面试常见问题总结

1、python中的list 是如何实现的 https://www.jianshu.com/p/J4U6rR

底层采用c实现 allocated 和 list_ resize

append o1，内存空间的增长方式为 0, 4, 8, 16, 25, 35, 46, 58, 72, 88,

insert 的操作是on量级的。

pop o1 remove on

2、python中字典的实现 http://python.jobbole.com/85040/

在Python中，字典是通hashtable实现的。也就是说，字典是一个数组，而数组的索引是键经过哈希函数处理后得到的。哈希函数的目的是使键均匀地分布在数组中。由于不同的键可能具有相同的哈希值，即可能出现冲突，高级的哈希函数能够使冲突数目最小化。

Ps： 数组不能作为dict 中的key，因为数组不能被hash

3，list 与tuple 有哪些区别 https://blog.csdn.net/weixin_39431596/article/details/78730731

tuple是一种有序列表，它和list非常相似，但是（但是前面的话也不都是废话）

tuple一旦初始化就不能修改，而且没有append() insert()这些方法，可以获取元素但不能赋值变成另外的元素

tuple中的元素要如何改变呢？？元素里面存饭的值为list

Tuple 与list 可以相互转换

4、模型训练的时候过拟合欠拟合的判断。https://blog.csdn.net/qq_28031525/article/details/60966234

从数据考虑：如果特征比样本多，会出现不满秩矩阵会发生过拟合

训练过层中：训练的loss下降，测试的loss上升

发生条件： 是否满秩

正则的具体在调节写什么？l1（稀疏），l2（平滑）

5、决策树id3，c4.5 cart算法总结 https://blog.csdn.net/qq_28031525/article/details/62217865

决策树模型可读性好，有助于人工分析；效率高只需要一次构建就能反复使用

6、pca降维方法是无监督的。根据最大方差来接行寻找

http://www.360doc.com/content/13/1124/02/9482_331688889.shtml

7、RF xgboost gbdt(gradient boosting decision tree) 内容整理

https://blog.csdn.net/qq_28031525/article/details/70207918

最近kaggle上最火的可能是 bgmlight model

Rf过程： 1、随机选择样本2、随机选择特征 3、构建决策树 4、投票表决

8、dl中避免过拟合的方法。https://blog.csdn.net/qq_28031525/article/details/78982555

1. L2正则化 2. Dropout 3. Data Augmentation 4. Early stopping 5、min-batch 6、batch normalization

min-batch https://blog.csdn.net/qq_28031525/article/details/78983043

当batch size 减小为1时，退化为SGD，此时将会丢失向量化处理的优势；

当batch size增大到m时，此时为batch gradient descent，此时每次循环将会需要很长时间。

相反如果采用mini-batch时，不仅能够利用向量化处理的优点，而且在不同时处理整个数据集时取得不错的效果。

ps：超参的调节使用cvgrid。sk中的网格调参法

9、RNN 产生梯度消失的原因以及LSTM 如何进行缓解的 https://blog.csdn.net/qq_28031525/article/details/79423450

对于梯度爆炸的问题，一般采用截断机制

梯度消失的产生原因：RNN在梯度向后传递的时候，优于相同的矩阵相乘次数太多，梯度倾向于逐渐消失，导致后面的节点无法更新参数，整个学习过程无法正常践行

手绘lstm的图，（gru内部结构太复杂，没看到让画的）

10、文本相似度计算 https://blog.csdn.net/qq_28031525/article/details/79596376

从传统的ngram model 到目前start-of-art 方法的

基于关键词匹配 ，jaccard相似度

基于向量：tfidf、word2vec、lsa。欧式距离、余炫相似度

深度学习：convnet （经过不同卷积核提取特征的拼接结果）

LSA 相比于tfidf 更好的捕获了单词之间的相关信息

http://www.cnblogs.com/kemaswill/archive/2013/04/17/3022100.html

11、海量数据的处理方式。

https://zhuanlan.zhihu.com/p/24383239

hashmap bitmap 更多关注分治的思想

12、cnn在NLP中的应用和 卷积和的不同

https://blog.csdn.net/qq_33414271/article/details/80043606

1、每个核的大小是不同的长度，滑动窗口不同，

2、padding与否分为宽卷积与窄卷积
3、通道数、在文本上可以立即为不同的词向量获取方式，（word2vec、glove等）
4、步长，步长越大得到的特征越稀疏

