**基于机器学习的学生毕业预测 郑龙天 指导教师 吴清锋 厦门大学**

2029320381452802095400

**本 科 毕 业 论 文 （设 计）**

**（主修专业）**

基于机器学习的学生毕业预测

**Student graduation prediction based on machine learning**

姓 名：郑龙天

学 号：24320142202548

学　 院：软件学院

专 业：软件工程

年 级：2014级

校内指导教师： 吴清锋 (教授)

**二〇一八年五月 日**

**厦门大学本科学位论文诚信承诺书**

本人呈交的学位论文是在导师指导下独立完成的研究成果。本人在论文写作中参考其他个人或集体已经发表的研究成果，均在文中以适当方式明确标明，并符合相关法律规范及《厦门大学本科毕业论文（设计）规范》。

该学位论文为（ ）课题（组）的研究成果，获得（ ）课题（组）经费或实验室的资助，在（ ）实验室完成（请在以上括号内填写课题或课题组负责人或实验室名称，未有此项声明内容的，可以不作特别声明）。

本人承诺辅修专业毕业论文（设计）（如有）的内容与主修专业不存在相同与相近情况。

学生声明（签名）：

年 月 日

致 谢

值此论文完成之际，谨向所有关心和支持我的人们致以诚挚的谢意！

首先，我要衷心地感谢我的导师XXX教授。从论文选题、内容和整体结构的确定，直至最后定稿，XXX老师都以极其负责的态度给予悉心指导，为我提出了许多宝贵的意见和建议，使我获益良多。他渊博的学识、严谨的治学态度以及朴实的学术作风时刻激励我不断努力完善自己，对我的悉心关怀和教诲也将鼓舞我在今后的学习和工作上不断努力向上。在此，谨向XXX老师致以最诚挚的感谢！

其次，还要感谢与我一起完成这个项目的所有团队成员。没有他们的帮助和共同努力，就没有项目的圆满成功，也就不会有本文的形成。在此，向他们表示衷心的感谢!

摘　要

随着校园信息化程度的完善，学生在校园内的各种信息数据变得越来越容易获得。而且在近几年来的时间积累下，学生信息的数据量越来越大，大量的数据如果只是用于无机的统计等处理显然是一种浪费。而本论文就是主要讲述如何恰当利用这些大量的数据来对校园工作产生很好的正面影响的。

学生的毕业情况是学校要处理的重点问题，学生能否顺利毕业关系到诸多问题。本论文结合现今校园大数据应用的背景，对学生预测系统进行了深入的剖析，包括设计思路以及详细的功能实现。讲述了通过机器学习和数据挖掘技术，利用学生信息对学生个体、群体进行多维画像分析，从多角度精确了解学生的相关行为，将学生和群体最大化的“透明化”，然后以此来进行毕业预测。

以这些为基础，进行特征的筛选，建立系统的算法模型，接着进行评估迭代，直到模型正确率满足系统要求，以此来构建学生的毕业预测功能。最后再以高度可用的展示模块来对系统功能进行封装展示，满足学校对于该功能的使用要求。

目前学生毕业预测系统已经稳定运行在Window、Linux系统上，实现了预期的目标。希望未来，本系统可以实现高通用化，使之应用到各个高校中，帮助各个高校更好的进行学校工作。

关键词：校园工作 大数据 数据挖掘 机器学习 学生画像

**Abstract**

With the improvement of the degree of campus information, students’ information data on the campus becomes more and more accessible. Moreover, under the accumulation of time in recent years, the amount of student information data is getting larger and larger. If a large amount of data is only used for inorganic statistics and other processing, it is obviously a waste. This paper is mainly about how to properly use these large amounts of data to have a positive impact on campus work.

The graduation situation of students is a key issue that the school has to deal with, and whether students can successfully graduate is related to many issues. This paper combines the background of today's campus big data application and analyzes students' prediction system in depth, including design ideas and detailed function implementation. It describes the use of student information to perform multi-dimensional portrait analysis of student individuals and groups through computer learning and data mining techniques, and screens features. Based on these, a systematic algorithm model is established, followed by an iterative evaluation until the accuracy of the model meets the system. Requirements to build a student’s graduation prediction function. Finally, a highly available display module is used to package and display the system functions to meet the school's requirements for using this function.

At present, the graduation forecasting system of students has been running on Window and Linux systems stably, achieving the expected goal. It is hoped that in the future, the system can be highly generalized and applied to various universities and colleges to help schools better perform school work.

**Key words:** campus work; big data; data mining; machine learning; student portrait

目 录

[Abstract IV](#_Toc512697776)

[Contents VII](#_Toc512697777)

[第一章 绪论 1](#_Toc512697778)

[1.1引言 1](#_Toc512697779)

[1.2论文组织结构 1](#_Toc512697780)

[第二章 系统相关技术概述 2](#_Toc512697781)

[2.1 Python机器学习框架Sklearn 2](#_Toc512697782)

[2.1.1 Sklearn概述 2](#_Toc512697783)

[2.1.2 Sklearn分类器 2](#_Toc512697784)

[2.1.3 Sklearn数据预处理 2](#_Toc512697785)

[2.1.4 Sklearn模型评估 2](#_Toc512697786)

[2.1 轻量级数据库Mysql 2](#_Toc512697787)

[3.2 Python建站框架Django 3](#_Toc512697788)

[3.3 图表设计工具Echarts 3](#_Toc512697789)

[3.4本章小结 3](#_Toc512697790)

[第三章 系统设计思路 3](#_Toc512697791)

[3.1数据分析和处理 4](#_Toc512697795)

[3.1.1数据表分析 4](#_Toc512697796)

[3.1.2数据处理 7](#_Toc512697797)

[3.2 特征表示 9](#_Toc512697798)

[3.3特征选择 9](#_Toc512697799)

[3.4模型设计和训练 9](#_Toc512697800)

[3.4.1模型设计流程分析 9](#_Toc512697802)

[3.4.2算法调研 10](#_Toc512697803)

[3.4.3建模研究 13](#_Toc512697804)

[3.5模型评估 13](#_Toc512697805)

[3.6模型持久化 14](#_Toc512697806)

[第四章 数据处理实现 14](#_Toc512697810)

[4.1源数据处理 14](#_Toc512697811)

[4.1.1多表聚合关联分析 14](#_Toc512697812)

[4.1.2学年数据处理 16](#_Toc512697813)

[4.1.3取全学号 16](#_Toc512697814)

[4.2学生特征数据处理 17](#_Toc512697815)

[4.2.1缺失值填补 17](#_Toc512697816)

[4.2.2特征离散化 17](#_Toc512697817)

[第五章 特征处理实现 18](#_Toc512697818)

[5.1特征表示实现 18](#_Toc512697819)

[5.2特征提取实现 20](#_Toc512697820)

[第六章 算法模型实现 21](#_Toc512697821)

[6.1弱分类器的建立 21](#_Toc512697822)

[6.2模型融合 22](#_Toc512697824)

[4.5模型评估实现 22](#_Toc512697833)

[4.6模型持久化实现 23](#_Toc512697834)

[第七章 系统展示 23](#_Toc512697836)

[第八章 总结与展望 23](#_Toc512697837)

[6.1总结 23](#_Toc512697838)

[6.2 工作展望 24](#_Toc512697839)

[参考文献 25](#_Toc512697840)

[附 录 26](#_Toc512697841)

**Contents**

[Chapter 1 Preface](#_Toc230076798) 1

[1.1 Introduction 1](#_Toc230076799)

[1.2 The Structure of This Dissertation 2](#_Toc230076800)

[Chapter 2 System Related Technologies Outline 4](#_Toc230076801)

[2.1 Unstructured Information Management 4](#_Toc230076806)

[2.1.1 Introduction of Unstructured Information 4](#_Toc230076807)

