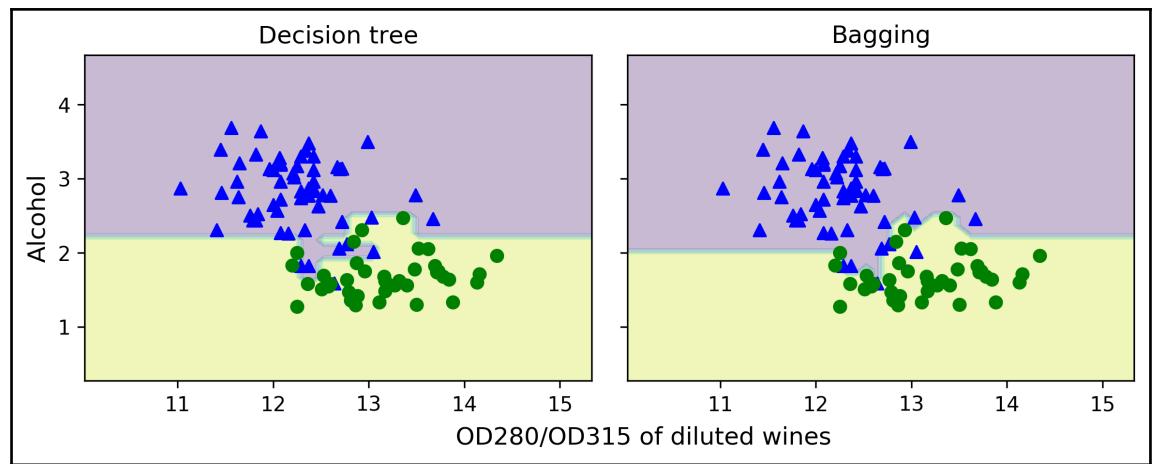


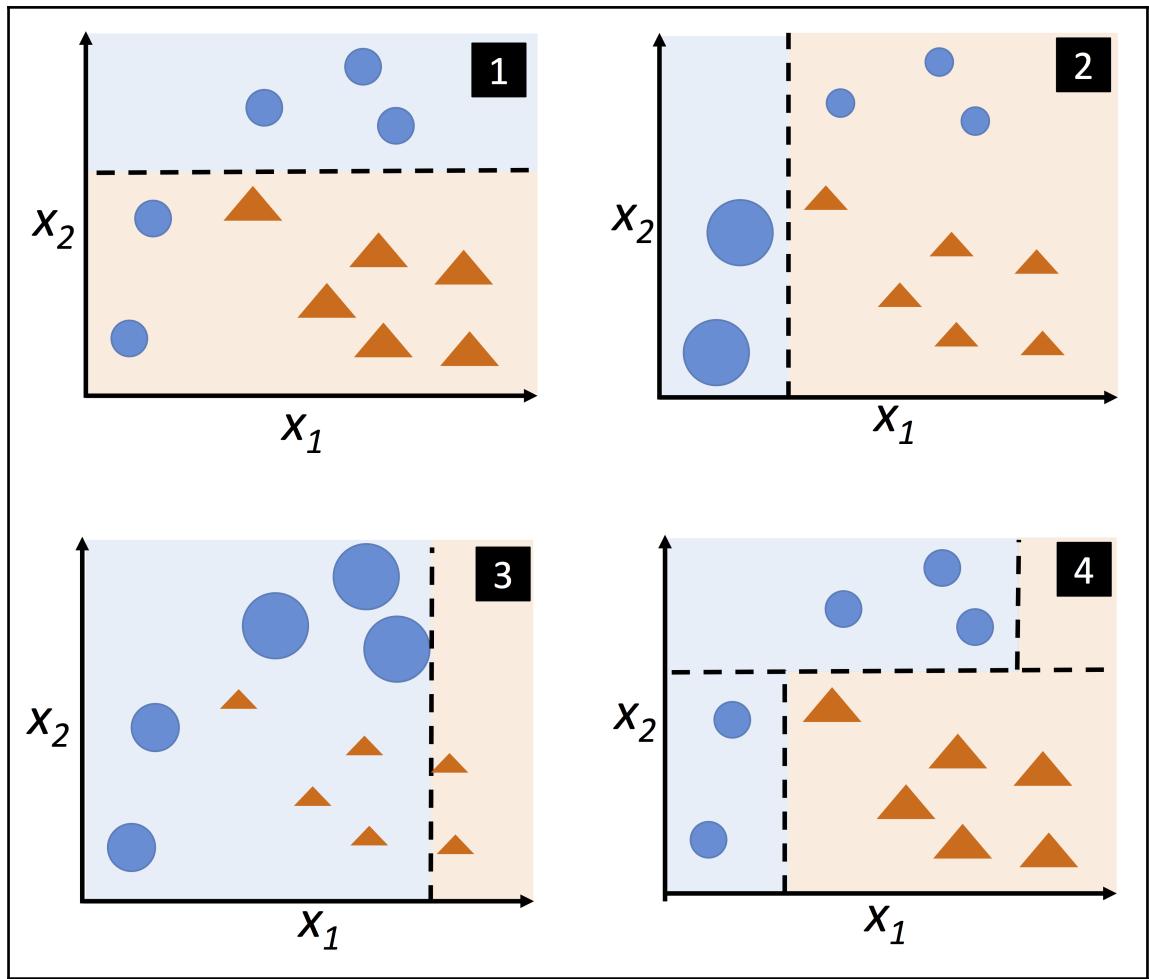




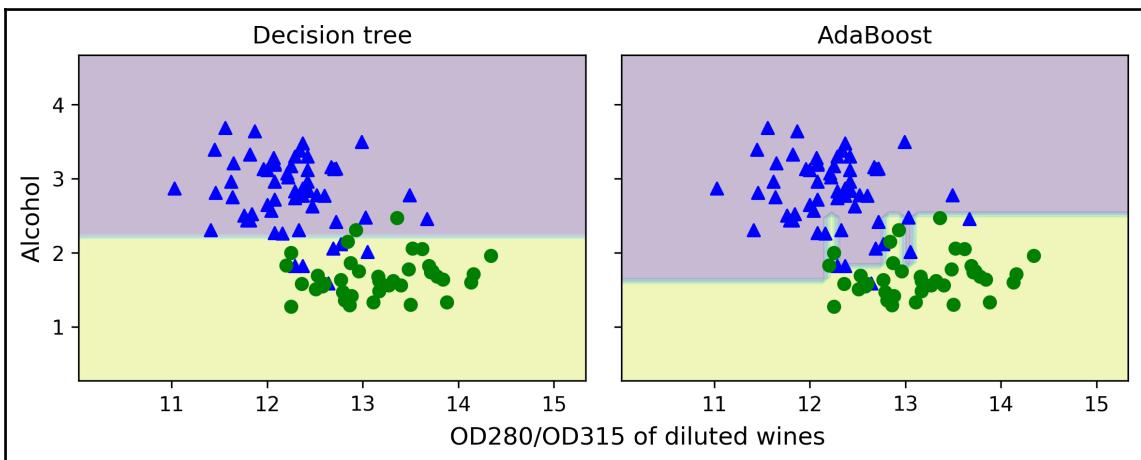
Sample indices	Bagging round 1	Bagging round 2	...
1	2	7	...
2	2	3	...
3	1	2	...
4	3	1	...
5	7	1	...
6	2	7	...
7	4	7	...

The diagram illustrates the output of the bagging process. Three arrows point from the rightmost column of the table to three separate components labeled  $C_1$ ,  $C_2$ , and  $C_m$ , representing different classifiers trained on different subsets of the data.





Sample indices	x	y	Weights	$\hat{y}(x \leq 3.0)$ ?	Correct?	Updated weights
1	1.0	1	0.1	1	Yes	0.072
2	2.0	1	0.1	1	Yes	0.072
3	3.0	1	0.1	1	Yes	0.072
4	4.0	-1	0.1	-1	Yes	0.072
5	5.0	-1	0.1	-1	Yes	0.072
6	6.0	-1	0.1	-1	Yes	0.072
7	7.0	1	0.1	-1	No	0.167
8	8.0	1	0.1	-1	No	0.167
9	9.0	1	0.1	-1	No	0.167
10	10.0	-1	0.1	-1	Yes	0.072



# Chapter 8: Applying Machine Learning to Sentiment Analysis

	review	sentiment
0	In 1974, the teenager Martha Moxley (Maggie Gr...	1
1	OK... so... I really like Kris Kristofferson a...	0
2	***SPOILER*** Do not read this, if you think a...	0

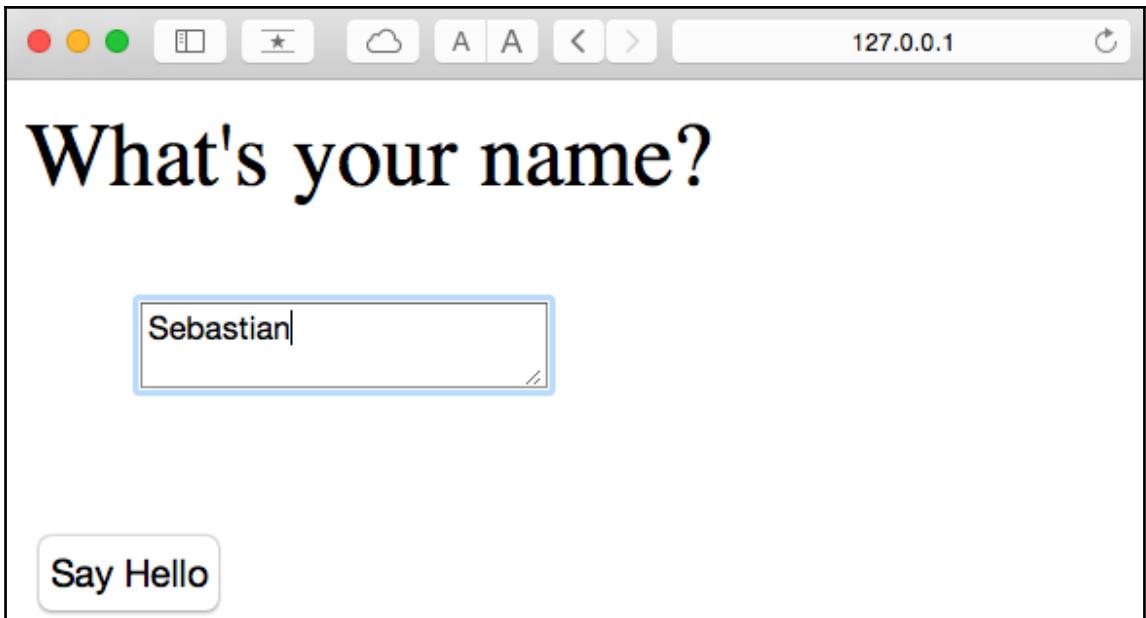
# Chapter 9: Embedding a Machine Learning Model into a Web Application

The screenshot shows the SQLite Manager interface. The title bar reads "SQLite Manager - /Users/sebastian/Desktop/ch09/reviews.sqlite". The left sidebar shows a tree view of the database structure under "reviews.sqlite": "Master Table (1)", "Tables (1)" (expanded to show "review\_db"), "Views (0)", "Indexes (0)", and "Triggers (0)". The main area displays the "review\_db" table with the following data:

rowid	review	sentiment	date
1	I love this movie	1	2017-04-24 00:14:38
2	I disliked this movie	0	2017-04-24 00:14:38

At the bottom, the status bar shows "SQLite 3.17.0 Gecko 53.0 0.8.3.1-signed.1-signed Shared Number of files in selected directory: 8 ET: 1 ms".

The screenshot shows a web browser window with the address bar set to "127.0.0.1". The main content area displays the text "Hi, this is my first Flask web app!".



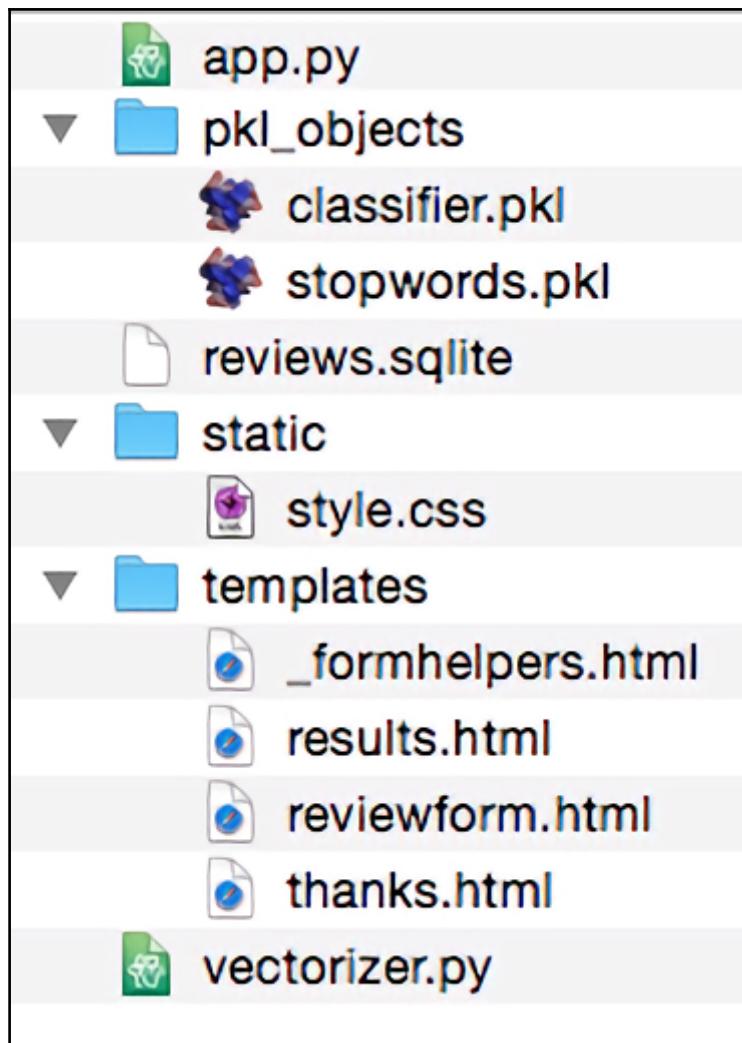
A screenshot of a web browser window with a light gray header bar. The address bar shows the URL "127.0.0.1". The main content area contains the following text:

**Please enter your movie review:**

Below this text is a large, empty rectangular input field with a thin black border. At the bottom left of the input field, there is a small double-line cursor icon. To the left of the input field is a "Submit review" button with a light gray background and a thin black border.

A screenshot of a web browser window titled "raschkas.pythonanywhere.com/results". The page displays a movie review: "I love this movie!". Below it, the text "Prediction:" is followed by "This movie review is **positive** (probability: 90.86%)." At the bottom, there are two buttons: "Correct" and "Incorrect", and a link "Submit another review".

A screenshot of a web browser window titled "raschkas.pythonanywhere.com/thanks". The page displays the message "Thank you for your feedback!" and a link "Submit another review".



A screenshot of a web browser window with a light gray header bar. The address bar shows the URL "127.0.0.1". The main content area contains the following text:

**Please enter your movie review:**

Below this text is a large, empty rectangular input field with a thin black border. At the bottom left of the input field, there is a small double-line symbol. To the left of the input field is a "Submit review" button with a light gray background and a thin black border.

The screenshot shows a web browser window for PythonAnywhere.com. The top navigation bar includes links for Send feedback, Forums, Help, Blog, Dashboard, Account, and Log out. Below the navigation is a horizontal menu with tabs: Consoles, Files (which is highlighted), Web, Schedule, and Databases. The main content area displays a file tree under the path / home / raschka / movieclassifier. It shows a directory named 'movieclassifier' containing files 'app.py', 'reviews.sqlite', and 'vectorizer.py'. There are also hidden directories like \_\_pycache\_\_ and pkl\_objects/. A quota message indicates the account is 3% full (16.4 MB of 512.0 MB quota). Below the file tree, there are input fields for creating new files ('Enter new file name, eg hello.py') and new directories ('Enter new directory name'). An orange button at the bottom left says 'Upload a file'.

pythonanywhere.com

Send feedback Forums Help Blog Dashboard Account Log out

Consoles Files Web Schedule Databases

/ home / raschka / movieclassifier Open Bash console here 3% full (16.4 MB of your 512.0 MB quota)

Directories

Enter new directory name New directory

\_\_pycache\_\_/  
pkl\_objects/  
static/  
templates/

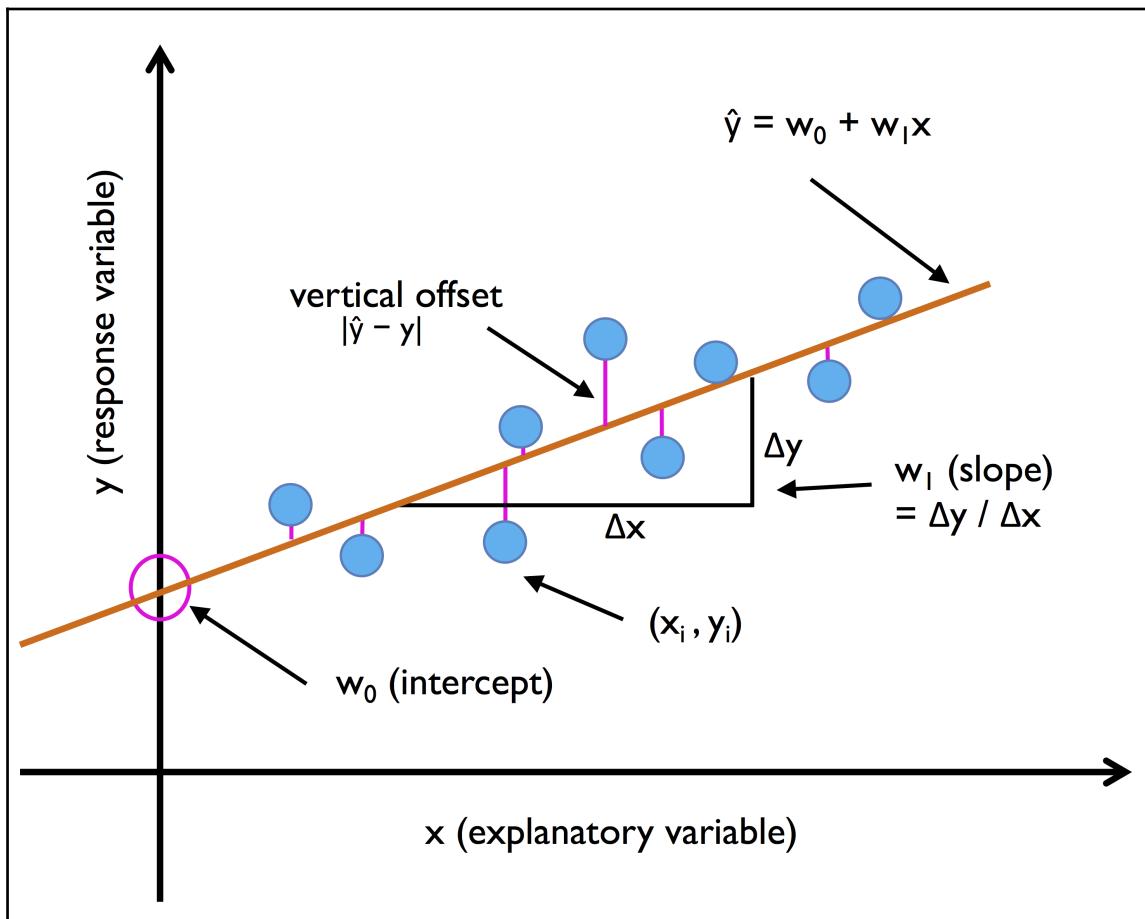
Files

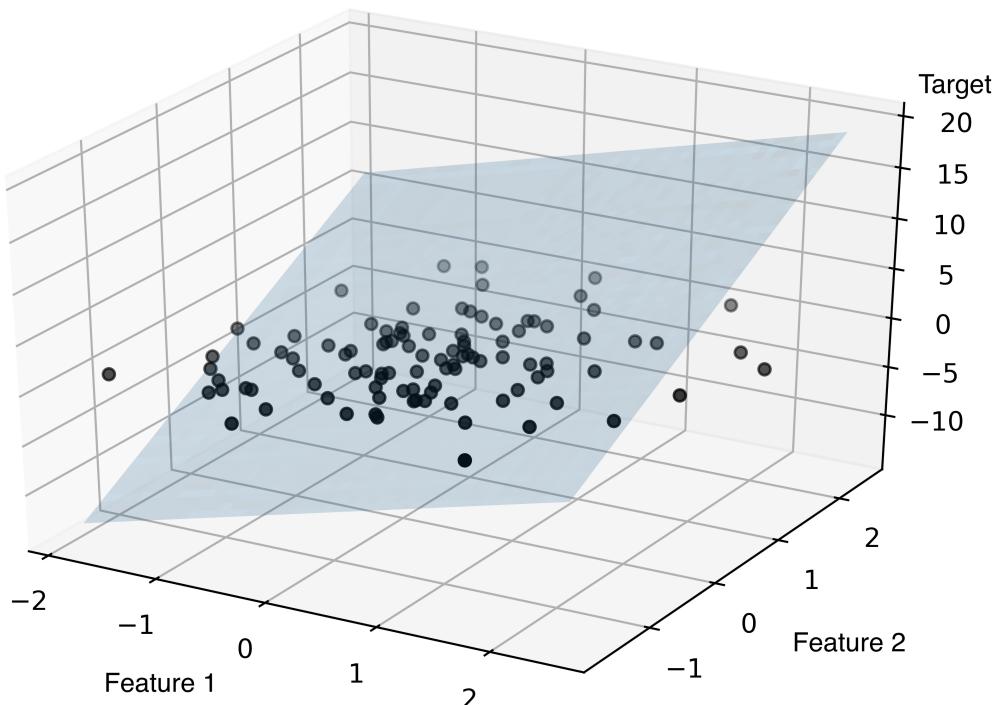
Enter new file name, eg hello.py New file

app.py reviews.sqlite vectorizer.py

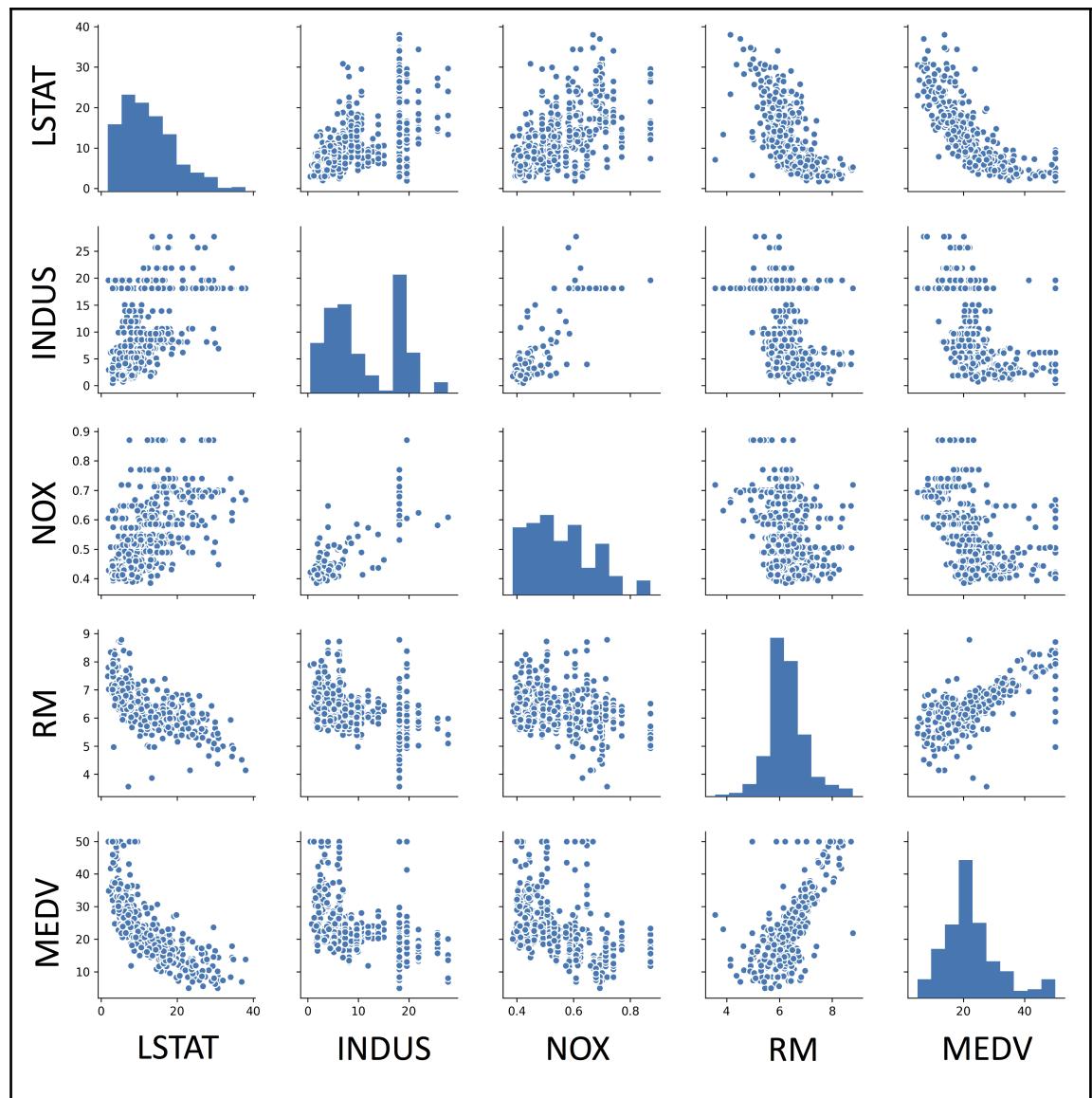
Upload a file

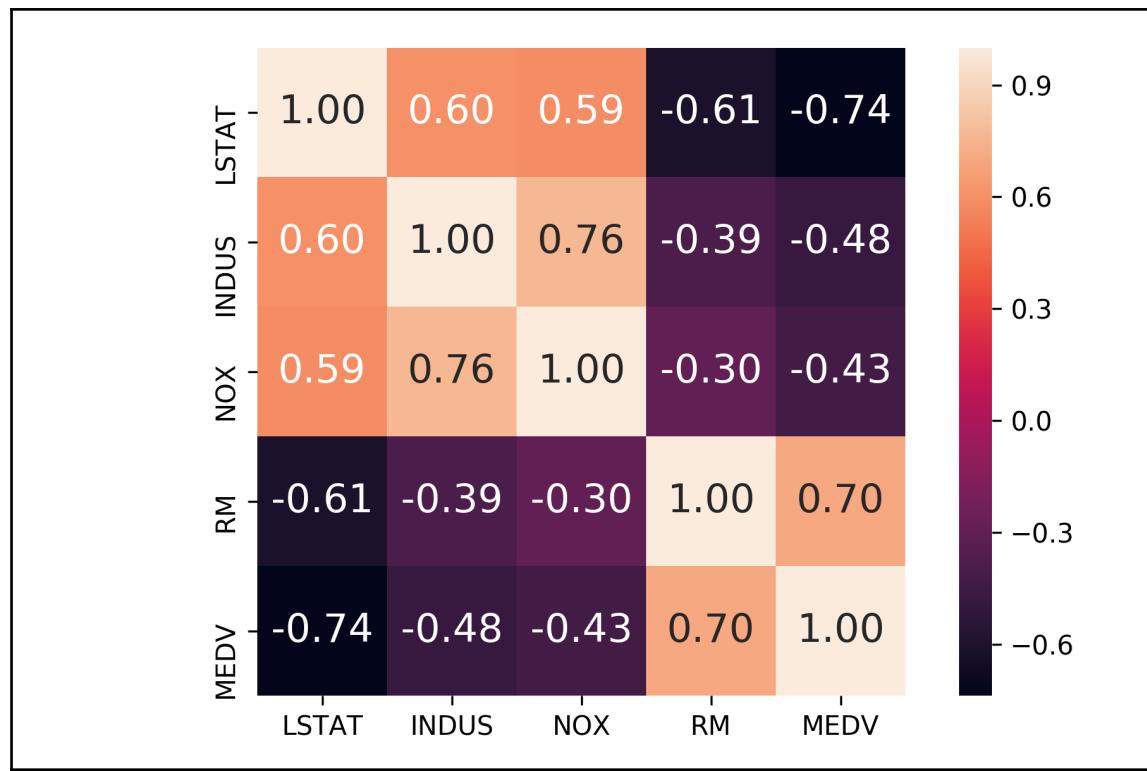
# Chapter 10: Predicting Continuous Target Variables with Regression Analysis

