**广东航空大数据创新大赛-GoAhead赛后总结**

1. **赛题背景**

本赛题提供了连接WIFI AP的人数表、安检旅客过关人数表、旅客进入－离开机场的行程表、航班排班表、机场登机口区域表、机场WIFI接入点坐标表共6张表。其中对于每张表的分析如下：

**连接WIFI AP的人数表：**提供了从9月10号到11月10号每隔一分钟的wifi AP记录到的人流量变化，通过观察同一wifi点每天人流量的变化可知，每天人流量的大体的趋势是基本一致的，所以这张表对我们预测未来的人流量有很大的帮助。

**安检旅客过关人数表：**提供了从9月10号到11月10号旅客的安检时间和航班号，从这一张表可以挖掘出不同航班对应旅客的大概人数，这张表结合航班排班表可以有效预测未来人流量的变化。

**旅客进入－离开机场的行程表：**提供了从9月10号到11月10号旅客的航班号、航班起飞时间、旅客出票时间，这张表可以统计出旅客出票时间和起飞时间的差异，得到提前到达机场的时间长短和旅客量之间的分布。

**航班排班表：**提供了从9月10号到11月12号的飞机排班情况，飞机排班的差异会带来旅客流量的大幅度变化，这张表非常重要。

机场登机口区域表、机场WIFI接入点坐标表：这两张表提供了登机口和wifi点的位置，wifi点越靠近登机口，受到登机口人流量的影响越大，这两张表为我们建模提供了必要的基石。

1. **赛题解读**

**目的：**预测未来两天(11.11~11.12)机场人流的分布密度

**已知：**白云机场每天百万级别的数据记录(9.10~11.10)

**评估指标 ：**

P:选手提交的旅客流量预测值，Q:机场WiFi AP实测旅客流量

本赛题把选手的预测值和真实值误差的平方和作为最终的误差函数，也就是说预测值和实际值差距越大，误差函数越大，评估指标越差。所以，我们很容易想到，人流量大的wifi点、人流量波动剧烈的wifi点，我们预测的误差会比较大，所以这些wifi点的预测对我们的目标函数起决定性作用，同时这些wifi点的位置也是有规律的，它们中有很大一部分都在登机口附近。这种地理因素显然会对人流量有较大的影响，所以，这是我们建模的重点。对于那种人流量很大同时又距离登机口较远的wifi点，我们在后期也进行了一些优化。

图1中左图是白云机场的地图，右图则是我们要预测的wifi点的位置，可以看出待预测的wifi点遍布机场的第一、二、三层，而且都在T1、T2、T3以及东西的E1~3和W1~3几块区域上。

