本文在收集天气温度数据已经发现了很多缺失值的情况，主要有空值，通常要根据实际业务情况来缺失值填补，比如用户所在省存在NAN，可以填补成“未知”。对于空值NAN当不影响特征含义的时候，可以填补成fillna(0)，即用0填补。如表3-4为7月份中一周的天气气温表，



其中2019年7月14日数据缺失，用拉个朗日插值对缺失值进行插补，关键核心代码如下：

代码清单4-1 拉格朗日法进行插补

1. #拉格朗日插值代码，index插入的位置，df为列向量，k为取前后的个数
2. **def** deal\_empty\_column(index, df, k=5):
3. **print**(index)
4. #取插补值索引前，后各5个数
5. y = df[list(range(index - k, index))
6. + list(range(index + 1, index + 1 + k))]
7. y = y[y.notnull()]
8. #调用拉格朗日插值函数
9. **return** lagrange(y.index, list(y))(index)

应用拉格朗日插值法算对表4-2中的缺失值进行插补，使用缺失值前后各5个未缺失的数据参与建模，得插值结果如下所示。



利用拉格朗日插值对这2015/2/21和2015/2/14的数据进行插补，结果是4275.255和4156.86，这两天都是周末，而周末的销售额一般要比周一到周五要多，所以插值结果比较符合实际情况。

2　异常值处理

异常值分析是检验数据是否有录入错误以及含有不合常理的数据。忽视异常值的存在是十分危险的，不加剔除地把异常值包括进数据的计算分析过程中，对结果会产生不良影响；重视异常值的出现，分析其产生的原因，常常成为发现问题进而改进决策的契机。

异常值是指样本中的个别值，其数值明显偏离其余的观测值。异常值也称为离群点，异常值的分析也称为离群点分析。

（1）简单统计量分析

可以先对变量做一个描述性统计，进而查看哪些数据是不合理的。最常用的统计量是最大值和最小值，用来判断这个变量的取值是否超出了合理的范围。如客户年龄的最大值为199岁，则该变量的取值存在异常。

（2）3σ原则

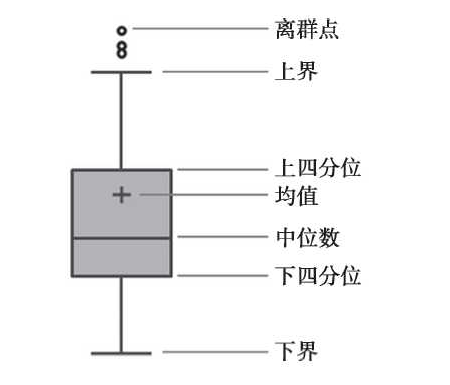
如果数据服从正态分布，在3σ原则下，异常值被定义为一组测定值中与平均值的偏差超过3倍标准差的值。在正态分布的假设下，距离平均值3σ之外的值出现的概率为P（｜x-μ｜>3σ）≤0.003，属于极个别的小概率事件。

如果数据不服从正态分布，也可以用远离平均值的多少倍标准差来描述。

（3）箱型图分析

箱型图提供了识别异常值的一个标准：异常值通常被定义为小于QL－1.5IQR或大于QU＋1.5IQR的值。QL称为下四分位数，表示全部观察值中有四分之一的数据取值比它小；QU称为上四分位数，表示全部观察值中有四分之一的数据取值比它大；IQR称为四分位数间距，是上四分位数QU与下四分位数QL之差，其间包含了全部观察值的一半。

箱型图依据实际数据绘制，没有对数据作任何限制性要求（如服从某种特定的分布形式），它只是真实直观地表现数据分布的本来面貌；另一方面，箱型图判断异常值的标准以四分位数和四分位距为基础，四分位数具有一定的鲁棒性：多达25％的数据可以变得任意远而不会很大地扰动四分位数，所以异常值不能对这个标准施加影响。由此可见，箱型图识别异常值的结果比较客观，在识别异常值方面有一定的优越性，如图3-1所示。



