金融科技学

李彦

liyan_zjgsu.163.com

分类

- 决策树：

算法	ID3	C4.5	CART
结点分裂度量	信息增益	信息增益率	Gini指数

- 随机森林： Bagging+DecisionTree
  - 森林：集成思想
  - 随机：**样本随机和特征随机**
  - 采用随机采样和随机属性的办法，随机森林足以保证不同的基分类器之间具有足够的不相关性，因而**不再需要对个体决策树进行剪枝**

聚类

- K-means：
  - 核心参数k：轮廓系数/肘部法则
  - 距离/相似性
- 层次聚类：Agnes
- Dbscan：
  - 核心参数{eps, MinPts}：k-距离
  - 密度可达关系

机器学习常见问题

- 数据质量问题
- 机器学习方法的选择
- 认识数据与数据预处理

数据质量问题

- 数据质量要求数据是完整的和真实的，并且具有一致性和可靠性
- **数据预处理占用整个机器学习项目60%的工作量**
- 问题
  - 数据量过少
  - 数据量过多
  - 维度灾难
  - 数据不完整
  - 异常数据
  - 重复数据
  - 数据不一致

数据量过少

- 数据样本需要覆盖与分析目标相关的维度
- 数据量增多，其中的规律会越发明显，也更易发现与分析目标相关的因素
  - 神经网络
  - 深度学习
- 一般来说，样本数量是特征数量的10~20倍为佳

数据量过多

- 数据量过多时，对全部数据集进行分析要耗费更多的计算资源，要求硬件配置较高，可应用数据采样技术随机提取样本子集。
- 对海量的同质化数据，可通过聚集技术按照时间、空间等属性进行汇总，减少数据数量。
- 数据集不平衡问题可能导致出现较大的结果误差，因此要对数据集应用采样技术或对异常数据进行复制，提高其占比。

维度灾难

- 当数据中的**特征过多**时，会出现维度灾难问题。
- 特别是在矩阵数据中，当冗余变量占比较高时，可用数据会变成稀疏矩阵，在分类算法处理时就无法可靠地进行类别划分，在聚类算法中则容易使聚类质量下降。
- 可采用线性代数的相关方法将数据从高维空间影射到低维空间:
  - 主成分分析（PCA）
  - 奇异值分解（SVD）

数据不完整

- 数据的种类多少直接影响数据挖掘方法的选择，可以通过编写程序抓取外部数据作为补充。
  - **数据缺失**也是数据不完整的一种表现，包括了空值、无效值等。
  - 需要针对不同原因对缺失值进行数据预处理，有多种方法可以操作
    - 采用众数、中位数、均值、最短距离等方法进行人为填充
    - 通过回归或贝叶斯定理等预测缺失值
    - 删除含有缺失值的数据

机器学习方法的选择

- 理解目标要求是机器学习方法选择的关键，首先对问题进行分析：如果数据集中有标签则进行**有监督学习**，反之则进行**无监督学习**

- 熟悉各类机器学习方法的特性是分析方法选择的基础，不仅需要了解如何**使用**各类分析算法，还要了解其实现的**原理**
  - 在选择模型前，要对数据进行描述性统计
  - 在几个可能模型中分析选择出较优的模型
  - 选择模型后，比较不同模型的泛化能力，反复调整参数使模型结果趋于稳定

Come back to the task of PartI

- 请思考：
  - 如何处理原始数据中的缺失值？
    - dropping first or filling first?
  - 如何确定特征？
    - 你能自己构造额外的特征吗？
  - 面对这样一个问题，如何选择合适的机器学习方法？

认识和处理数据

- 数据对象与属性类型
- 数据的描述性统计
- 数据预处理
  - 非数值型数据的处理：编码
  - 数据清洗：缺失值和异常值
  - 数据变换：规范化，离散化
- 特征处理
  - 特征选择
  - 特征构造
  - 特征提取

数据对象与属性类型

- 数据集由数据对象构成，一个数据对象代表一个实体
  - 例：银行贷款数据库：客户
  - 又称为样本、事例、实例、数据点、对象 等
- 数据对象由属性描述
  - 属性(Attribute，也称作维度、特征、变量)：一个数据字段表示一个数据对象的某个特征
  - 例：年收入、婚姻状况、是否拥有房产

序号	拥有房产（是/否）	婚姻状况（单身、已婚、离婚）	年收入（单位：万元）	无法偿还债务（是/否）
1	是	单身	12.5	否
2	否	已婚	10	否
3	否	单身	7	否
4	是	已婚	12	否
5	否	离婚	9.5	是
…	…	…	…	…

