机器学习-分类算法

2021年4月5日 21:11

参考：
《数据挖掘算法——常用分类算法总结》
https://blog.csdn.net/songguangfan/article/details/92581643
《一文读懂机器学习分类算法（附图文详解）》
https://zhuanlan.zhihu.com/p/82114104

常用的分类算法包括：

- LR（Logistic Regress，逻辑回归）
- KNN（K-Nearest Neighbor，K 最近邻近）
- SVM（Support Vector Machine，支持向量机）
- NBC（Naive Bayesian Classifier，朴素贝叶斯分类）
- 决策树（Decision Tree）
  https://www.cnblogs.com/molieren/articles/10664954.html
  ID3（Iterative Dichotomiser 3 迭代二叉树3 代）决策树
  C4.5 决策树
  C5.0 决策树
  CART 分类回归树 https://www.cnblogs.com/keye/p/10564914.html
  https://www.cnblogs.com/keye/p/10601501.html

集成方法

- 随机森林（Random Forest）
  https://blog.csdn.net/yangyin007/article/details/82385967
- GBDT:gradient boosting decision tree 梯度增强决策树
  https://www.cnblogs.com/bnuvincent/p/9693190.html

- ANN（Artificial Neural Network，人工神经网络）

分类器性能评价

- 混淆矩阵
- 接受者操作曲线（ROC）和曲线下的面积（AUC）

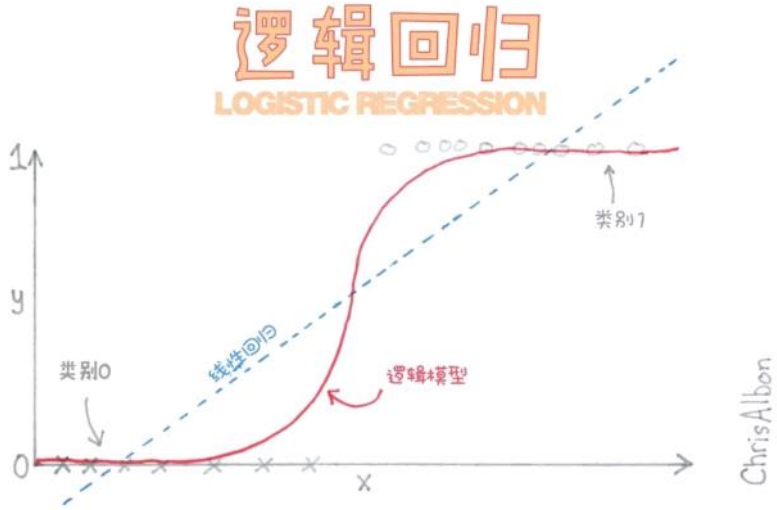
1. LR

a. 概述

LR 回归是当前业界比较常用的机器学习方法，用于估计某种事物的可能性。它与多元线性回归同属一个家族，即广义线性模型。简单来说多元线性回归是直接将特征值和其对应的概率进行相乘得到一个结果，逻辑回归则是在这样的结果上加上一个逻辑函数。在此选择LR 作为回归分析模型的代表进行介绍。

b. 详解

逻辑回归类似于线性回归，适用于因变量不是一个数值字的情况 (例如，一个“是/否”的响应)。它虽然被称为回归，但却是基于根据回归的分类，将因变量分为两类。



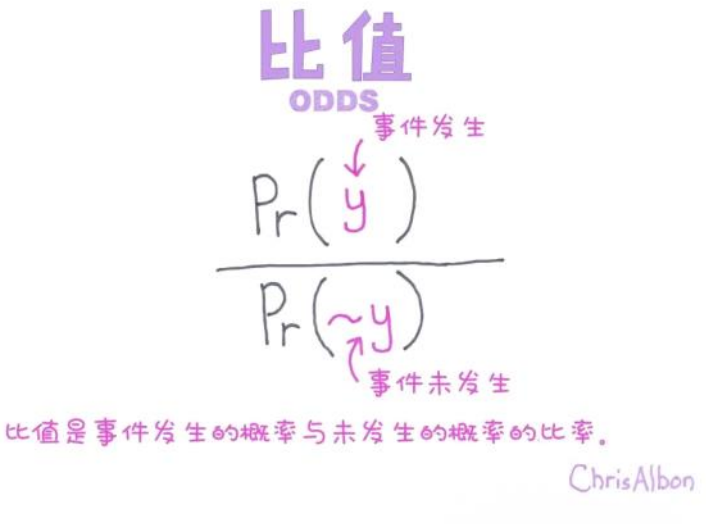
如上所述，逻辑回归用于预测二分类的输出。例如，如果信用卡公司构建一个模型来决定是否通过向客户的发行信用卡申请，它将预测客户的信用卡是否会“违约”。

$$y = b_0 + b_1 x$$

首先对变量之间的关系进行线性回归以构建模型，分类的阈值假设为0.5。

$$p = \frac{1}{1 + e^{-y}}$$

然后将Logistic函数应用于回归分析，得到两类的概率。
该函数给出了事件发生和不发生概率的对数。最后，根据这两类中较高的概率对变量进行分类。



c. LR算法的优点

- 对数据中小噪声的鲁棒性好
- LR 算法已被广泛应用于工业问题中
- 多重共线性并不是问题，它可结合正则化来解决

d. LR算法的缺点

- 对于非线性特征，需要转换
- 当特征空间很大时，LR的性能并不是太好

2. KNN

a. 概述

KNN 算法是Cover 和Hart 于1968 年提出的理论上比较成熟的方法，为十大挖掘算法之一。该算法的思路非常简单直观：如果一个样本在特征空间中的k 个最相似(即特征空间中最邻近)的样本中的大多数属于某一个类别，则该样本也属于这个类别。该方法在定类决策上只依据最邻近的一个或者几个样本的类别来决定待分样本所属的类别。

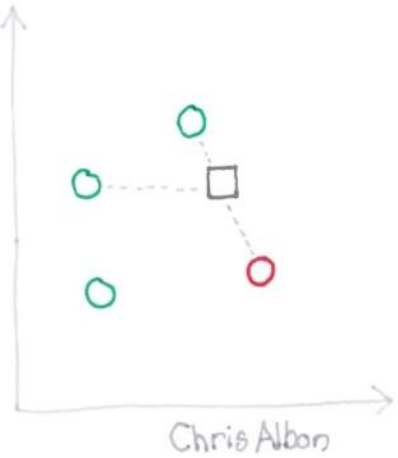
b. 详解

K-NN算法是一种最简单的分类算法，通过识别被分成若干类的数据点，以预测新样本点的分类。K-NN是一种非参数的算法，是“懒惰学习”的著名代表，它根据相似性（如，距离函数）对新数据进行分类。

KNN算法
k-nearest neighbors

-k代表参与计算的邻近单元数量;
-尺的取值比较重要;
-k值一般为奇数;
-如果我们有二进制特征，可以利用海明距离计算;
-以待分类单元与邻近单元的距离为权重进行分类;
-不适合用于大量数据的分类。

