# 1. 数据预处理

**问题：**

现实车载数据非常脏，对经纬度做可视化可发现大量漂移点、异常点。同时在处理时发现有文件存在重复数据，对数据的预处理花费了大量时间。

**解决方法：**

1. 首先对数据做EDA ，查看数据存在的异常

2. 在预处理时，要把所有可能性的问题考虑到，根据数据特性即EDA情况对数据做清洗

# 2. 建模不全面

**问题：**

在建模过程中，遇到困难，没有充分解决所给问题，天气数据没有用到，实际上用到天气数据会进一步丰满模型，考虑更多的因素。

**解决问题：**

对于模型的扩展，即应用天气模型的思路：

（1）调用百度地图API接口，进行坐标转换

（2）得到汽车行驶所有天 经过的城市、县，可根据经纬度的最大值，最小值找到汽车行驶边界

（3）由经过城市对应天气数据，得到天气、温度两个特征（天气数据中，只有这两个特征无缺失值且会影响到汽车行驶、油耗），湿度特征由于存在大量缺失值，因此可作为权重项加入（有的话可加分，没有的话就保持原始分数）。根据情况判断，得到天气得分

（4）将天气得分加入节能评价指标

# 3. 代码编写规范

## 3.1 效率问题

**问题：**

解决问题用到的主要模块是pandas，主要的数据结构是dataframe，主要的查找操作具体语句为 df.iloc[I,j] ， 对pandas dataframe操作不熟练，该语句效率及其低下。

**解决方法：**

参考<https://blog.csdn.net/sinat_38682860/article/details/85046081>

（1）data.iloc 可用data.at代替

（2）对于循环操作的改进，一直用的是for ,data.iloc进行值的更新，该博客提供了以下几种方法：

**1）使用itertuples() 和iterrows() 循环**

df.iterrows为DataFrame中的每一行产生（index，series）这样的元组。

df.itertuples为每一行产生一个**namedtuple**，并且行的索引值作为元组的第一个元素

**2) Pandas的 .apply()方法**

对一列进行操作，不用循环遍历该列的每一个数据，直接apply函数

df.apply( lambda row: fun(f1,f2) ,axis = 1 )

**3) 矢量化操作：使用.isin()选择数据**

相比于大量的条件判断 if-else语句，用 isin( range(start,end)) 矢量化一行代码可以解决，且速度更快

## 3.2 异常捕捉

**问题：**

在代码编写过程中，由于跑完450辆车才能得出结果，极其耗费时间，且没有保存中间结果，出现问题所有数据丢失

**解决方法：**

（1）编写异常捕捉模块：

考虑到会出现异常的问题，如是否结束判断、除法分母为0判断等

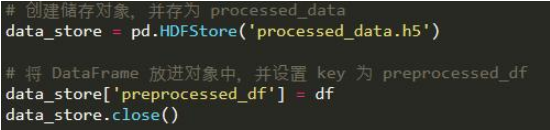
（2）中间结果保存

1）面对大型数据，错误难免发生，为避免之前得到的数据丢失造成的时间浪费，建议每跑50/100个数据就保存一下中间结果to\_csv

2）参考以上博文，使用HDFStore防止重新处理

“Pandas有一个内置的解决方案，它使用 **HDF5**，这是**一种专门用于存储表格数据阵列的高性能存储格式。** Pandas的 **HDFStore** 类允许你将DataFrame存储在HDF5文件中，以便可以有效地访问它，同时仍保留列类型和其他元数据。它是一个类似字典的类，因此您可以像读取Python dict对象一样进行读写。”

a. 存储HDF5文件



b. 读取HDF5文件

