一、项目说明

1、背景

EDR通常会收集到大量的有关用户登录行为的告警日志，但并不是所有登录行为都是异常的，因此需更准确地辨别哪些登录行为异常行为。

2、概述

登录行为聚类分析，主要是以EDR所采集的公司内各终端数月的win-eventlog登录行为告警数据为基础，对用户登录行为进行聚类，
通过多次聚类，得到异常集群，然后与业务人员一同对该集群进行分析，判断其是否产生异常行为。

二、登录行为聚类过程

1、数据读取：

数据类型：win-eventlog的登录成功行为告警数据
数据读取：（1）python连接数据库直接读取形成dataframe（实时更新）；（2）读取数据下载到本地处理（离线分析）。

2、数据预处理和参数确定：

（1）数据形式

如特征表数据字典，该数据集由57个特征组成的12月1日至今的登录行为告警数据。

（2）初步业务特征选择

业务筛选特征:
首先通过业务了解，对特征进行初步筛选，仅选择与登录成功行为相关的特征。
保留以下特征：

```
['dtdlcgs','jycdlcgsjd','j7tdlcgs','j14tdlcgs','j28tdlcgs','ljdlcgs',
 't1_zhmyms','t7_zhmyms', 't14_zhmyms','t28_zhmyms','lj_zhmyms',
 't1_jcxxs','t7_jcxxs','t14_jcxxs','t28_jcxxs','lj_jcxxs',
 't1_gzts','t7_gzts','t14_gzts','t28_gzts','lj_gzts',
 't1_zydzs','t7_zydzs','t14_zydzs','t28_zydzs','lj_zydzs',
 't1_dljcs','t7_dljcs','t14_dljcs','t28_dljcs','lj_dljcs']
```

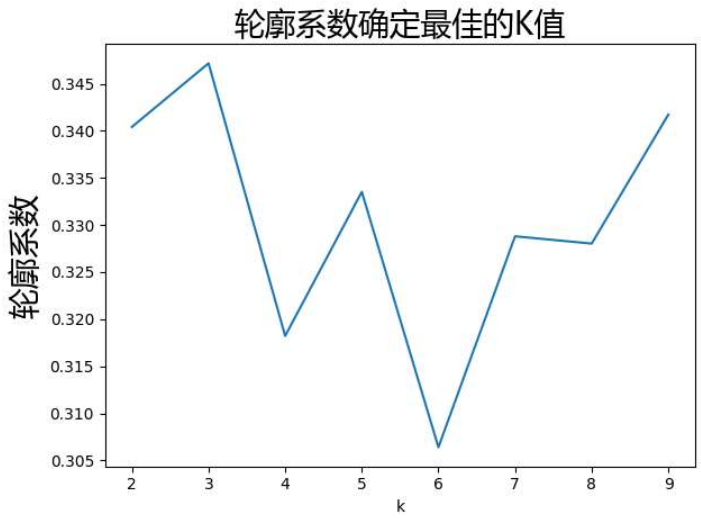
（3）缺失值处理、标准化

缺失值处理
删除缺失值：dataframe.dropna方法。
离散属性处理
pd.get_dummies独热编码，处理jycdlcgsjd离散变量，处理后得到73个特征
标准化处理
最大最小标准化

```
dataSet_sc = (dataSet-dataSet.min())/(dataSet.max()-dataSet.min())
```

