



Machine Learning

Fundamentals

김선녕(sykim.lecture@gmail.com)



Training Data

Learning Types

One-Hot Encoding

Confusion Matrix/F1 Score

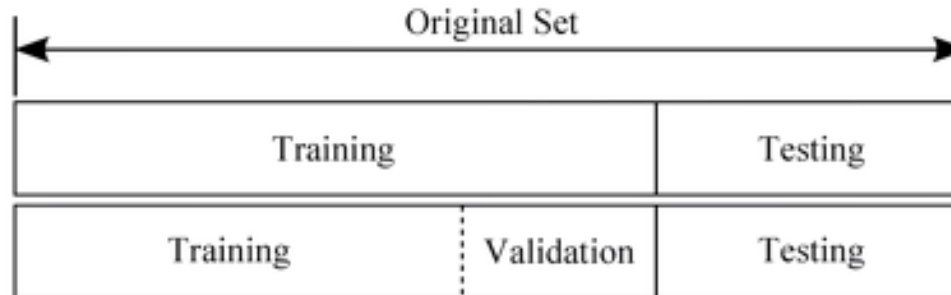
- 통계학자인 피셔(Fisher)교수가 1936년에 캐나다 동부 해안의 가스페 반도에 서식하는 3종의 붓꽃 (setosa, versicolor, virginica)을 50송이씩 채취하여 만들었다.
- Attribute : 150개 샘플 각각에 대해 꽃받침 길이, 꽃받침 너비, 꽃잎 길이, 꽃잎 너비를 측정 기록
 - 1. sepal length in cm
 - 2. sepal width in cm
 - 3. petal length in cm
 - 4. petal width in cm
 - 5. class: setosa, versicolour, virginica



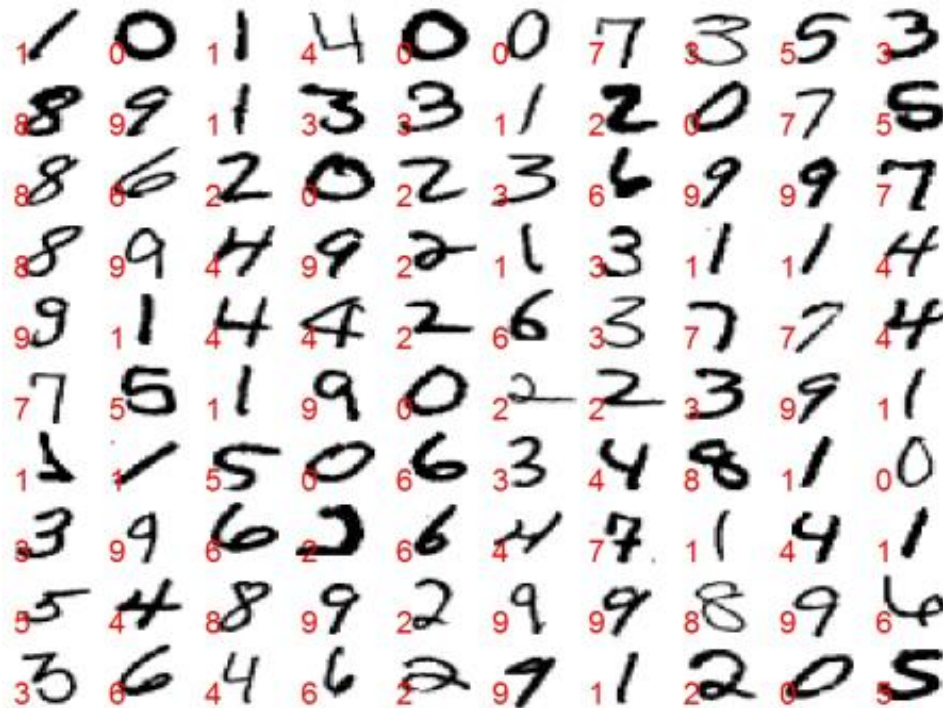
Sepal length ⬇		Sepal width ⬇		Petal length ⬇		Petal width ⬇		Species ⬇					
5.2		3.5		1.4		0.2		I. setosa					
4.9		3.0		1.4		0.2		I. setosa					
4.7		3.2		1.3		0.2		I. setosa					
4.6		3.1		1.5		0.2		I. setosa					
		6.4		3.2		4.5		1.5		I. versicolor			
		6.9		3.1		4.9		1.5		I. versicolor			
		5.5		2.3		4.0		1.3		I. versicolor			
				6.3		3.3		6.0		2.5		I. virginica	
				5.8		2.7		5.1		1.9		I. virginica	
				7.1		3.0		5.9		2.1		I. virginica	
				6.3		2.9		5.6		1.8		I. virginica	

- 훈련데이터(*Training data*) : 머신러닝 모델을 만들 때 사용
- 테스트데이터(*Test data*) : 모델이 얼마나 잘 작동하는 지 측정하는 데 사용
- 검증데이터(*Validation Data*) : *learning rate* 또는 *regularization*와 같은 것을 튜닝하는 데 사용. 모델 성능개선
- 7:3 혹은 8:2

Training, validation and test sets



- *MNIST : Modified National Institute of Standards and Technology database*
- 미국표준국(NIST)에서 수집한 필기 숫자(handwritten digits) 데이터
- 훈련데이터(training set) 60,000, 테스트데이터(test set) 10,000
 - *train – images – idx3 – ubyte.gz: training set images (9,912,422 bytes)*
 - *train – labels – idx1 – ubyte.gz: training set labels (28,881 bytes)*
 - *t10k – images – idx3 – ubyte.gz: test set images (1,648,877 bytes)*
 - *t10k – labels – idx1 – ubyte.gz: test set labels (4,542 bytes)*



-

[illegible]

- A dataset of Zalando's article images
- <https://github.com/zalandoresearch/fashion-mnist>
- 훈련데이터(training set) 60,000, 테스트데이터(test set) 10,000

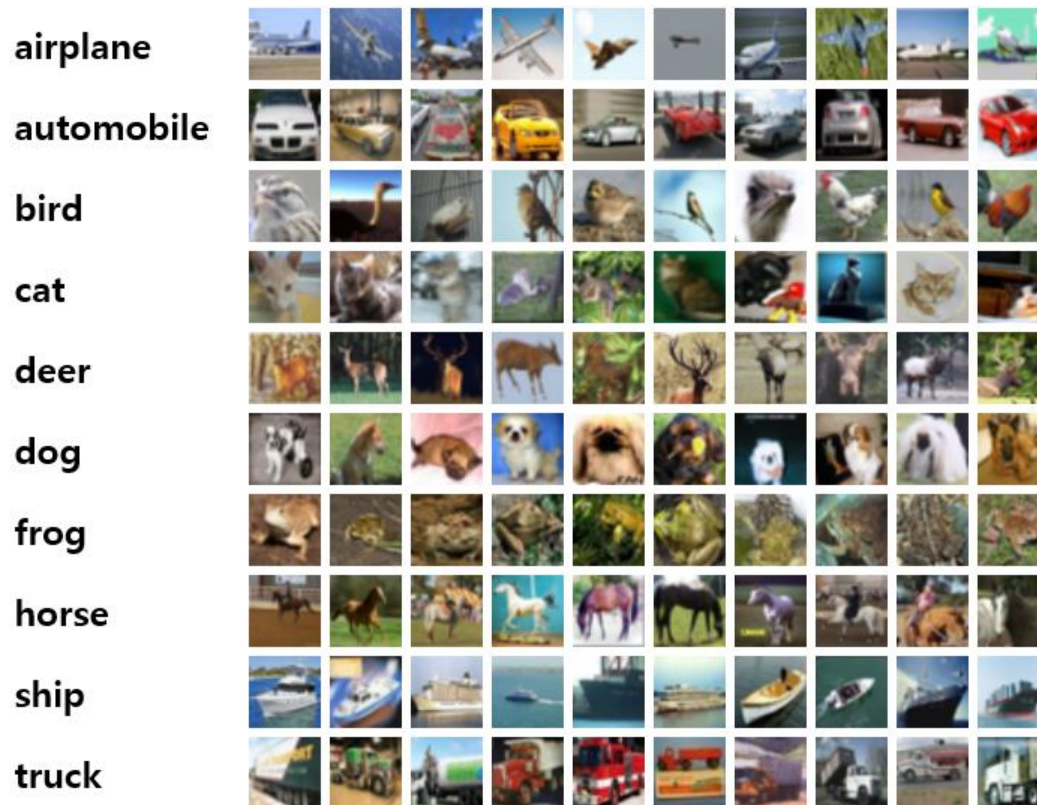
Label	Description
0	T – shirt/top
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Ankle boot



