1. **决策树**

假设是一个离散型随机变量，其概率分布函数为，则的信息量为：概率对数的负数，概率越小，信息量越大



信息熵：对于某个事件，有n种可能性，每一种可能性的概率为，则信息熵为所有信息量的期望：



联合熵：如果随机变量有多个，则信息熵就会变化成联合熵：



条件概率：



<https://www.cnblogs.com/fantasy01/p/4581803.html>

条件熵：为已知随机变量的条件下，随机变量的不确定性：



信息增益：待分类的集合的熵-按某个条件分类后的条件熵



在决策树算法中，分类效果越好，就会越小，相应的信息增益就会越大。

<https://blog.csdn.net/baimafujinji/article/details/50554664>

<https://www.cnblogs.com/maybe2030/p/4585705.html>

<https://www.cnblogs.com/fionacai/p/5894142.html>

Bootstrap抽样：“有放回地全抽”，（其实样本量也要视情况而定，不一定非要与原样本量相等），抽取的Bootstrap样本量与原样本相同，只是在抽样方式上采取有放回地随机抽，这样对于一个原始数据集就可以抽样多次，每次抽样的数据集就很有可能有重复数据，相应的也会剔除部分数据。

随机森林：为了解决决策树过拟合的问题，如果数据集的特征非常多，决策树算法就容易陷入过拟合（特征很多，会根据不同特征将训练集分隔成多个方块，不具备泛化能力）。

1. 采用Bootstrap抽样对训练集进行抽样；
2. 从样本的M个特征维度中，随机选取m（m<<M）个特征，只根据这个m个特征建立决策树；
3. 对根据之前两个步骤建立的多个决策树进行投票分析，得票最多的就是最终预测类别。

决策树的一些超参数：

1. 最大深度（max\_depth）：当决策树深度为时，它最多可以拥有片叶子。较大的深度往往会导致过拟合，这是因为过深的决策树可以记忆数据。而较小的深度会使得模型过于简单，导致欠拟合。
2. 每片叶子的最小样本数（min\_samples\_leaf）：当分裂结点时，每片叶子允许的最小样本数。当每片叶子的样本数量较小时，叶子上的样本数量也有可能过于稀少，此时模型将记忆数据，也就是过拟合。当每片叶子的样本数量较大时，决策树能够获得足够的弹性进行构建，这也许会导致欠拟合。
3. 每次分裂的最小样本数（min\_samples\_split）：结点允许分裂最小样本数，即如果结点样本数>min\_samples\_split，则允许分裂，否则不允许。
4. 最大特征数（max\_features）：寻找最佳分裂时，考虑的最大特征数目。
5. **朴素贝叶斯分类：假设各特征之间相互独立。**

贝叶斯公式：



为先验概率：即没有条件下，事件A的概率；

为后验概率：条件B发生后，事件A的概率；

<https://www.cnblogs.com/marc01in/p/4775440.html>

<https://www.cnblogs.com/by-dream/p/7884606.html>

例：

一所学校里面有60%的男生，40%的女生。男生总是穿长裤，女生则一半穿长裤一半穿裙子。

P(男)=0.6 P(女)=0.4

P(男|长裤)=1 P(女|长裤)=0.5

根据贝叶斯公式求P(长裤|男)，即穿长裤的人是男生的概率。

P(长裤|男)=P(男)P(男|长裤)/P(长裤)

P(长裤)=P(男,长裤)+P(女,长裤)=P(男)P(男|长裤)+P(女)P(女|长裤)=0.8

则：

P(长裤|男)=0.6\*1/0.8=0.75

P(长裤|女)=0.4\*0.5/0.8=0.25

朴素贝叶斯分类方法：

1. 设为一个待分类项，其中为的特征属性；
2. 设类别集合为，表示总共有n类。
3. 求出，，…
4. 如果，则。

朴素贝叶斯分类的核心就是计算第3步的条件概率。根据贝叶斯公式：



对于所有类别而言，是一个常数，可以忽略掉。

根据贝叶斯假设，的各个特征之间相互独立（“朴素”两个字的来源），所以：



拉普拉斯平滑：上述的公式会存在一个问题，如果某个特征没有再某个分类中出现过，如特征没有在中类别出现过，则为0，从而导致整个公式结果为0，显示不能如此计算。拉普拉斯平滑就是在计算不为0，而是赋一个很小的概率。具体操作如下：

