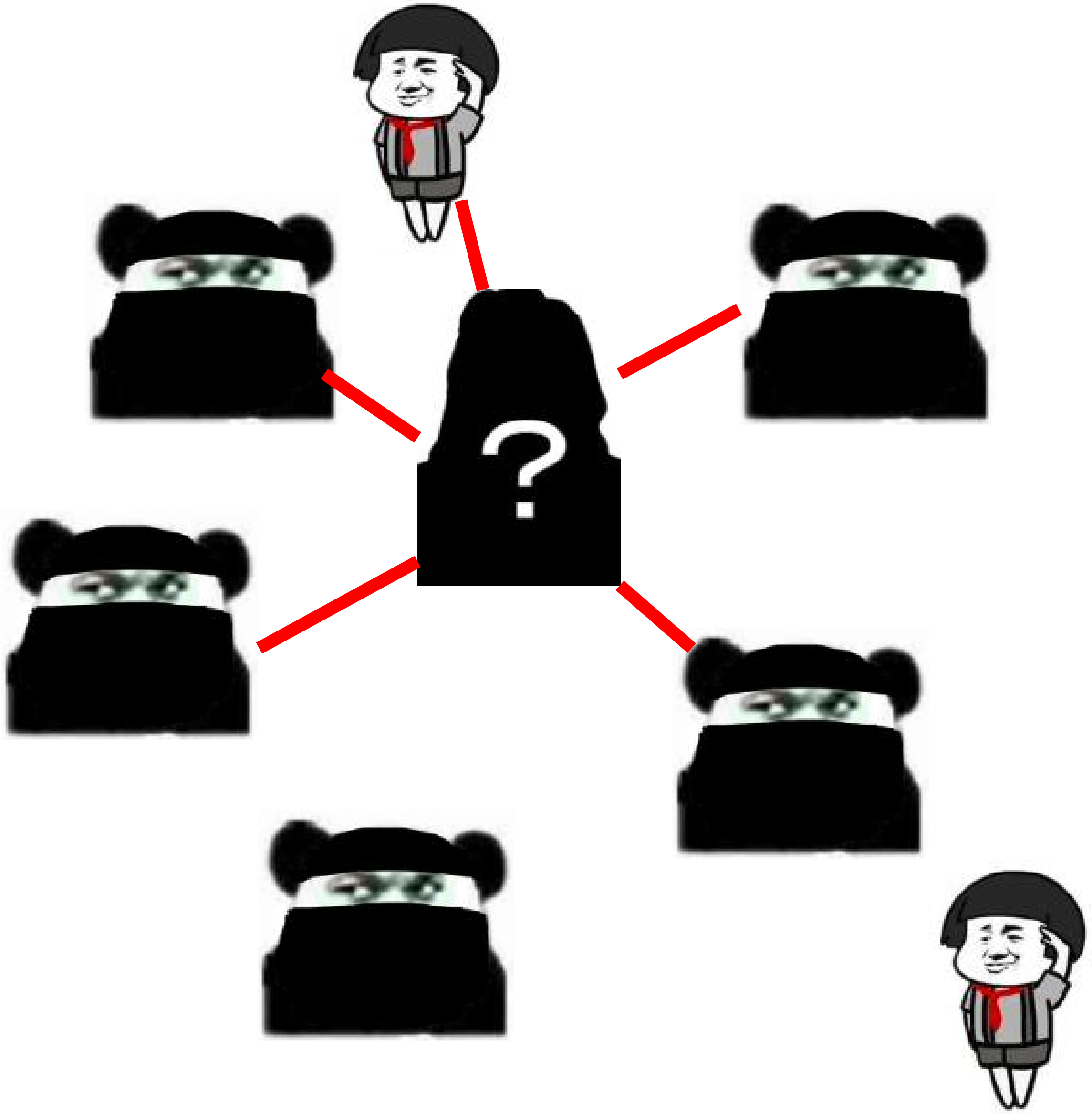
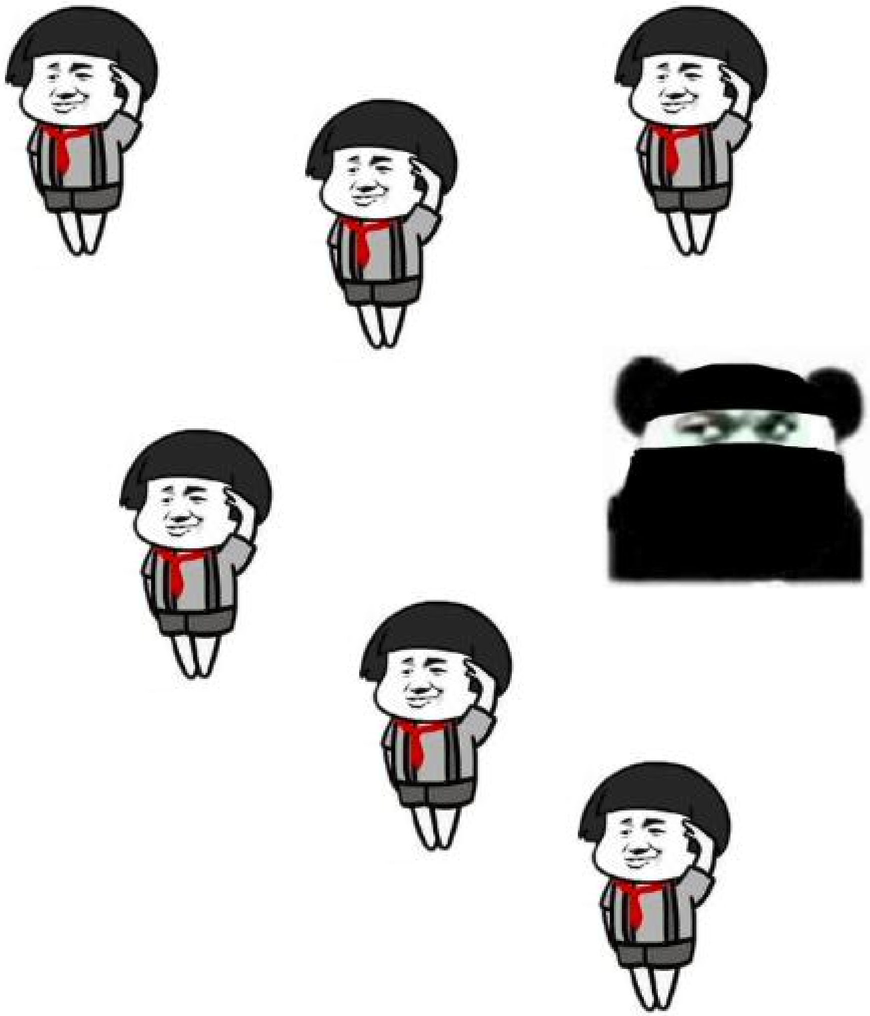
KNN （K-NearestNeighbor）

