·          无监督和有监督算法的区别？

有监督学习：对具有概念标记（分类）的训练样本进行学习，以尽可能对训练样本集外的数据进行标记（分类）预测。这里，所有的标记（分类）是已知的。因此，训练样本的岐义性低。

无监督学习：对没有概念标记（分类）的训练样本进行学习，以发现训练样本集中的结构性知识。这里，所有的标记（分类）是未知的。因此，训练样本的岐义性高。聚类就是典型的无监督学习。

·          SVM 的推导，特性？多分类怎么处理？

　　SVM是最大间隔分类器，几何间隔和样本的误分次数之间存在关系，IMG_256，其中IMG_257

从线性可分情况下，原问题，特征转换后的dual问题，引入kernel(线性kernel,多项式，高斯),最后是soft margin。

线性：简单，速度快，但是需要线性可分

多项式：比线性核拟合程度更强，知道具体的维度，但是高次容易出现数值不稳定，参数选择比较多。

高斯：拟合能力最强，但是要注意过拟合问题。不过只有一个参数需要调整。

多分类问题，一般将二分类推广到多分类的方式有三种，一对一，一对多，多对多。

一对一：将N个类别两两配对，产生N(N-1)/2个二分类任务，测试阶段新样本同时交给所有的分类器，最终结果通过投票产生。

一对多：每一次将一个例作为正例，其他的作为反例，训练N个分类器，测试时如果只有一个分类器预测为正类，则对应类别为最终结果，如果有多个，则一般选择置信度最大的。从分类器角度一对一更多，但是每一次都只用了2个类别，因此当类别数很多的时候一对一开销通常更小(只要训练复杂度高于O(N)即可得到此结果)。

多对多：若干各类作为正类，若干个类作为反类。注意正反类必须特殊的设计。

·          LR 的推导，特性？

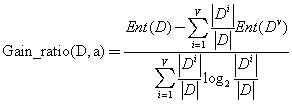
LR的优点在于实现简单，并且计算量非常小，速度很快，存储资源低，缺点就是因为模型简单，对于复杂的情况下会出现欠拟合，并且只能处理2分类问题(可以通过一般的二元转换为多元或者用softmax回归)。

·          决策树的特性？

决策树基于树结构进行决策，与人类在面临问题的时候处理机制十分类似。其特点在于需要选择一个属性进行分支，在分支的过程中选择信息增益最大的属性，定义如下

IMG_258

在划分中我们希望决策树的分支节点所包含的样本属于同一类别，即节点的纯度越来越高。决策树计算量简单，可解释性强，比较适合处理有缺失属性值的样本，能够处理不相关的特征，但是容易过拟合，需要使用剪枝或者随机森林。信息增益是熵减去条件熵，代表信息不确定性较少的程度，信息增益越大，说明不确定性降低的越大，因此说明该特征对分类来说很重要。由于信息增益准则会对数目较多的属性有所偏好，因此一般用信息增益率(c4.5)



其中分母可以看作为属性自身的熵。取值可能性越多，属性的熵越大。

Cart决策树使用基尼指数来选择划分属性，直观的来说，Gini(D)反映了从数据集D中随机抽取两个样本，其类别标记不一致的概率，因此基尼指数越小数据集D的纯度越高，一般为了防止过拟合要进行剪枝，有预剪枝和后剪枝，一般用cross validation集进行剪枝。

连续值和缺失值的处理，对于连续属性a,将a在D上出现的不同的取值进行排序，基于划分点t将D分为两个子集。一般对每一个连续的两个取值的中点作为划分点，然后根据信息增益选择最大的。与离散属性不同，若当前节点划分属性为连续属性，该属性还可以作为其后代的划分属性。

·          SVM、LR、决策树的对比？

SVM既可以用于分类问题，也可以用于回归问题，并且可以通过核函数快速的计算，LR实现简单，训练速度非常快，但是模型较为简单，决策树容易过拟合，需要进行剪枝等。从优化函数上看，soft margin的SVM用的是hinge loss,而带L2正则化的LR对应的是cross entropy loss，另外adaboost对应的是exponential loss。所以LR对远点敏感，但是SVM对outlier不太敏感，因为只关心support vector，SVM可以将特征映射到无穷维空间，但是LR不可以，一般小数据中SVM比LR更优一点，但是LR可以预测概率，而SVM不可以，SVM依赖于数据测度，需要先做归一化，LR一般不需要，对于大量的数据LR使用更加广泛，LR向多分类的扩展更加直接，对于类别不平衡SVM一般用权重解决，即目标函数中对正负样本代价函数不同，LR可以用一般的方法，也可以直接对最后结果调整(通过阈值)，一般小数据下样本维度比较高的时候SVM效果要更优一些。

