모델 검증

- 임계값에 따른 평가지표 확인

1.1.1 이진 분류의 평가지표

1.1.2 임계값과 평가지표

1.1.3 평가지표 - ROC 커브, AUC

1.1.4 다중 분류의 평가지표

학습 내용

- 예측을 0,1로 하는 것이 아니라 확률로 하기.
- 임계값을 조정하는 것에 따라 정밀도와 민감도가 변하는 것을 확인해 본다.

목차

이진 분류 예측 - 예측을 0,1로 하는 것이 아니라 확률로 해보기

- 400개(음성), 50개(양성) 으로 이루어진 불균형 데이터
- 사용 함수 : decision_function(), predict_proba()
  - decision_function을 0으로, predict_proba를 0.5의 임계값으로 사용

01. 데이터 준비 및 라이브러리 임포트

목차로 이동하기

In [1]:

```
import warnings
warnings.filterwarnings(action='ignore')
```

In [2]:

```
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
import mglearn
from sklearn.metrics import classification_report
import matplotlib.pyplot as plt
import numpy as np
```

In [10]:

```
from mglearn.datasets import make_blobs

### 데이터 만들기
X, y = make_blobs(n_samples=(400, 50),
                  centers=2,
                  cluster_std=[7.0, 2],          # 클러스터의 표준 편차
                  random_state=42)

print(X.shape, y.shape)

X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)
```

```
(450, 2) (450,)
```

In [11]:

```
X_train[0:10], y_train[0:10]
```

Out[11]:

```
(array([[-0.18299954,  3.77488037],
        [-2.73847051,  5.21031273],
        [ 4.14376924,  4.97596054],
        [ 5.21160962,  2.64208326],
        [ 1.78996928, 14.3168401 ],
        [ 1.09604925, 12.61078778],
        [-2.16954623,  3.19763531],
        [-8.04251881, 12.31456463],
        [-0.48077362, 23.54209172],
        [-8.287678  ,  6.76458524]]),
 array([0, 0, 0, 1, 0, 0, 0, 0, 0, 0]))
```

데이터 시각화

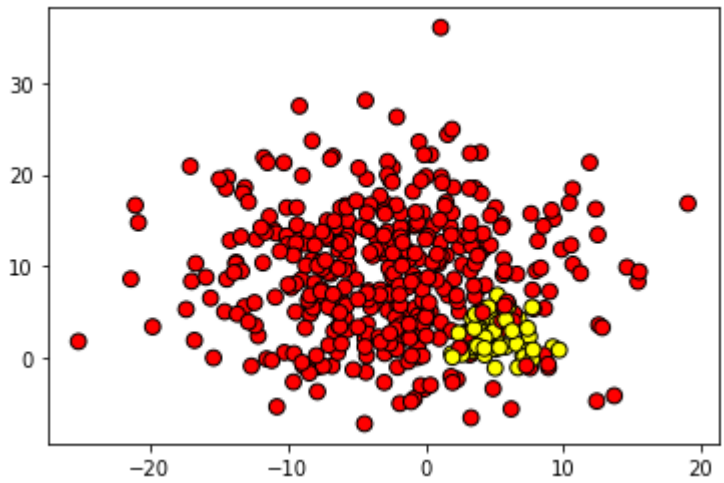
- 400개의 음성 클래스
- 50개의 양성 클래스

In [12]:

```
plt.scatter(X[:,0], X[:,1],
            c=y,
            cmap=plt.cm.autumn, s=60, edgecolors='k')
```

Out[12]:

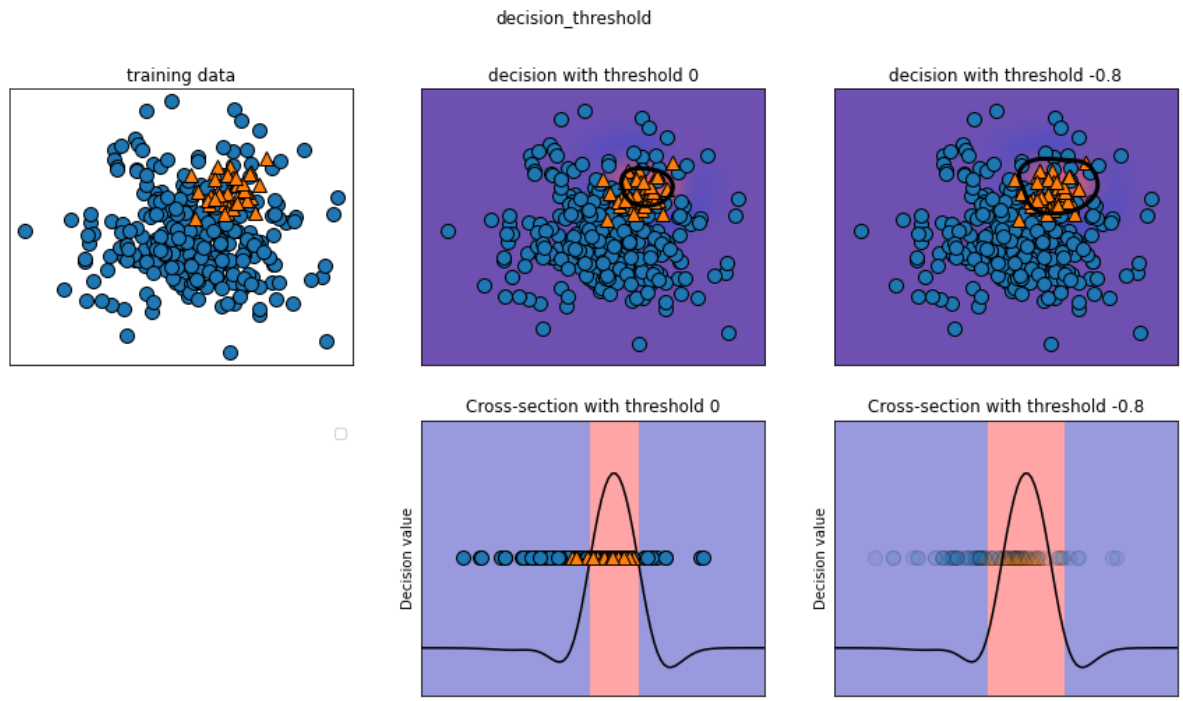
```
<matplotlib.collections.PathCollection at 0x7fa72ba8daf0>
```



임계값에 따른 값을 확인

In [13]:

```
mglearn.plots.plot_decision_threshold()
```



- 중앙 윗부분에 있는 검은 원은 decision_function이 정확히 0일 때의 임계점을 나타낸다.
- 원안의 포인트는 **양성 클래스**로 분류, 바깥쪽 포인트는 **음성 클래스**로 분류

재현율(recall) 조정해보기

- svc.predict()함수로 예측 시. 재현율을 조정하기 어려운 조건.
- decision_function()함수로 예측하여 임계값이 조정이 가능.

02. 모델(SVC) 예측 후, 평가 지표 확인

목차로 이동하기

- 모델 : SVC
- 정밀도, 민감도, f1-score 확인

In [14]:

```
svc = SVC(gamma=.05).fit(X_train, y_train)
pred = svc.predict(X_test)
print(classification_report(y_test, pred))
```

```
              precision    recall  f1-score   support

           0       0.99      0.93      0.96       107
           1       0.42      0.83      0.56         6

    accuracy                           0.93       113
   macro avg       0.70      0.88      0.76       113
weighted avg       0.96      0.93      0.94       113
```

- 클래스 1에 대해 상당한 작은 정밀도(0.35)를 얻었음. 재현율은 절반(0.67)
- 클래스 0의 샘플이 매우 많으므로 분류기는 소수인 클래스 (양성)1보다 클래스 (음성)0에 초점.