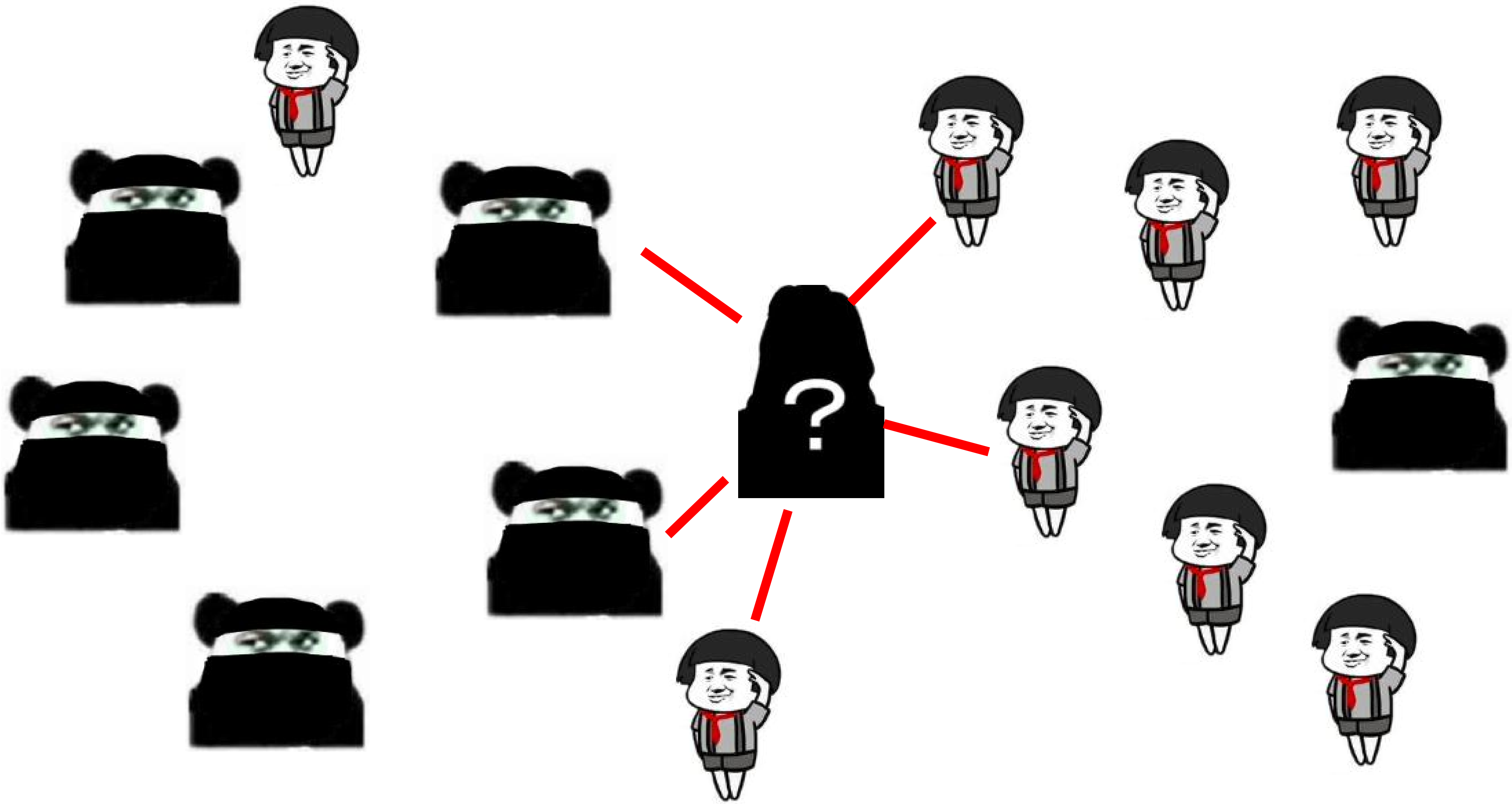
直观的理解

识别可疑人物



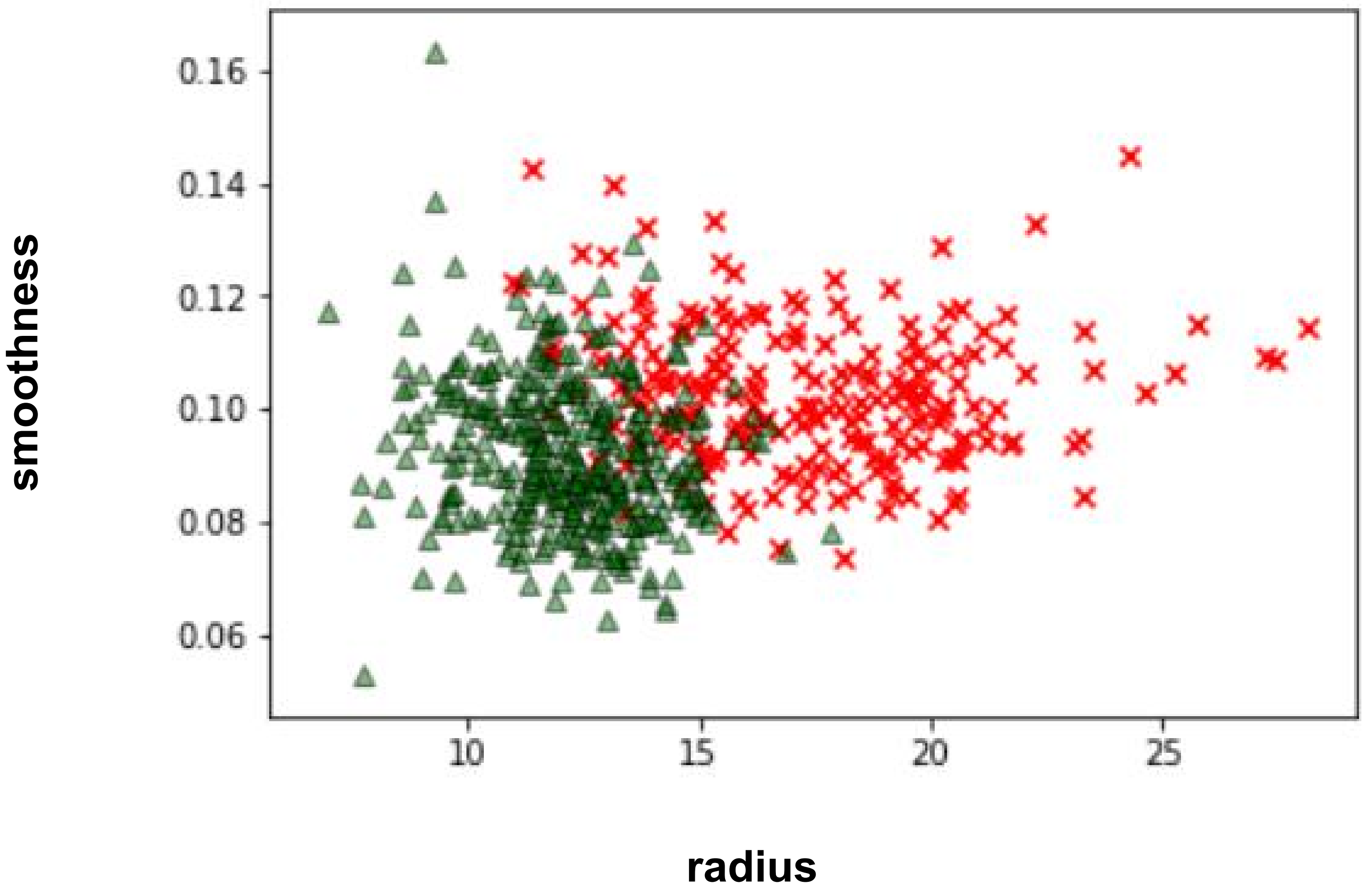


识别可疑人物



使用 KNN 算法

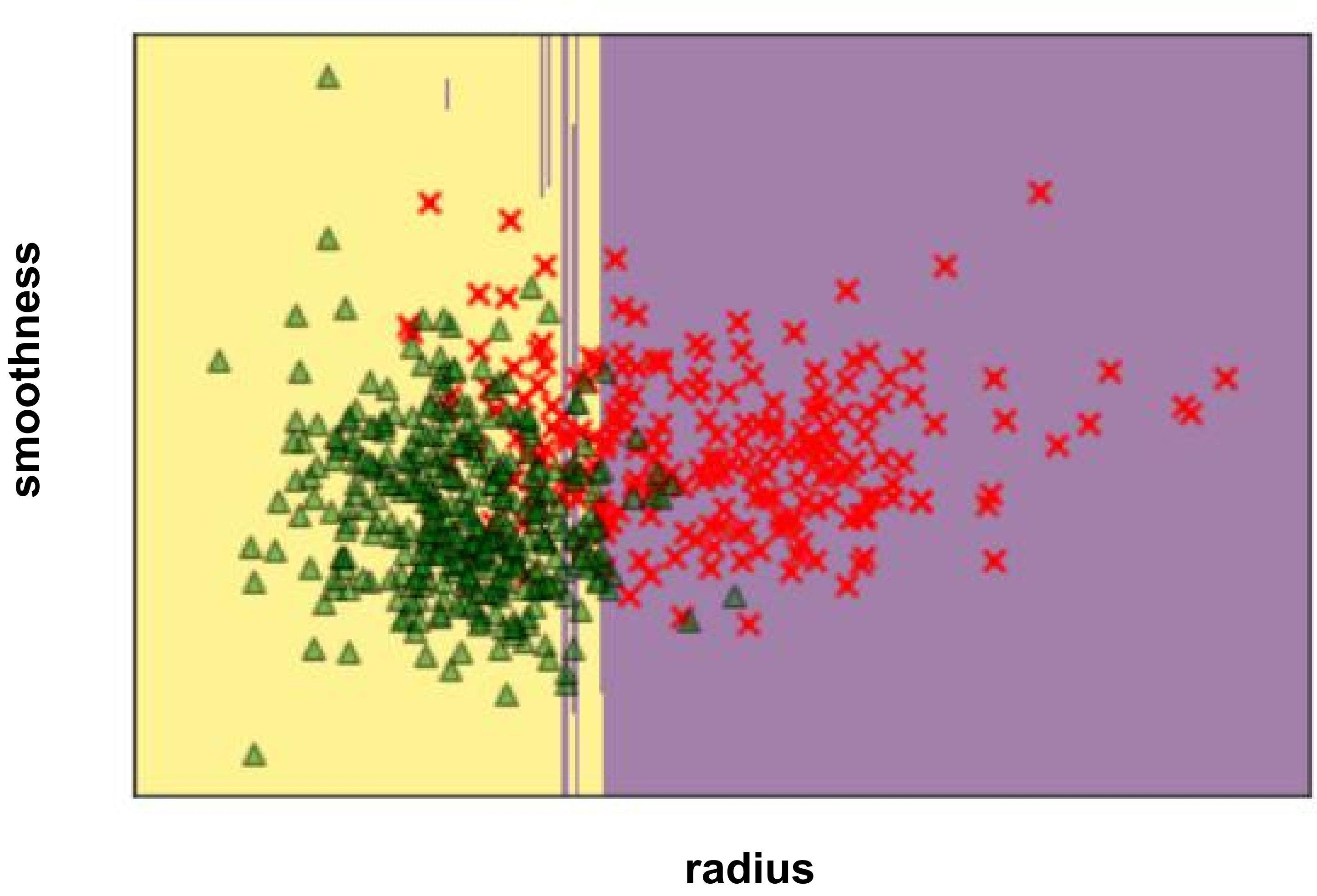
使用 KNN



model = KNeighborsClassifier()
model.fit(X_train, y_train)

training score: 0.9061
test score: 0.8531

决策边界 (decision boundary)



为什么第二个特征没有起到作用?

	0	1
0	17.99	0.1184
1	20.57	0.08474
2	19.69	0.1096
3	11.42	0.1425
4	20.29	0.1003
5	12.45	0.1278
6	18.25	0.09463
7	13.71	0.1189
8	13	0.1273
9	12.46	0.1186
10	16.02	0.08206
11	15.78	0.0971

