Experiment report

Student Number	1120182525	College	徐特立学院
Name	梁瑛平	Major	计算机科学与技术

Intelligent Transportation Experiment

1 Introduction .

With the opening of the era of mobile Internet, every traveler has become a contributor of traffic information. Super-large location data are processed and fused in the cloud to generate traffic information of the whole time without blind spots in the city. Smart transportation hopes to realize the prediction of traffic conditions and help social smart travel and intelligent control of urban traffic.

2 Aim

Based on the Internet traffic information to establish the algorithm model. Accurately predict the average travel time of each key road section in a certain period of time

3 Data set

Link data
Historical travel time data

4 Evaluation index

MAPE（Average absolute percentage error）

TTP: Predicted travel time
TTR: True travel time
N: The predicted number of Links
Ti: Number of time slices predicted on the i link

$$MAPE = \frac{\sum_{i=1}^{N}\sum_{j=1}^{T_i}\frac{\left|ttp_{i,j} - ttr_{i,j}\right|}{ttr_{i,j}}}{\sum_{i=1}^{N} T_i}$$

5 Experimental process

原始数据：

link_info 表里共存着每条路的 id，长度，宽度和类型,共有 132 条路：

	link_ID	length	width	link_class
0	4377906289869500514	57	3	1
1	4377906284594800514	247	9	1
2	4377906289425800514	194	3	1
3	4377906284525800514	839	3	1
4	4377906284422600514	55	12	1

link_top 里储存每一条路的上下游关系, in_links 里放着这条路的上游路 id，中间用#分割，而 out_links 里则给出了这条路的下游路 id；下游路可以理解为出路，上游路为入路：

	link_ID	in_links	out_links
0	4377906289869500514	4377906285525800514	4377906281969500514
1	4377906284594800514	4377906284514600514	4377906285594800514
2	4377906289425800514	NaN	4377906284653600514
3	4377906284525800514	4377906281234600514	4377906280334600514
4	4377906284422600514	3377906289434510514#4377906287959500514	4377906283422600514

travel_time 表里存着这 132 条路从 2017.4-2017.6 以及 2016.7 每天车通过路的平均旅行时间，统计的时间间隔为 2 分钟；除了 2016.4 到 6 月每天的信息，还有 2017.7 月每天 6:00-8:00, 13:00-15:00, 16:00-18:00 的记录，然后我们需要预测的就是 7 月每天在早高峰，午平峰，晚高峰三个时间段(8:00-9:00, 15:00-16:00, 18:00-19:00)每条路上的车平均旅行时间：

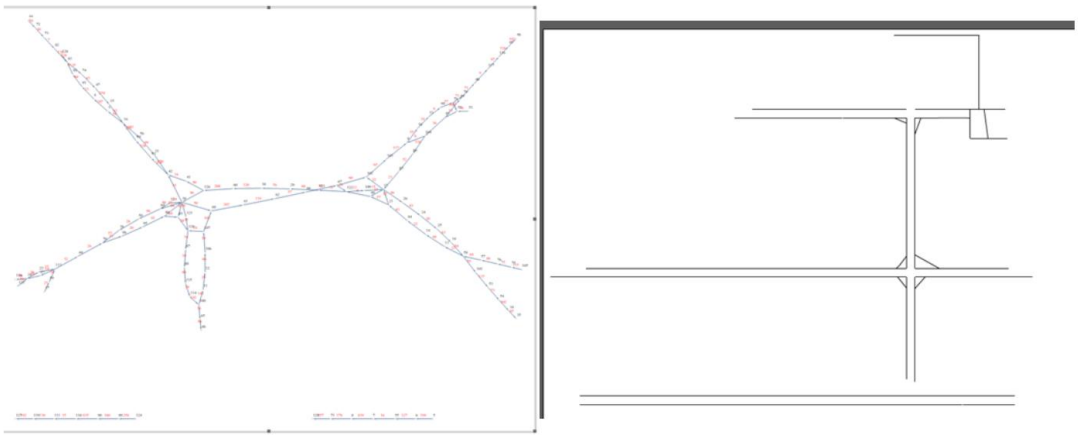
```
              link_ID        date                                time_interval  \
0  4377906283422600514  2017-05-06  [2017-05-06 11:04:00,2017-05-06 11:06:00)
1  3377906289434510514  2017-05-06  [2017-05-06 10:42:00,2017-05-06 10:44:00)
2  3377906285934510514  2017-05-06  [2017-05-06 11:56:00,2017-05-06 11:58:00)
3  3377906285934510514  2017-05-06  [2017-05-06 17:46:00,2017-05-06 17:48:00)
4  3377906287934510514  2017-05-06  [2017-05-06 10:52:00,2017-05-06 10:54:00)

   travel_time
0          3.0
1          1.0
2         35.2
3         26.2
4         10.4
```

