**正负样本构造trick**

**一、数据是模型的上限**

**二、Feed流场景下构造样本**

在Feed流场景下，使用曝光(展示)日志时，应该选择APP的SDK埋点的日志，而不是服务器Web接口返回的日志，因为Web接口返回的日志中的接种是会增加很多无效负样本；举个栗子，Web接口每次返回10条数据，但是APP屏幕最多展示3条数据，剩下的7条数据需要用户在Feed中滑动窗口后，才能算真正的曝光，但是很多用户可能并不会滑动窗口或者滑动幅度比较下，导致剩下的7条数据并未真正的在APP上曝光。针对上述问题，有如下几个解决方案：

针对同一个内容在不同时间对同一个用户曝光多次的情况，这时候训练集中可能出现同一个用户对同一个内容点击与不点击并存的情况，如果多次曝光的间隙非常短，考虑只使用其中的一次曝光数据。

根据YouTube的早期论文中描述，他们为了避免高度活跃用户对loss的影响，在训练集中对每个用户提取相同数量的训练样本。

根据《美团机器学习实战》书中所说，他们在Feed流场景中采用了Skip-Above的方式来提高效果。具体的做法就是，根据用户最后一次点击行为的位置，过滤掉最后一次点击之后的展示，可以人认为用户没有看到。当然可以选择，保留最后一次点击后的少数几个。

保留有行为的用户的负样本，其他负样本（没有过对应的正样本），进行随机负采样，实践证明这样做效果要优于只保留有行为的用户的负样本。

与上一个方式对立的，可以考虑去除只有曝光但没有点击操作的用户的样本（也就是说有的用户只有负样本，没有正样本），不过去除的话，那模型就只能够学习到活跃用户或者有意向用户的行为习惯，这就导致线下数据与线上数据分布有偏差，所以需要根据线上的AB-test测试效果来决定最终是否去除。

**三、训练样本的构造要点**

训练样本的构造方法，尽可能的采用线上模型预测样本时刻的特征作为之后训练样本中的数据，而不是使用离线回溯的方式去拼接训练样本中的各个特征，这就要求当线上模型在预测时，需要将喂给模型的特征做一次落地，比如传到kafka，后续再由相应程序进行解析生成之后的的训练样本。

**3.1平衡正负样本问题**

机器学习的训练数据不是看它分布是否平衡，而是看它是否符合原来的分布。如果符合原来的分布，那么训练误差最小化也就意味着整体分布误差的最小化，也就没必要进行平衡。

**3.2什么好的训练集？不平衡的训练集是否一定需要平衡？**

如果正负样本不是极端的不平衡，不太会影响模型学习效果。相反修改全量数据的原始分布，改的不好的话，有可能会降低模型的性能。

样本均衡的处理，一般很多机器学习案例数据都会叫做“专家样本”，会融入很多的业务洞察，专家样本做得好，几百个样本的模型，能击败几万甚至几十万样本的模型。

所以什么是好的训练集，没办法给一个准确的定义。那坏数据集相对来说就比较容易归纳了。

**3.3 坏数据集**

**3.3.1抽样不当的数据**

抽样数量太少，数据量太小不能代表全体样本。或者说正样本太少，就无法体现所有正样本的特性。但是具体的抽样数据量该多少没有一个具体的标准。有的文献说几十个样本就足够了，但是有的文献说需要几万个样本，所以没有统一的说法。但是，统一明确的一点是，抽样的数量一定要尽量的去代表全体数据。

抽样偏差。这个就比较容易理解了，抽样的数量要保证，同时质量也要保证。举个栗子，我们要通过一些穿着特征去或外貌特征判断一个人的性别，这时候如果去艺术学校去抽样，这时候艺术系的男生也会留长发、化妆等，这时候就会与社会整体的男性产生严重的偏差，没有办法表征全部样本。

**3.3.2不满足模型需求的的数据**

有些算法对数据的平衡性非常的敏感，以及对特征分布或者特征分段要求很高。对于这样的模型，就需要注意正负样本比例，否则性能可能显著下降。

**3.3.3不满足业务需求的数据：**

在特定的业务场景下，正样本本来就很少，此时训练出来的参数表达能力有限，模型是否会不符合业务的召回率需求。

举个栗子，在信用卡欺诈模型中，对于召回率的要求比较高（不希望漏掉一个欺诈用户），并且预测出来的数据还会经过人工审核，相对的对于准确率要求就低一些。但是在我们的原始数据中，正样本的比例本身就占比非常小了，或者正样本本身就是正太分布部分，但是在预测的时候，连长尾分布的部分也不能放过，（尽量的把所有欺诈用户召回），比如信用卡欺诈里有的超级用户虽然数量小，但是一次违约就是几十万，比几百个普通用户还严重，这种时候是否要用权值设置或者复制正样本的方式，来做识别增强。

