**《专业方向课程设计》大作业**

|  |
| --- |
| 乘用车零售量建模预测分析 |

题目：

|  |  |  |  |
| --- | --- | --- | --- |
| **姓名** | **学号** | **班级** | **成绩** |
| A | 1 | 软数1801 |  |
| B | 2 | 软数1801 |  |
| C | 3 | 软数1801 |  |

大连理工大学软件学院

2021年6月

目录

[1.项目简介 1](#_Toc76656600)

[1.1实验项目来源及业务背景 1](#_Toc76656601)

[1.2数据科学项目一般流程 2](#_Toc76656602)

[1.2.1 需求定义 2](#_Toc76656603)

[1.2.2 数据获取 3](#_Toc76656604)

[1.2.3 数据治理 3](#_Toc76656605)

[1.2.4 数据分析 3](#_Toc76656606)

[1.2.5 数据可视化 4](#_Toc76656607)

[1.3业务理解及本项目的具体步骤 4](#_Toc76656608)

[1.3.1 业务理解 4](#_Toc76656609)

[1.3.2 具体步骤 4](#_Toc76656610)

[2.数据探索 5](#_Toc76656611)

[2.1 描述性分析 5](#_Toc76656612)

[2.2 相关性分析 7](#_Toc76656613)

[2.3 分布分析 9](#_Toc76656614)

[2.4 缺失值分析 11](#_Toc76656615)

[2.5 异常值分析 14](#_Toc76656616)

[2.7 冗余值分析 15](#_Toc76656617)

[2.8 其他探索性分析 16](#_Toc76656618)

[2.9 数据清洗 17](#_Toc76656619)

[3.特征工程 18](#_Toc76656620)

[3.1 处理标称特征 18](#_Toc76656621)

[3.2 处理数值特征 19](#_Toc76656622)

[3.3 通过xgboost预测重新填充201612、201701的销量数据 24](#_Toc76656623)

[3.4 分组，引入时间特征 27](#_Toc76656624)

[4.模型构建 29](#_Toc76656625)

[4.1 划分特征和标签、训练集和测试集 29](#_Toc76656626)

[4.2 默认超参数观察模型效果 31](#_Toc76656627)

[4.3 选取默认参数效果较优的三个模型进行调参 33](#_Toc76656628)

[4.4 模型融合 34](#_Toc76656629)

[4.5 本地验证 34](#_Toc76656630)

[4.6 对天池测试集预测并保存为CSV 35](#_Toc76656631)

[5.项目总结及分析 35](#_Toc76656632)

# 1.项目简介

## 1.1实验项目来源及业务背景

“十三五”是汽车工业发展的新时期，经济形势复杂多变，汽车产业政策频出，因此准确预测汽车销量对于政府和企业都具有极其重要的现实意义。分区域的销量预测有利于各地政府准确地把握汽车市场发育与成长态势，及时调整宏观行业政策，同时有助于行业监管部门对汽车厂商实现产能监控。

对于企业，在市场环境瞬息万变的信息时代，需要充分了解消费者诉求，预见市场未来的需求量和可能存在的销售变化趋势，合理规划产能，正确制定生产计划，实施以销定产的生产策略。

现有汽车销量预测研究大多是宏观预测，预测对象是整个市场的总体销量，预测粒度宽泛。对于政府、行业监管部门以及汽车企业需要有更细化粒度的销量预测解决方案。

因此，尝试借助大数据及机器学习的力量，解决这一难题。本课题第一阶段用到的数据包括2012年1月-2017年10月盐城分车型销量配置数据，也允许使用任何可公开的外部数据辅助预测。在对这些数据进行初步分析的基础上，将进行数据预处理、特征提取、模型选择、参数优化等不同阶段实验。

项目的目的是预测2017年11月盐城分车型销量数据，预测2017年12月盐城分车型销量数据。并通过该实践项目提升大数据分析、建模与预测能力。

当前，大数据作为新一轮工业革命中最为活跃的技术创新要素，正在全面重构全球生产、流通、分配、消费等领域，对全球竞争、国家治理、经济发展、产业转型、社会生活等方面产生全面深刻影响。

大数据虽然孕育于信息通信技术，但它对社会、经济、生活产生的影响绝不限于技术层面。更本质上，它是为我们看待世界提供了一种全新的方法，即决策行为将日益基于数据分析，而不是像过去更多凭借经验和直觉。具体来讲，大数据将有以下作用。

1）对大数据的处理分析正成为新一代信息技术融合应用的结点。

移动互联网、物联网、社交网络、数字家庭、电子商务等是新一代信息技术的应用形态，这些应用不断产生大数据。

云计算为这些海量、多样化的大数据提供存储和运算平台。通过对不同来源数据的管理、处理、分析与优化，将结果反馈到上述应用中，将创造出巨大的经济和社会价值，大数据具有催生社会变革的能量。

2）大数据是信息产业持续高速增长的新引擎。

面向大数据市场的新技术、新产品、新服务、新业态会不断涌现。在硬件与集成设备领域，大数据将对芯片、存储产业产生重要影响，还将催生出一体化数据存储处理服务器、内存计算等市场。

在软件与服务领域，大数据将引发数据快速处理分析技术、数据挖掘技术和软件产品的发展。

3）大数据利用将成为提高核心竞争力的关键因素。

各行各业的决策正在从“业务驱动”向“数据驱动”转变。在商业领域，对大数据的分析可以使零售商实时掌握市场动态并迅速做出应对，可以为商家制定更加精准有效的营销策略提供决策支持，可以帮助企业为消费者提供更加及时和个性化的服务。在医疗领域，可提高诊断准确性和药物有效性。在公共事业领域，大数据也开始发挥促进经济发展、维护社会稳定等方面的重要作用。

4）大数据时代，科学研究的方法手段将发生重大改变。

例如，抽样调查是社会科学的基本研究方法，在大数据时代，研究人员可通过实时监测、跟踪研究对象在互联网上产生的海量行为数据，进行挖掘分析，揭示出规律性的东西，提出研究结论和对策。

## 1.2数据科学项目一般流程

数据科学项目的完整流程通常是这样的五步骤，如下图所示

图示

描述已自动生成

### 1.2.1 需求定义

需求定义是数据科学项目和数据科学比赛的较大不同之处，在真实情景下，我们往往对目标函数、自变量、约束条件都并不清晰。需要通过访谈、论文、文档等等形式对问题进行系统地分析，将实际问题量化为可以解决的抽象问题，确定自变量、约束条件以及目标函数。在真实情景下，需求往往是多变化的，临时的，如何把握好需求成为了整个项目后续推进的关键要素。

### 1.2.2 数据获取

数据获取的形式主要包括：现有数据库的调用、现有API的调用、自行设计的爬虫。

本项目中，我们使用的是比赛方提供的数据来进行分析与预测。

### 1.2.3 数据治理

数据治理第一步就是数据的定义，而数据的定义通过 Python的各种ORM框架和admin系统，可以非常出色地完成数据仓库的定义和管理。通过 airflow 我们又可以很好的对ETL过程做全流程的监控。所以，我们采用Python作为数据治理的工具。

### 1.2.4 数据分析

在python中，数据分析常用的工具包如下：

1.Numpy：

Numpy是python科学计算的基础包，它提供以下功能（不限于此）：

　　(1)快速高效的多维数组对象naarray

　　(2)用于对数组执行元素级计算以及直接对数组执行数学运算的函数

　　(3)用于读写硬盘上基于数组的数据集的工具

　　(4)线性代数运算、傅里叶变换，以及随机数生成

　　(5)用于将C、C++、Fortran代码集成到python的工具

2.pandas

pandas提供了使我们能够快速便捷地处理结构化数据的大量数据结构和函数。pandas兼具Numpy高性能的数组计算功能以及电子表格和关系型数据（如SQL）灵活的数据处理能力。它提供了复杂精细的索引功能，以便更为便捷地完成重塑、切片和切块、聚合以及选取数据子集等操作。

对于金融行业的用户，pandas提供了大量适用于金融数据的高性能时间序列功能和工具。

DataFrame是pandas的一个对象，它是一个面向列的二维表结构，且含有行标和列标。

3.matplotlib

matplotlib是最流行的用于绘制数据图表的python库。

4.Scipy

Scipy是一组专门解决科学计算中各种标准问题域的包的集合。

### 1.2.5 数据可视化

​ 数据可视化是在整个数据分析非常重要的一个辅助工具,可以清晰的理解数据，从而调整我们的分析方法。数据可视化能将数据进行可视化,更直观的呈现，使数据更加客观、更具说服力。数据可视化与python语言也有相应的结合，Matplotlib是一个Python 2D绘图库，它可以在各种平台上以各种硬拷贝格式和交互式环境生成出具有出版品质的图形。 Matplotlib可用于Python脚本，Python和IPython shell，Jupyter笔记本，Web应用程序服务器和四个图形用户界面工具包。

Matplotlib试图让简单的事情变得更简b单，让无法实现的事情变得可能实现。 只需几行代码即可生成绘图，直方图，折线图，条形图，错误图，散点图等。

为了简单绘图，pyplot模块提供了类似于MATLAB的界面，特别是与IPython结合使用时。 对于高级用户，可以通过面向对象的界面或MATLAB用户熟悉的一组函数完全控制线条样式，字体属性，轴属性等。

## 1.3业务理解及本项目的具体步骤

### 1.3.1 业务理解

本项目的主要目的是对乘用车零售量进行建模预测分析，分区域的销量预测有利于各地政府准确地把握汽车市场发育与成长态势，及时调整宏观行业政策，同时有助于行业监管部门对汽车厂商实现产能监控。对于企业，在市场环境瞬息万变的信息时代，需要充分了解消费者诉求，预见市场未来的需求量和可能存在的销售变化趋势，合理规划产能，正确制定生产计划，实施以销定产的生产策略。

现有汽车销量预测研究大多是宏观预测，预测对象是整个市场的总体销量，预测粒度宽泛。对于政府、行业监管部门以及汽车企业需要有更细化粒度的销量预测解决方案。

### 1.3.2 具体步骤

本项目的具体展开步骤分为如下几步：

1. 数据获取。项目的训练和预测数据都是平台给出，因此直接下载获得即可。

2. 数据探索。首先在宏观上理解数据，主要包括样本数据的总体情况分析预览、基本统计的特征分析以及对全局特征掌握，Label值情况、趋势、周期性等时序特征探索等，然后在细节上进行探索，主要工作集中在关键单变量EDA,变量和Label的关系，以及相应的可视化分析。

3. 特征工程。 进行数据样本的特征选择。进行建模训练之前的预处理。

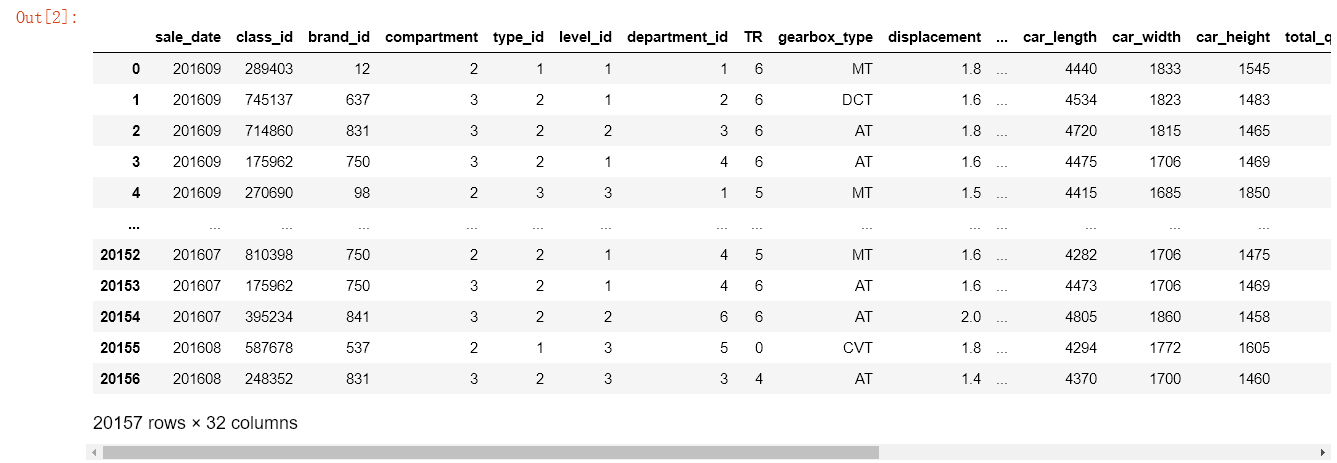
4. 分析建模。 根据数据样本特性，选择合适的模型进行建模分析。

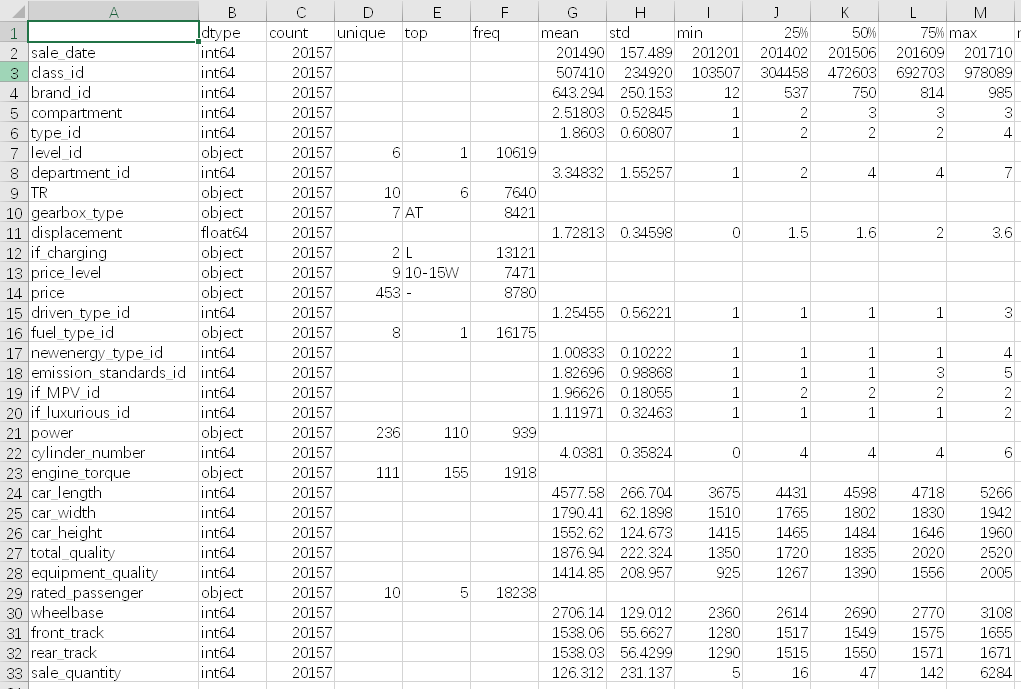
5. 模型分析。 根据之前工作的探索，选择最终合适的模型作为最终模型。

6. 总结分析。 总结项目，做出文档化分析说明。

# 2.数据探索

## 2.1 描述性分析





我们将所有描述性信息整理到了一张表中。

可以看到数据一共有20157行32列。

同一个 class\_id 的车可以在不同 sale\_date 反复卖。

而且 class\_id 相同不代表车相同，class\_id+各种配置，唯一决定一辆车。

预测最终结果的要求是 sale\_date+class\_id 做主码，分组求和。

可以看到所有的数据都没有直接明显的缺失np.nan。

观察到有为 0 的值，有可能是缺失，接下来需要进一步分析。

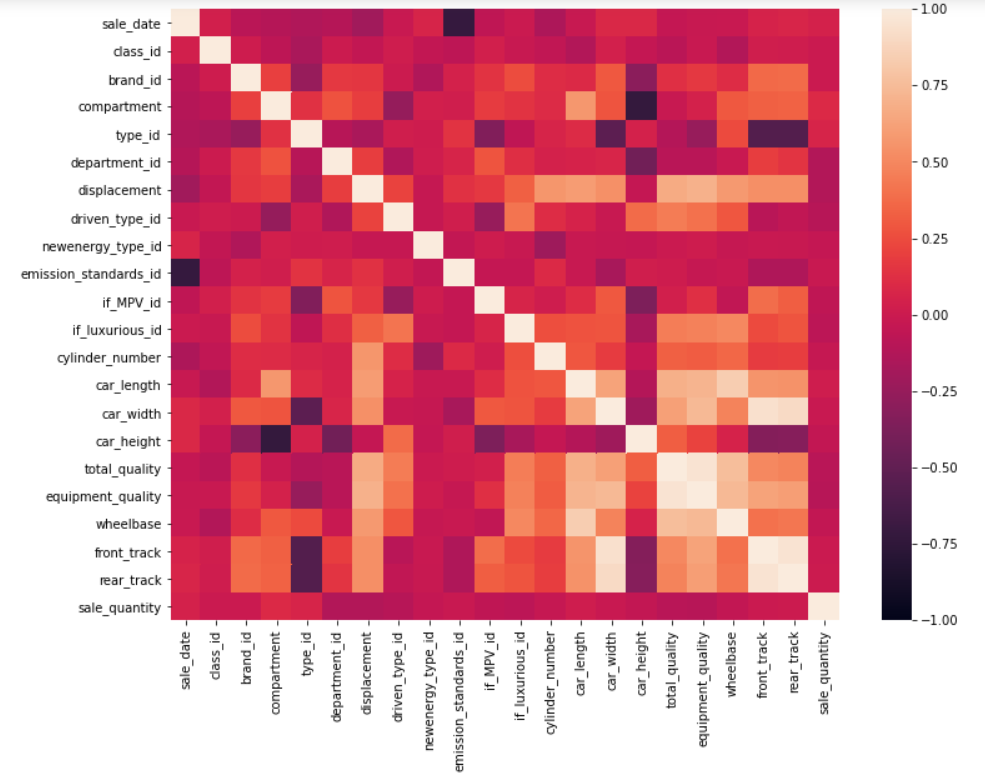
观察到有为 '-' 的值，显然是缺失，要进一步分析、填充。

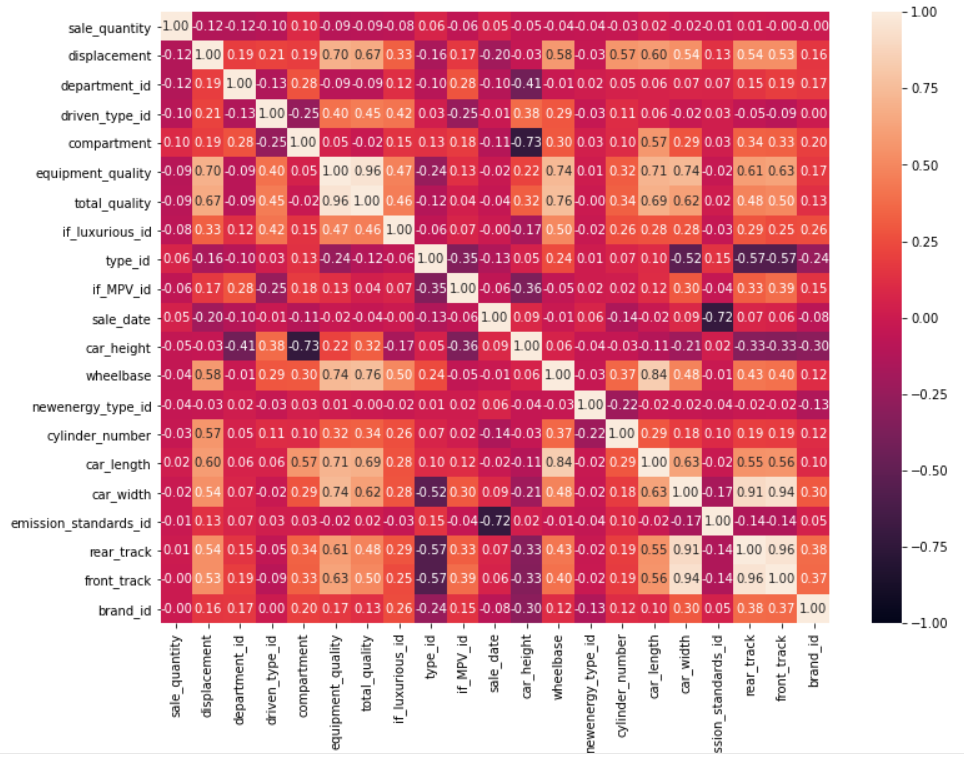
有一些标称特征，类型是int和float。

也有一些数值特征，由于缺失填充了横杠，类型是object。

这也是接下来需要处理的内容之一。

## 2.2 相关性分析





上图为相关性矩阵绘制的热图。

颜色特别浅或者颜色特别深就说明两个属性间的相关性很高。

相关性就是计算两个数据类型为数值型的向量数组间的相关性。

第二个图取的是与sale\_quantity相关性绝对值最高的前21个特征，包括它自己，然后就好巧不巧，数据类型为数值的正好就21个，和第一个图的特征数完全一致。

可以看到所有特征和销量的相关性都挺差的。

不过实际上数据类型为数值的标称特征没有计算这个的意义，它的作用是标识类别，数值上的大小一般都是没有意义的，这个相关性值代表不了它和标签的“相关性”。

然后发现和标签相关性最差的几个特征里好多都是数值特征，一些数据类型为数值的标称特征远排在它们之上，这是一件很离谱的事，可以说这几个数值特征和标签间的相关性差到了极点。

然后还发现许多特征与特征之间有十分强的相关性。

一般在处理缺失值时，可以利用特征之间的关联，采用分组填充的策略，一方面就是理解特征所表示的含义，另一方面就是用这个相关性矩阵热图来做参考。毕竟有时候特征字面义并不一定能理解的特别好，没有发现关联，那就需要参考这个相关性来挖掘数据间的关联。

## 2.3 分布分析

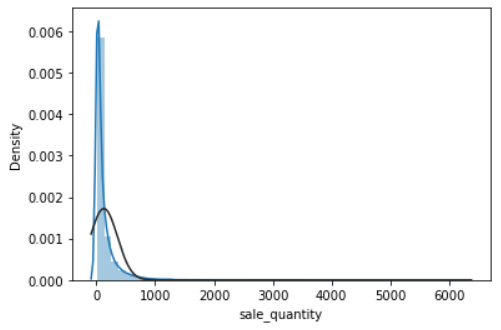


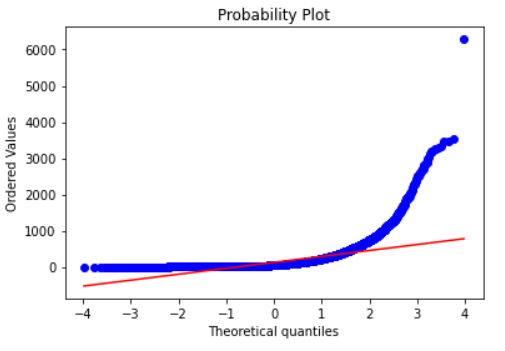
如图我们随便挑了几个属性做了下pairplot。

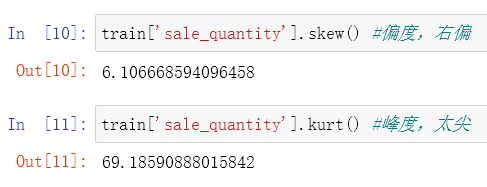
可以看到这些属性都和正态分布想去甚远，其他属性基本也都是这样，就不再进一步绘图了。