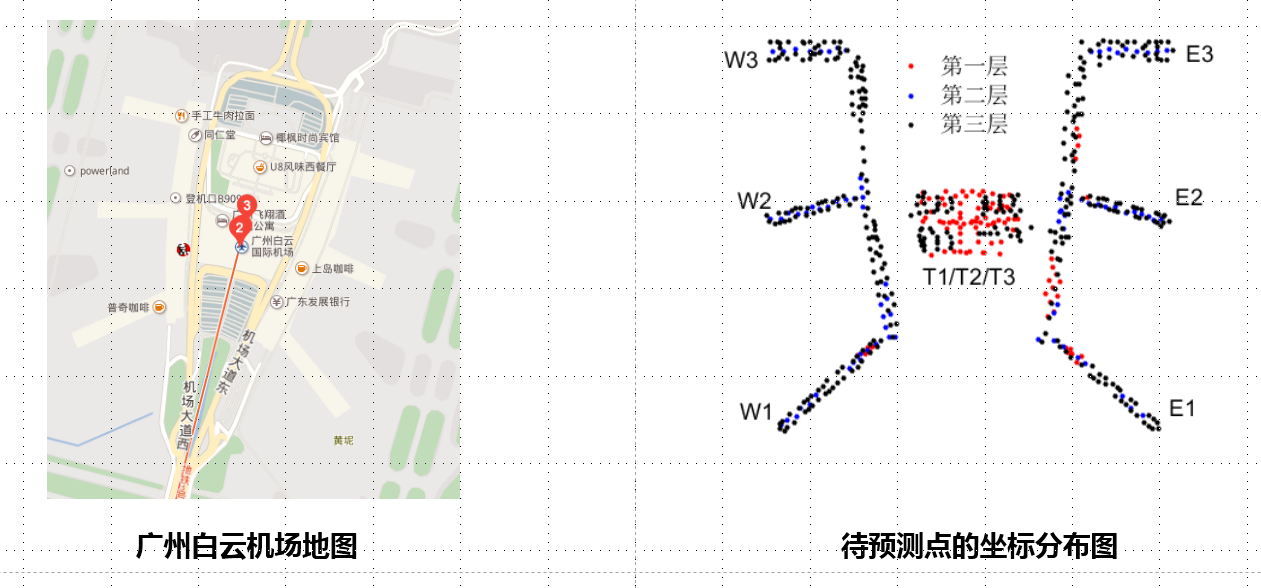


图1：左图是白云机场的地图，右图则是我们要预测的wifi点的位置，可以看出它们遍布整个机场的绝大部分区域，得到所有点的预测就可以通过Kriging 方法得到覆盖机场的‘面’的人流量的预测。

通过上面的一些分析，我们比赛的**整体思路框架**如下所示：



图2：初步分析后得到的比赛的整体思路。

原始数据经过预处理后，首先通过地理位置进行分类，由于赛题出题方已经给所有wifi点注明了楼层和所在机场区域，所以我们对于数据的分类是一项比较容易的工作。接下来，要预测长达两天的机场人流量的变化情况，我们有三种思路：

1. 使用时间序列分析，该方法基于随机过程理论和数理统计学方法，研究随机数据序列所遵从的统计规律，能够有效解决时间序列预测相关问题。但是考虑到这种方法在平台上实现比较复杂，我们只是在初赛中有使用到。
2. 使用回归模型，由于平台上只提供了GBDT和LR，而GBDT有良好的非线性预测能力，我们面临的问题也是一个非线性问题，所以我们直接选择了GBDT作为我们的回归模型。LR可以使用在后面的融合模型中。
3. 规则模型则主要是均值模型，考虑到每个wifi点每天的人流量的大致趋势是一致的，所以历史均值也是一个良好的估计值。

