一、作业讲解部分

关于召回阶段使用的模型
早期的时候，因为数据量不大、用户不多，占用的内存不大，因此ItemCF、userCF、热门推荐这些算法为主要的召回算法。（因为之前GPU还不成熟，内存也小）

现在，deepfm 这类的深度表示学习方法比较主流，特别是对于你说的用户多商品多这种场景，并且现在也正是深度学习火热的时候，相关研究也比较多。但也不是说传统的召回 cf 这类的完全不用了，传统方法更直接且成本较低，虽然研究价值少但是应用价值应该还是有的。

特征工程的思路：
- 需要处理垃圾用户，较少噪音，去除干扰
- 垃圾用户包括：僵尸用户、刷分用户、广告用户

推荐系统与收益的关系
- 推荐系统模型提升0.01，能带来好几百万的效益

名词解释
- ctr：点击率，有没有点击打开来看
- cvr：转化率，看了然后还去了那个网站，或者下载，买vip

看可推荐的情况
需要了解，还有多少内容是用户还没有看过的，也就是看看还可以推荐的程度有多少

二、多路召回方面

多路召回的思路
将召回的内容进行拼接，然后对对应的兴趣分数进行加总。即concat + groupby + sum

召回与排序的关系
召回 —> 得到召回数据集 —>进行建模排序
排序的数据集集，是由召回的数据构建的，因此排序建模效果完全取决于召回数据集的质量好坏。
排序是通过模型建模，通过得到的proba大小进行排序，而01则是转化过的标签结果

评估指标
内部评估效果—内部衡量推荐效果的好坏
- HR值： 即Hit Rate 命中率，上面是命中的hits，下面是用户总数。它代表着推荐了多少个，然后命中了多少个的含义
- mrr：考虑排序的命中率，命中1/排序位置
- mrp：直接对所有mrr取均值
- map：什么意思？它是一种什么概念

外部评估效果
- 访客数 UV
- 浏览数 PV
- 平均访问时长
- 转化率=转化次数/访问次数
- pv点击率=pv点击/pv
- uv点击率=点击uv/整个产品的
- uv 曝光点击率=点击量/曝光次数

满意度指标
- 留存率（x日后仍活跃的用户数/x日前的用户数）
- 停留时间长（实际播放时间OR进度条时长，一般前者）
- 播放完成率（播放时长/视频时长）

因此，衡量推荐系统的好坏，除了可以用内部的评估指标，也可以通过外部的实际效果来衡量。同样，在网页优化、推广、广告、引流效果上，也可以通过这些外部的评估指标来衡量效果。

FM方面

为什么要有FM?
在传统的CF以及MF场景下，仅仅只能使用两个维度（用户id、商品id）进行评估。但实际场景中，我们仍有许多特征可以用作建模的依据，此时CF与MF就出现了局限。因此，我们需要实现一个可以使用多个特征建模拟合用户兴趣分数的模型。

为什么不是逻辑回归
尽管逻辑回归可以使用多个特征进行建模，但是逻辑回归有几个弊端。其一，单纯的逻辑回归拟合非线性的能力较弱；其二，如果尝试通过多项式组合的方式以此增强模型的拟合能力，在大量onehot且特征两两组合之后，这必然就会出现数据大量稀疏的问题。然而逻辑回归对于稀疏的数据效果并不好，在极稀疏的数据集下没法取到真正的权值。

单阶+二阶组合的逻辑回归（FM的计算表达式）

$$y = w_0 + \sum_{i=1}^{n} w_i \cdot x_i + \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} \cdot x_i \cdot x_j$$

