Lab12

September 29, 2022

[ ]:

```python
import numpy as np
import matplotlib.pyplot as plt
from sklearn import datasets, svm
```

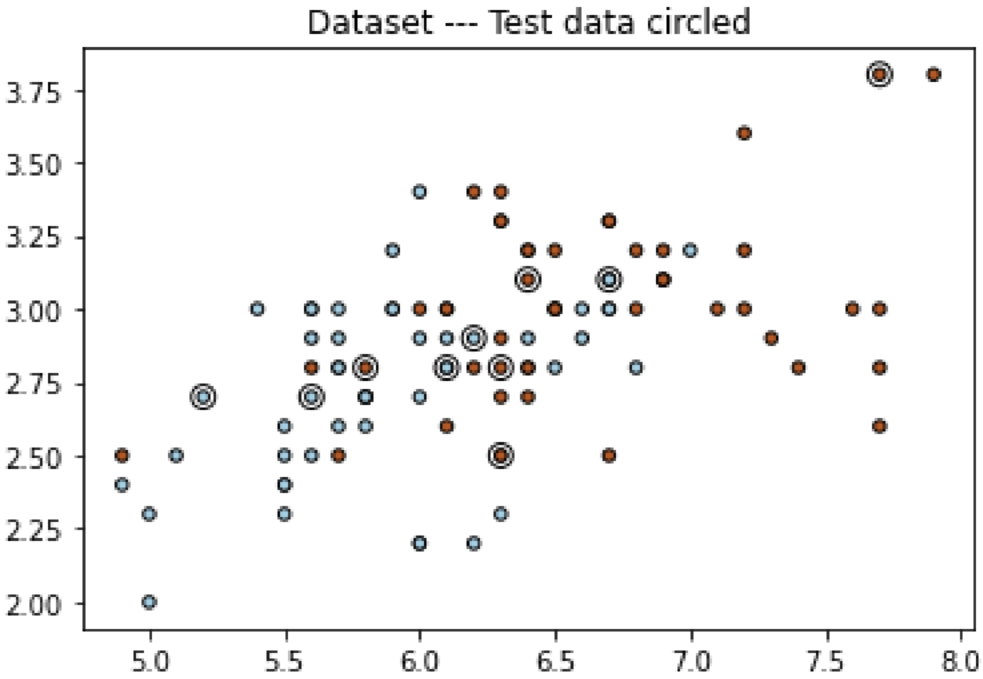
[ ]:

```python
#Plotting function
def plot_svm_kernels(clf, X, y, X_test):
    #Plot
    plt.figure()
    plt.clf()
    plt.scatter(X[:, 0], X[:, 1], c=y, zorder=10, cmap=plt.cm.Paired,
                edgecolor='k', s=20)

    # Circle out the test data
    plt.scatter(X_test[:, 0], X_test[:, 1], s=80, facecolors='none',
                zorder=10, edgecolor='k')

    plt.axis('tight')
    x_min = X[:, 0].min()
    x_max = X[:, 0].max()
    y_min = X[:, 1].min()
    y_max = X[:, 1].max()

    XX, YY = np.mgrid[x_min:x_max:200j, y_min:y_max:200j]
    Z = clf.decision_function(np.c_[XX.ravel(), YY.ravel()])

    # Put the result into a color plot
    Z = Z.reshape(XX.shape)
    plt.pcolormesh(XX, YY, Z > 0, cmap=plt.cm.Paired)
    plt.contour(XX, YY, Z, colors=['k', 'k', 'k'],
                linestyles=['--', '-', '--'], levels=[-.5, 0, .5])

    plt.title(clf.kernel)
    plt.show()
```

0.1 Radial Basis Function

[ ]:

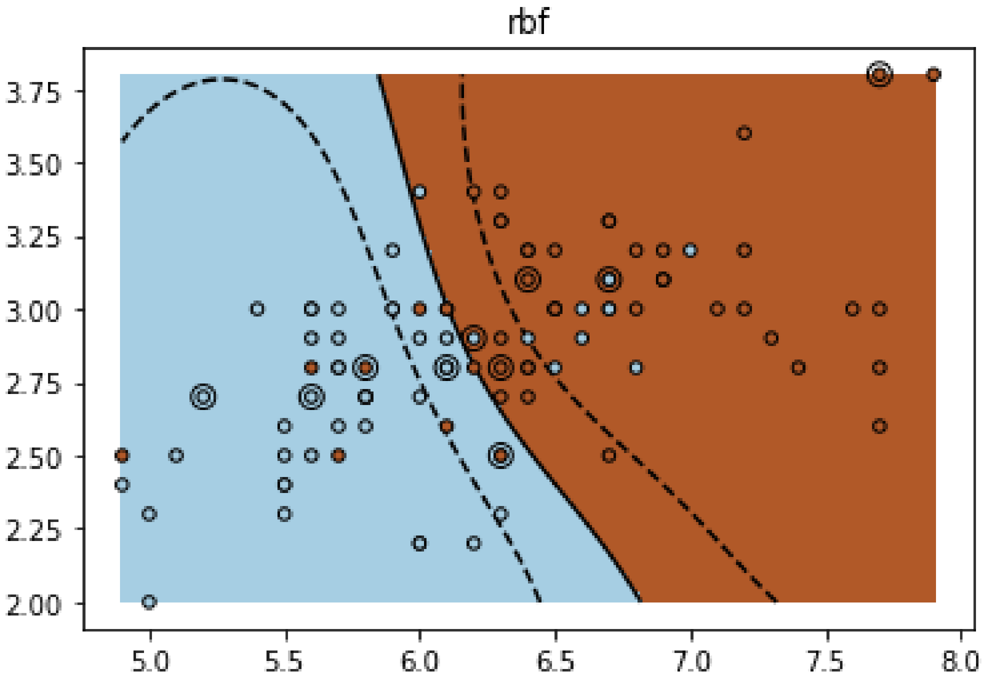
```
iris = datasets.load_iris()
X = iris.data
y = iris.target

X = X[y != 0, :2]
y = y[y != 0]

n_sample = len(X)

np.random.seed(0)
order = np.random.permutation(n_sample)
X = X[order]
y = y[order].astype(float)

X_train = X[:int(.9 * n_sample)]
y_train = y[:int(.9 * n_sample)]
X_test = X[int(.9 * n_sample):]
y_test = y[int(.9 * n_sample):]


#Plotting function
#Plot
plt.figure()
plt.scatter(X[:, 0], X[:, 1], c=y, zorder=10, cmap=plt.cm.Paired,
                edgecolor='k', s=20)

# Circle out the test data
plt.scatter(X_test[:, 0], X_test[:, 1], s=80, facecolors='none',
            zorder=10, edgecolor='k')

plt.axis('tight')
x_min = X[:, 0].min()
x_max = X[:, 0].max()
y_min = X[:, 1].min()
y_max = X[:, 1].max()

plt.title('Dataset --- Test data circled')
plt.show()
```



[ ]:

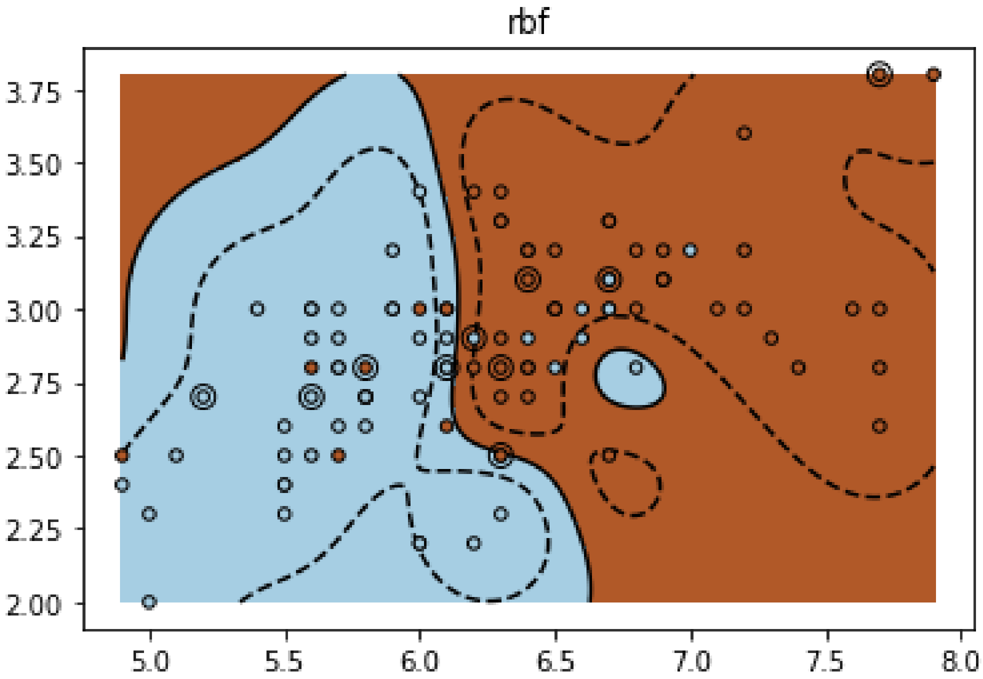
```
#Fit model
clf = svm.SVC(kernel='rbf', gamma=1)
clf.fit(X_train, y_train)

plot_svm_kernels(clf, X, y, X_test)
```

```
<ipython-input-2-aec1f050970f>:24: MatplotlibDeprecationWarning: shading='flat'
when X and Y have the same dimensions as C is deprecated since 3.3.  Either
specify the corners of the quadrilaterals with X and Y, or pass shading='auto',
'nearest' or 'gouraud', or set rcParams['pcolor.shading'].  This will become an
error two minor releases later.
  plt.pcolormesh(XX, YY, Z > 0, cmap=plt.cm.Paired)
```