属性的类型

- 类型:
  - **标称属性**（Nominal）：与名称有关
    - Hair_color = {black, blond, brown, grey, red, white}
  - **二元属性**（Binary）：是一种特殊的标称属性或布尔属性
    - 只有2个状态的名词性属性 (0 and 1):
    - 对称（Symmetric binary）：同等重要。例：性别
    - 非对称（Asymmetric binary）：非同等重要。例：医疗检查中的阴性和阳性
惯例：**将更重要的一方赋值为1**
  - **序数属性**（Ordinal）：值的顺序有意义，相邻值之差未知
    - Size = {small, medium, large}
  - 数字属性（Numeric）：数值的

离散 vs. 连续属性

- 离散属性（Discrete Attribute）：一个有限的或可数无限集的值
  - 例：邮政编码 - 有时表示为整数变量
  - 注：二元属性是离散属性的一个特殊情况
- 连续属性（Continuous Attribute）：属性值为实数
  - 例：temperature, height, or weight
  - 实际上，实值只能使用有限位数进行测量和代表
  - 连续属性通常表示为浮点型变量

数据的基本统计描述(Summary Measures)

- 描述性统计度量
  - 集中趋势：均值、中位数、众数……
  - 离散趋势：方差、标准差、极差、变异系数……
  - 分布特征：偏度、峰度……
- 可视化(visualization: matplotlib)
  - 直方图(hist)：概率分布
  - 散点图(scatter)：相关性
  - 折线图、箱型图……

数据预处理

- 原因
  - 数据在搜集时由于各种原因可能存在缺失、错误、不一致等问题
  - 用于描述对象的数据不能很好地反映潜在的模式
  - 描述对象的属性的数量有很多，有些属性是无用的或冗余的
- 任务
  - 数据编码（encoding）
  - 数据清洗（data cleaning）
  - 数据规范化（normalization）
  - 数据离散化（discretization）

数据编码：encoding

- 非结构化数据（文本、图像、语言、音乐）
- 数据编码是为了让计算机能够处理数据
  - 标称属性：性别、婚姻状况、学历……
  - 文本信息：one-hot

- 二分类：{class1，class2} → {0，1}
- 多分类：{class1, …, classN} → {0, …, N-1}
- 注：不是所有的无序变量都需要做数值化处理，*决策树、随机森林等树模型可能不需要处理*，视情况而定。

独热编码

- One-Hot编码，又称一位有效编码，主要是采用N位状态寄存器来对N个状态进行编码，每个状态都对应独立的寄存器位，并且在任意时候只有一位有效。
- 优点：
  - 简单，且保证无共线性
  - 对离散型特征使用one-hot编码会让特征之间的距离计算更加合理
  - 对离散型特征进行one-hot编码可以加快计算速度
- 缺点：
  - 稀疏矩阵
  - 解决方法：降维

数据清洗

- 缺失数据和噪声数据的处理，数据不一致的识别和处理
- 处理缺失数据:
  - 删除
  - 填充：如果数据集含有**分类属性**，一种简单的填补缺失值的方法为:
    - 将属于同一类的对象的该属性值的均值赋予此缺失值
    - 对于离散属性或定性属性，用众数代替均值
  - 预测
  - 插值
  - ……

简单方法的缺失值处理

序号	拥有房产（是/否）	婚姻状况（单身、已婚、离婚）	年收入（单位：万元）	无法偿还债务（是/否）
1	是	单身	12.5	否
2	否	已婚		否
3		单身	7	否
4	是	已婚	12	否
5	否	离婚	9.5	是
…	…	…	…	…

- 样本$2_{\text{年收入}}$=(12.5+7+12)/3
- 样本$3_{\text{是否拥有房产}}$=mode({是，是，否}) = 是

异常值处理

- 异常值分析:
  - 检测数据是否有输入错误或者含有不合常理的数据
- 异常值检查:
  - 简单统计量分析
  - 3σ准则
  - 箱型图分析
- 异常值处理:
  - 识别出噪音数据，将其**删除**。例：之前在介绍聚类算法DBSCAN时提到过，最终不属于任一个簇的孤立点(outlier) 可视作噪音。
  - 利用其它非噪音数据降低噪音的影响，起到平滑(smoothing)的作用

