第三次编程

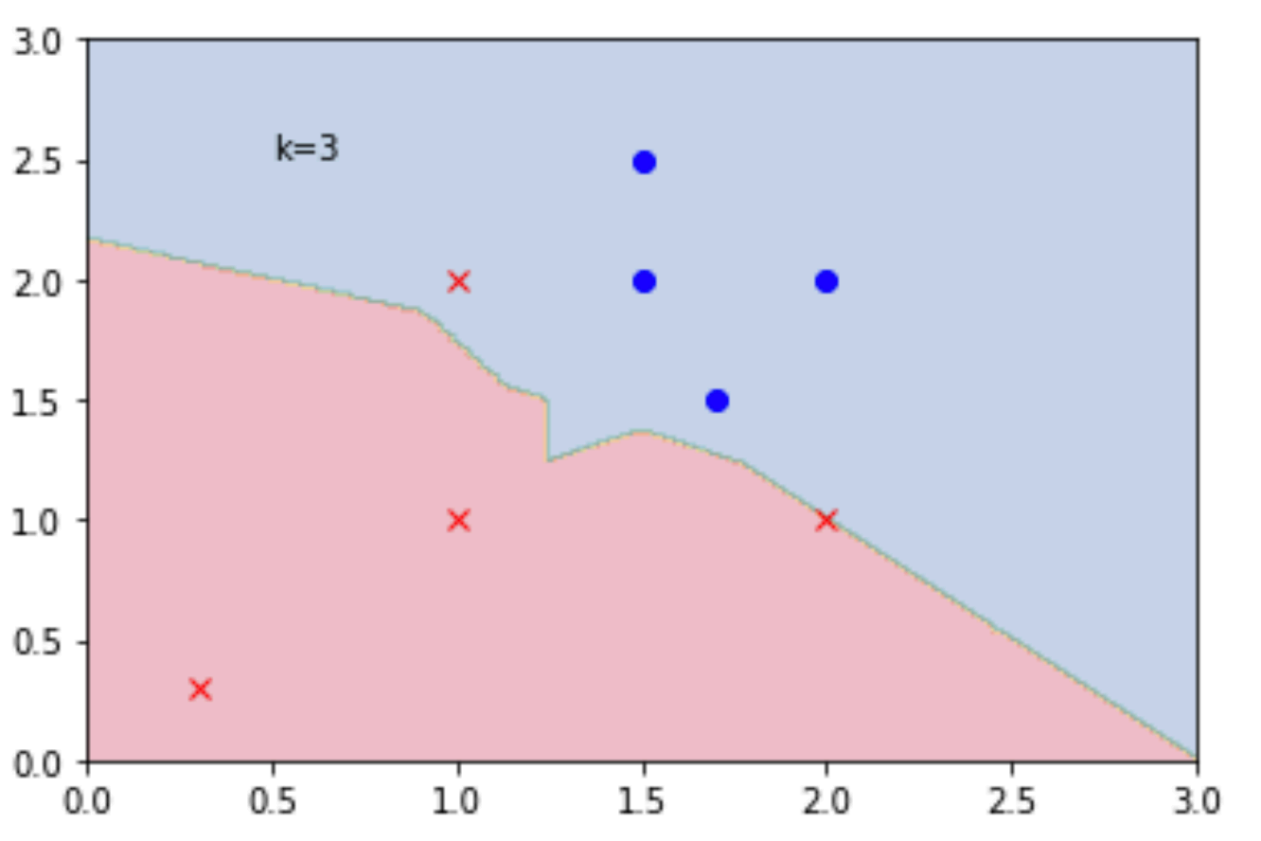
```
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image
# 数据集
p=[[1,0.3,2,1],# 正例
 [2,0.3,1,1]]
n=[[1.5,1.7,2,1.5],# 负例
 [2,1.5,2,2.5]]
p=np.array(p)
n=np.array(n)
def divide(dist,k,X,Y):# dist为一距离函数，k为KNN的参数，(X,Y)为数据的坐标
    ans_p=[np.sort(dist(p[0]-X[i],p[1]-Y[i]))for i in range(len(X))]
    ans_n=[np.sort(dist(n[0]-X[i],n[1]-Y[i]))for i in range(len(X))]
    t=[ans_p[i][int((k-1)/2)]>ans_n[i][int((k-1)/2)]for i in
    range(len(ans_p))]
    return np.array(t)# 返回分类结果
```

```
#欧式距离
def dist1(x,y):
    #return
np.sqrt(np.sum(np.square(x-y)))
    #return np.array(np.linalg.norm(x-
y))
    return
np.array(np.sqrt(np.power(x,2)+np.powe
r(y,2)))
#曼哈顿距离
def dist2(x,y):
    return
np.array(np.abs(x)+np.abs(y))
#切比雪夫距离
def exmaple_dist(x,y):
    return
np.max([np.abs(x),np.abs(y)],axis=0)
def plot(dist,k,ax):# 画图
    N=200
    #生成横坐标
    X=np.linspace(-0,3,N) #在区间内生成N
个间隔相同的数字
    Y=X #生成纵坐标
    #生成N*N个点
```

