

# 推荐系统面试题

## 目录

推荐系统.....	2
1. (爱奇艺) 数据不均衡的采样方法.....	2
2. (爱奇艺) 特征选择方法.....	3
3. (爱奇艺) Fm 原理.....	4
4. (百度) 给几个特征和标签, 怎么筛选有用的特征, 只用过滤法.....	4
5. (百度) 遗传算法的原理步骤.....	4
6. (百度) 模型和数据怎么存储? 分开存模型太大怎么办.....	5
7. (美团) 正负采样都哪些方法.....	5
8. (美团) sparse 特征比 dense 的优势在哪里.....	5
9. (美团) 一个推荐系统上线后, 是怎么给用户做商品推荐的.....	5
10. (腾讯) 传统机器学习方法与 DL 比较.....	5
11. (腾讯) 分布式训练框架, P-S 框架比一般的分布式框架有什么优势.....	7
12. learning-to-rank 的三大类算法: .....	7
13. (天眼查) GBDT+lr 与 Wide&Deep 算法的不同.....	7
14. ALS 算法是 LFm 算法在 spark 里的实现.....	9
15. feed 架构:.....	9
16. doc2vec.....	9
17. Embedding.....	10
18. 怎么保证在较短时间内返回推荐列表.....	11
19. 特征工程中定长序列和不定长序列处理.....	11
20. 公司具体的业务场景, 为什么选择这样的算法.....	11
21. Fm 与 DeepFm 的关系.....	12
22. 什么是基于模型推荐, 其中模型指的是什么, 如何预估模型的准确率? .....	12
23. 请用流程图画出推荐系统的整个处理流程.....	13
24. (BOSS 直聘) Fm 中如何处理数值型变量? 如何处理 multi-hot 分类变量? .....	13
25. (BOSS 直聘) 推荐系统常见的结构, 每层都需要做什么? .....	14
26. (BOSS 直聘) 在推荐系统召回中 Embedding 的方式有哪些? .....	14
27. (阿里高德) ItemCF 与 UserCF 之间的差别, 当物品冷启动时, 哪个更好一点? .	14
28. (阿里高德) DeepFm 模型中有一部分同时作为 Fm 部分与 Deep 部分的输入, 这种同时的输入与 Wide&Deep 中分开的输入有什么区别? .....	14
29. (360 金融) 常见的推荐架构, 以及召回算法, 以及召回的目的是什么? .....	15
30. (360 金融) 召回的常见方式有哪些? .....	15

## 推荐系统

### 1. (爱奇艺) 数据不均衡的采样方法

(1). 从数据角度:

主动获取: 获取更多的少量样本数据 (针对少量样本数据, 可以尽可能去扩大这些少量样本的数据集, 或者尽可能去增加他们特有的特征来丰富数据的多样性。譬如, 如果是一个情感分析项目, 在分析数据比例时发现负样本 (消极情感) 的样本数量较少, 那么我们可以尽可能在网站中搜集更多的负样本数量)

算法采样: 上采样、下采样、生成合成数据

ADASYN 采样方法:

ADASYN 为样本较少的类生成合成数据, 其生成的数据与更容易学习的样本相比, 更难学习。基本思想是根据学习难度的不同, 对不同的少数类的样本使用加权分布。其中, 更难学习的少数类的样本比那些更容易学习的少数类的样本要产生更多的合成数据。因此, ADASYN 方法通过以下两种方式改善了数据分布的学习: (1) 减少由于类别不平衡带来的偏差; (2) 自适应地将分类决策边界转移到困难的例子。

SMOTE 采样方法:

从少数类创建新的合成点, 以增加其基数。但是 SMOTE 算法也有一定的局限性。具体有两项, 一是在近邻选择时, 存在一定的盲目性。在算法执行过程中, 需要确定 K 值, 即选择几个近邻样本, 这个需要根据具体的实验数据和实验人自己解决。二是该算法无法克服非平衡数据集的数据分布问题, 容易产生分布边缘化的问题。由于负类样本的分布决定了其可选择的近邻, 如果一个负类样本处在负类样本的边缘, 则由此负类样本和近邻样本产生的样本也会处在边缘, 从而无法确定正负类的分类边界。下图是以前做的一个项目应用个各种采样方法做数据增强的情况。(效果不明显, 因为原始数据的分布重合太明显, 可视化不容易显示出效果))