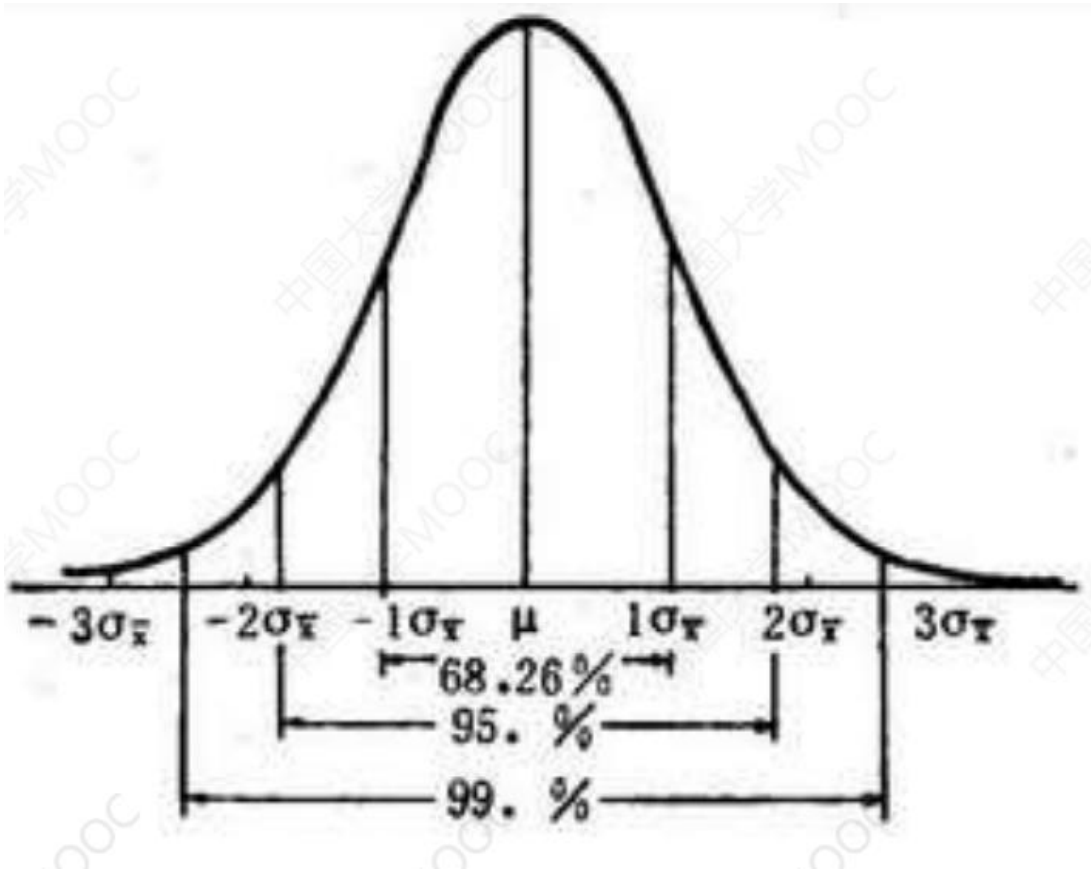
异常值分析

- 最常用的是**查看极值**并分析是否合理
  - 年龄=200
  - 价格出现负数
  - ……

异常值分析

- 1倍标准差之内：68.3%
- 2倍标准差之内：95.5%
- 3倍标准差之内：99.7%

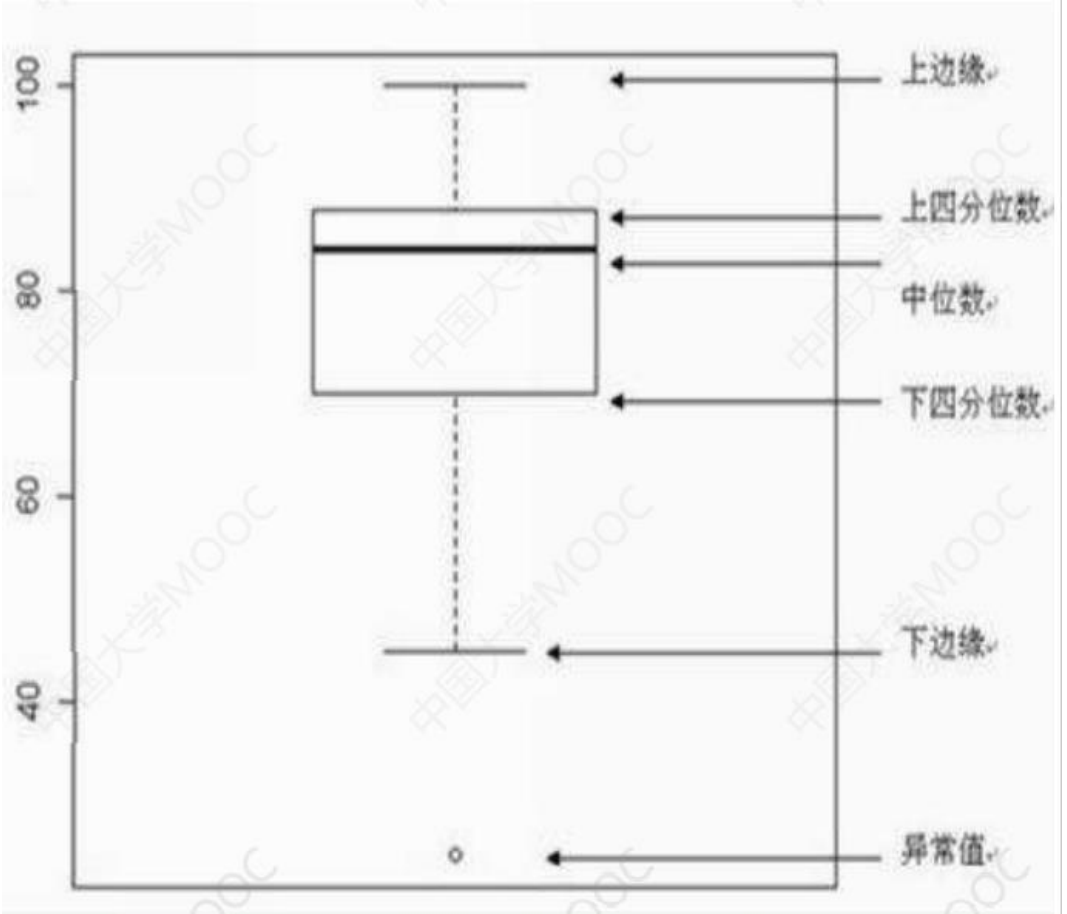
- 不符合标准分布：定义远离均值的多少倍标准差作异常值
- **3σ准则**：正态分布3倍标准差之外为异常



