```
import sklearn
import matplotlib.pyplot as plt
from sklearn import svm
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC


from sklearn.datasets import load_iris
iris = load_iris()
y1=iris.target
x1= iris.data


df = pd.DataFrame(x1,
                  columns = iris.feature_names)


df['species']=y1



df=df.drop(columns=["sepal width (cm)","sepal length (cm)"])


df
```

	petal length (cm)	petal width (cm)	species
0	1.4	0.2	0
1	1.4	0.2	0
2	1.3	0.2	0
3	1.5	0.2	0
4	1.4	0.2	0
...	...	...	...
145	5.2	2.3	2
146	5.0	1.9	2
147	5.2	2.0	2
148	5.4	2.3	2
149	5.1	1.8	2

150 rows × 3 columns

```
X=df[['petal length (cm)','petal width (cm)']]
X=X.to_numpy()



X=df[['petal length (cm)','petal width (cm)']]
```

```
X=X.to_numpy()
X=X[50:150,:]
y= df['species']
y=y.to_numpy()
y=y[50:150]
X,x_test,y, y_test=train_test_split(X,y,test_size=0.30)
model=SVC(kernel='linear')
model.fit(X, y)
pred=model.predict(x_test)
from sklearn.metrics import classification_report
print(classification_report(y_test, pred))
clf = svm.SVC(kernel='linear', C=1000)
clf.fit(X, y)

plt.scatter(X[:, 0], X[:, 1], c=y,s=30, cmap=plt.cm.Paired)


ax = plt.gca()
xlim = ax.get_xlim()
ylim = ax.get_ylim()


xx = np.linspace(xlim[0], xlim[1], 30)
yy = np.linspace(ylim[0], ylim[1], 30)
YY, XX = np.meshgrid(yy, xx)
xy = np.vstack([XX.ravel(), YY.ravel()]).T
Z = clf.decision_function(xy).reshape(XX.shape)

ax.contour(XX, YY, Z, colors='k', levels=[-1, 0, 1], alpha=0.5,
           linestyles=['--', '-', '--'])

ax.scatter(clf.support_vectors_[:, 0], clf.support_vectors_[:, 1], s=100,
           linewidth=1, facecolors='none', edgecolors='k')
plt.title("Class 1 and 2")
plt.show()
```

```
                  precision    recall  f1-score   support

               1       1.00      0.87      0.93        15
               2       0.88      1.00      0.94        15
```

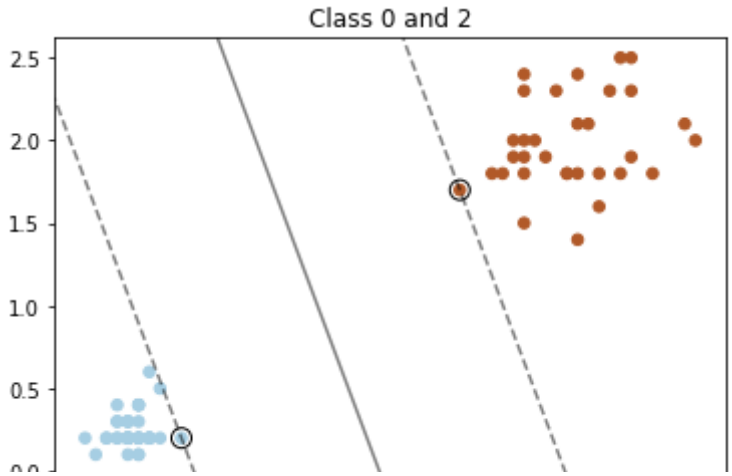
```
X=df[['petal length (cm)','petal width (cm)']]
X=X.to_numpy()
X=X[0:100,:]
y= df['species']
y=y.to_numpy()
y=y[0:100]
X,x_test,y, y_test=train_test_split(X,y,test_size=0.30)
model=SVC(kernel='linear')
model.fit(X, y)
pred=model.predict(x_test)
from sklearn.metrics import classification_report
print(classification_report(y_test, pred))
clf = svm.SVC(kernel='linear', C=1000)
clf.fit(X, y)

plt.scatter(X[:, 0], X[:, 1], c=y,s=30, cmap=plt.cm.Paired)


ax = plt.gca()
xlim = ax.get_xlim()
ylim = ax.get_ylim()

xx = np.linspace(xlim[0], xlim[1], 30)
yy = np.linspace(ylim[0], ylim[1], 30)
YY, XX = np.meshgrid(yy, xx)
xy = np.vstack([XX.ravel(), YY.ravel()]).T
#print(xy,clf.decision_function(xy).shape)
Z = clf.decision_function(xy).reshape(XX.shape)


ax.contour(XX, YY, Z, colors='k', levels=[-1, 0, 1], alpha=0.5,
           linestyles=['--', '-', '--'])

ax.scatter(clf.support_vectors_[:, 0], clf.support_vectors_[:, 1], s=100,
           linewidth=1, facecolors='none', edgecolors='k')
plt.title("Class 0 and 1")
plt.show()
```

```
                  precision    recall  f1-score   support

               0       1.00      1.00      1.00        16
               1       1.00      1.00      1.00        14

        accuracy                           1.00        30
       macro avg       1.00      1.00      1.00        30
    weighted avg       1.00      1.00      1.00        30
```

