区别于一般的二分类任务，对于用户流失预测这样的场景，会涉及到很多和时间相关的事情，为了更好的理解后面的内容，我们先了解一下下面几个和时间有关的概念：

**流失周期**：定义的是某用户没有发生某种关键行为的时间间隔

**流失预测周期**：定义的是每次离线做成批用户流失预测的时间间隔

**用户最近一次登录(last-login)的时间**：这个时间离当前时间的时间跨度可以建模为特征

**统计用户某行为的时间间隔**：这个时间段内的用户的某种行为的统计值会建模为一个特征

**增量训练的时间间隔**：为了让模型能持续的学习知识，可能需要每隔一段时间增量学习

接下来进入正题，关于用户流失/留存预测，用机器学习来建模的话，需要重点考虑下面几个问题（[**https://www.zhihu.com/question/68029067**](https://www.zhihu.com/question/68029067) **这个link中讲解的很清楚**，可以参考，强烈推荐阅读一下）：

**如何定义一个用户是流失的？**

**模型训练好以后，如何使用该模型来预测用户的流失**？

**数据集的时间窗口选择问题**？

**训练集和验证集如何切分**？

**用户流失预测模型的训练迭代问题**？

**关于用户流失预测场景下，强特征的挖掘**？

**在得到了针对用户的流失预测概率以后，如何利用这个概率**？

下面针对每个问题，我们来详细解释：

1. **如何定义一个用户是流失的？**

不合理的流失周期的划分会造成预测准确率低下（**因为一定存在回流的用户**），我们需要不断尝试周期划分，才能更为准确地预测流失。在这种情况下，选择恰当的定义方法显得至关重要。

对流失比较经典的定义是“**一段时间内未进行关键行为的用户”**，**关键点在于如何界定时间周期（流失周期）和关键行为（流失行为）**。不同的业务场景，这个关键行为一般会有区别。

**对于游戏用户的流失，可以选择“主动登录”这一行为作为是否流失的关键行为**。

注意：**这里可能需要考虑如何判定用户为主动登录，也就是可能这里需要考虑登录后的停留时间的问题**。

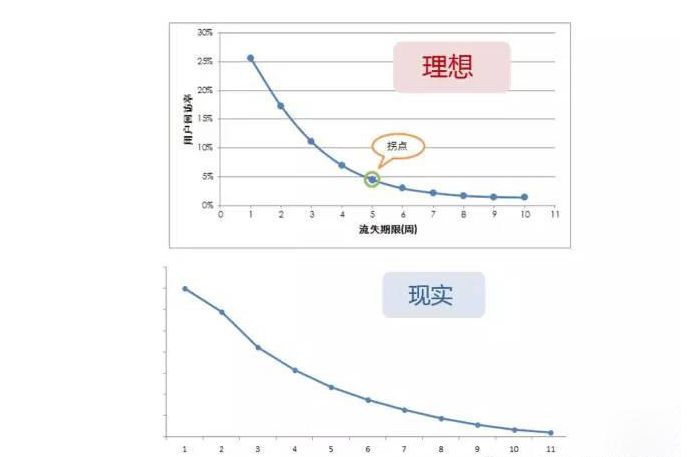
如果登录后马上就退出是否应该不算做有效登录；或者登录以后马上把这个应用切换到了后台来挂机，这种也不应该算做有效登录。也就是说，**应该把用户有登录动作，并且把应用放在前台，且停留了足够长时间（比如10分钟）才认为是一次“有效登录**”。

**对于流失周期的界定，常见有两种方法**：

1. **分位数界定法**

**可以使用90%分位数的方式来选择流失周期**。也就是根据用户连续两次登录的时间间隔从小到大来排序，90%分位数对应的时间间隔就作为流失周期。

1. **拐点界定法**

可以参考下图:  


注意上图中理想情况中的那个图，每个流失周期对应的用户回访率的计算需要计算流失用户数和回访用户数（用户回访率 = 回访用户数 ÷ 流失用户数 × 100%），具体计算过程参考如下：

流失用户数计算比较直接，比如流失周期是1周，那么下周没有进行登录的用户就认为是流失；但是**这里没有明确说明如果计算回访用户数**，**我理解应该会首先定义一个比较长的周期**，对于这里的例子来说，因为横坐标中有数据点的最大流失周期为10周，这里就把这个用户**回流周期**也设定为10周，对于流失周期是1周并且被判定为流失的用户如果在后续的10周内只要有有效登录就认为是回流用户；同理，对于流失周期是10周并且判定为流失的用户在后续的10周内只要有有效登录就认为是回流用户。（注意：**我这里是假设对不同的流失周期使用了一致的回流周期，这样设置是否合理以及回流周期具体多长，最好与业务运营人员确认**。）

