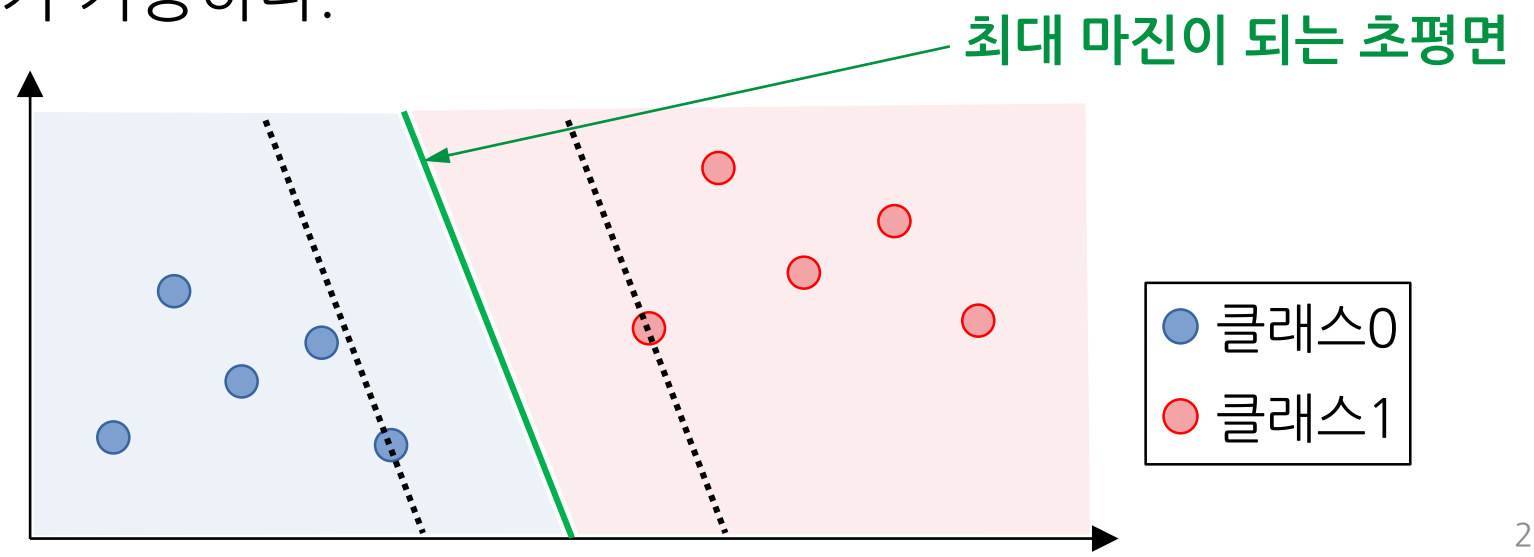
선형 SVM

소프트 마진 분류

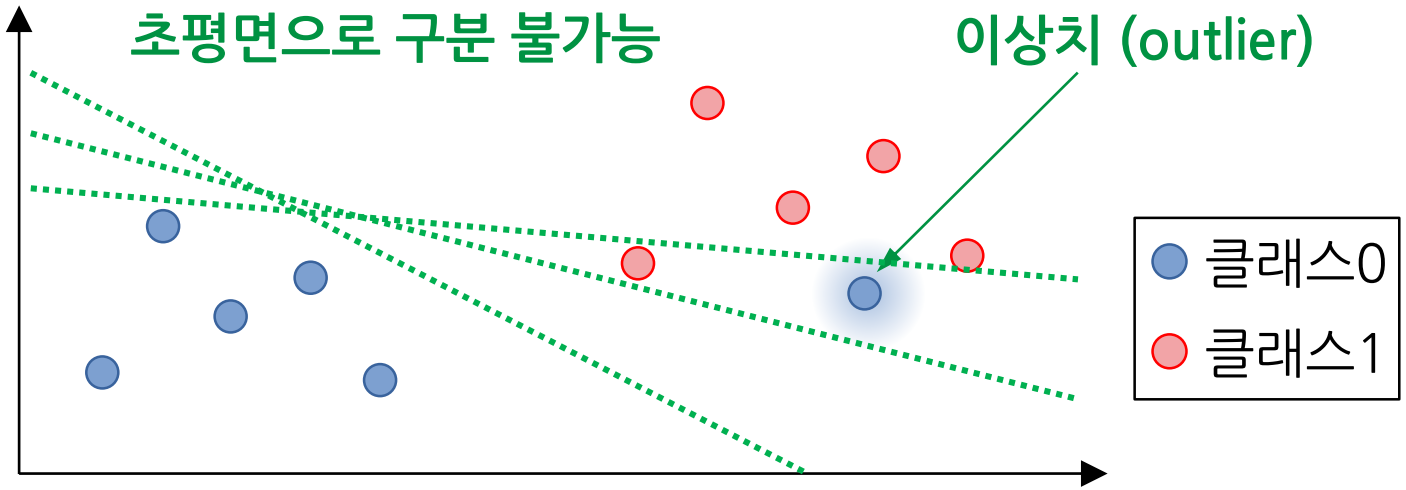
선형 SVM

- 하드 마진 (Hard Margin) 분류
  - 두 개의 클래스에 대해 최대 마진이 되는 초평면을 찾는다.
    - 이와 같이 하드 마진 분류를 수행하는 선형 SVM을 최대 마진 분류기(maximum margin classifier)라고 한다.
  - 모든 훈련 데이터들은 마진의 바깥쪽에 위치하게 된다.
  - 데이터들이 정확하게 선형적으로 구분되는 경우에만 분류가 가능하다.



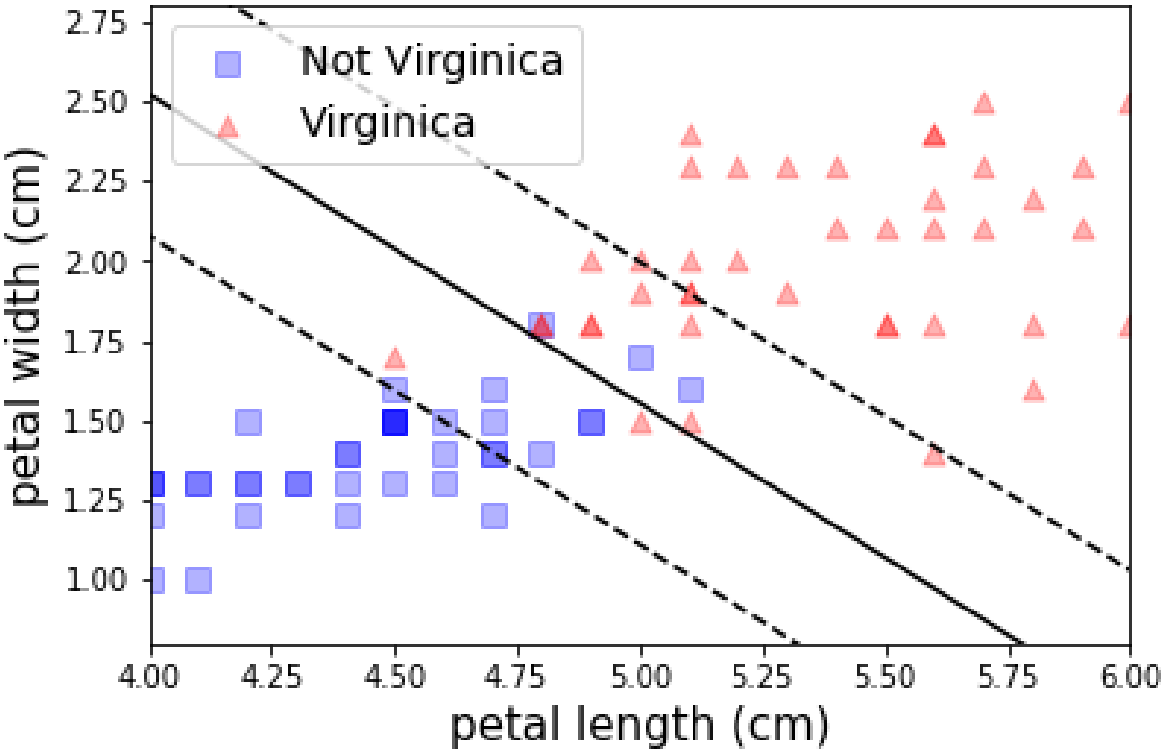
선형 SVM

- 하드 마진 분류의 한계
  - 모든 경우에 반드시 초평면이 존재하는 것은 아니다.
    - 데이터가 정확하게 선형적으로 구분되지 않는 경우에는 결정 경계를 찾는 것이 불가능하다.
  - 분류 모형이 일반화되기 어렵다.
    - 이상치가 존재할 경우, 초평면이 없거나 잘 일반화되지 않는다.



