Homework 9

Giagtos Stylianos

f3352410.

1) Correct 1,2
2) Correct 4
3) Correct 1,5
4) Correct 1,4,5
5) Correct 2,4
6) Correct 1,4
7) Correct 1,4
8) Correct ~~1~~ 3,4
9) Correct 1,3,4
10) Correct 2,3
11) Correct 1,4
12) Correct 1,2
13) Correct 2,4,5
14) Correct 2,3
15) Correct 1,3
16) Not valid: 2,4,5
17) Valid: 1,4
18) Correct: 4
19) Correct: 1, 4
20) Correct: 1,4
21) Correct: 1
22) Correct: 1,2
23) Correct: 1,4
24) Correct: 1,2,3
25) Correct: 1,2,3
26) Correct: 1,2,3
27) Correct: 1
28) Correct: 4
29) Correct: 1,3
30) Correct 2,4
31) Correct 2
32) Correct 3
33) Correct 2,4
34) Correct 2
35) Correct 2,4
36) Correct 2,3
37) Correct 2
38) Correct 2
39) Correct 1,3,6,8
40) Correct 3
41) Correct 1,3
~~42~~

42)

$\min J(\theta) = \min \frac{\|\theta\|^2}{2} = \min \frac{\theta^T \theta}{2}$ subject to $y_i(\theta^T x_i + \theta_0) \geq 1, i=1,\dots,N$

$y_i = 1$ for $x_i \in \omega_1$

$y_i = 0$ for $x_i \in \omega_2$

$L(\theta, \theta_0, \lambda) = \frac{1}{2}\theta^T\theta - \sum_{i=1}^N \lambda_i [y_i(\theta^T x_i + \theta_0) - 1]$

$\frac{dL}{d\theta} = 0 \Leftrightarrow \theta - \sum_{i=1}^N \lambda_i y_i x_i = 0 \Leftrightarrow \theta = \sum_{i=1}^N \lambda_i y_i x_i$

$\frac{dL}{d\theta} = 0 \Leftrightarrow -\sum_{i=1}^N \lambda_i y_i = 0 \Rightarrow \sum_{i=1}^N \lambda_i y_i = 0.$

$\theta = \frac{\theta^T\theta}{2} = \frac{1}{2}(\sum_{i=1}^N \lambda_i y_i x_i)^T(\sum_{j=1}^N \lambda_i y_i x_i)$

$= \frac{1}{2}\sum_{i=1}^N \sum_{j=1}^N \lambda_i \lambda_j y_i y_j x_i^T x_j.$

$L(\lambda) = J(\theta) - \sum_{i=1}^N \lambda_i [y_i(\theta^T x_i + \theta_0) - 1]$

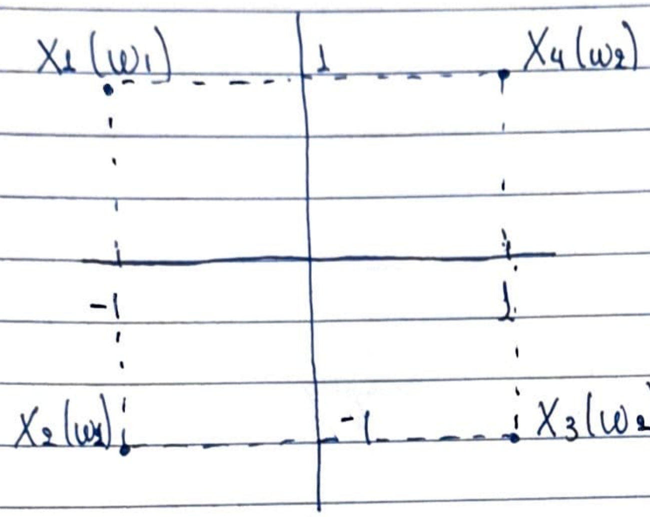
$= \frac{1}{2}\sum_{i=1}^N \sum_{j=1}^N \lambda_i \lambda_j y_i y_j x_i^T x_j - \sum_{i=1}^N \lambda_i [y_i(\sum_{i=1}^N \lambda_i y_i x_i^T x_i + \theta_0 - 1)]$

$= \sum_{i=1}^N \lambda_i - \frac{1}{2}\sum_{i=1}^N \sum_{j=1}^N \lambda_i \lambda_j y_i y_j x_i^T x_j$ Subject to $\sum_{i=1}^N \lambda_i y_i = 0$

$\lambda \geq 0$

Exercise 43

α)



Θεωρούμε την $x_1=0$ την ευθεία που διαχωρίζει τις κλάσεις ω_1, ω_2 καθώς ισαπέχει από τα κοντινότερα σημεία των δύο κλάσεων

b) Για $y_1=y_2=1$ και $y_3=y_4$ έχουμε

$$J(\lambda) = \sum_{i=1}^{4} \lambda_i - 1/2 \sum_{i=1}^{4} \sum_{j=1}^{4} \lambda_i \lambda_j y_i y_j x_i^T x_j$$

Άρα
$$J(\lambda) = \lambda_1+\lambda_2+\lambda_3+\lambda_4 - 1/2\,\lambda_1\lambda_2 \begin{bmatrix}-1 & 1\end{bmatrix}\begin{bmatrix}-1\\1\end{bmatrix} + 1/2\,\lambda_1\lambda_3\begin{bmatrix}-1 & 1\end{bmatrix}\begin{bmatrix}1\\-1\end{bmatrix}$$
$$+ 1/2\,\lambda_1\lambda_4\begin{bmatrix}-1 & 1\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix} - 1/2\,\lambda_2\lambda_1\begin{bmatrix}-1 & -1\end{bmatrix}\begin{bmatrix}-1\\1\end{bmatrix} + 1/2\,\lambda_2\lambda_3\begin{bmatrix}-1 & -1\end{bmatrix}\begin{bmatrix}1\\-1\end{bmatrix}$$
$$+ 1/2\,\lambda_2\lambda_4\begin{bmatrix}-1 & -1\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix} + 1/2\,\lambda_3\lambda_1\begin{bmatrix}1 & -1\end{bmatrix}\begin{bmatrix}-1\\1\end{bmatrix} + 1/2\,\lambda_3\lambda_2\begin{bmatrix}-1 & 1\end{bmatrix}\begin{bmatrix}-1\\1\end{bmatrix} -$$
$$- 1/2\,\lambda_3\lambda_4\begin{bmatrix}1 & -1\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix} + 1/2\,\lambda_4\lambda_1\begin{bmatrix}1 & 1\end{bmatrix}\begin{bmatrix}-1\\1\end{bmatrix} + 1/2\,\lambda_4\lambda_2\begin{bmatrix}1 & 1\end{bmatrix}\begin{bmatrix}-1\\-1\end{bmatrix}$$
$$- 1/2\,\lambda_4\lambda_3\begin{bmatrix}1 & 1\end{bmatrix}\begin{bmatrix}1\\-1\end{bmatrix} - 1/2\,\lambda_1^2\begin{bmatrix}-1 & 1\end{bmatrix}\begin{bmatrix}-1\\1\end{bmatrix} - 1/2\,\lambda_2^2\begin{bmatrix}-1 & -1\end{bmatrix}\begin{bmatrix}-1\\-1\end{bmatrix}$$
$$- 1/2\,\lambda_3\begin{bmatrix}1 & -1\end{bmatrix}\begin{bmatrix}1\\-1\end{bmatrix} - 1/2\,\lambda_4^2\begin{bmatrix}1 & 1\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix} =$$
$$= \lambda_1+\lambda_2+\lambda_3+\lambda_4 - \lambda_1^2 - \lambda_2^2 - \lambda_3^2 - \lambda_4^2 - 2\lambda_1\lambda_3 - 2\lambda_2\lambda_4$$

- $\frac{dJ(\lambda)}{d\lambda_1} = 1 - 2\lambda_1 - 2\lambda_3 = 0 \Rightarrow \lambda_1+\lambda_3 = 1/2$
- $\frac{dJ(\lambda)}{d\lambda_2} = 1 - 2\lambda_2 - 2\lambda_4 = 0 \Rightarrow \lambda_2+\lambda_4 = 1/2$

- $\sum_{i=1}^{4} \lambda_i y_i = 0 \Leftrightarrow \lambda_1+\lambda_2-\lambda_3-\lambda_4 = 0 \Leftrightarrow \lambda_1+\lambda_2 = \lambda_3+\lambda_4$
- $\theta = \sum_{i=1}^{4} \lambda_i y_i x_i = \lambda_1\begin{bmatrix}-1\\1\end{bmatrix} + \lambda_2\begin{bmatrix}-1\\-1\end{bmatrix} - \lambda_3\begin{bmatrix}1\\-1\end{bmatrix} - \lambda_4\begin{bmatrix}1\\1\end{bmatrix} = \begin{bmatrix}-1\\0\end{bmatrix}$

Για $i=1 \Rightarrow \lambda_1\left[y_1(\theta^T x_1+\theta_0) - 1\right] = 0 \Rightarrow \theta_0 = 0$

$g(x) = \theta^T x + \theta_0 \Rightarrow x_1 = 0.$

c) Η υπόθεση επιβεβαιώνεται από τα SVM.

EXERCISE 44

In [1]:
```
import scipy.io as sio
import numpy as np
import matplotlib.pyplot as plt


Dataset_a = sio.loadmat('HW9a.mat')

train_x_a = Dataset_a['train_X']
train_y_a = Dataset_a['train_y']
test_x_a = Dataset_a['test_X']
test_y_a = Dataset_a['test_y']




from sklearn import svm


def make_meshgrid(x, y, h=.02):
    """Create a mesh of points to plot in

    Parameters
    ----------
    x: data to base x-axis meshgrid on
    y: data to base y-axis meshgrid on
    h: stepsize for meshgrid, optional

    Returns
    -------
    xx, yy : ndarray
    """
    x_min, x_max = x.min() - 1, x.max() + 1
    y_min, y_max = y.min() - 1, y.max() + 1
    xx, yy = np.meshgrid(np.arange(x_min, x_max, h),
                         np.arange(y_min, y_max, h))
    return xx, yy


def plot_contours(ax, clf, xx, yy, **params):
    """Plot the decision boundaries for a classifier.

    Parameters
    ----------
    ax: matplotlib axes object
    clf: a classifier
    xx: meshgrid ndarray
    yy: meshgrid ndarray
    params: dictionary of params to pass to contourf, optional
    """
    Z = clf.predict(np.c_[xx.ravel(), yy.ravel()])
    Z = Z.reshape(xx.shape)
```

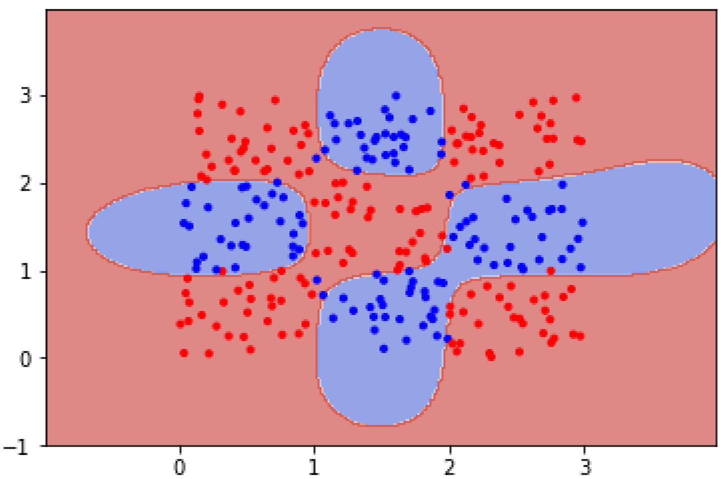
```
    out = ax.contourf(xx, yy, Z, **params)
    return out
```

In [2]:

```
#select kernel function:{'rbf','poly'}
clf = svm.SVC(kernel='rbf',C=10,gamma=1,degree=2)
clf.fit(train_x_a, train_y_a.reshape(270))

X00, X11 = train_x_a[:,0], train_x_a[:,1]
xx, yy = make_meshgrid(X00, X11)
fig, ax= plt.subplots(1, 1)
color= ['red' if l == 1 else 'blue' for l in train_y_a.reshape(270)]
plot_contours(ax, clf, xx, yy,
                  cmap=plt.cm.coolwarm, alpha=0.6)
ax.scatter(X00, X11,c=color, cmap=plt.cm.coolwarm, s=10, edgecolors='face')
plt.show()

clf.predict(test_x_a)
```