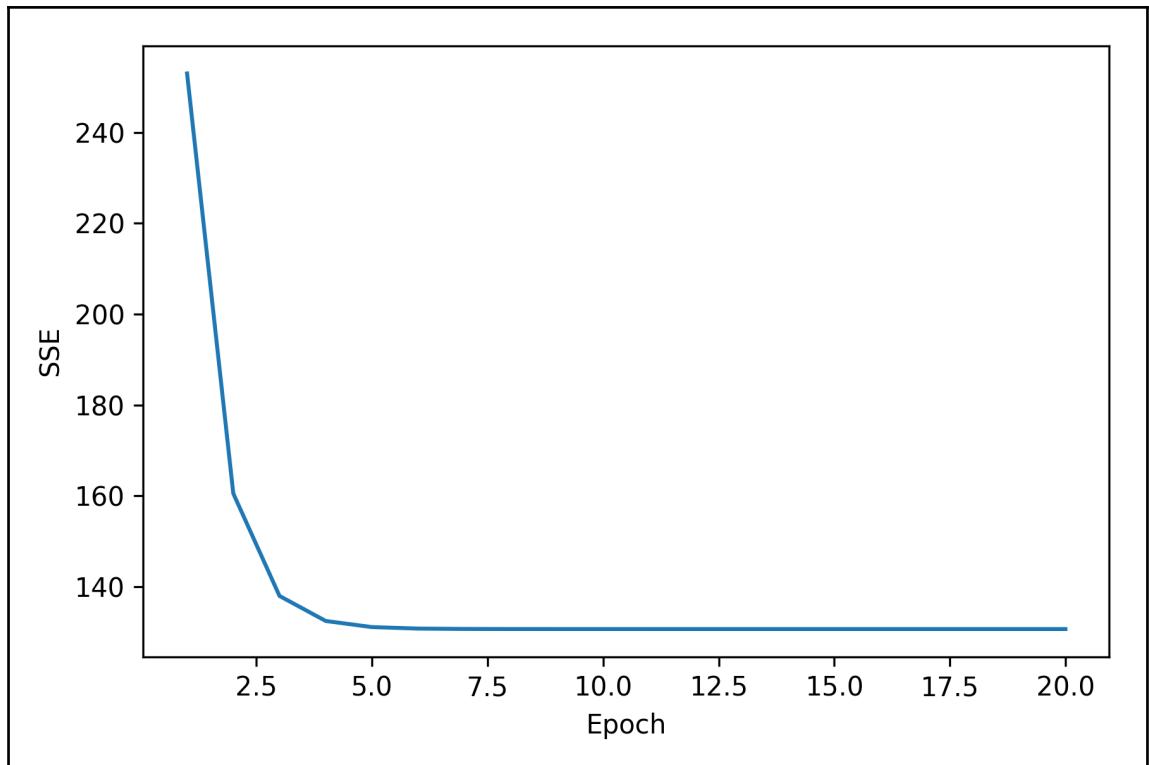


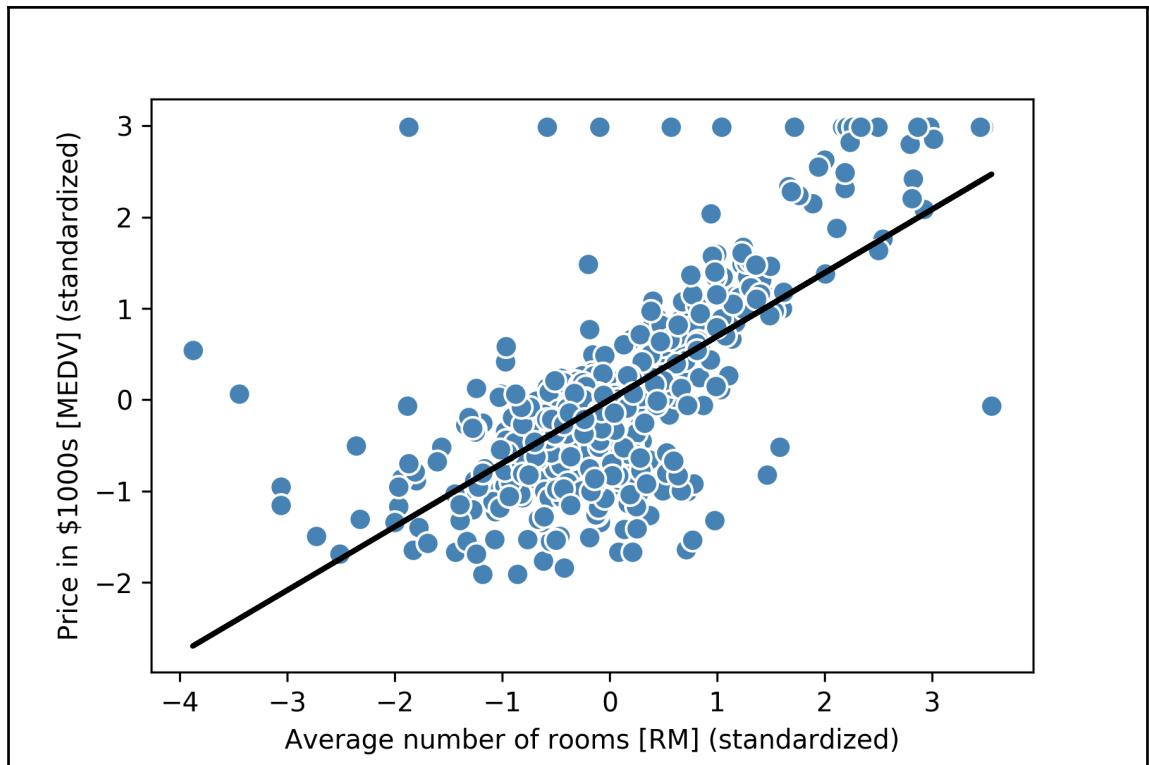


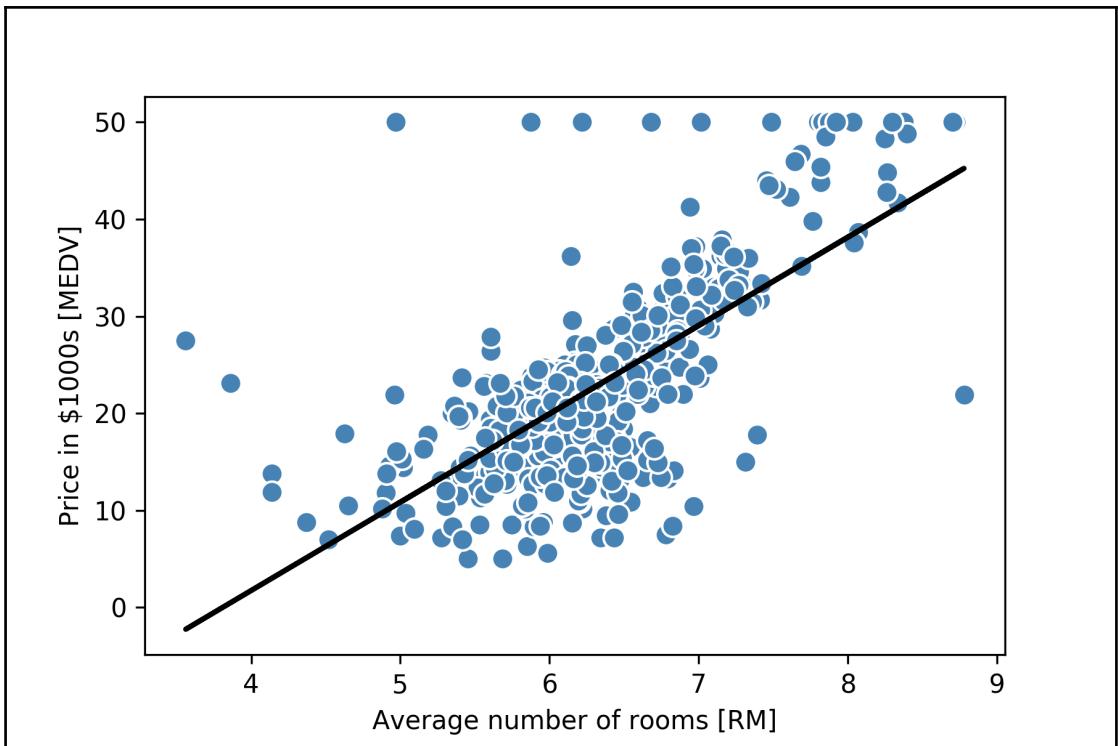
	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296.0	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242.0	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242.0	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222.0	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222.0	18.7	396.90	5.33	36.2