Class 0 and 1

```
X=df[['petal length (cm)','petal width (cm)']]
X=X.to_numpy()
a=X[0:50,:]
X=X[100:150,:]
X=np.concatenate((X,a))
y= df['species']
y=y.to_numpy()
b=y[0:50]
y=y[100:150]
y=np.concatenate((y,b))
X,x_test,y, y_test=train_test_split(X,y,test_size=0.30)
model=SVC(kernel='linear')
model.fit(X, y)
pred=model.predict(x_test)
from sklearn.metrics import classification_report
print(classification_report(y_test, pred))

clf = svm.SVC(kernel='linear', C=1000)
clf.fit(X, y)

plt.scatter(X[:, 0], X[:, 1], c=y,s=30, cmap=plt.cm.Paired)

ax = plt.gca()
xlim = ax.get_xlim()
ylim = ax.get_ylim()


xx = np.linspace(xlim[0], xlim[1], 30)
yy = np.linspace(ylim[0], ylim[1], 30)
YY, XX = np.meshgrid(yy, xx)
xy = np.vstack([XX.ravel(), YY.ravel()]).T
#print(xy,clf.decision_function(xy).shape)
Z = clf.decision_function(xy).reshape(XX.shape)


ax.contour(XX, YY, Z, colors='k', levels=[-1, 0, 1], alpha=0.5,
           linestyles=['--', '-', '--'])

ax.scatter(clf.support_vectors_[:, 0], clf.support_vectors_[:, 1], s=100,
           linewidth=1, facecolors='none', edgecolors='k')
plt.title("Class 0 and 2")
plt.show()
```

```
                   precision    recall  f1-score   support

                0       1.00      1.00      1.00        12
                2       1.00      1.00      1.00        18

         accuracy                           1.00        30
        macro avg       1.00      1.00      1.00        30
     weighted avg       1.00      1.00      1.00        30
```

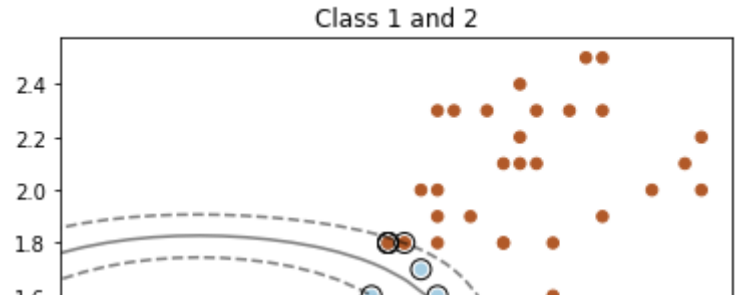



```
X=df[['petal length (cm)','petal width (cm)']]
X=X.to_numpy()
y= df['species']
y=y.to_numpy()
X,x_test,y, y_test=train_test_split(X,y,test_size=0.30,random_state=42)
for i in [0.001,1,1000]:
  model=SVC(C=i,kernel='linear')
  model.fit(X, y)
  pred=model.predict(x_test)

  print(classification_report(y_test, pred))
```

```
                  precision    recall  f1-score   support

               0       0.00      0.00      0.00        19
               1       0.41      1.00      0.58        13
               2       1.00      1.00      1.00        13

        accuracy                           0.58        45
       macro avg       0.47      0.67      0.53        45
    weighted avg       0.41      0.58      0.46        45
```