**如果曲线没有拐点的话，也可以依据产品经验来拍脑袋**，一般产品的回访率5%-10%（通过固定了回访率然后在曲线找到对应的流失周期），不管划分多长的时间周期作为流失周期都会存在回访，误差不可避免。

1. **模型训练好以后，如何使用该模型来预测用户的流失**（**和其他业务对比还比较特别**）？

**由于用户流失预测这个业务的特殊性，模型训练完以后一般不会马上用来做预测**（因为已有的老用户的流失和留存信息刚做了标注并被模型学习和验证来使用），所以一般会等一些时间比如几天或者几周（所谓的“**流失预测周期**”，比如可以把流失预测周期设置为1个月，正如下图中看到的），然后对下面两类人群进行流失预测（**一定要注意每次进行流失预测的人群，不要漏掉**！）：

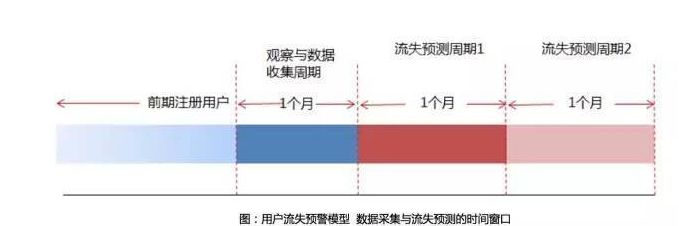
1. 对于新的从新用户转化为老用户的这群人来预测是否会流失；
2. 对于之前周期属于留存用户的这群人来进行预测是否会流失。
3. **数据集的时间窗口选择问题**？

在流失周期确定以后，我们需要来选择数据集的时间窗口。这里的时间窗口又包括两个（**这里的时间窗口并不是那么直接且显而易见的**）：

1. **从哪天开始收集用户的登录日志**？
2. **所有用户还是老用户**？

**一般流失预测更多针对的是老用户，而新的用户一般需要时间成长为老用户**。而这个新老用户的判定条件也是根据业务来具体设定的，比如累计玩游戏时间超过1个月的新用户就可以认为是老用户了。

1. **需要纳入数据集来训练的最早的时间如何选择**？



如上图所示，这里设定的流失周期是1个月，那么在准备开始训练的时间点往前推一个月这个时间窗口就是用来观察所有的“老用户”在这个时间窗口内是否有有效登录行为，如果有的话，表示该用户是留存用户，如果没有的话说明该用户有超过1个月时间没有有效登录行为了，标记为流失用户。

那对于上图提到的前期注册用户也就是我这里说的“老用户”，这个时间窗口往前推多少合适呢，也就是老用户的last-login最近一次登录时间在多久以前？我觉得主要有两个原则：

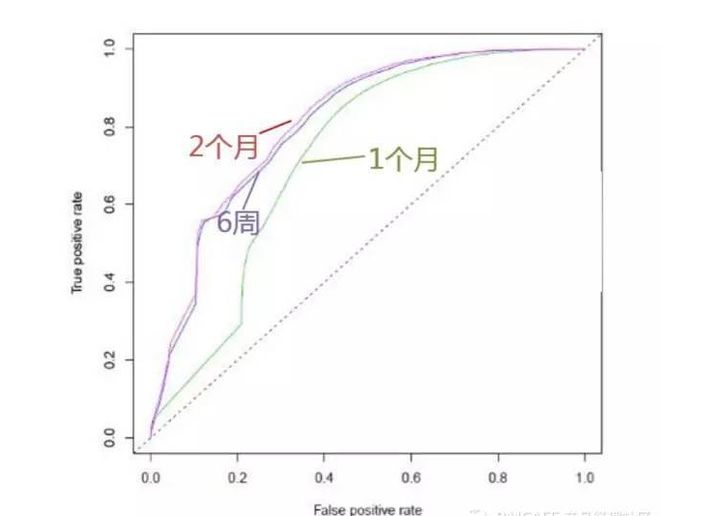
**数据样本要足够**（尤其是正样本也就是流失用户的样本要足够）；

**前推的时间窗口不要太长**（前推时间窗口太长的话，会把一些都已经流失很久的老用户的样本也纳入模型来学习，可能因为时效性等因素这些已经流失很久的老用户并不是我们这个流失预测模型所关心的人群；当然从对流失用户做归因分析的角度来说，这样的数据同样有意义，但那是另一个话题。）

1. **需要统计用户多长时间内的行为**？

这个也没有什么定论，这个时间窗口是用户last-login最近一次登录时间戳往前推的一段时间。

**一个方法就像这个link中提到的，通过对多个统计周期的特征来分别建模，然后在验证集上看ROC面积哪个更大**，如下图所示。



如上图所示，通过二元逻辑回归的ROC曲线可以进行评估。正如图所示6周的数据明显优于1个月（曲线右下方面积越大预测准确性越高），而2个月的数据只略优于6周，幅度有限，且时间成本较大，因此选择6周作为数据提取的周期。