Out[2]:

```
array([ 1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,
        1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1, -1, -1, -1, -1,
       -1, -1, -1, -1, -1, -1, -1, -1, -1,  1, -1,  1, -1,  1, -1, -1, -1,
        1, -1, -1, -1, -1, -1, -1, -1, -1,  1,  1,  1,  1,  1,  1,  1,  1,
        1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,
        1,  1,  1,  1,  1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
       -1, -1, -1, -1, -1, -1, -1,  1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
       -1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,
        1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1, -1, -1, -1,
       -1,  1, -1, -1, -1, -1,  1, -1, -1, -1, -1, -1, -1, -1, -1, -1,  1,
       -1, -1,  1,  1, -1, -1, -1, -1, -1, -1,  1,  1,  1,  1, -1,  1,  1,
        1,  1,  1,  1, -1,  1,  1, -1,  1,  1,  1,  1,  1,  1,  1,  1,  1,
        1,  1,  1,  1,  1,  1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
       -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
       -1, -1,  1,  1,  1,  1,  1,  1,  1,  1,  1, -1,  1,  1,  1,  1,  1,
        1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1, -1],
      dtype=int16)
```

In [1]:

```
import scipy.io as sio
import numpy as np
import matplotlib.pyplot as plt
from sklearn import svm
from sklearn.metrics import accuracy_score, classification_report
```

```python
# Load the dataset from HW9a.mat
Dataset_a = sio.loadmat('HW9a.mat')

# Extract the training and test sets
train_x_a = Dataset_a['train_X']  # Training features
train_y_a = Dataset_a['train_y']  # Training labels
test_x_a = Dataset_a['test_X']    # Test features
test_y_a = Dataset_a['test_y']    # Test labels

# Function to create a meshgrid for plotting decision boundaries
def make_meshgrid(x, y, h=.02):
    """Create a mesh grid to plot decision boundaries."""
    x_min, x_max = x.min() - 1, x.max() + 1
    y_min, y_max = y.min() - 1, y.max() + 1
    xx, yy = np.meshgrid(np.arange(x_min, x_max, h),
                         np.arange(y_min, y_max, h))
    return xx, yy

# Function to plot decision boundaries
def plot_contours(ax, clf, xx, yy, **params):
    """Plot decision boundaries for a classifier."""
    Z = clf.predict(np.c_[xx.ravel(), yy.ravel()])
    Z = Z.reshape(xx.shape)
    out = ax.contourf(xx, yy, Z, **params)
    return out

# Function to train and evaluate SVM with different kernels
def train_and_evaluate_svm(kernel, C=1.0, degree=3, gamma='scale'):
    """
    Train an SVM model with specified kernel and parameters and evaluate it.

    Parameters:
    - kernel: The kernel type ('linear', 'poly', or 'rbf')
    - C: Regularization parameter
    - degree: Degree of the polynomial kernel (relevant for 'poly')
    - gamma: Kernel coefficient (relevant for 'poly' and 'rbf')
    """
    # Create the SVM classifier
    clf = svm.SVC(kernel=kernel, C=C, degree=degree, gamma=gamma)
    clf.fit(train_x_a, train_y_a.flatten())  # Train the model

    # Make predictions on the test set
    test_pred = clf.predict(test_x_a)

    # Evaluate accuracy and print classification report
    accuracy = accuracy_score(test_y_a.flatten(), test_pred)
    print(f"\nAccuracy on the test set with {kernel} kernel: {accuracy:.4f}")
    print(classification_report(test_y_a.flatten(), test_pred))

    # Plot the decision boundaries
    X0, X1 = train_x_a[:, 0], train_x_a[:, 1]
    xx, yy = make_meshgrid(X0, X1)  # Generate meshgrid for plotting
    fig, ax = plt.subplots(1, 1, figsize=(8, 6))
    color = ['red' if label == 1 else 'blue' for label in train_y_a.flatten()]
    plot_contours(ax, clf, xx, yy, cmap=plt.cm.coolwarm, alpha=0.6)  # Plot decisio
```

```python
    ax.scatter(X0, X1, c=color, cmap=plt.cm.coolwarm, s=10, edgecolors='face')  # P
    plt.title(f"Decision Boundary with {kernel} kernel")
    plt.show()

# Train and evaluate SVM with different kernels and parameters
kernels = ['linear', 'poly', 'rbf']
C_values = [0.1, 1.0, 10.0]  # Experiment with different values of C
degree_values = [3, 4, 5]  # Only relevant for polynomial kernel
gamma_values = ['scale', 'auto']  # Only relevant for polynomial and RBF kernels

# Evaluate SVM with linear kernel
print("Evaluating SVM with Linear Kernel:")
for C in C_values:
    train_and_evaluate_svm('linear', C=C)

# Evaluate SVM with polynomial kernel
print("Evaluating SVM with Polynomial Kernel:")
for C in C_values:
    for degree in degree_values:
        train_and_evaluate_svm('poly', C=C, degree=degree)

# Evaluate SVM with RBF kernel
print("Evaluating SVM with RBF Kernel:")
for C in C_values:
    for gamma in gamma_values:
        train_and_evaluate_svm('rbf', C=C, gamma=gamma)
```

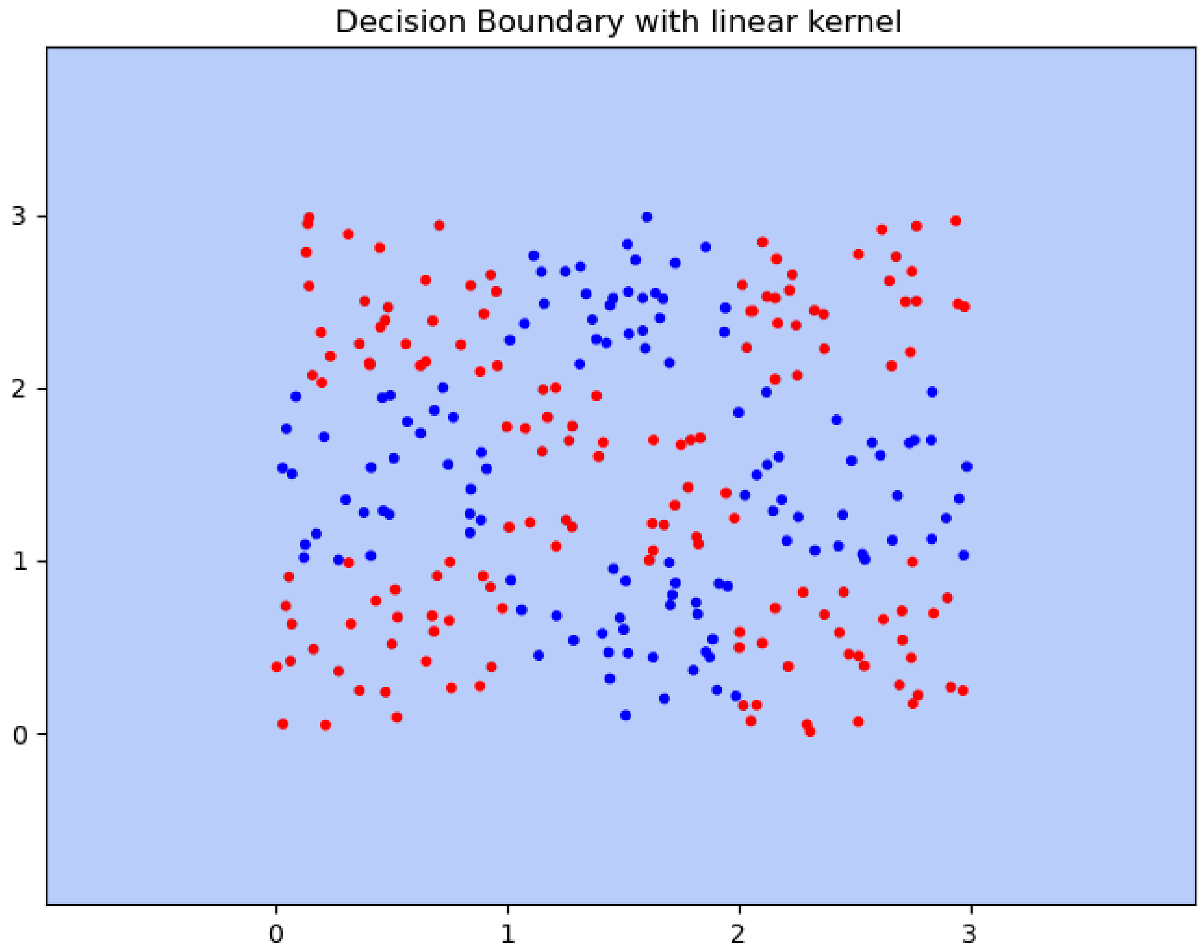
```
Evaluating SVM with Linear Kernel:

Accuracy on the test set with linear kernel: 0.5556
              precision    recall  f1-score   support

          -1       0.00      0.00      0.00       120
           1       0.56      1.00      0.71       150

    accuracy                           0.56       270
   macro avg       0.28      0.50      0.36       270
weighted avg       0.31      0.56      0.40       270
```

```
C:\Users\steli\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\steli\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\steli\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\steli\AppData\Local\Temp\ipykernel_20700\438884638.py:62: UserWarning: No d
ata for colormapping provided via 'c'. Parameters 'cmap' will be ignored
  ax.scatter(X0, X1, c=color, cmap=plt.cm.coolwarm, s=10, edgecolors='face')  # Plot
training points
```

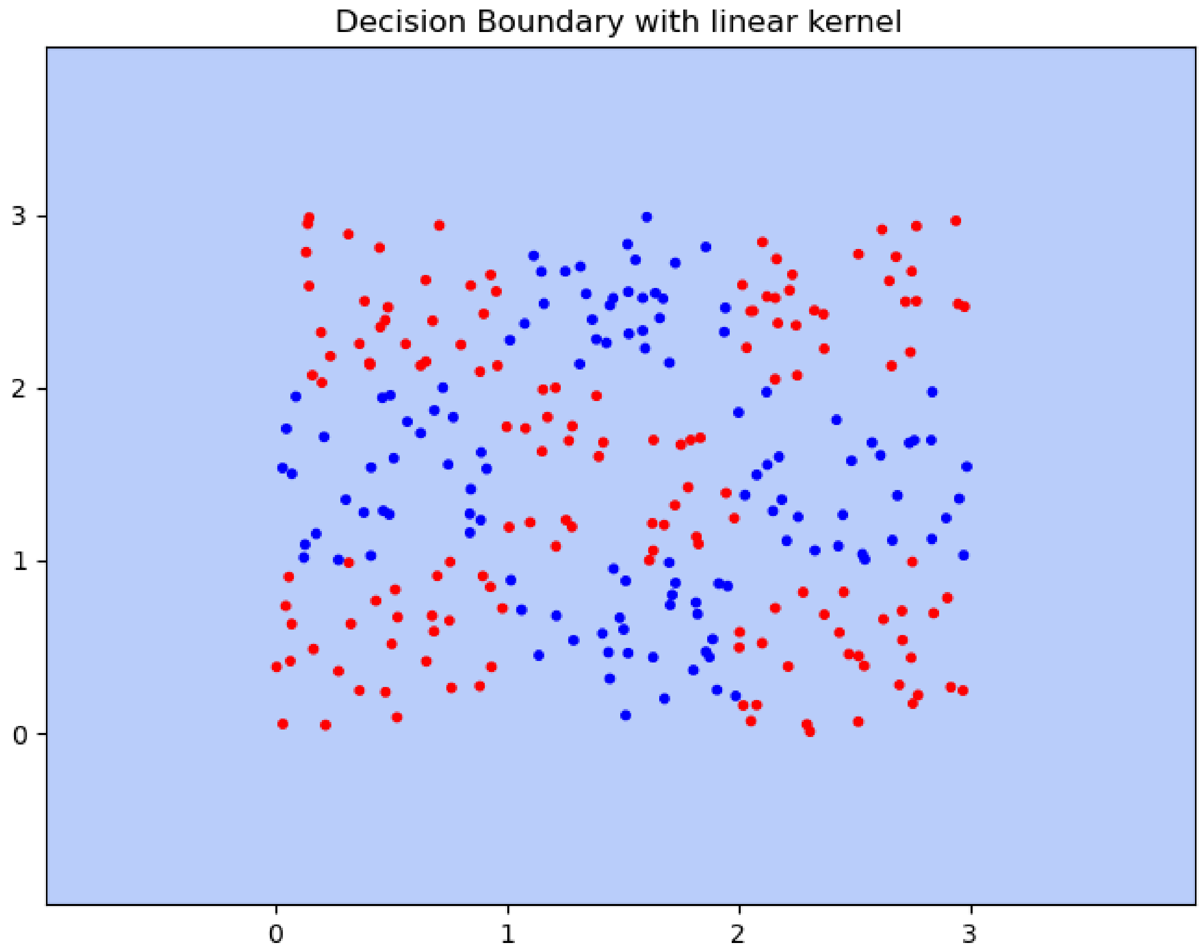



```
Accuracy on the test set with linear kernel: 0.5556
              precision    recall  f1-score   support

          -1       0.00      0.00      0.00       120
           1       0.56      1.00      0.71       150

    accuracy                           0.56       270
   macro avg       0.28      0.50      0.36       270
weighted avg       0.31      0.56      0.40       270
```

```
C:\Users\steli\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\steli\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\steli\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\steli\AppData\Local\Temp\ipykernel_20700\438884638.py:62: UserWarning: No d
ata for colormapping provided via 'c'. Parameters 'cmap' will be ignored
  ax.scatter(X0, X1, c=color, cmap=plt.cm.coolwarm, s=10, edgecolors='face')  # Plot
training points
```

