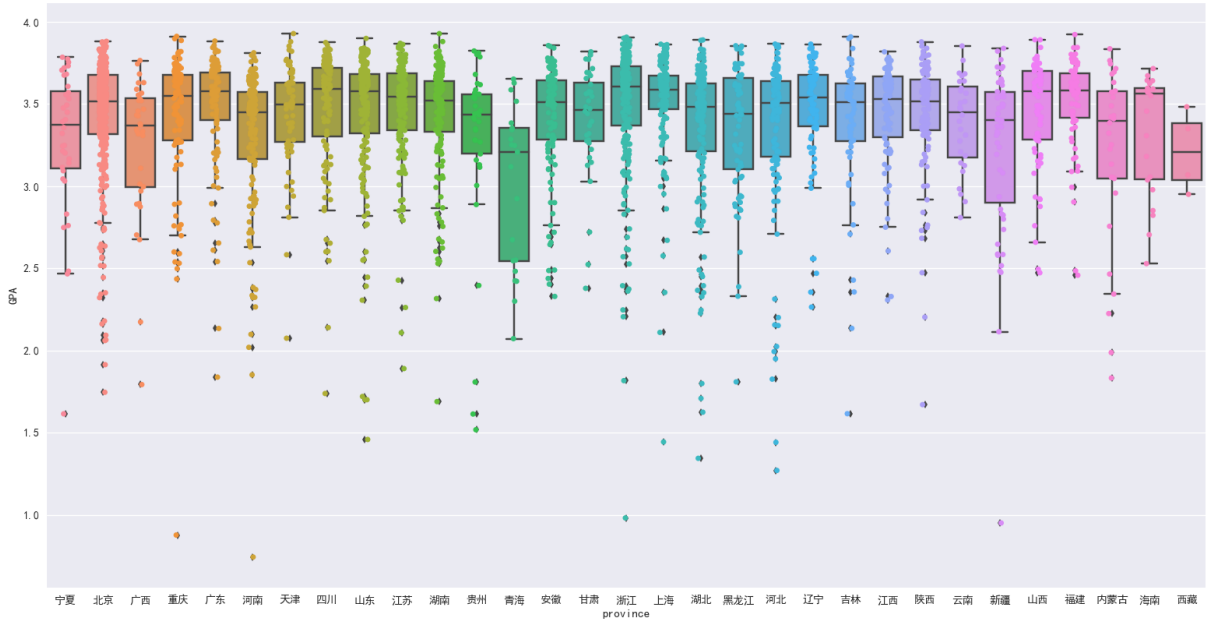
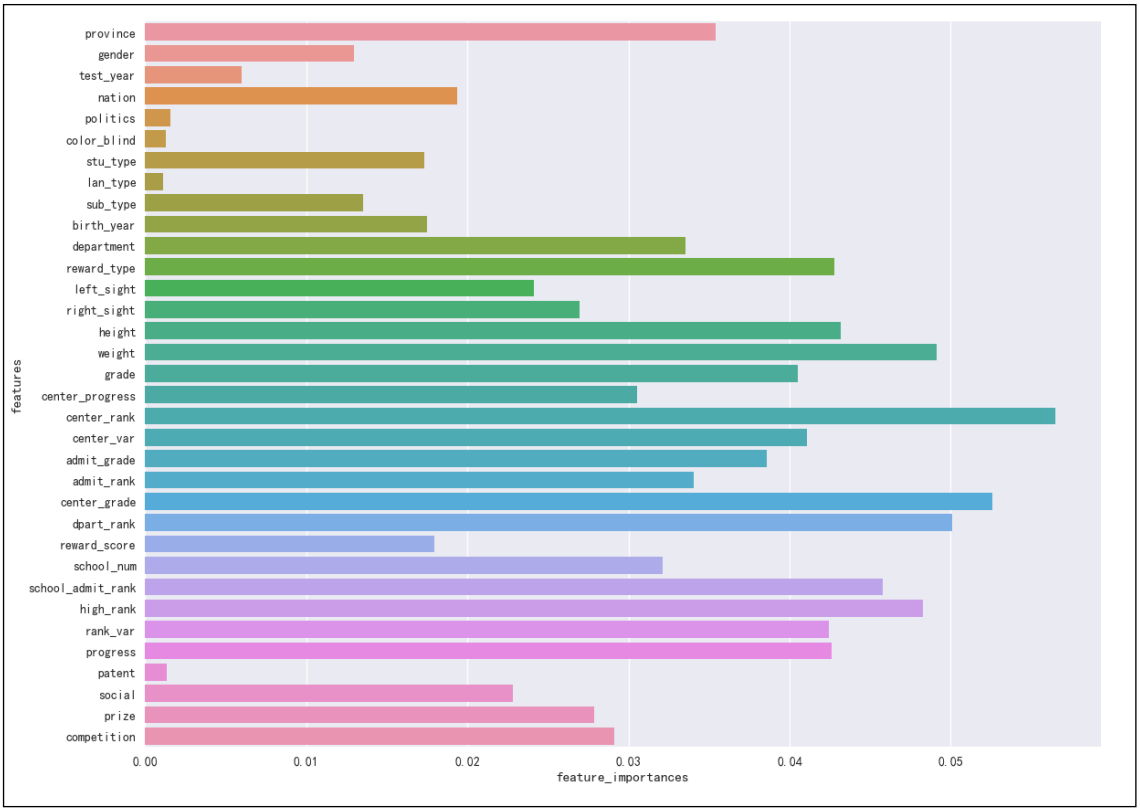
GPA PREDICTION

Wenjing Wang, Dexin Liu, Hao Jin

3rd November, 2017 — 10th December, 2017



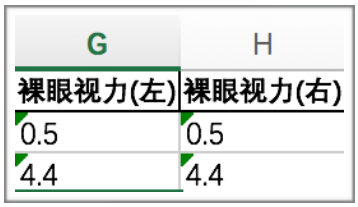


Introduction

GPA (Grade Point Average)，是北京大学广大劳苦本科生关心的重要指标之一。本次项目的主要目的就是寻找2015，2016级同学们在校期间的 GPA情况与同学们入学时的基本信息之间的关系。在此声明，我们的所有实验结果只是表明GPA与某些特征之间存在相关性，并不能表明两者之间任何的因果关系，更不能作为任何“院系炮”，“地图炮”等的理论依据。