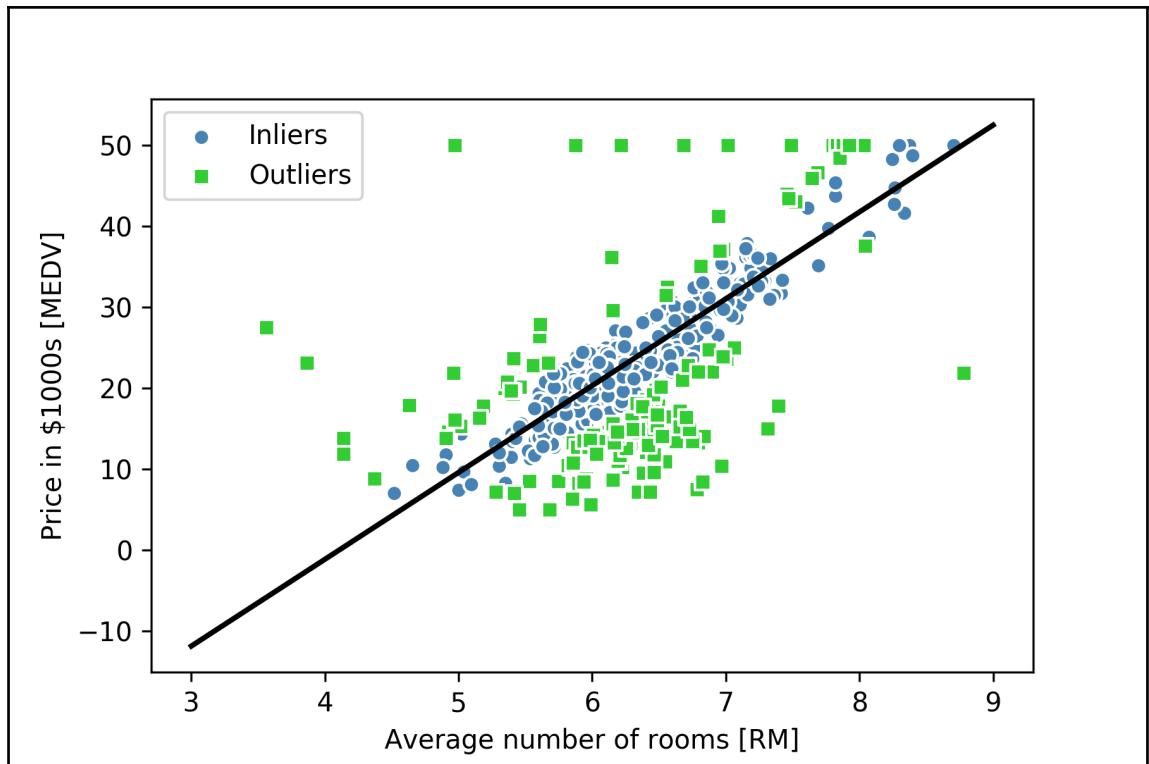


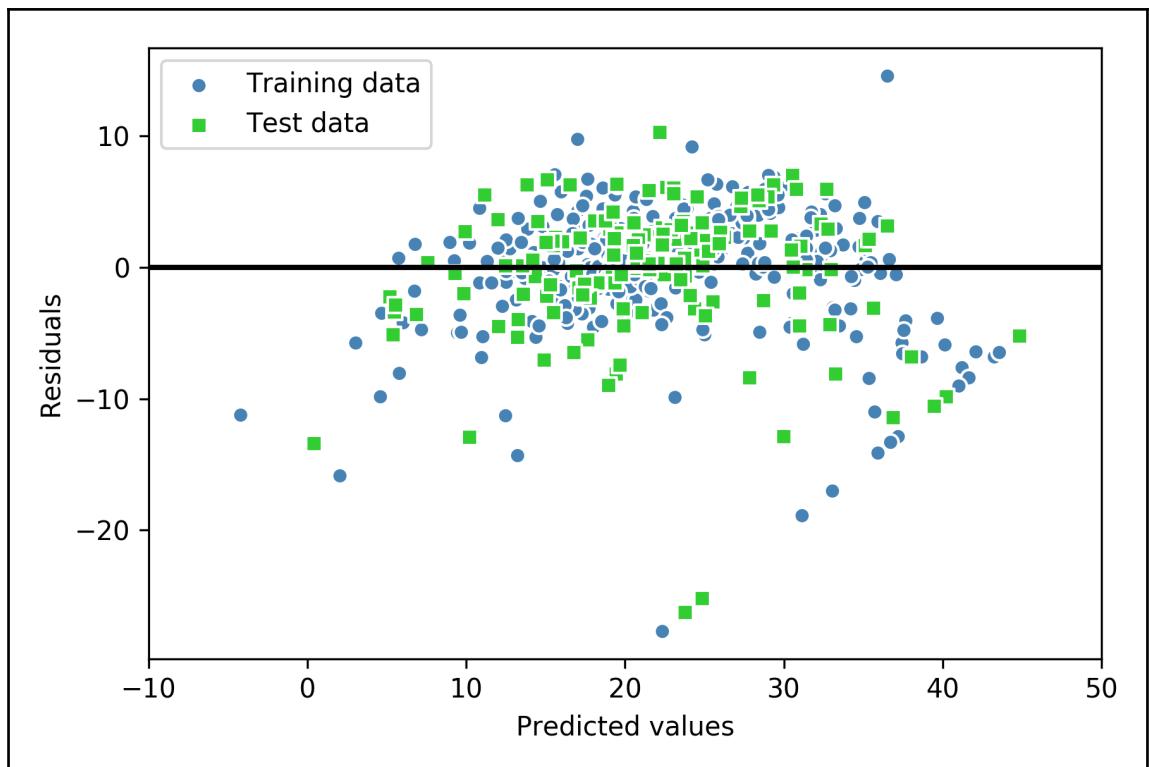


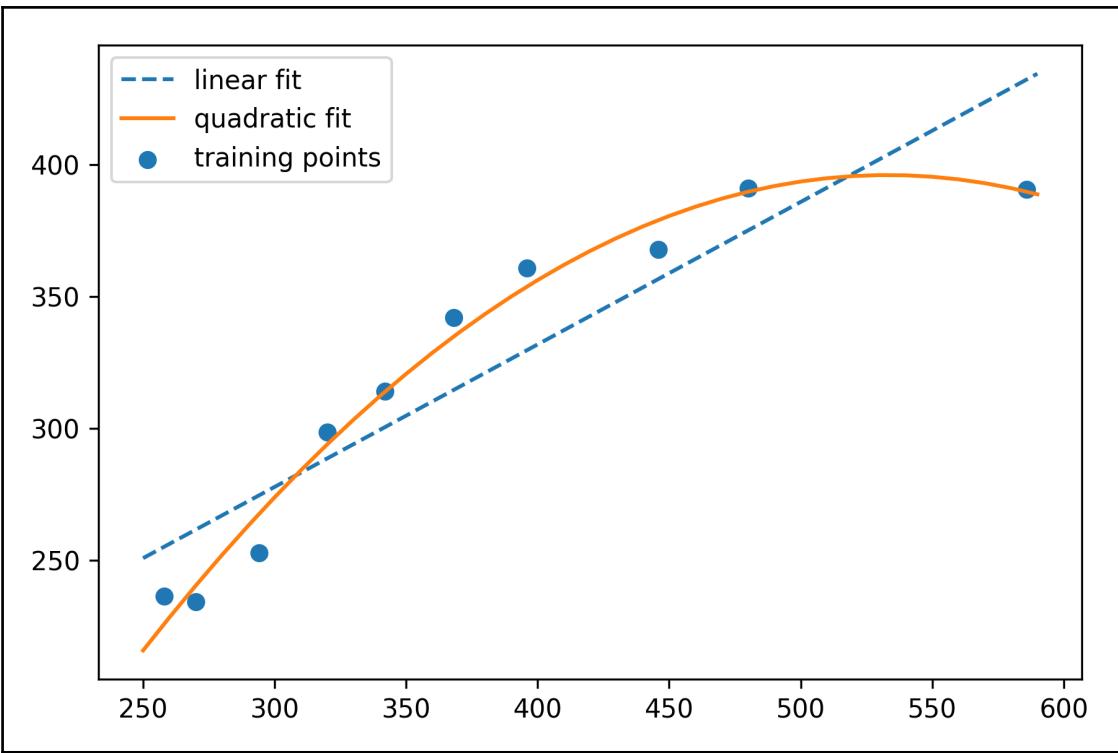


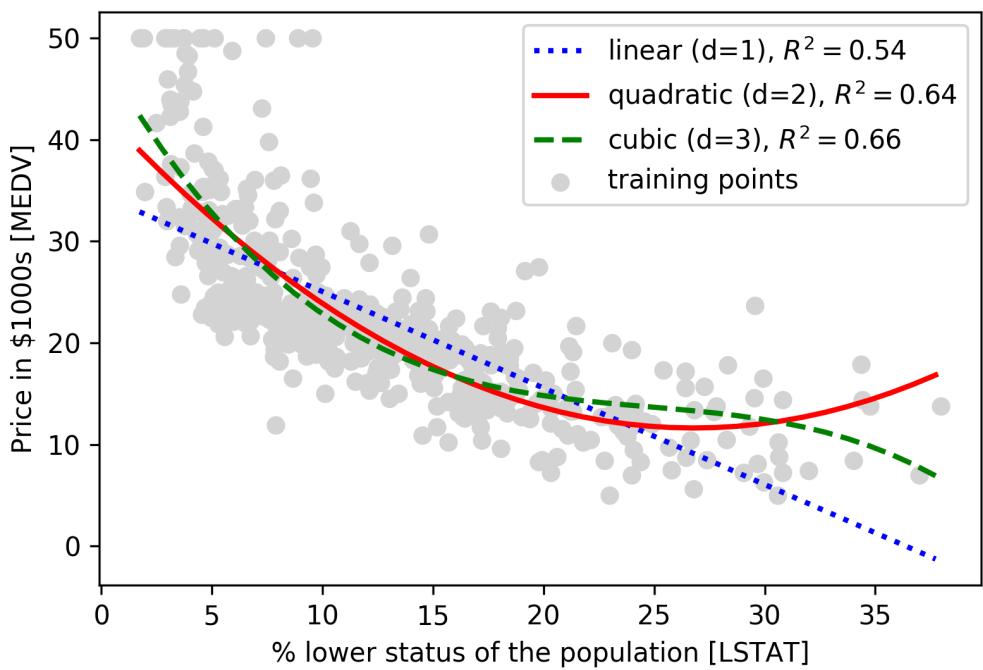


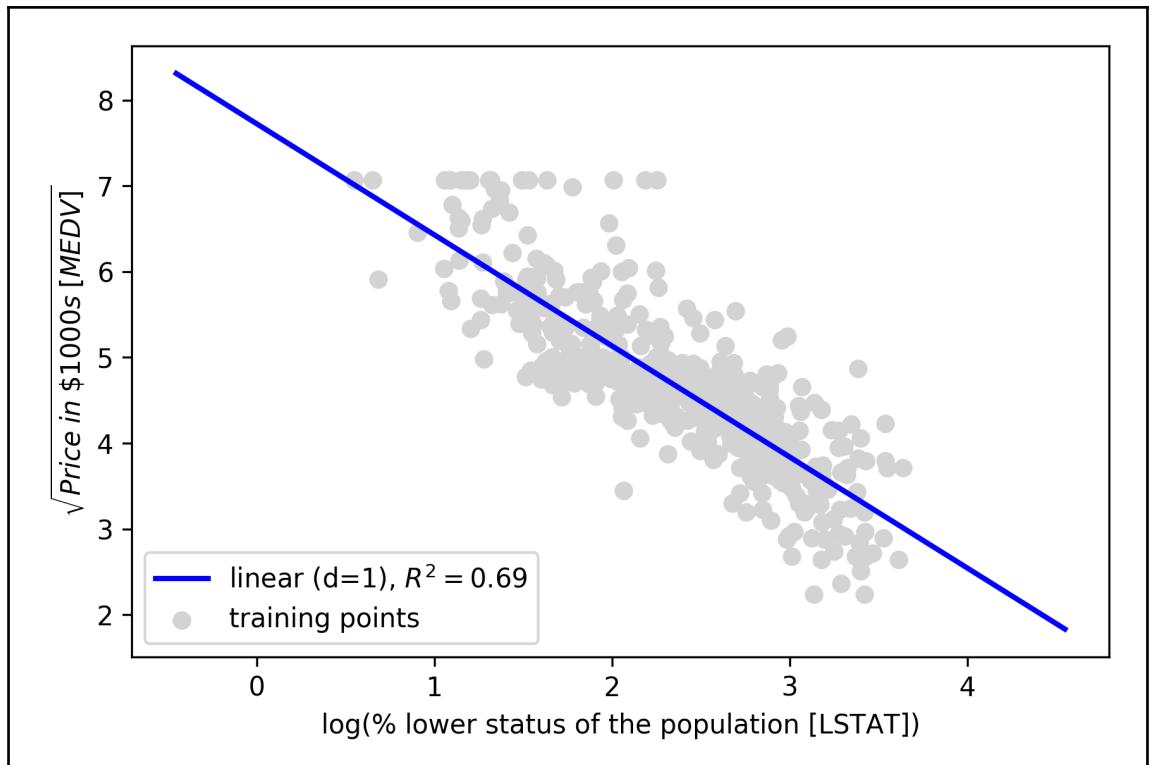


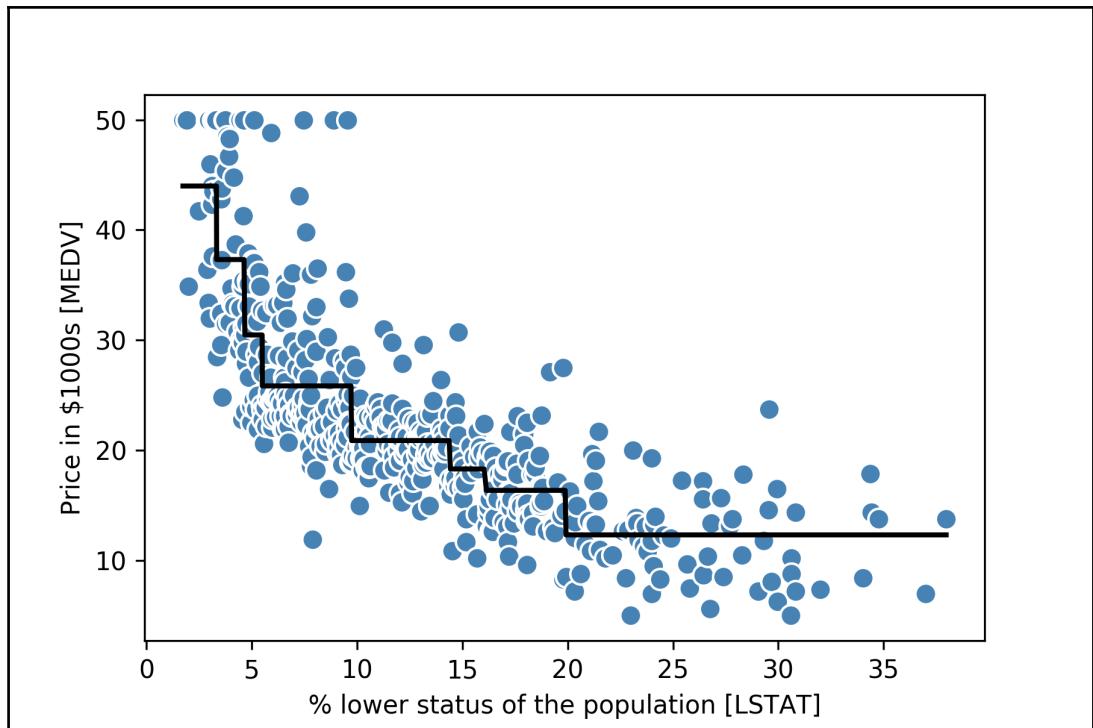


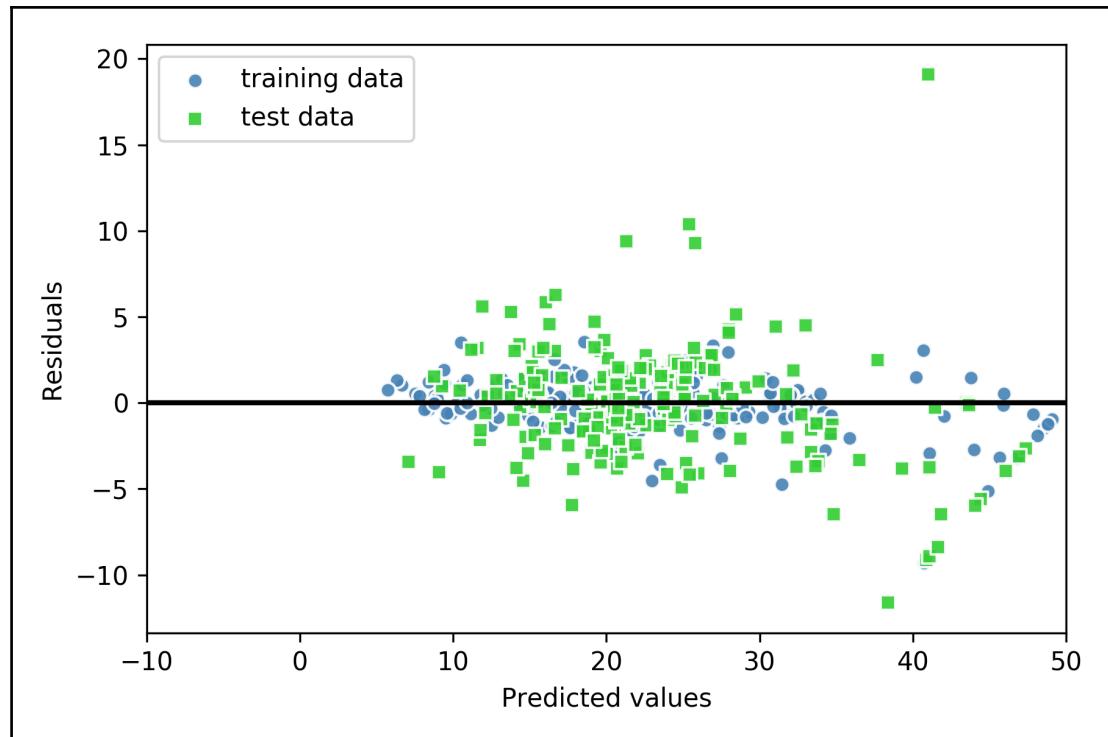




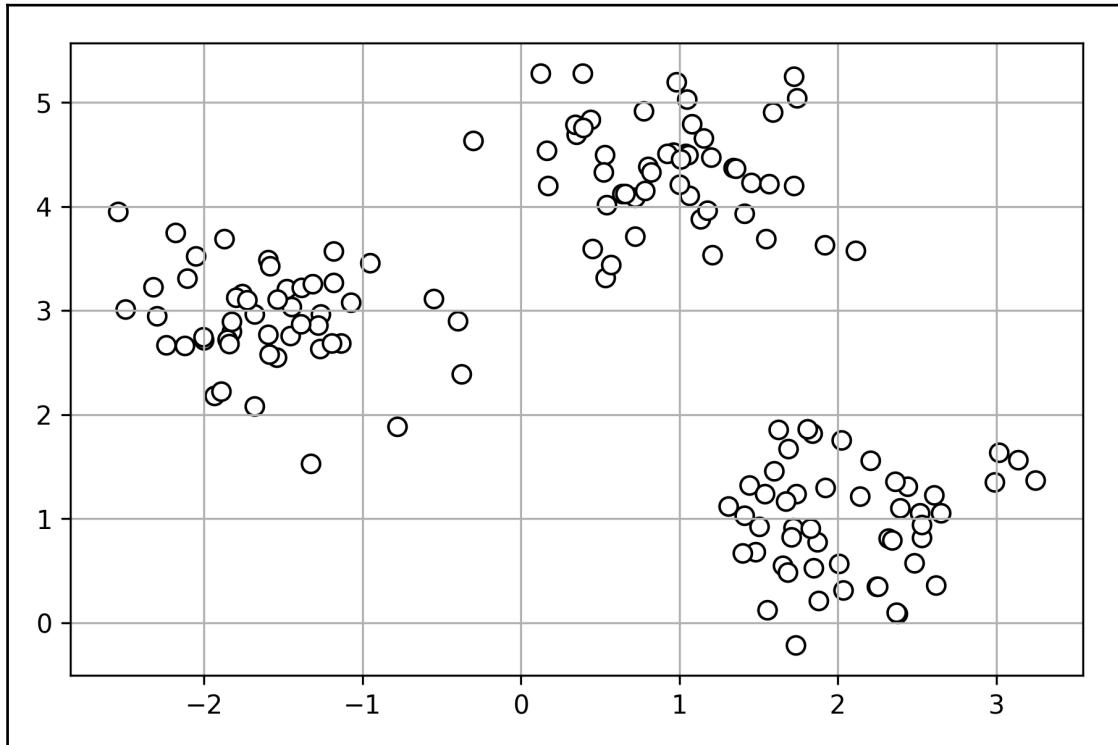


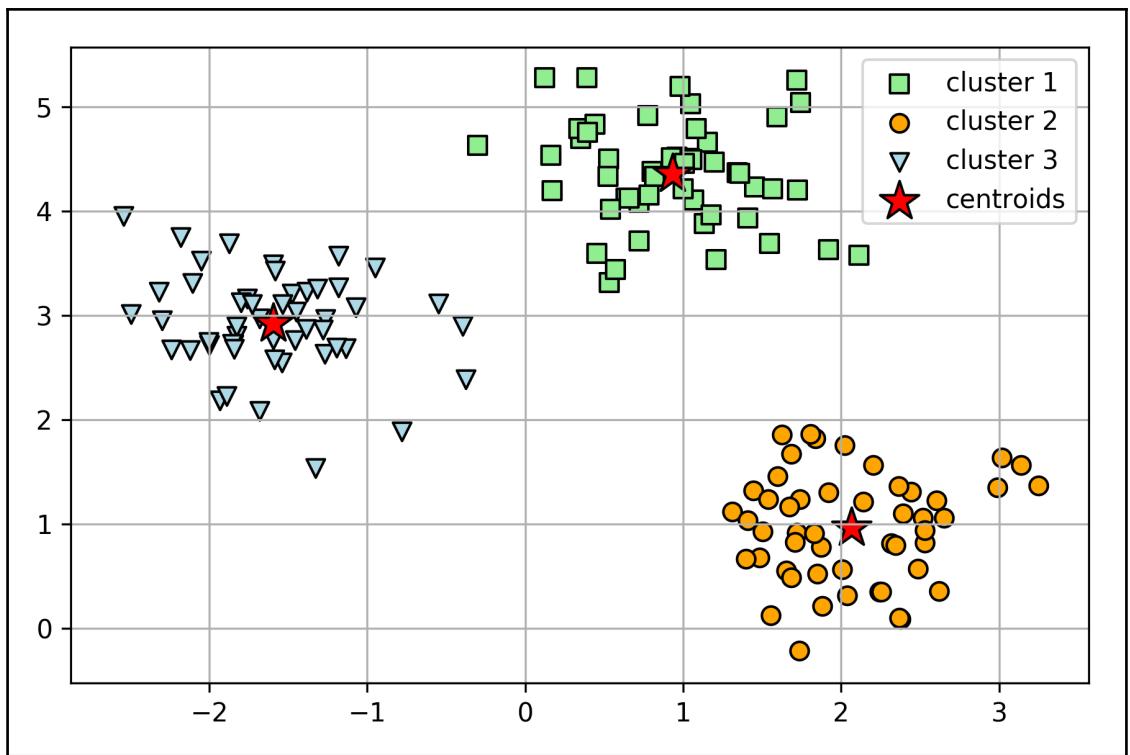


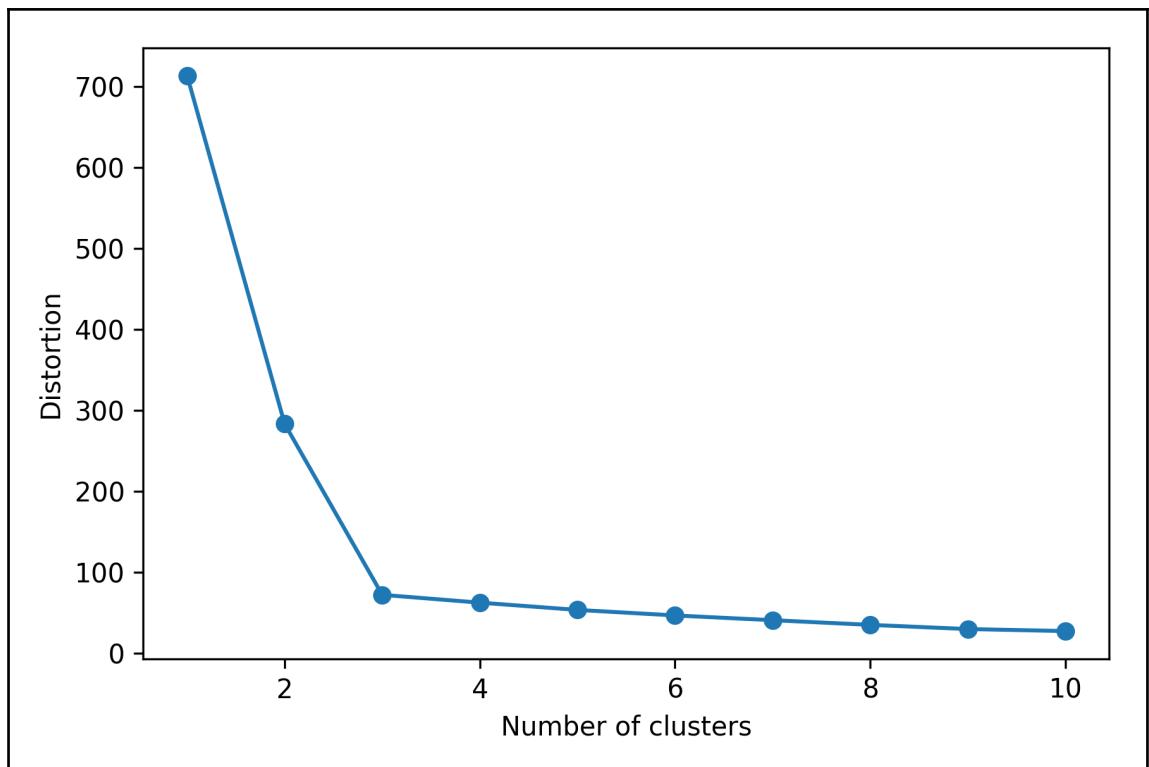


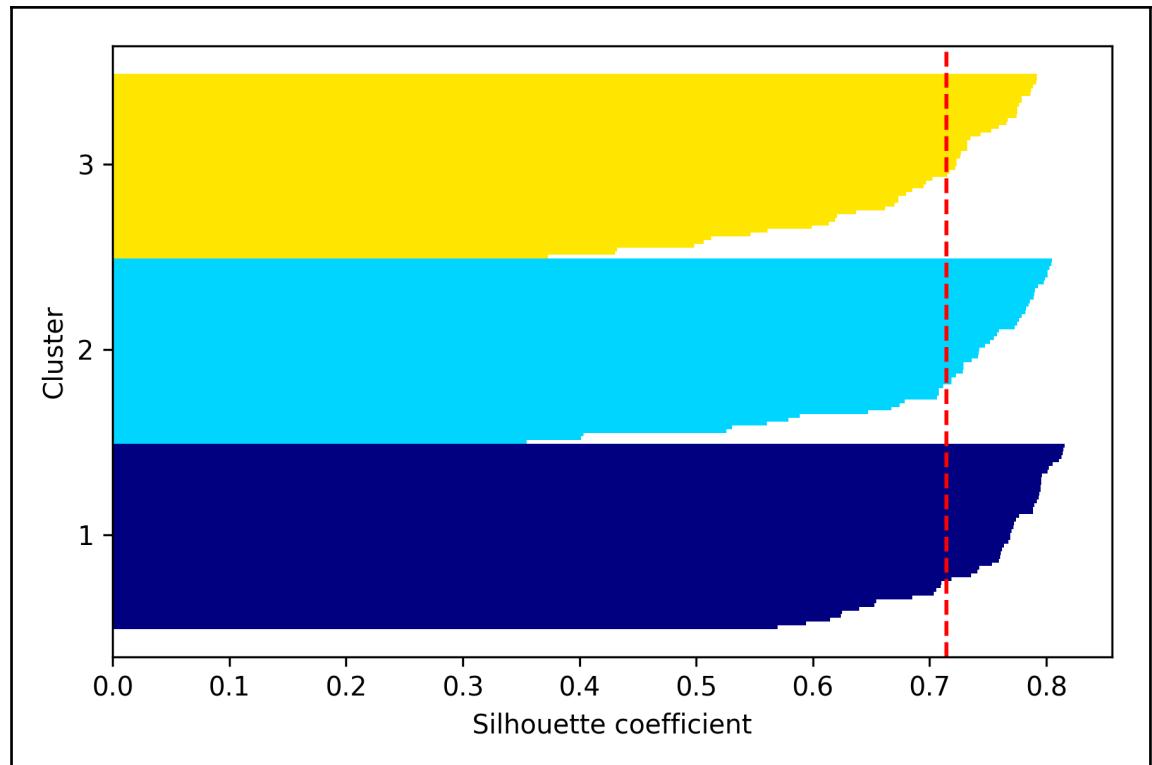


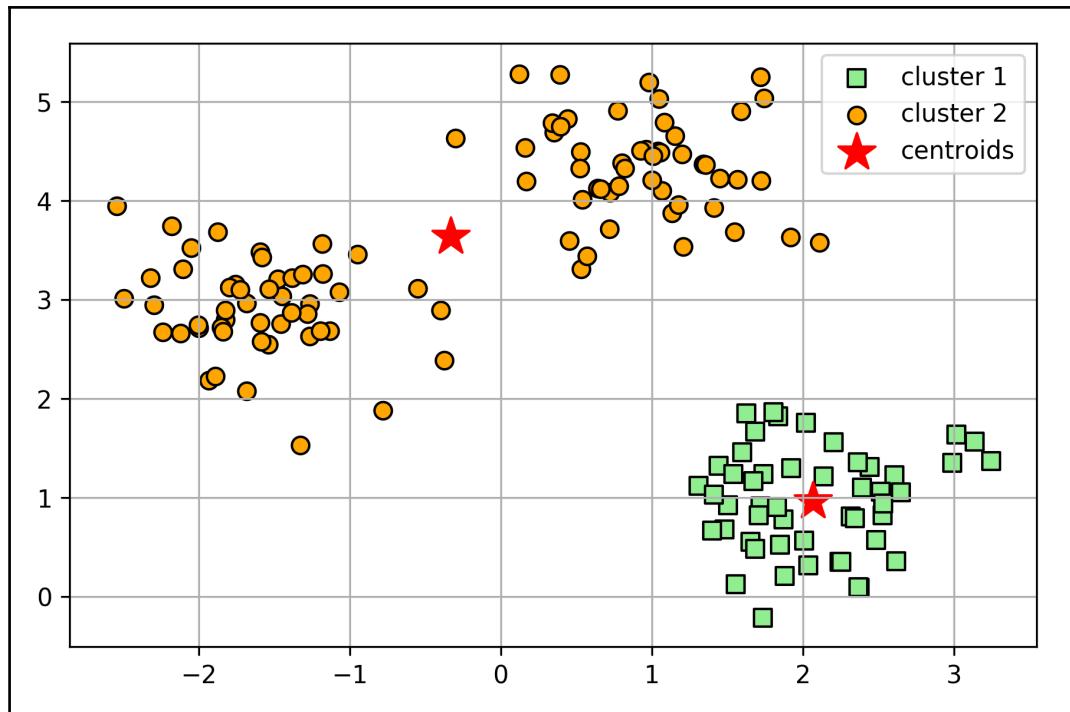
# Chapter 11: Working with Unlabeled Data – Clustering Analysis