·          GBDT 和随机森林的区别？

随机森林采用的是bagging的思想，bagging又称为bootstrap aggreagation，通过在训练样本集中进行有放回的采样得到多个采样集，基于每个采样集训练出一个基学习器，再将基学习器结合。随机森林在对决策树进行bagging的基础上，在决策树的训练过程中引入了随机属性选择。传统决策树在选择划分属性的时候是在当前节点属性集合中选择最优属性，而随机森林则是对结点先随机选择包含k个属性的子集，再选择最有属性，k作为一个参数控制了随机性的引入程度。

另外，GBDT训练是基于Boosting思想，每一迭代中根据错误更新样本权重，因此是串行生成的序列化方法，而随机森林是bagging的思想，因此是并行化方法。

·          如何判断函数凸或非凸？什么是凸优化？

首先定义凸集，如果x,y属于某个集合C,并且所有的IMG_260也属于c,那么c为一个凸集，进一步，如果一个函数其定义域是凸集，并且

IMG_261

则该函数为凸函数。上述条件还能推出更一般的结果，

IMG_262

如果函数有二阶导数，那么如果函数二阶导数为正，或者对于多元函数，Hessian矩阵半正定则为凸函数。

(也可能引到SVM，或者凸函数局部最优也是全局最优的证明，或者上述公式期望情况下的Jessen不等式)

·          如何解决类别不平衡问题？

有些情况下训练集中的样本分布很不平衡，例如在肿瘤检测等问题中，正样本的个数往往非常的少。从线性分类器的角度，在用IMG_263对新样本进行分类的时候，事实上在用预测出的y值和一个y值进行比较，例如常常在y>0.5的时候判为正例，否则判为反例。几率IMG_264反映了正例可能性和反例可能性的比值，阈值0.5恰好表明分类器认为正反的可能性相同。在样本不均衡的情况下，应该是分类器的预测几率高于观测几率就判断为正例，因此应该是IMG_265时预测为正例，这种策略称为rebalancing。但是训练集并不一定是真实样本总体的无偏采样，通常有三种做法，一种是对训练集的负样本进行欠采样，第二种是对正例进行升采样，第三种是直接基于原始训练集进行学习，在预测的时候再改变阈值，称为阈值移动。注意过采样一般通过对训练集的正例进行插值产生额外的正例，而欠采样将反例划分为不同的集合供不同的学习器使用。

·          解释对偶的概念。

一个优化问题可以从两个角度进行考察，一个是primal 问题，一个是dual 问题，就是对偶问题，一般情况下对偶问题给出主问题最优值的下界，在强对偶性成立的情况下由对偶问题可以得到主问题的最优下界，对偶问题是凸优化问题，可以进行较好的求解，SVM中就是将primal问题转换为dual问题进行求解，从而进一步引入核函数的思想。

·          如何进行特征选择？

特征选择是一个重要的数据预处理过程，主要有两个原因，首先在现实任务中我们会遇到维数灾难的问题(样本密度非常稀疏)，若能从中选择一部分特征，那么这个问题能大大缓解，另外就是去除不相关特征会降低学习任务的难度，增加模型的泛化能力。冗余特征指该特征包含的信息可以从其他特征中推演出来，但是这并不代表该冗余特征一定没有作用，例如在欠拟合的情况下也可以用过加入冗余特征，增加简单模型的复杂度。

在理论上如果没有任何领域知识作为先验假设那么只能遍历所有可能的子集。但是这显然是不可能的，因为需要遍历的数量是组合爆炸的。一般我们分为子集搜索和子集评价两个过程，子集搜索一般采用贪心算法，每一轮从候选特征中添加或者删除，分别成为前向和后先搜索。或者两者结合的双向搜索。子集评价一般采用信息增益，对于连续数据往往排序之后选择中点作为分割点。

常见的特征选择方式有过滤式，包裹式和嵌入式，filter,wrapper和embedding。Filter类型先对数据集进行特征选择，再训练学习器。Wrapper直接把最终学习器的性能作为特征子集的评价准则，一般通过不断候选子集，然后利用cross-validation过程更新候选特征，通常计算量比较大。嵌入式特征选择将特征选择过程和训练过程融为了一体，在训练过程中自动进行了特征选择，例如L1正则化更易于获得稀疏解，而L2正则化更不容易过拟合。L1正则化可以通过PGD,近端梯度下降进行求解。