可以使用的基本特征如图：

特征	类型	说明
link_ID	string	每条路段(link)唯一标识
link_seq	int	132条路段从1-132编号；
length	int	link长度(米)
width	int	link宽度(米)
link_class	int	link道路等级，如1代表主干道
date	string	日期，如2015-10-01
week	int	星期，根据日期映射到星期，从1到7
time_interval	string	时间段，如[2015-09-01 00:00:00,2015-09-01 00:02:00)
time_slot	int	时间片，根据时间段映射一天24小时，从1到720，每段2分
avg_travel_time	float	该时间片平均旅行时间的均值，反应集中趋势
inlinks_atl_1	float	车辆在该路段上（timeslot-上游平均旅行时间）时间段的平均旅行时间，上游最多4个路段汇入，如果小于4，则大于4的为0。 测试集中，该值是通过决策树回归预测出来的。
inlinks_atl_2	float	
inlinks_atl_3	float	
inlinks_atl_4	float	
inlinks_atl_1	float	车辆在该路段上（timeslot+平均旅行时间）时间段的平均旅行时间，下游最多4个路段汇出，如果小于4，则大于4的为0。 测试集中，该值是通过决策树回归预测出来的。
inlinks_atl_2	float	
inlinks_atl_3	float	
inlinks_atl_4	float	
travel_time	float	车辆在该路段上的平均旅行时间(秒)

地理图可视化后的结果：



基本思路：

这是一个关于时间序列预测的问题，并不是普通的回归问题，而是自回归，一般的回归问题比如最简单的线性回归模型:Y=a*X1+b*X2，我们讨论的是因变量 Y 关于两个自变量 X1 和 X2 的关系，目的是找出最优的 a 和 b 来使预测值 y=a*X1+b*X2 逼近真实值 Y. 而自回归模型不一样，在自回归中，自变量 X1 和 X2 都为 Y 本身，也就是说 Y(t)=a*Y(t-1)+ b*Y(t-2)，其中 Y(t-1)为 Y 在 t-1 时刻的值，而 Y(t-2)为 Y 在 t-2 时刻的值，换句话说，现在的 Y 值由过去的 Y 值决定，因此自变量和因变量都为自身，这种回归叫自回归。