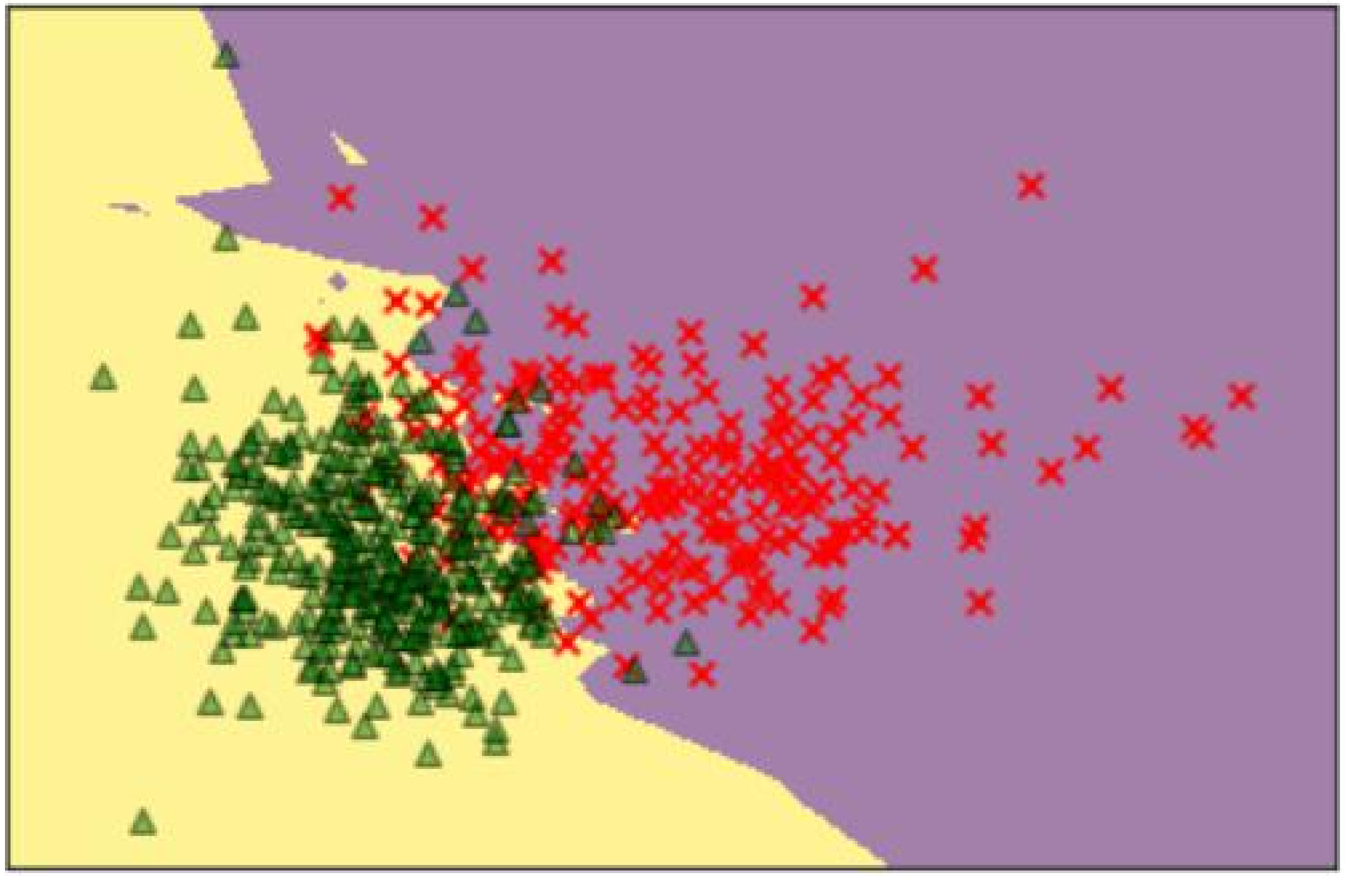
数据归一化

最值归一化

适用于分布有明显边界的情况
缺点：容易受到 outlier 影响

均值方差归一化

归一化后的结果



归一化前

training score: 0.9061

test score: 0.8531

归一化后