n 代表样本的特征数量，x_i 是第 i 个特征的值，w_0、w_i、w_{ij} 是模型参数。

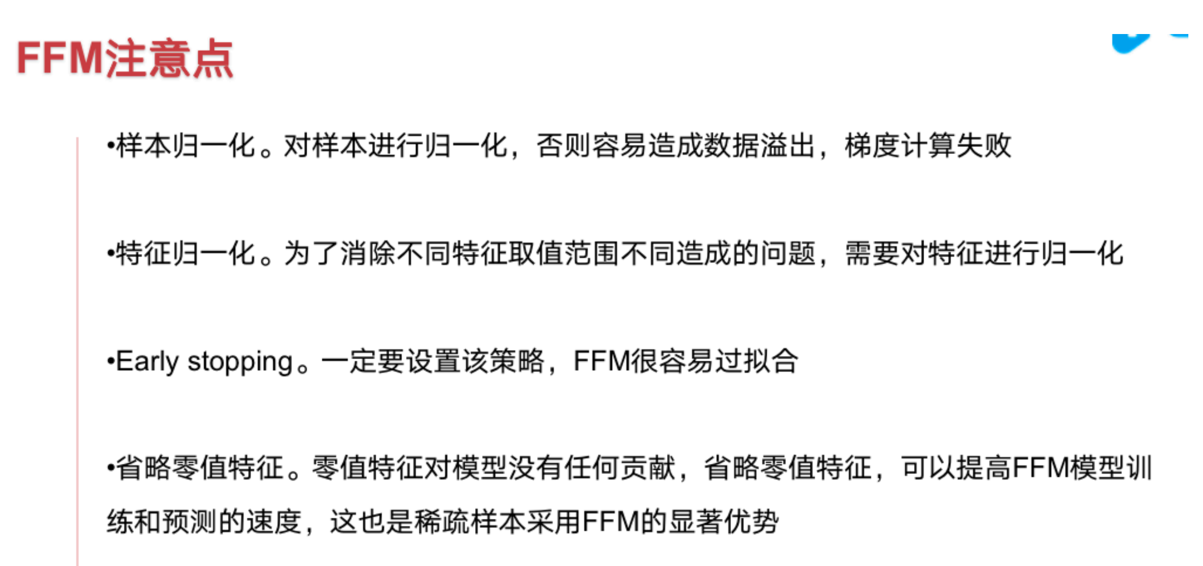
FM解决了什么问题
- 解决了逻辑回归拟合能力弱的问题—二阶特征组合
- 解决了稀疏性的问题—引用了MF拟合每一个权重，以此达到补全稀疏矩阵的问题
- 解决了参数量大的问题--通过矩阵对称公式性质，减少了一半的计算量。时间复杂度由 KN^2 变为 KN

FM也可以用于召回
因为FM中自带了特征二项式的组合，因此可以组合出userid+itemid的组合。这个情况下，它前面的w，我们可以视同为相似分数，然后以那个分数为依据，进行召回

FFM方面

FFM与FM的区别 — 带场数据集

FFM相对于FM而言，在数据集中加入了一个场的概念。因为场概念的出现：
1、使得原始的FM数据集变得具备可读性，知道哪些数据对应哪些特征
2、解决了FM数据集大量信息意义不明确，导致的信息间接流失的问题
3、在计算的时候，因着多出了场的概念，间接增加了一些隐特征，在特征的信息的提取效果上有了提升，可间接提升模型的拟合效果


FFM注意点

•样本归一化。对样本进行归一化，否则容易造成数据溢出，梯度计算失败

•特征归一化。为了消除不同特征取值范围不同造成的问题，需要对特征进行归一化

•Early stopping。一定要设置该策略，FFM很容易过拟合

•省略零值特征。零值特征对模型没有任何贡献，省略零值特征，可以提高FFM模型训练和预测的速度，这也是稀疏样本采用FFM的显著优势

libffm数据的转化
- 使用官方数据集进行转化
- 使用老师自写函数（保存到mytools里面）（写作业的时候尝试使用）

DeepFM方面

什么是DeepFM
DeepFM是DNN+FM的组合，它的出现是因着传统机器学习的背景下，超过二阶的特征组合计算量过于庞大，计算机难以进行计算，而高阶的组合又具备实际的意义。
因此为了解决这个问题，人们想到了使用Embedding+DNN的方式，实现对高阶特征组合计算的效果

DeepFM的结构组成
- 原始特征层：该部分为原始特征，该阶段还没有进行特征组合，特征组合是在FM、DNN内部完成
- embedding特征层：该层的目的是为了降维，将超稀疏超大维度的矩阵，使用几个隐变量来进行代替。达到降维的效果
- FM层：使用embedding的特征，进行1-2阶特征的计算
- DNN层：使用embedding的特征，进行高阶特征的计算
- 输出层：将FM、DNN的结果相加，得到DeepFM的结果

embedding使用方法的感悟：
- 语意表达、具体化：数个隐变量表达一个变量，使得之前没有关联的变量之间，有了共通但程度不同的隐变量。因此实现比较、对比的效果
- 降维度：使用少部分隐变量，用来代表多个变量，以此实现降维的效果