注意：由于之后用于流失预测的时候，会包含对那些从新用户转为老用户的人群进行预测，因此不适合把这个统计时间窗口弄的很长（**也就是说为了照顾这批人，需要把这个统计时间弄的不能太长，当然也不能太短**）。

这里举个例子比较容易理解，比如流失周期是1个月，当前是在3月1日准备进行训练，那么整个2月份的时间窗口是需要预留出来的，数据集中的某个特征比如统计用户离他最近一次登录（比如是1月1号）往前推1个月的累积消费金额，那么这个统计时间窗口对于这个用户就是去年的第12个月（**每个用户的统计时间窗口都不一样，因为是按照每个用户的last-login往前推固定时间段来的**）。

1. **训练集和验证集如何切分**？

**对于用户流失预测这样的场景，可能不方便严格按照时间线来切分训练集和验证集；而且，用户流失预测这样的场景并不是一个时间依赖相关的**。**因此对这样的数据集可以按照比例来切分**（**具体考虑到流失/留存类别样本的不均衡，这里做训练集和验证集切分的时候最好做分层按比例切分**）。

1. **用户流失预测模型的训练迭代问题**？

针对这个场景来说，模型的训练迭代可以选择的方式是：

**每隔一段时间做全量数据重新训练**；

**增量训练方式**

（当然还有其他的混合方式：全量数据训练，但用上一个训练周期的模型参数做初始化；小时间间隔增量训练+更大间隔的全量数据训练等）

对于用户流失预测模型的训练迭代方式，由于很可能会发生当前流失预测周期还是留存的用户下个流失预测周期就变成了流失用户（当然也可能反过来），这样用增量训练的方式的话，每个训练周期的用户的知识可以学习到（假设模型的容量足够大）；每隔一段时间做全量数据重新训练的话，可以通过把用户的旧的知识作为特征建模进来，即使这样也可能对用户旧的知识学习的没有增量训练那么好，因此我觉得可能用增量训练方式对这个用户流失预测场景更合适（**当然更合理的对比是做AB test**）。

**如果选择增量训练的方式，那么增量训练的频率选择多少**？

**增量训练的频率对于用户流失场景可能不太适合按天来增量做，因为需要等待一定的天数来积累新的样本**。

**增量训练频率或者我们叫训练周期和流失预测周期没有什么必然联系**。

比如预测流失周期是1个月，那么训练周期可以设置为3周或者5周（设置3周是想尽快在下次流失预测周期时就能利用到新的模型版本学习到的知识；设置5周是为了给更多的机会来积累新的样本）

1. **关于用户流失预测场景下，强特征的挖掘**？

**可以把一些能通过业务经验识别到的一些流失影响因素，或者作为特征建模到模型中，或者把这些因素影响到的样本从样本集中剔除**。

对于像产品功能更新了，或者年底积分折半了这样的小概率事件发生时，需要尽可能分析是否这些因素导致了用户的流失，但是**这样的小概率事件是否发生一般不作为特征放入模型，因为这个特征在每次训练的时候基本上都是一个值**（比如对于产品是否升级这个特征，可能在大多数训练的时候，这个特征是0，对目标任务没有什么辨识度，也就是和目标任务不相关）。

对于像比如游戏用户投诉次数这样的特征对于用户流失是一个强相关的特征，这样的特征需要多挖掘一些并放入模型中建模。

1. **在得到了针对用户的流失预测概率以后，如何利用这个概率**？

**如果想简单的判定该用户是否是流失用户，那么需要一个阈值**，这个阈值一般会基于cost成本/代价在验证集上通过对混淆矩阵来遍历阈值找到一个最优的。

具体可以参考AWS sagemaker的一个demo code：<https://github.com/awslabs/amazon-sagemaker-examples/blob/master/sagemaker_neo_compilation_jobs/xgboost_customer_churn/xgboost_customer_churn_neo.ipynb>

利用上面的方法来计算阈值的时候，需要针对不同的metric/指标做一个代价/成本的预估，**但是有些时候对于某些场景/业务不容易预估这个成本，这个时候可以考虑基于策略的方式来做**。

比如将使用App的频率和时长作为用户轻重度的划分标准，结合用户流失预测概率（参考下图，**其实这里仍然需要象限划分的，如何来划分留存和流失的界限文中并没有提及，我理解可能是根据大盘历史上对留存和流失用户数的统计然后基于分位数来做的**：比如大盘一年统计的流失用户的比例为20%，那么对离线预测的这批用户的流失预测概率进行从小到大的排序（不要去重），然后选取80%分位数的概率作为阈值，超过这个阈值的可以认为是流失的潜在用户），将用户划分为高价值、重点发展、重点转化、有待挽留等几种类型，分析每个类型用户不同的行为特点和使用痛点，采取针对性的运营策略。

