# 机器学习测试题

1. **什么是偏倚（bias）、方差（variable）均衡？**

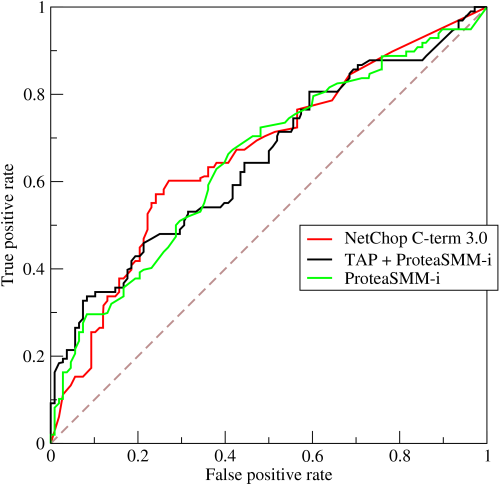
偏倚指的是模型预测值与真实值的差异，是由使用的学习算法的某些错误或过于简单的假设造成的误差。它会导致模型欠拟合，很难有高的预测准确率。  
方差指的是不同训练数据训练的模型的预测值之间的差异，它是由于使用的算法模型过于复杂，导致对训练数据的变化十分敏感，这样会导致模型过拟合，使得模型带入了过多的噪音。  
任何算法的学习误差都可以分解成偏倚、方差和噪音导致的固定误差。模型越复杂，会降低偏倚增加方差。为了降低整体的误差，我们需要对偏倚方差均衡，使得模型中不会由高偏倚或高方差。

1. **监督学习和非监督学习有什么不同？**

监督学习需要具有标签（label）的训练数据，比如做分类，你需要先对训练数据做标记，然后才能训练模型将数据分成你说需要的标记类。  
而非监督学习则不需要。

1. **KNN和k-means聚类由什么不同？**

k-Nearest Neighbors 是一种监督学习算法，而k-means 是非监督的。这两种算法看起来很相似，都需要计算样本之间的距离。knn算法需要事先已有标注好的数据，当你需要对未标注的数据进行分类时，统计它附近最近的k个样本，将其划分为样本数最多的类别中。k-means聚类只需要一些未分类的数据点和阀值，算法会逐渐将样本点进行分成族类。

1. **解释一下ROC曲线的原理**  
     
   ROC曲线是真正率和假正率在不同的阀值下之间的图形表示关系。通常用作权衡模型的敏感度与模型对一个错误分类报警的概率。  
   真正率表示：表示正的样本被预测为正占所有正样本的比例。  
   假正率表示：表示负的样本被预测为正占所有负样本的比例。  
   （0，0）点表示所有样本都被预测为负，此时阀值很大。  
   （1，1）点表示所有样本都被预测为正，此时阀值很小。

5. **定义一下prediction精准率、recall召回率**  
（准确率是accuracy）  
召回率就是Q4中的真正率。  
精准率指的是：正样本被预测为正所占所有预测为正样本数的比例

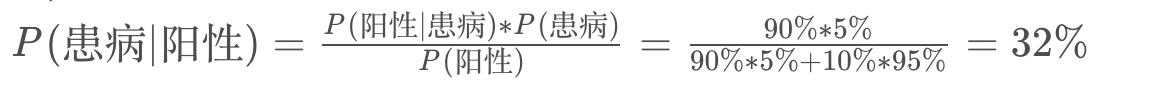
| **-** | **预测正** | **预测负** |
| --- | --- | --- |
| 真实正 | TP | FN |
| 真实负 | FP | TN |

recall：$\frac{TP}{TP+FN} $ precision: T P T P + F P \frac{TP}{TP+FP}*TP*+*FPTP*​

1. **什么是贝叶斯定理，它是如何使用在机器学习中的？**

贝叶斯定理会根据一件事发生的先验知识告诉你它后验概率。数学上，它表示为：一个条件样本发生的真正率占真正率和假正率之和的比例，即：  
文本

描述已自动生成

举个例子： 已知某疾病的患病概率为5%，现用某检验方法进行诊断，若患有该病，则有90%的几率检验结果呈阳性。但即使正常人使用该检验方法，也有10%的几率误诊而呈阳性。某人检验结果为阳性，求此人患病的概率。  