最后，对预测的值使用stacking的方式进行融合，即多个模型的值用线性（LR）/非线性（GBDT）模型估计出最优权重进而得到最终预测值。

1. **核心思路**

接下来介绍一下我们最终使用的算法模型：

**1.数据预处理**

首先，我们对数据进行了预处理，预处理包括数据筛选、数据清洗和训练集和测试集划分三个部分：

数据筛选：我们选择了连接wifi人数表、航班排班表、机场登机口区域表、机场wifi接入点坐标表这四张表来构造模型

数据清洗：主要是缺失值填充和去除脏数据。

训练集和测试集划分：由于要预测的是两天的数据，所以我们选择的线下训练集是9.11到11.08号，测试集选取的则是11.09到11.10号。

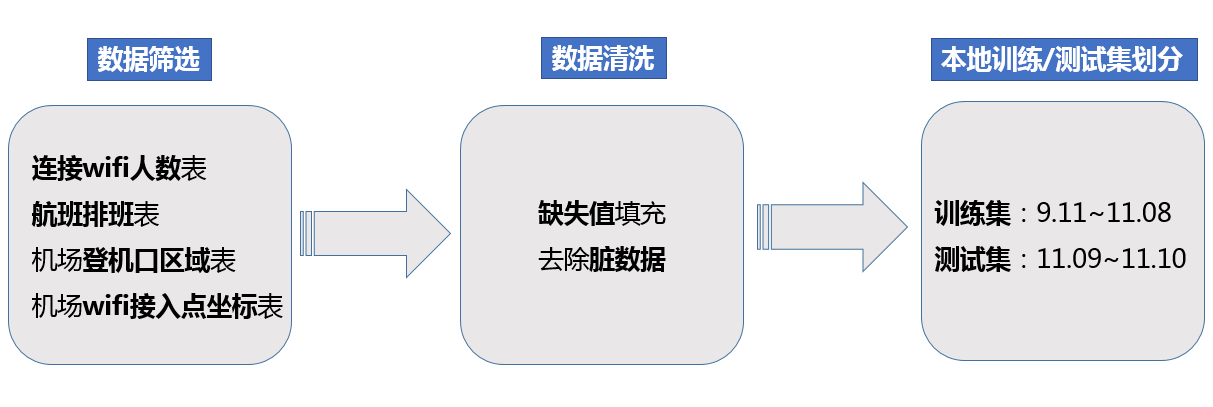


图3：数据预处理流程图。

**2.算法流程**

下图则是我们**算法模型的流程图**，我们首先对wifi点进行分类，由于wifi点的命名中已经蕴含了充分的地理信息，所以这种分类是非常简单的，如果待预测的wifi点靠近登机口，那么我们就使用第一类模型进行回归预测，考虑到待预测的wifi点的人流量变化主要受到附近登机口航班安排的影响，所以在第一类模型中，我们把登机口区域分为10类，分别是E1-1区、E1-3区、E2-3区等，这些区域具有不同数量的登机口，我们对这些区域分别提取特征、建立模型，由于这些模型具有类似的性质，所以我统称为第一类模型。wifi点如果远离登机口也就是处于登机走廊或者T1、T2、T3航站楼，考虑到航站楼的3层登机口比较多，T3区域必然会有一些wifi点也受到登机口的影响，为了筛选出这些登机口，我们在线下的测试集使用第二类和第三类模型进行预测，如果第二类模型效果优于第三类模型，那么我们就使用第二类模型进行回归预测，反之我们使用第三类模型，其它的点同样使用第三类模型进行预测。这里提出第二类模型主要是因为我们假设T3区域的点受到3楼所有登机口的影响，但是这些登机口太多了，所以我们不能按照第一类模型的方式来处理这个问题。

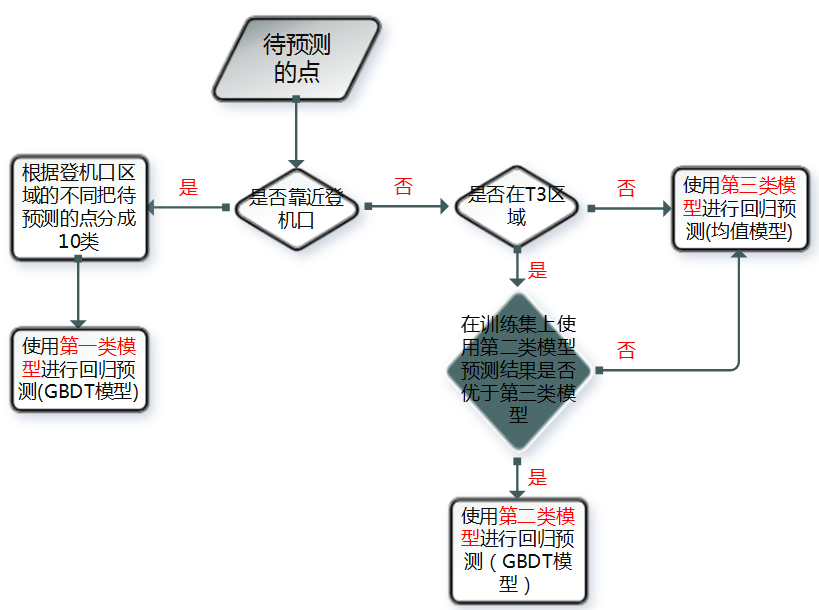


图4：算法模型的流程图

**3.第一类模型：**

第一类模型应用于靠近登机口的那些wifi点。

第一类模型包括特征提取还有模型选择两部分，特征提取是一个比较重要的问题，我们分析到飞机的航班安排会较大地影响到wifi点人流量。统计得到的航班晚点时间分布见图5，可以很明显地看出来晚点基本处于排班时间的后3个小时和前一个小时。

