|  |
| --- |
| **机器学习** |
| **梅赛德斯 - 奔驰 绿色制造** |
| 主 研 人：杜思君  参 研 人： |
| 审 核 人：  方 向：工业算法案例研究  版 本 号：A |
|  |

**二〇一九年二月**

**版本更新**

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **更新描述** | **作者** |
| 2019/02/15 | A | 初稿 | 杜思君 |

**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）** |
| 2018/10/31 | Mercedes-Benz Greener Manufacturing | 奔驰汽车拥有许多的功能（特征），那么针对如此多的可能特征组合优化测试系统的速度是复杂且耗时的，竞赛目的是优化奔驰车在测试时所用的时间 ，从而降低二氧化碳排放量，同时不降低测试的标准。 | 回归 |

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 11](#_Toc12357_WPSOffice_Level1)

[3.1 方案一 1](#_Toc32293_WPSOffice_Level2)1

[3.1.1 方案一数据预处理及特征工程部分方案 11](#_Toc16174_WPSOffice_Level3)

[3.1.2 方案一模型设计、建立部分方案 11](#_Toc32293_WPSOffice_Level3)

[3.1.3 方案一结果、排名等 14](#_Toc29922_WPSOffice_Level3)

[3.1.4 方案一算法流程图 14](#_Toc4787_WPSOffice_Level3)

[3.2 方案二 1](#_Toc29922_WPSOffice_Level2)5

[4. 算法比较 17](#_Toc22683_WPSOffice_Level1)

[表4-1 算法比较 17](#_Toc7253_WPSOffice_Level2)

[5. 总结与展望 17](#_Toc16174_WPSOffice_Level1)

[5.1 总结 17](#_Toc23198_WPSOffice_Level2)

[5.2 建模思路 17](#_Toc16619_WPSOffice_Level2)

1. 背景描述

自1886年第一辆汽车，奔驰专利汽车问世以来，梅赛德斯-奔驰(Mercedes-Benz)站在了汽车创新的重要位置上。这些创新包括有，例如具有防撞区的乘客安全单元，安全气囊和智能辅助系统。梅赛德斯-奔驰每年申请近2000项专利，这使得该品牌在欧洲高端汽车制造商中处于领先地位。戴姆勒（Daimler）的梅赛德斯 - 奔驰汽车是高端汽车行业的领导者。凭借众多功能和选项，让客户可以选择定制的他们心目中理想得梅赛德斯-奔驰汽车。  
 为了确保每一款独一无二的汽车在上路前的安全性和可靠性，戴姆勒的工程师们开发了一套强大且可靠的测试系统。但是，如果没有强大的算法方法做支撑，那么针对如此多的可能的功能组合的优化测试系统的速度是复杂且耗时的。作为世界上最大的高端汽车制造商之一，安全和效率对戴姆勒的生产线至关重要

**1.1 竞赛赛题描述**

在这次比赛中，戴姆勒公司希望参赛者们解决问题中的维度灾难（curse of dimensionality）并减少汽车在测试台上所花费的时间。参赛者们将使用代表梅赛德斯-奔驰汽车功能的不同排列的数据集来预测通过测试所需的时间。获胜算法将有助于加快汽车通过测试的速度，从而降低二氧化碳排放量，同时不降低戴姆勒公司的测试标准。

**1.2 评估指标描述**

竞赛中评估模型优劣的指标。

评估指标是值，也称为决定系数（coefficient of determination）

决定系数是用来衡量回归的好坏，换句话说就是回归拟合的曲线它的拟合优度，其计算公式如下图所示：

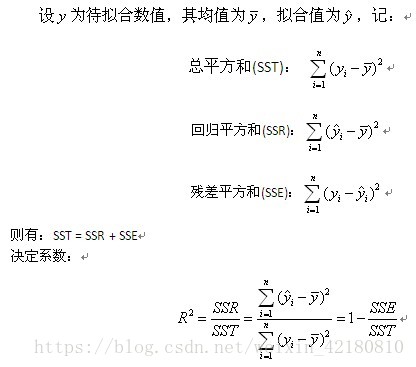


Figure 1.2-1 的计算公式（图源自CSDN博客）

1. 数据来源及描述性统计分析

**2.1 大赛数据来源**

赛题的数据源自戴姆勒股份公司（Daimler AG）旗下的梅赛德斯-奔驰汽车的测试数据。

以下是数据的超链接：

<test.csv> （测试集数据）

<train.csv> （训练集数据）

<sample_submission.csv>（提交样例）

**2.2 数据的描述性统计**

**2.2.1 数据基本情况描述：**

这些数据集包含一组匿名变量，每个变量一个代表梅赛德斯-奔驰汽车中的自定义功能。例如，一个变量可以是4WD, 也可以说增加空气悬架，或者是平视显示屏等。其中含有带字母的明确变量，只含有0/1的变量是二进制值。

目标值标记为:’y’，表示汽车通过每个变量测试所花费的时间（以秒为单位）

**2.2.2 数据字段介绍：**

以下是对训练集数据表字段的介绍，由于训练集一共含有378列数据，在此就不一一介绍说明，只作大致的综合概述。

|  |  |  |  |
| --- | --- | --- | --- |
|  | **变量含义** | **变量类型** | **缺失率** |
| ID | 汽车ID编号 | 离散（int64） | 0% |
| y | 汽车通过测试所花费的时间 | 离散（float64） | 0% |
| X0-X8 | 带字母的明确变量（共8个）  不含（X7） | 离散（object） | 0% |
| X10-X385 | 二进制变量（共计369个）  PS:这里的变量名不是完全按顺序依次排布的，部分变量不存在，如变量X9,X381没有出现 | 离散（int64） | 0% |

