**六、机器学习**

机器学习是数据分析师需要掌握的技能之一，但是机器学习掌握到何种程度，根据不同的数分岗位也有不同的要求。

如：纯业务型数据分析师（BI）只需要大致了解机器学习的基本原理和运用。技术型数据分析师（包括数据挖掘、数据科学等）需要较好的掌握机器学习原理、部分数学公式推导、模型之间的差异、建模方式以及实操等。

由于机器学习的知识点太过繁杂且专业化。每个模型都涉及到专业的推导，因此在知识点中大范围讲解模型原理并不合适。

本章会列举出数据分析求职面试时所有需要掌握的机器学习模型，做简单的介绍，并附上详细介绍的网址或电子书，同学们通过本章了解面试中掌握哪些模型，再分别进行学习即可。

**核心知识点：**

1. **机器学习基础概念**
2. **回归与分类**

机器学习的本质是用自变量（x）去拟合因变量(y)。回归与分类的区别在于因变量y是否连续。因变量是连续值，如预测上海未来的房价是多少，是回归问题。因变量是离散值，如预测上海未来的房价是否会涨，是分类问题。

1. **训练集、测试集、验证集**

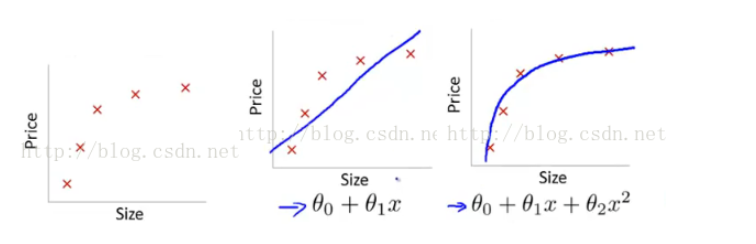
一个形象的比喻：将模型训练类比于我们参加高考。平日的课后习题是训练集，我们从课后习题中学习考试的知识。模拟考试是验证集，我们通过模考判断自己大致的成绩。最好的高考是测试集，直接判断我们自己学的好坏。

**一般将模型分成三份，训练集直接训练模型，验证集来初步判断模型的效果，根据验证集的结果调整模型中的参数，最后测试集来判断模型好坏**。

* 训练集（train set）：用于模型拟合的数据样本。
* 验证集（validation set）：是模型训练过程中单独留出的样本集，它可以用于调整模型的
* 测试集（test set）:用来评估模最终模型的泛化能力。但不能作为调参、选择特征等算法相关的选择的依据。

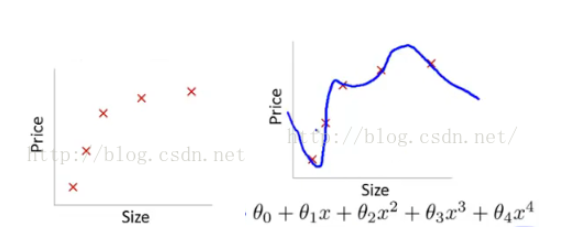
1. **过拟合与欠拟合**

* 欠拟合指模型没有很好地捕捉到数据特征，不能够很好地拟合数据



左图表示size与prize关系的数据，中间的图就是出现欠拟合的模型，不能够很好地拟合数据，如果在中间的图的模型后面再加一个二次项，就可以很好地拟合图中的数据了，如右面的图所示。

* 过拟合是指模型把数据学习的太彻底，以至于把噪声数据的特征也学习到了。过拟合在训练集上表现效果很好，但在验证集、测试集上效果较差。

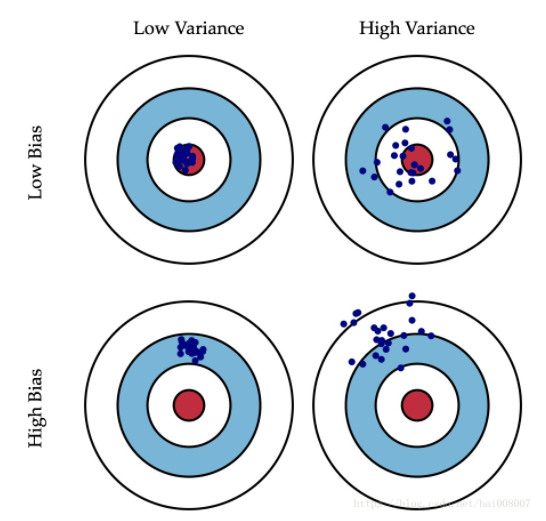


上面左图表示size和prize的关系，我们学习到的模型曲线如右图所示，虽然在训练的时候模型可以很好地匹配数据，但是很显然过度扭曲了曲线，不是真实的size与prize曲线。

1. **偏差与方差**

任何机器学习模型预测的结果都存在误差。而误差可以分为偏差与方差两种。

**偏差：**度量学习算法的期望预测与真实结果的偏离程度，也叫拟合能力，**判断模型本身的预测结果是否准。  
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，即刻画了数据扰动造成的影响**，判断模型多次训练的结果是否稳定。**



偏差与方差是冲突的，即同一个模型的偏差减少、方差增多，方差减少、偏差增多。这称为偏差-方差窘境。假定我们能够控制模型的训练程度，当模型训练不足时，拟合能力不够强，偏差主导了泛化错误率；随着训练程度的加深，模型的拟合能力逐渐增强，但也学习进去了数据的扰动，方差逐渐主导了泛化错误率；

因此，实际运用过程中，需要找到偏差与方差的平衡点。

1. **监督与非监督**

**监督学习与非监督学习的区别在于是否有明确的y值。**

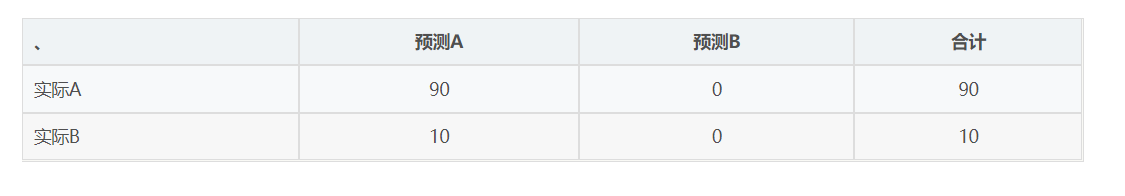
**监督学习（supervised learning）有明确的y值。**其任务是学习一个模型，使模型能够对任意给定的输入，对其相应的输出做出一个好的预测，如回归、分类等任务。

**非监督学习（unsupervised learning）没有明确的y值。**其直接对数据进行建模。没有给定事先标记过的训练范例，事先不知道输入数据对应的输出结果是什么。自动对输入的资料进行分类或分群，以寻找数据的模型和规律。如聚类算法。

1. **AUC和ROC**

AUC和ROC是二分类问题中常用的评估指标，判断分类结果的好坏，AUC在0-1之间，越接近1表示分类效果越好。

**当进行二分类问题时，分类结果会出现以下四种情况：**



**假定A为正样本、B为负样本:**

* 真正（True Positive , TP）被模型预测为正的正样本（实际为A，预测为A）
* 假负（False Negative , FN）被模型预测为负的正样本（实际为A，预测为B）
* 假正（False Positive , FP）被模型预测为正的负样本（实际为B，预测为A）
* 真负（True Negative , TN）被模型预测为负的负样本（实际为B、预测为B）

**根据这四个结果，再度引出两个指标：**

* **真正类率(true positive rate ,TPR),** 计算公式为TPR=TP/ (TP+ FN)，（被预测为正的正样本结果数 / 正样本实际数）。
* **假正类率(false positive rate, FPR),**计算公式为FPR= FP / (FP + TN)，（被预测为正的负样本结果数 /负样本实际数）。

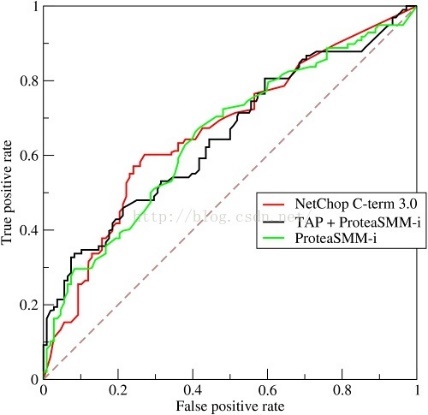
其中TPR表示正样本被预测正确的概率，假正类率表示负样本被预测错误的概率。

**进一步考虑以下四种情况。**

* 第一个点，(0,1)，即FPR=0, TPR=1，这意味着FN（false negative）=0，并且FP（false positive）=0。Wow，这是一个完美的分类器，它将所有的样本都正确分类。
* 第二个点，(1,0)，即FPR=1，TPR=0，类似地分析可以发现这是一个最糟糕的分类器，因为它成功避开了所有的正确答案。
* 第三个点，(0,0)，即FPR=TPR=0，即FP（false positive）=TP（true positive）=0，可以发现该分类器预测所有的样本都为负样本（negative）。
* 类似的，第四个点（1,1），分类器实际上预测所有的样本都为正样本。

**因此可以明确，假正类率相同时，真正类率越大，就是优秀的分类器。**

**ROC，是以横轴为FPR，纵轴为TPR，所画的曲线，对比不同的算法，即当FPR相同时，TPR越高则效果越好，因此曲线越接近左上角分类效果越准。**



**AUC，是ROC曲线下方的面积，可见面积小于1，面积越大即曲线越靠近左上角，效果越好。**

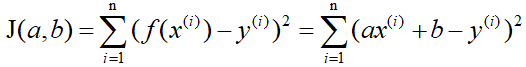
1. **模型（模型部分详见参考资料）**
   1. **线性回归**

线性回归是整个机器学习部分的基础。其基本思路为：有n组数据，自变量x(x1,x2,…,xn)，因变量y(y1,y2,…,yn)，然后假设它们之间的关系是：f(x)=ax+b。那么线性回归的目标就是如何选取a,b，能够让f(x)和y之间的差异最小

**线性回归的常考知识点包括：①最小二乘法②多重共线性问题。**

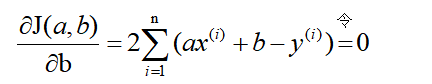
**（1）最小二乘法**

定义f(x)与y之间的差异公式：

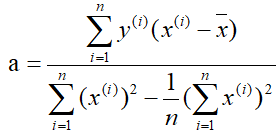


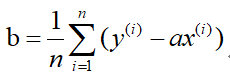
求解线性回归模型的参数，即a,b，方法为最小二乘法。

损失函数J(a,b)是凸函数，分别关于a和b对J(a,b)求偏导，并令其为零解出a和b。

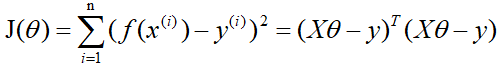


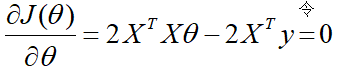
求得：





最小二乘法也可以用矩阵来表示参数：





在这里插入图片描述

**（2）多重共线性**

多重共线性问题指一个解释变量的变化引起另一个解释变量的变化。即线性回归的解释变量（X）之间存在较大的相关性。

使用最小二乘法进行参数估计，如若存在多重共线性，会造成线性回归的参数难以被准确估计。

原因在于通过最小二乘法，获得的参数估计函数为：

在这里插入图片描述

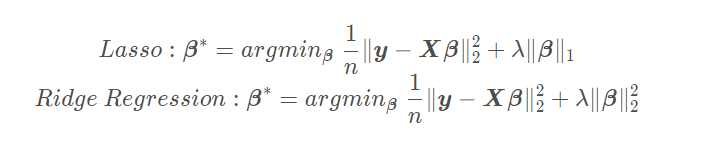
其中我们假设XTX是可逆的，但如果X存在多重共线性，若某特征变量之间完全相关，则XTX不可逆，参数无法计算。

解决多重共线性的方式包括：

* 高度相关的变量中只保留一个
* 将相关变量线性组合
* 利用降维模型将模型降维
* 使用岭回归和索回归

**2.2 岭回归（Ridge Regression）、索回归(Lasso Regression)**

岭回归与索回归都是普通线性回归，本质上也是优化模型，相比于普通线性回归，岭回归加入了L2正则化项，索回归加入了L1正则化项，故它们的目标函数如下：



**岭回归和索回归常考知识点包括：①与线性回归的区别②起到的主要作用。**

区别为岭回归加入了L2正则，L1加入了L1正则。作用为正则化防止过拟合，消除多重共线性影响，索回归还能起到特征降维、筛选有效特征的作用。

* 1. **逻辑回归**

逻辑回归是分类模型中的基础，其基础思想是在线性回归上加入sigmoid函数。

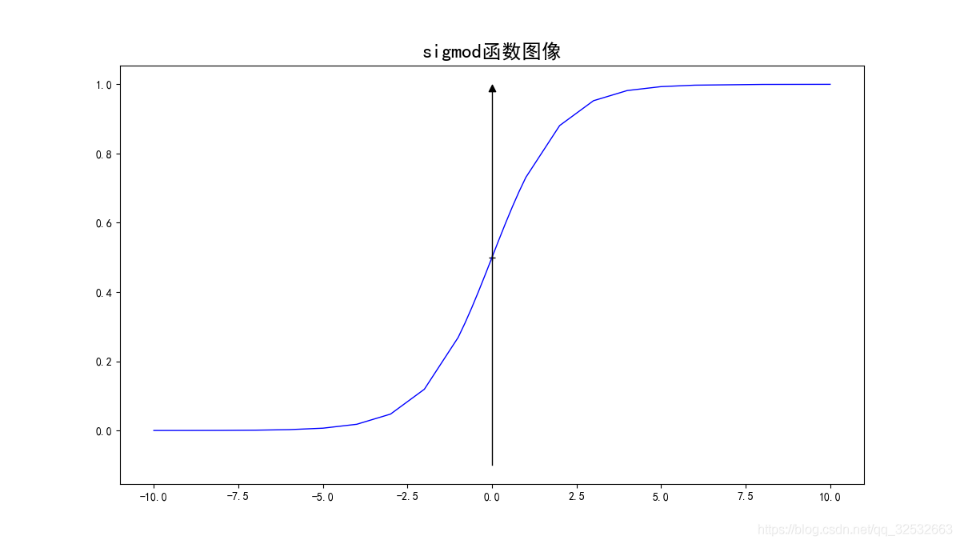
**逻辑回归的常考知识点包括：①Sigmoid函数②与线性回归的区别③损失函数。**

1. **Sigmoid函数**

Sigmoid函数的数学表达式为：



其函数图像为：



从图中可以看出，当z趋于-∞，g(z)趋于0，当z趋于∞，g(z)趋于1，且函数的值阈为(0,1)。

同时可以发现当z趋于5时，g(z)的值已经到0.99附近，z越大，g(z)越趋于1。

想一想我们平时碰到的概率，我们可能认为明天下雨的概率为0.3，天晴的概率为0.7。抛一枚硬币正面的概率为0.5，反面的概率也为0.5。概率也是介于0到1之间的一些数，很自然我们可以把sigmod函数的值域和概率联系起来。

**（2）与线性回归的区别**

逻辑回归的原理是用sigmoid函数把线性回归的结果从(-∞,∞)映射到(0,1)



即用线性回归公式作为sigmoid函数中的y，从而将线性回归结果划到（0,1）之间，结果会作为概率值。随后设定阈值，如阈值为0.5，则当模型预估结果>0.5，结果被分类为1，当模型预估结果<0.5，结果被分类为0。

**（3）逻辑回归的损失函数**

逻辑回归的损失函数，即分类模型中常用的损失函数，其具体公式为：

在这里插入图片描述



在这里插入图片描述

其中，g(xi)为模型给出的预估结果，yi为真实标签。

当yi=1时，结果为ln(g(xi))，此时g(xi)越大，则越接近1，损失函数越小。

当yi=0时，结果为ln(1-g(xi))，此时g(xi)越小，则越接近0，，损失函数越小。

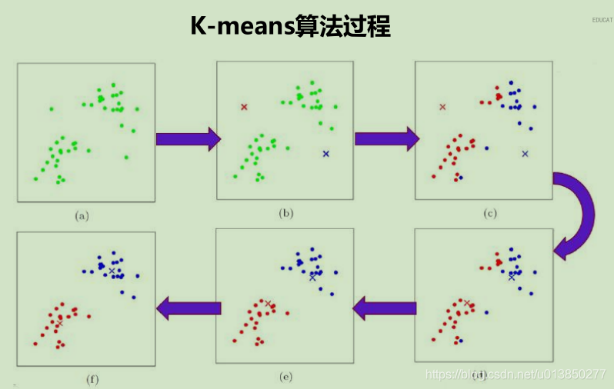
* 1. **KMeans算法**

KMeans是聚类中最基础的算法，原理相对简单，同样也是面试重点模型。

KMeans的核心考点时：（1）KMeans的具体步骤（2）如何确认KMeans的K值。

**（1）KMeans的具体步骤**

* 选取K个点做为初始聚集的簇心（也可选择非样本点）;
* 分别计算每个样本点到 K个簇核心的距离（这里的距离一般取欧氏距离或余弦距离），找到离该点最近的簇核心，将它归属到对应的簇；
* 所有点都归属到簇之后， M个点就分为了 K个簇。之后重新计算每个簇的重心（平均距离中心），将其定为新的“簇核心”；
* 反复迭代 2 - 3 步骤，直到达到某个中止条件。



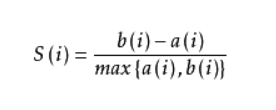
**（2）KMeans算法K的选取**

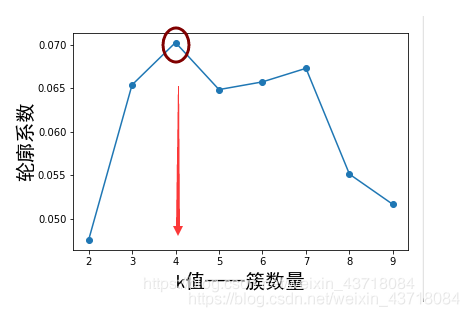
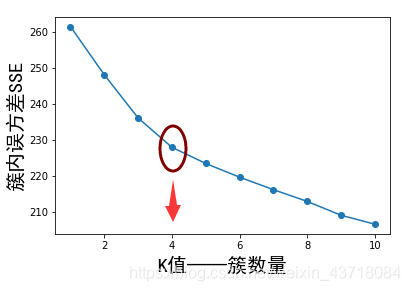
k值对最终结果的影响至关重要，而它却必须要预先给定。如何给定合适的k值，是面试中的常考问题。

确定K值的步骤为：**①首先给出评估聚类好坏的指标**，主要是SSE和轮廓系数两种。**②分别选取不同的K进行聚类**，计算聚类结果中的评估指标。**③选取表现最好的指标对应的K为最优K值。**

**SSE(簇内误方差)：**SSE利用计算误方差和，来实现对不同K值的选取后，每个K值对应簇内的点到中心点的距离误差平方和，理论上SSE的值越小，代表聚类效果越好，通过数据测试，SSE的值会逐渐趋向一个最小值。

**轮廓系数：**对于其中的一个点i①计算 a(i) = average(i向量到所有它属于的簇中其它点的距离)②计算 b(i) = min (i向量到各个非本身所 在簇的所有点的平均距离)





* 1. **决策树**

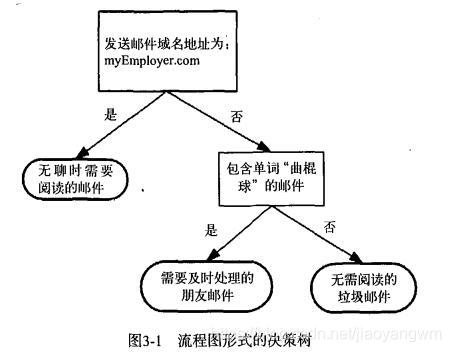
决策树是机器学习中重要的基础模型，为随机森林、Xgboost、LightGBM的基础，决策树可同时运用于分类或回归任务中，是非常重要必须掌握的模型、

**决策树的常考知识点包括：（1）决策树的基本步骤（2）信息增益、信息增益比、基尼指数（3）ID3算法、C4.5算法、CART树**

**(1)决策树的基本步骤**

决策树学习的算法通常是一个递归地选择最优特征，并根据该特征对训练数据进行分割，使得各个子数据集有一个最好的分类的过程。这一过程对应着对特征空间的划分，也对应着决策树的构建。

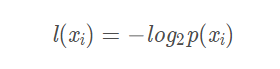
* + - * 开始：构建根节点，将所有训练数据都放在根节点，选择一个最优特征，按着这一特征将训练数据集分割成子集，使得各个子集有一个在当前条件下最好的分类。
      * 如果这些子集已经能够被基本正确分类，那么构建叶节点，并将这些子集分到所对应的叶节点去。
      * 如果还有子集不能够被正确的分类，那么就对这些子集选择新的最优特征，继续对其进行分割，构建相应的节点，如果递归进行，直至所有训练数据子集被基本正确的分类，或者没有合适的特征为止。
      * 每个子集都被分到叶节点上，即都有了明确的类，这样就生成了一颗决策树。



1. **熵、信息增益、信息增益比、基尼指数的计算**
   1. **熵**

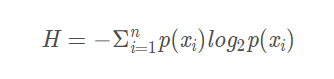
决策树选择特征，划分数据集的基本原则是：划分之后使原本无序的数据变得更加有序。为此，首先需要有衡量数据集是否有序的指标，即为熵。

**熵定义为信息的期望值，如果待分类的事物可能划分在多个类之中。**



其中，p(xi)为存在于某一特征为这个数值的概率。

为了计算熵，我们需要计算所有类别所有可能值所包含的信息期望值。



其中，n为分类数目，熵越大，随机变量的不确定性就越大。

* 1. **信息增益**

信息增益是相对于特征而言的。所以，特征X对训练数据集Y的信息增益，定义为集合D的经验熵H(Y)与特征A给定条件下Y的经验条件熵H(Y|X)之差，即：

* 1. **-信息增益比**

即信息增益比与经验熵H(Y)的比值。

* 1. **基尼指数**

1. **ID3算法、C4.5算法、CART树**

ID3算法：对应于使用信息增益

C4.5算法：对应于使用信息增益比

CART树：对应于使用基尼指数，二叉树

* 1. **随机森林**

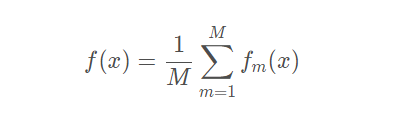
随机森林是基于决策树的集成模型。作为集成模型中较为简单的模型，比较适合面试官临时提问，或面试官让面试者介绍熟悉的模型时，如果对Xgboost、Lightgbm没有把握，而线性回归、决策树等模型太简单时，随机森林是比较适合介绍的模型。

**随机森林的常考知识点包括：（1）bagging（2）随机森林的具体步骤（3）随机森林的随机性（4）随机森林的关键参数与调参方向**

**（1）bagging**

Bagging即boostrap aggregating，其中boostrap是一种有放回的抽样方法，抽样策略是简单的随机抽样。

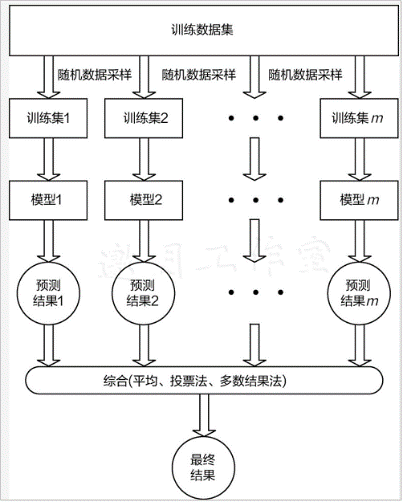
其**原理很直接，把多个基础模型放到一起，最后再求平均值**即可，这里可以把决策树当作基础模型，**其实基本上所有集成策略都是以树模型为基础的**，公式如下：



首先对数据集进行**随机采样**，分别训练多个树模型，最终将其结果整合在一起即可，思想还是非常容易理解的，其中最具代表性的算法就是**随机森林。**

**(2) 随机森林的具体步骤**

随机森林是机器学习中十分常用的算法，也是bagging集成策略中最实用的算法之一。那么随机和森林分别是什么意思呢？森林应该比较好理解，分别建立了多个决策树，把它们放到一起不就是森林吗？这些决策树都是为了解决同一任务建立的，最终的目标也都是一致的，最后将其结果来平均即可。



**（3）随机森林的随机性**

建立多个决策树模型取平均值，如果每个决策树模型相同，则均值也相同，此时随机森林没有意义。因此，应当对每颗决策树的建立引入随机性：

**第一种随机性是数据采样的随机性：**训练数据取自整个数据集中的一部分，如果每一个树模型的输入数据都是不同的，例如随机抽取80%的数据样本当作第一棵树的输入数据，再随机抽取80%的样本数据当作第二棵树的输入数据，并且还是有放回的采样，这就保证两棵树的输入是不同的，既然输入数据不同，得到的结果必然也会有所差异。

**第二种随机性是特征的随机性：**如果对不同的树模型选择不同的特征，结果的差异就会更大。例如，对第一棵树随机选择所有特征中的60%来建模，第二棵再随机选择其中60%的特征来建模，这样就把差异放大了，这就是第二重随机。

1. **随机森林的关键参数与调参方向**

随机森林的重要的参数包括：最大迭代次数(N\_estimator)、最大深度(max\_depth)、最大特征数(max\_features)、叶子节点最少样本数(min\_samples\_leaf)

* **最大迭代次数（N\_estimators）指**随机森林模型中建立的决策树数量。决策树数量越多，最后随机森林的结果综合的情况越多，预测结果往往更加精准，但是决策树的数量增多也会导致决策树计算时间变长，以及过度拟合情况发生。决策树的数量较少，则计算时间相对更快，但结果可能不够精准，出现欠拟合情况。在实际建模过程中，默认最大迭代次数为10，一般会将取值范围定到1-200之间。
* **决策树最大深度（Max\_depth）指**所有叶子结点的最深深度。决策树深度越深，则在决策树分裂过程中节点分裂的次数更多，最后结果的范围会更加精准，但是分裂次数增多也会导致决策树计算时间变长，以及过度拟合情况发生。分裂次数较少，会有较多的数据集中在同样的预测值中，计算时间相对快，但结果可能不够精准，出现欠拟合情况。在实际建模过程中，一般会将该参数取值范围定到10-100之间。
* **最大特征数（Max\_features）指**随机森林中单棵决策树可以使用的特征最大数量。其默认值通常用None表示，此时单棵决策树分裂过程中能够使用所有的的特征。当最大特征数较少时，可以使每棵树更加多元化，较多的考虑各种特征组合情况。当最大特征数较少时，单棵树的多样性降低。
* **叶子节点最少样本数（Min\_samples\_leaf）是**指限制叶子结点所需的最少样本数。即叶子结点中的样本只有大于该数时才会作为叶子结点存在，否则将会被去除。叶子结点最少样本数越大，则相应的叶子结点个数越少，越多的数据集中在同一个叶子结点中，模型精确度相应下降，但计算时间更快，可能出现欠拟合情况。当叶子节点最少样本数越多，则相应的叶子结点个数越多，模型精确度相对提升，但计算时间增加，可能出现过度拟合情况。此参数的取值可以是整数或者百分比，针对样本量较少的时候，可以直接将值设定为1。如果样本量较大的情况下，则建议增加该值。
  1. **Gbdt、Xgboost、LightGBM**

Xgboost、LightGBM基本是常规数据分析面试中机器学习板块的难度天花板。推荐求职技术向数据分析师的同学要清楚其中的推导细节，而对于业务向数据分析同学，简单了解即可，甚至可以不看。受限于这几个模型本身的复杂程度，很难用简短的篇幅概括原理，因此给出具体的学习地址，本处则罗列重要的考点。

**核心中的核心，常考知识点：GBDT、Xgboost、LightGBM模型之间的区别。**

此题基本是考到上述模型最常问的考题，因为三个模型之间的区别点代表了三个模型的核心关键，**一个题目就可以了解是否对三个模型的主体部分有充分掌握**。实际面试中，上述三模型的考题80%的概率就是讲述三模型的区别。

**（1）GBDT与Xgboost的区别**：

**损失函数的区别，**Xgboost在损失函数中利用泰勒展开式使用了二阶导的信息，而GBDT只采用了一阶导信息。

**正则化的区别，**Xgboost在损失函数中将树的复杂度加入到正则化项中，防止过拟合。

**（2）Xgboost与LightGBM的区别**

LightGBM的基本思想与Xgboost一致，其主要区别在于采用了很多加速训练的算法，使模型在不影响精度的情况了加快学习速度。

**在回答问题时，答案便是LightGBM的三种内置算法：**

* 1. GOSS算法。用于减少模型训练的样本数量。
  2. EFB算法。用于减少模型训练的特征数量。
  3. Histogram算法。用于加速特征划分的寻找最优点切割速度。

**LightGBM = Xgboost + GOSS + EFB + Histogram**

* 1. **其余模型**

机器学习的模型很多，除了上述介绍的常考模型外，还有包括SVM（支持向量机）、DBSCAN(聚类算法)、PCA、SVD（降维算法）等许多模型。但这种模型以作者经验来看，很少被面试官主动提问，但若简历上有这些模型相关项目的同学可能会被问到，需要自行准备。

因此这些模型的要求是如果掌握最好，未掌握也问题不大，可根据个人情况斟酌。

**面经真题：**

1、讲述逻辑回归的原理。（字节跳动）

2、讲述ROC、AUC的原理。（字节跳动）

3、bagging和boosting的区别（字节跳动）

4、XG boost和GBDT区别？以及优势（字节跳动）

5、说说随机森林和xgboost的区别（腾讯音乐）

6、缺失值怎么处理（兴业数金）

7、怎么解决过拟合问题（兴业数金）

8、监督学习与无监督学习的区别（兴业数金）

9、对聚类的了解（兴业数金）

10、什么是随机森林（兴业数金）

11、平时是否有使用过kmeans和knn，简单说明一下两者区别并检讨说明一下二者的逻辑（字节跳动）

12、线性回归和逻辑回归的区别（美团）

13、主成分分析PCA（美团）

14、聚类的评价指标、算法原理（顺丰）

15、XGB和LGB的区别优化（顺丰）

16、XGB调参、某某参数有啥用（顺丰）

17、随机森林是什么？随机森林和xgboost之间的区别（蚂蚁）

18、XGboost和AUC以及特征工程等等（滴滴）

19、归一化是否了解（字节跳动）

20、k-means和KNN的异同点（滴滴）

21、过拟合原因+解决方法（滴滴）

22、什么是多重共线性，有什么危害，如何解决？（腾讯）

23、讲讲lasso的用处（腾讯）

24、逐步回归法、lasso、岭回归各有什么优缺点（腾讯）

**参考资料：**

1. **训练集、验证集、测试集：**

<https://blog.csdn.net/kieven2008/article/details/81582591>

1. **偏差与方差**

<https://blog.csdn.net/hai008007/article/details/79777232>

1. **过拟合与欠拟合**

<https://blog.csdn.net/willduan1/article/details/84825471>

1. **监督与非监督**

<https://blog.csdn.net/sanmi8276/article/details/113849626>

1. **AUC、ROC**

<https://blog.csdn.net/zhihaoma/article/details/51166711>

1. **线性回归**

<https://blog.csdn.net/weixin_44697198/article/details/109405212>

https://blog.csdn.net/deephub/article/details/120696154

1. **岭回归、索回归**

<https://blog.csdn.net/weixin_42633850/article/details/105421717>

1. **逻辑回归**

<https://blog.csdn.net/qq_32532663/article/details/105763824>

1. **KMeans**

<https://blog.csdn.net/weixin_43718084>

1. **决策树**

<https://blog.csdn.net/jiaoyangwm/article/details/79525237>

1. **GBDT、Xgboost、LightGBM**

<https://blog.csdn.net/zpalyq110/article/details/79527653>

<https://blog.csdn.net/weixin_42462804/article/details/104352985>

https://blog.csdn.net/weixin\_39807102/article/details/81912566

<https://blog.csdn.net/weixin_42051011/article/details/104796070>