2018年11月4日星期日

上午11时12分

CERT5.2跳槽用户主观特征识别检测研究

背景：

CERT5.2中，场景二攻击者主要刻画了因为寻求更好待遇而跳槽的内部用户，这些用户在经历了猎头咨询、邮件自荐、拷贝重要资料等步骤后，最终离开单位。

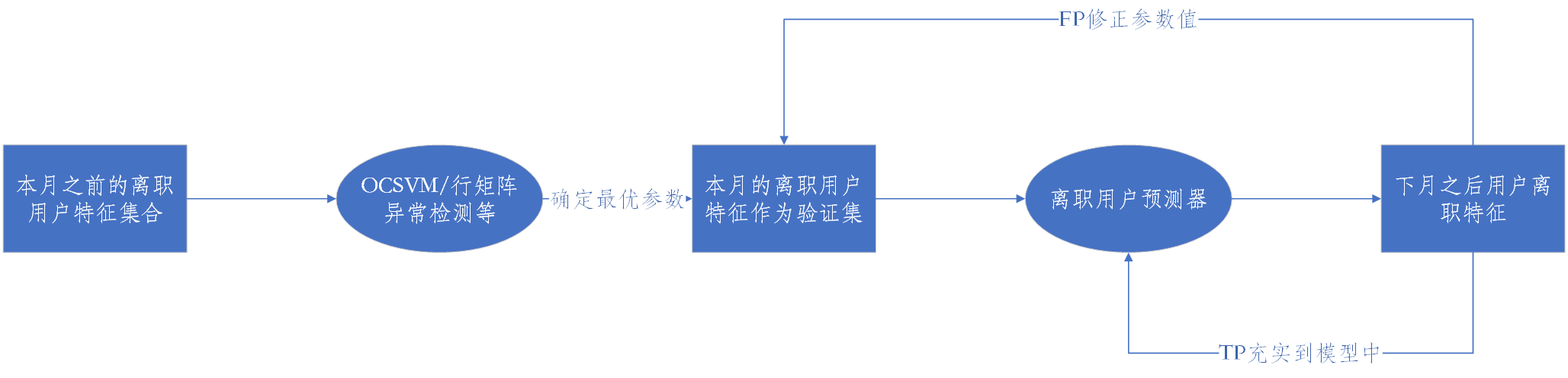
假设分析：

最早的研究工作聚焦于反生产行为CPB对于用户JS的影响，其关键来自于自身CPB的基础和周围工作环境的CPB影响；然后选择正常的用户进行了OCSVM训练，力图从中检测出JS可疑的用户。实验结果可以从4000个用户中识别出全部4个Insiders，而且识别出了400个用户集合。现在看来，实在没有必要在多种场景上验证主客观检测，因为数据场景描述中，仅明确了场景二跳槽攻击者受到工作满意度的影响；而工作满意度的影响主要是用户的上下班准时与否以及跳槽离职；

紧接着，今年十月份自己开始使用CERT5.2数据集进行重新研究，原因一是CERT5.2数据集中用户数量少了一半，分析起来更为高效；而且CERT5.2中也添加了满意度JS作为潜在动态变量，影响着用户的跳槽行为。然而，自己受到数据论文逻辑的影响，一直在努力刻画用户的人际关系（relationship），然而，在尝试了邮件联系、组织结构等关联后，得到的结果依旧不能很好地区分出高危用户。自己一方面不知晓论文数据构造时的方法，一方面也不想单纯以论文为指向进行方法研究，希望自己可以提出一种通用的行为预测方法。

因此，我们转而考虑如果直接从离职用户的角度来建模呢？朴素的想法：提取离职用户的诸多主观客观特征，然后一起使用聚类算法，查看目标用户聚类的的群簇所占当前离职用户的比例，以此作为该用户与离职用户的“距离”，比例越大，说明该用户越可能在下个月离职；

机器学习方法：对于离职用户而言，设定训练集、验证集与测试集，其中训练集使用OCSVM训练模型，而验证集用来选择最佳的参数；最终测试集用于实际测试；并且根据反馈的效果每月进行改进；



