|  |
| --- |
| **机器学习** |
| **Santander Customer Transaction Prediction** |
| 主 研 人：岳天昕  参 研 人： |
| 审 核 人：  方 向：工业算法案例研究  版 本 号：A |
|  |

**版本更新**

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **更新描述** | **作者** |
| 2018/10/31 | A | 初稿 |  |

**选题表格**

|  |  |  |  |
| --- | --- | --- | --- |
| 时间 | 竞赛名 | 竞赛背景描述（50字以内） | 类型（分类/回归） |
| 2019/4/18 | Santander Customer Transaction Prediction | 我们邀请Kagglers帮助我们确定哪些客户将来会进行特定交易，无论交易金额多少 | 分类 |

目录

[1. 背景描述 4](#_Toc25575_WPSOffice_Level1)

[1.1 竞赛赛题描述 4](#_Toc5940_WPSOffice_Level2)

[1.2 评估指标描述 5](#_Toc12357_WPSOffice_Level2)

[2. 数据来源及描述性统计分析 5](#_Toc5940_WPSOffice_Level1)

[2.1 大赛数据来源 5](#_Toc22683_WPSOffice_Level2)

[2.2 数据的描述性统计 5](#_Toc16174_WPSOffice_Level2)

[2.2.1 数据基本情况描述： 5](#_Toc5940_WPSOffice_Level3)

[2.2.2 数据字段介绍： 5](#_Toc12357_WPSOffice_Level3)

[2.2.3 数据描述性统计 6](#_Toc22683_WPSOffice_Level3)

[3. 优秀算法思路 6](#_Toc12357_WPSOffice_Level1)

[3.1 方案一 6](#_Toc32293_WPSOffice_Level2)

[3.1.1 方案一数据预处理及特征工程部分方案 6](#_Toc16174_WPSOffice_Level3)

[3.1.2 方案一模型设计、建立部分方案 6](#_Toc32293_WPSOffice_Level3)

[3.1.3 方案一结果、排名等 6](#_Toc29922_WPSOffice_Level3)

[3.1.4 方案一算法流程图 6](#_Toc4787_WPSOffice_Level3)

[3.2 方案二 7](#_Toc29922_WPSOffice_Level2)

[… 8](#_Toc4787_WPSOffice_Level2)

[4. 算法比较 8](#_Toc22683_WPSOffice_Level1)

[表4-1 算法比较 8](#_Toc7253_WPSOffice_Level2)

[5. 总结与展望 8](#_Toc16174_WPSOffice_Level1)

[5.1 总结 8](#_Toc23198_WPSOffice_Level2)

[5.2 建模思路 8](#_Toc16619_WPSOffice_Level2)

1. 背景描述

在Santander，我们的使命是帮助人们和企业繁荣发展。

我们一直在寻找方法来帮助客户了解他们的财务状况，并确定哪些产品和服务可以帮助他们实现货币目标。

我们的数据科学团队不断挑战我们的机器学习算法，与全球数据科学界合作，确保我们能够更准确地识别解决我们最常见挑战的新方法，二元分类问题，例如：客户是否满意？客户会购买此产品吗？客户可以支付这笔贷款吗？

**1.1 竞赛赛题描述**

关于竞赛题目的描述

在此挑战中，我们邀请Kagglers帮助我们确定哪些客户将来会进行特定交易，无论交易金额多少。

为此次竞赛提供的数据与我们可用于解决此问题的实际数据具有相同的结构。

**1.2 评估指标描述**

竞赛中评估模型优劣的指标：ROC曲线下的面积

1. 数据来源及描述性统计分析

**2.1 大赛数据来源**

大赛数据的来源：[Banco Santander](https://www.kaggle.com/santander)公司

这里是数据的超链接：

<https://www.kaggle.com/c/santander-customer-transaction-prediction/data>

**2.2 数据的描述性统计**

**2.2.1 数据基本情况描述：**

您将获得一个匿名数据集，其中包含数字要素变量，二进制目标列和字符串ID\_code列。

任务是预测测试集中目标列的值。

**2.2.2 数据字段介绍：**

做成表格形式（如果有多个数据表，请做多个数据字段介绍表格），格式如下：

**表2-1 训练数据表字段介绍**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 变量含义 | 变量类型 | 缺失率 |
| ID\_code | **Id** | 离散 | 0% |
| target | **目标值** | 0或1 | 0% |
| var\_0 | **数字变量** | 连续 | 0% |
| var\_1 | **数字变量** | 连续 | 0% |
| ………. | **数字变量** | 连续 | 0% |
| var\_199 | **数字变量** | 连续 | 0% |

**表2-2 测试数据表字段介绍**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 变量含义 | 变量类型 | 缺失率 |
| ID\_code | **Id** | 离散 | 0% |
| var\_0 | **数字变量** | 连续 | 0% |
| var\_1 | **数字变量** | 连续 | 0% |
| ………. | **数字变量** | 连续 | 0% |
| var\_199 | **数字变量** | 连续 | 0% |

**2.2.3 数据描述性统计**

包含（变量基础分布情况描述（均值、方差、分为数、分组计数、分布情况、相关情况分析等等）可以从论坛中寻找别人已经完成的描述性统计。形式包含不限于文字描述、图表等。如果难以找到可尽量做一些简单的描述性统计。

训练集包含：

ID\_code（字符串）;

目标值；

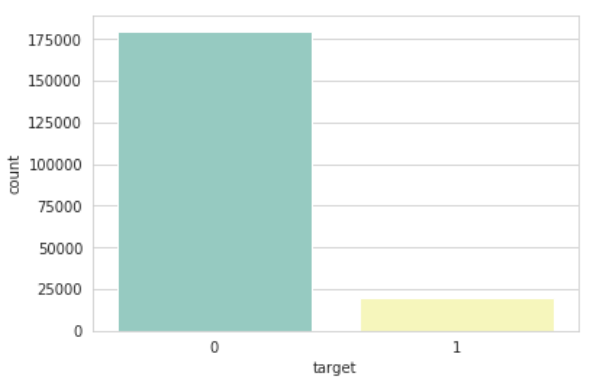
200个数值变量，从var\_0到var\_199命名;

