|  |
| --- |
| **机器学习** |
| **XX案例名** |
| 主 研 人：李琦  参 研 人：李琦 |
| 审 核 人：  方 向：工业算法案例研究  版 本 号：A |
|  |

**版本更新**

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **更新描述** | **作者** |
| 2018/10/31 | A | 初稿 | 李琦 |

**选题表格**

|  |  |  |  |
| --- | --- | --- | --- |
| 时间 | 竞赛名 | 竞赛背景描述（50字以内） | 类型（分类/回归） |
| 2018/10/31 | 全球城市计算AI挑战赛 | 大赛以“地铁乘客流量预测”为赛题，参赛者可通过分析地铁站的历史刷卡数据，预测站点未来的客流量变化。 | 回归 |

目录

[1. 背景描述](#_Toc10244_WPSOffice_Level1) [4](#_Toc10244_WPSOffice_Level1)

[1.1 竞赛赛题描述](#_Toc17943_WPSOffice_Level2) [4](#_Toc17943_WPSOffice_Level2)

[1.2 评估指标描述](#_Toc11268_WPSOffice_Level2) [4](#_Toc11268_WPSOffice_Level2)

[2. 数据来源及描述性统计分析](#_Toc17943_WPSOffice_Level1) [4](#_Toc17943_WPSOffice_Level1)

[2.1 大赛数据来源](#_Toc15022_WPSOffice_Level2) [4](#_Toc15022_WPSOffice_Level2)

[2.2 数据的描述性统计](#_Toc3411_WPSOffice_Level2) [5](#_Toc3411_WPSOffice_Level2)

[2.2.1 数据基本情况描述：](#_Toc17943_WPSOffice_Level3) [5](#_Toc17943_WPSOffice_Level3)

[2.2.2 数据字段介绍：](#_Toc11268_WPSOffice_Level3) [5](#_Toc11268_WPSOffice_Level3)

[2.2.3 数据描述性统计](#_Toc15022_WPSOffice_Level3) [6](#_Toc15022_WPSOffice_Level3)

[3. 优秀算法思路](#_Toc11268_WPSOffice_Level1) [11](#_Toc11268_WPSOffice_Level1)

[3.1 方案一](#_Toc6573_WPSOffice_Level2) [11](#_Toc6573_WPSOffice_Level2)

[3.1.1 方案一数据预处理及特征工程部分方案](#_Toc3723_WPSOffice_Level3) [11](#_Toc3723_WPSOffice_Level3)

[3.1.2 方案一模型设计、建立部分方案 16](#_Toc3723_WPSOffice_Level3)

[3.1.3 方案一结果、排名等](#_Toc19075_WPSOffice_Level3) [18](#_Toc19075_WPSOffice_Level3)

[3.1.4 方案一算法流程图](#_Toc3723_WPSOffice_Level1) [18](#_Toc3723_WPSOffice_Level1)

[3.2 方案二](#_Toc10034_WPSOffice_Level2) [20](#_Toc10034_WPSOffice_Level2)

[3.2.1 方案二数据预处理及特征工程部分方案](#_Toc31419_WPSOffice_Level3) [20](#_Toc31419_WPSOffice_Level3)

[3.2.2 方案二模型设计、建立部分方案](#_Toc31419_WPSOffice_Level1) [22](#_Toc31419_WPSOffice_Level1)

[3.2.3 方案二结果、排名等](#_Toc25567_WPSOffice_Level1) [25](#_Toc25567_WPSOffice_Level1)

[3.2.4 方案二算法流程图](#_Toc20308_WPSOffice_Level1) [25](#_Toc20308_WPSOffice_Level1)

[4. 算法比较](#_Toc18696_WPSOffice_Level1) [26](#_Toc18696_WPSOffice_Level1)

[表4-1 算法比较](#_Toc26772_WPSOffice_Level2) [26](#_Toc26772_WPSOffice_Level2)

[5. 总结与展望](#_Toc11933_WPSOffice_Level1) [26](#_Toc11933_WPSOffice_Level1)

[5.1 总结](#_Toc8585_WPSOffice_Level2) [26](#_Toc8585_WPSOffice_Level2)

[5.2 建模思路](#_Toc3723_WPSOffice_Level2) [26](#_Toc3723_WPSOffice_Level2)

1. 背景描述

2019年，杭州市公安局联合阿里云智能启动首届全球城市计算AI挑战赛，本次挑战赛的题目选定为“地铁乘客流量预测”。地铁目前是城市交通出行的主要工具之一，地铁站突发人流量的增加极容易引起拥塞，引发大客流对冲，造成安全隐患。因此，地铁运营部门和公安机关亟需通过流量预测技术提前部署相应的安保策略，保障市民安全出行。

参赛者将根据主办方提供的地铁人流相关数据，挖掘隐藏在背后的出行规律，准确预测各个地铁站点未来流量的变化。主办方希望通过这次挑战赛，用大数据和人工智能等技术助力未来城市安全出行。

**1.1 竞赛赛题描述**

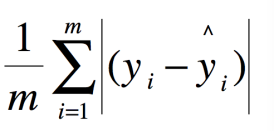
大赛以“地铁乘客流量预测”为赛题，参赛者可通过分析地铁站的历史刷卡数据，预测站点未来的客流量变化，帮助实现更合理的出行路线选择，规避交通堵塞，提前部署站点安保措施等，最终实现用大数据和人工智能等技术助力未来城市安全出行。

大赛开放了20190101至20190125共25天地铁刷卡数据记录,供选手搭建地铁站点乘客流量预测模型。同时大赛提供了路网地图，即各地铁站之间的连接关系表，存储在文件Metro\_roadMap.csv文件中供选手使用。

测试阶段，大赛将提供某天所有线路所有站点的刷卡数据记录，选手需预测未来一天00时至24时以10分钟为单位各时段各站点的进站和出站人次。

**1.2 评估指标描述**

评估指标用以评判选手对未来一天以10分钟为单位各时段各地铁站的出站和入站人次的总量预测是否准确，因此采用平均绝对误差（Mean Absolute Error，MAE）分别对入站人数和出站人数预测结果进行评估，最后再对两者取平均，得到最终评分



1. 数据来源及描述性统计分析

**2.1 大赛数据来源**

主办方：第十届中国(杭州)国际社会公共安全产品与技术博览会组委会

支持单位：杭州市公安局

这里是数据的超链接。

<https://tianchi.aliyun.com/competition/entrance/231708/information>

**2.2 数据的描述性统计**

**2.2.1 数据基本情况描述：**

大赛开放了20190101至20190125共25天地铁刷卡数据记录，共涉及3条线路81个地铁站约7000万条数据作为训练数据（Metro\_train.zip），供选手搭建地铁站点乘客流量预测模型。训练数据（Metro\_train.zip）解压后可以得到25个csv文件，每天的刷卡数据均单独存在一个csv文件中，以record为前缀。如2019年1月1日的所有线路所有站点的刷卡数据记录存储在record\_2019-01-01.csv文件中，以此类推。同时大赛提供了路网地图，即各地铁站之间的连接关系表，存储在文件Metro\_roadMap.csv文件中供选手使用。

预选赛阶段，测试集A集上，大赛将提供2019年1月28日的刷卡数据（testA\_record\_2019-01-28.csv），选手需对2019年1月29日全天各地铁站以10分钟为单位的人流量进行预测，淘汰赛和决赛将分别更新一批新的数据测试集B集和测试集C集。