数据增强: 加噪音增强模型鲁棒性、对不同性质的数据也可以做不同的 augmentation

改变权重：设定惩罚因子，如 libsvm 等算法里设置的正负样本的权重项等。  
惩罚多样本类别，其实还可以加权少样本类别

## (2). 从评价指标角度

x 谨慎选择 AUC 作为评价指标：对于数据极端不平衡时，可以观察观察不同算法在同一份数据下的训练结果的 precision 和 recall，这样做有两个好处，一是可以了解不同算法对于数据的敏感程度，二是可以明确采取哪种评价指标更合适。针对机器学习中的数据不平衡问题，建议更多 PR(Precision-Recall 曲线)，而非 ROC 曲线，具体原因画图即可得知，如果采用 ROC 曲线来作为评价指标，很容易因为 AUC 值高而忽略实际对少两样本的效果其实并不理想的情况。

不要只看 Accuracy：Accuracy 可以说是最模糊的一个指标了，因为这个指标高可能压根就不能代表业务的效果好，在实际生产中，我们可能更关注 precision/recall/mAP 等具体的指标，具体侧重那个指标，得结合实际情况看。

## (3). 从算法角度

选择对数据倾斜相对不敏感的算法。如树模型等。

集成学习(Ensemble 集成算法)。首先从多数类中独立随机抽取出若干子集，将每个子集与少数类数据联合起来训练生成多个基分类器，再加权组成新的分类器，如加法模型、Adaboost、随机森林等。

将任务转换成异常检测问题。譬如有这样一个项目，需要从高压线的航拍图片中，将松动的螺丝/零件判断为待检测站点，即负样本，其他作为正样本，这样来看，数据倾斜是非常严重的，而且在图像质量一般的情况下小物体检测的难度较大，所以不如将其转换为无监督的异常检测算法，不用过多的去考虑将数据转换为平衡问题来解决。

## 2. (爱奇艺) 特征选择方法

根据特征选择的形式，可分为三大类：

Filter(过滤法)：按照发散性或相关性对各个特征进行评分，设定阈值或者待选择特征的个数进行筛选

Wrapper(包装法)：根据目标函数（往往是预测效果评分），每次选择若干特征，或者排除若干特征

Embedded(嵌入法): 先使用某些机器学习的模型进行训练, 得到各个特征的权值系数, 根据系数从大到小选择特征 (类似于 Filter, 只不过系数是通过训练得来的)

### 3. (爱奇艺) Fm 原理

Fm 的全称是 Factorization Machines, 就是因子分解机的意思, 为什么叫因子分解呢, 就是因为他对传统的线性回归模型加了一个因子交叉项, 是由 Konstanz 大学 Steffen Rendle 于 2010 年最早提出的, 旨在解决稀疏数据下的特征组合问题。Fm 是一种比较灵活的模型, 通过合适的特征变换方式, Fm 可以模拟二阶多项式核的 SVM 模型、MF 模型、SVD++模型等, 相比 SVM 的二阶多项式核而言, Fm 在样本稀疏的情况下是有优势的; 而且, Fm 的训练/预测复杂度是线性的, 而二项多项式核 SVM 需要计算核矩阵, 核矩阵复杂度就是 N 平方。对于 Fm 而言, 我们可以加任意多的特征, 比如 user 的历史购买平均值, item 的历史购买平均值等