1. **为什么我们要称“朴素“贝叶斯？**

因为我们在用到它的时候，有一个很强的假设，现实数据中几乎不会出现的：我们假设特征之间是相互独立，也就是我们计算条件概率时可以简化成它的组件的条件概率乘积。

1. **L1、L2正则之间有什么不同？**

L2正则 对应的是加入2范数，使得对权重进行衰减，从而达到惩罚损失函数的目的，防止模型过拟合。保留显著减小损失函数方向上的权重，而对于那些对函数值影响不大的权重使其衰减接近于0。相当于加入一个gaussian prior。  
L1正则 对应得失加入1范数，同样可以防止过拟合。它会产生更稀疏的解，即会使得部分权重变为0，达到特征选择的效果。相当于加入了一个laplacean prior。

1. **你最喜欢的算法是什么？把它解释一下。**

这里我比较想说的是SVM，因为它的数学理论让我觉得很有意思，而且应用广泛，效果不错。先从线性可分讲起，然后是最大间隔原理。什么是支持向量？如何进行常数估计。转化成优化问题，对偶问题，kkt条件，拉格朗日方法求最值等。然后是非线性可分情况，软间隔，进行坐标变化。引入核函数。常见的：多项式核函数、指数核函数、高斯核函数。

1. **第一类误差和第二类误差有什么区别？**

第一类误差指的是假正率，第二类指的是假负率。简单来说，第一类误差意味着假设为真的情况下，作出了拒绝原假设的一种错误推断。第二类误差意味着假设为假的情况下，做出了接受原假设的一种错误判断。  
举个例子：第一类误差，你告诉一个男的他怀孕了。第二类误差，你告诉一个已经怀孕的女子，她没怀孕。

1. **什么是傅立叶变换？**

傅立叶变换指：一个满足某些条件的函数可以表示成三角函数或他们的积分形式的线性组合。

1. **概率和似然有什么区别？**

概率和似然都是指可能性，但在统计学中，概率和似然有截然不同的用法。概率描述了已知参数时的随机变量的输出结果；似然则用来描述已知随机变量输出结果时，未知参数的可能取值。例如，对于“一枚正反对称的硬币上抛十次”这种事件，我们可以问硬币落地时十次都是正面向上的“概率”是多少；而对于“一枚硬币上抛十次，我们则可以问，这枚硬币正反面对称的“似然”程度是多少。  
概率(密度)表达给定θ下样本随机向量X=x的可能性，而似然表达了给定样本X=x下参数θ1(相对于另外的参数θ2)为真实值的可能性。我们总是对随机变量的取值谈概率，而在非贝叶斯统计的角度下，参数是一个实数而非随机变量，所以我们一般不谈一个参数的概率，而说似然。

1. **什么是深度学习，它与机器学习算法之间有什么联系？**

深度学习是机器学习的一个子领域，它关心的是参照神经学科的理论构建神经网络，使用反向传播对大量未标注或半结构化的数据进行建模。从这个角度看，深度学习可以看成一种非监督学习算法，通过使用神经网络学习数据的表示。

1. **生成模型与判别模型有什么区别？**

生成模型会学习数据的分布；判别模型学习的是不同类型数据之间的区别，不学习数据内部特点。在分类问题上，判别模型会优于生成模型。  
判别模型求解的思路是：条件分布------>模型参数后验概率最大------->（似然函数\cdot 参数先验）最大------->最大似然  
生成模型的求解思路是：联合分布------->求解类别先验概率和类别条件概率  
常见的生成方法有混合高斯模型、朴素贝叶斯法和隐形马尔科夫模型等，常见的判别方法有SVM、LR等

1. **交叉检验如何用在时间序列数据上？**

与标准的k-folds 交叉检验不同，数据不是随机分布的，而是具有时序性的。如果模式出现在后期，模型仍然需要选择先前时间的数据，尽管前期对模式无影响。我们可以如下这么做：  
fold1：training[1], test[2]  
fold2：training[1 2], test[3]  
fold3：training[1 2 3], test[4]  
fold4：training[1 2 3 4], test[5]  
fold5：training[1 2 3 4 5], test[6]