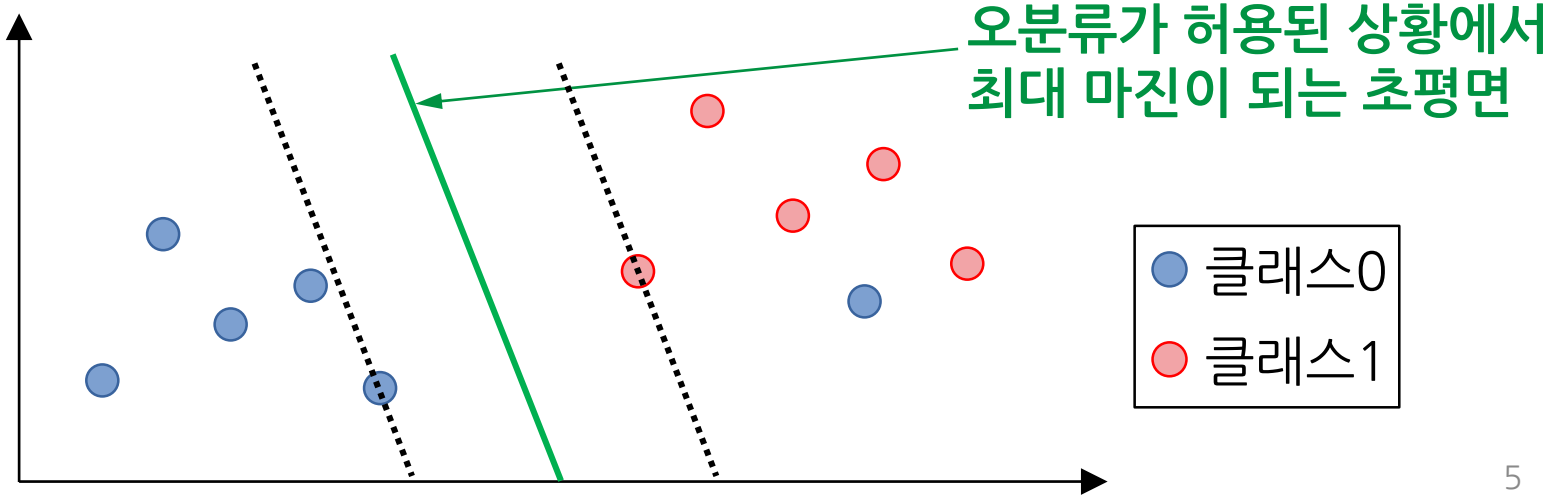
선형 SVM

- 하드 마진 분류의 한계
  - 붓꽃 데이터 IRIS에 대해 하드 마진 분류를 시도하더라도 하드 마진 분류가 실질적으로 진행되지 않는다.



선형 SVM

- 소프트 마진 (Soft Margin) 분류
  - 이러한 경우에는 어느 정도의 오류를 허용하면서 가급적 최대 마진이 되는 초평면을 찾는다.
    - 이와 같이 소프트 마진 분류를 수행하는 선형 SVM을 서포트 벡터 분류기(support vector classifier)라고 한다.
  - 잘못 분류되는 데이터가 있지만 초평면을 찾을 수 있다.
  - 과대적합을 방지하거나 줄일 수 있다.