### 4. (百度) 给几个特征和标签, 怎么筛选有用的特征, 只用过滤法

首先可以根据特征的分布差异 (方差) 来筛选, 去掉方差小的  
还可以计算特征和标签的相关度来排序筛选

### 5. (百度) 遗传算法的原理步骤

遗传算法可以用来求函数的极值。

(1) 用二进制编码来离散自变量, 码长根据离散精度来确定。码长为  $\log_2[(\max - \min)/\text{精度} + 1]$

(2) 交叉方法采用单点交叉

(3) 变异是根据变异概率反转子代某个位的值

(4) 选择策略采用轮盘赌策略, 令  $PP_i = \sum_{j=1}^i p_j$ ,  $PP_0 = 0$ , 其中  $PP_i$  为累计概率,  $p_i$  为个体的选择概率, 公式为  $p_i = \frac{\text{fitness}(x_i)}{\sum_{i=1}^{NP} \text{fitness}(x_i)}$ ,

其中  $\text{fitness}(x_i)$  为个体的适应度, 共轮转 NP 次, 每次轮转时, 产生随机数  $r$ , 当  $PP_{i-1} \leq r < PP_i$  时选择个体

## 算法步骤

基本遗传算法的基本步骤是：

1. 随机产生种群，
2. 用轮盘赌策略确定个体的适应度，判断是否符合优化准则，若符合，输出最佳个体及其最优解，结束，否则，进行下一步
3. 依据适应度选择再生个体，适应度高的个体被选中的概率高，适应度低的个体被淘汰
4. 按照一定的交叉概率和交叉方法，生成新的个体
5. 按照一定的变异概率和变异方法，生成新的个体
6. 由交叉和变异产生新一代种群，返回步骤2

## 6.（百度）模型和数据怎么存储?分开存模型太大怎么办

一方面可以做模型压缩（比如深度学习知识蒸馏、量化压缩）

另外一方面可以切片存储模型

## 7.（美团）正负采样都哪些方法

过采样：重复正比例数据，实际上没有为模型引入更多数据，过分强调正比例数据，会放大正比例噪音对模型的影响。

欠采样：丢弃大量数据，和过采样一样会存在过拟合的问题。

随机采样：由于随机过采样采取简单复制样本的策略来增加少数类样本，这样容易产生模型过拟合的问题，即使得模型学习到的信息过于特别而不够泛化。

## 8.（美团）sparse 特征比 dense 的优势在哪里

Memorization的一个典型模型就是使用大量的原始sparse特征和叉乘特征作为输入的LR模型，很多原始的dense特征通常也会被分桶离散化构造为sparse特征。这种做法的优点是：模型可解释高，实现快速高效，特征重要度易于分析，在工业界已被证明是有很有效的。这种做法的缺点是：1、需要大量人工进行的特征工程；2、容易过拟合。当我们进行细粒度的特征叉乘时，我们的模型实际上可以视作将训练集记忆下来，过拟合非常严重，较粗粒度的特征叉乘可以一定程度上缓解这个问题；3、无法捕捉训练数据中未曾出现过的特征对，即泛化能力较差。

## 9.（美团）一个推荐系统上线后，是怎么给用户做商品推荐的

采集 user 和 item 的 feature 构建特征，送入模型预测

## 10.（腾讯）传统机器学习方法与 DL 比较

机器学习

机器学习是人工智能的一个子集，它利用统计技术提供了向计算机“学习”数据的能力，而不需要复杂的编程。简单来说，机器学习可以被定义为一种科学，它使计算机像人类一样行动和学习，并通过以实际交互和观察的形式向他们提供信息和数据，以独立的方式提高他们的学习能力。机器学习鼓励各种行业的各种自动化跨度和任务，从分析恶意软件或数据安全公司到寻求有利交易的财务专家，都是机器学习的应用场景。

让我们举一个著名的音乐流媒体服务的例子，该服务必须决定应该向听众推荐哪个新的艺术家或歌曲。机器学习算法帮助听众选择具有相同品味的其他听众。在这种情况下，机器学习将作为虚拟助手工作，为用户提供有关音乐行业新口味和需求的信息，系统可以根据这些信息向听众推荐新歌。

### 深度学习

与特定于任务的算法不同，深度学习是基于学习数据的机器学习的子集。它的灵感来自被称为人工神经网络的功能和结构。深度学习通过学习将世界显示为更简单的概念和层次结构，以及基于不那么抽象的概念来计算更抽象的代表，从而获得巨大的灵活性和力量。尽管深度学习这个词现在已经说了好几年了，但是现在所有人都在大肆宣传，它正受到越来越多的关注。