2018年11月4日星期日

下午11时12分

为了进一步分析CERT6.2中离职用户的特点，将Insiders\_2的30个用户周围的离职时间进行分析。

以BYO1846为例：

Insider\_LaidOff\_0,RMB1821,FDS1841,TRC1838,

Insider\_LaidOff\_1,NWP1609,ZAD1621,TAG1610,MAR1075,JXH1061,KEW0198,LSM1382,VCF1602,KBC1390,TNB1616,IVS1411,SLL0193,MFM1400,CTT0639,HIS1394,MZO1066,

Insider\_LaidOff\_2,WSW1091,JHP1654,JDM0208,HSF1115,UAM1108,CIM1095,ZHB1104,PTM1432,CIF1430,WDT1634,DHS0204,DDR1649,HMK0653,MIB0203,HMS1658,

BYO1846： 2010-12月离职

RMB1821,2010-02

FDS841， 2010-03离职

TRC183，2010-09离职

2018年11月5日星期一

下午1时50分

通过分析用户的离职情况，发现了之前忽略的一个问题：

对于一个用户而言，其有作用的离职用于应明确定义为在其之前离职的用户，而之前自己考虑用户的离职关系，仅具体到了离职的月份，而并未细致考虑用户离职的日期。

比如对于BYO1846而言，其离职时间为2010-12-15，那么就不应当考虑2010-12-20离职的12/20/2010 16:00:30,MZO1066。

因此，需要重新更新CERT5.2中的离职用户信息表以及每个用户的离职关系表。

原本以为自己发现了离职后的邮件联系。

以RMB为例子

Rose Maisie Blackwell,RMB1821,22,19,48,14,32

Rose Maisie Blackwell

Rose Maisie Blackwell,RMB1821,Rose.Maisie.Blackwell@dtaa.com,Salesman,,1 - Executive,5 - SalesAndMarketing,2 - Sales,5 - RegionalSales,Donna Erin Black

BYO1846与RMB123的通信次数总共为60，但是发生在2个月的时间内

RMB1821,-0.428571428571,2.0,42436.0,0.0,5.0,145618.6,0.6,

RMB1821,-0.5,7.0,210405.142857,0.285714285714,21.0,165035.857143,0.238095238095,

**BYO1846收到的RMB邮件**

01/05/2010 10:25:56,BYO1846,PC-8402,Buffy.Yolanda.Ortiz@dtaa.com,Donna.Erin.Black@dtaa.com,,Rose.Maisie.Blackwell@dtaa.com,Receive,20958,

BYO1846发送给RMB的邮件

Line 1039: 02/09/2010

10:22:31,BYO1846,PC-8402,William.Orson.Sykes@dtaa.com;Rose.Maisie.Blackwell@dtaa.com;Germane.Fredericka.Velez@dtaa.com,,,Buffy.Yolanda.Ortiz@dtaa.com,Send,44280,

突然发现：

之前写的程序大多存在错误疏漏，导致由此得到的结果要么不准确，要么错失重要的结果！要命！

今后写实验程序，无论大小，均需要经过简单的示例验证才能进行下一步，每一步的结果必须尽量坚实有力，别都分析了两个礼拜，再来回头查看最初的结果，想不起算法代码！

2018年11月5日星期一

下午3时52分

下午自己重写了之前的邮件过滤程序，考虑多发多收邮件的情况，并加入了邮件发送与接收通信天数的新特征，以便于计算日均发送与接收邮件。

作为验证：

以BYO1846用户为例：

RMB1821,-0.5,**7.0,5.0**,210405.142857,0.285714285714,**21.0,94.0**,165035.857143,0.238095238095,

其中RMB用户的邮件名为Rose.Maisie.Blackwell

这里总共记录了5+94=99天，明显与只有60条记录不符合

RMB用户记录太多，不容易矫正，使用

WDS1286,-1.0,0.0,0.0,0.0,0.0,1.0,1.0,1451552.0,1.0,

二封收到的邮件也都遗失了，开工修补

Willa.Dora.Sweeney

在努力修改的过程中，一个下午过去了，丝毫没有进展，干脆重新写吧，像这样几百行的代码几乎无法修改再读。

重写的目标：

1. 记录与目标用户有关的所有邮件通讯中，体现出的邮件联系人；
2. 不再专注1vs1的邮件通讯，包括邮件1vsN与Nvs1的邮件通讯；
3. 记录特征扩展为9个：