13、NLP中attention总结
https://mp.weixin.qq.com/s/BeELZ213kp67RrU-yem8rg

14、在不均衡数据分类问题上的评价指标使用。
https://blog.csdn.net/pipisorry/article/details/51788927
AUC：ROC 曲线的线下面积，特别用于二分类上，其不关注具体的得分，只关注排序结果，使其特别适合排序问题的优化结果

15、对话系统 https://blog.csdn.net/qq_28031525/article/details/79855018

分类为：任务型的对话系统、闲聊型的chatbot
解决方法： 检索、生成（seq2seq）
里面有一个阿里的小蜜应用技术的变迁史 也已当作chatbot发展的变迁

16、意图识别。http://www.shuang0420.com/2017/04/27/NLP%E7%AC%94%E8%AE%B0%20-%20NLU%E4%B9%8B%E6%84%8F%E5%9B%BE%E5%88%86%E7%B1%BB/
该topic其实是文本分类的一种，难点在于domain和intern的确定上

17、各种中文处理工具的比较
https://blog.csdn.net/sinat_26917383/article/details/77067515
使用nltk处理中文的方法可以采用jieba将其分词后在借助cornnlp的工具进行应用 3.2之后才即成了cornnlp的分词函数

18、l1与l2的直观化理解 https://blog.csdn.net/jinping_shi/article/details/52433975
https://www.jianshu.com/p/8025b6c9f6fa

L1正则化可以产生稀疏权值矩阵，即产生一个稀疏模型，可以用于特征选择（稀疏规则算子 lasso regression）
L2正则化可以防止模型过拟合（overfitting）；一定程度上，L1也可以防止过拟合 （岭回归 ridge regression）（通过l2范数实现了对模型空间上的限制）

19、如何理解面向对象。https://blog.csdn.net/dsa63/article/details/17242111
类与对象的设计思想

20、LR的公式，以及原理（交叉熵）

23、gbdt lr 为什么比lr效果好？ https://blog.csdn.net/Losteng/article/details/78378958

1）gbdt连续特征的处理效果更好

2）gbdt+ffm（稀疏特征稠密话）+lr 更佳

24、高度稀疏的数据用树模型还是回归模型：回归模型

高维稀疏特征的时候，lr 的效果会比 gbdt 好，为什么?

这个问题我也是思考了好久，在平时的项目中也遇到了不少 case，确实高维稀疏特征的时候，使用 gbdt 很容易过拟合。

但是还是不知道为啥，后来深入思考了一下模型的特点，发现了一些有趣的地方。

假设有1w 个样本， y类别0和1，100维特征，其中10个样本都是类别1，而特征 f1的值为0，1，且刚好这10个样本的 f1特征值都为1，其余9990样本都为0(在高维稀疏的情况下这种情况很常见)，我们都知道这种情况在树模型的时候，很容易优化出含一个使用 f1为分裂节点的树直接将数据划分的很好，但是当测试的时候，却会发现效果很差，因为这个特征只是刚好偶然间跟 y拟合到了这个规律，这也是我们常说的过拟合。但是当时我还是不太懂为什么线性模型就能对这种 case 处理的好？照理说 线性模型在优化之后不也会产生这样一个式子：y = W1*f1 + Wi*fi+….，其中 W1特别大以拟合这十个样本吗，因为反正 f1的值只有0和1，W1过大对其他9990样本不会有任何影响。

后来思考后发现原因是因为现在的模型普遍都会带着正则项，而 lr 等线性模型的正则项是对权重的惩罚，也就是 W1一旦过大，惩罚就会很大，进一步压缩 W1的值，使他不至于过大，而树模型则不一样，树模型的惩罚项通常为叶子节点数和深度等，而我们都知道，对于上面这种 case，树只需要一个节点就可以完美分割9990和10个样本，惩罚项极其之小.