为了理解这个概念，举一个动物识别器的例子，它有助于识别给定的图像是狮子还是鹿。当我们将此解决为传统的机器学习问题时，我们将涉及特定的特征，比如说给定的动物是否有耳朵，是否有胡须或任何其他器官。简单来说，我们将定义面部特征，让系统识别动物。另一方面，在深度学习中，从第一步开始，深度学习将自动对关键特征进行定义和分类。深度学习将首先确定找出狮子或鹿的最相关因素。稍后它将开始识别形状和边缘的组合，以更深入地识别对象。例如，如果对象有耳朵或者有胡须。在定义了这些概念的连续分层识别之后，它将决定哪些特征负责找到正确的答案。

## 11. (腾讯) 分布式训练框架, P-S 框架比一般的分布式框架有什么优势

### 1. Efficient communication:

由于是异步的通信, 因此, 不需要停下来等一些机器执行完一个iteration (除非有必要), 这大大减少了延时。为机器学习任务做了一些优化(后续会细讲), 能够大大减少网络流量和开销;

### 2. Flexible consistency models:

宽松的一致性要求进一步减少了同步的成本和延时。parameter server 允许算法设计者根据自身的情况来做算法收敛速度和系统性能之间的trade-off。

### 3. Elastic Scalability:

使用了一个分布式hash表使得新的server节点可以随时动态的插入到集合中; 因此, 新增一个节点不需要重新运行系统。

### 4. Fault Tolerance and Durability:

我们都知道, 节点故障是不可避免的, 特别是在大规模商用服务器集群中。从非灾难性机器故障中恢复, 只需要1秒, 而且不需要中断计算。Vector clocks 保证了经历故障之后还是能运行良好;

### 5. Ease of Use:

全局共享的参数可以被表示成各种形式: vector, matrices 或者相应的sparse类型, 这大大方便了机器学习算法的开发。并且提供的线性代数的数据类型都具有高性能的多线程库。

## 12.learning-to-rank 的三大类算法:

Pointwise

Pairwise

Listwise

## 13. (天眼查) GBDT+lr 与 Wide&Deep 算法的不同

GBDT+LR 算法

通过 GBDT 进行特征组合之后得到的离散向量是和训练数据的原特征一块作为逻辑回归的输入, 而不仅仅全是这种离散特征

建树的时候用 ensemble 建树的原因就是一棵树的表达能力很弱, 不足以表达多个有区分性的特征组合, 多棵树的表达能力更强一些。GBDT 每棵树都在学习前面棵树尚存的不足, 迭代多少次就会生成多少棵树。

RF 也是多棵树, 但从效果上有实践证明不如 GBDT。且 GBDT 前面的树, 特征分裂主要体现在对多数样本有区分度的特征; 后面的树, 主要体现的是经过前 N 颗树, 残差仍然较大的少数样本。优先选用在整体上有区分度的特征, 再选用针对少数样本有区分度的特征, 思路更加合理, 这应该也是用 GBDT 的原因。

在 CRT 预估中, GBDT 一般会建立两类树(非 ID 特征建一类, ID 类特征建

一类), AD, ID 类特征在 CTR 预估中是非常重要的特征, 直接将 AD, ID 作为 feature 进行建树不可行, 故考虑为每个 AD, ID 建 GBDT 树。

非 ID 类树: 不以细粒度的 ID 建树, 此类树作为 base, 即便曝光少的广告、广告主, 仍可以通过此类树得到有区分性的特征、特征组合

ID 类树: 以细粒度的 ID 建一类树, 用于发现曝光充分的 ID 对应有区分性的特征、特征组合。

W&D 模型是将两部分输出的结果结合起来联合训练, 将 Deep 和 Wide 部分的输出重新使用一个逻辑回归模型做最终的预测, 输出概率值。

#### 操作流程

Retrieval : 利用机器学习模型和一些人为定义的规则, 来返回最匹配当前 Query 的一个小的 items 集合, 这个集合就是最终的推荐列表的候选集。

