Sparkify – 在线音乐服务用户流失分析

By Liya 2020 年 7 月 31 日

预测客户流失率是数据科学家和分析师在面向消费者的一类公司中经常遇到的一项具有挑战性的问题。作为 UDACITY 数据挖掘直通车纳米学位的结业项目中，我们尝试使用在线音乐服务 Sparkify 的日志数据来进行分析。

所谓流失用户，我们通常定义为从收费用户降级至免费用户，以及直接取消服务的用户，这两部分用户也很可能有重合。本项目中，我们只对取消服务的用户作分析，即若用户日志的`page`字段中包含`Cancellation Confirmation`事件，我们就标记他们为流失用户（churn）。

我们的目标是从日志数据中抽取特征以建立一个模型，并拿它来预测测试集中的用户标签（churn），并和实际的标签对比。如果预测的结果至少优于基准值（Baseline），我们就可以认为这个模型是有用的。而若能得出有效预测，Sparkify 平台就能够赶在用户取消服务之前给他们提供折扣或激励，以此降低流失率。

本项目使用的完整日志数据集大小为 12GB，考虑到计算资源，我们先使用它的一个 128MB 迷你子集来进行分析。以下是主要步骤：

1. 加载和清洗数据
2. 探索性数据分析
3. 建立特征和标签
4. 训练和评估模型
5. 选择最优模型并调参
6. 解读

加载和清洗数据

因为数据量巨大，我们需要用到专为大规模数据处理而设计的快速通用数据引擎 Apache Spark。这里我们首先创建一个 Spark 进程，然后读入. json 格式的日志数据集，这其中包含了 225 个不同用户产生的 278154 行数据。

因为日志数据本身比较规整，需要清理的部分不多。右边是数据字段一览：

其中 page 字段为用户在使用音乐服务时进行的各种操作，列表如下：

```
root
 |-- artist: string (nullable = true)
 |-- auth: string (nullable = true)
 |-- firstName: string (nullable = true)
 |-- gender: string (nullable = true)
 |-- itemInSession: long (nullable = true)
 |-- lastName: string (nullable = true)
 |-- length: double (nullable = true)
 |-- level: string (nullable = true)
 |-- location: string (nullable = true)
 |-- method: string (nullable = true)
 |-- page: string (nullable = true)
 |-- registration: long (nullable = true)
 |-- sessionId: long (nullable = true)
 |-- song: string (nullable = true)
 |-- status: long (nullable = true)
 |-- ts: long (nullable = true)
 |-- userAgent: string (nullable = true)
 |-- userId: string (nullable = true)
```

page	count
Cancel	52
Submit Downgrade	63
Thumbs Down	2546
Home	10082
Downgrade	2055
Roll Advert	3933
Logout	3226
Save Settings	310
Cancellation Conf...	52
About	495
Settings	1514
Add to Playlist	6526
Add Friend	4277
NextSong	228108
Thumbs Up	12551
Help	1454
Upgrade	499
Error	252
Submit Upgrade	159

我们发现日志中有一些 userId 为空白值的项目，经过观察，这些都是和用户登录相关的记录，对我们接下来的分析没有太多参考意义，故可以删除。

auth	page
Guest	Register
Guest	Error
Guest	Submit Registration
Guest	About
Guest	Help
Guest	Home
Logged Out	Home
Logged Out	About
Logged Out	Error
Logged Out	Login
Logged Out	Help