假设k=3，那么灰色方块将被预测为绿色，这是因为与它最邻近的三个单元中，两个是绿色的，一个是红色的。

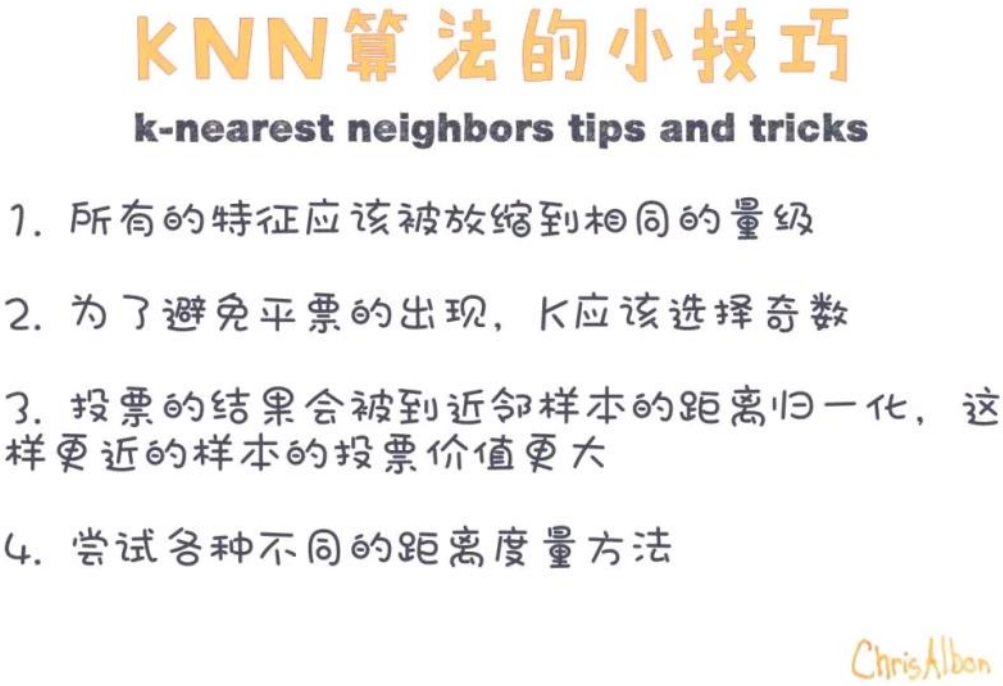


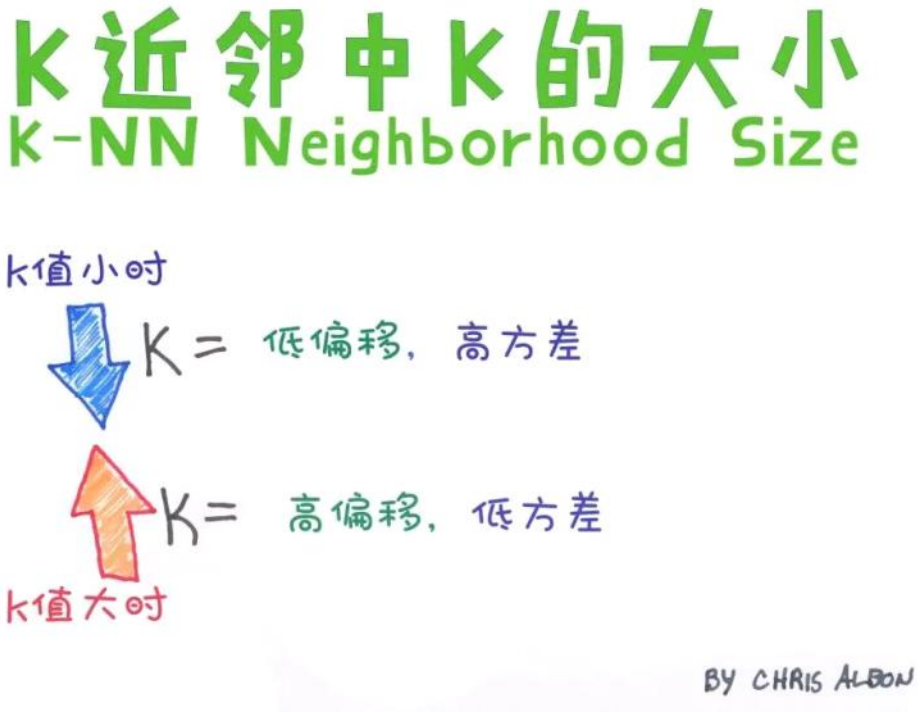
KNN算法
Does K-NN Learn

KNN算法
样本A最接近的k个样本本身不训练，他是懒惰的，仅仅记住所有数据

*译者注:
KNN可理解为一种死记硬背式的分类器，记住所有的训练数据，对于新的数据则直接和训练数据匹配，如果存在相同属性的训练数据，则直接用它的分类来作为新数据的分类。

Chris Albon





K-NN能很好地处理少量输入变量（p）的情况，但当输入量非常大时就会出现问题。

c. KNN算法的优点

- KNN 算法简单、有效
- KNN 算法适用于样本容量比较大的类域的自动分类
- 由于KNN 方法主要靠周围有限的邻近的样本，而不是靠判别类域的方法来确定所属类别的，因此对于类域的交叉或重叠较多的待分样本集来说，KNN 方法较其他方法更为适合

d. KNN算法的缺点

- KNN 算法计算量较大
- KNN 算法需要事先确定K 值
- KNN 算法输出的可解释不强
- KNN 算法对样本容量较小的类域很容易产生误分

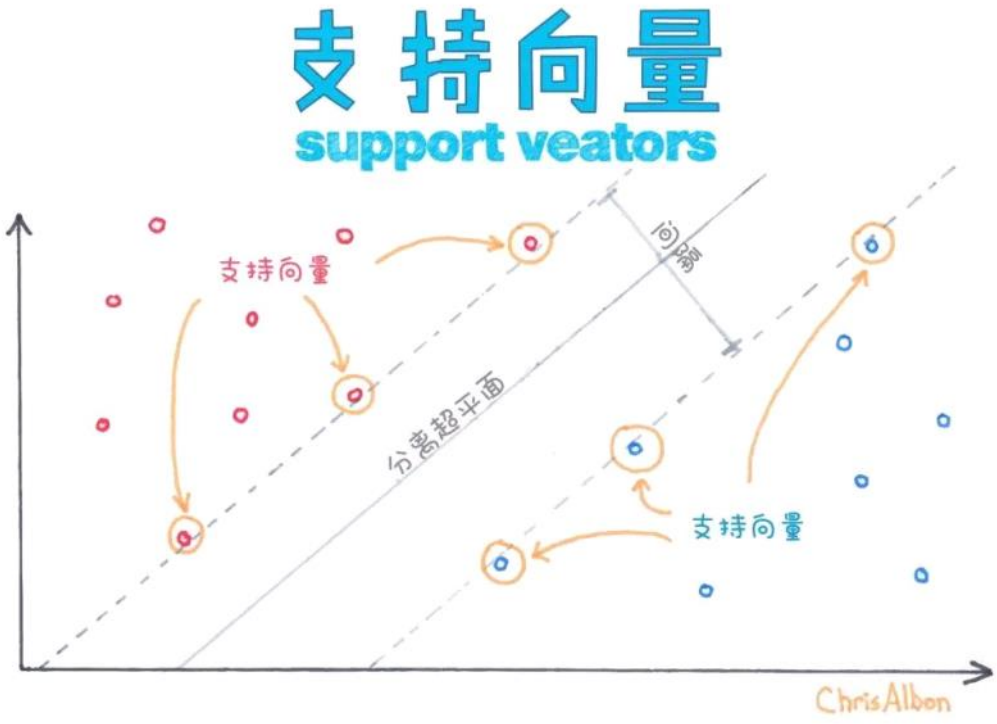
3. SVM

a. 概述

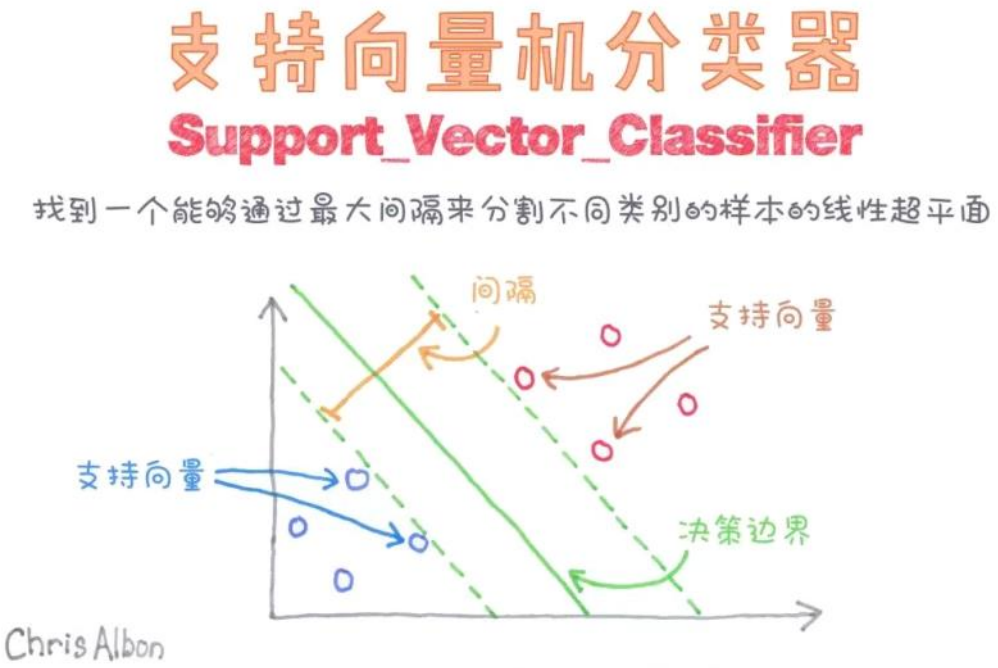
SVM 算法是建立在统计学习理论基础上的机器学习方法，为十大数据挖掘算法之一。通过学习算法，SVM 可以自动寻找出对分类有较好区分能力的支持向量，由此构造出的分类器可以最大化类与类的间隔，因而有较好的适应能力和较高的分准率。SVM 算法的目的在于寻找一个超平面H，该超平面可以将训练集中的数据分开，且与类域边界的沿垂直于该超平面方向的距离最大，故SVM 法亦被称为最大边缘算法。

