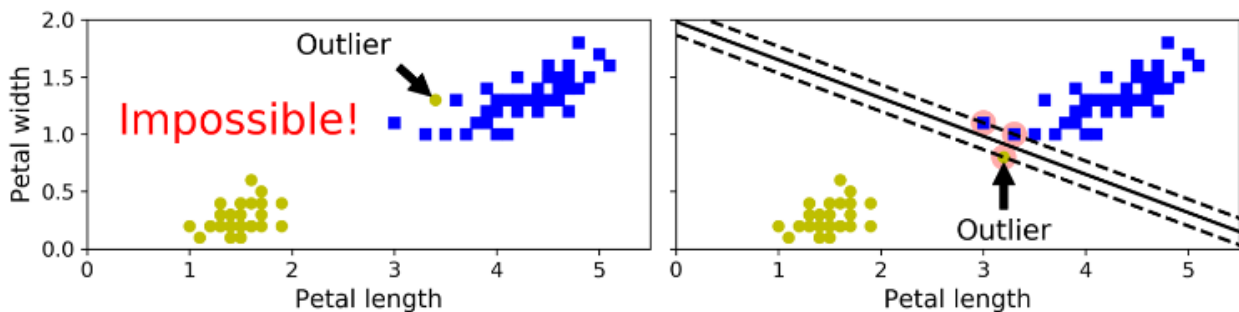


## طبقه بندی حاشیه سخت

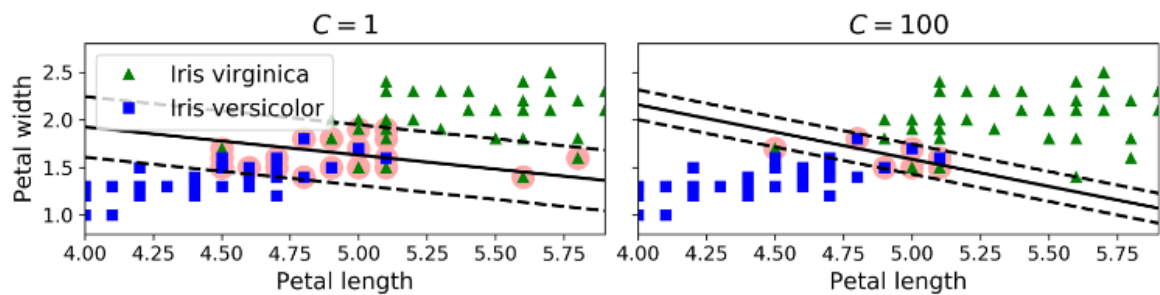
اگره تحمل کنیم که تمام نمونه ها باید خارج از فضای خالی میانی باشند و حتما باید در سمت مناسب قرار بگیرند، بهش میگویند Hard Margin Classification (طبقه بندی حاشیه سخت). این روش دو مشکل اساسی دارد. اول اینکه فقط برای داده‌هایی کار می‌کند که به صورت خطی جدا می‌شوند. دوم اینکه به داده‌های پرت به شدت حساس هست. شکلی که پایین می‌بینید دیتاست رو نشون میده که در سمت چپ یک داده پرت وجود داره و پیدا کردن یک Hard Margin برای اون غیر ممکن هست.



## طبقه بندی حاشیه نرم

برای جلوگیری از این مشکلات، از یک مدل انعطاف پذیر تر استفاده می‌کنیم. هدف ما پیدا کردن یک تعادل مناسب بین بیشترین مقدار فضای خالی میانی و محدود کردن Margin Violations هست (نقض حاشیه یعنی قرار گرفتن نمونه‌ها در فضای میانی یا در سمت اشتباه) به این کار میگویند Soft Margin Classification.

وقتی مدل‌های SVM رو با استفاده از Scikit-Learn می‌سازیم، یک سری هایپرپارامترها رو می‌تونیم تنظیم کنیم. یکی از این هایپرپارامترها  $C$  نام داره. اگر مقدار اون کم باشه، حاصل مدل سمت چپ شکل پایین میشه. با مقدار زیاد به شکل سمت راست می‌رسیم. نقض حاشیه اتفاق خوبی نیست و بهتره تا جایی که امکان داره، کمتر اتفاق بیفته. اگرچه، در این مورد، درسته که مدل سمت چپ نقض حاشیه‌های زیادی داره اما احتمالا بهتر عمومی‌سازی می‌کنه. در واقع هایپرپارامتر  $C$  مقدار مجازات مدل برای هر نمونه‌ای رو که اشتباه طبقه بندی میکنه، تعیین میکنه. هر چقدر  $C$  کمتر باشه، حاشیه نرم تر میشه.



اگر مدل شما **Overfit** شده باشه، می تونید با استفاده از هایپرپارامتر **C** اون رو **Regularize** کنید

مقدار **C** بزرگ: سوگیری کم، واریانس زیاد

مقدار **C** کوچک: سوگیری زیاد، واریانس کم

پس نتیجه می گیریم هر چه مقدار **C** بزرگتر باشد حاشیه سخت تر خواهد بود (**Hard Margin**) و بالعکس

هر چه مقدار **C** کوچکتر باشد حاشیه نرم تر خواهد بود (**Soft Margin**).

حال با استفاده از ایده بالا مسئله را حل می کنیم :

**# Create a linear SVM classifier with Hard Margin Classification**

```
svclassifier = svm.SVC(kernel='linear', C = 1000)
```

```
svclassifier.fit(x_train, y)
```

```
y_train, y_test = tts( y , test_size = 0.4965, random_state = 0)
```

```
pred = svclassifier.predict(x_test)
```

```
print(confusion_matrix(y_test,pred))
```

```
print(classification_report(y_test,pred))
```

```
[[ 0  0  0 252]
```

```
 [ 0  0  0 247]
```

```
 [ 0  0  0 245]
```

```
 [ 0  0  0 244]]
```

```
precision recall f1-score support
```

```
0    0.00    0.00    0.00    252
```

```
1    0.00    0.00    0.00    247
```

```
2    0.00    0.00    0.00    245
```

```
3    0.25    1.00    0.40    244
```

```
accuracy                0.25    988
```

```
macro avg    0.06    0.25    0.10    988
```

```
weighted avg    0.06    0.25    0.10    988
```

# Create a linear SVM classifier with soft Margin Classification

```
svclassifier = svm.SVC(kernel='linear' , C = 0.1)
```

```
svclassifier.fit(x_train, y)
```

```
y_train, y_test = tts( y , test_size = 0.4965, random_state = 0)
```

```
pred = svclassifier.predict(x_test)
```

```
print(confusion_matrix(y_test,pred))
```

```
print(classification_report(y_test,pred))
```

```
[[ 0  0  0 252]
```

```
 [ 0  0  0 247]
```

```
 [ 0  0  0 245]
```

```
 [ 0  0  0 244]]
```

```
precision recall f1-score support
```

```
0    0.00    0.00    0.00    252
```

```
1    0.00    0.00    0.00    247
```

```
2    0.00    0.00    0.00    245
```

```
3    0.25    1.00    0.40    244
```

```
accuracy                0.25    988
```

```
macro avg    0.06    0.25    0.10    988
```

```
weighted avg    0.06    0.25    0.10    988
```

همان طور که ملاحظه می کنید در این مثال در هر دو حالت (حاشیه نرم و حاشیه سخت) جواب های یکسان به دست آمد.