[2.1.2 Information Extraction 5](#_Toc230076808)

[2.2 Information Visualization 7](#_Toc230076809)

[2.3 Other Related Technologies Introduce 8](#_Toc230076802)

[2.3.1 Introduction of ASP.NET 8](#_Toc230076803)

[2.3.2 Introduction of ASP.NET AJAX 9](#_Toc230076804)

[2.3.3 ASP.NET Ajax Control Toolkit 11](#_Toc230076805)

[2.3.4 Dundas Chart Toolkit 12](#_Toc230076811)

[2.4 Summary 14](#_Toc230076812)

[Chapter 3 Unstructured Information Management](#_Toc230076813) 16

[3.1 The Design Philosophy of Enterprise Indicators](#_Toc230076814) 16

[3.2 The Design Philosophy of Enterprise Indicators Extraction 1](#_Toc230076815)7

[3.2.1 The Statistics Document’s Structure Analysis](#_Toc230076816) 17

[3.2.2 The Statistics Information Extraction Algorithm](#_Toc230076817) 24

[3.2.3 The Value of Statistic Extraction Algorithm](#_Toc230076818) 33

[3.3 The Design Philosophy of Information Visualization](#_Toc230076819) 35

[3.3.1 Information Visualization Model](#_Toc230076820) 35

[3.3.2 Information Visualization Base on Dundas Chart Toolkit](#_Toc230076821) 36

[3.4 Summary 38](#_Toc230076822)

[Chapter 4 Implementation of Enterprise Infromation Management](#_Toc230076823) 39

[4.1 System Profiler and Function](#_Toc230076824) 39

[4.1.1 Development Background and Overall Objective](#_Toc230076825) 39

[4.1.2 Functional Requirements and Module Division](#_Toc230076826) 40

[4.2 System Architecture](#_Toc230076827) 42

[4.3 Indicators of Statistics Module Design](#_Toc230076828) 43

[4.3.1 Statistical Analysis Module Design](#_Toc230076829) 43

[4.3.2 Questionnaire Management Module Design](#_Toc230076830) 45

[4.4 Document Management Module Design](#_Toc230076831) 46

[4.5 System Maintenance Module Design 48](#_Toc230076832)

[4.5.1 User Management Sub-module Design 48](#_Toc230076833)

[4.5.2 Role Management Sub-module Design](#_Toc230076834) 49

[4.5.3 Document Attribute Management Sub-module Design](#_Toc230076835) 49

[4.5.4 Database Backup and Restore](#_Toc230076836) 49

[4.7 Summary](#_Toc230076837) 49

[Chapter 5 System Testing and the Running Results](#_Toc230076838) 50

[5.1 System Test](#_Toc230076839) 50

[5.2 Running Results](#_Toc230076840) 50

[5.2.1 Statistical Analysis Module Running Results](#_Toc230076841) 50

[5.2.2 Questionnaire Management Module Running Results](#_Toc230076842)  52

[5.2.3 Document Management Module Running Results](#_Toc230076843) 54

[5.2.4 User Management Module Running Results](#_Toc230076844) 55

[5.2.5 Role Management Module Running Results 56](#_Toc230076845)

[5.2.6 Document Attribute Management Module Running Results](#_Toc230076846)  57

[5.2.7 Database Backup and Restore Module Running Results](#_Toc230076847) 58

[5.2.8 Improve the System-experience](#_Toc230076848) 59

[5.3 Summary](#_Toc230076849) 60

[Chapter 6 Conclusions and Future works](#_Toc230076850) 61

[6.1 Conclusions](#_Toc230076851) 61

[6.2 Future works](#_Toc230076852) 62

[References](#_Toc230076853) 63

Acknowledgements 64

# 绪论

## 1.1引言

在高校中，学生的主要目的就是学习先进的知识，所以完成学业可以说是学生最主要的目标了。但有些学生由于对时间和事情的安排不够合理，比如学分没有修满、必修课挂科等，可能会产生不顺利毕业的情况。学生需要对自己的学业完成情况负责，老师们也需要帮助提醒学生。然而就目前情况来看，如果要完成延毕提醒，比较困难。

随着信息技术的发展，高校以数字化信息和网络为基础，建立集教学、科研、管理、技术服务、生活服务等应用为一体的教育环境。学校的数据库中，保存有学生静态和动态信息。静态信息是学生的相对稳定的信息，包括学生户籍、学号、所属学院等信息；动态信息是随着用户行为不断变化的信息，包括校园卡消费信息、门禁信息、图书馆出入信息等。通过这些信息，可以为学校师生提供更好的服务。

本系统主要目标是将繁杂的学生信息，通过数据挖掘和机器学习技术，从多个维度刻画学生和群体信息，从多角度精确了解学生的相关行为，将学生和群体最大化的“透明化”，然后以此来进行毕业预测。

## 1.2论文组织结构

本论文共分为五章，论文首先分析了目前高校存在的一些问题，然后提出了可以利用已有的大数据来搭建大数据智慧平台来帮助干活。

论文具体安排如下：

第一章 简单介绍了企业和政府在信息化过程中遇到的非结构化信息管理困难和“认知过载”问题。针对北京市科委的指标统计分析问题，提出了毕业设计的背景、目标和研究价值。

第二章 概述系统中所使用的各项技术及的发展现状。

第三章 详细介绍本系统各方面的设计。

第四章 具体介绍系统对数据处理的实现。

第五章 详细介绍系统中对特征的处理。

第六章 讨论算法建模的相关事情，介绍如何构建更好的算法以支撑系统的运行。

第七章 展示最终系统的样貌以及交互界面。

第八章 最后论文总结了毕业设计所做的工作，并且指明了下一步的改进计划。主要是在信息抽取算法的改进，以及在用户体验方面的改进计划。

# 系统相关技术概述

## 2.1 Python机器学习框架Sklearn

### 2.1.1 Sklearn概述

sklearn是基于numpy和scipy的一个机器学习算法库，设计的非常优雅，它让我们能够使用同样的接口来实现所有不同的算法调用，借助SKlearn库，我们可以简单的实现复杂的机器学习算法的调用、比较，sklearn封装了特征选择，模型筛选，模型融合等算法的实现，sklearn还为我们提供了丰富的API文档供我们使用，是一款非常适合初学者的python学习库。

### 2.1.2 Sklearn分类器

系统主要使用的预测器是Sklearn中的分类器，sklearn提供了数目众多的分类器模型供用户选择，包括Naive Bayes、Decision Trees、SVM、Adaboost等。使用起来简单方便，还易于自定义优化。

### 2.1.3 Sklearn数据预处理

为了方便用户在开发时增加代码结构性，减少冗余和无效的代码，sklearn对数据的预处理提供了许多数据预处理方法。

### 2.1.4 Sklearn模型评估

为了方便用户在开发时增加代码结构性，减少冗余和无效的代码，sklearn对数据的预处理提供了许多模型评估方法。

## 2.1 轻量级数据库Mysql

MySQL是一个开放源码的小型关联式数据库管理系统，由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点。与其他的大型数据库例如Oracle、DB2、SQL Server等相比，MySQL自有它的不足之处，如规模小、功能有限等，但是这丝毫也没有减少它受欢迎的程度。对于一般的个人使用者和中小型企业来说，MySQL提供的功能已经绰绰有余，而且由于MySQL是开放源码软件，因此可以大大降低总体拥有成本。

## 3.2 Python建站框架Django

Django 是基于MVC思想的web开源框架，我们使用该框架主要是为了开发网站和成果的可视化展示的需求。但是Django框架不同与MVC的是采用了MTV模式，与其他框架相比，Django能够简便、快速的开发数据库驱动的网站，因为他有许多的第三方插件的支持和丰富的API供用户使用，甚至还可以添加自身的工具包，使得Django有很强的扩展性。

