# exp 1-数据预处理

## 引入$pandas$包和$numpy$包

```
import pandas as pd
import numpy as np
```

## 读取数据集

```
data = pd.read_csv('data/train_new.csv')
```

### 查看样本数量和特征数量

```
data.shape
```

```
(50000, 74)
```

### 检查读入数据的基本结构

```
data.head()
```

| | X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 | X9 | X10 | ... | X65 | X66 | X67 | X68 | X69 | X70 | X71 | X72 | Y | id |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 9.0 | 1458.0 | 17147.0 | 10.0 | 0.0 | 800.0 | 0.0 | NaN | 0.0 | 679.0 | ... | 7.0 | 581.0 | 2449.0 | 93.0 | 498.0 | 6.0 | 0.0 | 0.0 | 1 | 0 |
| **1** | 2.0 | 250.0 | 38.0 | 6.0 | NaN | 10000.0 | 0.0 | NaN | 1.0 | 12990.0 | ... | 31.0 | 796.0 | 7.0 | 122.0 | 406.0 | 5.0 | NaN | NaN | 1 | 1 |
| **2** | 2.0 | 1054.0 | 178.0 | 1.0 | 0.0 | 1000.0 | 0.0 | NaN | 1.0 | 18710.0 | ... | 230.0 | 732.0 | 29.0 | 78.0 | 10.0 | 6.0 | 0.0 | 0.0 | 0 | 2 |
| **3** | 10.0 | 1398.0 | 679.0 | 7.0 | 0.0 | 10000.0 | 0.0 | NaN | 1.0 | 19010.0 | ... | 11.0 | 36.0 | 113.0 | 82.0 | 35.0 | 6.0 | 0.0 | 0.0 | 1 | 3 |
| **4** | 2.0 | 1095.0 | 305.0 | 11.0 | 0.0 | 10000.0 | 0.0 | NaN | 2.0 | 16410.0 | ... | 93.0 | 395.0 | 50.0 | 48.0 | 491.0 | 5.0 | 0.0 | 0.0 | 0 | 4 |

5 rows × 74 columns

## 观察数据

### 数据缺失情况

```
missing_value_sum = data.isnull().sum()
```

```
missing_value_sum[0:10]
```

```
X1      5851
X2       390
X3       817
X4      4280
X5      8891
X6      3461
X7      4825
X8     48466
X9      4280
X10      955
dtype: int64
```

### 特征之间、特征与Label之间的相关度

#### .corr()

.corr() 计算相关系数矩阵，取值范围在[-1,1]，其中得数绝对值越大相关性越强，正数表示正相关，负数表示负相关

```
corr = data.corr()
```

```
corr[0:10]
```

| | X7 | X8 | X9 | X10 | ... | X65 | X66 | X67 | X68 | X69 | X70 | X71 | X72 | Y | id |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0510 | -0.021082 | -0.003949 | 0.002748 | ... | 0.003696 | 0.004220 | 0.002194 | -0.000705 | 0.002501 | -0.007212 | -0.008667 | -0.002907 | -0.007396 | 0.007816 | |
| 5850 | -0.002901 | 0.001000 | 0.055909 | ... | 0.637134 | 0.114426 | 0.127659 | -0.069287 | 0.198605 | 0.135124 | -0.001806 | -0.000636 | -0.052612 | -0.005470 | |
| 9100 | -0.013539 | 0.089222 | 0.021502 | ... | 0.110435 | 0.060165 | 0.997757 | -0.025897 | 0.096623 | 0.137174 | -0.025698 | -0.009129 | -0.012419 | -0.002800 | |
| 2175 | 0.020393 | 0.600068 | 0.003959 | ... | 0.166057 | 0.036125 | 0.146015 | -0.038666 | 0.079479 | 0.029898 | -0.009281 | -0.000339 | 0.066437 | -0.003480 | |
| 1668 | 0.300385 | -0.006101 | 0.002267 | ... | -0.005177 | 0.002582 | -0.010871 | 0.000487 | 0.008904 | -0.007493 | 0.196436 | 0.842313 | -0.008346 | -0.001466 | |
| 1095 | -0.010559 | -0.003911 | 0.003437 | ... | 0.010299 | 0.001217 | -0.005726 | 0.016657 | 0.004956 | -0.008236 | 0.003521 | 0.000449 | -0.009835 | 0.003314 | |
| 0000 | -0.003880 | -0.037396 | -0.010730 | ... | -0.035573 | -0.040136 | -0.028486 | 0.010183 | -0.056519 | -0.041117 | 0.010249 | -0.001397 | 0.016326 | 0.001224 | |
| 3880 | 1.000000 | 0.012443 | -0.000754 | ... | 0.013320 | 0.012779 | -0.013637 | -0.026441 | 0.088659 | 0.008881 | 0.296526 | 0.298793 | -0.020386 | -0.022960 | |
| 7396 | 0.012443 | 1.000000 | -0.001702 | ... | 0.100483 | -0.012501 | 0.088297 | -0.022689 | -0.011450 | 0.022129 | -0.012271 | -0.002344 | 0.038422 | -0.002440 | |
| 0730 | -0.000754 | -0.001702 | 1.000000 | ... | 0.064420 | 0.028534 | 0.021557 | 0.010355 | 0.049408 | -0.005804 | 0.007085 | 0.003051 | 0.000468 | -0.003107 | |