CIFAR-10 dataset (<https://www.cs.toronto.edu/~kriz/cifar.html>)

8

- 60,000 $32 * 32$ color images
 - 50,000 training images
 - 10,000 testing images
- 10 classes : 6,000 images per class



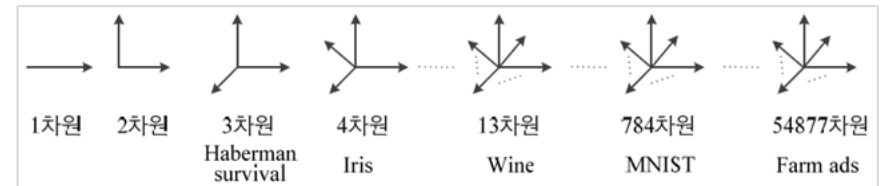
CIFAR-100 dataset (<https://www.cs.toronto.edu/~kriz/cifar.html>)


9

- 100 classes(20 superclasses group) containing 600 images each.
- 500 training images and 100 testing images per class.

20 Superclass	100 Classes
aquatic mammals	beaver, dolphin, otter, seal, whale
fish	aquarium fish, flatfish, ray, shark, trout
flowers	orchids, poppies, roses, sunflowers, tulips
food containers	bottles, bowls, cans, cups, plates
fruit and vegetables	apples, mushrooms, oranges, pears, sweet peppers
household electrical devices	clock, computer keyboard, lamp, telephone, television
household furniture	bed, chair, couch, table, wardrobe
insects	bee, beetle, butterfly, caterpillar, cockroach
large carnivores	bear, leopard, lion, tiger, wolf
large man-made outdoor things	bridge, castle, house, road, skyscraper
large natural outdoor scenes	cloud, forest, mountain, plain, sea
large omnivores and herbivores	camel, cattle, chimpanzee, elephant, kangaroo
medium-sized mammals	fox, porcupine, possum, raccoon, skunk
non-insect invertebrates	crab, lobster, snail, spider, worm
people	baby, boy, girl, man, woman
reptiles	crocodile, dinosaur, lizard, snake, turtle
small mammals	hamster, mouse, rabbit, shrew, squirrel
trees	maple, oak, palm, pine, willow
vehicles 1	bicycle, bus, motorcycle, pickup truck, train
vehicles 2	lawn-mower, rocket, streetcar, tank, tractor

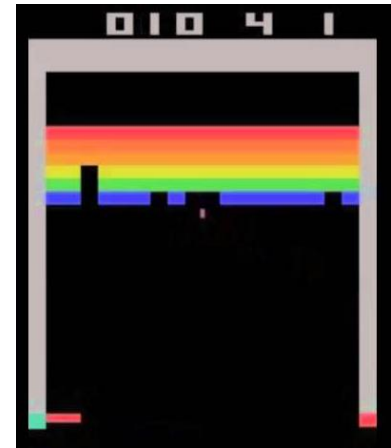
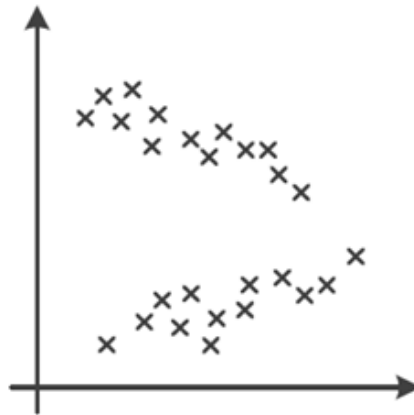
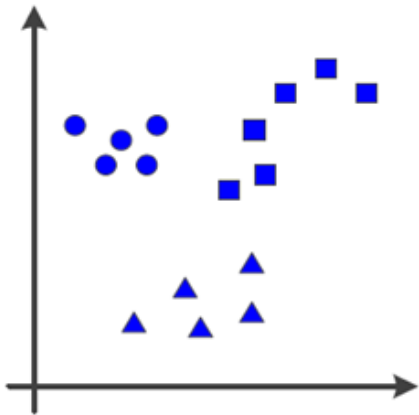
- Haberman Survival Data Set(<https://archive.ics.uci.edu/ml/datasets/>)
 - 유방암 수술을 받은 환자의 생존을 조사한 사례 데이터
 - Attribute : 3
 - Age of patient at time of operation (numerical)
 - Patient's year of operation (year - 1900, numerical)
 - Number of positive axillary nodes detected (numerical)
- Iris Data Set
 - Attribute : 4
 - sepal length in cm, sepal width in cm
 - petal length in cm, petal width in cm
- Wine Data Set (<https://archive.ics.uci.edu/ml/datasets/>)
 - 화학 분석을 사용하여 와인의 기원(origin of wine)을 결정
 - Attribute : 13
 - Alcohol, Malic acid , Ash, Alkalinity of ash, Magnesium, Total phenols, Flavonoids , Nonflavonoid phenols, Proanthocyanins, Color intensity, Hue, OD280/OD315 of diluted wines, Proline
- MNIST Data Set
 - Attribute : 784
 - 784 numbers
- Farm Ads Data set(<https://archive.ics.uci.edu/ml/datasets/>)
 - 다양한 농장 동물 관련 주제를 다루는 12 개의 웹 사이트에 있는 텍스트 광고에서 수집된 데이터
 - Attribute : 54,877
 - Text words 54877



