机器学习

1. 机器学习基础
   1. 定义
   2. 主要任务
      1. 分类：实例数据划分，通常为标称型数据
      2. 回归：预测数值型数据，通常为连续型数据
   3.  开发步骤
      1. 问题建模
      2. 获取数据
      3. 特征工程
      4. 模型训练与验证
      5. 模型诊断与调优
      6. 线上运行
   4. 机器学习相关库
      1. numpy

数值编程工具、矩阵数据类型、矢量处理。NumPy的全名为Numeric Python，是一个**开源的Python科学计算库**，它包括：

* 一个强大的N维数组对象ndrray
* 比较成熟的（广播）函数库；
* 用于整合C/C++和Fortran代码的工具包；
* 实用的线性代数、傅里叶变换和随机数生成函数

**NumPy的优点：**

* 对于同样的数值计算任务，使用NumPy要比直接编写Python代码便捷得多；
* NumPy的大部分代码都是用C语言写的，其底层算法在设计时就有着优异的性能，这使得NumPy比纯Python代码高效得多
* NumPy中的数组的存储效率和输入输出性能均远远优于Python中等价的基本数据结构，且其能够提升的性能是与数
* 组中的元素成比例的；

**基本操作：**

* 数组对象ndarray的创建
* 数组对象ndarray的数据类型
* 索引和切片
* 数组函数
* 线性代数

* + 1. pandas

基于NumPy和Matplotlib的数据处理库。包括数据分析和数据可视化，特有的数据结构： Series和DataFrame

**基本操作：**

* 创建Series、 DataFrame对象
* 查看数据
* 选择数据
* 缺失值处理
* 合并操作
* 分组
* 画图
* 对象导入和保存

**Series：**

如同列表一样，一系列数据，每个数据对应一个索引值。

**Series与List的区别：**

列表的索引只能是从 0 开始的整数， Series 数据类型在默认情况下，其索引也是如此。不过，区别于列表的是，Series可以自定义索引：

**DataFrame：**

是一种**二维的数据结构**，非常接近于电子表格或者类似 mysql 数据库的形式。它的竖行称之为 columns，横行跟前面的 Series 一样，称之为 index，也就是说可以通过 columns 和index 来确定一个元素的位置

。

* + 1. matplotlib

可视化绘图库

**基本操作：**

* Line plot（折线图）
* Scatter plot（散掉图）
* Histogram（直方图）
* Bar（条形图、柱状图）
* Box plot（箱型图）
  + 1. scikit-learn

**机器学习库**，基于NumPy, SciPy, Matplotlib,开源，涵盖分类，回归和聚类算法 代码和文档完备。

* sklearn的数据结构基于numpy和pandas;
* sklearn的数据计算基于scipy;
* sklearn的数据可视化基于matplotlib;
* Sklearn的六大基本功能

**功能：**

* 分类
* 回归
* 聚类
* 梯度衰减
* 模型选择
* 预处理

**三个步骤：**

1. 数据准备与预处理
2. 模型选择与训练
3. 模型验证与参数调优

* 1. 统计学习三要素

方法=模型+策略+方法

* + 1. 模型
    2. 策略

损失函数和风险函数

经验风险最小化和结构风险最小化

* + 1. 算法
    2. 模型评估

训练误差

测试误差

过拟合和

* + 1. 模型选择

正则化

交叉验证

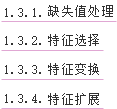
* 1. 泛化能力
     1. 泛化误差
     2. 泛化误差上界
  2. 生成模型与判别模型

1. 特征工程

**数据和特征决定了机器学习的上限**， 而模型和算法只是逼近这个上限而已。

特征工程是将原始数据，通过业务逻辑理解、数据变换、特征交叉与组合等方式，量化成模型训练和预测可直接使用的特征的过程。其中主要包括了**数据认知，数据清洗，特征提取，特征选择**四个部分。

* 基本特征
* 统计特征
* 复杂特征
* 自然特征



* 1. 获取与存储
  2. 缺失值处理
  3. 特征处理
     1. 数据清洗
     2. 数据预处理

标准化（StandardScaler）

区间缩放（MinMaxScaler）

归一化（Normalizer）

二值化（Binarizer）

哑编码（OneHotEncoder）

缺失值计算（Imputer）

多项式数据转换（PolynomialFeatures）

* + 1. 特征选择理论

过滤法（Filter）

包装法（Wrapper）

集成法（Embedded）

* + 1. 特征选择

方差选择法（VarianceThreshold）

相关系数法（SelectKBest）

卡方检验（SelectKBest）

互信息法（SelectKBest）

递归特征消除法（RFE）

基于惩罚项的特征选择法（SelectFromModel）

基于树模型的特征选择法（SelectFromModel）

* + 1. 嵌入法特征选择

正则项特征选择

树模型特征选择

* + 1. 特征扩展
    2. 更新特征

1. 算法分类
   1.  监督学习 (supervised learning)