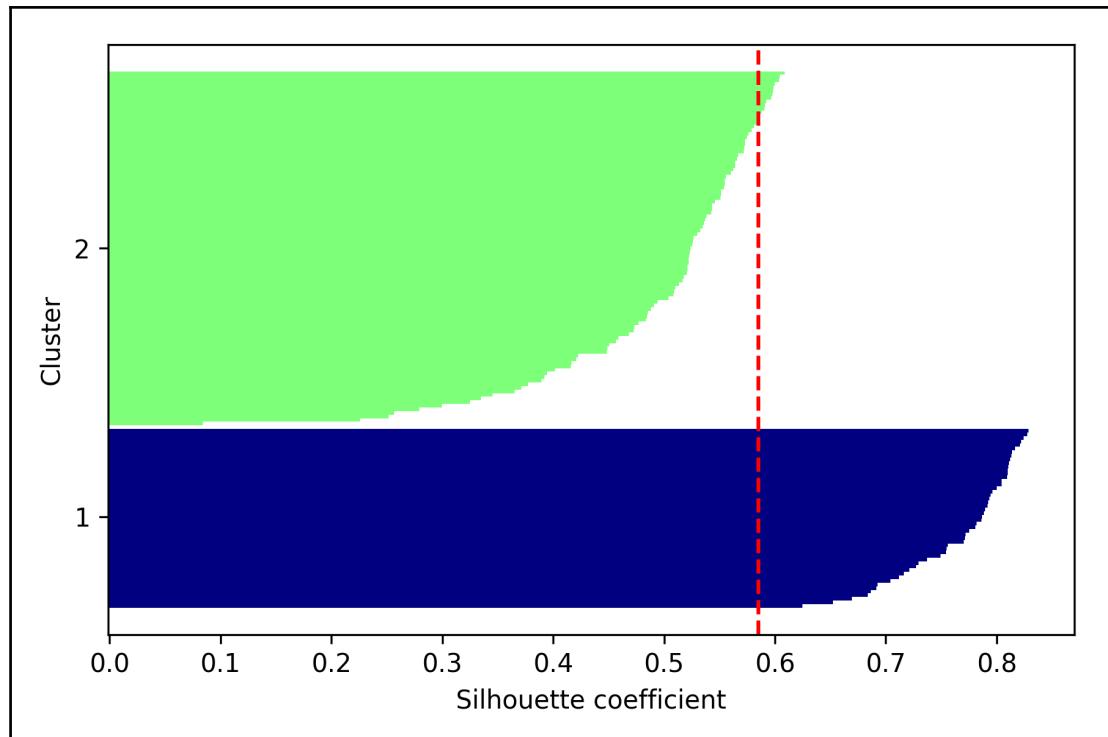


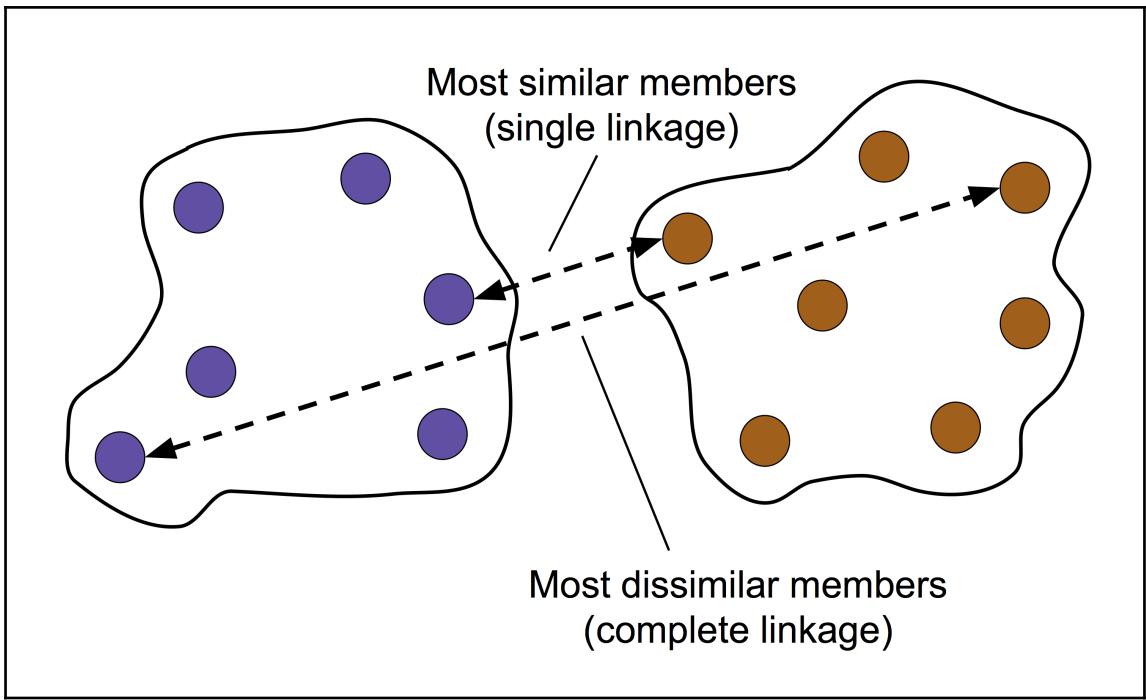








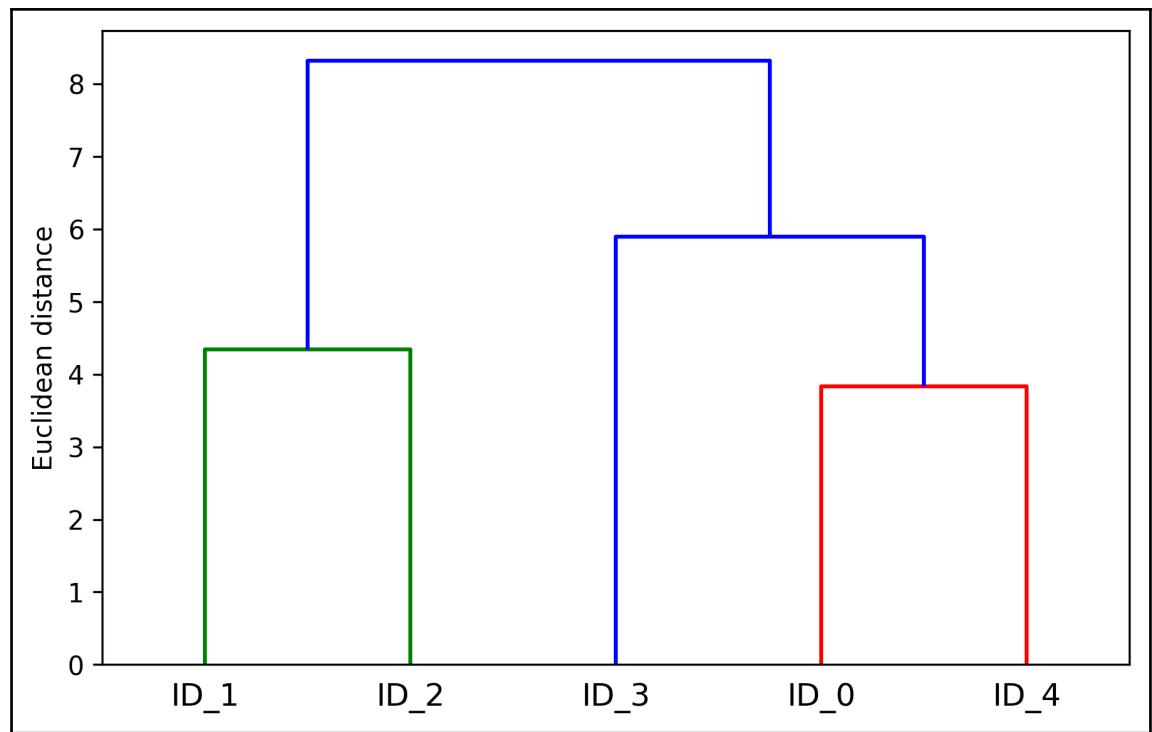


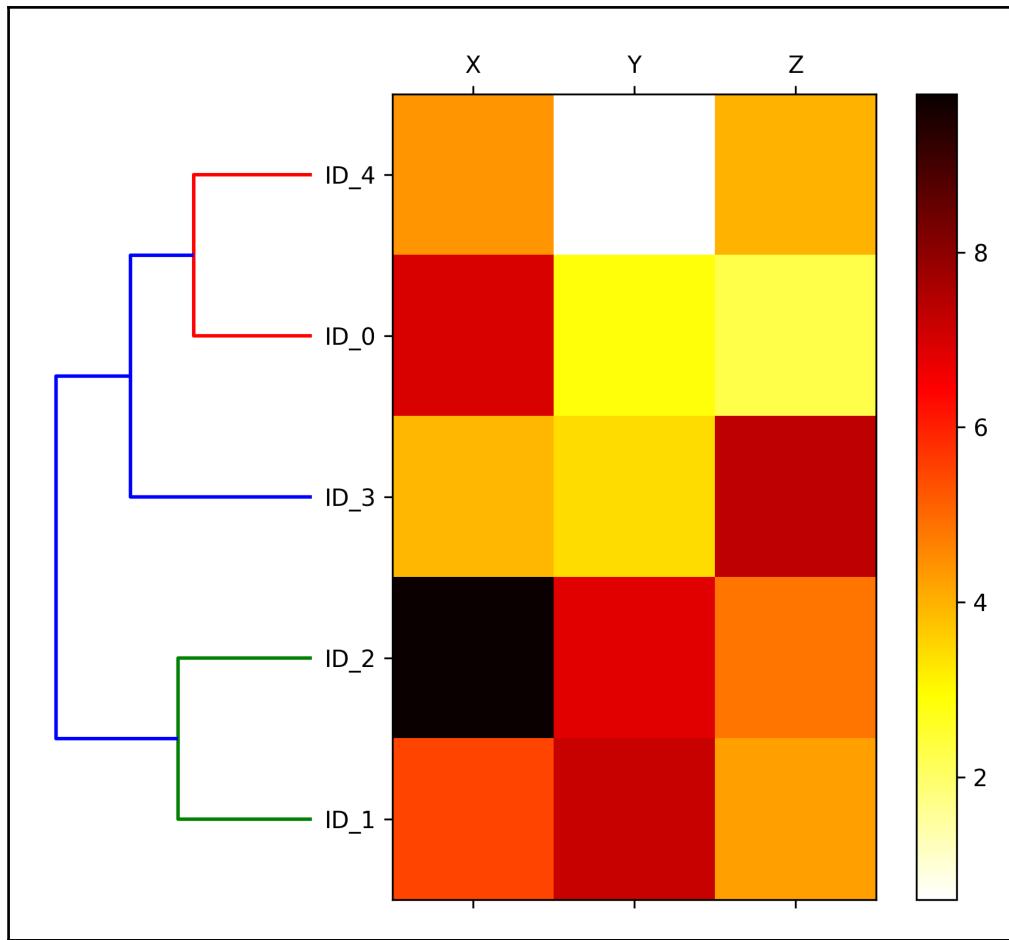


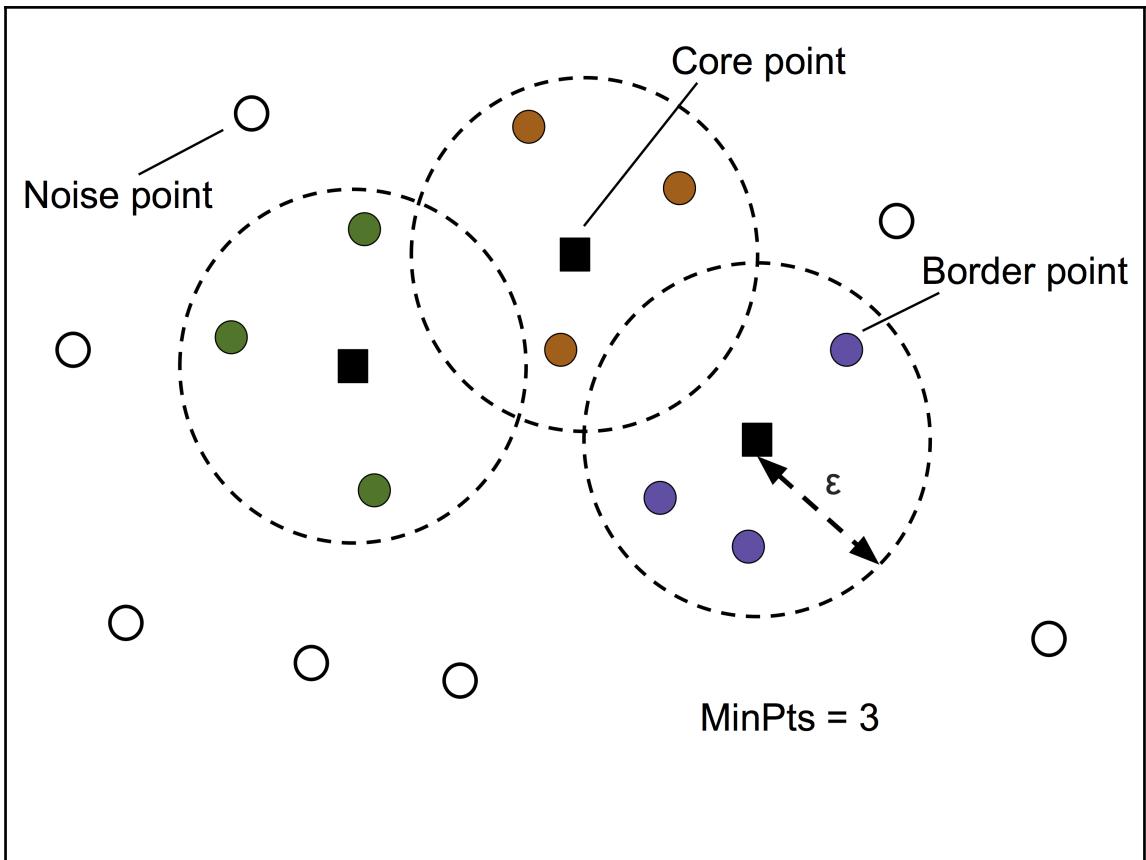
	<b>X</b>	<b>Y</b>	<b>Z</b>
<b>ID_0</b>	6.964692	2.861393	2.268515
<b>ID_1</b>	5.513148	7.194690	4.231065
<b>ID_2</b>	9.807642	6.848297	4.809319
<b>ID_3</b>	3.921175	3.431780	7.290497
<b>ID_4</b>	4.385722	0.596779	3.980443

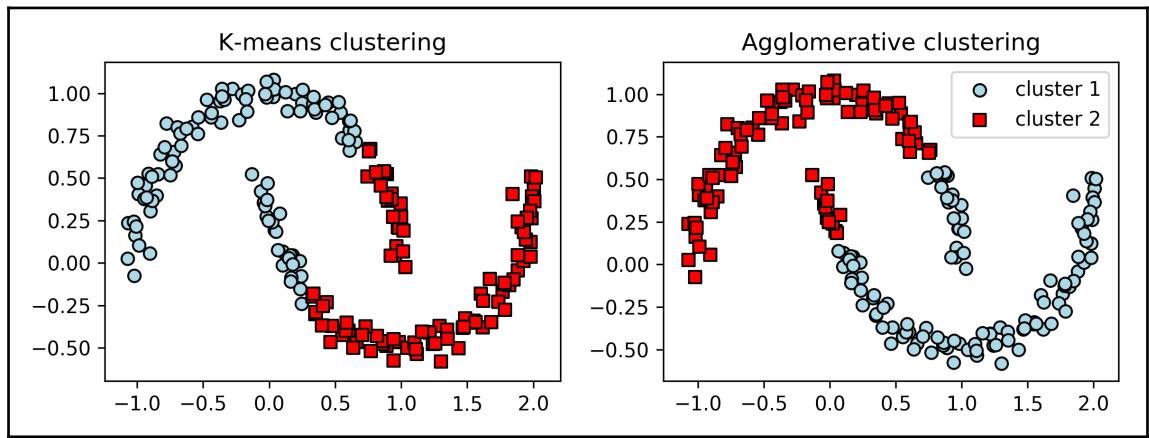
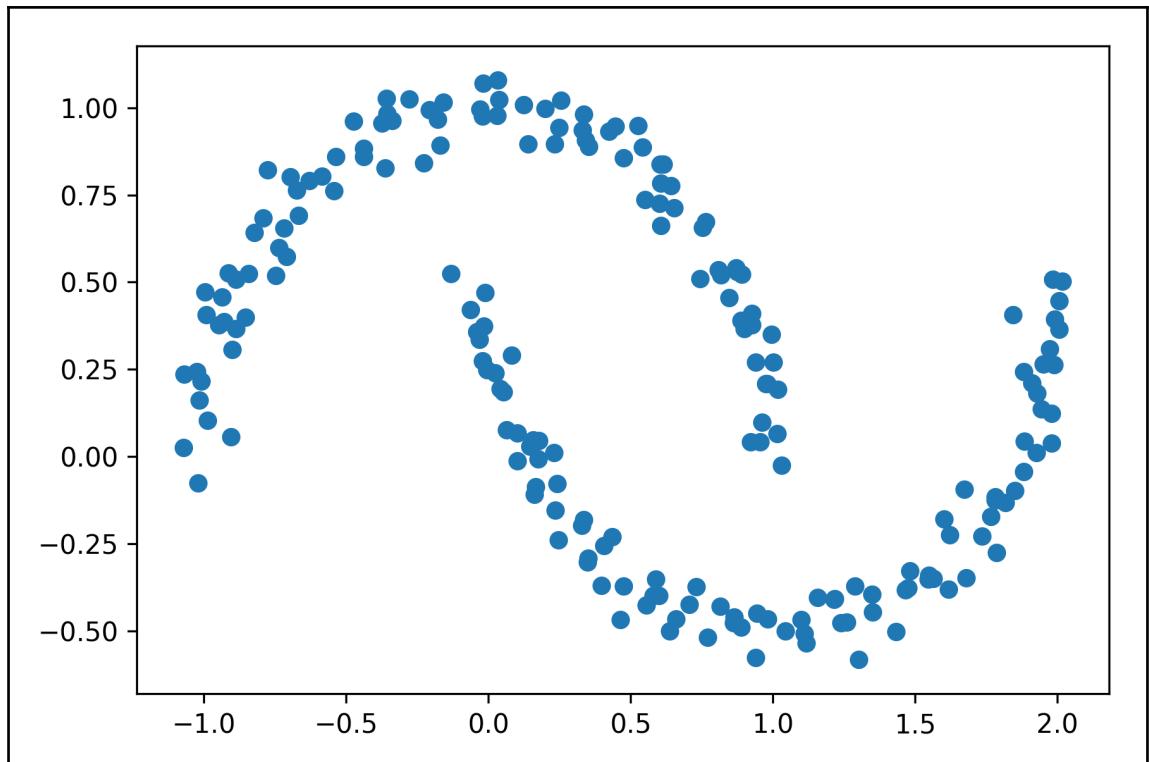
	<b>X</b>	<b>Y</b>	<b>Z</b>
<b>ID_0</b>	6.964692	2.861393	2.268515
<b>ID_1</b>	5.513148	7.194690	4.231065
<b>ID_2</b>	9.807642	6.848297	4.809319
<b>ID_3</b>	3.921175	3.431780	7.290497
<b>ID_4</b>	4.385722	0.596779	3.980443

	<b>row label 1</b>	<b>row label 2</b>	<b>distance</b>	<b>no. of items in clust.</b>
<b>cluster 1</b>	0.0	4.0	3.835396	2.0
<b>cluster 2</b>	1.0	2.0	4.347073	2.0
<b>cluster 3</b>	3.0	5.0	5.899885	3.0
<b>cluster 4</b>	6.0	7.0	8.316594	5.0

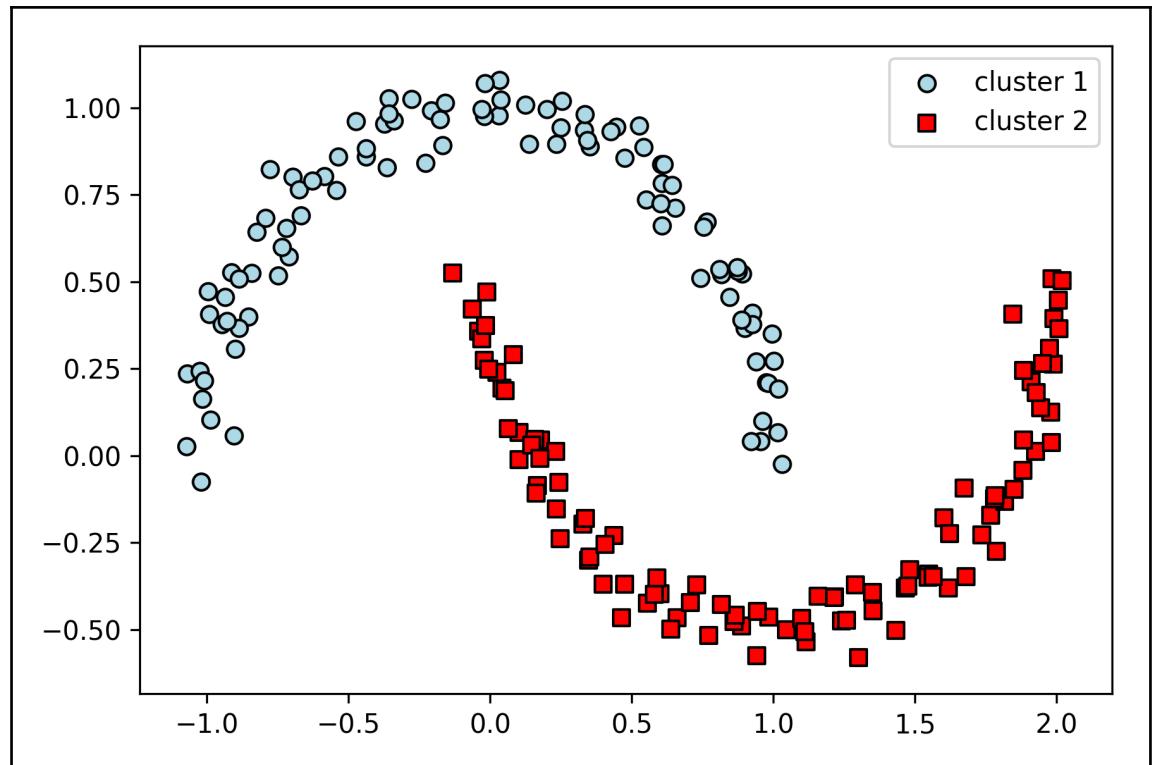




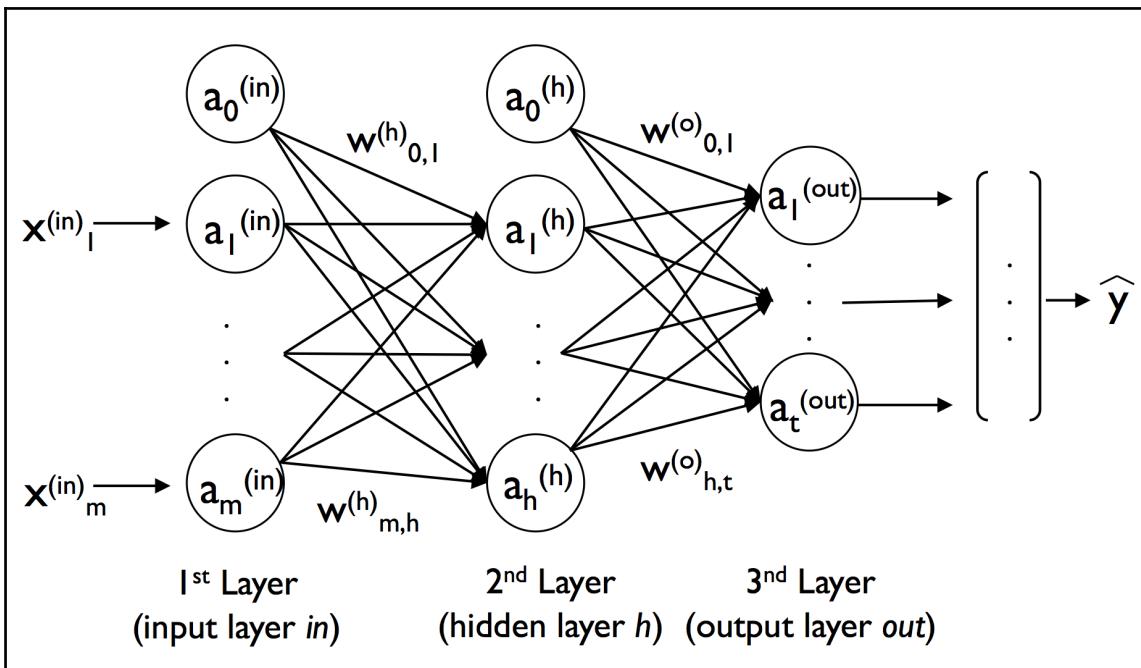
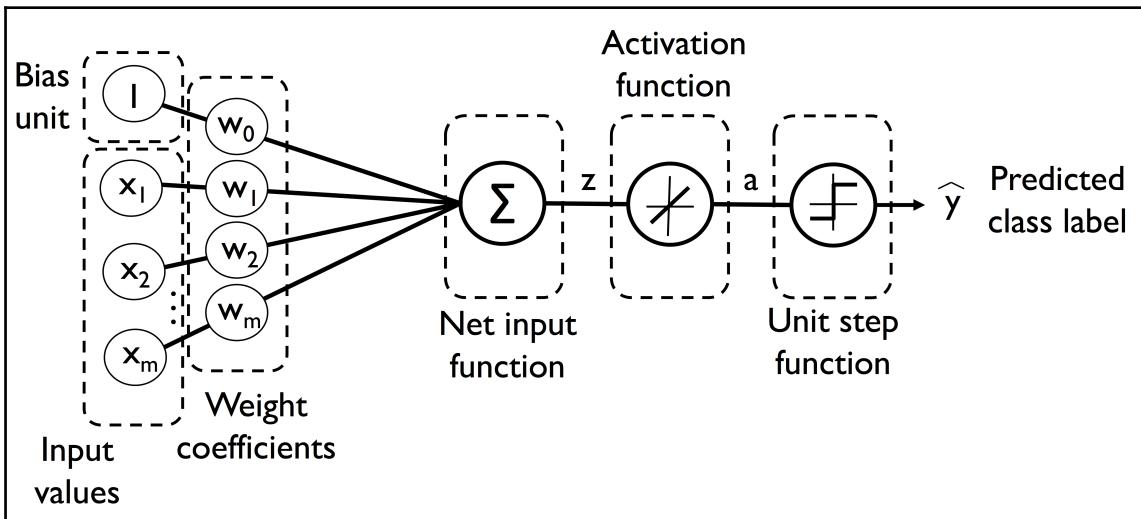




>



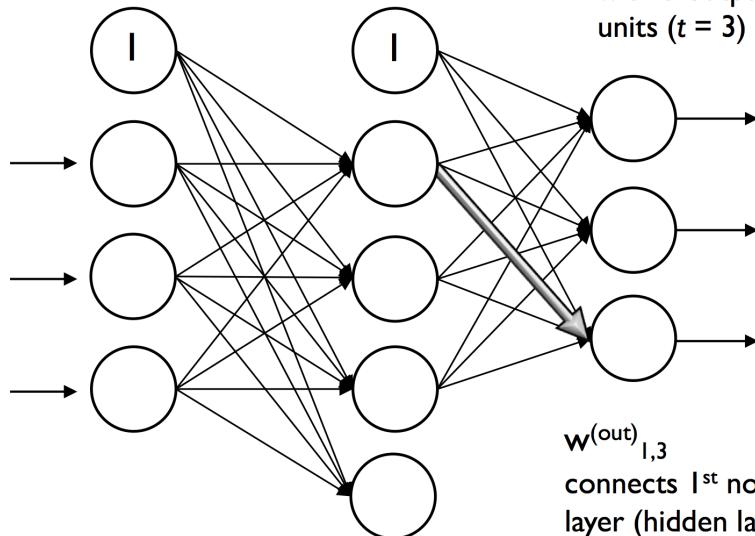
# Chapter 12: Implementing a Multilayer Artificial Neural Network from Scratch



Input layer with 3  
input units plus  
bias unit ( $m = 3+1$ )

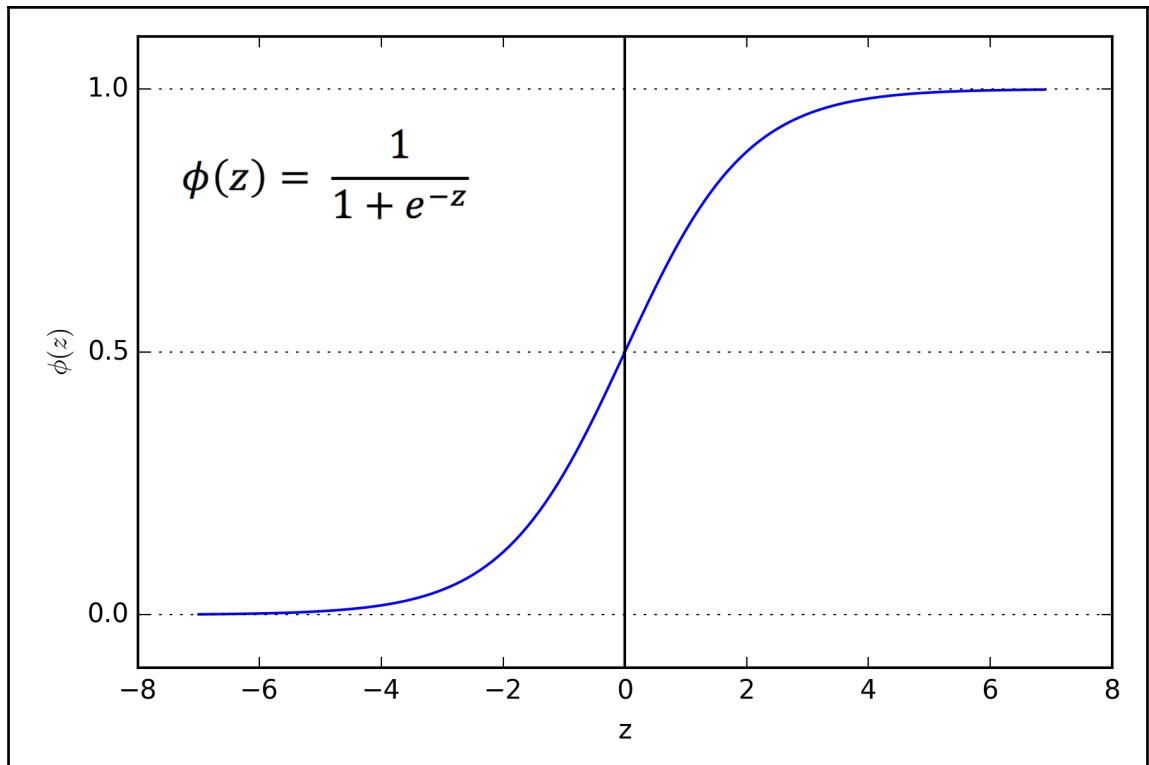
Hidden layer with 4  
hidden units plus bias  
unit ( $d = 4+1$ )

Output layer  
with 3 output  
units ( $t = 3$ )



Number of layers:  $L = 3$

$w^{(out)}_{1,3}$   
connects 1<sup>st</sup> non-bias neuron in the 2<sup>nd</sup>  
layer (hidden layer  $h$ ) to the 3<sup>rd</sup> unit in  
the 3<sup>rd</sup> layer (output layer  $out$ )