## 3.3 图表设计工具Echarts

ECharts，一个使用 JavaScript 实现的开源可视化库，可以流畅的运行在 PC 和移动设备上，兼容当前绝大部分浏览器，底层依赖轻量级的矢量图形库 [ZRender](https://github.com/ecomfe/zrender)，提供直观，交互丰富，可高度个性化定制的数据可视化图表。

Echarts是一款便捷高效的可视化工具，对于本系统网站而言，最终的结果是分析数据的价值，当然离不开数据的可视化问题，为了让用户对我们的结果看得更加直观，选择了Echarts可视化，Echarts实现了饼状图，热力图，云图等，让用户能够多角度看到结果。

## 3.4本章小结

本章从各个方面讲述了系统中用到的相关技术，包括sklearn、mysql、Django、Echarts，为后续的研究奠定了基础。

# 系统设计思路

本系统设计思路如图：

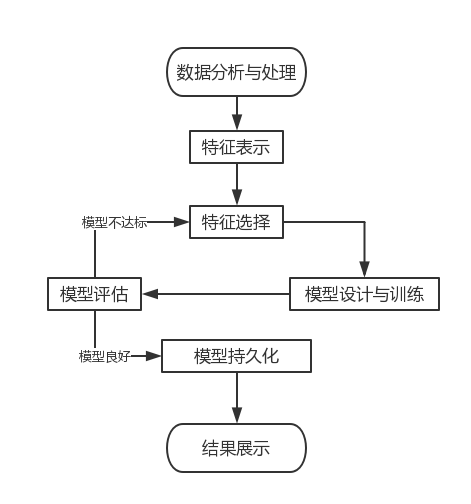


图3.1系统设计思路



## 3.1数据分析和处理

### 3.1.1数据表分析

本系统主要收集了学生校园卡消费记录、宿舍门禁记录、图书馆门禁记录、图书馆借阅记录、成绩信息，参加科创比赛以及获奖情况等方面的数据。

1. 图书借阅数据(library\_borrow)

|  |  |  |  |
| --- | --- | --- | --- |
| 学生id | 借阅日期 | 图书名称 | 图书编号 |
| 9708 | 2014/2/25 | "我的英语日记/ (韩)南银英著 (韩)卢炫廷插图" | "H315 502" |
| 6956 | 2013/10/27 | "解读联想思维: 联想教父柳传志", | "K825.38=76 547" |
| …… | …… | …… | …… |

该表的一条记录代表了一位学生的借书记录，我们通过主观分析，认为一个学生是否能够获得助学金，以及获得助学金的额度与该学生是否借阅某一些特定的书籍这种具体的数据并没有关联。反之，可能与该学生的总借阅量存在关系。于是，我们通过对该表进行统计分析，生成了一个特征值”library\_borrow”。该特征值代表每位学生的图书借阅总量，将作为训练集的特征之一。

1. 一卡通数据(card)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 学生id | 消费类别 | 消费地点 | 消费方式 | 消费时间 | 消费金额 | 剩余金额 |
| 1006 | POS消费 | 地点551 | 淋浴 | 2013/09/01 00:00:32 | 0.5 | 124.9 |
| 1406 | POS消费 | 地点78 | 其他 | 2013/09/01 00:00:40 | 0.6 | 373.82 |
| …… | …… | …… | …… | …… | …… | …… |

该表的一条记录代表了一位学生的一次消费记录，通过分析，我们发现消费类别一项的类型包括“charge”,”POS消费”，“交易冲正”，“卡冻结“等19项，我们分析认为对于每一位学生来说，学生卡充值平均金额，即该学生的消费类型为“charge”的消费记录的消费金额之和除以该学生的消费类型为“charge”的消费记录的条数与该学生是否为贫困生有密切关系。于是，我们通过对该表进行统计分析，生成了一个特征值” avg\_charge”。该特征值代表每位学生的每次充钱金额的平均值，将作为训练集的特征之一。此外，分析得出“cost\_amount”（学生历史总消费金额）、“cost\_variance”（学生消费方差）、“cost\_avg\_day\_superMarket”（学生每天超市消费平均值）、“cost\_avg\_day\_laundryroom”（学生每天洗衣房消费平均值）、“cost\_avg\_day\_dinnerHall”（学生每天食堂消费平均值）、“cost\_rate\_supermarket”（学生花费在超市的消费总额占总消费的比例）、“cost\_rate\_laundryroom”（学生花费在洗衣房的消费总额占总消费的比例）、“cost\_rate\_dinnerhall”（学生花费在食堂的消费总额占总消费的比例）、“cost\_times\_day\_supermarket”（学生每天超市消费平均值）、“cost\_times\_day\_dinnerhall”（学生每天食堂消费平均值）、“cost\_times\_day\_laundryroom”（学生每天洗衣房消费平均值）、“cost\_times”（学生消费总次数）、“balance\_rank”（学生卡内余额值在全体学生中的排名）、“card\_days”（学生card活跃天数）、“time6\_7costs”（学生每日6点-7点的消费总额）、“time7\_8costs”（学生每日7点-8点的消费总额）、“totaldinnercosts”（学生该学生日饭堂消费的总额）、“avgdayscosts”（学生的日平均消费）、“consumetimes11\_12”（学生每天 11点 - 12点消费的总次数）、“consumetimes0\_25”（学生单次消费金额在0-2.5元之间的次数）、“countcost0\_10”（学生当日总消费在0-10元范围内的总天数）、“cardrecharge”（学生卡充值总额）、“maxcost7\_8”（学生7点 -8点间的最大单笔消费的消费金额）、“below10\_rank”（学生日消费金额小于10天数占其card活跃天数的比例）、“below2\_5\_rank”（学生单次消费金额小于2.5次数占其总消费次数的比例）、“consume\_rank”（学生的消费排名）、“time7\_8consume\_avg”（学生7点 -8点间的平均消费）都与该学生是不是应该获得助学金有关联，故将他们都作为训练集的特征。

1. 图书馆门禁数据(library)

|  |  |  |
| --- | --- | --- |
| 学生id | 门禁编号 | 具体时间 |
| 3684 | 5 | 2013/09/01 08:42:50 |
| …… | …… | …… |

该表的一条记录代表了一位学生的进入或者离开图书馆的记录，图书馆的开放时间为早上7点到晚上22点，门禁编号数据在2014/02/23之前只有“编号”信息，之后引入了“进门、出门”信息，还有些异常信息为null。

“library\_time\_spand”：

根据这一特点，我们统计图书馆门禁数据后发现，有很多看似不合理的数据，例如某一学生在某日内有若干次进出图书馆的记录，且时间相隔较短；又如某学生在某天内有进出图书馆的记录，但最后一次是进门记录而不是离开记录。针对这些情况，我们有意忽视“门禁编号”的信息，对一个学生，将其当日记录时间最早的图书馆门禁记录所对应的记录时间作为该学生当日的进馆时间，将其当日记录时间最晚的图书馆门禁记录所对应的记录时间作为该学生当日的出馆时间，来求出该学生当日的图书馆学习时长。通过对所有记录进行统计，得出每一位学生的图书馆学习总时长“library\_time\_spand”，并将其作为训练集的特征之一。

“library\_times“ ：

由于在计算特征“library\_time\_spand”时忽略了每天进出图书馆的次数，为了减少这种方式可能产生的副作用，我们又分别统计每位学生进出图书馆的总次数“library\_times”,并将其作为训练集的特征之一。

1. 学生成绩数据(score)

|  |  |  |
| --- | --- | --- |
| 学生id | 学院编号 | 成绩排名 |
| 24320142202222 | x | x |
| …… | …… | …… |