b. 详解

支持向量机既可用于回归也可用于分类。它基于定义决策边界的决策平面。决策平面（超平面）可将一组属于不同类的对象分离开。



在支持向量的帮助下，SVM通过寻找超平面进行分类，并使两个类之间的边界距离最大化。



SVM中超平面的学习是通过将问题转化为使用一些某种线性代数转换问题来完成的。（上图的例子是一个线性核，它在每个变量之间具有线性可分性）。

对于高维数据，使用可使用其他核函数，但高维数据不容易进行分类。具体方法将在下一节中阐述。

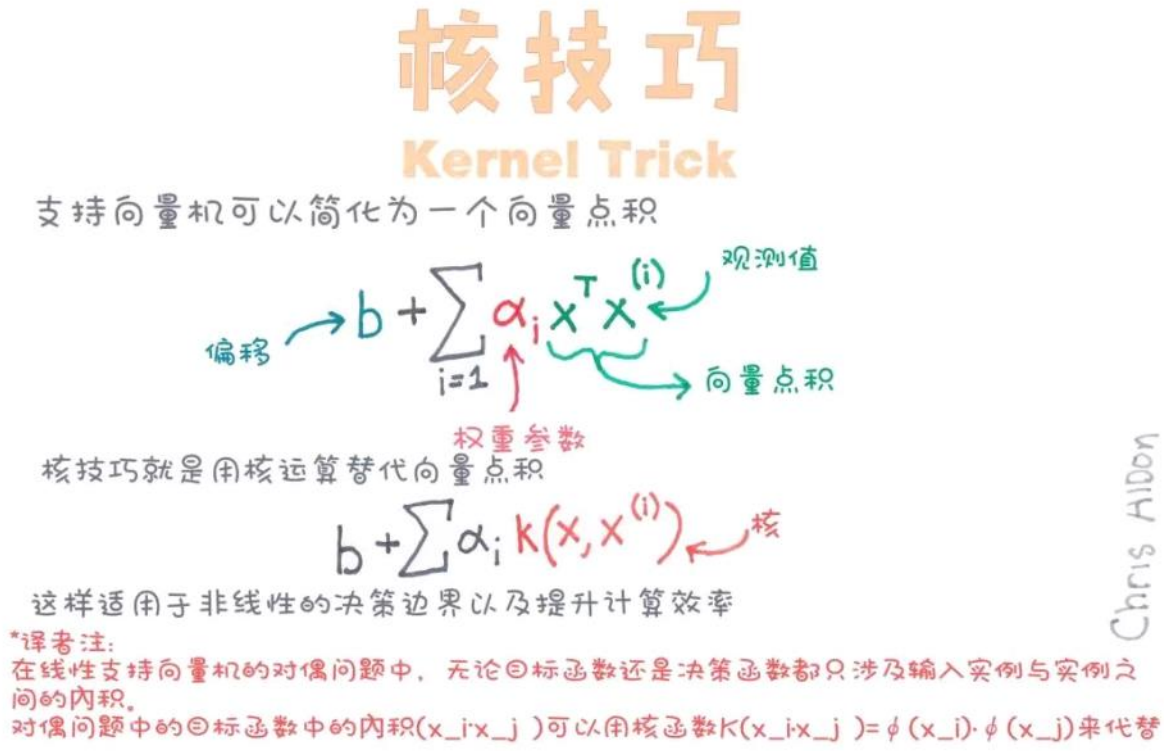
核支持向量机

核支持向量机将核函数引入到SVM算法中，并将其转换为所需的形式，将数据映射到可分的高维空间。

核函数的类型包括：

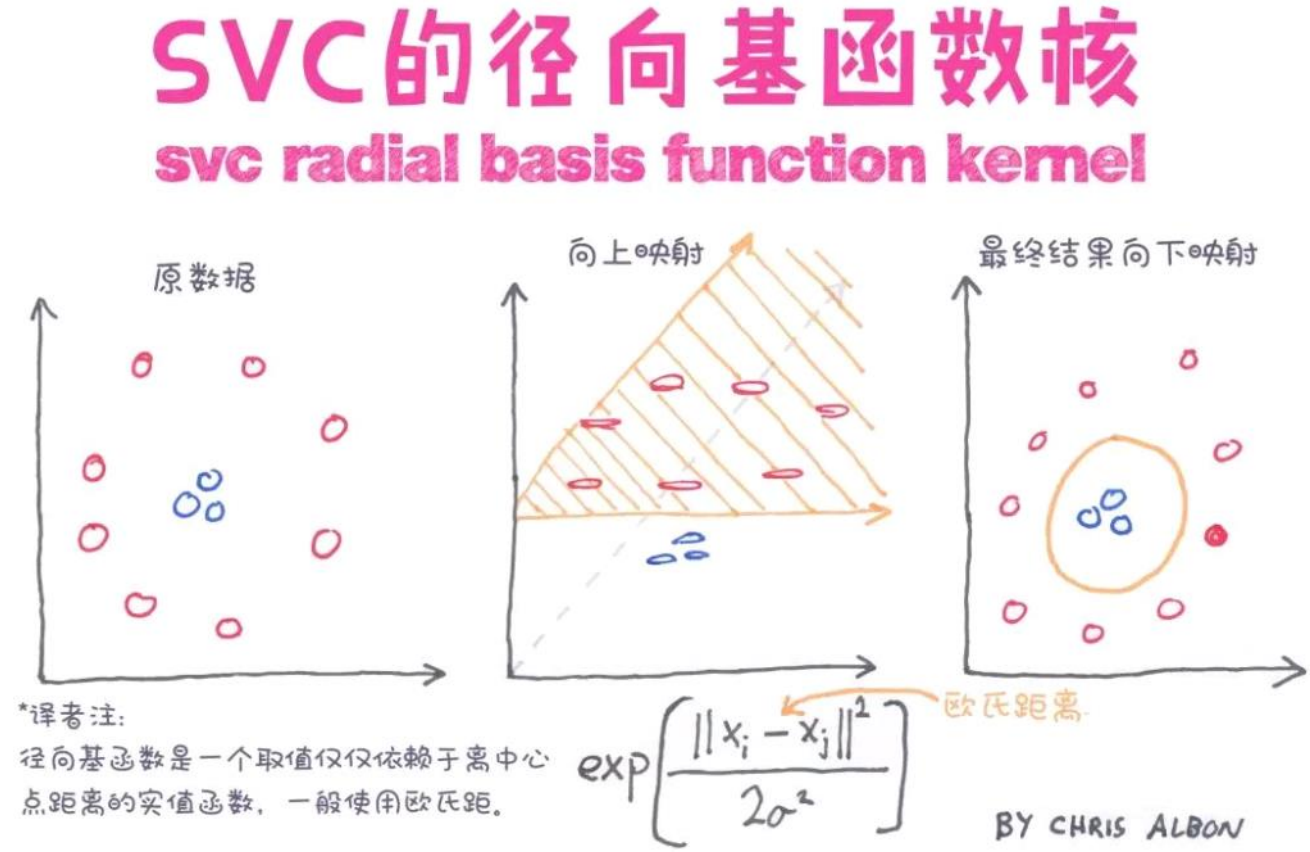
$$K(\mathbf{X}_i, \mathbf{X}_j) = \begin{cases} \mathbf{X}_i \cdot \mathbf{X}_j & \text{Linear} \\ (\gamma \mathbf{X}_i \cdot \mathbf{X}_j + C)^d & \text{Polynomial} \\ \exp(-\gamma |\mathbf{X}_i - \mathbf{X}_j|^2) & \text{RBF} \\ \tanh(\gamma \mathbf{X}_i \cdot \mathbf{X}_j + C) & \text{Sigmoid} \end{cases}$$

- 前文讨论的就是线性SVM。
- 多项式核中需要指定多项式的次数。它允许在输入空间中使用曲线进行分割。
- 径向基核（radial basis function, RBF）可用于非线性可分变量。使用平方欧几里德距离，参数的典型值会导致过度拟合。sklearn中默认使用RBF。
- 类似于与逻辑回归类似，sigmoid核用于二分类问题。



径向基核（RBF：Radial Basis Function ）

RBF核支持向量机的决策区域实际上也是一个线性决策区域。RBF核支持向量机的实际作用是构造特征的非线性组合，将样本映射到高维特征空间，再利用线性决策边界分离类。



因此，可以得出经验是：对线性问题使用线性支持向量机，对非线性问题使用非线性

核函数，如RBF核函数。

c. SVM 算法的优点

- SVM 模型有很高的分准率
- SVM 模型有很高的泛化性能
- SVM 模型能很好地解决高维问题
- SVM 模型对小样本情况下的机器学习问题效果好

d. SVM 算法的缺点

- SVM 模型对缺失数据敏感
- 对非线性问题没有通用解决方案，得谨慎选择核函数来处理

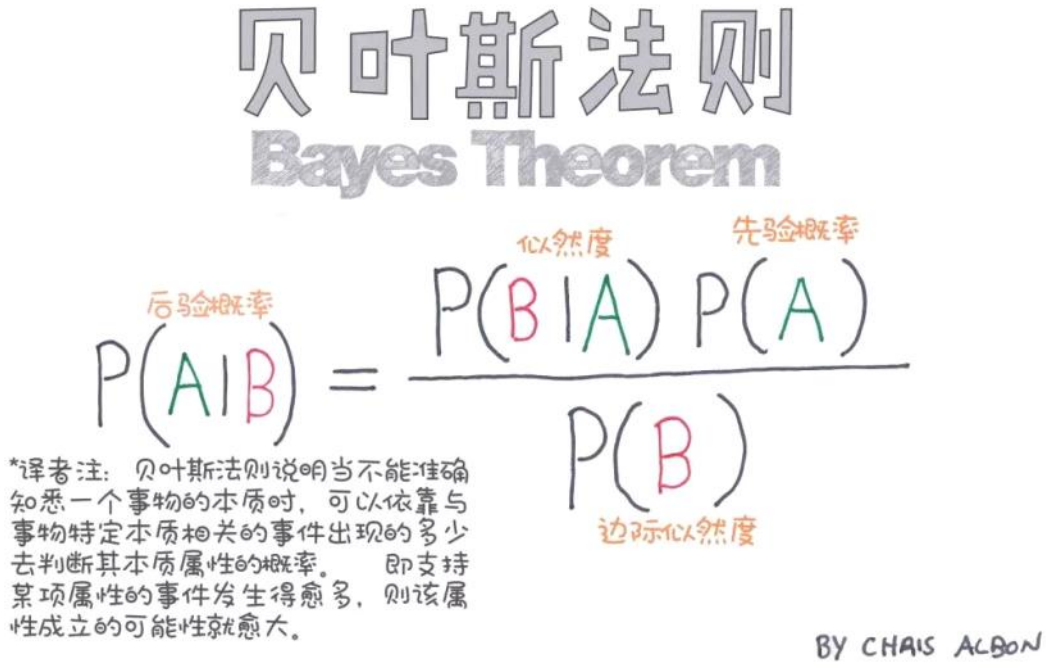
4. NBC

a. 概述

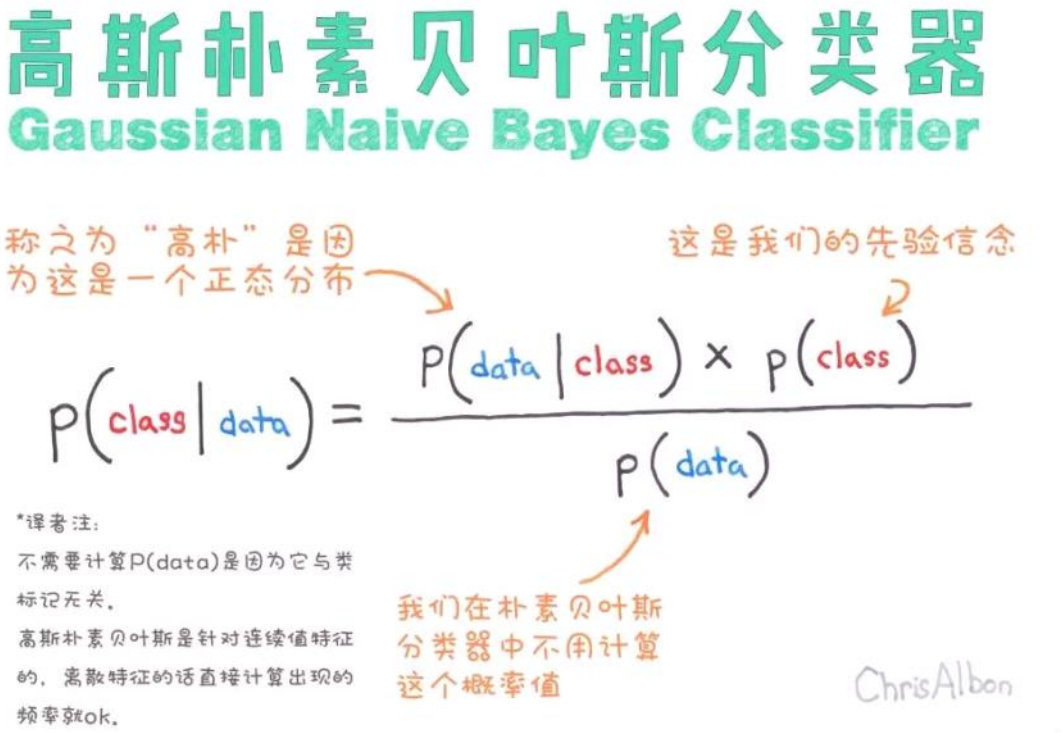
NBC 模型发源于古典数学理论，有着坚实的数学基础。该算法是基于条件独立性假设的一种算法，当条件独立性假设成立时，利用贝叶斯公式计算出其后验概率，即该对象属于某一类的概率，选择具有最大后验概率的类作为该对象所属的类。

b. 详解

朴素贝叶斯分类器建立在贝叶斯定理的基础上，基于特征之间互相独立的假设（假定类中存在一个与任何其他特征无关的特征）。即使这些特征相互依赖，或者依赖于其他特征的存在，朴素贝叶斯算法都认为这些特征都是独立的。这样的假设过于理想，朴素贝叶斯因此而得名。