然后对于这些特征我们可以用boxcox1p平滑。

然后对于标签，如果平滑，预测之后就要再转换回去，所以我们要进一步观察一下。







通过观察正态曲线拟合图，我们可以看到怎么看都不是正态，尖的离谱。

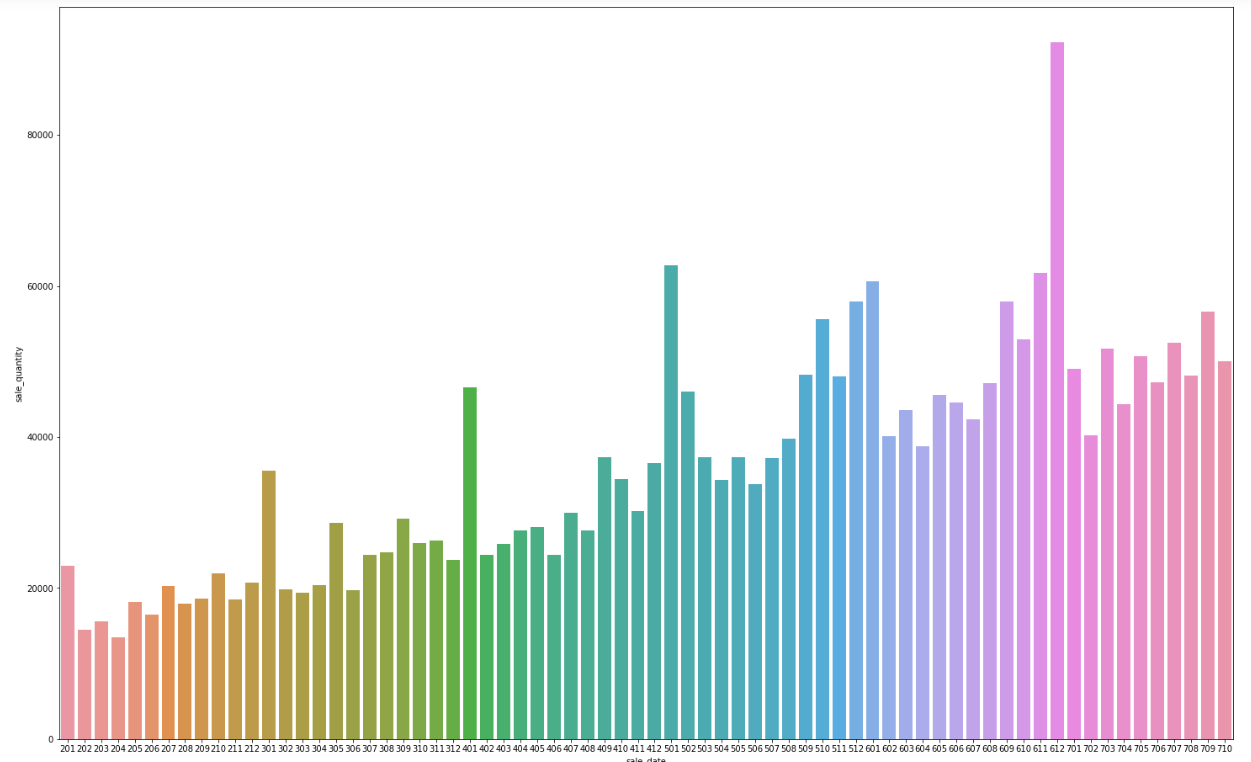
通过QQ图，QQ图直线截距是均值，斜率是标准差，可以看到那么大一个弯，完全就是个曲线，相去甚远，根本不是正态分布。

我们再查看一下偏度和峰度，可以明显看到偏度很大，峰度特别大，都是正数，是明显右偏并且特别尖。

可以用log1p平滑，预测后再用expm1转换回来。

那么说一下为什么我们要做这样一个平滑呢？有一些模型用到了大数定律和中心极限定理，有很多甚至是直接在正态分布的前提下做的推导。当然并不是说我们不平滑就不能用了，平滑一下可以在一定程度上降低误差。

接下来我们按照sale\_date分组，对销售量sum聚合，用条形图看了一下每个年月的总销售量的分布情况，如图所示。



可以看到15年到17年的波动是很相近的。

然后还可以看到，一般都是每年1月过年前后买车特别多。

但是有一个特例，就是2016年的12月特别多，2017年1月并没那么多。

而这显然是时间的原因，肯定是这个时间发生了什么事，于是我们百度搜素，发现是政策原因，有个什么优惠政策2016年12月31日就结束了。

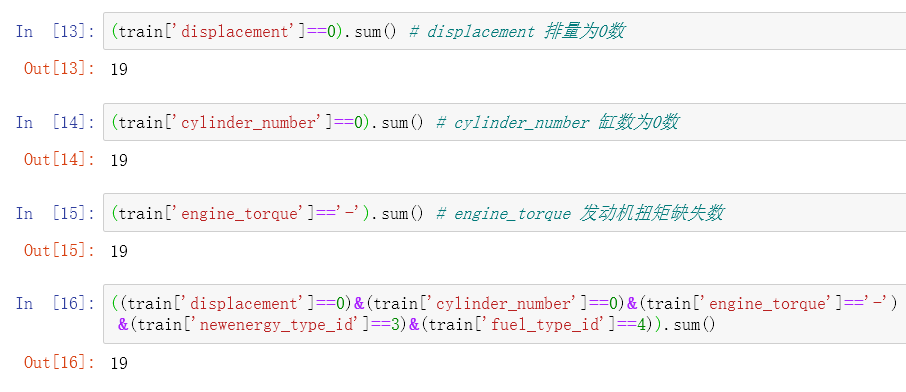
201612和201701是明显异常的，我们考虑利用车的特征进行训练预测重新填充这两个月的销量数据。

对于最终对201711和201712的预测。

我们考虑首先按class\_id+车的各种配置分组，唯一决定一辆车，然后拆分sale\_date特征，得到这个车在每个年月的销量的特征，没有就是0，然后预测这些唯一的车在接下来两个月的销量，最后按class\_id分组对销量求和就是最终要的结果了。

## 2.4 缺失值分析

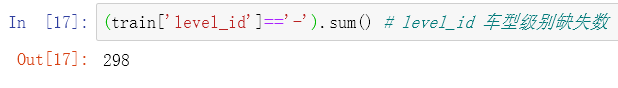
我们来进一步分析那些为0和横杠的数据。



首先关于排量、缸数以及发动机扭矩。

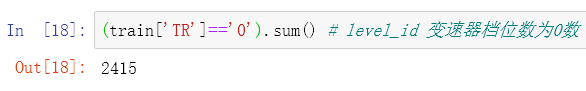
我们发现这几个0和横杠的数据行是完全对应的，并且对应新能源类型 3，燃料类型 4。经过研究以及查询相关资料，我们发现燃料类型4是纯电动燃料，纯电动，就只有电动机，没有发动机，所以没有发动机扭矩，是横杠。不需要内燃机，所以缸数是0，不烧油，排量就是0。由于发动机扭矩是数值特征而非标称特征，我们将横杠转换为0。

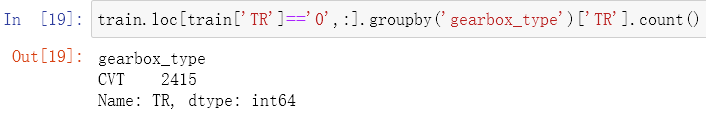
接下来是车型级别。



通过查询资料，我们了解到车型级别的评定只取决于轴距、车长、排量，所以这个可以考虑直接用这三个特征以及已知的车型级别进行拟合，通过预测填充缺失值，模型可以考虑直接选用svm，大间距区分开类别即可。

接下来是变速器档位数。

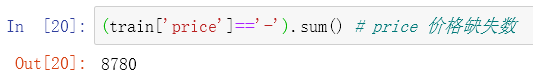




变速器档位数理应不能为0，所以我们考虑填充，而变速器档位数TR显然与变速器形式gearbox\_type是关联的，可以考虑按gearbox\_type分组填充众数。但是进一步研究发现，TR=0的全是gearbox\_type=CVT的，查询资料我们发现CVT是无级变速，无级变速没有换挡是平滑的加速，不存在确定的档位数，所以用'0'来表示档位数了。

于是这部分不是缺失，是单纯的没有无极变速档位数这个说法，作为数值大小肯定不太行，作为标识类别是完全没问题的，之后直接onehot就行，无需填充。

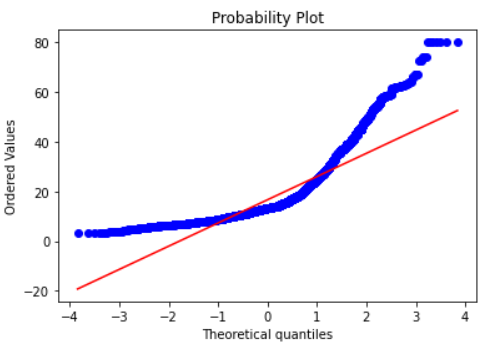
然后是成交价。



缺了很多，确实是缺了。

我们发现成交段一个缺失都没有，显然成交段和成交价是明显关联的，不过我们又发现成交价在成交段范围内还是有很大浮动，实际上成交价与车的各种特征都是有关的，但是拿出全部特征来预测这个价格完全没意义，所以我们还是考虑按成交段分组对成交价进行填充。

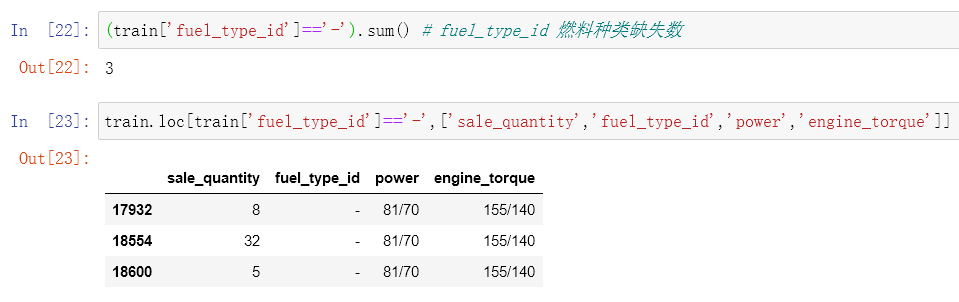
首先看一下成交价是否正态分布。



可以看到弯得离谱，显然不是正态。

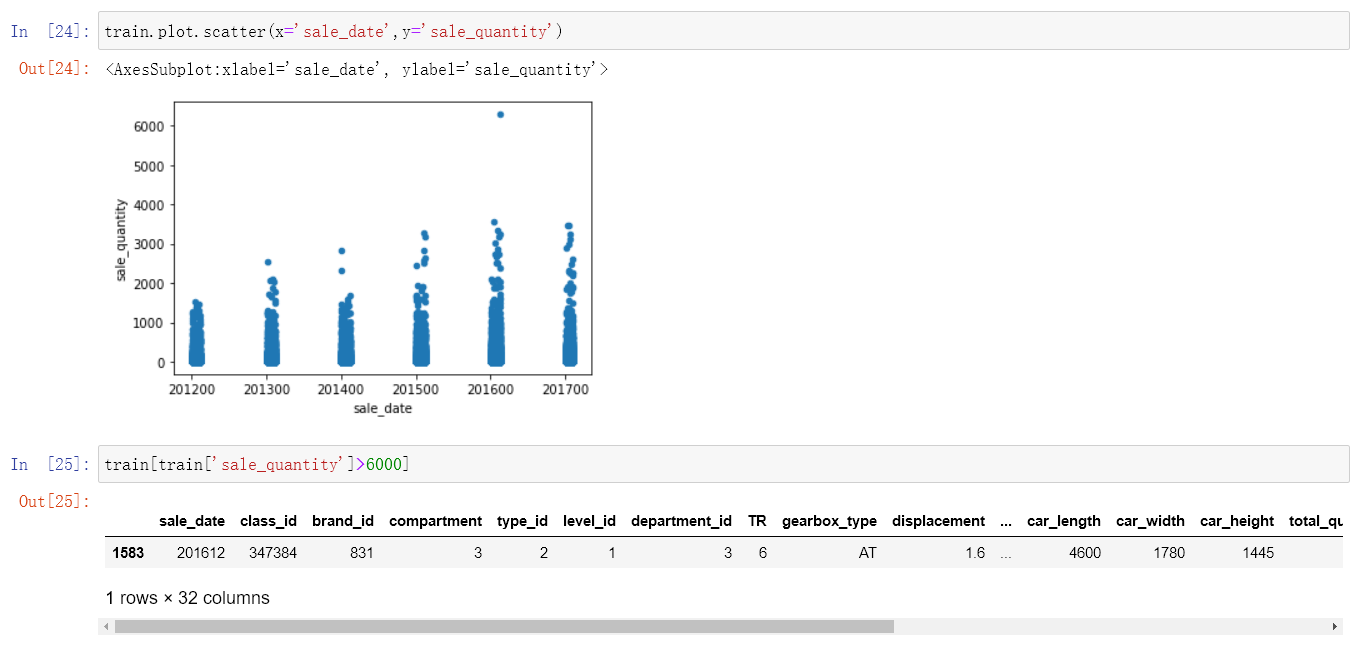
不是正态分布所以不用均值填充，使用中位数，所以成交价的填充策略就是按price\_level分组，取price的中位数对缺失值填充。

