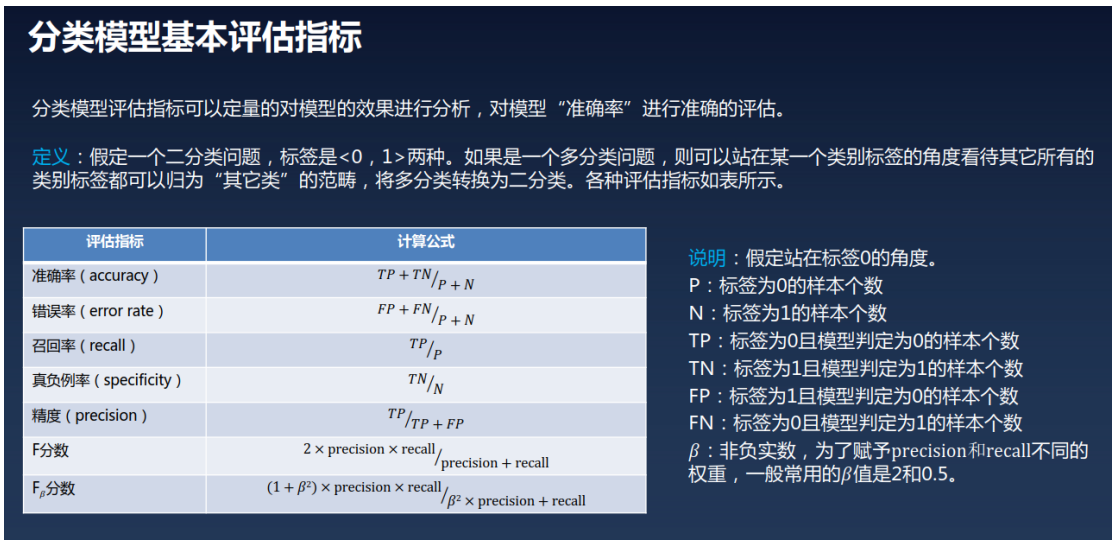
机器学习笔记

机器学习的模型评估
分类评估
回归评估

分类模型基本评估指标
分类模型评估指标可以定量的对模型的效果进行分析，对模型“准确率”进行准确的评估。
定义：假定一个二分类问题，标签是<0，1>两种。如果是一个多分类问题，则可以站在某一个类别标签的角度看待其它所有的
类别标签都可以归为“其它类”的范畴，将多分类转换为二分类。各种评估指标如表所示。

说明：假定站在标签 0 的角度。
P：标签为 0 的样本个数
N：标签为 1 的样本个数
TP：标签为 0 且模型判定为 0 的样本个数
TN：标签为 1 且模型判定为 1 的样本个数
FP：标签为 1 且模型判定为 0 的样本个数
FN：标签为 0 且模型判定为 1 的样本个数
β：非负实数，为了赋予 precision 和 recall 不同的
权重，一般常用的β值是 2 和 0.5。



评估指标 计算公式
准确率（accuracy） $TP+TN\ P+N$
错误率（error rate） $FP+FN\ P+N$

召回率（recall）	TP/P
真负例率（specificity）	TN/N
精度（precision）	$TP/TP+FP$
F 分数	2×precision×recall precision+recall
Fβ分数	(1+β2)×precision×recall β2 ×precision+recall

分类基本评估指标的使用

分类模型评估指标的使用跟具体的业务场景相关，而且因为可以站在不同类别标签的角度去计算指标，这样就出产生（7 x 类别
个数）个基本评估指标，需要在具体的业务场景中进行分析。
业务场景举例 1：银行判定贷款申请者有无风险。此业务场景下，银行既希望能够扩大贷款业务、又希望降低坏账风险，所以对
于“有风险”和“无风险”两种类别的判定评估都需要考虑，各项指标（错误率除外）都要尽可能的高。

业务场景举例 2：某制造产线判定某生产设备有无故障。此业务场景下，首先要考虑的是故障的判定要准，所以以故障类别计算
的召回率要接近 100%，在此前提下，specificity 要尽可能的高，越高越好。