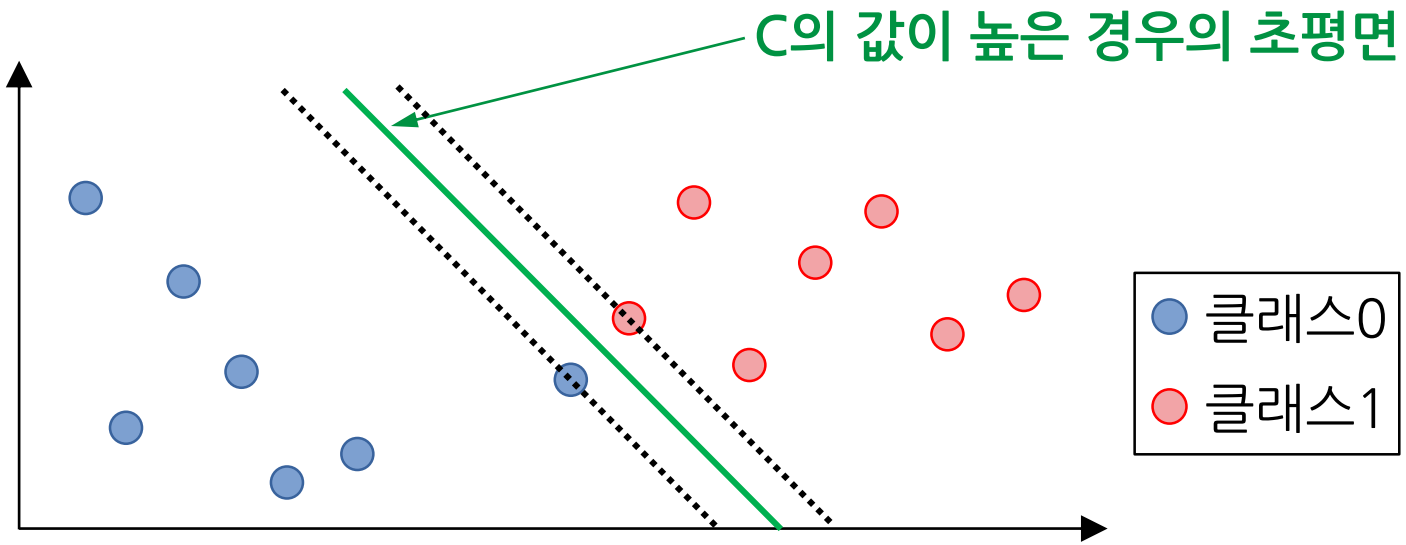
선형 SVM

- 소프트 마진 분류
  - 하이퍼파라미터 C(cost)를 이용하여 허용할 오류의 수준을 결정한다.
  - 즉, “원래 데이터와 다른 클래스로 분류되는 경우를 얼마나 많이 허용할 것인가”를 조정하는 규제 값이다.

※ 매개변수 C는 이론적으로 소프트 마진 분류를 위해 도입된 여유 변수(slack variable) ζ 의 값에 적용되는 변수이다.