异常值分析

- 上四分位数：全部数据中有1/4的数据比它大，记作QU
- 下四分位数：全部数据中有1/4的数据比它小，记作QL
- $[QL, QU]$之间包含了一半的数据，记$IQR= QU - QL$
- $[QL - 1.5IQR, QU + 1.5IQR]$即为箱型图上下界，界外的值视作异常值

- **箱型图分析**：箱型图外部为异常



一致性分析

- 不一致性：数据的矛盾性和不相容性

姓名	上班方式	…	是否有车
xxx	xxx	…	xx
xxx	开车	…	否
xxx	xxx	…	xx

姓名	满意度	…	频率
xxx	xxx	…	xx
xxx	10	…	很少去
xxx	xxx	…	xx

- ？ ？ ？

数据变换

- 数据变换主要是对数据进行规范化处理，将数据转换成适当的形式，以适用于挖掘任务及算法的需要
  - 思考：为什么作变换？——假设数据分布不符合方法的要求
  - 简单函数变换是对原始数据进行某些数学函数变换，常用的变换包含平方、开方、 取对数、差分运算等。
    - 时间序列分析：差分
    - 取值范围较宽的分布：对数变换
  - 归一化/标准化
  - 连续数据离散化

数据规范化

- 数据规范化通过将属性的取值范围进行统一，避免不同的属性在数据分析的过程中具有不平等的地位。
- 目的：消除指标间量纲和取值范围差异
- 常用线性变换方法
  - **最小-最大规范化**（min-max normalization）
    通过对原始数据进行线性变换，使数据均落在[**0**, **1**]区间内
  - **零-均值规范化**（z-score）
    将原始数据变换到均值为0，标准差为1的分布中
- 注：基于树的方法是基于比较的，不需要做规范化处理

连续数据离散化

- 目标：将连续数据变换为分类属性
- 任务：确定区间数和映射方式
- 非监督离散化不需要使用分类属性值，相对简单。
  有**等宽离散化、等频率散化**、聚类等方法：
  - 等宽散化将属性值划分为宽度一致的若干个区间
  - 等频散化将属性值划分为若干个区间，每个区间内的数量相等

- 例：假设14个客户的属性“年收入”的取值按顺序为：
  [20，40，50，58，65，80，80，82，86，90，96，105，120，200]
  - 利用等距离离散化，区间的个数为4，则区间间距为(200-20)/4=45，则4个箱的区间分别为[20，65)，[65，110)，[110，155)，[155，200]
  - 利用等频率离散化，每箱3个值，则4个箱分别为[20，40，50]，[58，65，80，80]，[82，86，90]，[96，105，120，200]

数据离散化

- 监督离散化：通过选取能够**极大化区间纯度**的临界值来进行划分
- 例：
  - 首先将数据集D按照属性A的取值进行排序
  - 设v是A的一个取值，将数据集D以条件A<=v和A>v分割为D1和D2
  - 计算分割前后的信息增益（以决策树为例）
  - 以信息增益最大的v作为分割的阈值

特征工程

- 定义：简而言之，从数据到变量
  - 特征工程是利用数据所在领域的相关知识来构建特征，使得机器学习算法发挥其最佳的过程。它是机器学习中的一个基本应用，实现难度大且代价高。
- 地位：数据和特征是上限，算法和训练是逼近这个上限
  - “挖掘特征是困难、费时且需要专业知识的事，应用机器学习其实基本上是在做特征工程。”
- 实质：连接原始数据与模型