测试集包含：

ID\_code（字符串）;

200个数值变量，从var\_0到var\_199命名;

查看训练集目标值的分布：



**特征的密度图**  
现在让我们看一下训练数据集中变量的密度图。  
我们用不同的颜色表示目标值为0和1的值的分布





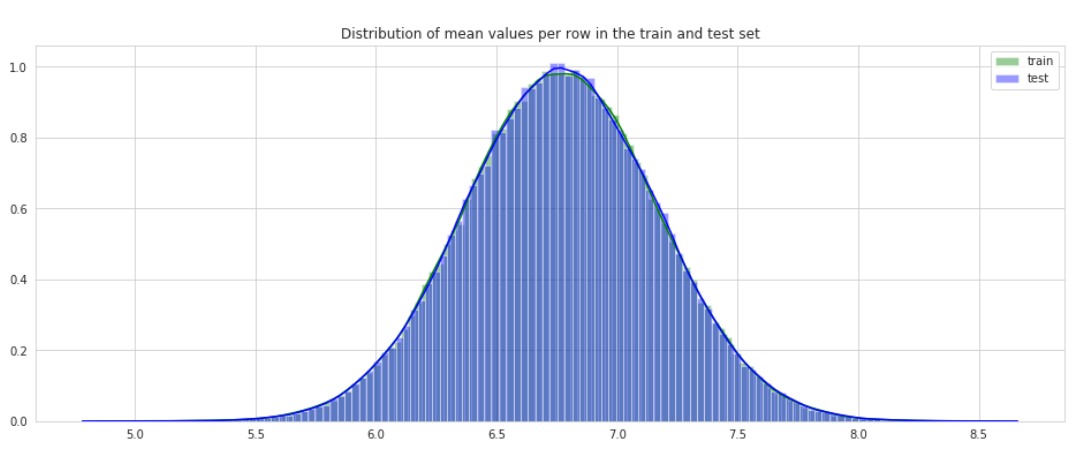
我们可以观察到，存在相当多的特征，两个目标值具有显着不同的分布。  
例如，var\_0，var\_1，var\_2，var\_5，var\_9，var\_13，var\_106，var\_109，var\_139等等。  
此外，某些特征（如var\_2，var\_13，var\_26，var\_55，var\_175，var\_184，var\_196）显示可转变为双变量分布的分布。  
我们将在未来考虑到这一点，为我们的pred选择特征。

现在我们查看训练集测试集中特征分布相同的特征：

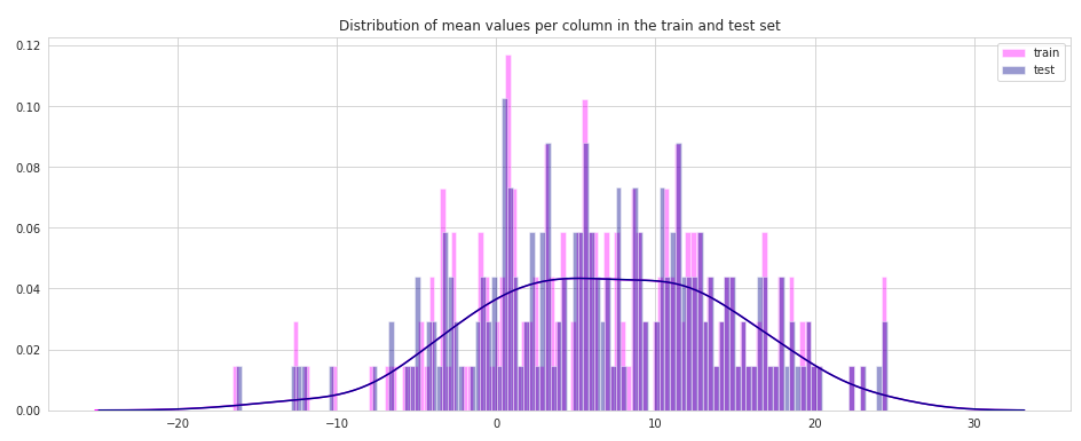




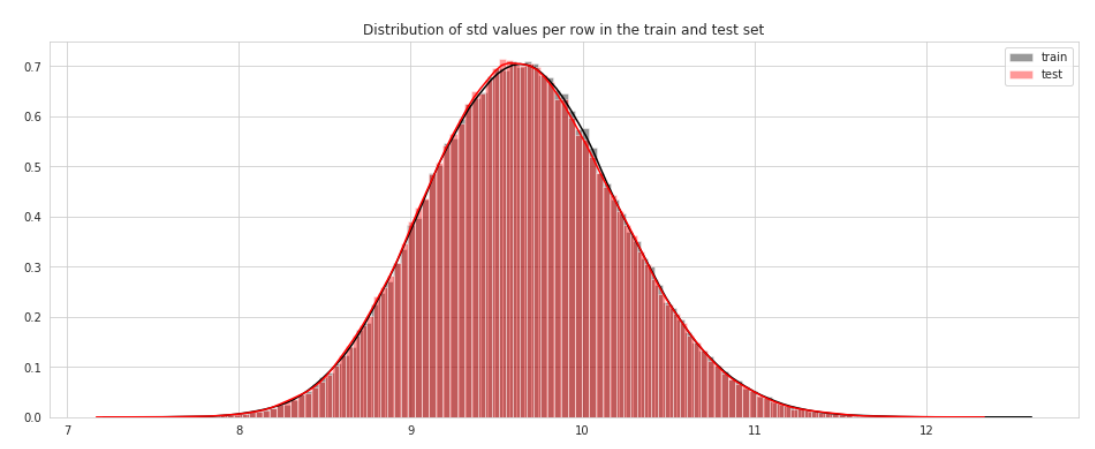
**均值和std的分布¶**  
让我们检查训练和测试集中每行平均值的分布



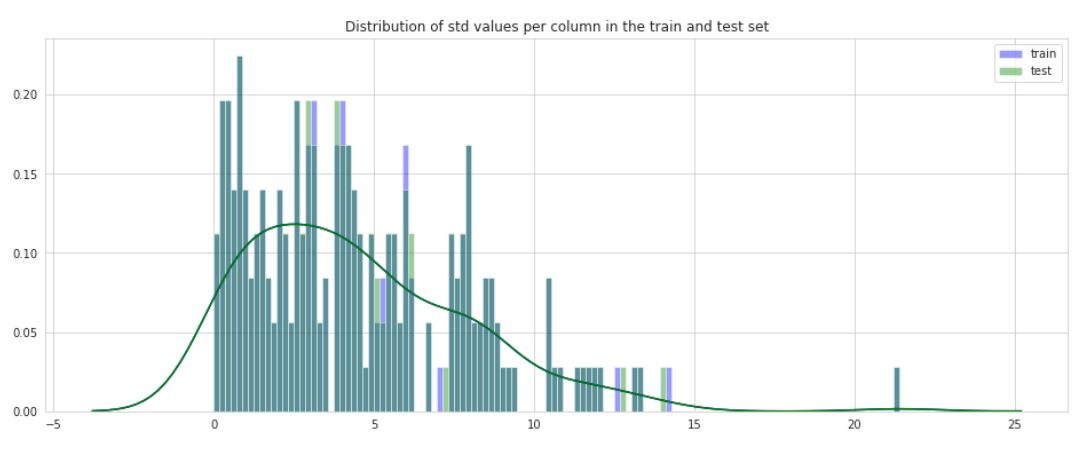
让我们查看训练和测试集中每列平均值的分布：



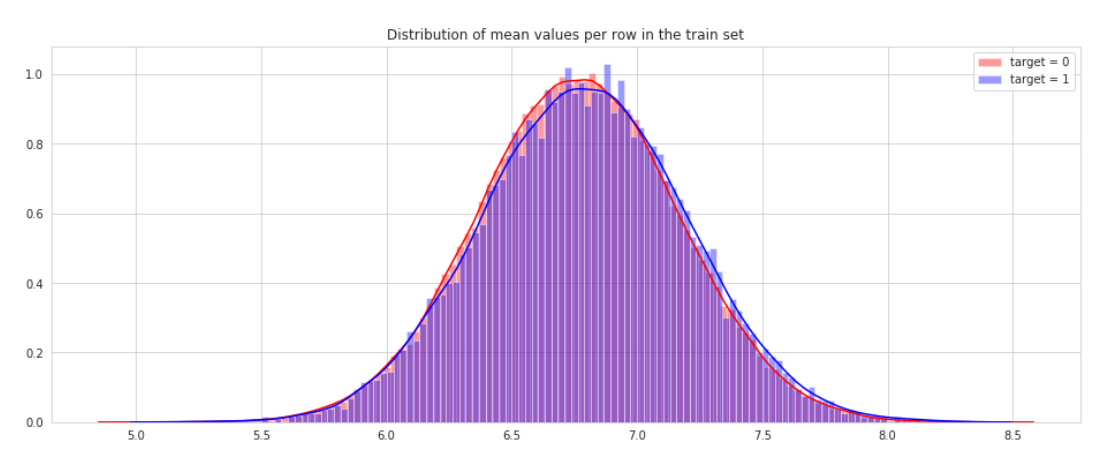
让我们显示训练和测试数据集的每行值的标准偏差的分布：



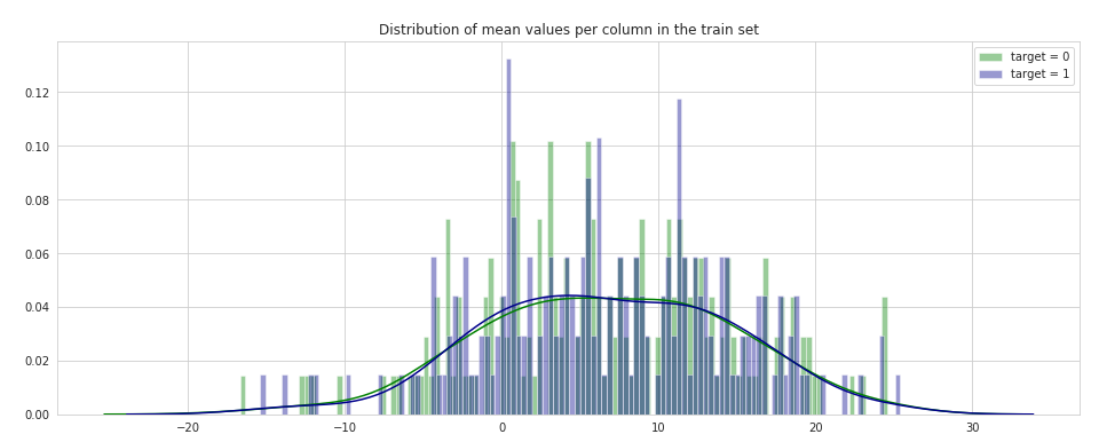