监督学习的**训练集的所有结果都是已知**的。

**特点：**

* 根据已知类别的样本进行学习
* 数据集的创建和分类 + 模型训练 + 效果验证 + 模型使用

监督学习一般使用两种类型的目标变量：

**标称型：**

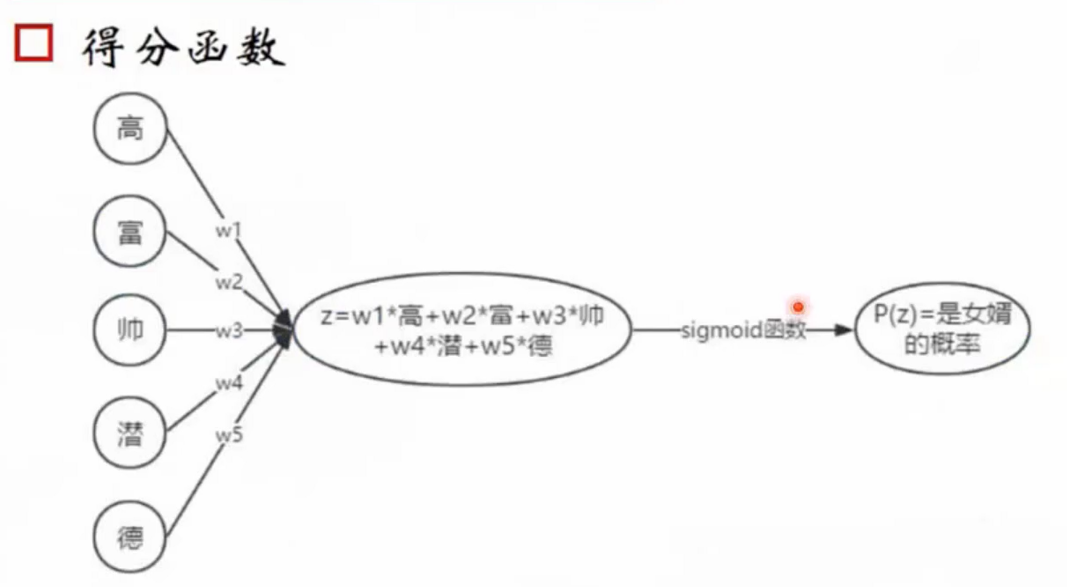
目标变量的结果**只在有限目标集中取值**。如真与假，动物分类{爬行类，鱼类，哺乳类...}

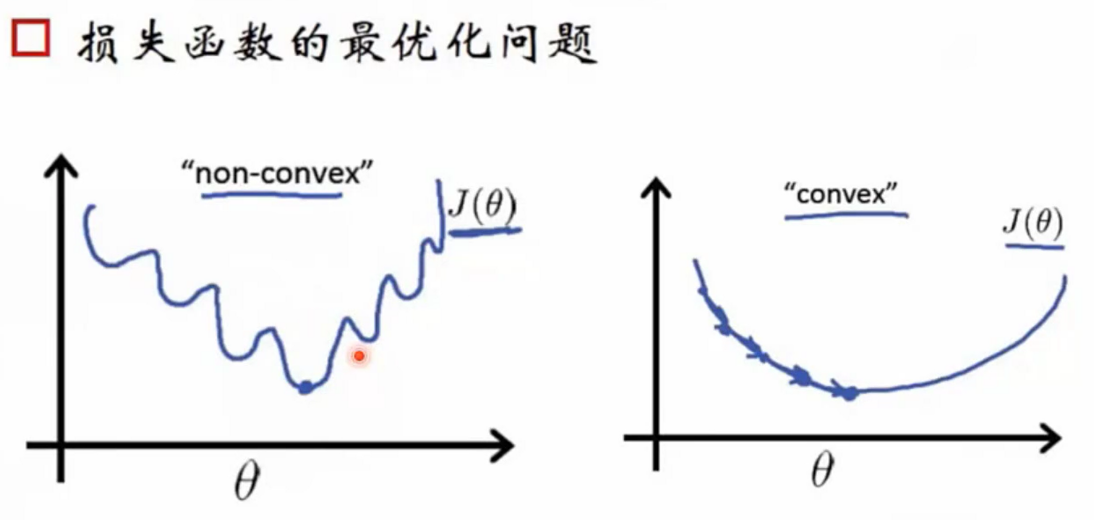
**数值型：**

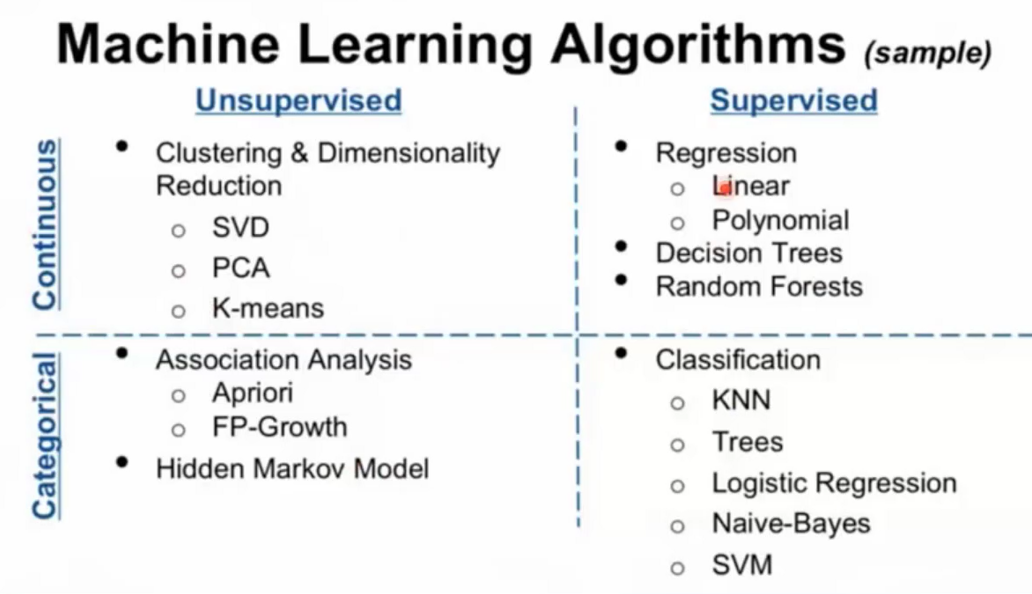
目标变量**可以从无限的数值集合中取值**。如0.100、42.01、100.73。主要用于回归分析。