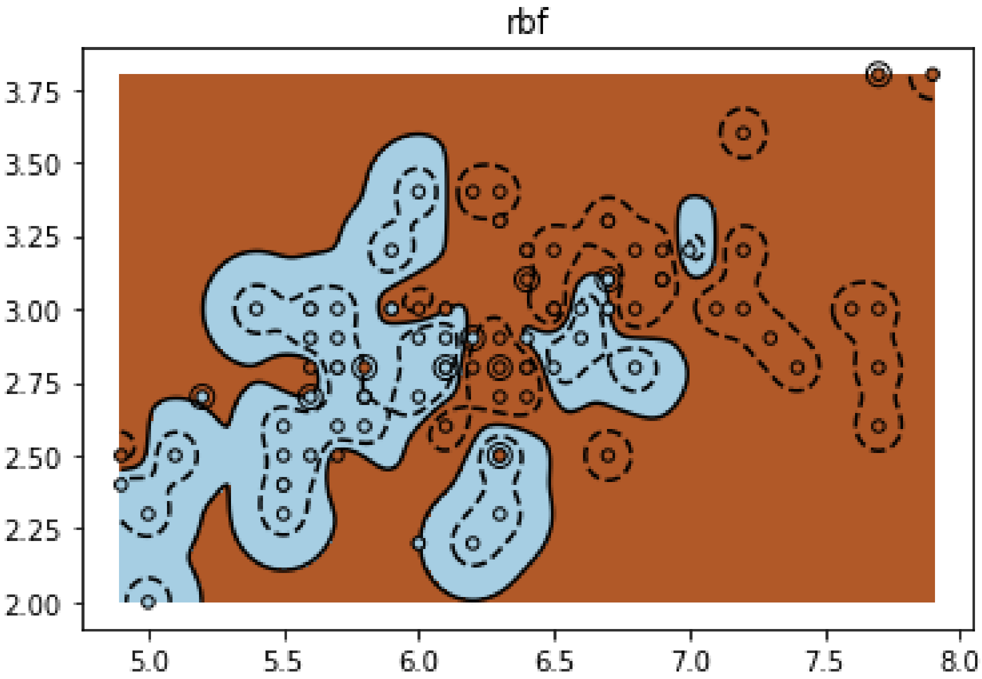
```
[ ]: #Fit model
     clf = svm.SVC(kernel='rbf', gamma=10)
     clf.fit(X_train, y_train)

     plot_svm_kernels(clf, X, y, X_test)
```

```
<ipython-input-2-aec1f050970f>:24: MatplotlibDeprecationWarning: shading='flat'
when X and Y have the same dimensions as C is deprecated since 3.3.  Either
specify the corners of the quadrilaterals with X and Y, or pass shading='auto',
'nearest' or 'gouraud', or set rcParams['pcolor.shading'].  This will become an
error two minor releases later.
  plt.pcolormesh(XX, YY, Z > 0, cmap=plt.cm.Paired)
```



```
[ ]: #Fit model
     clf = svm.SVC(kernel='rbf', gamma=100)
     clf.fit(X_train, y_train)

     plot_svm_kernels(clf, X, y, X_test)
```

```
<ipython-input-2-aec1f050970f>:24: MatplotlibDeprecationWarning: shading='flat'
when X and Y have the same dimensions as C is deprecated since 3.3.  Either
specify the corners of the quadrilaterals with X and Y, or pass shading='auto',
'nearest' or 'gouraud', or set rcParams['pcolor.shading'].  This will become an
error two minor releases later.
  plt.pcolormesh(XX, YY, Z > 0, cmap=plt.cm.Paired)
```