**2.2.2 数据字段介绍：**

****用户刷卡数据表（record\_2019-01-xx.csv）:****

**表2-1**

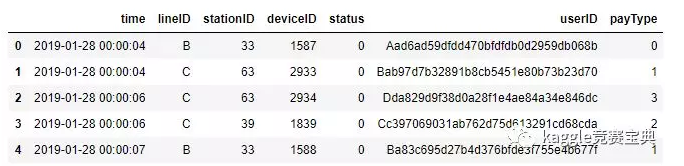
|  |  |  |  |
| --- | --- | --- | --- |
|  | 变量含义 | 变量类型 | 缺失率 |
| **time** | 刷卡发生时间 | string | 0% |
| **lineID** | 地铁线路ID | string 分类 | 0% |
| **stationID** | 地铁站ID | int 分类 | 0% |
| **deviceID** | 刷卡设备编号ID | int | 0% |
| **status** | 进出站状态  0为出站，1为进站 | int 分类 | 0% |
| **userID** | 用户身份ID | string | 0% |
| **payType** | 用户刷卡类型 | int 分类 | 0% |

注：  
1.record\_2019-01-xx.csv文件中，除第一行外，每行包含一条用户的刷卡记录。  
2.对于userID，在payType为3时无法唯一标识用户身份。即此userID可能为多人使用，但在一次进出站期间可以视为同一用户。对于其他取值的payType，对应的userID可以唯一标识一个用户。

****路网地图（Metro\_roadMap.csv）****  
大赛提供了各地铁站之间的连接关系表，相应的领接矩阵存储在roadMap.csv中，其中包含一个81x81的二维矩阵。文件中首行和首列表示地铁站ID（stationID），columns为0-80，rows为0-80。  
roadMap[i][j] = 1 表示stationID为i的地铁站和stationID为j的地铁站直接相连；  
roadMap[i][j] = 0 表示stationID为i的地铁站和stationID为j的地铁站不相连。

**2.2.3 数据描述性统计（以给的测试集1月28号的数据为例）**

**1.数据概览**

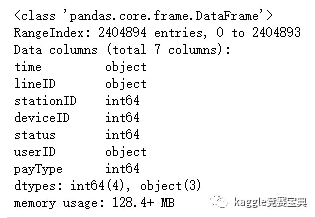


**2.数据类型&大小**

每天的记录包含7种信息,4个int64类型的数据,3个object类型的数据;

每个数据集大概有128.4M左右的大小

数据集中一共有2404894个样本



### 单变量信息探索**.**

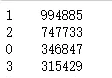
### **lineID**



一共有3条线

B线路乘坐的顾客数最多,A线路乘坐的人数最少

1. **payType**



一共有4种不同的支付方式

类型为1的支付人数最多,3是变动的人数(也就是外地游客去购买的地铁卡),从下面来看还是当地人多一些

1. **status**

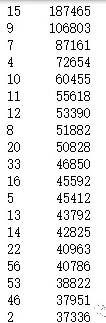


status一共有两种情况,0(出站)和1(进站)

0大于1,也就是说出站人数要大于进站的人数

推测：有些工作人员可能第二天才下班,所以需要注意这些工作人员的下班时间,也就是第二天凌晨的时间段等.这部分的预测可以单独处理.

1. **stationID**



stationID一共有80个不同的值

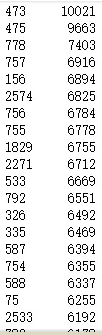
stationID为15,9,7这些站点的人数比较多,可能是一些大型商业地点.

stationID对应的人数不存在奇异的情况,例如0之类的,最少的也有7100多号人的流动。

1. **deviceID**

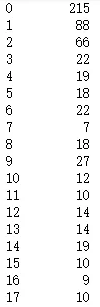
deviceID一共有1700个不同的值

deviceID为473,474,475的最多,也存在一些极少的情况,如3305,1838



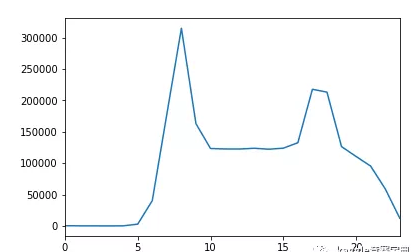
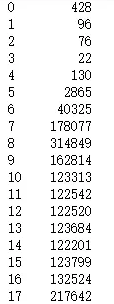
1. **time**

* 先将时间转化为10分钟为单位



发现在凌晨的时候还零零散散的有人,可能是之前忘记出站的和工作人员的流动情况

* 以小时为单位观察



发现早上7点,8点,9点的人数是最多的,因为这个时候很多人去上班了,所以算上班高峰期间.

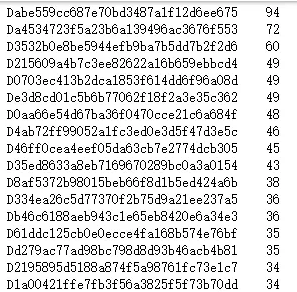
下午5、6点的时候也有一波高峰,这个时候大家都下班了,所以人数会有突变.

1. **userID**

有些userID很奇怪,有着非常高的出入频率,应该是工作人员之类的;

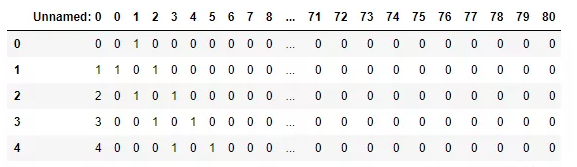
有些userID只有一次出入的情况;

还有很多userID的出现频率是1，共6269个



## **Metro\_roadMap数据集分析**

官方额外给出了地铁地图的信息,所以对此进行简单的分析

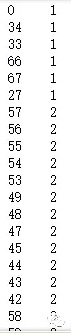


### **中转站&终点站和始发站分析**

通过计算每个站点的连接度的读数,可以发现：

10,5,51这几站有很多点,这些点应该就是几个地铁线的换乘口

0,33,45,66,67,27这六个站应该就是始发站或者终点站,因为这些站只有一个连接口



3. 优秀算法思路

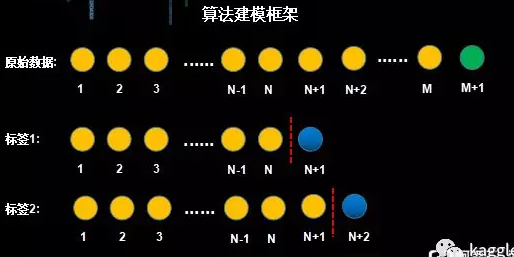
**3.1 方案一**

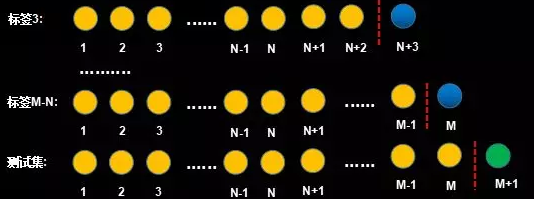
**3.1.1** 方案一数据预处理及特征工程部分方案

**1.提取特征**

数据我们采用滑窗滚动（天）的方式进行构建，这样可以防止因为某一天存在奇异值而导致模型训练走偏。最后将所有滚动滑窗的标签以及特征进行拼接形成我们最终的训练集。

滑窗的方式可以参考下图:



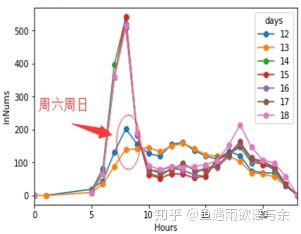
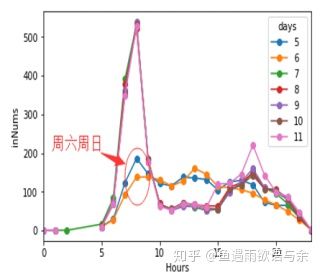


对于常见得时序问题时，都可以采样这种方式来提取特征，构建训练集。

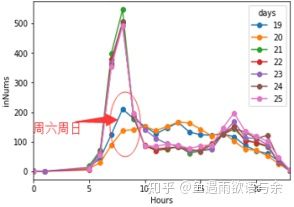
1. **特征处理**

上面滑窗滚动需要选择分布于测试集类似的进行label的构建才能取得较好的结果，所以在此之前我们需要对分布差异大的数据进行删除。

这里我们进行了简单得EDA来分析label得分布情况。（好的EDA能够帮助你理解数据，挖掘更多细节，在比赛中必不可少）

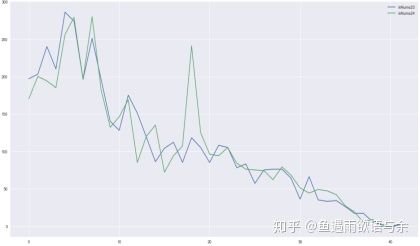


5号-10号各时刻入站流量分布 12号-18号各时刻入站流量分布



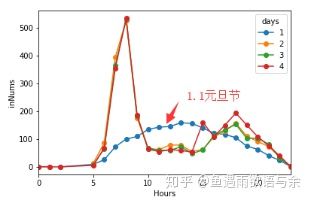
19号-25号各时刻入站流量分布

从三幅图中可以看出周末与周内分布有很大差异，所以我们将测试集为周末和测试集为周内经行区别对待，保证训练集分布的稳定。



23号和24号入站流量分布

可以看出相同时间段流量突然相差巨大。可以考虑是因为突发性活动，特别事件等因素影响。

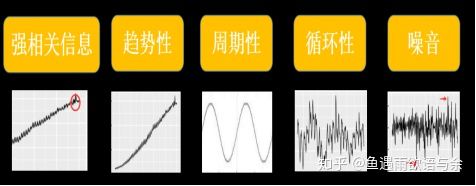


元旦节及之后几天的入站流量分布

节假日的信息和非节假日的分布差异非常大,所以我们也选择将其删除。

## **特征工程**

有了框架，下面就是如何对每个站点不同时刻的流量信息进行刻画，此处需要切身地去思考影响地铁站点流量的因素，并从能使用的数据中思考如何构造相关特征来表示该因素。最终通过大量的EDA以及分析,我们通过以下几个模块来对地铁流量的特征进行构建。



### **强相关性信息**

强相关性信息主要发生在每天对应时刻，所以我们分别构造了小时粒度和10分钟粒度的出入站流量特征。考虑到前后时间段流量的波动因素，所以又添加上个时段和下个时段，或者上两个和下两个时段的流量特征。同时还构造了前N天对应时段的流量。更进一步，考虑到相邻站点的强相关性，添加相邻两站对应时段的流量。



### **趋势性**

挖掘趋势性也是我们提取特征的关键，我们主要构造特征定义如下：

IMG_256

即表示前后时段的差值，这里可以是入站流量也可以是出战流量。同样，我们考虑了每天对应当前时段，每天对应上个时段等。当然我们也可以考虑差比：

IMG_257

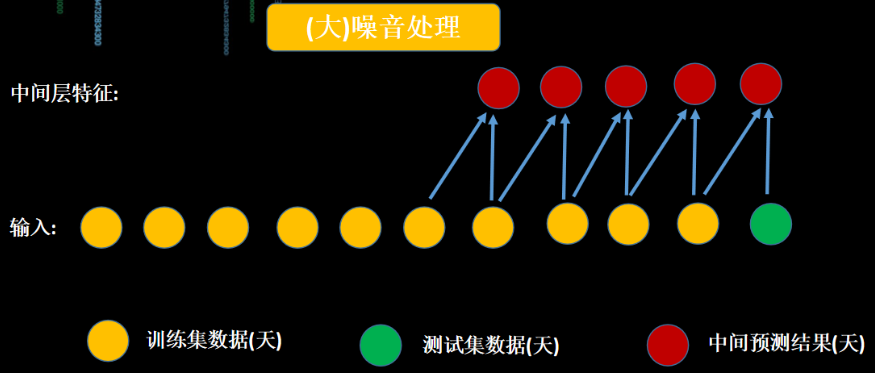


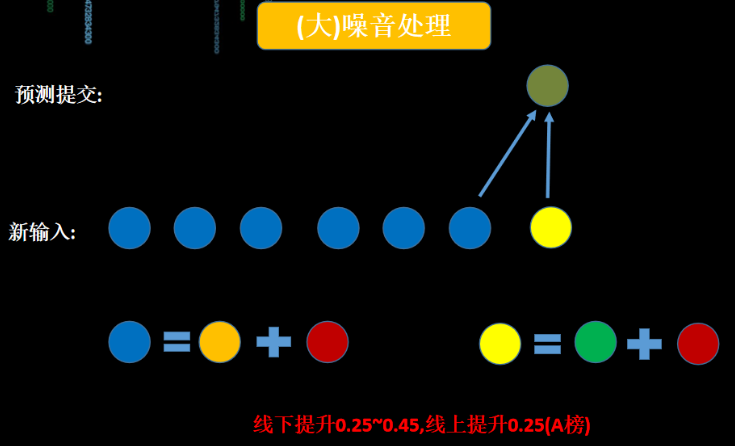
### **周期性**

由于周末分布类似，工作日分布类似。所以我们选择对应日期对应时间段的信息进行特征的构建,具体地:



### **噪声**





### **stationID相关特征**

主要来挖掘不同站点及站点与其它特征组合得热度

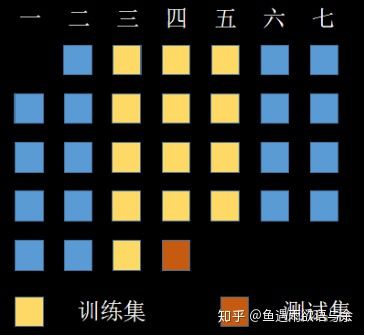
**3.1.3** 方案一模型设计、建立部分方案

## **模型训练&融合**

模型训练方面我们主要有三个方案，分别是**传统方案**、**平滑趋势**和**时序stacking**。最后将这三个方案预测的结果根据线下验证集的分数进行加权融合。

**由于C榜分数得优越性，所以此处主要阐述C榜的方案。**

**1.传统方案**



由于C榜测试集为周内数据，所以我们移除了周末数据，保证分布基本一致，为了保持训练集的周期性，我们移除了周一和周二。这也作为我们最基本的方案进行建模。

### **2.平滑趋势**

我们设计了一种处理奇异值的方法，也就是第二个方案平滑趋势。方案思想是，对于周内分布大体相同的日期，如果相同时刻流量出现异常波动，那么我们将其定义为奇异值。然后选取与测试集有强相关性的日期作为基准，比如C榜测试集为31号，那么选择24号作为基准，对比24号与其它日期的相对应时刻的站点流量情况。这里我们构造其它日期对应24号时刻流量的趋势比，根据这个趋势比去修改对应时刻中每个10分钟的流量。因为小时的流量更具稳定，所以根据小时确定趋势比，再修改小时内10分钟的流量。对流量进行修改后再进行传统方案的建模，这里我们回保留周一和周二的数据。