G	H
裸眼视力(左)	裸眼视力(右)
500	500

图(a) 裸眼视力错误数据



G	H
裸眼视力(左)	裸眼视力(右)
0.5	0.5
4.4	4.4

图(b) 裸眼视力数据标准不统一

W	X	Y	Z
大类	高三排名	成绩方差	进步情况
筑梦	0.0028169	1.0607E-06	0.00044864
普通			
自主	0.01612903	7.9416E-05	0.01026966

图(c) 高三排名数据大量缺失

实验采用的原始数据集来源于GPA.tar.gz。原始数据内容不在这里赘述，但是我们要在这里说明原始数据充满了各种各样的噪声。首先，一些数据本身天然存在错误，例如图(a)中被记录为500的裸眼视力。其次，一些数据本身存在标准的不统一，例如图(b)裸眼视力存在新表，老表两种标准。最后，一些数据本身在录入的过程中就有严重的缺失：例如图(c)高三排名等来自高中的信息有大量的缺失。

然而，即使清理了原始数据之中的噪声，原始数据仍旧是不适宜直接作为预测的直接依据。这一点直观上并不难去理解。直接将原始数据作为预测依据，传统回归模型无法考虑特征之间的交叉关系，例如不同省份高考的高考分数总分不同，神经网络模型虽然可以考虑特征之间的交叉关系，但是数据量过少，会导致神经网络模型在训练集上过度拟合，这些都不是我们所期望的。这样的情况，就意味着我们要主动提取特征之间的交叉特征，以及显式地求解一些基于全局的信息，比如说高考排名。

在生成比较充分的特征之后，我们开始使用提取的特征对模型进行训练。总体上，我们把训练集分划成Train,Validation两部分，通过Validation上的mse大小来进行模型的调参。进一步的实验也表明，单一模型对GPA的拟合显得力不从心，我们采用了包含多种模型的boosting方法以求性能上的进一步提高。

Data Preprocessing

I. 数据清理

一、错误数据的清理

1）在“裸眼视力(左)”、“裸眼视力（右）”两栏中，有一些数据使用了非“.”符号，我们将其纠正为“.”符号。还有一些错误数据如图(a)，我们进行了纠正。

2）在“高三排名”一栏，我们发现了“13.65231788”、“87.02150538”等异常数据。我们删除了这些数据

二、数据的格式统一和标准化

1）在“裸眼视力(左)”、“裸眼视力（右）”两栏中，我们发现数据有的使用了旧视力表，有的使用了新视力表。我们将所有视力数据都转化成新视力表。

2）在“科类”一栏，我们发现，同样是文史，数据有“文科普通类”、“文史”、“文科”、“文史类”多种表达，并且与学科类型一栏重复度很大。我们选择直接删除“科类”一栏。

三、空缺数据的填充

1）在“高三排名”、“成绩方差”、“进步情况”三栏，我们对用0值填充和用均值填充两种方法均进行了尝试。

2）在“成绩”一栏，如果一个学生的值为空缺，我们使用他在“投档成绩”一栏中的值进行填充。

3）我们发现，有三个保送生在“成绩”和“投档成绩”两栏都是空缺，并且其中只有一人在测试集里面。我们选择直接将他们从训练集中剔除，并在提交时在全北大GPA均值的基础上加一个随机小量赋给在测试集的那个保送生一个GPA。

4）其他空缺值，我们都用0来填充。

II. 交叉特征提取

一、分类别中心化：

1）对于同一高考年份，同一生源学校的同学，我们将其“高三排名”、“成绩方差”、“进步情况”这三栏进行了中心化以及标准化。[center_rank, center_var, center_progress]

2）对于同一高考年份，同一高考省份，同一考试类别的同学，我们将其“投档成绩”，“成绩”这两栏进行了中心化以及标准化。经历了中心化和标准化，我们可以在接下来的数据处理中无视高考年份，高考省份以及考试类别的区别，直接对所有同学的高考分数进行处理。[center_grade]

3）对于同一院系，同一高考年份的同学，我们将其“竞赛成绩”，“获奖数目”，“社会活动”这三栏进行了中心化以及标准化。[center_competition, center_reward, center_social]

二、全局排序：

1）对于同一高考年份，同一考试类别，同一高考省份的同学，我们根据其高考提档分进行了排名，并除以了该类别的高考总人数，得到了归一化的比率。[admit_rank]

2）对于同一高考年份，同一院系的同学，我们根据其中心化标准化的高考提档分进行了排名，并除以了该类别的总人数，得到了归一化的比率。[depart_rank]

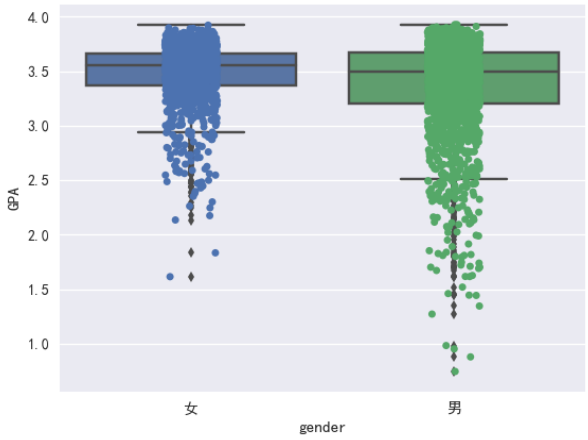
三、加权评估：

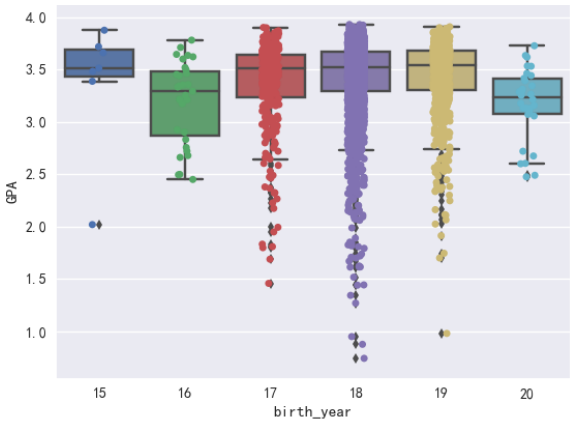
1）我们将两年内北大在一所学校中录取的人数，赋给这个学校作为一个权值。[school_num]

