

```
In [16]: import numpy as np
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

```
In [17]: iris=datasets.load_iris()
```

```
In [18]: x=iris.data[:100,:2]
y=iris.target[:100]
```

```
In [19]: x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=
```

```
In [20]: svm_classifier=SVC(kernel='linear',C=1.0)
svm_classifier.fit(x_train,y_train)
```

```
Out[20]: SVC(kernel='linear')
```

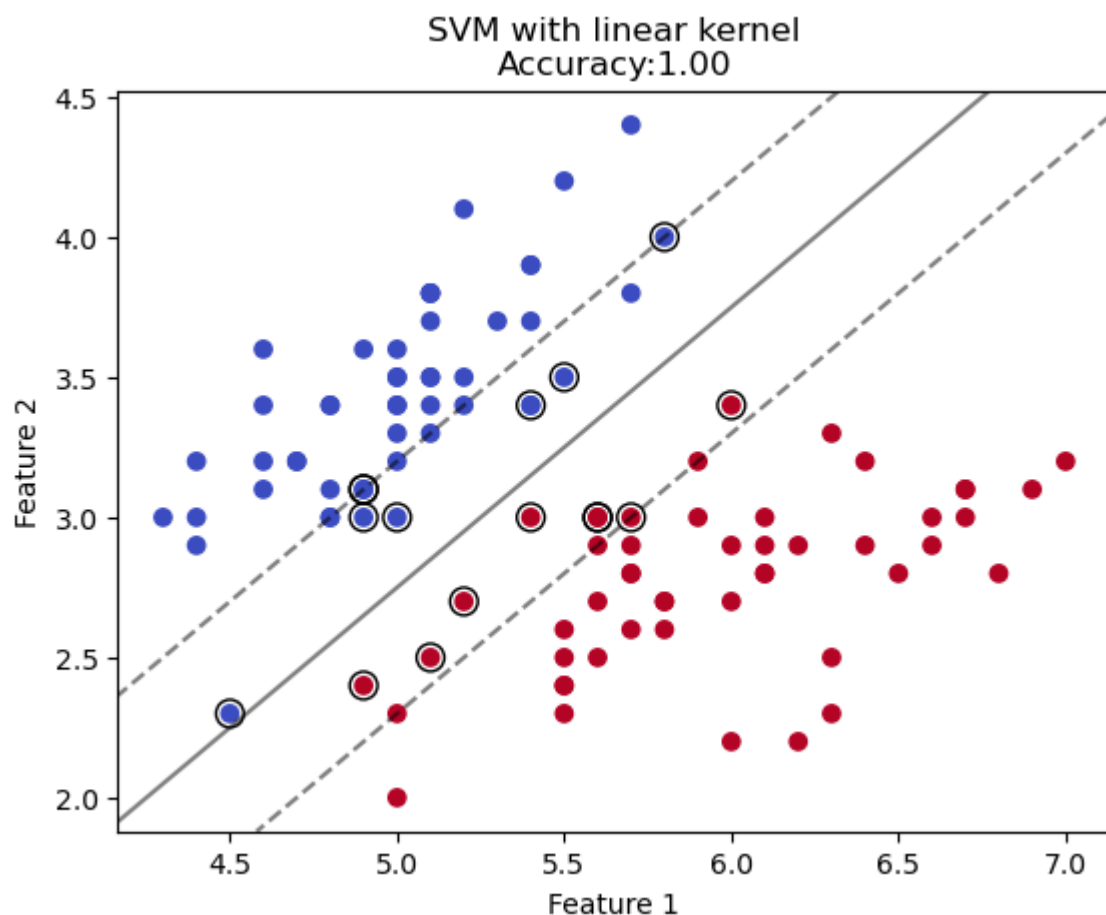
```
In [21]: y_pred=svm_classifier.predict(x_test)
```

```
In [22]: accuracy=accuracy_score(y_test,y_pred)
```

```
In [23]: print("Accuracy:",accuracy)
```

```
Accuracy: 1.0
```

```
In [30]: plt.scatter(x[:,0],x[:,1],c=y,cmap='coolwarm')
ax=plt.gca()
xlim=ax.get_xlim()
ylim=ax.get_ylim()
xx,yy=np.meshgrid(np.linspace(xlim[0],xlim[1],50),
                  np.linspace(ylim[0],ylim[1],50))
Z=svm_classifier.decision_function(np.c_[xx.ravel(),yy.ravel()])
Z=Z.reshape(xx.shape)
plt.contour(xx,yy,Z,colors='k',levels=[-1,0,1],alpha=0.5,linestyles=['--','-'],
plt.scatter(svm_classifier.support_vectors[:,0],svm_classifier.support_vector
plt.title(f'SVM with linear kernel\nAccuracy:{accuracy:.2f}')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.show()
```



In []: