1.《统计学习方法》，这书可能不是最全的，但是讲得最精髓，薄薄一本，适合面试前突击准备。统计学习的核心步骤：模型、策略、算法，你应当对logistic、SVM、决策树、KNN及各种聚类方法有深刻的理解。能够随手写出这些算法的核心递归步的伪代码以及他们优化的函数表达式和对偶问题形式。

2.数学知识方面，你应当深刻理解矩阵的各种变换，尤其是特征值相关的知识。

3.算法方面：理解常用的优化方法：梯度下降，牛顿法，随机搜索算法（基因，蚁群），最起码要理解到：梯度下降法是平面来逼近局部，牛顿法是利用曲面逼近局部

4.其实最重要的还是将机器学习应用在搜索，广告，垃圾过来，安全，推荐系统中。对业务要有深刻的理解这是最重要的只有做了实际的项目才能有体会，项目是**最大的加分项啊**

5.svm的原理，svm核函数

6.k-means，如何利用hadoop的实现

7. naive bayes和logistic regression的区别

8. LDA的原理和推导

1. 做广告点击率预测，用哪些数据什么算法
2. 推荐系统的算法中最近邻和矩阵分解各自适用场景
3. 用户流失率预测怎么做（游戏公司的数据挖掘都喜欢问这个）: 数据不平衡问题和时间序列分析
4. 一个游戏的设计过程中该收集什么数据
5. 如何从登陆日志中挖掘尽可能多的信息
6. 这些问题的特点是很基础很简单，因为实际中很少用复杂的算法，复杂的算法不好控制
7. Message passing interface(MPI)是高性能计算领域开发并行应用的标准API。所以我想MPI在机器学习系统中也应该有广泛应用
8. 我面的推荐，问了各类协同过滤的好与坏。
9. 然后我说我做过LDA，问我，Dirichlet Distribution的定义和性质，并问我，为什么它和multinomial distribution是共轭的，顺便问了我啥叫共轭分布。
10. 现实应用中的Top-N推荐问题和学术研究中的评分预测问题之间有什么不同。
11. 问我ItemCF的工程实现，面对大数据如何实现，又追问了有没有什么工程优化算法。这个问题我没答好，一开始我说了一个MapReduce模型，他问能不能更快一点，我就卡那了。。。最后面试官告诉我，不能只从算法角度分析，要从系统设计分析，利用内存来减小MapReduce的吞吐量。
12. 最后考了我一个基本概念，什么叫判别模型什么叫生成模型。
13. 数据结构算法水题+常用机器学习算法推导+模型调优细节+业务认识
14. 面试官一般不会因为你不会某个算法不要你，往往是一些涉及到调优，底层原理的模糊让人觉得你很单薄。如果思维逻辑也不清晰，那基本上就会有问题了。
15. 在机器学习中, L^2和L^1范数分别对应L^2和L^1正则化, 详情参考线性模型中的**岭回归(**[Ridge Regression](http://scikit-learn.org/stable/modules/linear_model.html#ridge-regression)**)**和**套索回归(**[Lasso](http://scikit-learn.org/stable/modules/linear_model.html#lasso)**)**.
16. Kmeans必须懂 KNN懂原理不用写 SVM、MR和Pagerank多看看，遇到问题多用这些方法讨论（不关乎信仰，面试官都懂）
17. 个人比较成功的面筋是看好面试官的cd间隔放大招：比如看简历沉没的时候主动说high light一下，有条件就把线上的project拿出来秀；代码写完主动说说能有什么改进思路，把面试官拐跑，就不会被他带到挖好的坑上来了。
18. 信自己的数学和逻辑直觉，不要信Model。
19. 字符串反转
20. 快排
21. 如何提高Python的运行效率
22. 写一个简单的正则匹配表达式(将文本中的123.4匹配出来)
23. KNN（分类与回归）
24. CART（回归树用平方误差最小化准则，分类树用基尼指数最小化准则）
25. Logistics（推导）
26. GBDT（利用损失函数的负梯度在当前模型的值作为回归问题提升树算法中的残差的近似值，拟合一个回归树）
27. 随机森林（Bagging+CART）
28. SVM与随机森林比较
29. 改变随机森林的训练样本数据量，是否会影响到随机森林学习到的模型的复杂度
30. Logistics与随机森林比较
31. GBDT与随机森林比较
32. 自己实现过什么机器学习算法
33. 推荐算法（基于用户的协同过滤，基于内容的协同过滤）
34. 如何做一个新闻推荐
35. Map-reduce，Hadoop
36. 一个袋子里有很多种颜色的球，其中抽红球的概率为1/4，现在有放回地抽10个球，其中7个球为红球的概率是多少？（伯努利试验）
37. 介绍大顶堆和小顶堆
38. 二叉树的前中后遍历
39. 手写二叉树前序递归遍历算法（千万不要忘记异常处理！）
40. 介绍二叉树前序遍历非递归遍历算法
41. list有哪几种添加元素的方法，能否从表头插入元素？(append, extend和insert, insert能从表头插入元素, 但是时间复杂度为O(n).)
42. 如何获取list中最后一个元素
43. a = [1, 2, 3, 4], b = a, b[0] = 100, 请问print(a)结果是什么
44. [list是怎样实现的](http://www.jianshu.com/p/J4U6rR)
45. 介绍项目
46. 项目相比别人有什么优劣
47. 项目的数据从哪里来
48. 项目的特征向量的归一化与异常处理
49. 项目的下载量
50. 目前在研究什么
51. 参加天猫大数据推荐算法成绩
52. 线性分类器与非线性分类器的区别及优劣；
53. 特征比数据量还大时，选择什么样的分类器？
54. 对于维度很高的特征，你是选择线性还是非线性分类器？
55. 对于维度极低的特征，你是选择线性还是非线性分类器？
56. 如何解决过拟合问题？
57. [L1和L2正则的区别](http://blog.csdn.net/zouxy09/article/details/24971995/)，如何选择L1和L2正则？
58. 随机森林的学习过程；
59. 随机森林中的每一棵树是如何学习的；
60. 随机森林学习算法中CART树的基尼指数是什么？
61. 如何找到第k大的数？
62. 机器学习知识中问的很细，不仅需要考虑算法本身，还需要考虑应用。
63. 常用的数据结构及应用场景（list，dict，tuple）

给定一个分类器p，它有0.5的概率输出1，0.5的概率输出0。

1. **Q1**：如何生成一个分类器使该分类器输出1的概率为0.25，输出0的概率为0.75？ **Ans**：连续进行两次分类，两次结果均为1则输出1，其余情况（10,01,00）均输出0。
2. **Q2**：如何生成一个分类器使该分类器输出1的概率为0.3，输出0的概率为0.7？ **Tip**：小明正在做一道选择题，问题只有A、B和C三个选项，通过抛一个硬币来使选择3个选项的概率相同。小明只需抛连续抛两次硬币，结果正正为A，正负为B，负正为C，负负则重新抛硬币。 **Ans**：连续进行4次分类（2^4=16 > 10），结果前3种情况则输出1，结果接下来7种情况则输出0，其余情况重新进行分类。
3. **Q1**：给定一个1T的单词文件，文件中每一行为一个单词，单词无序且有重复，当前有5台计算机。请问如何高效地利用5台计算机完成文件词频统计工作？ **Ans**（有问题的）：将1T文件切分为5份，分配给5台计算机。每台计算机进行词频统计工作，输出一个结果为{单词：频数}的字典结果文件。将5台计算机生成的5个结果文件合并。
4. **Q2**：每台计算机需要计算200G左右的文件，内存无法存放200G内容，那么如何统计这些文件的词频？ **Ans**（不是最优）：首先将文件排序，然后遍历利用list存储结果即可。（不能用字典，因为200G统计出来的结果会很大，没有那么大的内存存放字典。由于经过排序操作，遍历存储并不会使结果丢失，所以用list存储结果即可，每当一个list即将占满内存，则将其写入文件，然后清空list继续存储结果。）
5. **Q3**：如何将1T的文件均匀地分配给5台机器，且每台机器统计完词频生成的文件只需要拼接起来即可（即每台机器统计的单词不出现在其他机器中） **Ans1**（不是很好）：对1T文件中的单词进行抽样，获得其概率分布，遍历文件，然后根据首字母的概率均匀分配至5台计算机，如a到e的概率均为0.04, 0.04\*5=0.2，则将所有以a-e的单词放入第1台计算机，若z的概率为0.2，则把所有以z开头的单词放入第5台计算机。缺点：不具有可扩展性，如果有100台计算机，那么可能就需要2个字母计算了，则程序就要改变。还有可能出现2台机器中有相同的单词。 **Ans2**（不是最优）：遍历文件，对于每一个单词，获得单词中各字母的ASCII码值，然后将ASCII值之和取余。则每台机器中的单词必定是不一样。
6. 百度NLP部门电面考的基础很多，但是关于机器学习、数据挖掘方面的知识一个都没有问，坑爹啊！！！亏我准备了好久关于机器学习方面（特别是SVM，随机森林和AdaBoost）的东西！！！所以即使是投递机器学习、数据挖掘和自然语言处理相关的岗位，关于编程基础（C/C++）和算法还是得准备
7. 决策树的模型很重要，不会问你ID3这些简单的，甚至CART都不会，不过会问GBDT，随机森林。
8. 如一个热门微博排行榜等
9. 实现一个分布式的矩阵向量乘的算法。。。
10. 最速下降法和共轭梯度法 wolfe条件 最速下降法和共轭梯度法的收敛速度如何判断。。。
11. 约束优化的KKT条件 KKT条件在边界区域的搜索行为的物理意义是什么。。。
12. 实现一个分布式的topN算法。。。
13. 为什么可以使用logistic回归。。。  
    这个我不懂。。。 好吧 hr让我去看看《离散选择方法》这本书。。。 算是学到
14. 不要没事去搭建hadoop, spark集群!!!

不要没事去搭建hadoop, spark集群!!!

不要没事去搭建hadoop, spark集群!!!

初学者不要关注性能调优!!!

初学者不要关注性能调优!!!

初学者不要关注性能调优!!!

牢记上面两点忠告, 这将节省你大量宝贵的时间。懂得取舍也是一种智慧。

我的建议, 下载完安装包, 配置成单机版, 开搞。

学习hadoop, 首先要了解什么是倒排索引. 然后尝试用mapreduce实现inner join, 解决top-k问题, 解决简单的好友推荐问题, 之后可以深入了解一下hive里的join, group by, distribute by的实现, 搞清楚怎么样把一段SQL拆分成各个stages，深入总结一下，什么样的情况下必须要有merge stage，假如给你十亿的样本，你该怎么借鉴mr的基本思想实现全局排序？

学习spark, 推荐用scala. 首先了解rdd和dsstream, 用mllib从头到尾实现一下官网的决策树, 逻辑斯蒂回归, 线性回归, 找到成就感, 在这个过程中你也许会踩到不少坑, 也会学习到一些基础的也是核心的概念, 比如何时需要persist, 何时需要cache, 何时需要广播变量, 全局accumulator, rdd stage划分, 为什么需要避免在class内的函数的rdd里引用外部变量, driver和executor区别.

这些是基础, 搞懂这些, 再尝试去系统性地看书, 看资料. 在此之前, 网加博客就能让你掌握6-7成的技能.

搞懂以上知识点，然后再深入进阶，推荐一些进阶资料:

1. hadoop相关的不想多说, 网上太多了, 推荐一下hulu的董西城大神的深入hadoop技术内幕>系列.

2. spark相关:

**[Spark Streaming 源码解析系列] (**[lw-lin/CoolplaySpark](https://github.com/lw-lin/CoolplaySpark/tree/master/Spark%20Streaming%20%E6%BA%90%E7%A0%81%E8%A7%A3%E6%9E%90%E7%B3%BB%E5%88%97)**)**

**[**[**data-algorithms-book**](https://github.com/mahmoudparsian/data-algorithms-book)**-Scala Spark Solutions] (**[mahmoudparsian/data-algorithms-book](https://github.com/mahmoudparsian/data-algorithms-book/tree/master/src/main/scala/org/dataalgorithms)**)**

**[SparkDeepDoc](**[codlife/SparkDeepDoc](https://github.com/codlife/SparkDeepDoc)**)**

**[Mastering Apache Spark 2](**[Mastering Apache Spark 2](https://www.gitbook.com/book/jaceklaskowski/mastering-apache-spark/details)**)**

最后的忠告, 多动手, 少看书, 多一些实践, 少一些套路. bless~有人问spark为什么不推荐用python写。当然可以用python，不过仅限于小型的项目。像我现在做的一个项目，用scala写的，模型训练和评估的代码量已经接近2万行了。用python写的话我没信心能把控得像现在这么好，另外scala 代码量一大idea都感觉hold不住了，很难想象丢到pycharm里会变成什么样子。以前用flask写过中等规模的服务，代码量上去后ide连自动补全都失效了。scala写的话，错误提示失效了，至少类型检查和自动补全还勉强能用。

1. 另外推荐一下用scala或java写大型程序的一个小技巧。方法签名和返回值很复杂时建议用idea的quickdoc获取。高效而且准确。尤其是传嵌套的数据结构进去时，这是静态语言的一大优点。
2. http://blog.csdn.net/sunpeng19960715/article/details/54745922

一直想写写数据科学家面试中机器学习有关的问题，无奈杂事缠身又懒癌发作导致迟迟没有动笔。最近找工季来临，不少同学找我咨询数据科学家面试的问题，借此机会把自己的一点点想法和经验教训记录下来。

作为数据科学家的核心技能之一，机器学习在数据科学家日常工作中的重要性不言而喻。正因如此，机器学习是数据科学家面试中极为重要的一环。但是机器学习涵盖范围广，同时涉及诸多复杂的数学知识，精通所有主流算法及其细节对于绝大多数人而言都是很难实现的。同时，机器学习的面试又是一个理论与实践紧密结合的方式，对于初学者来说，无疑大大增加了面试准备的难度。但是，面试并不是考试，对于知识点的侧重与考试也是有所不同，只要准备方法得当，抓住主要矛盾，掌握面试的规律，就能够游刃有余的应对大部分机器学习方面的面试。

接下来我将以垃圾邮件分类器为例，谈一下数据科学家面试中是如何对机器学习进行考察。假设我们做过一个垃圾邮件分类器的项目。为了建立这个分类器，我们首先对数据进行清理及预处理，如缺失数据的处理、数据的归一化等。在获得初始特征向量后，用PCA进行了特征选择。利用特征选择得到的特征向量及对应数据，训练一个随机森林的分类器作为我们的垃圾邮件分类器。针对的这样一个项目，有这样几个点可以进行挖掘和准备。

**1. 项目简介**

如何向面试官介绍你做过的项目，这是一个非常基础、非常常见但是又充满技巧的问题。首先，项目简介不应过于冗长，力争用最短的几句话勾勒出项目的框架。其次，数据科学相关项目通常是业务与技术并存，因此，既要突出项目过程中解决的技术难题及应用的相关技术，又应该兼顾项目带来的业务上的影响。

**2. 模型简介**

这类问题同样是机器学习面试中最普遍最常见的一类问题，面试的形式一般为介绍一个你最喜欢的模型，或是介绍项目中应用的某种模型。与项目简介相同，模型简介也应力求简洁，用最简短的几句话，讲清楚模型是用了什么样的原理完成了怎样的目标。wikipedia中关于随机森林的定义给我们提供了一个非常好的学习模板，可以用来借鉴：

Random forests is a notion of the general technique of random decision forests that are an ensemble learning method(怎样的方法) for classification, regression and other tasks(解决了什么问题), that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees(基本原理).

但是，wikipedia中大部分模型的描述更偏书面化的表达，并不适合原封不动地照搬到面试中。我们需要将它转化为更口语化的表达。

**3. 模型的优缺点**

模型的优缺点与模型简介是紧密相关的，可以将两个问题结合起来一起准备。比如之前我们谈到了什么是随机森林，紧接着可以谈一下随机森林有什么优点，如：a. 对于很多数据集表现良好，精确度比较高；b. 不容易过拟合；c. 可以得到变量的重要性排序；d. 既能处理离散型数据，也能处理连续型数据，且不需要进行归一化处理； e. 能够很好的处理缺失数据；f. 容易并行化等等。同时，将理论与实践结合也是非常好的切入点，如随机森林的诸多优点是如何体现在垃圾邮件分类器项目中，这样的结合能更好的展示出面试人对于模型的理解及掌控。

**4. 模型原理及相关技术细节**

模型简介与模型优缺点的问题属于概念性问题，偏向于考察面试人是否了解某种模型，而更进一步的则是对于模型原理及相关技术细节的考察，比如模型假设、目标函数、优化过程、算法收敛性等。所谓知其然，又知其所以然，这是对于面试人的进一步要求。

例如，在模型的优缺点中，我们提到了随机森林可以对变量重要性进行排序。相应地，我们应该能够解释随机森林是如何对变量重要性进行排序，有哪几种常见的排序指标，比如利用OOB误分率的改变或者分裂时信息增益的变化等。当然，问题并非到此终止，基于上面提到的两种常见的变量重要性排序指标，又可以衍生出新的问题。例如，针对OOB误分率这个指标，解释一下什么是OOB，随机森林中OOB是如何计算的，它有什么样的优缺点；针对信息增益，同样会有很多与之有关的问题，如什么是信息增益，如何计算信息增益，什么是熵，什么是GINI指数，他们之间的区别是什么，他们之间的区别会对建树产生怎样的影响等。

再如，在垃圾邮件分类器项目中，有一部分数据存在缺失，而随机森林具有处理缺失数据的优点，建模的过程中我们充分利用了这一特性。那么，与之相关的问题可能会是，随机森林为什么会有这个优点？随机森林是怎样对缺失数据进行训练及预测？

**5. 模型的横向比较**

模型原理及相关技术细节的考察属于对机器学习知识深度考察的范畴，与之对应的是机器学习知识广度方面的考察。广度上的考察主要有两大部分，一方面是从理论上对不同算法进行横向比较，如模型假设，优化方法等。另一方面，是结合实际案例对不同算法进行横向比较，这要求面试人不仅仅要熟知不同模型的原理及技术细节，更需要将抽象的理论与具体的实践结合，在实际案例中对算法进行比较。

在垃圾邮件分类器项目中，随机森林被用作最终的分类器模型。面试官可能会就此提出以下问题：为什么选择随机森林而非其他模型，比如朴素贝叶斯或者支持向量机。一般来说，面试者可以从数学理论和工程实现两个方面进行比较回答。从理论上讲，数据表现出来的特征，以及模型所基于的假设都是很好的突破口;从工程实现上讲，实现的难易程度，是否易于scale都是可以考虑的点。

**6. 开放性问题**

除了对机器学习知识深度和广度上的考察，开放性问题也是面试中经常会遇到的问题，对于初学者来说这也是最难准备的一类问题。一方面这类问题很难在教科书中见到，没有固定的问题清单；另一方面，这类问题没有标准答案，很多时候是对过往经验的总结。针对这一类问题，更多的是靠平时工作学习过程中多思考、多总结、多积累，临阵抱佛脚很难起到效果。

再次回到垃圾邮件分类器项目，这个项目中有多个开放性问题可以被提问。比如，1. 邮件数据存在缺失，通常情况下，如何对缺失数据进行处理？2. 垃圾邮件分类是一个非平衡数据集分类的问题，针对这一类问题，我们应该如何进行建模。3. 项目中，PCA被用于特征选择，除此而外，还有哪些方法可以用来进行特征选择？

**7. 准备材料**

在准备机器学习的过程中，我主要用了如下的材料：

A. Stanford CS229 Machine Learning.

B. CMU 10-701 Introduction to Machine Learning.

C. The Elements of Statistical Learning. By Trevor Hastie, Robert Tibshirani and Jerome Friedman.

D. Pattern Recognition and Machine Learning. By Christopher Bishop.

A.和B.是Stanford和CMU机器学习课程的课件，里面涵盖了各种常用算法，应该力求掌握这些算法。C.和D.是经典中的经典，难度适中，内容没有太理论，语言也没有太晦涩，是机器学习内功修炼的不二法门。

关于材料再说句题外话，我之前是个资料收集整理爱好者，总是在努力充实自己的资料库，总是担心遗漏任何有用的材料。但是后来才逐渐意识到，资料求精不求多，存在脑子里面的叫知识，存在硬盘里叫文档。