# 智慧交通预测挑战赛

## 比赛题目

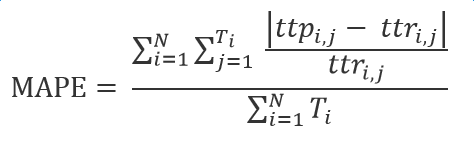
基于已知某路段平均旅行时间对未来该路段平均旅行时间进行预测

## 比赛数据

共提供了2016年与2017年3、4、5、6月份与2017年7月份平均旅行时间作为训练集，2017年7月份旅行时间作为测试集。各个路段的长宽属性。各个路段的上下游关系。

## 评价指标

采用MAPE评价指标



## 数据分析

并没有对数据进行宏观分析。

## 数据预处理

1. 采用3δ准则对数据进行降噪。在日期维度上与时间维度上对数据进行处理，对超过avg+3δ或小于avg-3δ的数据进行修补。
2. 采用了数据平滑的方法。在日期维度上与时间维度上对数据进行处理，创建滑动窗口对数据进行平滑，平滑后的数据比平滑前更具有稳健性。
3. 缺失值填补。很多个路段存在数据不全的时段，对这些时段使用模型进行预测时容易面临特征不全的问题，采用直接用历史时段数据中位数拟合的方法。

## 特征提取

1. **日期维度：**
2. 历史同时段旅行时间中位数。
3. 历史同时段旅行时间均值。
4. 历史同时段旅行时间最大值。
5. 历史同时段旅行时间最小值。
6. 历史同时段旅行时间标准差。
7. 历史同时段旅行时间中位数在所有时段旅行时间中位数中的排名。
8. 历史同时段旅行时间中位数在所有时段旅行时间中位数中的排名占比。
9. 历史同时段旅行时间均值在所有时段旅行时间中位数中的排名。
10. 历史同时段旅行时间均值在所有时段旅行时间中位数中的排名占比。
11. 历史同时段旅行时间中位数与所有时段旅行时间中位数的均值的差值。
12. 历史同时段旅行时间均值与所有旅行时间均值的均值的差值。
13. 通过平滑后数据提取的以上特征。
14. **时间维度**
15. 测试集目标时间前一小时平均旅行时间的中位数。
16. 测试集目标时间前一小时平均旅行时间的均值。
17. 测试集目标时间前一小时平均旅行时间的最大值。
18. 测试集目标时间前一小时平均旅行时间的最小值。
19. 测试集目标时间前一小时平均旅行时间的方差。
20. 测试集目标时间前一小时平均旅行时间中位数在历史目标时间前一小时平均旅行时间的排名。
21. 测试集目标时间前一小时平均旅行时间中位数在历史目标时间前一小时平均旅行时间的排名占比。
22. 测试集目标时间前一小时中位数与历史测试集前一小时平均时间中位数的均值的差值。
23. 测试集三个目标时间前一小时均值数与历史测试集前一小时平均时间均值的均值的差值。
24. 测试集三个目标时间前一小时平均旅行时间的中位数。
25. 测试集三个目标时间前一小时平均旅行时间的均值。
26. 测试集三个目标时间前一小时平均旅行时间的最大值。
27. 测试集三个目标时间前一小时平均旅行时间的最小值。
28. 测试集三个目标时间前一小时平均旅行时间的方差。
29. 通过平滑后数据提取的以上特征。
30. **星期维度**
31. 所有周一周二……周日历史旅行时间的中位数。
32. 通过平滑后数据提取的以上特征。

**4. 道路属性**

1. 各个路段的路长。

2. 各个路段的路宽。

3. 各个路段的路长与路宽的乘积。

## 建模

1. **模型选择**

主要采用了XGBOOST模型进行建模。

1. XGBOOST模型简介：

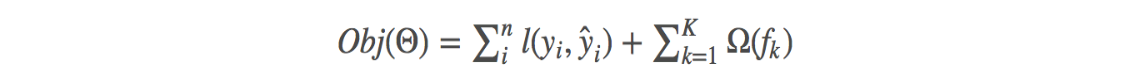
Xgboost是一个开源软件库，提供了C++、JAVA、Python等接口，具有伸缩性、可移动性、并发性等特点。且由于其训练速度快，效果好的特性，在许多数据挖掘比赛中崭露头角。

相较于其他基于bagging策略的决策树而言，基于boosting策略的Xgboost的准确率和训练时间有显著的提高。Xgboost实现的是一种通用的Tree boosting策略，此算法中有个经典的子算法梯度提升决策树（GBDT），其原理为首先采用训练集和样本真值训练一棵树，然后采用这棵树预测训练集，由于预测值和真实值存在偏差，所以二者相减可以得到残差，接下来训练第二颗树，此时不再使用真值，而使用其残差作为标签，两棵树训练完成后，可以进一步得到新的残差，然后训练第三棵树，使残差一步步缩小。树的总量可以人为指定，也可以通过监控某些指标来停止训练。