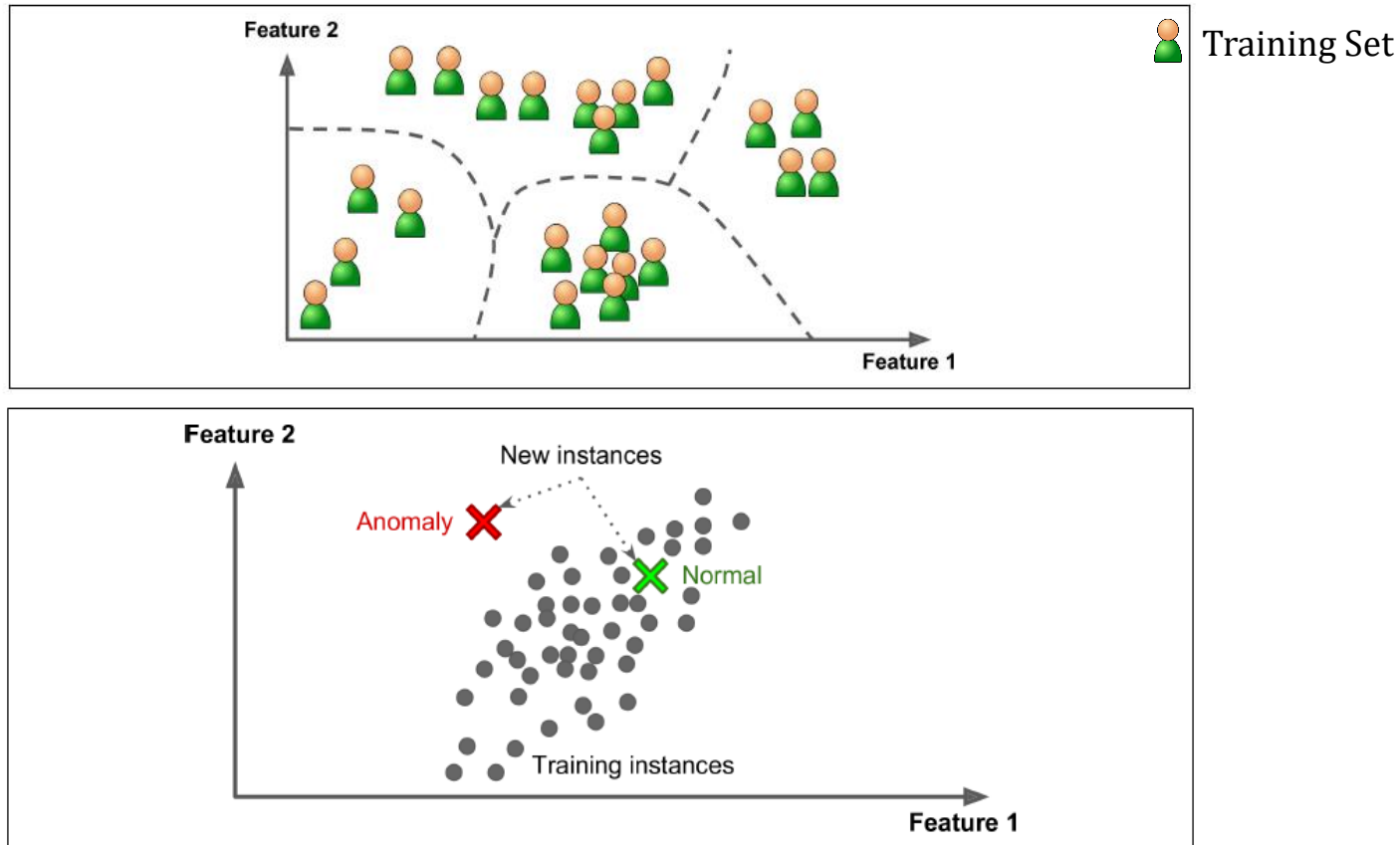


Training Data
Learning Types
One-Hot Encoding
Confusion Matrix/F1 Score

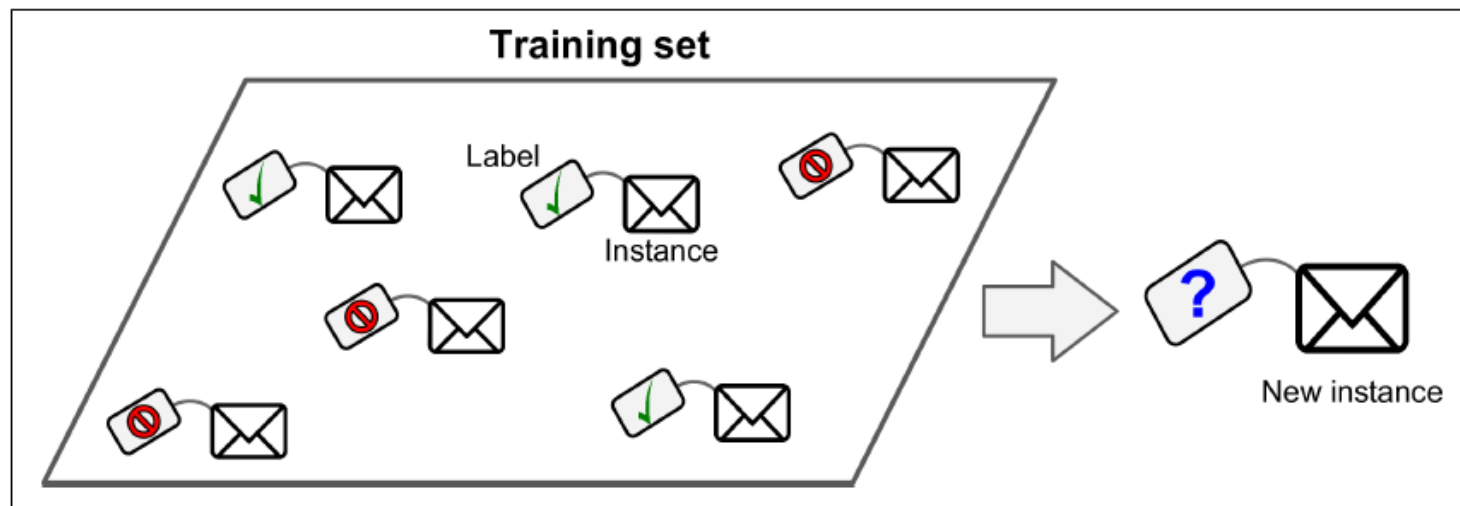
- 지도학습(*Supervised Learning*)
 - 모든 훈련 데이터(Training Data)에 레이블(Label, 정답)의 정보를 갖는다.
- 비지도학습(*Unsupervised Learning*)
 - 모든 훈련 데이터에 레이블의 정보가 없다.
- 준지도 학습(*Semi – supervised Learning*)
 - 지도학습 + 비지도학습
- 강화학습(*Reinforcement Learning*)
 - 어떤 환경을 탐색하는 에이전트가 현재의 상태를 인식하여 어떤 행동을 취한다. 그러면 그 에이전트는 환경으로부터 포상을 얻게 된다.
 - 에이전트가 앞으로 누적될 포상을 최대화하는 일련의 행동으로 정의되는 정책을 찾는 방법