成绩排名的计算方式是将所有成绩按学分加权求和，然后除以学分总和，再按照学生所在学院排序

”score”：

该表的一条记录代表了一位学生的成绩情况，还包含了他所属的学院，我们通过分析，认为一个学生是否能够获得助学金，以及获得助学金的额度与该学生的成绩存在联系。于是，我们将原数据中的“成绩排名”作为特征“score”并且作为训练集的特征之一。

“scorerank\_divided\_by\_stunum”:

由于每个学院的学生总数不同，所以如果单独使用“score”作为特征，可能会导致一些问题。所以我们又在原有数据上统计出各个学院的总人数，并用“scorerank\_divided\_by\_stunum”（学生成绩排名除以所属学院学生总人数）作为特征，来平衡“score”带来的负面影响。

1. 助学金数据(subsidy)

|  |  |
| --- | --- |
| 学生id | 助学金金额 |
| 10 | 0 |
| …… | …… |

该表的一条记录表示已知的某一位学生获得的助学金金额，其值可能去4个，分别是0,1000,1500,2000。这是我们最终需要对测试集进行预测后得到的结果。

“propotion\_of\_1000”、“propotion\_of\_1500”、“propotion\_of\_2000”：

这三个值分别代表某个学生所在学院获得1000助学金、1500助学金、2000助学金的人数占所有获得助学金学生人数的比例。因为每个学院的情况不一样，学校给每个学院安排的助学金名额也可能存在较大差异，故计算出这三个特征，并将其作为训练集的特征。

### 3.1.2数据处理

对于数据挖掘技术而言，为了让系统的可靠性，真实性增强，必须数据量充足，数值合理，标准，易于处理。但由于数据源的支持力度有限，所获取的学生真实数据有限，存在数据不平衡、数据缺失等问题。

在所给的数据集中，各种类型学生的人数比例相差比较大，此时用标准算法去解决一定会很困难。传统算法往往偏向于多数类，因为他们的损失函数在没考虑数据分布的情况下优化如错误率等量。最坏的情况是，小类别样本会被认为是大类别的异常值而被忽略，学习算法简单的生成一个平凡分类器，将每个样本都分类为大类别。

各类别人数比例不平衡，获得助学金人数远少于没获奖人数，如下图。

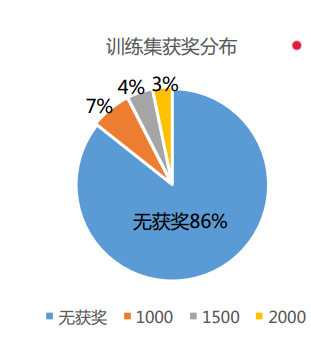


图3.1.2.1 获奖人数分布不均

数据处理就是要解决这些奇怪的问题。

1. 部分学生的部分属性缺失，系统对于该部分数据进行了缺失值的填补；
2. 由于被打上毕业不顺标签的学生数目过少，通过分层抽样、过采样处理类保证打上顺利毕业标签的学生和毕业不顺的学生训练样本均衡，使得前者与后者的比例小于4:1；
3. 因为训练数据过少，所以使用GAN网络模型生成更多合理的数据用于模型训练；
4. 在进行了上述的数据处理之后，系统采用了离散化特征的处理方式，以保证特征的高可用性。由于学生毕业预测是一个分类问题，需要学生的特征是离散的，不然原数据可能存在极端和异常的数据，可能导致模型训练出错。具体处理方式：
   1. 选取一个未转化的特征；
   2. 所有记录按照该特征上的值进行升序排序，得到一个有序的结果集合；
   3. 分别取出该集合的1/4处的记录，1/2处的记录，3/4处的记录，以及集合的最后一条记录；
   4. 用取出的记录在当前选择特征上的取值作为分界的依据，将所有记录的该特征转化为1,2,3,4这4者之一；
   5. 若还有未转化的特征则返回1；
   6. 结束；

## 3.2 特征表示

即学生画像，根据学生平时的行为，为学生打上相应的标签，来帮助我们更好地认识该学生，从而为该学生提供相应的服务。



图3.2.1 某学生的画像图

## 3.3特征选择

在进行了特征表示之后，我们要在这些特征中舍去一部分无关的属性，从而利用留下来的这些属性可以更准确地提供服务。利用被选择出来的特征，模型可以更好xxx。如图为毕业预测所筛选出来的主要特征：

图3.3.1挂科预警选取的主要特征

## 3.4模型设计和训练



### 3.4.1模型设计流程分析

在对数据进行预处理和特征处理之后后，尝试构建不同模型进行训练（主要为适合离散型属性的非线性模型），最后根据定制的模型评估方法对模型进行评估，剔除对结果影响很小的属性，需要考虑模型的精度、泛化能力等，目的是得出最理想的模型。

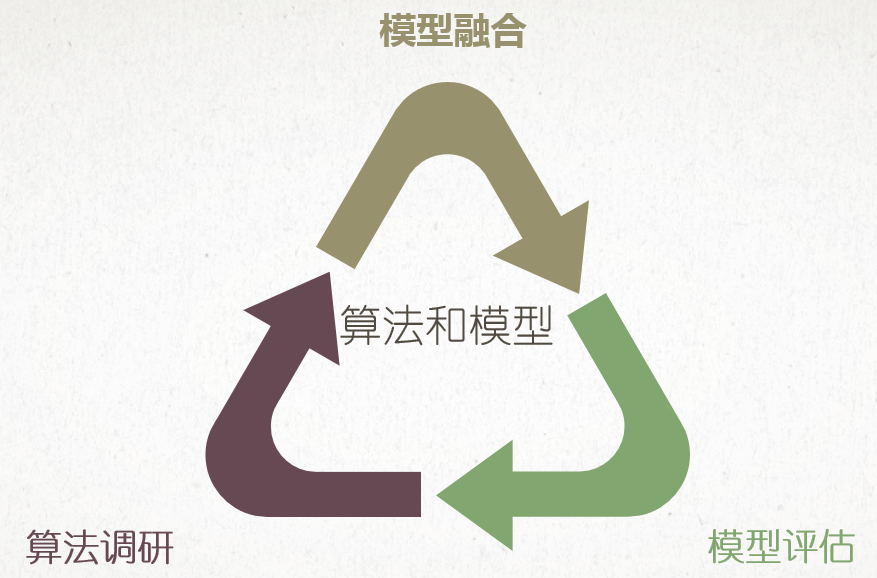


图3.4.1.1模型构建流程

每一次迭代都对选取的特征进行不断的增删，经过多次的迭代，使模型更加精准，直至达到预期要求。

### 3.4.2算法调研

由于系统主要解决为监督学习的分类问题（所谓监督学习说白了很简单，就是给定一堆样本，每个样本都有一组属性和一个类别，这些类别是事先确定的，那么通过学习得到一个分类器，这个分类器能够对新出现的对象给出正确的分类。这样的机器学习就被称之为监督学习）。所以这里调研了较为出名的分类，包括以下几种：

#### Naive Bayes朴素贝叶斯

朴素贝叶斯模型发源于古典数学理论，有着坚实的数学基础，以及稳定的分类效率。通过某对象的先验概率，利用贝叶斯公式计算出其后验概率，即该对象属于某一类的概率，选择具有最大后验概率的类作为该对象所属的类。

优点：

1. NBC模型所需估计的参数很少，对缺失数据不太敏感，算法也比较简单。

缺点：

1. 属性个数比较多或者属性之间相关性较大时，分类效率下降
2. 需要知道先验概率。
3. 分类决策存在错误率

#### Decision Trees（决策树）

决策树是附加概率结果的一个树状的决策图，是直观的运用统计概率分析的图法。机器学习中决策树是一个预测模型，它表示对象属性和对象值之间的一种映射，树中的每一个节点表示对象属性的判断条件，其分支表示符合节点条件的对象。树的叶子节点表示对象所属的预测结果。