Table 2.2.2-1 train数据表字段介绍

测试集（test）数据表，除无y值以外含义和类型与训练集（train）数据表数据字段一致，且均无缺失值，故不再单独阐述。

**2.2.3 数据描述性统计**

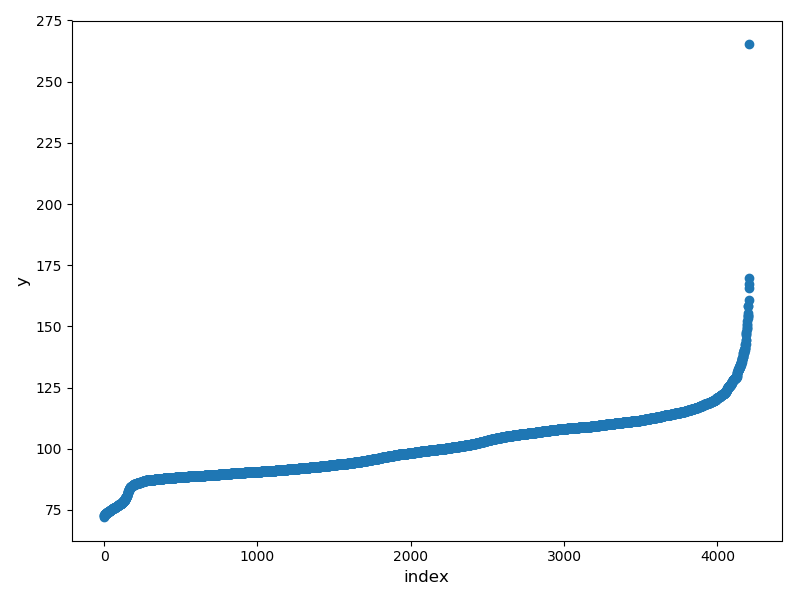
1. 对目标值y的统计分析

Figure 2.2.3-1： y在训练集中的散点分布图

从上图我们不难看出，有一个数据点，明显偏离（高于）其他数据。

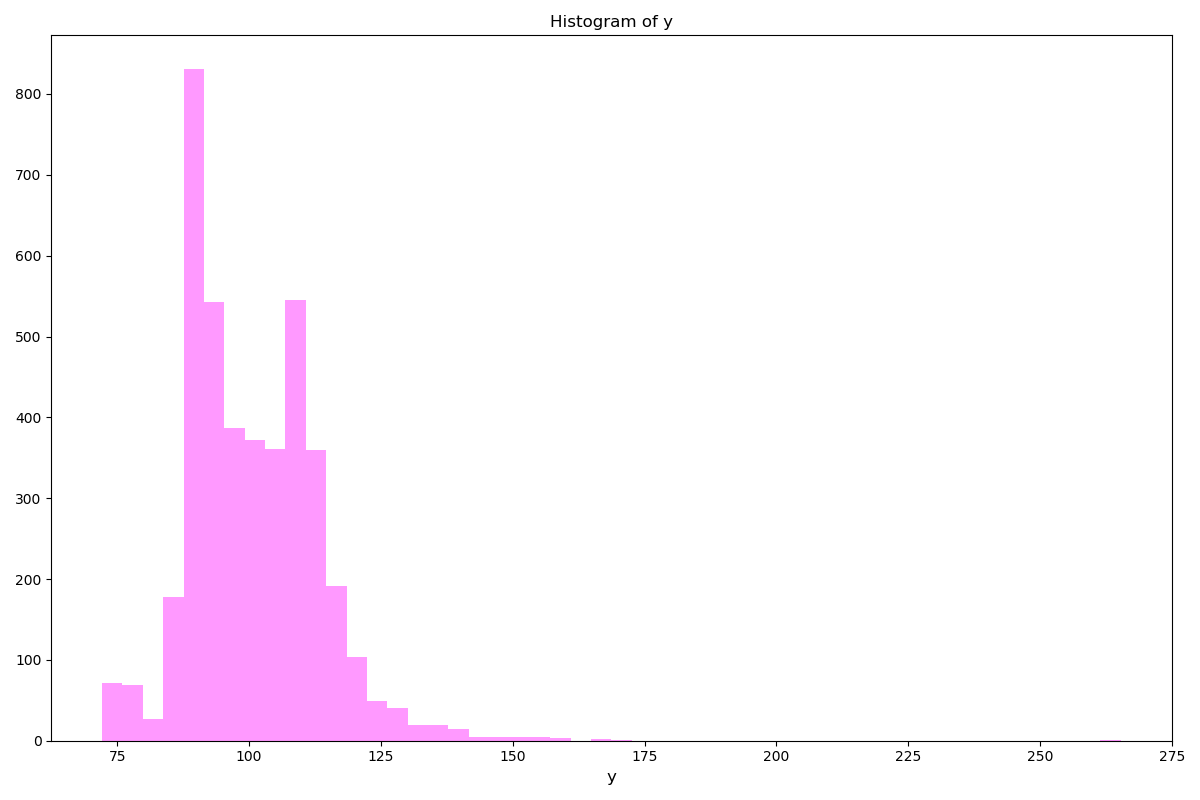


Figure 2.2.3-2: y在训练集中的直方分布图

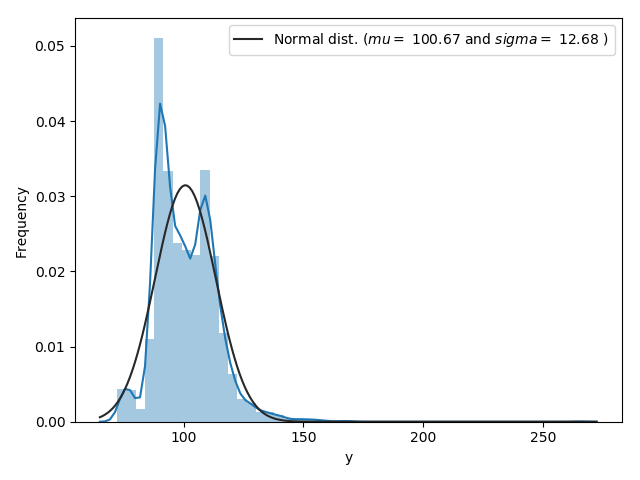


Figure 2.2.3-3: y在训练集中的正态分布图

由Figure 2.2.3-2与Figure 2.2.3-3，可以了解到出y值大致的分布情况

1. 对整数型变量的统计

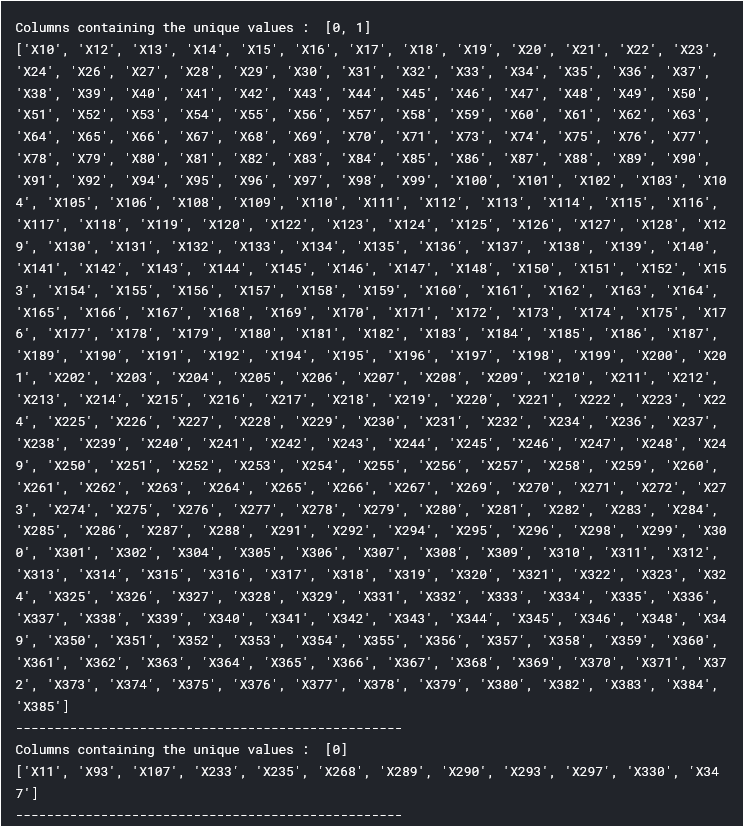
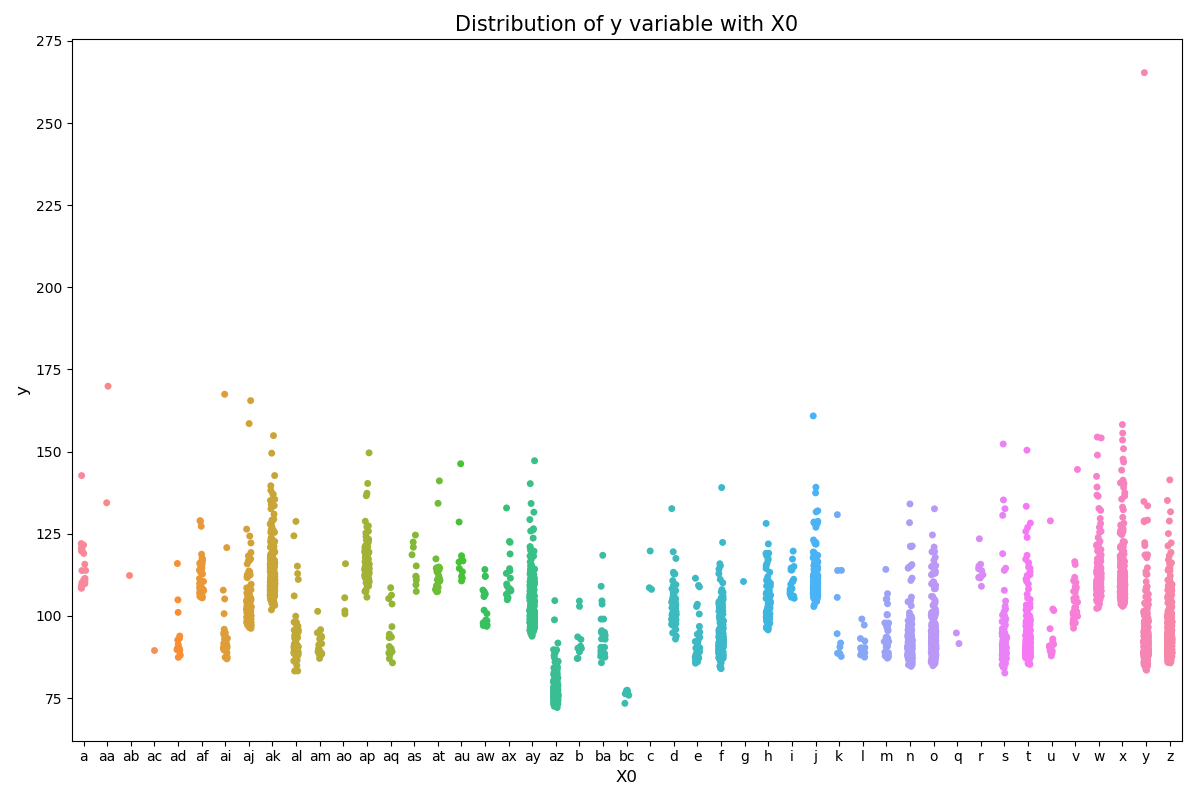
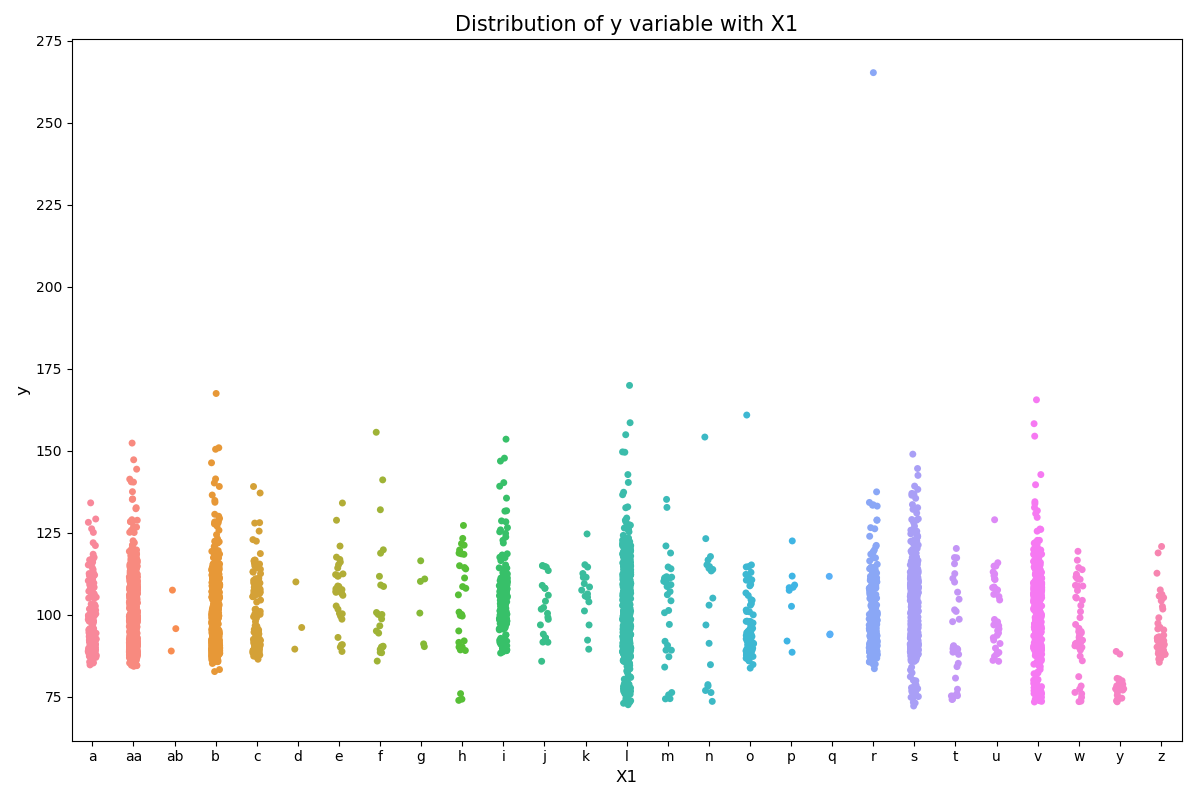
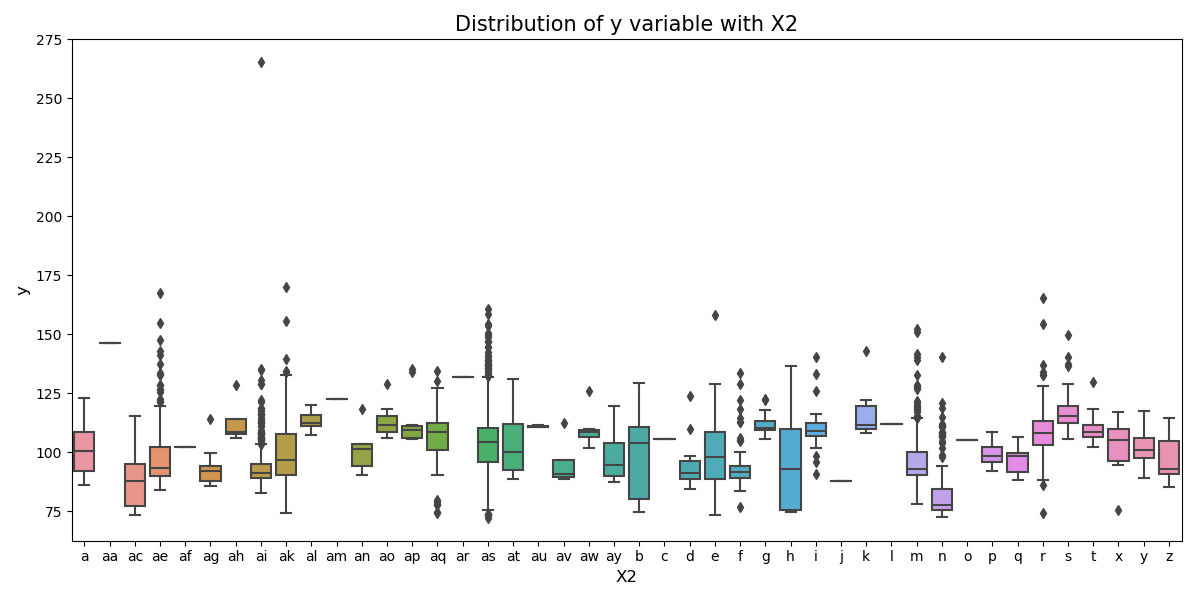
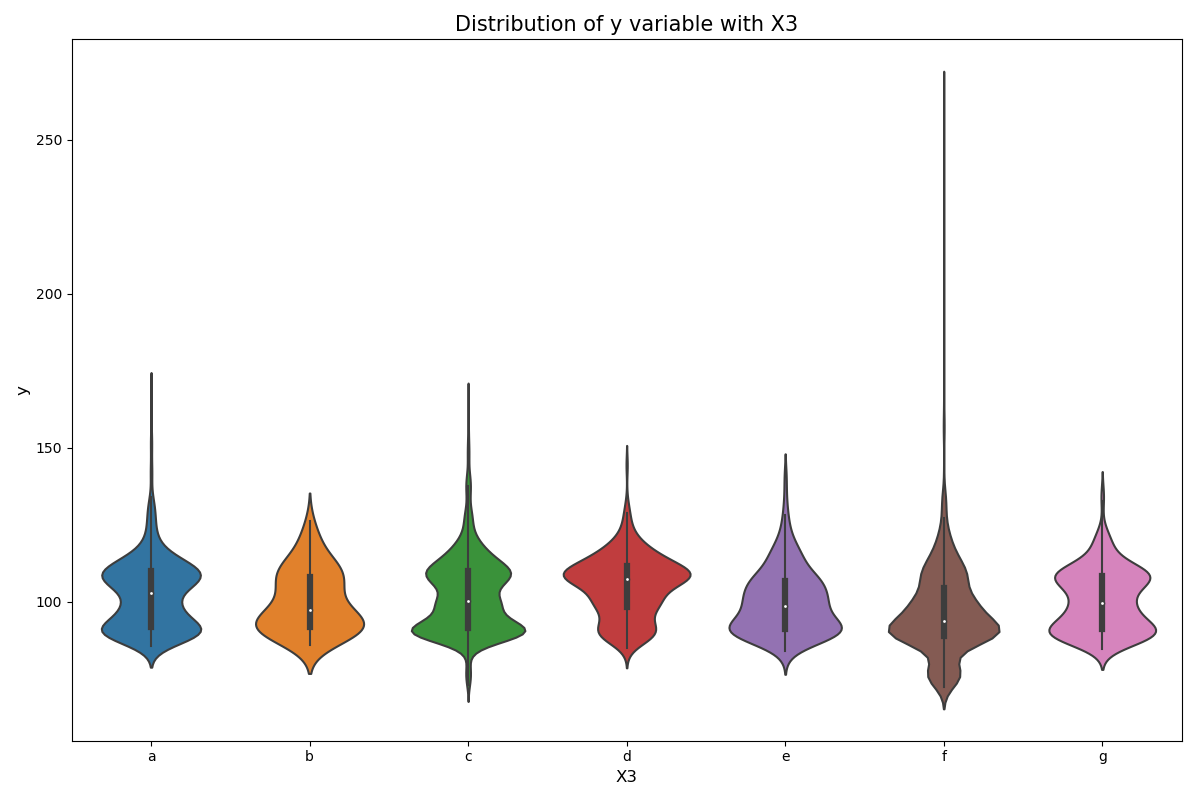