1. **如何对决策树进行剪枝？**

剪枝是决策树发生过拟合后，为了降低模型复杂度，提高模型准确率的一种做法。可以分为自上而下和自下而上两种。常见的方法有：误差降低剪枝（REP）和代价复杂度剪枝（CCP）。  
REP简单的来说就是对树的每一个结点进行剪枝，如果剪掉某个结点不会降低模型准确率，那么将其剪掉。这种启发式的做法实际上就是为了最大化准确率。

1. **模型的精度和模型的性能哪个对你更重要？**

许多机器学习的模型会有高的精度，但是预测能力也就是泛化能力较低，如何去理解它们呢？  
精度只是模型性能的一部分，有可能是会产生误导的那个。对于具有倾斜的数据集，比如要从大量的金融数据中识别出少量的诈骗数据，一个精度高的模型可能会告诉你没有诈骗，然而这样的模型预测是没有意义的。所以，不要总是把精度当作模型最重要的事。

1. **什么是F1数，怎么使用它？**

F1数是衡量模型性能的一个指标。它是模型精准率和召回率的加权平均，1表示最好，0表示最差。在分类问题中有时精准率和召回率不会同时都高，那么我们可以使用F1数。

1. **如何处理一个不平衡的数据集？**

不平衡的数据集：比如二分类问题中，一类数据有90%，而另一类只有10%。我们可以轻易的得到90%准确率的模型，但是它对第二类的预测值为0。那么我们需要对这样的数据进行处理：

1. 收集更多的数据，使其达到平衡
2. 使用重复采样
3. 使用不同的算法

重要的是：你注意到了数据的不平衡导致的问题，以及如何去解决它们。

1. **什么时候你应该使用分类而不是回归？**

分类会产生离散的数值，使得数据严格的分为不同类。回归会得到连续的值，使你更好的区分独立点之间的区别。当你需要知道你的数据明确的属于那些类时你可以用分类。

1. **举个例子，说明使用集成学习会很有用。**

集成学习通过组合一些基学习算法来优化得到更好的预测性能，通常可以防止模型的过拟合使模型更具有鲁棒性。  
你可以列举一些集成学习的例子，如bagging、boosting、stacking等，并且了解他们是如何增加模型预测能力的。

1. **你如何确保你的模型没有过拟合？**

过度拟合的训练数据以及数据携带的噪音，对于测试数据会带来不确定的推测。有如下三种方法避免过拟合：

1. 保持模型尽可能地简单：通过考量较少的变量和参数来减少方差，达到数据中消除部分噪音的效果。
2. 使用交叉检验的手段如：k-folds cross-validation。
3. 使用正则化的技术如：LASSO方法来惩罚模型中可能导致过拟合的参数。
4. **如何评估你的机器学习模型的有效性？**

首先你需要将数据分成训练集和测试集，或者使用给交叉验证方法分割。然后你需要选择度量模型表现的metrics，如F1数、准确率、混淆矩阵等。更重要的是，根据实际情况你需要理解模型度量的轻微差别，以便于选择正确的度量标准。

1. **如何评估一个LR model？**

23题的一个子问题。首先你需要知道LR的目标是什么？（分类或预测）然后通过举一些例子来说明。

1. **什么是核技巧，有什么用处？**

核技巧使用核函数，确保在高维空间不需要明确计算点的坐标，而是计算数据的特征空间中的内积。这使其具有一个很有用的属性：更容易的计算高维空间中点的坐标。许多算法都可以表示称这样的内积形式，使用核技巧可以保证低维数据在高维空间中运用算法进行计算。