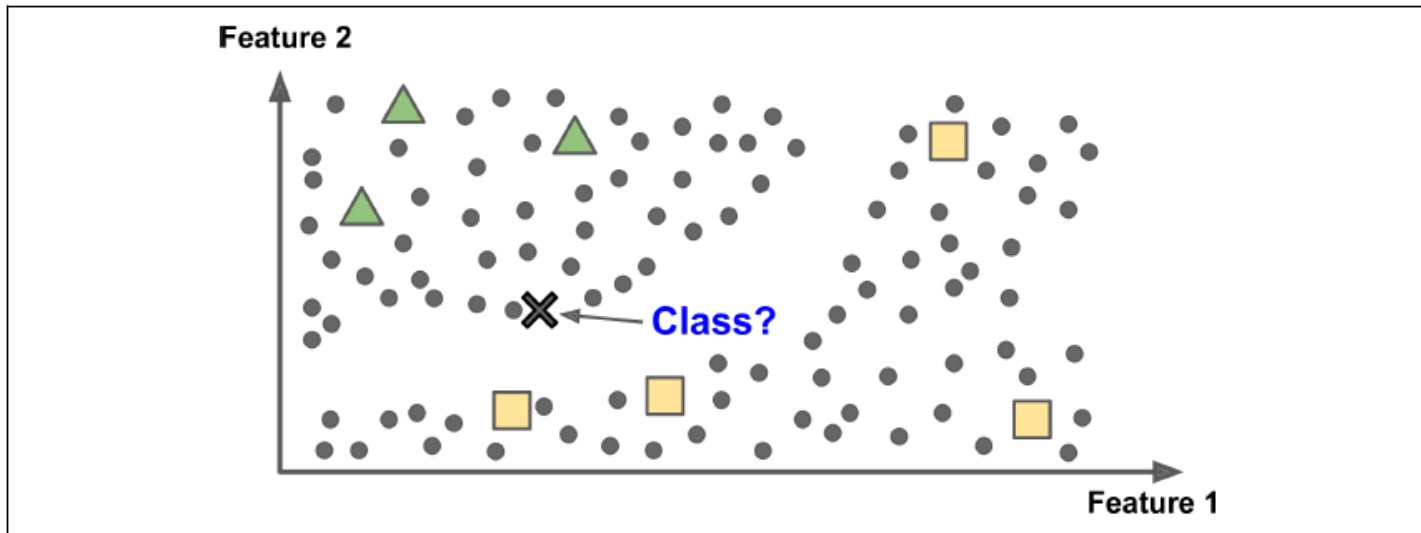
- 군집(clustering)
 - k-평균(k-means)
 - 계층군집분석(hierarchical cluster analysis)
 - 이상치탐지(outlier detection)
- 차원축소(dimension reduction)



- k – 최근접 이웃(k – nearest neighbors)
- 선형회귀(*linear regression*)
- 로지스틱회귀(*logistic regression*)
- 서포트벡터머신(*SVM, support vector machine*)
- 결정트리(*decision tree*)와 랜덤 포레스트(*random forest*)
- 신경망(*neural networks*)



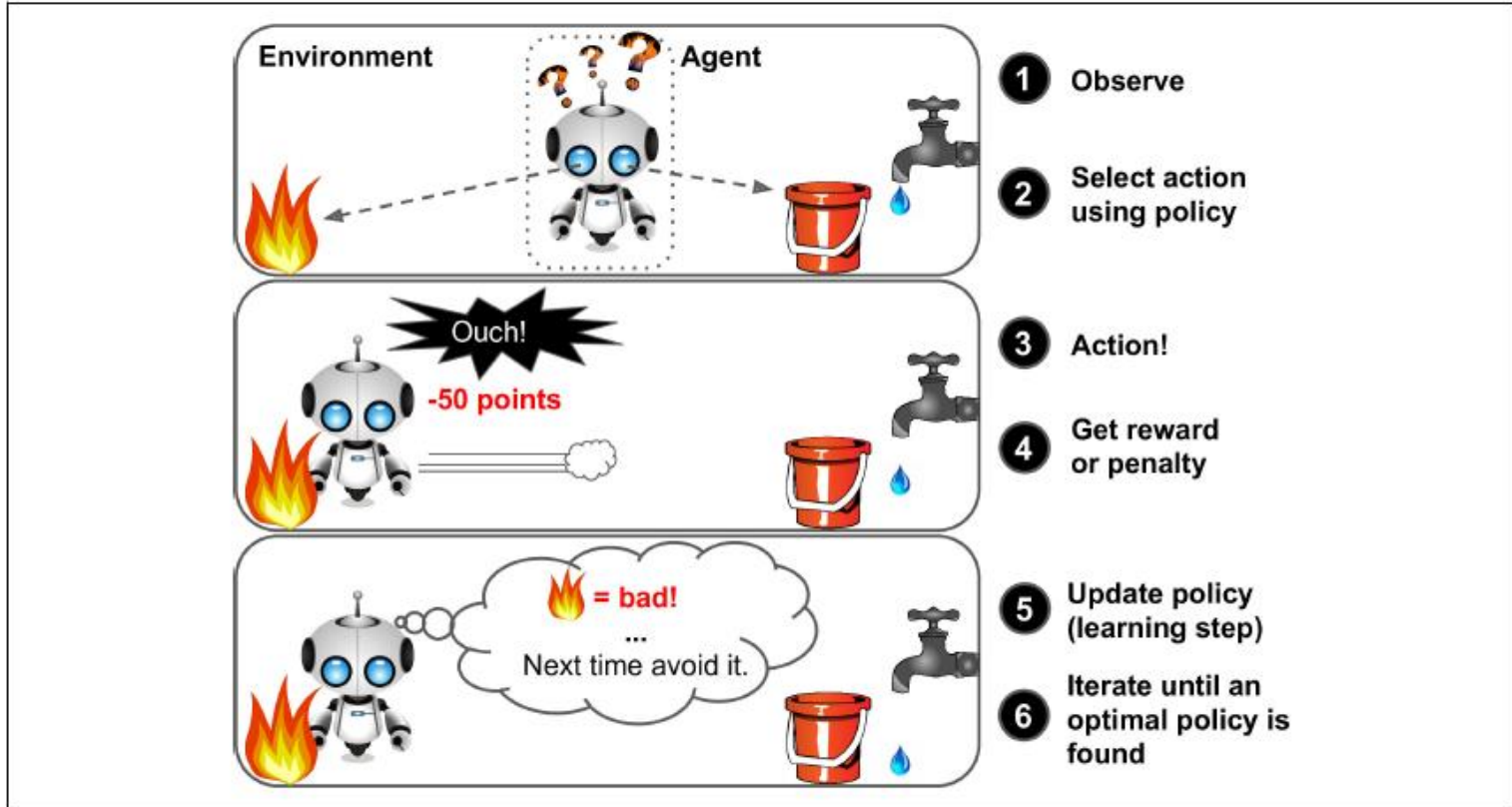
- 데이터에 레이블을 다는 것은 일반적으로 시간과 비용이 많이 든다.
- 일부만 레이블이 있는 데이터



두개의 클래스(삼각형과 사각형)를 사용한 준지도 학습

새로운 샘플(곱셈기호)이 레이블이 있는 사각형 클래스에 더 가깝지만 레이블이 없는 샘플(원)이 이 샘플을 삼각형 클래스로 분류하는 데 도움을 준다

- 행동실행에 따라 보상이나 벌점을 받음 - ③④
- 정책수정(학습) : 최적의 정책을 찾을 때까지 반복 - ⑤⑥



(예) 삶의 만족도 = $\theta_0 + \theta_1 \times 1\text{인당GDP}$

Table 1-1. Does money make people happier?

