一句话逻辑

1. **过拟合与欠拟合？如何解决？**

（１）过拟合是模型使用的参数过多而导致模型对训练数据过度拟合，以至于用该模型来预测其他测试样本的时候效果不好。欠拟合则刚好相反，是由于模型使用的参数过少，以至于得到的模型难以拟合训练数据的现象

（２）增加数据量、适当减小参数数量、正则化来解决过拟合。增加训练数据量是根据大数定理，当数据量足够大时，训练模型会无限逼近实际，而减小参数数量则是人工根据模型的需要，剔除那些跟得到模型不太相关的参数来实现

**２．为什么正则化可以防止过拟合呢？**

简单地理解就是，通过加入一个正则项，在最小化新的代价函数的时候，正则项使得预测值与真实值之间的误差并不会达到最小，也就是说它是不会去完美拟合的，降低模型复杂度，这样也就防止了过拟合，提高了机器学习模型的泛化能力，正则化函数项的作用是选择经验风险与模型复杂度同时较小的模型，对应结构风险最小化。

**３．L1正则和L2正则作用、区别？**

L1正则可以产生稀疏权值矩阵，会把不重要的特征直接置零，鲁棒性更强，对异常值更不敏感

L2正则可以让参数衰减，选择更多的特征，这些特征都会趋近于０，防止模型过拟合

区别：相比L1，L2 计算起来更方便，可以直接求导获得取最小值时各个参数的取值

**４.　模型偏差和方差，各个模型的偏差与方差的特点**

偏差：描述的是预测值的期望与真实值之间的差距。偏差越大，越偏离真实数据

如何解决：选择正确的模型，避免过小的数据集，合适的模型复杂度，找更好、更多的特征。

方差：描述的是预测值的变化范围，离散程度，方差越大，数据的分布越分散

如何解决：增大数据集，减少特征，正则化，交叉验证

泛化误差可以分解为：偏差+方差+噪声

一般来说简单的模型会有一个较大的偏差和较小的方差，复杂的模型偏差较小方差较大，欠拟合偏差大，过拟合方差大。

RF：实现了有放回取样，特征随机，多次训练取均值的效果，所以方差会比较小。但是由于每棵树都是弱学习器，偏差较大。

GBDT：N棵树组成了相关关联，层层递进的超级学习者，它的方差一定是比较大的。但由于它的学习能力比较强，所以，它的偏差是很小的，而且树的棵树越多，学习能力就越强，偏差就越小

XGBOOST中，我们选择尽可能多的树，尽可能深的层，来减少模型的偏差；  
通过交叉验证，通过正则化，来减少模型的方差

**５.　奥卡姆剃刀原理**

这个原理称为“如无必要，勿增实体”，即“简单有效原理”

其含义是：在其他条件一样的情况下，选择简单的那个

**６. 经验风险与结构风险**

机器学习过程采用的策略就是使损失函数最小化，这其实就是一个最小化误差的过程。也就是说，最小化误差可通过最小化损失函数达成，这个损失我们称之为“经验风险”， 在模型能够较好地匹配已知数据的前提下，模型越简单越好。而模型的“简单”程度可通过模型的参数情况来度量，所以一般采用在经验损失的基础上加上一项关于模型参数复杂度的规则化函数J（w）来平衡。加上约束项的模型我们称之为“结构风险”

**７.　模型的评估指标（分类模型、回归模型、聚类模型各有哪些评估指标）**

分类模型：六大模型评估指标：

Accuracy（准确率）： 全量样本中被预测正确的样本站样本总数的比例；

Precision（精确度）: 所有预测值为正例的样本中，真实值为正例的样本所占的比例；

Recall（查全率）： 所有真实值为正例的样本中被预测正确的比例；

F-score（P和R的平衡指标）： 精确度和查全率的调和平均数；

Specificity（特异度）： 所有真实值为反例的样本中被预测正确的比例；

FPR（假正率）； 所有真实值为反例的样本中被预测错误的比例；

回归模型：回归模型任务目标是使得预测值能尽量拟合实际值，因此常用的性能度量方式主要有绝对误差和均方误差两种。1．MAE绝对误差：预测点与真实点之间距离之差的绝对值的平均值2．MSE均方误差：预测点与实际点之间距离之差平方和的均值

聚类模型：簇内误差平方和，簇内紧密性，簇间中心间隔，轮廓系数（越接近１越好）

**8.　泰勒公式，有什么用？**

是用一个函数在某点的信息，描述其附近取值的公式

简单讲就是用一个多项式函数去逼近一个给定的函数，在有限步内得到足够精确的值，机器学习中主要应用于梯度迭代。

**9.　优化算法（常见的有哪些，梯度下降法和牛顿法介绍下）**

梯度下降法：梯度就是函数的导数方向，往梯度下降的方向迭代求解时误差的收敛速度最快

随机梯度：固定步长λ改为动态步长λk，可以有效避免陷入局部最优的情况

def gd(x\_start, step, g):

x = x\_start

for i in range(20):

grad = g(x)

x -= grad \* step

print '[ Epoch {0} ] grad = {1}, x = {2}'.format(i, grad, x)

if abs(grad) < 1e-6:

break

return x

**def** **f**(x):损失函数

**return** x **\*** x **-** 2 **\*** x **+** 1

**def** **g**(x): 损失函数求导

**return** 2 **\*** x **-** 2

另外一个在梯度下降中十分重要的东西，那就是动量——momentum：

Nesterov梯度下降会预知你的下一步将会时到坡的对面去，所以会提示你提前刹车，避免过度冲到坡的对面去。这包含了一种提前计算下一步的梯度，来指导当前梯度的想法。就是每次计算梯的时候用下一次的权重，这样下一次的变化就会反应在这次计算的梯度上，如果下一次权重计算的梯度很大，则说明更新权重的时候更应该把下一次的梯度给加上去。