优点：

1. 决策树易于理解和解释.人们在通过解释后都有能力去理解决策树所表达的意义。
2. 对于决策树，数据的准备往往是简单或者是不必要的.其他的技术往往要求先把数据一般化，比如去掉多余的或者空白的属性。
3. 能够同时处理数据型和常规型属性，其他技术往往要求数据属性单一。
4. 决策树是一个白盒模型。如果给定一个观察的模型，那么根据所产生的决策树很容易推出相应的逻辑表达式。
5. 易通过静态测试来对模型进行评测，表示有可能测量该模型的可信度。
6. 在相对短的时间内能够对大型数据源做出可行且效果良好的结果。
7. 可以对有许多属性的数据集构造决策树。
8. 决策树可很好地扩展到大型数据库中，同时它的大小独立于数据库的大小。

缺点：

1. 对于那些各类别样本数量不一致的数据，在决策树当中,信息增益的结果偏向于那些具有更多数值的特征。
2. 决策树处理缺失数据时的困难。
3. 过度拟合问题的出现。
4. 忽略数据集中属性之间的相关性。

决策树算法包括:

##### ID3

ID3 算法是由Ross Quinlan发明的，建立在“奥卡姆剃刀”的基础上，越简单的决策树越优于越大的决策树（Be Simple），ID3算法中，根据信息论的信息增益来进行评估和特征的选择，每次选择信息增益最大的特征作为判断模块。ID3算法可以用于划分标称型数据集，没有剪枝的过程，为了去除过度数据匹配的问题，可通过裁剪合并相邻的无法产生大量信息增益的叶子节点（例如设置信息增益阀值）。使用信息增益的话其实是有一个缺点，那就是它偏向于具有大量值的属性–就是说在训练集中，某个属性所取的不同值的个数越多，那么越有可能拿它来作为分裂属性，而这样做有时候是没有意义的，

##### C4.5

以信息增益率为衡量标准实现对数据归纳分类

优点：

产生的分类规则易于理解，准确率较高

缺点：

在构造树的过程中，需要对数据集进行多次的扫描和排序，因而导致算法的低效

##### CART

以基于最小距离的尼基指数估计函数为衡量标准对数据进行递归分类

优点：

抽取规则简便且易于理解；面对存在缺失值、变量数多等问题时非常稳健

缺点：

要求被选择的属性只能产生两个子节点；类别过多时，错误可能增加的较快

#### SVM （支持向量机）

建立一个最优决策超平面，使得该平面两侧距离平面最近的两类样本之间的距离最大化，从而对分类问题提供良好的泛化能力

优点：

1. 可以解决小样本情况下的机器学习问题。
2. 可以提高泛化性能。
3. 可以解决高维问题。
4. 可以解决非线性问题。
5. 可以避免神经网络结构选择和局部极小点问题。
6. 更好的泛化能力，解决非线性问题的同时避免维度灾难，可找到全局最优

缺点：

1. 对缺失数据敏感
2. 运算效率低，计算时占用资源过大

#### ADABOOST

针对同一个训练集训练不同的分类器(弱分类器)，然后把这些弱分类器集合起来，构成一个更强的最终分类器(强分类器)

优点：

1. 高精度，简单无需做特征筛选，不会过度拟合。

缺点：

1. 训练时间过长，执行效果依赖于弱分类器的选择。

### 3.4.3建模研究

研究各种不同算法的实现原理和它们的适用场景，经过多个模型的相互比较，根据评估指标对比模型在训练集上交叉验证的表现，选择对项目最有利的模型。

在具体研究过程中，针对不同的学生行为以及种类，我们可以通过构建模型来分析。可以从如下几方面进行考虑：

1）选择适当的模型：在得到好的属性之后，针对数据的特征，如离散值众多等特点，选用适当的模型进行预测。在比较各个模型的优劣之后，选择较为合适的模型进行细致的调参。这是有效辨别敏感用户的另一种方式。

2) 多个维度分析建模：可以从多个角度入手来分析建立模型，例如抽取学生日消费额进行研究，构建月消费分析模型，从中分析学生家庭情况是否存在异常行为，尤其是贫困生，可以针对月消费额变化情况提供是否接收为贫困援助对象等。

3) 不同模型的集成: 不同模型具有不同的偏好，当我们从多个角度用不同的模型对问题进行了预测时就可以得到多个具有不同偏好、对结果具有不同偏差的模型。有效的结合这些模型，让他们互相纠正。这不失为提高模型精度的一种好方式

### 3.5模型评估

   我们建立模型之后，接下来就要去评估模型，确定这个模型是否好用。在实际情况中，我们会用不同的度量去评估我们的模型。监督学习的主要任务就是用模型实现精准的预测。我们希望自己的机器学习模型在新数据（未被标注过的）上取得尽可能高的准确率，希望用训练数据训练得到的模型能适用于待测试的新数据。

由于系统要解决的问题为二分类，这是一个相对比较好评估的问题，所以采用较为常见的召回率来评估。

对于数据测试结果有下面4种情况：

TP: 预测为正，实际为正

TN: 预测为负，实际为负

FP:预测为正，实际为负

FN: 预测为负，实际为正

精确率、准确率：Accuracy=(TP+TN)/(TP+TN+FN+FP)

精准率、查准率： P = TP/ (TP+FP)

召回率、查全率： R = TP/ (TP+FN)

真正例率(同召回率、查全率)：TPR = TP/ (TP+FN)

假正例率：FPR =FP/ (FP+TN)

F1-score: 2\*TP/(2\*TP + FP + TN)

[准确率与召回率](https://baike.baidu.com/redirect/ad41hzblSUyJF9XtkWVy0f8Tq6fKgUFYStqBmqKB5nAjZv8NHEhdg2aKzJYI18oz3zPoSE5JylUwufeeXnrHn1mVmzf6MsBiM0I2UJsZcA-f-eeB3lc) ．CSDN．2013-06-08[引用日期2017-05-30]

## 3.6模型持久化

在训练完模型之后，最好有一种方法来将模型持久化以备将来使用，而无需重新训练,模型持久化对于我们的训练有很重要的作用。

* 如果我们的模型比较复杂，训练数据比较多，那么我们的模型训练就会耗时很长，如果在训练过程中出现某些不可预计的错误，导致我们的训练意外终止，那么我们将会前功尽弃。为了避免这个问题，我们就可以通过模型持久化来暂存我们训练过程中的临时数据。
* 如果我们训练的模型需要提供给用户做离线的预测，那么我们只得到预测值就可以了，这个时候只需用户提供一个输入，我们就可以通过模型得到一个输出给用户。



# 数据处理实现

## 4.1源数据处理

### 4.1.1多表聚合关联分析

本系统的数据表众多，一个学生的数据分散在各个不同的表里，一个表里有多条相同学生的不同数据记录。为了更加全面的描述学生，系统将在分散的表中统计计算出学生的各个特征值。