这也就是为什么在高维稀疏特征的时候，线性模型会比非线性模型好的原因了：带正则化的线性模型比较不容易对稀疏特征过拟合。

25.LSTM的源代码（tensorflow），让每行解释含义　三个门的设计
https://blog.csdn.net/guolindonggld/article/details/79273992

27.自然语言处理中同义词，反义词，相关词的识别情况，怎么解决
英文的可以使用nltk的wordnet实现，
中文的同义词可以使用word2vec计算得到，
https://www.cnblogs.com/chenbjin/p/7106139.html 一个新的基于word2vec的模型来搞的

28.分词的详细过程，比如说，分词好了模型，但是突然火了一个新词王者荣耀，怎么让模型匹配
最简单的方式就是添加字典，定义垂领域的字典
但说到新词的发现上，（关于新词语义的判定，方法就有很多了，典型任务包括相关词发现、领域词扩展，等等。我觉得像latent topic models、word representation、explicit semantic analysis等模型都可以用于解决这个问题。）来自刘知远老师的回答

29.经典的lstm加上hmm的分词结构是什么?
这应该是问 lstm+crf吧进行分词，
通过LSTM网络的处理，相当于得到了一个比较好的对输入数据的表示方法，LSTM单元最终输出的向量即可以看成是输入数据的一种表示形式，最终在打标签阶段，一般都采用softmax进行处理，不过这种方法在处理输出标签直接有强烈关系的数据时，效果还是有限的。特别是在实际的序列标注任务时，由于神经网络结构对数据的依赖很大，数据量的大小和质量也会严重影响模型训练的效果，故而出现了将现有的线性统计模型与神经网络结构相结合的方法，效果较好的有LSTM与CRF的结合。简单来说就是在输出端将softmax与CRF结合起来，使用LSTM解决提取序列特征的问题，使用CRF有效利用了句子级别的标记信息。

30.word2vec的具体原理，cbow实现过程，n-skim的算法实现，以及后续的负采样，层次softmax用法的原因
https://www.jianshu.com/p/14ec26a5892b

31.深度学习中relu模块在那里使用，有什么作用
https://blog.csdn.net/bojackhosreman/article/details/69372087
引入非线形因素，使数据变的更好分

32.word2vec与wordembedding的区别?
个人理解是，word embedding 是一个将词向量化的概念，来源于Bengio的论文《Neural probabilistic language models》，中文译名有"词嵌入"。
word2vec是谷歌提出一种word embedding 的工具或者算法集合，采用了两种模型(CBOW与skip-gram模型)与两种方法(负采样与层次softmax方法)的组合，比较常见的组合为 skip-gram+负采样方法。（word2vec+hsoftmax=fasttext）

33.为什么参数稀疏会好？有什么数学原理么？
https://www.jianshu.com/p/8025b6c9f6fa
这样才能模型才符合。奥卡姆剃刀原理： 在所有的可能选择的模型中，我们应该选择能够很好的解释已知数据并且十分简单的模型
最小化误差是为了让我们的模型拟合我们的训练数据，而参数化是防止我们的模型过分拟合我们的训练数据，确保泛化能力。
追求数据稀疏的原因：1）特征选择：更好的选择出重要特征 2）使模型更好解释

34.你使用过哪些损失函数?
铰链损失（hinge loss）：svm的最大边界分类 https://blog.csdn.net/u010976453/article/details/78488279。 （svm的loss 可看成是l2范数与hinge loss 的和）
交叉熵损失（softmax loss）：lr的损失和softmax的分类中 深度学习模型的分类过程
平方损失 （square loss）：最小二乘法
均方根误差（mse loss）：回归问题
指数损失 （exponential loss）：adaboost集成学习算法中