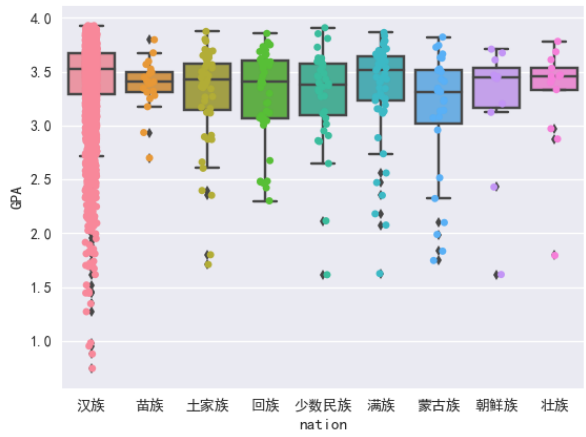
2）我们将两年内北大在一所学校中录取的所有人的投档成绩在省内的排名，求平均，并把这个平均排名赋给这所学校。[school_admit_rank]

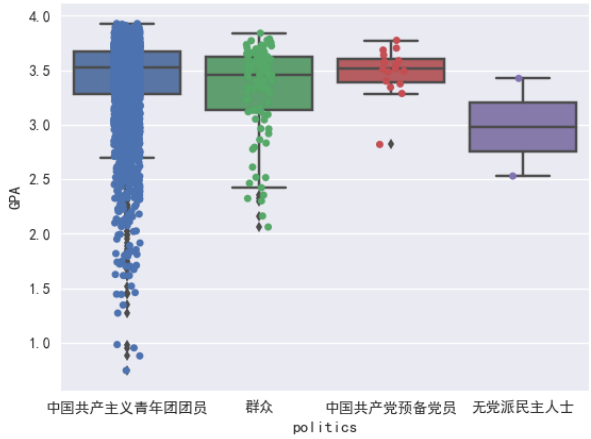
III. 特征选择

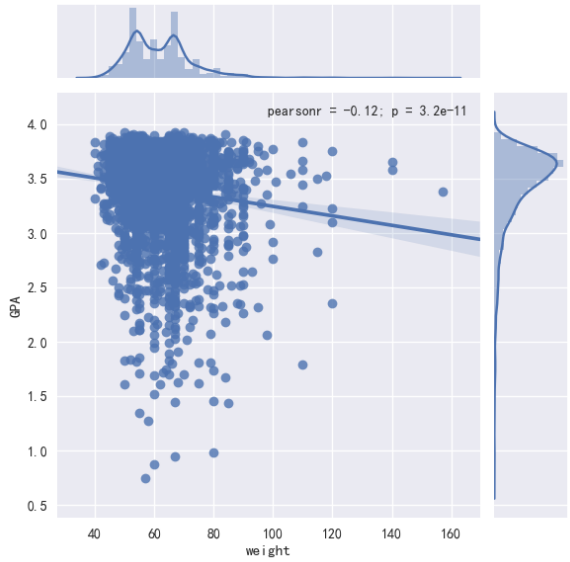
特征选择，一般是基于两种方式：创建模型之前，根据格图中散点的分布情况；根据特征选择之后，模型在validation集上的mse结果。格图如下所示：