令为在垃圾邮件类别中，单词出现的概率；定义：

令为垃圾邮件中，单词出现的个数。

令为垃圾邮件，所有单词总数目。则：



拉普拉斯平滑：分子+1，分母加特征总数，这样可以保证概率和为1



Udacity中的例子，患病的概率：

人患某种病的概率

P(S)=0.00001

健康的概率

P(H)=0.9999

检测为阳性的准确率为99%（这个概率指的是如果人检测为阳性，准确率为99%，还有%1是本身为阴性，结果误检测为阳性），即：

患病时检测为阳性的概率

P(+|S)=0.99

健康时误检测为阳性时的概率

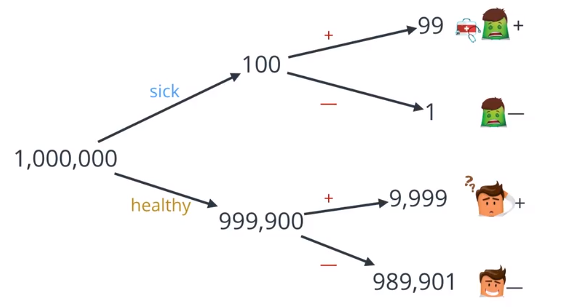
P(+|H)=0.01

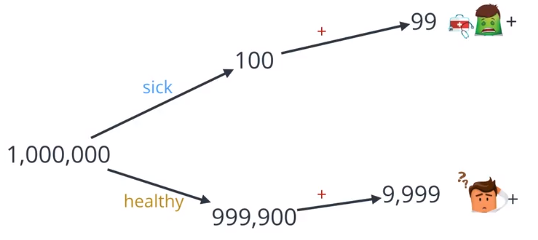
求P(S|+)

P(S|+)=P(S)P(+|S)/P(+)

P(+)=P(S)P(+|S)+P(H)P(+|H)=0.0001\*0.99+0.9999\*0.01

P(S|+)=0.0001\*0.99 / (0.0001\*0.99+0.9999\*0.01)=0.0098





从上述计算来看，即使检测准确率达到99%，检测为阳性并且人患病的准确率最终为0.0098，小于1%，而检测为阳性但人健康的概率（假阳性）却超过了99%，这是由于检测的误差1%远大于人患病的概率。

所以在医学上检测准确率必须非常高，如果准确率达到99.999%，则检测为阳性并且人患病的概率为：

P(S|+)=0.91

朴素贝叶斯算法是一种生成模型，分别计算样本属于某一个类别的概率，然后概率最大的就是样本的类别。

1. **支持向量机（SVM）**

在Logistic回归中：



当时，，预测y=1。

当时，，预测y=0。

如果，那么我们就能很准确的预测y=1，反之，如果，就能很准确的预测y=0.

通过Logistic回归学习得到的参数，如果使得y=1时，，y=0时，，那么可以认为模型是良好的。

如果数据集一定可以线性分隔，Logistic回归考虑全局，得到的分隔模型将正类和负类区别开，而SVM则希望靠近分隔线的元素能尽可能的远离分隔线，不关心那些已经远离分隔线的元素，注重局部。这也是SVM和Logistic的不同。

在SVM中，符号约定：







函数间隔：定义一个训练样本，是特征，是结果标签。



如果希望函数间隔取一个较大的值（这样根据之前的推断，可以更好的区分样本是正类，还是负类），当时，，当时，。因此，函数间隔代表了样本是正类还是负类的确信度。

全局样本的函数间隔：



所有训练样本中，最坏的情况。

几何间隔：训练样本到分隔面的垂直距离



其中表示向量的模。

当时，几何间隔就变成了函数间隔。函数间隔存在一个问题，我们可以通过将和乘以一个倍数来提高函数间隔，尽管这样对模型并无意义。而几何间隔则不存在这个问题，提高了和，则也会跟着一起提高。