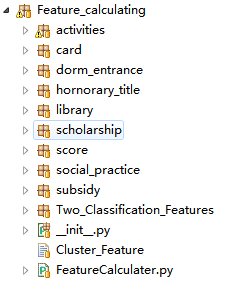


图4.1.1.1 系统中特征计算代码结构

@my\_exception\_handler

def calculate(self):

sql = "update students set activity\_avg\_level=%s"

self.executer.execute(sql, float(0))

sql="select Stu\_num,DATE\_FORMAT(Start\_time, '%Y-%m'),sum(Active\_level)/count(\*) from stu\_in\_activities group by stu\_num,DATE\_FORMAT(Start\_time, '%Y-%m')"

self.executer.execute(sql)

result = self.executer.fetchall()

for re in result:

if re is None:

pass

else:

re[1].split('-')

if int(re[1][5:7]) < 9:

sql="update students set activity\_avg\_level=(activity\_avg\_level+%s)/2 where student\_num=%s"

num = self.executer.execute(sql, (float(re[2]), str(re[0]) + (str)(int(re[1][0:4]) - 1)))

if num == 0:

self.add\_student(str(re[0]) + (str)(int(re[1][0:4]) - 1))

sql2 = "update students set activity\_avg\_level=%s where student\_num=%s"

self.executer.execute(sql2, (float(0),str(re[0]) + (str)(int(re[1][0:4]) - 1)))

self.executer.execute(sql, (float(re[2]), str(re[0]) + (str)(int(re[1][0:4]) - 1)))

else:

sql = "update students set activity\_avg\_level=(activity\_avg\_level+%s)/2 where student\_num=%s "

num = self.executer.execute(sql, (float(re[2]), str(re[0]) + str(re[1][0:4])))

if num == 0:

self.add\_student(str(re[0]) + str(re[1][0:4]))

sql2 = "update students set activity\_avg\_level=%s where student\_num=%s"

self.executer.execute(sql2, (float(0),str(re[0]) + (str)(int(re[1][0:4]))))

self.executer.execute(sql, (float(re[2]), str(re[0]) + str(re[1][0:4])))

然后以学生号为连接属性，将上面提到的各种特征进行连接，得到一张总表，在总表中，每个学生的信息只有一行，其主键为学生号，其余字段为上述所有特征的值，在行末再添加一个该学生是否可以毕业的值作为标签。

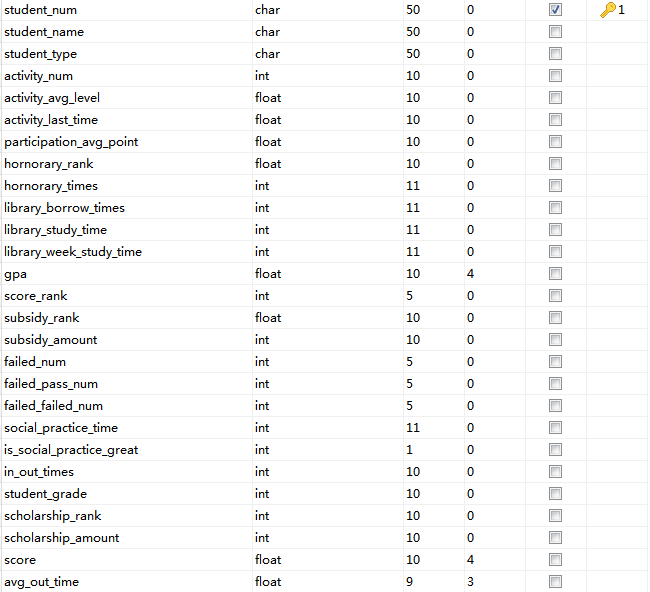


图 4.1.1.2多表聚合后的总表局部结构

### 4.1.2学年数据处理

由于每个学生每个学年都有数据，所以针对每个学生在每学年都生成一个样本，方便数据的训练。

### 4.1.3取全学号

由于数据源本身问题，存在部分学生在某表中有记录，而在其他表中无记录的情况，这就给最终表的生成造成的影响。如果直接使用学生号作为连接属性去生成最终表，那么会有很多学生因为在某一张表中没有记录而导致其在最终的表里没有记录。为解决这一问题，系统采取的方法是：

1. 判断总表中是否存在该学生的记录；
2. 如果有，直接更新特征值；
3. 如果没有，将当前学生学号插入到总表中；
4. 至此我们就可以得到一张涵盖所有学生信息的表。

具体代码如下：

sql = "update students set activity\_avg\_level=(activity\_avg\_level+%s)/2 where student\_num=%s "

num = self.executer.execute(sql, (float(re[2]), str(re[0]) + str(re[1][0:4])))

if num == 0:

self.add\_student(str(re[0]) + str(re[1][0:4]))

sql2 = "update students set activity\_avg\_level=%s where student\_num=%s"

self.executer.execute(sql2, (float(0),str(re[0]) + (str)(int(re[1][0:4]))))

self.executer.execute(sql, (float(re[2]), str(re[0]) + str(re[1][0:4])))

## 4.2学生特征数据处理

### 4.2.1缺失值填补

这里使用了sklearn.preprocessing中的Imputer来定义的MyImputer类来进行缺失值的填补。

代码如下：

class MyImputer():

def \_\_init\_\_(self):

from sklearn.preprocessing import Imputer

self.transformer=Imputer(

missing\_values='NaN',

strategy='mean',

axis=0,

verbose=0,

copy=True)

在系统获取数据时，就会调用到该类的transformer来进行缺失值填补。

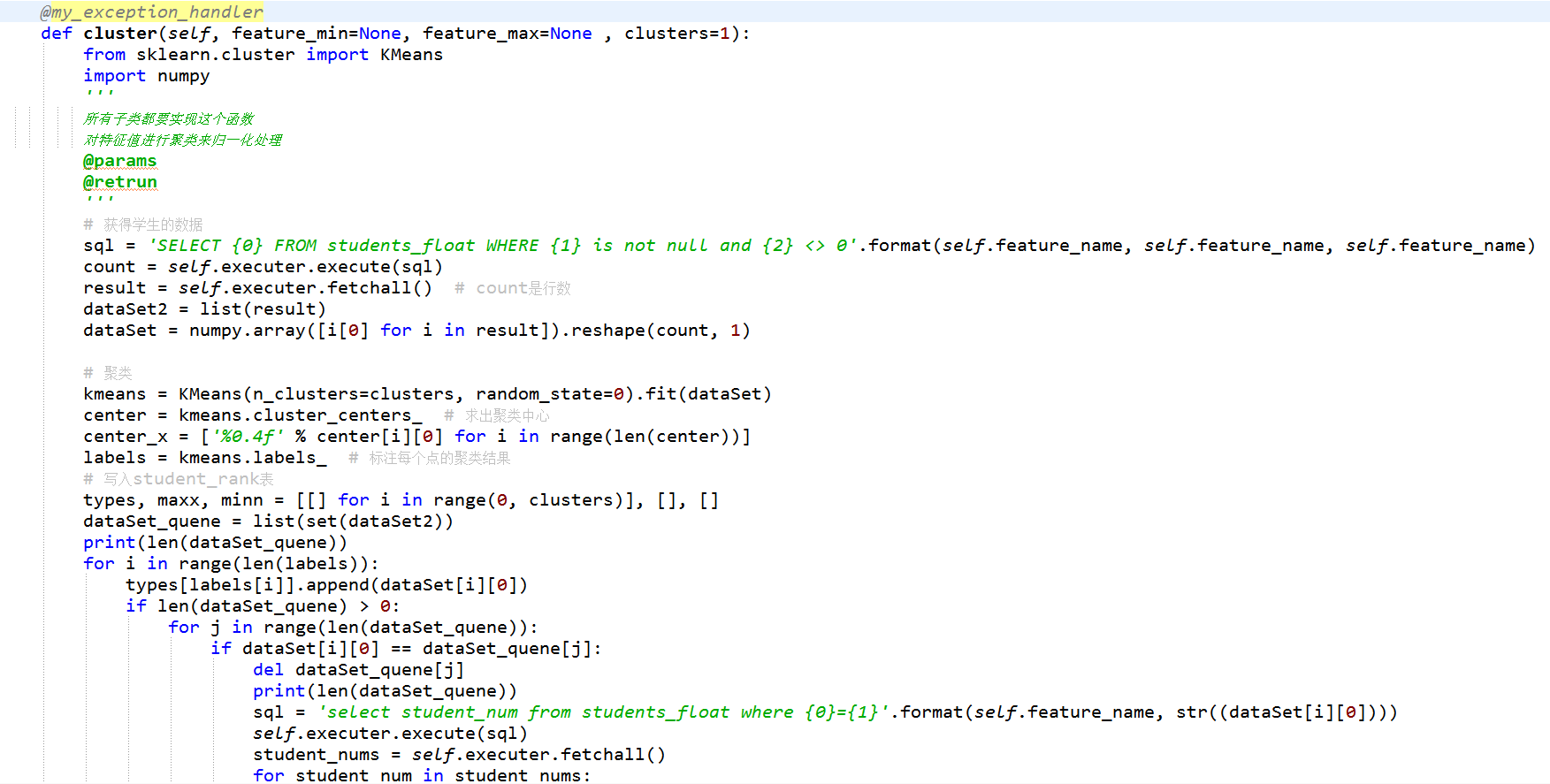
def pre\_process(self, X):

from Background\_program.b\_Sample\_processing.PreProcessing.MyImputer import MyImputer