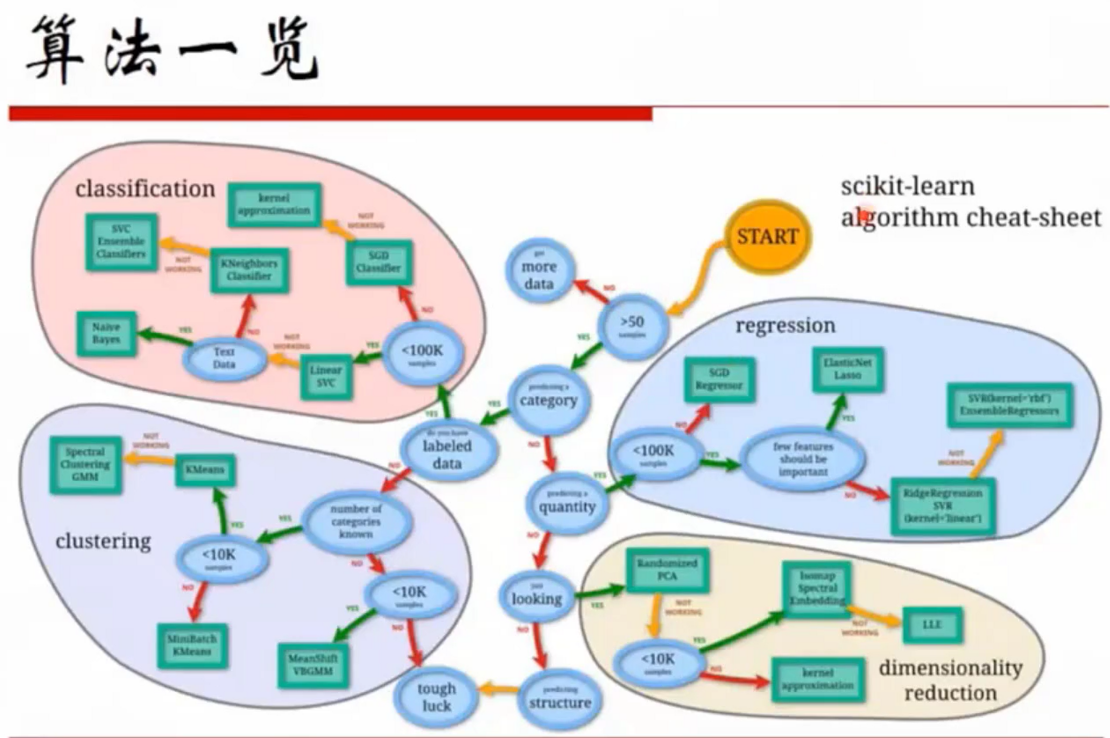
**分类与回归的区别：**

分类是对离散变量的预测，回归是对连续变量的预测



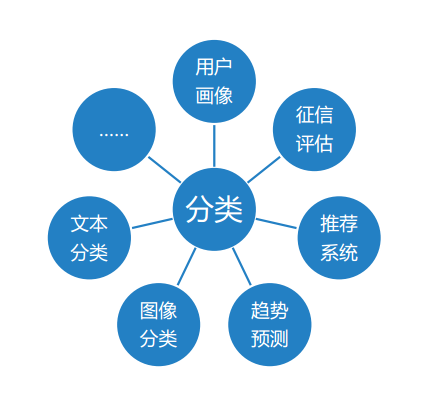
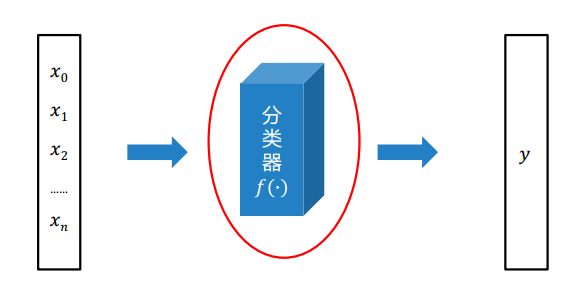