35.各种最优化方法比较 拟牛顿法和牛顿法区别，哪个收敛快？为什么?
传统机器学习中常用的机器学习方法：梯度下降、批量梯度下降、随机梯度下降、牛顿法、拟牛顿法、共轭梯度法 （各种算法的优缺点：http://www.voidcn.com/article/p-efgnrash-bhs.html）（各种算法的公式描述：http://www.cnblogs.com/maybe2030/p/4751804.html）
深度学习的时候会用到的优化方法：sgd、adagrad、adadelta、adam、adamax、nadam （https://zhuanlan.zhihu.com/p/22252270）
看完的感受：传统方法更多的是从数学理论上的求解优化、深度学习则更加的从实现角度进行完善。
梯度下降：梯度下降法是对函数进行最小化的方法，在梯度的方向上（全部数据）
随机梯度下降：（一个样本）
批量梯度下降：（一个batch）
三者都是在一阶导数（梯度）上找最优优解，差异之体现在每次的数据量上，三种方式都需要自定义学习率

牛顿法：牛顿法里使用了二阶梯度信息（海参矩阵），从而加快了每一步的收敛速度。但缺点就是每次计算都要二阶信息，计算量大——》诞生了拟牛顿法，将海参矩阵替换为一个近似的矩阵

深度学习上优化方法：sgd
Sgd：选择学习率困难，对所有的参数更新使用同样的学习率，
缺点：1、对于稀疏数据会更佳缓慢
2、容易手链到局部最优
Adagrad：对学习率进行了一个约束，前期能够放大梯度，后去能够约束梯度
非常适合处理稀疏梯度
缺点：1、仍依赖一个全局设置的学历率

adadelta：优化的adagrad，不依赖了学习率，
训练初中期加速效果很快，训练后期反复在进步最小值附近抖动

Rmsprop：是adadelte的一个特例：
适合处理非平稳目标，对rnn效果很好

adam：本质上是导动量项的rmsprop，他利用梯度的一阶矩阵估计和二阶梯度估计动态调整每个参数的学习率，adam的优点主要在于经过偏置矫正后，每一次迭代学习率都有个范围，是参数比较平稳。
适用于大数据集和高维空间，需要的内存比较小

36. 深度学习的优化方法有哪些？ sgd、adam、adgrad区别？ adagrad详细说一下？为什么adagrad适合处理稀疏梯度？（https://zhuanlan.zhihu.com/p/22252270）

adagrad 对稀疏特征采用高的更新速率，而对其他特征采用相对较低的更新速率，来适应稀疏数据的学习过程

adam：对内存需求较小，适合在大数据集和高维空间的机器学习

37.什么是梯度消失，标准的定义是什么？
明确点：发生在反向传播过程中，使用基于梯度下降的优化策略的时候，梯度的更新就是对激活函数进行求道，随着网络层数增加的时候，最终求出的梯度更新以指数的形式增加。当数值大于一 发生梯度爆炸、数值小于1发生梯度消失
发生条件：1、网络层数多。2 激活函数不合理
改善：1、修改网络结构。2 换激活函数 relu 代替sigmoid

38.DNN的初始化方法有哪些？ 为什么要做初始化？ kaiming初始化方法的过程是怎样的？ https://blog.csdn.net/u012151283/article/details/78230891

为啥：为了让神经网络在训练过程中学习到有用的信息，需要在参数更新时的梯度不为0。全零初始化方法在前向传播过程中会使得隐层神经元的激活值均未0，在反向过程中根据BP公式，不同维度的参数会得到相同的更新。 需要破坏这种“对称性”。

方法：

标准初始化：激活函数的输入值的均值为0，方差为常量1/3

xavier初始化：目的（为了使得网络中信息更好的流动，每一层输出的方差应该尽量相等。）https://blog.csdn.net/shuzfan/article/details/51338178 推到过程

he（kaiming）初始化：均值为0方差为(8)或者(15)的高斯分布。

xgboost原理介绍：

40. xgboost在什么地方做的剪枝，怎么做的？
两种方式：预剪枝，后剪枝
预剪枝：在分裂的增益小于设定的阈值时停止分类
后剪枝：达到最大深度后停止在剪枝，优点如下：

- 当分裂时遇到一个负损失时，GBM会停止分裂。因此GBM实际上是一个贪心算法。
- XGBoost会一直分裂到指定的最大深度(max_depth)，然后回过头来剪枝。如果某个节点之后不再有正值，它会去除这个分裂。
- 这种做法的优点，当一个负损失（如-2）后面有个正损失（如+10）的时候，就显现出来了。GBM会在-2处停下来，因为它遇到了一个负值。但是XGBoost会继续分裂，然后发现这两个分裂综合起来会得到+8，因此会保留这两个分裂。

