**Lab. Tutorial Module 3**

**In [1] :**

%matplotlib inline

import numpy as np

import matplotlib.pyplot as plt

from scipy import stats

# use seaborn plotting defaults

import seaborn as sns; sns.set()

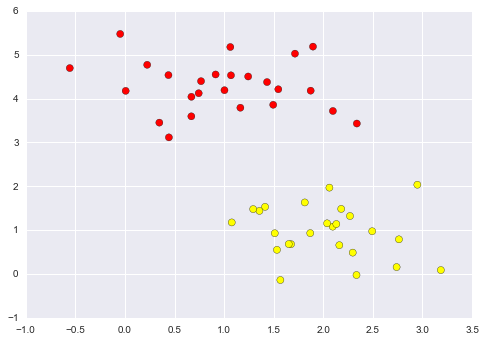
**In[2]:**

from sklearn.datasets.samples\_generator import make\_blobs

X, y = make\_blobs(n\_samples=50, centers=2,

random\_state=0, cluster\_std=0.60)

plt.scatter(X[:, 0], X[:, 1], c=y, s=50, cmap='autumn');



**In [3]:**

xfit = np.linspace(-1, 3.5)

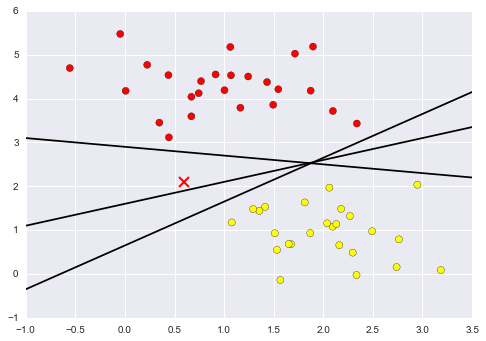
plt.scatter(X[:, 0], X[:, 1], c=y, s=50, cmap='autumn')

plt.plot([0.6], [2.1], 'x', color='red', markeredgewidth=2, markersize=10)

for m, b in [(1, 0.65), (0.5, 1.6), (-0.2, 2.9)]:

plt.plot(xfit, m \* xfit + b, '-k')

plt.xlim(-1, 3.5);



**In[4]:**

xfit = np.linspace(-1, 3.5)

plt.scatter(X[:, 0], X[:, 1], c=y, s=50, cmap='autumn')

for m, b, d in [(1, 0.65, 0.33), (0.5, 1.6, 0.55), (-0.2, 2.9, 0.2)]:

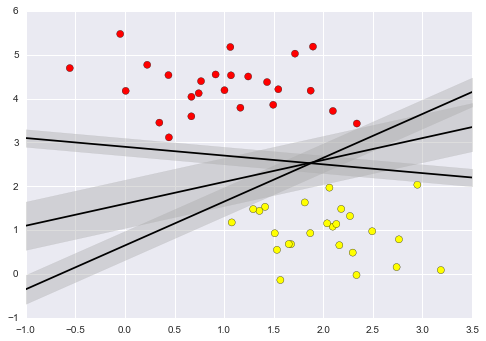
yfit = m \* xfit + b

plt.plot(xfit, yfit, '-k')

plt.fill\_between(xfit, yfit - d, yfit + d, edgecolor='none',

color='#AAAAAA', alpha=0.4)

plt.xlim(-1, 3.5);



**In[5]:**

from sklearn.svm import SVC # "Support vector classifier"

model = SVC(kernel='linear', C=1E10)

model.fit(X, y)

**Out [5]:**

SVC(C=10000000000.0, cache\_size=200, class\_weight=None, coef0=0.0,

decision\_function\_shape=None, degree=3, gamma='auto', kernel='linear',

max\_iter=-1, probability=False, random\_state=None, shrinking=True,

tol=0.001, verbose=False)

**In[6]:**

def plot\_svc\_decision\_function(model, ax=None, plot\_support=True):

"""Plot the decision function for a 2D SVC"""

if ax is None:

ax = plt.gca()

xlim = ax.get\_xlim()

ylim = ax.get\_ylim()