·          为什么会产生过拟合，有哪些方法可以预防或克服过拟合？

一般在机器学习中，将学习器在训练集上的误差称为训练误差或者经验误差，在新样本上的误差称为泛化误差。显然我们希望得到泛化误差小的学习器，但是我们事先并不知道新样本，因此实际上往往努力使经验误差最小化。然而，当学习器将训练样本学的太好的时候，往往可能把训练样本自身的特点当做了潜在样本具有的一般性质。这样就会导致泛化性能下降，称之为过拟合，相反，欠拟合一般指对训练样本的一般性质尚未学习好，在训练集上仍然有较大的误差。

欠拟合：一般来说欠拟合更容易解决一些，例如增加模型的复杂度，增加决策树中的分支，增加神经网络中的训练次数等等。

过拟合：一般认为过拟合是无法彻底避免的，因为机器学习面临的问题一般是np-hard,但是一个有效的解一定要在多项式内可以工作，所以会牺牲一些泛化能力。过拟合的解决方案一般有增加样本数量，对样本进行降维，降低模型复杂度，利用先验知识(L1,L2正则化)，利用cross-validation，early stopping等等。

·          什么是偏差与方差？

泛化误差可以分解成偏差的平方加上方差加上噪声。偏差度量了学习算法的期望预测和真实结果的偏离程度，刻画了学习算法本身的拟合能力，方差度量了同样大小的训练集的变动所导致的学习性能的变化，刻画了数据扰动所造成的影响，噪声表达了当前任务上任何学习算法所能达到的期望泛化误差下界，刻画了问题本身的难度。偏差和方差一般称为bias和variance，一般训练程度越强，偏差越小，方差越大，泛化误差一般在中间有一个最小值，如果偏差较大，方差较小，此时一般称为欠拟合，而偏差较小，方差较大称为过拟合。

偏差：IMG_266

方差：IMG_267

·          神经网络的原理，如何进行训练？

神经网络自发展以来已经是一个非常庞大的学科，一般而言认为神经网络是由单个的神经元和不同神经元之间的连接构成，不够的结构构成不同的神经网络。最常见的神经网络一般称为多层前馈神经网络，除了输入和输出层，中间隐藏层的个数被称为神经网络的层数。BP算法是训练神经网络中最著名的算法，其本质是梯度下降和链式法则。

·          介绍卷积神经网络，和 DBN 有什么区别？

卷积神经网络的特点是卷积核，CNN中使用了权共享，通过不断的上采用和卷积得到不同的特征表示，采样层又称为pooling层，基于局部相关性原理进行亚采样，在减少数据量的同时保持有用的信息。DBN是深度信念网络，每一层是一个RBM，整个网络可以视为RBM堆叠得到，通常使用无监督逐层训练，从第一层开始，每一层利用上一层的输入进行训练，等各层训练结束之后再利用BP算法对整个网络进行训练。

·          采用 EM 算法求解的模型有哪些，为什么不用牛顿法或梯度下降法？

用EM算法求解的模型一般有GMM或者协同过滤，k-means其实也属于EM。EM算法一定会收敛，但是可能收敛到局部最优。由于求和的项数将随着隐变量的数目指数上升，会给梯度计算带来麻烦。

·          用 EM 算法推导解释 Kmeans。

k-means算法是高斯混合聚类在混合成分方差相等，且每个样本仅指派一个混合成分时候的特例。注意k-means在运行之前需要进行归一化处理，不然可能会因为样本在某些维度上过大导致距离计算失效。k-means中每个样本所属的类就可以看成是一个隐变量，在E步中，我们固定每个类的中心，通过对每一个样本选择最近的类优化目标函数，在M步，重新更新每个类的中心点，该步骤可以通过对目标函数求导实现，最终可得新的类中心就是类中样本的均值。

·          用过哪些聚类算法，解释密度聚类算法。

k-means算法，聚类性能的度量一般分为两类，一类是聚类结果与某个参考模型比较(外部指标)，另外是直接考察聚类结果(内部指标)。后者通常有DB指数和DI，DB指数是对每个类，找出类内平均距离/类间中心距离最大的类，然后计算上述值，并对所有的类求和，越小越好。类似k-means的算法仅在类中数据构成簇的情况下表现较好，密度聚类算法从样本密度的角度考察样本之间的可连接性，并基于可连接样本不断扩展聚类蔟得到最终结果。DBSCAN(density-based spatial clustering of applications with noise)是一种著名的密度聚类算法，基于一组邻域参数IMG_268进行刻画，包括IMG_269邻域，核心对象(邻域内至少包含IMG_270个对象)，密度直达(j由i密度直达，表示j在i的邻域内，且i是一个核心对象)，密度可达(j由i密度可达，存在样本序列使得每一对都密度直达)，密度相连(xi,xj存在k,i,j均有k可达)，先找出样本中所有的核心对象，然后以任一核心对象作为出发点，找出由其密度可达的样本生成聚类蔟，直到所有核心对象被访问过为止。