Country	GDP per capita (USD)	Life satisfaction
Hungary	12,240	4.9
Korea	27,195	5.8
France	37,675	6.5
Australia	50,962	7.3
United States	55,805	7.2

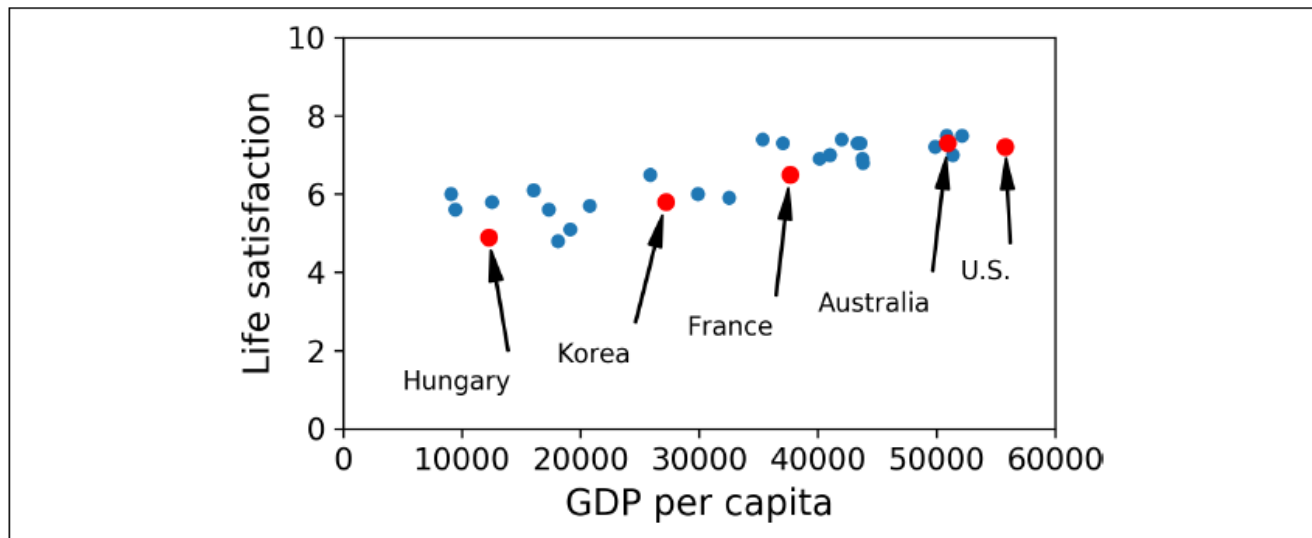


Figure 1-17. Do you see a trend here?

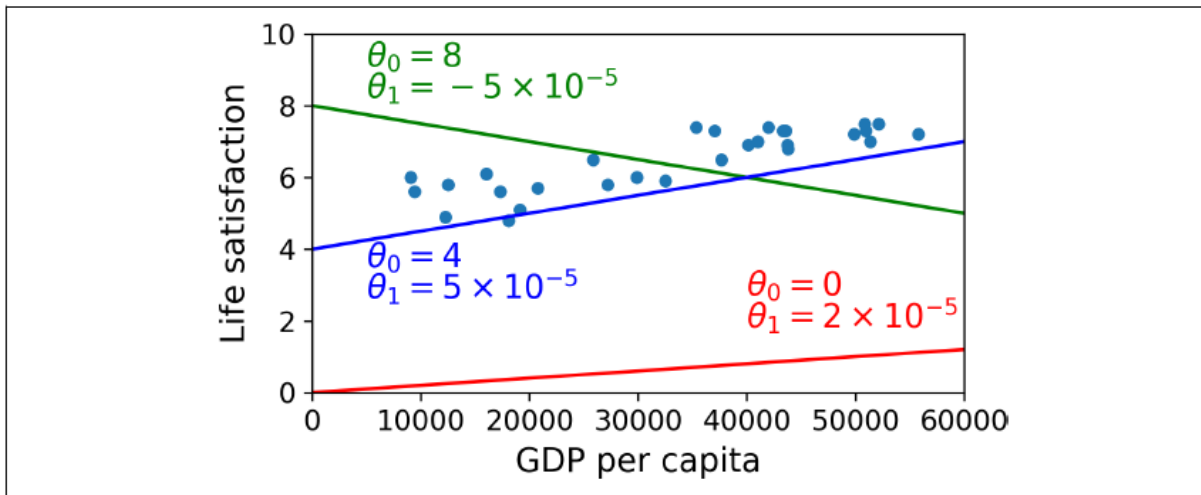


Figure 1-18. A few possible linear models

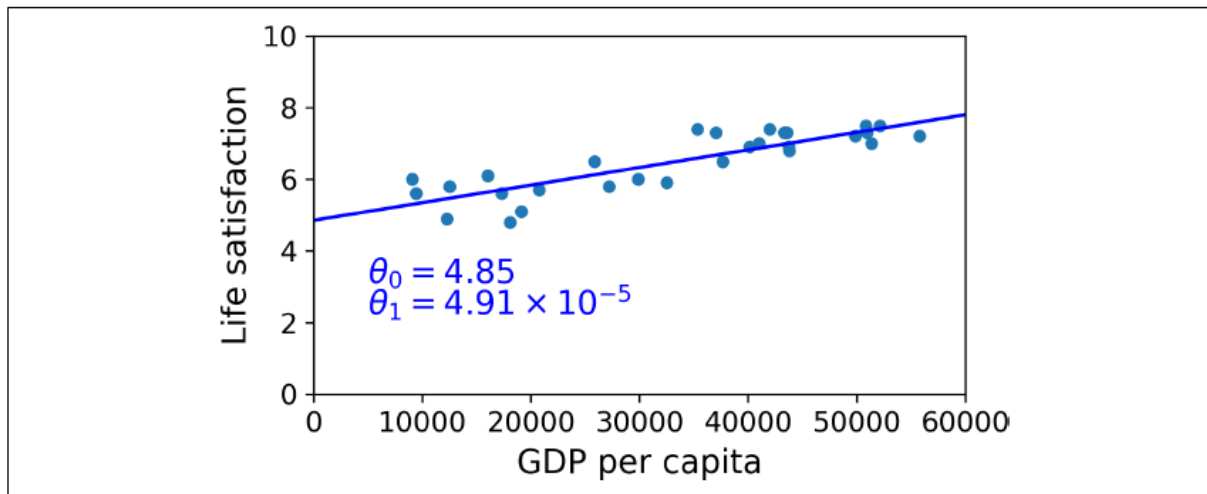
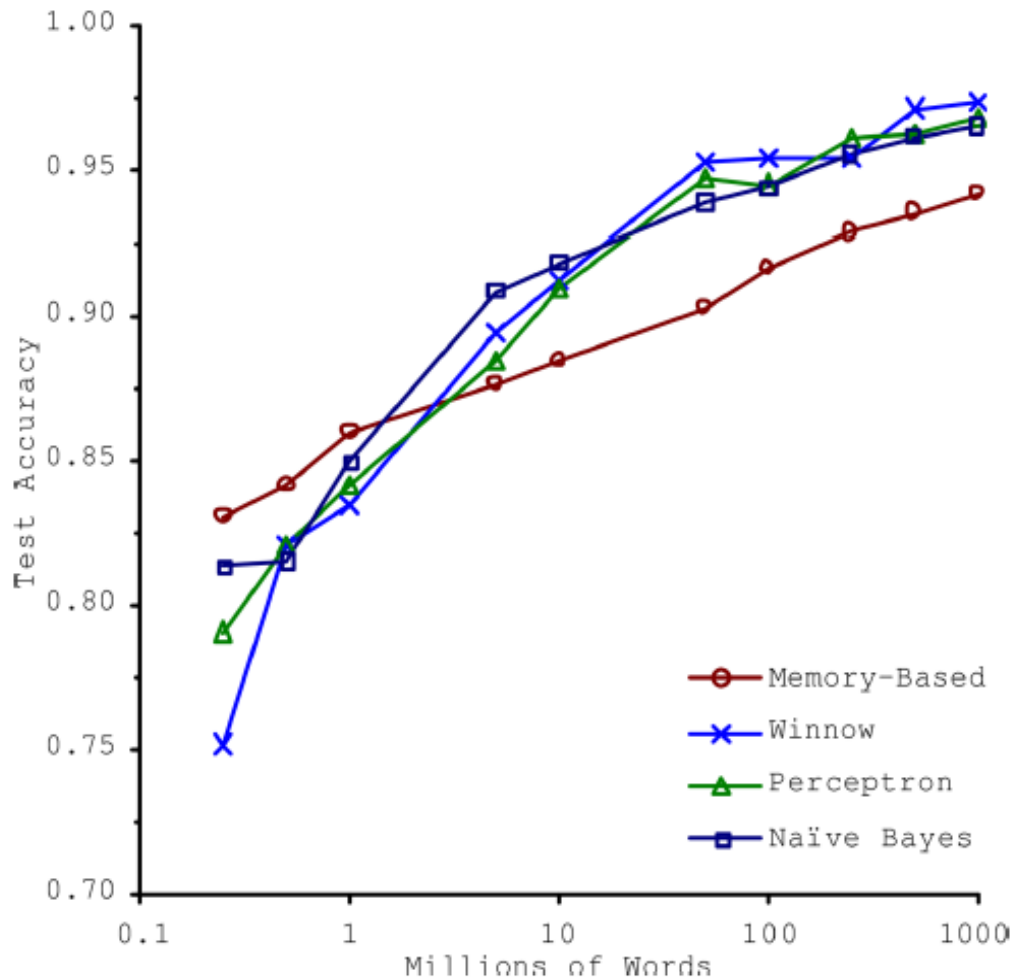