在朴素贝叶斯的基础上，高斯朴素贝叶斯根据二项（正态）分布对数据进行分类。

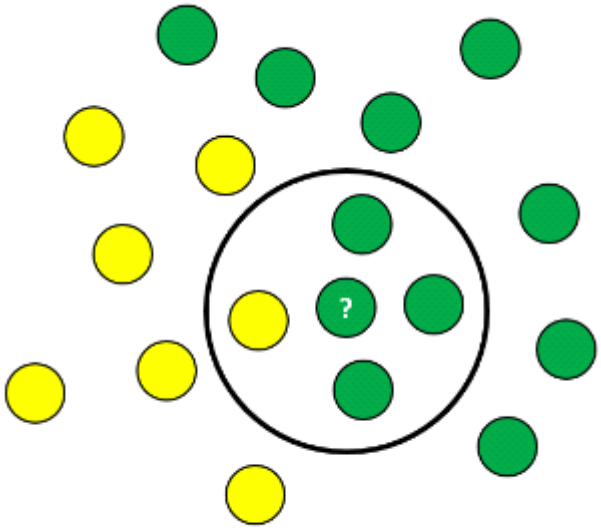


P(class|data) 表示给定特征（属性）后数据属于某类（目标）的后验概率。给定数据，其属于各类的概率大小就是我们要计算的值。

P(class)表示某类的先验概率。

P(data|class)表示似然，是指定类别时特征出现的概率。

P(data)表示特征或边际似然的先验概率。



NB Classification Example

步骤

1、计算先验概率P(class) = 类中数据点的数量/观测值的总数量P(yellow) = 10/17P(green) = 7/17

2、计算边际似然P(data) = 与观测值相似的数据点的数量/观测值的总数量P(?) = 4/17该值用于检查各个概率。

3、计算似然P(data/class) = 类中与观测值相似的数量/类中点的总数量P(?/yellow) = 1/7P(?/green) = 3/10

4、计算各类的后验概率

$$p(class/data) = \frac{P(data/class) * P(class)}{P(data)}$$

$$P(yellow/?) = \frac{1/7 * 7/17}{4/17} = 0.25$$

$$P(green/?) = \frac{3/10 * 10/17}{4/17} = 0.75$$

5、分类

$$P(class1/data) > P(class2/data)$$

$$P(green/?) > P(yellow/?)$$

某一点归于后验概率高的类别，因为从上可知其属于绿色类的概率是75%根据其75%的概率这个点属于绿色类。

多项式、伯努利朴素贝叶斯是计算概率的其他模型。朴素贝叶斯模型易于构建，不需要复杂的参数迭代估计，这使得它对非常大的数据集特别有用。

c. NBC算法的优点

- NBC算法逻辑简单，易于实现
- NBC算法所需估计的参数很少
- NBC 算法对缺失数据不太敏感
- NBC 算法具有较小的误差分类率
- NBC 算法性能稳定，健壮性比较好

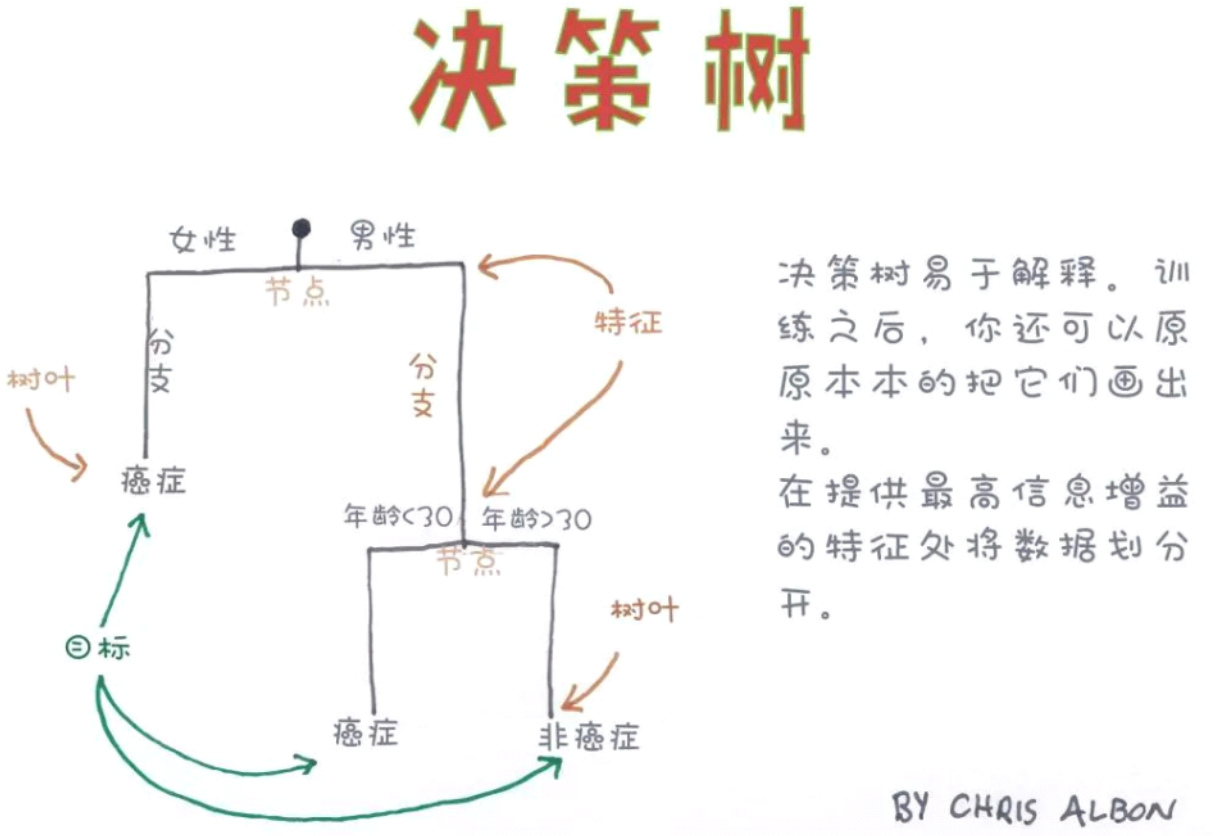
d. NBC算法的缺点

- 在属性个数比较多或者属性之间相关性较大时，NBC 模型的分类效果相对较差
- 算法是基于条件独立性假设的，在实际应用中很难成立，故会影响分类效果

5. Decision Tree

决策树以树状结构构建分类或回归模型。

它通过将数据集不断拆分为更小的子集来使决策树不断生长。最终长成具有决策节点（包括根节点和内部节点）和叶节点的树。最初决策树算法它采用采用Iterative Dichotomiser 3（ID3）算法来确定分裂节点的顺序。



