

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.svm import SVC
import data
```

```
class KSVMWrap:
```

```
    """
    Metode:
    __init__(self, X, Y_, param_svm_c=1, param_svm_gamma='auto'):
        Konstruira omotač i uči RBF SVM klasifikator
        X, Y_:      podatci i točni indeksi razreda
        param_svm_c:  relativni značaj podatkovne cijene
        param_svm_gamma: širina RBF jezgre
    """
```

```
def __init__(self, X, Y_, param_svm_c=1, param_svm_gamma='auto'):
    self.model = SVC(C=param_svm_c, gamma=param_svm_gamma)
    self.model.fit(X, Y_)
```

```
def predict(self, X):
    return self.model.predict(X)
```

```
def get_scores(self, X):
    return self.model.decision_function(X)
```

```
def support(self):
    return self.model.support_
```

```
if __name__ == "__main__":
    np.random.seed(100)
    X, Y_ = data.sample_gmm_2d(6, 2, 10)
    svm = KSVMWrap(X, Y_)
    Y = svm.predict(X)

    Y_ = np.hstack(Y_)
    accuracy, pr, M = data.eval_perf_multi(Y, Y_)
    print("Accuracy: ", accuracy)
    print("Precision / Recall: ", pr)
    print("Confussion Matrix: ", M)

    bounding_box = (np.min(X, axis=0), np.max(X, axis=0))
    data.graph_surface(svm.get_scores, bounding_box, offset=0.5)
    data.graph_data(X, Y_, Y, special=svm.support())
    plt.show()
```