让我们检查训练和测试数据集中每列值的标准偏差的分布：



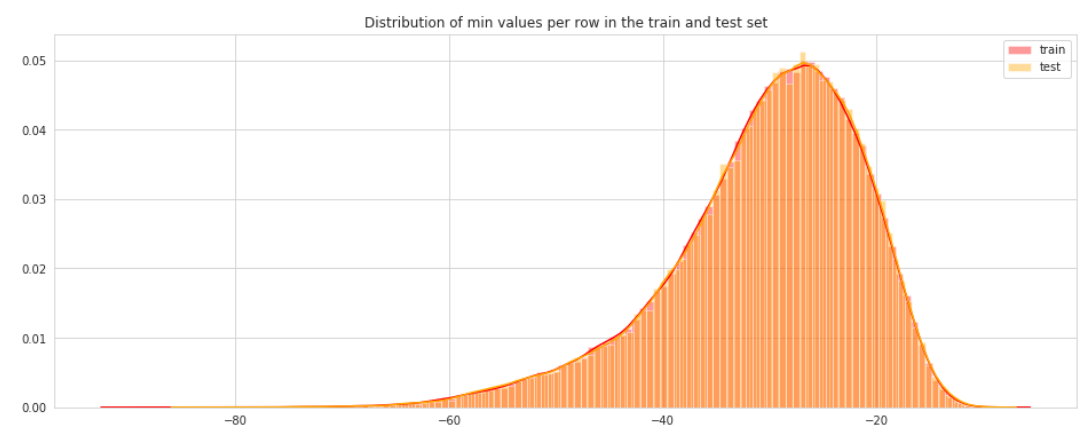
我们现在检查训练数据集中每行平均值的分布，按目标值分组：



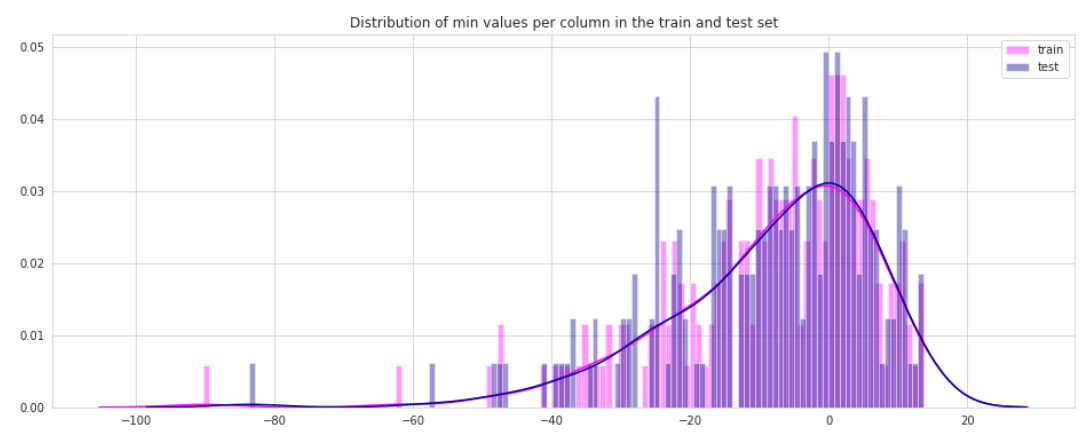
我们现在检查训练数据集中每列平均值的分布，按目标值分组：



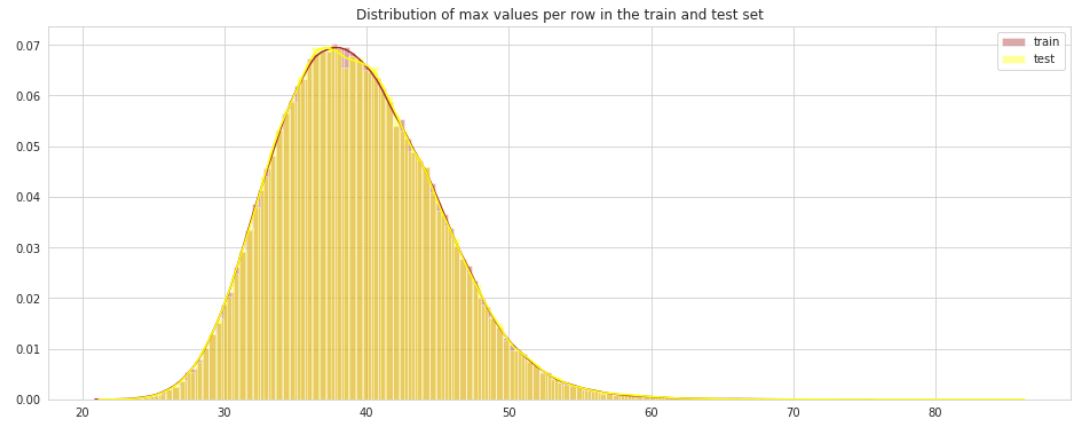
**最小和最大的分布**  
让我们查看训练和测试集中每行的最小分布：



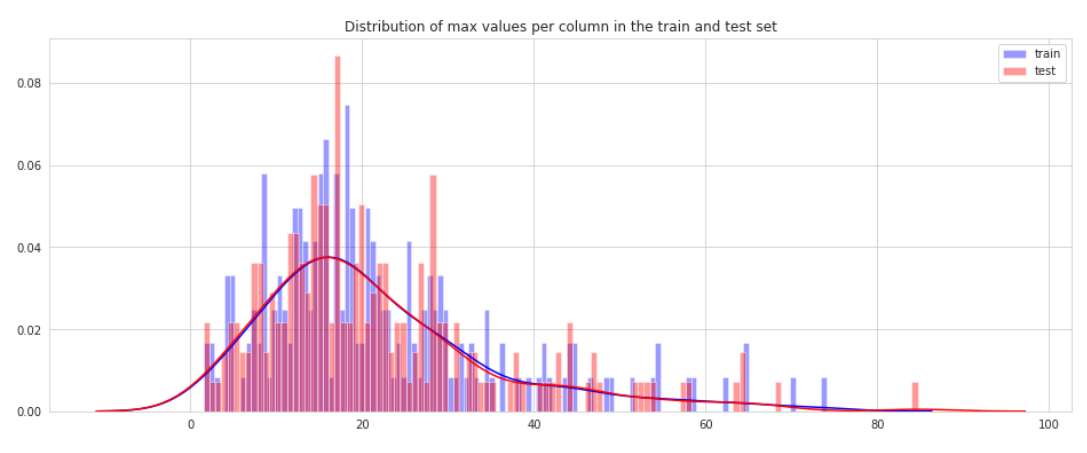
观察到两个较长值的长队列，扩展到测试集的-80。  
现在让我们展示训练和测试装置中每列的最小分布：



