Spark 项目 2020年3月11日
数据挖掘直通车班

预测Spotify平台用户流失情况

项目概述

Spotify平台采用订阅模式，盈利主要来源于广告以及订阅费用。为了提高平台利润，需要尽可能多地吸引用户并且尽力避免用户流失。为了尽可能挽留即将流失的用户，需要给予一定的优惠服务。平台收集用户的使用数据，根据用户的注册信息以及过往的行为历史，使用机器学习方法，预测用户流失的概率。

问题陈述

项目目标是预测用户流失的概率，需要完成以下步骤：

1 加载和清洗Spotify平台数据
2 探索性数据分析，充分了解数据集
3 特征工程，筛选/建立与预测用户流失情况相关的重要特征
4 使用特征工程建立的特征，搭建模型预测用户未来流失的情况
5 根据评估指标进行模型优化

该分析建模过程暂时使用完整数据集的子集进行

评估指标

预测用户流失情况属于监督学习中的分类问题，由于流失用户人数与留存用户人数之比大约为1:3，分布不平衡，所以使用F1 Score，F1 Score兼顾了precision与recall，是precision与recall的harmonic mean。
precision = the number of correct positive results / the number of all positive results
recall = the number of correct positive results / the number of samples that should be identified as positive

Compute the F1 score, also known as balanced F-score or F-measure
The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the F1 score are equal. The formula for the F1 score is:
F1 = 2 * (precision * recall) / (precision + recall)
In the multi-class and multi-label case, this is the average of the F1 score of each class with weighting depending on the average parameter.

——From Scikit-Learn Document

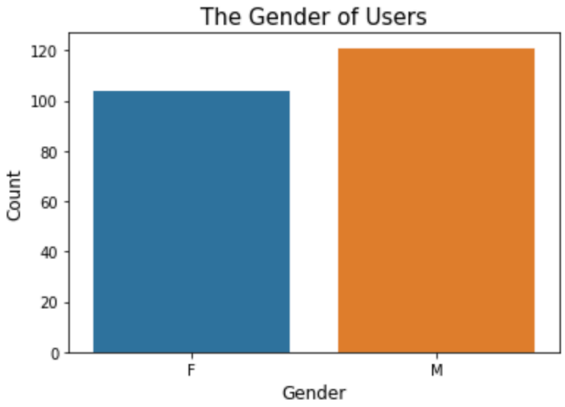
加载和清洗Spotify平台数据

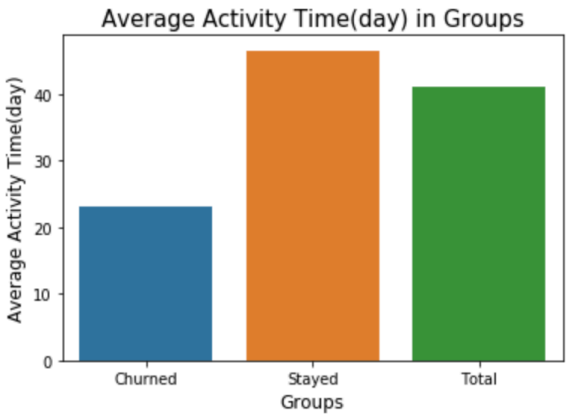
排除userId为空的数据，因为这些数据是在用户未登录的状态下产生的，与本项目预测用户流失情况的需求无关