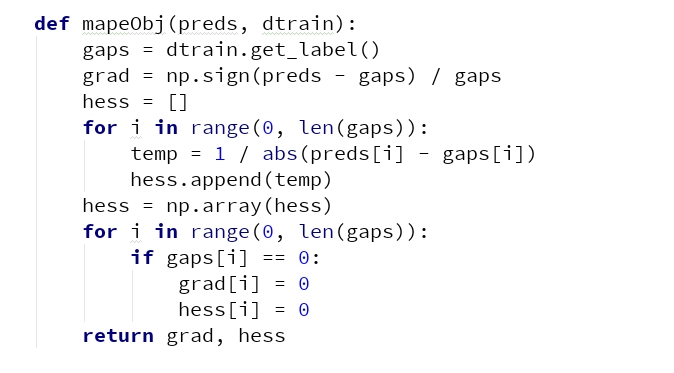
而相较于GBDT而言，Xgboost有许多改进的地方。一是将其损失函数从平方损失推广到二阶可导的损失，GBDT使用平方损失函数可以使残差一步步逼近于零，但是如果换用其它损失函数，则使用残差将不再能够保证。GBDT使用的残差是泰勒展式到一阶的结果，而Xgboost是展开到二阶的结果。二是加入了正则化项，可以控制模型的复杂程度，模型越复杂，对于训练集的误差可以很低，但是对于测试集的误差却很高，越容易发生过拟合(overfitring)，Xgboost加入了L2正则项，可以有效控制模型的复杂度。三是剪枝策略，Xgboost会一直分裂到最大深度，然后进行剪枝，如果发现某个节点之后不再有正损失，就去除这个分裂，这是一种全局最优的策略。

1. 参数调优介绍。
2. 通用参数介绍：
3. booster[default: gbtree]：选择每次迭代的模型，gbtree梳妆模型，gbliner线性模型。
4. silent[default:0]：是否开启静默模式。
5. nthread[default: max]：选择线程数。
6. booster参数介绍：
7. eta[default: 0.3]：步长，减少每一步的权重，提高模型鲁棒性。
8. min\_child\_weight[default: 1]：决定最小叶子节点样本权重和，避免过拟合。
9. max\_depth[default: 6]：树的最大深度，防止过拟合。
10. max\_leaf\_nodes：树的最大节点或叶子数量（此参数会掩盖max\_depth）。
11. gamma[default: 0]：指定了节点分裂所需要最小函数下降值。
12. max\_delta\_step[default: 0]：限制每棵树权重改变的最大步长。
13. subsample[default: 1]：对于每棵树随机采样的比例，避免过拟合。
14. colsample\_bytree[default: 1]：控制每棵树随机采样列数的占比。
15. colsample\_bylevel[default: 1]：控制树的每一级的每一次分裂对列数的采样占比。
16. lambda[default: 1]：权重的L2正则化项。
17. alpha[default: 1]：权重的L1正则化项。
18. scale\_pos\_weight[default: 1]：当样本十分不平衡时，设定一个正值使其更快收敛。
19. 自定义目标函数和评价函数：
20. 目标函数：

目标函数通常由预测误差项和正则项组成。

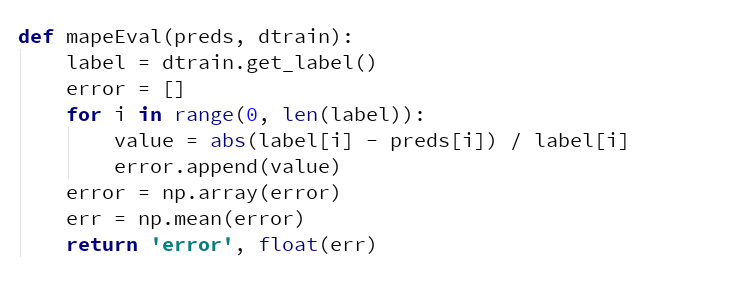


误差项代表数据在数据集上的拟合程度，正则项代表模型的复杂程度。Xgboost的目标函数需要返回目标函数的一阶和二阶梯度。



1. 评价函数：

本次比赛采用MAPE评价指标，为了在建模时实时通过交叉验证获取模型状态，选择更好的迭代停止时机，定义了模型的评价函数。



1. 建模流程：
2. 构建线下线上训练集，保证参数调优。任务是预测7月份时段的平均旅行时间，所以我们选择了使用5月份作为线下训练集，6月份作为线下测试集，6月份作为线上训练集，7月份作为线上测试集进行建模。
3. 特征变换。对所有数值特征进行了ZSCORE变换，增加了每个特征蕴含的信息量，并对数据进行了归一化，减少了迭代所需要的时间。