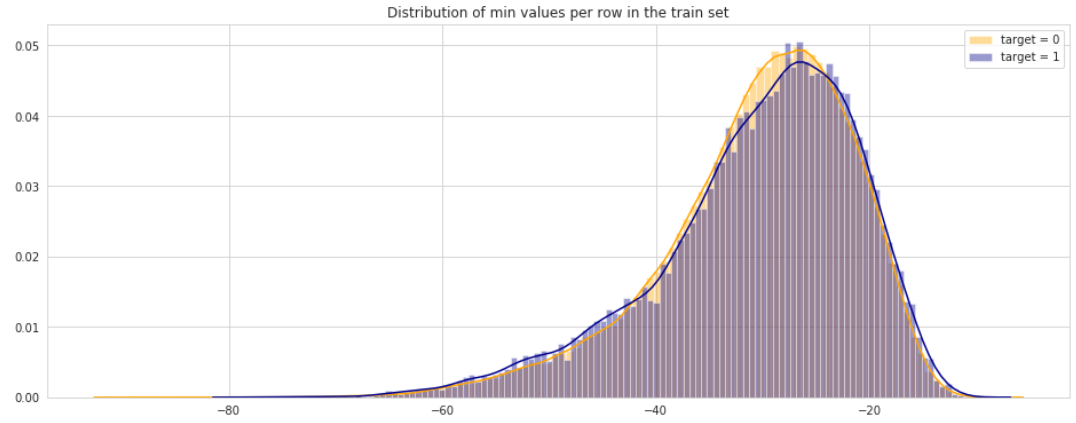
我们现在查看训练和测试集的每行最大值的分布：



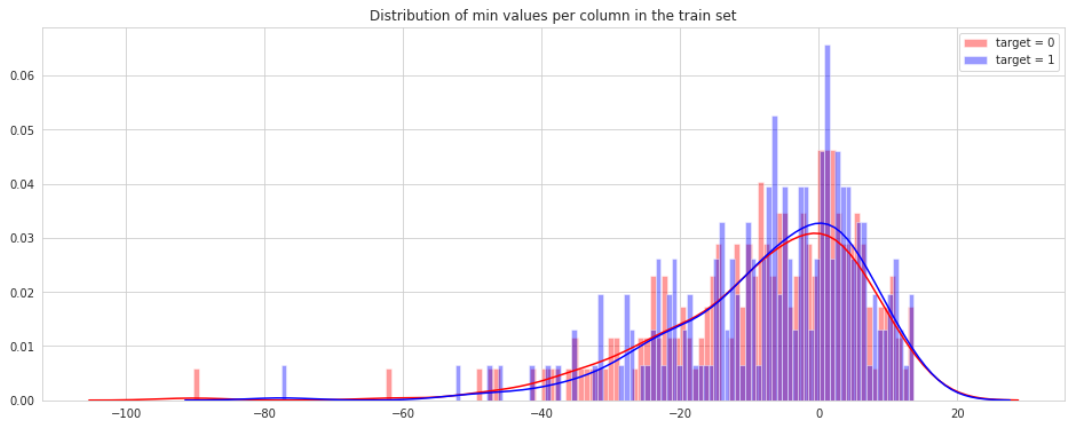
现在让我们看一下训练和测试集的最大值分布：



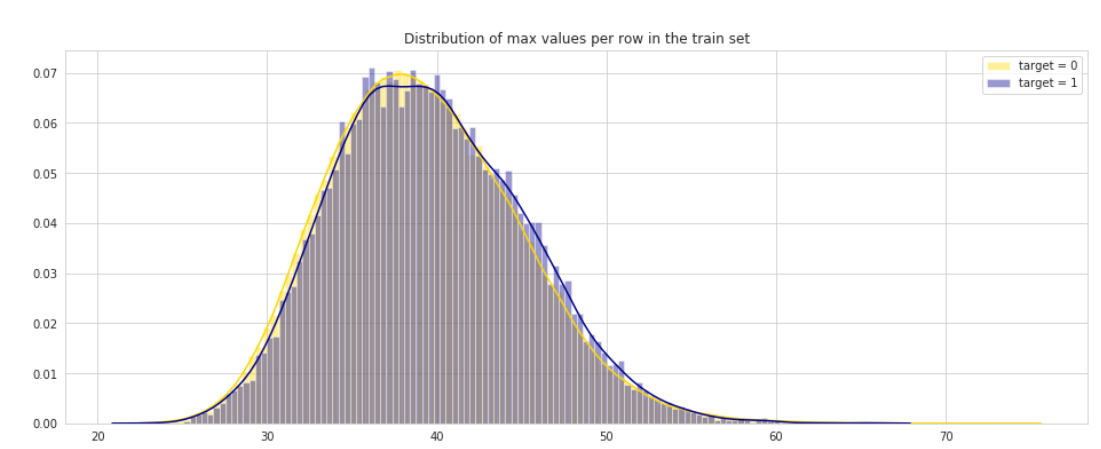
现在让我们显示训练集中每行的最小值的分布，以目标（0和1）的值分开：



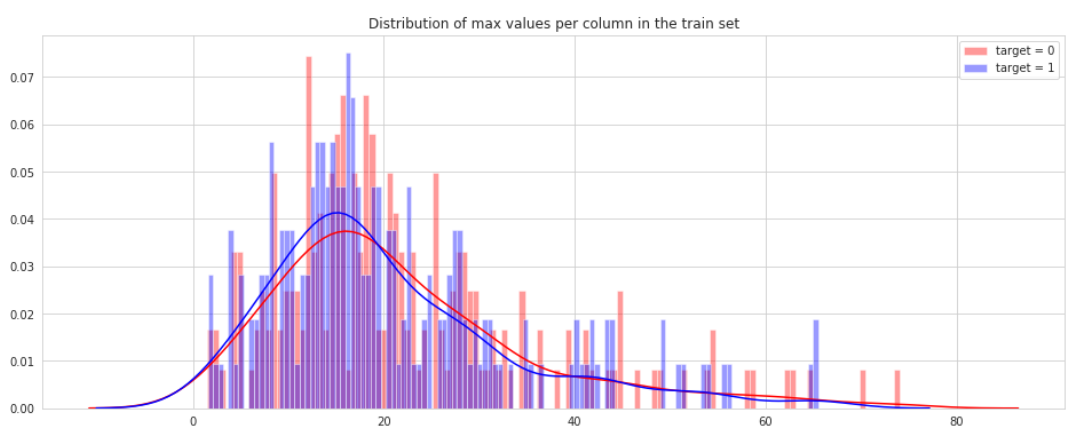