信息熵和信息增益用于被用来构建决策树。

信息熵

信息熵是衡量元素无序状态程度的一个指标，即衡量信息的不纯度。

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

Entropy

信息熵是衡量元素的无序状态的程度的一个指标，或者说，衡量信息的不纯度。

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

Entropy

直观上说地理解，信息熵表示一个事件的确定性程度。信息熵度量样本的同一性，如

果样本全部属于同一类，则信息熵为0；如果样本等分成不同的类别，则信息熵为1。

信息增益

信息增益测量独立属性间信息熵的变化。它试图估计每个属性本身包含的信息，构造决策树就是要找到具有最高信息增益的属性（即纯度最高的分支）。

$$Gain(T,X) = Entropy(T) - Entropy(T,X)$$

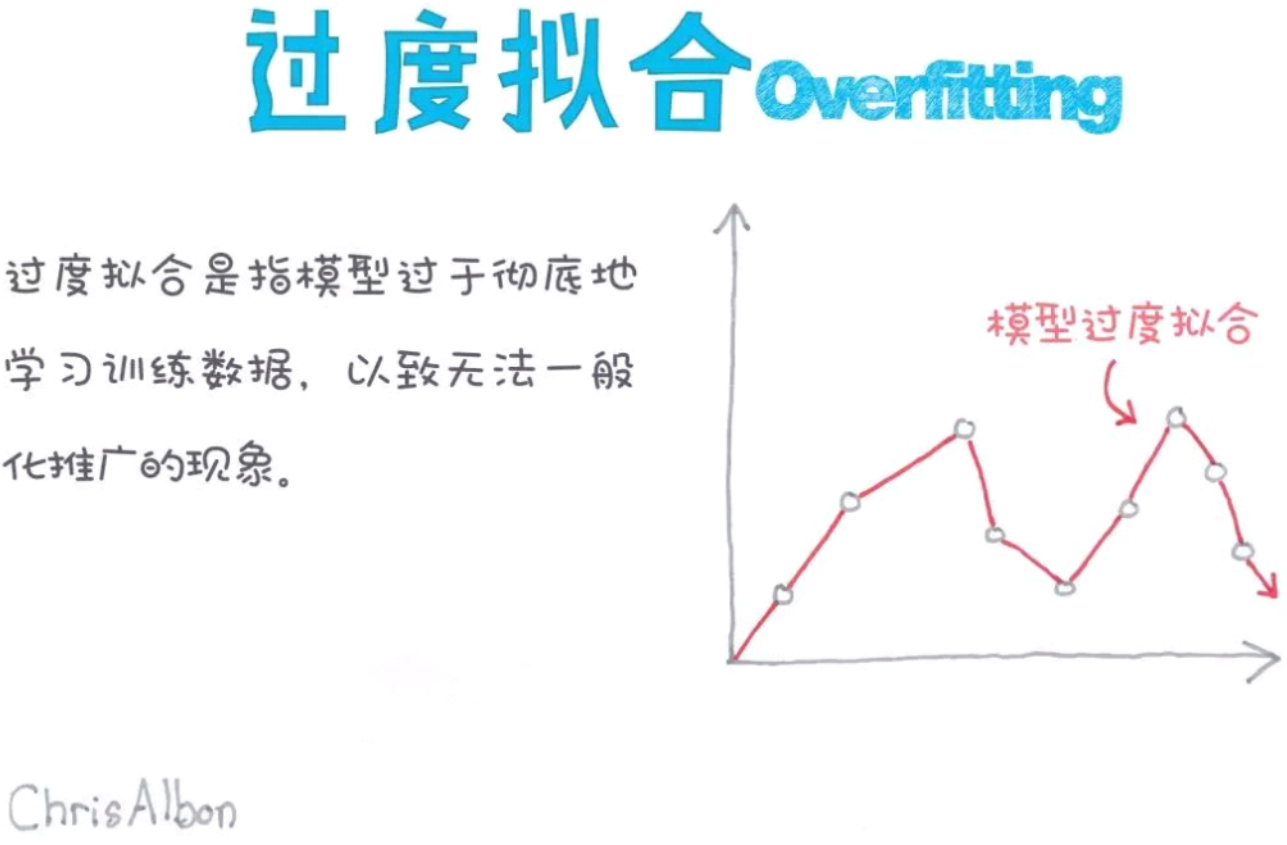
信息增益测量独立属性间的信息熵的变化。它试图估计每个属性本身包含的信息，构造决策树就是要找到具有最高信息增益的属性（即纯度最高的分支）。

$$Gain(T,X) = Entropy(T) - Entropy(T,X)$$

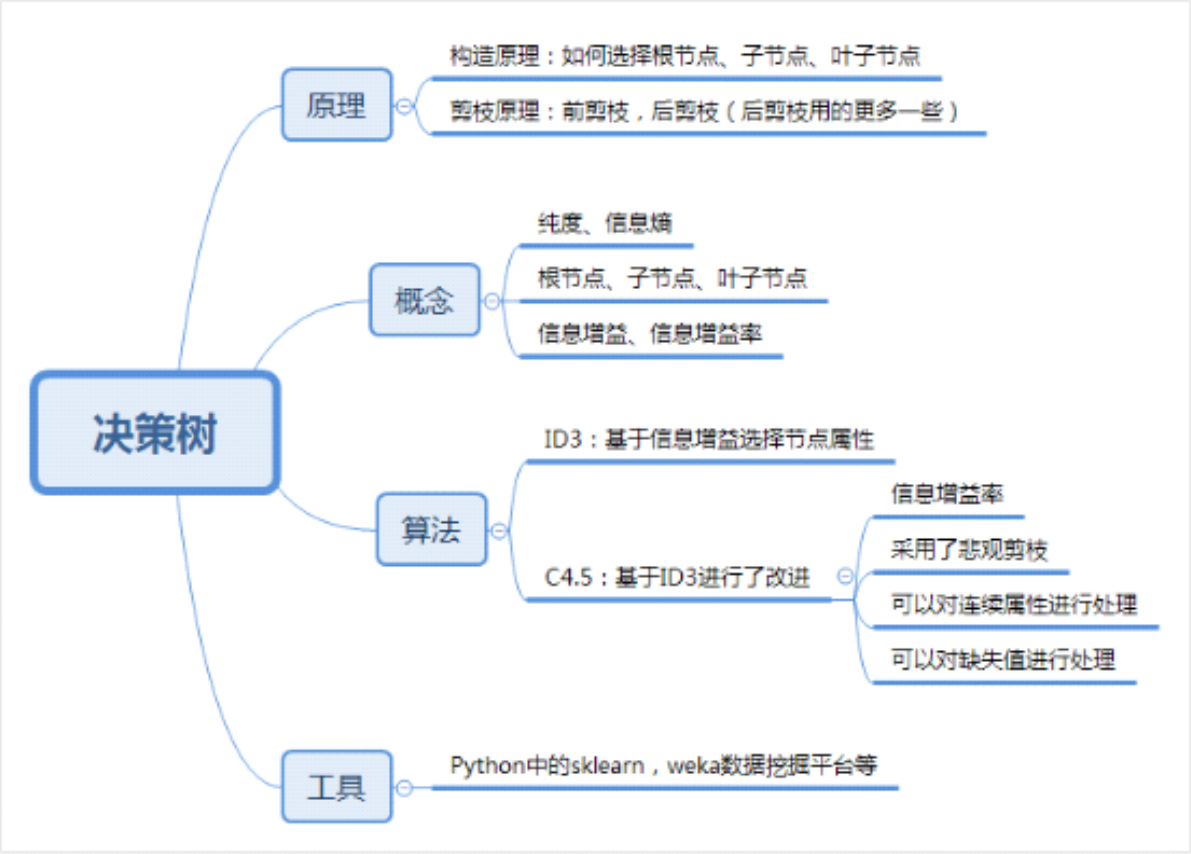
其中Gain（(T,X）)是特征X的信息增益。Entropy(T)是整个集合的信息熵，第二项Entropy(T,X)是特征X的信息熵。

采用信息熵进行节点选择时，通过对该节点各个属性信息增益进行排序，选择具有最高信息增益的属性作为划分节点，过滤掉其他属性。

决策树模型存在的一个问题是容易过拟合。因为在其决策树构建过程中试图通过生成长一棵完整的树来拟合训练集，因此却降低了测试集的准确性。



▲通过剪枝技术可以减少小决策树的过拟合问题。



https://www.cnblogs.com/molieren/articles/10664954.html

ID3

a. 概述

ID3 算法是一种基于决策树的分类算法，该算法是以信息论为基础，以信息熵和信息增益为衡量标准，从而实现对数据的归纳分类。信息增益用于度量某个属性对样本集合分类的好坏程度。ID3 算法的时间复杂度为O(n*|D|*log|D|)。

b. ID3算法的优点

- ID3 算法建立的决策树规模比较小
- 查询速度快

c. ID3算法的缺点

- 不适合处理连续数据
- 难以处理海量数据集
- 建树时偏选属性值较大的进行分离，而有时属性值较大的不一定能反应更多的数据信息

C4.5

a. 概述

C4.5 算法是ID3 算法的修订版，采用信息增益率来加以改进，选取有最大增益率的分割变量作为准则，避免ID3 算法过度的适配问题。

b. C4.5算法优点

- C4.5 继承了ID3 优点
- 在树构造过程中进行剪枝
- 能对不完整数据进行处理
- 能够完成对连续属性的离散化处理
- 产生的分类规则易于理解，准确率较高\
- 用增益率来选择属性，克服了用增益选择属性时偏向选择取值多的属性

c. C4.5 算法缺点

- 构造树时，需要对数据集进行多次的顺序扫描和排序，因而导致算法的低效
- 只适合于能驻留于内存的数据集，当训练集达到内存无法容纳时程序无法运行

d. 实例

C4.5 用于遥感分类过程中，首先依据通常的方式建立第一个模型。随后建立的第二个模型聚焦于被第一个模型错误分类的记录。以此类推，最后应用整个模型集对样本进行分类，使用加权投票过程把分散的预测合并成综合预测。Boosting 技术对于噪声不大的数据，通常通过建立的多模型来减少错误分类的影响，提高分类精度。

C5.0

a. 概述

C5.0 算法是 Quinlan 在C4.5 算法的基础上改进而来的产生决策树的一种更新的算法，它除了包括C4.5 的全部功能外，还引入许多新的技术，其中最重要的技术是提升（Boosting）技术，目的是为了进一步提高决策树对样本的识别率。同时C5.0 的算法复杂度要更低，使用更简单，适应性更强，因此具有更高的使用价值。

b. C5.0算法的优点

- C5.0 模型能同时处理连续和离散的数据
- C5.0 模型通常不需要很长的训练时间
- C5.0 引入Boosting 技术以提高分类的效率和精度
- C5.0 模型易于理解，模型推出的规则有非常直观的解释
- C5.0 模型在面对数据遗漏和特征很多的问题时非常稳健