17. lightgbm和xgboost有什么区别？他们的loss一样么？ 算法层面有什么区别？

lgb 选招最佳分割点的方式变成了直方图算法

带有深度限制的leaf-wise叶子生长策略

18 lightgbm有哪些实现，各有什么区别？

xgboost的剪枝打分函数是什么形式？
就是gain 那个公式形式https://blog.csdn.net/u010159842/article/details/77503930
在计算每次分类点的时候对树进行剪枝

xgb与gbdt的不同？https://www.zhihu.com/question/41354392
1、gbdt以cart为基分类器，xgb还支持线形分类器，
2、gbdt只用到了一阶导数，xgb用到了二阶
3、xgb加入了正则项，优于gbdt
4、skrinkage 缩减，相当于学习率的设置
5、xgb支持列抽样
6、对缺失数据的处理，将其放到左右两边分别计算gini，选择小的那边，自动学习分裂方向
7、特征选择上支持并行操作

自我理解：
1、xgb借鉴了rf的思想支持列列抽样（对特征采样），减少过过拟合
2、xgb支持行采样（不是样本少了会过拟合吗？可以理解成一个batch数据增加了样本的多样性）

50. 激活函数有哪些，在什么场景中使用，优缺点是什么，为什么会梯度消失
https://blog.csdn.net/NOT_GUY/article/details/78749509

	优点	缺点
sigmoid	1）便于求导的平滑函数 2）能压缩数据， 3）适合用于前向传播	1）容易出现梯度消失问题 2）输出均值不为0 3）幂运算相对耗时
tanh	均值为0 数据是对称的	存在梯度消失，和幂运算的问题
relu	1）梯度不会饱和，解决了梯度消失的问题 2）计算复杂度低 3）适合于后向传播	1）输出不是以0为中心的 2）会出现神经元坏死现象 3）不会数据压缩，数据的幅度会随着模型层数的增加不断扩张
leakly relu	解决了神经元坏死问题	表现不一定比Relu好
ELU	均值接近0，不会有坏死神经元	计算量大，表现不一定比Relu好

51.集成学习介绍（boosting bagging stacking原理）
stacking的优点为啥你要用？stacking框架是集成了不同的算法，充分利用不同算法从不同的数据空间角度和数据结构角度的对数据的不同观测 https://zhuanlan.zhihu.com/p/27493821
三种原理介绍：https://blog.csdn.net/Mr_tyting/article/details/72957853
Boosting： adaboost 残差拟合不断的减小误差，串行
bagging： rf 对样本的随机采用，并行 （对于Bagging需要注意的是，每次训练集可以取全部的特征进行训练，也可以随机选取部分特征训练，例如随机森林就是每次随机选取部分特征）notethat：随机选取特征、样本都可以、为了增加模型的泛化性能
Stacking：一种更加精巧的融合模型设计 有个很形象的图：https://blog.csdn.net/qq_18916311/article/details/78557722
还有一个blending：将每个模型的输出直接相加或者取平均， https://blog.csdn.net/u012969412/article/details/76636336
我自己的理解：stacking将预测出来的结果不是进行了简单的投票或者平均，而是更多的学习了他们的特征进行了重新选择

52.蓄水池抽样算法 https://blog.csdn.net/Azeyunyun/article/details/52550510
等概率抽样问题，在知道样本数量或者不知道样本数量（样本太多不能够统计的时候），实现等概率的抽取其中topK个，采用蓄水池算法

53.无序数组中找出第k大的树 https://blog.csdn.net/wangbaochu/article/details/52949443
1）先排序然后输入第k个位置上的数
2）改进的快速排序方法
3）大数据是采用堆排序，非常适合内存有限的情况，海量数据
考官一般是问这个解法：简历一个K的小根堆。x

54.麦穗问题的解法。 https://www.imooc.com/article/39772?block_id=tuijian_wz
从一个无序的数组中只能选择一次，能找到最大的数的办法
1、求出前 k 个数中的最大值 max_base，
从 k+1 开始，如果这个数大于 max_base，则取出这个数作为结果，
如果第 k+1 个数到最后都小于max_base，则取出最后一个数作为结果。
重复抽取 100000 次，以频率估算概率，

37%概率问题，的数学计算得到 概率公式
扩展到恋爱问题上，降低选择的标准能够得到更大的满足感

55.cnn的几点问题

1、卷积核为啥是单数?