分别是：

邮件收发比，发送邮件总数，发送邮件行为的天数，发送的总字数，发送的总附件，接收的邮件总数，接收的邮件行为天数，接收的总字数，接收的总附件个数

下午下班时继续上述编程，终于将send邮件部分编写完毕，该部分针对用户Allen.Ashton.Buckley的send邮件JLF1315进行了验证，并且在两次send时准确记录了特征变化

再次联系用户 JLF1315 此时特征为： [2.0, ['2010-01-02', '2010-01-08'], '2053925619', 0.0]

46行用户验证通过

再次收到用户 HEB1313 的来信，此时特征为： [2.0, ['2010-01-04', '2010-01-08'], '4219039812', 0.0]

2018年11月6日星期二

上午6时31分

早上根据昨天上午的实验继续写完，得到一个测试用例：关于测试用JLF1315, 程序计算其特征为：

['**JLF1315**', -0.037037037037037035, 39.0, ['2010-01-02', '2010-01-08', '2010-01-15', '2010-01-20', '2010-03-02', '2010-03-10', '2010-03-11', '2010-03-15', '2010-04-05', '2010-04-28', '2010-06-08', '2010-06-21', '2010-07-01', '2010-07-13', '2010-07-14', '2010-08-01', '2010-08-15', '2010-08-18', '2010-09-14', '2010-10-27', '2010-11-08', '2010-11-22', '2010-11-26', '2011-01-01', '2011-01-05', '2011-01-21', '2011-01-22', '2011-01-31', '2011-02-08', '2011-02-11', '2011-02-25', '2011-03-02', '2011-03-06', '2011-03-08', '2011-04-11', '2011-04-26', '2011-04-29', '2011-05-07', '2011-05-20'], '205392561920672327903445045955445791862721575392523788226109278011566317211413342623935703297084173526349305378530332529320190013443932970257255835323768306042227323941514613339191602331919519118389124874', 10.0, 42.0, ['2010-01-07', '2010-01-20', '2010-01-30', '2010-02-01', '2010-02-16', '2010-02-23', '2010-02-25', '2010-03-03', '2010-03-08', '2010-03-09', '2010-03-29', '2010-04-07', '2010-05-05', '2010-05-19', '2010-05-21', '2010-05-27', '2010-06-22', '2010-07-16', '2010-08-05', '2010-08-25', '2010-09-03', '2010-09-17', '2010-10-13', '2010-10-18', '2010-11-30', '2010-12-27', '2011-01-18', '2011-01-25', '2011-02-03', '2011-02-25', '2011-03-02', '2011-03-04', '2011-03-07', '2011-03-09', '2011-03-20', '2011-03-23', '2011-03-26', '2011-04-28', '2011-05-12', '2011-05-17', '2011-06-01'], '39127215378323637807698253472621429909226062697438649256654034982324868012522546306681909514010903207130248403337728570227647534532888616353751988590645027774225667442611109947233290332941182635125916211252008610459224342156043725248', 23.0]

email\_feat is **['JLF1315'**, **-0.037037037037037035**, 39.0, ['2010-01-02', '2010-01-08', '2010-01-15', '2010-01-20', '2010-03-02', '2010-03-10', '2010-03-11', '2010-03-15', '2010-04-05', '2010-04-28', '2010-06-08', '2010-06-21', '2010-07-01', '2010-07-13', '2010-07-14', '2010-08-01', '2010-08-15', '2010-08-18', '2010-09-14', '2010-10-27', '2010-11-08', '2010-11-22', '2010-11-26', '2011-01-01', '2011-01-05', '2011-01-21', '2011-01-22', '2011-01-31', '2011-02-08', '2011-02-11', '2011-02-25', '2011-03-02', '2011-03-06', '2011-03-08', '2011-04-11', '2011-04-26', '2011-04-29', '2011-05-07', '2011-05-20'], 9453780.0, 10.0, 42.0, ['2010-01-07', '2010-01-20', '2010-01-30', '2010-02-01', '2010-02-16', '2010-02-23', '2010-02-25', '2010-03-03', '2010-03-08', '2010-03-09', '2010-03-29', '2010-04-07', '2010-05-05', '2010-05-19', '2010-05-21', '2010-05-27', '2010-06-22', '2010-07-16', '2010-08-05', '2010-08-25', '2010-09-03', '2010-09-17', '2010-10-13', '2010-10-18', '2010-11-30', '2010-12-27', '2011-01-18', '2011-01-25', '2011-02-03', '2011-02-25', '2011-03-02', '2011-03-04', '2011-03-07', '2011-03-09', '2011-03-20', '2011-03-23', '2011-03-26', '2011-04-28', '2011-05-12', '2011-05-17', '2011-06-01'], 21172138.0, 23.0]