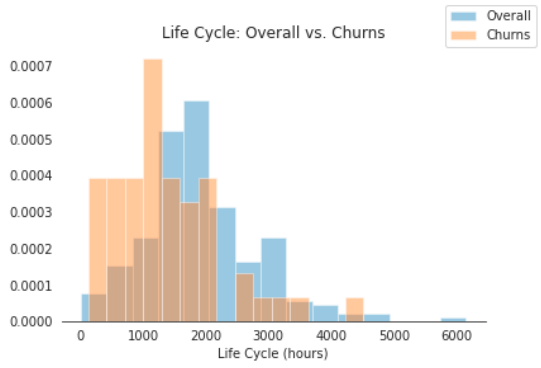
探索性数据分析

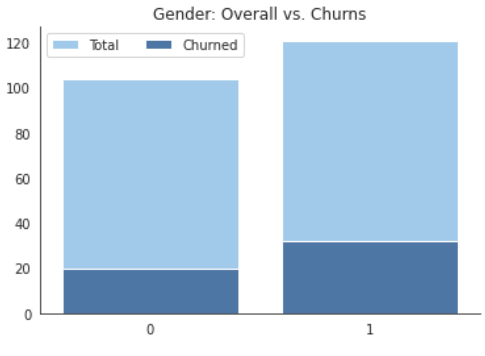
首先我们需要对流失用户进行定义：这里我们把最终取消了服务（即 page 中的 Cancellation Confirmation 事件）的用户定义为流失用户，因此我们新建字段 churn，将流失用户设置为 1，未设置用户设置为 0。

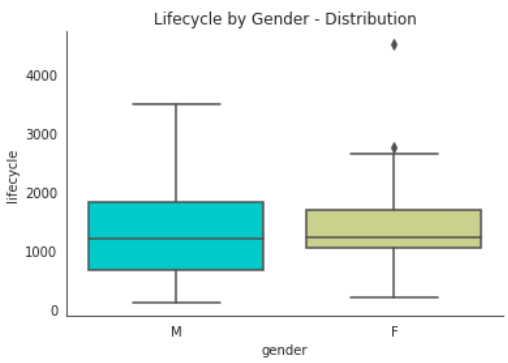
churn 也将作为后续模型训练时时的 label（标签）。

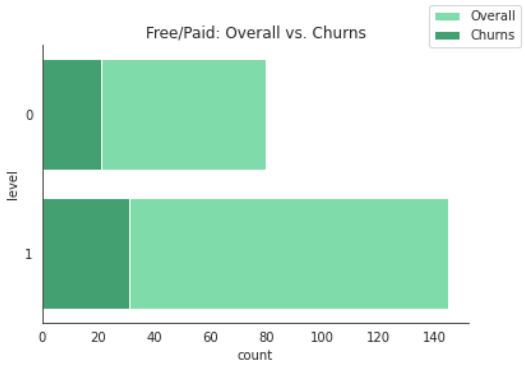
接下来我们应用 spark.sql 及可视化方法对可能的影响用户流失的因素逐一分析，以期找出流失和非流失用户表现不同的因素。我们将抽取这些字段来作为模型训练用的 feature（候选特征）。

我们首先能想到的是用户的注册时长（生命周期）、性别和账户等级（付费/免费）。注册时长可由数据集中用户最后一次操作的日期与注册日期之差得到。对这 3 个字段，我们分别比较其流失用户与全体用户的差异，结果如下图：