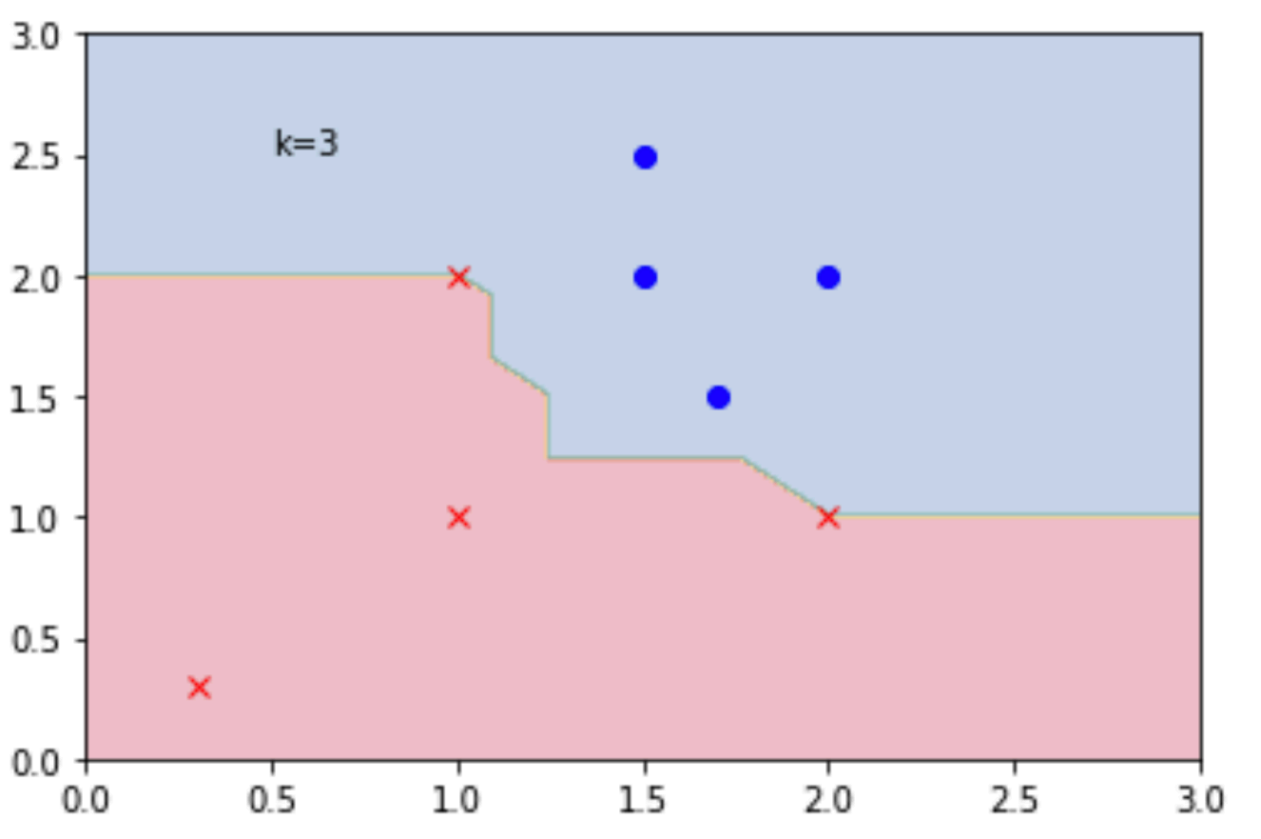
```
    X,Y=np.meshgrid(X,Y) #根据输入的坐标向量生成对应的坐标矩阵
    X=X.reshape(1,N*N)[0] #将横坐标化为向量形式
    Y=Y.reshape(1,N*N)[0]
    #根据模型对生成的N*N个点进行预测
    predict=divide(dist,3,X,Y)
    #绘制图像  根据预测值和对应坐标
    ax.contourf(X.reshape(N,N),
Y.reshape(N,N),
predict.reshape(N,N),cmap=plt.cm.Spectral,alpha=0.3)
    ax.plot(p[0],p[1],'rx')
    ax.plot(n[0],n[1],'bo')
    plt.text(0.5,2.5,"k="+str(k))
#if __name__ == '__main__':
print(dist1(n[0],n[1]))
print(dist2(n[0],n[1]))
print(dist1(p[0],p[1]))
print(dist2(p[0],p[1]))
print(exmaple_dist(p[0],p[1]))
#plot(exmaple_dist,3,plt.subplot(111))
#plot(dist1,3,plt.subplot(111))
plot(dist2,3,plt.subplot(111))
```