### .cov()

.cov() 计算协方差矩阵，得数为正，说明正相关，为负说明负相关，为0则不相关

```
cov = data.cov()
```

```
cov[0:10]
```

| | X65 | X66 | X67 | X68 | X69 | X70 | X71 | X72 | Y | id |
|---|---|---|---|---|---|---|---|---|---|---|
| 55067e+00 | 3.941626e+00 | 3.481346e+00 | -1.399093e-01 | 1.128594e+01 | -2.066750e-02 | -6.255876e+04 | -1.047307e+08 | -1.204056e-02 | 3.811577e+02 | |
| 33739e+04 | 2.426242e+04 | 4.552788e+04 | -3.021596e+03 | 1.974078e+05 | 8.728339e+01 | -2.926994e+06 | -5.007664e+09 | -1.950439e+01 | -6.031520e+04 | |
| 44956e+04 | 4.632109e+04 | 1.313362e+06 | -3.477416e+03 | 3.496892e+05 | 3.124526e+02 | -1.611054e+08 | -2.792869e+11 | -1.694448e+01 | -1.136803e+05 | |
| 55324e+02 | 8.015638e+01 | 5.419578e+02 | -1.819511e+01 | 8.254087e+02 | 1.994962e-01 | -1.591938e+05 | -2.847267e+07 | 2.566609e-01 | -4.031002e+02 | |
| 09143e+10 | 1.462705e+10 | -1.115866e+11 | 5.487122e+08 | 2.310737e+11 | -1.048108e+08 | 8.386457e+15 | 1.725747e+20 | -8.221648e+07 | -4.307433e+11 | |
| 27134e+04 | 1.836822e+04 | -1.511804e+05 | 5.343448e+04 | 3.559235e+05 | -3.577035e+02 | 2.055031e+08 | 1.268163e+11 | -2.640661e+02 | 2.650293e+06 | |
| 114929e-01 | -1.050871e+00 | -1.242195e+00 | 5.518620e-02 | -6.941358e+00 | -3.144931e-03 | 1.950237e+03 | -1.287988e+06 | 7.450623e-04 | 1.675803e+00 | |
| 04682e+10 | 9.661805e+10 | -1.150278e+11 | -2.550035e+10 | 2.727680e+12 | 1.695845e+08 | 6.762558e+16 | 3.241901e+20 | -2.541891e+08 | -8.583093e+12 | |
| 65826e+01 | -8.755588e+00 | 1.035784e+02 | -3.374583e+00 | -3.765519e+01 | 4.662955e-02 | -6.647651e+04 | -6.220060e+07 | 4.685442e-02 | -8.919649e+01 | |
| 68810e+05 | 1.328615e+05 | 4.623299e+05 | 2.692343e+04 | 1.084304e+06 | -2.010323e+02 | 7.226618e+08 | 1.532019e+12 | 1.033799e+01 | -2.042078e+06 | |