全局几何间隔：



最优间隔分类器（optimal margin classifier）：

目标函数1：使得几何间隔最大



约束条件：

其中，保证了函数间隔=几何间隔。

上述的优化问题中，约束条件难以进行局部最优搜索，所以需要变换：

目标函数2：函数间隔除以向量的模最大



约束条件：

其中，

凸函数：

<https://www.cnblogs.com/always-fight/p/9377554.html>

目标函数不是凸函数，难以使用梯度下降等局部搜索算法求解，需要继续变换。

将全局函数间隔取为1，即。目标函数2变为：



演变为目标函数3：



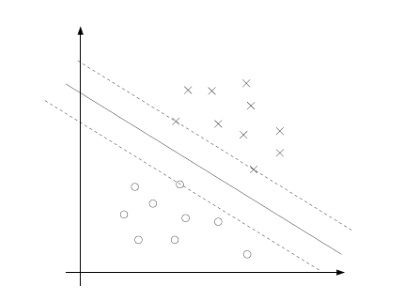
约束条件：

根据拉格朗日对偶问题（不细究其具体推导了）：

目标函数3的拉格朗日对偶问题为：



其中，称为拉格朗日算子。



上图中实线是最大间隔超平面，虚线是函数间隔为1的超平面，其上的点的拉格朗日算子，其它点的拉格朗日算子，上图中两条虚线上的3个点称为支持向量。

对上述的拉格朗日对偶问题求偏导：



令其分别为0：



从而得到如下问题：其中，，均为实数



进一步推导：



其中将向量内积表示为。

目标问题4：



约束条件：

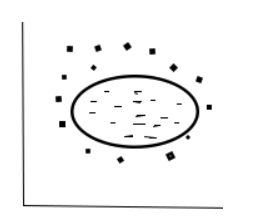


根据之前的求解：



之前如果有新的样本需要分类，需要根据和进行一次线性运算，现在有了后，只需要将新的样本与之前的样本进行运算就可以了，而且只有支持向量的，其它的，所以新的样本只要与支持向量进行运算就可以了。

核函数：对于非线性问题（与线性不可分不同，后续有说明），需要进行非线性变换，将非线性问题变成线性问题。



上图中，利用直线无法进行分隔，但利用椭圆则能有效的分隔。

令表示映射后的特征向量：

则目标函数4转换为目标函数5：



约束条件：



核函数：由于映射到特征空间的维数可能很高，直接计算比较困难，令：



目标函数5：



约束条件：





常用的核函数：

线性核：



多项式核：



高斯核（）：



也就是所谓的径向基核函数，其中为sklearn中的gamma参数，越大曲线越尖锐，越小曲线越平缓。

拉普拉斯核（）：



软间隔支持向量机和松弛变量（）：

如果出现线性不可分的情况，意味着某些样本点的函数间隔不满足大于等于1的条件，样本点落在最大间隔超平面和边界超平面（函数间隔=1的超平面）之间了。

对每个样本点引入一个松弛变量：

目标函数3转变为：



约束条件：

拉格朗日对偶问题为：



其中，为拉格朗日算子。

求其偏导数：



令其分别为0：



目标函数4最终转变为：



约束条件：



相比对其他的机器学习算法，SVM没有对特征数据的分布进行任何假设，如果不清楚特征数据的分布，选择SVM相对比较好。

1. **集成方法**

将各种模型连接起来，获取到更好的模型。有两种最常见的集成方法：bagging（boost aggregating，自助聚集）和boosting（增强学习）。

Bagging：

1. 从原始样本集中抽取训练集。每轮使用Bootstraping的方法抽取n个训练样本（sklearn中的方法可以控制是否有放回抽样，并控制抽取的样本数），共进行k轮抽取，得到k个训练集，并认为k个训练集是相互独立的。
2. 每次使用一个训练集得到一个模型，k个训练集得到k个模型（根据具体问题采用不同的分类或回归方法）
3. 对于分类问题：将上步得到的k个模型采用投票方式得到分类结果；对于回归问题，计算上述模型的均值作为最后的结果。所有模型的权重是相同的。

AdaBoosting：使用全部的样本，每轮训练改变样本的权值，减小训练正确的样本的权值，增大错误样本的权值。

公式推导：

训练样本集为，其中；

初始化训练样本的权值分布：

对于训练轮次：

1. 使用具有权值分布的训练数据集进行学习，得到弱分类器；
2. 计算在训练数据集上的分类误差率：



1. 计算在强分类器（最终的分类器）中所占的权重：

假设经过m-1轮计算后，得到，则：



其中为所占的权重。

AdaBoosting采用指数损失，损失函数为：



令，为每一轮迭代的样本权值，则：



根据特性，如果分类正确与符号相同，如果分类错误与符号相反，则：



其中，分类误差率，则：