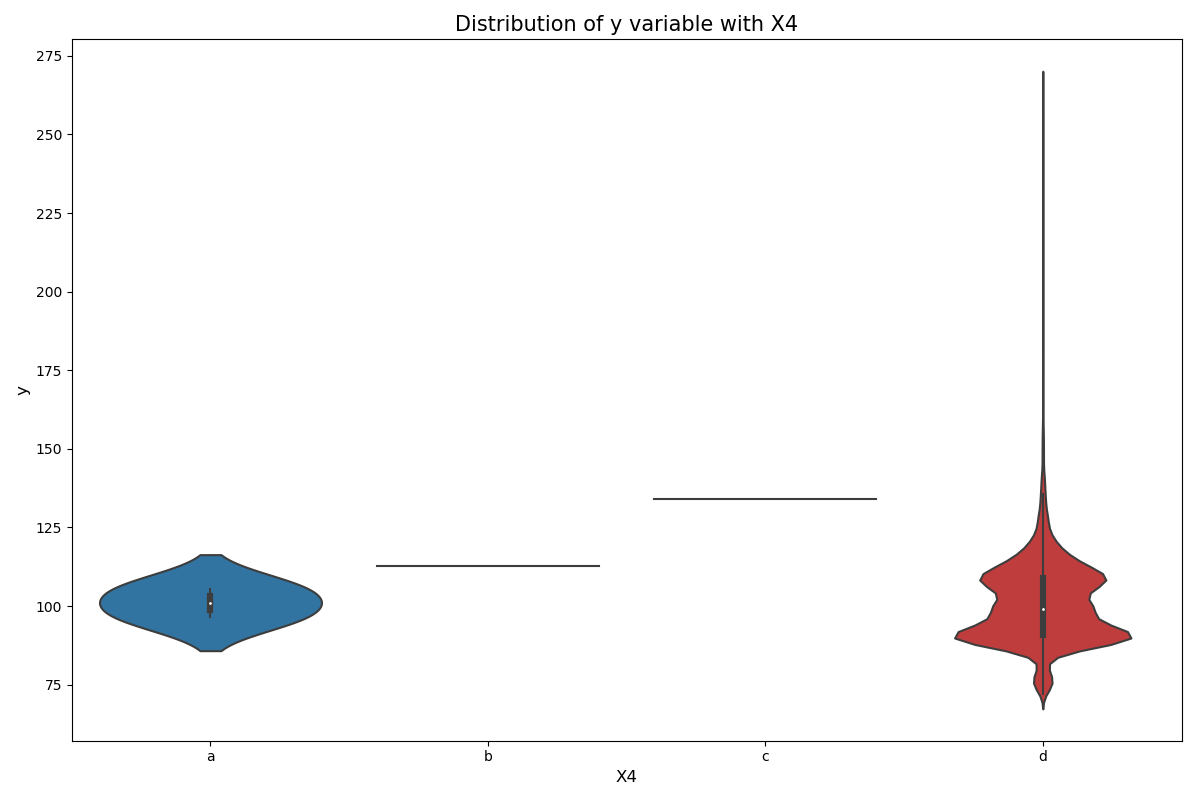
Figure 2.2.3-4: integer型变量的分布情况

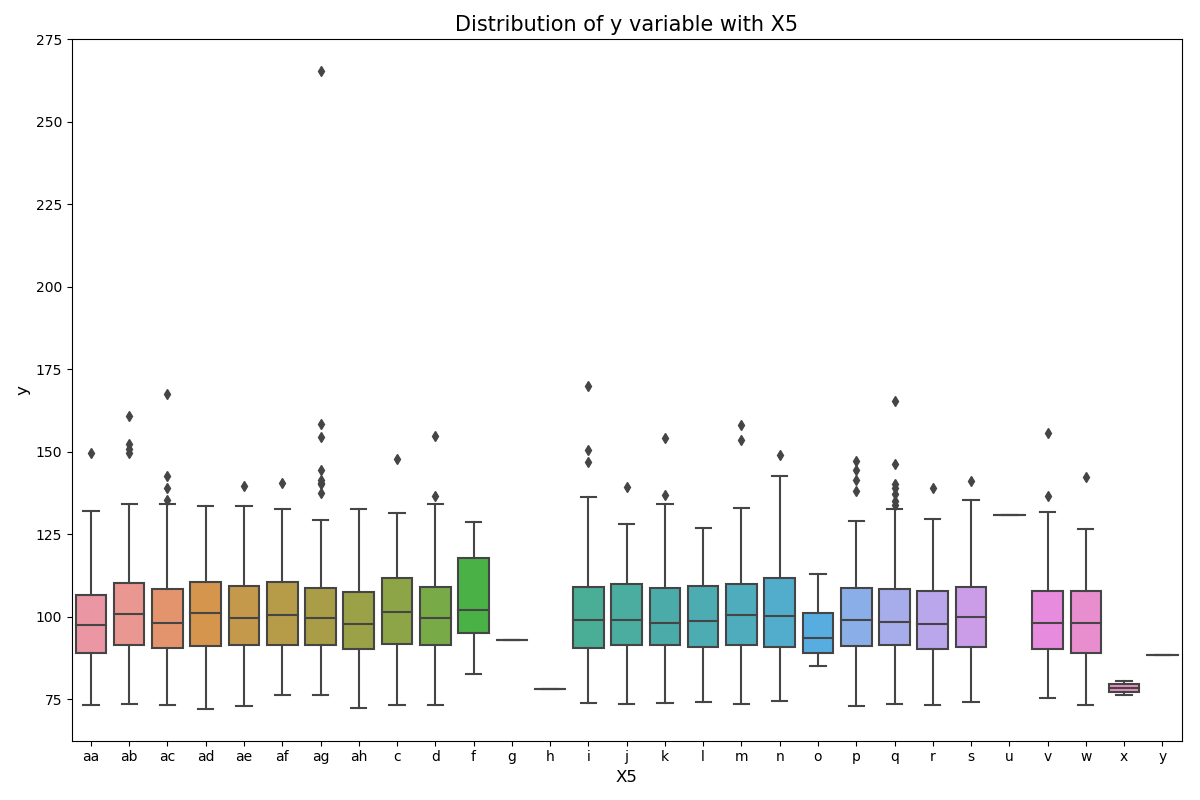
由Figure 2.2.3-4 我们不难看出，所用的整数型变量列均为二进制变量，其中部分变量行仅含有0值（可以考虑在模型中删除这些变量行）。

1. 对字母型变量的统计分析(结合y的分布)









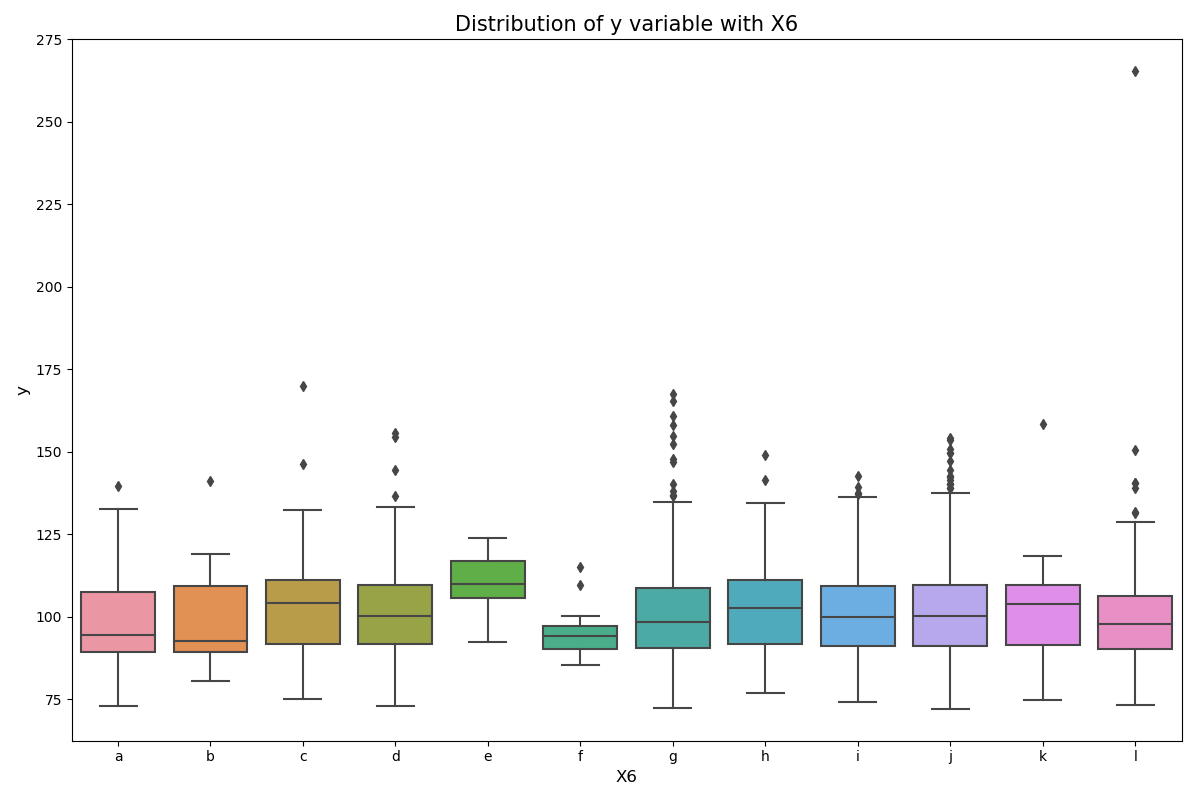
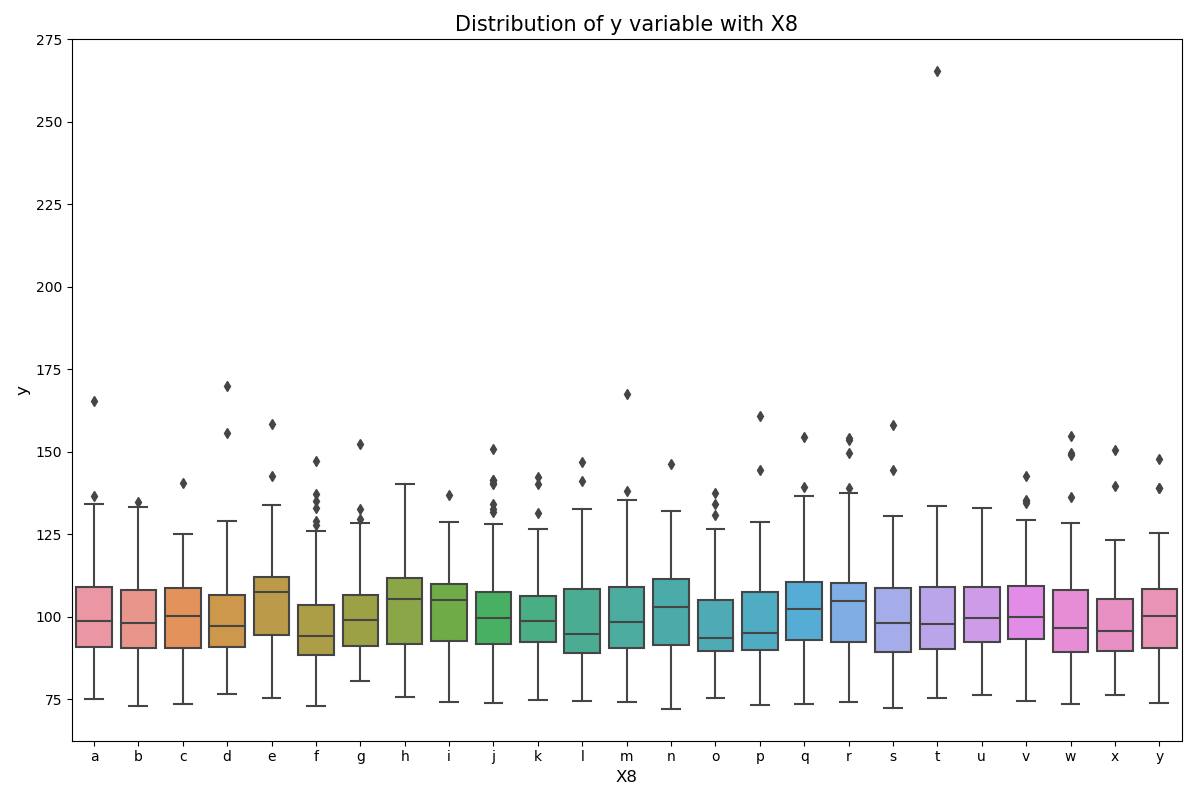
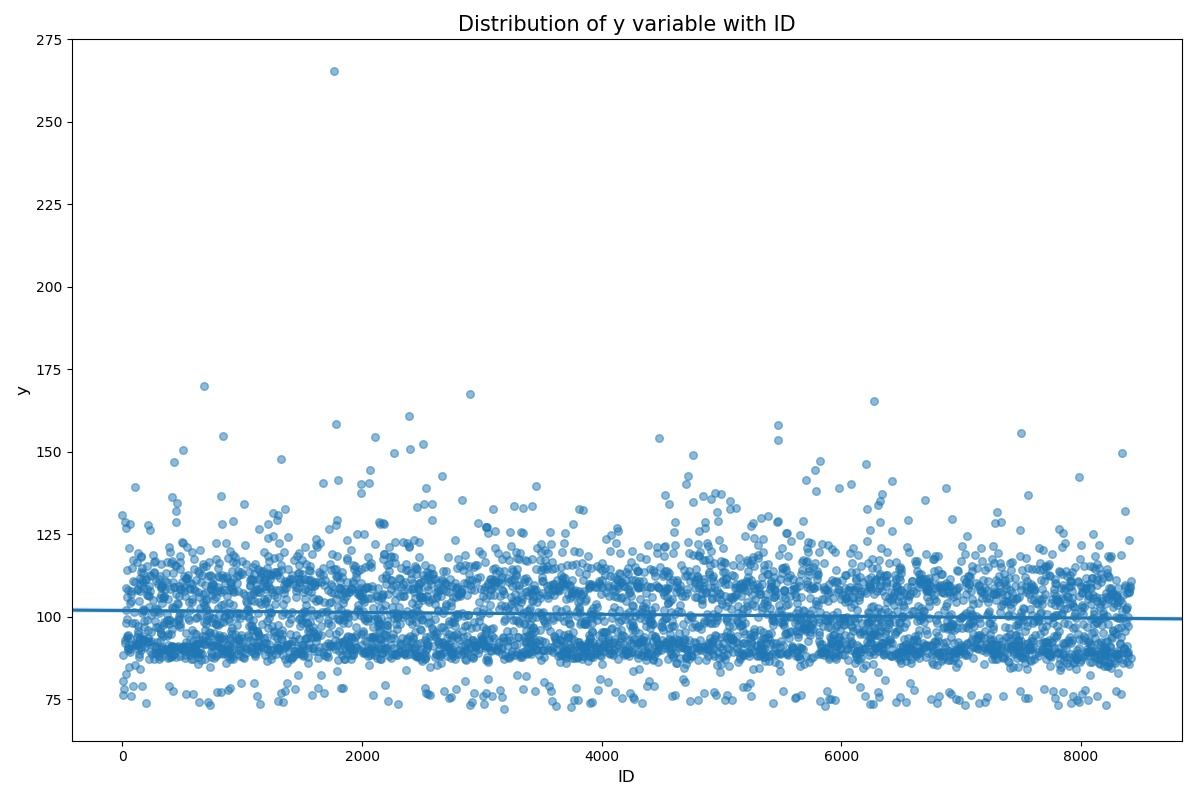


Figure 2.2.3-5 字母型变量字母出现与y值分布统计图（合集）

使用了多种绘图方式，选择了最适合该变量的图示，以便观测y值在每个变量上每个字母型取值上的分布情况。

1. 变量ID的统计分析

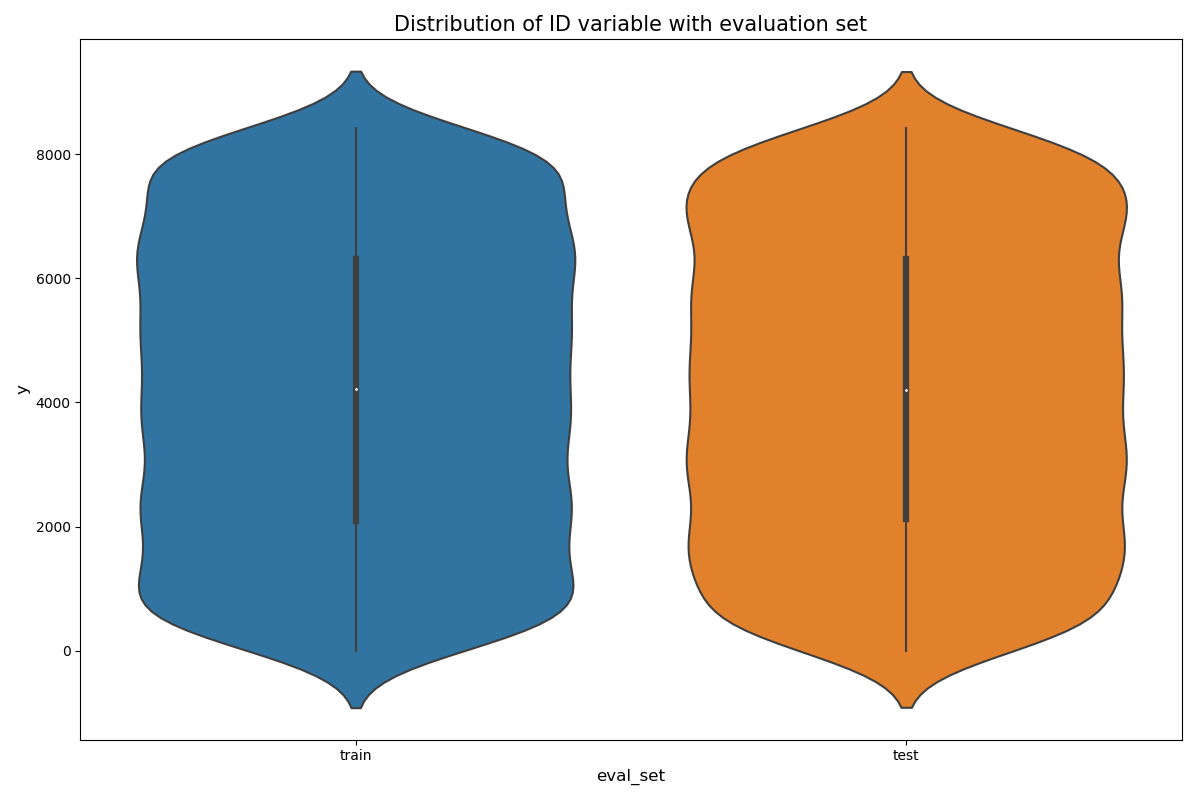


Figure 2.2.3-5 变量ID与y值间分布的统计图（1.训练集中；2.训练集与测试集的对比）

从图1中我们可以看出ID变量似乎略有下降的趋势，图2看起来像是ID变量在训练集和测试集中的随机划分。

e） 重要特征

这里采用xgboost model和random forest两种算法得到对特征的评分，从而选取重要的特征

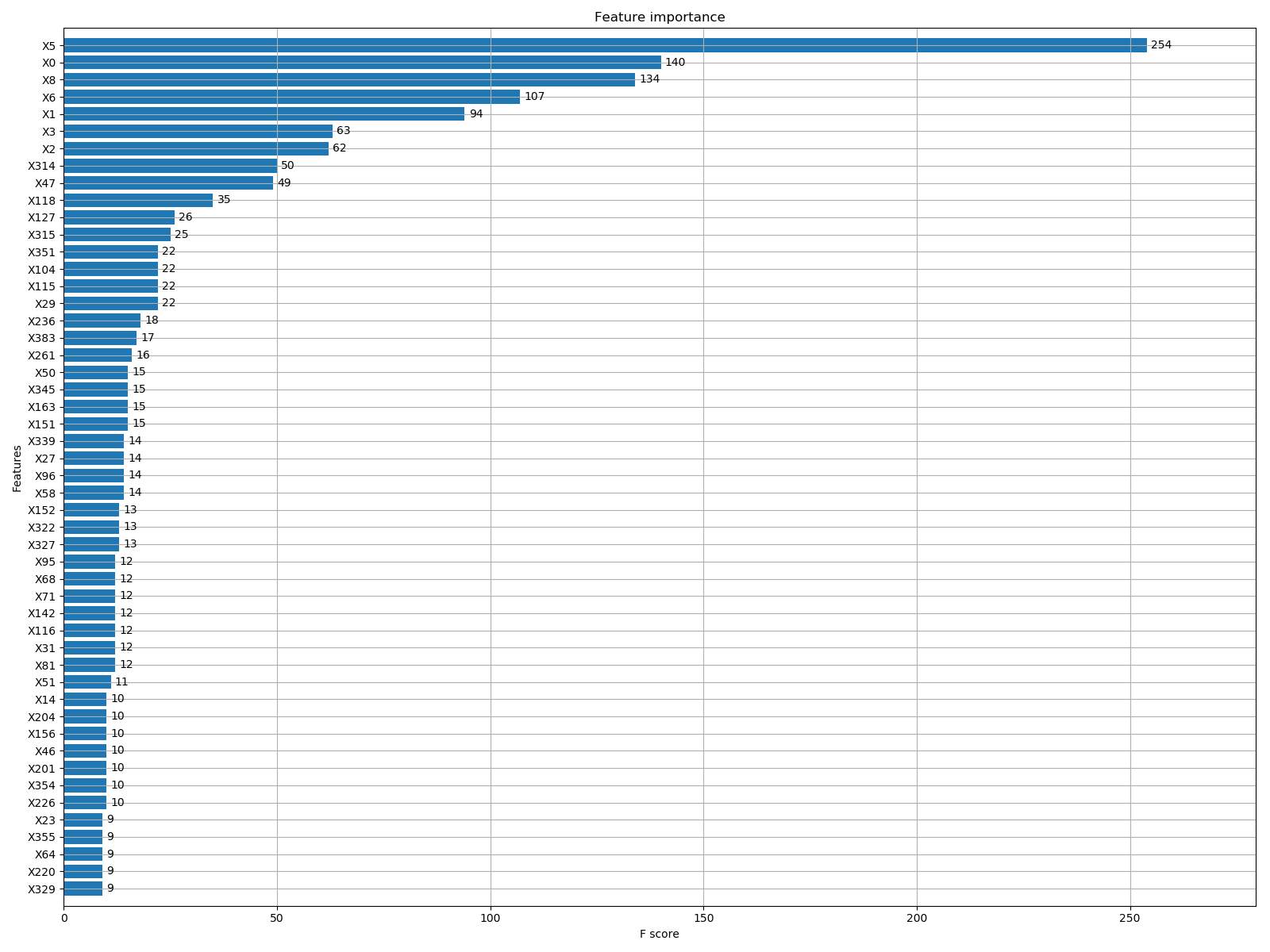


Figure 2.2.3-6 用xgb得到的特征变量评分

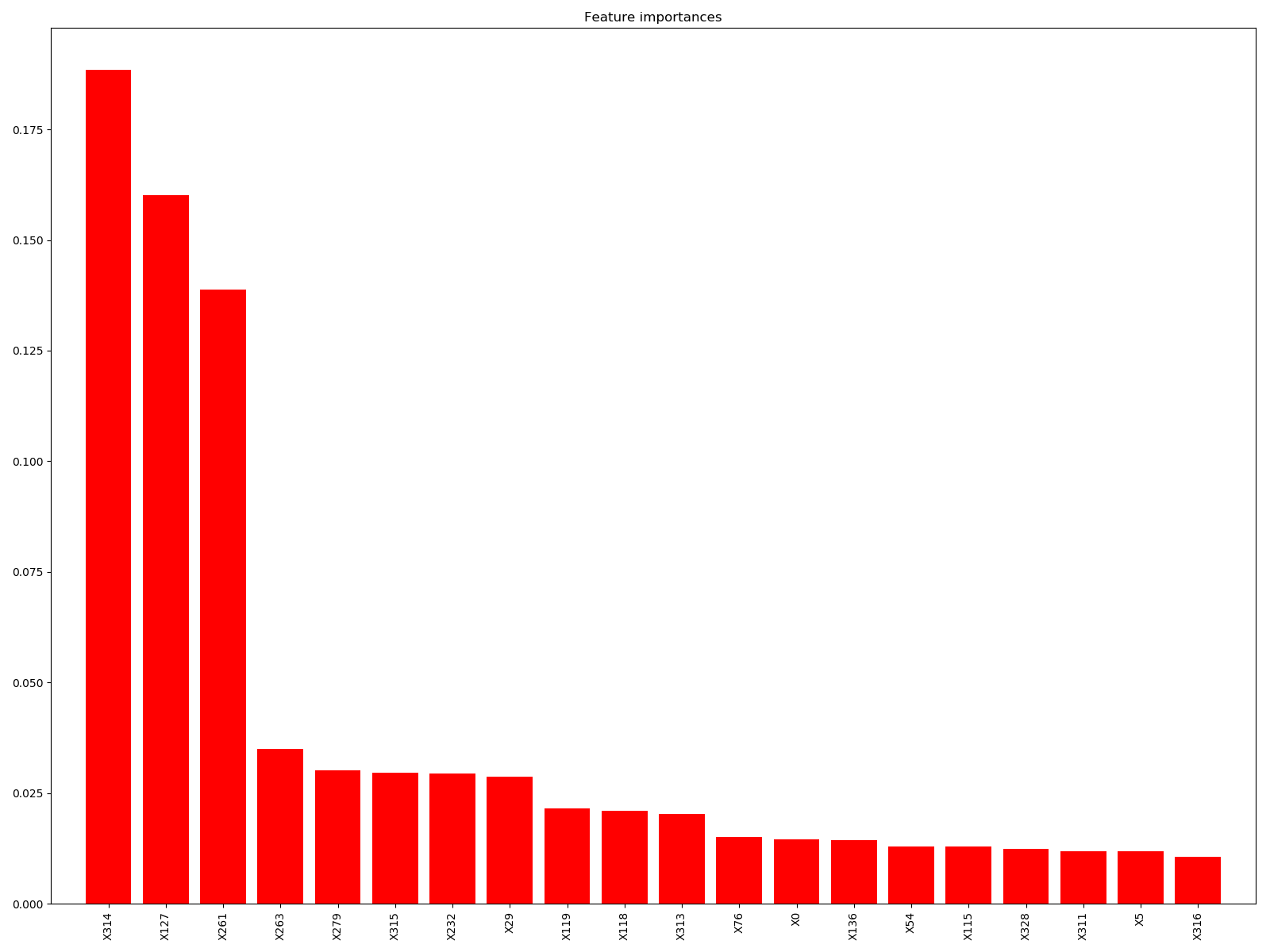


Figure 2.2.3-7 由random forest得到的特征变量评分

由上面两图可以看出，在由xgb model 得到的重要特征里确定的分类型字母变量占据较为重要的位置，但用随机森林算法得到的重要特征与用xgb model得到的结果有比较大的差异。

3. 优秀算法思路

* 1. **方案一**

**3.1.1**方案一数据预处理及特征工程部分方案

1. **对object 型变量进行序列化数值化:**

从sklearn调用LabelEncoder()函数对不连续的文本进行编号，由图*Table2.2.2-1*可以看出object型即前面提到的字母型分类变量（X0-X8,无X7），使用fit命令对训练集和测试集同时作用后，再用transform命令分别作用与训练集和测试集。

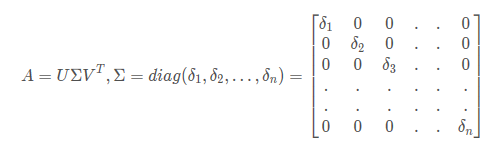
1. **数据降维**

* **截断奇异值分解（TruncatedSVD）**

1. 功能/原理介绍：

截断奇异值分解（Truncated singular value decomposition，TSVD）是一种矩阵因式分解（factorization）技术，将矩阵 M 分解成 U ， Σ 和 V 。它与PCA很像，只是SVD分解是在数据矩阵上进行，而PCA是在数据的协方差矩阵上进行。通常，SVD用于发现矩阵的主成份。对于病态矩阵，目前主要的处理办法有预调节矩阵方法、区域分解法、正则化方法等，截断奇异值分解技术TSVD就是一种正则化方法，它牺牲部分精度换去解的稳定性，使得结果具有更高的泛化能力。

对于原始数据矩阵A(N\*M) ，N代表样本个数，M代表维度，对其进行SVD分解：



TSVD与一般SVD不同的是它可以产生一个指定维度的分解矩。例如，有一个 n×n 矩阵，通过SVD分解后仍然是一个 n×n 矩阵，而TSVD可以生成指定维度的矩阵。这样就可以实现降维了。

1. 调用

从sklearn.decomposition库里调用TruncatedSVD，使用fit\_transform 命令作用于训练集（去除y值），再使用transform命令作用于测试集。得到有TSVD作用后的训练集和测试集。

* **主成分分析(PCA)**

1. 功能/原理介绍：

PCA（主成分分析）是一种在尽可能减少信息损失的情况 下找到某种方式降低数据的维度的方法。通常来说，我们期望得到的结 果，是把原始数据的特征空间（n 个 d 维样本）投影到一个小一点的子 空间里去，并尽可能表达的很好（就是说损失信息最少）。

1. 调用过程同TSVD

* **快速独立成分分析（FastICA）**

1. 功能/原理介绍：

FastICA算法也叫固定点算法(Fixed-Point)算法，是一种快速寻优迭代算法，采用批处理的方式，每一步迭代由大量的样本数据参与运算。FastICA有基于峭度，基于似然最大，基于负熵最大等形式。

b) 调用过程同TSVD

* **高斯随机投影（**GaussianRandomProjection**）**

1. 功能/原理介绍：

通过将原始输入空间投影到随机生成的矩阵来降低维度。

1. 从sklearn.rondom\_projection中调用GaussianRandomRrojection()，然后对训练集和测试集作用

* **稀疏随机投影（**SparseRandomProjection**）**

1. 功能/原理介绍

使用稀疏随机矩阵，通过投影原始输入空间来降低维度。稀疏矩阵可以替换高斯随机投影矩阵来保证相似的嵌入质量，且内存利用率更高，投影数据的计算更快。

1. 调用方式同上

**3. 增强特征空间**

融合数据降维之后得到的五个训练集和测试集（TSVD, PCA, FastICA , GRP, SRP），构造最终的训练集和测试集（但构造出来的finaltrainset和finaltestset仅用于后续的stacked model）

**3.1.2** 方案一模型设计、建立部分方案

**1. 构建xgboost模型**

a) 参数设置：作者没有展示调参的过程，这里就直接展示最终参数的设置

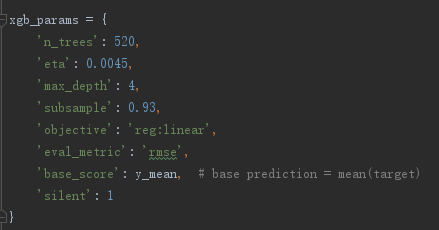


Figure3.1.2-1模型的参数设置

b) 模型训练：

将xgb模型迭代次数设置为1250（即num\_boost\_round=1250），用仅进行过数值化后的训练集（先将其储存的对象Dmatrix中）进行训练。然后对测试集继续预测得到y\_pred。

1. **构建stacking模型**

a） 构建stacking模型：

定义一个stacking 类（class）, 使用BaseEstimator和TransformerMixin 作为开发接口。

b) 模型训练：

使用带正交化的LassoLarsCV模型（即设置参数normalize=True）和梯度提升回归算法（GBR）作为初级学习器，以默认参数的LassoLarsCV作为次级学习器，传入stacking类，得到模型。用之前由五种数据降维方法融合得到的finaltrainset来训练这个stacking模型，最后使用训练好的模型得到预测结果results.

1. **模型评估方法**

在本地计算R2值，用训练集的y值与stacking模型对训练集预测得到的结果乘以一个系数加上 xgb模型对训练集预测得到的结果乘以一个系数来计算本地的R2值。（两系数和为1，具体计算公式为stacking\_y\*0.2855+xgb\_y\*0.7145）

1. **模型提交结果**

最终的提交结果是由xgb模型得到的结果与stacking模型得到的结果简单融合而成，融合公式为xgb结果y\_pred\*0.75+stacking结果results\*0.25,

上传最后的结果。

**3.1.3** 方案一结果、排名等

得分：R2 值：0.55424

排名：rank: 6/3835

**3.1.4** 方案一算法流程图

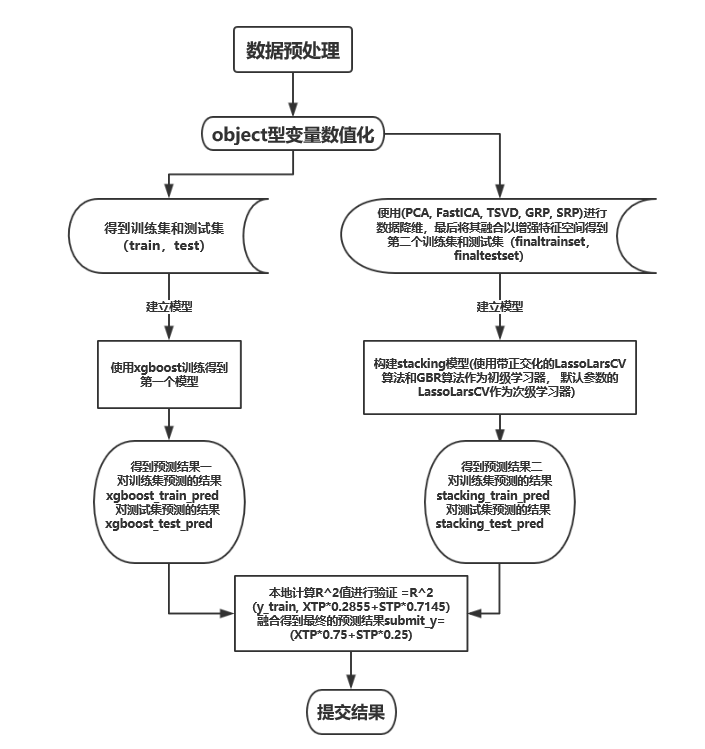


Figure3.1.4-1:算法流程图

**3.2 方案二**

该方案来着top1选手的分享描述，由于分享中只有部分有效信息和选手的大致流程，故只能做大致的描述。

**3.2.1**方案二数据预处理及特征工程部分方案

1. 特征探索

数据特征分析：

1. 匿名（没有带名字的提示）
2. 只有二进制和分类（容易过度拟合）
3. 只是二进制变量的双向交互：67528
4. 相对于特征的数量的数据集较小
5. 高质量数据：显然无需处理受污染的数据

2. 特征组合/特征相关性, 特征重组

1. 成对变量：（X314, X315）; (X118, X119); (X47, X48);
2. 对ID变量的分析：需要保留ID型变量
3. 三元相关变量（X118, X314, X315）;
4. 独立变量： X279, X232, X261, X29;
5. 将变量X122加到X128上；
6. 将X0的11个等级将其重新编码为新的一个变量，将这个变量命名为rexX0
7. 应用了分类特征的单热编码（one-hot），即原始X0到X8以及为交互变量创建的。如果一个总和超过50，则保留一个热编码变量。
8. 数据转化：

使用决策树算法（decision tree）将字母型分类特征转换为数值型特征。

**3.2.2** 方案二模型设计、建立部分方案

1. 模型建立

使用XGBoost算法建立了两个模型，分别是模型A和模型B；模型A提取了900个特征来训练；模型B使用了900+1个特征，最后stacking方法得到预测结果。

2. 特征筛选：

* + - * 1. 使用xgb算法对特征重要性进行评估（如Figure2.2.3-6），模 型A 选取了前229个变量；模型B选取了208个变量。
        2. 在模型中包含增益高于减值的特征;使用的切割值为0.1％，模型A为中为53个，模型B中为47个 。

3. 模型验证：

两种模型均使用30验证交叉折叠，在整个模型建立过程中使用xgboost和30倍的交叉折叠验证。

4. 模型训练：

a） 使用筛选好对模型A的训练集对A模型进行训练，得到模型A；

b） 在模型B的建立中，使用30折交叉验证用来对做30折的stacking融合，得到模型B。

5. 预测结果：

将模型A和模型B得到的预测结果取平均值，得到最终的提交结果。

**3.2.3** 方案二结果、排名等

得分：R2值：0.5555；

排名： rank：1/3835。

**3.2.4** 方案二算法流程图

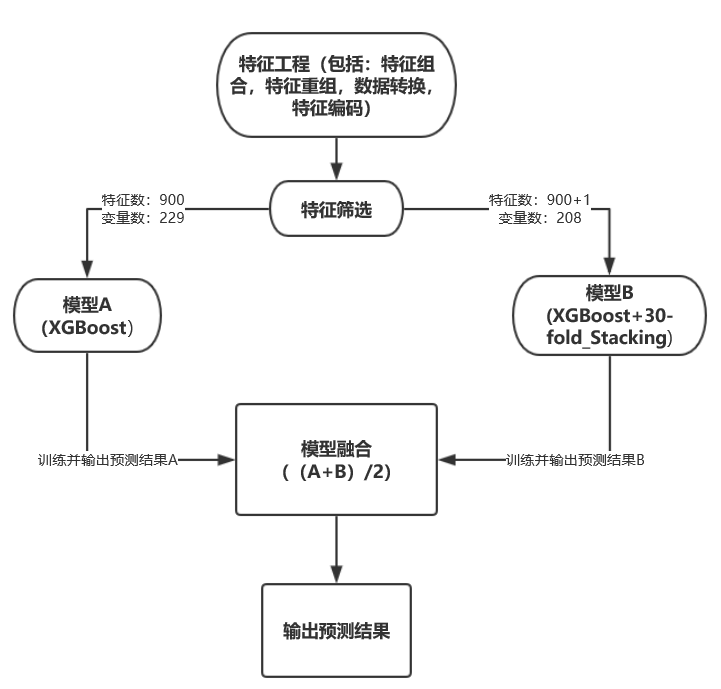


Figure3.2.4-1:方案二流程图

4. 算法比较

|  | **评估指标** | **特征工程** | **基础算法** | **基本库** |
| --- | --- | --- | --- | --- |
| **算法1** | **R2值** | **PCA, FastICA等** | **Xgboost，stacking** | **Sklearn，sgboost** |
| **算法2** | **R2值** | **特征组合，选取等** | **Xgboost** | **Sklearn，xgboost** |

此类算法在模型建立方面都大同小异，大多数参赛者都会使用Xgboost和stacking进行模型堆叠以获取更高的评分，而制胜之处还是在参赛者的特征工程，排名第一的参赛者显然做了更加细致的特征工程，取得了更好的成绩。

5. 总结与展望

**5.1 总结**

本题是一道典型的回归问题，难点在于维度灾难，即数据集有许多匿名的未知变量且相对数据的维度来说，数据量较小，这就给训练带来了很大难度，并且非常容易过拟合。所以会考虑对数据进行降维，筛选主要的特征变量进行训练。

在收集整理的过程中也学习到了这些优秀参赛者的一些方法和经验，为今后的学习和工作提供了宝贵的经验。

**5.2 建模思路**

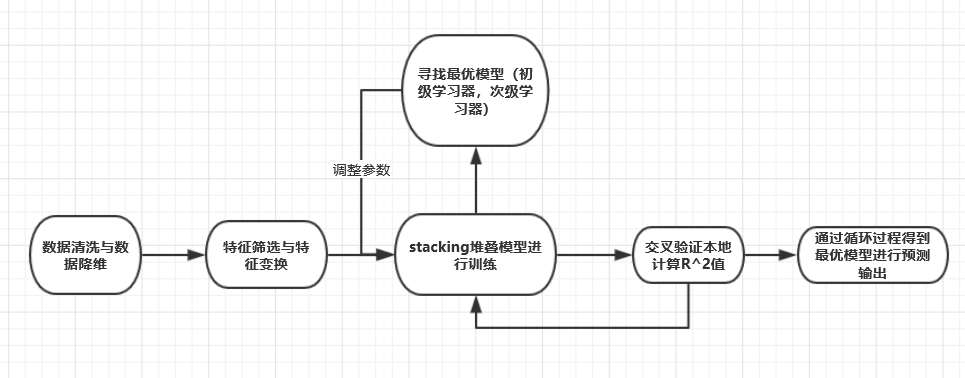


Figure5.2-1大致的建模思路