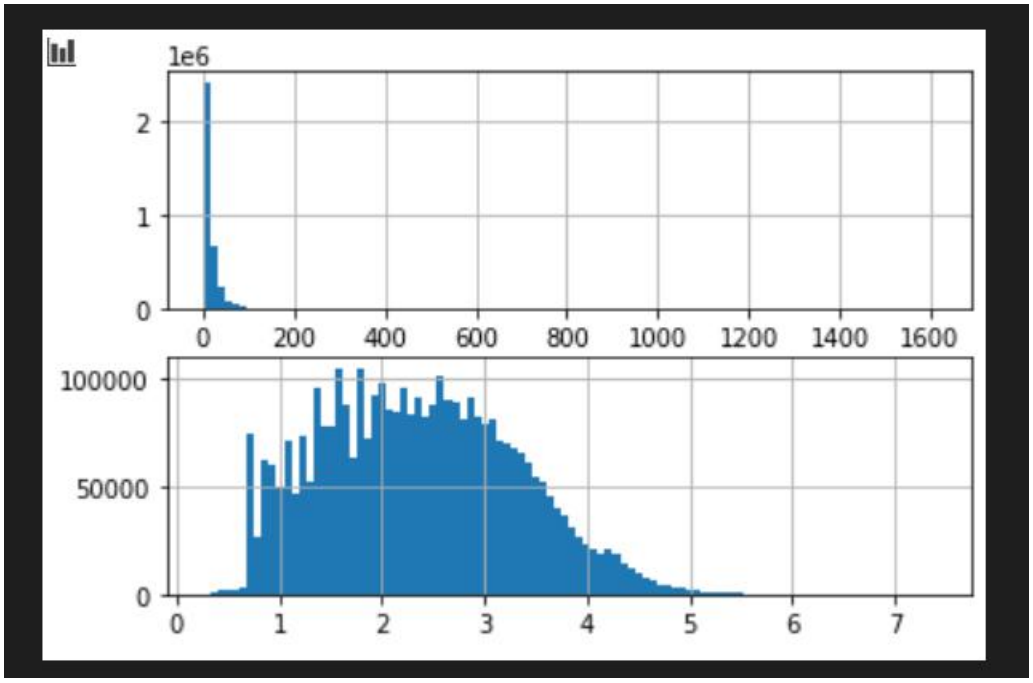
根据题目给出的信息，除了路本身的信息外，训练数据基本上只有旅行时间，而我们要预测的也是未来的平均旅行时间，而且根据我们的常识，现在的路况跟过去一段时间的路况是很有关系的，因此该问题应该是一个自回归问题，用过去几个时刻的交通状况去预测未来时刻的交通状况。

传统的自回归模型有自回归模型（AR）、移动平均模型（MA）、自回归移动平均模型（ARMA）以及差分自回归移动平均模型（ARIMA），这些自回归模型都有着严格理论基础，讲究时间的平稳性，需要对时间序列进行分析才能判断是否能使用此类模型。