1. **怎么理解Dropout？**

Dropout 是深度学习中的一种正则化方法，指在深度学习网络的训练过程中，对于神经网络单元，按照一定的概率将其暂时从网络中丢弃。注意是暂时，对于随机梯度下降来说，由于是随机丢弃，故而每一个mini-batch都在训练不同的网络。由于大规模神经网络具有两个显著缺点：计算费时和过拟合。问了解决过拟合，一般采用essenble方法，即训练多个模型进行组合，但是这对于大型神经网络是不可取的。Dropout可以被看作是集成大量深度神经网络的使用Bagging方法，比如对于具有N个结点的网络，使用dropout随机进行丢弃后，可以看成2n2n个“瘦小“的神经网络的集合，达到了一种廉价的Bagging集成近似，但是训练的参数却是不变的。从而摆脱了费事的问题，能够方便且功能强大的训练和评估指数级数量的神经网络。一个关于dropout的重要见解是，通过随机行为训练网络并平均多个随机决定进行预测，实现了一种参数共享的Bagging形式。这种参数共享不一定基于包括和排除。原则上，任何一种随机的修改都是可以接受的。在实践中，我们必须选择让神经网络能够学习对抗的修改类型。  
更进一步的观点是：dropout不仅仅是训练一个bagging的集成模型，而且是共享隐藏单元的集成模型。受生物学想法启发：有性繁殖涉及两个不同生物体之间交换基因，进化产生的压力使得基因不仅是良好的，而且要准备好不同有机体之间的交换，这样的基因和这些特点对环境的变化是非常稳健的，因为它们一定会正确适应任何一个有机体或模型不寻常的特性。因此Dropout正则化每个隐藏单元不仅是一个很好的特征，要在很多情况下是良好的特征，dropout会强迫一个神经单元，和随机挑选出来的其他神经单元共同工作，达到好的效果。消除减弱了神经元节点间的联合适应性，增强了泛化能力。  
另外还有一种观点是：Dropout是施加到隐藏单元的掩码噪音。就是对于每一个dropout后的网络，进行训练时，相当于做了Data Augmentation，因为，总可以找到一个样本，使得在原始的网络上也能达到dropout单元后的效果。 比如，对于某一层，dropout一些单元后，形成的结果是(1.5,0,2.5,0,1,2,0)，其中0是被drop的单元，那么总能找到一个样本，使得结果也是如此。这样，每一次dropout其实都相当于增加了样本。

1. **为什么random forest具有特征选择的功能？**

首先我们知道，决策树是一种易解释的模型，它可以对特征的重要性进行排序，而随机森林是由决策树构建而成的，一定程度上了揭示了它也是是由特征选择的功能的。那么随机森林如何对特征进行选取呢？方法如下：

* 平均不纯度减少 mean decrease impurity：随机森林由多个决策树构成。决策树中的每一个节点都是关于某个特征的条件，为的是将数据集按照不同的响应变量一分为二。利用不纯度可以确定节点（最优条件），对于分类问题，通常采用基尼不纯度或者信息增益，对于回归问题，通常采用的是方差或者最小二乘拟合。当训练决策树的时候，可以计算出每个特征减少了多少树的不纯度。对于一个决策树森林来说，可以算出每个特征平均减少了多少不纯度，并把它平均减少的不纯度作为特征选择的值。
* 平均精确率减少 Mean decrease accuracy：度量每个特征对模型精确率的影响。主要思路是打乱每个特征的特征值顺序，并且度量顺序变动对模型的精确率的影响。很明显，对于不重要的变量来说，打乱顺序对模型的精确率影响不会太大，但是对于重要的变量来说，打乱顺序就会降低模型的精确率。

1. **random forest有哪些重要的参数？**

重要的参数有：构建随机森林的决策树的数量；每棵树的最大深度；每棵树包含的最大特征数。随机森林通过对样本随机采样，随机选取特征来构建决策树，通过k棵决策树组合成最终的分类器。

1. **KNN和LR有什么本质区别？**

KNN与LR的本质区别有：

1. knn是惰性学习算法，不会去自主学习特征权重。它没有训练过程，基本原理就是找到训练数据集里面离需要预测的样本点距离最近的k个值（距离可以使用比如欧式距离，k的值需要自己调参），然后把这k个点的label做个投票，选出一个label做为预测。这是与LR最本质的区别。
2. knn是基于距离的，LR基于概率。所以knn需要事先对数据进行归一化，而LR对数据没有太多约束。
3. LR适用与高维稀疏数据，而knn对于大型数据的预测结果很糟糕。