* + 1. 分类（classification）

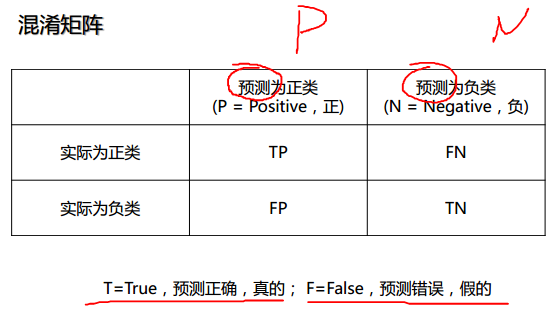
1. **问题定义：**



**分类问题分为：**

* 二分类（TA是男是女？）
* 多分类（TA的年终奖评定S?A?B?C?），其中又分为互斥和非互斥多分类

1. **评价指标（混淆矩阵）：**





* **真正率（ True Positive Rate, TPR），也称灵敏度（sensitivity）：**graphic
* **真负率 (True Negative Rate, TNR），特指度（specificity）：**graphic
* **假正率 （False Positive Rate, FPR）：**graphic
* **假负率（ False Negative Rate , FNR）：**graphic
* **精准率/查准率 (Precision)：**graphic
* **召回率/查全率 (Recall)，等于 真正率TPR：**graphic
* **ROC (receiver operating characteristic curve)：**graphic

二分类模型返回一个概率值，通过调整阈值，即大于该阈值为正类，反之负类，可以得到多个**（FPR，TPR）**点，描点画图得到的曲线即为ROC曲线。 ROC曲线是根据一系列不同的二分类方式（分界值或决定阈），以真阳性率（灵敏度）为纵坐标，假阳性率（1-特异度）为横坐标绘制的曲线。传统的诊断试验评价方法有一个共同的特点，必须将试验结果分为两类，再进行统计分析。ROC曲线的评价方法与传统的评价方法不同，无须此限制，而是根据实际情况，允许有中间状态，可以把试验结果划分为多个有序分类，如正常、大致正常、可疑、大致异常和异常五个等级再进行统计分析。**越靠近（0,1）效果越好。**

* **AUC(Area Under Curve)：**

AUC（Area Under Curve）被定义为ROC曲线下的面积，显然这个面积的数值不会大于1。又由于ROC曲线一般都处于y=x这条直线的上方，所以**AUC的取值范围在0.5和1之间。**使用AUC值作为评价标准是因为很多时候ROC曲线并不能清晰的说明哪个分类器的效果更好，而作为一个数值，对应**AUC越大的分类器效果更好**

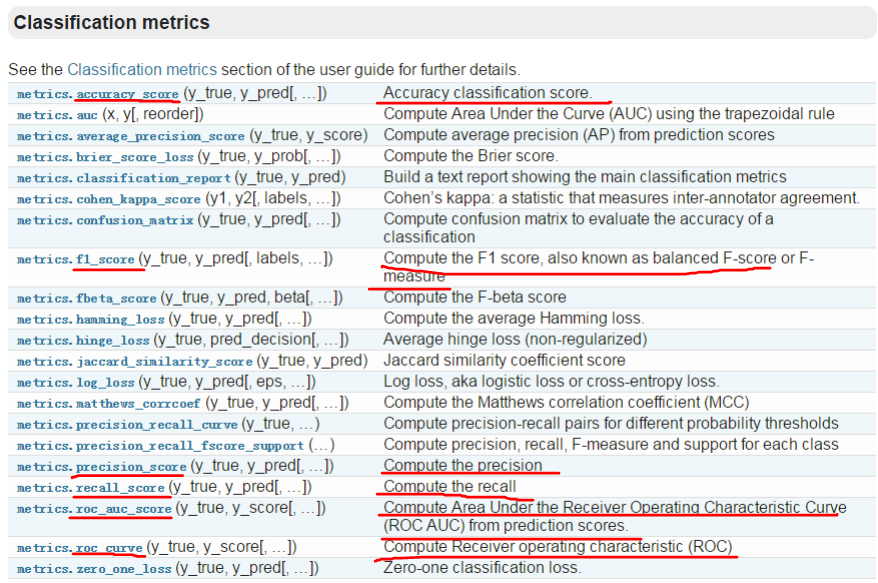
* **精确率和召回率的调和平均数（ F1-score）：**graphic**——》** graphic
* **Accuracy（准确率/正确率）=（TP+TN）/ (P+N)：**