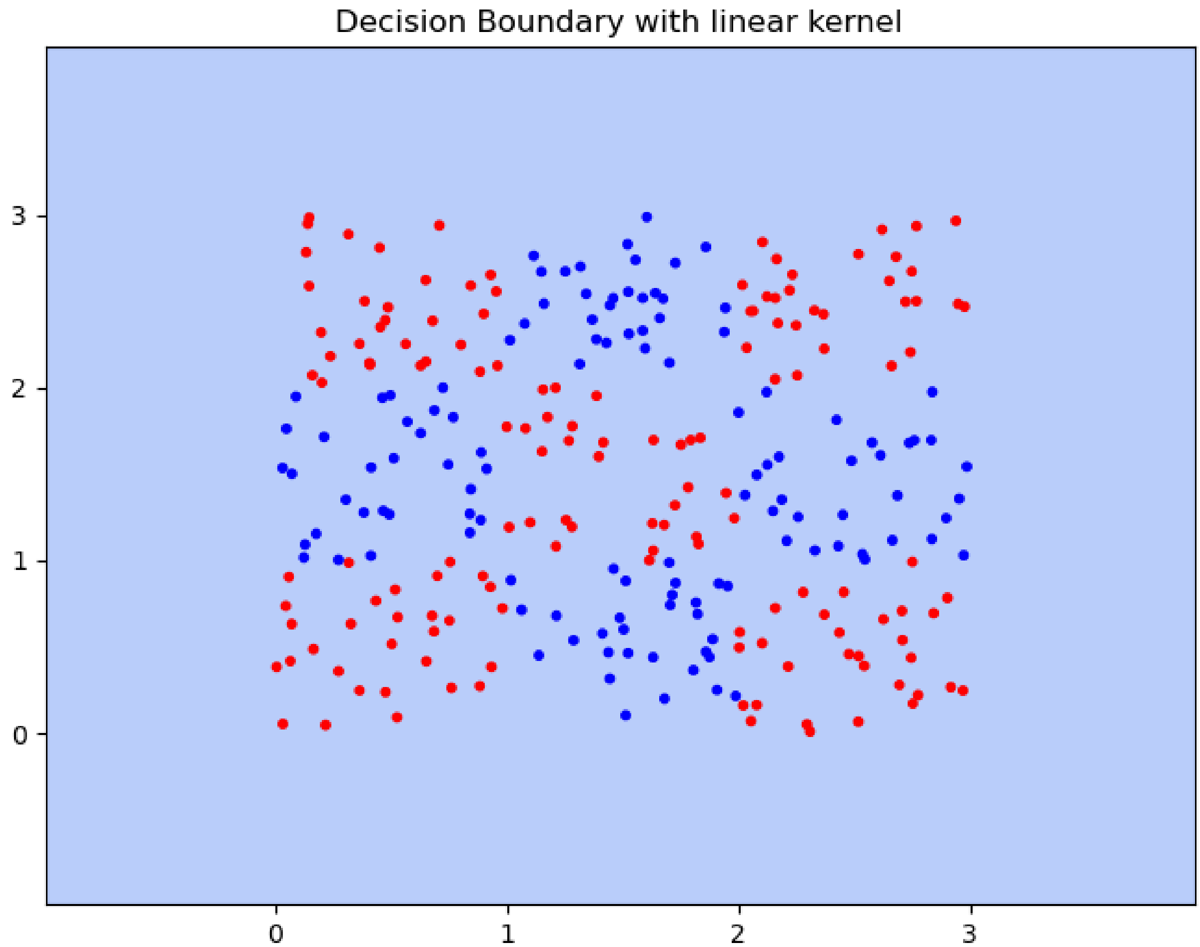



```
C:\Users\steli\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\steli\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\steli\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
Accuracy on the test set with linear kernel: 0.5556
              precision    recall  f1-score   support

          -1       0.00      0.00      0.00       120
           1       0.56      1.00      0.71       150

    accuracy                           0.56       270
   macro avg       0.28      0.50      0.36       270
weighted avg       0.31      0.56      0.40       270
```

```
C:\Users\steli\AppData\Local\Temp\ipykernel_20700\438884638.py:62: UserWarning: No d
ata for colormapping provided via 'c'. Parameters 'cmap' will be ignored
  ax.scatter(X0, X1, c=color, cmap=plt.cm.coolwarm, s=10, edgecolors='face')  # Plot
training points
```



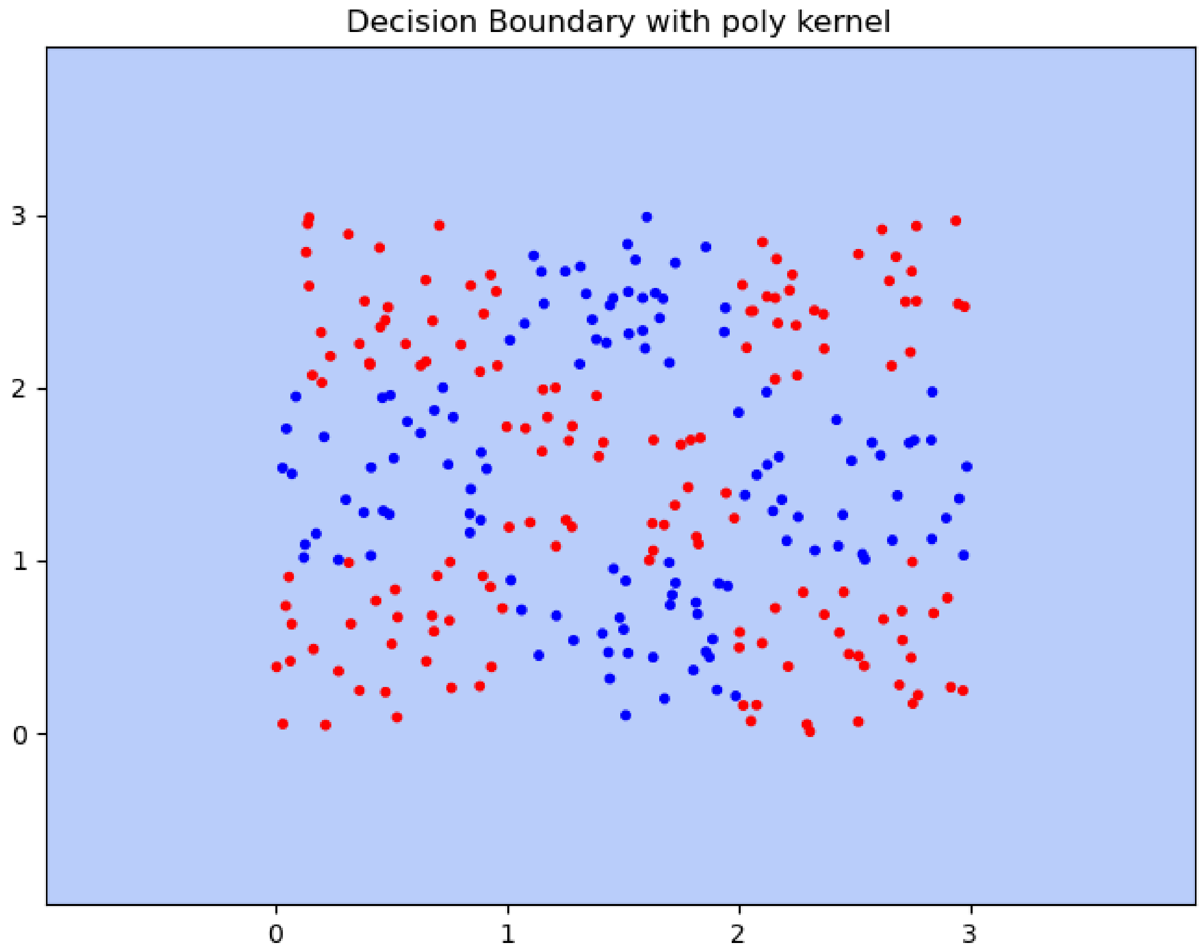

```
C:\Users\steli\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\steli\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\steli\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
Evaluating SVM with Polynomial Kernel:

Accuracy on the test set with poly kernel: 0.5556
              precision    recall  f1-score   support

          -1       0.00      0.00      0.00       120
           1       0.56      1.00      0.71       150

    accuracy                           0.56       270
   macro avg       0.28      0.50      0.36       270
weighted avg       0.31      0.56      0.40       270
```

```
C:\Users\steli\AppData\Local\Temp\ipykernel_20700\438884638.py:62: UserWarning: No d
ata for colormapping provided via 'c'. Parameters 'cmap' will be ignored
  ax.scatter(X0, X1, c=color, cmap=plt.cm.coolwarm, s=10, edgecolors='face')  # Plot
training points
```

