Karanjot Singh

In [153]:

```
import pandas as pd
import numpy as np
import seaborn as sb
from sklearn import datasets
import matplotlib.pyplot as plt
sb.set()
from sklearn.model_selection import train_test_split
```

In [154]:

```
df = sb.load_dataset('iris')
iris = datasets.load_iris()
df.head()
```

Out[154]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

In [155]:

```
x = iris.data[:,:2]
y = iris.target
x_train,x_test,y_train,y_test = train_test_split(x,y,train_size = 0.7)
#print(x)
```

In [156]:

```
from sklearn.svm import SVC
svm_clf = SVC(gamma = 0.0000001,kernel = 'rbf',probability = True)
svm_clf.fit(x,y)

#SVC(C=1.0,cache_size=200, class_weight=None, coef0=0.0,decision_function_shape='ovr',
 degree=3, gamma=1e-07, kernel='rbf',max_iter=-1, probability=True, random_state=None,
 shrinking=True, tol=0.001,verbose=False)
#SVC(C=1.0, kernel='rbf', degree=3, gamma=0.0, coef0=0.0, shrinking=True, probability=F
alse,tol=0.001, cache_size=200, class_weight=None, verbose=False, max_iter=-1, random_s
tate=None)
pp = svm_clf.predict(x_test)
svm_clf.score(x_test, y_test)
```

Out[156]:

```
0.8222222222222222
```

In [157]:

```
svm_clf1 = SVC(gamma = 10,kernel = 'rbf',probability = True)
svm_clf1.fit(x,y)

#SVC(C=1.0,cache_size=200, class_weight=None, coef0=0.0,decision_function_shape='ovr',
 degree=3, gamma=1e-07, kernel='rbf',max_iter=-1, probability=True, random_state=None,
 shrinking=True, tol=0.001,verbose=False)
#SVC(C=1.0, kernel='rbf', degree=3, gamma=0.0, coef0=0.0, shrinking=True, probability=F
alse,tol=0.001, cache_size=200, class_weight=None, verbose=False, max_iter=-1, random_s
tate=None)
pp = svm_clf1.predict(x_test)
svm_clf1.score(x_test, y_test)
```

Out[157]:

```
0.8888888888888888
```

In [158]:

```
from sklearn.svm import SVC
svm_clf2 = SVC(gamma  = 100,kernel = 'rbf',probability = True)
svm_clf2.fit(x,y)

#SVC(C=1.0,cache_size=200, class_weight=None, coef0=0.0,decision_function_shape='ovr',
 degree=3, gamma=1e-07, kernel='rbf',max_iter=-1, probability=True, random_state=None,
 shrinking=True, tol=0.001,verbose=False)
#SVC(C=1.0, kernel='rbf', degree=3, gamma=0.0, coef0=0.0, shrinking=True, probability=F
alse,tol=0.001, cache_size=200, class_weight=None, verbose=False, max_iter=-1, random_s
tate=None)
pp = svm_clf2.predict(x_test)
svm_clf2.score(x_test, y_test)
```

Out[158]:

```
0.9555555555555556
```

In [159]:

```
x_min,x_max = x[:,0].min()-1,x[:,0].max()+1
y_min,y_max = x[:,1].min()-1,x[:,1].max()+1
print(x_min,x_max)
print(y_min,y_max)
```

```
3.3 8.9
1.0 5.4
```

In [161]:

```
xx,yy = np.meshgrid(np.arange(x_min,x_max,0.1),np.arange(y_min,y_max,0.1))
plt.subplots(2,2,sharex = 'col',sharey = 'row',figsize = (10,8))
title = ['SVM_0.0','SVM_10','SVM_100']
for i,clf in enumerate((svm_clf,svm_clf1,svm_clf2)):
    plt.subplot(2,2,i+1)
    plt.subplots_adjust(wspace=0.4,hspace=0.4)
    z = clf.predict(np.c_[xx.ravel(),yy.ravel()])
    z = z.reshape(xx.shape)
    plt.contourf(xx,yy,z,cmap = plt.cm.coolwarm,alpha = 0.8)
    plt.scatter(x[:, 0], x[:, 1], c=y, cmap=plt.cm.coolwarm)
    plt.xlabel('Sepal length')
    plt.ylabel('Sepal width')
    plt.xlim(xx.min(), xx.max())
    plt.ylim(yy.min(), yy.max())
    plt.xticks(())
    plt.yticks(())
    plt.title(title[i])
plt.show()
```