吴恩达课程：1、方便padding 2、基数有中心点

2、参数量的计算。

56.最大子序列和，最长递增子序列，最长公共子串，最长公共子序列，字符串编辑距离，最长不重复子串，最长回文子串

子串是连续的。子序列是不连续的

查看端口是否被占用。lsof -i：80

建堆的时间复杂度： O（n） 空间复杂度O（1）

https://blog.csdn.net/YuZhiHui_No1/article/details/44258297

堆排序的过程： 把待排序的元素按照大小在二叉树位置上排列，排序好的元素要满足**父节点的元素要大于等于其子节点**（对其子节点的左右大小没要求）；整个过程叫做堆化过程，

其实整个排序过程主要核心就是堆化过程，堆化过程一般是用父节点和他的孩子节点进行比较，取最大的孩子节点和其进行交换；但是要注意这个应该是个逆过程，先排序好子树的顺序，然后在一步步往上走，到排序根节点上。

word2vec 无监督的

概率题：扔硬币连续两次出现正面

答：假设期望次数是E，我们开始扔，有如下几种情况：

- 扔到的是反面，那么就要重新仍，所以是0.5*(1 + E)
- 扔到的是正面，再扔一次又反面了，则是0.25*(2 + E)
- 扔到两次，都是正面，结束，则是0.25*2

所以递归来看E = 0.5*(1 + E) + 0.25*(2 + E) + 0.25*2，解得E = 6

有N个不同颜色的球，有放回的取，每次取1个，取M次，即Uniq(M)={不同颜色的个数}，求Uniq(M)的数学期望. (全排列问题)
https://blog.csdn.net/wangbaochu/article/details/52975746

a,b~U[0,1]，互相独立 求Max(a,b) 期望，期望是：https://www.zhihu.com/question/27913379. 答案：1/sqrt(pi) ~ 0.56

画lstm、gru的图，画cnn的图

1. 两个有序数组求中位数(leetcode)
2. 判断平衡二叉树(剑指offer)
3. 最长上升子序列(lintcode)
4. 二叉树转双向链表(剑指offer)
5. LRU cache实现(leetcode)
6. House Robber(leetcode)

3、手写代码 数组中第k大的数

数据结构bst搜索树，非遍历情况实现
二叉树的平均深度，非递归的方法，最优实现
算法题：链表翻转、判断平衡二叉树、最长公共子序列、海量数据topk问题、

单链表拟置
输出链表的倒数第k个节点
二叉树的广度优先搜索。
算法题 数组中和最大子序列
特别大的数据，每个数据有一个url地址，统计访问前20的url地址（先将url地址按照hash映射到不同文件或机器，然后再归并）

给你1-3的随机数，怎么生成1-7的随机数 并求其均值
https://blog.csdn.net/sgbfblog/article/details/7753012

堆排序原理、时空复杂度 （堆排序问的还挺多）

比价caffe，tf，mxnet这些框架的实现，优缺点
https://blog.csdn.net/MyArrow/article/details/52064608
这个介绍是16年的了，经过两年的更新有些缺点已经修改了

lr的各种问题总结：https://blog.csdn.net/u011734144/article/details/79717470

什么是python的生成器？https://foofish.net/what-is-python-generator.html

svm核函数的选择？https://blog.csdn.net/BigBzheng/article/details/51055374
ng的回答：https://toutiao.io/posts/115520/app_preview
没啥条件就选择livsvm的核函数默认参数 ，否则调参困难