在数据预处理时，异常值是否剔除，需视具体情况而定，因为有些异常值可能蕴含着有用的信息。异常值处理常用方法见表4-3。

表 3 产品表

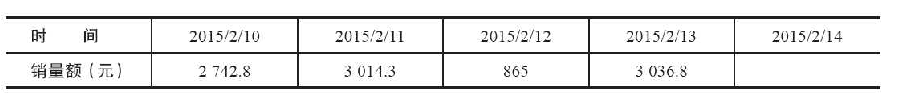
|  |  |
| --- | --- |
| **异常值处理方法** | **方法介绍** |
| 删除含有异常值的记录 | 直接将含有异常值的记录删除 |
| 视为缺失值 | 将异常值视为确实值，利用缺失值处理的方法进行处理 |
| 平均值修正 | 可用前后两个观测值的平均值修正该异常值 |
| 不处理 | 直接在具有异常值的数据集上进行挖掘建模 |

将含有异常值的记录直接删除的方法简单易行，但缺点也很明显，在观测值很少的情况下，这种删除会造成样本量不足，可能会改变变量的原有分布，从而造成分析结果的不准确。视为缺失值处理的好处是可以利用现有变量的信息，对异常值（缺失值）进行填补。

在很多情况下，要先分析异常值出现的可能原因，再判断异常值是否应该舍弃，如果是正确的数据，可以直接在具有异常值的数据集上进行挖掘。

在数据集的订单车辆表中，还车的经纬度可能会出现缺失值和异常值，如表3-1中数据所示：

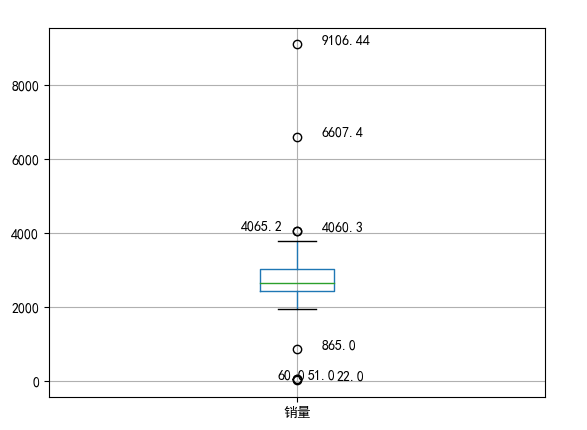
表3-1 订单车辆表数据示例



从数据中可以发现，其中有部分数据是缺失的，因为数据记录和属性较多，使用人工分辨的方法是不切实际的，本文这里使用编写程序来检测出含有异常值以及缺失率。

在Python的Pandas库中，只需要读入数据，然后使用describe（）函数就可以查看数据的基本情况。其中count是非空值数，通过len（data）可以知道数据记录为10000000条，因此缺失值数为700。另外，提供的基本参数还有平均值（mean）、标准差（std）、最小值（min）、最大值（max）以及1/4、1/2、3/4分位数（25％、50％、75％）。更直观地展示这些数据，并且可以检测异常值的方法是使用箱线图。其核心关键检测代码如代码清单3-1所示

1. #以下参数都是经过调试的，需要具体问题具体调试
2. **for** i **in** range(len(x)):
3. # 处理临界情况， i》0时
4. **if** i>0:
5. plt.annotate(y[i],xy=(x[i],y[i]),xytext=(x[i] +0.05-0.8 / (y[i] - y[i - 1]), y[i]))
6. **else**:
7. plt.annotate(y[i],xy=(x[i], y[i]), xytext=(x[i] + 0.08, y[i]))
8. plt.show()