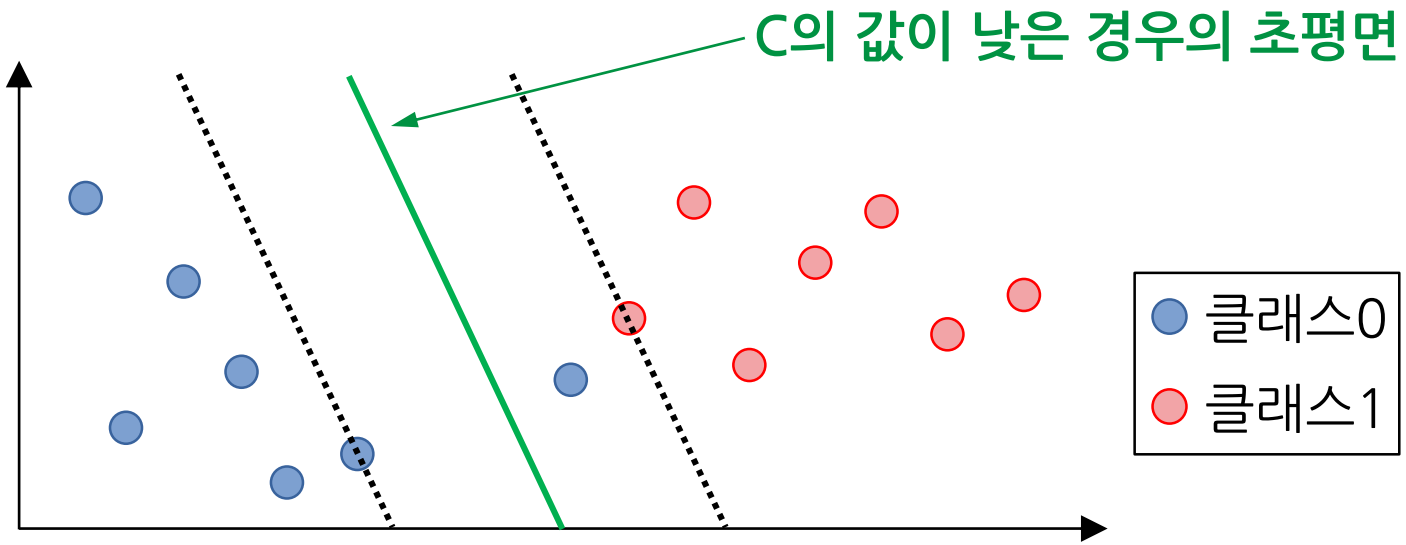
선형 SVM

- 소프트 마진 분류
  - C의 값이 높은 경우, 오류에 대해서 더 엄격하게 적용한다.
    - 마진이 작아진다.
    - 오분류율이 낮아진다.
    - 과대적합이 될 수 있다.