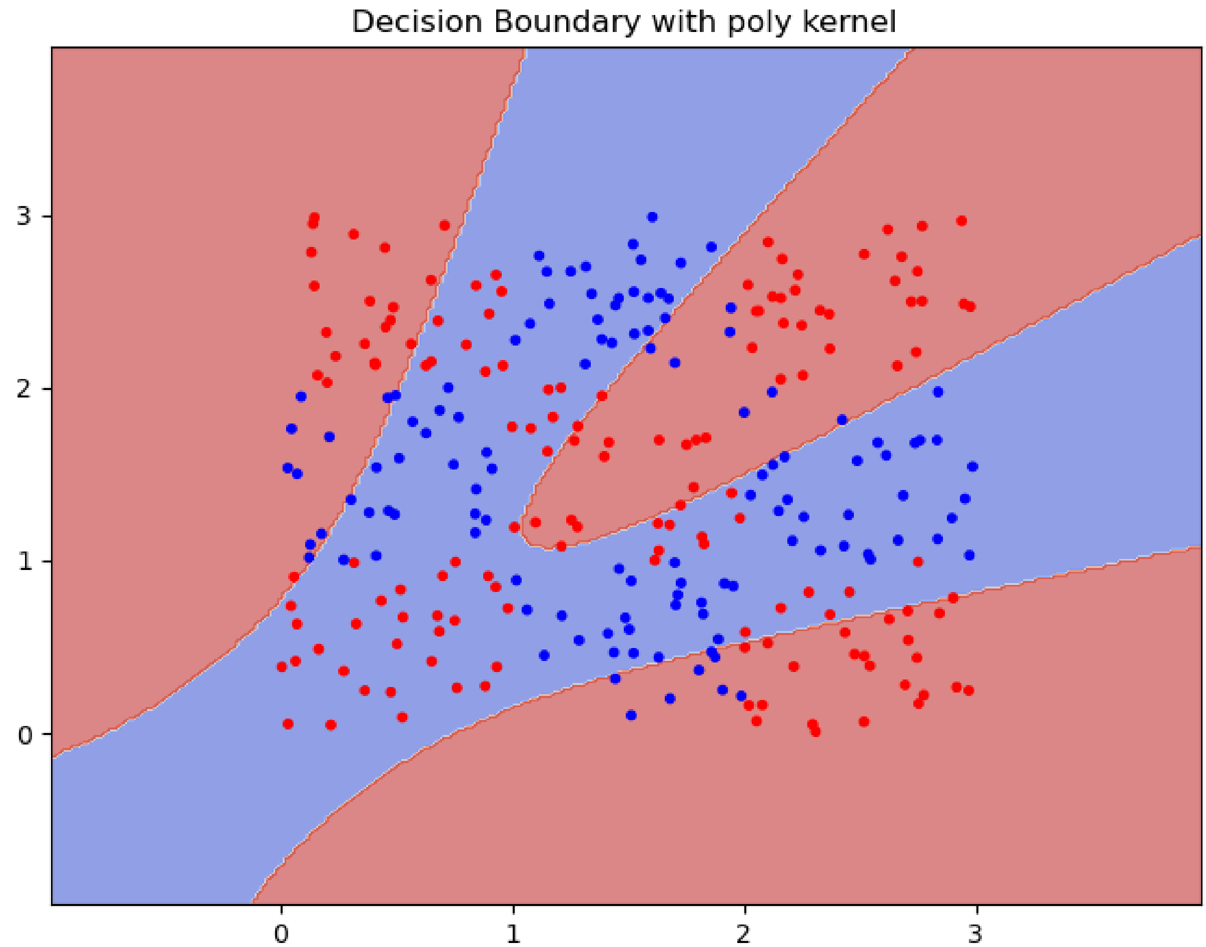



```
Accuracy on the test set with poly kernel: 0.6444
              precision    recall  f1-score   support

          -1       0.58      0.72      0.64       120
           1       0.72      0.59      0.65       150

    accuracy                           0.64       270
   macro avg       0.65      0.65      0.64       270
weighted avg       0.66      0.64      0.64       270
```


```
C:\Users\steli\AppData\Local\Temp\ipykernel_20700\438884638.py:62: UserWarning: No d
ata for colormapping provided via 'c'. Parameters 'cmap' will be ignored
  ax.scatter(X0, X1, c=color, cmap=plt.cm.coolwarm, s=10, edgecolors='face')  # Plot
training points
```

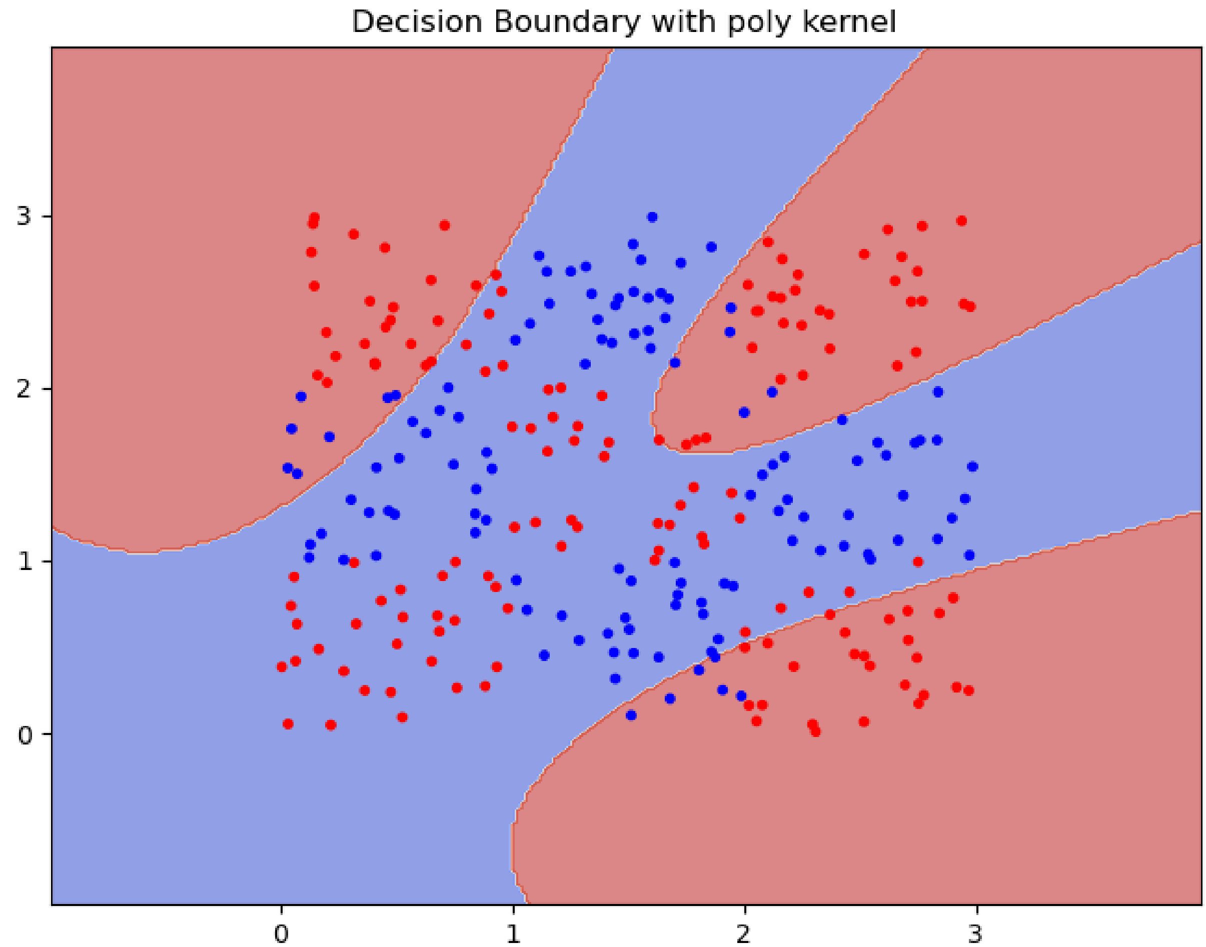



```
Accuracy on the test set with poly kernel: 0.6815
              precision    recall  f1-score   support

          -1       0.60      0.82      0.70       120
           1       0.80      0.57      0.66       150

    accuracy                           0.68       270
   macro avg       0.70      0.70      0.68       270
weighted avg       0.71      0.68      0.68       270
```

```
C:\Users\steli\AppData\Local\Temp\ipykernel_20700\438884638.py:62: UserWarning: No d
ata for colormapping provided via 'c'. Parameters 'cmap' will be ignored
  ax.scatter(X0, X1, c=color, cmap=plt.cm.coolwarm, s=10, edgecolors='face')  # Plot
training points
```

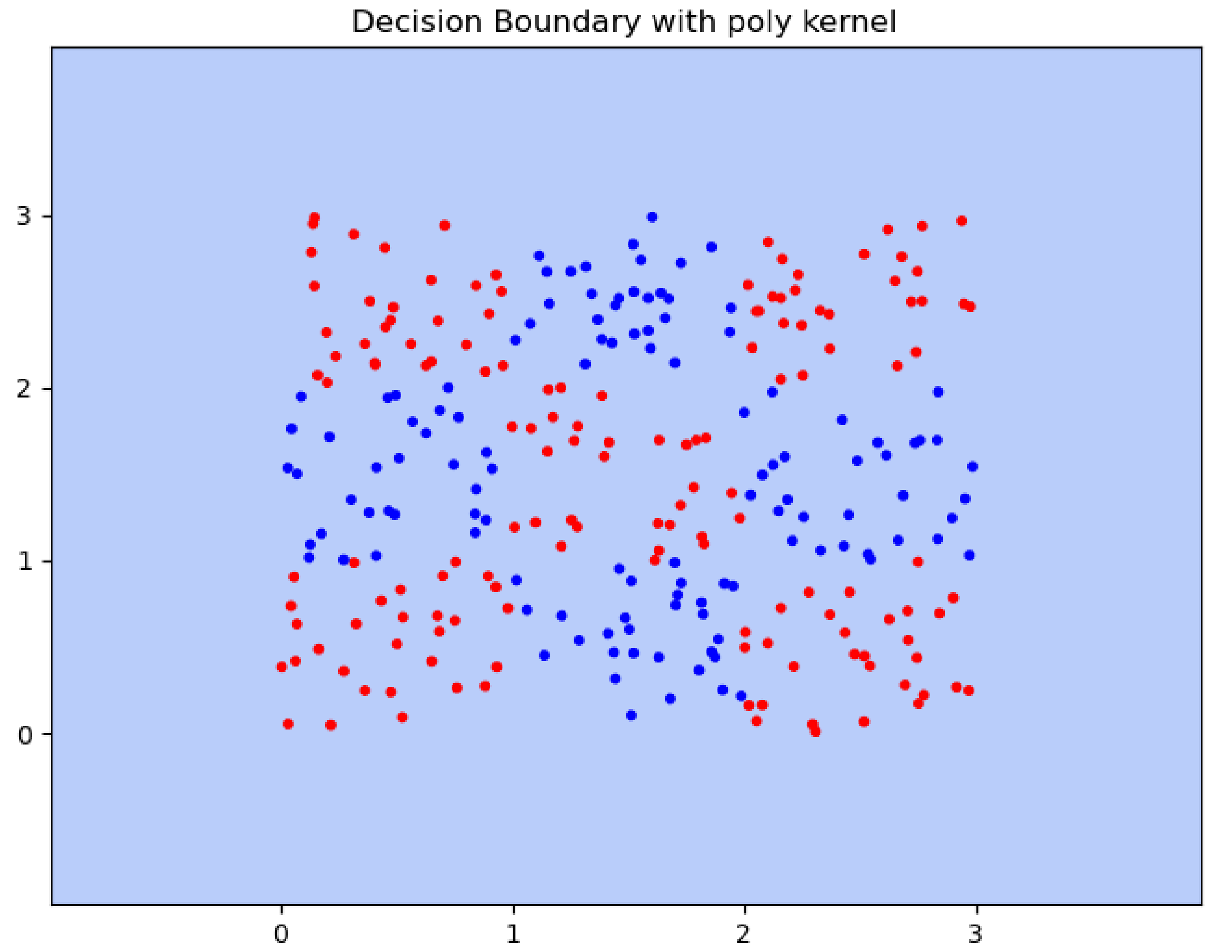



```
Accuracy on the test set with poly kernel: 0.5556
              precision    recall  f1-score   support

          -1       0.00      0.00      0.00       120
           1       0.56      1.00      0.71       150

    accuracy                           0.56       270
   macro avg       0.28      0.50      0.36       270
weighted avg       0.31      0.56      0.40       270
```

```
C:\Users\steli\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\steli\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\steli\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\steli\AppData\Local\Temp\ipykernel_20700\438884638.py:62: UserWarning: No d
ata for colormapping provided via 'c'. Parameters 'cmap' will be ignored
  ax.scatter(X0, X1, c=color, cmap=plt.cm.coolwarm, s=10, edgecolors='face')  # Plot
training points
```