为什么之前没有学特征工程

- 目的角度：解释性优先于预测性
  - 变量本身：大部分时间，变量原值有很好的解释性
  - 构造模型：我们只关心某个变量（核心解释变量、控制变量）
  - 综合以上：降维、抽象化基本上不存在

- 实践角度

特征选择、特征提取与特征构造

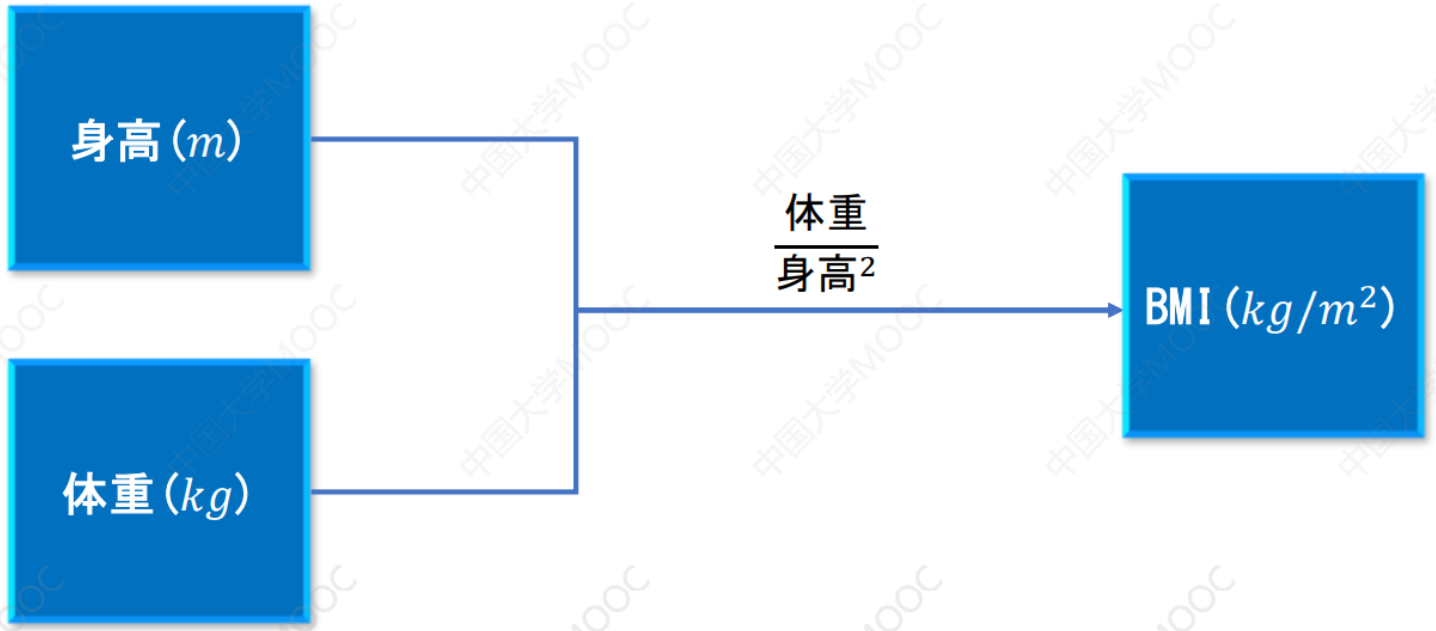
- **特征选择(Feature Selection)：**是指从属性集合中选择那些重要的、与分析任务相关的子集的过程

- **特征提取(Feature Extraction)：**通过对属性进行重新组合，自动地构建新的特征，将原始特征转换为一组具有明显物理意义或者统计意义或核的特征

- **特征构建(Feature Construction)：**从原始数据中人工构建新的特征

特征构建

- 需要很强的洞察力和分析能力，以及花大量的时间去研究真实的数据样本，思考问题的潜在形式和数据结构



特征选择

- 当数据预处理完成后，我们需要选择有意义的特征输入机器学习的算法和模型进行训练。通常来说，从两个方面考虑来选择特征:
- **特征是否发散**：如果一个特征不发散，例如方差接近于0，也就是说样本在这个特征上基本上没有差异，这个特征对于样本的区分并没有什么用。
- **特征与目标的相关性**：这点比较显见，与目标相关性高的特征，应当优选选择。除移除低方差法外，本文介绍的其他方法均从相关性考虑。
- 根据特征选择的形式又可以将特征选择方法分为3种：Filter、Wrapper、Embedded