评估指标的使用原则：
1、确立标签重要性。要确定哪一个类别是要非常关注的类别。
2、重要类别的召回率、精度需要制定一个高的标准。比如必须达到 xx%。
3、非重要类别的指标尽可能的高。

分类模型其它评估指标
1、 混淆矩阵：当面临一个多分类问题，且每个类别的权重几乎等同时，利用混淆矩阵进行模型评估。

单位为样本个数	预测类别			
实际类别		类别1	类别2	类别3
	类别1	500	10	5
	类别2	2	300	1
	类别3	3	4	200

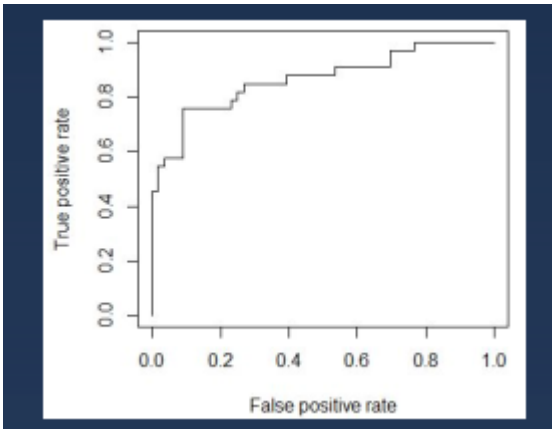
单位为样本个数 预测类别

注：矩阵对角线上的值
越大越好，其它位置的
值越小越好

2、ROC 曲线：当一些分类器（比如基于神经网络）给出的判定值并不是类别标签，而是一些数值，那么就需要给这些数值一个

阈值去产生类别，阈值不同，则类别判定结果不同，这样就产生了基于阈值的 recall 和（1-specificity），每一对这样的两个值

看做平面上的一个点，多个阈值产生的点相连就产生了 ROC 曲线。ROC 曲线越凸，表明模型效果越好。



3、AUC 值：AUC 值为 ROC 曲线所覆盖的区域面积，AUC 越大，分类器分类效果越好。

AUC = 1：是完美分类器，采用这个预测模型时，不管设定什么阈值都能得出完美预测。

绝大多数预测的场合，不存在完美分类器。

0.5 < AUC < 1：优于随机猜测。这个分类器（模型）妥善设定阈值的话，能有预测价值。

AUC = 0.5：跟随机猜测一样（例：丢铜板），模型没有预测价值。

AUC < 0.5：比随机猜测还差；但只要总是反预测而行，就优于随机猜测。

回归评估指标的计算

回归评估指标的计算

典型回归方法

线性回归

KNN 回归

分类回归树（基于平方误差）

……

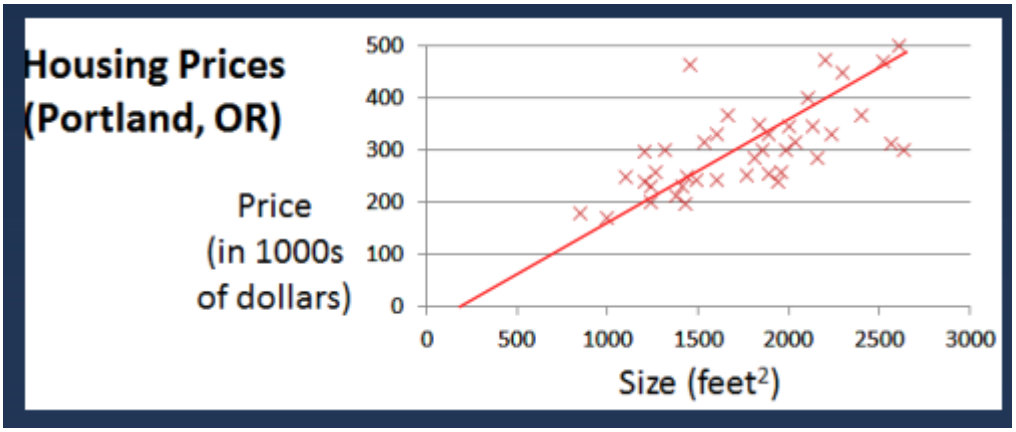
回归问题的评估指标

平均绝对误差（ mean absolute error，MAE ）

Size in feet² (x)	Price ($) in 1000's (y)
2104	460
1416	232
1534	315
852	178
...	...