牛顿法：牛顿法是求解无约束最优化问题的常用方法，是二阶收敛，梯度下降是一阶收敛，所以牛顿法更快。  
通俗地讲，比如你想找一条最短的路径走到一个盆地的底部，梯度下降法是每次从你当前所处位置选一个坡度最大的方向走一步，而牛顿法在选择方向时，不仅会考虑坡度是否够大，还会考虑你走了一步之后，坡度是否会变得更大。也就是说，牛顿法比梯度下降法看得更远一点，因此能更快地走到底部。  
或者从几何上说，牛顿法就是用一个二次曲面去拟合你当前所处位置的局部曲面，而梯度下降法是用一个平面去拟合当前的局部曲面，通常情况下，二次曲面的拟合会比平面更好，所以牛顿法选择的下降路径更符合真实的最优下降路径

拟牛顿法：牛顿法虽然收敛速度快，但是需要计算黑塞矩阵的逆矩阵  ，而且有时目标函数的黑塞矩阵无法保持正定，从而使得牛顿法失效。为了克服这两个问题，人们提出了拟牛顿法。这个方法的基本思想是：不用二阶偏导数而构造出可以近似黑塞矩阵的正定对称阵。

阻尼牛顿法：当初始点x0远离极小值点时，牛顿法可能不收敛。原因之一是牛顿方向d=−∇2f（xk）−1∙∇f（xk）不一定是下降方向，经迭代，目标函数值可能上升。此外，即使目标函数值是下降的，得到的点xk+1也不一定是沿牛顿方向最好的点或极小值点。因此人们提出阻尼牛顿法对牛顿法进行修正。  
阻尼牛顿法在牛顿法的基础上增加了动态步长因子λk

**10.　激活函数**

神经网络为什么需要激活函数：首先数据的分布绝大多数是非线性的，而一般神经网络的计算是线性的，引入激活函数，是在神经网络中引入非线性，强化网络的学习能力。所以激活函数的最大特点就是非线性

Sigmoid、tanh函数：sigmod值域为 [0, 1 ]和tanh值域为 [−1, 1 ]就属于典型容易进入梯度饱和区的函数，即自变量进入某个区间后，梯度变化会非常小，就会发生梯度消失的情况。表现在图上就是函数曲线进入某些区域后，越来越趋近一条直线，梯度变化很小，梯度饱和会导致训练过程中梯度变化缓慢，。从而造成模型训练缓慢

ReLU函数：值域为 [0,+∞]，D:\360安全浏览器下载\qq\514456035\Image\C2C\514456035-1153253923-EA8AE31FD8A7A6074F0BACF98B22B867.JPG，导数为　D:\360安全浏览器下载\qq\514456035\Image\C2C\514456035-4083894657-27881BC1373A9591ABD2C3F824C1155E.JPG　　为什么现在这么多网络都更多地使用ReLU函数呢？重要原因就是其稀疏激活性，ReLU将x<0的输出置为0，就是一个去噪音，稀疏矩阵的过程，它的作用是如果计算出的值小于0，就让它等于0，否则保持原来的值不变。这是一种简单粗暴地强制某些数据为0的方法，同时大大地提高了训练速度

ReLU在x>0下，导数为常数1的特点：导数为常数1的好处就是在“链式反应”中不会出现梯度消失，但梯度下降的强度就完全取决于权值的乘积，这样就可能会出现梯度爆炸问题。解决这类问题：一是控制权值，让它们在（0，1）范围内；二是做梯度裁剪，控制梯度下降强度，如ReLU(x)=min(6, max(0,x))

ReLU在x<0下，输出置为0的特点：

leaky-ReLU函数：为了防止模型的‘Dead’情况，后人将x<0部分并没有直接置为0，而是给了一个很小的负数梯度值ａ为常数，一般设置 0.01，但是效果不是很稳定，所以在实际中 Leaky ReLu 使用的并不多。

**11.　生成模型与判别模型**

在机器学习中任务是从属性X预测标记Y

判别模型求的是P(Y|X)，即后验概率；而生成模型最后求的是P(X,Y)，即联合概率。

从本质上来说：判别模型之所以称为“判别”模型，是因为其根据X“判别”Y；而生成模型之所以称为“生成”模型，是因为其预测的根据是联合概率P(X,Y)，如何“生成”的依赖于P(X,Y)，那么最后的预测结果选哪一个Y呢？那就选“生成”概率最大的那个吧~

生成模型：给定输入x产生输出y的生成关系:朴素贝叶斯和隐马尔可夫模型

特点：可以还原出联合概率分布，判别不能，当存在隐变量时，生成方法仍可以使用，判别不行。

判别模型：由数据直接学习决策函数，或者概率分布作为预测模型：Kｎｎ，感知机，决策树，LR，最大熵模型，SVM，BOOSTING,条件随机场。

特点：学习准确度更高，能定义特征并使用特征，可以简化学习问题。

12．**核函数**

常用的核函数如下：线性核函数，高斯核函数，多项式核函数。核函数和映射没有关系**。**核函数只是用来计算映射到高维空间之后的内积的一种简便方法。核函数是二元函数，输入是映射之前的两个向量，其输出等价于两个向量映射之后的内积。你并不需要知道具体对应哪种映射，表达式是什么，你需要知道的是核函数肯定对应于某一种映射即可。

线性核函数：https://img-blog.csdn.net/20150724004048762　　　高斯核函数：　https://img-blog.csdn.net/20150724004209557　　多项式核函数：https://img-blog.csdn.net/20150724004123784

高斯核函数对于数据中的噪音有着较好的抗干扰能力，其参数决定了函数作用范围，σ越大，那么高斯核函数的局部影响范围就会越大，超过了这个范围，特征上的权重实际上衰减得非常快，数据的作用就“基本消失”。

**13.　最小二乘法**

它通过最小化误差的平方和寻找数据的最佳函数匹配。利用最小二乘法可以简便地求得未知的数据，并使得这些求得的数据与实际数据之间误差的平方和为最小https://img-blog.csdnimg.cn/img_convert/0d0470d79401a535c5c03b2214bdbbb3.png，最小二乘法需要计算 的逆矩阵，有可能它的逆矩阵不存在，这样就没有办法直接用最小二乘法了，此时梯度下降法仍然可以使用，当样本特征 ｎ 非常的大的时候，计算  的逆矩阵是一个非常耗时的工作，建议超过10000个特征就用迭代法吧。或者通过主成分分析降低特征的维度后再用最小二乘法。