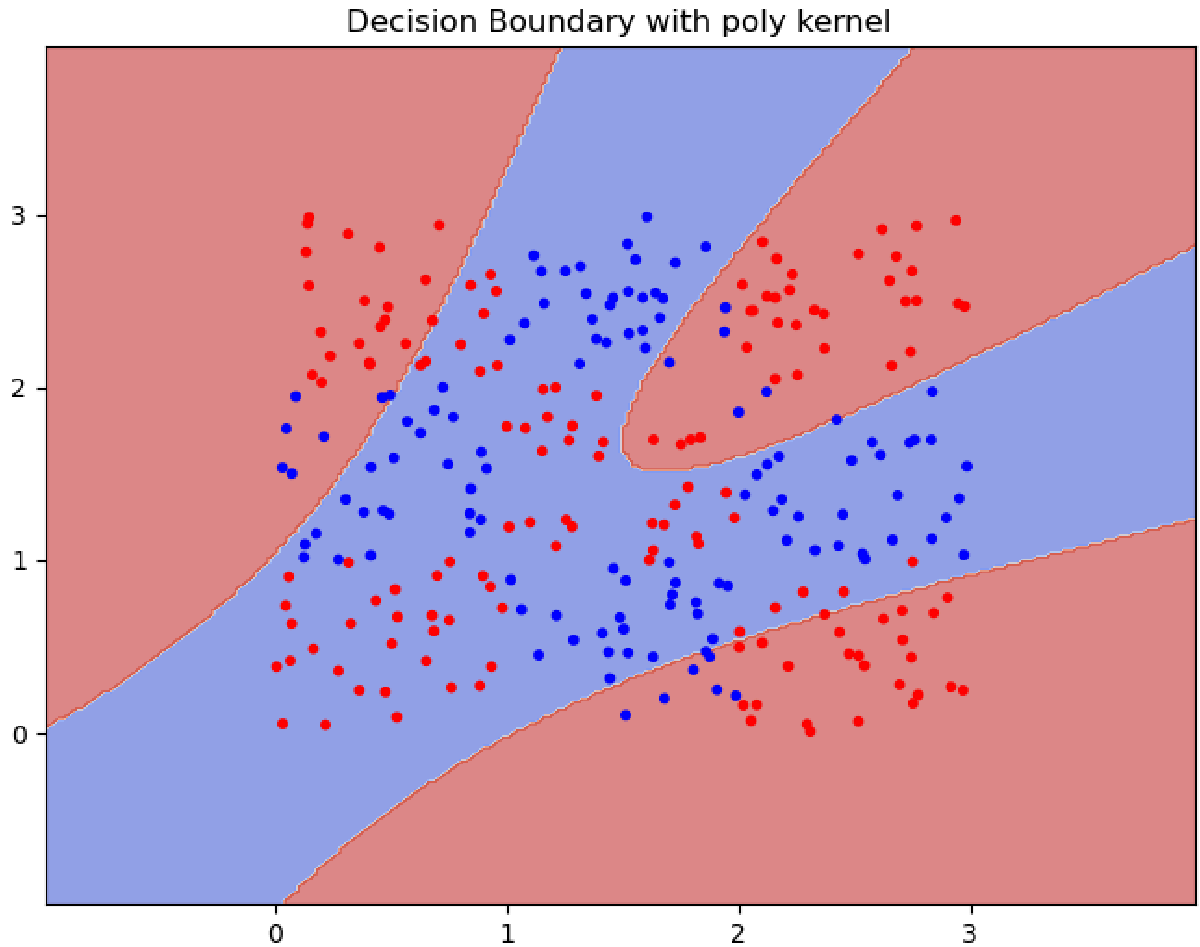



```
Accuracy on the test set with poly kernel: 0.6444
              precision    recall  f1-score   support

          -1       0.58      0.76      0.65       120
           1       0.74      0.55      0.63       150

    accuracy                           0.64       270
   macro avg       0.66      0.66      0.64       270
weighted avg       0.67      0.64      0.64       270
```

```
C:\Users\steli\AppData\Local\Temp\ipykernel_20700\438884638.py:62: UserWarning: No d
ata for colormapping provided via 'c'. Parameters 'cmap' will be ignored
  ax.scatter(X0, X1, c=color, cmap=plt.cm.coolwarm, s=10, edgecolors='face')  # Plot
training points
```

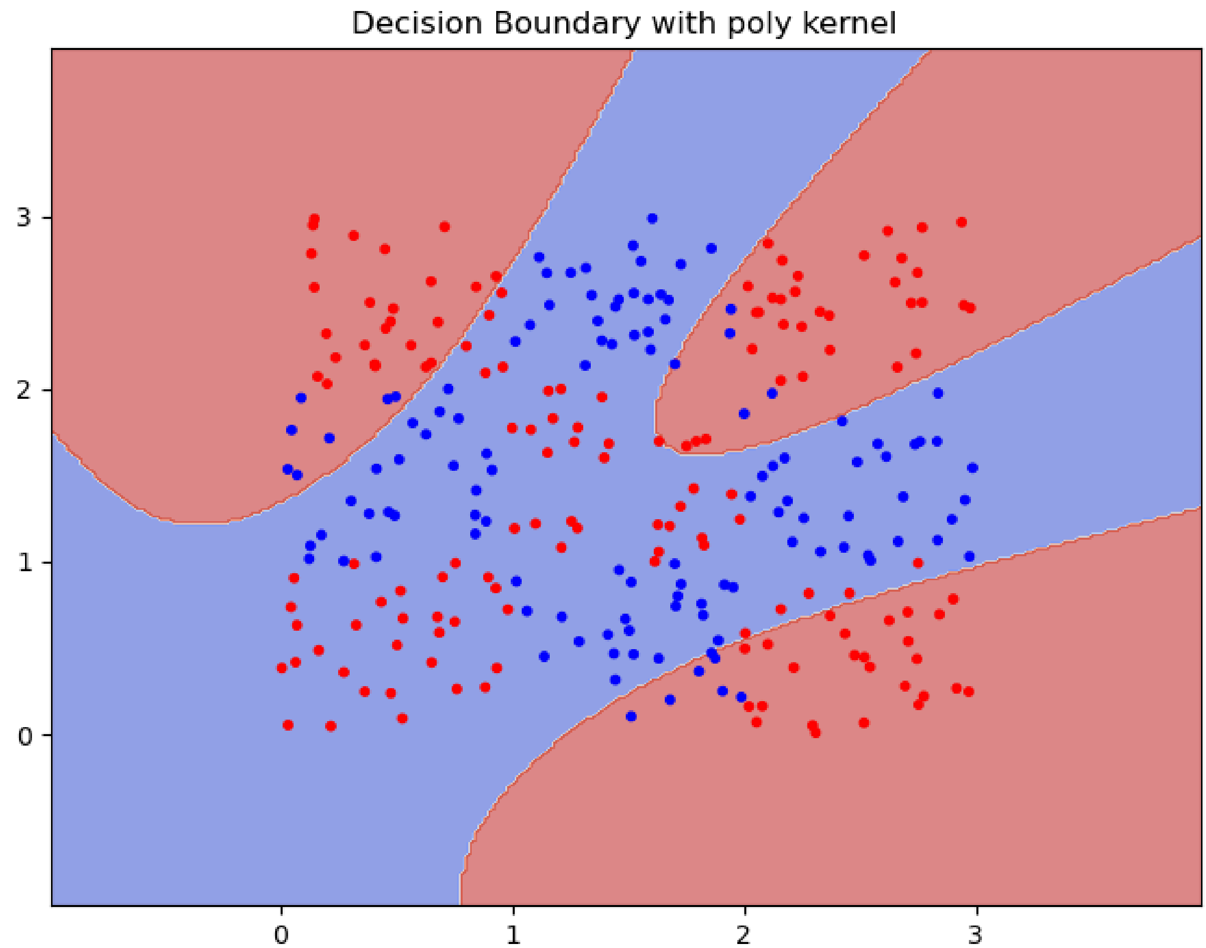



```
Accuracy on the test set with poly kernel: 0.6741
              precision    recall  f1-score   support

          -1       0.60      0.82      0.69       120
           1       0.79      0.56      0.66       150

    accuracy                           0.67       270
   macro avg       0.70      0.69      0.67       270
weighted avg       0.71      0.67      0.67       270
```

```
C:\Users\steli\AppData\Local\Temp\ipykernel_20700\438884638.py:62: UserWarning: No d
ata for colormapping provided via 'c'. Parameters 'cmap' will be ignored
  ax.scatter(X0, X1, c=color, cmap=plt.cm.coolwarm, s=10, edgecolors='face')  # Plot
training points
```

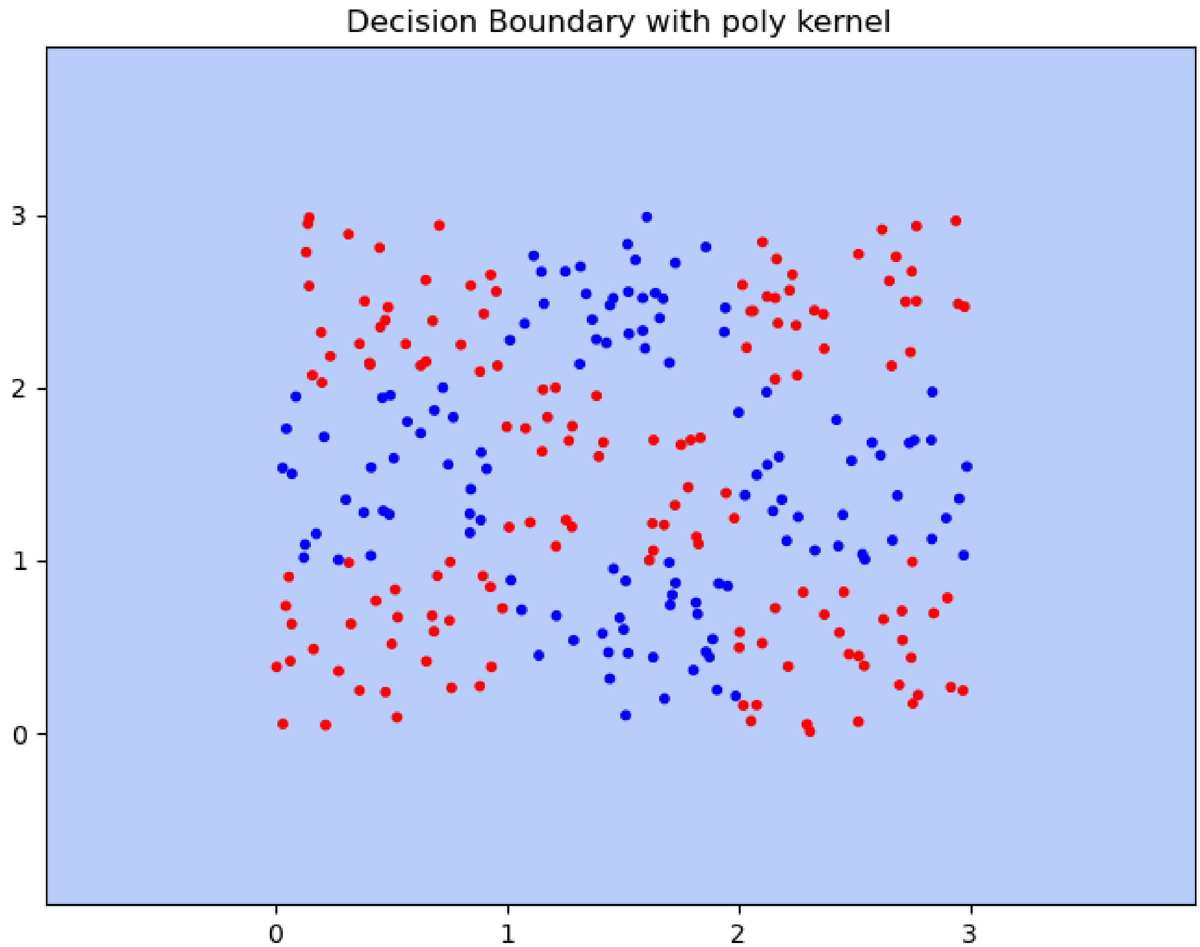



```
Accuracy on the test set with poly kernel: 0.5556
              precision    recall  f1-score   support

          -1       0.00      0.00      0.00       120
           1       0.56      1.00      0.71       150

    accuracy                           0.56       270
   macro avg       0.28      0.50      0.36       270
weighted avg       0.31      0.56      0.40       270
```

```
C:\Users\steli\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\steli\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\steli\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\steli\AppData\Local\Temp\ipykernel_20700\438884638.py:62: UserWarning: No d
ata for colormapping provided via 'c'. Parameters 'cmap' will be ignored
  ax.scatter(X0, X1, c=color, cmap=plt.cm.coolwarm, s=10, edgecolors='face')  # Plot
training points
```