# 处理缺失数据

## 默认值填充

将缺失值填充为-1

```
data1 = data
```

```
data1.fillna(-1, inplace=True)
```

```
data1.head()
```

| | X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 | X9 | X10 | ... | X65 | X66 | X67 | X68 | X69 | X70 | X71 | X72 | Y | id |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 9.0 | 1458.0 | 17147.0 | 10.0 | 0.0 | 800.0 | 0.0 | -1.0 | 0.0 | 679.0 | ... | 7.0 | 581.0 | 2449.0 | 93.0 | 498.0 | 6.0 | 0.0 | 0.0 | 1 | 0 |
| **1** | 2.0 | 250.0 | 38.0 | 6.0 | -1.0 | 10000.0 | 0.0 | -1.0 | 1.0 | 12990.0 | ... | 31.0 | 796.0 | 7.0 | 122.0 | 406.0 | 5.0 | -1.0 | -1.0 | 1 | 1 |
| **2** | 2.0 | 1054.0 | 178.0 | 1.0 | 0.0 | 1000.0 | 0.0 | -1.0 | 1.0 | 18710.0 | ... | 230.0 | 732.0 | 29.0 | 78.0 | 10.0 | 6.0 | 0.0 | 0.0 | 0 | 2 |
| **3** | 10.0 | 1398.0 | 679.0 | 7.0 | 0.0 | 10000.0 | 0.0 | -1.0 | 1.0 | 19010.0 | ... | 11.0 | 36.0 | 113.0 | 82.0 | 35.0 | 6.0 | 0.0 | 0.0 | 1 | 3 |
| **4** | 2.0 | 1095.0 | 305.0 | 11.0 | 0.0 | 10000.0 | 0.0 | -1.0 | 2.0 | 16410.0 | ... | 93.0 | 395.0 | 50.0 | 48.0 | 491.0 | 5.0 | 0.0 | 0.0 | 0 | 4 |

5 rows × 74 columns

# 平均值填充

```
data2 = data
```

```
data2.fillna(data2.mean(),inplace=True)
```

```
data2.head()
```

| | X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 | X9 | X10 | ... | X65 | X66 | X67 | X68 | X69 | X70 | X71 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 9.0 | 1458.0 | 17147.0 | 10.0 | 0.000000e+00 | 800.0 | 0.0 | 6.957551e+08 | 0.0 | 679.0 | ... | 7.0 | 581.0 | 2449.0 | 93.0 | 498.0 | 6.0 | 0.000000 | 0. |
| **1** | 2.0 | 250.0 | 38.0 | 6.0 | 4.235226e+08 | 10000.0 | 0.0 | 6.957551e+08 | 1.0 | 12990.0 | ... | 31.0 | 796.0 | 7.0 | 122.0 | 406.0 | 5.0 | 146266.002699 | 1. |
| **2** | 2.0 | 1054.0 | 178.0 | 1.0 | 0.000000e+00 | 1000.0 | 0.0 | 6.957551e+08 | 1.0 | 18710.0 | ... | 230.0 | 732.0 | 29.0 | 78.0 | 10.0 | 6.0 | 0.000000 | 0. |
| **3** | 10.0 | 1398.0 | 679.0 | 7.0 | 0.000000e+00 | 10000.0 | 0.0 | 6.957551e+08 | 1.0 | 19010.0 | ... | 11.0 | 36.0 | 113.0 | 82.0 | 35.0 | 6.0 | 0.000000 | 0. |
| **4** | 2.0 | 1095.0 | 305.0 | 11.0 | 0.000000e+00 | 10000.0 | 0.0 | 6.957551e+08 | 2.0 | 16410.0 | ... | 93.0 | 395.0 | 50.0 | 48.0 | 491.0 | 5.0 | 0.000000 | 0. |

