项目报告

2150998 张诚睿

代码：https://github.com/Neo0214/Geoscience-Big-Data

一、 预处理

基本的预处理标准为空值填充，特别处理之处在于通话时间的三个属性。

start_time、end_time、raw_dur 三者现实逻辑关系是知二得三，因此需要直接剔除确实属性超过两个的记录，只缺一个的记录可以计算补全缺失值。此处跨天结束的 end_time 我们全部按照日式计时法——24 点、25 点等表示凌晨，这样在后续数据使用时，如果需要同时使用开始结束时间，反而方便统一处理，不需要的地方只要-24 就可以回到标准值。

实现方式详见 src/pre.py

二、 用户基本行为

两个基本行为统计的总体实现类似，我们首先按主叫号码排序，然后依次遍历整个数据行，将同一主叫号码的记录合并统计数据，最后输出即可。

其中，每日平均通话次数的统计上，整个数据集包括二月份的通话记录，我们取数据集涵盖的时间 29 天为总天数；各个时间段通话时长所占比例的统计上，我们将所有记录计算为秒钟数，然后给各时段统计包含的通话总秒时间，最后计算比例。

实现方式详见 src/func/call_per_day.py、src/func/percentage_in_segments.py

结果见 result/call_per_day.xlsx、result/percentage_in_segments.xlsx

三、 用户行为特征

用户行为特征有很多角度可以出发去统计，我们首先聚焦于用户打电话的基本习惯上。

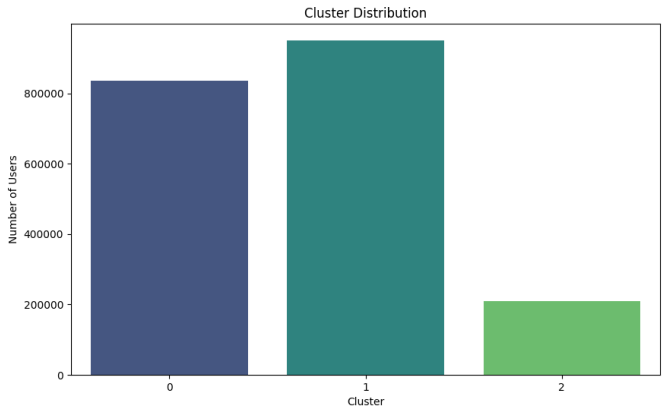
我们选取通话时长、开始结束时间（按小时标记）、通话类型作为使用的特征属性

```
features = df[['call_duration', 'start_hour', 'end_hour',
'call_type']]
```

为了避免不同量纲对聚类的影响，我们再标准化一次，之后分成三类，查看结果

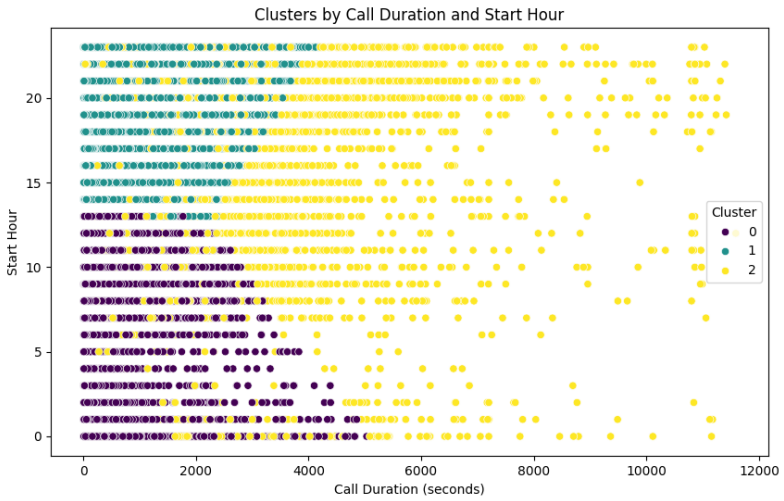
```
# 特征标准化
scaler = StandardScaler()
scaled_features = scaler.fit_transform(features)

# 使用 KMeans 聚类
n_clusters = 3
kmeans = KMeans(n_clusters=n_clusters, random_state=42)
df['cluster'] = kmeans.fit_predict(scaled_features)
```

聚类结果如下：



	call_durati	start_hour	end_hour	call_type
0	81.71218	10.11655	10.13587	1.001945
1	102.9437	17.20308	17.23069	1
2	221.7074	14.365	14.40347	2.173432

