

```

import sklearn as sklearn;
import numpy as np ;
import matplotlib.pyplot as plt ;
from sklearn.neighbors import KNeighborsClassifier
from matplotlib.colors import ListedColormap

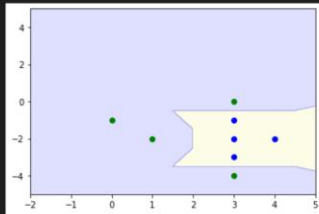
# Create the dataset
X = np.array( [[4,-2],[3,-1],[3,-3],[3,-2],[3,0],[3,-4],[1,-2],[0,-1]]);
y = np.array([0,0,0,0,1,1,1,1]);
knn = KNeighborsClassifier(n_neighbors=1);
knn.fit(X, y);

def plot_decision_boundary(clf , axes):
    xp=np.linspace(axes[0], axes[1], 300) #均匀300个横坐标
    yp=np.linspace(axes[2], axes[3], 300) #均匀300个纵坐标
    x1, y1=np.meshgrid(xp, yp) #生成300x300个点
    xy=np.c_[x1.ravel(), y1.ravel()] #按行拼接，规范成坐标点的格式
    y_pred = clf.predict(xy).reshape(x1.shape) #训练之后平铺
    custom_cmap = ListedColormap(['#fafab0','#9898ff'])
    plt.contourf(x1, y1, y_pred, alpha=0.3, cmap=custom_cmap)

plot_decision_boundary(knn, axes=[-2, 5, -5, 5])
#画两种类型的点
p1=plt.scatter(X[y==0,0], X[y==0, 1], color='blue')
p2=plt.scatter(X[y==1,0], X[y==1, 1], color='green')
#设置注释
plt.show()

```

✓ 1.4s



```

import sklearn as sklearn;
import numpy as np ;
import matplotlib.pyplot as plt ;
from sklearn.neighbors import KNeighborsClassifier
from matplotlib.colors import ListedColormap

# Create the dataset
X = np.array( [[4,-2],[3,-1],[3,-3],[3,-2],[3,0],[3,-4],[1,-2],[0,-1]]);
y = np.array([0,0,0,0,1,1,1,1]);
knn = KNeighborsClassifier(n_neighbors=3);
knn.fit(X, y);

def plot_decision_boundary(clf , axes):
    xp=np.linspace(axes[0], axes[1], 300) #均匀300个横坐标
    yp=np.linspace(axes[2], axes[3], 300) #均匀300个纵坐标
    x1, y1=np.meshgrid(xp, yp) #生成300x300个点
    xy=np.c_[x1.ravel(), y1.ravel()] #按行拼接，规范成坐标点的格式
    y_pred = clf.predict(xy).reshape(x1.shape) #训练之后平铺
    custom_cmap = ListedColormap(['#fafab0','#9898ff'])
    plt.contourf(x1, y1, y_pred, alpha=0.3, cmap=custom_cmap)

plot_decision_boundary(knn, axes=[-2, 5, -5, 5])
#画两种类型的点
p1=plt.scatter(X[y==0,0], X[y==0, 1], color='blue')
p2=plt.scatter(X[y==1,0], X[y==1, 1], color='green')
#设置注释
plt.show()

```

✓ 1.4s

