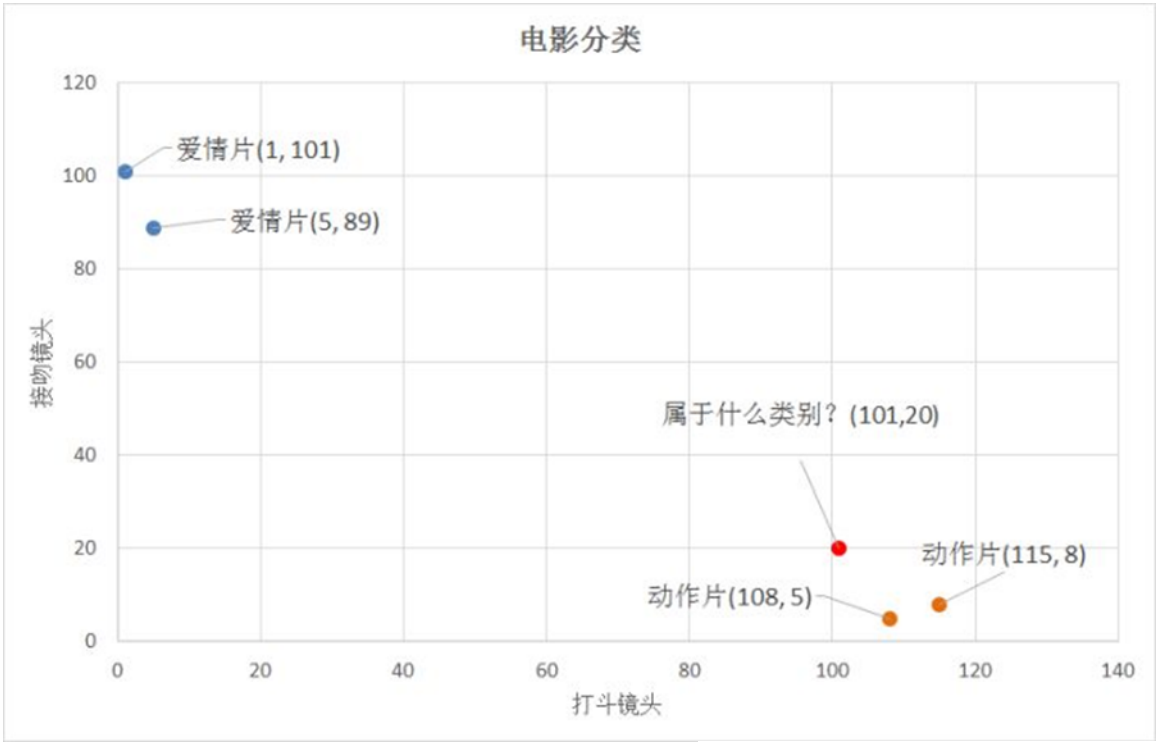
1、KNN（K-近邻算法）

K-近邻算法是最基础的非监督学习算法之一，在如下这张表中，预测一个电影是爱情片还是动作片，所使用的特征值是打斗镜头和接吻镜头，由此可想，它是一个二维平面上的预测分类算法。

电影名称	打斗镜头	接吻镜头	电影类型
电影1	1	101	爱情片
电影2	5	89	爱情片
电影3	108	5	动作片
电影4	115	8	动作片



现在将特征以及分类标签写在图中，利用公式 $|AB| = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ 即可计算出预测点与其余所有点的平方根号距离，

(101,20)->动作片(108,5)的距离约为 16.55

(101,20)->动作片(115,8)的距离约为 18.44

(101,20)->爱情片(5,89)的距离约为 118.22

(101,20)->爱情片(1,101)的距离约为 128.69

此时如果选择 K=3，则按照距离从小到大排序，我们选择前 3 个数据，动作片出现的频率为三分之二，因此我们预测该店对应的电影为动作片。

该算法的本质是非显示学习算法，因为预测点直接遍历所有的样本，计算距离，而不是通过学习特征与标签的关系。

关于二维以上 K-近邻算法，计算时可采用欧氏距离。

$$\mathrm{d}(\mathbf{p},\mathbf{q}) = \mathrm{d}(\mathbf{q},\mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}.$$

2、KNN 算法优缺点

优：

1、简单明了、可做分类

2、用于数值型和离散型数据

4、对异常值不敏感

缺：

1、时空复杂度高

2、样本不平衡会严重影响算法效能

3、无内在数据含义