0

1

2

3

4

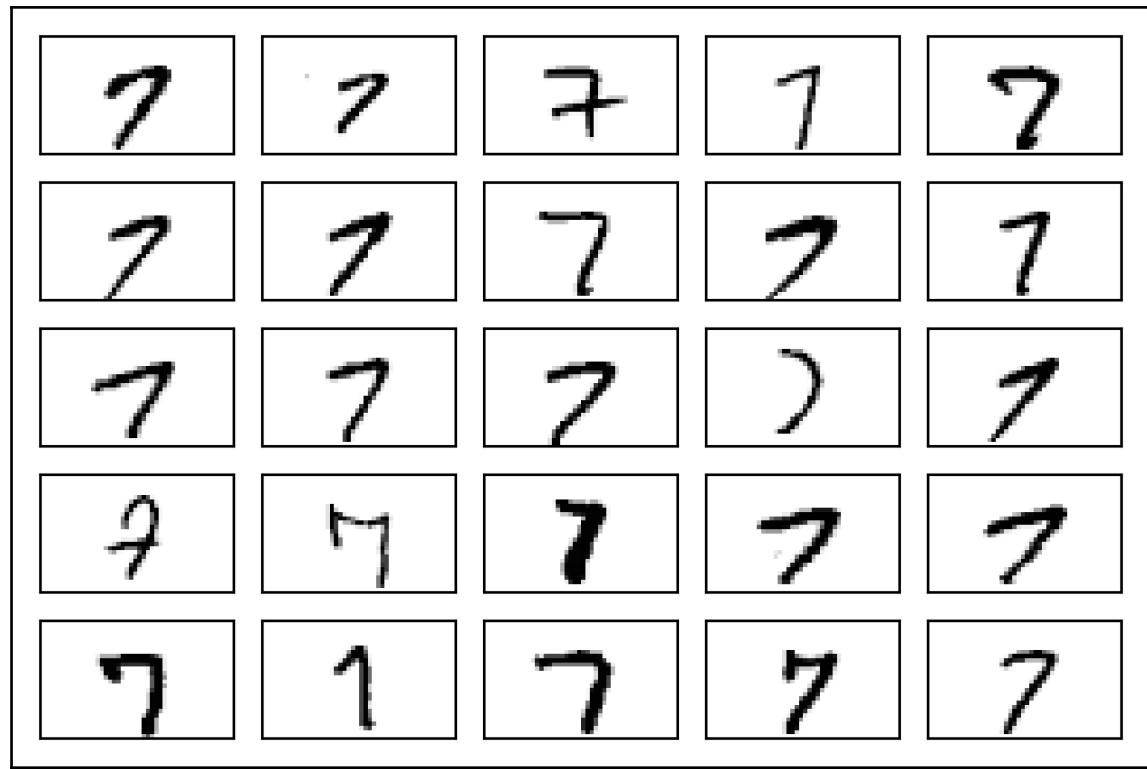
5

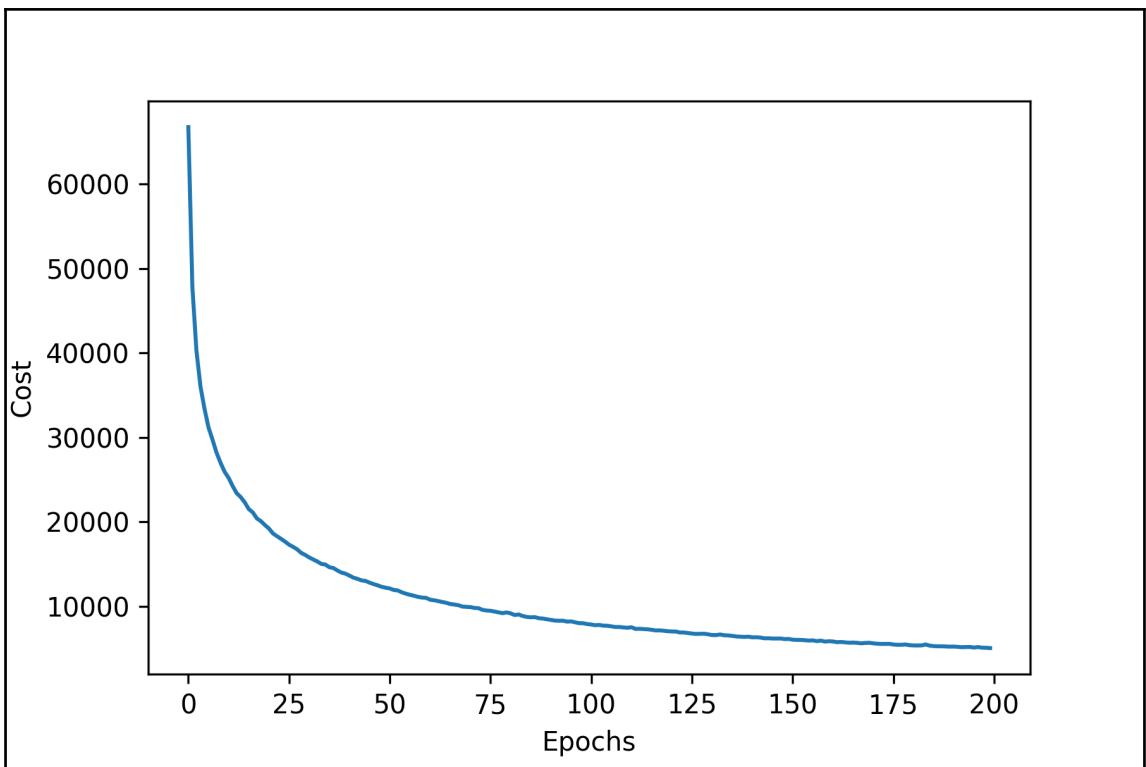
6

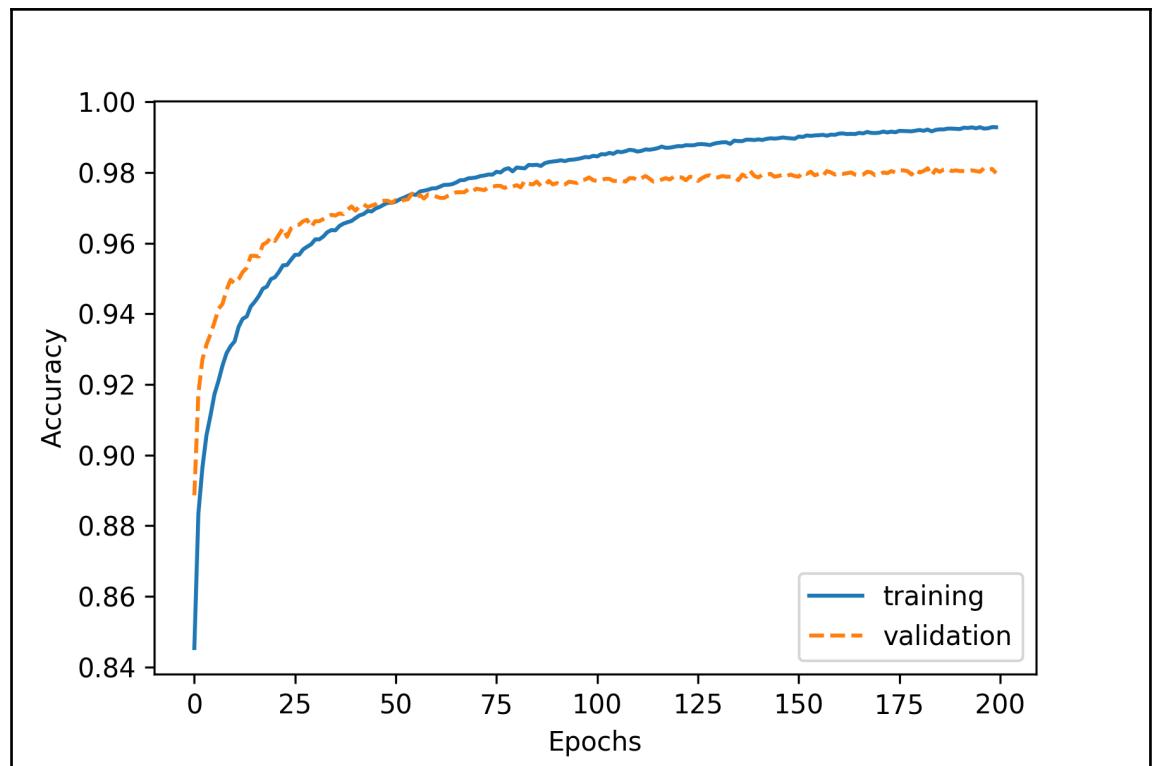
7

8

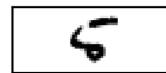
9



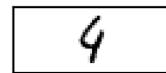




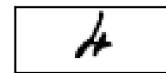
1) t: 5 p: 6



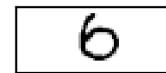
2) t: 4 p: 9



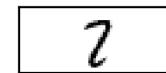
3) t: 4 p: 2



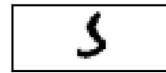
4) t: 6 p: 0



5) t: 2 p: 7



6) t: 5 p: 3



7) t: 3 p: 7



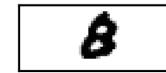
8) t: 6 p: 0



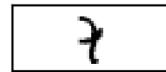
9) t: 3 p: 5



10) t: 8 p: 0



11) t: 7 p: 1



12) t: 3 p: 7



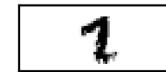
13) t: 1 p: 8



14) t: 2 p: 6



15) t: 2 p: 8



16) t: 7 p: 3



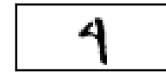
17) t: 8 p: 4



18) t: 5 p: 8



19) t: 4 p: 9



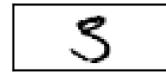
20) t: 9 p: 7



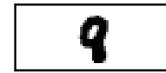
21) t: 2 p: 7



22) t: 3 p: 5



23) t: 8 p: 9

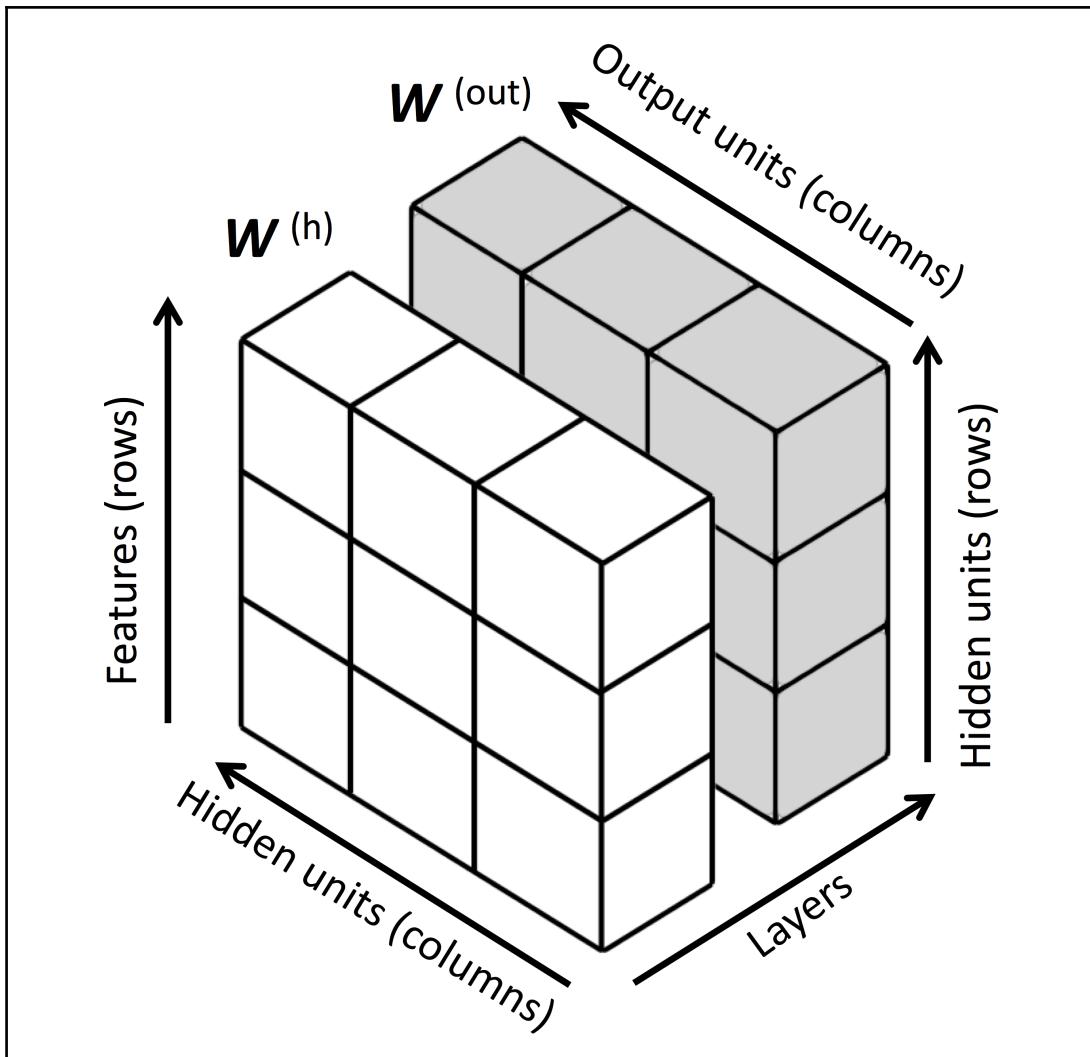


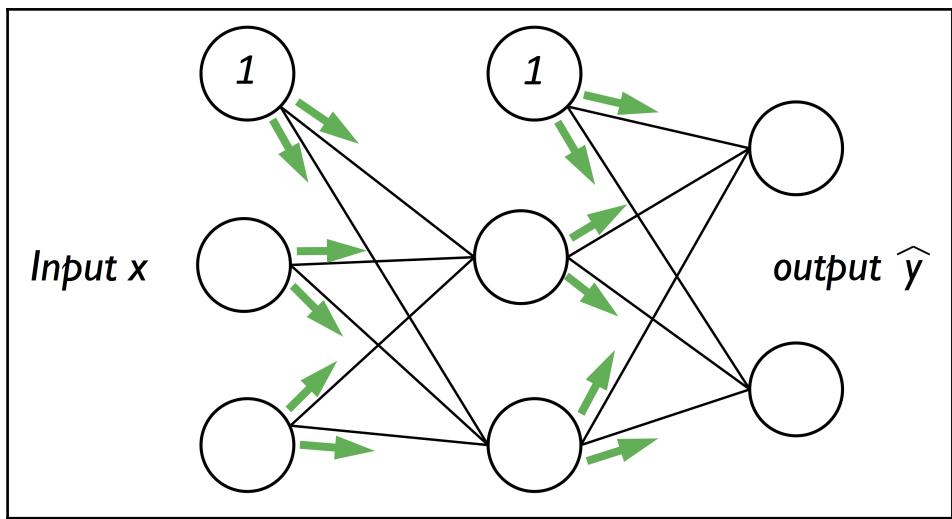
24) t: 5 p: 4



25) t: 1 p: 2



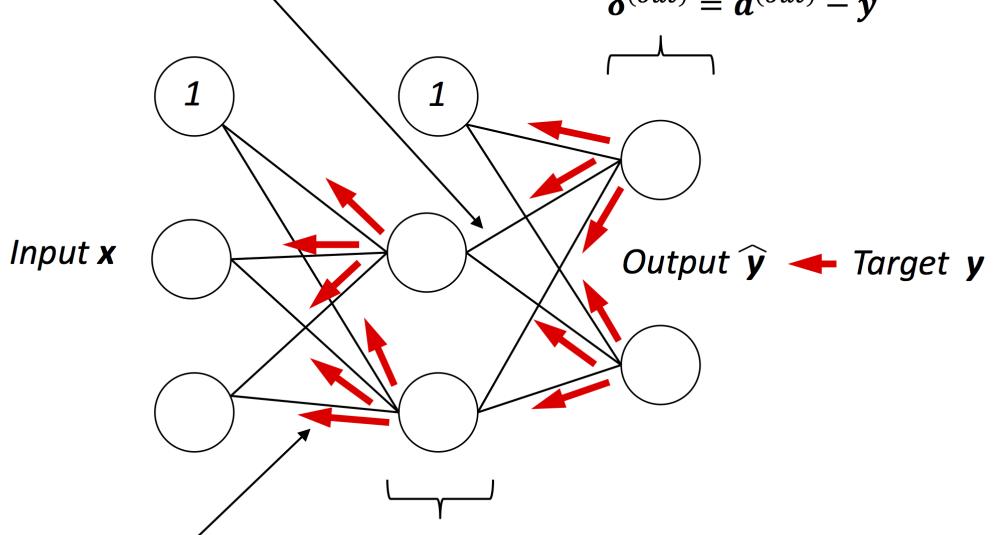




Compute the gradient:

$$\frac{\partial}{\partial w_{i,j}^{(out)}} J(\mathbf{W}) = a_j^{(h)} \delta_i^{(out)}$$

Error term of the output layer:  
 $\delta^{(out)} = \mathbf{a}^{(out)} - \mathbf{y}$

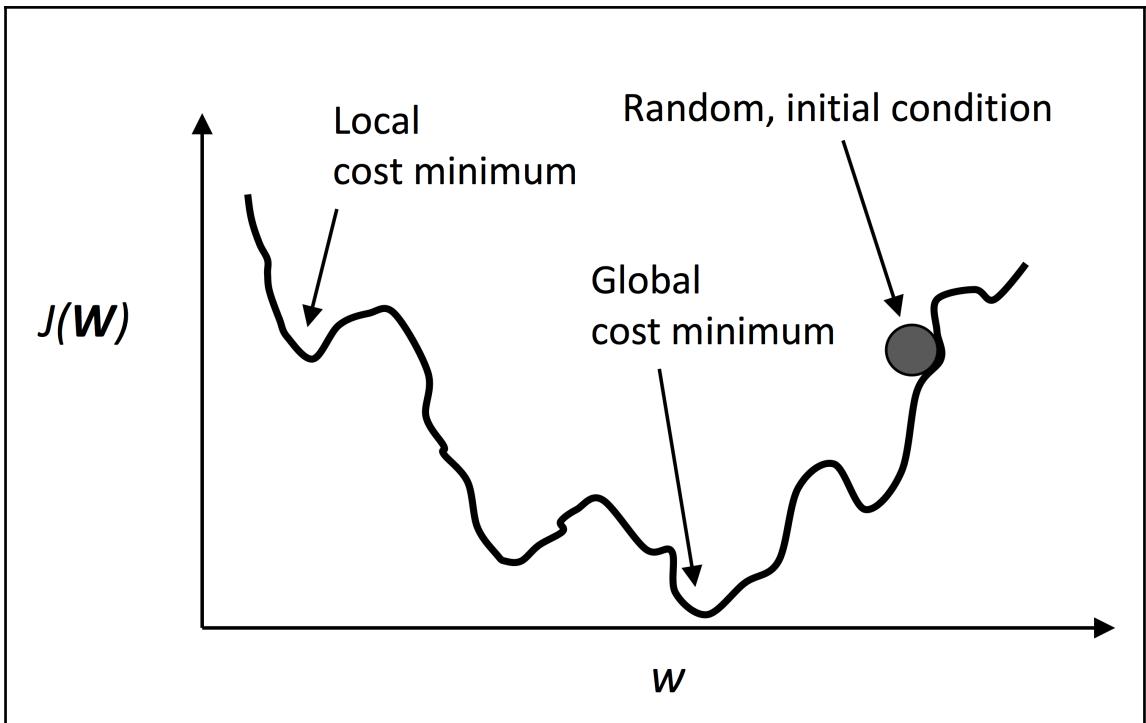


Compute the gradient:

$$\frac{\partial}{\partial w_{i,j}^{(h)}} J(\mathbf{W}) = a_j^{(in)} \delta_i^{(h)}$$

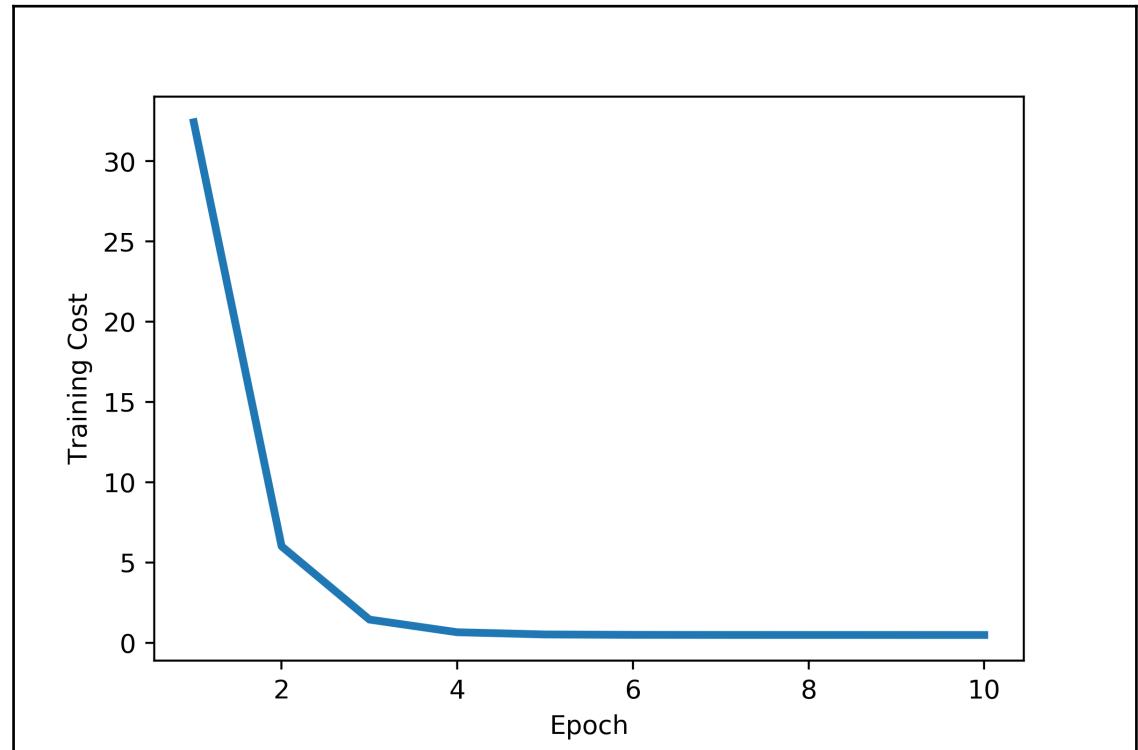
Error term of the hidden layer:

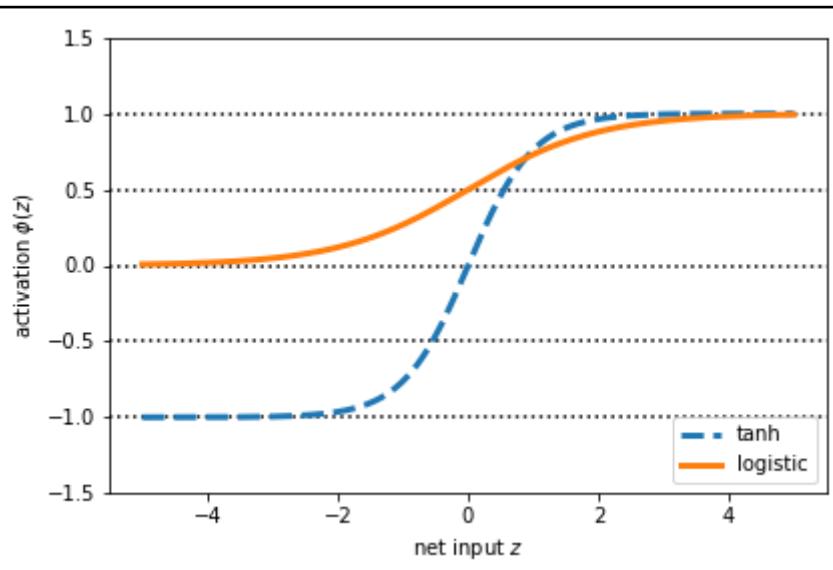
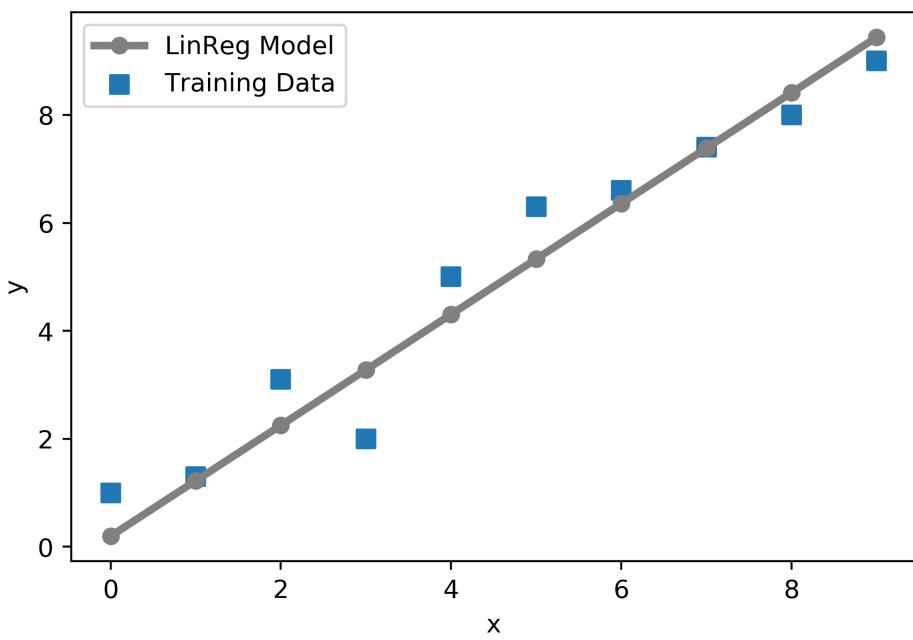
$$\delta^{(h)} = \delta^{(out)} (\mathbf{W}^{(out)})^T \odot \frac{\partial \phi(z^{(h)})}{\partial z^{(h)}}$$

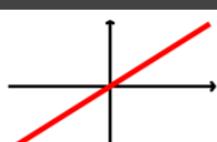
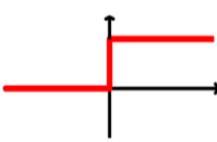
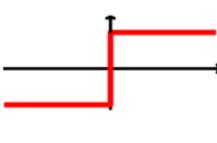
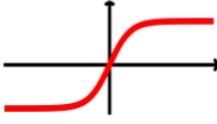
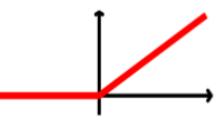


# Chapter 13: Parallelizing Neural Network Training with TensorFlow

Specifications	Intel® Core™ i7-6900K Processor Extreme Ed.	NVIDIA GeForce® GTX™ 1080 Ti
Base Clock Frequency	3.2 GHz	< 1.5 GHz
Cores	8	3584
Memory Bandwidth	64 GB/s	484 GB/s
Floating-Point Calculations	409 GFLOPS	11300 GFLOPS
Cost	~ \$1000.00	~ \$700.00



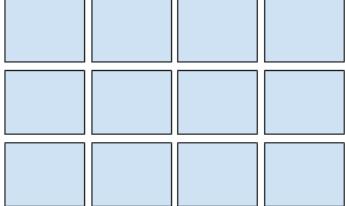


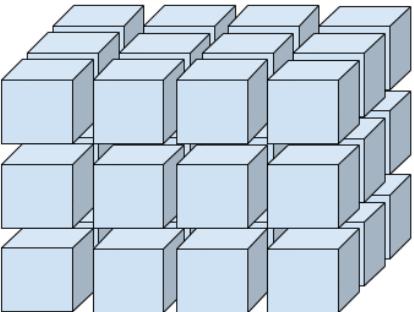
Activation Function	Equation	Example	1D Graph
Linear	$\phi(z) = z$	Adaline, linear regression	
Unit Step (Heaviside Function)	$\phi(z) = \begin{cases} 0 & z < 0 \\ 0.5 & z = 0 \\ 1 & z > 0 \end{cases}$	Perceptron variant	
Sign (signum)	$\phi(z) = \begin{cases} -1 & z < 0 \\ 0 & z = 0 \\ 1 & z > 0 \end{cases}$	Perceptron variant	
Piece-wise Linear	$\phi(z) = \begin{cases} 0 & z \leq -\frac{1}{2} \\ z + \frac{1}{2} & -\frac{1}{2} \leq z \leq \frac{1}{2} \\ 1 & z \geq \frac{1}{2} \end{cases}$	Support vector machine	
Logistic (sigmoid)	$\phi(z) = \frac{1}{1 + e^{-z}}$	Logistic regression, Multilayer NN	
Hyperbolic Tangent (tanh)	$\phi(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	Multilayer NN, RNNs	
ReLU	$\phi(z) = \begin{cases} 0 & z < 0 \\ z & z > 0 \end{cases}$	Multilayer NN, CNNs	

