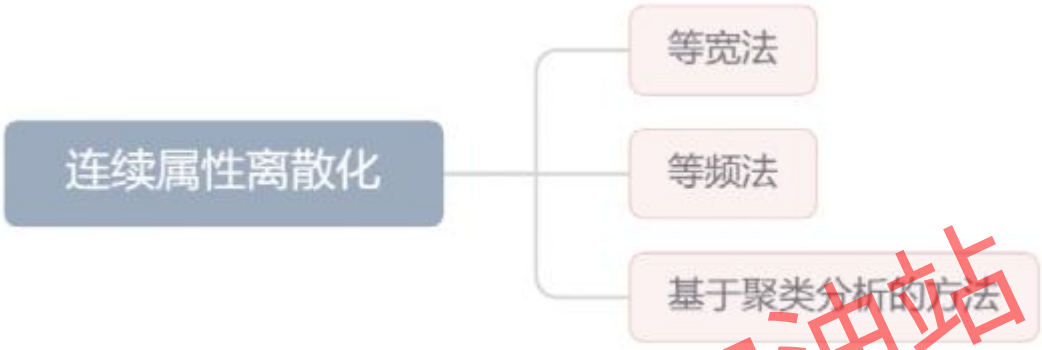
数据变换——连续变量离散化

连续变量离散化（又称数据分箱），是一种数据预处理方法，用于减少次要观察误差对模型的影响，降低模型过拟合的风险，使模型更加稳定。 但并非所有问题都需要做数据分箱，主要在一些分类算法中，如决策树 ID3 算法，要求自变量都是离散型。 常见的连续变量离散化方法主要有以下三类：



（1）等宽法： 将属性的值域分成具有相同宽度的区间，区间的个数由数据本身的特点决定或者用户指定，类似于制作频率分布表。

（2）等频法： 将相同数量的记录放进每个区间。

上述两个方法操作简单但都需要人为地规定划分区间的个数，等宽法的缺点在于对离群点比较敏感，因为离群点的出现会使得有些区间包含许多数据而另一些则数据很少，这样会严重损害所建立的决策模型。等频法避免了该问题，但却可能将相同数据值分到不同的区间以满足每个区间中固定的数据个数。

（3）（一维）聚类分析法： 首先将连续属性的值用聚类算法（如 K-Means）进行聚类，然后再将聚类得到的簇进行处理，合并得到一个簇的连续属性值做同一标记。

MATLAB 提供了 discretize 函数以实现等宽法分箱。

[Y,E] = discretize(X,N)

输入数组 X，将数组划分为 N 个宽度一致的区间（即“箱”，bin）。

输出数组每个元素所在的区间的序号 Y，以及每个区间的边界 E。

每个区间默认包含左边界，即 a <= x < b

Y = discretize(X,edges, ‘IncludedEdge’,side)
输入数组 X，按照 edges 进行分箱；

edges： 区间边界，输入为值递增的数值向量。 edges 中的连续元素形成离散的区间，discretize 使用这些区间 划分 X 中的数据。 默认情况下，每个区间都包括区间的左边界，除了最后一个区间，它包含区间的左右边界。

‘IncludedEdge’： 指定每个区间包含的边界。 side 为数值型参数，默认为’left’，每个区间包含左边界，a <= x < b； 设为’right’则每个区间包含右边界，即 a < x << span="">= b。
举 2 个栗子：

```
% 等宽法分箱
    X = randi(10,1,10)
    [Y,E] = discretize(X,3)
% 自定义每个区间的边界进行分箱
    data = [1 1 2 3 6 5 8 10 4 4]
    edges = 2:2:10
    Y = discretize(data,edges)
```

Y 表示数据的每个元素属于哪个 bin。 由于值 1 超出了区间范围，因此 Y 会在这些元素位置包含 NaN 值。

```
X =
   6   6   9   3   4   2  10   7   5   7

Y =
   2   2   3   1   1   1   3   2   2   2

E =
   2   5   8  11
```

```
data =
   1   1   2   3   6   5   8  10   4   4

edges =
   2   4   6   8  10

Y =
  NaN  NaN   1   1   3   2   4   4   2   2
```