#### Ranking:

a. 收集更细致的用户特征, 如:

User features (年龄、性别、语言、民族等)

Contextual features (上下文特征: 设备, 时间等)

Impression features (展示特征: app age、app 的历史统计信息等)

b. 将特征分别传入 Wide 和 Deep 一起做训练。在训练的时候, 根据最终的 loss 计算出 gradient, 反向传播到 Wide 和 Deep 两部分中, 分别训练自己的参数 (Wide 组件只需要填补 Deep 组件的不足就行了, 所以需要比较少的 cross-product feature transformations, 而不是 full-size Wide Model)

训练方法是用 mini-batch stochastic optimization。

Wide 组件是用 FTRL (Follow-the-regularized-leader) + L1 正则化学习。

Deep 组件是用 AdaGrad 来学习。

c. 训练完之后推荐 TopN

所以 Wide&Deep 模型尽管在模型结构上非常的简单, 但是如果想要很好的使用 Wide&Deep 模型的话, 还是要深入理解 业务, 确定 Wide 部分使用哪部分特征, Deep 部分使用哪些特征, 以及 Wide 部分的交叉特征应该如何去选择。



## 14.ALS 算法是 LFM 算法在 spark 里的实现

spark 在协同过滤这一块却一直以来都只有 ALS 一种算法。Hadoop 中的机器学习算法库 Mahout 就集成了多种推荐算法，不但有 user-cf 和 item-cf 这种经典算法，还有 KNN、SVD，Slope one 这些。交替最小二乘法，它只是是一种优化算法的名字，被用在求解 spark 中所提供的推荐系统模型的最优解。

## 15.feed 架构:

多路召回，实时更新一些特征，用模型预测

考察特征重要性是，固定其他特征，单看要考察特征，特征变化前后和转化点击相关性

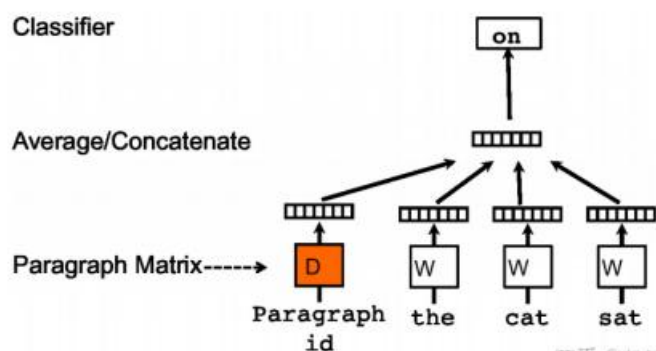
## 16.doc2vec

许多机器学习算法需要的输入是一个固定长度的向量，当涉及到短文时，最常用的固定长度的向量方法是词袋模型。尽管它很流行，但是词袋模型存在两个主要的缺点：一个是词袋模型忽略词序，如果两个不同的句子由相同的词但是顺序不同组成，词袋模型会将这两句话定义为同一个表达；另一个是词袋模型忽略了句法，这样训练出来的模型会造成类似 ‘powerful ‘， ‘strong’ 和 ‘Paris’ 的距离是相同的，而其实 ‘powerful’ 应该相对于 ‘Paris’ 距离 ‘strong’ 更近才对。

