**SVM with polynomial kernel**

from sklearn.datasets import load\_iris

import matplotlib.pyplot as plt

from sklearn.svm import SVC

import numpy as np

import pandas as pd

iris=load\_iris()

X = iris.data[:,:2]

Y = iris.target

x\_min, x\_max = X[:, 0].min() - .5, X[:, 0].max() + .5

y\_min, y\_max = X[:, 1].min() - .5, X[:, 1].max() + .5

# Plot the training points

plt.scatter(X[:, 0], X[:, 1], c=Y,cmap=plt.cm.Set1,edgecolor='k')

plt.xlabel('Sepal length')

plt.ylabel('Sepal width')

plt.title('Scatter plot')

plt.show()

 # Polynomial kernel SVM

svc =SVC(kernel='poly',degree=4, C=1,gamma='auto').fit(X, Y)

h = (x\_max / x\_min)/100

xx, yy = np.meshgrid(np.arange(x\_min, x\_max, h),np.arange(y\_min, y\_max, h))

Z = svc.predict(np.c\_[xx.ravel(), yy.ravel()])

Z = Z.reshape(xx.shape)

plt.contourf(xx, yy, Z,cmap=plt.cm.Set1,alpha=.5)

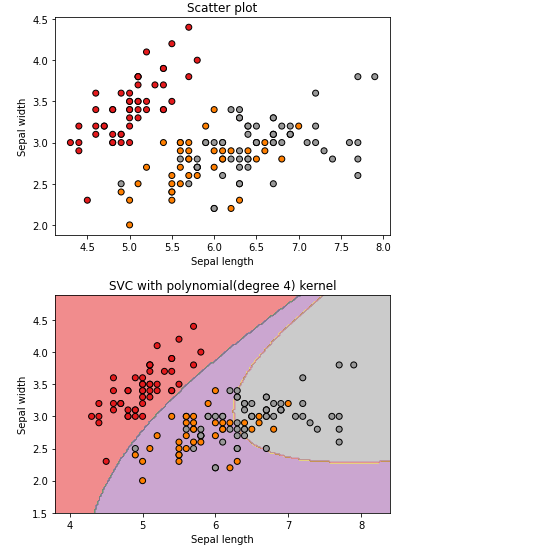
plt.scatter(X[:, 0], X[:, 1], c=Y,cmap=plt.cm.Set1,edgecolor='k')

plt.title('SVC with polynomial(degree 4) kernel')

plt.xlabel('Sepal length')

plt.ylabel('Sepal width')

plt.show()



**SVM with linear kernel**

from sklearn.datasets import load\_iris

import matplotlib.pyplot as plt

from sklearn.svm import SVC

import numpy as np

import pandas as pd

iris=load\_iris()

X = iris.data[:,:2]

Y = iris.target

x\_min, x\_max = X[:, 0].min() - .5, X[:, 0].max() + .5

y\_min, y\_max = X[:, 1].min() - .5, X[:, 1].max() + .5

# Plot the training points

plt.scatter(X[:, 0], X[:, 1], c=Y,cmap=plt.cm.Set1,edgecolor='k')

plt.xlabel('Sepal length')

plt.ylabel('Sepal width')

plt.title('Scatter plot')

plt.show()

 # Linear SVM

svc =SVC(kernel='linear', C=1,gamma='auto').fit(X, Y)

h = (x\_max / x\_min)/100

xx, yy = np.meshgrid(np.arange(x\_min, x\_max, h),np.arange(y\_min, y\_max, h))

Z = svc.predict(np.c\_[xx.ravel(), yy.ravel()])

Z = Z.reshape(xx.shape)

plt.contourf(xx, yy, Z,cmap=plt.cm.Set1,alpha=.5)

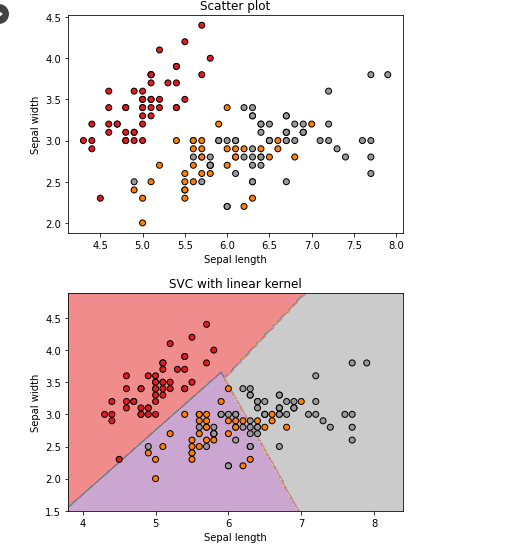
plt.scatter(X[:, 0], X[:, 1], c=Y,cmap=plt.cm.Set1,edgecolor='k')

plt.title('SVC with linear kernel')

plt.xlabel('Sepal length')

plt.ylabel('Sepal width')

plt.show()



**SVM with rbf kernel**

from sklearn.datasets import load\_iris

import matplotlib.pyplot as plt

from sklearn.svm import SVC

import numpy as np

import pandas as pd

iris=load\_iris()

X = iris.data[:,:2]

Y = iris.target

x\_min, x\_max = X[:, 0].min() - .5, X[:, 0].max() + .5

y\_min, y\_max = X[:, 1].min() - .5, X[:, 1].max() + .5

# Plot the training points

plt.scatter(X[:, 0], X[:, 1], c=Y,cmap=plt.cm.Set1,edgecolor='k')

plt.xlabel('Sepal length')

plt.ylabel('Sepal width')

plt.title('Scatter plot')

plt.show()

 # Rbf kernel SVM

svc =SVC(kernel='rbf', C=1,gamma=1).fit(X, Y)

h = (x\_max / x\_min)/100

xx, yy = np.meshgrid(np.arange(x\_min, x\_max, h),np.arange(y\_min, y\_max, h))

Z = svc.predict(np.c\_[xx.ravel(), yy.ravel()])

Z = Z.reshape(xx.shape)

plt.contourf(xx, yy, Z,cmap=plt.cm.Set1,alpha=.7)

plt.scatter(X[:, 0], X[:, 1], c=Y,cmap=plt.cm.Set1,edgecolor='k')

plt.title('SVC with rbf kernel with C=1 and gamma=1')

plt.xlabel('Sepal length')

plt.ylabel('Sepal width')

plt.show()