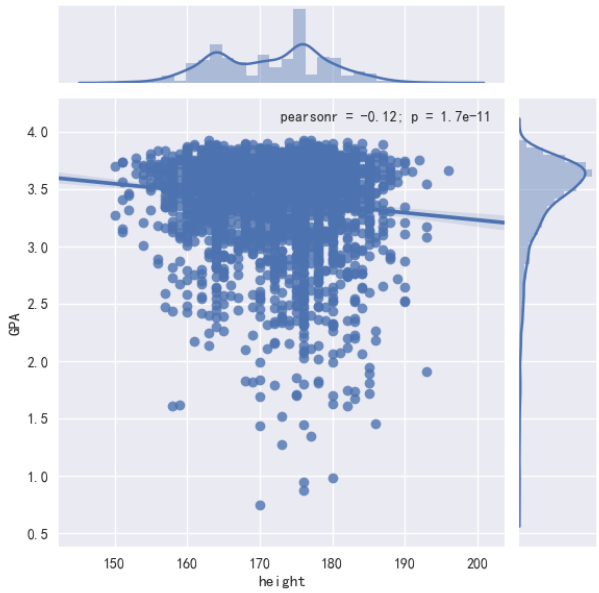


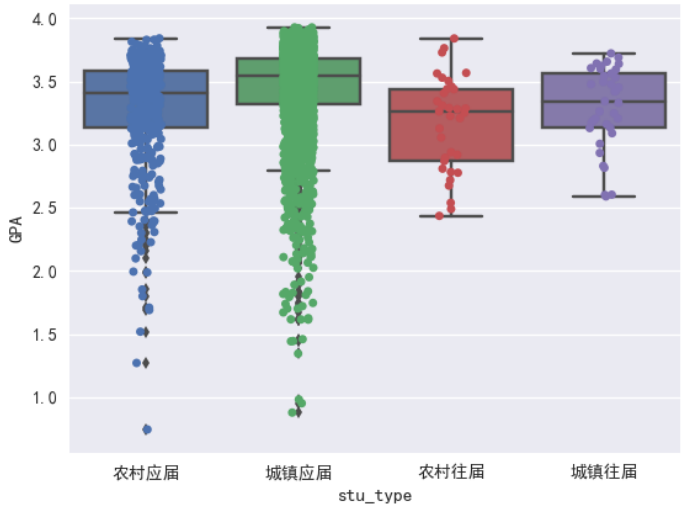


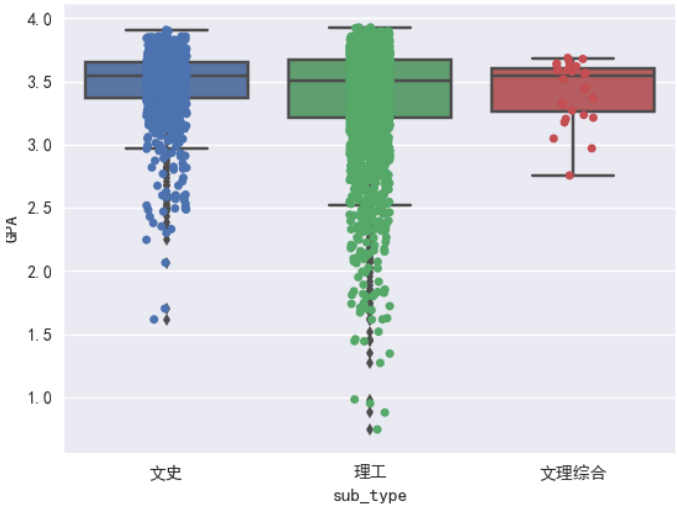


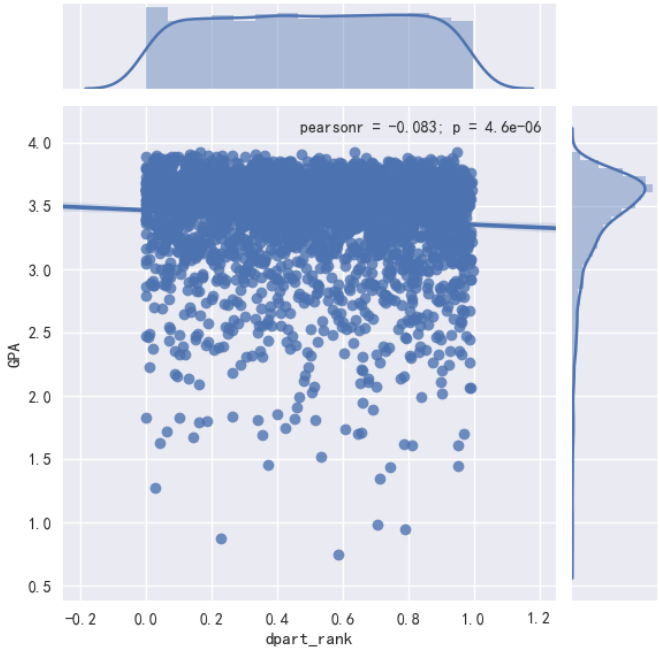


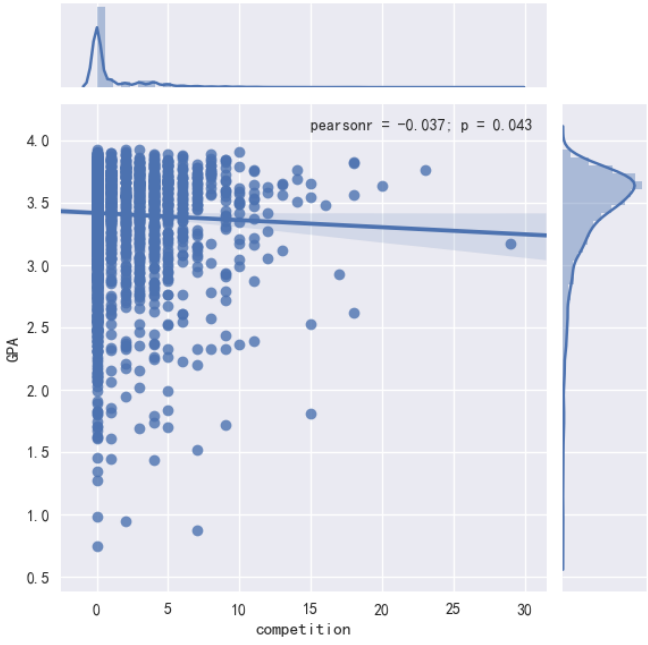

pearsonr = -0.12; p = 3.2e-11
GPA
weight

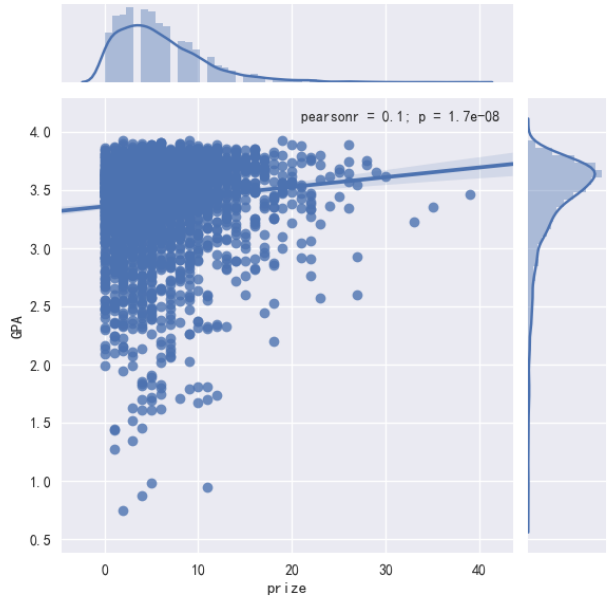

pearsonr = -0.12; p = 1.7e-11
GPA
height

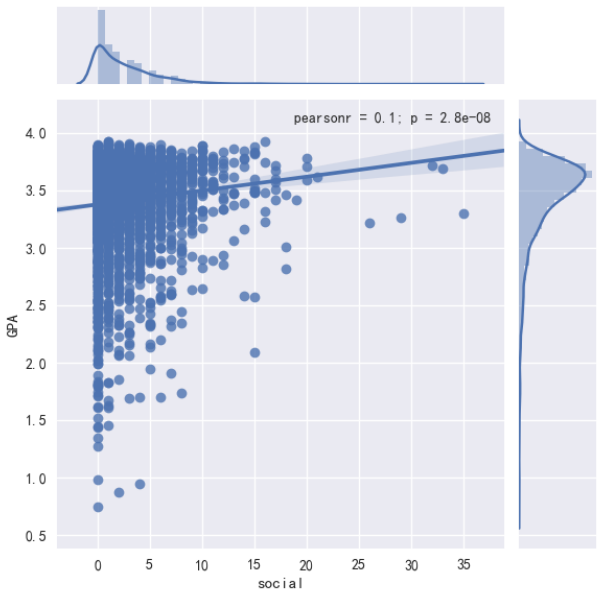

GPA
农村应届
城镇应届
农村往届
城镇往届
stu_type

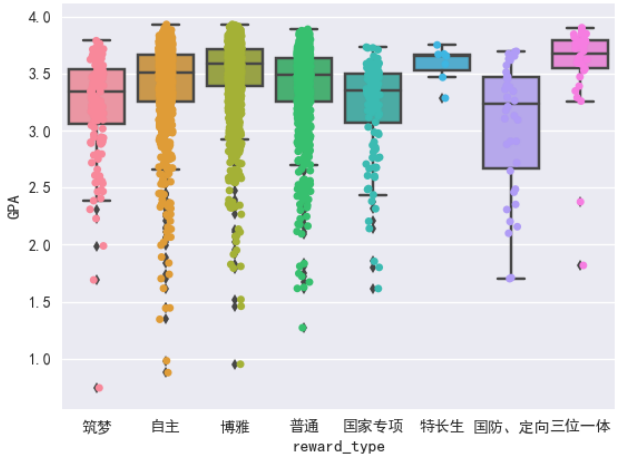

GPA
文史
理工
文理综合