2018年11月6日星期二

上午10时11分

经过昨天下班与今早的努力，终于重写了整个代码。

并且经过了阶段性验证，从而具备了较高的可信度。

以一个具体用户为例说明特征文件各个字段

CBW1826,-1.0,0,[],0,0,1.0,[2011-05-16],26462.0,0.0,1,1

各个特征字段意义分别为：

* User\_id
* EmailRatio
* Cnt\_Send
* Send\_Days
* Cnt\_Send\_Size on Each Cnt\_Send
* Cnt\_Attach
* Cnt\_Recv
* Recv\_Days
* Cnt\_Recv\_Size on Each Cnt\_Recv
* Cnt\_Attach
* Cnt\_Send\_days
* Cnt\_Recv\_days

2018年11月7日星期三

下午3时5分

周三的实验

今天开始重新启动JS建模。

首先需要研究的一个问题是，在CERT5.2中离职单位的用户一共越有237人，这些人里面有主动离开的“跳槽/离职”，也有因为工作原因被“解雇”而被动离开的，那么是否可以从数据特征中区分出两类人群呢？

我们先来看一个典型的主动离职者：BYO1846

其离职前最后一天为2010-12-15日，当天依旧存在五类行为记录

1. 收到的邮件依旧有许多来自于组织内部的邮件；
2. File：8次文件拷贝到移动设备的行为；
3. 正常上班，略微早下班

再看看RMB1821，离开单位时间为2010-02-09

最后一天存在三类记录。

我们来重点看看BYO1846最后一天与RMB1821最后一天的所有记录

通过编写实验2程序，我们分析了BYO1846与RMB1821用户最后一天的邮件通信，其邮件内容完全是随机生成的组合，不能作为判断离职/解雇的依据。

如果不能从数据上区分，那么我们的方法明显只有两个：

1. 从场景描述上，将明确的攻击者作为离职，而剩余离开单位的人员作为解雇；（即场景1-2-3（共69个用户，还有237-69=168人被看作解雇），最后一个场景的攻击者未离开单位）
2. 不作区分，直接考虑所有离开单位的人群；

2018年11月7日星期三

下午5时4分

接下来，利用下班后的一个小时，重点根据离职用户的时间点，重新更新用户的离职关系表。

不再重新书写代码，而是在原先代码的基础上进行修改、添加比较时间的部分。

基本思路是：读取得到原始未考虑时间的Leave\_Users\_Relationship列表，然后结合离职用户的时间信息Leave\_Users，筛选去掉时间不合适的用户。

过滤前：

0 层次长度为： 1 ['MMK1532', 'No:']

0 层次长度为： 0 ['Insider\_LaidOff\_0']

0 层次长度为： **11** ['Insider\_LaidOff\_1', 'WMH1300', 'JRC1963', 'BAR1328', 'RDP1751', 'DAS1320', 'HKK0881', 'CEW1960', 'ILG0879', 'DEO1964', 'SMS0432', 'VSB1317']

0 层次长度为： **21** ['Insider\_LaidOff\_2', 'CBC1504', 'JSB0860', 'KDP1706', 'FKH0864', 'RKW1936', 'BMR0865', 'SCO1719', 'ZJN1492', 'SAF1942', 'CHP1711', 'NAO1281', 'REF1924', 'JDJ1949', 'YBH1926', 'SNK1280', 'IAJ1729', 'DCA0857', 'LKC0405', 'VAH1292', 'TMT0851', 'JIP1503']