MyImputer().transformer.fit\_transform(X)

### 4.2.2特征离散化

因为系统解决的是分类问题，所以要求进行特征的离散化处理，离散化的处理分散在每个特征计算类下面。



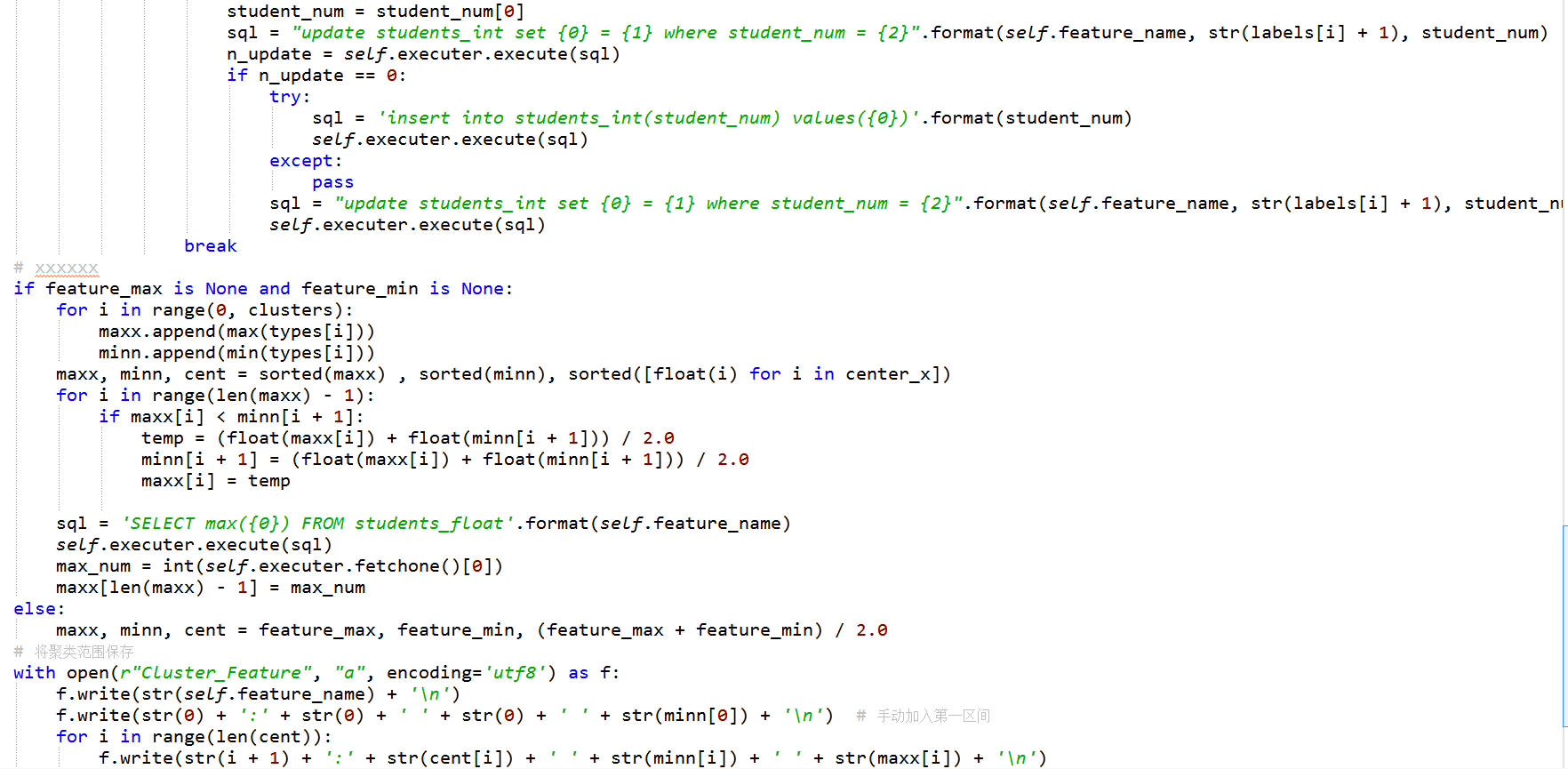


图 4.2.2.1特征离散化的实现

在进行完特征的计算之后，依次调用cluster聚类函数，进行该处理。

# 特征处理实现

## 5.1特征表示实现

在系统中为了更好地展示与处理学生，系统为学生建立了专门的学生类。

class Student():

'''

学生类，要画像的对象

'''

def \_\_init\_\_(self, student\_num, features=None, label="1"):

self.student\_num = student\_num

# self.school\_year = school\_year

try:

self.features = list(features)

except:

pass

self.label = label

def setStudent\_num(self, student\_num):

'''

设置学号

@params string student\_num:学生学号

@retrun

'''

self.student\_num = student\_num

def getStudent\_num(self):

'''

获取学号

@params

@retrun string self.student\_num:学生学号

'''

return self.student\_num

def getAll(self):

'''

返回特征+标签

@params

@retrun list[[]] features\_and\_labels:[特征,标签]

'''

features\_and\_labels = self.features + [self.label]

return features\_and\_labels

经过之前的一系列操作，最终得到了完整的数据集。该数据集共包含82个数据项，数据项 ’student\_num’为主键，其余81项作为特征项。特征项如下：

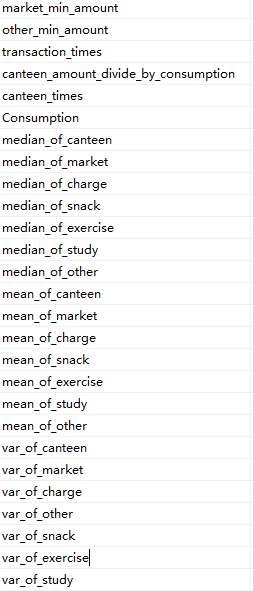
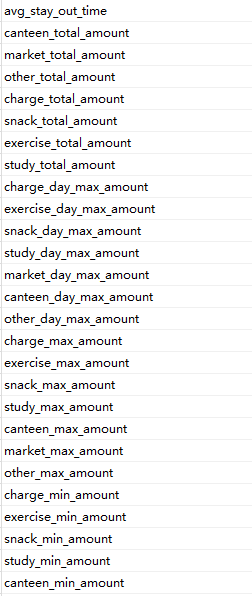
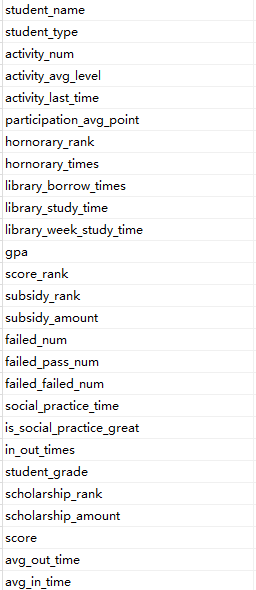