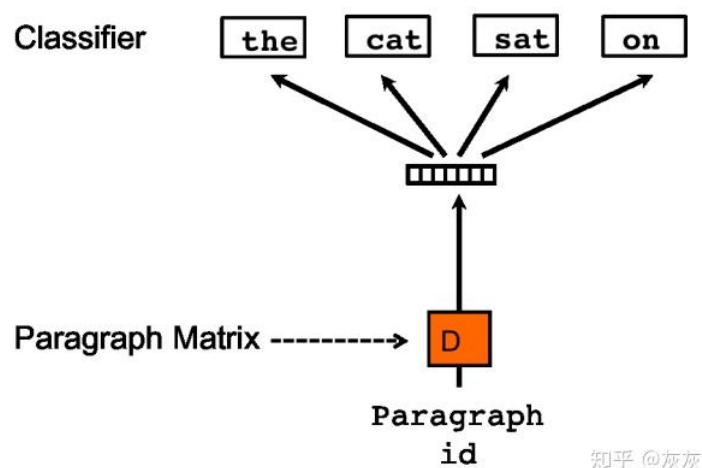
Doc2vec 又叫 Paragraph Vector，是 Tomas Mikolov 基于 word2vec 模型提出的，他的一些优点有：比如不用固定句子长度，接受不同长度的句子做训练样本。Doc2vec 是一个无监督学习算法，该算法用于预测一个向量来表示不同的文档，该模型的结构潜在的克服了词袋模型的缺点。

Doc2vec 模型是受到了 word2vec 模型的启发，word2vec 里预测词向量时，预测出来的词是含有词义的，比如上文提到的词向量 ‘powerful’ 会相对于 ‘Paris’ 离 ‘strong’ 距离更近，在 Doc2vec 中也构建了相同的结构。所以 Doc2vec 克服了词袋模型中没有语义的缺点。假设现在存在训练样本，每个句子都是训练样本。和 word2vec 一样，Doc2vec 也有两种训练方式，一种是 PV-DM(Distributed Memory Model of paragraph vectors) 类似于 word2vec 中的 CBOW 模型，如图

一:



另一种是 PV-DBOW(Distributed Bag of Words of paragraph vector) 类似于 Word2vec 中的 skip-gram 模型，如图二：



## 17.Embedding

在深度学习实验中经常会遇 Embedding 层, 然而网络上的介绍可谓是相当含糊。比如 Keras 中文文档中对嵌入层 Embedding 的介绍除了一句“嵌入层将正整数（下标）转换为具有固定大小的向量”之外就不愿做过多的解释。那么我们为什么要使用嵌入层 Embedding 呢？主要有这两大原因：

1、使用 One-hot 方法编码的向量会很高维也很稀疏。假设我们在做自然语言处理中遇到了一个包含 2000 个词的字典，当使用 One-hot 编码时，每一个词会被一个包含 2000 个整数的向量来表示，其中 1999 个数字是 0，如果字典再大一点，这种方法的计算效率会大打折扣。

训练神经网络的过程中，每个嵌入的向量都会得到更新。通过上面的图片我

们就会发现在多维空间中词与词之间有多少相似性,这使我们能可视化的了解词语之间的关系,不仅仅是词语,任何能通过嵌入层 Embedding 转换成向量的内容都可以这样做。

## 18. 怎么保证在较短时间内返回推荐列表

这是一个工程问题

① 首先,线上模型部署一定不会用 python 的,如果是 Java 的话,可以用 PMML,使用 jpmm1 可以直接部署

② 其次,从召回过来的数据,一定要组织成一批数据一起预估,每一条都是 [user, item, 交互特征] 组成的特征向量,一起预测然后拆分

③ 进一步的优化涉及到架构和开发了,在大厂,并发量非常大,通常会对模型切片部署到多台服务器上,然后在接受请求端可以做负载均衡

## 19. 特征工程中定长序列和不定长序列处理

如果是用户行为序列,在机器学习算法里,大多数情况下是统计之后,作为统计值特征使用;在深度学习里,可以用 word2vec 对他们先建模,得到 Embedding 以后放到后续 CTR 网络用,也可以用 LSTM 之类的先对序列建模,然后拿到 hidden state 作为用户的动态信息表示,送进后续 CTR 网络学习。

## 20. 公司具体的业务场景,为什么选择这样的算法

回答的时候,结合整个开发的便捷性,架构的一致性回答就好了

比如 Java 为主业务开发语言的公司,数据生态一般都是 spark 的,这样 spark 做数据处理特征工程 ETL + spark MLlib/xgboost4j 就是很合理的搭配。