0 层次长度为： **87** ['Insider\_LaidOff\_3', 'JAT1218', 'ADL1898', 'KSW0708', 'LAS0256', 'GWO1660', 'MIB1265', 'TCP0380', 'CDO0684', 'CDG0770', 'CLL0306', 'QSG1150', 'QAP0266', 'OCW1127', 'PKS1187', 'GER0350', 'BSS0847', 'OCD1985', 'MPF0690', 'USM0703', 'RFP1918', 'FKS1696', 'CRD0272', 'AWW0718', 'EJO0236', 'ESP1198', 'MMR1458', 'JIB1258', 'ZIE0741', 'DTB0722', 'EPG1196', 'DMP0344', 'MDS0680', 'CNM0787', 'NBL1190', 'OSS1463', 'ALT1465', 'WHG1669', 'WFV0687', 'STH0353', 'RPJ1159', 'JKB0287', 'DNJ0740', 'ELM1123', 'DXF1662', 'SCI0778', 'ISW0738', 'AYG1697', 'LMW0837', 'ICB1890', 'NTG1667', 'PCK0271', 'DHR1157', 'ZVW1475', 'BRG0728', 'HPM0360', 'ACA1126', 'KJG1121', 'JOE1672', 'UKM0845', 'KVF1143', 'DCC1119', 'JDB1163', 'NEG0281', 'FZG0389', 'MGB1235', 'KHW0289', 'VRP0267', 'CAB1189', 'JAL0811', 'AMS1236', 'ALW0764', 'CQR1172', 'EGM1222', 'AMB0745', 'MBW1149', 'WHB1247', 'XBK0246', 'ZEH0685', 'JRB0759', 'JUP1472', 'WWW0701', 'HJO0779', 'DCV1185', 'KMO0382', 'CWW1120', 'HSN0675', 'DJH0253']

0 层次长度为： 119 ['Insider\_LaidOff\_4', 'RMB1821', 'FAM0495', 'WSW1091', 'SDL0541', 'CTH1812', 'JBG1375', 'LRF0549', 'JHP1654', 'NWP1609', 'JDM0208', 'HSF1115', 'FDS1841', 'NTV1777', 'GMM1037', 'MAF0467', 'ZAD1621', 'XMG1579', 'HBH0111', 'KBC0818', 'TAG1610', 'EAL1813', 'WBP0828', 'NWH0960', 'BRM0126', 'MAR1075', 'GWH0961', 'JXH1061', 'HFF0560', 'MMB0556', 'UAM1108', 'KEW0198', 'BNS0484', 'LSM1382', 'GFM1815', 'CIM1095', 'VCF1602', 'CKP0630', 'JKM1790', 'PBC0077', 'PTV0067', 'KBC1390', 'ZHB1104', 'PTM1432', 'MGM0539', 'IHC0561', 'OJC0930', 'SIS0042', 'CIF1430', 'KLB0918', 'TRC1838', 'TNB1616', 'IVS1411', 'WDT1634', 'DHS0204', 'SLL0193', 'MTD0971', 'DDR1649', 'MFM1400', 'TTR1792', 'GWG0497', 'MCP0611', 'CTT0639', 'HMK0653', 'VVG0624', 'MIB0203', 'LLW0179', 'RAT0514', 'GKW0043', 'NAH1366', 'PLF1030', 'TPO1049', 'ICB1354', 'RRS0056', 'MJA1784', 'HMS1658', 'BYO1846', 'EJV0094', 'HIS1394', 'GCB0118', 'HXP0976', 'MZO1066', 'KRC1348', 'DPK0954', 'KJH0475', 'JJW1785', 'RBC1624', 'LVF1626', 'ITA0159', 'CKL0652', 'ZKP0542', 'LJM1807', 'KCM0466', 'KFS1029', 'PTH0005', 'CDB1594', 'NIV1608', 'LCB1869', 'WSK1857', 'IJM0603', 'ELT1370', 'SQC1072', 'TMC0934', 'GTN1021', 'FIM0605', 'ETW0002', 'KSS1005', 'LMM0167', 'MEW0485', 'GPO1020', 'OKM1092', 'XAM0376', 'HBW0057', 'MTP1582', 'ACB0220', 'JKB1843', 'SLC1865', 'VPA0974', 'ACE1431', 'LAH0463']

过滤后：

0 层次长度为： 1 ['MMK1532', 'No:']

0 层次长度为： 0 ['Insider\_LaidOff\_0']

0 层次长度为： 11 ['Insider\_LaidOff\_1', 'WMH1300', 'JRC1963', 'BAR1328', 'RDP1751', 'DAS1320', 'HKK0881', 'CEW1960', 'ILG0879', 'DEO1964', 'SMS0432', 'VSB1317']