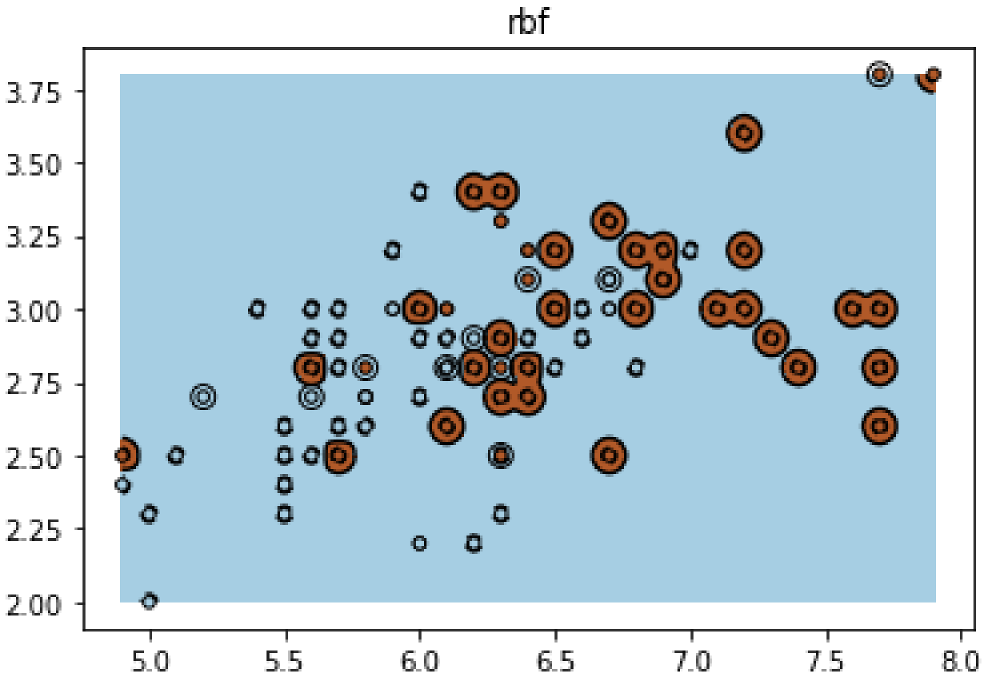


rbf

[ ]:

```
#Fit model
clf = svm.SVC(kernel='rbf', gamma=1000)
clf.fit(X_train, y_train)

plot_svm_kernels(clf, X, y, X_test)
```

```
<ipython-input-2-aec1f050970f>:24: MatplotlibDeprecationWarning: shading='flat'
when X and Y have the same dimensions as C is deprecated since 3.3.  Either
specify the corners of the quadrilaterals with X and Y, or pass shading='auto',
'nearest' or 'gouraud', or set rcParams['pcolor.shading'].  This will become an
error two minor releases later.
  plt.pcolormesh(XX, YY, Z > 0, cmap=plt.cm.Paired)
```



$$f(y) = \sum_{i=1}^{n} \alpha_i K(y, x_i)$$

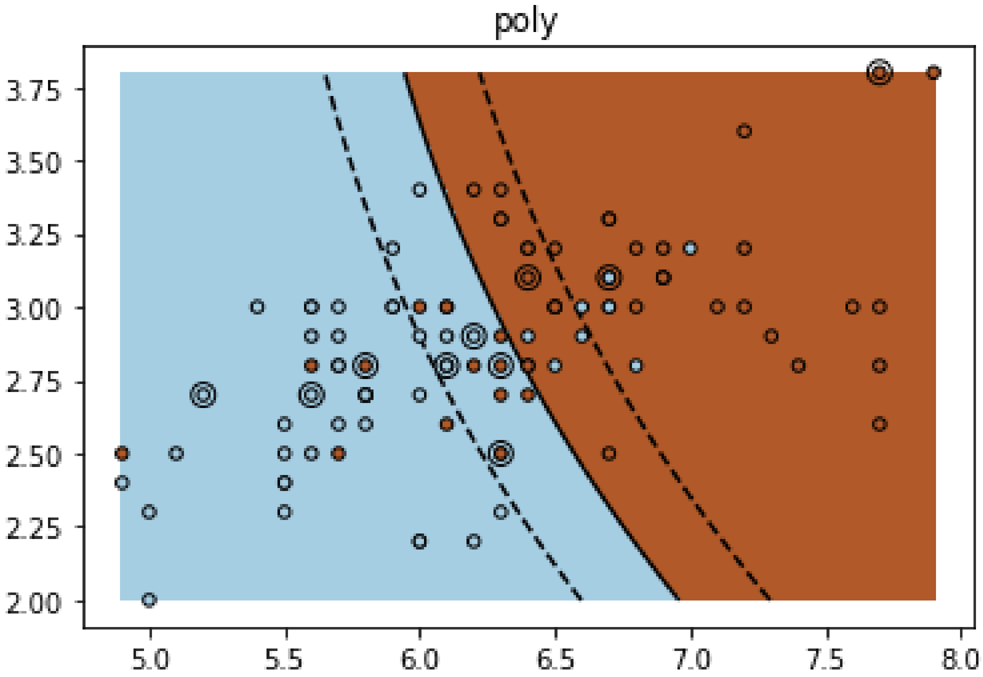
1 Polynomial kernels

[ ]:

```
#Fit model
for p in range(3,10):
    print('Fit with p =', p)
    clf = svm.SVC(kernel='poly', degree=p)
    clf.fit(X_train, y_train)

    plot_svm_kernels(clf, X, y, X_test)
```