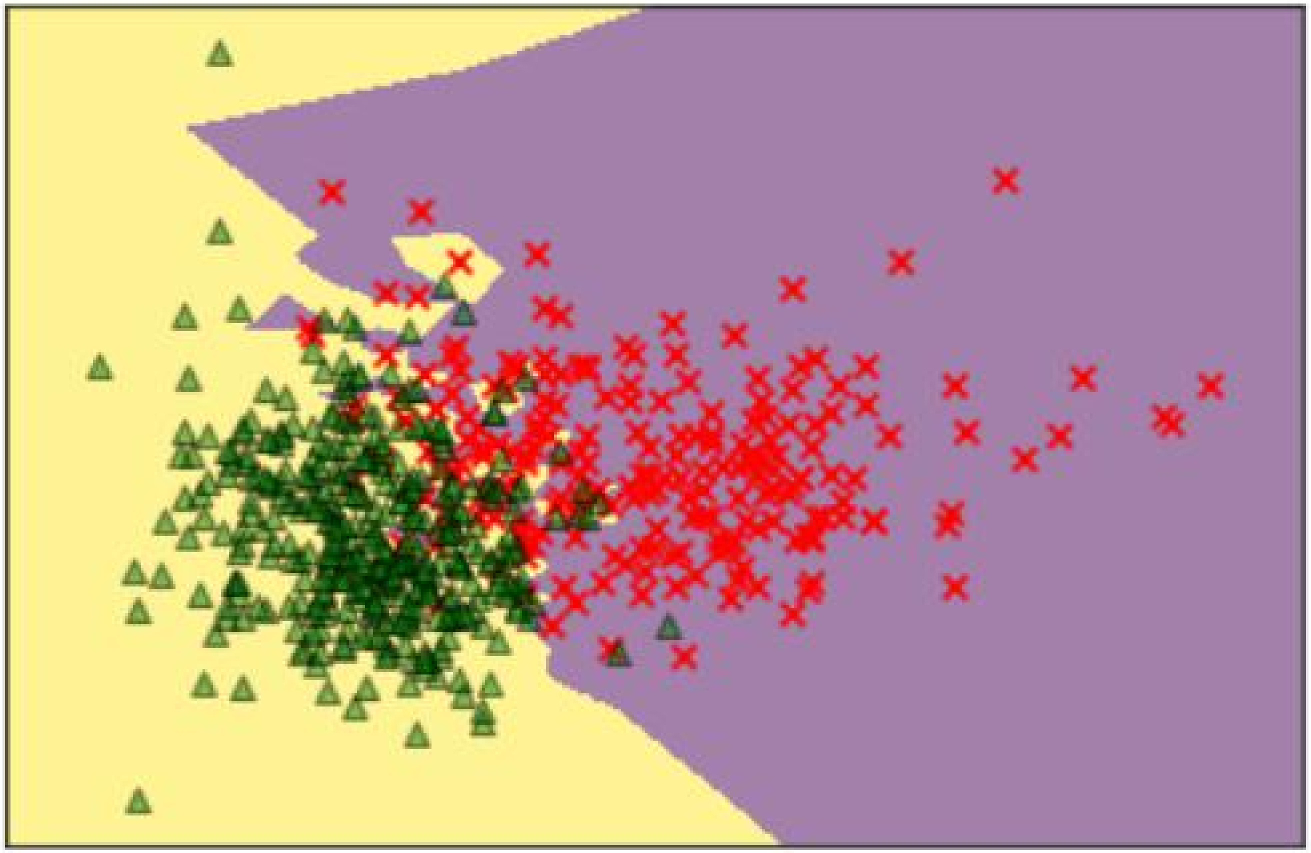
training score: 0.9225

test score: 0.8881

KNN 算法的参数

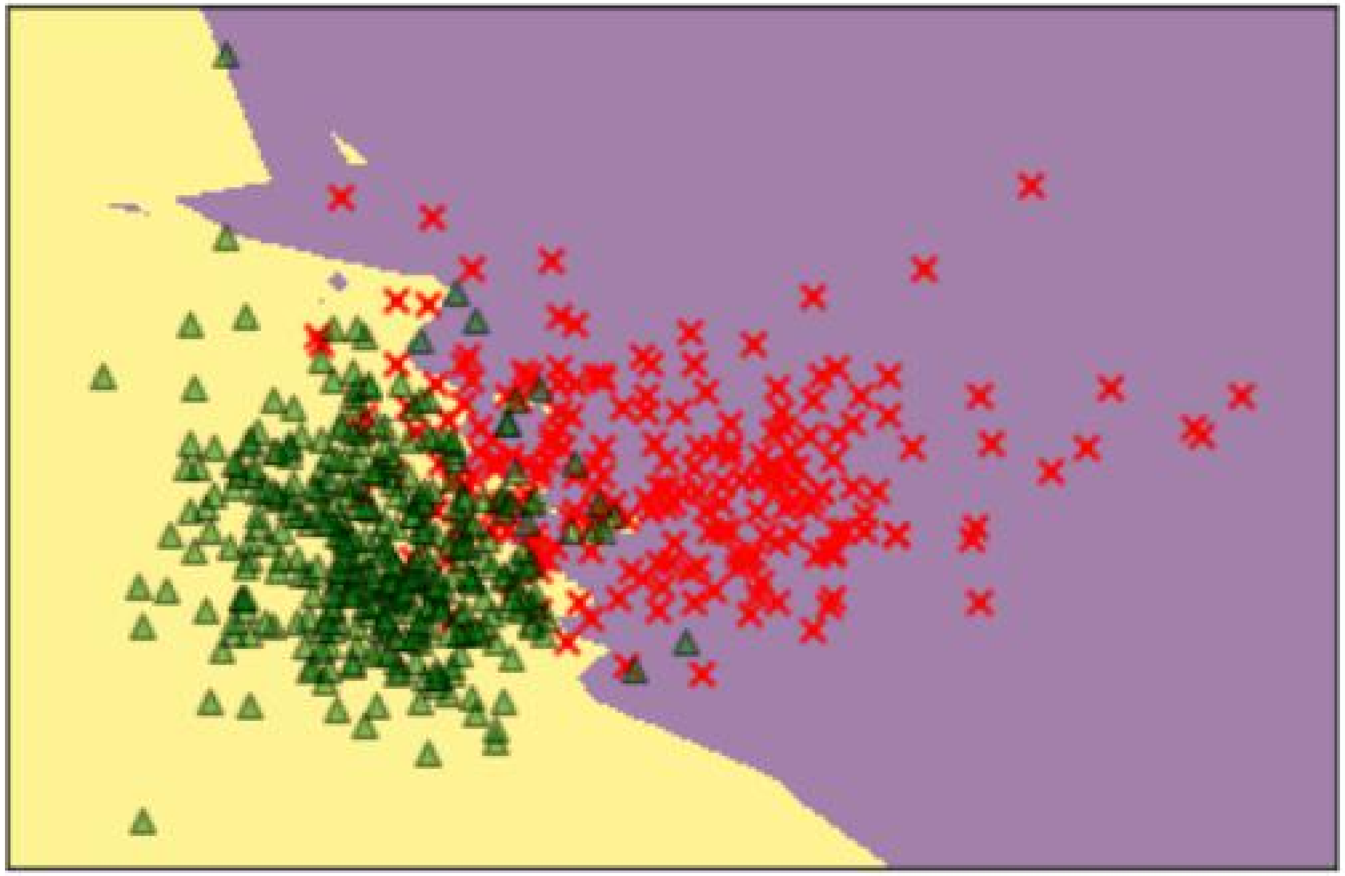
选择参数 K

model = KNeighborsClassifier(n_neighbors=1)



