二〇一八年十一月二十七日星期二

下午5时6分

鉴于8月份以来一直进行实验，如今已经是第三个月，自己努力在本月有个好的结果。然而，最近实验笔记虽然从1写到了7，但是十分杂乱，也没有系统的整理，故今天正好准备重回RLV的分析思路，于是趁热打铁来整理下。

1. 基于离职人际关系的低满意度分析

基本假设1：用户的人际关系中，如果有离职的朋友，则会对用户本身的工作满意度造成负面影响，其具体的分析为：(1)如果朋友主动离职跳槽，说明当前单位的待遇或者工作情况不尽如人意，自己会不满足于当前的工作现状，萌生跳槽念头；（2）如果朋友被单位解雇，则出于友情会同情朋友，进而对单位产生不满；

基本假设2：跳槽离职行为受到反生产行为CPB-O与CPB-I的显著影响，而反生产行为受到工作满意度的影响；因此通过刻画用户的工作满意度情况，有可能预测员工由离职影响的离职意愿高低；

基本公式：

我们分析认为，用户的离职朋友的影响可以从三个维度刻画：

① 用户与离职人员的人格距离：当前的计算方法是考虑“物以类聚，人以群分”，因此假设OCEAN越接近的人越容易成为朋友，实际计算采取的OCEAN分数的欧氏距离；

② 用户与离职人员的组织结构距离，一般假设同一个团队、同一个部门中更容易认识形成联系，因此这里采用LDAP的四元结构作为用户的单位位置标识，而后进行四位异或码表示的0-15作为用户与离职人员的单位距离；

③ 用户与离职人员的邮件通讯往来，最关键的数据，直接体现了二者工作中的联系情况，当前阶段主要采用了以下几个维度：

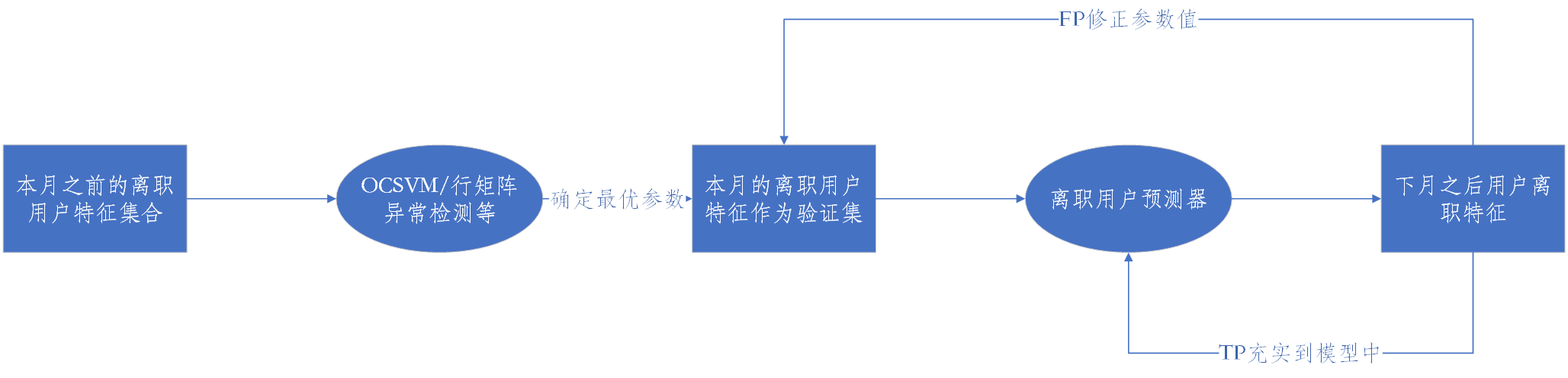
user\_id,leave\_contact\_user,O\_Score,C\_Score,E\_Score,A\_Score,N\_Score,Dis\_OCEAN,OS\_Code\_1,OS\_Code\_2,OS\_Code\_3,OS\_Code\_4,Dis\_OS,Email\_Ratio,Cnt\_Send,Avg\_Send\_Size,Avg\_Send\_Attach,Cent\_Recv,Avg\_Recv\_Size,Avg\_Recv\_Attach,Cnt\_Send\_Days,Cnt\_Recv\_Days

自己初步设想了，整体的循环应包含两个大模块：

* 预测器模块：细分为数据准备模块、特征构建模块、RelationLevel计算模块、特征保存模块（特征+RL数值）以及最终的结果输出模块；
* 检验反馈模块：细分为下个月标签提取模块（离职用户+迟到早退用户）、FN+FP计算模块、预测阈值反馈模块

而我们计算的公式则是：

整体的检测框架为单月分析预测：



接下来需要选择一个传统方法漏掉的用户进行分析，查看是哪个因素更为重要，从而对公式进行调整



二〇一八年十一月二十八日星期三

下午4时34分

我们今天的目标，有两点小想法：

1. 现在计算用户当月的RLV时依据着OCEAN差距，OS差距，通信天数，以及邮件通信信息量四个维度进行，那么，是否这四个维度中有着对于区分Insiders更为重要的元素呢？比如对于关系亲密度而言，可能邮件通信更重要？
2. CERT5.2数据集是否区分了离职和解雇的员工呢？对于Insiders而言的离职联系人与普通用户的离职联系人之间是否有什么没有发现的区别呢？