特征选择

- Filter（过滤方法）：按照发散性或者相关性对各个特征进行评分，设定阈值或者待选择阈值的个数，选择特征。
  - 方差选择法
  - 相关系数法
  - 互信息法
  - 卡方检验
  - Relief算法
  - ……
- 优点：算法的通用性强，省去了分类器的训练步骤，算法复杂性低，因而适用于大规模数据集，可以快速去除大量不相关的特征，作为特征的预筛选器非常合适
- 缺点：倾向于选择冗余的特征，因为不考虑特征之间的相关性。由于算法的评价标准独立于特定的学习算法，所选的特征子集在分类准确率方面通常低于wrapper方法。

特征选择

- Wrapper（封装方法）：用选取的特征子集对样本集进行分类，分类的精度作为衡量特征子集好坏的标准，经过比较选出最好的特征子集
  - 递归特征消除法
  - 特征干扰法
- 优点：考虑了特征与特征之间的关联性，wrapper方法找到的特征子集分类性能通常更好
- 缺点：wrapper方法选出的特征通用性不强，当改变学习算法时，需要针对该学习算法重新进行特征选择，由于每次对子集的评价都要进行分类器的训练和测试，所以算法计算复杂度很高，尤其对于大规模数据集来说，算法的执行时间越长。

特征选择

- Embedded（嵌入方法）：特征选择算法本身作为组成部分嵌入到学习算法里（例：决策树生成的过程也就是特征选择的过程）
  - 基于惩罚项的特征选择法
  - 基于树模型的特征选择法

- 决策树算法在树增长过程的每个递归步都必须选择一个特征，将样本划分成较小的子集，选择特征的依据通常是划分后子节点的纯度，划分后子节点越纯，则说明划分效果越好，可见决策树生成的过程也就是特征选择

sklearn

- sklean.feature_selection

案例：违约预测

客户违约预测模型搭建

模型预测及评估

(2) 预测不违约&违约概率

其实分类决策树模型本质预测的并不是准确的0或1的分类，而是预测其属于某一分类的概率，可以通过如下代码查看预测属于各个分类的概率：

```
y_pred_proba = clf.predict_proba(X_test)
```

客户违约预测模型搭建

模型预测及评估

(2) 预测不违约&违约概率

此时获得的y_pred_proba就是预测的属于各个分类的概率，它是一个二维数组，下表展示的便是最后五组数据的不违约&违约概率。

- 第一列数据是预测为第一类结果0，也即不违约的概率
- 第二列数据则是预测为第二类结果1，也即违约的概率

这两个概率的和为1：

最后五组数据的不违约&违约概率	
0.86	0.14
0.56	0.44
0.56	0.44
0.04	0.96
0.25	0.75

客户违约预测模型搭建

模型预测及评估

(2) 预测不违约&违约概率

二分类问题默认是以0.5作为阈值来预测属于哪一类，因为如果某一类的概率大于0.5，则该类的概率必然大于另一类。实际应用也可以根据需要调节阈值，比如设定只要违约概率大于0.3，就认为该用户会违约。

想单纯的查看违约概率，即查看y_pred_proba的第二列，可以采用如下代码：

```
y_pred_proba[:,1]
```

客户违约预测模型搭建

模型预测及评估

(2) 预测不违约&违约概率

之前已经利用准确度来衡量了模型的预测效果，不过在商业实战中一般不会以准确度作为模型的评估标准，因为准确度很多时候并不可靠。

举个例子：倘若100个客户里有10个人违约，而如果模型预测所有客户都不会违约，虽然这个模型没有过滤掉一个违约客户，但是模型的预测准确度仍然能达到90%，显然这个较高的准确度并不能反映模型的优劣。

客户违约预测模型搭建

模型预测及评估

(3)模型预测效果评估

在商业实战中，我们更关心下面两个指标：

真正率（命中率）	True Positive Rate（TPR）	TPR=TP/ (TP+ FN)
假正率（假警报率）	False Positive Rate（FPR）	FPR= FP / (FP + TN)

客户违约预测模型搭建

模型预测及评估

(3) 模型预测效果评估

其中TP、FP、TN、FN的含义如下表所示，这个表也叫作混淆矩阵：

	1（预测违约）	0（预测不违约）	合计
1 （实际违约）	True Positive (TP) 正确肯定	False Negative (FN) 漏报	TP + FN
0 （实际不违约）	False Positive (FP) 虚报	True Negative (TN) 正确否定	FP + TN