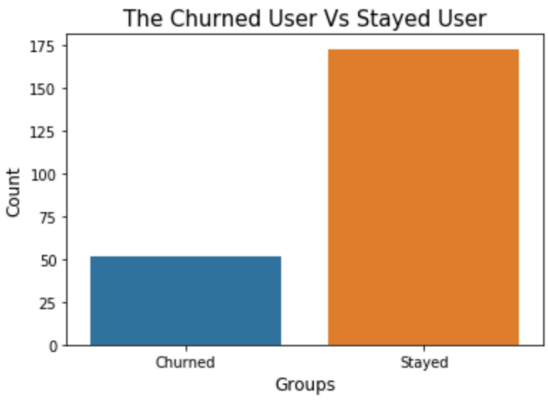
探索性数据分析 & 特征工程

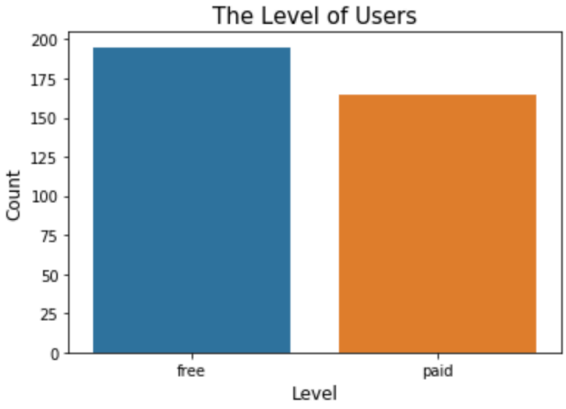
1 对数据集中流失用户与留存用户进行标记，发现在这个小数据集中流失用户与留存用户的比例约为1:3。
2 数据产生于2018年10月1日至12月31日。
3 流失用户平均活跃天数为23天，留存用户平均活跃天数46天

4 用户性别分布大致均衡，有部分用户同时拥有free与paid两个level









由于我们需要做的是根据用户的注册信息以及历史行为（浏览网页的时间分布）预测未来用户流失还是留存。在探索性分析中要通过分析找到流失用户组与留存用户组两组之间存在差别的特征。 要排除两组之间的人数不同以及平均活跃时间不同这两个因素对用户行为统计结果的影响。

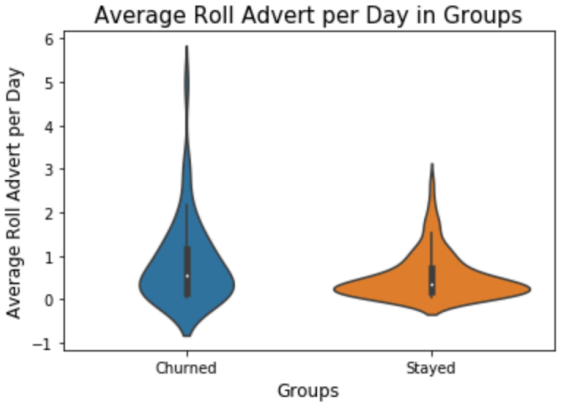
因为后续使用机器学习方法的目的是预测用户流失的概率，以对流失概率高的用户采取一定的优惠措施。机器学习方法针对的是个体而不是总体，所以两组之间的样本数量差异会干扰

机器学习方法的特征选取。在预测的时候某名用户在未来留存或流失是未知的，所以样本每名用户的活跃时间对机器学习方法没有参考价值

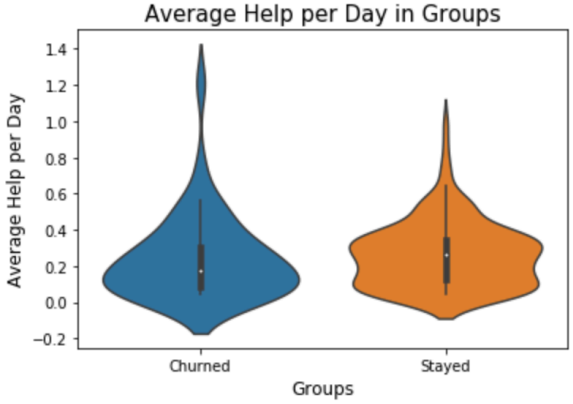
基于控制变量的考虑排除两组之间人数的差别以及平均活跃时间不同这两个因素可以更好地考察两组之间其他特征的差异（如播放音乐的数量）

经过探索性分析，有以下特征可以用来预测未来用户留存或流失：
1 用户每天平均Roll Advert的数量
2 性别
3 用户平均每天Help数量
4 用户平均每天Setings数量
5 用户平均每天Add Friend数量
6 用户平均每天Downgrade数量
7 用户平均每天Thumbs Up数量
8 用户每天平均播放音乐数量
9 用户平均每天Add to Playlist数量
按照流失用户组与留存用户组特征差异由大到小排序