对于给定的测试数据集，分类器**正确分类的样本数与总样本数之比**，从某种意义上得到一个分类器是否有效，但它并不总是能有效的评价一个分类器的工作

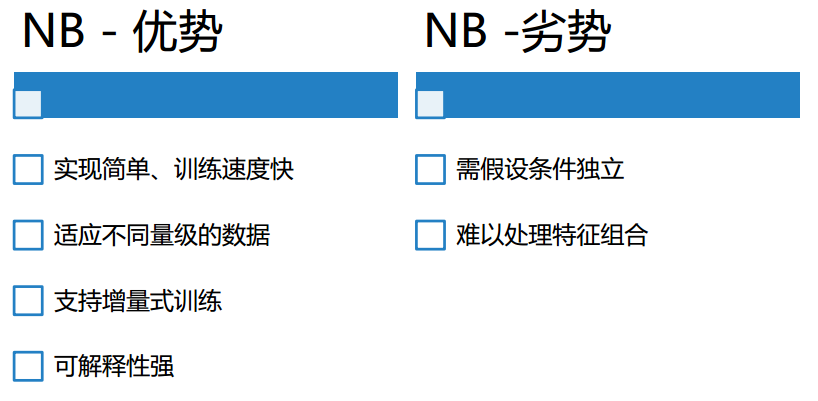
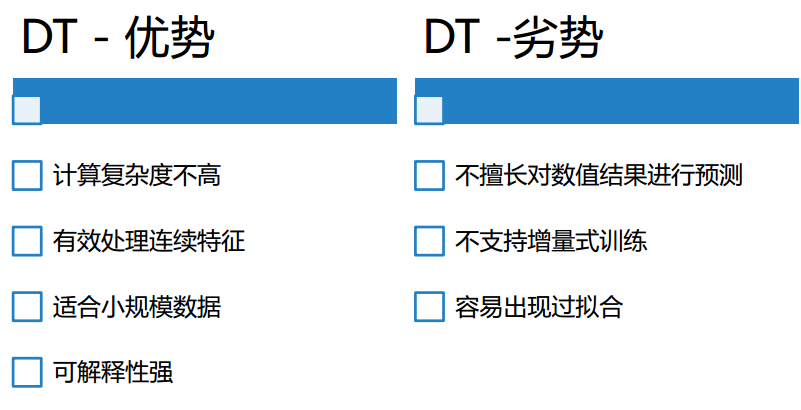
* **Error rate（错误率）= (FP+FN) / (P+N)：**

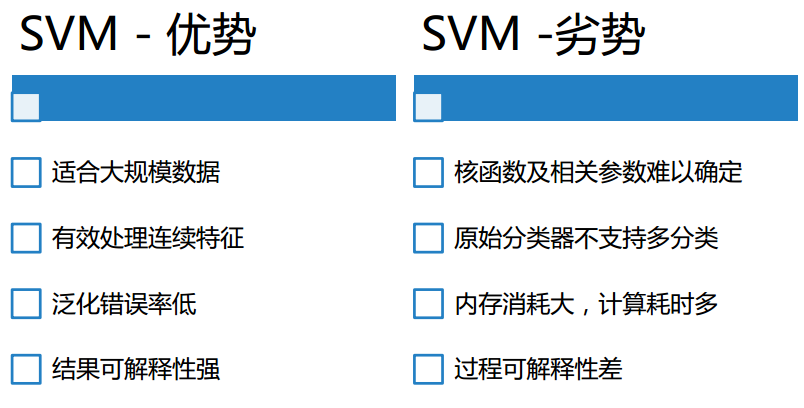
对于给定的测试数据集，分类器**错误分类的样本数与总样本数之比**，分对与分错是互斥事件，所以accuracy =1 - error rate

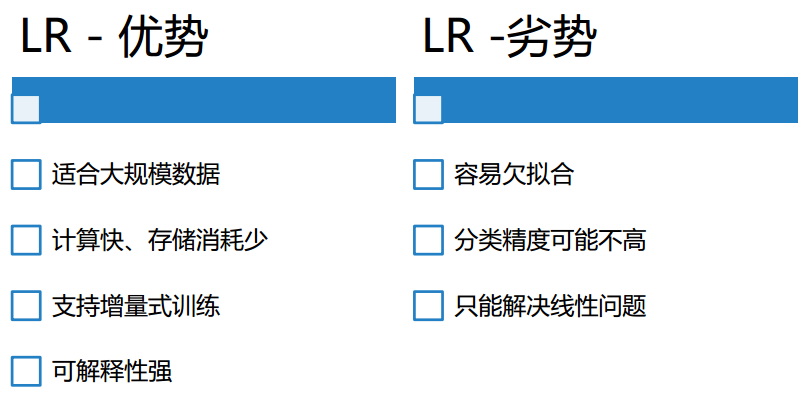
1. **API参考**



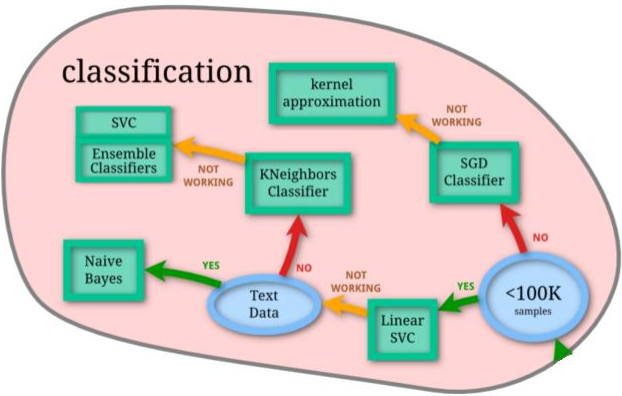
1. **分类算法对比**



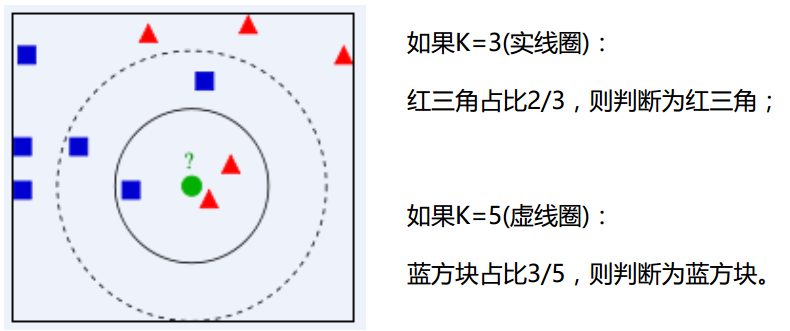
1. **算法选择**



K近邻（KNN）

* **定义：**

K最近邻 (KNN，K-Nearest Neighbor)数据挖掘分类技术中最简单的方法之一，所谓K最近邻，就是K个最近的邻居，每个样本都可以用它最接近的K个邻居来代表