**具体步骤：**

* 删除周六周日
* 平滑24号之前日期对应24号的时刻流量趋势
* 常规训练

### **3.时序Stacking**

因为历史数据中存在一些未知的奇异值,例如某些大型活动会导致某些站点在某些时刻流量增加,这些数据的影响很大,为了减小此类数据的影响,我们用了时序stacking的方式进行解决,如果模型预测结果和我们的真实结果相差较大,那么此类数据就是异常的,方案的可视化如下,通过下面的操作,我们线下和线上都能得到稳定的提升。



采用rf,lgb,gbdt,mlp做训练；节省时间

### **4.模型融合**



三个方案各具优势，线下的表现的相关性也较低，经过过融合后线下的结果更加稳定，最终我们依线下CV的表现对其进行加权融合。

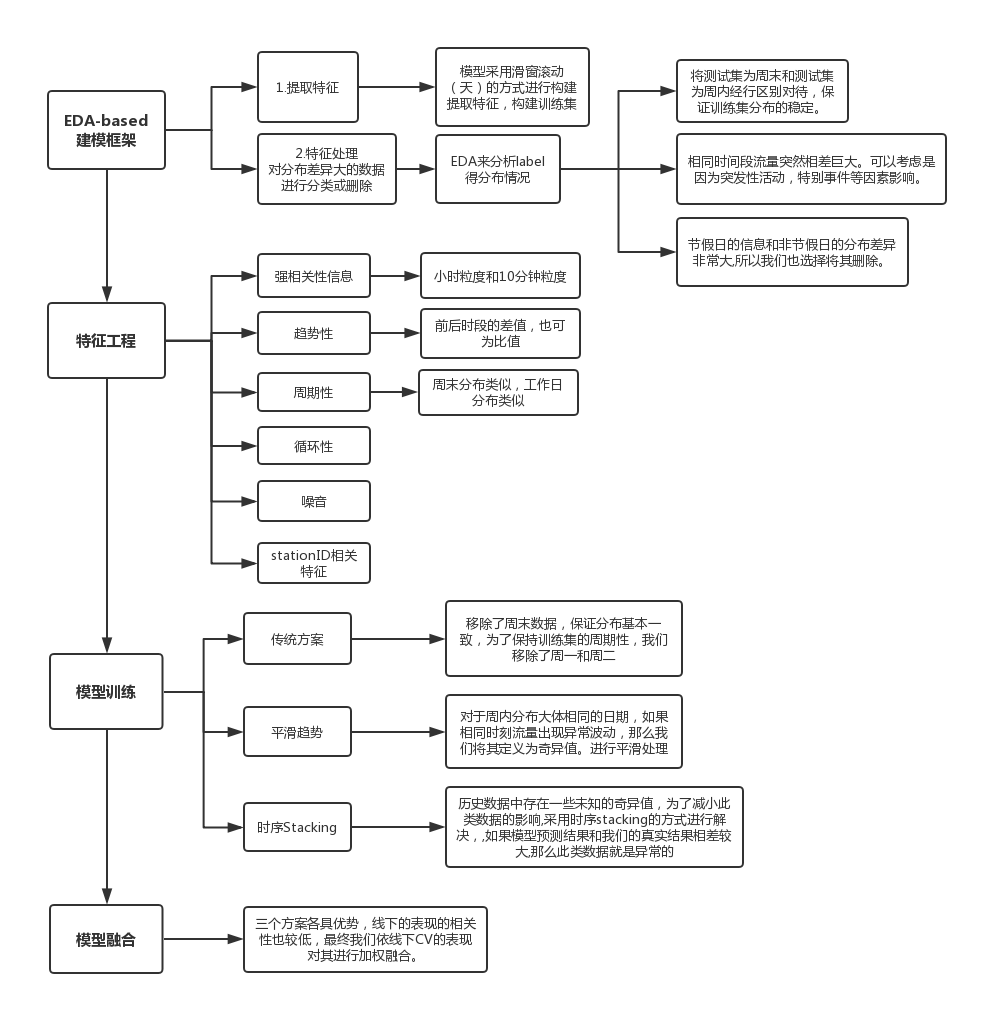
**3.1.3** 方案一结果、排名等

A榜：11.22第一

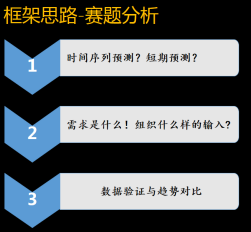
BC榜：第二

**3.1.4** 方案一算法流程图

（这个部分需要自己整理一下绘制，流程图可以用processon画）

****

**3.2 方案二**



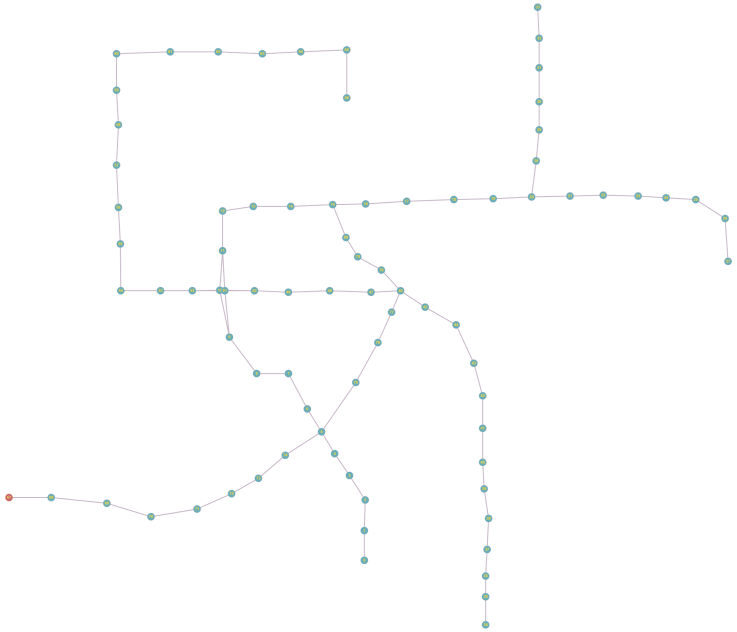
分析一下这是一个怎样的问题。从数据来看，显然这是一个关于时间序列预测的问题。但是这个时序题具有怎样的特质呢？是中短期时间序列？还是长期时间序列？

通过提交文件形式可以发现，每一天需要预测144个时间段，且总共只有25天作为训练数据，数据规律性以天呈现周期时间变化，可以认为这是一个中短期时序的短期预测问题。

由于赛题测试集中只包含待预测日期前一天的数据记录，因此输入数据的形式（或结构）并不是可以自由组织，根据现有的测试集进行输入调整可能是一种比较好的选择。当测试集分别为工作日和周末时，线下验证应该保持相应的结构，具体的验证方式也应有所不同，我们可以尝试将预测结果与原始趋势进行对比，确保预测结果的趋势可靠性。