sub_type


pearsonr = -0.083; p = 4.6e-06
GPA
dpart_rank

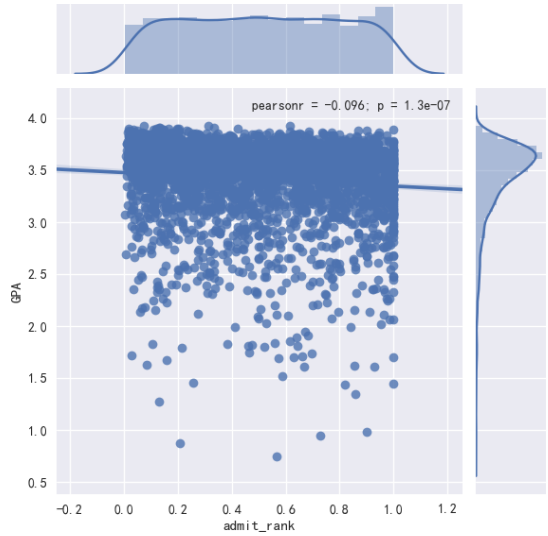

pearsonr = -0.037; p = 0.043
GPA
competition

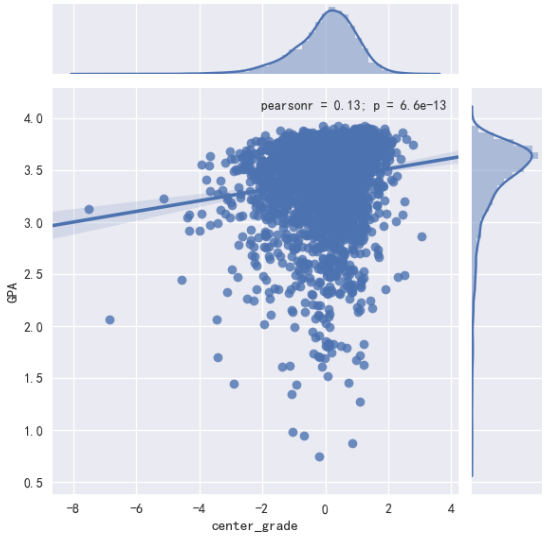

pearsonr = 0.1; p = 1.7e-08
GPA
prize

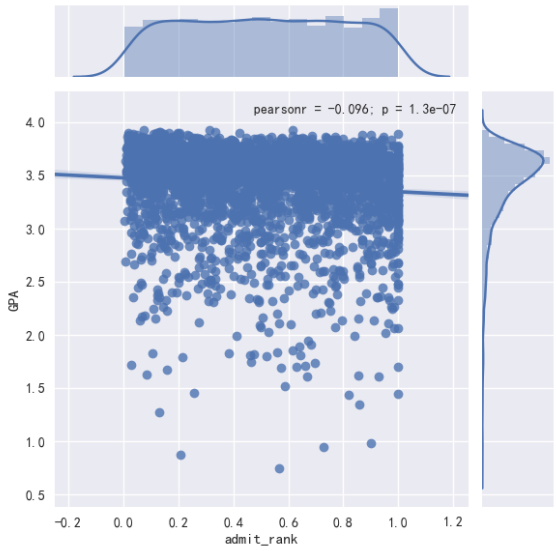

pearsonr = 0.1; p = 2.8e-08
GPA
social

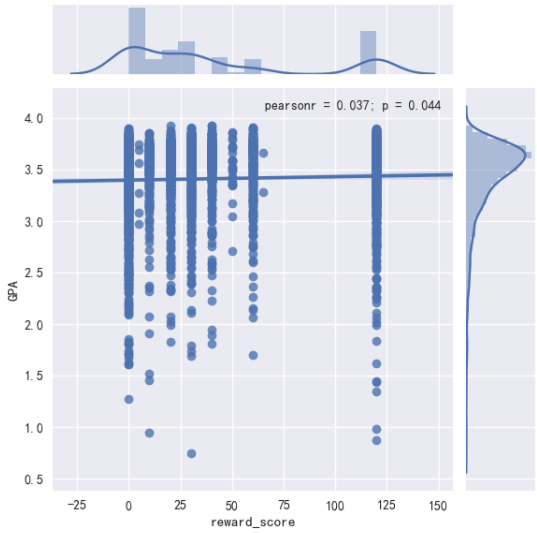

GPA
筑梦
自主
博雅
普通
国家专项
特长生
国防、定向三位一体
reward_type

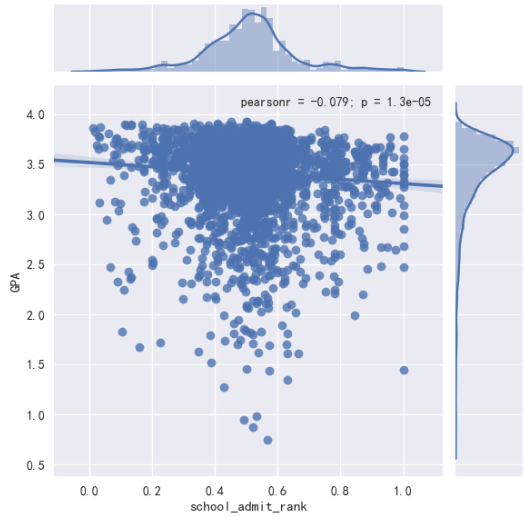

GPA
2015
2016
test_year