像上面的栗子中，就很有必要通过提高正样本的数量，来覆盖所有正样本的情况，并降低负样本的数量，使得正样本特征得到更强的表达。

**3.4解决样本不均衡问题的具体方法**

**3.4.1 采样处理**

采样是解决样本不均衡问题的相对简单且常用的方法，包括过采样和欠采样两种

【过采样】

过采样（也叫上采样，over-sampling）方法通过增加分类中少数类样本的数量来实现样本数量的均衡，最直接的方法是简单赋值少数类样本形成多条记录。

这种方法的缺点是如果样本特征少，很可能导致过拟合问题。

改进的过采样方法，通过在少数类中加入随机噪声、干扰数据或通过一定规则产生新的合成样本。

产生新的合成样本的算法有SMOTE算法。

【欠采样】

欠采样（也叫下采样，under-sampling）方法通过减少分类中多数类样本的样本数量来实现样本均衡，最直接的方法是随机的丢掉一些多数类样本来减少多数类样本的规模。

缺点是会丢失多数类样本中的一些重要信息。

总体而言，过采样和欠采样更适合大数据分布不均衡的情况，尤其是过采样方法应用更加广泛。

**3.4.2 类别权重**

通过正负样本的惩罚权重解决样本不均衡的问题的思想是在算法实现过程中，对于分类中不同样本数量的类别分别赋予不同的权重（一般思路分类中的小样本量的类别权重高，大样本量的类别权重低），然后进行计算和建模。

使用这中方法时不需要对样本本身做额外处理， 只需要在算法模型的参数中进行相应的设置即可。（调整超参数）

很多算法中都有基于类别参数的调整设置，以sklearn中的SVM为例，通过在class\_weight:{dict,'balanced'}中针对不同类别针对不同权重，来手动的指定不同类别的权重。如果使用其默认的方法balanced，那么SVM会将权重设置为与不同类别样本数量呈反比的权重来做自动均衡处理，计算公式为：n\_samples/(n\_classes \* np.bincount(y))

如果在算法本身实现的时候有这样的参数可以调整，这样的方法更加简单且有效。

**3.4.3 集成方法**

组合/集成方法指的是在每次生成训练集是使用所有分类中的小样本量，同时从分类中的大样本量中随机抽取数据来与小样本量合并构成训练集，这样反复多次会得到很多训练集，从而训练处多个模型。最后在应用时，使用组合/集成的方法（比如，投票，加权投票等）产生分类预测结果。

例如，在数据集中的正、负样本分别为100和10000条，比例为1：100，此时可以将负样本随机切分为100份，每份100条数据，然后每次形成训练集时使用所有的正样本（100条）和随机抽取的负样本（100条）形成新的训练数据集。如此反复可以得到100个模型。然后继续集成表决。

这种解决问题的思路类似于随机森林，在随机森林中，虽然每个小决策树的分类能力很弱，但是通过大量的“小决策树”组合形成“森林”就具有良好的模型预测能力。

如果计算资源充足，并且对于模型的时效性要求不高的话，这是一个不错的选择。

**3.4.4 特征选择方法**

上述的集中方法都是基于数据行的操作，通过不同的方法来使得不同类别的样本数据条数均衡，除此以外，还可以考虑使用特征选择的方法选择特征。

一般情况下，样本不均衡也会导致特征分布不均衡（表征正样本的特征肯定和表征负样本的特征不同，否则就无法区分了），但是如果小类别样本量具有一定的规模，那么意味着其特征值的分布较为均匀，可以通过选择具有显著型的特征配合参与解决样本不均衡问题，也能在一定程度上提高模型效果。

怎么理解上面的意思呢，非常简单，如果某些特征分布极其不均匀，说明这个会造成分类的不稳定，那么我们就需要把这样的特征剔除，选择哪种具有很好的表征正样本（小类样本）的特征。以便提高模型的性能。

**四、搜索排序的样本构造不足之处**

1/3的搜索会话没有点击行为，这些样本里面的数据不应该被直接扔掉。数据如下所示：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | session\_num | total\_session | session\_ratio | imp\_count | total\_imp | imp\_ratio |
| click | 185905 | 301531 | 61.65% | 1883976 | 2594070 | 72.62% |
| noclick | 115626 | 301531 | 38.35% | 710094 | 2594070 | 27.38% |