然后是燃料种类。

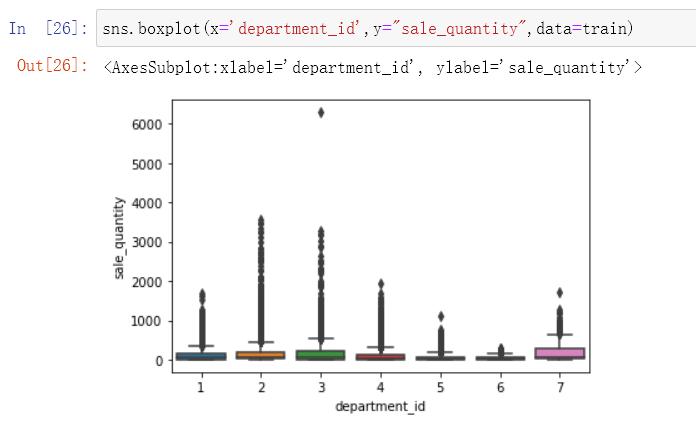


可以看到就缺了三条，然后经过研究发现这三条恰好又是在power马力和engine\_torque发动机扭矩上比较特别的三条，它的power和engine\_torque是多值的，查询相关资料，我们发现一个车不可能有两个发动机，所以也不会有两个马力，如果这个数据是两个发动机的意思，那问题很大，但也可能是两种发动机配置其他配置相同的车的销量求和合并得到的结果。但是这样又该怎么处理呢？我们不可能保留这种多值的，它被单独视作一个类别是不合理的，而且燃料类型该怎么填充也是个谜，缺失的分作一类也是有点不合适。拆开的话销量该怎么分，对半分合适吗，种种问题都是不确定的。而比较明确的一件事就是这三行销量特别特别少，所以我们决定直接删掉这三行，没大影响。

## 2.5 异常值分析

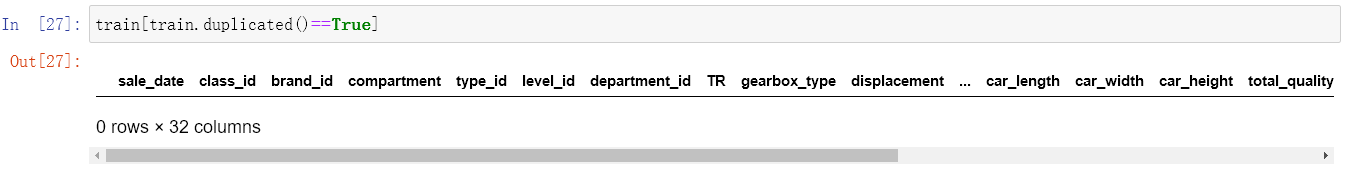


可以最上面那个点的销量明显异常，考虑重新预测填充，但是发现恰好在201612，我们本来就打算对201612重新预测填充，所以就正好不用单独处理这个点了。



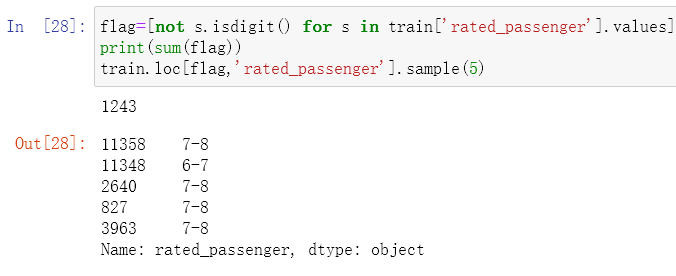
然后我们本来打算通过专门分析异常值的箱线图来分析下异常值，但以这个图为例，箱线图看到的结果就是基本全是异常值。数据确实分布得太差了，但是你给它全删了你还预测个啥，所以对于本项目通过箱线图来看是没啥意义的，就不进一步分析了。

## 2.7 冗余值分析



没有任何一行是完全冗余的，实际上就算冗余了也不能按往常的手法给删掉它，因为你最终结果要的销量是时序问题下的分组加和，不是单纯地你按车辆配置预测一行的销量。你不能随便去删掉一行。如果冗余顶多销量求和聚合一下，但是就算不合并也没啥差别。所以就是按照常规分析步骤简单看一下情况。

## 2.8 其他探索性分析



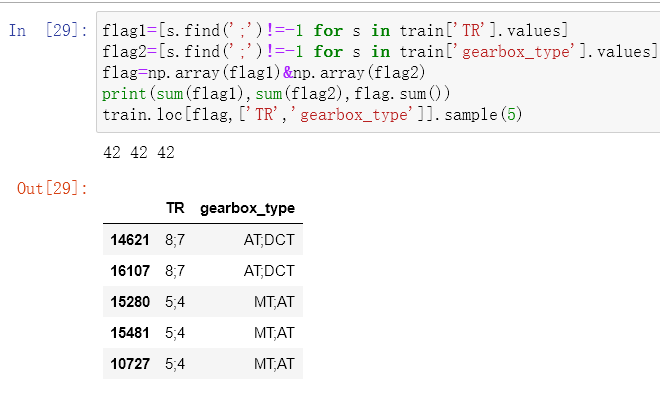
发现载客量有范围型的

载客量这里虽然是标称特征，但实际上也有一定数值特性

做onehot的同时最好也保留原来的数值特性

范围型载客数发现都是差1人

考虑直接平均数 0.5



发现有变速器类型和档位一一对应的多值数据。

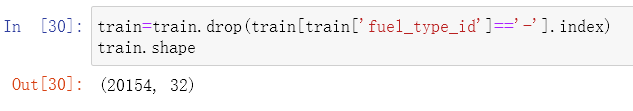
通过研究，我们发现是手自一体，不算多值，是单独的类别。

所以问题不大，保持现状，之后直接onehot即可。

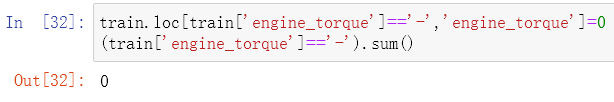
## 2.9 数据清洗

按照之前的分析，进行逐步清洗。

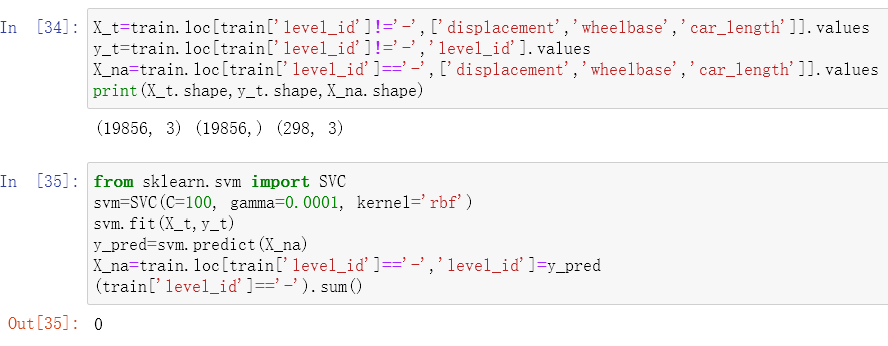
燃料种类缺失的三行直接删除。



发动机扭矩横杠转为0。



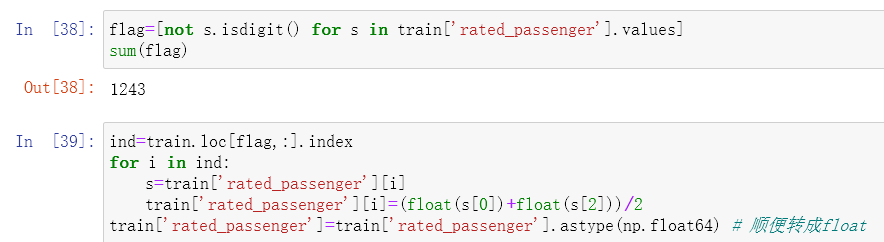
通过svm利用排量、轴距、车长预测填充缺失的车型级别。



按成交段分组，用中位数填充缺失的成交价

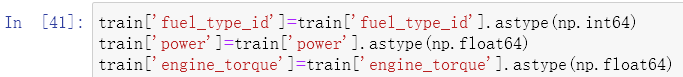


将范围型载客量取均值，并顺便转为float

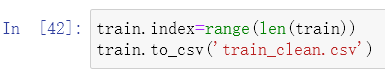


接下来对一些本应是数值特征以及数值大小有意义的标称特征把object数据类型转换为数值类型，其中比较有意思的是我们发现fuel\_type\_id有int和str混合的情况，1,3, ‘1’, ‘3’ 这样。

对于其他object类型直接onehot即可，没必要转换数据类型。



数据探索到此结束，清洗干净后的数据保存为csv，以备后续使用。



# 3.特征工程

## 3.1 处理标称特征

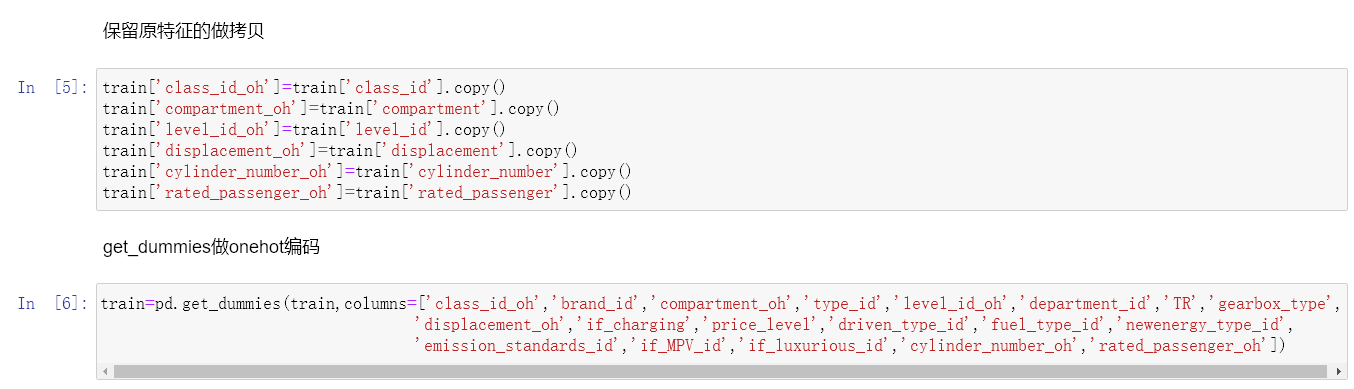
直接onehot的特征：brand\_id、type\_id、department\_id、TR、gearbox\_type、if\_charging、price\_level、driven\_type\_id、fuel\_type\_id、newenergy\_type\_id、emission\_standards\_id、if\_MPV\_id、if\_luxurious\_id。

class\_id需要做主码属性，所以拷贝一份做onehot。

厢数、车型级别、排量、缸数、载客量的大小有一定的意义，所以保留原特征，拷贝一份做onehot。

比较棒的是有price\_level，数值特征处理时就不需要price增加分段onehot特征了，直接price\_level的onehot就行了。

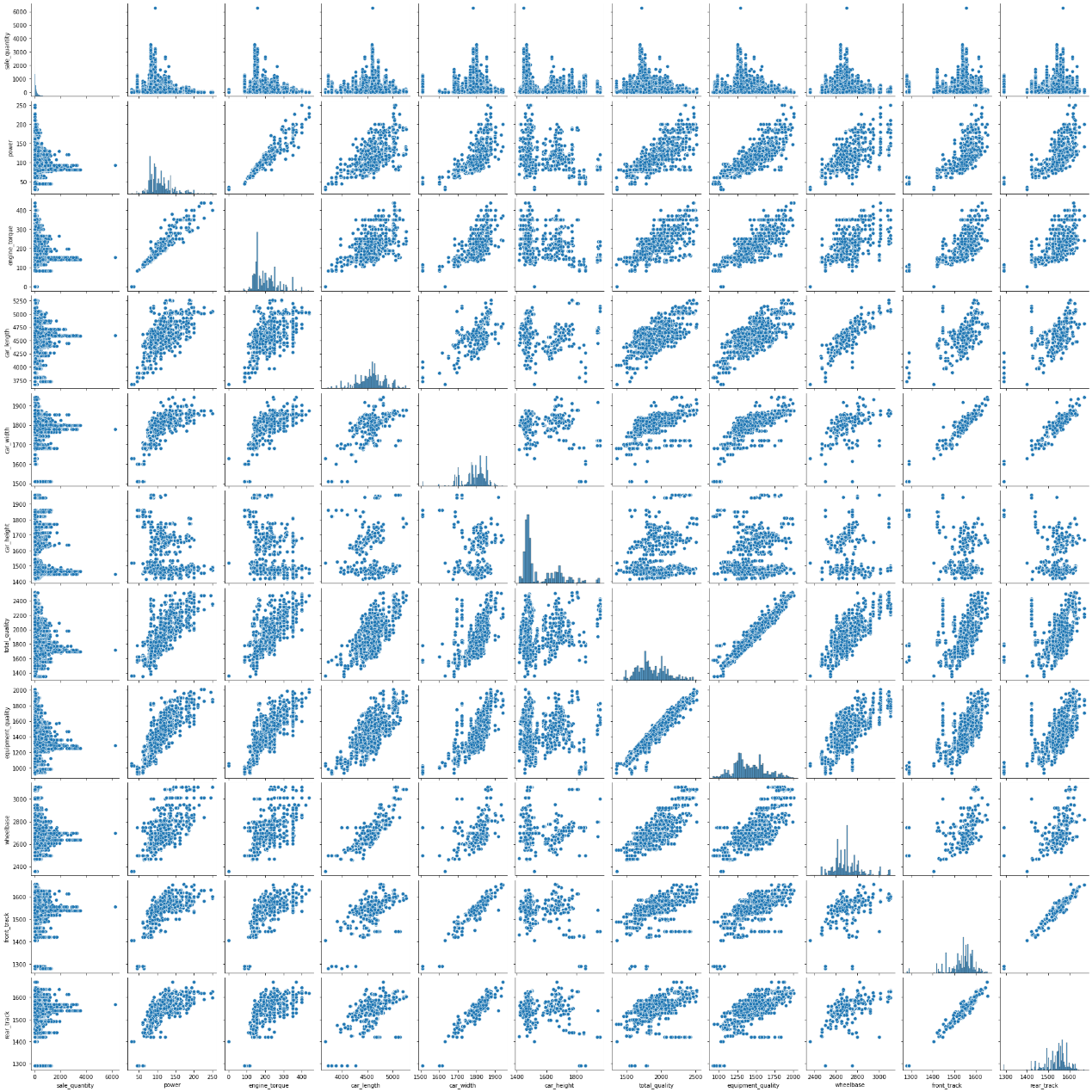
对于标称特征，做onehot最大的好处就是可以彻底消除只用于标识类别的标称量中的数值大小的影响。这是LabelEncoder直接离散化编码所不具备的。