如果拟合函数不是线性的，这时无法使用最小二乘法，需要通过一些技巧转化为线性才能使用，此时梯度下降仍然可以用．

**14．线性回归**

分为简单线性回归和[多元线性回归](https://so.csdn.net/so/search?from=pc_blog_highlight&q=%E5%A4%9A%E5%85%83%E7%BA%BF%E6%80%A7%E5%9B%9E%E5%BD%92" \t "_blank)，线性回归，不要忘了**前提假设是y和x呈[线性关系](https://www.zhihu.com/search?q=%E7%BA%BF%E6%80%A7%E5%85%B3%E7%B3%BB&search_source=Entity&hybrid_search_source=Entity&hybrid_search_extra={"sourceType":"article","sourceId":"72513104"}" \t "https://www.zhihu.com/_blank)**，如果两者不是线性关系，就要选用其他的模型啦。



使用残差作为损失函数，使用均方误差最小化目标函数的方法称为最小二乘法。分别对W和b求偏导得出W,b

优点：直接，快速，可解释性好。缺点：需要严格的假设，需处理异常值，存在共线性，自相关，异方差等问题

**15. LOGISTIC回归**

使用概率作为输出结果使得样本在距离很小的差别下不再强制地输出+1和-1这两种天壤之别的结果，而是通过概率的方式告诉你结果可能是多少，同时也告诉你预测的不确信程度。这样子看起来让人比较安心一点不是吗？

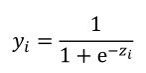
实际上，“分类”是应用逻辑回归的目的和结果，但中间过程依旧是“回归”。为什么这么说？因为通过逻辑回归模型，我们得到的计算结果是0-1之间的连续数字，可以把它称为“可能性”（概率）。对于上述问题，就是：客户购买某个商品的可能性，借款人违约的可能性。

然后，给这个可能性加一个阈值，就成了分类。例如，算出贷款违约的可能性>0.5，将借款人预判为坏客户

我们常常用**概率(Probability)** 来描述一个事件发生的可能性。

而**似然性(Likelihood)** 正好反过来，意思是一个事件实际已经发生了，反推在什么参数条件下，这个事件发生的概率最大。

**可知线性回归是逻辑斯蒂回归的基础，但是线性回归用于拟合一个具体的连续值，属于真正的回归问题，而逻辑斯蒂回归得到的是概率值，解决的其实是二分类问题**

Logistic回归模型就是将线性回归的结果输入一个Sigmoid函数，将回归值映射到0~1，表示输出为类别的概率。Sigmoid函数：，逻辑回归是用来解决分类问题的，这里有一个前提假设，就是样本服从0-1分布，也就是伯努利分布n=1的情况（伯努利分布指的是对于随机变量X有, 参数为p(0<p<1)，如果它分别以概率p和1-p取1和0为值。EX= p,DX=p(1-p)），需要用最大似然估计的方法来评判模型的好坏，采用梯度上升法最大化log-likelihood。让总体分布尽量与样本的分布趋同，就是总体的分布与样本分布具有最大的相似性，然后再来求取模型中的参数，这样就可以得到比较符合最大似然估计的模型。最大似然估计的方法要求似然函数的最大值，损失函数在其前面加上负号，就是求最小值。

优点：１.直接对分类的概率建模，无需实现假设数据分布，从而避免了假设分布不准确带来的问题。2.不仅可预测出类别，还能得到该预测的概率，这对一些利用概率辅助决策的任务很有用；3.对数几率函数是任意阶可导的凸函数，有许多数值优化算法都可以求出最优解。（梯度下降法，牛顿法）

缺点：１.Logistic回归模型本质上还是一种线性模型，只能做线性分类，不适合处理非线性的情况，一般需要结合较多的人工特征处理使用。2.Logistic回归对正负样本的分布比较敏感，所以要注意样本的平衡性

经典线性模型的优化[目标函数](https://www.zhihu.com/search?q=%E7%9B%AE%E6%A0%87%E5%87%BD%E6%95%B0&search_source=Entity&hybrid_search_source=Entity&hybrid_search_extra={"sourceType":"article","sourceId":"47200561"}" \t "https://www.zhihu.com/_blank)是最小二乘，而逻辑回归则是[似然函数](https://www.zhihu.com/search?q=%E4%BC%BC%E7%84%B6%E5%87%BD%E6%95%B0&search_source=Entity&hybrid_search_source=Entity&hybrid_search_extra={"sourceType":"article","sourceId":"47200561"}" \t "https://www.zhihu.com/_blank)

**16.　SVM**

1. **软间隔支持向量机：**SVM 想要的就是找到各类样本点到超平面的距离最远，考虑到实际情况下有些样本点存在混叠的情况，将允许支持向量机在一些样本上出错，增加一个松弛变量ξi，即引入“软间隔”的概念，软间隔支持向量机的最终模型仅与支持向量有关，通过采用hinge合页损失函数仍保持了稀疏特性
2. **非线性支持向量机：**实际中很多数据可能并不只是带有异常点这么简单，而是完全非线性可分的。引入核函数， Xｉ 与Xｊ 在高维特征空间的内积等于它们在原始样本空间中通过核函数K（ｘ，ｙ）  计算的结果，将二维线性不可分样本映射到高维空间中，让样本点在高维空间线性可分。

**优点：**１. 有严格的数学理论支持，可解释性强，不依靠统计方法　２. 采用核技巧之后，可以处理非线性分类/回归任务　3. 最终决策函数只由少数的支持向量所确定，计算的复杂性取决于支持向量的数目，而不是样本空间的维数，这在某种意义上避免了“维数灾难”4. SVM模型具有较好的泛化能力，因为其本身的优化目标是结构风险最小，而不是经验风险最小

缺点：１. 训练时间长　2. 模型预测时，预测时间与支持向量的个数成正比。当支持向量的数量较大时，预测计算复杂度较高。3. 不方便解决多分类问题

**17.　KNN模型**

**K近邻分类**的思想就是：对于任意一个新的样本点，我们可以在这M个已知类别标签的样本点中选取K个与其距离最接近的点作为它的最近邻点，然后统计这K个最近邻点的类别标签，采取多数投票表决的方式，即把这K个最近邻点中占绝大多数类别的点所对应的类别拿来当作要预测点的类别。

