特征选择：

筛选掉不需要的一些特征

数据预处理：

1. 数值型 归一化
2. 非数值型 --转化数值型
3. 缺失值的处理
   1. 删除
   2. 填充0
   3. 填充平均值

Pandas

一维数组Series

二维数组 DataFrame

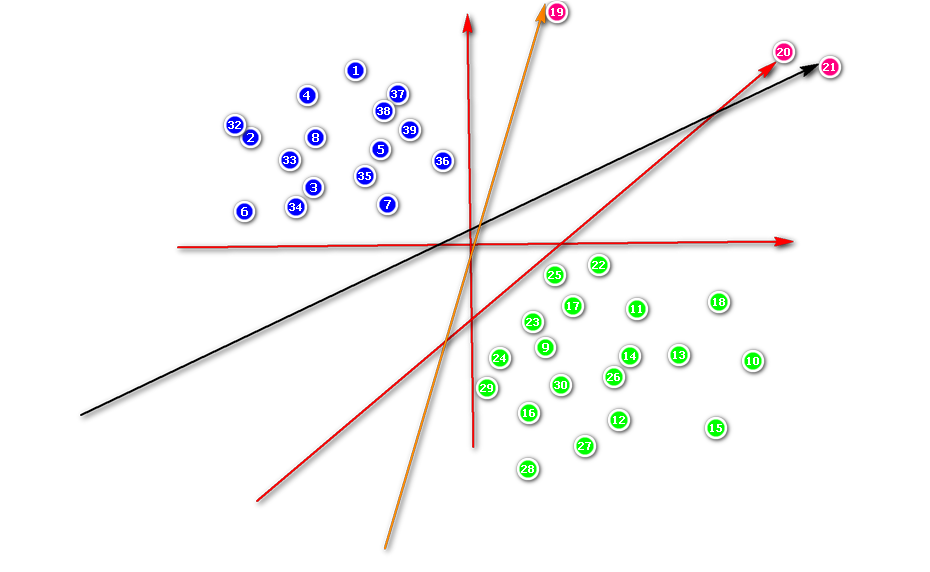
三维数组 Panel 面板

SVM算法

支持向量机

支持向量：决定分割线的位置的一些样本点

机：算法



点到直线的距离:

D=(（a\*x+by+c）/a^2+b^2)^0.5

逻辑回归：

If f(x)>0 Sign(f(x))=1

F(x)<=0 sign(f(x))=0

SVM：

If f(x)>1 Sign(f(x))=1

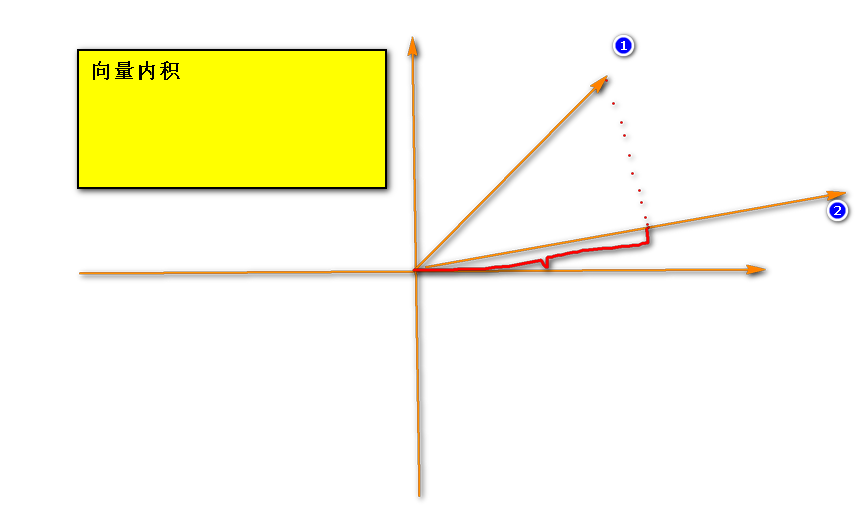
F(x)<=-1 sign(f(x))=0

F(x)=W^T\*X

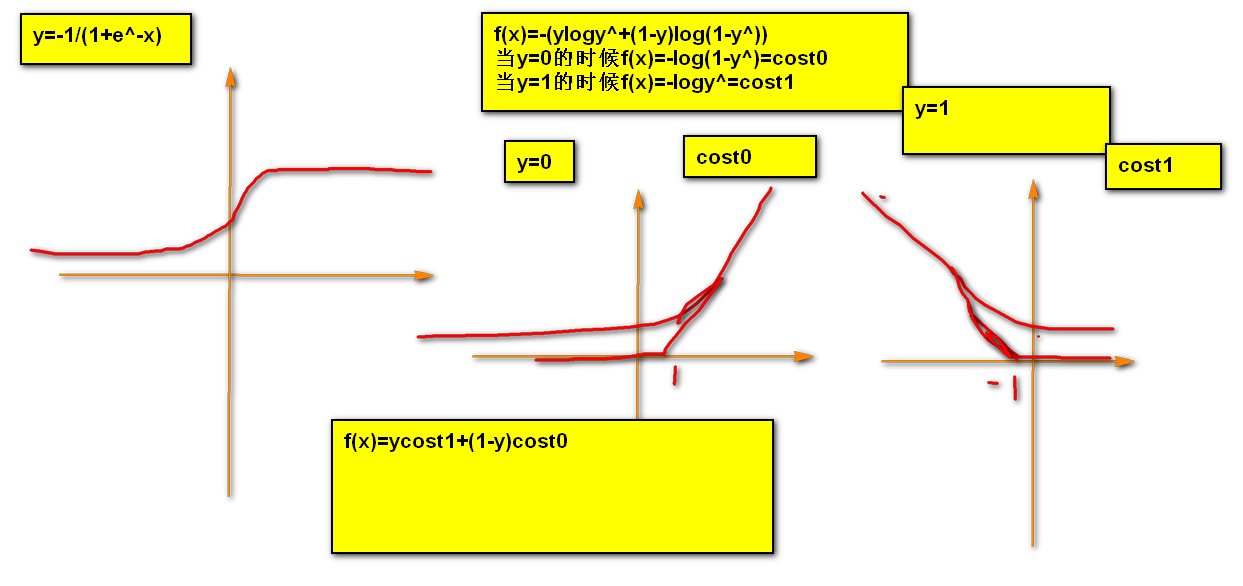
向量的范数：向量中所有元素的平方和再开方

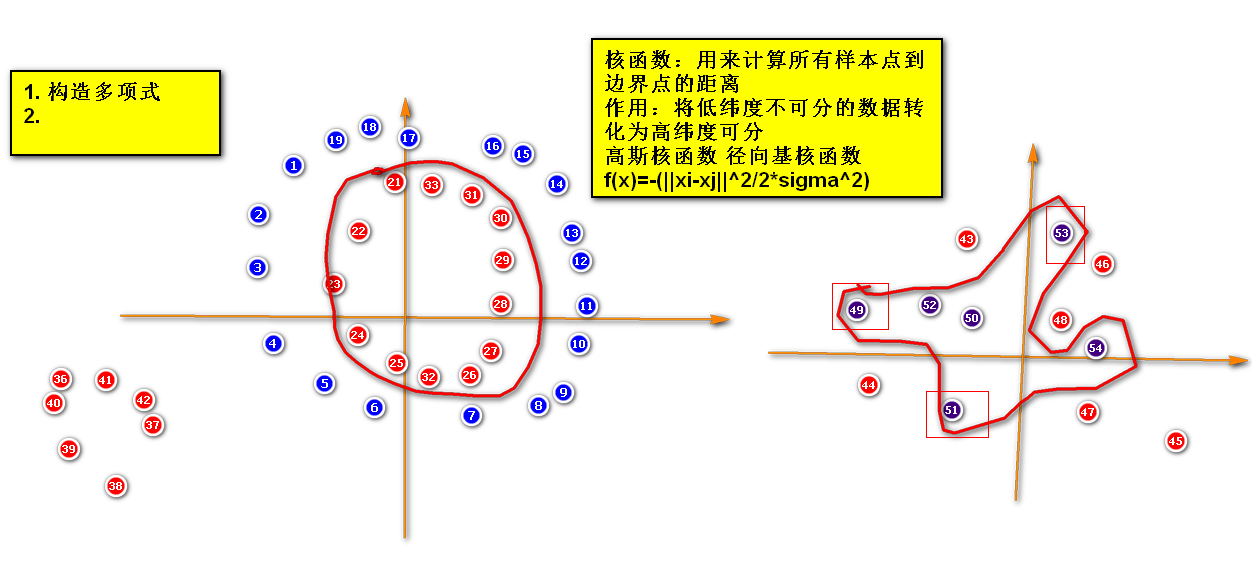
||W ||=(w1^2+w2^2+…+wn^2)^0.5

向量的内积：一个向量到另一个向量的投影的长度，记作p



支持向量到分割线法向量的内积





超平面：分割类别的面

SVM核心思想：利用核函数将低纬度不可分转化为高纬度可分，并且在迭代中找到最合适的超平面的位置，是算法的泛化能力更强，鲁棒性更好

原生的SVM只能做二分类,多分类的SVM一般称作SVC，做回归的叫做SVR

无监督学习：

没有明确的目标变量

聚类算法

K-means算法：

1. 指定k个质心
2. 计算每个点到这k个质心的距离 欧式距离 余弦距离 杰卡德距离
3. 把该样本点划分到距离最近的质心，形成一个簇
4. 计算每个簇的平均值，将距离平均值最近的点作为新的质心
5. 重复2-4步，直到质心的变化不大达到最大迭代次数为止
6. 最终输出每个质心作为新的类别

中文分词：jieba

Kmeans算法的问题：

人类事先不太清楚有多少类

DBSCAN基于密度的聚类算法

基于数据的分布密度进行聚类

MinPts：最少点数

Eps邻域：给定对象半径Eps内的邻域称为该对象的Eps邻域

核心对象：如果对象的Eps邻域至少包含最小数目MinPts的对象，则称该对象为核心对象

边界点：边界点不是核心点，但落在某个核心点的邻域内

噪音点：既不是核心点，也不是边界点的任何点

直接密度可达：给定一个对象集合D，如果p在q的Eps邻域内，而q是一个核心对象，则称对象p 从对象q出发时是直接密度可达的(directly density-reachable)

密度可达：对于点Q，如果存在p1,p2,...,pn且p1=p2,...,pn−1=pn,pn=Q，即p1到pn都是直接（密度）可达的，那么Q对于p1（密度）可达

密度相连：如果存在对象O∈D，使对象p和q都是从O关于Eps和MinPts密度可达的，那么对象p到q是关于Eps和MinPts密度相连的(density-connected)。