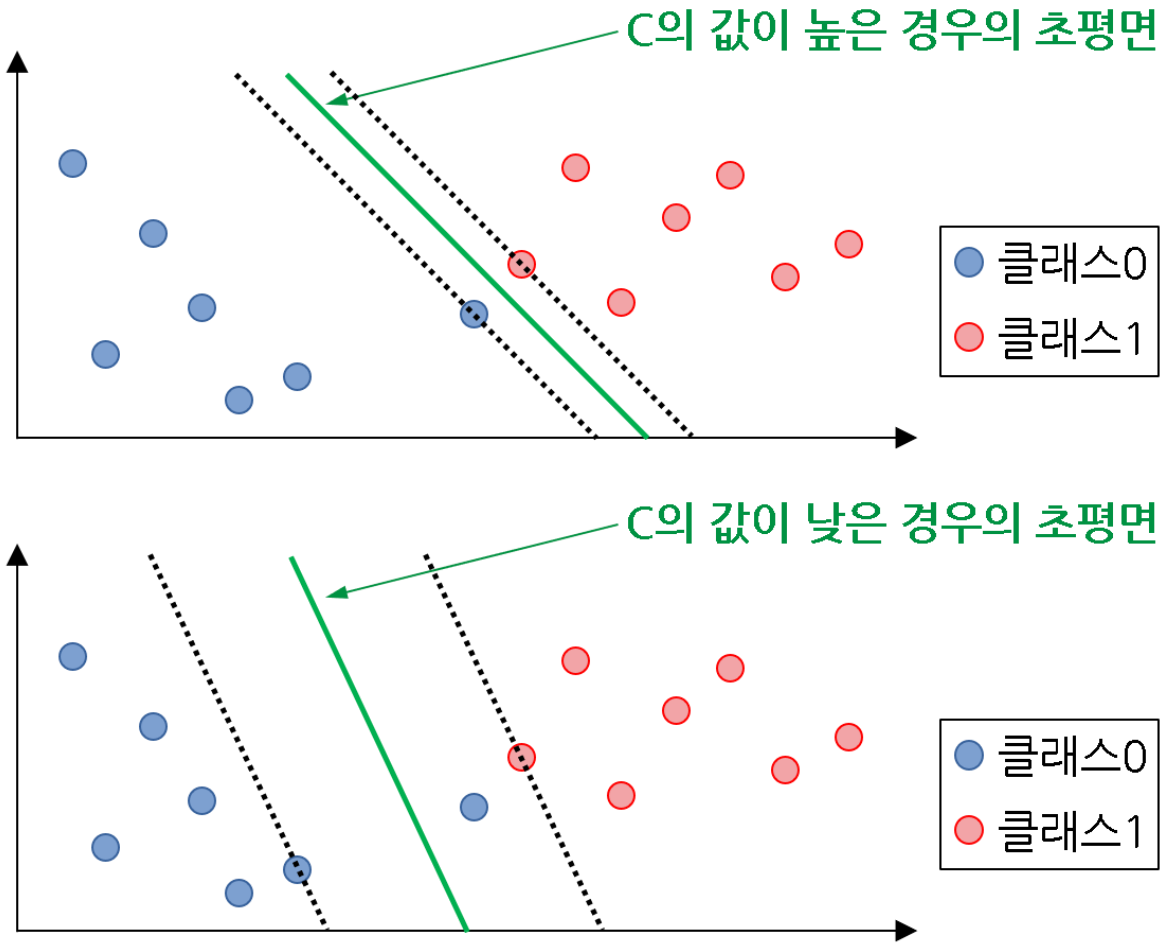
선형 SVM

- 소프트 마진 분류
  - C의 값이 낮은 경우, 오류에 대해서 덜 엄격하게 적용한다.
    - 마진이 커진다.
    - 오분류율이 높아진다.
    - 과소적합이 될 수 있다.



선형 SVM

- 소프트 마진 분류
  - 매개변수 C의 영향 함께 보기



선형 SVM

- 사이킷런으로 선형 SVM (소프트 마진) 분류 수행
  ① svm 모듈에 있는 LinearSVC를 또는 SVC를 이용하여 선형 SVM 객체를 생성한다.
  - 매개변수 C는 공통적으로 사용되며, 허용할 오류의 정도를 결정하는 하이퍼파라미터 값이다.
  - 매개변수 loss는 LinearSVC에서만 사용되는 손실 함수 이름이다. 기본값은 'squared_hinge'이며, 필요하다면 'hinge'로 변경한다.
  - 매개변수 kernel은 SVC에서만 사용되는 커널 트릭의 이름이다. 기본값은 'rbf'이며, 선형 SVM에서는 'linear'로 변경하여 적용한다.

선형 SVM

- 사이킷런으로 선형 SVM (소프트 마진) 분류 수행

  ① **svm** 모듈에 있는 **LinearSVC**를 또는 **SVC**를 이용하여 선형 SVM 객체를 생성한다.

```
1  import sklearn.svm as svm
2
3  clf = svm.SVC(C=1, kernel="linear")
```

  ※ 아래와 같이 LinearSVC에서 loss='hinge'로 설정하여 생성해도 의미적으로는 동일하지만, 적용된 구현 기법이 SVC와 다소 다르기 때문에 완전히 똑같은 결과가 나오지는 않는다.

```
1  clf = svm.LinearSVC(C=1, loss="hinge")
```

선형 SVM

- 사이킷런으로 선형 SVM (소프트 마진) 분류 수행

  ② 선형 SVM 객체에 대하여 **fit** 메소드를 이용하여 훈련한다.

  (붓꽃 데이터를 이용하였으며, 훈련 및 검증 데이터를 분리하지 않고 원본 데이터 전체를 학습에 사용하였다.)

```
1  import sklearn.datasets as d
2
3  iris = d.load_iris()
4  X = iris.data[:, (2, 3)]
5  y = (iris.target == 2).astype(np.int)
6
7  clf.fit(X, y)
```

선형 SVM

- 사이킷런으로 선형 SVM (소프트 마진) 분류 수행

③ 실행 객체 또는 분류 모형에 대하여 **predict** 메소드를 이용하여 예측을 수행한다.

```
1 X_test1 = [[5.5, 1.9]]
2 clf.predict(X_test1)
```

```
array([1])
```

```
1 X_test2 = [[4.7, 1.8]]
2 clf.predict(X_test2)
```

```
array([0])
```

④ **accuracy_score** 함수 또는 **score** 메소드를 이용하여 정확도를 구한다.

선형 SVM

- 사이킷런으로 선형 SVM (소프트 마진) 분류 수행
  - 매개변수 C의 값을 변화시키면 마진의 너비가 달라지고, 이에 따라 분류 결과도 달라지는 것을 확인할 수 있다.