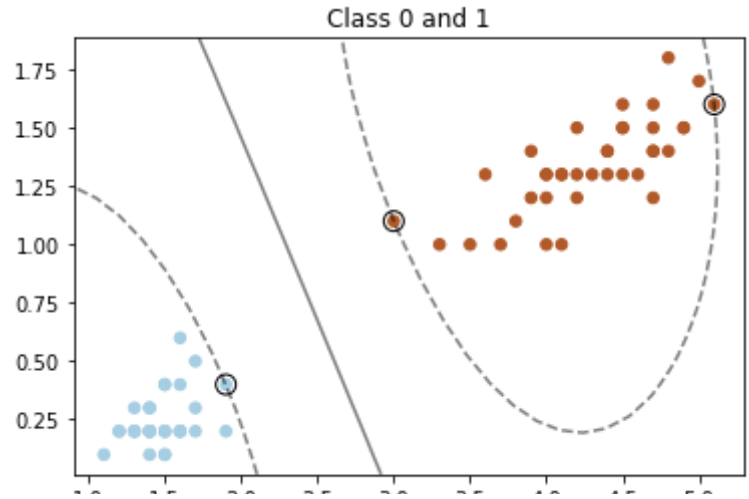
```
X=df[['petal length (cm)','petal width (cm)']]
X=X.to_numpy()
X=X[50:150,:]
y= df['species']
y=y.to_numpy()
y=y[50:150]
X,x_test,y, y_test=train_test_split(X,y,test_size=0.30)
model=SVC()
model.fit(X, y)
pred=model.predict(x_test)
from sklearn.metrics import classification_report
print(classification_report(y_test, pred))
clf = svm.SVC(kernel='rbf', C=1000 )
clf.fit(X, y)

plt.scatter(X[:, 0], X[:, 1], c=y,s=30, cmap=plt.cm.Paired)


ax = plt.gca()
xlim = ax.get_xlim()
ylim = ax.get_ylim()

xx = np.linspace(xlim[0], xlim[1], 30)
yy = np.linspace(ylim[0], ylim[1], 30)
YY, XX = np.meshgrid(yy, xx)
xy = np.vstack([XX.ravel(), YY.ravel()]).T
#print(xy,clf.decision_function(xy).shape)
Z = clf.decision_function(xy).reshape(XX.shape)

ax.contour(XX, YY, Z, colors='k', levels=[-1, 0, 1], alpha=0.5,
           linestyles=['--', '-', '--'])

ax.scatter(clf.support_vectors_[:, 0], clf.support_vectors_[:, 1], s=100,
           linewidth=1, facecolors='none', edgecolors='k')
plt.title("Class 1 and 2")
plt.show()
```

```
                precision    recall  f1-score   support

             1       0.93      1.00      0.97        14
             2       1.00      0.94      0.97        16

      accuracy                           0.97        30
     macro avg       0.97      0.97      0.97        30
  weighted avg       0.97      0.97      0.97        30
```




```
X=df[['petal length (cm)','petal width (cm)']]
X=X.to_numpy()
X=X[0:100,:]
y= df['species']
y=y.to_numpy()
y=y[0:100]
X,x_test,y, y_test=train_test_split(X,y,test_size=0.30)
model=SVC(kernel='rbf')
model.fit(X, y)
pred=model.predict(x_test)
from sklearn.metrics import classification_report
print(classification_report(y_test, pred))
clf = svm.SVC(kernel='rbf', C=1000)
clf.fit(X, y)

plt.scatter(X[:, 0], X[:, 1], c=y,s=30, cmap=plt.cm.Paired)

ax = plt.gca()
xlim = ax.get_xlim()
ylim = ax.get_ylim()


xx = np.linspace(xlim[0], xlim[1], 30)
yy = np.linspace(ylim[0], ylim[1], 30)
YY, XX = np.meshgrid(yy, xx)
xy = np.vstack([XX.ravel(), YY.ravel()]).T
#print(xy,clf.decision_function(xy).shape)
Z = clf.decision_function(xy).reshape(XX.shape)

ax.contour(XX, YY, Z, colors='k', levels=[-1, 0, 1], alpha=0.5,
           linestyles=['--', '-', '--'])

ax.scatter(clf.support_vectors_[:, 0], clf.support_vectors_[:, 1], s=100,
           linewidth=1, facecolors='none', edgecolors='k')
plt.title("Class 0 and 1")
plt.show()
```

```
                  precision    recall  f1-score   support

               0       1.00      1.00      1.00        18
               1       1.00      1.00      1.00        12

        accuracy                           1.00        30
       macro avg       1.00      1.00      1.00        30
    weighted avg       1.00      1.00      1.00        30
```




```
X=df[['petal length (cm)','petal width (cm)']]
X=X.to_numpy()
a=X[0:50,:]
X=X[100:150,:]
X=np.concatenate((X,a))
y= df['species']
y=y.to_numpy()
b=y[0:50]
y=y[100:150]
y=np.concatenate((y,b))
X,x_test,y, y_test=train_test_split(X,y,test_size=0.30)
model=SVC(kernel='rbf')
model.fit(X, y)
pred=model.predict(x_test)
from sklearn.metrics import classification_report
print(classification_report(y_test, pred))

clf = svm.SVC(kernel='rbf', C=1000)
clf.fit(X, y)

plt.scatter(X[:, 0], X[:, 1], c=y,s=30, cmap=plt.cm.Paired)


ax = plt.gca()
xlim = ax.get_xlim()
ylim = ax.get_ylim()

xx = np.linspace(xlim[0], xlim[1], 30)
yy = np.linspace(ylim[0], ylim[1], 30)
YY, XX = np.meshgrid(yy, xx)
xy = np.vstack([XX.ravel(), YY.ravel()]).T

Z = clf.decision_function(xy).reshape(XX.shape)
```

```
ax.contour(XX, YY, Z, colors='k', levels=[-1, 0, 1], alpha=0.5,
           linestyles=['--', '-', '--'])

ax.scatter(clf.support_vectors_[:, 0], clf.support_vectors_[:, 1], s=100,
           linewidth=1, facecolors='none', edgecolors='k')
plt.title("Class 0 and 2")
plt.show()
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        18
           2       1.00      1.00      1.00        12

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30
```