

# 中国科学技术大学

# 工程硕士学位论文



## 基于 TAP 平台的 高血压风险预测系统设计与实现

作者姓名：	李俊
学科专业：	软件工程
校内导师：	郑浩然 副教授
企业导师：	张丽娟 高级工程师
完成时间：	二〇一六年八月十一日

University of Science and Technology of China  
A dissertation for master's degree  
of engineering



**The Design and Implementation of  
Hypertension Risk System  
Based on TAP Platform**

Author's Name:	Jun Li
speciality:	software engineering
Supervisor:	Prof. Haoran Zheng
Advisor:	Lijuan Zhang senior engineer
Finished time:	August 11 <sup>th</sup> , 2016

## 中国科学技术大学学位论文原创性声明

本人声明所呈交的学位论文,是本人在导师指导下进行研究工作所取得的成果。除已特别加以标注和致谢的地方外,论文中不包含任何他人已经发表或撰写过的研究成果。与我一同工作的同志对本研究所做的贡献均已在论文中作了明确的说明。

作者签名: \_\_\_\_\_

签字日期: \_\_\_\_\_

## 中国科学技术大学学位论文授权使用声明

作为申请学位的条件之一,学位论文著作权拥有者授权中国科学技术大学拥有学位论文的部分使用权,即:学校有权按有关规定向国家有关部门或机构送交论文的复印件和电子版,允许论文被查阅和借阅,可以将学位论文编入有关数据库进行检索,可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。本人提交的电子文档的内容和纸质论文的内容相一致。

保密的学位论文在解密后也遵守此规定。

☐ 公开    ☐ 保密 (\_\_\_\_ 年)

作者签名: \_\_\_\_\_

签字日期: \_\_\_\_\_

## 摘要

随着医学技术的发展，在高血压风险预测方面，国内外很早就尝试使用新技术进行研究寻求突破，这一过程诞生了很多优秀成果，但由于受当时技术水平和一些硬件性能的限制，使得该领域的发展比较缓慢。

这两年云计算和大数据技术的蓬勃发展，加上处理器性能的不断提高，使得开发高准确率的高血压风险预测系统成为可能。在国外已经形成了不少类似的健康风险预估系统，而在国内这一方面还很欠缺。

本论文首先从国内外发展情况出发，结合当前市场需求和主流的技术，使用优秀的云计算平台 TAP 平台，研究并选择了合适的算法模型来完成高血压风险预测模块程序的设计与实现，结合具体的数据特征并添加和完善了数据处理量化模块程序，最终建立起了一套对高血压风险预测的系统。本论文使用工程化的方法和图表对系统的实现过程进行描述，并对相应的模块进行详细的测试，最终实现了这样一个运算速度快又预测准确率高的高血压风险预测系统。

本论文设计并实现的高血压风险预测系统是在前人的基础进行的，使用了最新的云计算平台和最新的硬件设备，保证了本系统的高运算速度，同时本系统对原始数据进行的规则化处理，使得系统对数据的兼容性很强，此外本系统使用了随机森林算法模型，可以很好的提高本系统预测的准确率。本系统的设计方法和实现过程也为其他疾病预测系统的开发提供参考性建议。

**关键词：**TAP 平台，高血压，大数据处理

## ABSTRACT

With the development of medical technology, people attempt to use new technologies to seek a breakthrough in hypertension risk prediction. This process that gave birth to a lot of good results, but due to the technology and hardware level at that time, make the development relatively slowly.

These years the rapid development of cloud computing and big data technology, and constantly improve the processor performance, make development high accuracy hypertension risk prediction system is possible. it have formed many similar health risk forecast system at abroad, but this is still very lack in China.

In this thesis, departure from domestic and foreign developments, combined with the current market demand and mainstream technology, using excellent platform cloud computing platform TAP, researched in Algorithm Model TAP platform-related, choose the appropriate model and complete the hypertension risk prediction module program, design and implementation along with specific data features added and improved data processing quantization module program, establish a hypertension risk prediction systems. In this paper, the use of engineering methods and diagrams of the system implementation process will be described, and the corresponding module for detailed testing, finally realized that is fast operation and a prediction of hypertension risk prediction system with high accuracy.

This paper design and implement of hypertension risk prediction system was conducted on the basis of predecessors, it use the latest cloud computing platform and the latest hardware equipment to ensure the high speed of this system, At the same time, we design the corresponding original data processing rules for this system that makes the system compatibility of the data is very strong, In addition this system uses the random forest algorithm model, it can well improve prediction accuracy. The design method and implementation process of this system for the development of other diseases forecasting system provide referential suggestions.

**Keywords:** TAP Platform , Hypertension, Big data handing

## 目 录

摘 要.....	I
ABSTRACT .....	II
目 录.....	III
第 1 章 绪 论 .....	1
1.1 选题的依据与意义 .....	1
1.2 高血压风险预测系统国内外发展现状 .....	2
1.2.1 国内外发展现状.....	2
1.2.2 存在的问题.....	4
1.3 本文主要工作内容 .....	4
1.4 论文的组织 .....	5
第 2 章 TAP 平台优势分析 .....	7
2.1 引言 .....	7
2.2 TAP 平台架构优势 .....	7
2.3 平台核心技术优势分析 .....	10
2.3.1 OpenStack 技术分析.....	10
2.3.2 CDH 技术分析 .....	11
2.3.3 Cloud Foundry 技术分析 .....	11
2.4 本章小结 .....	12
第 3 章 需求分析与系统概要设计.....	13
3.1 引言 .....	13
3.2 总体需求分析 .....	13
3.3 功能性需求分析 .....	14

3.4	非功能性需求分析 .....	16
3.5	系统概要设计 .....	17
3.5.1	目标与概述 .....	17
3.5.2	系统总体设计 .....	18
3.5.3	量化程序开发架构 .....	19
3.5.4	高血压风险预测系统总体模块设计 .....	19
3.5.5	方法和工具 .....	21
3.6	本章小结 .....	22
第 4 章	数据处理量化模块的设计 .....	23
4.1	引言 .....	23
4.2	原始数据特征分析 .....	23
4.3	数据处理规则设计 .....	27
4.3.1	汉字的量化 .....	28
4.3.2	连续值的分类量化 .....	29
4.3.3	关联其他字段的字段分类与量化 .....	30
4.4	数据处理模块总体流程设计 .....	31
4.5	本章小结 .....	33
第 5 章	高血压预测模块的设计 .....	34
5.1	引言 .....	34
5.2	TAP 平台中数据模型优缺点对比 .....	34
5.3	模型训练数据的特征分析 .....	35
5.4	各算法模型对本系统适应性分析 .....	36
5.4.1	分类树模型适应性分析 .....	36
5.4.2	主成分分析方法模型适应性分析 .....	36
5.4.3	支持向量机模型适应性分析 .....	37
5.4.4	K-Means 算法模型适应性分析 .....	37

5.4.5	随机森林算法模型适应性分析.....	38
5.5	高血压预测模块总体流程设计 .....	38
5.6	补充--随机森林算法原理简述.....	40
5.7	本章小结 .....	41
第 6 章	数据处理量化模块程序的实现.....	42
6.1	引言 .....	42
6.2	数据处理量化模块程序的实现 .....	44
6.2.1	数据处理量化模型程序相关类的设计与实现.....	45
6.2.2	数据处理量化模块程序运行过程实现 .....	50
6.3	本章小结 .....	52
第 7 章	高血压预测模块的实现.....	53
7.1	引言 .....	53
7.2	TAP 平台的搭建与运行 .....	53
7.2.1	硬件环境准备 .....	53
7.2.2	软件环境准备 .....	54
7.2.3	TAP 平台的部署和运行 .....	54
7.3	高血压风险预测应用程序的实现 .....	56
7.3.1	高血压预测模块交互过程.....	57
7.3.2	连接 TAP 平台 .....	58
7.3.3	增加数据源.....	59
7.3.4	创建并操作 Frames.....	60
7.3.5	模型训练及预测.....	61
7.4	本章小结 .....	63
第 8 章	系统核心模块的测试与分析.....	64
8.1	引言 .....	64
8.2	测试的方案 .....	64



## 目 录

---

8.3	测试需求描述 .....	65
8.4	测试执行情况 .....	66
8.4.1	功能测试 .....	66
8.4.2	兼容性测试 .....	71
8.4.3	性能测试 .....	72
8.4.4	结果分析 .....	74
8.5	本章小结 .....	75
第9章	总结与展望 .....	76
9.1	总 结 .....	76
9.2	展 望 .....	77
参考文献	.....	78
致 谢	.....	81

---

## 表目录

表 4.1	部分字段缺失率 .....	26
表 5.1	国内外知名算法优缺点对比 .....	34
表 6.1	Form 类部分方法 .....	42
表 6.2	Form 类部分事件 .....	43
表 7.1	高血压风险预测系统硬件开发环境 .....	53
表 7.2	高血压风险预测系统软件开发环境 .....	54
表 8.1	测试需求描述 .....	65
表 8.2	Feature 选择测试用例 .....	66
表 8.3	数据处理规则设置测试用例 .....	67
表 8.4	数据处理测试用例 .....	68
表 8.5	连接 TAP 并导入数据源测试用例 .....	69
表 8.6	创建 Frame 测试用例 .....	70
表 8.7	模型训练测试用例 .....	70
表 8.8	数据预测测试用例 .....	71
表 8.9	兼容性测试说明 .....	71
表 8.10	模块功能消耗平均时间 .....	73
表 9.1	高血压风险预测系统的优缺点 .....	76

## 图目录

图 2.1	TAP 体系架构 .....	8
图 3.1	数据管理员需求用例 .....	15
图 3.2	预测观察员需求用例 .....	16
图 3.3	高血压风险预测系统的总体架构 .....	18
图 3.4	量化程序客户端的体系结构 .....	19
图 3.5	系统的主要工作流程 .....	20
图 4.1	部分数据 1 .....	24
图 4.2	部分数据 2 .....	24
图 4.3	部分字段代码对应说明 .....	25

---

图 4.4	部分字段值分类 .....	26
图 4.5	Sex 字段分析 .....	28
图 4.6	Sex 字段规则设计 .....	28
图 4.7	CI 字段分析 .....	29
图 4.8	CI 字段规则设计 .....	29
图 4.9	Cr 字段分析 .....	30
图 4.10	Cr 字段规则设计 .....	30
图 4.11	数据处理量化模块工作流程 .....	32
图 5.1	高血压预测模块工作流程 .....	39
图 6.1	数据处理量化模块主页面 .....	44
图 6.2	数据处理量化模块项目结构 .....	45
图 6.3	数据处理模块类 .....	46
图 6.4	数据处理窗口 .....	47
图 6.5	feature 选择窗口 .....	48
图 6.6	数据处理规则编写窗口 .....	49
图 6.7	数据处理模块时序 .....	51
图 7.1	数据上传界面 .....	56
图 7.2	运行实例创建界面 .....	57
图 7.3	高血压预测模块交互过程 .....	58
图 7.4	创建证书文件 .....	59
图 7.5	随机森林算法预测结果 .....	63
图 8.1	数据导入时间折线 .....	72
图 8.2	数据处理时间折线 .....	73
图 8.3	预测准确率折线 .....	74

## 第1章 绪论

在健康医疗方面随着人们生活水平的不断提高，对其的关注日益加强，而作为慢性病中最普遍的高血压疾病，对该病的检测和预防研究一直从未停止。在一些传统的预测方法中，结果往往可信度不高。本章对传统预测方法在高血压风险预测的应用做了阐述，并分析了国内外这方面发展的现状，然后介绍了传统预测方法的不足和在使用过程中遇到的问题。接着介绍本论文的主要研究内容，最后对论文后续章节的安排进行了简单的介绍。

### 1.1 选题的依据与意义

人们日益增长的需求是推动互联网技术一次次进步的动力，互联网技术的不断发展，与人们的生活联系日益紧密，同时新的互联网技术又催生人们新的需求。云计算技术的出现源于人们对互联网技术相关服务的使用、增加和交付模式的改变<sup>[1]</sup>，云计算技术在出现之后的短短时间里飞速发展，按照调研机构 IDC 的考察报告陈述，一直到 2015 年年底，在私有云的 IT 基础设施的消费购买方面同比增长了 19.1%，总共支出达 124 亿美元；而在公有云的 IT 基础设施的消费购买方面则同比增长 28.2%，总共支出达 204 亿美元<sup>[2]</sup>。云计算技术在物联网、智能交通、智慧城市、云计算、手机支付、视频监控、医疗信息化<sup>[3]</sup>等领域应用的越来越广。

在医疗领域一直是人们关注的焦点，云计算技术在这一领域的应用自然是重中之重。而高血压疾病又是医疗领域最常见也是危害性最大的一种慢性疾病，引起心、脑、血管、肾脏疾病的最主要危险因素就是高血压疾病，该疾病会导致脑卒、心力衰竭、慢性肾脏病等多种并发症，而且致残、致死的概率比较高，同时还严重消耗了医疗和社会的资源，给家庭和国家带来严重的负担。根据《中国高血压防治指南 2010》一书描述，截至 2010 年底，在我国已经有 3553.8 万例的高血压患者在各地的医疗机构管理之下，但是 2002 年的全国高血压调查显示高血压患者的知晓率仅有 30.2%，控制率仅有 6.1%<sup>[4]</sup>。鉴于此高血压一直是重点研究领域，而人们在高血压风险预测方面也研究出一系列优秀成果，比如基

于神经网络算法的预测方法、基于决策树算法的预测方法、基于 Logistic 回归算法的预测方法。

本应用系统是在传统方法的基础之上，运用云计算技术，克服由于无法处理大量数据而采用小量数据导致拟合度不高，进而导致预测准确率降低的问题，并包含了对数据的清理、量化等预处理步骤，建立起对高血压风险预测高度拟合的数据模型，有效的提高了对高血压风险预测的准确率和效率，从而满足在医学检测方面的实质性的要求，为进一步对患者进行身体健康管理提供依据。

本应用系统有效的结合当前热门技术与传统预测方法，并在此基础上进行发展，优化数据处理过程，减少对大量数据的处理时间，建立起结构完整的疾病预测框架，为其他疾病比如糖尿病等的预测提供了很有效的参考意见，后人可以在此基础上快速的完成对其他疾病预测程序的实现。

## 1.2 高血压风险预测系统国内外发展现状

### 1.2.1 国内外发展现状

#### 1) 国外发展现状

在国外，人们通过在计算机内建立知识库，把规则化形式化的专家知识以及经验存入其中，用符号推理的方式，从而在相应的医疗领域形成专家系统，进行医疗诊断。数学模型首次被引入临床医学是在 1959 年由美国的 Ledley 等人引入的，并且他们还提出将布尔代数以及 Bayes 定理用做计算机诊断的，这也是从来没有过的创举；到了 1966 年，Ledley 又提出“计算机辅助诊断”这一概念，并且由此后来还发展成了计量医学；1976 年，医学专家系统-MYCIN 研制成功，该系统是由美国斯坦福大学的 Shortliffe 等人开发的，这个系统以鉴别细菌感染和治疗的优秀功能而著名，提示他们在建立该系统时还形成了一套完整专家系统的开发理论。1982 年，著名的 Internist-I 内科计算机辅助诊断系统面世，该系统是由美国匹兹堡大学的 Miller 等创建，该系统的知识库中含有 572 种疾病，约 4500 多种症状；1991 年，来自美国哈佛医学院的 Barnett 等人又开发了“解释”这一软件，该软件包含有 2200 多种疾病以及 5000 多种症状<sup>[5]</sup>。

除此之外，还有很多较为知名的应用系统。在 1948 年，美国的 Framingham

风险评估模型（Framingham Risk Source, FRS）就已经开始着手建立，该模型采用直接评分和多元回归等多种分析方法根据年龄、性别、吸烟、HDL-C 等风险因子来预测心血管类疾病的风险概率，成绩显著。而欧洲也在 2003 年也启动了系统性冠心病风险评估 SCORE 计划，这个计划旨在建立了一个欧洲区域的系统性的欧洲冠心病风险评估计划模型（Systematic Coronary Risk Evaluation, SCORE），这个风险评估模型适用于欧洲的临床实践。世界卫生组织（WHO）也在 2008 年发表了 WHO/ISH 风险预测图，该图用于降低心血管风险，以及冠心病等疾病的预防<sup>[6]</sup>。

虽然国外的有些系统已经相当完善了，但是由于我国国民体质以及生活习惯上和外国人有很大的不同，直接使用国外的系统用于国内高血压的风险预测，结果并不是很好，必须加以改进，难度较大。

## 2) 国内发展现状

当前国内对高血压的预测主要还停留在医学手段之上，在应用系统上的发展不是很多，而医学手段又由于医生个人经验和当前医疗手段的限制，从而最后预测的结果并不十分准确。比如一项特别的“冷加压”试验就曾在南京医科大学被专门组织起来，该试验是用来研究以及预测检测者今后获得高血压的风险几率，该试验是通过将检测者的手臂浸泡在盛满冰水的桶里，浸泡时间为 1 分钟，之后测量其血压变化情况，从而根据血压变化情况来预测该检测者以后患高血压的风险几率<sup>[7]</sup>。还有的是根据检测者的家族高血压病史来预测检测者今后是否会得高血压，此类方法不仅预测结果不准确，而且对个别人个别特殊情况的处理也不是很好。

除了建立在医学知识上的一些预测手段外，我国目前在预测应用系统方面也是有一定的发展的，1997 年，一个仿人疾病诊断的专家系统模型被张红梅等提出。张玉璞开发了基于波形分析的心血管疾病诊断专家系统，不过该系统只开发了原型系统<sup>[6]</sup>。再比如我们台湾徐光红等人就发明了采用基于实例的分类方法的手段对高血压病人进行诊断，准确率达到 70%，还有张诚丁等人利用数据挖掘技术挖掘高血压和高血脂的共同风险因素，并用得到的共同风险因素来预测高血压<sup>[8]</sup>。这些研究都得到了验证，而且都有较高的准确率，但是准确率依旧不是很高。

虽然应用系统建设不是很多，但在理论研究方面国内涌现出不少的成果，在基于神经网络技术的高血压风险预测，基于使用大数据技术进行分析的高血

压预测，基于决策分类树的高血压预测等方面有着不少的论文发表，这对本系统的建立起到很大的帮助作用。

### 1.2.2 存在的问题

尽管国内外高血压风险预测技术发展迅速，但存在的高血压风险预测系统还不是十分成熟，高血压风险预测技术仍然存在着一些不足。具体来说有以下几点：

#### 1) 不准确性

尽管高血压风险预测技术已经发展的比较好，国内外的一些高血压风险预测系统都已经可以运用于临床实践，而且也有了比较好的预测准确率。但是目前的系统要达到 90%左右都很困难，因此目前的高血压预测系统还存在着许多的不准确和不稳定，高血压风险预测准确性还有待提高。

#### 2) 性能不足

目前存在的高血压风险预测系统由于发展时间较早，鉴于当时的技术水平，系统所使用的技术都比较陈旧，并且当时系统所使用的硬件以及硬件上使用的技术架构都远比不上如今的水平，面对与日俱增的数据和计算量，这些高血压风险预测系统无疑存在着很严重的性能瓶颈，优待以如今的主流技术对其进行更新升级。

#### 3) 不适合国内，移植难度较大

如今比较成熟的高血压风险预测系统大都是在国外发展起来的，这些系统都是按照外国人的体质和身体体检数据建立起来的，这些系统对外国人的身体适应性很好，但是由于我国国民体质和外国人的体质有很大的不同，这些依照外国人体质而建立的高血压风险预测系统就不适合用于我国国民的预测，而对于这种大型的复杂的系统不管是技术原因还是人为原因，将其移植难度都很大。所以借鉴国外系统发展自己的系统很有必要。

## 1.3 本文主要工作内容

本文主要描述的是在高血压风险预测系统的开发过程中两个核心模块的开发任务，并描述了对这两个模块进行测试的过程，具体涉及的工作内容如下：

- 1) 参与并协助云计算平台 TAP 的搭建，包括服务器的摆放，线路的连接，系统配置，以及相关软件的安装；
- 2) 参与数据处理量化模块程序的设计和开发工作，主要是对数据处理规则的总结分析以及设计，并且完成该模块程序的编写工作；
- 3) 研究高血压风险预测数据模型，选择合适的模型并完成高血压风险预测模块程序的设计与实现，并且完成该模块程序的开发工作；
- 4) 对所开发的程序进行测试，完成相应功能测试、性能测试，并根据测试的结果对程序进行修复和优化。

## 1.4 论文的组织

本论文将从绪论、背景知识、需求分析和概要分析、高血压风险预测系统核心模块的设计、系统核心模块的实现、测试与分析以及结论这九个章节来详细介绍本系统。具体章节安排详述如下。

### 第一章 绪论

主要介绍论文的选题来源和依据，包括高血压风险预测技术的发展情况，以及在高血压风险预测方面国内外的现状和存在的问题。然后结合选题分析论文的主要研究内容，并对论文的组织安排进行简单介绍。

### 第二章 TAP 平台优势分析

通过介绍本系统底层云平台的组织架构，分析 TAP 平台架构的优越性，同时详细的分析该平台所采用的云计算技术，对主要的几个技术的优点进行进一步的阐述说明，阐明本系统为什么采用该平台的缘由。

### 第三章 需求分析与系统概要设计

结合用户的实际使用需求，对本系统在使用过程中存在的一些问题和不足进行详细的分析，得到具体的用户需求。并使用分析结果进行简单的概要设计，最后给出系统总体架构图和系统总体流程图。

### 第四章 数据处理量化模块的设计

根据需求分析和概要设计，明确本系统的设计目标。通过分析原始数据的数据特征并设计相对的处理规则，并在此基础上形成对该模块的总体设计。

### 第五章 高血压预测模块的设计

分析 TAP 平台提供的几个主要的数据模型算法，结合数据特征进行对比



分析，并从中挑选出最适合本论文的模型算法，并以此形成对该模块的总体设计。

#### 第六章 数据处理量化模块的实现

根据本模块的设计，结合相应的编程技术，对本模块的实现过程进行描述，并通过类图，时序图等直观的展示出程序的运行过程。

#### 第七章 高血压预测模块的实现

根据本模块的设计，先对本模块依赖的 **TAP** 平台的搭建过程进行简单描述，之后详细的介绍使用 **TAP** 平台提供的接口进行本模块程序的实现过程。

#### 第八章 系统核心模块的测试与分析

对上面实现的两个系统的核心模块模块进行测试，验证其准确性，可靠性，并通过相应的测试数据对本系统进行结果分析。

#### 第九章 总结与展望

对本系统的设计和实现进行总结，指出了设计中现有的一些不足之处，并对其未来的发展情况进行展望。

## 第2章 TAP 平台优势分析

本章主要介绍高血压风险预测系统所基于的云计算平台 TAP 平台，通过分析其架构和所使用到的技术的特点进行分析，凸显该平台的优越性，从而阐明本系统选择建立在 TAP 之上的必要性和本系统性能的优越性。

### 2.1 引言

Trusted Analytics Platform(TAP)是一个云计算平台，在该平台上数据专家和应用开发者能够创建和运行由大规模数据分析驱动的应用程序。

TAP 平台简化了大量在不同种类的用户实例和解决方案基础上进行知识发现和预测模型应用中的图形分析和机器学习。该平台使用一个可扩展、模块化的框架，在特征工程、图形架构、图像分析和机器学习之间提供一个分析管道。通过统一的图形和基于实体的机器学习，机器学习开发者可以合并实体附近的关系信息来生成一个更加优越的预测模型，该模型能更好的代表数据中的上下文信息。同时使用高级别的 python 数据编程抽象来大大缓解集群运算和并行处理的复杂性，使 TAP 平台的所有功能都可以在大规模数据量下正常运行。除此之外，TAP 平台也是基于插件架构的可扩展的，平台将全方位的分析 and 机器学习的任何解决方案合并到工作流中并对外提供迭代式的多样性的接口，从而减少研究人员理解方面的开销，提高集成性和效率性。

### 2.2 TAP 平台架构优势

本系统的 TAP 平台的体系架构如图 2.1 所示，TAP 平台是一个优秀的云计算平台，该平台简化了图像分析和机器学习。可以通过大量不同种类的用户用例和解决方案来建立预测模型和用于大数据的知识挖掘，使用独一无二的图形分析和机器学习来为开发者提供尽可能的符合实际情况的准确预测数据。TAP 平台里的所有功能都是可以大规模应用的，同时运用高级别的 Python 抽象数据编程来缓解集群计算和并行处理的复杂性。并且 TAP 平台支持插件架构是完全

可扩展的，如此便为图形分析和机器学习的任何解决方案提供了一个统一的工作流程，为开发者节省开销，只需使用 TAP 平台提供的统一的、集成的、高效率的接口就行。

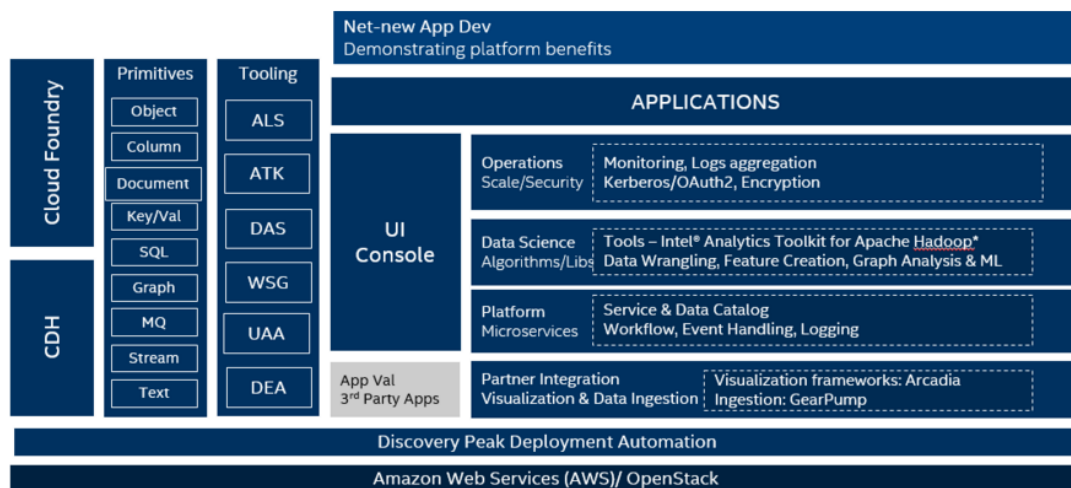


图 2.1 TAP 体系架构

从 TAP 平台的体系架构图中可以看出 TAP 平台基础硬件平台是基于 Openstack 或者 AWS 的，Openstack 是开源的免费的而且其功能强大，在自己的服务器上适合使用 Openstack 进行基础平台的搭建，而在没有服务器或者购买服务器自己搭建平台代价很大的情况下可以考虑使用 AWS 作为 TAP 平台的基础平台，AWS 有着 EC2, S3, ELB 等等一系列性能不错而且价格相对较低的优质服务，最重要的是使用 AWS 的服务可以随着项目规模的增大而动态的添加服务器以满足你的需求，这些操作只需要简单的配置些许文件就可以，这大大减少传统的硬件服务器升级的成本和管理成本。本高血压风险预测系统选择在自己的服务器上搭建 TAP 平台，主要原因是考虑到 AWS 的服务要通过网络连接，网络传输速度比较缓慢不如本地服务器搭建平台的快速，而且不管是部署 Openstack 还是部署 TAP 平台都比较简单。

在基础平台 Openstack 之上是 Data Platform，这一层 TAP 是基于 CDH 的，CDH 是 Cloudera 公司开发的简化版的 hadoop，是在 hadoop 基础上进行的一些封装，能够让用户可以更低成本的使用 hadoop 而不是需要经过传统的 hadoop 那样的繁琐安装和配置。CDH 所在的数据平台层主要的组件包括 HBase, HDFS, Kafka, Spark, YARN, Zookeeper 等，HBase 是一个开源数据库，该数据库是分

布式的面向列的；HDFS 是一个分布式文件系统；Kafka 是一种分布式发布订阅系统，该系统具有高吞吐量的特点；Spark 是一个通用并行框架，该框架类似于 Hadoop MapReduce，是一种开源集群计算环境；YARN 是一种通用资源管理系统，该系统可为上层的应用提供统一的资源管理和调度服务；而 Zookeeper 则是一个应用程序协调服务，该服务是分布式的，所以可以为分布式应用提供一致性服务。这些组件组成了 TAP 平台的强大数据存储，处理以及为应用层协调服务和资源能力的基石。

在 Data Platform 之上是 Application Platform 即应用服务层，这一层 TAP 平台主要是使用 Cloud Foundry，Cloud Foundry 是一个开源的，是属于 PaaS 层的云平台，使用 Cloud Foundry 可以使开发人员可以在极短的时间内进行应用的部署和扩展，而无须担心任何基础架构的问题。TAP 的应用服务层主要有 Director，Blob store，Workers，Message bus，Health monitor，Agents，Cloud platform integration 这几个组件组成。除了 Cloud Foundry 为 TAP 平台应用服务层提供的那些核心服务外，TAP 平台在应用服务层还有 Key/value stores，Document stores，Relational stores，Memcache stores，Graph stores，Message queues 等一些可以根据需要自行添加的服务。这么多的服务足见 TAP 平台的强悍，而这也是我们选择 TAP 平台作为高血压风险预测系统底层平台的一个重要原因。

TAP 平台不仅仅是将现在主流的一些云计算技术堆叠在一起，而是以这些技术为基础，整合这些技术，充分发挥这些技术的特长，强强联合并且加以以自己的协调处理技术，比如在数据平台和基础硬件平台之间 TAP 平台有一层 Discovery Peak Deployment Automation，这一层的添加能够使 PaaS 层和 IaaS 层能够更好的结合在一起，另外在数据平台层和应用服务层 TAP 平台还添加了一个 Shared Services，这一服务主要是用来为应用服务层提供一些数据平台层多点架构的一组共享服务，并且可以用来管理上面我们提到的应用服务层的那些服务。

通过对 TAP 平台体系结构的了解，TAP 平台所拥有的强大数据存储处理能力，应用协调资源管理能力，以及对开发人员非常友好的应用部署和扩展能力，这些能力促使我们的高血压风险预测系统选择了它，TAP 平台的使用会让我们的血压风险预测系统性能更好，运行更快，响应更迅速，对大数据的支持更好，对复杂模型的处理预测更加符合我们的预期。

下面将会对 TAP 平台中使用到的云计算技术进行进一步的阐述和分析，从

而使大家能够更好的理解高血压风险预测系统是一个性能优秀，技术先进的系统。

## 2.3 平台核心技术优势分析

TAP 平台的优越性不仅体现在其平台架构的优越性，扩展性，还体现在该平台所采用技术的先进性，本节将通过对 TAP 平台所采用的一些技术进行分析，从侧面展现出其优势所在。

### 2.3.1 OpenStack 技术分析

TAP 平台采用的 OpenStack 是当前最优秀的云计算技术之一，正如其名，Open 开放之意，Stack 则为堆砌，OpenStack 合起来就是软件堆积的集合，事实上 OpenStack 指的是一些开源软件并且这些软件组合成了一整套功能强大的系统，该集合方便企业或者服务提供商建立和运行自己的云计算和存储设备，Rackspace 和 NASA 当年因各自需要开发出 Nova 和 Swift，后来一拍即组合成了最初的 OpenStack，随着 OpenStack 的发展和其他一些公司的加入，OpenStack 逐渐发展成了以 Nova 计算服务、Swift 存储服务、Glance 镜像服务、Keystone 认证服务和 Horizon 界面 UI 服务几大服务为核心的开源服务软件集合，即为现在的 OpenStack。

OpenStack 在众多开源的项目中其发展是非常迅速的，其中蕴含着无数的开发者的智慧，使得 OpenStack 提供的几种服务尤为强大。其中的 Nova 是一个 OpenStack 的弹性计算的控制器，在 OpenStack 的整个实例生命周期中所有的动作都是由 Nova 来进行处理，Nova 作为 OpenStack 中管理的角色，它负责管理整个云计算的资源、网络、授权等等事务，即使 Nova 并没有虚拟化方面的能力，但是 Nova 可以和虚拟机进行交互，并以 web 服务 api 的形式对外提供处理接口。

除了 Nova，OpenStack 的 Glance 服务更为 OpenStack 提供一套虚拟机镜像发现、注册和检测的系统，该镜像服务支持多种虚拟机镜像格式，更能够方便的对虚拟机的镜像进行管理。

Swift 作为 OpenStack 的对象存储器，它有着分布式的持续的虚拟对象存储功能，类似于 AWS 的 S3，Swift 可以跨越节点进行百万级的对象存储，该服务

有着支持海量存储，大文件存储，数据冗余管理，对象安全存储等功能及特点。

OpenStack 提供的 Keystone 则是提供认证和访问策略的服务，能够对其他服务比如 Swift、Glance、Nova 等进行认证和授权，是提供 OpenStack 安全保障的很重要的一部分。

除了上面提到的这些 OpenStack 还有一个 Horizon 服务，该服务就是一个为管理和控制 OpenStack 服务的 web 界面，在 Horizon 的 web 控制面板上可以很方便的对其他服务进行管理，简化用户使用 OpenStack 的难度。

正是由于 OpenStack 具有如此之多的优秀服务，从而使 TAP 平台在数据存储，安全验证等方面处于一流水准，OpenStack 奠定了 TAP 性能优秀的基石。

### 2.3.2 CDH 技术分析

TAP 平台使用的另一个云计算技术是 CDH，该技术是 cloudera 公司对 hadoop 的安装过程进行一些简化，对 Hadoop 部分组件进行再一次的封装，从而给用户提供一个易学易用的 hadoop 及其相关组件的封装。使用 CDH 进行 hadoop 的安装十分的简单，在其官方文档上的三种安装方式都十分简单。

Cloudera 的 CDH 是对 hadoop 做的这些封装，和在 hadoop 的基础上做的这些优化，使得 Cloudera CDH 在兼容性、安全性以及稳定性等方面都比 Apache hadoop 更加优秀。

Cloudera CDH 有非常清晰的 hadoop 版本划分方式，并且总是应用最新的 bug 修复，更新速度也较 Apache 快速很多，同时 CDH 提供 Kerberos 安全认证，安全性非常高，除此之外 CDH 还文档齐全，安装、升级和使用都非常之方便。

### 2.3.3 Cloud Foundry 技术分析

云计算技术在 IaaS、PaaS、SaaS 三种不同方向进行发展，在 PaaS 领域，Cloud Foundry 是一个功能强大且应用非常广泛的技术。VMware 在推出 Cloud Foundry 之后，Cloud Foundry 就以其对多种框架、语言、运行时环境以及云平台和服务的强大支持能力而大受欢迎，使用 Cloud Foundry 可以使开发人员在极短的时间之内把应用部署起来，并且也可以同时进行应用的扩展，而在这一过程中开发人员不需要担心基础架构的问题。

Cloud Foundry 使用将应用和应用依赖的服务隔离开的做法来使应用的部署更加灵活，同时 Cloud Foundry 本身也是高度模块化的分布式系统，可以自由的部署在 OpenStack 等公有云、私有云、混合云等云环境之上。

Cloud Foundry 的架构中，整个平台的流量入口是 Router 这一组件，负责将外部用户请求以及平台内部的请求分发到相应的组件之中。而 Authentication 是一个包含了登录服务和验证服务的用户通道。在 app 的整个生命周期都是由 Cloud Foundry 中 Cloud Controller 负责，用户可以通过 cf 命令行工具和 Cloud Foundry 进行交互。除了这些 Cloud Foundry 还有 Message Bus 这样一个内部组件通信的媒介，这样的一个分布式消息队列系统让 Cloud Foundry 可以松散的耦合在一起。

正是有着 OpenStack、CDH、Cloud Foundry 这些优秀的云计算技术的支撑，才使得 TAP 平台在性能，运算速度，服务协调等等方面都比一般的平台更为的优秀，而这正是本系统采用此平台的依据，采用该平台能使本系统预测的更快，预测的更准。

### 2.4 本章小结

本章主要介绍高血压风险预测系统使用到 TAP 平台，通过对 TAP 的核心技术 OpenStack、CDH、Cloud Foundry 等等一些云计算技术的分析，从而直观的展示出 TAP 平台的优秀性。从本章的介绍之中可以看出本高血压风险预测系统所采用的云计算技术都是很先进，功能非常强大的，这也从侧面反映出本系统性能的过人之处。

通过本章的了解，明确本系统所采用平台技术之优秀，在这样的技术背景之下本系统应运而生，从而引出下一章对本系统需求分析的介绍。

## 第3章 需求分析与系统概要设计

本章通过从高血压风险预测的总体需求出发，详细分析系统用户实际需求，结合本系统的具体内容，针对这些需求进行分析，并给出简单的概要设计。

### 3.1 引言

医学经过了长久的发展，无数的先贤医师在医术上倾注毕生心血，一种种的疾病被发现，一种种的医疗预防方式被公布，人类的寿命在这些医师的努力下，不断的被延长。从原始的畏疾忌医求助神明，到现在的有病求医，都是人们信赖医学技术的表现。

高血压病很好的体现了这一进程，很久以前医生都不知道血压这一概念，直到近代生理学之父 William Harvey 出版了《心血运动论》之后，医学界才知道“血液循环”这一现象。而即使如此医生却也没办法能够测量出血压的大小。到 1896 年意大利医生里瓦罗基发明出了后来广泛流传的血压计，医生才有可安全测血压的工具，患者终于可以获知自己血压数值，至此高血压的检测工具一直是血压计。

显然血压计能够用来测量你是否患有高血压疾病，但是当你获知你已经患有高血压病的时候已经为时已晚，如果能在你患有高血压病之前就可以知道你是否会患有高血压，那么对个人健康来说是非常重要的。于是在这种需求之下，无数的医生以及医疗机构就进行了新一轮的研究和发明。

在需求的驱动下，国内外就诞生了几种专家系统和诊断系统，这些将科技与传统医学结合起来的方式是医学在新科学技术环境下产生的新的需求。同时鉴于新时代人们的生活方式和习惯，催生出了一系列的需求。

### 3.2 总体需求分析

从高血压的历史发展和高血压风险预测历代系统的使用情况分析，高血压风险预测的总体一般性需求如下。



#### 1) 方便的操控方式

高血压预测系统被设计出来之后不管是给医生使用还是给用户自己使用，方便的操作方式是系统使用者必然的需求，不管系统多么的复杂难懂，展现给用户的界面都应该是简单，傻瓜式的。如果系统的按钮经常不知道在哪里，或者启用一个功能按键次数过多，都会给用户带来不便，得不到良好的交互体验。

#### 2) 快捷的体验

当今社会人们的生活节奏都很快，特别是在大都市的居民，每分每秒都是匆匆忙忙而又珍贵的，所以高血压风险预测系统就应当满足用户体验的快捷性，让用户能够在很短的时间内完成预测的过程，而不是经过漫长的等待才等到结果的出现，这样的体验肯定是极差的。

#### 3) 多功能

随着人们对健康的关注，未来人们关注的肯定不止是高血压这种疾病，有可能还包含糖尿病等多种疾病，那我们的预测系统就不能单一的只能用来预测高血压这种疾病，需要对其他疾病的预测进行支持。另外我们的系统应该不只是对疾病进行预测，还能通过预测结果对用户的生活方式或者习惯提供针对性的建议，帮助用户更好的健康发展，满足用户多需求的特点。

#### 4) 易扩展性

作为高血压疾病的预测系统，其中包含的医学知识，技术都是很复杂的，建立起这样一个系统的代价是极大的，那么，作为这样一个系统那就应当是可扩展性强的，这样可以节约资源和成本。

#### 5) 可靠性

作为一个预测系统，其主要责任在于在人们还没有患病之前就将这种可能性预测出来，从而来预防该病的发生。所以预测的可靠性是非常重要的，并且除了预测结果的可靠性，还包括预测过程的可靠性和预测结果的有依据性。