我们在这里显示训练集中每列的最小值的分布：



现在让我们看一下训练组中每行最大值的分布：



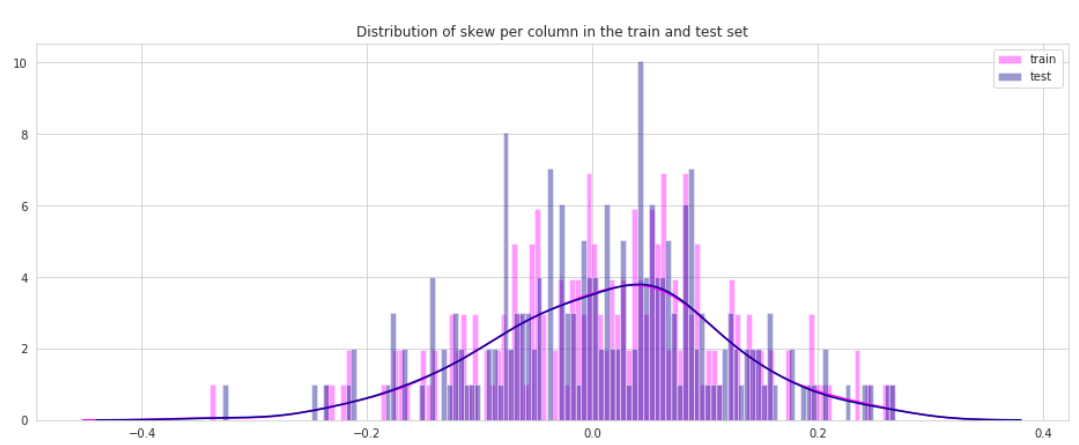
让我们显示测试集中每列的最大值的分布：



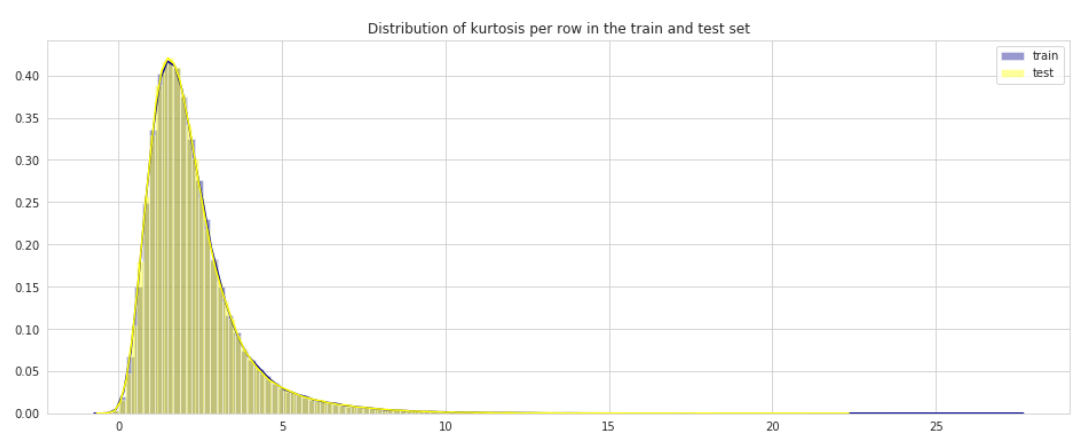
**偏斜和峰度的分布**  
现在让我们看看每行和每列的偏斜值的分布是什么。  
让我们首先看一下训练和测试集中每行计算的偏度的分布：