再比如自己的业务场景下,如果数据是稠密形态,类别型变量不多,这个时候就特别适合 xgboost

如果数据是稀疏态,或者类别型数据很多,就适合用深度学习,对类别 ID 型变量做 Embedding 处理

## 21.Fm 与 DeepFm 的关系

Fm 通过对每一维特征的隐变量内积来提取特征组合，最后的结果也不错，虽然理论上 Fm 可以对高阶特征组合进行建模，但实际上因为计算复杂度原因，一般都只用到了二阶特征组合。对于高阶特征组合来说，我们很自然想到多层神经网络 DNN。DeepFm 目的是同时学习低阶和高阶的特征交叉，主要由 Fm 和 DNN 两部分组成，底部共享同样的输入

## 22.什么是基于模型推荐，其中模型指的是什么，如何预估模型的准确率？

基于模型的推荐是与基于记忆的推荐相对应的。

基于记忆的推荐使用用户对物品的历史行为来统计得到物品间的相似度（或用户间的相似度），从而给用户推荐从来没见过的商品。

基于模型的推荐使用一些机器学习算法来对用户向量或者物品向量进行计算，然后建立模型来预测用户对于新的物品的得分。

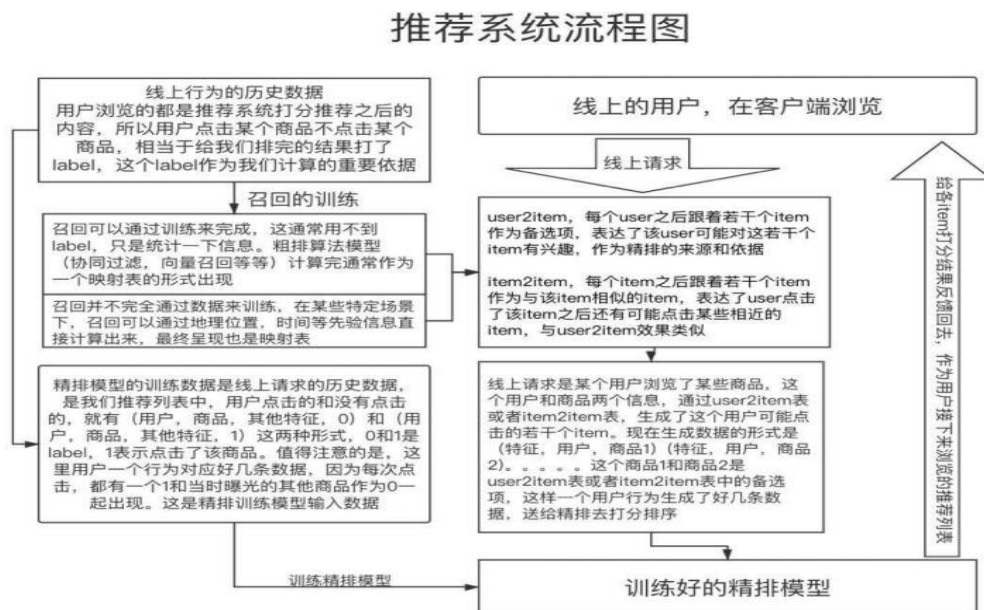
这里的模型应该是指矩阵分解、贝叶斯网络等机器学习算法。

1. 准备包含用户的历史行为记录的数据集。然后按时间将数据集分为训练集和测试集。

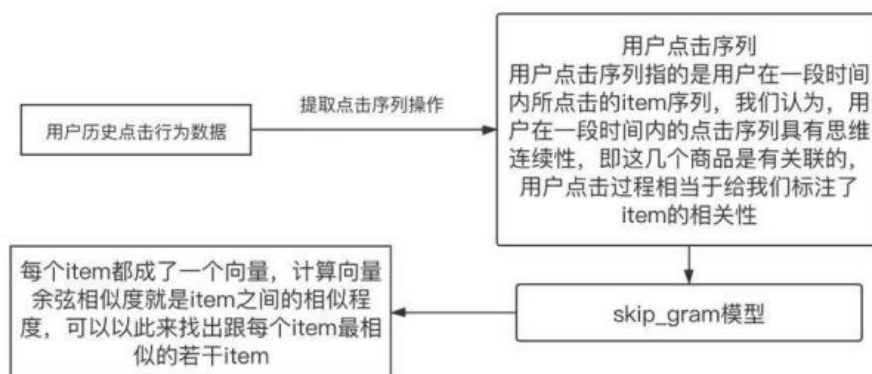
2. 最后，通过在训练集上建立的用户的行为与兴趣模型预测用户在测试集上的行为，并计算预测行为与测试