# create grid to evaluate model

x = np.linspace(xlim[0], xlim[1], 30)

y = np.linspace(ylim[0], ylim[1], 30)

Y, X = np.meshgrid(y, x)

xy = np.vstack([X.ravel(), Y.ravel()]).T

P = model.decision\_function(xy).reshape(X.shape)

# plot decision boundary and margins

ax.contour(X, Y, P, colors='k',

levels=[-1, 0, 1], alpha=0.5,

linestyles=['--', '-', '--'])

# plot support vectors

if plot\_support:

ax.scatter(model.support\_vectors\_[:, 0],

model.support\_vectors\_[:, 1],

s=300, linewidth=1, facecolors='none');

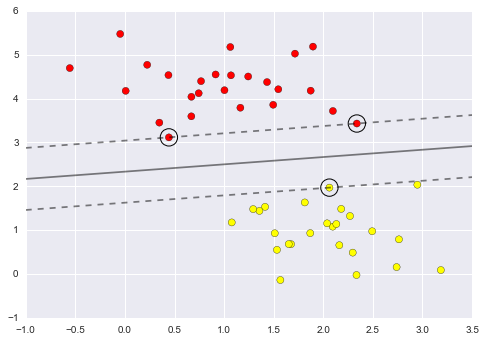
ax.set\_xlim(xlim)

ax.set\_ylim(ylim)

**In[7]:**

plt.scatter(X[:, 0], X[:, 1], c=y, s=50, cmap='autumn')

plot\_svc\_decision\_function(model);



**In[8]:**

model.support\_vectors\_

**Out[8]:**

array([[ 0.44359863, 3.11530945],

[ 2.33812285, 3.43116792],

[ 2.06156753, 1.96918596]])

**In[9]:**

def plot\_svm(N=10, ax=None):

X, y = make\_blobs(n\_samples=200, centers=2,

random\_state=0, cluster\_std=0.60)

X = X[:N]

y = y[:N]

model = SVC(kernel='linear', C=1E10)

model.fit(X, y)

ax = ax or plt.gca()

ax.scatter(X[:, 0], X[:, 1], c=y, s=50, cmap='autumn')

ax.set\_xlim(-1, 4)

ax.set\_ylim(-1, 6)

plot\_svc\_decision\_function(model, ax)

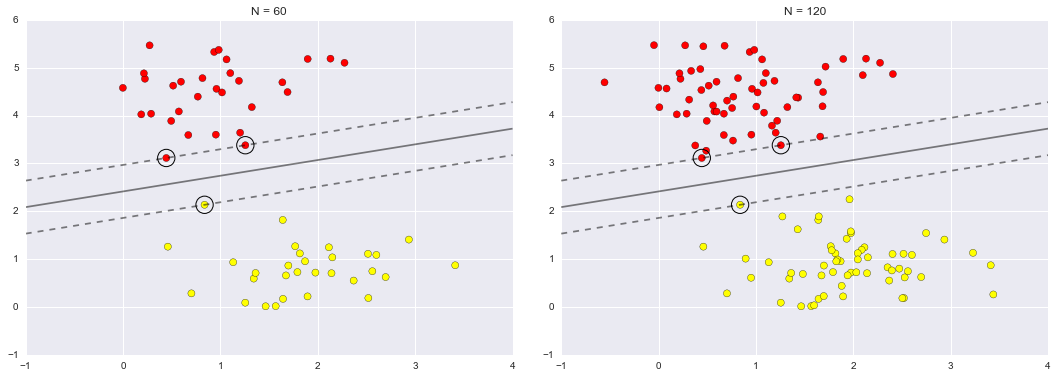
fig, ax = plt.subplots(1, 2, figsize=(16, 6))

fig.subplots\_adjust(left=0.0625, right=0.95, wspace=0.1)

for axi, N in zip(ax, [60, 120]):

plot\_svm(N, axi)

axi.set\_title('N = {0}'.format(N))



**In[10]:**

from ipywidgets import interact, fixed

interact(plot\_svm, N=[10, 200], ax=fixed(None));