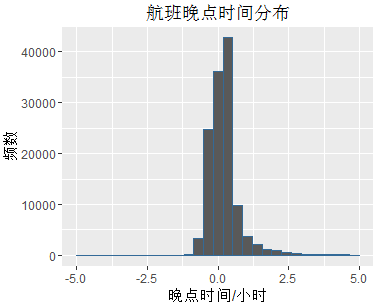


图5：航班晚点时间分布图

所以，我们使用滑动时间窗的办法：利用航班排班表，提取每个登机口的待预测时间点前60分钟——后180分钟（每隔5分钟）起飞的飞机数量，即每个登机口提取48维特征。另外，我们也提取了时间信息作为我们的特征，这也是非常重要的，比如乘客在凌晨2、3点比较困，连接wifi的概率很可能降低。

考虑到平台提供的非线性模型只有GBDT模型，所以我们选取了GBDT模型，并在测试集上进行参数调优，选取的最优参数如下：

* 学习率：0.05
* 树的数目：500
* 训练采集特征比例：1
* 训练采集样本比例：0.6
* 树最大深度：3
* 叶节点最小样本树20

图6是我们使用第一类模型的预测示例，三条线分别代表均值、第一类模型的预测结果以及实际值，可以看出模型1相对于均值的预测时有明显的提升的。当然，这里需要指出一点，在我们的线下测试集中，绝大部分wifi点都有不小的提升，整体来看的话，相对于均值预测，线下我们的结果提升了9w+，线上则是6w+。

同时，我们在这个图中可以发现另一个问题，在预测的第一天前一段时间，预测值和真实值是有明显差异的，但是由于真实值是连续的，也就是11月11号0点0分的预测值必然和11月10号23点59分的数据高度相关，所以显然可以通过对这一部分进行额外处理来提高我们的模型，这一处理的方法见第三类模型的融合方法部分。

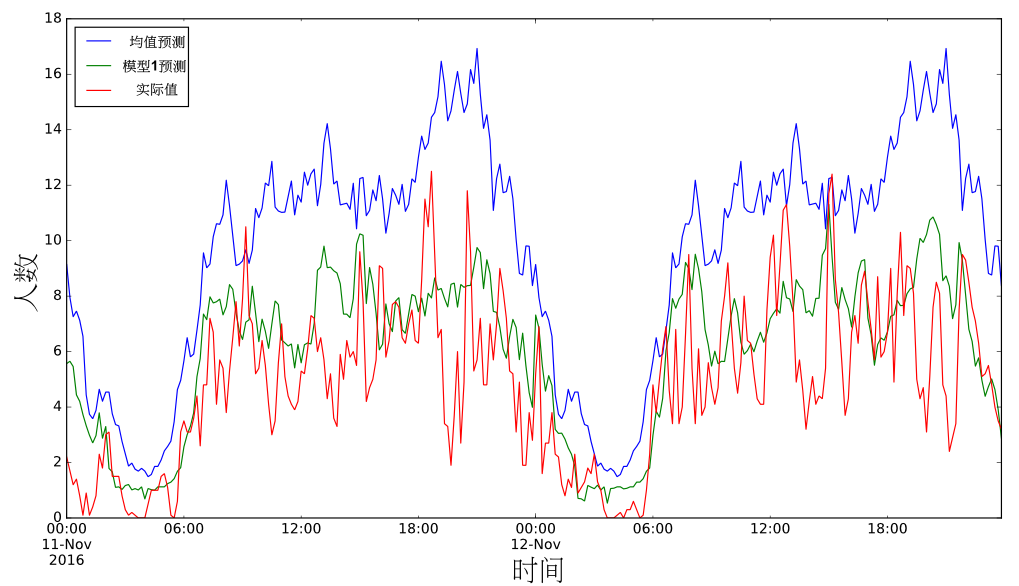


图6：使用第一类模型的预测示例，模型1相对于均值的预测时有明显的提升

**4.第二类模型：**

第二类模型与第一类模型比较相似，考虑到T3区域受到众多登机口的影响，所以我们直接使用航班排班表提取6个大的登机区域（E1-3/W1-3）的待预测时间点前60分钟——后180分钟（每隔5分钟）起飞的飞机数量，即每个大的登机区域提取48维特征。