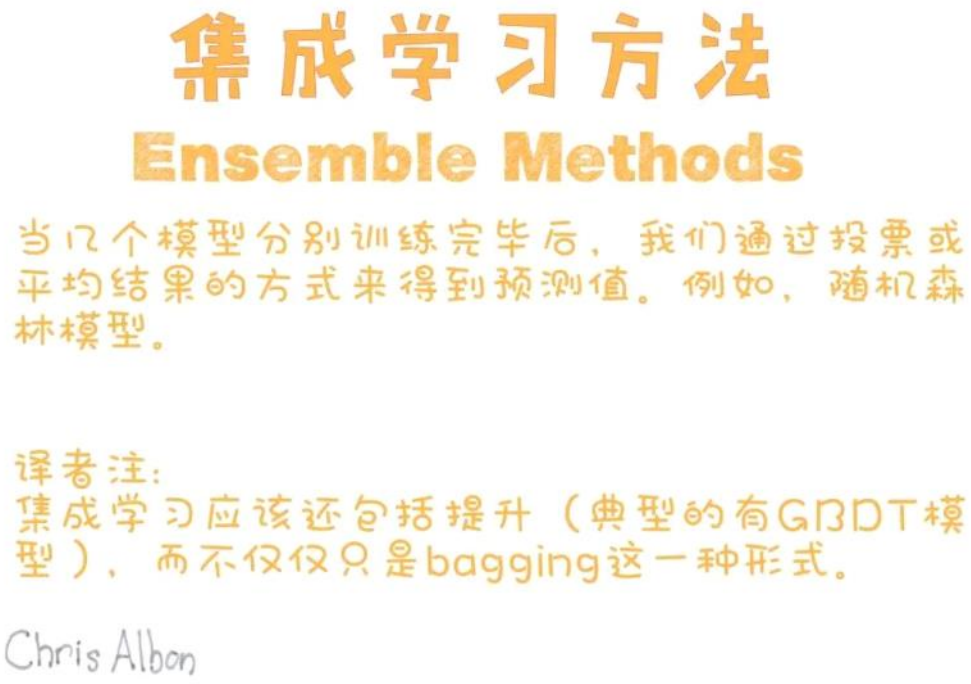
c. C5.0算法的缺点

- 目标字段必须为分类字段

d. 实例

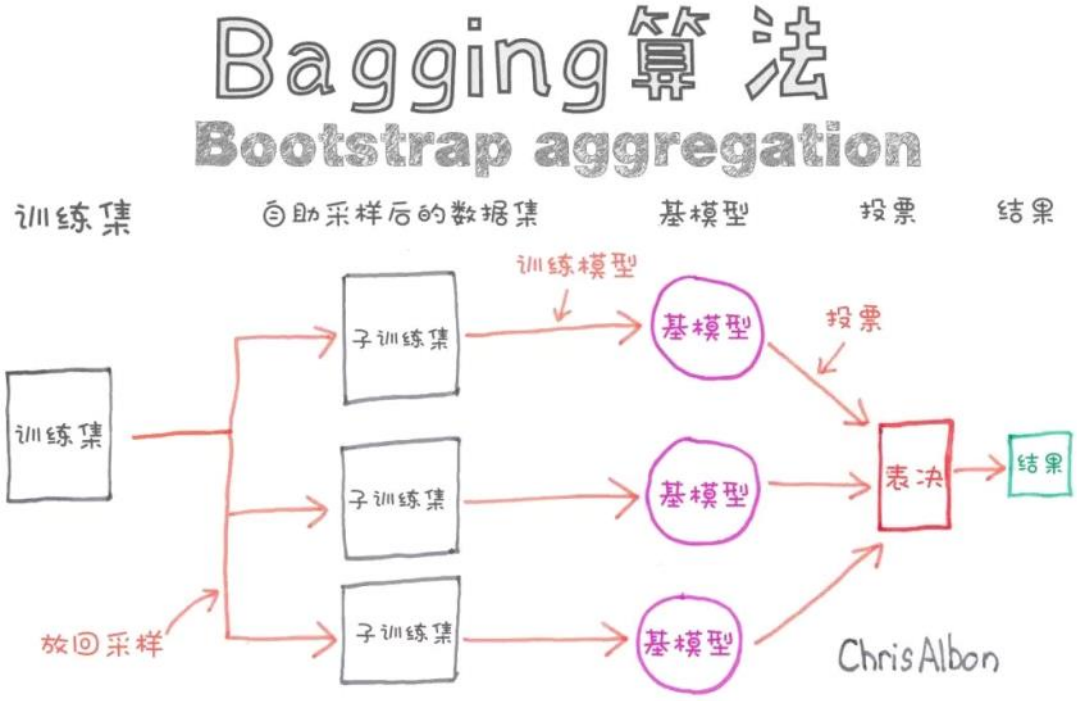
美国地质调查局(USGS)在进行土地覆盖分类项目过程中研发了支持决策树分类的软件。软件分类模块主要是针对庞大数据量的数据集进行数据挖掘，找出特征，然后建立规则集进行决策分类。在分类模块中采用C5.0 模型来完成决策树分类、形成分类文件，实现遥感影像的分类。

集成算法是一个模型组。从技术上说，集成算法是单独训练几个有监督模型，并将训练好的模型以不同的方式进行融合，从而达到最终的得预测结果。集成后的模型比其中任何一个单独的模型都有更高的预测能力。

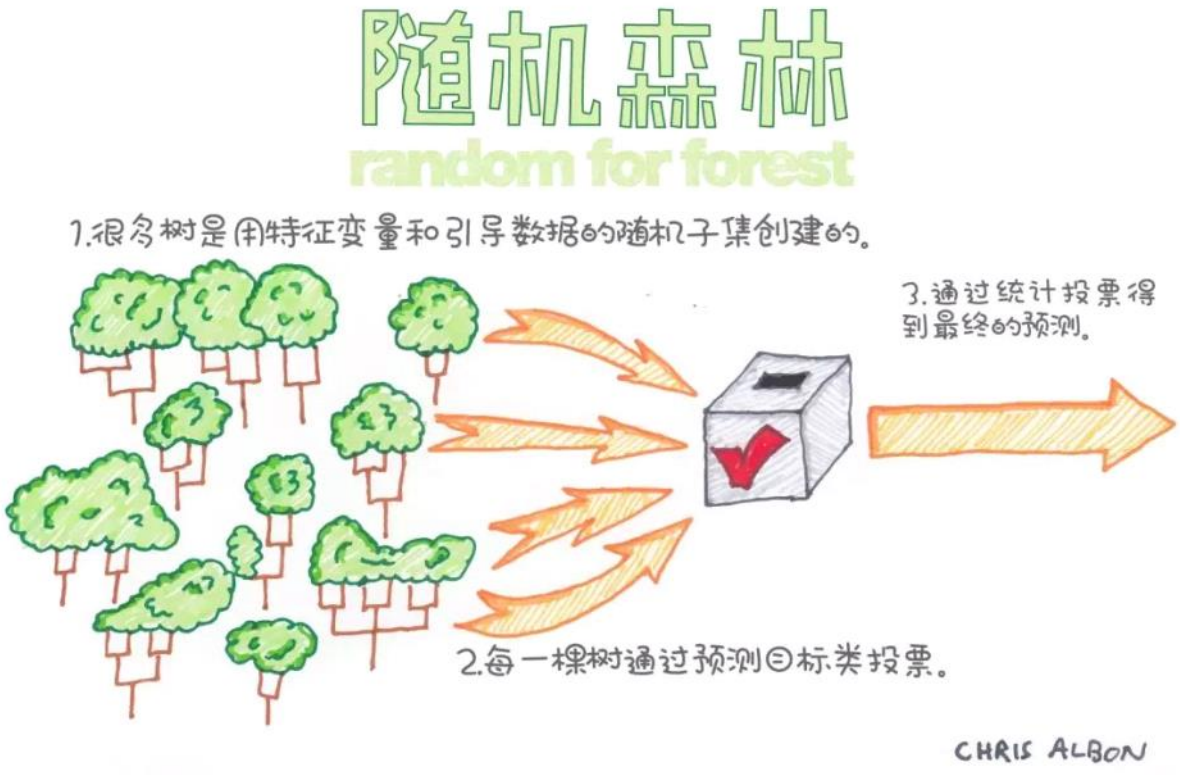


6. 随机森林

随机森林分类器是一种基于装袋（bagging）的集成算法，即自举助聚合法(bootstrap aggregation)。集成算法结合了多个相同或不同类型的算法来对对象进行分类（例如，SVM的集成，基于朴素贝叶斯的集成或基于决策树的集成）。



集成的基本思想是算法的组合提升了最终的结果。

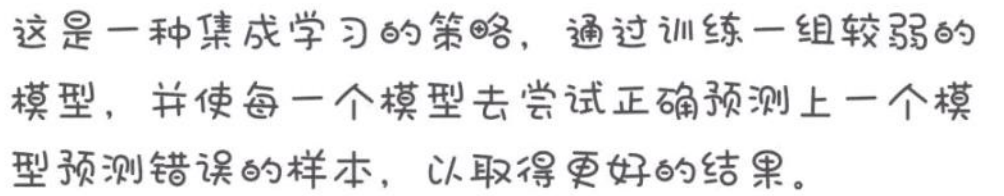


深度太大的决策树容易受过拟合的影响。但是随机森林通过在随机子集上构建决策树防止过拟合，主要原因是它会对所有树的结果进行投票的结果是所有树的分类结果的投票，从而消除了单棵树的偏差。

随机森林在决策树生增长的同时为模型增加了额外的随机性。它在分割节点时，不是搜索全部样本最重要的特征，而是在随机特征子集中搜索最佳特征。这种方式使得决策树具有多样性，从而能够得到更好的模型。

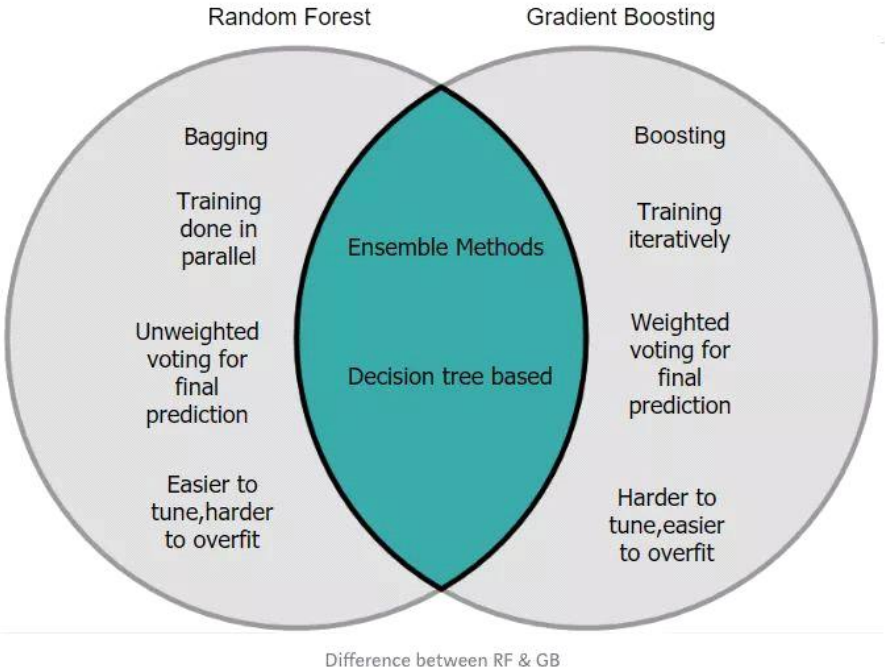
7. 梯度上升

梯度提升分类器是一种提升集成算法。提升(boosting)算法是为了减少偏差而对弱分类器的而进行的一种集成方法。与装袋（bagging）方法构建预测结果池不同，提升算法是一种分类器的串行方法，它把每个输出作为下一个分类器的输入。通常，在装袋算法中，每棵树在原始数据集的子集上并行训练，并用所有树预测结果的均值作为模型最终的预测结果；梯度提升模型，采用串行方式而非并行模式获得预测结果。每棵决策树预测前一棵决策树的误差，因而使误差获得提升。