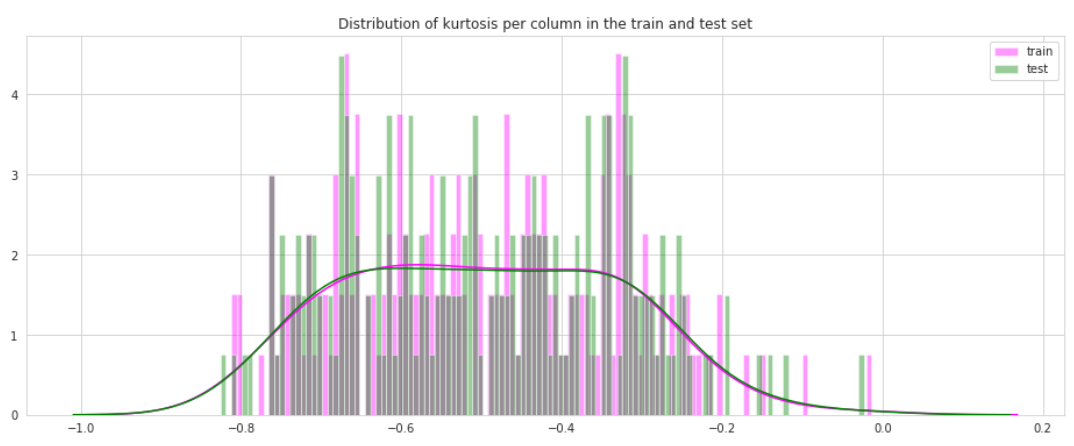
让我们首先看一下训练和测试集中每列计算的偏度分布：



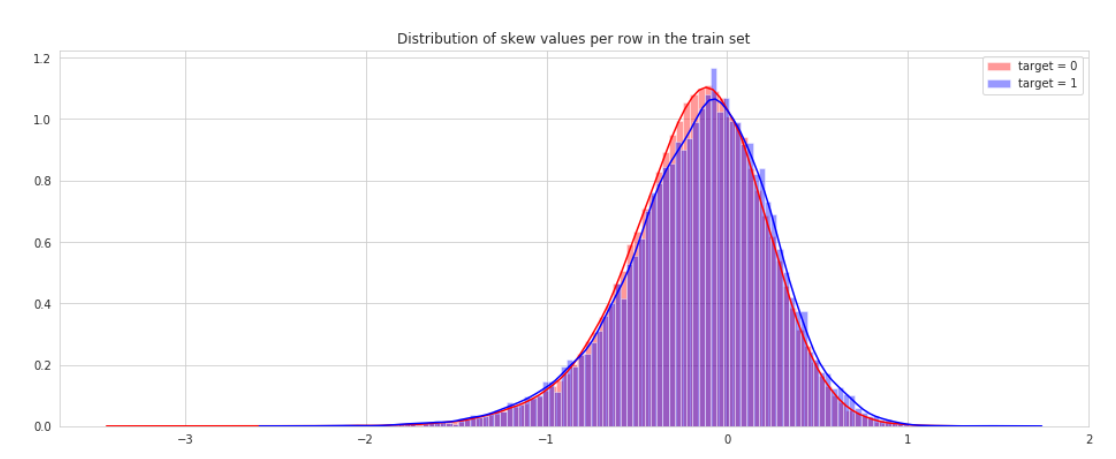
现在让我们看看每行和每列的峰度值的分布是多少。  
让我们首先看一下在训练和测试集中每行计算的峰度分布：



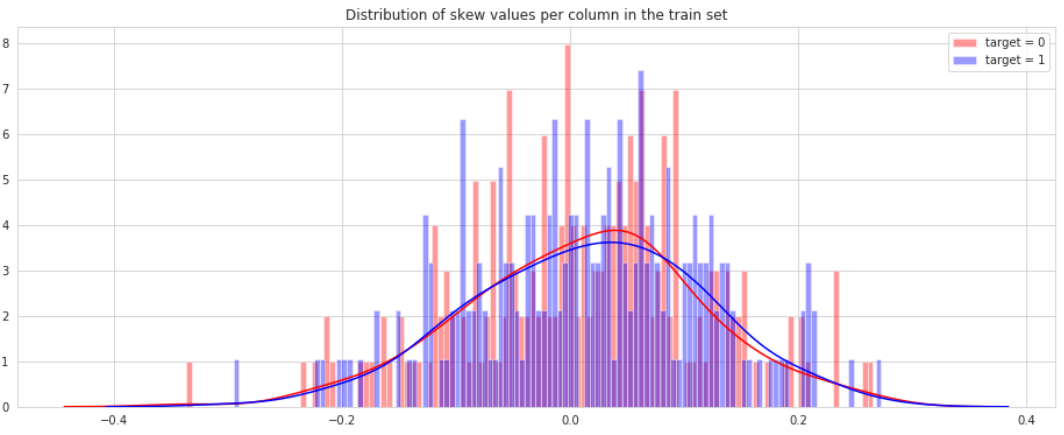
让我们首先看一下在训练和测试集中每列计算的峰度分布：



现在让我们看一下列表中行的偏度分布，分别为目标0和1的值。

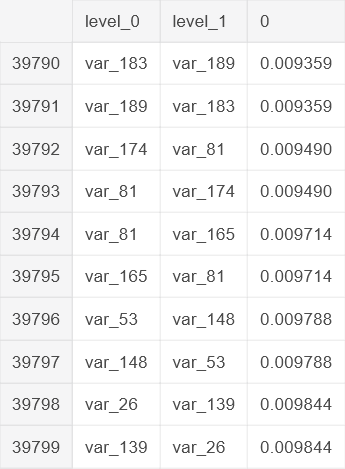


现在让我们看一下训练集中列的偏度分布，它们是针对目标0和1的值分开的。

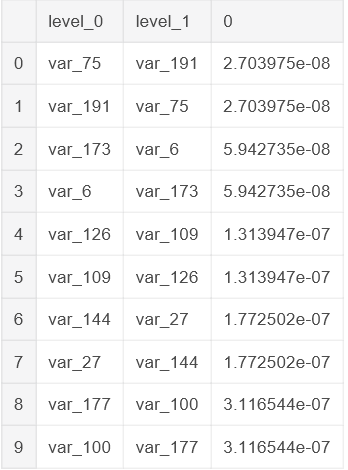


特征相关性：

我们现在计算训练集中特征之间的相关性。  
下表显示了前10个最相关的特征。



下表显示了前10个最不相关的特征。



3. 优秀算法思路

这个部分麻烦寻找竞赛结果排名至少top10的思路分享（天池大赛会在技术圈置顶，kaggle的话需要自己在交流区搜索一下）优秀算法思路可以包含多个，可以选择2-3个算法方案进行描述。

**3.1 方案一**

**3.1.1** 方案一数据预处理及特征工程部分方案

我们创建了200个类别的特征，有五个类别相对应：

该值至少出现在目标== 1且无0的数据中的另一个时间

此值至少在目标== 0且不为1的数据中出现

此值在目标== 0和1的数据中至少再出现两次

该值在数据中是唯一的;

该值在数据+测试中是唯一的（仅包括实际测试样本）

另外200个（每个原始特征一个）特征是数字的，我们称之为“not unique feat”，并且对应于原始特征替换数据+测试集中唯一的值与特征的平均值

在一些EDA之后，关键的发现是训练集和测试集中的不同值的数量不同，我开始使用LGBM，因为它快速而强大且易于使用。  
像许多人一样，我开始看到CV / LB的改进与功能的计数编码。

我查看了我的LGBM树（只有3片叶子很容易做到），并注意到树木正在使用唯一性信息。  
在这个观察之后，我开始围绕唯一性构建特征。 仅使用训练数据和“has one feat”，我可以达到.910 LB. 添加其他200“非独特专长”，。914LB。  
下一步是使用数据+测试来发现唯一值。 它在CV上工作得很好，给出了> .92x的结果，但不适用于测试！

正如许多人注意到的那样，数据和测试中每个特征的唯一值计数是非常不同的！ 所以我知道测试中有一小部分样本我无法识别会带来> .92x LB. 我此刻与Silogram合作。 第二天，他给我发了一个链接到@ YaG320的非常重要的内核（rick和morty的粉丝是最好的！）“假样本列表和公共/私人LB分裂”。 我立即明白，这是发现数据+测试中独一无二的价值的关键！