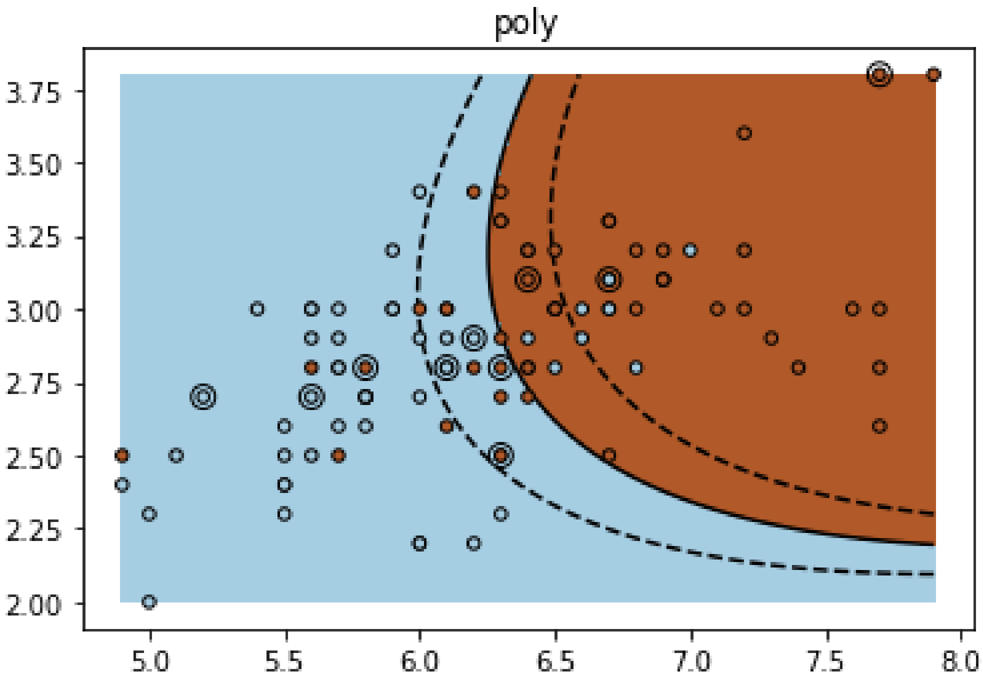
```
Fit with p = 3
```

```
<ipython-input-2-aec1f050970f>:24: MatplotlibDeprecationWarning: shading='flat'
when X and Y have the same dimensions as C is deprecated since 3.3.  Either
specify the corners of the quadrilaterals with X and Y, or pass shading='auto',
'nearest' or 'gouraud', or set rcParams['pcolor.shading'].  This will become an
error two minor releases later.
  plt.pcolormesh(XX, YY, Z > 0, cmap=plt.cm.Paired)
```



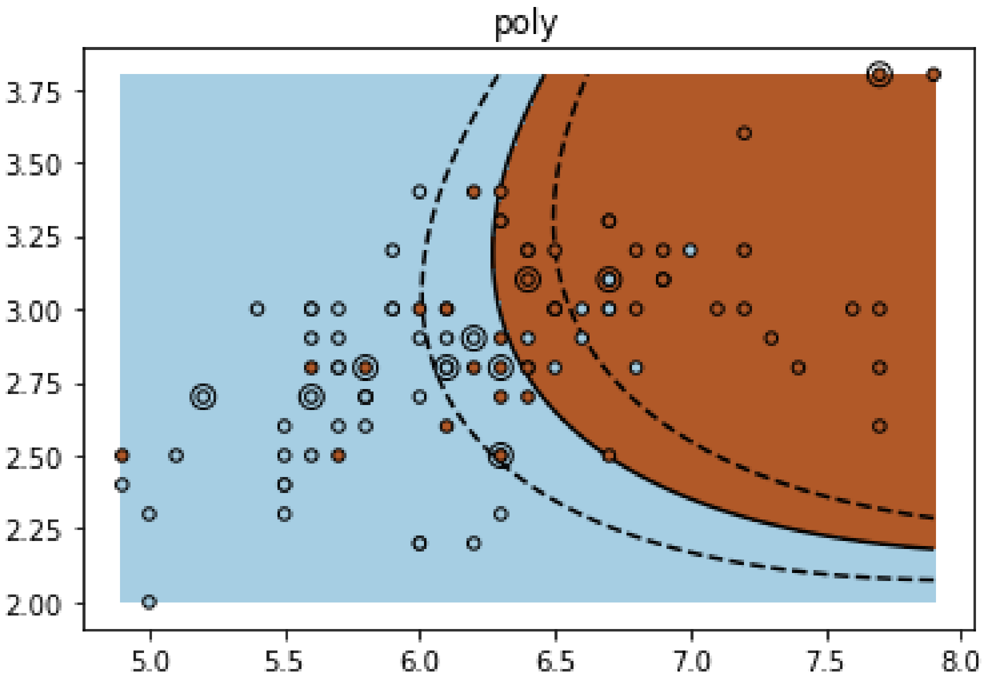
```
Fit with p = 4
```

```
<ipython-input-2-aec1f050970f>:24: MatplotlibDeprecationWarning: shading='flat'
when X and Y have the same dimensions as C is deprecated since 3.3.  Either
specify the corners of the quadrilaterals with X and Y, or pass shading='auto',
'nearest' or 'gouraud', or set rcParams['pcolor.shading'].  This will become an
error two minor releases later.
  plt.pcolormesh(XX, YY, Z > 0, cmap=plt.cm.Paired)
```