5 rows × 74 columns

# 删除不完整的行

## 删除任何包含空值的行

```
data3 = data
```

```
data3.dropna(inplace=True)
```

```
data3.head()
```

| | X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 | X9 | X10 | ... | X65 | X66 | X67 | X68 | X69 | X70 | X71 | X72 | Y |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **33** | 2.0 | 668.0 | 113.0 | 7.0 | 1048576.0 | 20000.0 | 0.0 | 3145728.0 | 2.0 | 11900.0 | ... | 110.0 | 281.0 | 18.0 | 49.0 | 447.0 | 6.0 | 1597992.0 | 381525.0 | 0 |
| **170** | 6.0 | 2089.0 | 684.0 | 4.0 | 614400.0 | 5000.0 | 0.0 | 11400.0 | 0.0 | 3000.0 | ... | 173.0 | 866.0 | 114.0 | 100.0 | 1092.0 | 6.0 | 0.0 | 614400.0 | 0 |
| **573** | 6.0 | 1309.0 | 31.0 | 11.0 | 5345280.0 | 10000.0 | 0.0 | 30000.0 | 4.0 | 14060.0 | ... | 352.0 | 855.0 | 5.0 | 62.0 | 882.0 | 6.0 | 337356.0 | 2655573.0 | 1 |
| **663** | 2.0 | 146.0 | 187.0 | 1.0 | 3072.0 | 3000.0 | 0.0 | 700.0 | 0.0 | 25900.0 | ... | 146.0 | 656.0 | 187.0 | 61.0 | 31.0 | 6.0 | 3072.0 | 3072.0 | 1 |
| **3879** | 1.0 | 485.0 | 29.0 | 2.0 | 1830635.0 | 5000.0 | 0.0 | 30000.0 | 1.0 | 17165.0 | ... | 52.0 | 247.0 | 9.0 | 134.0 | 44.0 | 6.0 | 2281123.0 | 1266909.0 | 1 |

5 rows × 74 columns

## 删除所有值都为空的行

```
data.dropna(how='all',inplace=True)
```

```
data.head()
```

| | X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 | X9 | X10 | ... | X65 | X66 | X67 | X68 | X69 | X70 | X71 | X72 | Y | id |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 9.0 | 1458.0 | 17147.0 | 10.0 | 0.0 | 800.0 | 0.0 | NaN | 0.0 | 679.0 | ... | 7.0 | 581.0 | 2449.0 | 93.0 | 498.0 | 6.0 | 0.0 | 0.0 | 1 | 0 |
| **1** | 2.0 | 250.0 | 38.0 | 6.0 | NaN | 10000.0 | 0.0 | NaN | 1.0 | 12990.0 | ... | 31.0 | 796.0 | 7.0 | 122.0 | 406.0 | 5.0 | NaN | NaN | 1 | 1 |
| **2** | 2.0 | 1054.0 | 178.0 | 1.0 | 0.0 | 1000.0 | 0.0 | NaN | 1.0 | 18710.0 | ... | 230.0 | 732.0 | 29.0 | 78.0 | 10.0 | 6.0 | 0.0 | 0.0 | 0 | 2 |
| **3** | 10.0 | 1398.0 | 679.0 | 7.0 | 0.0 | 10000.0 | 0.0 | NaN | 1.0 | 19010.0 | ... | 11.0 | 36.0 | 113.0 | 82.0 | 35.0 | 6.0 | 0.0 | 0.0 | 1 | 3 |
| **4** | 2.0 | 1095.0 | 305.0 | 11.0 | 0.0 | 10000.0 | 0.0 | NaN | 2.0 | 16410.0 | ... | 93.0 | 395.0 | 50.0 | 48.0 | 491.0 | 5.0 | 0.0 | 0.0 | 0 | 4 |

5 rows × 74 columns

```
data.shape
```

```
(50000, 74)
```

## 删除行数据中非空值少于10个的行

```
data.dropna(thresh=10,inplace=True)
```

```
data.head()
```

| | X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 | X9 | X10 | ... | X65 | X66 | X67 | X68 | X69 | X70 | X71 | X72 | Y | id |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 9.0 | 1458.0 | 17147.0 | 10.0 | 0.0 | 800.0 | 0.0 | NaN | 0.0 | 679.0 | ... | 7.0 | 581.0 | 2449.0 | 93.0 | 498.0 | 6.0 | 0.0 | 0.0 | 1 | 0 |
| **1** | 2.0 | 250.0 | 38.0 | 6.0 | NaN | 10000.0 | 0.0 | NaN | 1.0 | 12990.0 | ... | 31.0 | 796.0 | 7.0 | 122.0 | 406.0 | 5.0 | NaN | NaN | 1 | 1 |
| **2** | 2.0 | 1054.0 | 178.0 | 1.0 | 0.0 | 1000.0 | 0.0 | NaN | 1.0 | 18710.0 | ... | 230.0 | 732.0 | 29.0 | 78.0 | 10.0 | 6.0 | 0.0 | 0.0 | 0 | 2 |
| **3** | 10.0 | 1398.0 | 679.0 | 7.0 | 0.0 | 10000.0 | 0.0 | NaN | 1.0 | 19010.0 | ... | 11.0 | 36.0 | 113.0 | 82.0 | 35.0 | 6.0 | 0.0 | 0.0 | 1 | 3 |
| **4** | 2.0 | 1095.0 | 305.0 | 11.0 | 0.0 | 10000.0 | 0.0 | NaN | 2.0 | 16410.0 | ... | 93.0 | 395.0 | 50.0 | 48.0 | 491.0 | 5.0 | 0.0 | 0.0 | 0 | 4 |