下面，我们看看LR与SVM的相同与不同处。

相同处：

* 都是线性分类算法；
* 都属于判别模型；
* 都是监督学习算法；
* 都适用于高维数据，在工业界都得到了广泛使用。

不同处：

* 本质上，损失函数不同。LR是log损失函数，SVM是Hinge损失函数，这是0-1损失函数的一种变体。不同的loss function代表了不同的假设前提，也就代表了不同的分类原理。简单来说，​逻辑回归方法基于概率理论，假设样本为1的概率可以用sigmoid函数来表示，然后通过极大似然估计的方法估计出参数的值。支持向量机​基于几何间隔最大化原理，认为存在最大几何间隔的分类面为最优分类面。
* 支持向量机只考虑局部的边界线附近的点，而逻辑回归考虑全局。SVM决策面的样本点只有少数的结构支持向量，当在支持向量外添加或减少任何样本点对分类决策面没有任何影响；而在LR中，每个样本点都会或多或少影响决策面的结果。因此，线性SVM不直接依赖于数据分布，分类平面不受一类点影响；LR则受所有数据点的影响，如果数据不同类别strongly unbalance，一般需要先对数据做balancing。
* 在解决非线性问题时，支持向量机采用核函数的机制，而LR通常不采用核函数的方法。分类模型的结果就是计算决策面，模型训练的过程就是决策面的计算过程。通过上面的第二点不同点可以了解，在计算决策面时，SVM算法里只有少数几个代表支持向量的样本参与了计算，也就是只有少数几个样本需要参与核计算（即kernal machine解的系数是稀疏的）。然而，LR算法里，每个样本点都必须参与决策面的计算过程，也就是说，假设我们在LR里也运用核函数的原理，那么每个样本点都必须参与核计算，这带来的计算复杂度是相当高的。所以，在具体应用时，LR很少运用核函数机制。​
* ​线性SVM依赖数据表达的距离测度，所以需要对数据先做normalization，LR不受其影响。一个基于概率，一个基于距离！同时会有：feature scaling会使得gradient descent的收敛更好。
* SVM的损失函数就自带正则,损失函数中的1/2||w||21/2||w||2项，这就是为什么SVM是结构风险最小化算法的原因. 而LR必须另外在损失函数上添加正则项！因此，两个模型对参数的敏感程度不同，Linear SVM比较依赖penalty的系数和数据表达空间的测度，而（带正则项的）LR比较依赖对参数做L1 regularization的系数。

1. **stacking是什么？需要注意哪些问题？**

stacking 是一种不同于bagging、boosting的组合模型算法，重点在于对不同学习器的组合。具体过程如下：

* 划分数据集为两个不相交的集合 A 与 B
* 在集合A上训练m个学习器
* 在集合B上测试这m个学习器
* 将上述预测结果作为输入，把正确的label作为输出，训练一个高层学习器

stcking属于集成学习 ensemble learning 的一种。  
集成学习是将多个学习算法组合在一起来获得更好效果的一种方法。  
机器学习中，监督学习可以描述成：对于一个具体问题，从一堆假设空间中搜索一个具有较好且相对稳定预测效果的模型。  
即使假设空间中包含很好的假设，有时我们也难以找出。集成方法通过组合多个假设以期望得到一个更优的假设，也就是说通过组合多个弱学习器以构成一个强学习器。其中组合的弱学习模型可以是相同类型的，也可以是不同的。  
集成方法对大量数据和不充分数据都有很好的效果，对于大数据，将其分成若干小数据集，分别进行训练，最后组合在一起。如果数据量少，可以通过boostrap进行抽象，得到多个数据集，分别训练后再组合。  
集成方法用于评估测试的时候，需要更多的计算。可以认为集成学习使用计算换预测效果。同时这也造成了模型中的参数相比单一模型包含的信息较少，导致了太多的冗余。