K近邻模型主要有三个要素：K值的选择、距离度量方法（一般为欧式）、分类决策规则。K近邻有3个参数对结果影响较大，一个是数据归一化，另一个是K值的选择，还有一个是距离的度量方式。优点：简单且易于实现，比较适合多分类问题，对异常值不大敏感，无数据输入假定。缺点：预测精度一般，对于大部分问题比不上基于树的方法。当样本存在范围重叠时，K近邻的分类精度很低。即便分类一个样本也要计算所有数据，在大数据环境下很不适应。

**K近邻回归：**决策规则为：即把这K个最近邻训练样本实例xi的输出值yi的平均作为待预测实例xj的值

1. **朴素贝叶斯**

朴素贝叶斯分类器就是一个对所有可能性求概率的模型，最后输出结果中哪种可能性高就输出哪种

朴素贝叶斯是一个基于贝叶斯定理和特征条件独立假设的分类方法，属于生成模型。基于条件独立假设求出输入—输出的联合概率P（X,Y），然后对于新的输入xi，利用贝叶斯定理求出后验概率P（yi|xi），即该对象属于某一类的概率，然后选择具有最大后验概率的类作为该对象所属的类别，达到分类预测的目的。朴素贝叶斯的最大化后验概率其实相当于最小化期望风险原则  
优点：过程简单且速度较快，因为它的预测过程实际就是进行概率的乘积。对于多分类问题也很有效，计算复杂度不会有大幅度的上升

缺点：属性个数比较多或者属性之间相关性较大时，分类效果不好。对于连续数值型变量特征，要求它服从正太分布

**19.　决策树**

从最初的没有信息，到最后的百分百确信，这是一个减少不确定性的过程。在决策树根节点的最初，我们先假设信息熵为1，表示我们的一无所知。到叶节点时假设信息熵为0，表示非常确信。那么使用决策树决策的过程就是我们不断减少信息熵的过程，直到它降为0。？我们多么希望能用尽可能少的判断次数就能确定苹果的好坏，放到信息熵的环境下，就是我们多希望信息熵能下降得快一点。这就涉及到决策树的构建了，我们该怎么构建才能使得这棵决策树的信息熵在判断分支中下降得最快呢？这就是信息增益。

因此**Gini指数越小，说明数据集越集中**，也就是纯度越高。它的概念等价于信息熵，熵越小说明信息越集中，两者的概念是非常近似的。所以当我们使用Gini指数来作为划分依据的时候，选择的是切分之后Gini指数尽量小的切分方法，而不是尽量大的。

决策树是一种树状结构模型，可以进行基本的分类与回归，是集成方法经常采用的基模，各种决策树的主体思想大同小异，都主要涉及三要素，分别是特征选择、决策树的生成和决策树的剪枝。  
特征选择：信息增益（如果一个属性的信息增益量越大，这个属性作为一棵树的根节点就能使这棵树更简洁），　信息增益比（特征X的信息增益g（Y,X）与特征X的取值的熵H（X）的比值），基尼系数（以用来度量任何不均匀分布，且介于0~1之间的数（0指完全相等，1指完全不相等），计算出数据集某个特征所有取值的基尼指数后，就可以得到利用该特征进行样本划分产生的基尼指数增加值（GiniGain）；决策树模型在生成的过程中就是递归选择 GiniGain 最小的节点作为分叉点）

当我们面对的不是一个类别预测问题，而是一个连续结果值预测的回归问题时，ID3决策树（以信息增益作为特征的选择标准）和C4.5决策树（信息增益比）均无法处理。回归问题的结果取值是很多个连续值，因此不能像分类问题一样采用信息增益或基尼系数等特征选择标准。CART既可以用于分类任务，又可以用于回归任务。

CART 决策树用于分类任务：采用基尼系数作为特征选择标准，对结果的处理方式是对每一个叶子节点里面包含的所有样本的类别进行统计，然后选择样本类别占多数者对应的类别标签作为该叶子节点的类别标签

CART决策树用于回归任务：采用平方误差最小化准则进行特征选择，把最终各个叶子中所有样本对应结果值的均值或者中位数当作预测的结果输出  
CART分类树与回归树采用二叉树来对每一个特征进行划分

决策树剪枝：L（α,Ti）=L（Ti） + α|Ti| （正则化项α|Ti|（|Ti|是子树Ti的叶子节点的数量），式中的L（Ti）是关于叶子节点数的递减函数，α|Ti|是关于叶子节点数的递增函数

优点：时间复杂度较小，适合高维数据，决策树算法可处理连续和种类字段，可以用于小数据集，准确性高: 挖掘出来的分类规则准确性高, 便于理解, 决策树可以清晰的显示哪些字段比较重要, 即可以生成可以理解的规则

缺点：对噪声比较敏感，容易过拟合，在训练集噪声较大时得到的模型容易过拟合，处理特征关联性较强的数据时表现不太好。

**20.　RF**

RF默认采用CART作为基学习器，而且它在Bagging模型的基础上再进一步，每次训练基学习器时，除对样本随机采样外，对样本的特征也进行随机采样，树模型本身具有较强的非线性拟合能力

优点：１.RF各个基学习器之间没有强关联，因此学习过程可以分开，实现并行化　２. RF每次训练时，各个基学习器只是抽取样本的部分特征进行训练，因此对于样本特征维度很高的情况，RF仍能高效地训练模型　３. RF的每个基学习器只是随机抽取部分样本和部分特征进行学习，因此模型的泛化能力较强　４. 采用的基学习器是CART决策树，而CART决策树对缺失值不敏感，因此RF对部分特征缺失也不敏感。５. RF训练模型后可以顺便输出各个特征对预测结果的重要性，因此可以辅助我们进行特征选择。  
缺点：１. 决策树模型的一个缺点是对噪声比较敏感，所以RF在某些噪声较大的样本集上容易陷入过拟合

**21.　ADABOOST（adaptive）**

“关注”被错分的样本，“器重”性能好的弱分类器：提升树就是一种采用加法模型与前向分步算法，并以决策树为基分类器的提升方法，针对同一个训练集训练不同的分类器，越难区分的样本在训练过程中会变得越重要。通过每次降低个体学习器的分类误差，加大效果好的个体学习器的重要性，得到最终的集成学习器。同时给分类误差率较小的基分类器以更大的权值，给分类误差率较大的基分类器以小的权重值。采用的是指数损失