# Chapter 14: Going Deeper – The Mechanics of TensorFlow

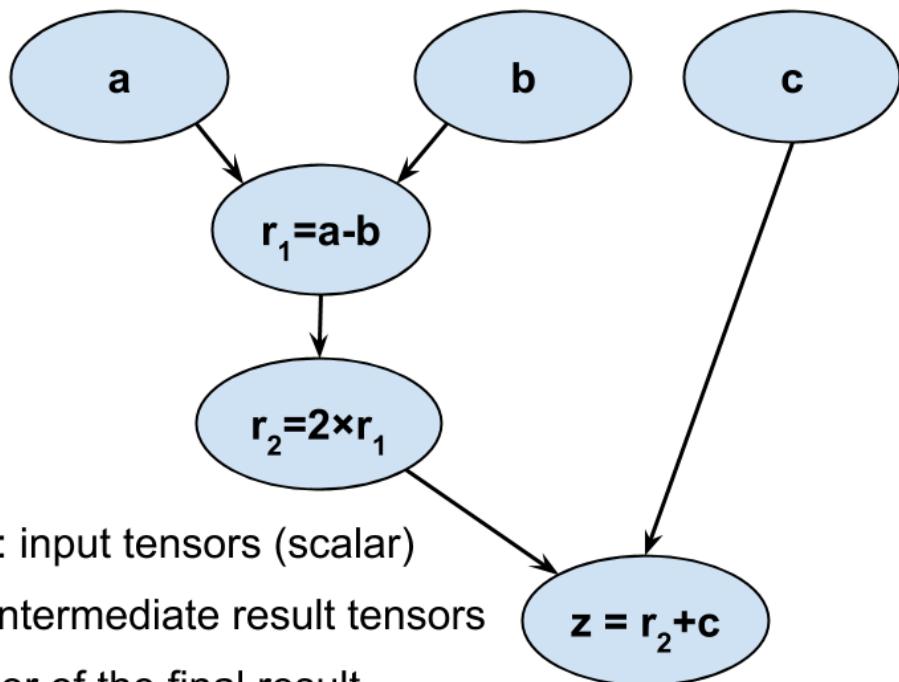
Rank 0:   
(scalar)

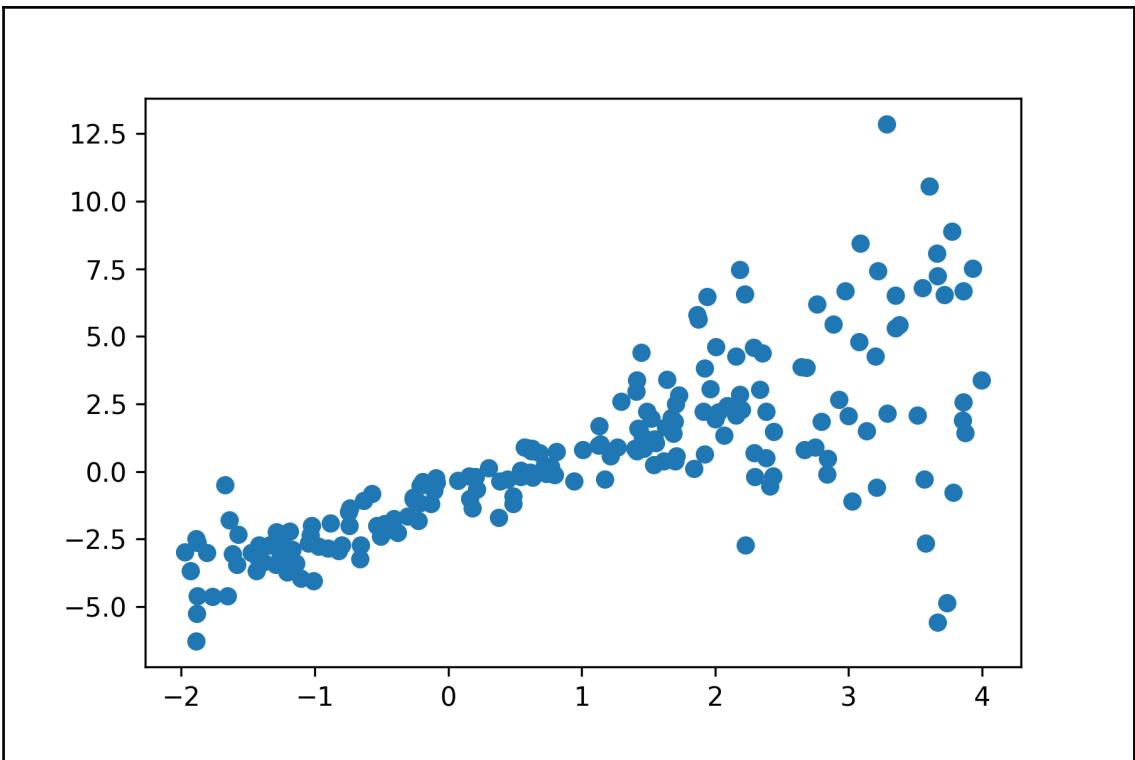
Rank 1:   
(vector)

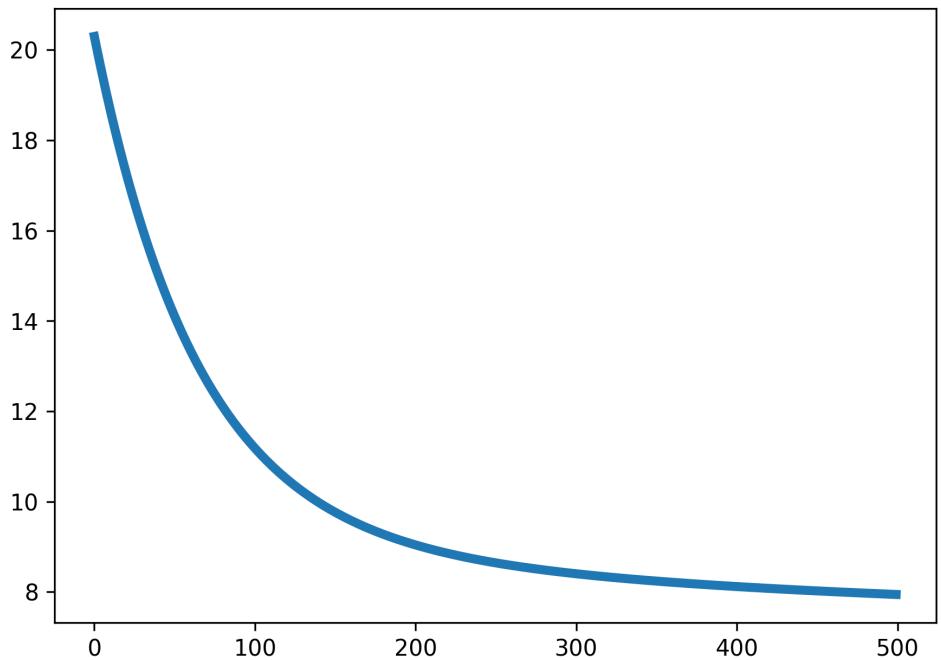
Rank 2: (matrix)  


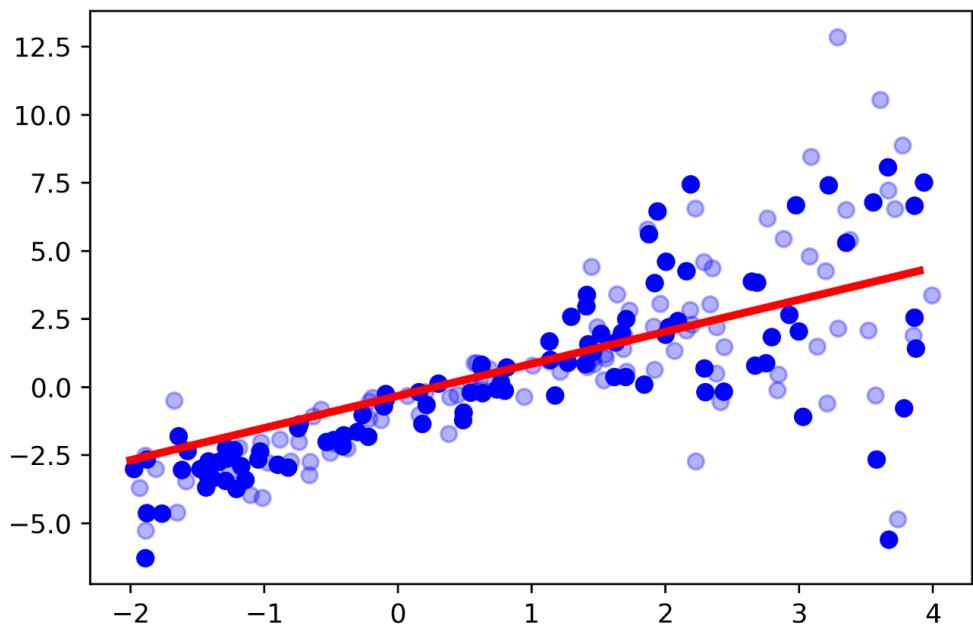
Rank 3: 

## Computation graph implementing the equation $z = 2 \times (a - b) + c$

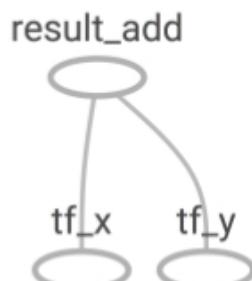




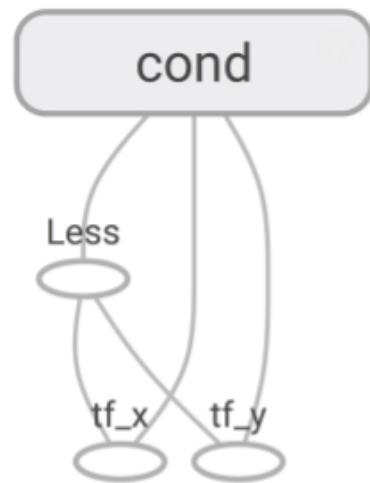




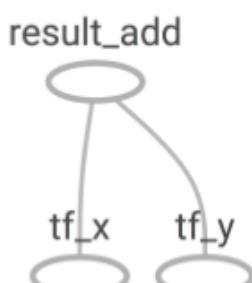
Python if



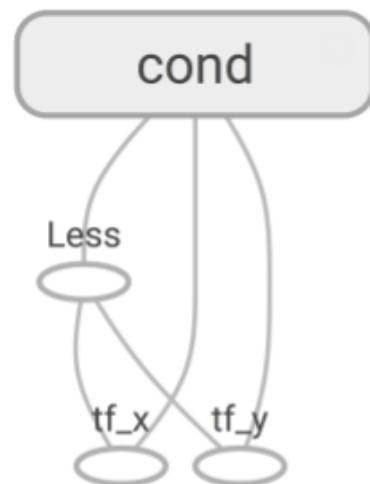
`tf.cond(...)`

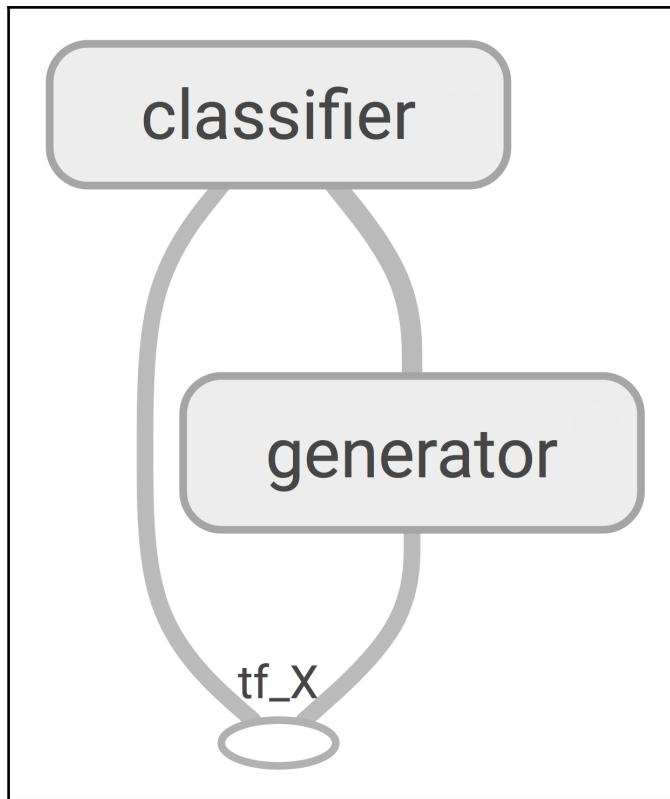


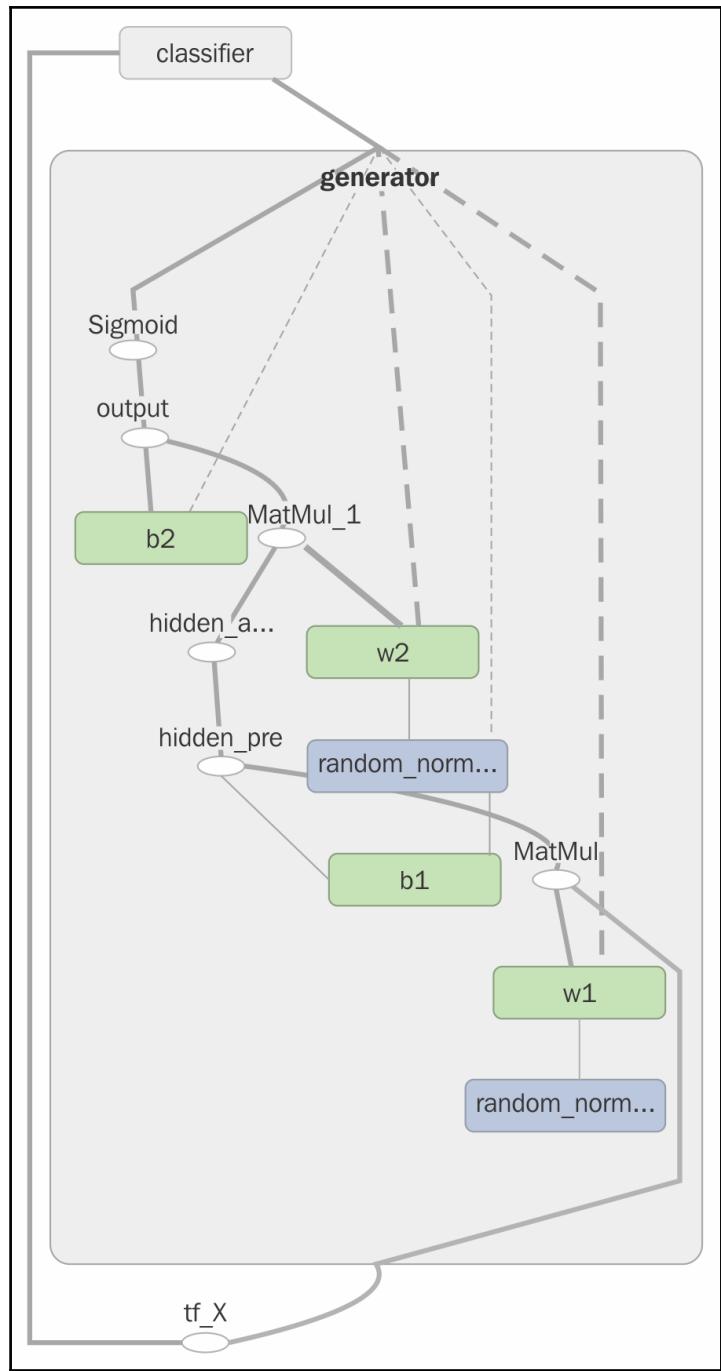
Python if

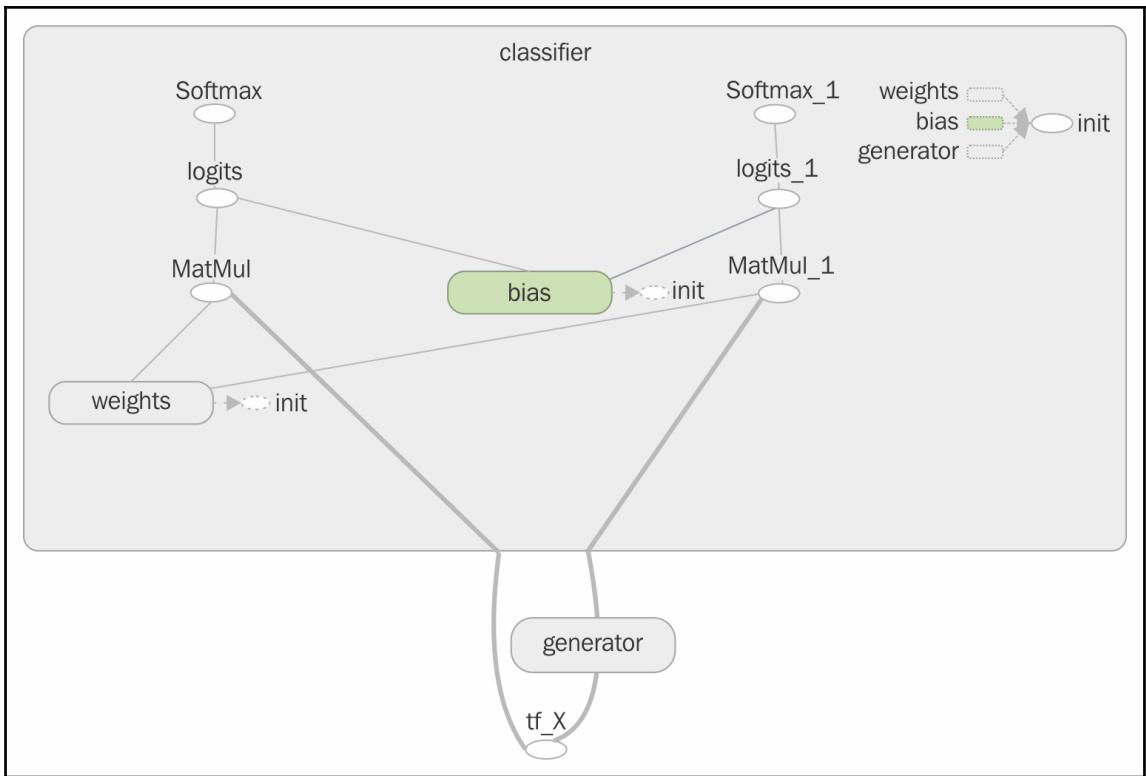


`tf.cond(...)`

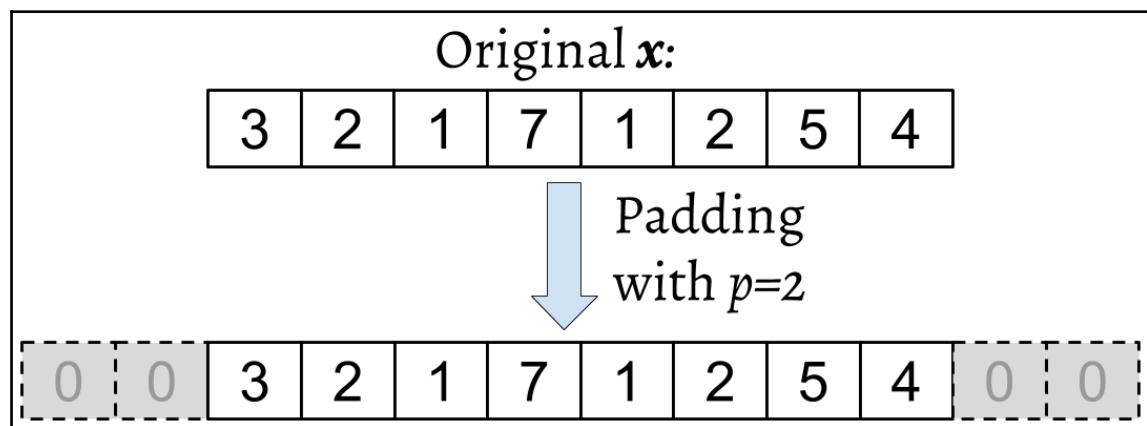
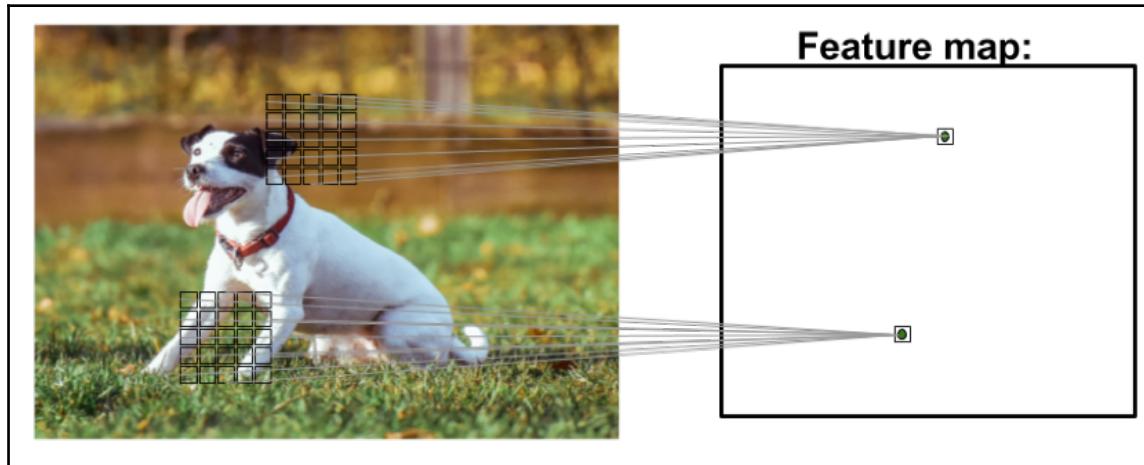


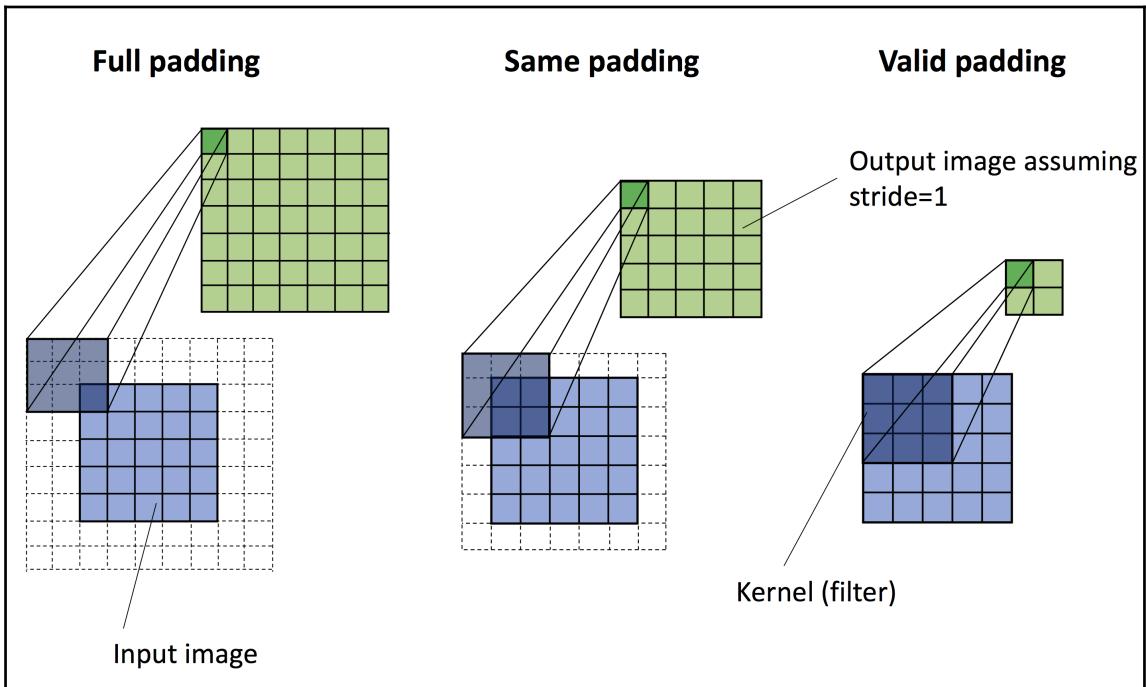
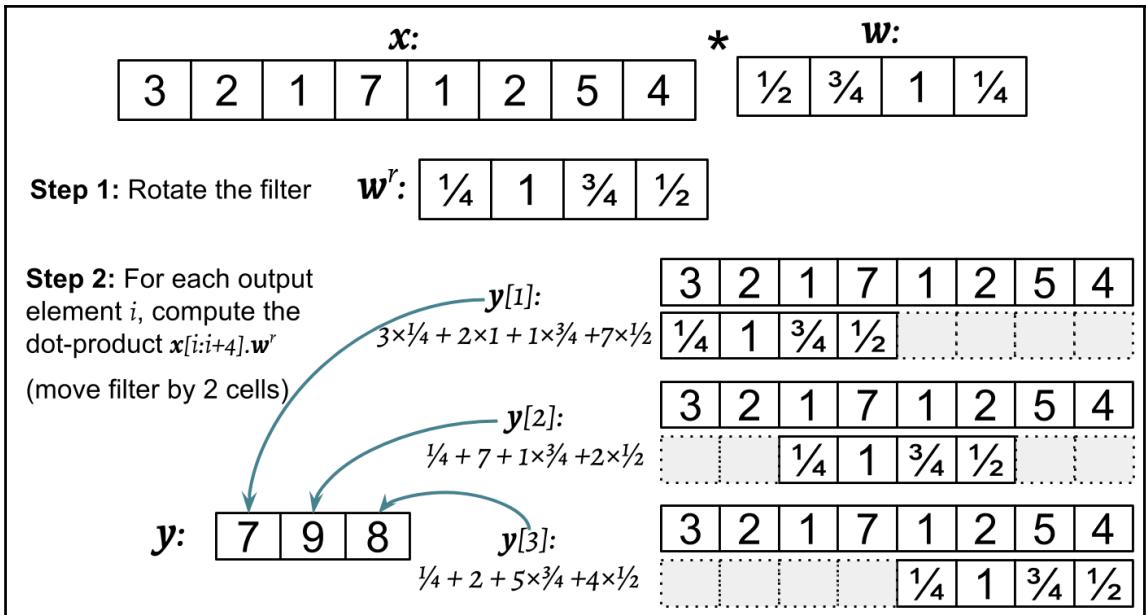