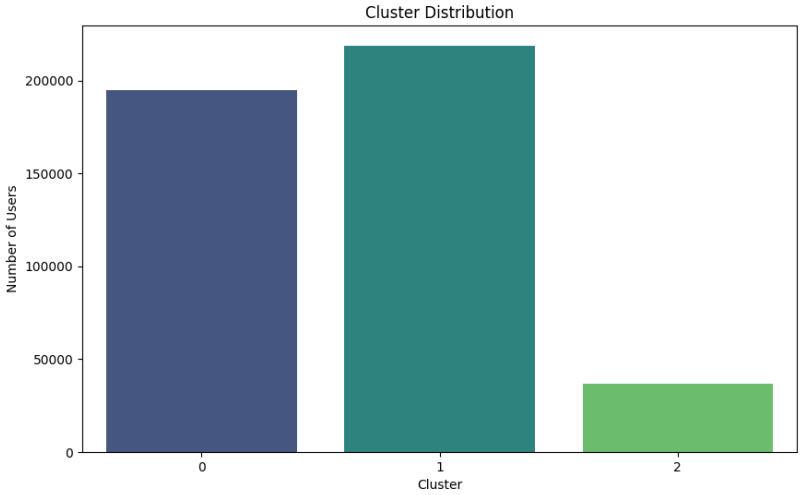
其中，从开始通话时间和通话时长来绘制散点图，可以很明显的看到三类不同打电话特点的记录。



无论是从数据还是散点图上看，Cluster0 记录喜欢在上午打电话，而且通话时长都不长；Cluster1 的记录喜欢在下午和晚上通话，时间也不长；Cluster2 群

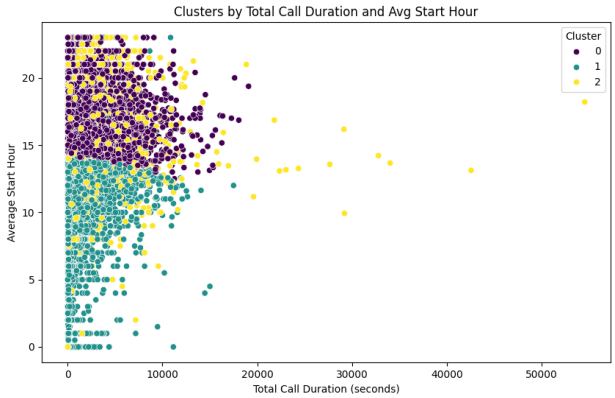
体主要区别在于通话时长普遍更长，而且长途和漫游多。

有了对记录的分析，我们现在转向对用户的聚类。针对同一用户的不同记录，我们将其合并，并取均值作为他的一般通话特征。聚类结果如下：

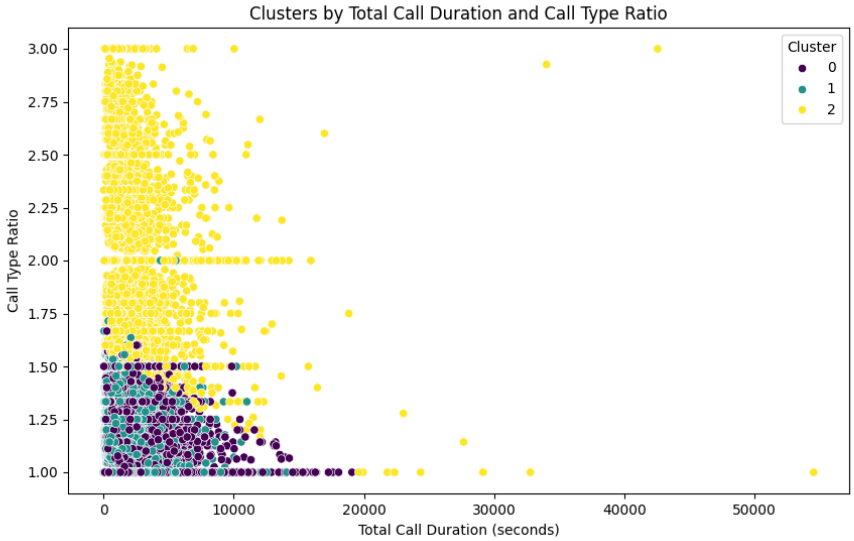


cluster	total_call_duration	avg_start_hour	avg_end_hour	call_type_ratio	cluster
0	493.8902855	16.47948305	16.51259923	1.047953148	0
1	422.5886462	11.86111158	11.88231666	1.045698186	1
2	653.4135416	14.70040465	14.74466586	2.118850055	2

散点图如下：



不难看出，0 和 1 簇的主要区分在于上午还是下午（包括晚上）通话，并且通话时间都不长。而 2 簇的特点在上图并不明显，他们主要在于喜欢打漫游和长途。用下面坐标轴的散点图可以很明显的看出这点。



在通话时间上簇 2 并没有很明显的特点，但是他们的市话明显比例更低，漫游和长途很多。

那么我们针对这一结果，考虑用随机森林去实现对通话类型的分类。上面的聚类中，通话类型是取用类型代表值的平均值来作为提取的用户特征，有逻辑上的道理，但是离散的属性值显然用离散的方式处理是更好的。因此，分类任务上，我们尝试将用户按通话类型分成三类。其中判定用户所属的通话类型的方法是取他所有通话记录中通话类型最多的类别。

我们选用的属性包括：

```
return {
    'called_city': group['called_city'].mode()[0],
    'calling_roam_city': group['calling_roam_city'].mode()[0],
    'called_roam_city': group['called_roam_city'].mode()[0],
    'start_hour': group['start_hour'].mean(),
    'raw_dur': group['raw_dur'].mean(),
    'calling_cell': group['calling_cell'].mode()[0],
    'call_type': group['call_type'].mode()[0]
}
```