依据上述问题，决定分别进行下研究。

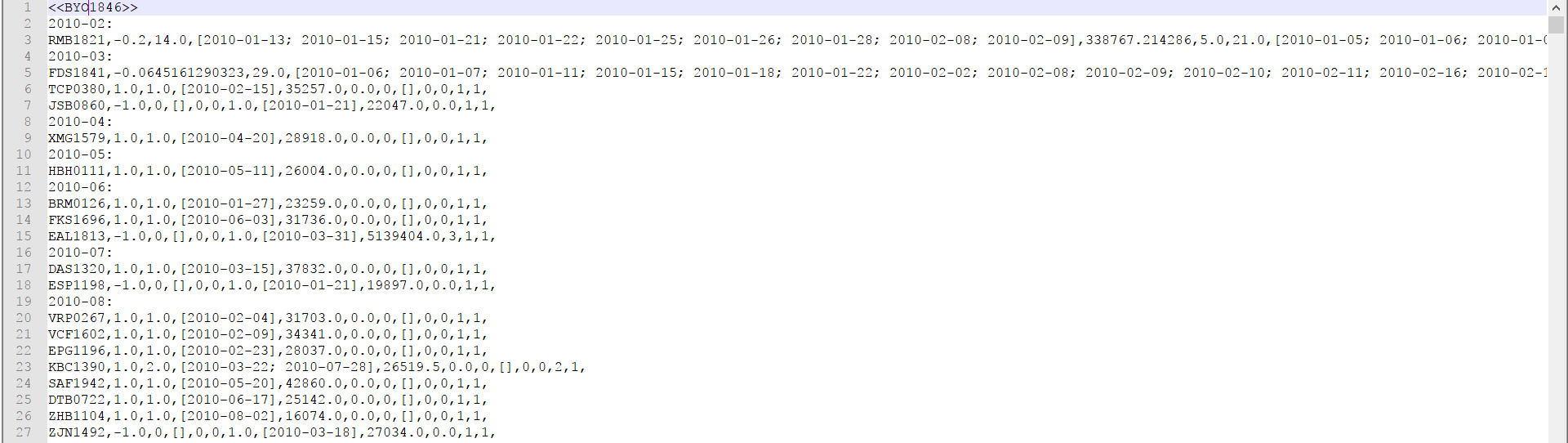
首先，我们：

1. 获取Insiders\_2的用户以及离职的时间
2. 获取在Insiders\_2离职前，提前离职的邮件联系人以及联系特征；
3. 将上述结果汇总成为一张表格供（2）分析；
4. 选择出Insiders中的几个用户，依次列出有记录的，其process特征在当月scale后的结果，从而可以分析出该用户哪个维度的特征更为偏离；

在编写程序的过程中，突然发现一个严重的问题：自己统计用户邮件特征的时候没有考虑到2010-01月份的记录！虚惊一场！当时自己已经考虑了，计算了月份，统计了所有离职月份前的邮件通信。

任务③完成！

三类攻击场景的联系人数据保存在CERT5.2\_Insiders-\*\_EmailFeats.csv中，该程序可以十分方便的在后续需要时分析正常用户的数据，以进行对比。





2018年11月29日星期四

上午11时7分

早上。

依据重新提取的基于Team决定的上下班时间，对于CERT5.2全体用户在2010-01：2011-04月份的出勤数据进行了分析，结果其结果同使用标准时间大同小异，单从单月预测结果来看（即预测2010-08月离职用户，仅使用2010-07月份用户出勤记录），30个Insiders\_2中，有10个用户使用中位数无法区分，准确而言是其中9个用户缺勤为0！，另外21个用户勉强可以采用Over中位数的形式进行区分。

即：由于9个用户的出勤堪称完美，因此从出勤迟到早退的角度分析，无论如何分析都无法弥补上述9个用户的不足。

为了避免以后对于方法的疏忽模糊，现对当前实验的出勤分析方法进行简单的小结。

1. 提取的特征形式：

* User\_id, workon\_time, workoff\_time, cnt\_late, cnt\_early, month\_work\_days, cnt\_ledays, ledays
* BGR0917,6.5,18.0,15.0,1.0,29,-1（末字段-1表示不存在同时迟到早退的天）
* MYB0686,6.5,18.5,7.0,7.0,21,2,2010-07-19,2010-07-27,（2010-07\_early\_late\_team\_feats.csv）

1. 计算当月缺勤风险指数的方法是：



上述公式体现了分别考虑迟到与早退的归一化结果带来的风险，其中可能多次计算同时迟到与早退的天。

依据上述结果，依靠当月排序Over\_Median的方法，30个用户中可以区分出21个用户，1个用户略低于中位数，9个用户缺勤为0.

因此，如果仅考虑同时具有跳槽+缺勤表现的Insiders而言，或许可以少分析近一半的用户；

而对于直接跳槽的这种用户，仅能依靠分析其离职联系人的影响。

另外，今天上午还顺手分析了全体用户的LEDdays的分布，Insiders\_2的结果并不同意，尤其是只有1天同时迟到早退的用户排序十分靠后，难以分析：

1487 ['MGB1235', '1.0', '266.0', '0.00375939849624']

1488 ['JAL0811', '1.0', '291.0', '0.00343642611684']

2018年11月29日星期四

下午的主要工作依旧在来回调试程序，发现了原先编程时的错误，由于数据量巨大，因此验证其来极其麻烦。

我们整理了CERT5.2中，Insiders2的与离职的用户的邮件特征，得到两个文件：

* CERT5.2\_Insiders-2\_EmailFeats\_Pure.csv：仅考虑到该用户离职前离职用户的联系
* CERT5.2\_Insiders-2\_EmailFeats\_All.csv：考虑到了整个周期内与离职用户的联系
* 即：All文件中涵盖了一些这样的用户：这些用户在Insiders离职后才离职，但是在Insiders离职前有着密切的邮件往来；