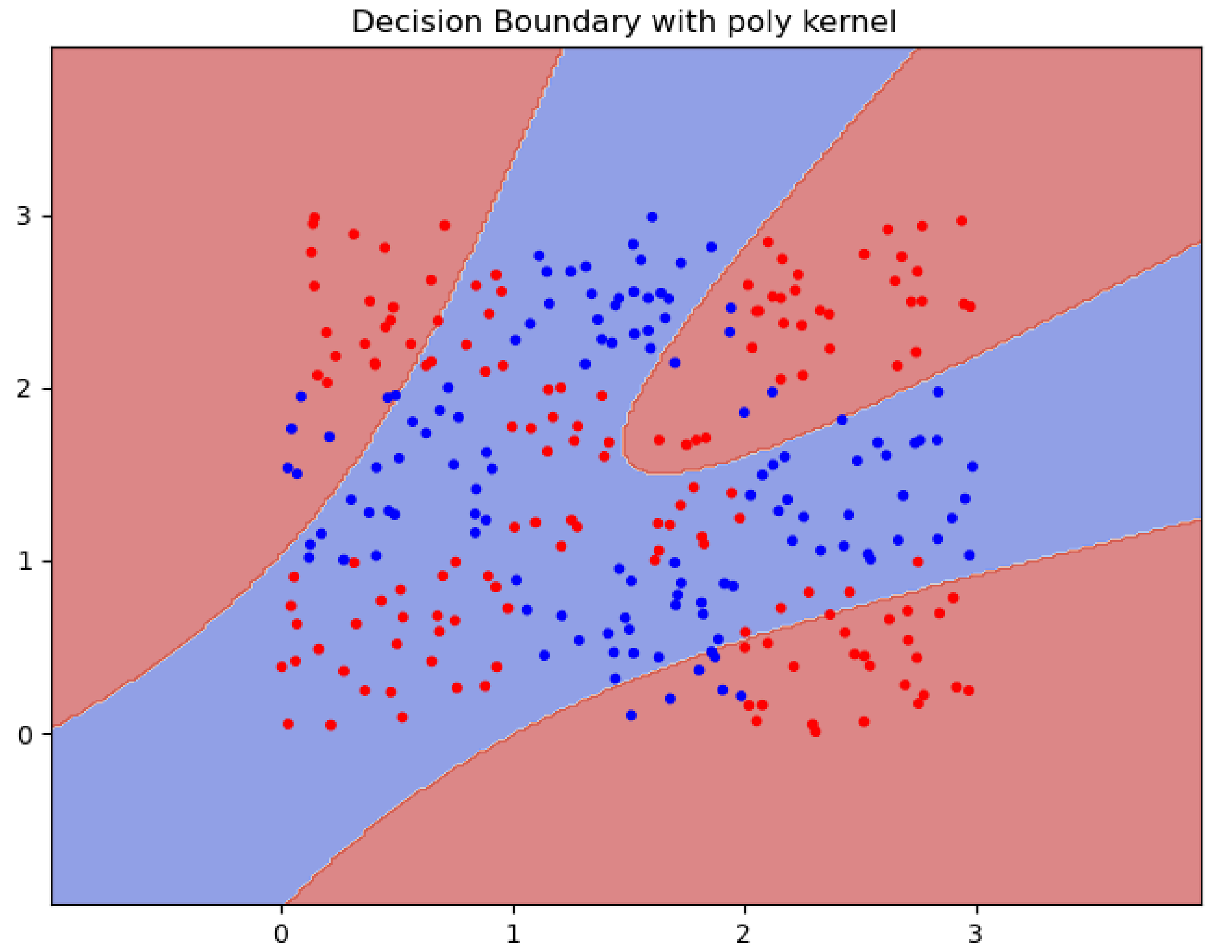



```
Accuracy on the test set with poly kernel: 0.6444
              precision    recall  f1-score   support

          -1       0.58      0.76      0.65       120
           1       0.74      0.55      0.63       150

    accuracy                           0.64       270
   macro avg       0.66      0.66      0.64       270
weighted avg       0.67      0.64      0.64       270
```

```
C:\Users\steli\AppData\Local\Temp\ipykernel_20700\438884638.py:62: UserWarning: No d
ata for colormapping provided via 'c'. Parameters 'cmap' will be ignored
  ax.scatter(X0, X1, c=color, cmap=plt.cm.coolwarm, s=10, edgecolors='face')  # Plot
training points
```

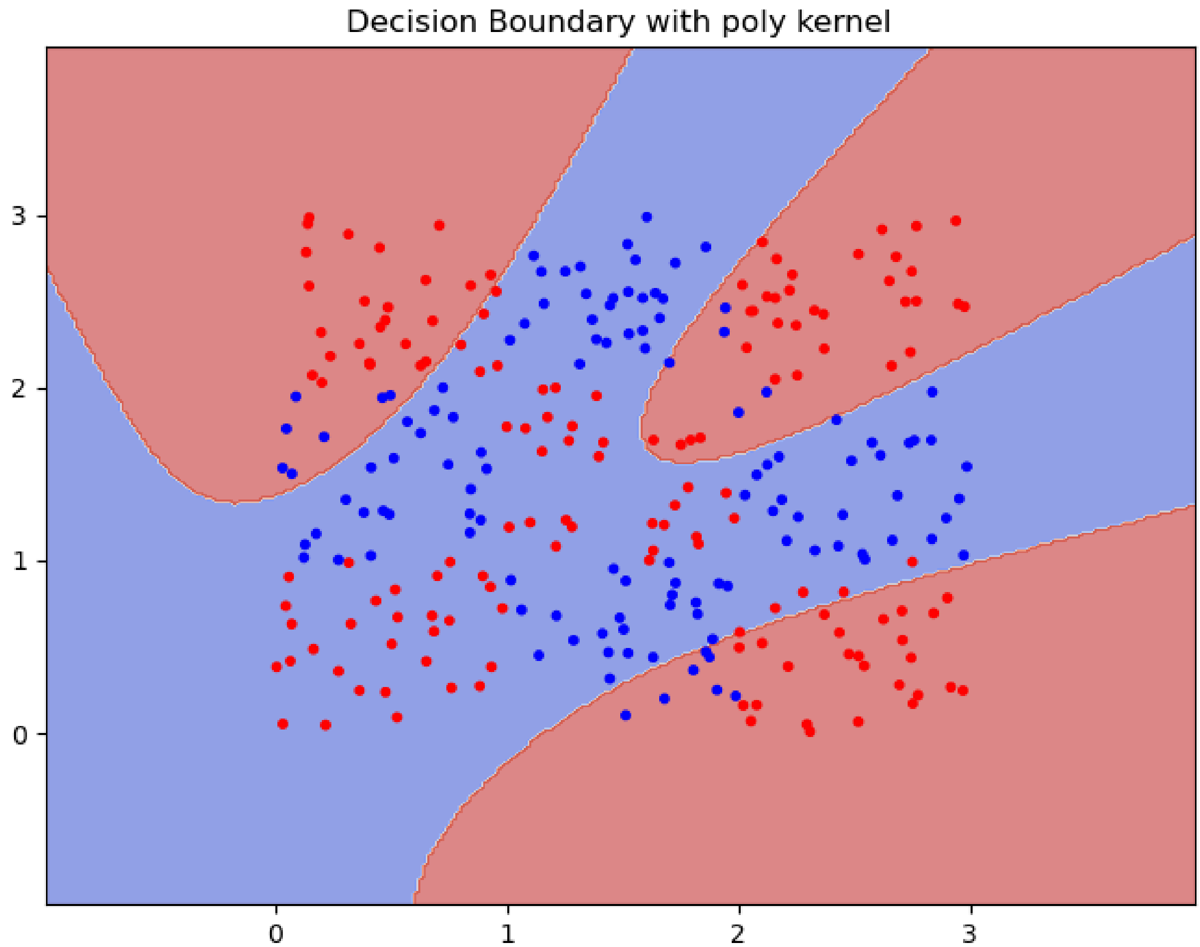



```
Accuracy on the test set with poly kernel: 0.6815
              precision    recall  f1-score   support

          -1       0.61      0.80      0.69       120
           1       0.79      0.59      0.67       150

    accuracy                           0.68       270
   macro avg       0.70      0.69      0.68       270
weighted avg       0.71      0.68      0.68       270
```

```
C:\Users\steli\AppData\Local\Temp\ipykernel_20700\438884638.py:62: UserWarning: No d
ata for colormapping provided via 'c'. Parameters 'cmap' will be ignored
  ax.scatter(X0, X1, c=color, cmap=plt.cm.coolwarm, s=10, edgecolors='face')  # Plot
training points
```



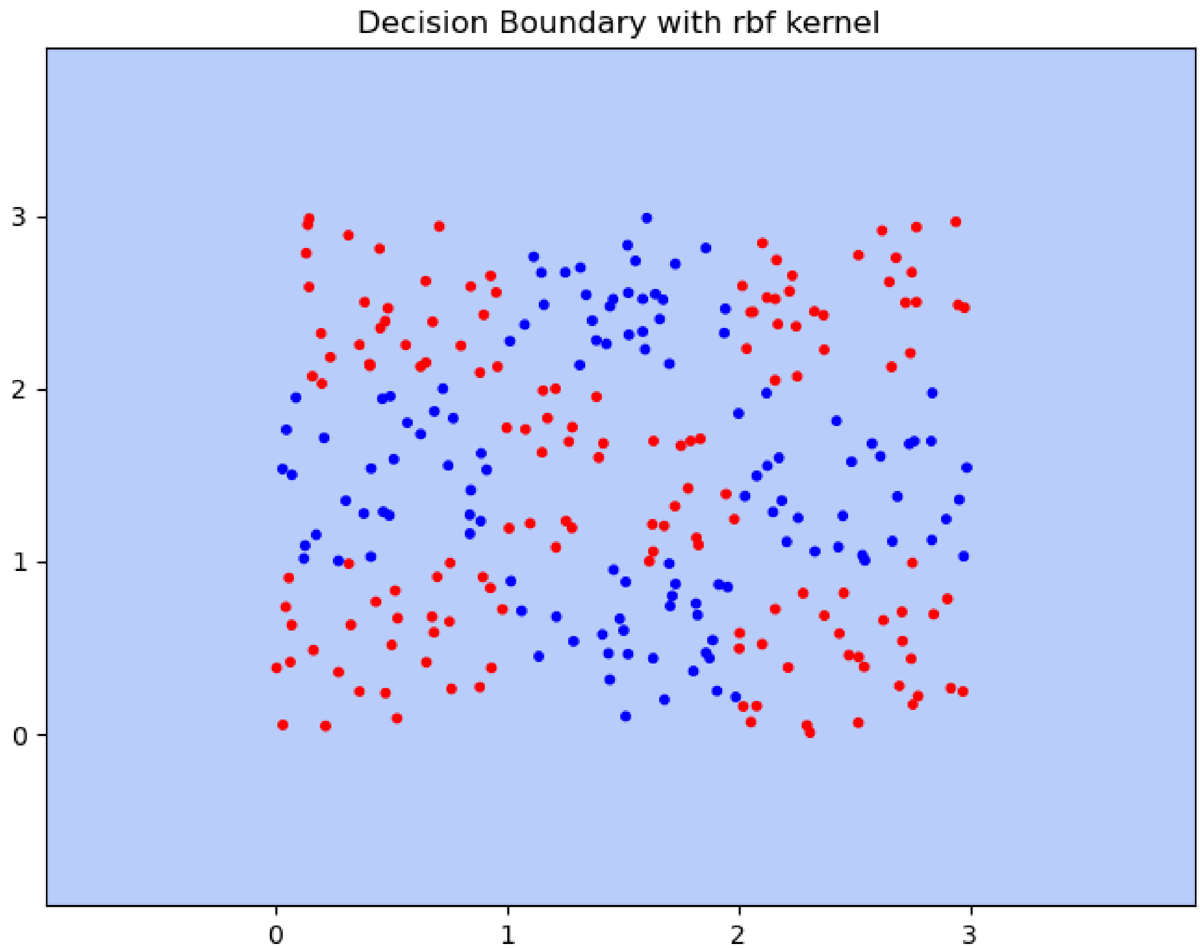

```
C:\Users\steli\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
Evaluating SVM with RBF Kernel:

Accuracy on the test set with rbf kernel: 0.5556
              precision    recall  f1-score   support

          -1       0.00      0.00      0.00       120
           1       0.56      1.00      0.71       150

    accuracy                           0.56       270
   macro avg       0.28      0.50      0.36       270
weighted avg       0.31      0.56      0.40       270
```

```
C:\Users\steli\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\steli\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\steli\AppData\Local\Temp\ipykernel_20700\438884638.py:62: UserWarning: No d
ata for colormapping provided via 'c'. Parameters 'cmap' will be ignored
  ax.scatter(X0, X1, c=color, cmap=plt.cm.coolwarm, s=10, edgecolors='face')  # Plot
training points
```