0 层次长度为： 21 ['Insider\_LaidOff\_2', 'CBC1504', 'JSB0860', 'KDP1706', 'FKH0864', 'RKW1936', 'BMR0865', 'SCO1719', 'ZJN1492', 'SAF1942', 'CHP1711', 'NAO1281', 'REF1924', 'JDJ1949', 'YBH1926', 'SNK1280', 'IAJ1729', 'DCA0857', 'LKC0405', 'VAH1292', 'TMT0851', 'JIP1503']

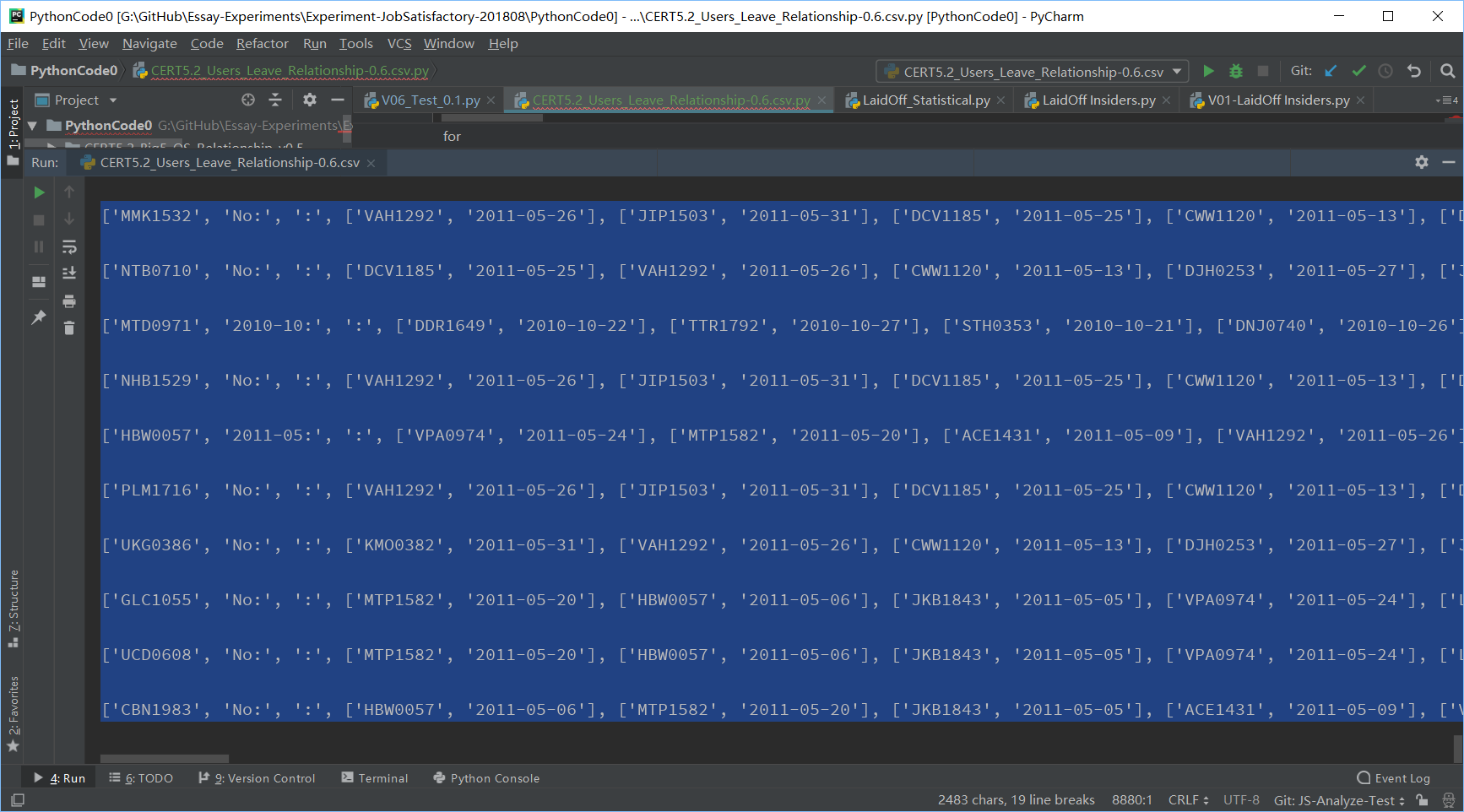
0 层次长度为： 87 ['Insider\_LaidOff\_3', 'JAT1218', 'ADL1898', 'KSW0708', 'LAS0256', 'GWO1660', 'MIB1265', 'TCP0380', 'CDO0684', 'CDG0770', 'CLL0306', 'QSG1150', 'QAP0266', 'OCW1127', 'PKS1187', 'GER0350', 'BSS0847', 'OCD1985', 'MPF0690', 'USM0703', 'RFP1918', 'FKS1696', 'CRD0272', 'AWW0718', 'EJO0236', 'ESP1198', 'MMR1458', 'JIB1258', 'ZIE0741', 'DTB0722', 'EPG1196', 'DMP0344', 'MDS0680', 'CNM0787', 'NBL1190', 'OSS1463', 'ALT1465', 'WHG1669', 'WFV0687', 'STH0353', 'RPJ1159', 'JKB0287', 'DNJ0740', 'ELM1123', 'DXF1662', 'SCI0778', 'ISW0738', 'AYG1697', 'LMW0837', 'ICB1890', 'NTG1667', 'PCK0271', 'DHR1157', 'ZVW1475', 'BRG0728', 'HPM0360', 'ACA1126', 'KJG1121', 'JOE1672', 'UKM0845', 'KVF1143', 'DCC1119', 'JDB1163', 'NEG0281', 'FZG0389', 'MGB1235', 'KHW0289', 'VRP0267', 'CAB1189', 'JAL0811', 'AMS1236', 'ALW0764', 'CQR1172', 'EGM1222', 'AMB0745', 'MBW1149', 'WHB1247', 'XBK0246', 'ZEH0685', 'JRB0759', 'JUP1472', 'WWW0701', 'HJO0779', 'DCV1185', 'KMO0382', 'CWW1120', 'HSN0675', 'DJH0253']