原理：对弱分类器的要求一般是足够简单，并且是低方差和高偏差的。因为训练的过程是通过降低偏差来不断提高最终分类器的精度

**22.　GBDT**

梯度提升简单来说就是利用损失函数的负梯度将当前模型的值来作为提升树算法中残差的近似替代。模型的结果是一组回归分类树组合(CART Tree Ensemble)： T1…Tｋ 。其中 Tｊ学习的是之前 ｊ－１棵树预测结果的残差，这种思想就像准备考试前的复习，先做一遍习题册，然后把做错的题目挑出来，在做一次，然后把做错的题目挑出来在做一次，经过反复多轮训练，取得最好的成绩。

AdaBoost是通过提升错分数据点的权重来定位模型的不足而Gradient Boosting是通过算梯度 （gradient）来定位模型的不足。因此相比AdaBoost, Gradient Boosting可以使用更多种类的目标函数。

梯度提升分类树GBDT：对数损失函数或者指数损失函数

梯度提升回归树GBRT：平方损失函数（缺点：一个异常值造成的损失由于二次幂而被过分放大，会影响到最后得到模型在测试集上的表现），Hｕｂｅｒ损失函数，对异常值鲁棒性较强

从决策边界来说，线性回归的决策边界是一条直线，逻辑回归的决策边界根据是否使用核函数可以是一条直线或者曲线，而GBDT的决策边界可能是很多条线。如果GBDT和线性回归或逻辑回归在某个问题上表现接近，那么我们应该选择相对比较简单的线性回归或逻辑回归。具体选择哪一个算法还是要根据实际问题来决定。

**23. XGBOOST**

XGBoost（eXtreme Gradient Boosting）是GBDT的一种改进形式，具有很好的性能，XGBoost采用的是解析解思维，即对损失函数进行二阶泰勒展开（传统的GBDT/GBRT模型只用到了损失函数的一阶导信息（一阶泰勒展开）），求得解析解，然后用这个解析解作为“增益”来辅助建立 CART 回归树，最终使得整体损失达到最优。

优点：１.XGBoost 支持自定义损失函数，只要满足定义的损失函数二阶可导即可，这大大增加了处理问题的灵活性。２.在GBDT的基础上加入了一个正则化项，用于控制模型的复杂度，正则化项里面包含了树的叶子节点数和各个叶子节点输出值的平方之和。从方差—偏差角度来看，正则化项可以降低模型的方差，使学习出来的模型更加简单，防止模型过拟合。３.XGBoost 模型借鉴了随机森林的做法，支持对特征进行抽样，这也可以起到降低过拟合风险和减少计算量的作用。４. 当遇到一个负“增益”时，GBDT/GBRT 会马上停止分裂，但 XGBoost会一直分裂到指定的最大深度，然后回过头来剪枝。如果某个节点之后不再有负值，则会除掉这个分裂；但是如果负值后面又出现正值，并且最后综合起来还是正值，则该分裂会被保留。５. XGBoost 对于不同节点遇到的特征缺失将采用不同的处理方式，并且会逐渐学习出处理缺失值的方式，当后面再遇到有缺失特征时就可以按学习出的处理方式进行处理，这样更加科学。６. XGBoost支持并行：XGBoost的并行不是在tree粒度上，而是在特征粒度上。xgboost在训练之前，预先对数据进行了排序，然后保存为block结构，后面的迭代中重复地使用这个结构，大大减小计算量。这个block结构也使得并行成为了可能，在进行节点的分裂时，需要计算每个特征的增益，最终选增益最大的那个特征去做分裂，那么各个特征的增益计算就可以开多线程进行。XGBoost为什么执行效率那么高，在这里做一个简单的介绍。

首先是用于并行学习的列块(column block)，在树的学习中最耗时的是在各个特征中寻找分裂点，而寻找分裂点中最耗时的是对特征值进行排序。为了减少排序的消耗，XGBoost会把排好顺序的数据保存到内存块中，称为”block"。这些内存块仅需在训练前计算一次，后续迭代中即可重复使用。

**24. 　K-MEANS**

K-Means聚类过程时，涉及四个关键点，即聚类簇数K值的选择、K个聚类中心点的初始值选择、距离度量方式（样本点之间的距离度量用得比较多的还是欧氏距离）、损失函数的选择（各个簇中样本向量到对应簇均值向量的均方误差）。只适合簇型数据，对其他类型的数据聚类效果一般。当数据类别严重不平衡时，聚类效果不佳。

k-means++改进方法  
第1步：先随机从M个样本点中选择一个样本点作为聚类中心，设其为μ1，然后计算各个样本点到该聚类中心的距离D:\360安全浏览器下载\qq\514456035\Image\C2C\CCE30E57714744307B0C67970450D541.png；选择距离最远的一个样本点，将其作为第二个聚类中心μ2。

**26. 层次聚类**

层次聚类（Hierarchical Clustering），顾名思义，就是一层一层地进行聚类。它是一类算法的总称，主要通过从下往上不断合并簇，或者从上往下不断分离簇形成嵌套的簇。我们把从下往上对小类进行聚合的方式叫作凝聚法，把从上往下将大类分割成小类的方式叫作分裂法