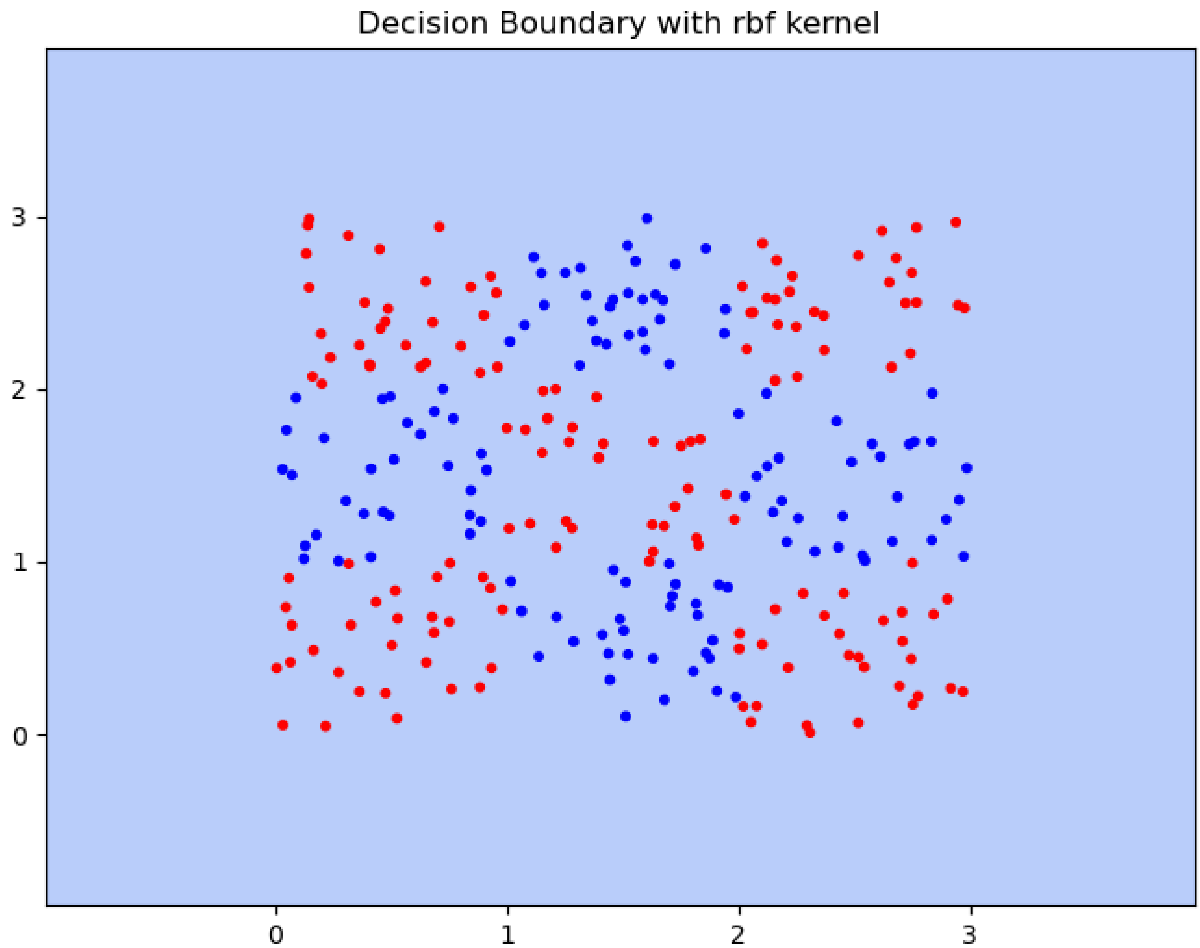

```
C:\Users\steli\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\steli\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\steli\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1531:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

```
Accuracy on the test set with rbf kernel: 0.5556
              precision    recall  f1-score   support

          -1       0.00      0.00      0.00       120
           1       0.56      1.00      0.71       150

    accuracy                           0.56       270
   macro avg       0.28      0.50      0.36       270
weighted avg       0.31      0.56      0.40       270
```

```
C:\Users\steli\AppData\Local\Temp\ipykernel_20700\438884638.py:62: UserWarning: No d
ata for colormapping provided via 'c'. Parameters 'cmap' will be ignored
  ax.scatter(X0, X1, c=color, cmap=plt.cm.coolwarm, s=10, edgecolors='face')  # Plot
training points
```

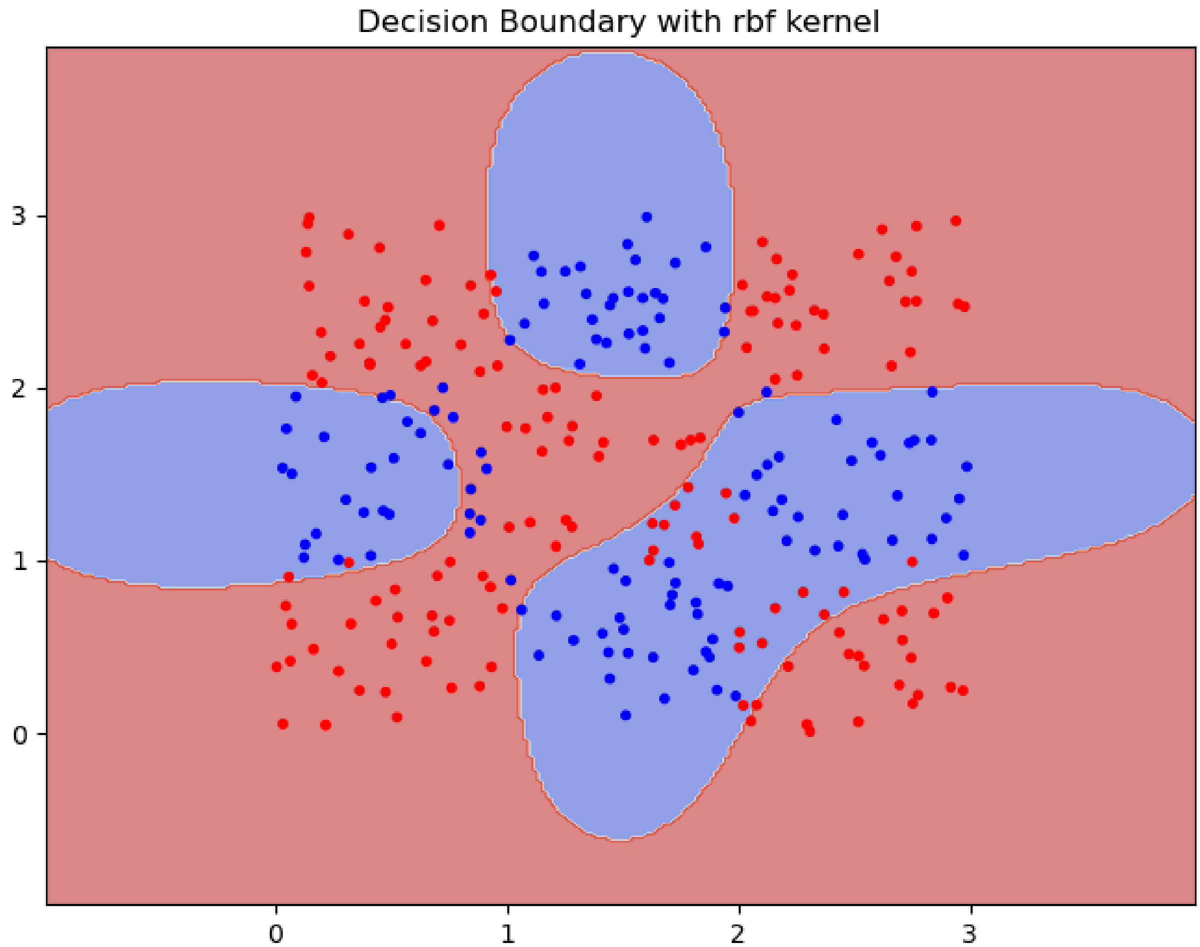



```
Accuracy on the test set with rbf kernel: 0.8889
              precision    recall  f1-score   support

          -1       0.86      0.90      0.88       120
           1       0.92      0.88      0.90       150

    accuracy                           0.89       270
   macro avg       0.89      0.89      0.89       270
weighted avg       0.89      0.89      0.89       270
```

```
C:\Users\steli\AppData\Local\Temp\ipykernel_20700\438884638.py:62: UserWarning: No d
ata for colormapping provided via 'c'. Parameters 'cmap' will be ignored
  ax.scatter(X0, X1, c=color, cmap=plt.cm.coolwarm, s=10, edgecolors='face')  # Plot
training points
```

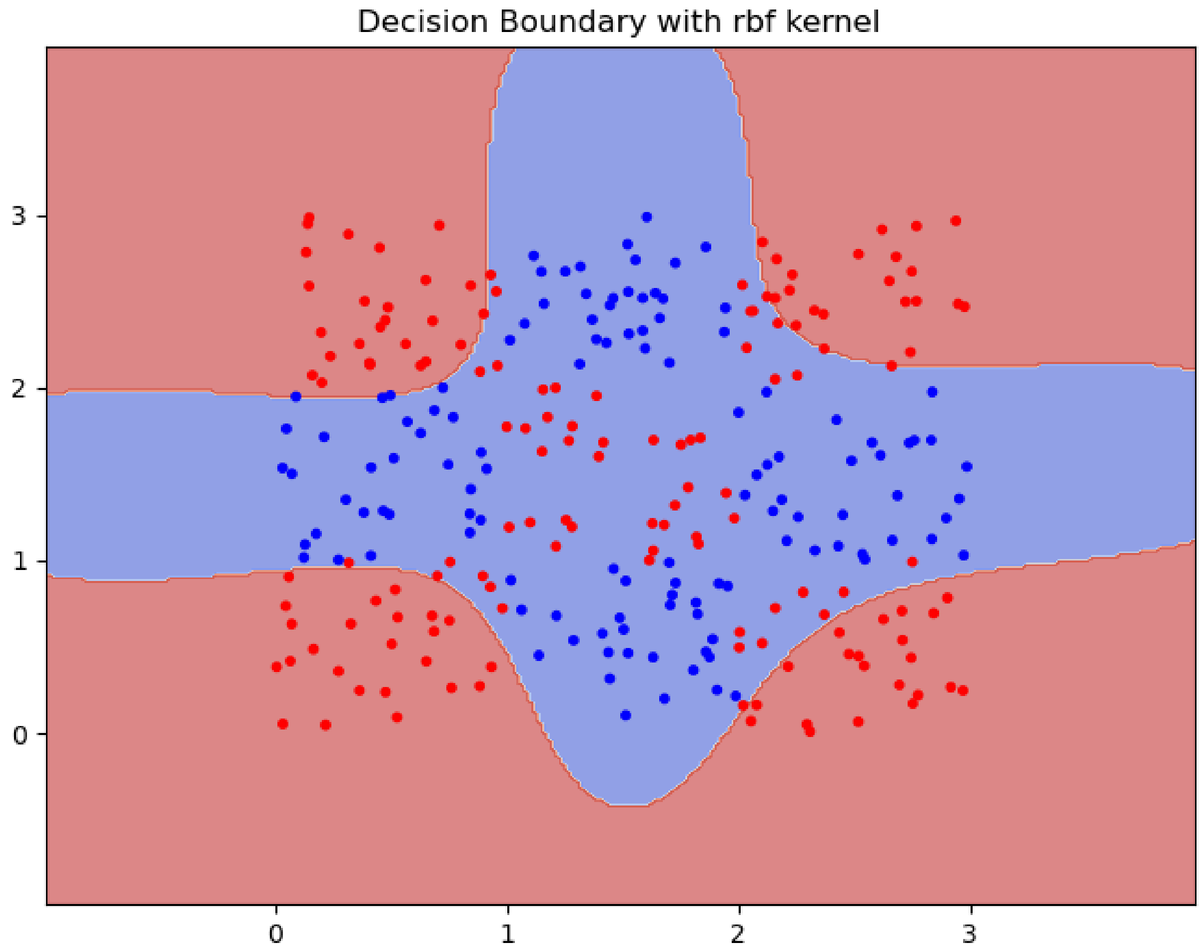