0 层次长度为： 119 ['Insider\_LaidOff\_4', 'RMB1821', 'FAM0495', 'WSW1091', 'SDL0541', 'CTH1812', 'JBG1375', 'LRF0549', 'JHP1654', 'NWP1609', 'JDM0208', 'HSF1115', 'FDS1841', 'NTV1777', 'GMM1037', 'MAF0467', 'ZAD1621', 'XMG1579', 'HBH0111', 'KBC0818', 'TAG1610', 'EAL1813', 'WBP0828', 'NWH0960', 'BRM0126', 'MAR1075', 'GWH0961', 'JXH1061', 'HFF0560', 'MMB0556', 'UAM1108', 'KEW0198', 'BNS0484', 'LSM1382', 'GFM1815', 'CIM1095', 'VCF1602', 'CKP0630', 'JKM1790', 'PBC0077', 'PTV0067', 'KBC1390', 'ZHB1104', 'PTM1432', 'MGM0539', 'IHC0561', 'OJC0930', 'SIS0042', 'CIF1430', 'KLB0918', 'TRC1838', 'TNB1616', 'IVS1411', 'WDT1634', 'DHS0204', 'SLL0193', 'MTD0971', 'DDR1649', 'MFM1400', 'TTR1792', 'GWG0497', 'MCP0611', 'CTT0639', 'HMK0653', 'VVG0624', 'MIB0203', 'LLW0179', 'RAT0514', 'GKW0043', 'NAH1366', 'PLF1030', 'TPO1049', 'ICB1354', 'RRS0056', 'MJA1784', 'HMS1658', 'BYO1846', 'EJV0094', 'HIS1394', 'GCB0118', 'HXP0976', 'MZO1066', 'KRC1348', 'DPK0954', 'KJH0475', 'JJW1785', 'RBC1624', 'LVF1626', 'ITA0159', 'CKL0652', 'ZKP0542', 'LJM1807', 'KCM0466', 'KFS1029', 'PTH0005', 'CDB1594', 'NIV1608', 'LCB1869', 'WSK1857', 'IJM0603', 'ELT1370', 'SQC1072', 'TMC0934', 'GTN1021', 'FIM0605', 'ETW0002', 'KSS1005', 'LMM0167', 'MEW0485', 'GPO1020', 'OKM1092', 'XAM0376', 'HBW0057', 'MTP1582', 'ACB0220', 'JKB1843', 'SLC1865', 'VPA0974', 'ACE1431', 'LAH0463']