图5.1.1特征字段

## 5.2特征提取实现

最终数据表体现出特征项数目繁多的特点。此时数据集主要的问题主要体现两个方面：

* + - 1. 特性项过多很可能导致在后续的模型训练中出现对训练集过拟合的情况，从而使得训练所得的模型在测试集上的表现不尽人意；
      2. 上述81个特征项对不同的预测功能的适用性相差较大，如果直接将所有特征项投入训练很有可能会画蛇添足，导致所得的模型效果不好。因此，有必要进行特征选择，对特征项进行筛选，对特定的预测目标选出特定的特征项进行训练。

在sklearn提供的算法模型中包含了score函数，该函数的会返回每一个特征项的“贡献度”评分，通过比较不同特征项所获的评分，可以为每一个预测功能模块单独筛选出一个特征项集合用于该模块的模型训练。

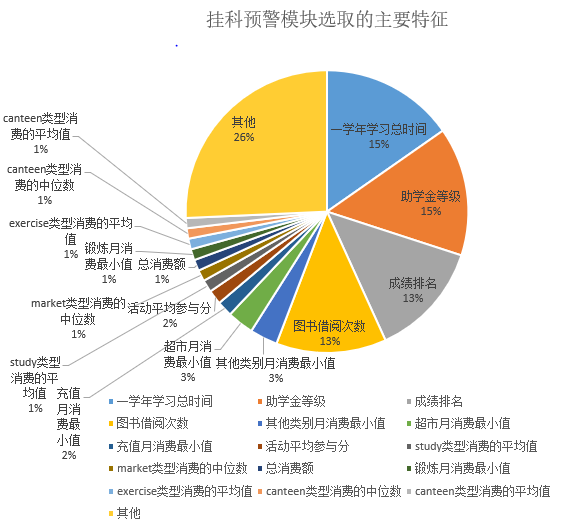


图5.2.1毕业预测选取的主要特征

# 算法模型实现

## 6.1弱分类器的建立



经过一些研究，最终发现除了Naive Bayes下的MultinomialNB和BernoulliNB表现不够好，其他模型都符合要求。这里面主要原因在于大部分属性是离散性的，且维度较高而且彼此间有相互联系，例如学生各类消费比等级、学生消费方差等属性，这对于构建非线性模型是很有帮助。

图 6.1.1各算法在同一数据集下的正确率

## 6.2模型融合



由于单一的分类器在某些方面存在局限性，比如：适于对有高纬度特征的数据进行分类，但是要求各特征间是独立的、和适于对各特征间联系较高的，但对高纬度特征的数据处理困难，这两种算法如果通过某种办法进行优势互补就能得到一个可以处理高纬度且各维度间互有联系的数据的算法模型。

使用是VotingClassifier的软投票机制（即不同分类器对分出的不同类所占的权重比值不一样，相比于硬投票，它是不同分类器占不同比值，对所分的类直接作用）。根据对不同分类器设置分类前的权重来使最后的模型偏向某一正确率最高的分类器(坐标数字为按上述的几个算法顺序赋予的权值组合)。

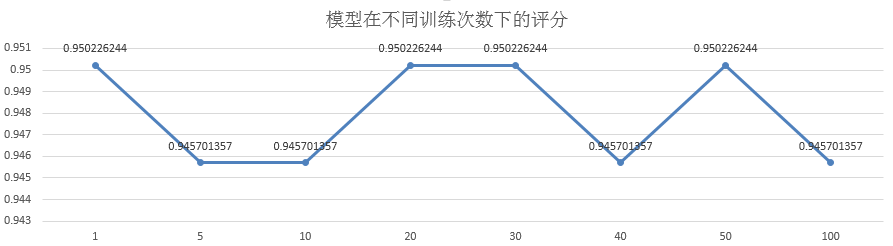


图6.2.1 模型在不同训练次数下的评分

所有最后选择训练次数为20。

## 6.3模型评估实现

系统使用sklearn .metrics.precision\_score来封装的，得到recall\_score类。

class recall\_score():

def \_\_init\_\_(self, y\_true, y\_pred):

from sklearn.metrics import recall\_score

self.evaluate\_score = recall\_score(y\_true, y\_pred)

def get\_evaluate\_score(self):

return self.evaluate\_score

def get\_model\_evaluater(self, y\_true, y\_predict):

'''

获得模型评估器，这里用正确率来评价

@params

@retrun

'''

from background\_program.d\_Model\_evalueting.Classification import recall\_score

model\_evalueter = recall\_score(y\_true, y\_predict)

return model\_evalueter

## 4.6模型持久化实现

def persistence\_model(self):

#将模型持久化

joblib.dump(self.estimater, 'train\_model')

def load\_model(self):

#从文件中加载模型

self.estimater = joblib('train\_model')



# 系统展示

# 总结与展望

## 6.1总结

此次项目中，我们团队基于python编程语言，借助Sciki-learn, Pandas,Numpy等开源数据分析模块，采用了多种数据挖掘方法，对学生数据进行深度分析，分别进行了数据预处理，特征工程筛选有意义的特征，预测模块挖掘建模，挖掘结果可视化，并从不同角度进行了统计分析，从而实现了多方位的数据分析挖掘。对所分析的数据，团队通过多次实验、比较分析，寻找并实现了有效的挖掘算法对数据进行建模挖掘。最终，我们是采用Django框架进行网站的快速开发，并结合echarts灵活的可视化方式，进行多维度的效果展示分析，让用户能够更加直观的看出结果，起初受限于数据量的问题，因为没有足够的数据，我们得出的模型和结果是没有那么强的说服力的，所以我们后期采用的是GAN网络通过学习已有数据源的分布特征，来填补缺失值和产生新的数据源，这些数据相对比随机生成的数据，更加真实、有价值，也为我们后期的模型验证起到很大的帮助。经过我们团队数月的努力，本项目目前完成了助学金预测，奖学金预测，失联预警，成绩预测等模块的预测，并为相关的模块生成了较好的模型融合器，对于新的数据，可以把它放入模型融合器快速的进行验证、分析。

后期，我们的系统将往特征选择更加高效智能化方向研究，对于任何的预测模块和任意的特征字段，能够快速筛选出有价值的特征进行训练，生成有效的模型融合器。

虽然只是作为一个大创项目的一小部分，但是这次小项目中我们得到很多有用的知识和手段，这为未来继续帮助大创的进行提供了很好的理论基础和经验。

## 6.2 工作展望

模型评分不够高。

当然，学生毕业预测系统中还存在着许多的不足，我将对它进行进一步的完善和改进：

1、学生毕业预测的算法还有待提高。我将对毕业预测的算法进行进一步改进，使它能够更加合理的解决其他高校中的问题，进一步提高算法的通用性。

2、系统的功能较为单一，在接下来的开发测试中，将加入更多实用的功能。

3、学生毕业预测系统的界面比较简单，缺乏交互效果，因此在后续的工作中，将对原有的界面设计进行进一步的改进。

4、系统的维护工作，以及文档的完善。

6、代码的优化。

# 参考文献

[1] 常青.大数据技术在高校智慧校园中的应用[J].信息与电脑,2016,(4):24-25.

[2] 刘敏斯,陈少波.大数据时代高校智慧校园建设研究[J].软件导刊,2015,14(8):6-8.

[3] 冯玖,李俊玲,张海霞,等.基于数据挖掘的校园一卡通数据应用研究——以石家庄为例[J].石家庄学院学报,2017,19(3):53-58.

[4] 王光宏,蒋平.数据挖掘综述[J].同济大学学报,2004,32(2):246-252.

[5] 刘卓,崔忠伟.大数据技术在高校智慧校园中的应用[J].软件导刊,2015,14(8):224-225.

[6] 李婷,傅钢善.国内外教育数据挖掘研究现状及趋势分析[J].现代教育技术,2010,20(10):21-25.

# 附 录

……………………