5 rows × 74 columns

```
data.shape
```

```
(49851, 74)
```

### 删除label为空的行

```
data.dropna(subset=['Y'],inplace=True)
```

```
data.head()
```

| | X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 | X9 | X10 | ... | X65 | X66 | X67 | X68 | X69 | X70 | X71 | X72 | Y | id |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 9.0 | 1458.0 | 17147.0 | 10.0 | 0.0 | 800.0 | 0.0 | NaN | 0.0 | 679.0 | ... | 7.0 | 581.0 | 2449.0 | 93.0 | 498.0 | 6.0 | 0.0 | 0.0 | 1 | 0 |
| **1** | 2.0 | 250.0 | 38.0 | 6.0 | NaN | 10000.0 | 0.0 | NaN | 1.0 | 12990.0 | ... | 31.0 | 796.0 | 7.0 | 122.0 | 406.0 | 5.0 | NaN | NaN | 1 | 1 |
| **2** | 2.0 | 1054.0 | 178.0 | 1.0 | 0.0 | 1000.0 | 0.0 | NaN | 1.0 | 18710.0 | ... | 230.0 | 732.0 | 29.0 | 78.0 | 10.0 | 6.0 | 0.0 | 0.0 | 0 | 2 |
| **3** | 10.0 | 1398.0 | 679.0 | 7.0 | 0.0 | 10000.0 | 0.0 | NaN | 1.0 | 19010.0 | ... | 11.0 | 36.0 | 113.0 | 82.0 | 35.0 | 6.0 | 0.0 | 0.0 | 1 | 3 |
| **4** | 2.0 | 1095.0 | 305.0 | 11.0 | 0.0 | 10000.0 | 0.0 | NaN | 2.0 | 16410.0 | ... | 93.0 | 395.0 | 50.0 | 48.0 | 491.0 | 5.0 | 0.0 | 0.0 | 0 | 4 |

5 rows × 74 columns

```
data.shape
```

```
(50000, 74)
```

# 数据变换与离散化

## 缩放(使用最大最小值规范化)

```
numeric_feats = data.dtypes[data.dtypes!="object"].index
```

```
data[numeric_feats] = data[numeric_feats].apply(lambda x:(x-x.min())/(x.max()-x.min()))
```

```
data.head()
```

| | X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 | X9 | X10 | ... | X65 | X66 | X67 | X68 | X69 | X70 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0.818182 | 0.094984 | 0.145874 | 0.163934 | 0.0 | 0.008626 | 0.0 | NaN | 0.00 | 0.002204 | ... | 0.003107 | 0.427520 | 0.125006 | 0.013330 | 0.022994 | 0.250000 |
| **1** | 0.181818 | 0.016287 | 0.000323 | 0.098361 | NaN | 0.009481 | 0.0 | NaN | 0.04 | 0.003510 | ... | 0.015536 | 0.585725 | 0.000357 | 0.017486 | 0.018746 | 0.208333 |
| **2** | 0.181818 | 0.068664 | 0.001514 | 0.016393 | 0.0 | 0.008644 | 0.0 | NaN | 0.04 | 0.004117 | ... | 0.118591 | 0.538631 | 0.001480 | 0.011180 | 0.000462 | 0.250000 |
| **3** | 0.909091 | 0.091075 | 0.005776 | 0.114754 | 0.0 | 0.009481 | 0.0 | NaN | 0.04 | 0.004148 | ... | 0.005179 | 0.026490 | 0.005768 | 0.011753 | 0.001616 | 0.250000 |
| **4** | 0.181818 | 0.071336 | 0.002595 | 0.180328 | 0.0 | 0.009481 | 0.0 | NaN | 0.08 | 0.003873 | ... | 0.047644 | 0.290655 | 0.002552 | 0.006880 | 0.022671 | 0.208333 |