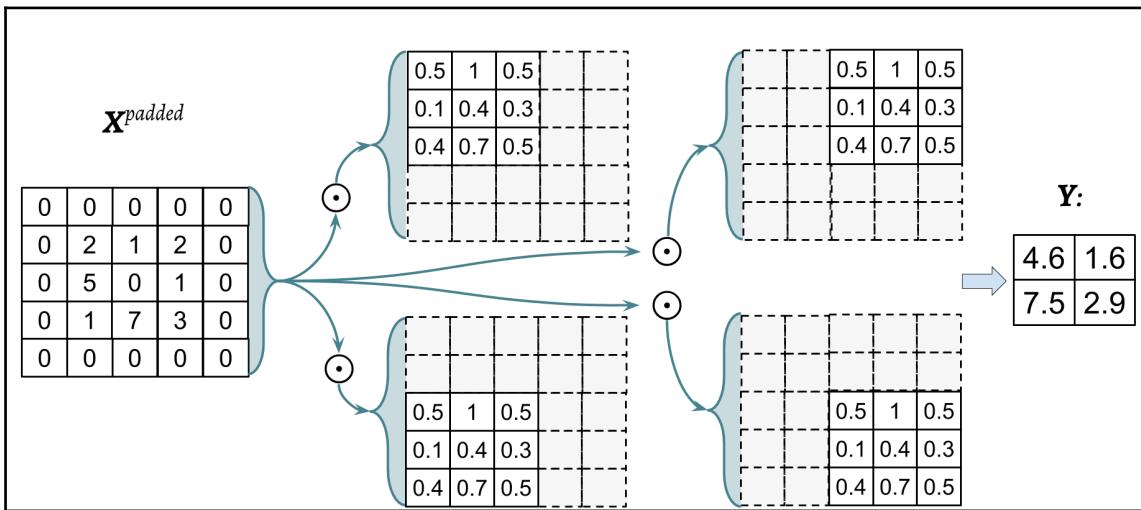
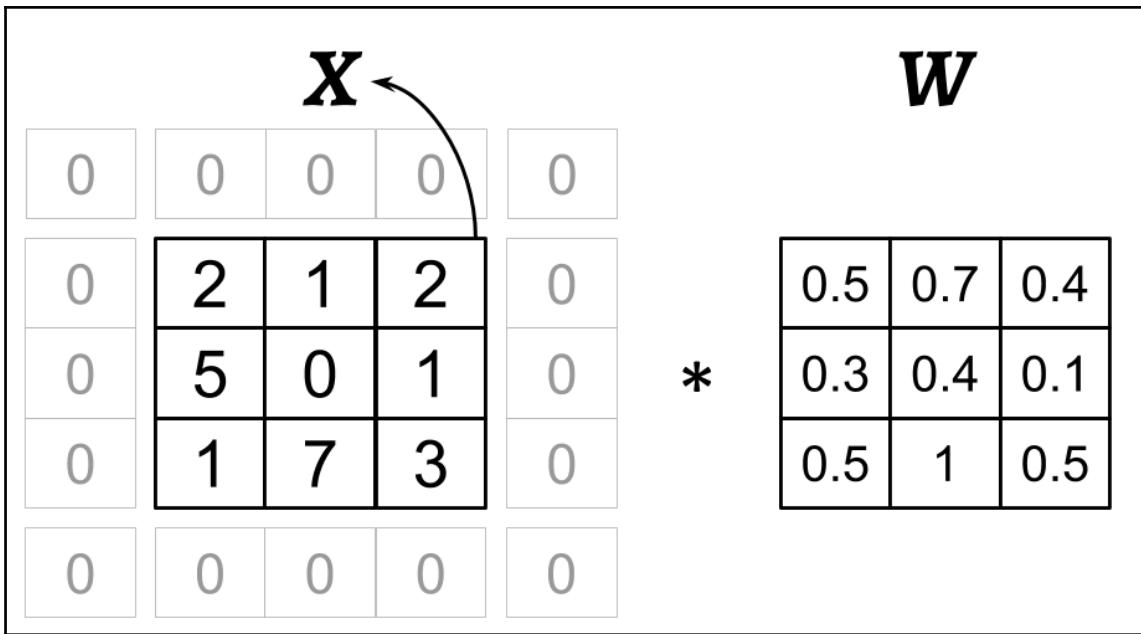


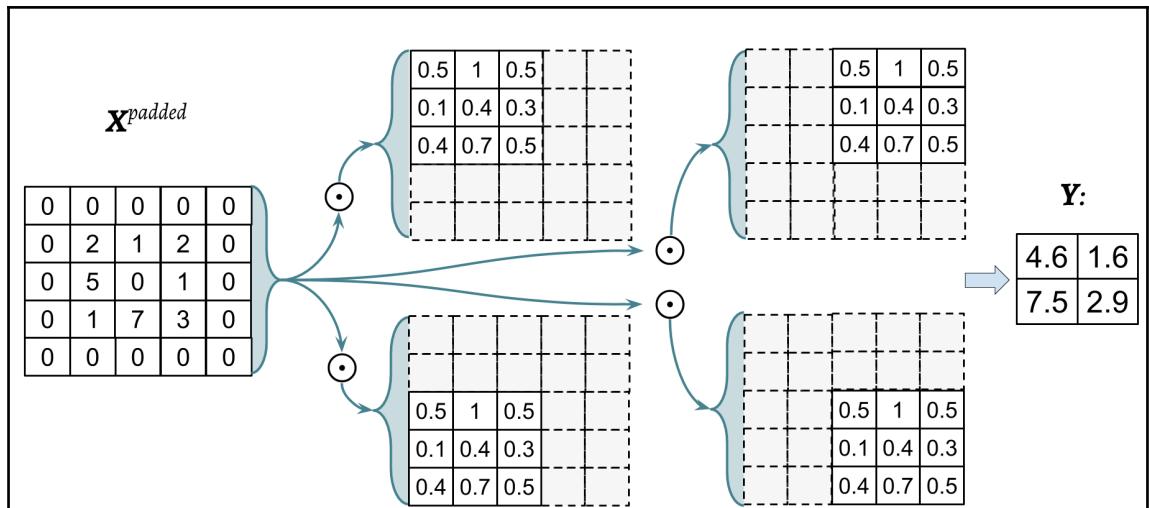


# Chapter 15: Classifying Images with Deep Convolutional Neural Networks





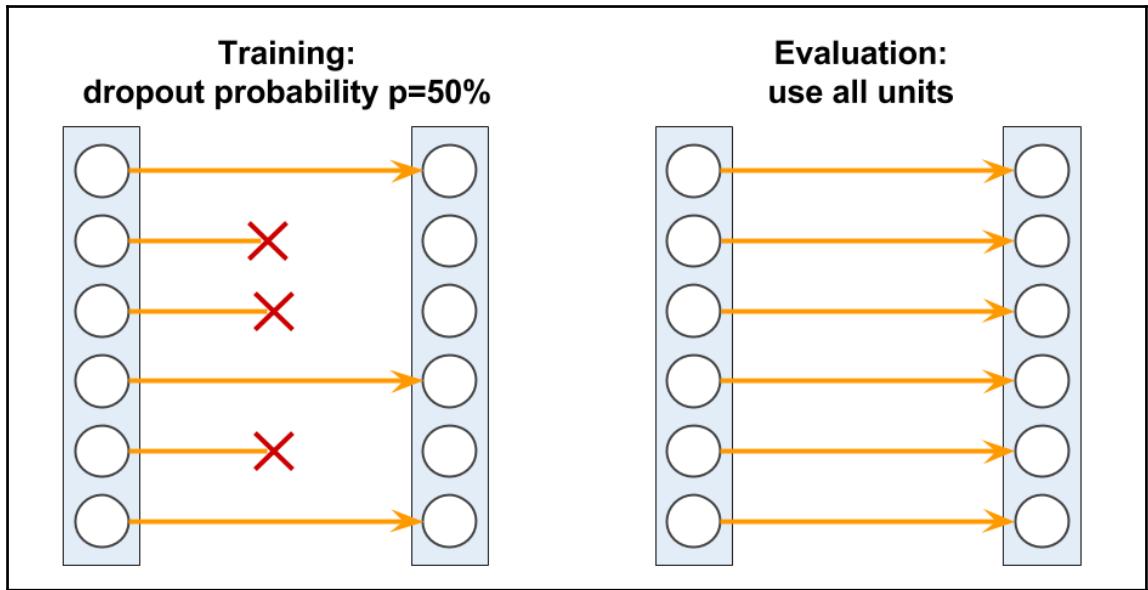
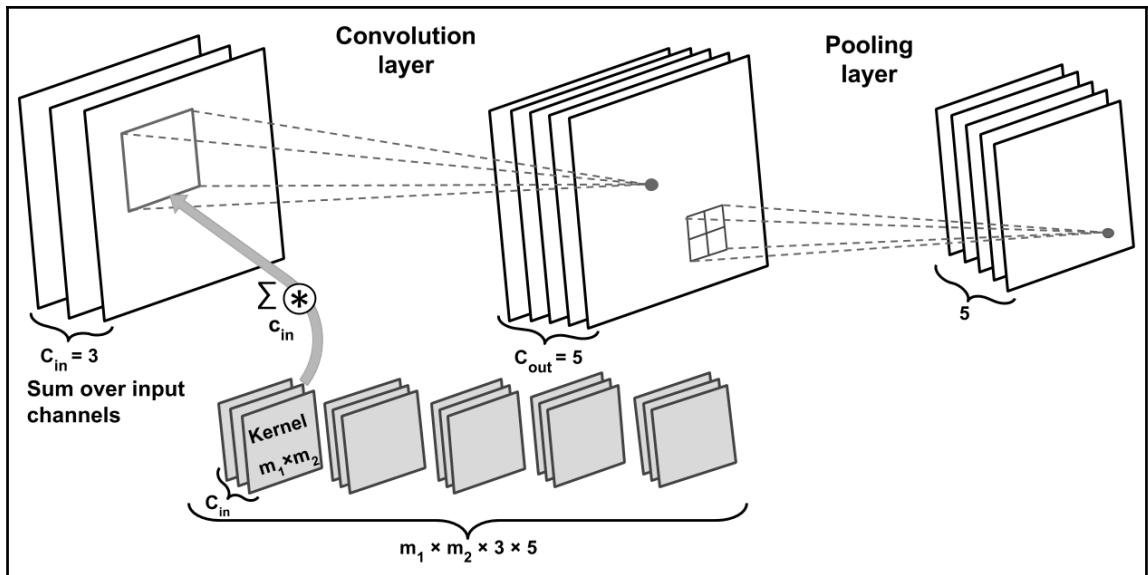


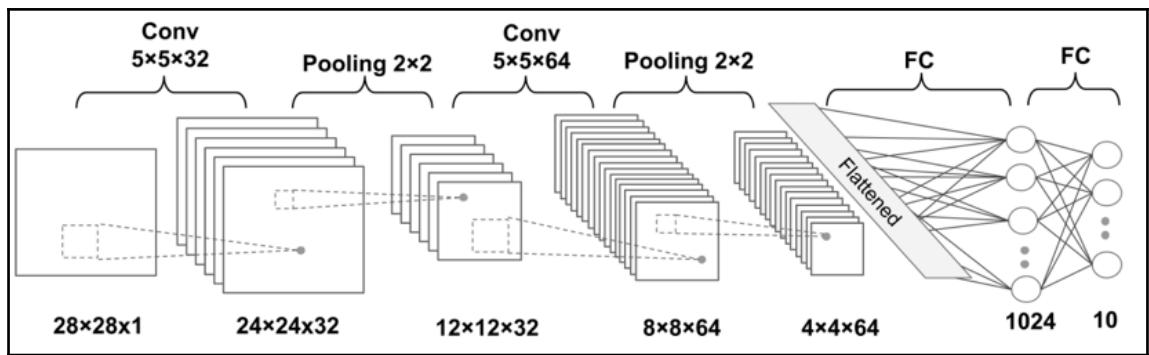


$$X_1 = \begin{bmatrix} 10 & 255 & 125 & 0 & 170 & 100 \\ 70 & 255 & 105 & 25 & 25 & 70 \\ 255 & 0 & 150 & 0 & 10 & 10 \\ 0 & 255 & 10 & 10 & 150 & 20 \\ 70 & 15 & 200 & 100 & 95 & 0 \\ 35 & 25 & 100 & 20 & 0 & 60 \end{bmatrix}$$

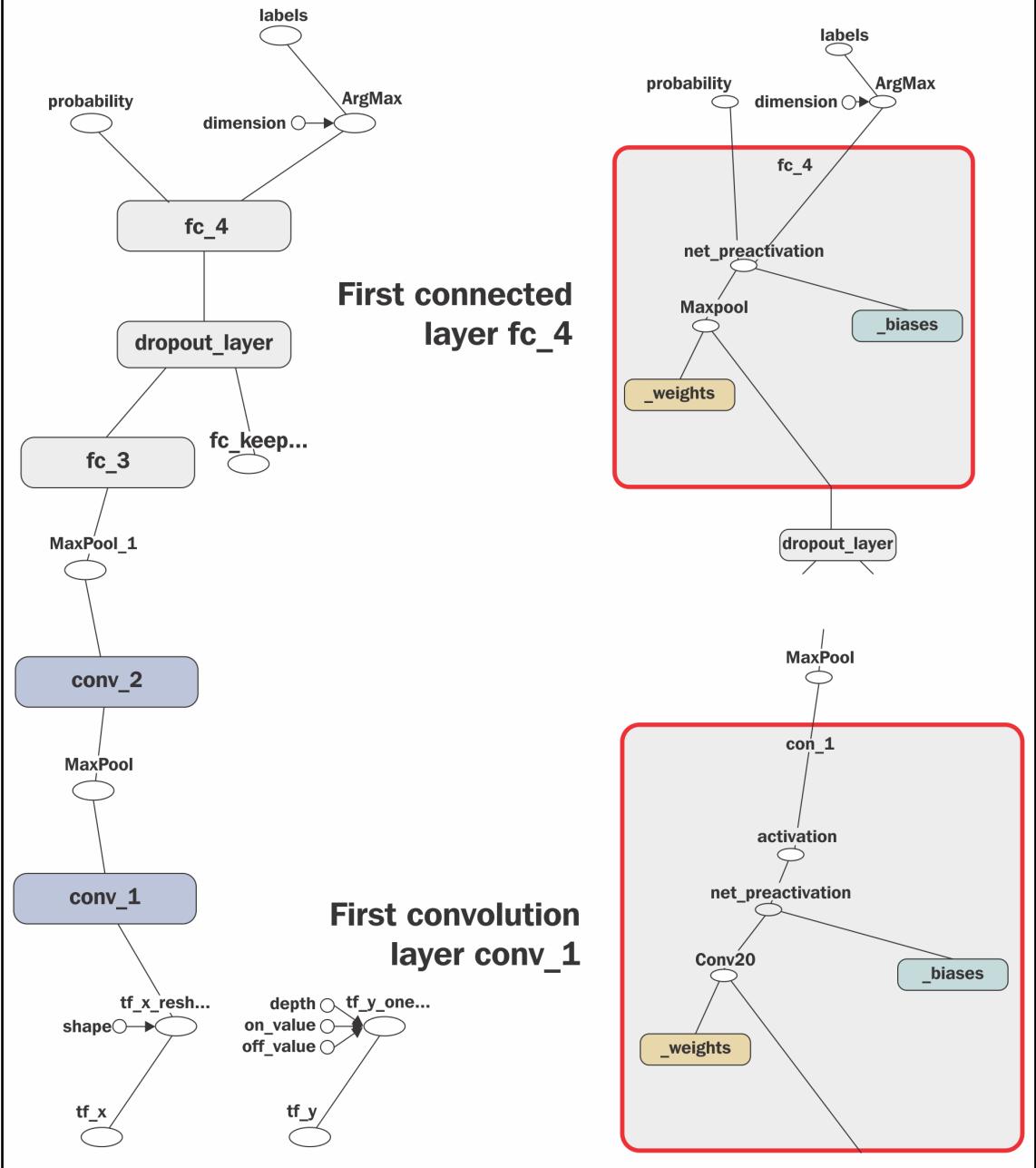
$$X_2 = \begin{bmatrix} 100 & 100 & 100 & 50 & 100 & 50 \\ 95 & 255 & 100 & 125 & 125 & 170 \\ 80 & 40 & 10 & 10 & 125 & 150 \\ 255 & 30 & 150 & 20 & 120 & 125 \\ 30 & 30 & 150 & 100 & 70 & 70 \\ 70 & 30 & 100 & 200 & 70 & 95 \end{bmatrix}$$

max-pooling  $P_{2 \times 2}$   $\rightarrow$   $\begin{bmatrix} 255 & 125 & 170 \\ 255 & 150 & 150 \\ 70 & 200 & 95 \end{bmatrix}$

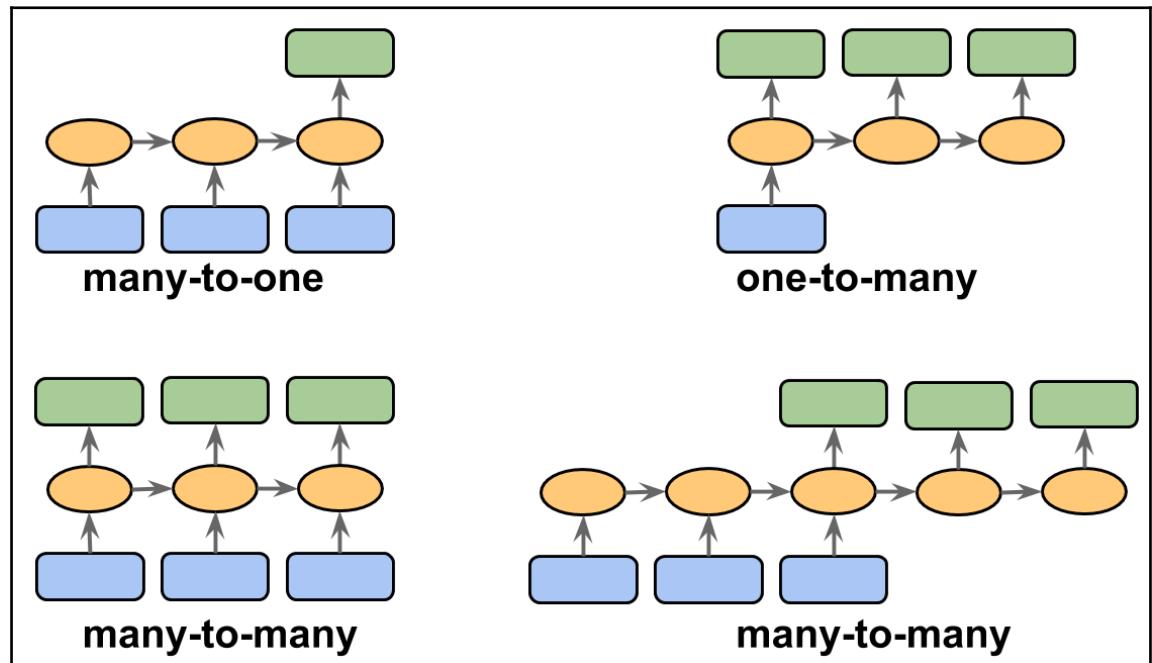
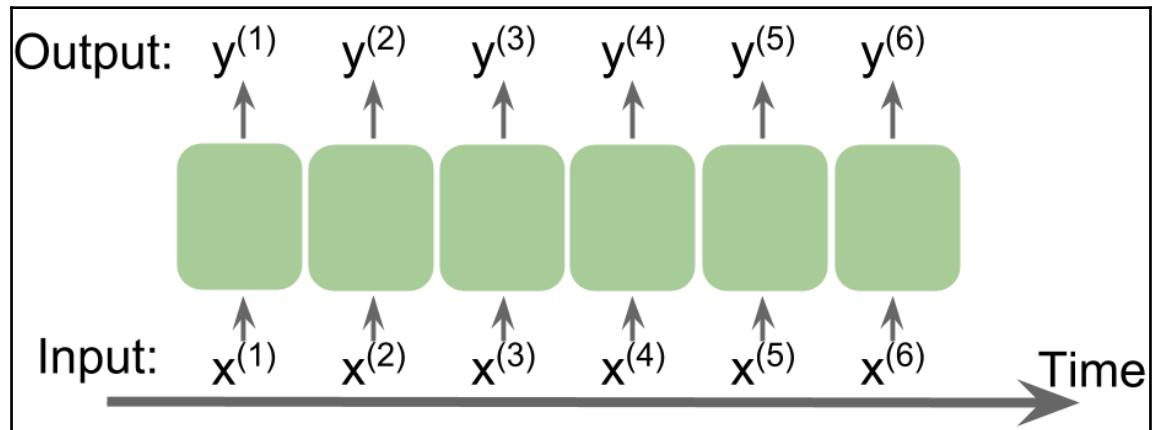




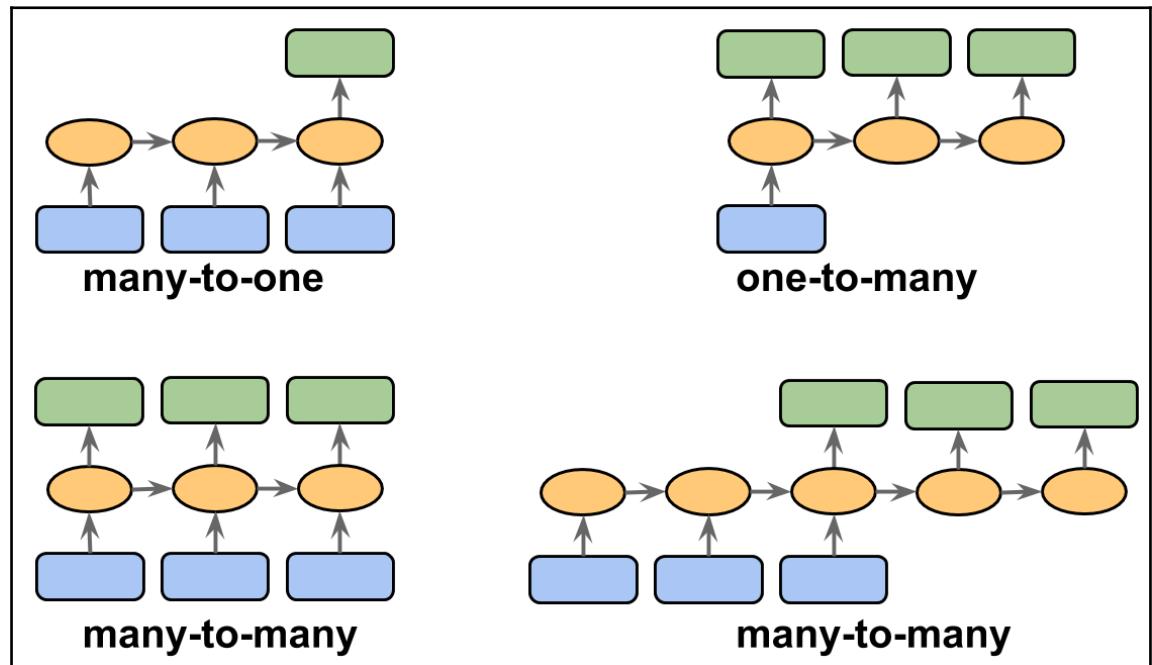
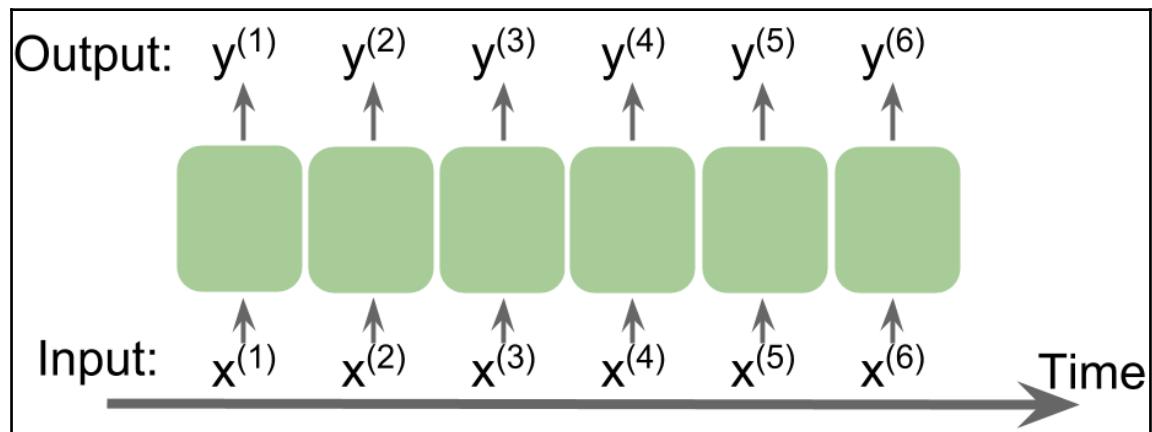
## Graph of the multilayer CNN