Python 的is 和==有什么区别？http://www.iplaypy.com/jinjie/is.html
这道题要理解python内置的数据结构，每个元素存放了三个属性，id type value，is是比较元素的id是否相同，==比较的元素的value是否相同

python执行函数的解析顺序？http://hanjianwei.com/2013/07/25/python-mro/
1）经典类 的深度便利
2）

最新出土的面经：
1、线性回归的共线性，如何解决，为什么深度学习不强调
共线性：多变量线性回归中，变量之间由于存在高度相关关系而使回归估计不准确。共线性会造成冗余，导致过拟合。解决方法：排除变量的相关性 / 加入权重正则。

共线性会导致权重参数巨大改变，参数的方差很大不稳定。https://blog.csdn.net/xo3ylAF9kGs/article/details/78623269

解决办法：添加l2正则化 岭回归分析

2、任取圆上的三个点，其组成直角、锐角、钝角三角形的概率
https://www.zhihu.com/question/19824740
锐角三角形其要求圆心在三角形内，
概率分别为直角无穷小、锐角 1/4 钝角3/4

3、用骰子掷出1–7
一个骰子掷两次，共36种结果，令其中的三十五种1–7，剩下的一种对应【再掷两次】
比如：第一个骰子不为6时，第二个骰子掷出几就输出几（1–6，每种有五个可能）。
第一个骰子为6，且第二个骰子不为6，输出7（有五个可能）。两个骰子都是6，重掷。
于是1-7的概率都是5/36。而且永远掷6的概率是0，一定会在有限次内得到输出

4、一条绳子切成三段能组成三角形的概率。 1/4
https://blog.csdn.net/QIBAOYUAN/article/details/5978325

5、python中的try except finally 执行顺序
try中如果含有raise异常则执行到except 否则执行finally

6、决策树将一个特征全部乘以2 会有什么影响
答：没影响， 在寻中分裂点的时候 都是按照自身特征维度寻找的。

7、sql基本知识
http://www.dscademy.com/languages/sql/

8、概率论的知识，机器学习知识点总结
http://www.dscademy.com/

9、逻辑回归与线性回归的区别
https://blog.csdn.net/JoyceWYJ/article/details/51596797
一个用于分类一个用于回归

10、决策树是线性的吗？不是

11、牛顿法和拟牛顿法、梯度下降、随机梯度下降方法的介绍
特性比较：http://www.dscademy.com/#optimization

12、1*1 的卷积核的作用？https://blog.csdn.net/haolexiao/article/details/77073258
那么1×1卷积核有什么作用呢，如果当前层和下一层都只有一个通道那么1×1卷积核确实没什么作用，但是如果它们分别为m层和n层的话，**1×1卷积核可以起到一个跨通道聚合的作用，所以进一步可以起到降维（或者升维）的作用，起到减少参数的目的。**

13、海量数据寻找中位数
https://blog.csdn.net/yusiguyuan/article/details/38930789

14、最短路径算法
Floyd https://blog.csdn.net/u011285477/article/details/75096303

15、python的深浅拷贝以及应用场景
https://www.cnblogs.com/zhuifeng-mayi/p/9179647.html

16、如何理解SVM的超参数C
C是松弛变量，当C趋于无穷大时，必须每个变量都取0，问题华为了硬间隔SVM；当C趋于0时，对参数W没有任何限制，接近全0，没有任何意义。C越大训练集上的表现越好，容易造成过拟合；C越小，则要求越低容易造成欠拟合

17、gbdt如何计算变量的重要性？基尼不纯度
rf如何计算变量的重要性？ 带外错误率。 如果袋外错误率的增加较大, 说明该特征较重要, 因此使用该值作为特征的重要性分值。

18、几种优化算法的介绍：
牛顿法和拟牛顿法：
https://blog.csdn.net/wtq1993/article/details/51607040

梯度下降算法：按照导数下降最快的方向就行求解，可用于解线性优化问题
非线形优化问题：使用牛顿饭和拟牛顿法
牛顿法：每次迭代都要计算二阶泰勒展开运算量大

19、doc2vec原理介绍，
每个段落表示成为一个向量，对应矩阵D种的一个向量，每一个词表示为一个向量，对一个w中的一个列向量，段落向量和词向量通过取平均值的方式活着链接来对上下文中的下一个词进行预测