选择参数 K

model = KNeighborsClassifier(n_neighbors=5)



default

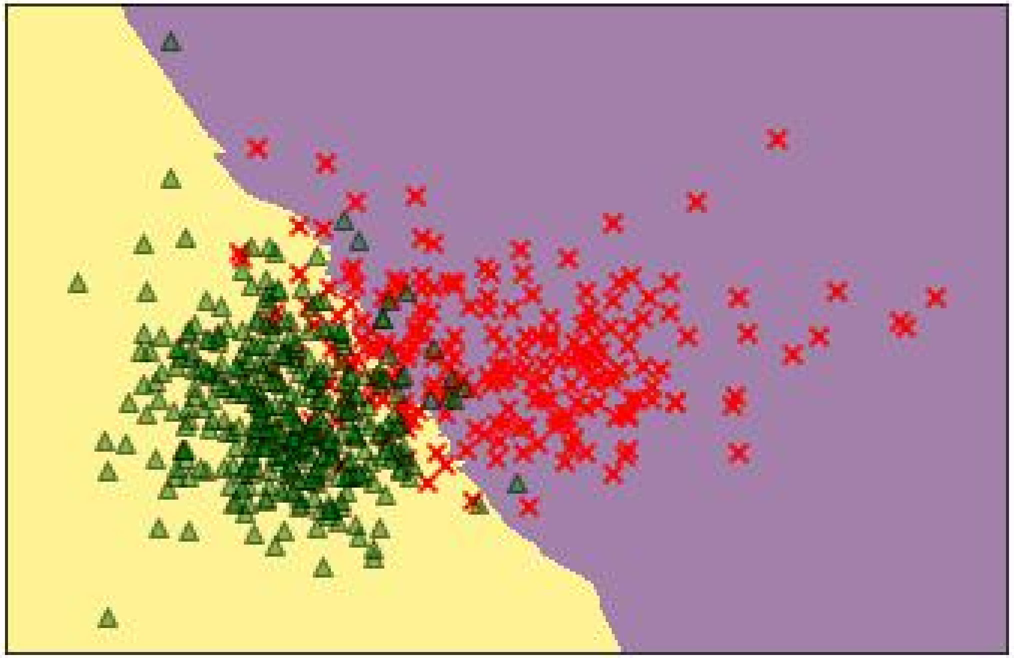
选择参数 K

如果 K 值太大？

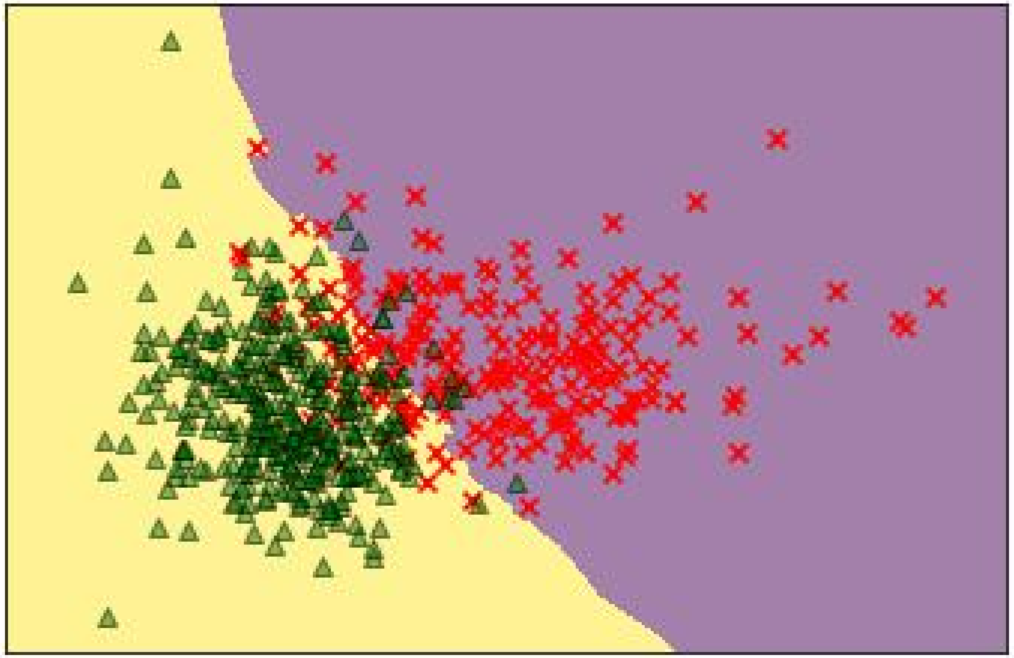
选择参数 K

$$经验数值 \approx \sqrt{样本量}$$

K = 20



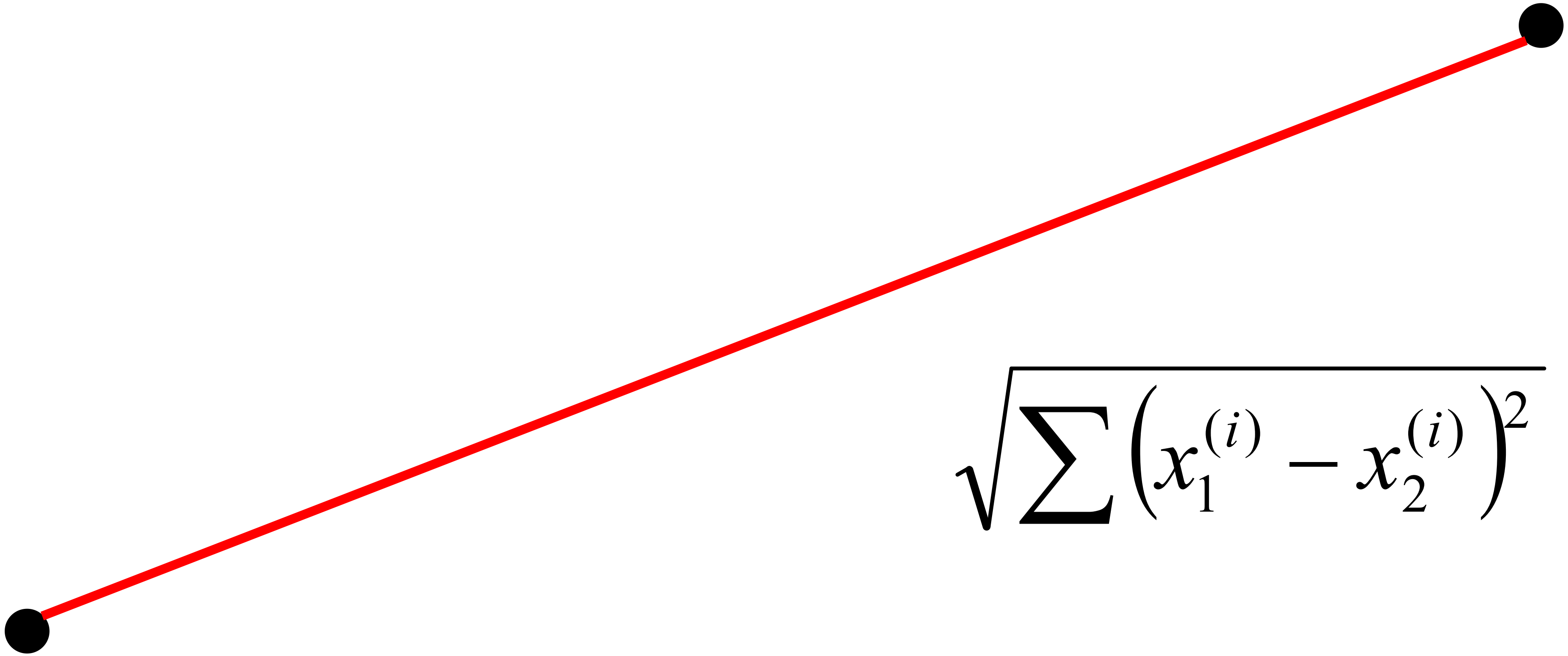
K = 30



选择参数 K

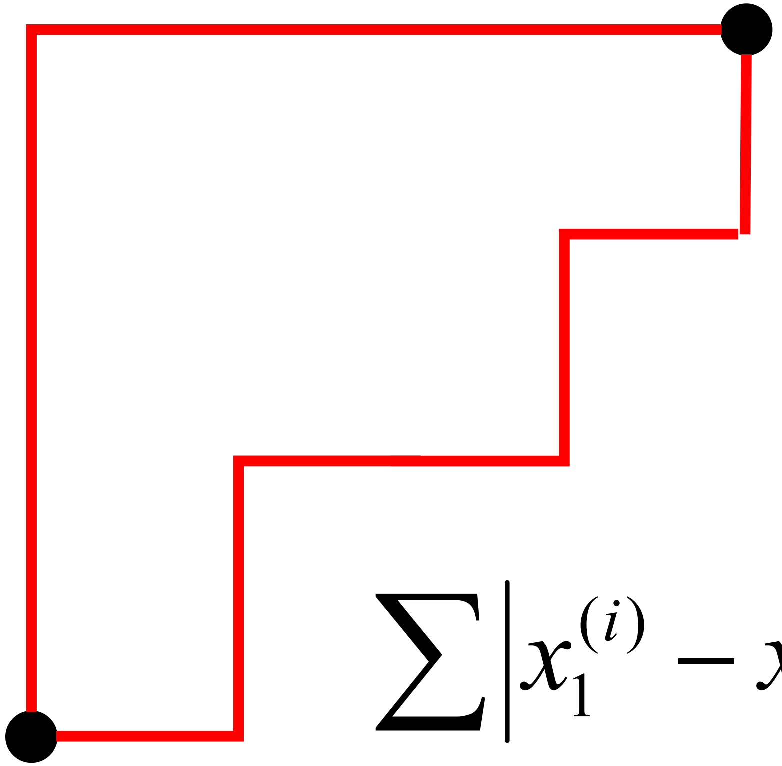
K	效果
太小	过拟合
太大	欠拟合
经验值	sqrt(N)

欧式距离 (Euclidean Distance)



$$\sqrt{\sum \left(x_1^{(i)} - x_2^{(i)} \right)^2}$$

曼哈顿距离 (Manhattan Distance)



$$\sum \left| x_1^{(i)} - x_2^{(i)} \right|$$

一般形式

Euclidean Distance

$$\sqrt{\sum\left(x_1^{(i)}-x_2^{(i)}\right)^2} = \left[\sum\left|x_1^{(i)}-x_2^{(i)}\right|^2\right]^{1/2}$$

Manhattan Distance

$$\sum\left|x_1^{(i)}-x_2^{(i)}\right| = \left[\sum\left|x_1^{(i)}-x_2^{(i)}\right|^1\right]^{1/1}$$

闵可夫斯基距离 (Minkowski Distance)

$$\left[\sum\left|x_1^{(i)}-x_2^{(i)}\right|^{\mathrm{p}}\right]^{1/\mathrm{p}}$$