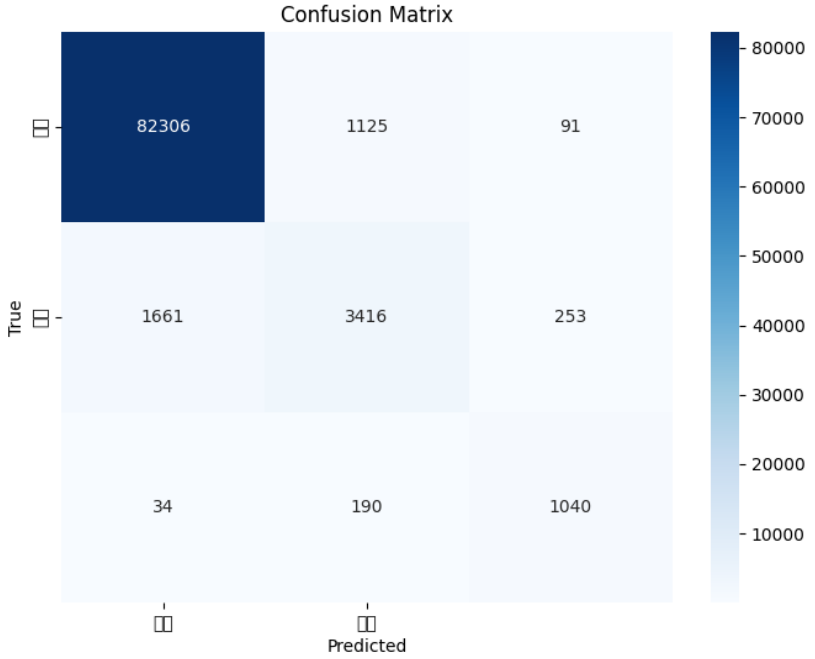
处理的原则是离散属性值按离散处理，逻辑上连续的值取平均处理。（时间虽然是离散的，但是逻辑上是连续的，因此取平均）

取数据的 20%作为测试集，训练结果如下：

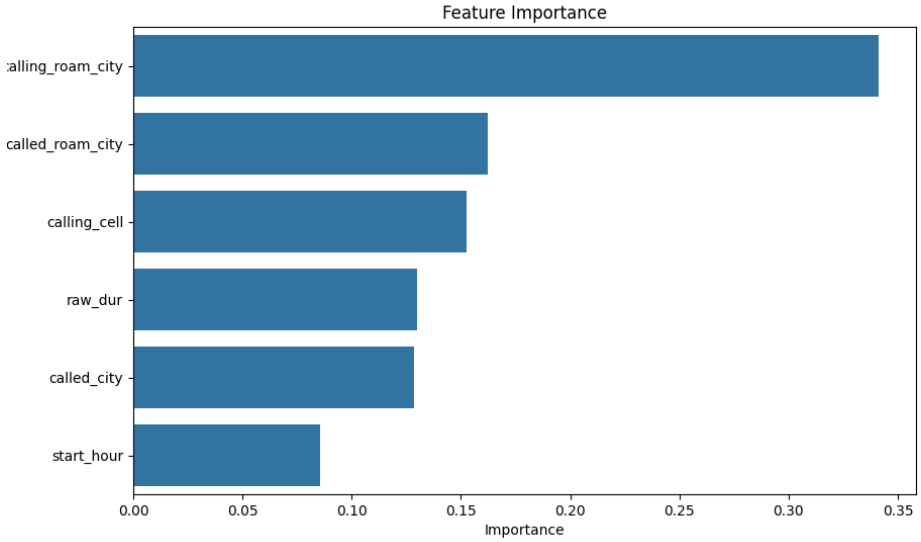
Classification Report:

	precision	recall	f1-score	support
1	0.98	0.99	0.98	83522
2	0.72	0.64	0.68	5330
3	0.75	0.82	0.79	1264
accuracy			0.96	90116
macro avg	0.82	0.82	0.82	90116
weighted avg	0.96	0.96	0.96	90116

可以看出类别 1（市话）的精确度最高，预测效果很好。漫游和长途相比要差一些，不过性能也在可接受范围之内。市话的效果好可能是受数据量大导致的更加拟合，而 2 和 3 可能是因为漫游和长途的通话者经常出差、从事外贸等，导致二者本身现实上的特征就相近，尤其是出差用户，漫游和长途的界限更不明显，最终导致这两类的分类效果不如市话。



从混淆矩阵来看，结论和上面一样。对角线上的值明显高于非对角线值。再看不同特征对模型的影响，如下图：



漫游城市和目标电话城市影响最大，这是显而易见的。而电话时间也有明显的影响，很可能是因为漫游长途占主导的用户所从事的行业、家庭状况等有趋同性。例如，外贸行业有时差，北漂们会在晚上给家人打长时间电话等。