同样，提取时间信息，共2维特征（第n天，第m个时间段）。

我们最后一天对T3区域的小部分wifi点应用了这类模型，线下提升了2w左右，线上则是5k左右。说明这类模型同样是有效的。

**5.第三类模型：**

使用均值模型，取待预测时间点对应的过去12天历史值的均值，并经过以下处理得到最终预测值：

* 无相移滤波：线下调参得到滤波系数：

a=[1.0, 0.3249197],b=[0.33754015, 0.33754015];

* 使用融合模型预测第一天的前100min的人流量:

pred1\*0.22+pred2\*0.78;

pred1:最接近的历史值, pred2:均值模型得到的预测值;

加权系数通过线下调参后得到。

均值模型选12天的原因: 选择历史均值的天数并不是越多越好，因为机场排班的变化、机场晚点等因素都会带来较大的影响，但是太少也不行, 假设每天机场人流量由一种统一的模型和噪声构成，噪声服从标准高斯分布，那么N（N>1）天平均后，噪声服从（0，σ2/N）分布，通过一些计算就可以知道N天数据的平均估计值要比一天的好，而且是N越大越好。我们通过线下线上的分析发现12天左右的历史均值一直是比较稳定而良好的估计值。

第三类模型的预测结果见图7，相对于纯粹的均值，使用滤波方法得到的结果是比较平缓的，相当于滤掉了信号中的一些高频成分，减少了可能存在的离异值对结果的影响。需要说明的是这种提升在个别wifi点是不显著的，不过整体而言，也就是对所有的wifi点进行上述滤波处理后，线上、线下都整体提升了大概2w+~3w+。

使用融合方法后线上、线下整体的提升上在5k~7k。

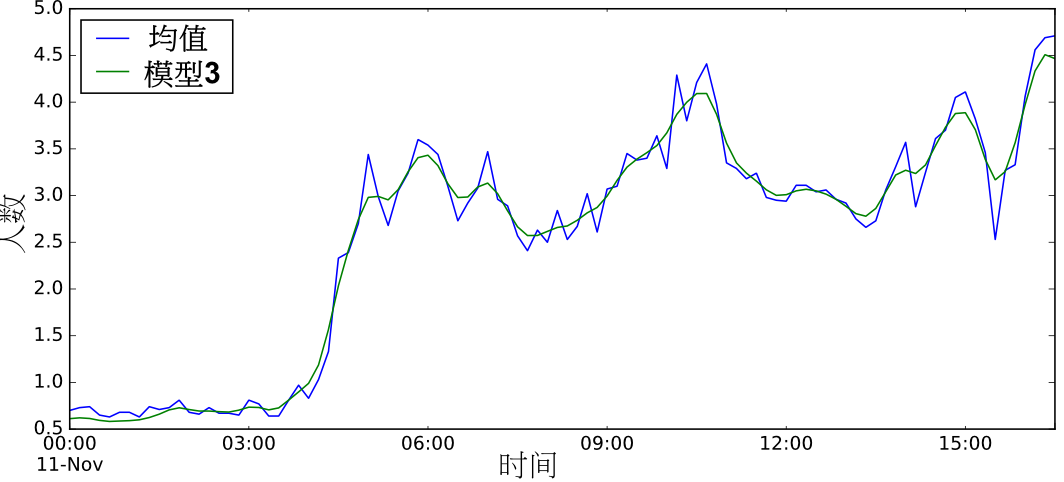


图7：第三类模型的预测结果，相对于直接均值的预测，模型三得到的结果更加平缓，减少了可能存在的离异值对结果的影响。

**四、关键代码：**

使用滑动时间窗进行特征提取：

for (int day\_ct = 0; day\_ct < 63; day\_ct++)

{

for (int i = 0; i < 144; i++)

{

StringBuffer sBuffer = new StringBuffer();

for (int j = 0; j < depart\_minute\_diff.size(); j++) //离开的飞机，每5分钟做一次统计

{

int[] features = new int[48];

ArrayList<Integer> temparr = depart\_minute\_diff.get(j);

for (int k = 0; k < temparr.size(); k++)