均方根误差（root mean squared error，RMSE）

均方误差（Mean Squared Error，MSE）



平均绝对误差（ mean absolute error，MAE ）

$$\frac{1}{m}\sum_{i=1}^{m}\left|(y_i-\hat{y}_i)\right|$$

均方根误差（root mean squared error，RMSE）

$$\sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i-\hat{y}_i)^2}$$

均方误差（Mean Squared Error，MSE）

$$\frac{1}{m}\sum_{i=1}^{m}(y_i-\hat{y}_i)^2$$

回归评估指标的使用原则

回归评估指标计算的都是真实值和预测值之间的误差：

1、使用误差的方式从总体上来判断模型的好坏，原则是误差越小越好，那么这个“小”的程度需要提前规定。比如

“平均绝对误差不能超过真实值平均值的 10%”，这就是一个程度规定，程度规定要根据具体场景具体制定。

2、可以使用“接受度”的概念将依据误差的评估进行转换，比如设定一个接受度的值为 80%，即当预测值与真实值

之间的比值处于 0.8-1.2 之间则视为为“可接受”，然后设定一个可接受的样本数，比如“测试集中必须有 90%以上

的样本达到可接受状态”，这样就相当于用分类评估的思想去处理回归评估。

2、 可以根据实际场景的不同设定各种规则，以满足模型验收需求为前提。

客户分群场景定义

客户分群是指在营销、客户管理等场景下需要对客户进行分群，从而对不同的群体进行不同的营销策略制定、不同的管理方式等，一般来说进行客户分群时，客户本身的信息是不带标签的信息，需要使用聚类算法找到客户之间内在的联系，将相同的客户分在一起。

数据说明

字段说明

字段名	含义	类型	描述
Channel	渠道	Int	渠道
Region	区域	Int	区域
Fresh	生鲜类	Int	生鲜类
Milk	奶制品	Int	奶制品
Grocery	杂货	Int	杂货
Frozen	冷冻类	Int	冷冻类
Detergents_Paper	洗涤类	Int	洗涤类
Delicassen	熟食类	Int	熟食类
id	客户ID	Int	客户标识