## 3.2 处理数值特征

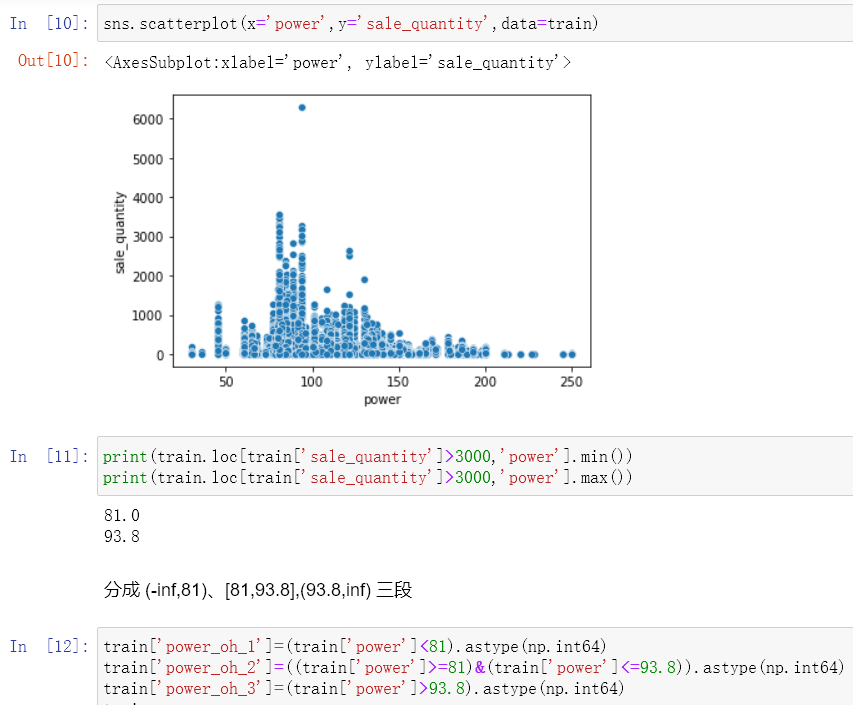
对于数值特征，一个比较常见的处理手段就是划分成几段，增加新的分段onehot特征，这样做的好处是，一是有时候数值特征的数值相关性很弱，反而划定范围后能明显看到标识类别的效果，二是有时候预测结果可能用分段函数可以更好的拟合，不妨设想把分段函数合并成一个函数，这是很难的一件事，对于这种明显有分段特性的情况，你拿一个函数去拟合，拟合起来很困难很复杂，并且欠拟合和过拟合都会更容易发生。当然我们只能是一个假说函数，我们不能写成分段的形式，那样就是多个函数了，但是可以增加一定的特征来实现分段函数的合并，我们只需要加入这样的划分范围分段的onehot特征，这些0/1二值属性可以起到分段函数的效果。

首先我们先pairplot整体观察一下所有这些数值特征。

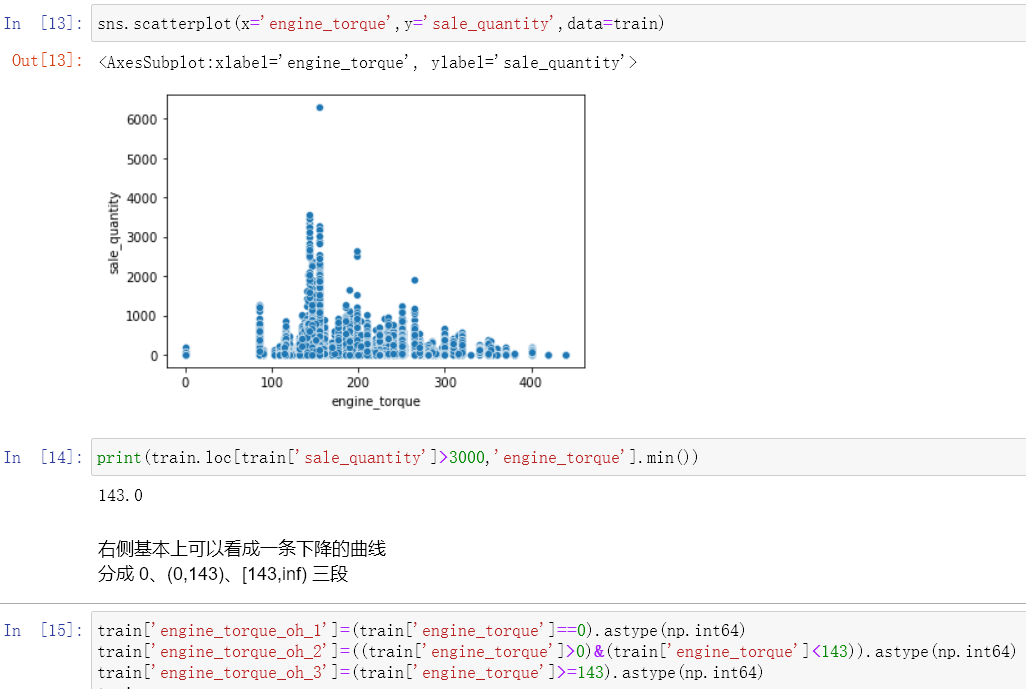


首先可以看到这些特征的分布都是和正态有较大偏差的，有必要平滑。其次可以看到和销量间的关系基本上都是中间高、两边低。其中 car\_length、car\_width、rear\_track、front\_track 这几个特征波动相对大一些。观察数据探索时的相关性矩阵，我们发现这几个特征和销量间的相关性最多不超过0.02，而且我们前面也探讨过这个问题，要知道他们还是数值型变量，所以这几个量的相关性实在是太差，处理的价值都不是很大，不处理对预测效果的影响估计也不好，所以我们决定直接删掉这四个特征。然后看wheelbase，虽然也是中间高两边低，但是中间看到只有两竖列的散点高出来，而且两竖列中间还凹下去一列，如果不看高出来的那一小部分，感觉就是一坨放在那，变化幅度不是很明显，再看它和销量的相关性0.04，在数值特征中仅高于前面说的那四个，所以我们决定直接删掉wheelbase这个特征。

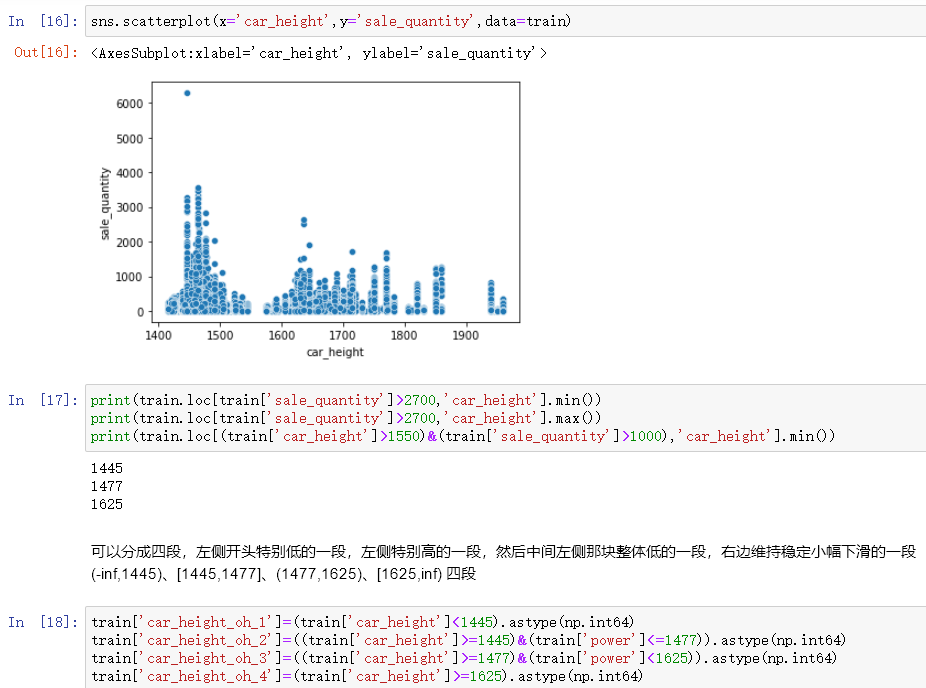
然后接下来对每个需要分段的数值特征我们进一步分析其分段点。



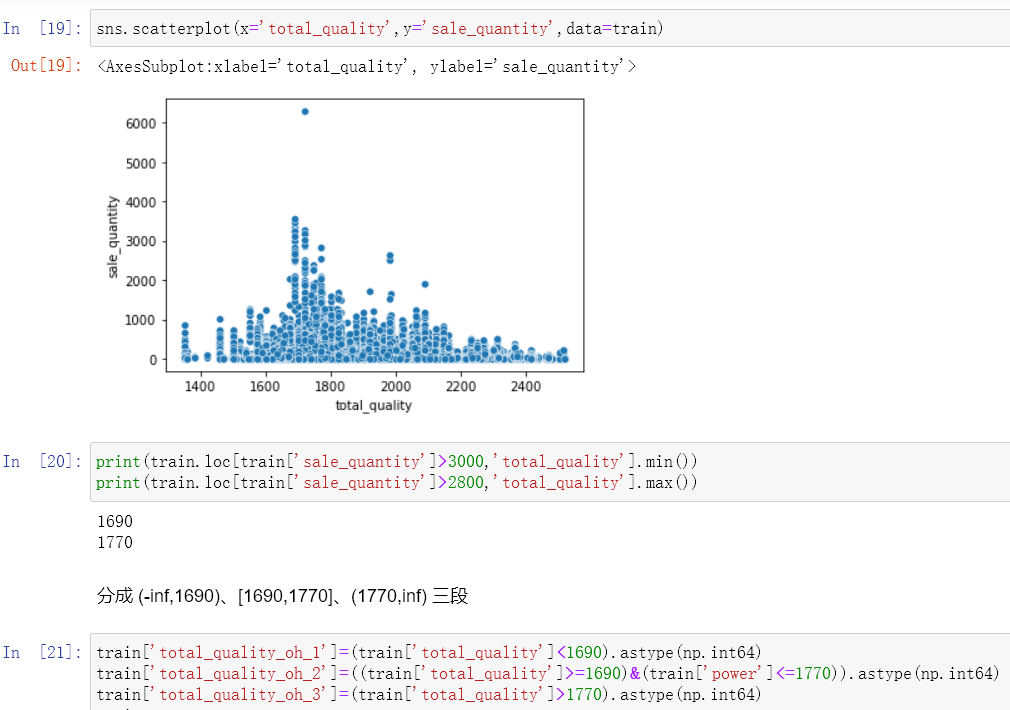
对power，我们分段成(-inf,81)、[81,93.8],(93.8,inf)三段