2018年11月7日星期三

下午11时48分

通过实验，我们发现，严格时间后多数用户周围的离职用户数量大为减少，由此可以得到两个列表：一个新的Leave\_Relatonship与对应的因为时间而减少的用户，上述结果对于全职用户默认使用最后一个月的第一天作为deadline来过滤。



2018年11月8日星期四

上午9时21分

早上过来，将实验程序完成，并输出到文件，随机挑一个用户验证：

以用户REF1924， 2010-11离职为例：

通过过滤掉了原先统计的AYG1697:2010-11-16

在新生成的Leave\_Relationship\_Order中查看确认：

原先中Insider\_LaidOff\_3存在SCI0778

过滤后的文件中无该用户

验证成功

2018年11月8日星期四

上午11时37分

进一步进行验证，首先验证了场景二的30个用户是否存在非空的离职联系人？

结果验证通过：

场景二各个用户的离职联系人匹配结果数量为：

0 ['BYO1846', 33.0] 1 ['CHP1711', 17.0]

2 ['CIF1430', 8.0] 3 ['CKP0630', 2.0]

4 ['DCC1119', 33.0] 5 ['GWG0497', 20.0]

6 ['HIS1394', 34.0] 7 ['HMS1658', 17.0]

8 ['HSN0675', 43.0] 9 ['HXP0976', 12.0]

10 ['ICB1354', 19.0] 11 ['ITA0159', 18.0]

12 ['JAL0811', 29.0] 13 ['KSS1005', 27.0]

14 ['LVF1626', 38.0] 15 ['MCP0611', 6.0]

16 ['MDS0680', 18.0] 17 ['MGB1235', 10.0]

18 ['NAH1366', 7.0] 19 ['OKM1092', 27.0]

20 ['OSS1463', 8.0] 21 ['RRS0056', 20.0]

22 ['SIS0042', 12.0] 23 ['SNK1280', 41.0]

24 ['TMT0851', 49.0] 25 ['TNB1616', 17.0]

26 ['TRC1838', 17.0] 27 ['VCF1602', 10.0]

28 ['WDT1634', 5.0] 29 ['ZIE0741', 10.0]

2018年11月8日星期四

下午4时45分

开始自11月4日星期日，今天周四，预定了下周三回京的车票，今天明天两天务必完成实验；下周一/二可以进一步修正。

现在已经有了一个基本思路：

* 按月分析，预测下一个月的JS风险；
* 每月根据实际结果反馈，修正预测比例阈值；
* 每月的预测结果对错以离职和工作不守时为标准；

具体如何调整，需要在已有大体程序框架后再进一步补充完善。

在正式开始大程序框架前，先来写一个分析用户正常上下班时间的模块。

由于最早的离职者出现在2010-02，因此我们只使用最初的第一个月2009-12作为预测

由于CERT5.2构造数据时，用户的登录登出时间在一个固定时间附近小波动，因此，我们可以统计登录登出的时间段次数，从而投票选出次数最多的时间区间，然后将该区间的上限作为上班时间；同理登出次数最多的时间区间的上限作为下班时间。

现试着提取所有用户在2009-12月份的Logon/Login数据作为分析

原本计划从2009-12月开始建立所有用户的登录登出数据，不想对于JBI1134而言，虽然2009-12就在单位了，但是知道2010-01-02才有登录记录，因此，需要同时利用2009-12与2010-01才能建立用户的上下班时间

同时考虑两个月，用户JBI1134通过验证：

01/02/2010 02:24:51,JBI1134,PC-0168,Logon,

01/02/2010 02:38:28,JBI1134,PC-0168,Logoff,

01/02/2010 04:55:52,JBI1134,PC-2320,Logon,

01/02/2010 05:02:28,JBI1134,PC-2320,Logoff,

01/02/2010 06:36:55,JBI1134,PC-5439,Logon,

01/02/2010 06:38:42,JBI1134,PC-5439,Logoff,

01/02/2010 06:50:00,JBI1134,PC-7235,Logon,

01/02/2010 07:13:59,JBI1134,PC-2226,Logon,

01/02/2010 07:16:08,JBI1134,PC-2226,Logoff,

01/02/2010 10:41:45,JBI1134,PC-7235,Logon,

01/02/2010 14:10:33,JBI1134,PC-7872,Logon,

01/02/2010 14:29:38,JBI1134,PC-7872,Logoff,

01/02/2010 16:04:00,JBI1134,PC-7235,Logoff,