特征工程：

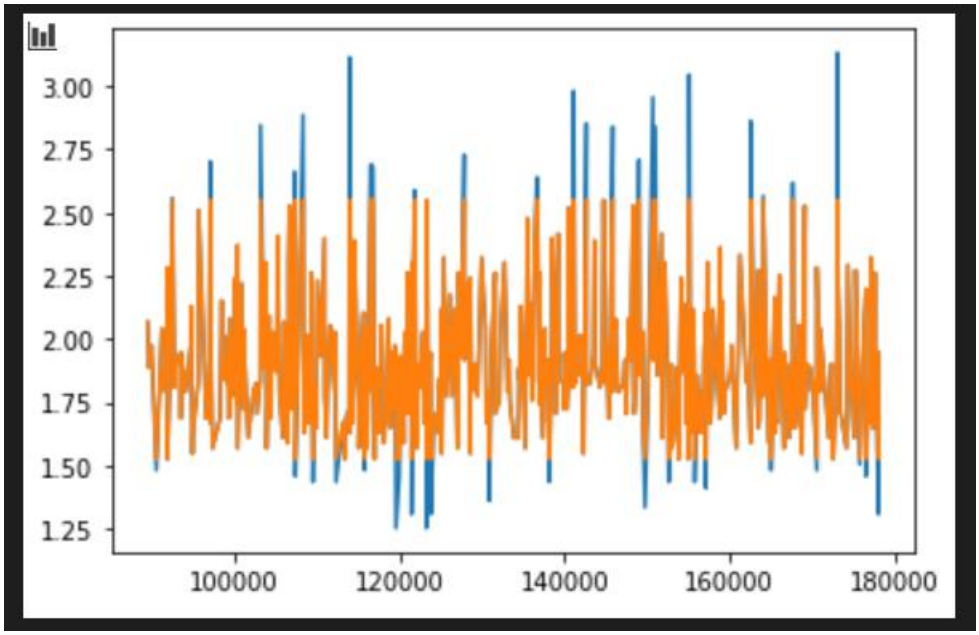
分布均衡化：

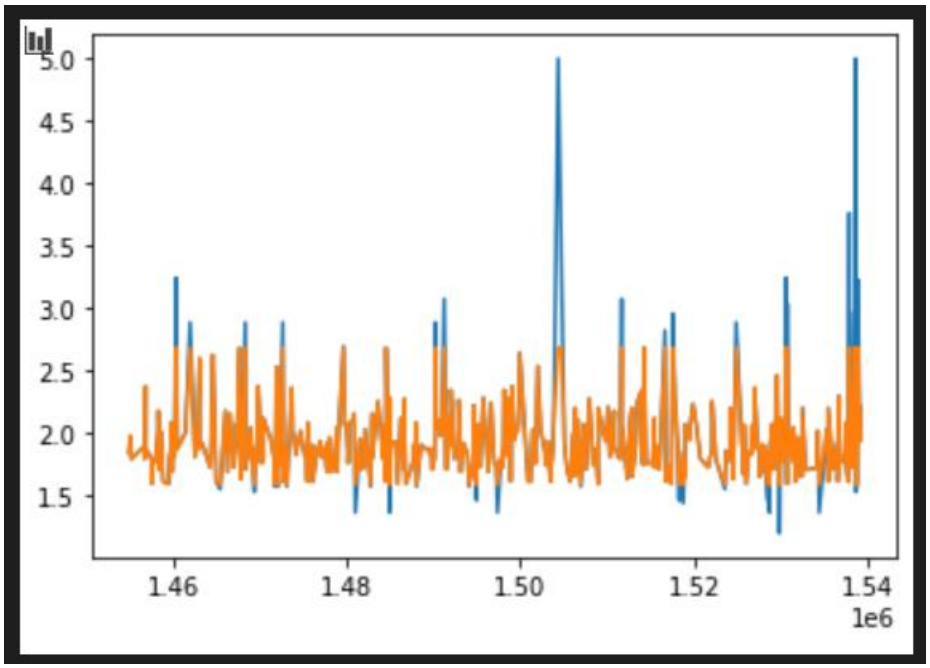
查看数据中预测时间分布，时间分布较为集中（数值较小的 label 比较多，数值较大的 label 比较少），因此加上 log 变换再查看均衡化后的分布。处理后类似正态分布，比较适合模型来处理：



处理离群点：

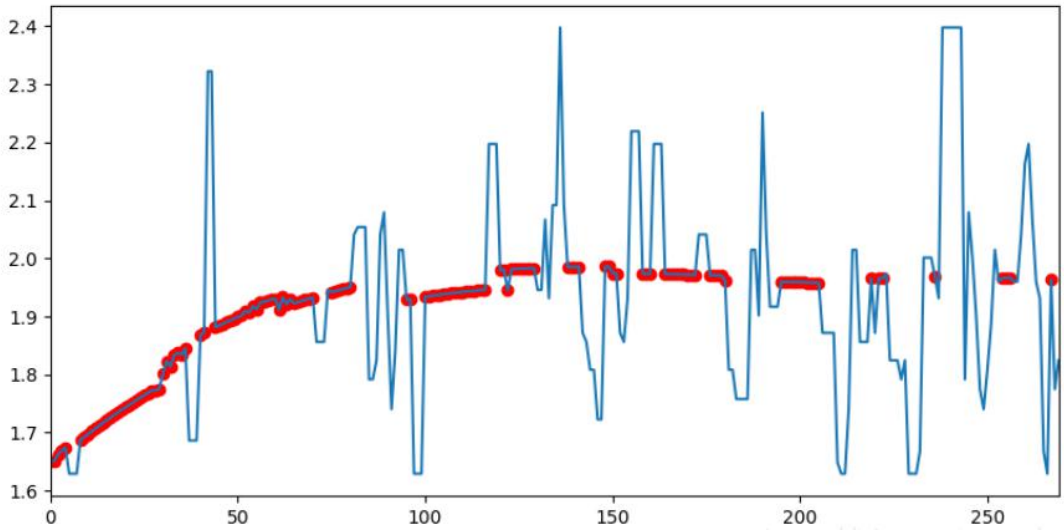
即使做了 log 变换后，还是有部分 travel_time 值过于大，为了消除一些离群点的影响，我们对 travel_time 做一个百分位的裁剪 clip，我们把上下阈值设为 95 百分位和 5 百分位，即将所有大于上阈值的 travel_time 归为 95 百分位数，而小于小阈值的 travel_time 设为 05 百分位数：