（4）聚类簇数选择

轮廓系数法
采用轮廓系数法,簇心个数区间设置为3到8，对于不同的k值计算聚类模型的轮廓系数值，确定当前聚类阶段的簇数k-certain

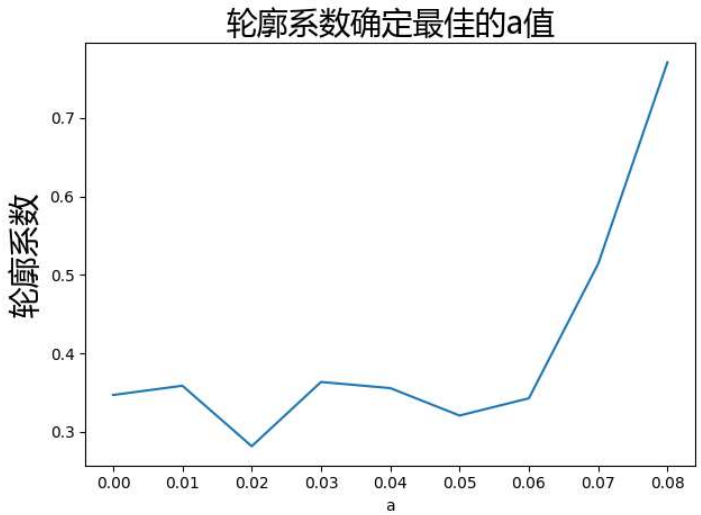



```
K = range(3,8) # 设置个数区间
coef = []
for k in K:
    km = KMeans(n_clusters=k,random_state = 0).fit(dataSet_sc) #构建kmeans模型并训练
    score = silhouette_score(dataSet_sc, km.labels_,sample_size=915) # 计算对应模型的轮廓系数
    coef.append(score)
plt.plot(K,coef) # K为x轴输出，coef是y轴输出
plt.xlabel('k')
font = FontProperties(fname=r'c:\windows\fonts\msyh.ttc', size=20)
plt.ylabel(u'轮廓系数', fontproperties=font)
plt.title(u'轮廓系数确定最佳的K值', fontproperties=font)
plt.show()
```

（5）二次特征选择

方差阈值法进行特征选择

统计各特征的方差,得到最小方差，最大方差，设定最小阈值，最大阈值，按10等分取步长.
根据不同的阈值，剔除方差大于阈值的特征，得到不同的特征矩阵，然后训练k=k-certain的聚类模型，计算轮廓系数值，
得到阈值a,筛选特征。



3、模型训练

(1)形成新dataframe

根据特征筛选的结果，标准化后的dataSet_sc剔除未被选择的特征生成新的dataframe特征矩阵new_data

```
如：
[False  True False False False False  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True]
False表示剔除，True表示保留。
drop_fea为剔除特征组成的列表。
newdata为dataSet_sc剔除相关特征后形成的数据集。
new_data = dataSet_sc.drop( drop_fea,axis=1)
```

(2) 训练模型

训练模型，簇数k=k_certain,初始化簇心方法init为kmeans++，n_init默认为10，选择最优结果。

```
model = KMeans(n_clusters=k-certain, random_state=0,max_iter=1000).fit(new_data.iloc[:,:].values)
```

(3) 模型结果输出

简单打印结果

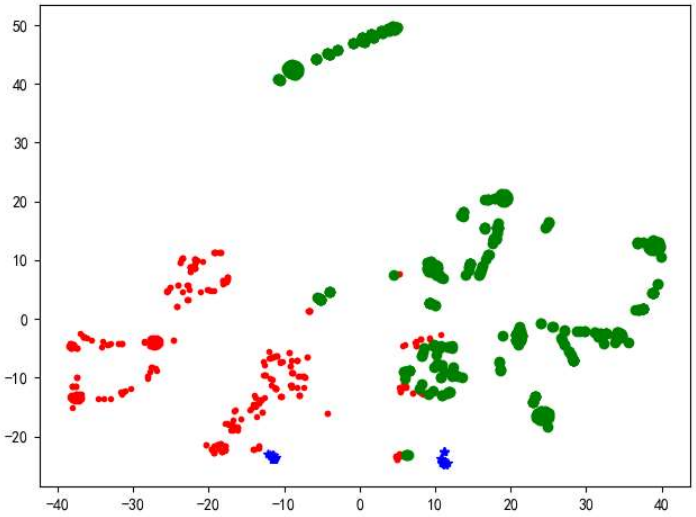
统计各类别种的样本数目，得到聚类中心，进行横向连接。

```
r1 = pd.Series(model.labels_).value_counts() #统计各个类别的数目
r2 = pd.DataFrame(model.cluster_centers_) #找出聚类中心
r = pd.concat([r2, r1], axis = 1) #横向连接（0是纵向），得到聚类中心对应的类别下的数目
         0         1         2         3   ...        69        70        71   0
0  0.217459  0.381137  0.375133  0.326451  ...  0.060000  0.500000  0.060000   50
1  0.003974  0.004813  0.003880  0.003232  ...  0.032401  0.346097  0.050074  679
2  0.003275  0.004007  0.003323  0.002924  ...  0.047847  0.388756  0.039474  836
```