```
Accuracy on the test set with rbf kernel: 0.8000
              precision    recall  f1-score   support

          -1       0.70      0.97      0.81       120
           1       0.96      0.67      0.79       150

    accuracy                           0.80       270
   macro avg       0.83      0.82      0.80       270
weighted avg       0.84      0.80      0.80       270
```


```
C:\Users\steli\AppData\Local\Temp\ipykernel_20700\438884638.py:62: UserWarning: No d
ata for colormapping provided via 'c'. Parameters 'cmap' will be ignored
  ax.scatter(X0, X1, c=color, cmap=plt.cm.coolwarm, s=10, edgecolors='face')  # Plot
training points
```

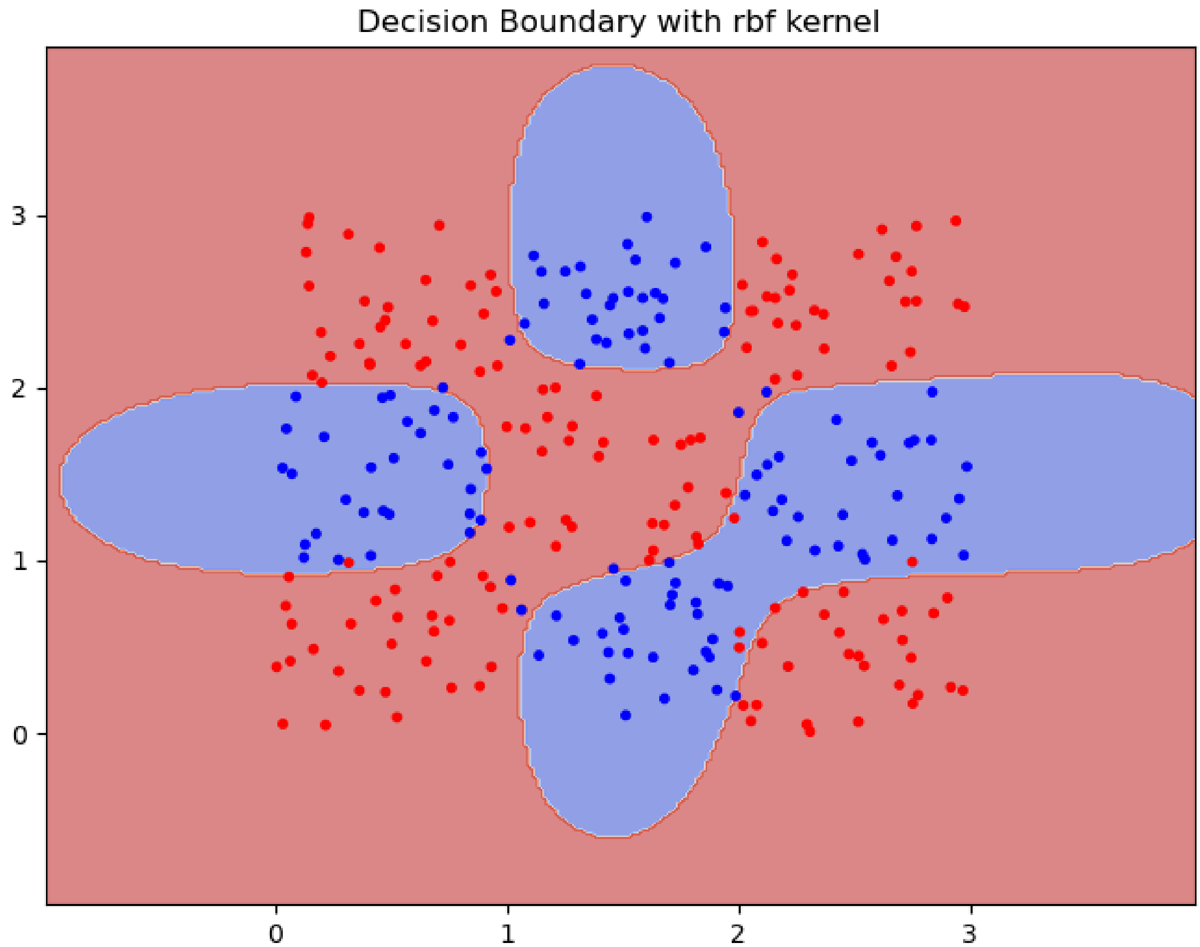



```
Accuracy on the test set with rbf kernel: 0.9185
              precision    recall  f1-score   support

          -1       0.95      0.87      0.90       120
           1       0.90      0.96      0.93       150

    accuracy                           0.92       270
   macro avg       0.92      0.91      0.92       270
weighted avg       0.92      0.92      0.92       270
```

```
C:\Users\steli\AppData\Local\Temp\ipykernel_20700\438884638.py:62: UserWarning: No d
ata for colormapping provided via 'c'. Parameters 'cmap' will be ignored
  ax.scatter(X0, X1, c=color, cmap=plt.cm.coolwarm, s=10, edgecolors='face')  # Plot
training points
```

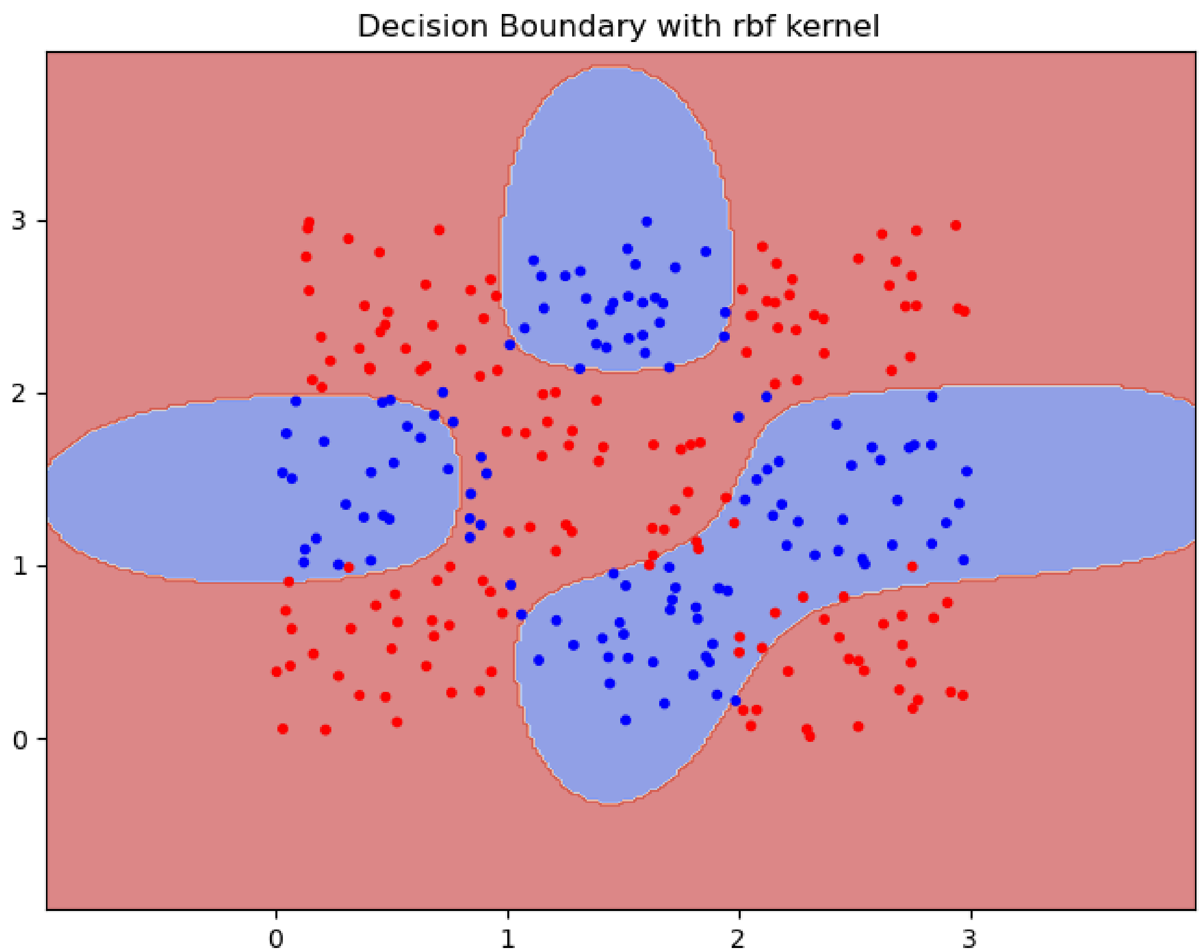



```
Accuracy on the test set with rbf kernel: 0.8963
              precision    recall  f1-score   support

          -1       0.92      0.84      0.88       120
           1       0.88      0.94      0.91       150

    accuracy                           0.90       270
   macro avg       0.90      0.89      0.89       270
weighted avg       0.90      0.90      0.90       270
```

```
C:\Users\steli\AppData\Local\Temp\ipykernel_20700\438884638.py:62: UserWarning: No d
ata for colormapping provided via 'c'. Parameters 'cmap' will be ignored
  ax.scatter(X0, X1, c=color, cmap=plt.cm.coolwarm, s=10, edgecolors='face')  # Plot
training points
```



Accuracy Results for Each Kernel:

- **Linear Kernel**:
  - Accuracy: `[0.556, 0.556, 0.556]` (for `C` values of 0.1, 1.0, and 10.0)
  - The performance of the linear kernel is consistently low, with an accuracy of about **55.6%** for all values of `C`. This suggests that the data is not linearly separable, and a more complex kernel is required to capture the underlying patterns.
- **Polynomial Kernel**:
  - Accuracy: `[0.556, 0.644, 0.681, 0.556, 0.644, 0.674, 0.556, 0.644, 0.681]` (for varying `C` and `degree` values)
  - The polynomial kernel performs better than the linear kernel, with the accuracy improving up to **68.1%** when `C=10` and `degree=3`. However, the performance fluctuates slightly based on the choice of `C` and `degree`. It seems to perform best for certain parameter combinations but still doesn't outperform the RBF kernel.
- **RBF Kernel**:
  - Accuracy: `[0.556, 0.556, 0.889, 0.8, 0.919, 0.896]` (for varying `C` and `gamma` values)

- The RBF kernel performs significantly better than both the linear and polynomial kernels. The highest accuracy of **91.9%** is achieved with `C=10` and `gamma=scale`, demonstrating that the RBF kernel is much better at capturing the non-linear relationships in the data.

Comments on the Results:

1. **Linear Kernel**:

    - The **linear kernel** consistently provides poor performance (around **55.6%** accuracy). This suggests that the dataset is not linearly separable, and using a linear kernel is not suitable for this task.
2. **Polynomial Kernel**:

    - The **polynomial kernel** performs better than the linear kernel, reaching an accuracy of **68.1%** at best. This shows that the polynomial kernel is more flexible, but its performance is still limited compared to the RBF kernel. The performance fluctuates with different values of `C` and `degree`.
3. **RBF Kernel**:

    - The **RBF kernel** outperforms both the linear and polynomial kernels by a large margin. With an accuracy of **91.9%** (the highest value), the RBF kernel is clearly the best choice for this dataset. The RBF kernel can capture complex non-linear patterns in the data, which is why it performs so much better.

Conclusion:

- **Best Performance**: The **RBF kernel** with `C=10` and `gamma=scale` provides the best accuracy of **91.9%**, making it the most suitable kernel for this dataset.
- **Linear Kernel**: The linear kernel is unsuitable for this dataset, with consistently low accuracy.
- **Polynomial Kernel**: While the polynomial kernel performs better than the linear one, its performance is still not as good as the RBF kernel.

Therefore, the **RBF kernel** is the recommended choice for this problem.