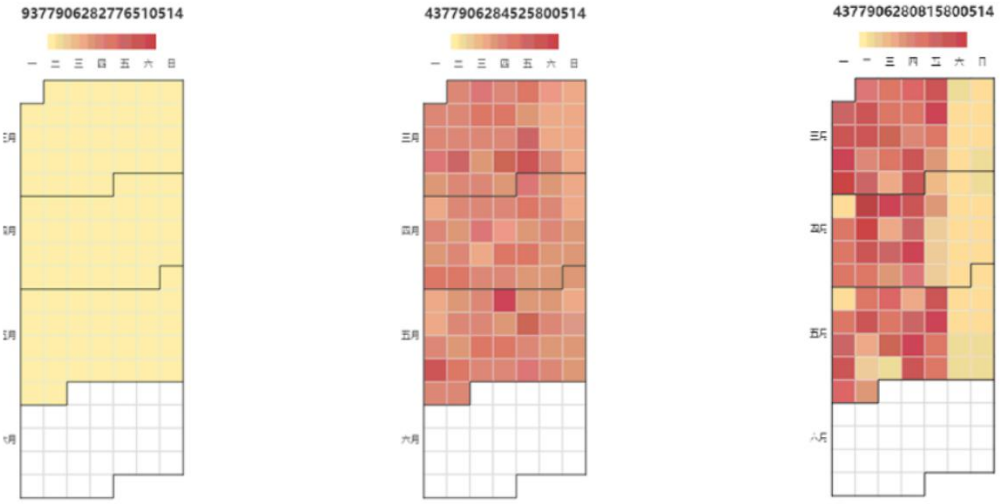
补全缺失值：

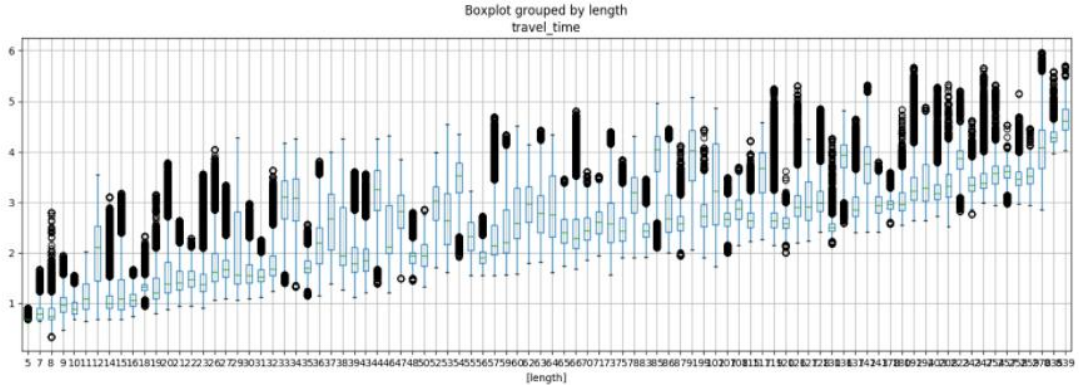
自动补全的方法就是用预训练一个模型去补全缺失值：训练已有的数据，把缺失的数据当做是要预测的数据，有很多模型可以补全缺失值，例如随机森林等，只要 feature 构造合理，这种补全的方法要比前面提到的手动的方法效果要好一些。我们可以看一下补全的效果，我们画出某路某天的 travel_time 变化如下图，红色的部分是补全的数据，蓝线为原来的数据，可以看出补全的数据比较保守，基本贴近于 hour_trend：