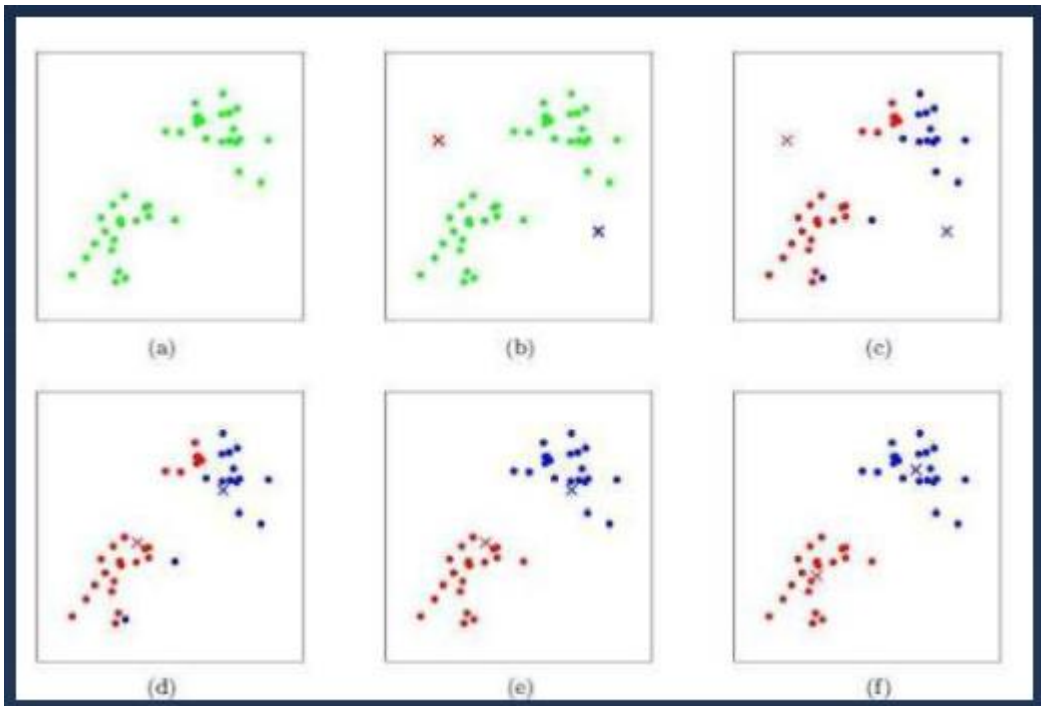
数据示例

Channel	Region	Fresh	Milk	Grocery	Frozen	Detergents_ Paper	Delicatessen	id
2	3	12669	9656	7561	214	2674	1338	1
2	3	7057	9810	9568	1762	3293	1776	2
2	3	6353	8808	7684	2405	3516	7844	3
1	3	13265	1196	4221	6404	507	1788	4
2	3	22615	5410	7198	3915	1777	5185	5
2	3	9413	8259	5126	666	1795	1451	6
2	3	12126	3199	6975	480	3140	545	7

算法和建模

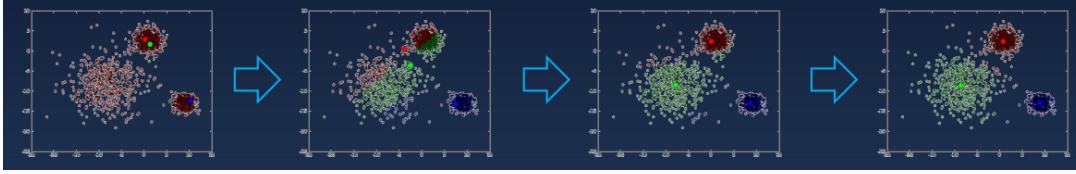
划分聚类算法：k-means 原理
输入：k（簇的数目）、要进行分类的数据集 D
输出：k 个簇的集合
过程：
1）从 D 中以某种规则选择 k 个样本或者 k 个在值域范围内的点作为初始簇的中心
2）计算簇中心之外的每个样本和每个簇中心的距离，将样本归属于最近的的簇。
3）计算簇内均值，将均值作为簇的中心。
4）训练 2 和 3，直到簇中心的变化小于一定的阈值或者即便簇中心变动，但是簇内样本不变动。

k-means 的优缺点：
优点：
1、能够比较快速的收敛。
2、在大数据集上是相对可伸缩和有效的。
3、当无法计算均值时，可以通过定义一些簇中心（比如众数簇心）
来改写算法。
缺点：
1、严重依赖于 k 值的确定。
2、不适合发现非球形簇或者大小差别非常大的簇。
3、对噪声和离群点非常的敏感。