Figure 1-19. The linear model that fits the training data best

- 충분하지 않은 양의 훈련데이터



- 대표성 없는 훈련 데이터

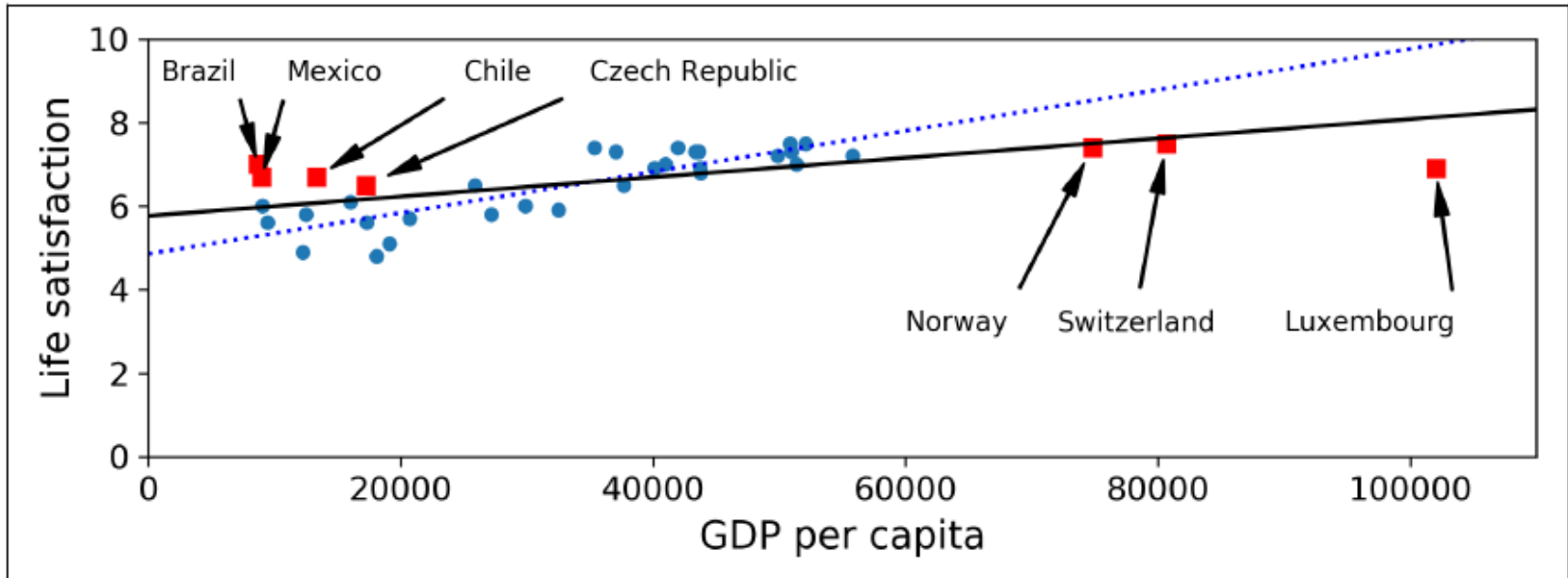
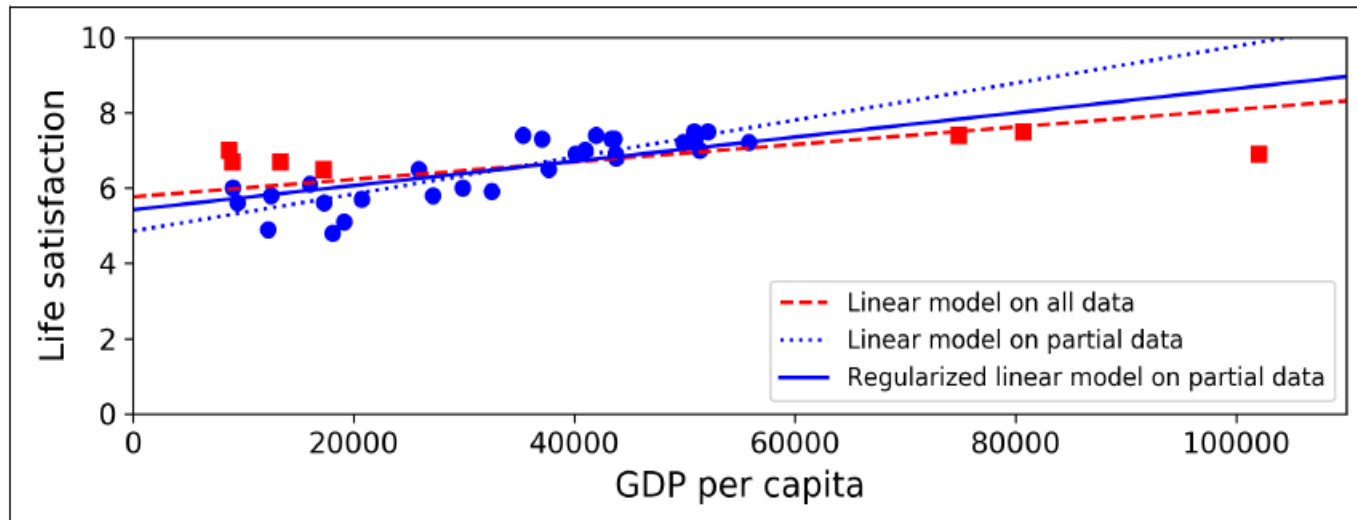
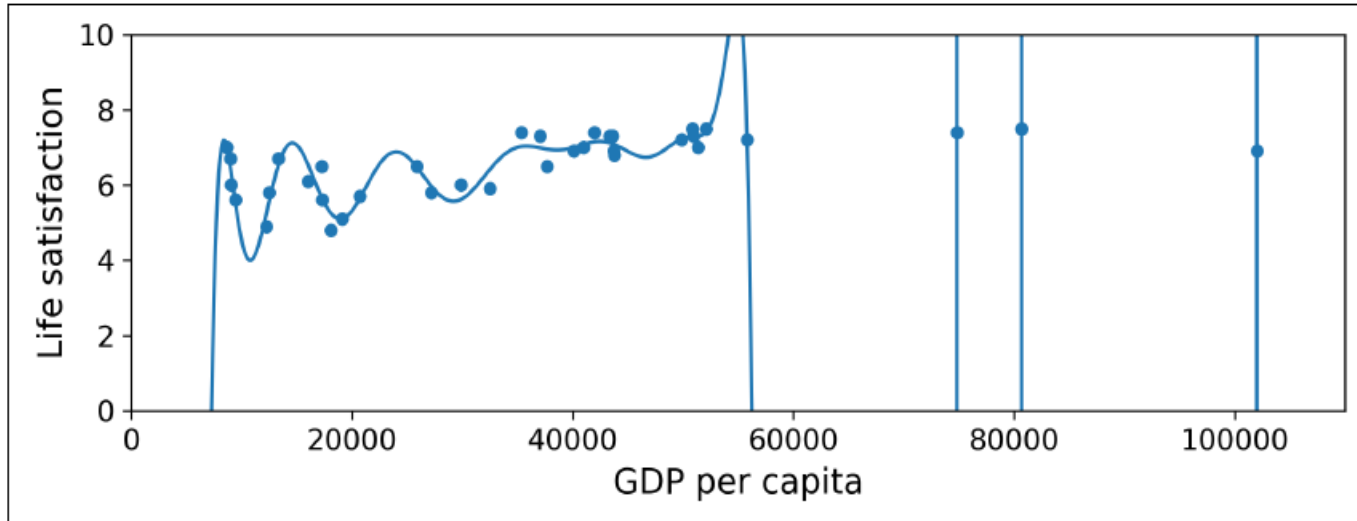