·          聚类算法中的距离度量有哪些？

聚类算法中的距离度量一般用闽科夫斯基距离，在p取不同的值下对应不同的距离，例如p=1的时候对应曼哈顿距离，p=2的情况下对应欧式距离，p=inf的情况下变为切比雪夫距离，还有jaccard距离，幂距离(闽科夫斯基的更一般形式),余弦相似度，加权的距离，马氏距离(类似加权)作为距离度量需要满足非负性，同一性，对称性和直递性，闽科夫斯基在p>=1的时候满足读来那个性质，对于一些离散属性例如{飞机，火车，轮船}则不能直接在属性值上计算距离，这些称为无序属性，可以用VDM(Value Diffrence Metrix)，属性u上两个离散值a,b之间的VDM距离定义为

IMG_271

其中IMG_272表示在第i个簇中属性u上a的样本数，样本空间中不同属性的重要性不同的时候可以采用加权距离，一般如果认为所有属性重要性相同则要对特征进行归一化。一般来说距离需要的是相似性度量，距离越大，相似度越小，用于相似性度量的距离未必一定要满足距离度量的所有性质，例如直递性。比如人马和人，人马和马的距离较近，然后人和马的距离可能就很远。

·          解释贝叶斯公式和朴素贝叶斯分类。

贝叶斯公式，IMG_273，最小化分类错误的贝叶斯最优分类器等价于最大化后验概率。

基于贝叶斯公式来估计后验概率的主要困难在于，条件概率IMG_274是所有属性上的联合概率，难以从有限的训练样本直接估计得到。朴素贝叶斯分类器采用了属性条件独立性假设，对于已知的类别，假设所有属性相互独立。这样，朴素贝叶斯分类则定义为

IMG_275

如果有足够多的独立同分布样本，那么IMG_276可以根据每个类中的样本数量直接估计出来。在离散情况下先验概率可以利用样本数量估计或者离散情况下根据假设的概率密度函数进行最大似然估计。朴素贝叶斯可以用于同时包含连续变量和离散变量的情况。如果直接基于出现的次数进行估计，会出现一项为0而乘积为0的情况，所以一般会用一些平滑的方法，例如拉普拉斯修正，

IMG_277

这样既可以保证概率的归一化，同时还能避免上述出现的现象。

·          L1和L2正则化的作用，解释。

L1正则化是在代价函数后面加上IMG_278，L2正则化是在代价函数后面增加了IMG_279，两者都起到一定的过拟合作用，两者都对应一定的先验知识，L1对应拉普拉斯分布，L2对应高斯分布，L1偏向于参数稀疏性，L2偏向于参数分布较为稠密。

* TF-IDF是什么？

　　TF指Term frequecy,代表词频,IDF代表inverse document frequency,叫做逆文档频率，这个算法可以用来提取文档的关键词，首先一般认为在文章中出现次数较多的词是关键词，词频就代表了这一项，然而有些词是停用词，例如的，是，有这种大量出现的词，首先需要进行过滤，比如过滤之后再统计词频出现了中国，蜜蜂，养殖且三个词的词频几乎一致，但是中国这个词出现在其他文章的概率比其他两个词要高不少，因此我们应该认为后两个词更能表现文章的主题，IDF就代表了这样的信息，计算该值需要一个语料库，如果一个词在语料库中出现的概率越小，那么该词的IDF应该越大，一般来说TF计算公式为(某个词在文章中出现次数/文章的总词数)，这样消除长文章中词出现次数多的影响，IDF计算公式为log(语料库文章总数/(包含该词的文章数)+1)。将两者乘乘起来就得到了词的TF-IDF。传统的TF-IDF对词出现的位置没有进行考虑，可以针对不同位置赋予不同的权重进行修正，注意这些修正之所以是有效的，正是因为人观测过了大量的信息，因此建议了一个先验估计，人将这个先验估计融合到了算法里面，所以使算法更加的有效。

* 文本中的余弦距离是什么，有哪些作用？

　　余弦距离是两个向量的距离的一种度量方式，其值在-1~1之间，如果为1表示两个向量同相，0表示两个向量正交，-1表示两个向量反向。使用TF-IDF和余弦距离可以寻找内容相似的文章，例如首先用TF-IDF找出两篇文章的关键词，然后每个文章分别取出k个关键词(10-20个)，统计这些关键词的词频，生成两篇文章的词频向量，然后用余弦距离计算其相似度。