从图3-2中可以看出，箱型图中的超过上下界的7个销售额数据可能为异常值。结合具体业务可以把865、4060.3、4065.2归为正常值，将22、51、60、6607.4、9106.44归为异常值。最后确定过滤规则为：日销量在400以下5000以上则属于异常数据，经分析，这些异常数据可能是程序发布版本或者第三方网络差小概率情况造成的异常数据，结合本数据集数据量比较大，异常数据相对比较少（1000000>235），且异常数据的值不具有任何意义，所以这里针对异常数据采取的办法是：编写过滤程序，删除这里异常数据。

1. 数据集成

数据挖掘需要的数据往往分布在不同的数据源中，数据集成就是将多个数据源合并存放在一个一致的数据存储（如数据仓库）中的过程。

4.2.1　实体识别

实体识别是指从不同数据源识别出现实世界的实体，它的任务是统一不同源数据的矛盾之处，常见形式如下。

（1）同名异义

数据源A中的属性ID和数据源B中的属性ID分别描述的是车辆编号和订单编号，即描述的是不同的实体。

（2）异名同义

数据源A中的order\_dt和数据源B中的order\_date都是描述销售日期的，即A.order\_dt=B.order\_date

（3）单位不统一

描述同一个实体分别用的是国际单位和中国传统的计量单位。

检测和解决这些冲突就是实体识别的任务。

4.2.2　冗余属性识别

数据集成往往导致数据冗余，例如，

1）同一属性多次出现；

2）同一属性命名不一致导致重复。

仔细整合不同源数据能减少甚至避免数据冗余与不一致，从而提高数据挖掘的速度和质量。对于冗余属性要先分析，检测到后再将其删除。

本文数据集都是用户骑行的数据存储到mysql中，针对用户骑行订单车辆表数据量大，按照订单号末位取模的策略进行了水平分表，水平分库分表能够有效的缓解单机和单库的性能瓶颈和压力，突破IO、连接数、硬件资源等的瓶颈。使用的时候需要进行union 将小表合并查询。如图下表

Order\_bike\_0

## 3.3 数据分析及可视化探索

### 4.2.1 特征工程的介绍

### 4.2.2 归一化

数据规范化（归一化）处理是数据挖掘的一项基础工作。不同评价指标往往具有不同的量纲，数值间的差别可能很大，不进行处理可能会影响到数据分析的结果。为了消除指标之间的量纲和取值范围差异的影响，需要进行标准化处理，将数据按照比例进行缩放，使之落入一个特定的区域，便于进行综合分析。如将工资收入属性值映射到[-1，1]或者[0，1]内。

数据规范化对于基于距离的挖掘算法尤为重要。

（1）最小-最大规范化

最小-最大规范化也称为离差标准化，是对原始数据的线性变换，将数值值映射到[0，1]之间。

转换公式如下：

其中，max为样本数据的最大值，min为样本数据的最小值。max-min为极差。离差标准化保留了原来数据中存在的关系，是消除量纲和数据取值范围影响的最简单方法。这种处理方法的缺点是若数值集中且某个数值很大，则规范化后各值会接近于0，并且将会相差不大。若将来遇到超过目前属性[min，max]取值范围的时候，会引起系统出错，需要重新确定min和max。