梯度提升树的工作流程：

- 使用浅层决策树初始化预测结果。
- 计算残差值（实际预测值）。
- 构建另一棵浅层决策树，将上一棵树的残差作为输入进行预测。
- 用新预测值和学习率的乘积作为最新预测结果，更新原有预测结果。
- 重复步骤2-4，进行一定次数的迭代（迭代的次数即为构建的决策树的个数）。



Difference between RF & GB

8. ANN

人工神经网络（ANN）算法就是一组连续的输入/输出单元，其中每个连接都与一个权相关。在学习阶段，通过调整神经网络的权，使得能够预测样本的正确类标号来学习。

- ANN算法的优点
  - 能处理数值型及分类型的属性
  - 分类的准确度高，分布并行处理能力强
  - 对包含大量噪声数据的数据集有较强的鲁棒性和容错能力
- ANN算法的缺点
  - 不能观察之间的学习过程
  - 学习时间过长，甚至可能达不到学习的目的
  - 对于非数值型数据需要做大量数据预处理工作
  - 输出结果难以解释，会影响到结果的可信度和可接受程度
  - 神经网络需要大量的参数，如网络拓扑结构、权值和阈值的初始值

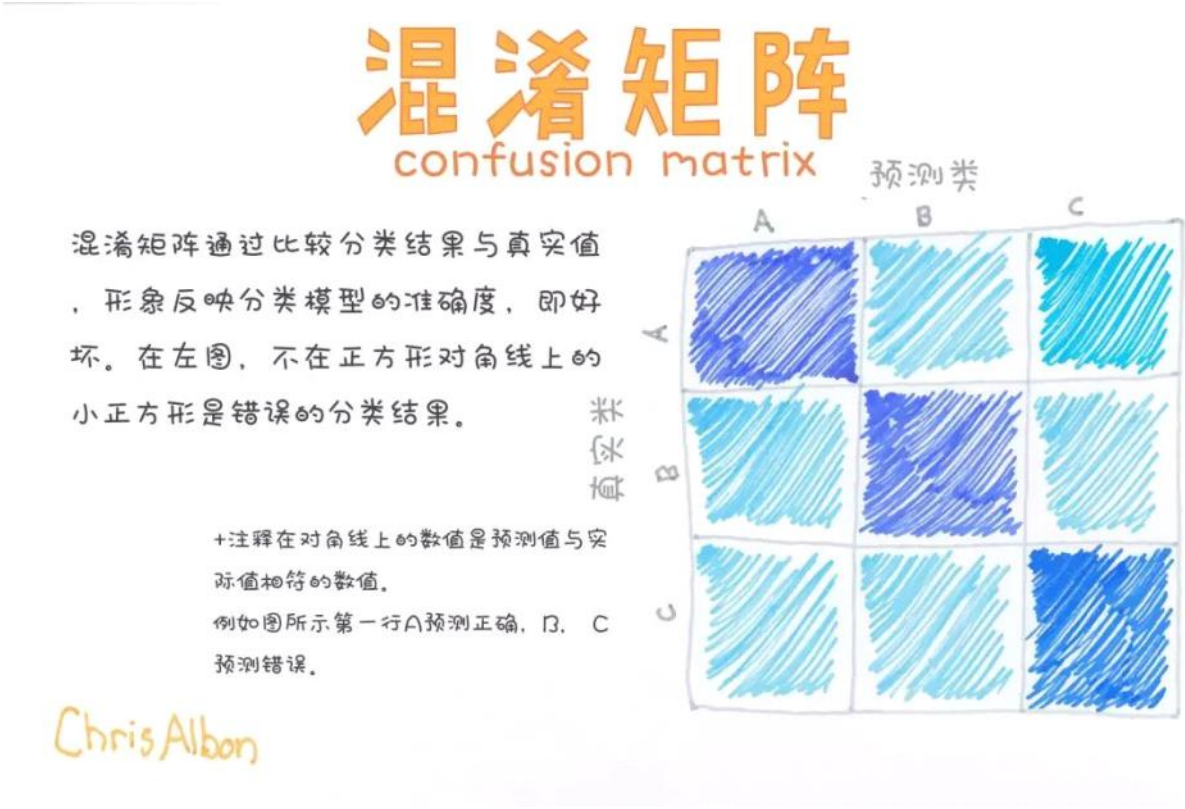
小结

算法名称	收敛时间	是否过度拟合	是否过渡拟合缺失数据敏感度	训练数据量
NBC	快	存在	不敏感	无要求
LR	快	存在	敏感	无要求
SVM	一般	存在	敏感	小数据量
ID3	快	存在	不敏感	小数据集
C4.5	快	存在	不敏感	小数据集
C5.0	快	不存在	不敏感	大数据集
ANN	慢	存在	敏感	大数据集
KNN	快	存在	敏感	数据量多

分类器性能评价

- **混淆矩阵**

  混淆矩阵是一张表，这张表通过对比已知分类结果的测试数据的预测值和真实值表来描述衡量分类器的性能。在二分类的情况下，混淆矩阵是展示预测值和真实值四种不同结果组合的表。



多分类问题的混淆矩阵可以帮助你确认错误模式。

对于二元分类器：

		Predicted Class	
		No	Yes
Observed Class	No	TN	FP
	Yes	FN	TP

TN	True Negative
FP	False Positive
FN	False Negative
TP	True Positive

Model Performance

Accuracy	= (TN+TP)/(TN+FP+FN+T
Precision	= TP/(FP+TP)
Sensitivity	= TP/(TP+FN)
Specificity	= TN/(TN+FP)

Binary Confusion Matrix

- **假正例&假负例**

  假正例和假负例用来衡量模型预测的分类效果。假正例是指模型错误地将负例预测为正例。假负例是指模型错误地将正例预测为负例。主对角线的值越大（主对角线为真正例和真负例），模型就越好；副对角线给出模型的最差预测结果。

- **假正例**

  下面给出一个假正例的例子。比如：模型将一封邮件分类为垃圾邮件（正例），但这封邮件实际并不是垃圾邮件。这就像一个警示，错误如果能被修正就更好，但是与假负例相比，它并不是一个严重的问题。

作者注：个人观点，这个例子举的不太好，对垃圾邮件来说，相比于错误地将垃圾邮件分类为正常邮件（假负例），将正常邮件错误地分类为垃圾邮件（假正例）是更严重的问题。

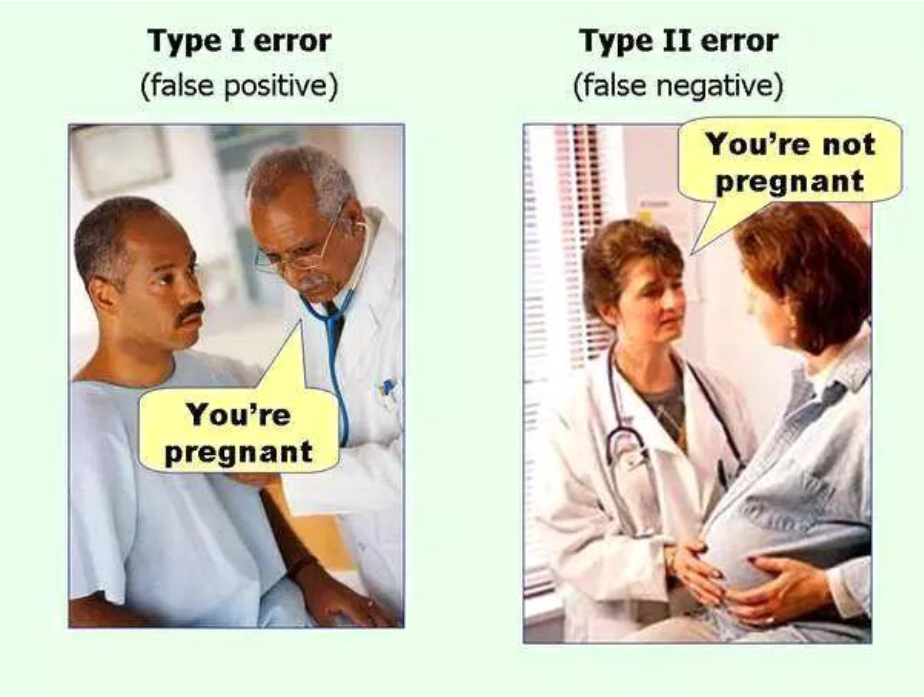
假正例（I型错误）——原假设正确而拒绝原假设。

$$FPR = \frac{False\ Positives}{False\ Positives + True\ Negatives}$$

- **假负例**

  假负例的一个例子。例如，该模型预测一封邮件不是垃圾邮件（负例），但实际上这封邮件是垃圾邮件。这就像一个危险的信号，错误应该被及早纠正，因为它比假正例更严重。

  假负例（II型错误）——原假设错误而接受原假设



上图能够很容易地说明上述指标。左图男士的测试结果是假正例因为男性不能怀孕；右图女士是假负例因为很明显她怀孕了。
从混淆矩阵，我们能计算出准确率、精度、召回率和F-1值。