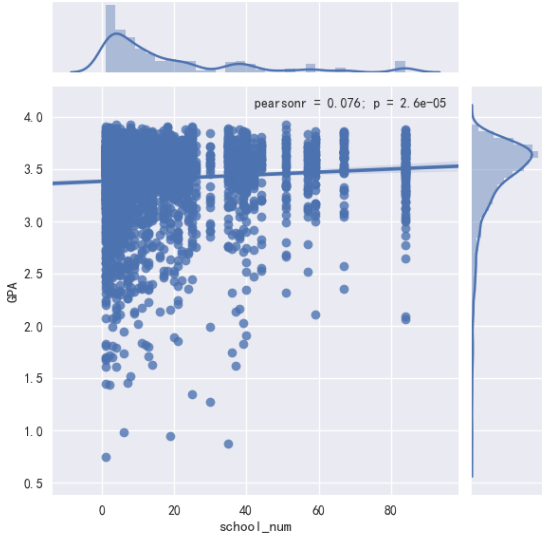

pearsonr = -0.096; p = 1.3e-07
GPA
admit_rank

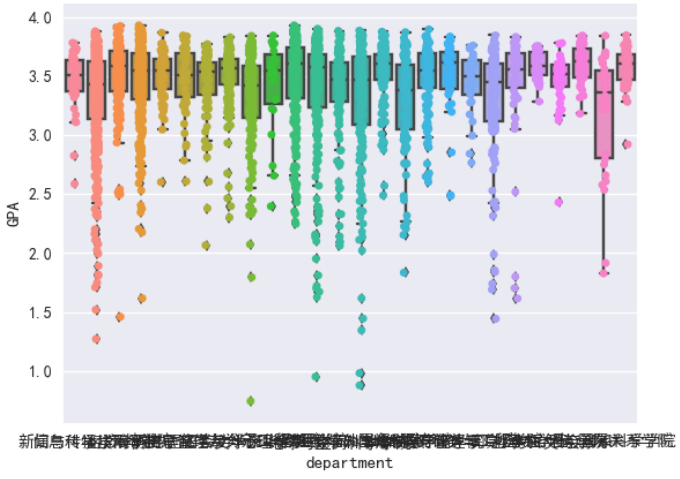

pearsonr = 0.13; p = 6.6e-13
GPA
center_grade

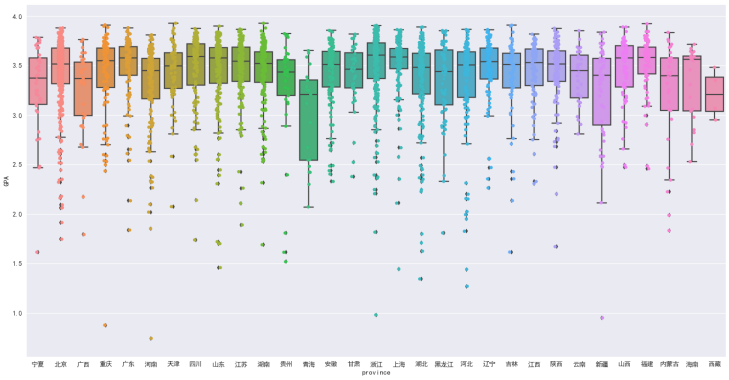












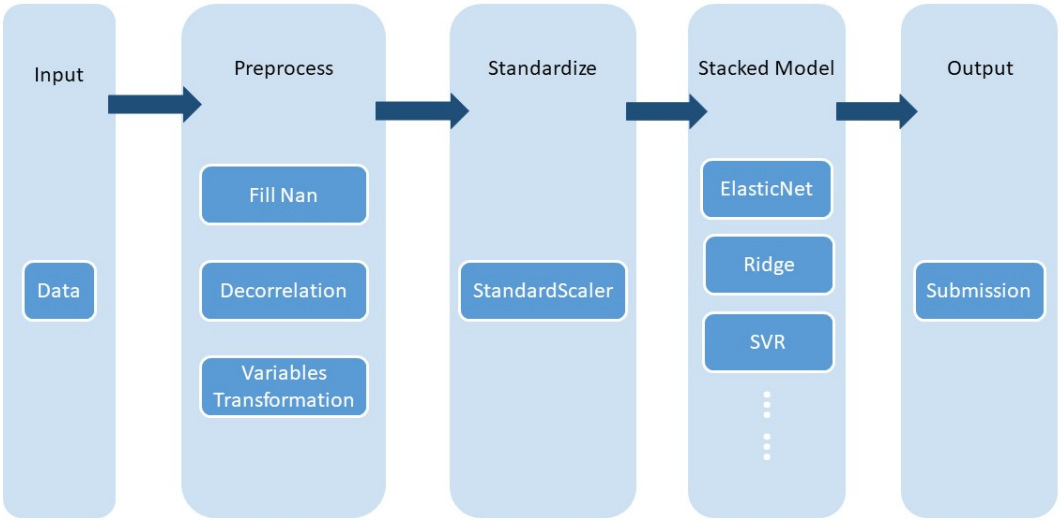
最后，我们选取的特征有：

数值特征——admit_rank, school_num, center_grade, social, center_social, school_admit_rank, dpart_rank
reward_score, center_reward, competition, center_competition, height, weight, high_school