Figure 1-21. A more representative training sample

- 낮은 품질의 데이터
 - 에러, 이상치, 잡음
- 관련성 없는 특성
 - 머신러닝 프로젝트의 핵심요소는 훈련에 사용할 좋은 특성들을 찾는 것

- 훈련데이터 과대적합





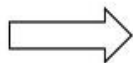
Training Data
Learning Types
One-Hot Encoding
Confusion Matrix/F1 Score

- 다중 분류에서 인코딩 방법으로 출력 값의 형태가 정답 레이블은 1(true), 그외의 레이블은 0(false)인 배열
- 한 개의 요소만 1이고 나머지는 0인 N차원의 벡터로 표현

(예1) [0,0,0,1,0] 4번째 인덱스만 1이고 나머지는 0. 즉, 4번째 인덱스가 정답.


(예2)

Index	Job
1	Police
2	Doctor
3	Student
4	Teacher
5	Driver



One hot encoded data					
[1	0	0	0	0]
[0	1	0	0	0]
[0	0	1	0	0]
[0	0	0	1	0]
[0	0	0	0	1]

airplane	[1 0 0 0 0 0 0 0 0 0]	1
automobile	[0 1 0 0 0 0 0 0 0 0]	2
bird	[0 0 1 0 0 0 0 0 0 0]	3
cat	[0 0 0 1 0 0 0 0 0 0]	4
deer	[0 0 0 0 1 0 0 0 0 0]	5
dog	[0 0 0 0 0 1 0 0 0 0]	6
frog	[0 0 0 0 0 0 1 0 0 0]	7
horse	[0 0 0 0 0 0 0 1 0 0]	8
ship	[0 0 0 0 0 0 0 0 1 0]	9
truck	[0 0 0 0 0 0 0 0 0 1]	10



Training Data
Learning Types
One-Hot Encoding
Confusion Matrix/F1 Score

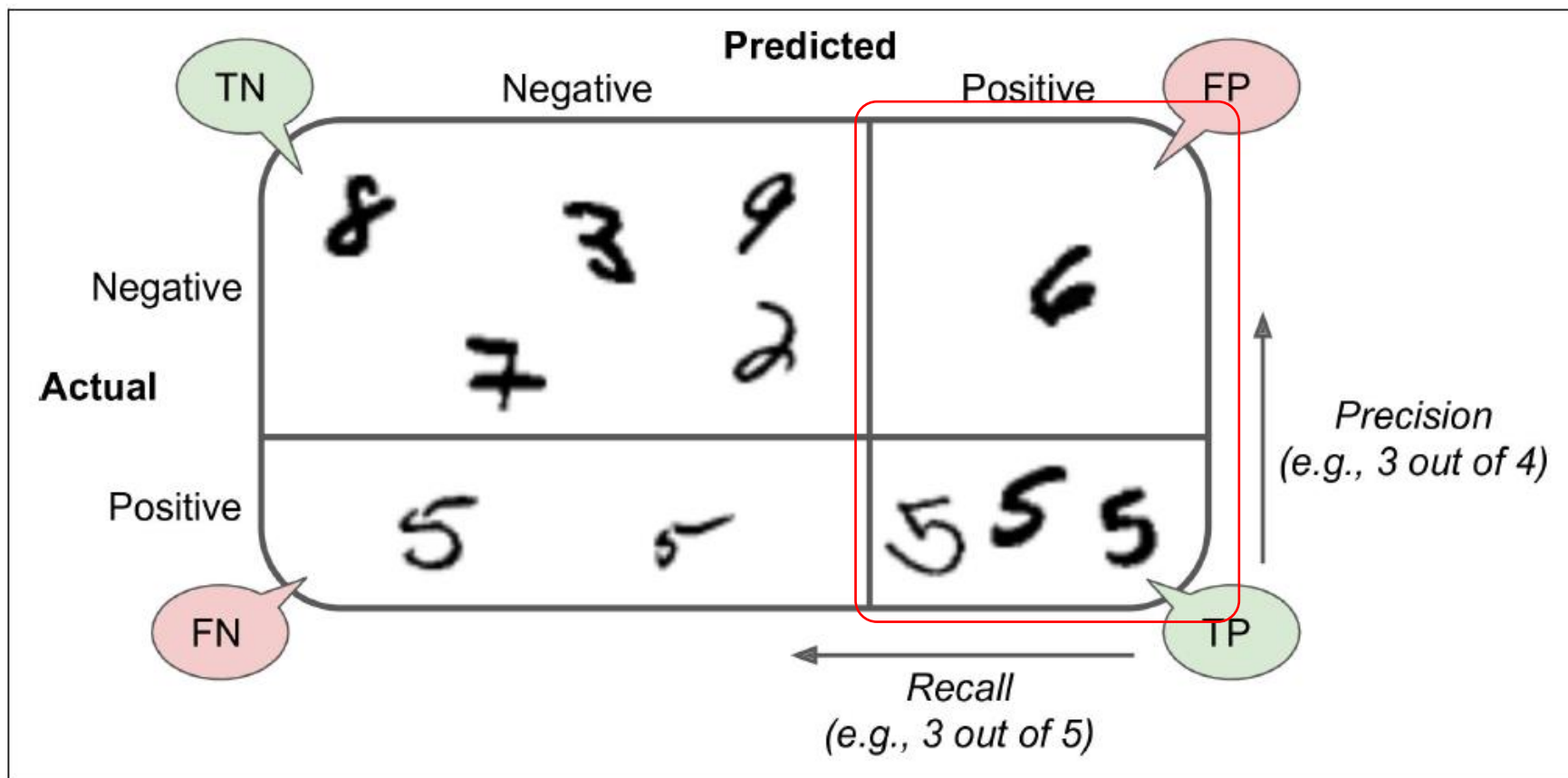
- 예측된 분류가 얼마나 잘 예측되었는지 평가할 때 사용되는 테이블
- sci-learn의 confusion_matrix 함수 사용
- 정확도(Accuracy)
 - 모델이 얼마나 정확한지 평가하는 척도
$$\frac{TP + TN}{(TP + TN + FP + FN)}$$
- 정밀도(Precision)
 - True라고 예측한 값 중에서 실제 값이 True인 수치(비율)
$$\frac{TP}{(TP + FP)}$$
- 재현도(Recall), 민감도(Sensitivity), TPR
 - True인 경우에 True로 잘 예측한 수치(비율)
$$\frac{TP}{(TP + FN)}$$
- 특이도(Specificity)
 - False인 경우에 False로 잘 예측한 수치(비율)
$$\frac{TN}{(TN + FP)}$$