客户违约预测模型搭建

模型预测及评估

(3) 模型预测效果评估

一个优秀的客户违约预测模型，我们希望真正率（TPR）尽可能的高，即能尽可能地揪出坏人，同时也希望假正率（FPR）能尽可能的低，即不要误伤好人。

然而这两者往往成正相关性，因为一旦当调高阈值，比如认为违约率超过90%的才认定为违约，那么会导致假正率很低，但是真正率也很低。

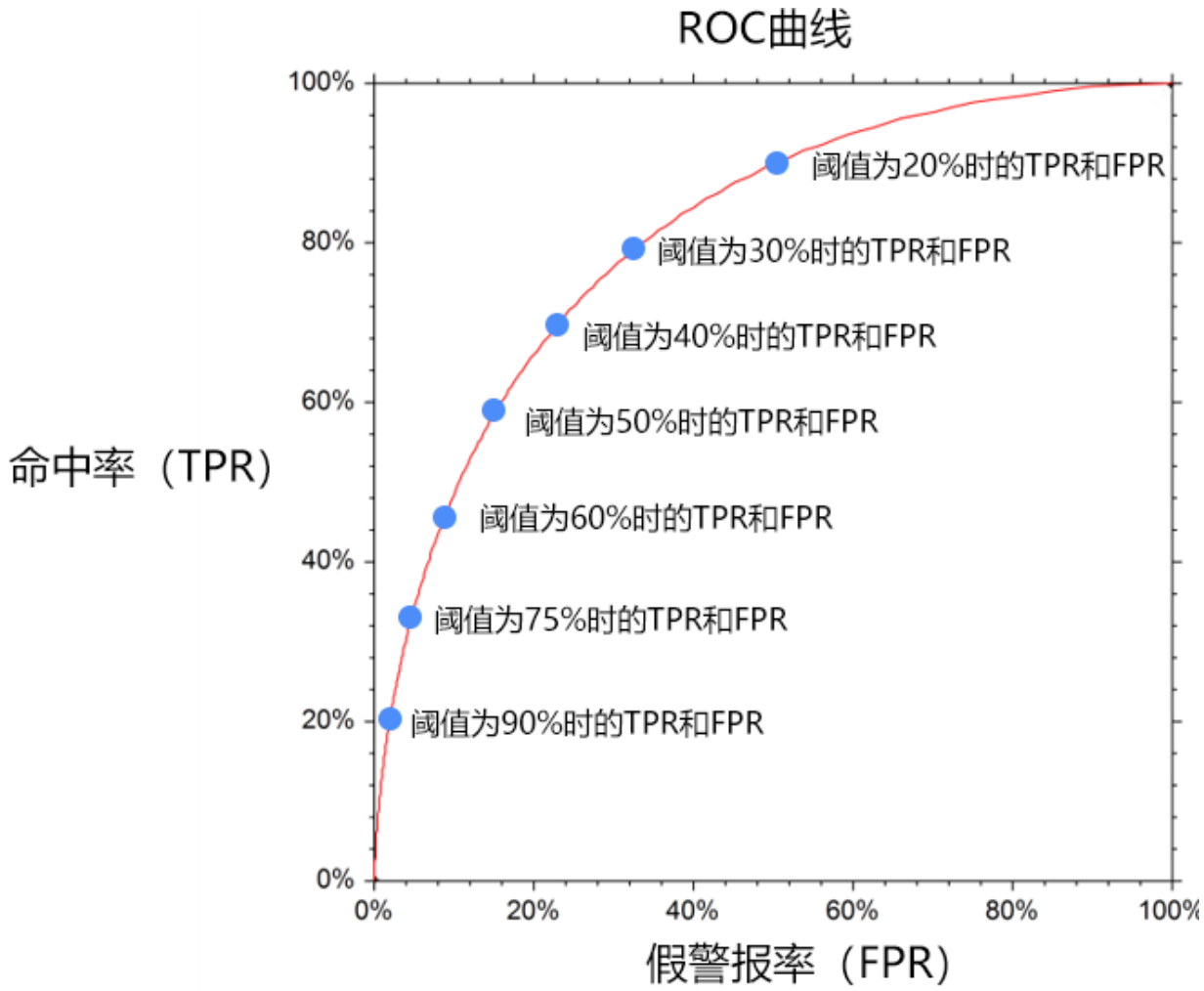
而如果降低阈值的话，比如认为违约率超过10%就认定为违约，那么真正率就会很高，但是假正率也会很高。

