```
from sklearn.svm import SVC
import matplotlib.pyplot as plt
import numpy as np
```

```
tpts = np.array([[1,2], [1,5], [2,2], [3,3.5], [7,5], [9,4], [8,2], [4,8]]) #training_points
labels = [1, 1, 1, 1, 0, 0, 0, 1]
```

```
tpts
```

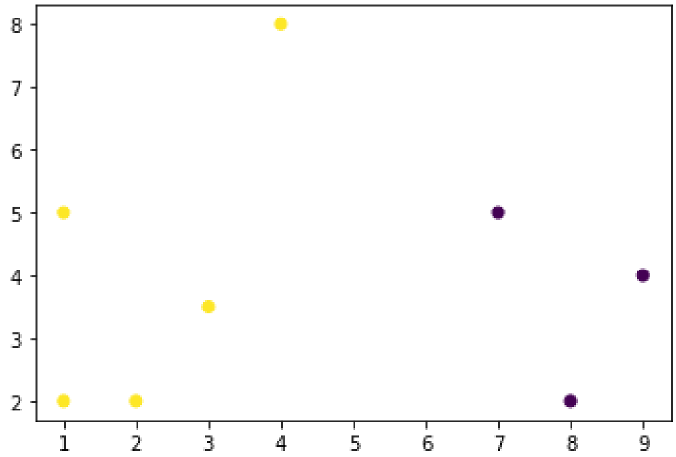
```
array([[1. , 2. ],
       [1. , 5. ],
       [2. , 2. ],
       [3. , 3.5],
       [7. , 5. ],
       [9. , 4. ],
       [8. , 2. ],
       [4. , 8. ]])
```

```
tpts.shape
```

```
(8, 2)
```

```
plt.scatter(tpts[:,0], tpts[:, 1], c=labels)
```

```
<matplotlib.collections.PathCollection at 0x7ffa7
```



```
clf = SVC(kernel='linear') #classifier
```

```
%%time
clf.fit(tpts, labels)
```

```
CPU times: user 1.22 ms, sys: 0 ns, total: 1.22 m
Wall time: 2.67 ms
SVC(C=1.0, break_ties=False, cache_size=200, clas
    decision_function_shape='ovr', degree=3, gamm
    max_iter=-1, probability=False, random_state=
    tol=0.001, verbose=False)
```

## 테스트해보기

```
predl = clf.predict([[3, 3]])
print(predl[0])
```

```
predl = clf.predict([[7, 4]])
print(predl[0])
```

```
predl = clf.predict([[7, 4.1]])
print(predl[0])
```

## 그래프 그려보기

```
clf.coef_
```

```
clf.intercept_
```

```
def getXY(clf):
  w = clf.coef_[0]
  i = clf.intercept_[0]
  print('weights = ', w)
  print('intercept = ', i)

  a = -w[0] / w[1]
  b = -i / w[1]
  x = np.linspace(1, 9)


  y = a * x + b

  return x, y

x, y = getXY(clf)
```

이민아 완료
2020년 4월 22일
(2020년 4월 22일 수정됨)

기울기 구현하는 방법 중의 일부

이민아
2020년 4월 22일

*완료됨으로 표시됨*

이민아
2020년 4월 22일

*다시 열림*

이민아 완료
2020년 4월 22일
(2020년 4월 22일 수정됨)

b에 대해 직선이 y축을 지나는 점을 표현하는 것

이민아
2020년 4월 22일

*완료됨으로 표시됨*

이민아
2020년 4월 22일

```
x
```

*다시 열림*

```
y
```

```
plt.scatter(tpts[:,0], tpts[:, 1], c=labels)
plt.plot(x, y, '-')
```