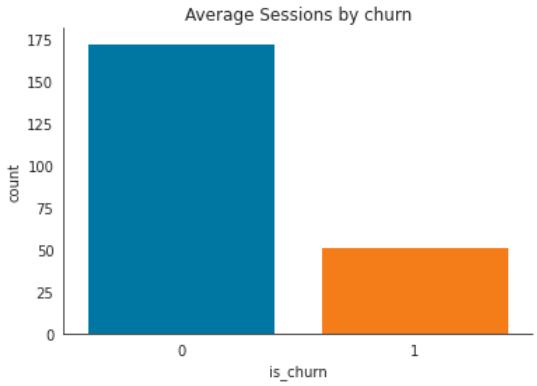


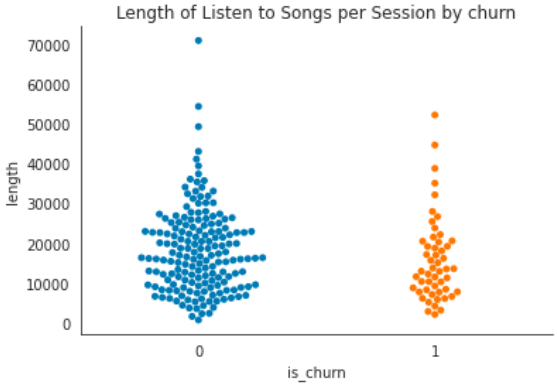


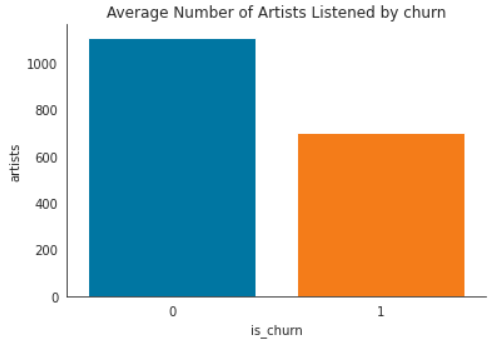


我们能观察到，流失和非流失用户生命周期的分布和峰值都存在差异，男性用户流失比例更大，付费用户更倾向于取消服务。这些都可以作为候选特征。

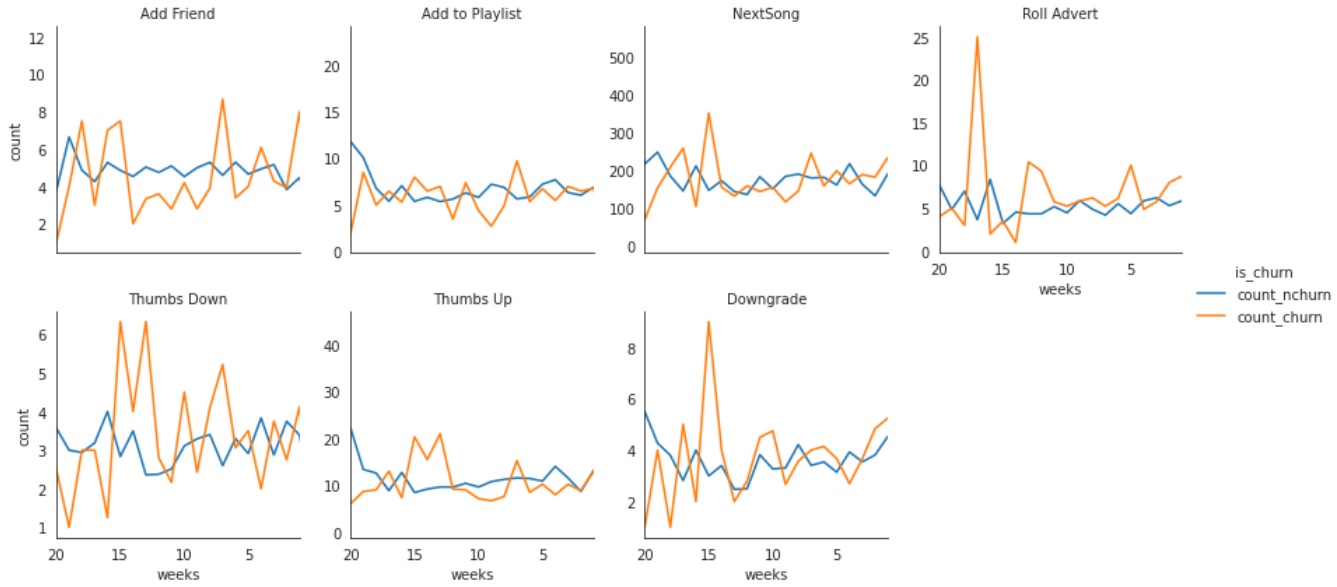
接下来我们有理由假设非流失用户更频繁地使用服务（更多进程数），每次使用服务时播放过更多的歌曲，也尝试过更多不同歌手，结果和我们的猜测是符合的，所以它们也都可以当作候选特征:







最后我们对用户的各种操作（page）作折线图，以周为单位统计各个操作标签的数量变化，看这些项目中是否能观察到流失和非流失用户间的分离情况。目视观察结果是 Add Friend、Roll Advert、Thumbs Down、Thumbs Up、Downgrade 几个操作似乎分离更多一些，但并不是十分显著。我们把它们都加入候选特征。



建立特征和标签

通过探索性数据分析，我们总共选择了 13 个字段作为 Feature，把它们和作为 Label 的 churn 一起合并到 14 x 255 的数据表中，并进行向量化、规一化等预处理，建立模型的基础就准备好了。

训练和评估模型

我们先把日志数据集划分为训练集、测试集和验证集库 （即 Train, Test and Validation），比例为 8:1:1。其中训练集和验证集用于模型训练，测试集只用于评估最终的预测结果。

这里我们从 spark.ml 中导入 5 个机器学习模型：Logistic Regression, Gradient Boosted Trees, Support Vector Machines, Random Forest，对它们分别使用默认参数进行训练，这是本项目中最消耗时间和计算资源的一步。