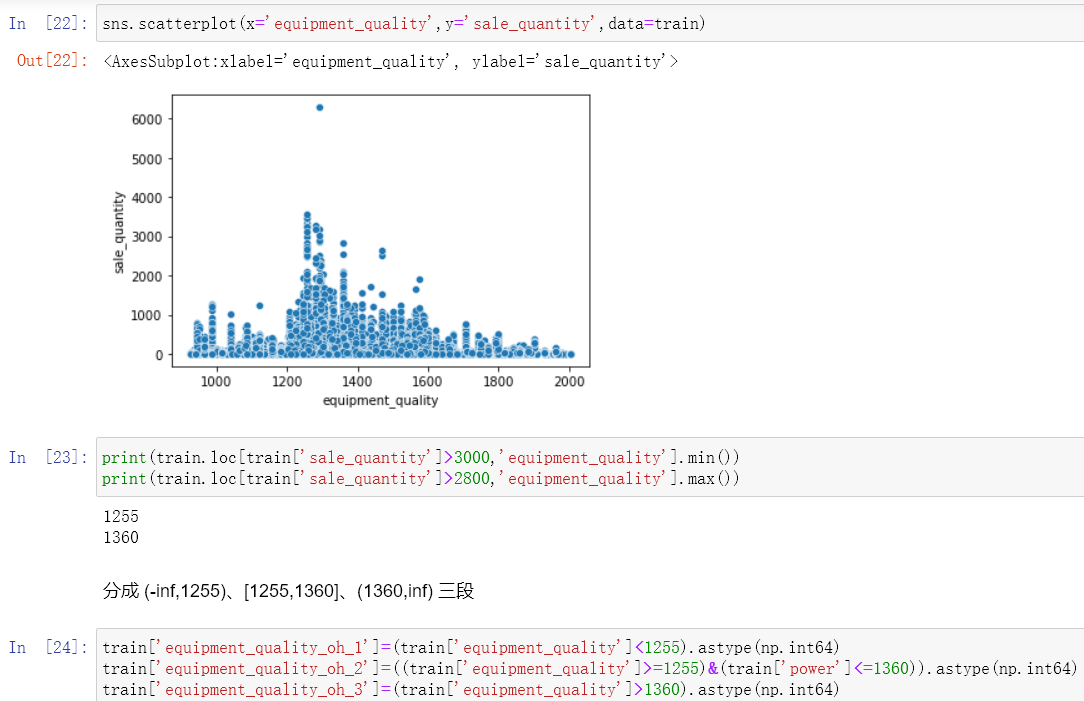
对engine\_torque，我们分段成0、(0,143)、[143,inf)三段。



对car\_height，我们分段成(-inf,1445)、[1445,1477]、(1477,1625)、[1625,inf]四段。



对 total\_quality，我们分段成(-inf,1600)、[1690,1770]、(1770,inf)三段。

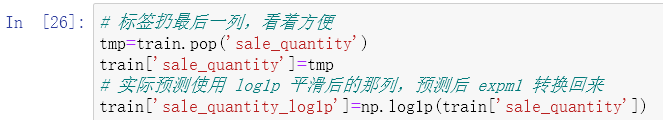


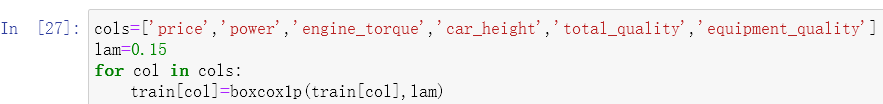
对equipment\_quality，我们分段成(-inf,1255)、[1255,1360]、(1360,inf)三段。

接下来处理sale\_date。不妨设想假如现在的实际情况是每过一个月涨100台，那么销量关于现在的sale\_date值会拟合成怎样的曲线。如果是从201611到201612，看起来没啥问题，而如果是从201612到201701呢？201611、201612、201701是连续的三个月，然而201701-201612=89，如果让你表示出它们的连续性，你会怎么表示呢？这很难，基本上你能想到的办法就是整数除以及取模获得年和月，然而哪有模型会拟合出带整数除和取模的假说函数呢？你这样的sale\_date放在这里就是不合理的。可以考虑的方案一是拆出年和月两列特征，二是干脆转换成月数，比如201201是1，那201301就是13。那么选哪个，肯定是第一个。不妨设想，1年真的和12个月等价吗？这是一个很有意思的问题，看起来就像在问1+1真的等于2吗，然而实际上这两个问题也仅仅是形式上相近。当你的前提还停留在按一定规律逐月变化时，那1年和12个月确实等价，如果从201612到201701骤变呢，月数做特征能很好地去拟合这个信息吗，那样的曲线将会很复杂。当然不论是最初的六位数，还是最后谈到的月数，我们都不能说它不可行，但是很显然拆成年、月两列明显能更好地表示出有关时间的信息。月份上的连续性变化，以及同月不同年，因年份变化而带来的变化。结合我们的可视化分析，可以看到销量的按月波动整体按年来看有着近似性，而且有着明显的年与年间的整体变化。然后就是sale\_date原特征我们保留了，因为要做主码属性。



然后之后我们对除了sale\_date和sale\_quantity以外的数值特征都做了boxcox1p平滑。然后对sale\_quantity做了log1p平滑。

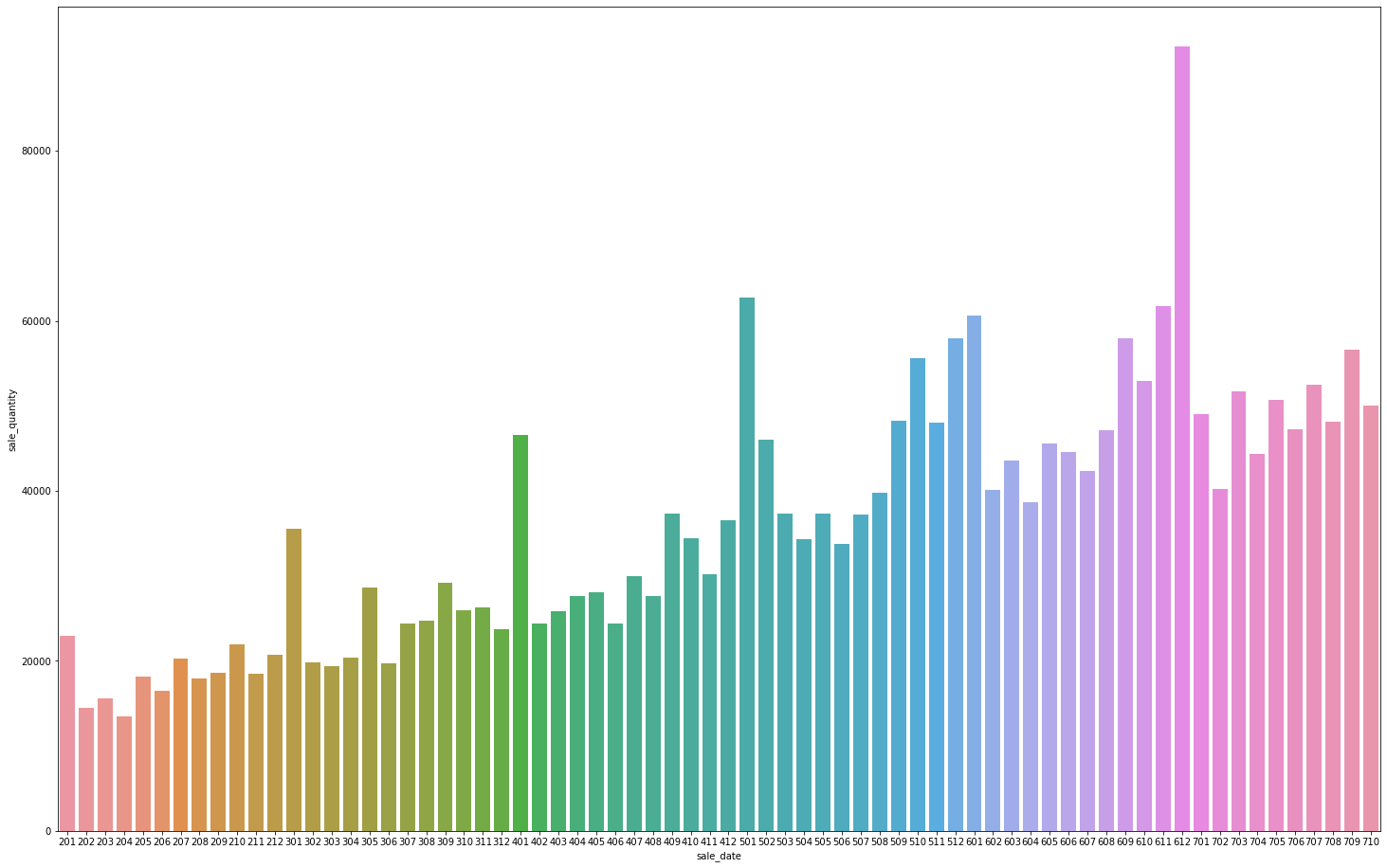




## 3.3 通过xgboost预测重新填充201612、201701的销量数据

在构造完这些特征后，我们就可以先预测一次，把201612、201701重新填充，这个预测任务和最终的预测任务不同的是，这个预测一行一行的车辆特征是完全已知的，而最后的预测任务核心是时序。

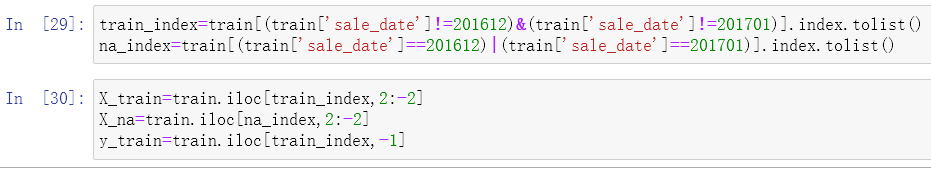
我们再次看一下按date的销量分布图。



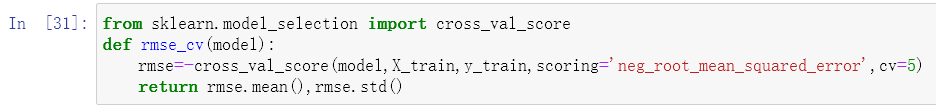
可以看到201612和201701确实很反常。

我们决定直接选用xgboost预测，这个预测填充的任务只需要稳定就好。我们不需要它特别好，但是不能差，最重要的是稳定的好。根据我们的个人经验，xgboost是比较合适的。

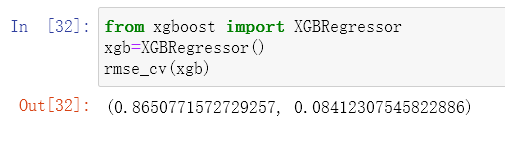
用不在201612、201701的数据做训练，如图所示。



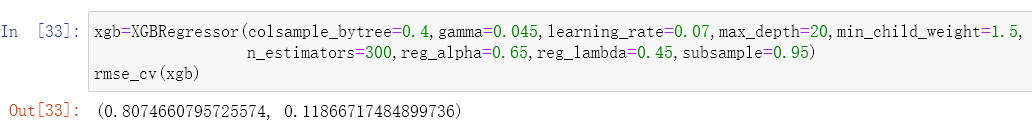
定义了和题目要求的误差计算方式一致的rmse的五折交叉验证函数，其评分用于调参，如图所示。