1. **算法步骤：**

1. **计算**已知类别数据集中的点与当前之间的**距离**

2. 按照距离递增次序**排序**

3. **选取**与当前点距离最小的**k个点**

4. **确定前k个点**所在的类别的出现**频率**

5. 返回前k个点出现**频率最高的类别作为当前点的预测分类**

* **三要素：**
* k值的选择：
* 距离度量：
* 分类规则选择：

* **特点：**

**优点：精度高，**对异常值不敏感，无数据输入假定

**缺点：计算复杂度高，空间复杂度高**

* **API参考：**

graphic

**n\_neighbor：**

需要设定的K值

**weights:**

uniform：权重相等

distance：权重随距离的增加而减小

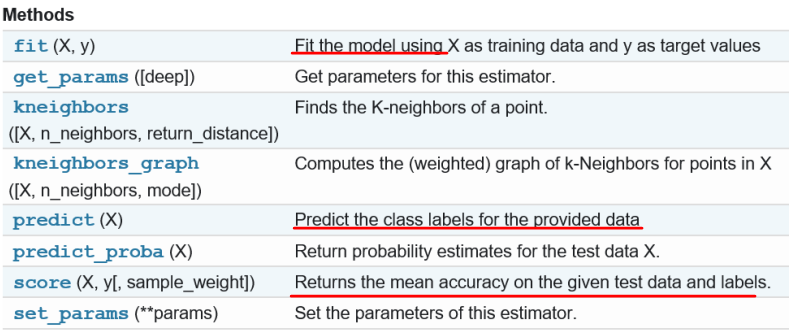
**algorithm：**

ball\_tree： 是为了克服KD树高维失效而发明的，其构造过程是以质心C和半径r分割样本空间，每一个节点是一个超球体

kd\_tree： 是依次对K维坐标轴，以中值切分构造的树，每一个节点是一个超矩形，在维数小于20时效率最高

brute：

auto：



决策树（Decision tree）

**决策树与随机森林**

1. 树模型概述
2. 决策树算法
3. Bagging & 随机森林算法
4. Boosting & GBDT

贝叶斯（Nave Bayes）

逻辑回归（Logistic Regression）

支持向量机（SVM）

核函数

最小序列优化（SMO）

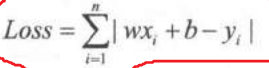
神经网络（NN）

* + 1. 回归（regression）
* **评价指标**
* Variance(方差)
* Bias（偏差）
* MSE（均方误差）
* RMSE（均方根误差）
* R-Squard（R方）
* MAE（平均绝对误差）

**回归分类：**

线性回归（Linear Regression）：  **y = f(x) = wx + b**

w 为的1 x n行矩阵，x为n x 1的列矩阵，wb为矩阵w与矩阵x的内积，b为偏置项。

损失函数定义： 

非线性回归：

K近邻（KNN）

支持向量机（SVM）

线性回归（LinearRegression）

贝叶斯回归（BayesRegression）

常用的参数估计方法

极大似然估计

最大后验估计

贝叶斯估计

最大熵估计

混合模型估计

决策树（Decision Tree）

RF

GBDT

* 1.  无监督学习 (unsupervised learning)

无监督学习，所有的结果都是未知的。

**特点：**

* 缺乏足够的先验知识
* 难以人工标注类别或进行人工类别标注的成本太高
* 根据类别未知(没有被标记)的样本来学习
  + 1. 聚类（cluster）

**本质**：寻找联系紧密的事物进行区分，将数据划分成有意义或有用的簇

**目标**：簇内的对象相互之间是相似的，而不同组中的对象是不同的

**核心**：相似度计算

* **评价指标**

**内部评价**

* compactness（紧密性CP）
* Separation（间隔性SP）
* Davies-Bouldin Index（戴维森堡丁指数DB）/（分类适确性指标DBI）
* Dunn Validity Index（邓恩指数DVI）

**外部评价**

* Cluster Accuracy（准确性CA）
* Rand index（兰德指数RI）
* Adjusted Rand index（调整兰德指数ARI )
* Normalized Mutual Information（标准互信息NMI）
* Mutual Information（互信息MI）

* **距离度量**
* 曼哈顿距离
* 欧几里得距离
* 明可夫斯基距离
* 切比雪夫距离

* **相似度量**
* 向量空间余弦相似度
* 皮尔森相关系数

划分聚类

K-means

K-medoids

CLARANS

DIANA

层级聚类

BIRCH

Chameleon

基于密度

OPTICS

DBSCAN

FDC

基于图论

Spectral

基于网络

STING

CLIQUE

基于模型

COBWeb

CLASSIT

SOM

* + 1. 关联规则
  1. 半监督学习(semisupervised learning)