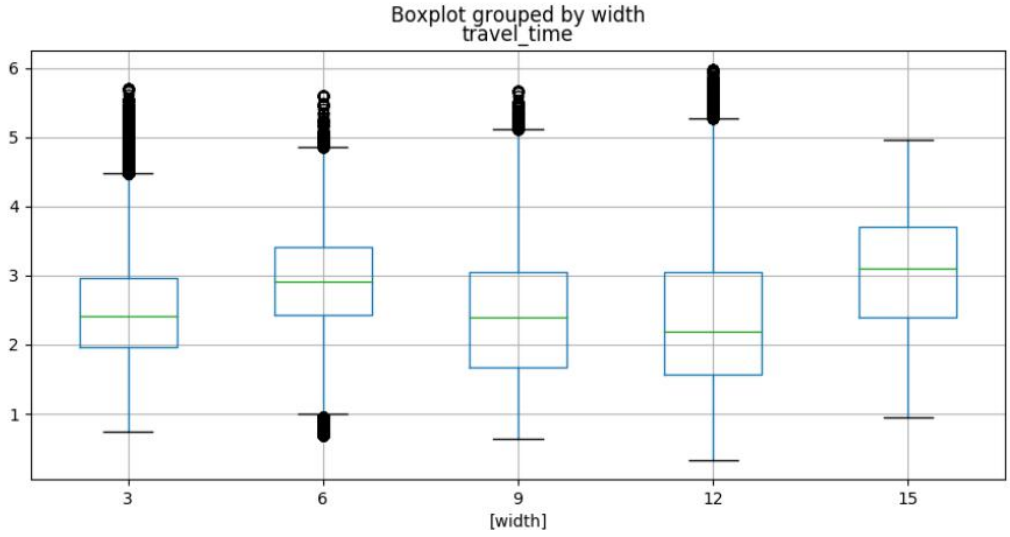


相关性分析：

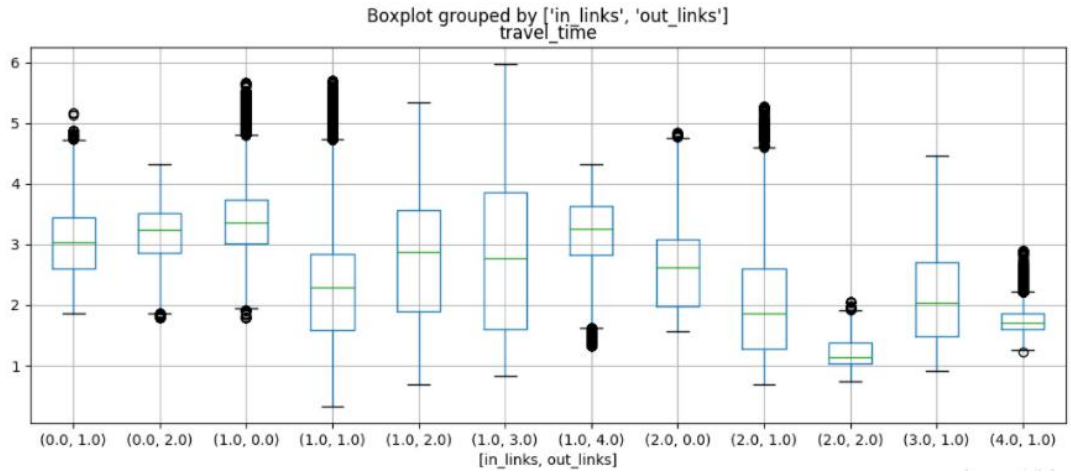
路的长度和宽度：毫无疑问，路的长度与 travel_time 是成正比的，路越长，travel_time 越大，所以路的长度特征应该是非常重要的，由下图也可以看出他们存在正相关关系。