{

int temp = temparr.get(k)+480-i\*10-day\_ct\*1440;

if (temp>=0 && temp<180)

{

features[temp/5]++;

}

if (temp<0 && temp>-60)

{

temp = -temp;

features[temp/5+36]++;

}

}

for (int k = 0; k < features.length; k++)

{

sBuffer.append(String.valueOf(features[k])+",");

}

}

sBuffer.deleteCharAt(sBuffer.length()-1);

forward(Long.valueOf(day\_ct+1),Long.valueOf(i+1),sBuffer.toString());

}

}

无相移滤波：

ArrayList<Double> y = Filtfilt.doFiltfilt(B,A,to\_list(predict));//滤波，B、A是滤波参数，滤波器filtfilt

融合模型：

if((!Double.isNaN(past\_value)) && ispast)

{

for (int i = 0; i < 10; i++)

{

predict[i] = 0.22\*past\_value+0.78\*predict[i];

}

}

GBDT模型（SQL）:

PAI -name GBDT -project algo\_public -DfeatureSplitValueMaxSize="500" -DlossType="3" -DrandSeed="0" -DnewtonStep="1" -Dshrinkage="0.05" -DmaxLeafCount="32" -DlabelColName="target" -DinputTableName="e1\_1\_train" -DminLeafSampleCount="20" -DsampleRatio="0.6" -DmaxDepth="3" -DmetricType="0" -DmodelName="e1\_1\_model" -DfeatureRatio="1" -Dp="1" -Dtau="0.6" -DtestRatio="0" -DfeatureColNames="day,idx,f1,f2,f3,f4,f5,f6,f7,f8,f9,f10,f11,f12,f13,f14,f15,f16,f17,f18,f19,f20,f21,f22,f23,f24,f25,f26,f27,f28,f29,f30,f31,f32,f33,f34,f35,f36,f37,f38,f39,f40,f41,f42,f43,f44,f45,f46,f47,f48,f49,f50,f51,f52,f53,f54,f55,f56,f57,f58,f59,f60,f61,f62,f63,f64,f65,f66,f67,f68,f69,f70,f71,f72,f73,f74,f75,f76,f77,f78,f79,f80,f81,f82,f83,f84,f85,f86,f87,f88,f89,f90,f91,f92,f93,f94,f95,f96,f97,f98,f99,f100,f101,f102,f103,f104,f105,f106,f107,f108,f109,f110,f111,f112,f113,f114,f115,f116,f117,f118,f119,f120,f121,f122,f123,f124,f125,f126,f127,f128,f129,f130,f131,f132,f133,f134,f135,f136,f137,f138,f139,f140,f141,f142,f143,f144,f145,f146,f147,f148,f149,f150,f151,f152,f153,f154,f155,f156,f157,f158,f159,f160,f161,f162,f163,f164,f165,f166,f167,f168,f169,f170,f171,f172,f173,f174,f175,f176,f177,f178,f179,f180,f181,f182,f183,f184,f185,f186,f187,f188,f189,f190,f191,f192" -DtreeCount="500";

**五、总结：**

我们的总结包括模型总结和经验总结两部分：

1. **模型总结：**

算法优势：

* + - * 先分类后回归
      * 规则模型使用滤波方法平滑数据，使用融合方法进一步提升结果
      * 回归模型简单、有效
      * 泛化能力优秀

后续提升的思路：

在我们上述的模型中，其实还有一些问题我们没有足够的时间来解决，如果时间充分、平台运行流畅的话应该还会有不小幅度的提升。

* 进一步优化模型参数
* GBDT对异常值比较敏感，航班晚点容易造成异常值，利用二者关系可以进一步优化模型
* 考虑其它类型的滤波器（有很多滤波器实现复杂所以并未考虑）
* 对更多的wifi点构造模型
* 模型融合(stacking)

1. **经验总结：**

比赛心得：

* + - * 仔细分析数据，不可盲目堆特征、堆模型；
      * 充分利用线上评测的机会；
      * 坚持就是胜利：Just Go Ahead and you will be rewarded.