## 0. 环境说明

本方案的代码环境为 python3.6，在运行代码之前需要确保安装以下库：

（1） xgboost 和 lightgbm：两个回归树模型
（2） sklearn：机器学习库
（3） geopy：计算球面距离的库
（4） ge：graph embedding 的算法库，安装方式见 github：https://github.com/shenweichen/GraphEmbedding
（5） 其他工具的库：networkx,pickle,tqdm

## 1. 代码目录结构

代码目录下共有三个文件夹和代码的 jupyter 文件：

```
lishuangli 4.0K 5月   10 11:36 cache
lishuangli 4.0K 5月   10 12:04 data
lishuangli  14K 5月   10 11:36 Fusion.ipynb
lishuangli  58K 5月   10 11:36 GraphEmbedding.ipynb
lishuangli  19K 5月   10 11:36 model-RULE.ipynb
lishuangli  34K 5月   10 12:09 model-TREE.ipynb
lishuangli   38 5月   10 11:36 output
```

（1）data 目录是输入的数据，该目录下已经存在补充的天气数据 weather.csv，在运行代码前需要将赛题提供的三个数据集 area_passenger_index.csv，area_passenger_info.csv 和 datafountain_competition_od.txt 以及提交样例文件 test_submit_example.csv 先移动到 data 目录，data 目录下的文件如下：

```
lishuangli  14M 5月   10 11:58 area_passenger_index.csv
lishuangli 105K 5月   10 11:58 area_passenger_info.csv
lishuangli 555M 5月   10 11:59 datafountain_competition_od.txt
lishuangli 4.1M 5月   10 12:04 test_submit_example.csv
lishuangli  955 5月   10 11:59 weather_data.csv
```

（2）cache 目录是代码运行产生的中间数据，最后包括区域空间表征向量文件，机器学习模型的预测结果和统计规则模型的预测结果等数据。
（3）output 目录是模型融合后的最终输出结果 test_submission_final.csv
（4）四个 jupyter 代码文件包括 GraphEmbedding.ipynb（空间特征学习）、model-TREE.ipynb（机器学习模型）、model-RULE.ipynb（统计规则模型）和 Fusion.ipynb（模型融合）四个文件。

## 2. 机器学习模型 model-TREE.ipybn

（1）在运行机器学习模型代码之前，需要先运行 GraphEmbedding.ipynb 用随机游走算法 node2vec 生成区域空间特征，由于该代码运行时间较长（完整生成大

概需要 1h），在提交的方案代码中 cache 文件下已经有生成好的区域空间特征：

```
lishuangli 4.0K 5月   10 11:36 area_embedding_0
lishuangli 4.0K 5月   10 11:36 area_embedding_1
```

所以可以直接运行机器学习模型的代码；如果要重新生成区域空间特征（如改变参数，换 graph embedding 算法），需要从头运行一遍 GraphEmbedding.ipynb，并在机器学习模型代码中修改为相应的读取文件路径。

（2）该代码文件分为几个部分：

（a）读取数据（load_data），定义两个模型（tree model: xgoost & lightgbm），定义特征提取函数（feature extraction）；首先依次 run 这几个模块

（b）然后为统计规则模型生成 base（运行 1. base flow for RULE）

（c）用机器学习模型进行小时级别的预测（运行 2. hour-level flow prediction）

（d）训练工作日的机器学习模型进行预测（运行 3. hour-level flow prediction for weekday）

（e）说明：运行完之后会在 cache 目录下生成相应的中间预测文件。运行以上代码时，注释”validation”的代码块表示验证过程，生成最终线上预测结果时可以跳过。

## 3. 统计规则模型 model-RULE.ipynb

（1） 统计规则模型运行之前首先要保证 cache 目录下有 base_lgb.pkl 文件（机器学习模型的基础人群密度预测结果，由上一代码 model-RULE.ipynb 生成）

（2） 该代码文件分为几个部分：

a) 首先读取数据（load_data）

b) 然后进行数据处理（1. merge area_info and area_flow），把区域属性信息的 DataFrame 和区域人群密度的 DataFrame 进行 merge，得到 area_df

c) 定义三个时间段级别的增长趋势因子（2. hour-aware growth index）

d) 运行统计规则代码（3. Rule application），然后保存生成结果即可

## 4. 模型融合 Fusion.ipybn

（1） 模型融合代码运行之前保证 cache 目录下有 model-TREE.ipynb 和 model-RULE.ipybn 生成中间结果文件：

```
lishuangli 8.5K 5月   10 11:36 growth_index.pkl
lishuangli 1.3M 5月   10 11:36 pred_flow_hour_level_lgb.pkl
lishuangli 655K 5月   10 11:36 pred_flow_hour_level_xgb.pkl
lishuangli 1.7M 5月   10 11:36 pred_flow_rule.pkl
lishuangli 935K 5月   10 11:36 weekday_pred_flow_hour_level_lgb.pkl
lishuangli 468K 5月   10 11:36 weekday_pred_flow_hour_level_xgb.pkl
```

其中 growth_index.pkl 和 pred_flow_rule.pkl 分别是统计规则模型生成的增长趋势因子文件和人群密度预测结果；

pred_flow_hour_level_lgb.pkl 和 pred_flow_hour_level_xgb.pkl 分别是 lightgbmh 和 xgboost 小时级别的人群密度预测结果；

weekday_pred_flow_hour_level_lgb.pkl 和 weekday_pred_flow_hour_level_xgb.pkl

分别是 lightgbmh 和 xgboost 针对工作日的人群密度预测结果。

（2） 运行 Fusion.ipynb 之后最终的预测结果文件输出在 output 目录下的 test_submission_final.csv，提交该文件即为 B 榜 0.13971750 的结果

Ps：由于随机游走算法具有随机性，用该算法重新生成空间特征后的最终预测结果可能不完全是 B 榜的 0.13971750，但应该相近