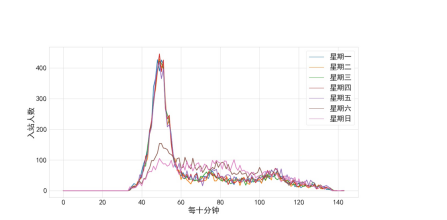
**3.2.1** 方案二数据预处理及特征工程部分方案

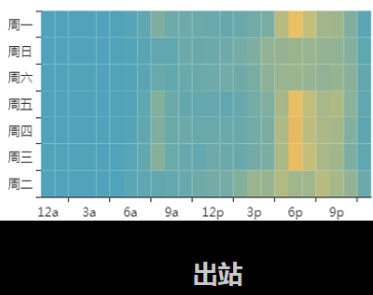
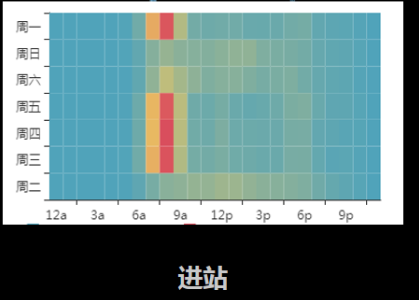
根据roadMap可以得到部分地铁站台分布示意图。通过与网上实际地铁运行线路对比，可以找到数据中地铁ID的一一对应关系，这部分信息确认可以为后面的模型建立提供指导。



**时间维度分析**

2019年一月第二周（1.14~1.20）每日10分钟入站人数对比

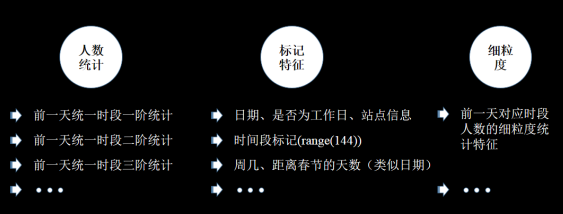




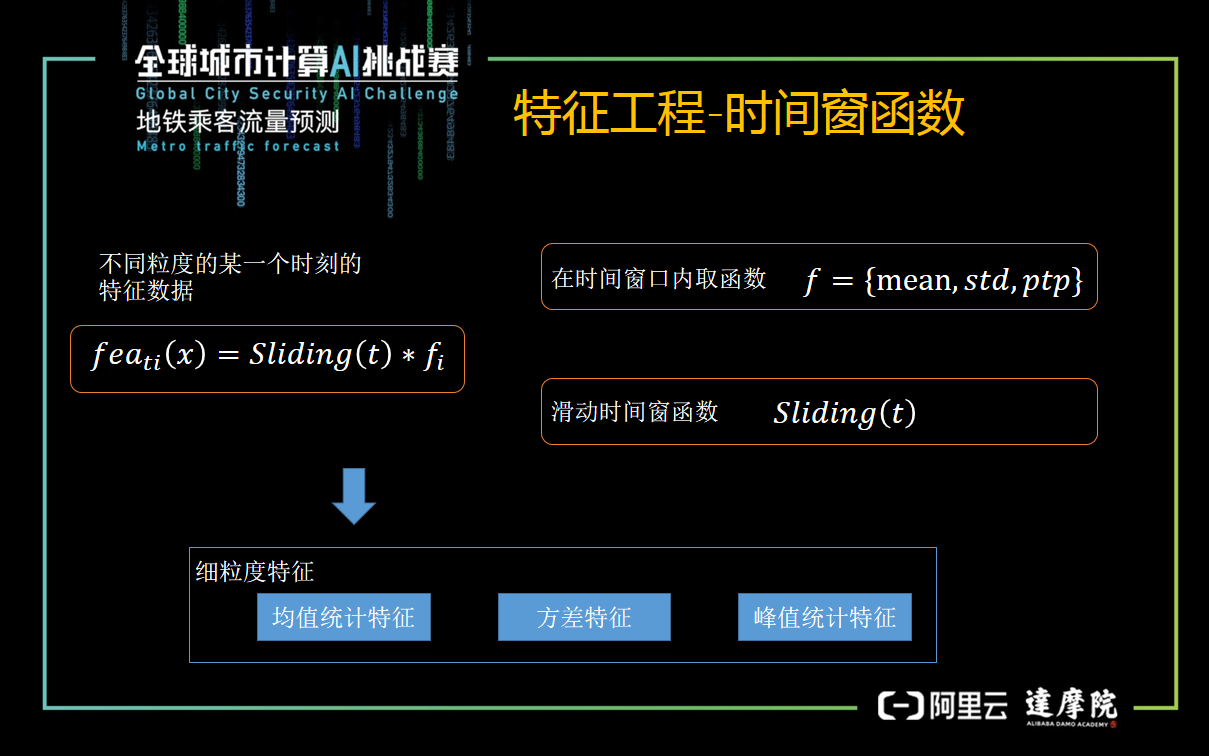
* 进站、出站的峰值时间不同，进站峰值在8点附近，出站峰值在6点附近
* 周六周日明显与工作日进出站人数不同