参数 p 的影响

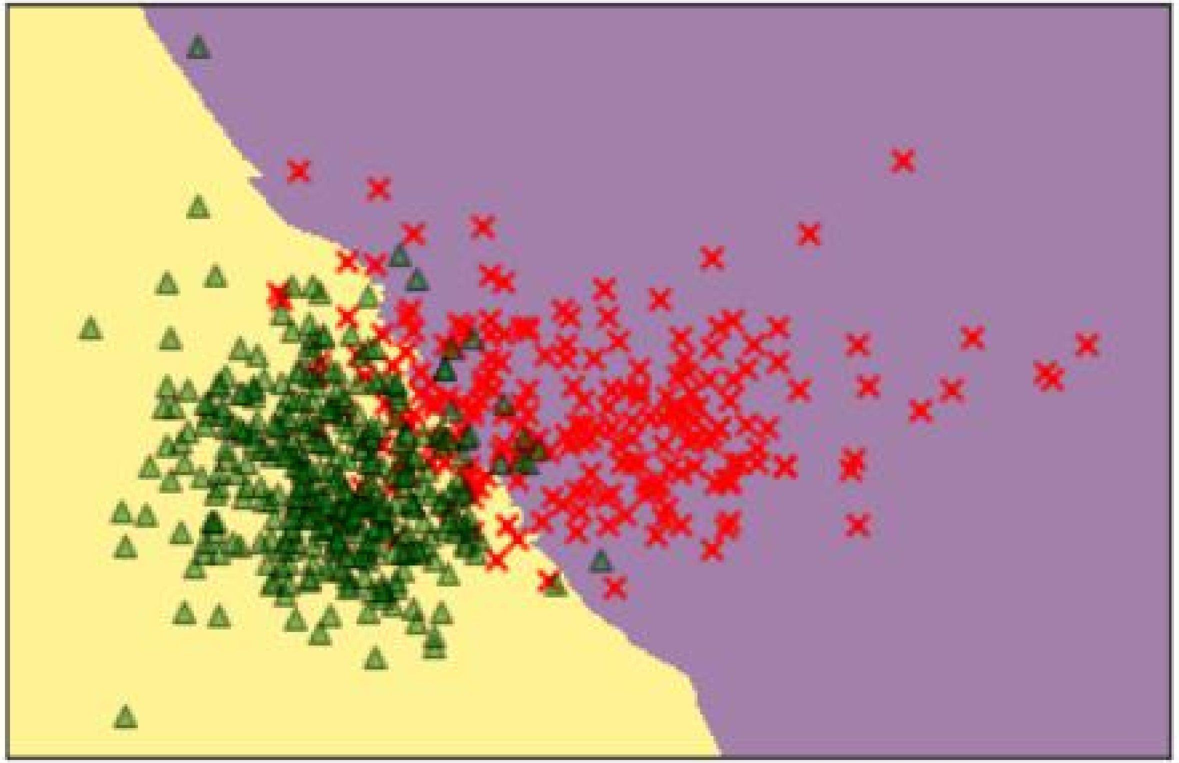
p large

more focus on large values

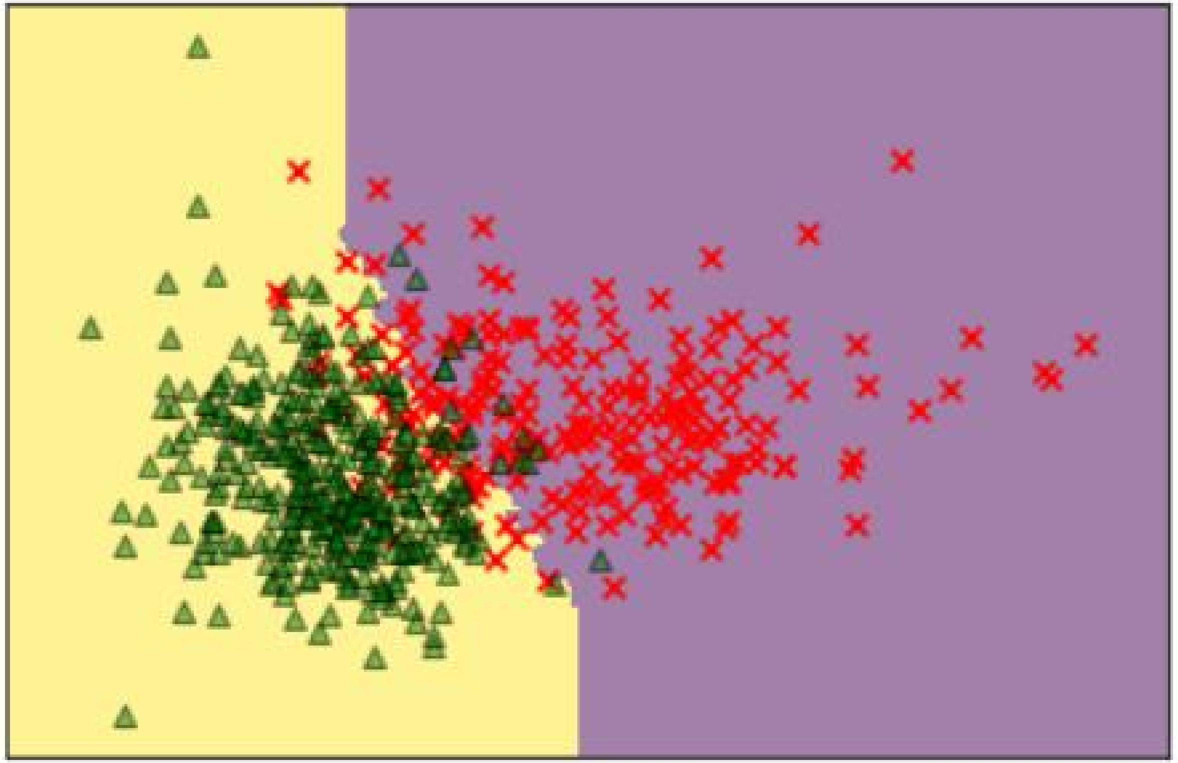
more sensitive to outliers

参数 p 的影响

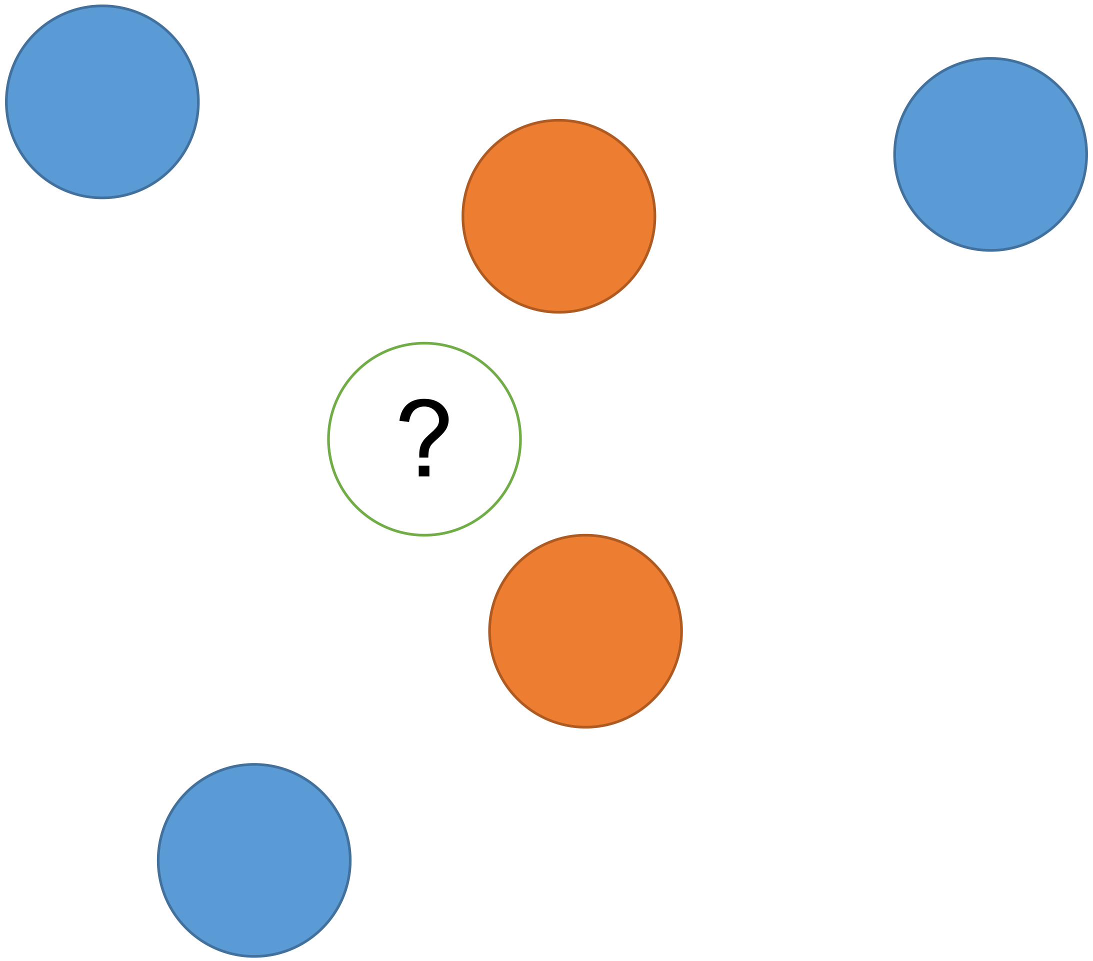
KNeighborsClassifier(n_neighbors=20, **p = 2**)



KNeighborsClassifier(n_neighbors=20, **p = 1**)