k=3效果

欧式距离

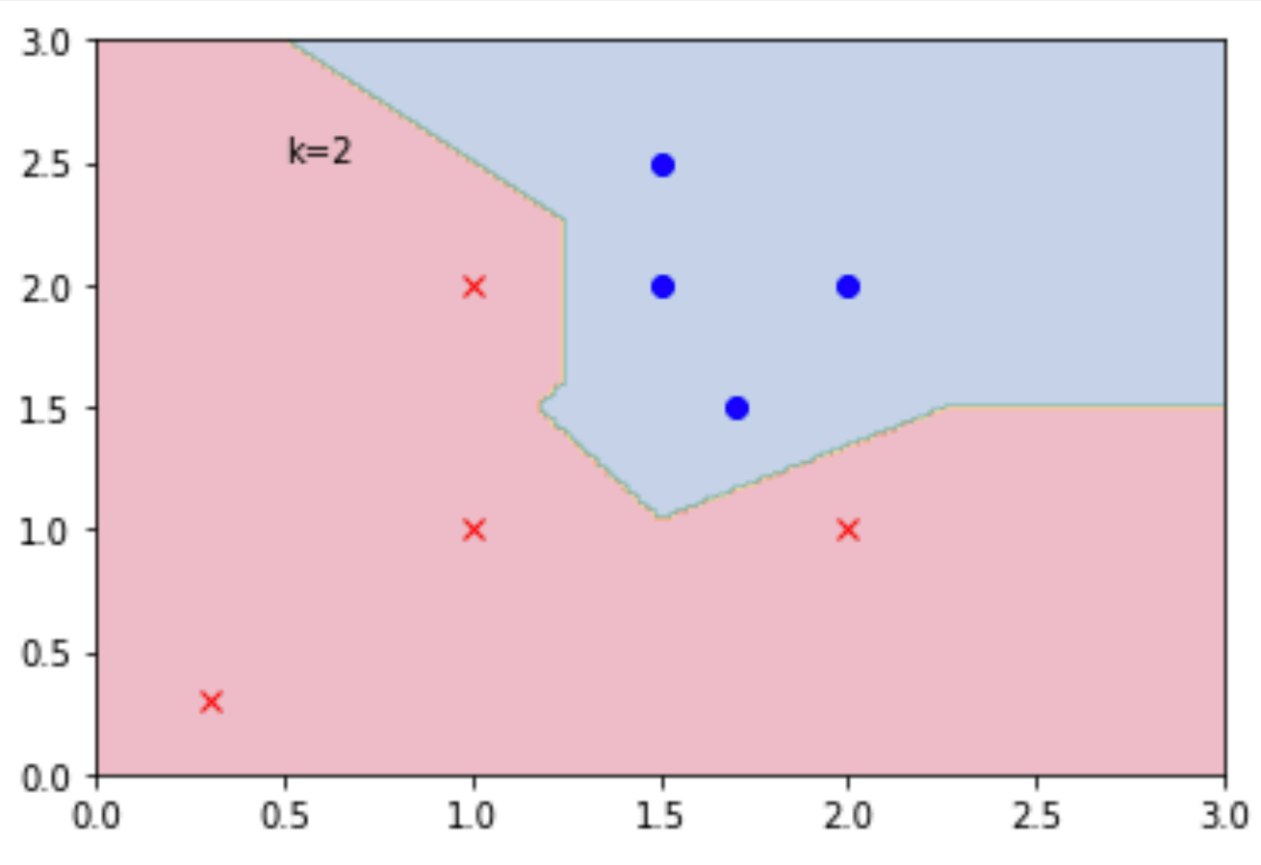



曼哈顿距离

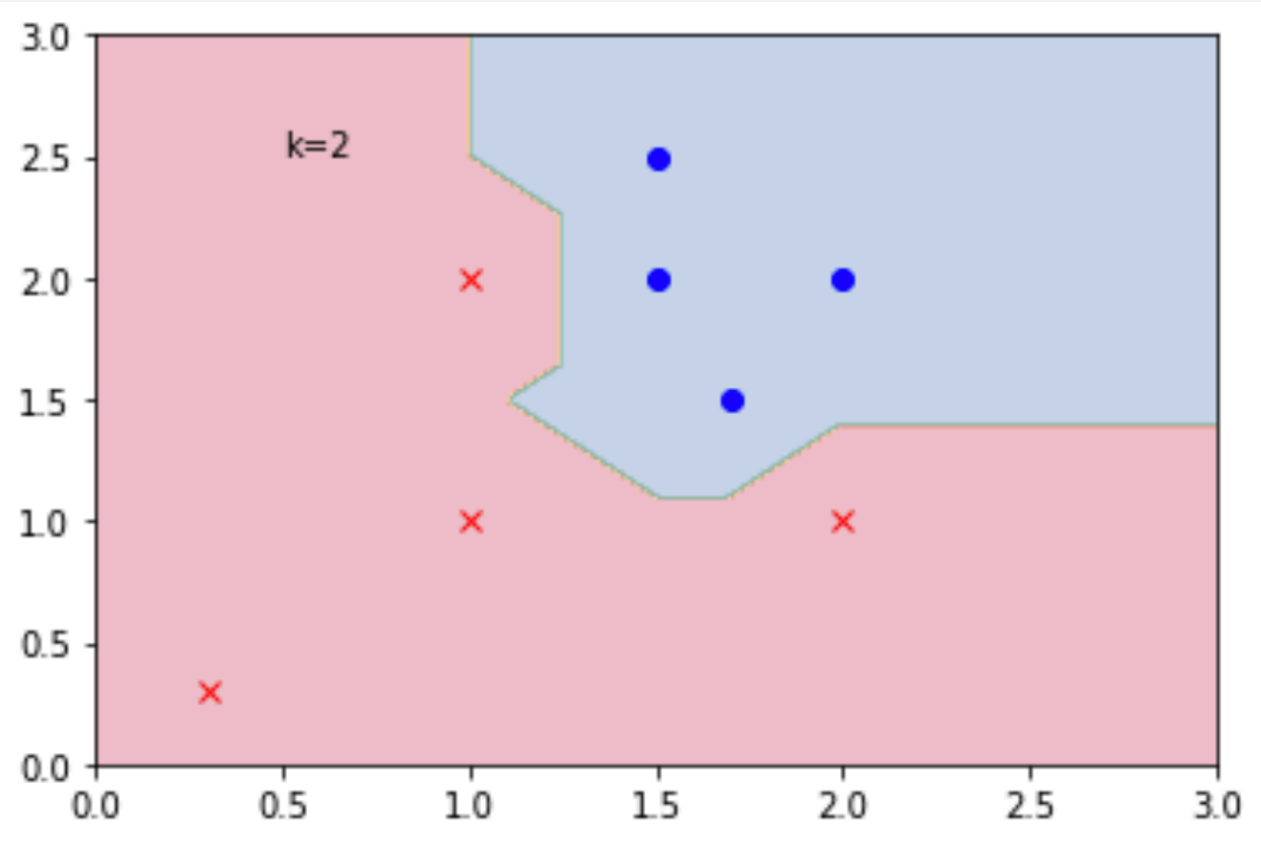



k=2效果

欧式距离


k=2
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0

曼哈顿距离


k=2
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0

切比雪夫距离