（2）零-均值规范化

零-均值规范化也称标准差标准化，经过处理的数据的均值为0，标准差为1。转化公式为：

其中x为原始数据的均值，σ为原始数据的标准差，是当前用得最多的数据标准化方法。

（3）小数定标规范化

通过移动属性值的小数位数，将属性值映射到[-1，1]之间，移动的小数位数取决于属性值绝对值的最大值。

转化公式为：

下面通过对一个矩阵使用上面3种规范化的方法处理，对比结果。其程序如代码清单4-2所示。

(data - data.min())/(data.max() - data.min()) #最小-最大规范化

(data - data.mean())/data.std() #零-均值规范化

data/10\*\*np.ceil(np.log10(data.abs().max())) #小数定标规范化

### 4.2.1 PCA

在大数据集上进行复杂的数据分析和挖掘需要很长的时间，数据规约产生更小但保持原数据完整性的新数据集。在规约后的数据集上进行分析和挖掘将更有效率。

数据规约的意义在于：

降低无效、错误数据对建模的影响，提高建模的准确性；

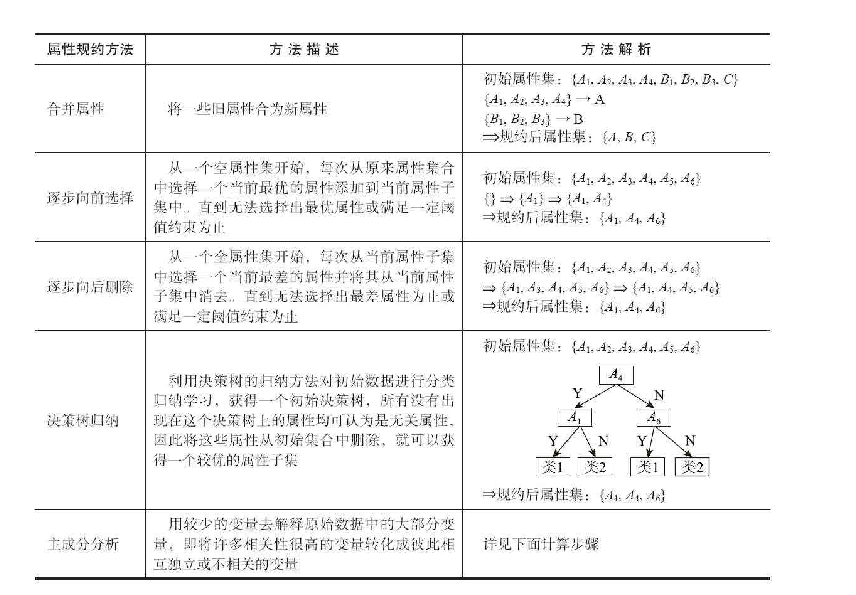
少量且具代表性的数据将大幅缩减数据挖掘所需的时间；

降低储存数据的成本。

4.4.1　属性规约

属性规约通过属性合并来创建新属性维数，或者直接通过删除不相关的属性（维）来减少数据维数，从而提高数据挖掘的效率、降低计算成本。属性规约的目标是寻找出最小的属性子集并确保新数据子集的概率分布尽可能地接近原来数据集的概率分布。属性规约常用方法见表4-6。

表4-6　属性规约常用方法



逐步向前选择、逐步向后删除和决策树归纳是属于直接删除不相关属性（维）方法。主成分分析是一种用于连续属性的数据降维方法，它构造了原始数据的一个正交变换，新空间的基底去除了原始空间基底下数据的相关性，只需使用少数新变量就能够解释原始数据中的大部分变异。在应用中，通常是选出比原始变量个数少，能解释大部分数据中的变量的几个新变量，即所谓主成分，来代替原始变量进行建模。

主成分分析[6]的计算步骤如下。

1. 设原始变量X1，X2，…，Xp的n次观测数据矩阵为：

2）将数据矩阵按列进行中心标准化。为了方便，将标准化后的数据矩阵仍然记为X。

3）求相关系数矩阵R，R=（rij）p×p，rij的定义为：

4）求R的特征方程det（R-λE）＝0的特征根λ1≥λ2≥λp＞0。

5）确定主成分个数m：，α根据实际问题确定，一般取80％。

6）计算m个相应的单位特征向量：

7）计算主成分：

在Python中，主成分分析的函数位于Scikit-Learn下：

　sklearn.decomposition.PCA(n\_components = None, copy = True, whiten = False)

PCA的案例

### 4.2.1 数据特征分析

；数据特征分析要求我们在数据挖掘建模前，通过**频率分布分析**、对比分析、帕累托分析、**周期性分析、相关性分析**等方法，对采集的样本数据的特征规律进行分析，以了解数据的规律和趋势，为数据挖掘的后续环节提供支持。

### 4.2.4 小波变换待定