**27. 密度聚类**

密度聚类算法假设聚类结构能够通过样本分布的紧密程度确定，其从样本密度的角度考察样本之间的可连接性，并且基于可连续样本不断扩展聚类簇，以获得最终的聚类结果。

**28. 谱聚类**

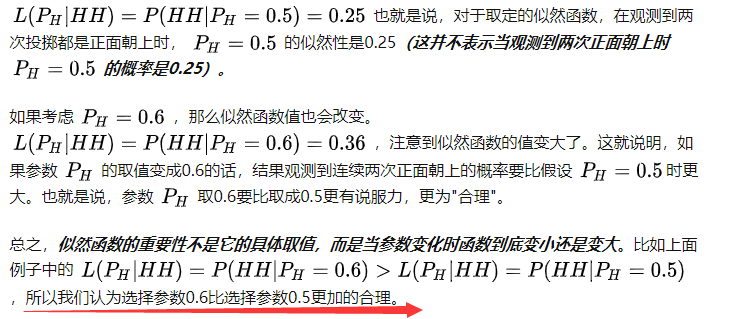
谱聚类（Spectral Clustering）是从图论中演化出来的算法，也是一种广泛使用的聚类算法。与传统的 K-Means 算法或层次聚类算法等相比，谱聚类对数据分布的适应性更强，聚类效果比较优秀，同时聚类的计算量也小很多。  
谱聚类的主要思想：把所有的样本数据看作空间中的点，点之间用边连接起来形成图。图里面边的权重与点之间的距离有关，距离较远的两点之间的边权重值较低，而距离较近的两点之间的边权重值较高。对所有数据点组成的图进行切图，让切图后不同子图间边的权重和的值尽可能小，而子图内边的权重和的值尽可能大，从而达到聚类的目的，

**29. 似然函数和最大似然估计**

概率描述的是：指定参数后，预测即将发生事件的可能性；有一个箱子，装有形状相同的黑色球和白色球共100个，其中黑色球有90个，白色球10个，现在从箱子中任取一个球，结果是黑色球的概率？

似然描述的是：在已知某些观测所得到的结果时，对有关事物的性质的参数进行估计；有一个箱子，装有形状相同的黑色球和白色球100个，其中一种颜色90个，另一种颜色球10个，现在从箱子中任取一球，结果所取得的球是黑色球，箱中黑色球是90个可能性是多少？

假设硬币投出时会有 Pｈ的概率朝上，有１-Pｈ的概率反面朝上，当然这里的Pｈ是未知的，你并不知道他是0.5还是0.6甚至是1。

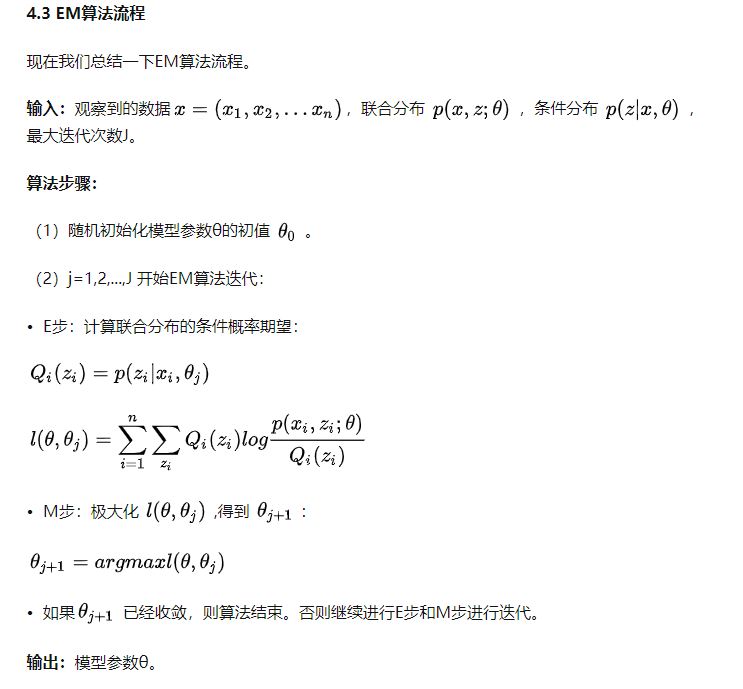


最大似然估计：现在已经拿到了很多个样本（你的数据集中所有因变量），这些样本值已经实现，最大似然估计就是去找到那个（组）参数估计值，使得前面已经实现的样本值发生概率最大。因为你手头上的样本已经实现了，其发生概率最大才符合逻辑。

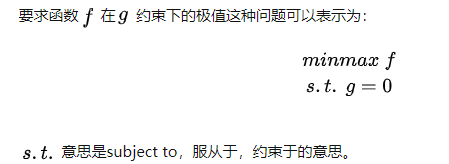
**30. EM算法**

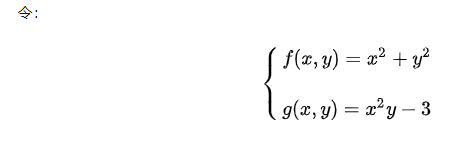
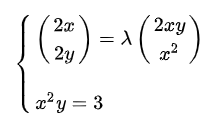
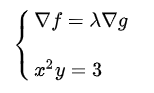
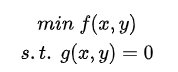
EM算法是一种迭代优化策略，由于它的计算方法中每一次迭代都分两步，其中一个为期望步（E步），另一个为极大步（M步），所以算法被称为EM算法（Expectation-Maximization Algorithm）。

其基本思想是：E-Step 主要通过观察数据和现有模型来估计参数，然后用这个估计的参数值来计算似然函数的期望值，而 M-Step 是寻找似然函数最大化时对应的参数。由于算法会保证在每次迭代之后似然函数都会增加，所以函数最终会收敛，直到参数不再更新为止。EM算法收敛的优劣很大程度上取决于其初始参数。



**31. 拉格朗日乘子法、对偶**

拉格朗日乘子即为λ：

令：求

在约束最优化问题中，常常利用拉格朗日对偶性(Lagrange duality)将原始问题转为对偶问题，通过解决对偶问题而得到原始问题的解。１.对偶问题的对偶是原问题；2无论原始问题是否是凸的，3对偶问题都是凸优化问题；4对偶问题可以给出原始问题一个下界；5当满足一定条件（KKT）时，原始问题与对偶问题的解是完全等价的；

**32. 特征分解与SVD奇异值分解**

两种分解用途：节省存储（分解成三个小矩阵）、降维（以用到特征值分解的PCA为代表）、去噪（小奇异值大概率上是噪声，是某些非重要和非本质特征，因此可以去掉），两个分解的目的都是提取出一个矩阵最重要的特征