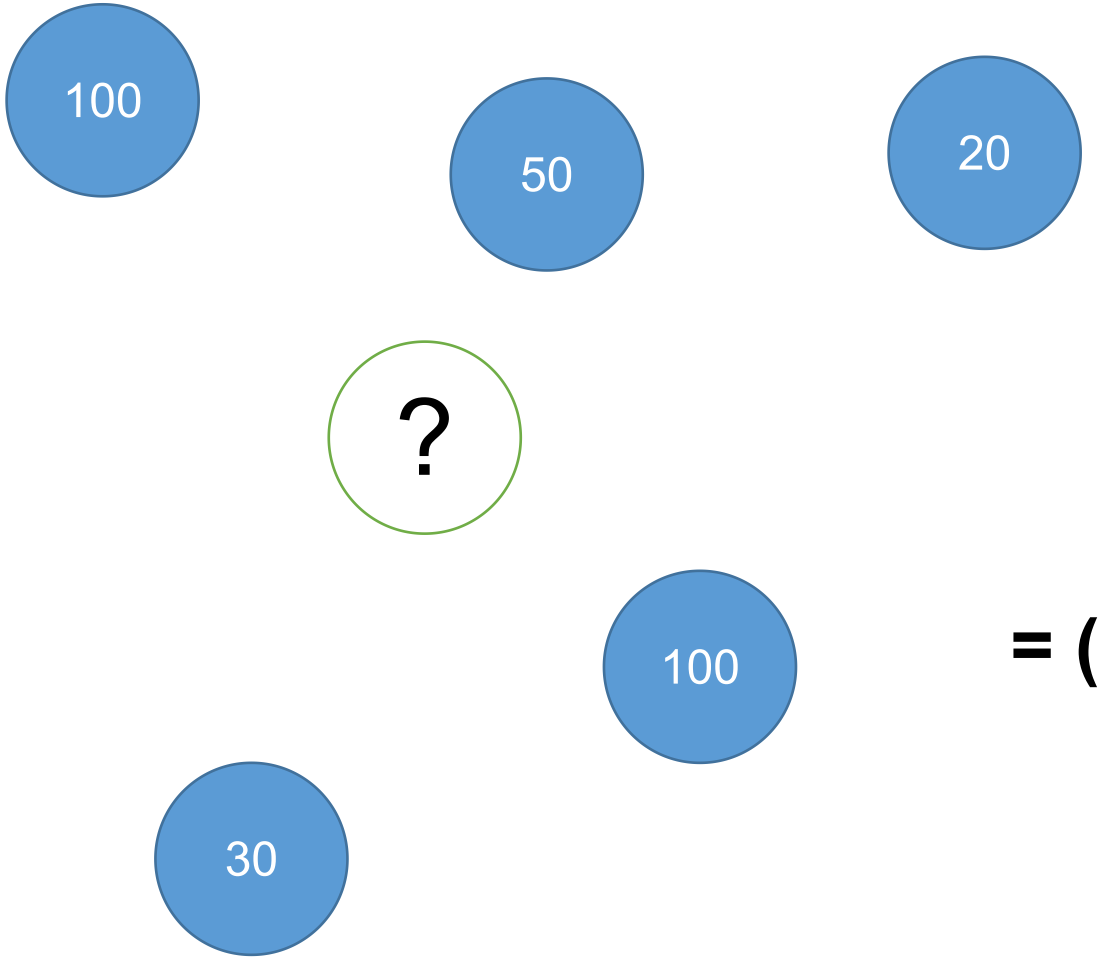


距离作为权重



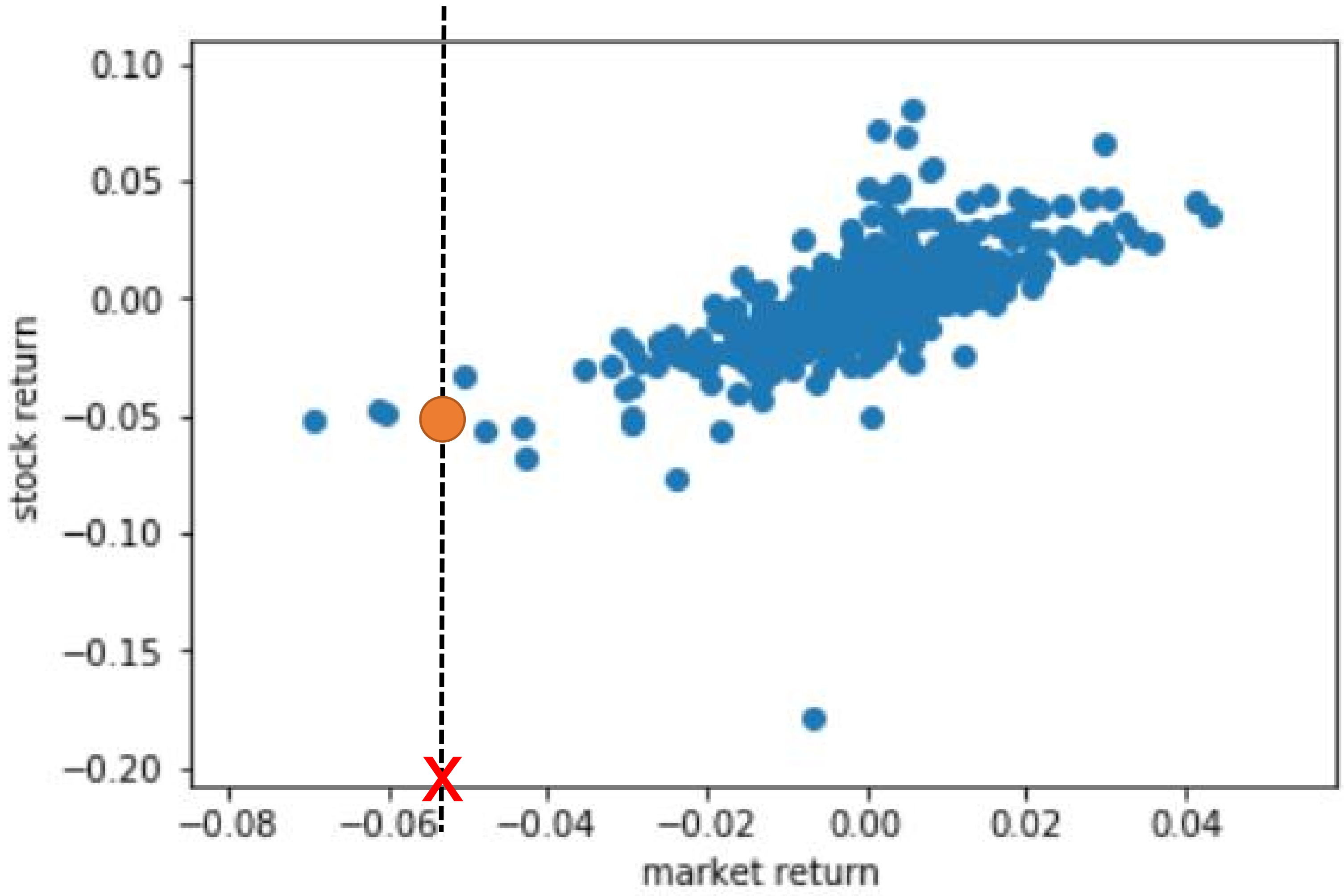
将 KNN 用于回归

将 KNN 用于回归

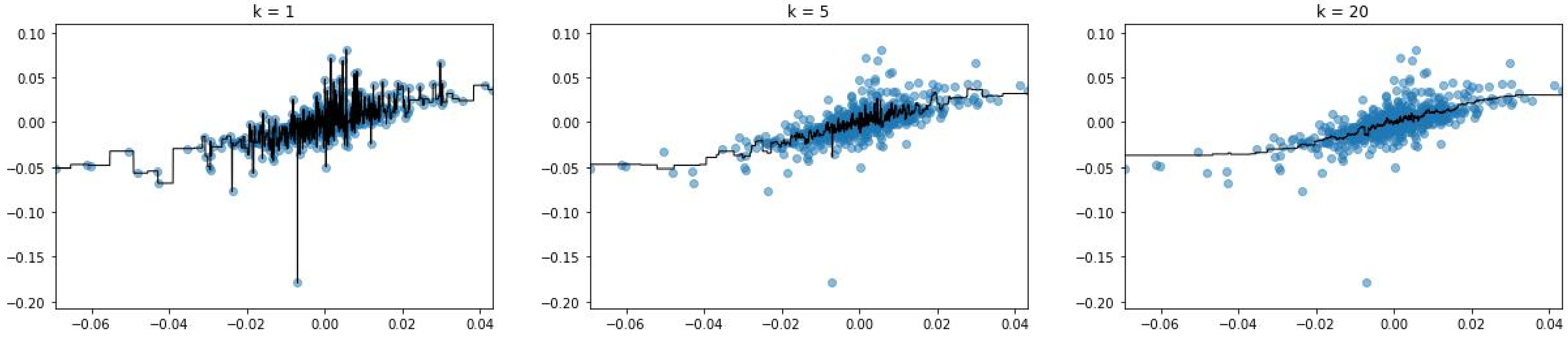


$$= (100 + 50 + 20 + 30 + 100) / 5$$

将 KNN 用于回归


0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
stock return
−0.08
−0.06
−0.04
−0.02
0.00
0.02
0.04
market return
X

将 KNN 用于回归



KNN 算法shi's总结

优点	缺点
容易理解 无需调整即可用于多分类问题 可用于分类和回归	计算复杂度高 对于大数据计算速度慢 事实上并没有进行任何训练 没有形成模型