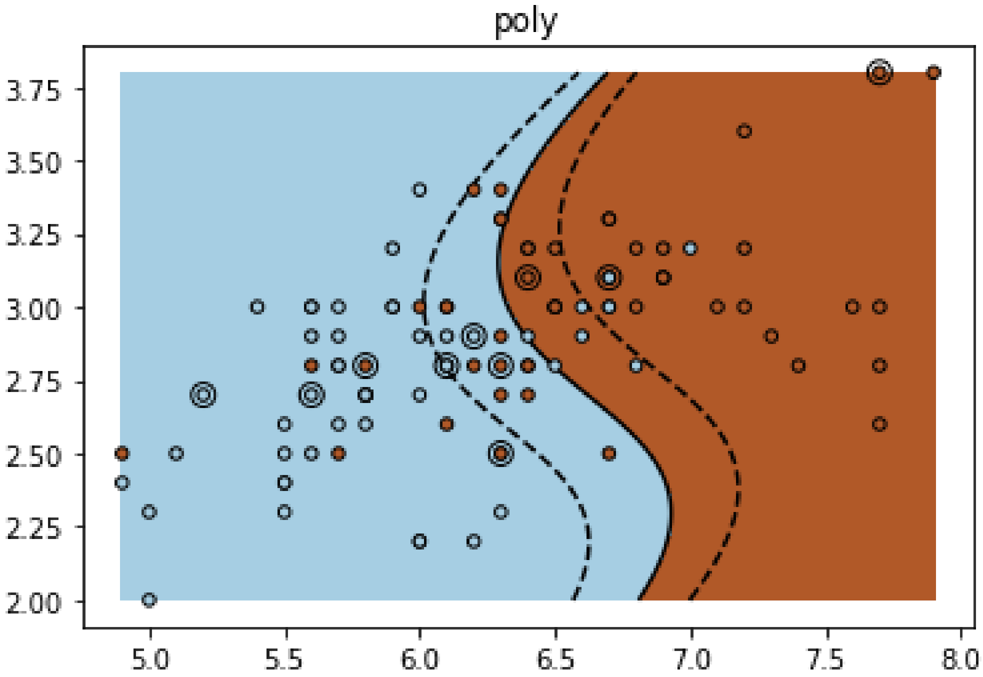
```
Fit with p = 5
```

```
<ipython-input-2-aec1f050970f>:24: MatplotlibDeprecationWarning: shading='flat'
when X and Y have the same dimensions as C is deprecated since 3.3.  Either
specify the corners of the quadrilaterals with X and Y, or pass shading='auto',
'nearest' or 'gouraud', or set rcParams['pcolor.shading'].  This will become an
error two minor releases later.
  plt.pcolormesh(XX, YY, Z > 0, cmap=plt.cm.Paired)
```



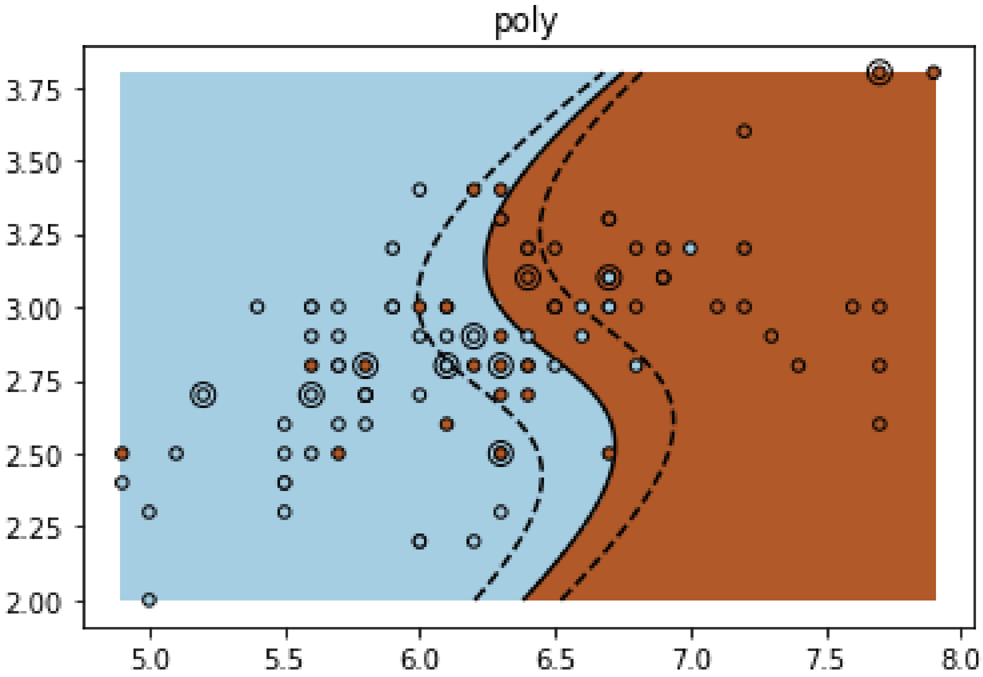
```
Fit with p = 6
```

```
<ipython-input-2-aec1f050970f>:24: MatplotlibDeprecationWarning: shading='flat'
when X and Y have the same dimensions as C is deprecated since 3.3.  Either
specify the corners of the quadrilaterals with X and Y, or pass shading='auto',
'nearest' or 'gouraud', or set rcParams['pcolor.shading'].  This will become an
error two minor releases later.
  plt.pcolormesh(XX, YY, Z > 0, cmap=plt.cm.Paired)
```