小部分结果已知。

**特点：**

* 少量的标注样本和大量的未标注样本
* 尽量少的人工，带来尽量大的价值
* 平滑假设(Smoothness Assumption)
* 聚类假设(Cluster Assumption)
* 流形假设(Manifold Assumption)
  + 1. 最大期望（EM）
    2. 生成模型（Generative Model）
    3. 图论推导（Graph Inference）
    4. 拉普拉斯支持向量机（Laplacian SVM）
  1. 强化学习(reinforcement learning)

**特点：**

* 是否越来越接近目标（回报函数，reward function）
* 输入数据直接反馈到模型，模型必须对此立刻作出调整
* 常见的应用场景包括动态系统以及机器人控制

* + 1. Q-Learning
    2. 时间差学习（Temporal difference learning）

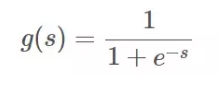
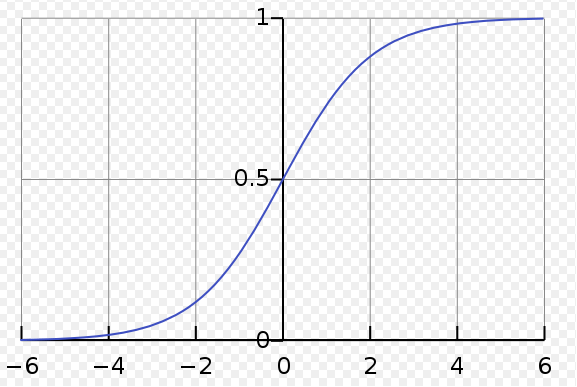
1. 项目实战
   1. O2O优惠券使用预测
   2. Credit Card Fraud Detection
   3. Click Throughrate Prediction
2. 知识总结
   1. 损失函数函数的选择
      1. 分类

* **损失函数（Loss Function）**

用来**估量模型的预测值 f(x) 与真实值 y 的不一致程度**。若损失函数很小，表明机器学习模型与数据真实分布很接近，则模型性能良好；若损失函数很大，表明机器学习模型与数据真实分布差别较大，则模型性能不佳。我们训练模型的主要任务就是使用优化方法来寻找损失函数最小化对应的模型参数。

一般来说，二分类机器学习模型包含两个部分：线性输出 s 和非线性输出 g(s)。其中，线性输出一般是模型输入 x 与 参数 w 的乘积，简写成：

**s = wx**；非线性输出一般是 **Sigmoid 函数**，其表达式如下：

g(s) 值被限定在 [0,1] 之间，若 s ≥ 0，g(s) ≥ 0.5，则预测为正类；若 s < 0，g(s) < 0.5，则预测为负类。

* **正类和负类的表示**

通常有两种方式：

* **一种是用 {+1, -1} 表示正负类；**
* **另一种是用 {1, 0} 表示正负类**。下文默认使用 {+1, -1}，因为这种表示方法有种好处。

如果使用 {+1, -1} 表示正负类，我们来看预测类别与真实类别的四种情况：

* **s >= 0, y = +1: 预测正确**
* **s >= 0, y = -1: 预测错误**
* **s < 0, y = +1: 预测错误**
* **s < 0, y = -1: 预测正确**

通用表达：

* **若 ys >= 0，则预测正确**
* **若 ys < 0，则预测错误**

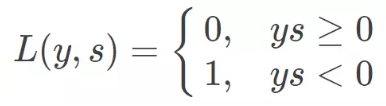
这里的 ys 类似与回归模型中的残差 s - y。因此，在比较分类问题的各个损失函数的时候，我们就可以把 ys 当作自变量 x 轴即可，这样更加方便。

0-1 Loss

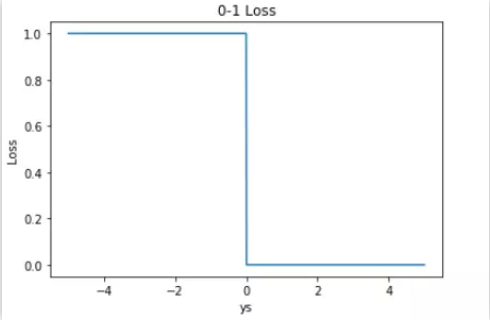
0-1 Loss 是最简单也是最容易直观理解的一种损失函数。对于二分类问题:

* **如果预测类别 y\_hat 与真实类别 y 不同，则 L=1；**
* **如果预测类别 y\_hat 与 真实类别 y 相同，则 L=0（L 表示损失函数）。**

**0-1 Loss 的表达式为：**



**0-1 Loss 的曲线如下图所示：**

 ys为自变量

0-1 Loss 的特点就是非常直观容易理解。但是它存在两个缺点：

* 0-1 Loss 对每个错分类点都施以相同的惩罚（损失为 1），这样对犯错比较大的点（ys 远小于 0）无法进行较大的惩罚，所有犯错点都同等看待，这不符合常理，不太合适。
* 0-1 Loss 不连续、非凸、不可导，难以使用梯度优化算法。