5 rows × 74 columns

## 规范化(零均值规范化)

```
numeric_feats = data.dtypes[data.dtypes!="object"].index
```

```
data[numeric_feats] = data[numeric_feats].apply(lambda x:(x-x.mean())/(x.std()))
```

```
data.head()
```

| | X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 | X9 | X10 | ... | X65 | X66 | X67 | X |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1.029154 | 0.538781 | 5.607751 | -0.070962 | -0.020829 | -0.103187 | -0.095701 | NaN | -0.798870 | -0.253708 | ... | -0.910740 | 0.440512 | 4.743993 | 0.2348 |
| **1** | -1.044798 | -1.042146 | -0.473350 | -0.569373 | NaN | 0.062788 | -0.095701 | NaN | -0.404132 | 0.016640 | ... | -0.726820 | 1.218165 | -0.475500 | 0.7416 |
| **2** | -1.044798 | 0.010060 | -0.423590 | -1.192385 | -0.020829 | -0.099579 | -0.095701 | NaN | -0.404132 | 0.142251 | ... | 0.798183 | 0.986677 | -0.428477 | -0.0272 |
| **3** | 1.325433 | 0.460258 | -0.245518 | -0.444770 | -0.020829 | 0.062788 | -0.095701 | NaN | -0.404132 | 0.148839 | ... | -0.880087 | -1.530746 | -0.248937 | 0.0426 |
| **4** | -1.044798 | 0.063717 | -0.378450 | 0.053640 | -0.020829 | 0.062788 | -0.095701 | NaN | -0.009394 | 0.091743 | ... | -0.251693 | -0.232248 | -0.383592 | -0.5515 |

5 rows × 74 columns

# 离散化

## 等深分箱

```
dataX65_bin = pd.qcut(data.X65,q=10,duplicates='drop')
```

```
dataX65_bin
```

```
0         (0.999, 12.0]
1          (27.0, 44.0]
2        (199.0, 288.0]
3         (0.999, 12.0]
4         (88.0, 115.0]
              ...
49995    (199.0, 288.0]
49996    (115.0, 150.0]
49997    (115.0, 150.0]
49998     (0.999, 12.0]
49999      (12.0, 27.0]
Name: X65, Length: 50000, dtype: category
Categories (10, interval[float64]): [(0.999, 12.0] < (12.0, 27.0] < (27.0, 44.0] < (44.0, 64.0] ... (115.0, 150.0] < (150.0,
199.0] < (199.0, 288.0] < (288.0, 1932.0]]
```

将data中属性X65分为10箱，其中若边界值不唯一，则弃掉。

## 等宽分箱

```
dataX66_bin = pd.cut(data.X66,bins=[100,200,300,400,500,600])
```

```
dataX66_bin
```

```
0        (500.0, 600.0]
1                   NaN
2                   NaN
3                   NaN
4        (300.0, 400.0]
              ...
49995               NaN
49996               NaN
49997    (400.0, 500.0]
49998               NaN
49999    (400.0, 500.0]
Name: X66, Length: 50000, dtype: category
Categories (5, interval[int64]): [(100, 200] < (200, 300] < (300, 400] < (400, 500] < (500, 600]]
```

将data中属性X66分为5个箱子，值为100-200,200-300,300-400,400-500,500-600.

# 特征构造（交叉）

```
def add_cross_feature(data,feature1,feature2):
    comb_index = data[[feature1,feature2]].drop_duplicates()
    comb_index[feature1+'_'+feature2]=np.arange(comb_index.shape[0])
    data = pd.merge(data, comb_index, 'left', on=[feature1,feature2])
    return data
```

```
comb_index = data[['X34','X36']].drop_duplicates()
```

```
comb_index['X34'+'_'+'X36'] = np.arange(comb_index.shape[0])
```

```
data1 = pd.merge(data,comb_index,'left',on=['X34','X36'])
```

```
data1.X36.corr(data.Y)
```

```
0.0013384845715645115
```

```
data1.X34.corr(data.Y)
```

```
0.1412437064881158
```

```
data1.X34_X36.corr(data.Y)
```

```
-0.00877721933255106
```

将X34与X36特征按照去重合并的方式进行交叉构造，得到新的特征X34_X36

## 数据集切分

指定训练集和测试集比例对数据集进行划分

```
num_train = int(data.shape[0]*0.8)
train_data = data[:num_train]
test_data = data[num_train:]
train_data.shape, test_data.shape
```

保存处理后的数据集：

```
train_data.to_csv("train.csv")
test_data.to_csv("test.csv")
```