모델의 임계값을 활용하여 0,1 개수 조정

- 임계값을 0에서 -0.8로 낮추기
- 임계값을 0에서 -0.8로 조정시 양성 클래스(1)의 개수가 늘어난다.

In [15]:

```
pred = svc.decision_function(X_test)
print(pred[0:10])
np.min(pred), np.max(pred)
```

```
[-1.0167542   0.72583536 -1.17766946 -1.00425497 -1.0002495  -0.999771
82
 -1.07285711 -1.2206812  -1.24018502 -1.30361098]
```

Out[15]:

```
(-1.509707253620952, 1.6245457437087478)
```

In [16]:

```
decision_0 = svc.decision_function(X_test)   > 0      # 임계값을 0으로
decision_m08 = svc.decision_function(X_test) > -.8    # 임계값을 -0.8로

# TP - 잘 맞추는 것을 늘린다.
print("임계값 0 일때    : 1(양성) 개수 :", decision_0.sum() )
print("임계값 -0.8 일때 : 1(양성) 개수 :", decision_m08.sum() )
```

```
임계값 0 일때    : 1(양성) 개수 : 12
임계값 -0.8 일때 : 1(양성) 개수 : 18
```

- 임계값을 변경하여 역으로 1의 개수가 늘고 0의 개수가 줄어든다.

In [17]:

```
print("임계값 0 일때    : 0(음성) 개수 :", len(decision_0) - decision_0.sum())
print("임계값 -0.8 일때 : 0(음성) 개수 :", len(decision_m08) - decision_m08.sum() )
```

```
임계값 0 일때    : 0(음성) 개수 : 101
임계값 -0.8 일때 : 0(음성) 개수 : 95
```

In [18]:

```
y_pred_0 = svc.decision_function(X_test)   > 0
y_pred_08 = svc.decision_function(X_test) > -.8
```

In [19]:

```
# 임계값 0
print(classification_report(y_test, y_pred_0))
```

```
              precision    recall  f1-score   support

           0       0.99      0.93      0.96       107
           1       0.42      0.83      0.56         6

    accuracy                           0.93       113
   macro avg       0.70      0.88      0.76       113
weighted avg       0.96      0.93      0.94       113
```

In [20]:

```
print(classification_report(y_test, y_pred_08))
```

```
              precision    recall  f1-score   support

           0       1.00      0.89      0.94       107
           1       0.33      1.00      0.50         6

    accuracy                           0.89       113
   macro avg       0.67      0.94      0.72       113
weighted avg       0.96      0.89      0.92       113
```

임계값을 낮추는 것은

- 정밀도(precision) **0.42에서 0.33로 낮아지고**
- 재현율(recall)-sensitivity(민감도)는 **0.83에서 1로 올라감.**
- 결론적으로 **1(양성)의 수가 늘어나기 때문에 TP(진짜 양성)의 개수가 늘어난다.**

Review

- 정밀도(precision)
  - TP/(TP + NP) : **예측 양성 전체 중**에 정확하게 잘 맞추었을까?

- 재현율(recall) :
  - TP/(TP + FN) : 실제 양성 데이터의 얼마나 잘 맞추었을까?
  - 다른 말로 **민감도(sensitivity), 적중률(hit rate), 진짜 양성 비율(TPR)**이라고 합니다.

- F1-score

$$F = 2 * \frac{정밀도 * 재현율}{정밀도 + 재현율}$$

실습

- 임계값을 0보다 큰 값으로 조정해 보고 재현율(recall)를 확인해 보기
- 임계값을 0으로 하고 재현율(recall)과 기타 평가지표를 확인해 보기

기타 방법

- predict_proba()메서드는 출력이 0에서 1 사이로 고정
  - 보통은 0.5를 임계값-이는 양성과 음성이 50%분류이다.
  - 임계값을 높이는 것은 양성이 분류될 확률이 많이 나올 때, 수행