客户违约预测模型搭建

模型预测及评估

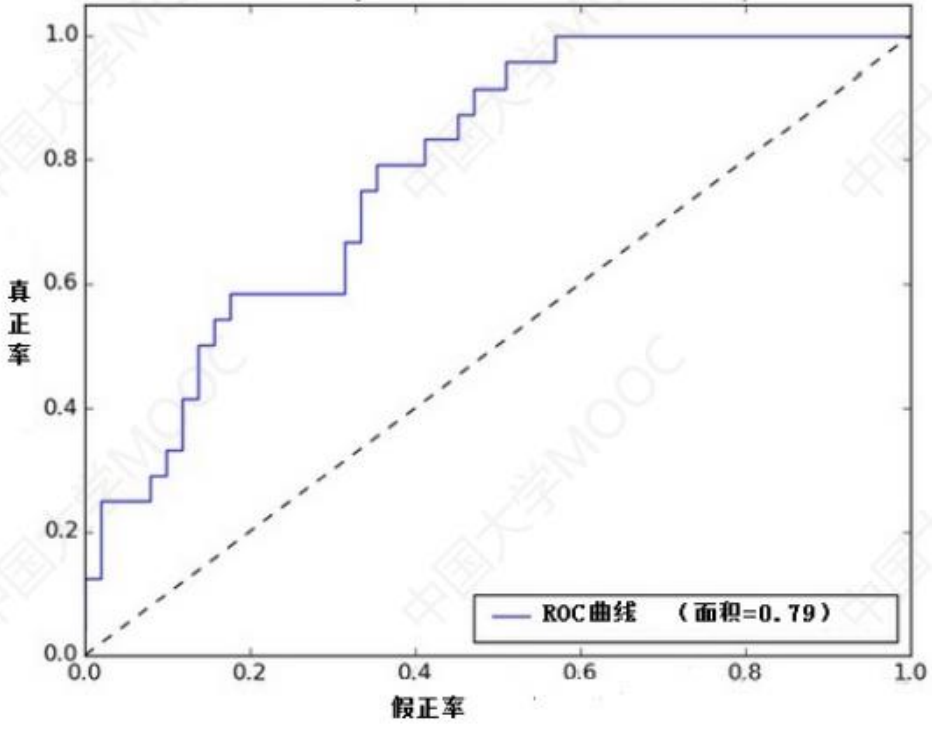
(3) 模型预测效果评估

因此为了衡量一个模型的优劣，数据科学家根据不同阈值下的真正率和假正率绘制了如下的曲线图，称之为ROC曲线：



机器学习算法的性能度量——分类

ROC曲线（receiver operating characteristic curve）



根据分类结果计算得到ROC空间中相应的点，连接这些点形成ROC曲线

真正率(TPR)：预测为正的正样本数 /正样本实际数

$$TPR=TP/(TP+FN)$$

假正率(FPR)：预测为正的负样本数 /负样本实际数

$$FPR=FP/(FP+TN)$$

靠近左上角的ROC曲所代表的分类器准确性最高

PR（precision recall curve）：precision 对 recall的曲线

机器学习算法的性能度量——分类

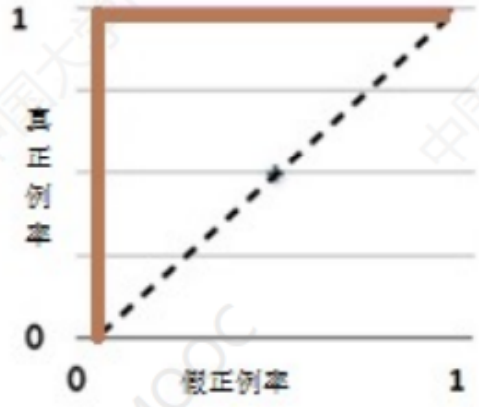
- **AUC**（area under curve）：ROC曲线下的面积(ROC的积分)

AUC = 1：100%完美识别正负类，不管阈值怎么设定都能得出完美预测；

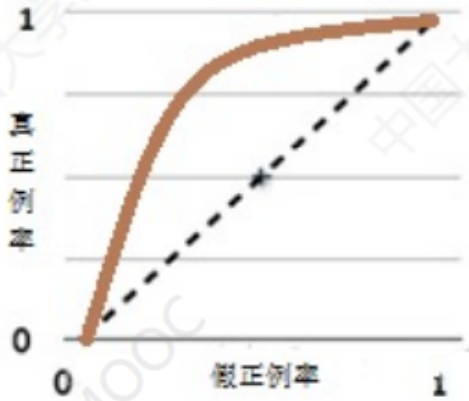
0.5<AUC<1：优于随机猜测。这个分类器（模型）妥善设定阈值的话，可能有预测价值；

AUC=0.5：跟随机猜测一样（例：随机丢N次硬币，正反面的概率为50%），模型无预测价值；

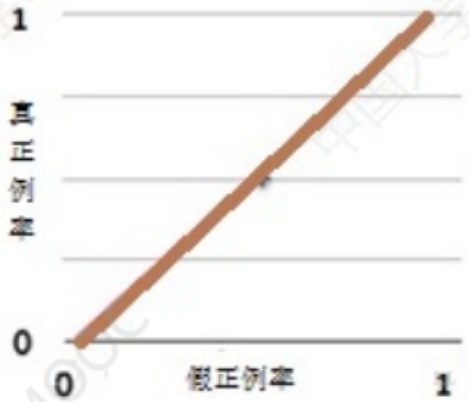
AUC<0.5：比随机猜测还差，不存在AUC<0.5的情况



AUC=1



AUC=0.8



AUC=0.5

Thank you