因此，实际应用中，0-1 Loss 很少使用。

Cross Entropy Loss

Cross Entropy Loss 是**非常重要**的损失函数，也是应用最多的损失函数之一。

二分类问题的交叉熵 Loss 主要有两种形式，下面分别详细介绍。

第一种形式是基于输出标签 label 的表示方式为 {0,1}，也最为常见。

它的 Loss 表达式为：

这个公式是如何推导的呢？很简单，从极大似然性的角度出发，预测类别的概率可以写成：

我们可以这么来看，当真实样本标签 y = 1 时，上面式子第二项就为 1，概率等式转化为：

当真实样本标签 y = 0 时，上面式子第一项就为 1，概率等式转化为：

我们希望的是概率 P(y|x) 越大越好。首先，我们对 P(y|x) 引入 log 函数，因为 log 运算并不会影响函数本身的单调性。则有：

我们希望 log P(y|x) 越大越好，反过来，只要 log P(y|x) 的负值 -log P(y|x) 越小就行了。那我们就可以引入损失函数，且令 Loss = -log P(y|x) 即可。则得到损失函数为：

我们来看，当 y = 1 时：

因为，

所以，代入到 L 中，得：

这时候，Loss 的曲线如下图所示：

从图中明显能够看出，s 越大于零，L 越小，函数的变化趋势也完全符合实际需要的情况。

当 y = 0 时：

对上式进行整理，同样能得到：

这时候，Loss 的曲线如下图所示：

从图中明显能够看出，s 越小于零，L 越小，函数的变化趋势也完全符合实际需要的情况。

第二种形式是基于输出标签 label 的表示方式为 {-1,+1}，也比较常见。它的 Loss 表达式为：

下面对上式做个简单的推导，我们在 0 小节说过，ys 的符号反映了预测的准确性。除此之外，ys 的数值大小也反映了预测的置信程度。所以，从概率角度来看，预测类别的概率可以写成：

分别令 y = +1 和 y = -1 就能很容易理解上面的式子。

接下来，同样引入 log 函数，要让概率最大，反过来，只要其负数最小即可。那么就可以定义相应的损失函数为：

这时候，我们以 ys 为横坐标，可以绘制 Loss 的曲线如下图所示：

其实上面介绍的两种形式的交叉熵 Loss 是一样的，只不过由于标签 label 的表示方式不同，公式稍有变化。标签用 {-1,+1} 表示的好处是可以把 ys 整合在一起，作为横坐标，容易作图且具有实际的物理意义。

总结一下，交叉熵 Loss 的优点是在整个实数域内，Loss 近似线性变化。尤其是当 ys << 0 的时候，Loss 更近似线性。这样，模型受异常点的干扰就较小。 而且，交叉熵 Loss 连续可导，便于求导计算，是使用最广泛的损失函数之一。

Hinge Loss

Exponential Loss

Modified Huber Loss

Softmax Loss

* + 1. 回归

1. 机器学习算法总结
   1. 其他
      1. 支持向量机（SVM）
      2. 特征选择算法
      3. Algorithm accuracy evaluation
      4. Performance measures
   2. 降维算法
      1. 主成分分析法（PCA）
      2. 主成分回归（PCR）
      3. 偏最小二乘回归（PLSR）
      4. 萨蒙映射
      5. 多维尺度分析法（MDS）
      6. 投影寻踪法（PP）
      7. 线性判别分析法（LDA）
      8. 混合判别分析法（MDA）
      9. 二次判别分析法（QDA）
      10. 灵活判别分析法（Flexible Discriminant Analysis，FDA）
   3. 模型融合算法
      1. Boosting
      2. Bagging
      3. AdaBoost
      4. 堆叠泛化（混合）
      5. GBM 算法
      6. GBRT 算法
      7. 随机森林
   4. 深度学习算法
      1. 深度玻尔兹曼机（DBM）
      2. 深度信念网络（DBN）
      3. 卷积神经网络（CNN）
      4. 栈式自编码算法（Stacked Auto-Encoder）
   5. 人工神经网络
      1. 感知机
      2. 反向传播算法（BP 神经网络）
      3. Hopfield网络
      4. 径向基函数网络（RBFN）
   6. 关联规则学习

从大规模数据集中寻找物品间的**隐含关系**被称为**关联分析（association analysis）**或者**关联规则学习（association rule learning）**

* + 1. Apriori 算法

Subtopic

* + 1. Eclat 算法
  1. 聚类算法
     1. K-均值
     2. K-中位数
     3. EM 算法
     4. 分层聚类算法
  2. 贝叶斯算法
     1. 朴素贝叶斯算法
     2. 高斯朴素贝叶斯算法
     3. 多项式朴素贝叶斯算法
     4. AODE 算法
     5. 贝叶斯信念网络（BBN）
     6. 贝叶斯网络（BN）
  3. 决策树算法
     1. 分类和回归树（CART）
     2. ID3 算法
     3. C4.5 算法和 C5.0
     4. CHAID 算法
     5. 单层决策树
     6. M5 算法
     7. 条件决策树
  4. 正则化算法
     1. 岭回归
     2. LASSO 算法
     3. Elastic Net
     4. 最小角回归算法（LARS）
  5. 基于实例的学习算法
     1. k-邻近算法（kNN）
     2. 学习矢量量化算法（LVQ）
     3. 自组织映射算法（SOM）
     4. 局部加权学习算法（LWL）
  6. 回归算法
     1. 普通最小二乘回归（OLSR）
     2. 线性回归
     3. 逻辑回归
     4. 逐步回归
     5. 多元自适应回归样条法（MARS）
     6. 局部估计平滑散点图（LOESS）