

임계값에 따른 평가지표 확인

학습 내용

- 임계값을 조정하는 것에 따라 정밀도와 민감도가 변하는 것을 확인해 본다.

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_function`을 0으로, `predict_proba`를 0.5의 임계값으로 사용

01 데이터 준비하기

In [1]:

```
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
```

In [2]:

```
from mglearn.datasets import make_blobs
X, y = make_blobs(n_samples=(400, 50),
                  centers=2,
                  cluster_std=[7.0, 2],
                  random_state=22)

print(X.shape, y.shape)

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

(450, 2) (450,)

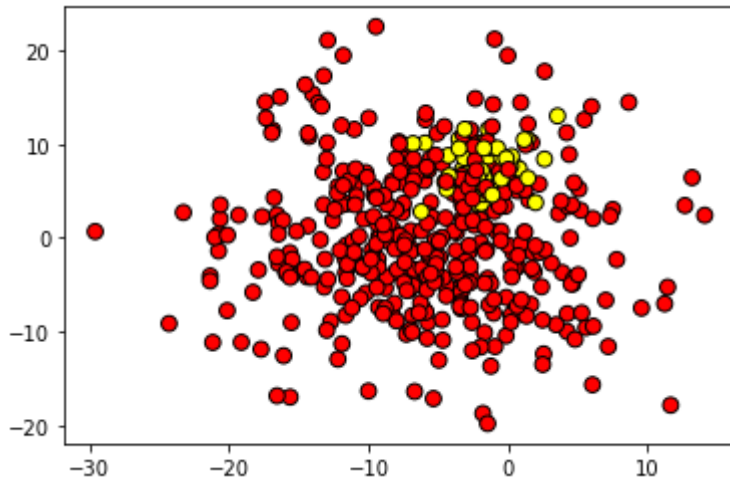
```
C:\Users\Wtoto\Anaconda3\lib\site-packages\sklearn\utils\deprecation.py:86: FutureWarning: Function make_blobs is deprecated; Please import make_blobs directly from scikit-learn
warnings.warn(msg, category=FutureWarning)
```

In [3]:

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

Out[3]:

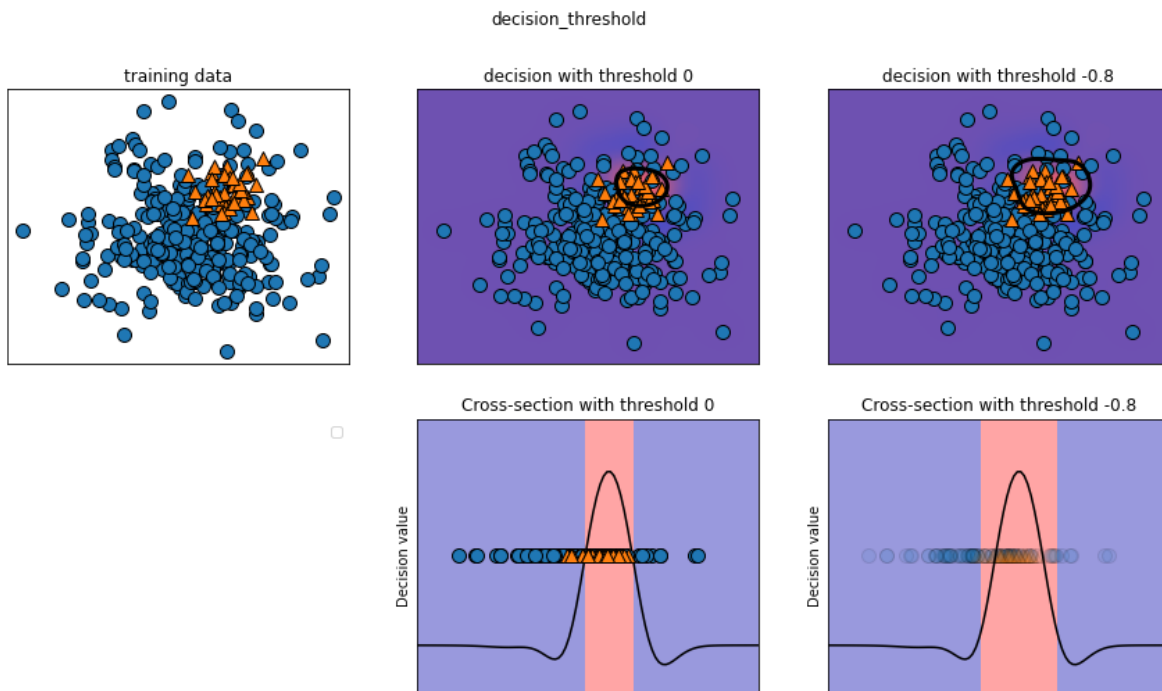
<matplotlib.collections.PathCollection at 0x1717e7af3d0>



임계값에 따른 값을 확인

In [4]:

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



- 중앙 윗부분에 있는 검은 원은 `decision_function`이 정확히 0일 때의 임계점을 나타낸다.
- 원안의 포인트는 양성 클래스로 분류, 바깥쪽 포인트는 음성 클래스로 분류

모델 선택 및 학습 후, 예측

In [5]:

```
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	0.97	0.89	0.93	104
1	0.35	0.67	0.46	9
accuracy			0.88	113
macro avg	0.66	0.78	0.70	113
weighted avg	0.92	0.88	0.89	113

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

임계값을 0에서 -0.8로 낮추기

- 1의 개수가 늘어난다.

In [6]:

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

# 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 : 17
 임계값 -0.8 일때 : 1(양성) 개수 : 28

- 0의 개수가 줄어든다.

In [7]:

```
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 : 96
 임계값 -0.8 일때 : 0(음성) 개수 : 85

In [8]:

```
y_pred_lower_threshold = svc.decision_function(X_test) > -.8
```

In [9]:



```
print(classification_report(y_test, y_pred_lower_threshold))
```

	precision	recall	f1-score	support
0	1.00	0.82	0.90	104
1	0.32	1.00	0.49	9
accuracy			0.83	113
macro avg	0.66	0.91	0.69	113
weighted avg	0.95	0.83	0.87	113

임계값을 낮추는 것은

- 정밀도(precision) 0.35에서 0.32로 낮아지고
- 재현율(recall)-sensitivity(민감도)는 0.67에서 1로 올라감.
- 결론적으로 1(양성)의 수가 늘어나기 때문에 TP(진짜 양성)의 개수가 늘어난다.

정밀도와 재현율

- 정밀도(precision)
 - $TP/(TP + NP)$: 예측 양성 것중에 얼마나 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.

실습

- 임계값을 0보다 큰 값으로 조정해 보고 recall를 확인해 보기
- 임계값을 0으로 하고 recall과 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