在有会话的搜索记录下，我们对最后一次点击行为之前的样本进行了skip\_above的选择，但是后面的样本没有进行随机采样。

举个栗子：

|  |  |  |
| --- | --- | --- |
|  | Session1 | Session2 |
| doc\_1 | 0 | 0 |
| doc\_2 | 0 | 0 |
| … | 0 | 0 |
| doc\_15 | 1 | 0 |
| doc\_16 | 0 | 0 |
| doc\_17 | 0 | 0 |
| … | 0 | 0 |

因此我们应该对Session1进行随机的进行选择。

1：若是session 下点击数目非空，则点击的最后一条到session 下曝光的最后一条应该随机进行选择。

2：若是session 下点击数目为空，则全部结果随机进行选取。

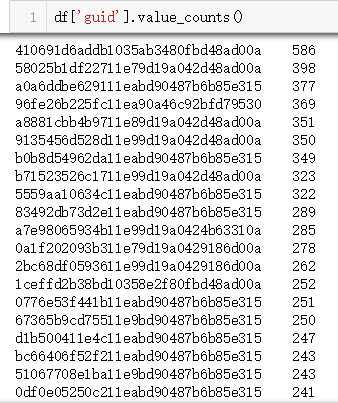
另外头部对整体的影响太显著了，需要对不同的用户平衡样本

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| top | thres | idf\_count | cdf\_count | total\_count | idf | cdf |
| 1% | 90 | 281802 | 281802 | 2594070 | 0.108633 | 0.108633 |
| 2% | 70 | 242374 | 524176 | 2594070 | 0.093434 | 0.202067 |
| 3% | 58 | 169762 | 693938 | 2594070 | 0.065442 | 0.267509 |
| 4% | 48 | 156126 | 850064 | 2594070 | 0.060186 | 0.327695 |
| 5% | 40 | 137683 | 987747 | 2594070 | 0.053076 | 0.380771 |
| 10% | 22 | 419267 | 1407014 | 2594070 | 0.161625 | 0.542396 |
| 15% | 15 | 242338 | 1649352 | 2594070 | 0.09342 | 0.635816 |
| 20% | 11 | 158791 | 1808143 | 2594070 | 0.061213 | 0.697029 |
| 25% | 9 | 142839 | 1950982 | 2594070 | 0.055064 | 0.752093 |
| 30% | 7 | 113597 | 2064579 | 2594070 | 0.043791 | 0.795884 |
| 34% | 6 | 59070 | 2123649 | 2594070 | 0.022771 | 0.818655 |
| 43% | 5 | 131515 | 2255164 | 2594070 | 0.050698 | 0.869354 |
| 52% | 4 | 100380 | 2355544 | 2594070 | 0.038696 | 0.90805 |
| 62% | 3 | 81291 | 2436835 | 2594070 | 0.031337 | 0.939387 |
| 76% | 2 | 95940 | 2532775 | 2594070 | 0.036984 | 0.976371 |
| 100% | 1 | 61295 | 2594070 | 2594070 | 0.023629 | 1 |

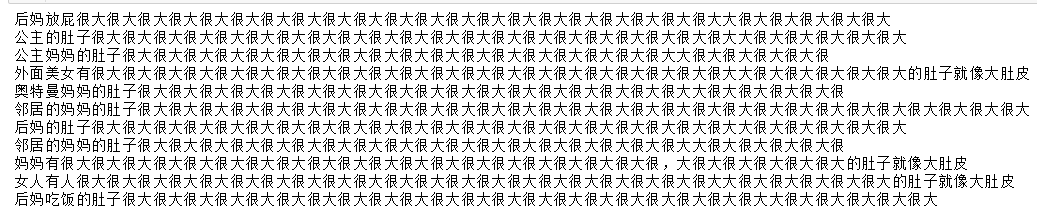
**五、搜索排序的样本构造原因及指导思想**

**5.1 平衡guid top的影响**

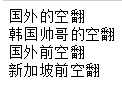
鉴于某个guid (用户)一个小时内搜索的query 大多数是很相关的或者错误的。因此限定对每个guid 只选择一个其中的一个qid来构造训练样本



410691d6addb1035ab3480fbd48ad00a



9135456d528d11e99d19a042d48ad00a



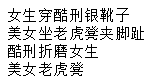
b7ed371d311511e99d19a0424b63310a



a7e98065934b11e99d19a0424b63310a



38cc320de2f011e79d19a0420722850a



**5.2 有点击的session 正负样本构造思想**

最后一个点击之前的没点击的样本就是负样本（skip\_above）,后面的样本按照一个比例进行选择。

**5.3无点击的session 正负样本构造思想**

无点击的若是选择pointwise, 那按照一定比例随机产生负样本即可。

若是选择训练模式为pairwise。因为不知道到底是因为排序的顺序用户不满意，还是因为呈现的内容用户不满意。