输出原始数据及其所属簇类

```
r = pd.concat([new_data, pd.Series(model.labels_, index = new_data.index)], axis = 1)  #详细输出每个样本对应的
r.columns = list(new_data.columns) + [u'聚类类别'] #重命名表头
r.to_csv(r'E:\EDR_log_analysis\数据\聚类结果.csv') #保存结果
```

利用TSNE进行数据降维展示聚类结果



(4) 迭代聚类

输出多次聚类的结果

迭代聚类，直到各集群内的样本数目没有数量级上的明显差别。

如：

第一次聚类结果：

	0	1	2	3	...	28	29	30	0
0	0.418783	0.803030	0.764792	0.761314	...	0.900000	0.950000	0.950000	24
1	0.005150	0.614731	0.005805	0.004565	...	0.297518	0.304526	0.304526	685
2	0.003190	0.629779	0.003906	0.003237	...	0.179206	0.192056	0.192290	856

第二次聚类结果：

	0	1	2	3	...	28	29	30	0
0	0.008172	0.636818	0.010710	0.011190	...	0.215000	0.231875	0.232187	800
1	0.012168	0.611942	0.014693	0.014675	...	0.375527	0.384318	0.384318	711
2	0.000057	0.521212	0.000100	0.000116	...	0.250000	0.250000	0.250000	30

第三次聚类结果：

	0	1	2	3	...	28	29	30	0
0	0.004466	0.682318	0.006665	0.007494	...	0.025641	0.032967	0.032967	273
1	0.022927	0.809725	0.026840	0.025029	...	0.503876	0.511628	0.511628	258
2	0.017351	0.582645	0.020006	0.016401	...	0.250000	0.250000	0.250000	22
3	0.009315	0.604936	0.011945	0.012408	...	0.330523	0.352445	0.352867	593
4	0.005887	0.487173	0.007528	0.008668	...	0.275342	0.282877	0.282877	365

第四次聚类结果：

	0	1	2	3	...	28	29	30	0
0	0.012612	0.861405	0.016313	0.016264	...	0.319121	0.346899	0.346899	387
1	0.004425	0.674567	0.006273	0.007159	...	0.011673	0.019455	0.019455	257
2	0.005887	0.475811	0.007430	0.008622	...	0.275585	0.282895	0.282895	342
3	0.024154	0.800973	0.028246	0.026189	...	0.510288	0.512346	0.512346	243
4	0.003486	0.260140	0.004789	0.006070	...	0.341346	0.356731	0.357692	260

(5) 对多次聚类结果进行分析

迭代聚类过程中，已经将聚类结果中的异常集群标注为xxx_ourlier_an,表示它是与其他集群有明显异常的。接下来会将该集群内的样
内的样本进行对比分析，并与专业人士探讨其异常原因。

4、测试数据

模型保存与再利用

利用joblib模块进行模型的保存和索引使用。

模型测试

将近一周的用户登录数据，输入到各次迭代的聚类模型中，通过与各模型总的簇心计算距离比较，看哪些登录行为归类为异常集群，则判断该登录行为是异常的。