C = 1

```
1  X_test1 = [[5.5, 1.9]]
2  clf.predict(X_test1)
```

```
array([1])
```

```
1  X_test2 = [[4.7, 1.8]]
2  clf.predict(X_test2)
```

```
array([0])
```

C = 100

```
1  X_test1 = [[5.5, 1.9]]
2  clf.predict(X_test1)
```

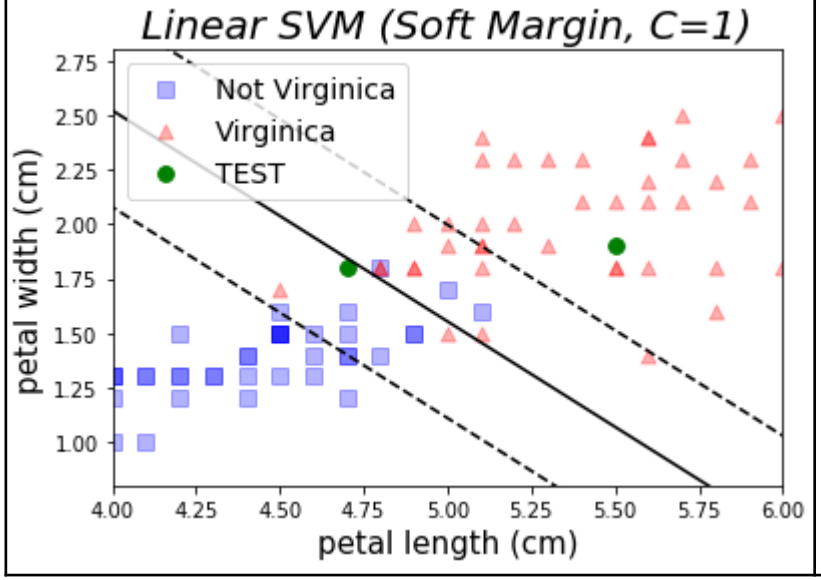
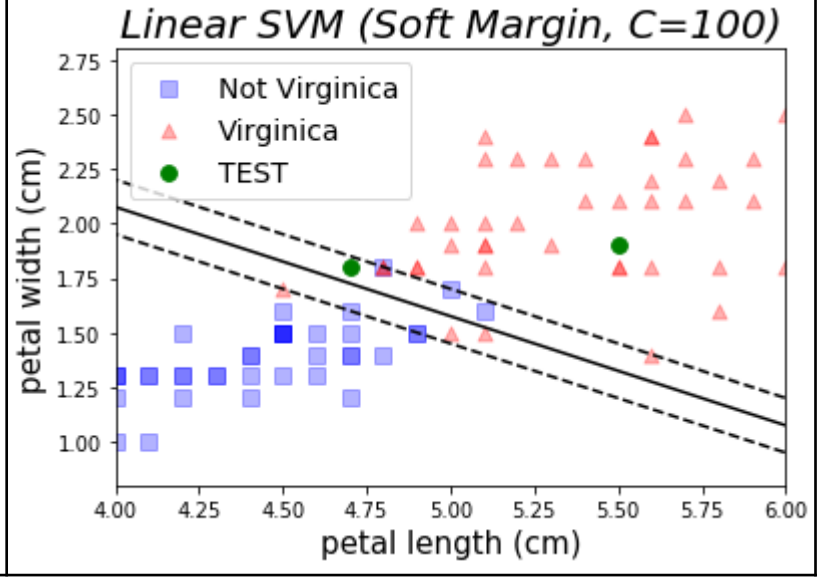
```
array([1])
```

```
1  X_test2 = [[4.7, 1.8]]
2  clf.predict(X_test2)
```

```
array([1])
```

선형 SVM

- 사이킷런으로 선형 SVM (소프트 마진) 분류 수행
  - 매개변수 C의 값을 변화시키면 마진의 너비가 달라지고, 이에 따라 분류 결과도 달라지는 것을 확인할 수 있다.