我们此时使用LGBM获得了LB .921，这些是我们最后使用的特征。

**3.1.2** 方案一模型设计、建立部分方案

我们使用标准的10倍分层交叉验证和多个种子进行最终混合。

我们使用随机扩充（重复和随机播放16次样本，目标== 1,4，目标== 0）制作LGBM并添加伪标签（2700个最高预测测试点为1，2000最低为0）。 我们的LGBM执行.92522 Public，.92332私有。

我们的第二个模型是具有特定结构的NN：

与许多人一样，这个想法是以相同的方式（即使用相同的权重集）独立地处理属于同一组（原始/具有一个/不唯一）的所有特征。 这将创建此特征值的嵌入。 我们区别于下一步：我们对这200个嵌入进行了加权平均，然后我们将其输入到密集层以进行最终输出。 这可确保以相同方式处理每个功能。 权重由另一个NN生成。 这个想法与注意力网络的做法非常相似。 当然，一切都是端到端优化的。

我们添加了动画扩充（对于每个批处理，随机播放属于目标的特征值== 1 / target == 0），它得到了.92497私有。 添加伪标签（5000最高和最低3000）增加私人至.92546。

我们的最终提交是这两个型号的混合，权重为2.1 NN / 1 LGBM。

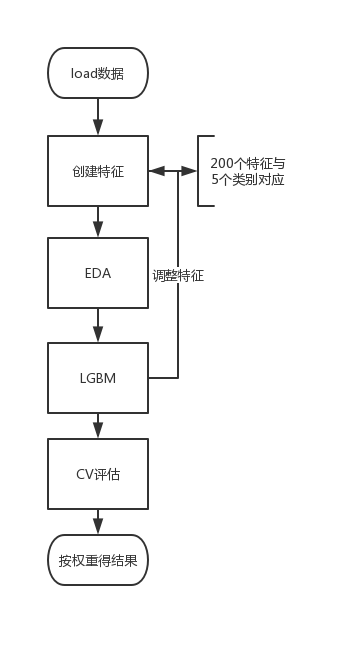
**3.1.3** 方案一结果、排名等

TOP1

Score = 0.92573

**3.1.4** 方案一算法流程图

（这个部分需要自己整理一下绘制，流程图可以用processon画）



**图3-1**

**3.2 方案二**

**3.2.1** 方案二数据预处理及特征工程部分方案

1.一半的测试集数据是虚假的

2. 计算训练集+实际测试数据的所有特征的唯一值。

如果还在计算中包含假数据，则不会在LB上显示提升。 换句话说，需要创建200个额外的特征，表示所有数字特征的唯一值计数。 例如，他们有一些明确的属性。 这可以使lightgbm模型达到0.914x水平。

3. 增强。由于特征是不相关的（Braden也在这里解释，你可以生成多个人工样本。我们使用的增强类似于此。所有区别的是组合所有先前的1,2。例如当你进行增强并且你从一列中采用随机值，您需要确保您还为该列的这些值采样唯一计数（根据组合训练集+实际测试集计算）。尝试不同的增强比率（我们发现增加正数20次和负数 2次是最佳lb性能的最佳值，你可以得到lightgbm到0.920x。

**3.2.2** 方案二模型设计、建立部分方案

1. NN比lightgbms更好，但需要Input的输入形状（shape =（200,2））。 第一个代表特征，第二个代表与这些特征相关的唯一计数。 通过这种转换+扩充，他们可以轻松地在公共LB中达到0.924。 这个nn的架构和扩充可以在这里看到

2. 对于lightgbm，它帮助在原始特征上使用（-4,4）的minmaxscaler然后执行Xn \*\* countn，其中计数被剪切在1到3之间

3. 同样对于lightgbm，它帮助在原始特征上使用（-4,4）的minmaxscaler，然后执行countn \*\* Xn，其中计数被剪切在1和3之间

4. Alaso xn \*\* - 3也有所帮助

5. 获得几乎另一个+0.0005非常重要的是伪标签

我们试图为每个列构建一个不同的模型，但它没有很好地工作。

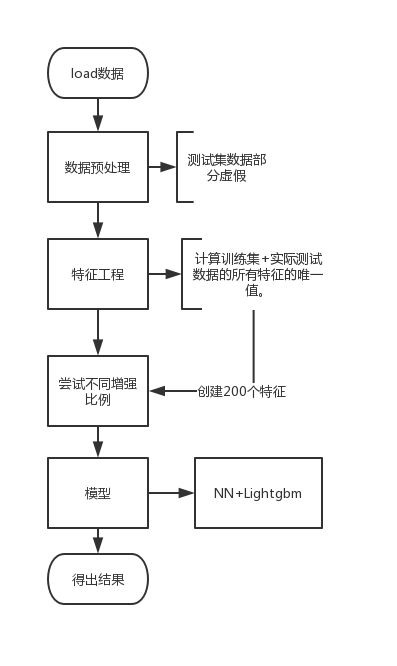
堆叠增加了大约+0.001。 NN在堆叠方面明显优于任何其他方法。

**3.2.3** 方案二结果、排名等

TOP5

Score = 0.92440

**3.2.4** 方案二算法流程图



4. 算法比较

描述几种算法的基本情况及效果对比，以表格形式：

例如： 特征工程和建模部分可以描述建模用到的基础算法名称（PCA、LSTM、时间衍生等）基本库指用到的基本库（sklearn等）

**表4-1 算法比较**

|  | **评估指标** | **特征工程** | **基础算法** | **基本库** |
| --- | --- | --- | --- | --- |
| **算法1** | **CV** | **Sklearn**  **Fastai** | **LGBM** |  |
| **算法2** |  | **Sklearn** | **NN+Lightgbm** | **Sklearn**  **Lightgbm** |

5. 总结与展望

**5.1 总结**

本题特征变量特别多，有近200个，如何选取处理特征就非常重要了，排行榜前几名的选手都选择了自己创建特征值。

**5.2 建模思路**

1.根据相关性筛选数据

2.数据清洗+数据降维

3.选取模型，调整参数（考虑Lightgbm+）

4.添加伪标签

5.CV评估

6.得出结果