这里选用 F1 score 作为主要优化指标， 因为我们希望在准确率和召回率间取得平衡，即希望尽可能不漏掉可能流失的用户，又不希望把优惠额度浪费在太多本没有流失风险的用户上。而 F1 Score 在平衡二者上有较好的参考意义。

另外训练时间也是一个重要的参考指标，尤其是我们要处理的数据量非常大，以加倍的训练时间（也意味着更多的计算资源）来换取 F1 Score 的少量提升并不划算。

同时我们需要确保模型预测能力至少要优于基线值 Baseline（这里我们定义基线值为：对所有用户都判定为非流失用户，亦即所有用户的 Label 都为 0）

以下是各模型训练结果和基线的比较：

	accuracy	f1	runtime
Random Forest	0.857143	0.839827	36.0904
Logistic Regression	0.857143	0.839827	233.662
Gradient-Boosted Trees	0.714286	0.72619	358.514
Baseline	0.777778	0.680556	-
Naive Bayes	0.714286	0.595238	3.89333
Support Vector Machines	0.714286	0.595238	686.562

选择最优模型并调参

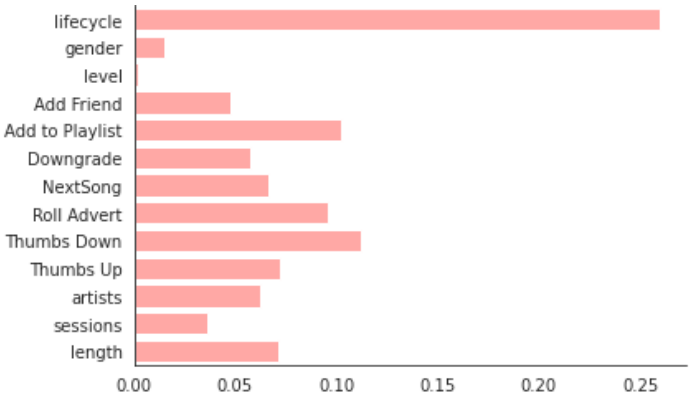
观察以上评估表，我们可以看到 Random Forest 和 Logistic Regression 评分相当，F1-Score 均为 0.84，且优于基线值，而 Random Forest 运行时间明显要短很多。所以我们选定 Random Forest 为最优模型。

为了得到更好的预测结果，我们对 Random Forest 模型设置多个预选参数，并使用 paramGrid 和 CrossValidator 进行批量调参。这次我们用测试集数据来评估最终预测结果，最后得出 F1-Score 为 0.790。

与我以往做过的项目相比，这一步花费在模型训练上的时间特别长。尤其在调参时，如果一开始时选择的参数不佳，后续反复尝试更是花费成倍的时间，而这还权权只是针对 128M 子集的训练。这也让我更直观地认识到数据质量、算法和模型选择对一个项目花费时间和资源的影响。

解读

我们使用 featureImportances 列出各特征的影响力占比：



可以看出，对用户流失影响最大的因素为 lifecycle，亦即注册时长，而我们一般猜测的因素如播放数量和时间、是否付费，以及社交频度等，虽然也是影响因素，相较之下并没有体现出想象中的影响力，尤其是性别和账户等级，影响力非常小。多数用户在 2000 小时左右就取消了服务。

若按照这一结论，我们应该在用户注册满 2000 小时左右就给他们提供优惠券。

局限及改进空间

我们最终得到的 F1-Score 0.790 并不是一个很好的结果，只比 Baseline 高出 16%。可能的原因是，参与分析的子集中一共只包含 225 个用户，这对训练来说可能是不够的。如果将最终模型运用到完整数据集中，应该能得到好的结果。

Downgrade 应该是一个很重要的因素，但在此次特征分析中占比不算高，应该也是跟用户数太少有关系。而在完整数据集中针对所有 Downgrade 用户单独作分析，可能能够得到更有意义的结论。

此外，未取消服务的用户可能包括两种情况：重度用户和注册之后很少使用的用户，不区别这二者的话，可能会对我们的分析结果造成偏差。

另外我们还可以针对用户取消服务之前几天、几周内的行为作更精细化的分析，也许能够找出某些行为模式来。

分析过程用到的主要依赖库

PySpark.sql, PySpark.ml, Pandas, Matplotlot.pylpot, Seaborn

Github 代码库

https://github.com/liyapink/Sparkify/