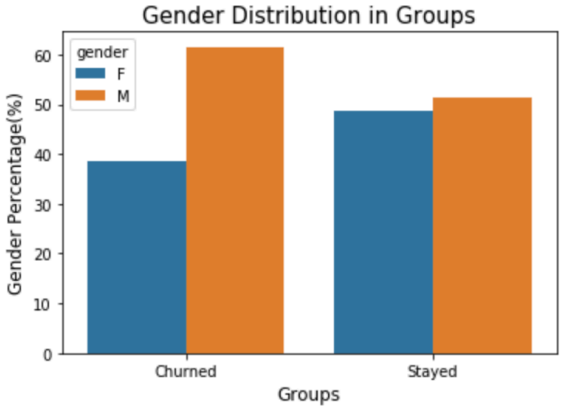
1 用户每天平均Roll Advert的数量



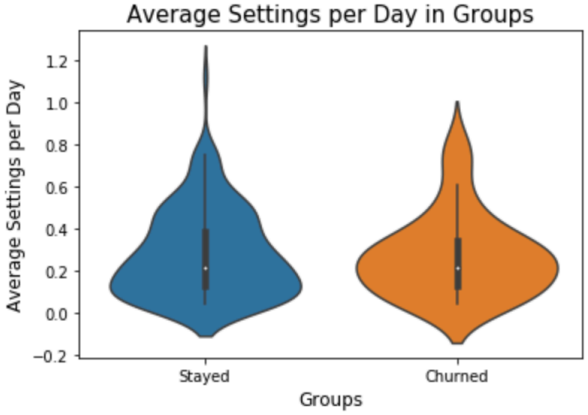
3 用户平均每天Help数量



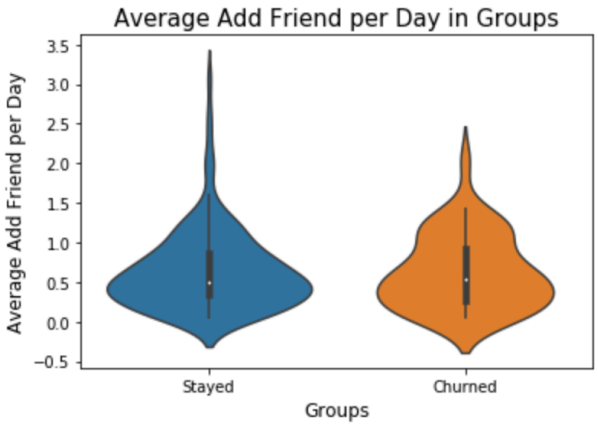
2 性别



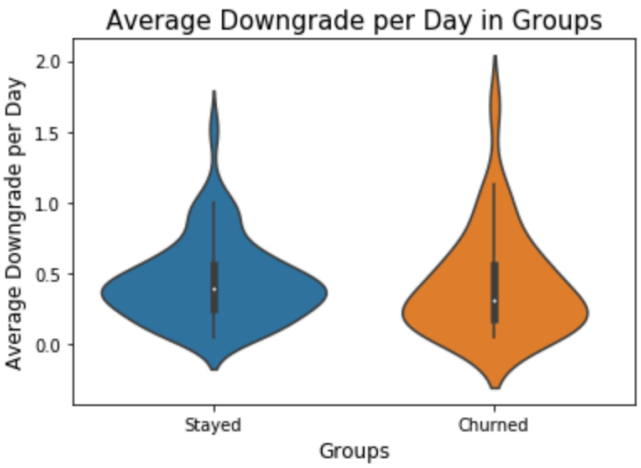
4 用户平均每天Setings数量



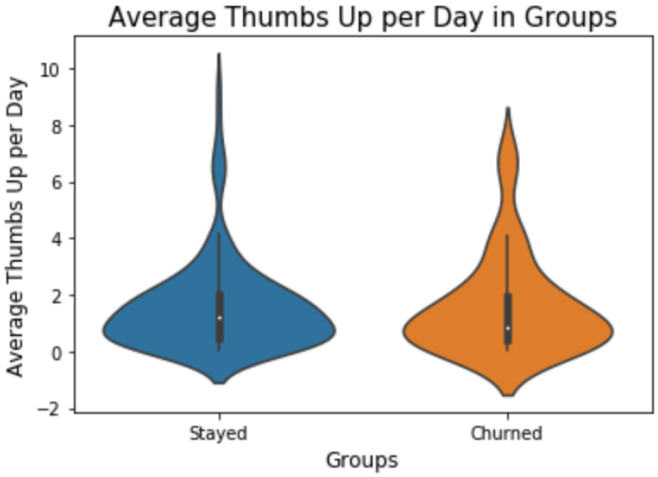
5 用户平均每天Add Friend数量



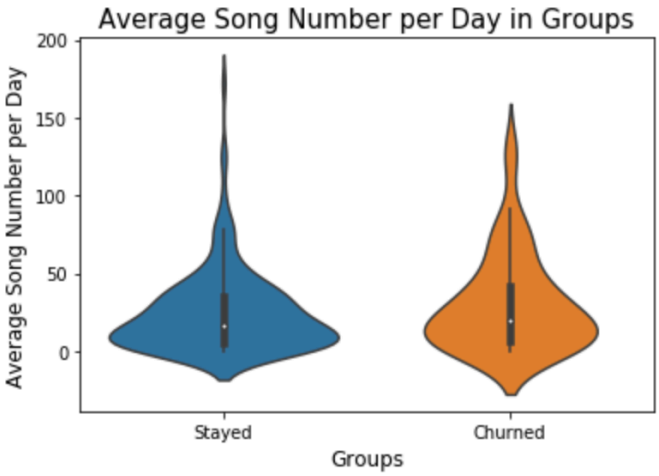
6 用户平均每天Downgrade数量



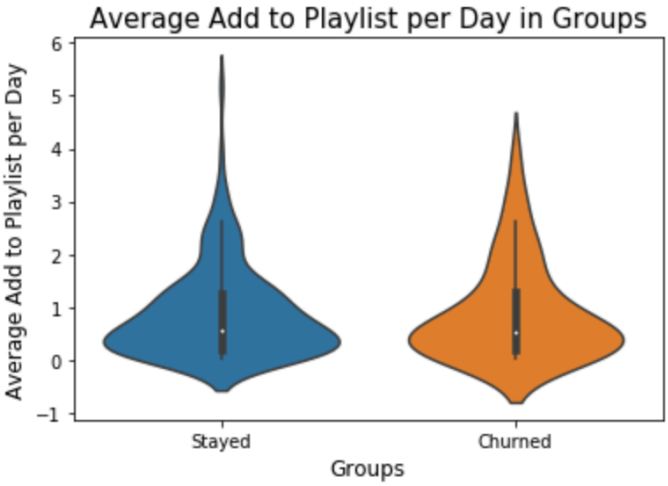
7 用户平均每天Thumbs Up数量



8 用户每天平均播放音乐数量



9 用户平均每天Add to Playlist数量



建模 & 根据指标进行模型优化

由于Spark本身的局限性，机器学习算法选用随输入数据大小线性扩展到算法，项目处理的是分类问题，可以选用LogisticRegression, LinearSVC, RandomForest, GradientBoosting四种分类器。

一 首先建立基准模型，考虑到该问题为非均衡分类，共有两种基准模型

1 预测所有用户全部留存
Accuracy: 0.7647058823529411
F1 Score: 0.6627450980392157
2 预测所有用户全部流失
Accuracy: 0.23529411764705882
F1 Score: 0.0896358543417367
训练调参得到的模型必须好于基准模型

二 使用默认模型参数在训练集上完成训练，并在测试集上进行测试，记录F1 Score，训练时间等评估指标

1 LogisticRegression
Training Time: 354.55540561676025
Training F1 Score: 0.7522275366712448