特征值分解条件：ｎ个线性无关的特征向量矩阵。特征值分解可以得到特征值与特征向量，特征值表示的是这个特征到底有多重要，而特征向量表示这个特征是什么，可以将每一个特征向量理解为一个线性的子空间，我们可以利用这些线性的子空间干很多的事情。不过，特征值分解也有很多的局限，比如说变换的矩阵必须是方阵。比如说有N个学生，每个学生有M科成绩，这样形成的一个N \* M的矩阵就不可能是方阵

但是对于奇异值分解来说，所有实数矩阵都可以进行。奇异值分解是一个有着很明显的物理意义的一种方法,它可以将一个比较复杂的矩阵用更小更简单的几个子矩阵的相乘来表示,这些小矩阵描述的是矩阵的重要的特性。在很多情况下，前10%甚至1%的奇异值的和就占了全部的奇异值之和的99%以上了。也就是说，我们也可以用前r大的奇异值来近似描述矩阵，这里定义一下部分奇异值分解：称作「奇异值」可能无法顾名思义迅速理解其本质，那咱们换个说法，称作「主特征值」，你可能就迅速了然了。

 SVD可以获取另一个方向上的主成分，而PCA只能获得单个方向上的主成分。通过SVD可以得到PCA相同的结果，但是SVD通常比直接使用PCA更稳定。因为PCA需要计算X⊤X的值，对于某些矩阵，求协方差时很可能会丢失一些精度。两者都是矩阵分解的技术，一个直接分解SVD，一个是对协方差矩阵操作后分解PCA

**33.　PCA主成分分析(无监督)**

其基本思想是：找出原始数据里最主要的方面来代替原始数据，使得在损失少部分原始信息的基础上极大地降低原始数据的维度。PCA 的本质是将方差最大的方向作为其主成分，然后将最相关的特征投影到同一个主成分上，从而达到降维的效果。投影的标准是保留最大方差。

有可能舍弃了一些看似无用的信息，但是这些看似无用的信息恰好是重要信息，只是在训练集上没有很大的表现，所以 PCA 也可能加剧了过拟合. PCA 不仅将数据压缩到低维，它也使得降维之后的数据各特征相互独立.

当对训练集进行 PCA 降维时，也需要对验证集、测试集执行同样的降维。而对验证集、测试集执行零均值化操作时，均值必须从训练集计算而来，不能使用验证集或者测试集的中心向量。

核化PCA  
前面讲到的PCA方法将样本特征从高维投影到低维时，采用的是一种线性映射的方式，即默认存在一个线性超平面，可以让我们对样本数据点进行投影。但在很多实际任务中，可能并不存在这样一个比较理想的线性超平面，所以研究者想到能否用一个非线性的超平面来进行投影呢？答案是肯定的。  
前面我们看到，在线性投影中，当需要将原始数据集从N降到N＇维时，只求解XXTW=λW就可得到投影矩阵W，然后利用zi=WTxi即可达到降维的目的。类似于支持向量机中的思想，我们引入核函数，先将需要降维的样本数据进行一次非线性映射，对映射后得到的结果使用线性PCA的流程

**34.　LDA线性判别分析**

与 PCA 相同的是，LDA 也基于某个投影矩阵对样本点进行降维；与PCA不同的是，LDA需要利用样本的类别信息，是一种有监督降维方法。用一句话概括LDA的思想就是：投影后类内方差最小，类间方差最大。即我们要将数据投影到低维空间，并希望投影后每一种类别数据的投影点尽可能接近，而不同类别数据的类别中心之间的距离尽可能的大。LDA 和 PCA 的主要区别在于采用不同的思想，得到不同的投影矩阵。二者默认都假设被降维的样本数据服从高斯分布。

**35．词向量**

通常需要首先将语言数学化，词向量就是用来将语言中的词进行数学化的一种方式。

一种最简单的词向量方式是 one-hot representation，就是用一个很长的向量来表示一个词，向量的长度为词典的大小，向量的分量只有一个 1，其他全为 0， 1 的位置对应该词在词典中的位置。但这种词表示有两个缺点：（1）容易受维数灾难的困扰，尤其是将其用于 Deep Learning 的一些算法时；（2）不能很好地刻画词与词之间的相似性

另一种就是Distributed Representation 这种表示，通过训练将某种语言中的每一个词映射成一个固定长度的短向量，将所有这些向量放在一起形成一个词向量空间，而每一向量则为该空间中的一个点，在这个空间上引入“距离”，则可以根据词之间的距离来判断它们之间的（词法、语义上的）相似性了。

有了Word2Vec模型训练的词向量后，我们可以做的事情很多，最常见的有文本分类和文本聚类。具体做法就是：对测试集中每一篇文档中的词向量求一个平均值，这个平均值就相当于代表了这一篇文章的语义，然后通过比较该平均词向量和训练集文档中各文档对应的平均词向量的距离，就可以预测这篇文档最有可能属于哪一类了。但是上述对文档中所有词向量取平均的处理方式有一个小小的缺点，那就是会抹灭文档中各个词的排列顺序产生的作用。为了解决这一问题，就有了下面要介绍的Doc2Vec模型了。  
Doc2Vec模型除在Word2Vec模型的基础上增加了一个段落向量（注意，这里实际是将一整篇文档当作一个段落了）外，其他部分和Word2Vec模型几乎是完全一样的

**36.　深度神经网络（Deep Neural Networks， 简称DNN）**

DNN可以理解为有很多隐藏层的神经网络，DNN内部的神经网络层可以分为三类，输入层，隐藏层和输出层,如下图示例，一般来说第一层是输入层，最后一层是输出层，而中间的层数都是隐藏层。神经网络的基本原理主要涉及前向传播和反向传播两个过程，所谓的DNN前向传播算法就是利用若干个权重系数矩阵W,偏倚向量b来和输入值向量x进行一系列线性运算和激活运算，从输入层开始，一层层的向后计算，一直到运算到输出层，得到输出结果为值。在进行DNN反向传播算法前，我们需要选择一个损失函数，来度量训练样本计算出的输出和真实的训练样本输出之间的损失，DNN可选择的损失函数有不少，为了专注算法，这里使用最常见的均方差来度量损失，损失函数有了，用梯度下降法迭代求解每一层的w,b。

**37.模型调参**

模型调参一直是建模人员向往的高地，但这要求建模人员具有强大的综合能力。例如对数据业务的理解、算法本质的理解、合适的调试工具、做实验的方法和策略、细致的观察和分析能力，等等