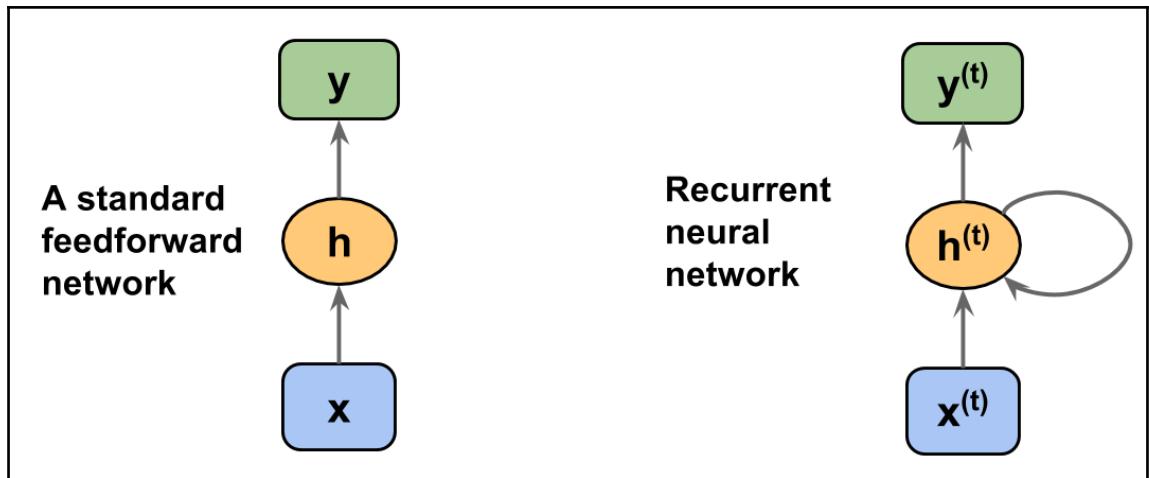


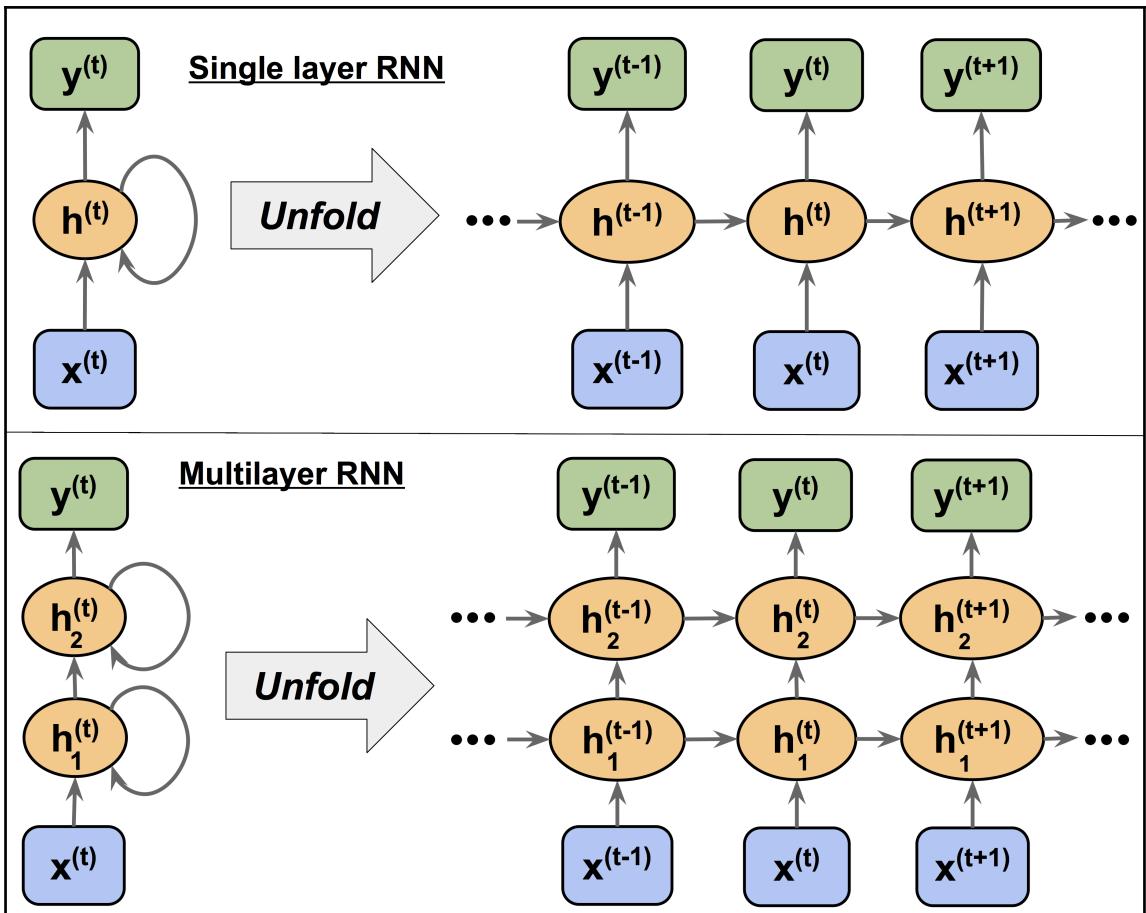
# Chapter 16: Modeling Sequential Data Using Recurrent Neural Networks

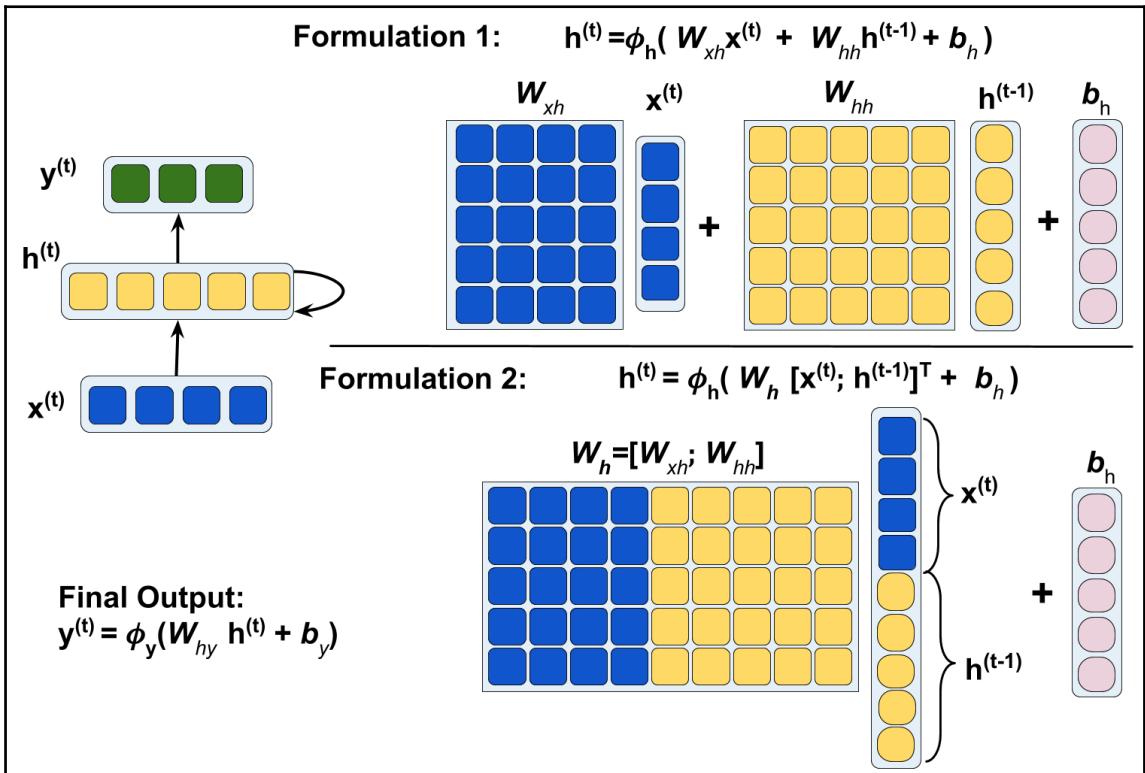
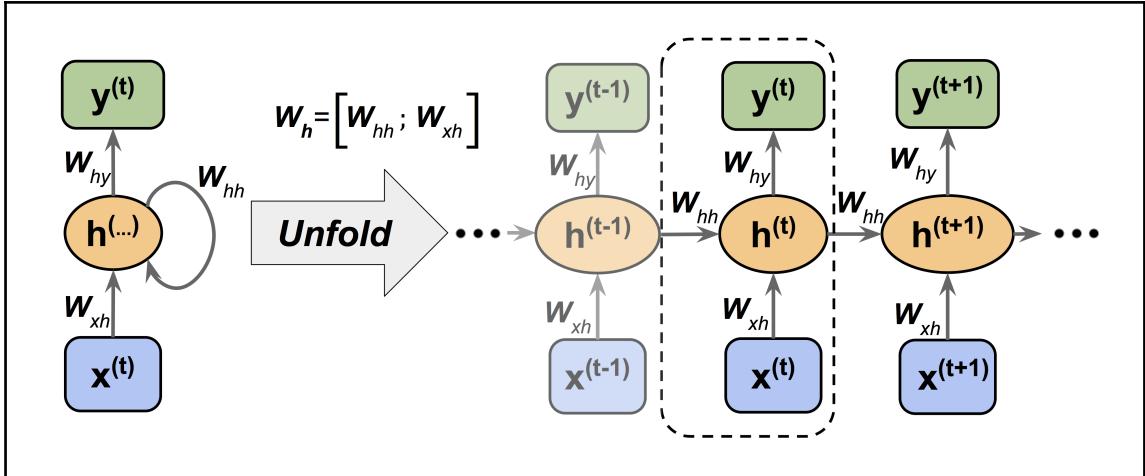


# Chapter 16: Modeling Sequential Data Using Recurrent Neural Networks

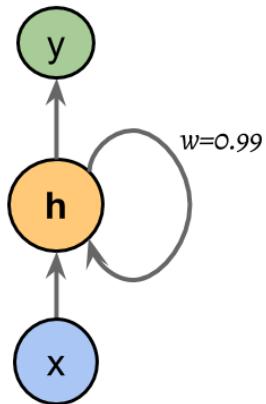




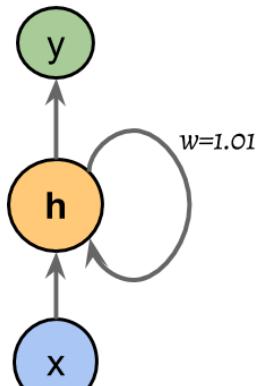




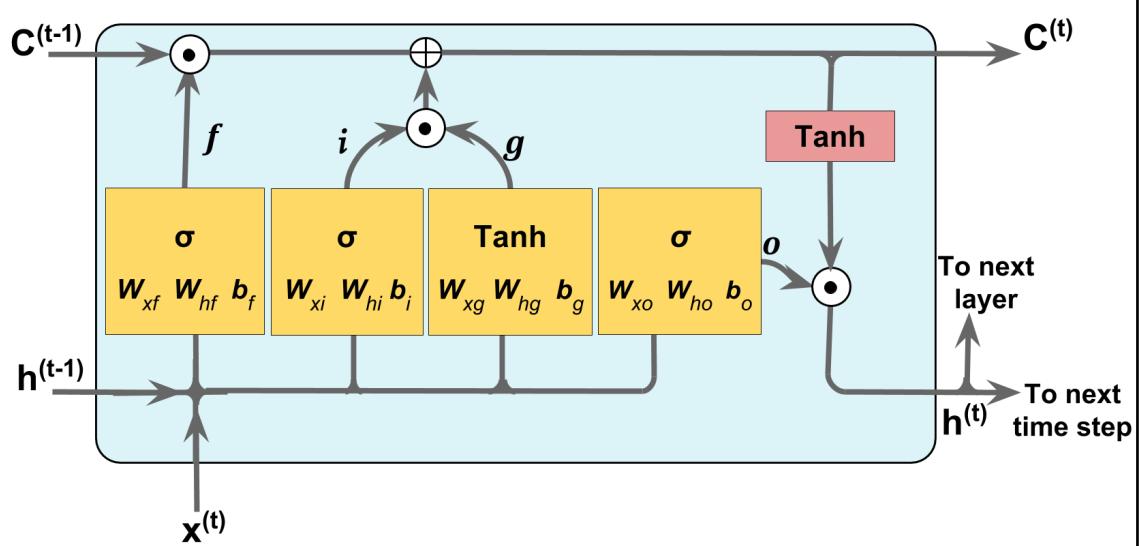
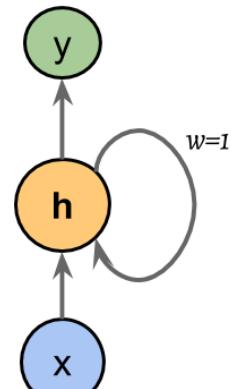
**Vanishing gradient:**  $|w_{hh}| < 1$

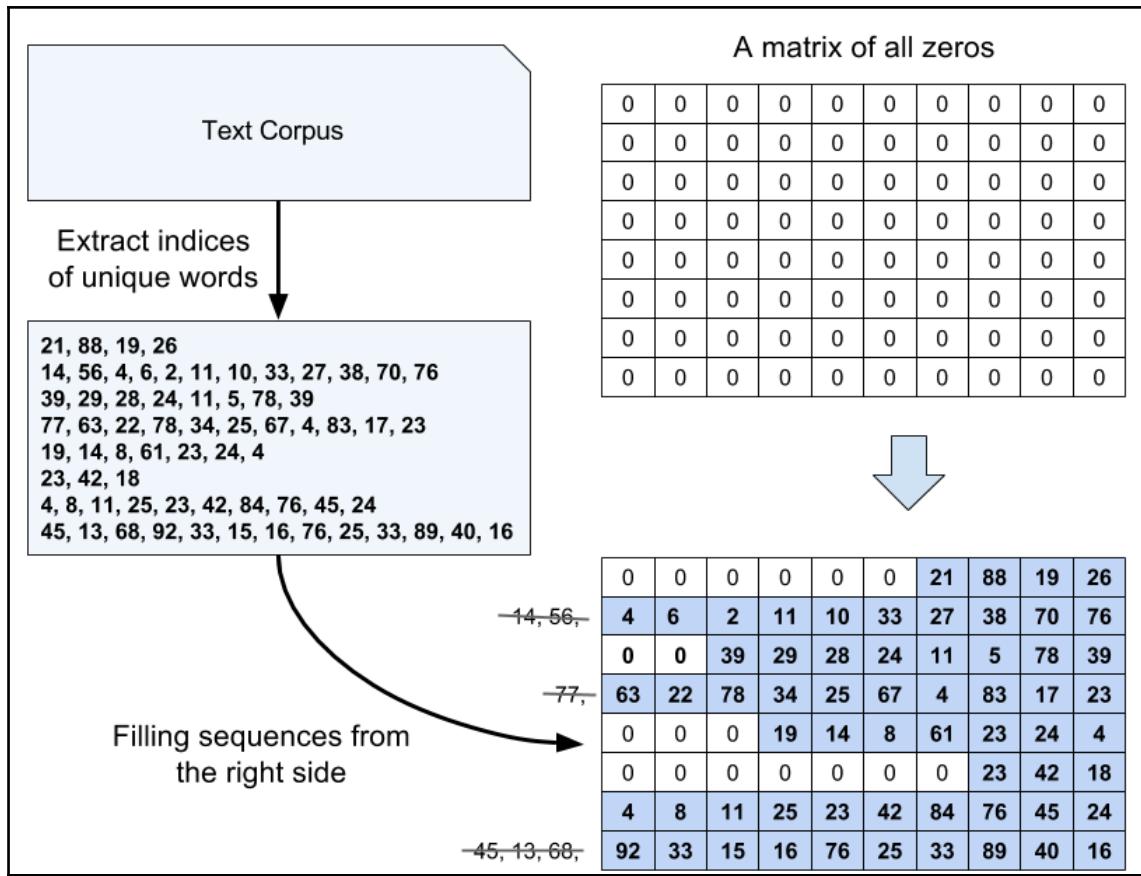


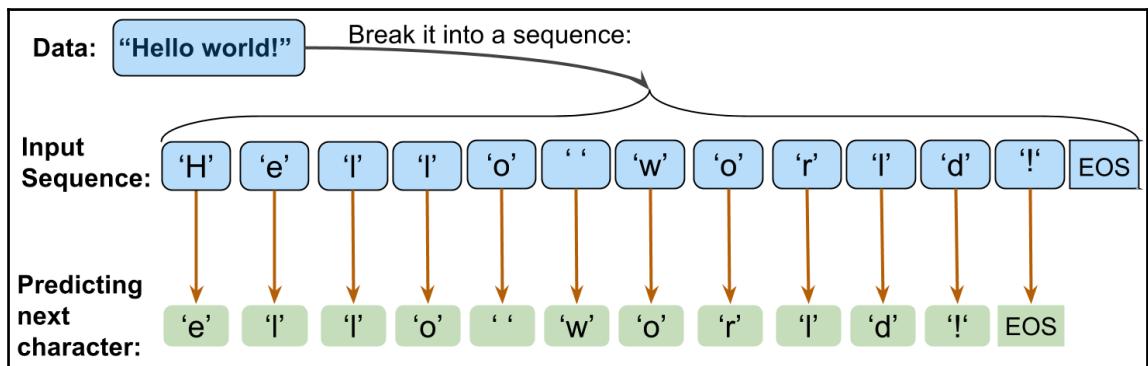
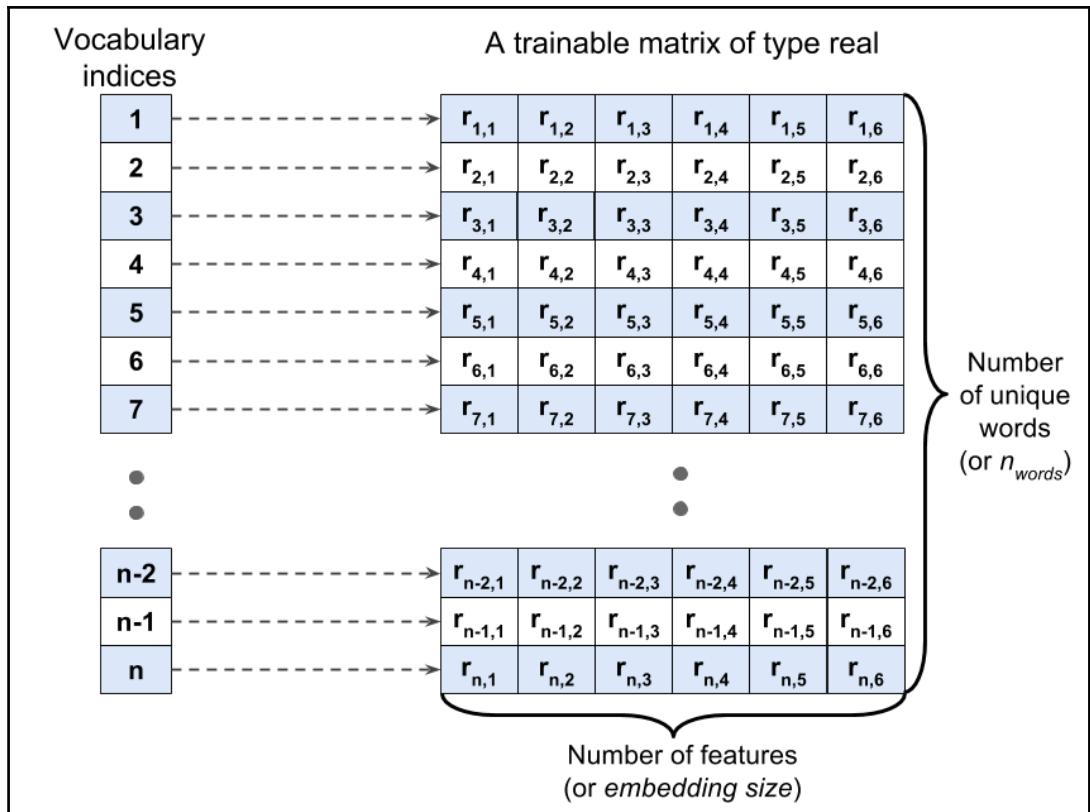
**Exploding gradient:**  $|w_{hh}| > 1$



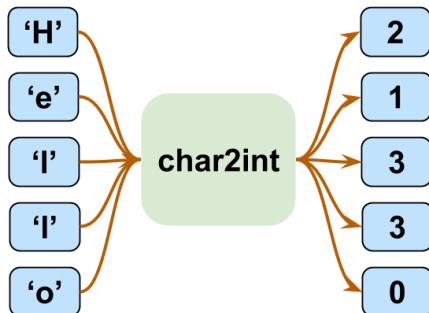
**Desirable:**  $|w_{hh}| = 1$



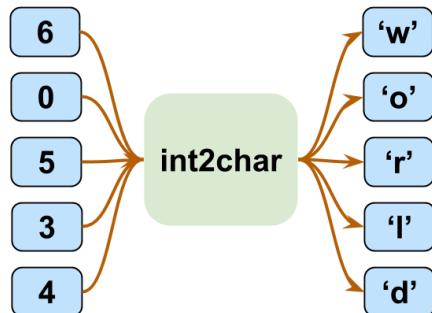




Mapping characters to integers



Mapping integers to characters



### Text Corpus

Convert text into a long sequence of integers

49, 29, 29, 29, 5, 19, 27, 0, 7, 3, 36, 65, 27,  
41, 31, 0, 26, 4, 31, 27, 86, 10, 27, 3, 84, 67,  
12, 0, 80, 31, 27, 58, 31, 0, 36, 28, 0, 75, 19,  
22, . . . , 52, 84, 19, 31, 0, 22

Create sequences x and y

#### Sequence x:

49, 29, 29, 29, 5, 19, 27, 0, 7, 3, 36, 65, 27,  
41, 31, 0, 26, 4, 31, 27, 86, 10, 27, 3, 84, 67,  
12, 0, 80, 31, 27, 58, 31, 0, 36, 28, 0, 75, 19,  
22, . . . , 52, 84, 19, 31, 0, 22

#### Sequence y:

49, 29, 29, 29, 5, 19, 27, 0, 7, 3, 36, 65, 27,  
41, 31, 0, 26, 4, 31, 27, 86, 10, 27, 3, 84, 67,  
12, 0, 80, 31, 27, 58, 31, 0, 36, 28, 0, 75, 19,  
22, . . . , 52, 84, 19, 31, 0, 22

#### Training data array x:

49, 29, 29, 29, 5, 19, . . . , 41, 31  
73, 11, 56, 0, 36, 28, . . . , 72, 45  
19, 22, 31, 67, 12, 0, . . . , 12, 0  
. . .  
22, 51, 51, 0, 51, 52, . . . , 86, 42

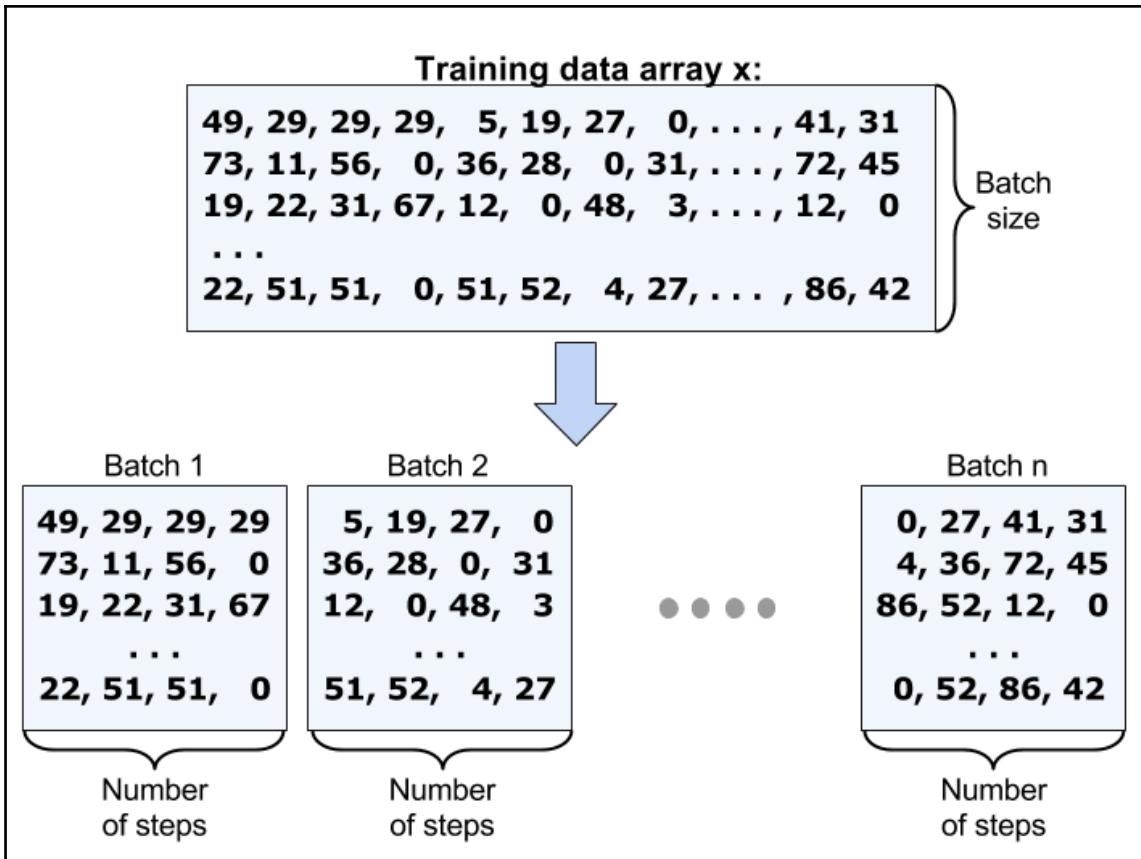
Batch size

#### Training data array y:

29, 29, 29, 5, 19, 27, . . . , 31, 0  
11, 56, 0, 36, 28, 0, . . . , 45, 4  
22, 31, 67, 12, 0, 48, . . . , 0, 86  
. . .  
51, 51, 0, 51, 52, 4, . . . , 42, 0

Batch size

Number of batches X Number of steps



The stall soues tay and the hates,  
 The perse in there is that so the meanes this made there

Ham. Ile teath thes are this makere of a driane,  
 Why shis mestend the Casst of is singe,  
 In this to this, to mers it is for marth,  
 Ase hinees sim thig talid ow a tote andere,  
 In histhene tistere shere this wile and my Lord:  
 And tit mighe the secleer allost heruen, and that hash to sall and hears,  
 If you his moses tonger and mout ofr mesting a forte tis at

Pomin. Where in you dist and sintere shan shall