		Predicted Class	
		Positive	Negative
Real Class	Positive	TP (True Positive)	FN (False Negative) <i>Type II Error</i>
	Negative	FP (False Positive) <i>Type I Error</i>	TN (True Negative)

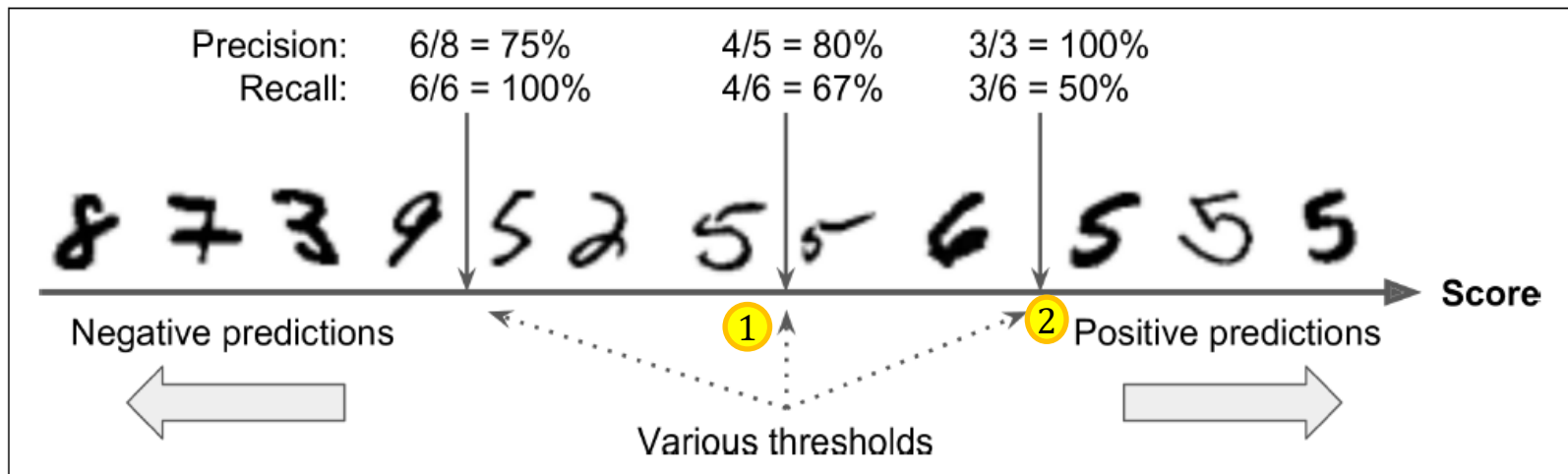
Confusion Matrix

		Predicted Class			
		Positive	Negative		
Real Class	Positive	TP (True Positive)	FN (False Negative) <i>Type II Error</i>	Sensitivity (Recall) $\frac{TP}{(TP + FN)}$	TPR (True Positive Rate) = Sensitivity $\frac{TP}{(TP + FN)}$
	Negative	FP (False Positive) <i>Type I Error</i>	TN (True Negative)	Specificity $\frac{TN}{(TN + FP)}$	FPR (False Positive Rate) $1 - \text{Specificity}$ $= \frac{FP}{(TN + FP)}$
		Precision PPV (Positive Predictive Value) $\frac{TP}{(TP + FP)}$	NPV (Negative Predictive Value) $\frac{TN}{(TN + FN)}$	Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$	

$$F1(F - \text{Score}) = 2 * \frac{\text{Precision} * \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}}$$



- 임계값(*threshold*)보다 크면 양성 클래스에 할당하고 그렇지 않으면 음성클래스에 할당
- 임계값(*threshold*) 이 1인 경우
 - 4개의 TP(숫자 5)와 1개의 FP(숫자 6)
 - 정밀도(*Precision*) : 80%(5 개중 4개)
 - 재현도(*Recall*) : 67%(숫자5는 6개중 4개)
- 임계값(*threshold*) 이 2인 경우
 - FP(숫자6)이 TN이 되어 정밀도 (*Precision*) : 100%
 - TP하나가 FN이 되어 재현도 (*Recall*) : 50%
- Trade-off : 임계값이 높을 수록 재현율은 낮아지고 정밀도는 높아진다.



```
import pandas as pd

# Loading the Breast Cancer Wisconsin dataset
df = pd.read_csv('https://archive.ics.uci.edu/ml/'
                 'machine-learning-databases'
                 '/breast-cancer-wisconsin/wdbc.data', header=None)

df.head()

df.shape  # (569, 32)

from sklearn.preprocessing import LabelEncoder

X = df.loc[:, 2:].values
y = df.loc[:, 1].values
le = LabelEncoder()
y = le.fit_transform(y)
le.classes_  # ['B', 'M']
le.transform(['M', 'B'])  # [1. 0]
```

```
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline
from sklearn.svm import SVC

pipe_svc = make_pipeline(StandardScaler(),
                          SVC(random_state=1))
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, stratify=y, random_state=1)

from sklearn.metrics import confusion_matrix

pipe_svc.fit(X_train, y_train)
y_pred = pipe_svc.predict(X_test)
confmat = confusion_matrix(y_true=y_test, y_pred=y_pred)
print(confmat)
'''
[[71  1]
 [ 2 40]]
'''
```

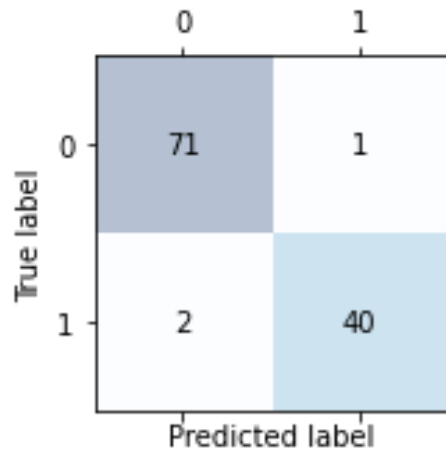


```
import matplotlib.pyplot as plt

fig, ax = plt.subplots(figsize=(2.5, 2.5))
ax.imshow(confmat, cmap=plt.cm.Blues, alpha=0.3)
for i in range(confmat.shape[0]):
    for j in range(confmat.shape[1]):
        ax.text(x=j, y=i, s=confmat[i, j], va='center', ha='center')

plt.xlabel('Predicted label')
plt.ylabel('True label')

plt.tight_layout()
plt.show()
```



```
# test dataset
confmat = confusion_matrix(y_true=y_test, y_pred=y_pred, labels=[1, 0])
print(confmat)
'''
[[40  2]
 [ 1 71]]
'''

from sklearn.metrics import precision_score, recall_score, f1_score

print('Precision: %.3f' % precision_score(y_true=y_test, y_pred=y_pred))
print('Recall: %.3f' % recall_score(y_true=y_test, y_pred=y_pred))
print('F1: %.3f' % f1_score(y_true=y_test, y_pred=y_pred))
'''
Precision: 0.976
Recall: 0.952
F1: 0.964
'''
```