Testing Accuracy: 0.7941176470588235
Testing F1 Score: 0.7262656475019387

2 LinearSVC
Training Time: 589.9088156223297
Training F1 Score: 0.6757330259205702
Testing Accuracy: 0.7647058823529411
Testing F1 Score: 0.6627450980392157

3 RandomForest
Training Time: 264.60654640197754
Training F1 Score: 0.730103036013706
Testing Accuracy: 0.7941176470588235
Testing F1 Score: 0.7262656475019387

4 GradientBoost
Training Time: 1057.0708003044128
Training F1 Score: 0.6992624501393876
Testing Accuracy: 0.7941176470588235
Testing F1 Score: 0.706288032454361

综合考虑训练时间与F1 Score，以及模型的特性，选择LogisticRegression与RandomForest两种模型进行超参数调整。因为LinearSVC与LogisticRegression原理相似，都取决于不同类别之间分界超平面附近的数据点，但LogisticRegression的训练时间明显少于LinearSVC。GradientBoost需要不断迭代，速度较慢，超参数调整需要耗费更多时间。

三 参数调整优化

1 LogisticRegression
依次设定regularization parameter为0.0, 0.01, 0.1, 1.0, 10.0，分类器的稳健性不断增加
参数调整优化后
```
Training F1 Score: 0.7746464359884134
Testing Accuracy: 0.7941176470588235
Testing F1 Score: 0.7262656475019387
```