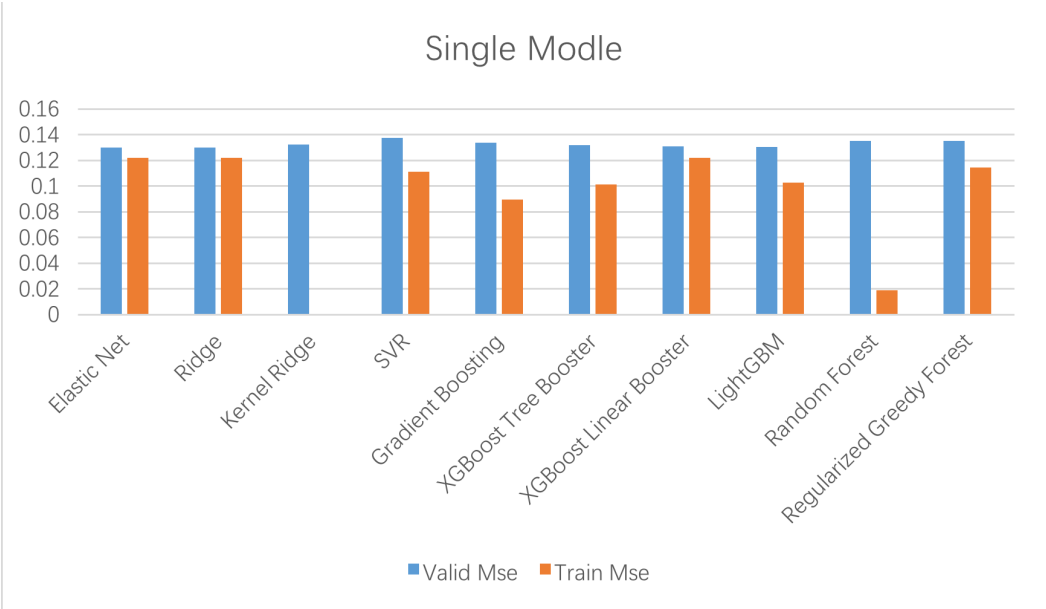
离散特征——province, politics, gender, birth_year, nation, test_year, stu_type, sub_type, department,
reward_type

Model Generation

I.总体流程



II.模型选择



一、线性模型

本项目的训练集数据量较小（3000*110），理论上使用线性模型会有很好的效果。我们选择了三种线性模型，Elastic Net（Lasso回归和Ridge回归的结合）、Ridge和Kernel

Ridge。经过网格搜索参数后的三种线性模型已经可以很好的描述数据，在单模型中表现最好。在进行模型综合之前，我们使用Elastic Net单模型取得了0.1324的最好成绩。

二、SVM

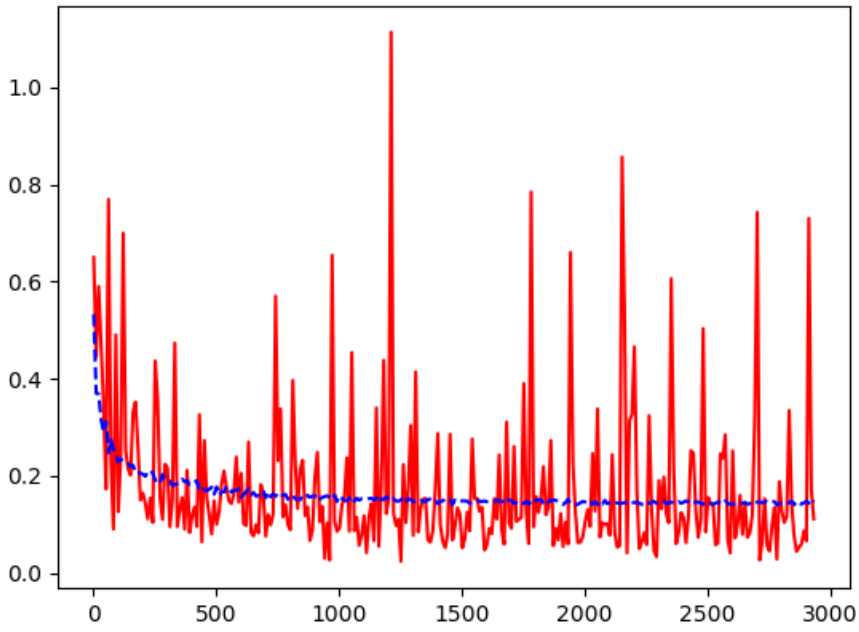
SVM模型在这个命题中出现了泛化性不足的问题。调整过C值和gamma值减小过拟合后依然只能在valid集上得到0.1337的mse。但在模型综合时我们仍然加入了SVM，它能对大部分数据有较好的描述。

三、Boosting模型

我们在线性模型和SVM外，还选择了6种boosting模型来增强对偏离中心条目的预测能力。最终挑选了集成于sklearn中的树形模型：Gradient Boosting Regressor和Random Forest Regressor、XGBoost中树形和线性booster、对每个树节点进行正则化的树形模型Regularized Greedy Forest、以及训练较快的LightGBM。参数调整后它们都能得到0.130左右的valid集mse。