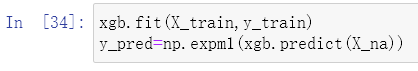
默认参数的评分。

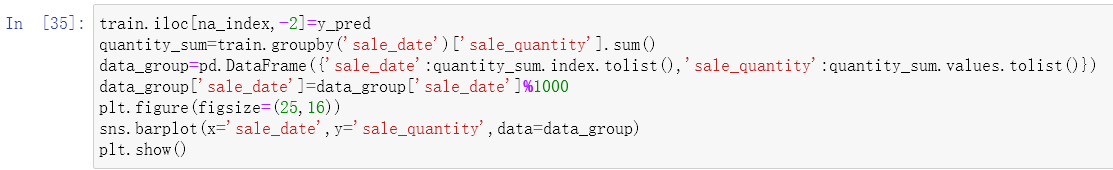


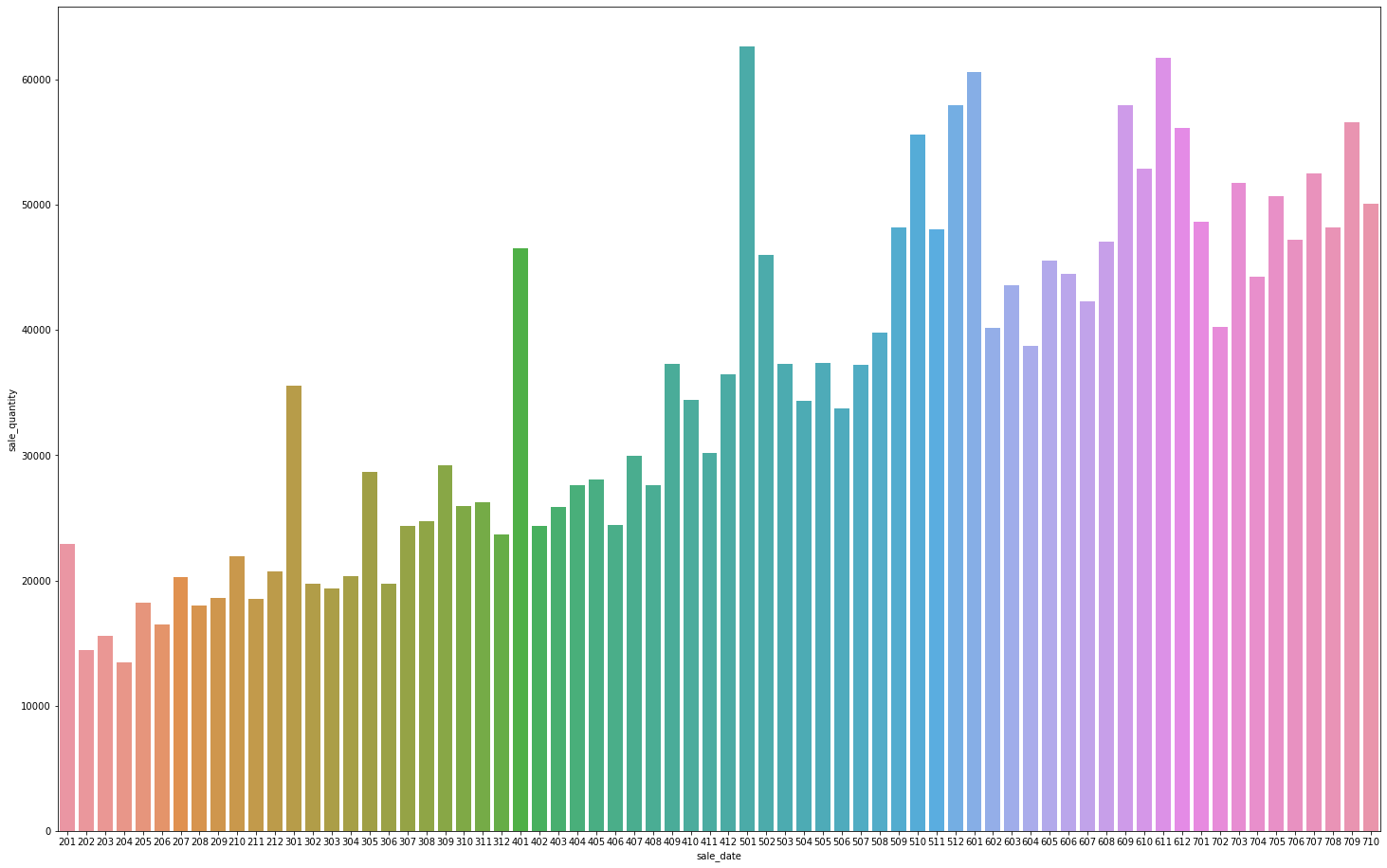
然后通过网格搜索和手动瞎试我们调了一波参，具体调参过程在此省略。



对异常的部分预测重新填充，并观察效果。

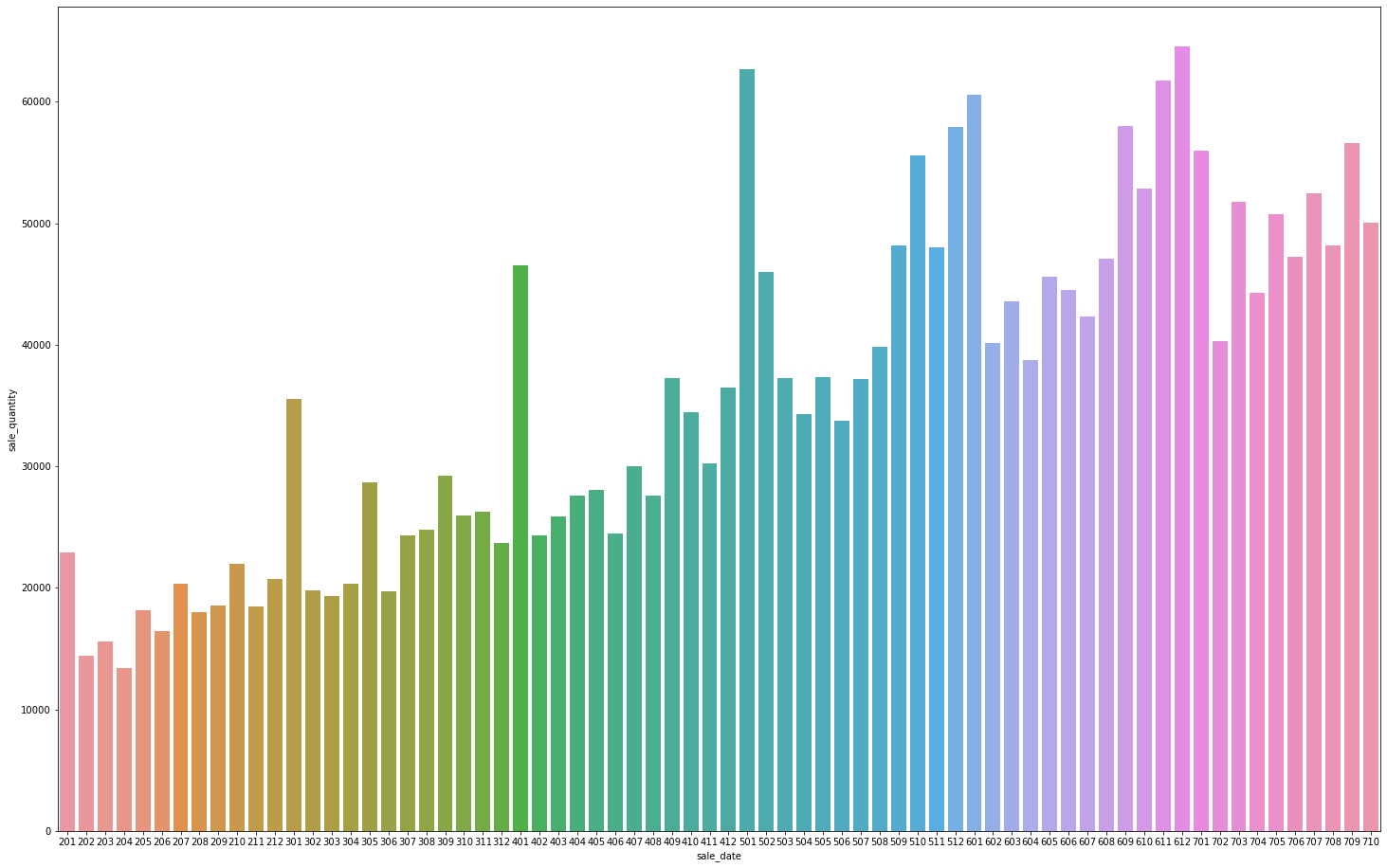






可以看到感觉预测得似乎有点过于保守了，可以手动整体倍乘扩大一下。而且201701是不是预测得有点太少了呢？往年的1月全都是特别高。而且这时候我们又意识到了一个问题，201611是不是也有点多？然后我们观察到数据中很多'大小月波动'，连续的涨一个月降一个月。2016的波动看起来像是反着来的，而且2016整体确实都多一点，把201611和201510放在一起横向比较来看，感觉其实201611的也没那么多，看起来挺正常的。2016和2017的波动也很相似。按照这个波动，201701比201612小一点点也可以认为是正常的。那么这两个月的预测是不是太保守有点小呢？我个人认为是，我最终将这两个月的预测结果扩大了1.15倍进行填充，我认为这是合理的。

画出的条形图就是这样的。



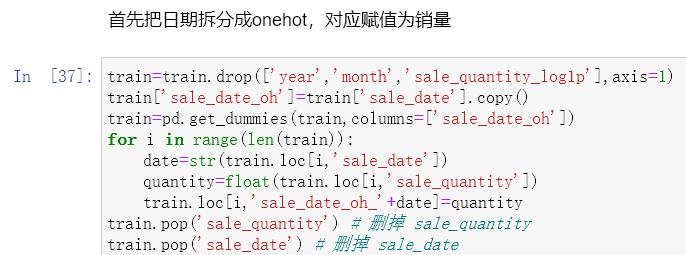
感觉正常多了。

## 3.4 分组，引入时间特征

为了预测接下来的两个月，我们要进一步引入时间特征。

对于201711和201712的预测任务，其特点是，不是单纯的根据车辆特征预测，因为没有确切的说卖了哪些车，哪些具体配置的车。所以基于时序的历史销售信息就很重要了。

我们的思路就是，拆分sale\_date特征，每个月的销量单独做一列特征，因为class\_id+各种配置唯一决定一辆车，那就按class\_id+各种配置分组，sum聚合合并得到其在每个年月下的销量，这样就是预测这些唯一的车辆在11、12月会卖多少，然后最后按要求按class\_id分组sum聚合即可。而这时候原来的year、month、sale\_date、sale\_quantity特征也都不需要了，直接drop掉了。这部分具体处理手法如下图所示。