C = 1	C = 100
	

선형 SVM

다중 클래스 분류

다중 클래스 분류

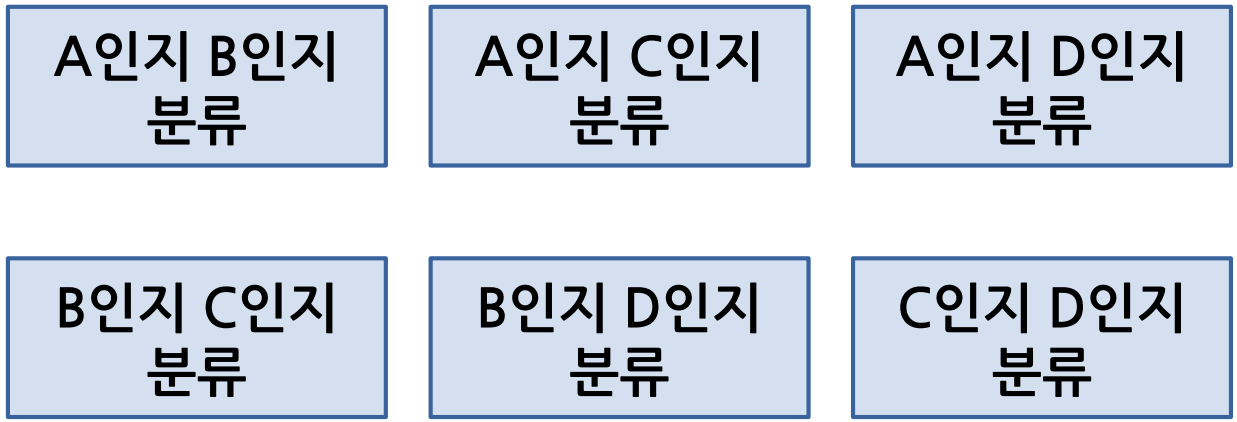
- 다중 클래스 분류 (Multi-class Classification)
  - 다중 클래스 분류(multi-class classification)는 말 그대로, 두 종류가 아닌 여러 개의 클래스를 분류하는 것을 의미한다.
  - 다항 분류(multinomial classification)라고도 한다.

  - 분석 기법에 따라서 이진 분류만 가능한 경우도 있고, 다항 분류까지 가능한 경우도 있다.
  - 이진 분류만 가능한 방법이라도 이진 분류기 자체를 여러 개 조합하여 다항 분류를 수행할 수 있다.

다중 클래스 분류

- OvO (One-Vs-One) 전략
  - K개 클래스에 대하여 2개의 클래스 조합을 선택하여 분류하는 과정을 모든 조합에 대해서 수행한다.
  - 각 분류를 통해서 판별된 결과를 기반으로 가장 많은 결과값을 획득한 클래스를 최종 결과로 선택한다.

예) 클래스가 4종류일 때 (A, B, C, D)



다중 클래스 분류

- OvA 또는 OvR (One-Vs-All 또는 One-Vs-Rest) 전략
  - K개 클래스에 대하여 각 클래스 별로 소속 여부를 판별하는 분류를 수행한다.
  - 각 분류를 통해서 판별된 결과를 기반으로 가장 많은 결과 값을 획득한 클래스를 최종 결과로 선택한다.

예) 클래스가 4종류일 때 (A, B, C, D)

선형 SVM

- 선형 SVM의 다중 클래스 분류
  - SVM은 기본적으로 이진 분류기이지만, OvO 전략을 적용하여 다항 분류를 수행할 수 있다.
  - 별도의 설정은 필요 없으며, 데이터가 다중 클래스로 구성되어 있으면 자동적으로 OvO 방식으로 분류가 수행된다.
    - 아래의 경우, y는 'setosa', 'versicolor', 'virginica'의 3종이 존재하므로 다중 클래스이다.

```
1  import sklearn.datasets as d
2
3  iris = d.load_iris()
4  X = iris.data[:, (2, 3)]
5  y = iris.target
```

선형 SVM

- 선형 SVM의 다중 클래스 분류
  - 이후의 과정은 이진 분류의 경우와 동일하게 진행하면 된다.

```
1  import sklearn.model_selection as ms
2  import sklearn.svm as svm
3  import sklearn.metrics as mt
4
5  X_train, X_test, y_train, y_test = ₩
6  ms.train_test_split(X, y, test_size=0.3, random_state=0)
7
8  clf = svm.SVC(kernel="linear").fit(X_train, y_train)
9
10 y_pred = clf.predict(X_test)
11
12 score = mt.accuracy_score(y_test, y_pred)
13 print("정확도:", round(score, 3))
```

```
정확도: 0.978
```

선형 SVM

- 선형 SVM의 다중 클래스 분류
  - 붓꽃 데이터의 다중 클래스 분류를 수행한 결과이다.