- **准确率**

准确率是模型预测正确的部分。

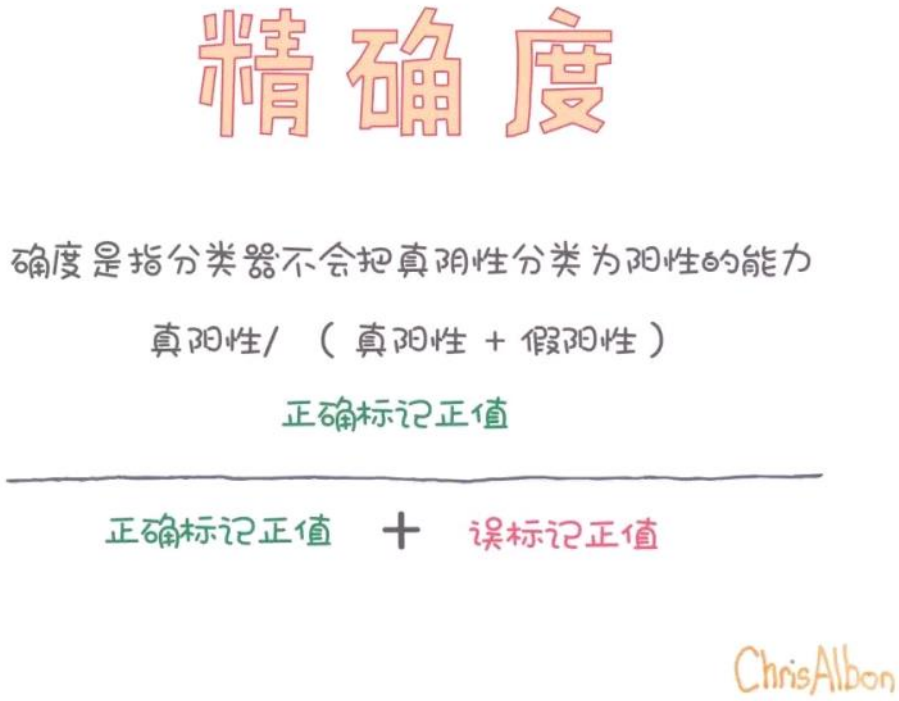
准确率的公式为：

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

当数据集不平衡，也就是正样本和负样本的数量存在显著差异时，单独依靠准确率不能评价模型的性能。精度和召回率是衡量不平衡数据集的更好的指标。

- **精度**

精度是指在所有预测为正例的分类中，预测正确的程度为正例的效果。



▲精度越高越好。

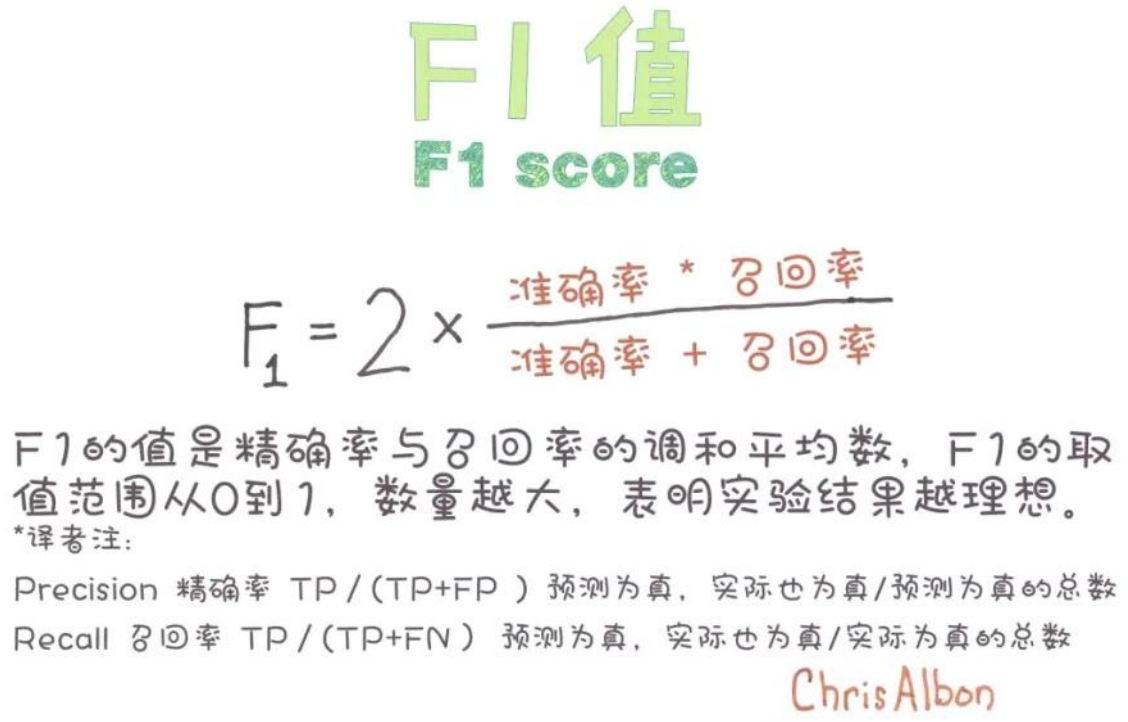
- **召回率**

召回率是指在所有预测为正例（被正确预测为真的和没被正确预测但为真的）的分类样本中，召回率是指预测正确的程度。它，也被称为敏感度或真正率（TPR）。

▲召回率越高越好。

- **F-1值**

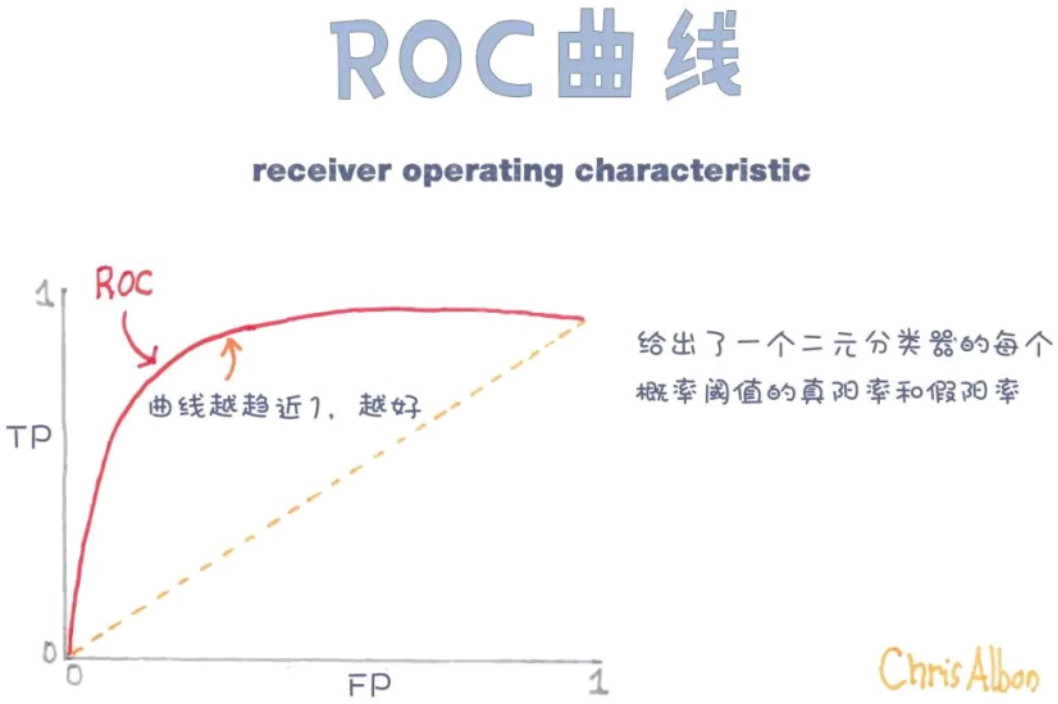
通常实用的做法是将精度和召回率合成一个指标F-1值更好用，特别是当你需要一种简单的方法来衡量两个分类器性能时。F-1值是精度和召回率的调和平均值。



普通的通常均值将所有的值平等对待，而调和平均值给予较低的值更高的权重，从而能够更多地惩罚极端值。所以，如果精度和召回率都很高，则分类器将得到很高的F-1值。

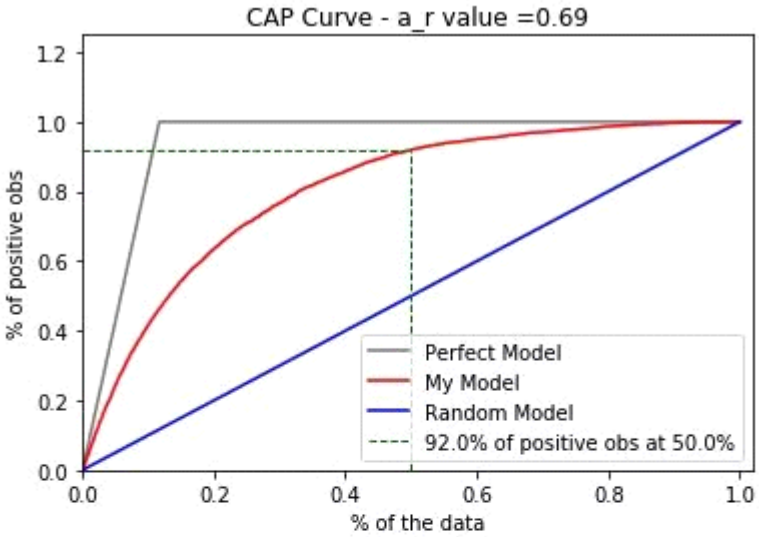
- **接受者操作曲线（ROC）和曲线下的面积（AUC）**

  ROC曲线是衡量分类器性能的一个很重要指标，它代表模型准确预测的程度。ROC曲线通过绘制真正率和假正率的关系来衡量分类器的敏感度。如果分类器性能优越，则真正率将增加，曲线下的面积会接近于1.如果分类器类似于随机猜测，真正率将随假正率线性增加。AUC值越大，模型效果越好。



- **累积精度曲线**

  CAP代表一个模型沿y轴为真正率的累积百分比与沿x轴的该分类样本累积百分比。CAP不同于接受者操作曲线（ROC，绘制的是真正率与假正率的关系）。与ROC曲线相比，CAP曲线很少使用。



CAP Curve

$$CAPCurve(x,y) = (\frac{(TP+FP)}{n}, \frac{TP}{(TP+FN)})$$

$$Where, n = TP + FN + FP + TN$$

以考虑一个预测客户是否会购买产品的模型为例，如果随机选择客户，他有50%的概率会购买产品。客户购买产品的累积数量会线性地增长到对应客户总量的最大值，这个曲线称为CAP随机曲线，为上图中的蓝色线。而一个完美的预测，准确地确定预测了哪些客户会购买产品，这样，在所有样本中只需选择最少的客户就能达到最大购买量。这在CAP曲线上产生了一条开始陡峭一旦达到最大值就会维持在1的折线，称为CAP的完美曲线，也被称为理想曲线，为上图中灰色的线。

最后，一个真实的模型应该能尽可能最大化地正确预测，接近于理想模型曲线。