**特征工程**

1. 基础特征



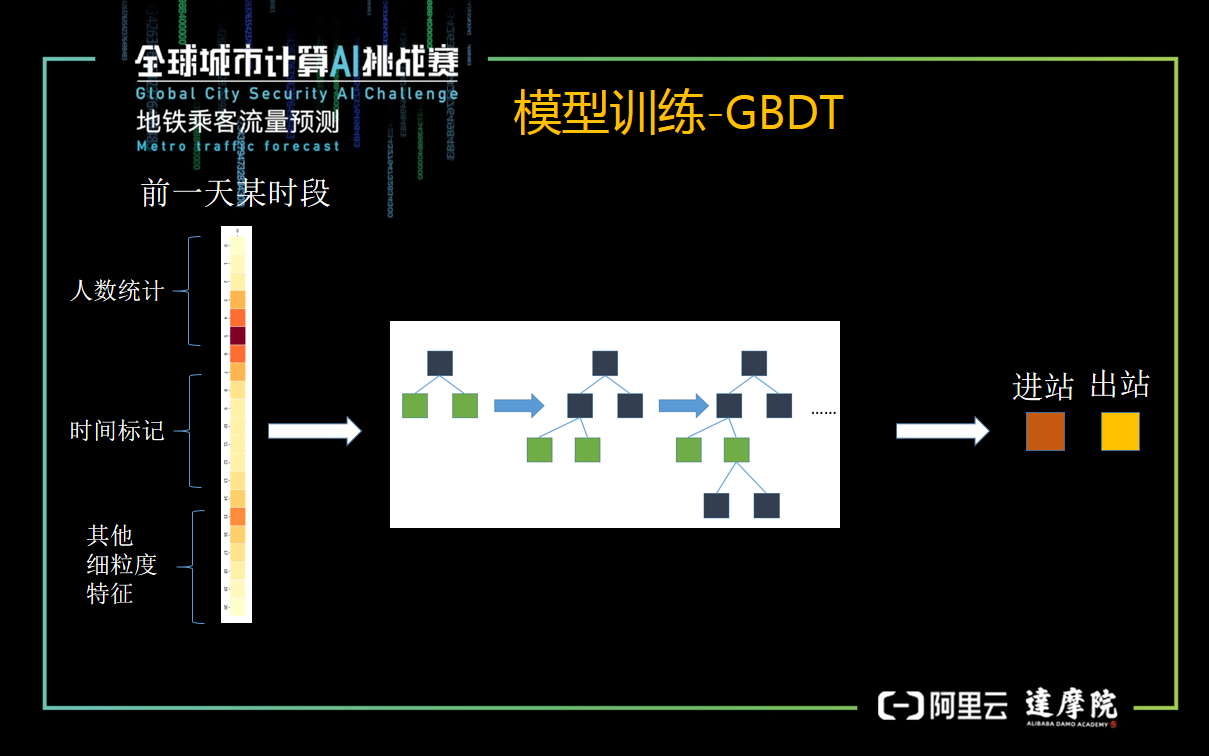
**2.时间窗函数**



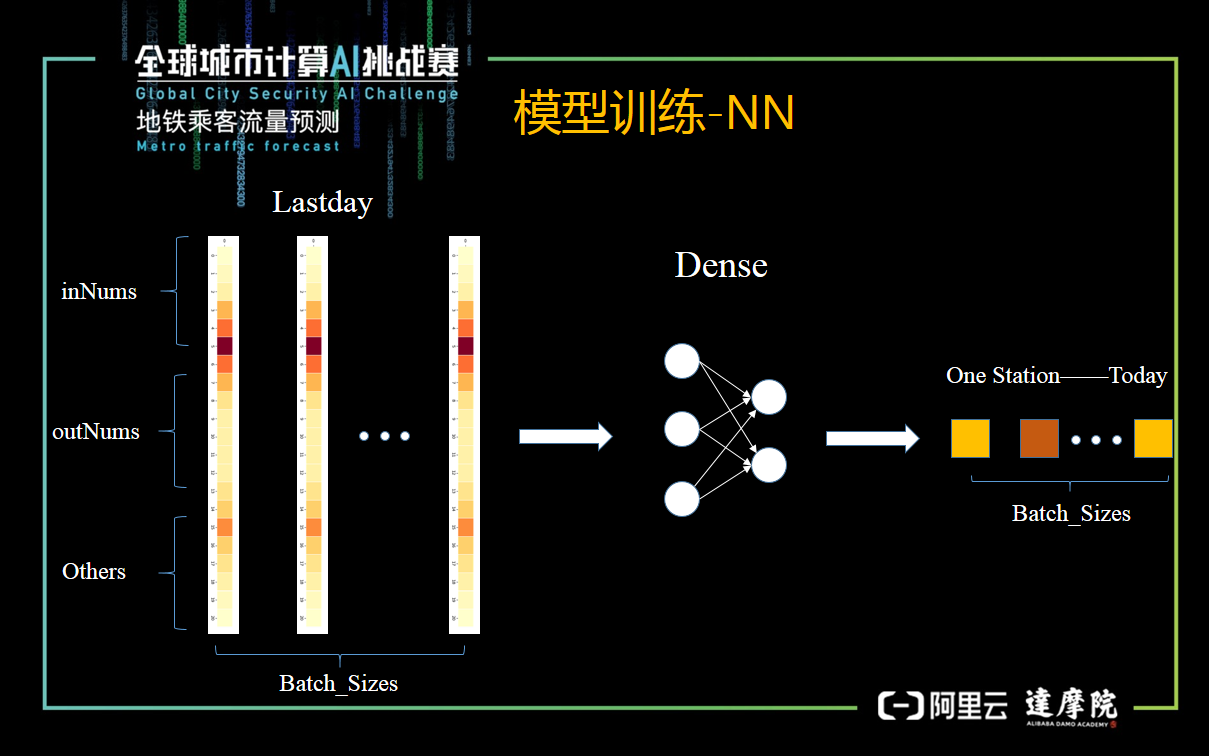
**3.2.2** 方案二模型设计、建立部分方案

**模型训练**

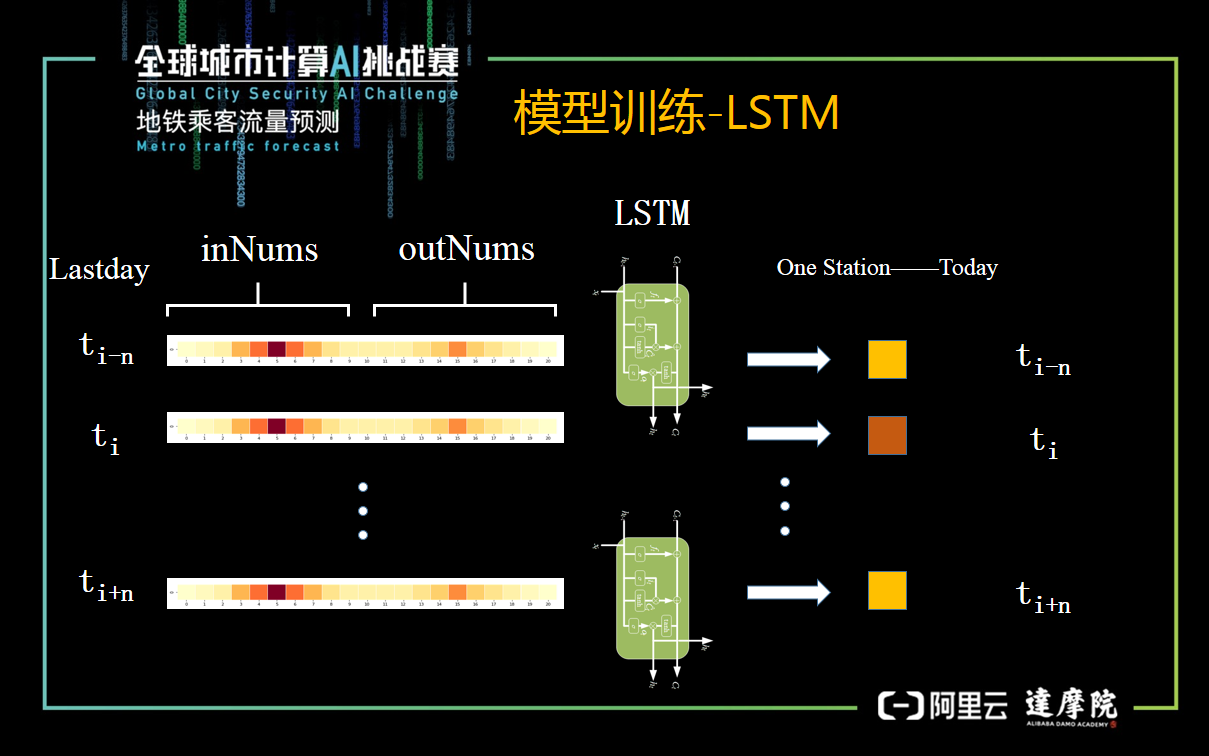
**1.GBDT**



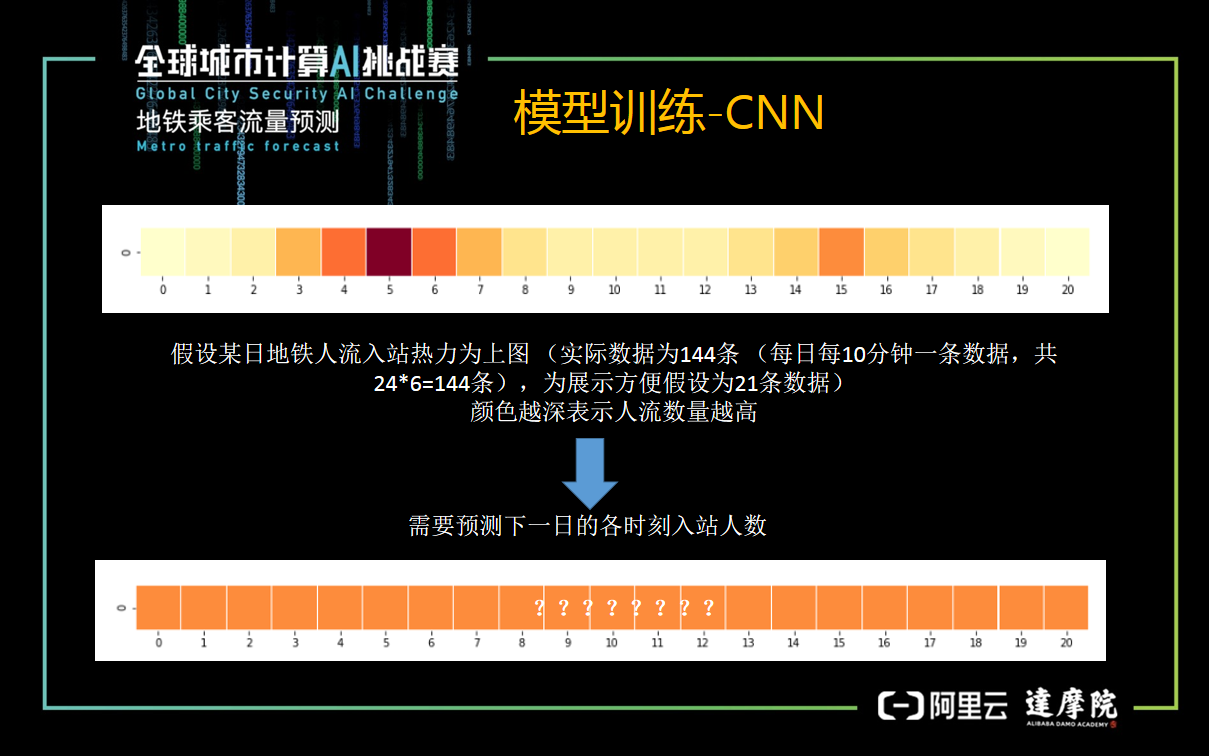
**2.NN**

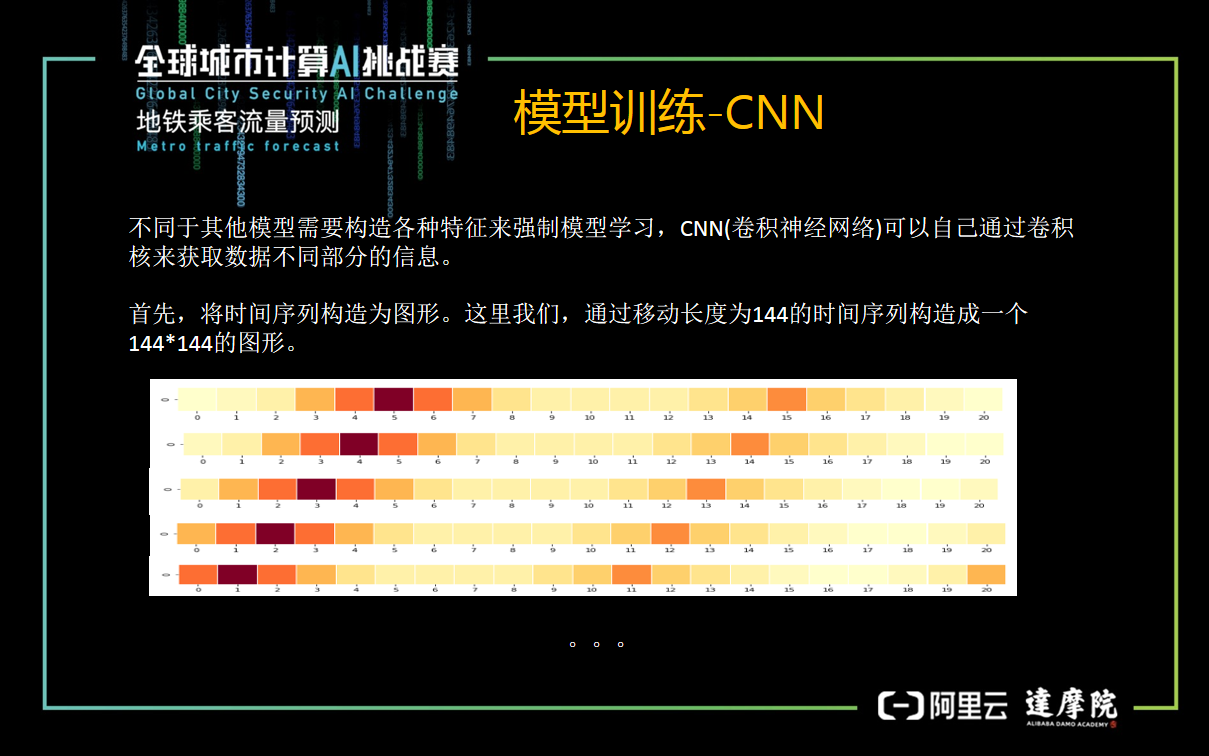


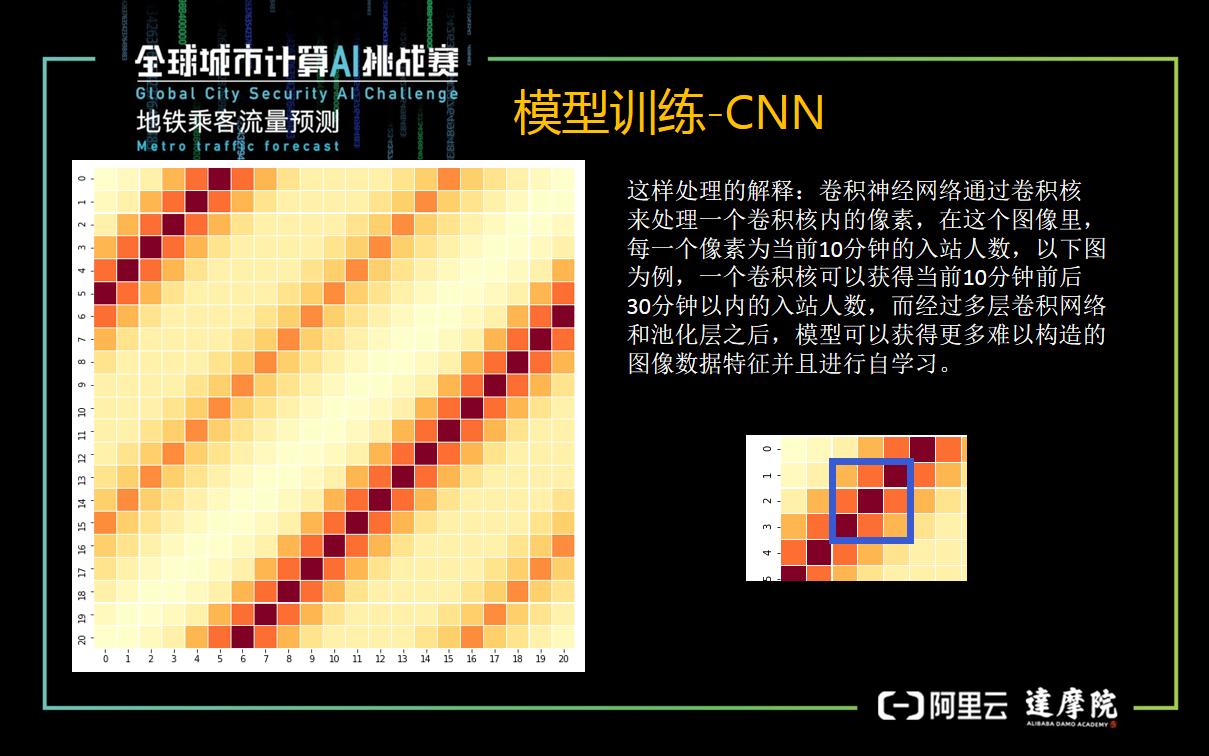
**3.LSTM**



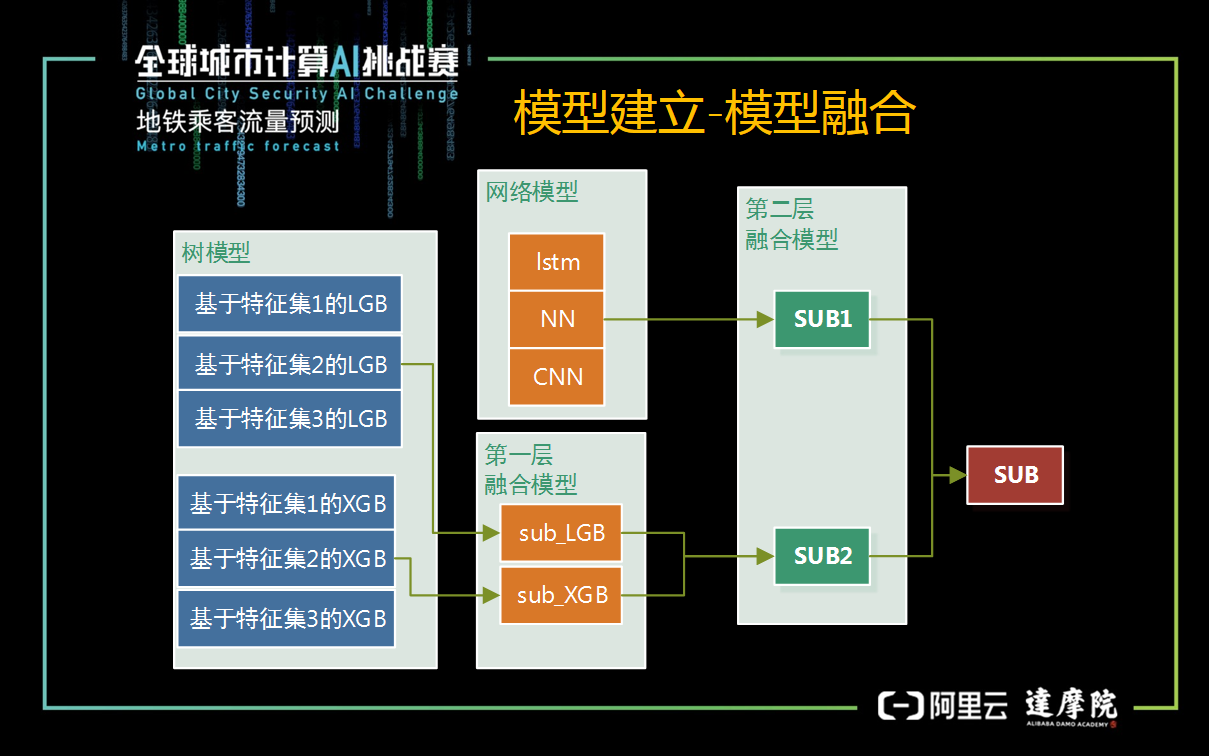
**4.CNN**







**模型融合**

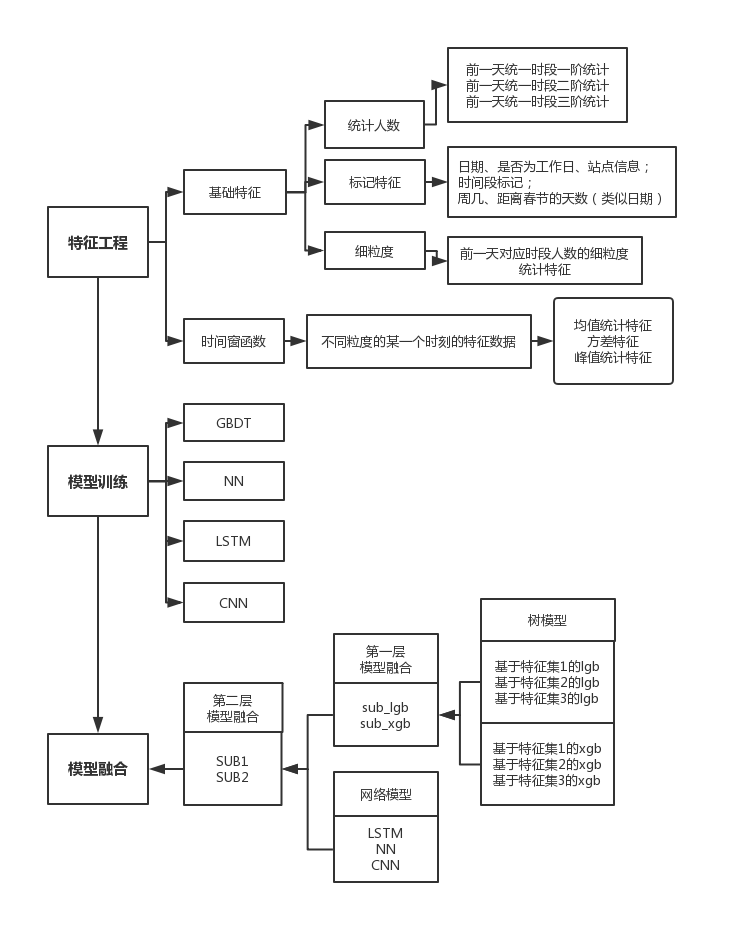


**3.2.3** 方案二结果、排名等

结果：11.44