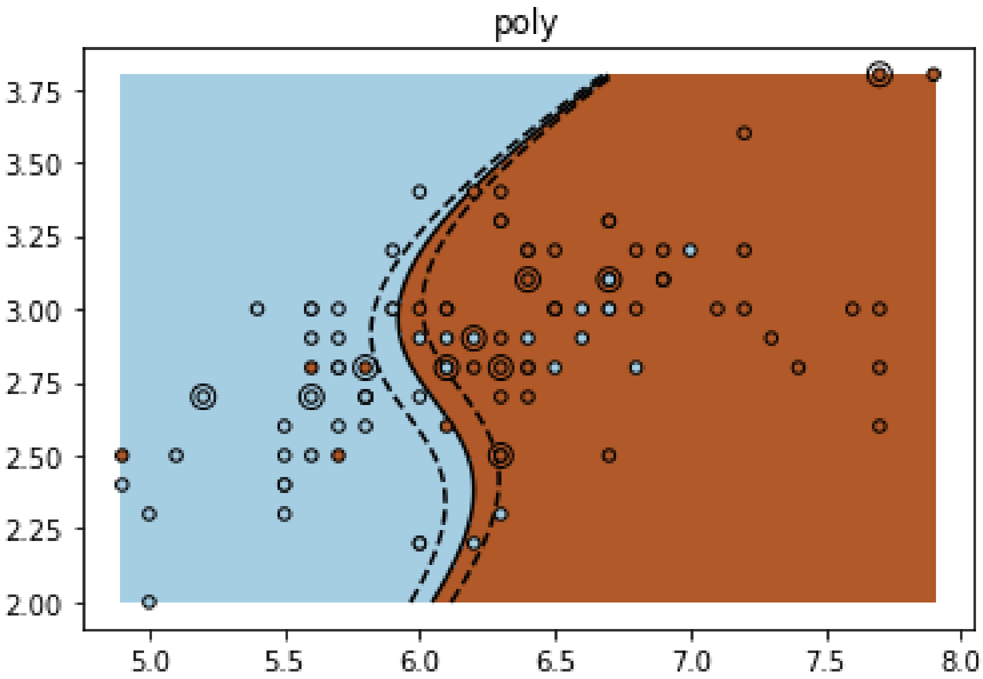
```
Fit with p = 7
```

```
<ipython-input-2-aec1f050970f>:24: MatplotlibDeprecationWarning: shading='flat'
when X and Y have the same dimensions as C is deprecated since 3.3.  Either
specify the corners of the quadrilaterals with X and Y, or pass shading='auto',
'nearest' or 'gouraud', or set rcParams['pcolor.shading'].  This will become an
error two minor releases later.
  plt.pcolormesh(XX, YY, Z > 0, cmap=plt.cm.Paired)
```

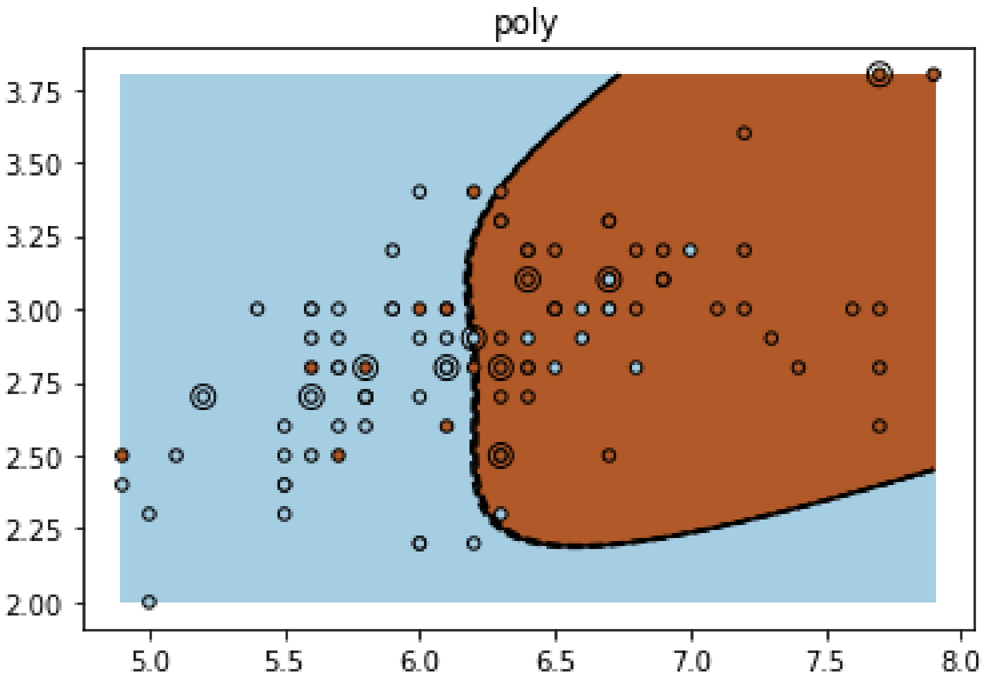


```
Fit with p = 8
```

```
<ipython-input-2-aec1f050970f>:24: MatplotlibDeprecationWarning: shading='flat'
when X and Y have the same dimensions as C is deprecated since 3.3.  Either
specify the corners of the quadrilaterals with X and Y, or pass shading='auto',
'nearest' or 'gouraud', or set rcParams['pcolor.shading'].  This will become an
error two minor releases later.
  plt.pcolormesh(XX, YY, Z > 0, cmap=plt.cm.Paired)
```