集上实际行为的重合度作为预测准确率。

## 23.请用流程图画出推荐系统的整个处理流程



## 向量召回I2I方法



## 24.（BOSS 直聘） Fm 中如何处理数值型变量？如何处理 multi-hot 分类变量？

Fm 模型与 Ffm 模型天然为类别型特征设计，如果不处理数值型特征，以年龄为代表，那 1 岁和 99 岁的在二阶特征交叉部分得到的结果差距极大。相当于对隐向量做了缩放。

基于以上理由要对数值型特征进行离散化处理。离散化处理的方式有很多可以等距分桶，等频分桶，也可以根据业务知识分桶。

## 25. (BOSS 直聘) 推荐系统常见的结构, 每层都需要做什么?

用户请求

召回

排序

展示

召回+排序是当前推荐系统的一个重要构建方式

Embedding 是一种对非数字、高度稀疏等特点的特征进行表征的重要方法。

问题的抽象、设计是一个问题成功解决的重要一环

特征的合理构建和整理 (即特征工程, 有的公司甚至专门设置了有关岗位)

多领域的结合与应用

## 26. (BOSS 直聘) 在推荐系统召回中 Embedding 的方式有哪些?

基于用户点击行为序列做 item2item 的 Embedding , 基于 word2vec

基于用户点击行为使用 DNN 的方式做 user 与 item 的 Embedding。user 的 Embedding 是最后 relu 层的输出, item 的 Embedding 是最后 softmax 层的权重。

## 27. (阿里高德) ItemCF 与 UserCF 之间的差别, 当物品冷启动时, 哪个更好一点?

从原理上讲, ItemCF 是计算物品与物品间的相似度。UserCF 是计算用户与用户间的相似度。

当新物品来临时, UserCF 效果比较好, 一旦用户对物品产生行为, 可以将新物品推荐给相似用户。

当新用户来临时, ItemCF 效果比较好, 一旦新用户对物品产生相似度, 可以将物品的相似物品推荐给用户。

## 28. (阿里高德) DeepFm 模型中有一部分同时作为 Fm 部分与 Deep 部分的输入, 这种同时的输入与 Wide&Deep 中分开的输入有什么区别?

Wide 模型部分由 LR 替换为 Fm。Fm 模型具有自动学习交叉特征的能力, 避免

了原始 Wide & Deep 模型中浅层部分人工特征工程的工作。共享原始输入特征。DeepFm 模型的原始特征将作为 Fm 和 Deep 模型部分的共同输入，保证模型特征的准确与一致。

是考察 Wide&Deep 中 Wide 与 Deep 部分的区别

Wide 部分负责记忆，Deep 部分负责泛化。

## **29. (360 金融) 常见的推荐架构，以及召回算法，以及召回的目的是什么？**

常见的推荐架构可以看做数据层，召回层，排序层。

数据层包括数据生成和数据存储，主要是利用各种数据处理工具对原始日志进行清洗，处理成格式化的数据，落地到不同类型的存储系统中，供下游的算法和模型使用。

召回根据用户的部分特征，从海量物品库中找出可能用户感兴趣的少部分物品交给排序环节。

排序层主要是利用机器学习的模型对召回层筛选出来的候选集进行精排序。

召回强调快，尽快从海量数据中筛选出少量可能需要的物品。

## **30. (360 金融) 召回的常见方式有哪些？**

基于兴趣的召回

基于地点的召回

基于协同过滤算法的召回

基于热门的召回等等。

召回可以看做单特征排序模型的排序结果