给定一组数据D和具有可调N个参数的算法 ，调参的目的就是在由参数组成的N维向量集 里，挑选一组参数λ，使得算法在训练集学习后能在验证集取得最小的损失。调算法学习不到的参数，模型调参到底是调哪些参数呢？答案是调超参数（Hyperparameter），简称超参。

随机种子是最常见的建模作弊的手段（或者建模工程师自己也没有意识到），当通过不断调整随机种子而择优选择的模型，就是作弊的模型。3个是否是超参的直观判断方法：1. 模型无法直接从数据中估算的参数；2.是否必须手动指定的参数（包含接口中的缺省的一些参数）；3.算法接口中的输入参数（排除与算法无关的参数，如输入数据等）。先训练复杂的模型并在后续逐渐简化，一定要铭记：参数只有更好，没有最好，应适可而止，朝着项目总体目标推进。在调参前通常需要了解3个调参要素：目标函数、搜索域和优化算法

目标函数：即定义要优化的目标/评估标准以判断模型，一般来说即最小化损失、最大化验证集上模型指标

搜索域：也称为域空间、配置空间，即超参的搜索空间，亦称为超参空间。实践中需关注参数的上下界、取值范围等。

搜索算法：在实践中，我们需选用某种调参搜索算法，例如常见的、下文要介绍的网格搜索、随机搜索、贝叶斯搜索以及它们组合而成的混合搜索

例如逻辑回归中只有L2的正则且solver参数为‘liblinear’时，对偶求解参数dual才生效。

参数分组：机器学习是偏差和方差权衡的艺术，一组是有利于偏差的参数、一组是有利于方差的参数。

**38. 特征选择与特征组合**

特征选择：是指选择获得相应模型和算法最好性能的特征集。1.计算每一个特征与响应变量的相关性：工程上常用的手段有计算皮尔逊系数和互信息系数，皮尔逊系数只能衡量线性相关性而互信息系数能够很好地度量各种相关性。2. 构建单个特征的模型，通过模型的准确性为特征排序，借此来选择特征　3. 通过L1正则项来选择特征：L1正则方法具有稀疏解的特性，因此天然具备特征选择的特性，但是要注意，L1没有选到的特征不代表不重要，原因是两个具有高相关性的特征可能只保留了一个，如果要确定哪个特征重要应再通过L2正则方法交叉检验　4. 训练能够对特征打分的预选模型：RandomForest和Logistic Regression等都能对模型的特征打分，通过打分获得相关性后再训练最终模型；5.  通过特征组合后再来选择特征（对非线性规律进行编码）：如对用户id和用户特征最组合来获得较大的特征集再来选择特征，这种做法在推荐系统和广告系统中比较常见，这也是所谓亿级甚至十亿级特征的主要来源，原因是用户数据比较稀疏，组合特征能够同时兼顾全局模型和个性化模型　6. 通过深度学习来进行特征选择：目前这种手段正在随着深度学习的流行而成为一种手段，尤其是在计算机视觉领域，原因是深度学习具有自动学习特征的能力　7.离散化　8.合成特征（比如：两个特征相除）

我们最终获得的预测能力将远远超过任一特征单独的预测能力。例如，如果狗狗在下午 5 点主人下班回来时（快乐地）叫喊，可能表示对主人满意度的正面预测结果。如果狗狗在凌晨 3 点主人熟睡时（也许痛苦地）哀叫，可能表示对主人满意度的强烈负面预测结果。

**39.特征工程**

什么是特征工程呢？一个非常简单的例子，现在出一非常简答的二分类问题题，请你使用逻辑回归，设计一个身材分类器。输入数据X:身高和体重 ，标签为Y:身材等级（胖，不胖）。显然，不能单纯的根据体重来判断一个人胖不胖，姚明很重，他胖吗？显然不是。针对这个问题，一个非常经典的特征工程是，BMI指数，BMI=体重/(身高^2)。这样，通过BMI指数，就能非常显然地帮助我们，刻画一个人身材如何。甚至，你可以抛弃原始的体重和身高数据

所以说，特征工程就是通过X，创造新的X'。基本的操作包括，衍生（升维），筛选（降维）。说起来简单，实际中，衍生和筛选都是困难重重，甚至需要非常专业的专家知识。  
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，应当优先选择。

sklearn中的feature\_selection库来进行特征选择：方差选择法，相关系数法，卡方检验，互信息法，递归特征消除法，基于惩罚项的特征选择法， 基于树模型的特征选择法，降维

**40.样本不平衡**

在现实收集的样本中，正负类别不均衡是现实数据中很常见的问题，会存在“长尾现象”，也就是所谓的“二八原理”。一个分类器往往 Accuracy 将近90%，但是对少数样本的判别的 Recall 却只有10%左右。这对于我们正确找出少数类样本非常不利。一般而言，正负样本比例超过1:3，分类器就已经会倾向于负样本的判断（表现在负样本Recall过高，而正样本 Recall 低，而整体的 Accuracy依然会有很好的表现）。在这种情况下，我们可以说这个分类器是失败的，因为它没法实现我们对正类人群的定位。如果数据不但不平衡，样本数还非常少，这样的问题就非常棘手。

过采样是把小众类复制多份，每次生成新数据点时加入轻微的随机扰动，欠采样是从大众类中剔除一些样本。多次欠采样（放回采样，这样产生的训练集才相互独立）产生多个不同的训练集，进而训练多个不同的分类器。

模型上选对样本不均衡问题不敏感的模型，如决策树；或者调整分类阈值，使得更倾向与类别少的数据。

**41.模型选择**

数据的维度大小，数据的质量和数据的特征属性；你可以利用的计算资源；你所在的项目组对该项目的时间预计；你手上的数据能应用在哪些项目中；

**42.互信息**

互信息指的是两个随机变量之间的关联程度，即给定一个随机变量后，另一个随机变量不确定性的削弱程度，因而互信息取值最小为0，意味着给定一个随机变量对确定一另一个随机变量没有关系，最大取值为随机变量的熵，意味着给定一个随机变量，能完全消除另一个随机变量的不确定性。