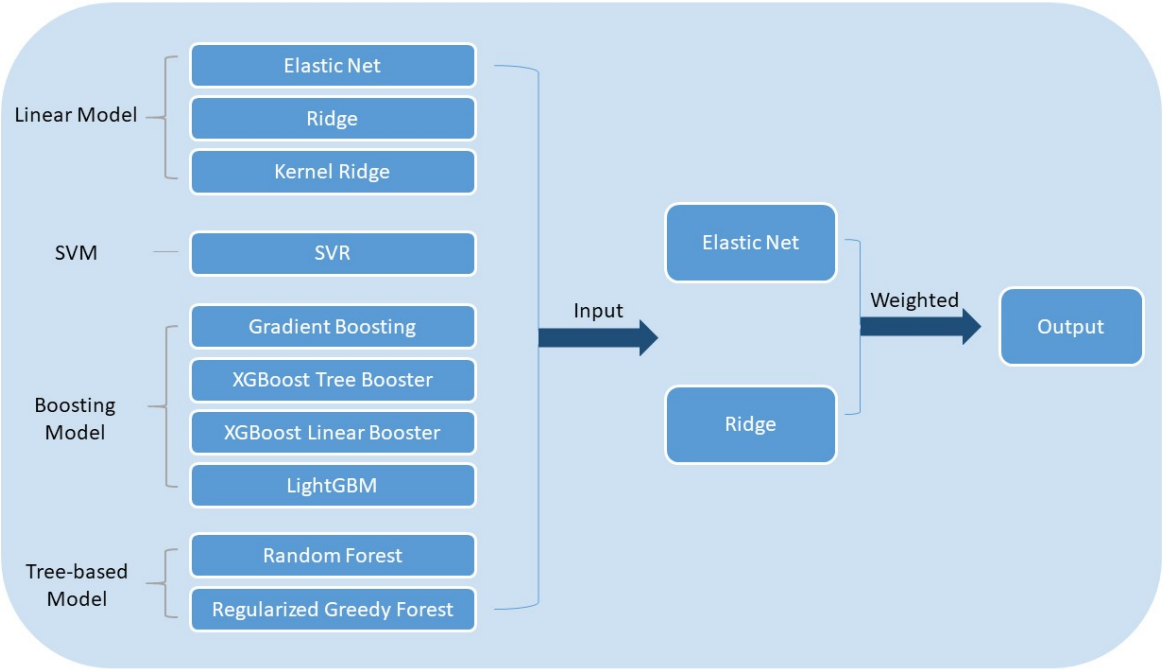
四、MLP和深度模型

除已经使用的10种模型外，我们还尝试使用了两层和三层的神经网络。由于数据规模太小，训练中会出现严重的过拟合情况；即使减小网络大小，并加入了dropout、early-stopping等缓解过拟合的方法，最终的表现仍然不尽人意。

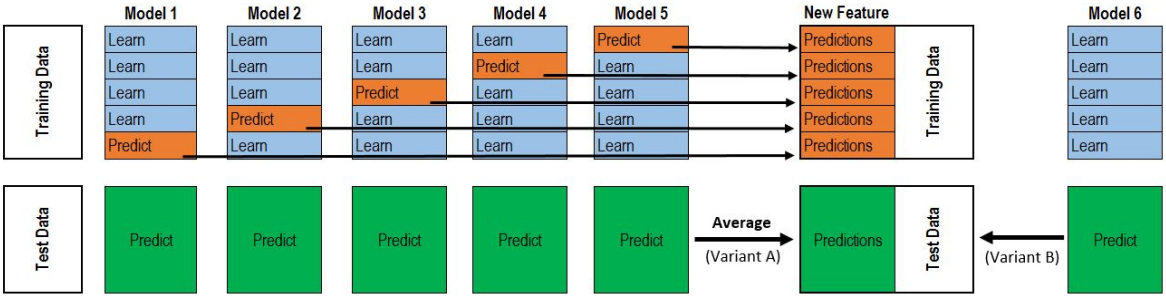


上图中红色曲线为*valid*集的*mse*，蓝色为训练集*mse*。可以看到训练进行下去泛化性不会有任何的提升。在最终提交的版本中，我们没有选择将深度方法的结果放入融合模型中。

III.模型融合



在常用的模型ensemble方法（Bagging、Boosting、Blending和Stacking）中，我们选择使用一个三层Stacking模型组合各个模型的结果。整个Stacking的过程类似于CV验证：将训练集分为5份，对每个基本模型进行5轮训练，使用依次使用其中的4分作为训练集训练，预测余下一份的结果，5轮后得到训练集大小的预测数据；同时在每轮中对测试集进行预测，对每个基本模型来说测试集的预测结果为5轮结果的均值；在第二层中，我们输入（训练集上的预测结果*基本模型数量）的数据进行训练；最终输出为第二层预测结果的加权组合。



使用Stacking进行ensemble后预测效果得到了明显的提升，从单模型到Stacking模型，每加入一个有效模型（与其他模型相关度不高），CV结果和提交结果都会有0.005的mse减小，这说明ensemble能够很好地平衡各个模型的特性，过拟合情况得到缓解。

IV.模型训练和验证

一、模型训练

首先使用GridSearchCV对各个单模型网格搜索，直到各个模型都能达到相近的水平（mse在0.13左右）。模型融合时，第二层选择较为简单的模型（线性模型），最终确定使用Ridge回归和KernelRidge回归，并且同样使用网格搜索，确定第二层模型的参数与权重。

二、CV验证

提交结果前，在整个训练集上进行5-fold分割并CV验证，这样得到的结果与提交结果比较接近。

APPENDIX

- model文件夹中包含实验代码：grid_search.py, NN_regression.py, NN_train.py, regression.py
- result文件夹中包含自11月25日至12月10日的294次实验结果
- data文件夹中包含我们除噪之后的数据集ALLDATA.xlsx以及提取的交叉特征