扩展内容

1、什么是Double的数据格式(需要查询)
Double型数据，即双精度浮点型，是计算机使用的一种资料型别，double(双精度浮点数)使用 64 位（8字节） 来储存一个浮点数。 因此，在paddle中，float64 也被识别为double类型

double : 双精度浮点数
float : 单精度浮点数

两者的主要区别如下：
1、在内存中占有的字节数不同：在机内存占4个字节，double在机内存占8个字节。
2、有效数字位数不同：float有效数字8位，double有效数字16位。
3、数值取值范围：float的表示范围：-3.40E+38~3.40E+38，double的表示范围：-1.79E+308~-1.79E+308.
4、在程序中处理速度不同：一般来说，CPU处理单精度浮点数的速度比处理双精度浮点数快，如果不声明，默认小数为double类型，所以如果要用float的话，必须进行强转

2、就是实际工作中，数据量很少，我怎么获得更多的训练集数据
- 要是总样本量就很少，那就考虑使用数据增强，比如SMOTE；要是有大量未标记数据，可以看看迁移学习。
- 这个问题基本没啥特别好的解决方法，基本就是特征筛选去除无用特征，减少特征维度；然后数据增强使用简单模型进行拟合。数据量少意味着信息极有可能不齐全，在残缺的信息下训练的结果也是残缺的。因此，即便通过技术手段增加数据量，意义也不大，因为数据量虽然变多了，但是蕴含的信息还是这么多。 与其如此，不如在特征上下点功夫，在现有特征上提取更多、更直接的信息，或许对提升模型效果更有帮助
- 小样本情况下，样本蕴含的x和y的关系可能并不完整，所以过多特征反而可能会相当于引入噪声，导致模型过拟合。所以在小样本下需要借助一些统计方法删掉没用特征，相当于删去特征。增强数据是为了让模型充分学习。假如说我们在特征筛选部分已经确定留下来的特征都是和目标有关的，那我们就需要模型去拟合我们认为的这种正确关系。但对于模型来说，模型越复杂需要的学习样本越多，所以小样本下需要一方面增强数据一方面采用简单模型

经验与技巧（需要研究的部分）

1、多路召回的实现（现成数据，进行召回）（尝试建模排序）（尝试使用surprise进行双路召回+构建数据集+建模排序）

2、pandas中提取列表的方法
生成列表的内置方法 agg(list) （需要尝试）

3、数学优化方法（+1防止为0）
+1的目的是为了防止分子分母为0，最终的结果是要算相似度的吧

```

numerator = ratings.select("rating").count()


num_users = ratings.select("userId").distinct().count()
num_items = ratings.select("movieId").distinct().count()

denominator = num_users * num_items

sparsity = (1.0 - (numerator * 1.0)/ denominator) * 100
print("The ratings dataframe is ", "%.2f" % sparsity + "% empty.")
```

```
The ratings dataframe is  99.53% empty.
```

给好好学习、微笑助教
助教你好，不晓得好好学习和微笑是两个助教，还是一个助教两个号～
一直以来在课堂上以及在作业中都有看到你在与我互动。每一次作业都能感受到你有很认真的在看，并且给出相关的建议。可惜的是我并不是名企班的，看完建议之后没法跟你继续讨论。
另外，在课堂上你的每一次回答问题都很具体、到位，我能感受到你的诚意与认真回答的态度。
之前有几次我碰到了核心课上的问题想要跟你讨论，我也跟班班询问过你的联系方式，可是被班班拒绝了，可能这是你们的公司规定吧，工作号不可以加学生。
其实一直以来都很想加一下你的微信，不管是在作业上的讨论也好，还是在课程上的讨论也好，或者是未来竞赛上的请教也好。

所以，之后能否加一下个人微信，或者个人QQ呢？希望有机会还是能继续跟你讨论的。

我的联系方式是：
QQ：583747834
weixin：zby0409