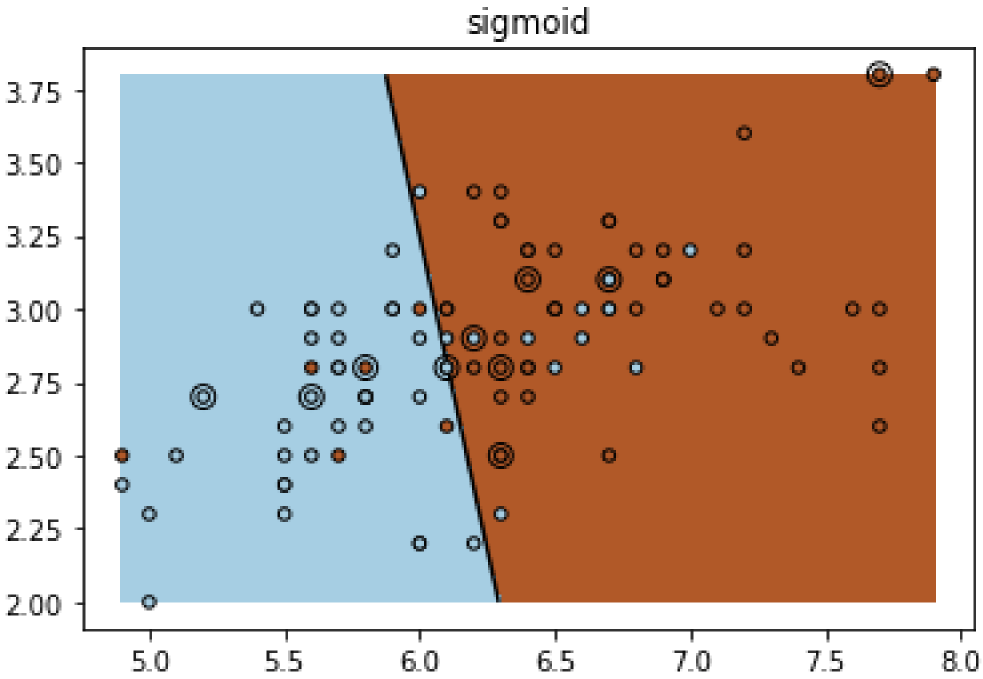
```
Fit with p = 9

<ipython-input-2-aec1f050970f>:24: MatplotlibDeprecationWarning: shading='flat'
when X and Y have the same dimensions as C is deprecated since 3.3.  Either
specify the corners of the quadrilaterals with X and Y, or pass shading='auto',
'nearest' or 'gouraud', or set rcParams['pcolor.shading'].  This will become an
error two minor releases later.
  plt.pcolormesh(XX, YY, Z > 0, cmap=plt.cm.Paired)
```



1.1 Sigmoid kernels

```
[ ]: #Fit model
     # for p in range(3,10):
     #     print('Fit with p =', p)
     clf = svm.SVC(kernel='sigmoid', gamma=.01)
     clf.fit(X_train, y_train)

     plot_svm_kernels(clf, X, y, X_test)
```

```
<ipython-input-2-aec1f050970f>:24: MatplotlibDeprecationWarning: shading='flat'
when X and Y have the same dimensions as C is deprecated since 3.3.  Either
specify the corners of the quadrilaterals with X and Y, or pass shading='auto',
'nearest' or 'gouraud', or set rcParams['pcolor.shading'].  This will become an
error two minor releases later.
  plt.pcolormesh(XX, YY, Z > 0, cmap=plt.cm.Paired)
```


sigmoid
3.75
3.50
3.25
3.00
2.75
2.50
2.25
2.00
5.0
5.5
6.0
6.5
7.0
7.5
8.0