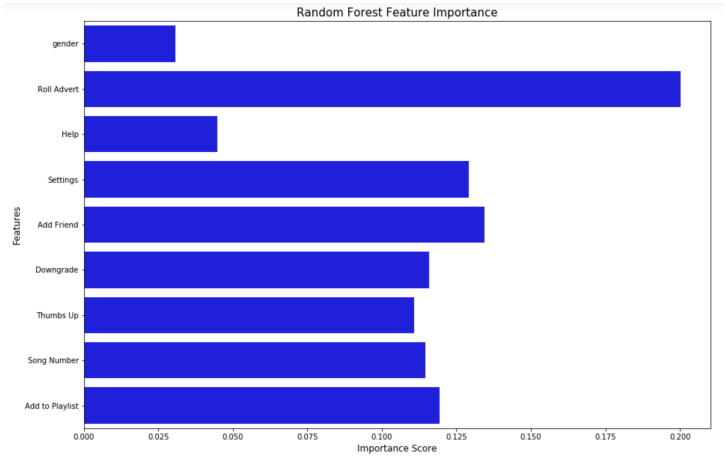
最佳的LogisticRegression模型regularization parameter为0.0,不使用regularization。最佳的LogisticRegression模型训练集的F1 Score为0.775,测试集的F1 Score为0.726，没有明显的过拟合。由于LogisticRegression模型只与分界的超平面附近的数据点有关，所以模型足够的稳健。模型进一步改善的余地也有限，因为最佳的LogisticRegression模型不使用regularization，无法再提升模型的灵活性。

2 RandomForest
依次设定树的数量为10,20,30,40,50，树的深度为3,5,7。分类器随着树的数量增加而稳定性增加
参数调整优化后
```
Training F1 Score: 0.7408733234394707
Testing Accuracy: 0.8235294117647058
Testing F1 Score: 0.7797160243407708
```

最佳的RandomForest模型树的数量为10，树的深度为7。最佳的RandomForest模型训练集的F1 Score为0.741，测试集的F1 Score为0.780，没有明显的过拟合。还可以继续增加树的深度以及减少树的数量来改善RandomForest模型的性能，提升其灵活性，但代价是模型的稳健性下降，更容易被干扰，也更容易过拟合。DecisionTree模型非常灵活，但极易过拟合，缺乏稳健性；RandomForest模型作为DecisionTree的集成方法，主要解决其稳健性差，容易过拟合的问题。

最佳模型是树的数量为10，树的深度为7的RandomForest模型。

特征重要性



观察发现最重要的特征是用户每天平均Roll Advert的数量，其次是用户平均每天Add Friend数量，用户平均每天Settings数量

改进

1 模型还有进一步改进的空间，如果使用更精确的GridSearch，但代价是需要更长的模型训练时间和成本。另外Spark能够使用的机器学习方法远少于sklearn等机器学习库，还有一些机器学习方法未能使用，比如神经网络/朴素贝叶斯等，神经网络可能带来更好的效果，但在大数据集上训练的成本也更高。如果Spark支持RandomSearch，BayesianSearch，可能做到更有效地搜索参数组合。

2 这只是在Workspace的子数据进行的分析与预测，下一步需要在AWS的完整数据集上进一步分析与预测。有一些用来预测用户流失情况的特征可能不适用于完整数据集，整个流程有可能要重新调整。

3 模型需要在robust与F1 Score之间找到平衡。使用更灵活的模型可以更好的拟合数据，提升F1 Score，但代价是容易过拟合，不够robust，容易被outliers干扰；使用更稳健的模型F1 Score较低，数据拟合程度不够好，甚至欠拟合，好处是不容易被outliers干扰。需要根据具体的应用场景去设置评估指标的阈值，选择更灵活的模型或者更稳健的模型。

项目的难点以及有趣的地方

1 Spark的调试过程中出现了一些Bug，但报错信息不好理解。查阅了stack overflow等平台的解决方案

2 Spark使用惰性评估，数据的统计信息显示以及可视化速度较慢或者无法实现，数据探索性分析的最佳解决方案通常是转化成Pandas后再进行处理

3 Spark的转换器可以灵活自定义，可以引入很多有趣的变换来提升模型的性能

参考资料：

1 8 Tactics to Combat Imbalanced Classes in Your Machine Learning Dataset
https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/
2 Learning Apache Spark with Python
https://runawayhorse001.github.io/LearningApacheSpark/classification.html
3 PySpark Pandas Seaborn Document