划分聚类算法：k-means 图示

成功案例

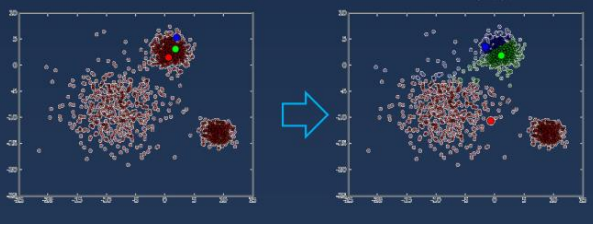


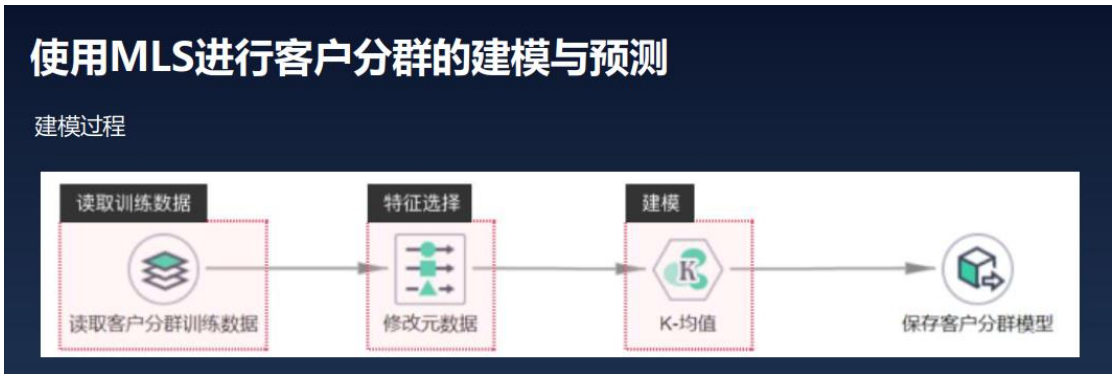
最小化目标函数

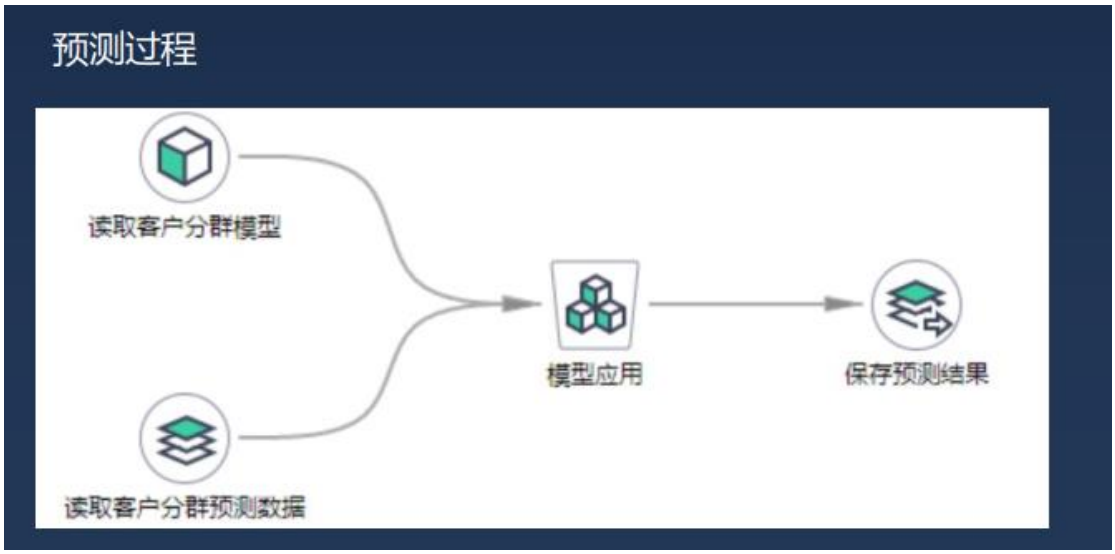
$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|x_n - \mu_k\|^2$$

$$\mu_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}}$$

失败案例







k-means 初始化方法

K 值设定:
提供一些基于经验的 k 值，或者一个 k 值的范围，通过比较由不同的 k 得到的聚类结果，来确定最佳的 k 值。
初始簇心选择：两种方式
1、随机初始簇心：k 个簇的簇心的初始化都是随机的，随机选择 k 个样本作为簇心。
2、有限最大距离簇心:
1）随机选择 1 个样本作为第 1 个簇心
2）随机选择 m 个样本，计算这些样本和当前所有簇心的距离 dist
3）在 2 的计算结果中选择 dist 最大的样本作为下 1 个簇心。
4）循环 2 和 3，直到选出 k 个簇心。
终止规则：两种方式
1、完全终止，即在当前 k 值设定下，簇完全不发生更新时停止
2、设定一个次数，当簇的更新次数达到此值时即停止

K-means 优化

1、检测离群点，删除离群点，优化簇的聚集程度。

2、尝试不同的 k-means 初始化方法，以寻找更好的簇。