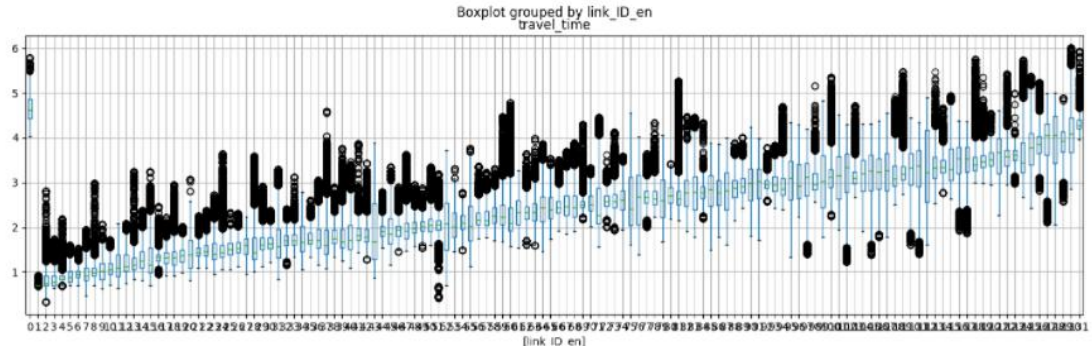




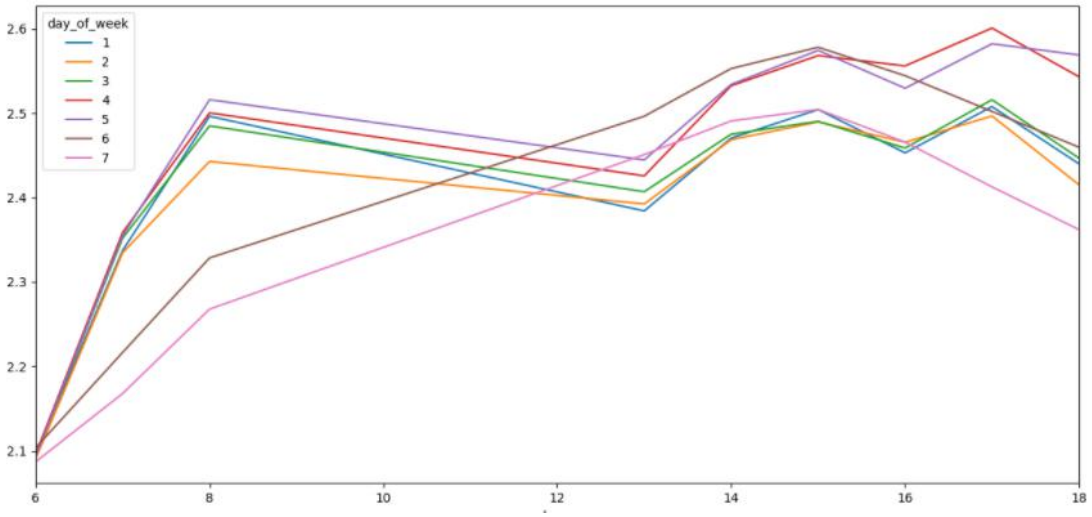
上下游关系：对于每一条路来说，基本上都有上游和下游，但是有的路处于尽头，本身就没有上游或者下游，我们这里只根据上游和下游的数量进行划分，统计每一条路的上游和下游路的个数，然后画出箱线图。其中 in_links 和 out_links 分别表示的上游路和下游路的数量。



ID 特征： scikit 库有直接将 ID 映射到标签的工具，映射后 ID 特征变为了从 1 到 132 离散的数字：



基本时间特征：根据常识来说，车在工作日是比较多的，travel_time 相对大，而在一天之内，上下班高峰期也直接影响 travel_time，而且假期也是很影响大家的出行的. 我们可以跟上面一样对 week_day 和 hour 以及 vacation 分别画出箱线图看看，但我后来发现了 week_day 和 hour 是有一定的关联的，比如周一的早上 8 点与周末的早上 8 点是完全不一样的，下面给一天之内每个小时平均 travel_time 的变化情况。