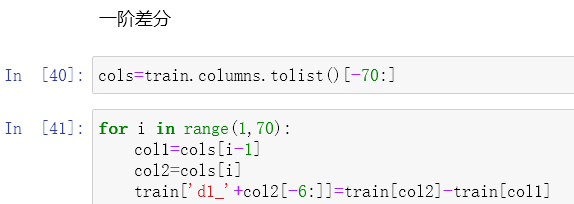


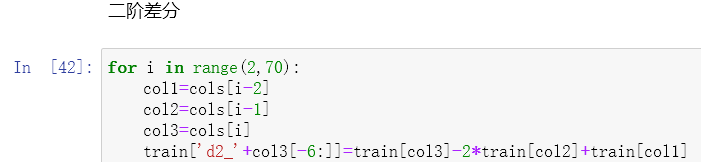
然后我们要强化其时序信息，新增特征。

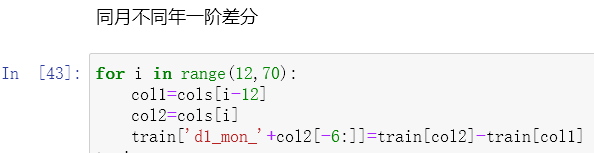
由于是预测接下来两个月，考虑引入一阶差分和二阶差分。

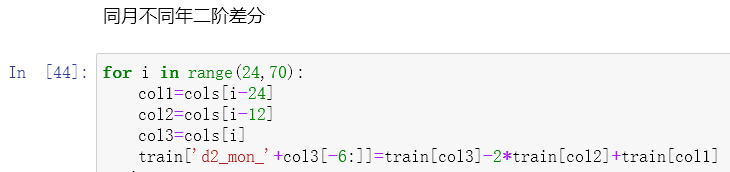
由于年度波动近似，考虑引入同月不同年的一阶差分和二阶差分。

本来还想引入环比和同比，但是我们拆分的按销售量矩阵过于稀疏，不适合做除法计算。

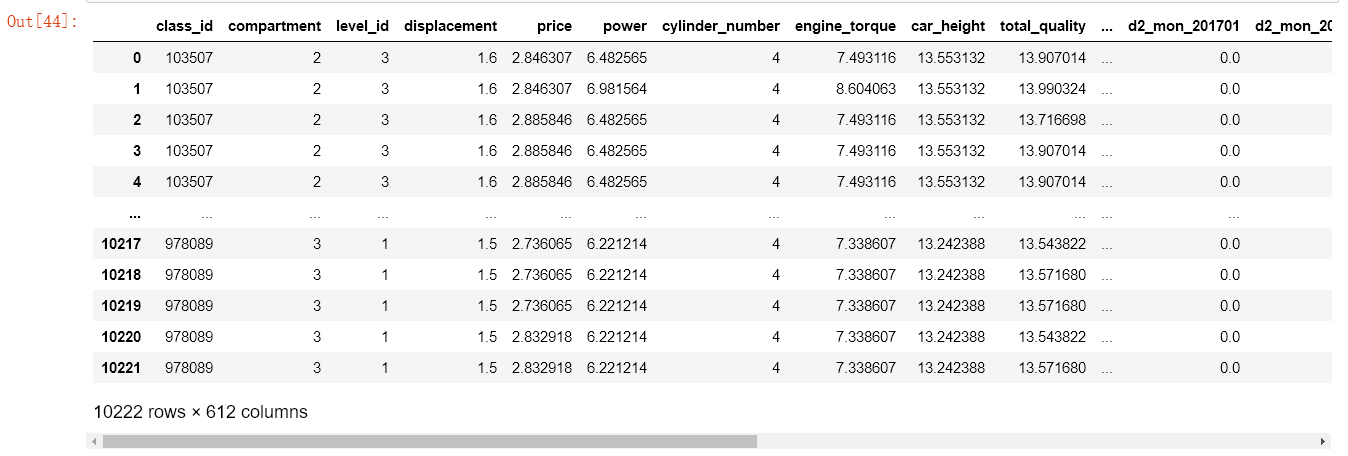








最终得到的就是这样一个10222\*612的数据，当然在具体的预测任务之前还要做一些对应性的调整，这个在之后的模型构建部分我们会谈到。



我们将其保存为csv，以备后续使用。

# 4.模型构建

## 4.1 划分特征和标签、训练集和测试集

对于标签，我们考虑用波动程度近似的15年和16年的销量做标签来进行训练，来预测17年。然后加入一列标签所在年份的特征。用201510、201610的销量做标签，预测201710的销量，进行本地验证。用201511、201611的销量做标签，预测201711的销量，提交A榜。用201512、201612的销量做标签，预测 201712的销量，提交B榜。由于利用交叉验证评分调参训练得太慢了太浪费时间了，所以我们只打算对其中一组预测进行调参。B 榜201712的数据预测难度相对更大，因为有201612这个异常的前例。所以我们主打A榜201711，我们调参都是基于201710和201711作标签的训练集。然后直接拿这个挑出来的参数和模型分别对其他数据集进行训练测试。在划分特征和标签的过程中需要注意的是，需要删除和训练标签相关的差分特征，原因很简单，你预测的标签是未知的，它和前后月间差多少是未知的，都知道了那加加减减都能算出来了，那还预测什么？然后最后的预测结果需要按照 class\_id 分组，对标签做 sum 聚合，还要记得把predict\_date补充上，按提交要求来。



上图是11月的训练集特征和标签，测试集的特征。



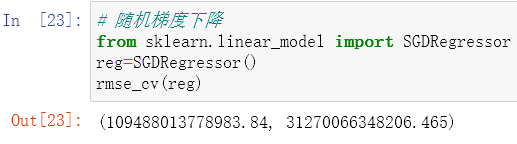
上图是12月的训练集特征和标签，测试集的特征。

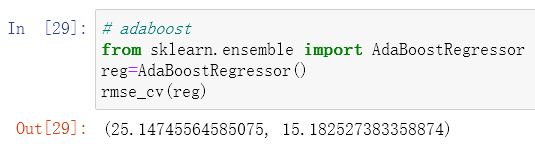


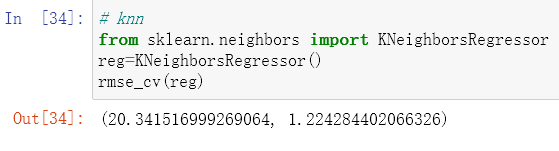
上图是10月的训练集特征和标签，以及本地验证集的特征和真实标签。

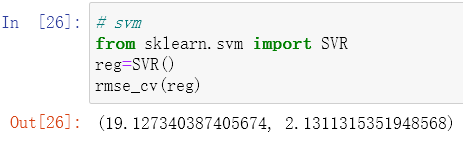
## 4.2 默认超参数观察各模型效果

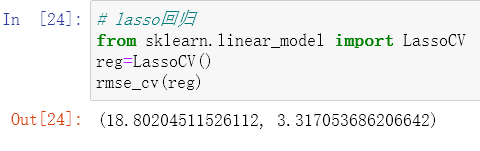
我们选取了线性回归、岭回归、随机梯度下降、lasso回归、弹性网络、svm、决策树、极限树、adaboost、bagging、极限随机树、梯度提升决策树、随机森林、knn、限定半径最近邻回归树、xgboost、xgboostrf、lgbm，共计18个模型。其中限定半径最近邻回归树RadiusNeighbors由于One or more samples have no neighbors而无法正常训练。其余17个模型按照五折交叉验证计算的rmse均值结果好坏升序展示如下。

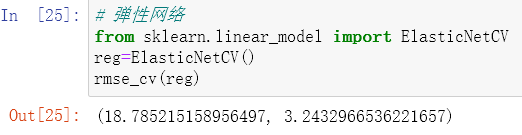


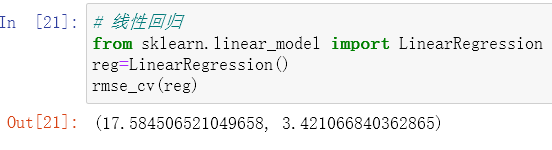


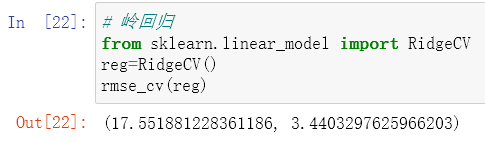


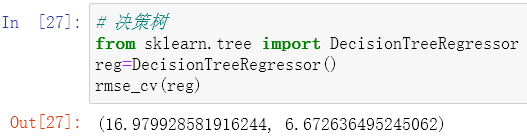


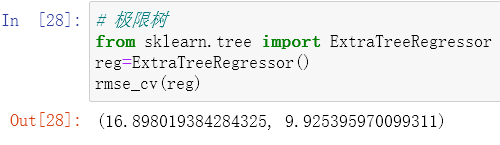


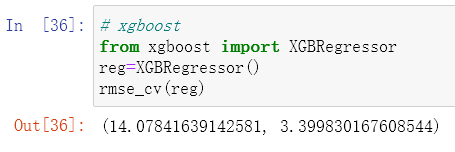


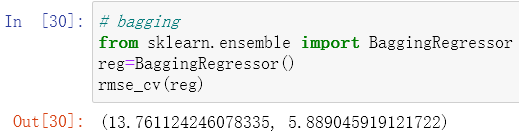


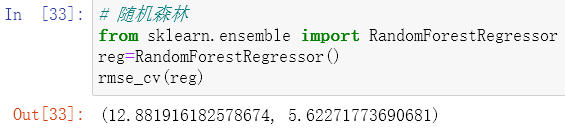


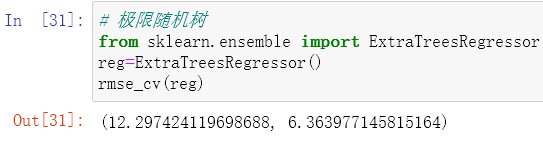


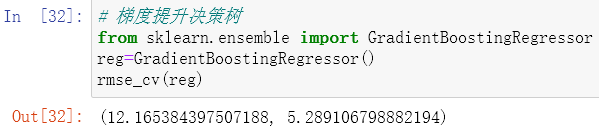


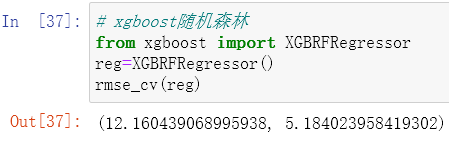


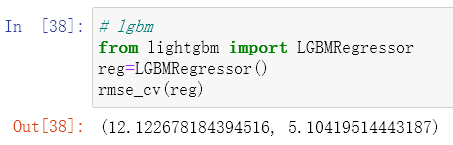












## 4.3 选取默认参数效果较优的三个模型进行调参

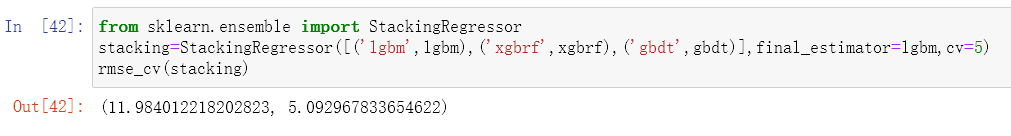
接下来，根据默认参数的效果，我们选取了lgbm、xgboostrf、gbdt这三个模型进行进一步调参。

我们的调参方式是网格搜素以及手动玄学瞎试，具体过程就省略了，这里仅保留最终的结果，如下所示。



## 4.4 模型融合

我们的模型融合是使用的是懒人方法stacking，基回归器使用的是lgbm、xgbrf和gbdt，元回归器使用的是lgbm。相比于对不同模型预测结果按百分比权重加和，stacking最大的缺点是训练太慢。我们的结果如下所示。

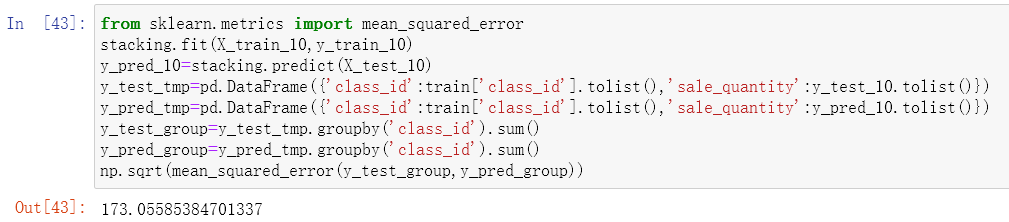


你可能会说这个融合后的怎么看起来比单独的调参后的lgbm和xgbrf还要差一点点呢。关于这个问题我以前也查过很多次，应该说stacking不会让预测结果出现明显变差的可能性，一般来讲都是变得更好，训练那么久也不是白训练的。这里我认为stacking应该消除了一定的过拟合，至少可以看到交叉验证评分的方差变小了一点。实际上这个结果还是不错的，至少比每调参的所有模型都要好。

## 4.5 本地验证

然后我们对201710的销量进行了本地验证。

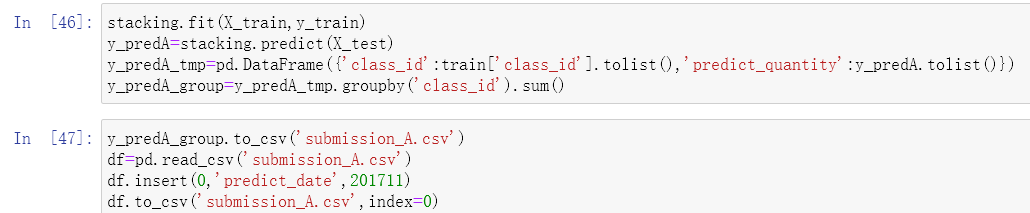
最终结果是按照要求的按class\_id分组sum聚合。

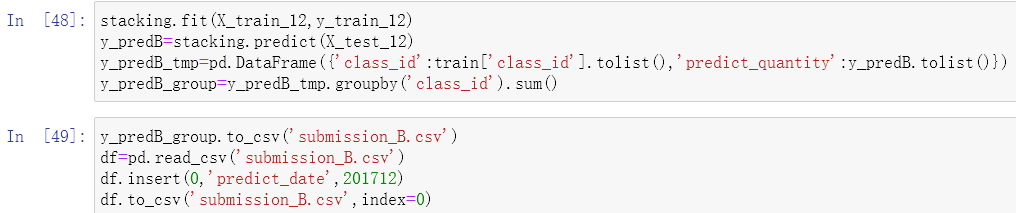


rmse173，很不错。10月的整体销量会比11月低一些，如果是对11月的评测应该rmse还会大一些，但是差别不会很大。我们这个验证结果是完全有参考价值的，我们对11、12月的预测和对10月的预测手法上是完全一样的，我们的验证评分方式也是完全按照要求来的没有问题。我们认为提交在线评测的结果应该会很不错，至少进复赛应该是很轻松的。

## 4.6 对天池测试集预测并保存为CSV

最后我们对测试集进行了预测，并将结果保存为了submission\_A.csv和submission\_B.csv，以备后续在线提交。





# 5.项目总结及分析

暂时还没有线上评测，到底做的好不好，还不太好说，个人感觉最后线上提交结果应该会不错，不会比本地的rmse=172高太多。

然后本项目是完全独立完成的，由于以前做过类似的新人赛，有一些经验，所以做得还是挺轻松的。个人认为，从数据探索到特征工程再到最后的模型构建，感觉做的都还不错，美中不足的是：本项目特征工程的处理总体而言还是有些套路化，没有什么出彩的地方，主要问题还是经验不足，比方说增加来自现实生活的相关特征，本项目我只能想到是否闰年 (感觉没啥用所以最后没加) 。解决方案就是不断积攒经验，对于本项目，我打算之后有空了以后去看一看其他参赛大佬是怎么做的，多积攒一些经验。然后是模型调参，我们的缺点是基本上就是瞎调+网格搜索，之后要好好去看一看各个模型怎么实现的，理解其原理和本质。