排名：第三

**3.2.4** 方案二算法流程图



4. 算法比较

**表4-1 算法比较**

|  | **评估指标**  MAE | **特征工程** | **基础算法** | **基本库** |
| --- | --- | --- | --- | --- |
| **算法1** | 11.22 | 强相关性信息；趋势性；周期性；循环性；噪音；stationID相关特征 | rf,lgb,gbdt,mlp | Pandas,sklearn, Keras,[Tensorflow](http://www.baidu.com/link?url=Oi_andISViwcMG3l4Yee30scMO9CdZSTnFMKYRCnbbSgRXgKliMeDk6pqKvreDF8neL1i5iXXLzQvyif8F2ns0sQfJTmXpJQSYIDSzuSyfC" \t "https://www.baidu.com/_blank) |
| **算法2** | 11.44 | 统计人数；标记特征；细粒度；不同粒度的某一个时刻的特征数据 | GBDT，NN,LSTM,CNN | Pandas,sklearn, Keras,[Tensorflow](http://www.baidu.com/link?url=Oi_andISViwcMG3l4Yee30scMO9CdZSTnFMKYRCnbbSgRXgKliMeDk6pqKvreDF8neL1i5iXXLzQvyif8F2ns0sQfJTmXpJQSYIDSzuSyfC" \t "https://www.baidu.com/_blank) |

5. 总结与展望

**5.1 总结**

**本次竞赛的难点：**

1. 自己构建训练集

比赛给了25天的进出站刷卡的情况。参赛者需要将每天的情况进行统计，变成每个站点，每十分钟的进出站的人数。并且将这个根据统计出的时间序列，进行分析分解，构建特征。

1. 如何将各站点间的联系加以运用

比赛提供了各个站点间的连接情况，要想办法如何将这个信息分析并加到特征中。

**5.2 建模思路**

特征工程：

1. 删除第一天节日的情况
2. 将周末与工作日分开
3. 添加特征：上周同一天，前一天

建模：

lr,rf,lgb,gbdt

加权平均

后处理：

将凌晨基本没人的时间段改为0