doc2vec还是借助word2vec的基础来进行的

4 hmm分词的步骤（基于统计，标注是BMES，5个基本部件初始概率矩阵、发射矩阵、转移矩阵、观察序列、隐含序列，hmm条件独立性，目标函数（联合概率））
5 推CRF（哭，还要推CRF，我先说了最大团的概念，然后继续往下推，推得不是很好，然后面试官笑了笑说这个有点难，不用往下推了）
https://blog.csdn.net/JackyTintin/article/details/79261981
CRF就是在给定X下的条件概率分布P(Y|X)，当我们计算Y中每一个节点Yv时，只需要考虑与Yv有链接的边的Y的集合中的节点和X中的集合节点，因为没有边链接额及节点与Yv完全独立

CRF与神经网络的结合在序列标注任务上效果更佳了？BI+LSTM对整个输入序列就行特征提取和建模，用非线性的模型建模发射得分；转移得分用另外的P表示，作为CRF自身的额参数，相对于常规的用于神经网络训练的目标函数，CRT数代参数的损失函数（CREF对于输出之间的关系进行了建模）

6 写出朴素贝叶斯公式，哪个是先验概率，哪个是后验概率，为什么叫朴素（？,?

其朴素体现在这个算法的简易性上，最训的 所有变量都是相互独立的假设。但是通常情况下很难做到

判别式模型：直接对条件概率进行建模。常见的判别式模型有 线性回归模型、线性判别分析、支持向量机、神经网络
生成式模型：会对x和y的联合分布p(x,y) 建模，通过贝叶斯公式来求的。常见的生成式模型有 隐马尔可夫模型、朴素贝叶斯模型、高斯混合模型、LDA
两种方式的优缺点：
2、生成式模型最终得到的错误率会比判别式模型高，但是其需要更少的训练样本就可以使错误率收敛
3、生成式模型容易过拟合，而判别式模型要解决凸优化问题
4、当添加新的类别时，生成式模型不需要重新训练，而判别式模型需要
5、生成式模型可以更好的使用无标签数据

Softmax 与cross-entropy的区别：
softmax用于计算概率分布，交叉熵用于度量两个概率分布之间的相似性
https://www.zhihu.com/question/65288314/answer/244557337
softmax 的推到
http://www.52nlp.cn/fasttext

李纪为博士解决安全性回复的双loss论文：
其中采用了两种方式修改了最大似然损失，
1、添加了本身生成句子信息的熵值来判断
2、t反向生成s的函数来计算

Catalan 数问题满足
h(0) =1 ,h(1) = 1
h(n)= h(0)*h(n-1)+h(1)*h(n-2) + ... + h(n-1)h(0) (n>=2)
使用场景：
二叉树的构建数量
一个栈无穷大元素的出栈顺序问题
2n个人买票问题
将一个凸多边形区域分成三角形区域的方法数

对评价指标BLEU的理解：
https://blog.csdn.net/wwj_748/article/details/79686042
rouge 的计算：
https://blog.csdn.net/qq_25222361/article/details/78694617
困惑度
https://blog.csdn.net/jiaqiang_ruan/article/details/77989459?locationNum=2&fps=1

CRF详细介绍 与crf 与hmm的比较
https://blog.csdn.net/class_brick/article/details/78252026
Hmm 维特比算法的实例介绍
https://blog.csdn.net/zb1165048017/article/details/48578183

GBDT 为什么是在拟合残差?
https://www.cnblogs.com/ScorpioLu/p/8296994.html 推导

GBDT 面试常问问题总结?
https://www.cnblogs.com/ModifyRong/p/7744987.html
GBDT 的特征重要性计算： 计算每个特征在每棵树上的重要性,
https://blog.csdn.net/yangxudong/article/details/53899260?locationNum=13&fps=1
RF的特种重要性选择：会有两种方式
https://blog.csdn.net/lqqlqqlqqlqq/article/details/78857411
OOV方式：
http://blog.sina.com.cn/s/blog_7103b28a0102w7q1.html

基尼不纯度与熵的关系： （基尼不纯度就是基尼系数）
http://onmyway-1985.iteye.com/blog/2083384

不均衡数据分类问题的处理办法：
https://blog.csdn.net/heyongluoyao8/article/details/49408131