训练模型：

输入数据：

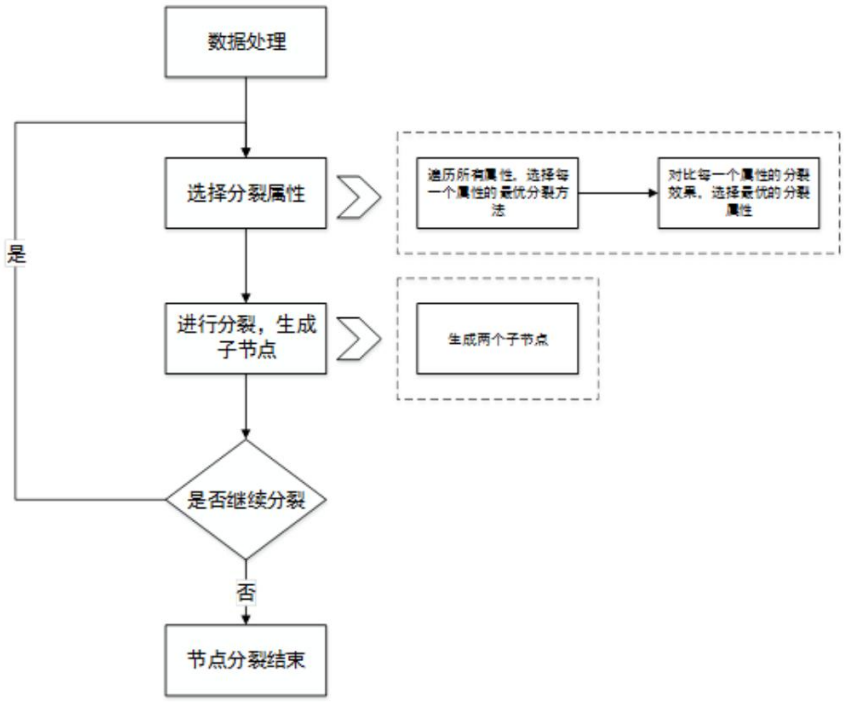
训练模型时直接用上面的 feature 训练对应的 travel_time 就可以，但是时间序列 feature 在交叉验证和测试的时候就不能跟训练一样了，因为当我们在预测出 t 时刻的 travel_time 后，需要把这个 travel_time 作为预测 t+1 时刻 travel_time 的 lagging1 特征，这个 lagging 特征是需要根据上次预测的结果进行更新的，如此反复直到预测到最后一个时刻的 travel_time。

验证数据：

5 次交叉验证。

模型选择：

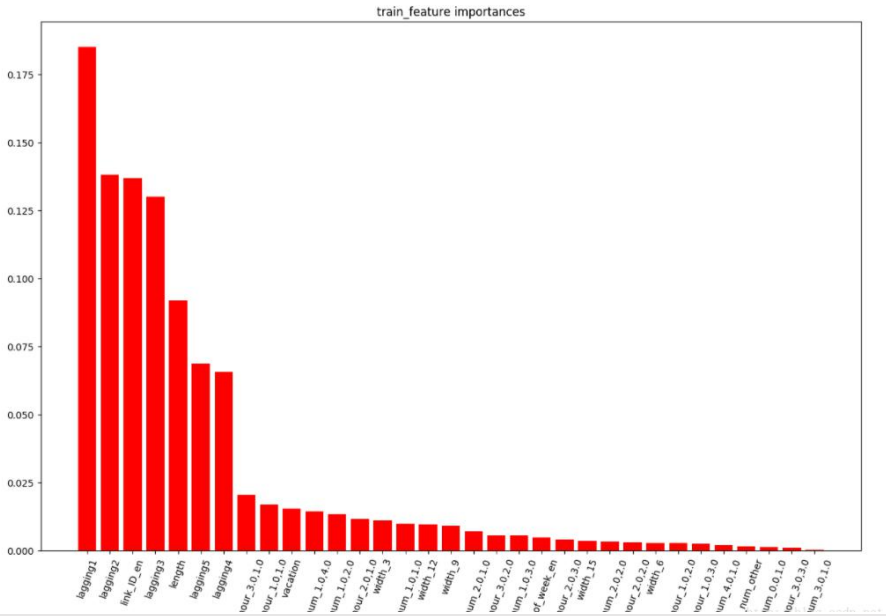
模型使用 XGBRegressor 提供的 gbtree，即 CART。分类与回归树（Classification and Regression Trees, CART）是由四人帮 Leo Breiman, Jerome Friedman, Richard Olshen 与 Charles Stone 于 1984 年提出，既可用于分类也可用于回归。该算法对于回归树，采用样本方差衡量节点纯度。节点越不纯，节点分类或者预测的效果就越差。



6 Experimental results and analysis

```
-[INFO] train: MAPE=0.286701
-[INFO] valid: MAPE=0.293537
```

分数不是很高的原因可能是有些无用特征没有删除而导致了过拟合。



7 Gain and experience