### 3.3 功能性需求分析

通过上面对高血压风险预测系统总体需求的分析，结合到本系统的具体用户角色，即高血压风险预测系统中的数据管理员和预测观察员这两种用户，其角色对系统的功能性需求如下。

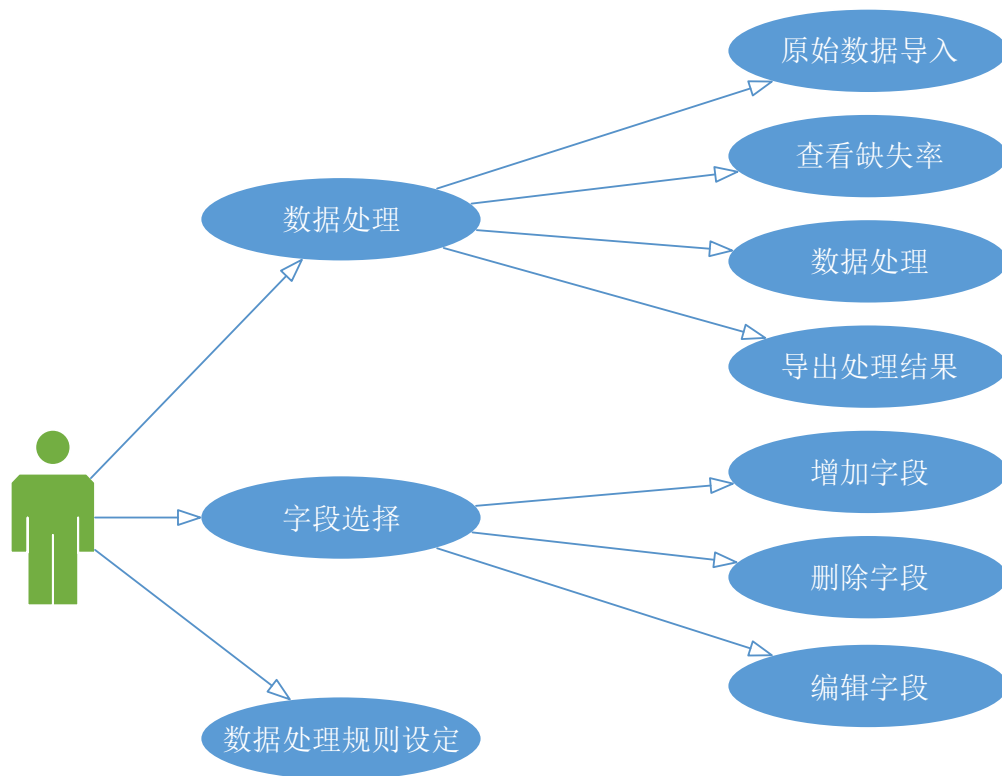


图 3.1 数据管理员需求用例

高血压风险预测系统中很重要的一个操作对象就是数据管理员。面对采集到众多格式不一，种类繁多，数据量大的原始数据，只有经过数据管理员的整合、填补、量化等操作才能符合后面训练模型以及进行预测的要求。数据库管理员在导入原始数据之后应能查看各字段的缺失率，如此才能为字段的选择提供决策支持，同时数据管理员也能将处理好的数据导出为我们指定的格式进行存储。数据管理员通过字段选择功能选择出我们需要处理的字段，筛选出我们需要的字段，节省我们处理数据的时间，除此之外数据管理员还能在字段上进行修改，修改字段名或者字段类型甚至删除字段。数据的处理依赖一定的处理规则，而数据管理员也能根据实际需求修改相应规则。数据管理员需求用例图如图 3.1 所示。

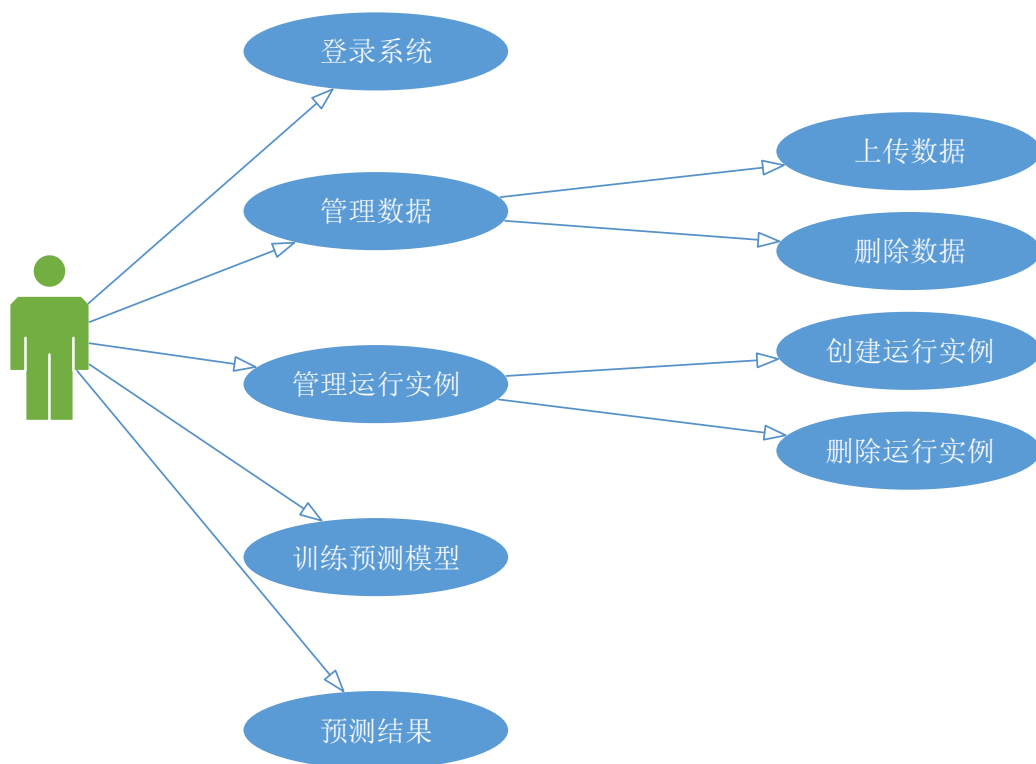


图 3.2 预测观察员需求用例

预测观察员用户主要是在云计算平台（TAP）上进行操作，操作界面是一个网站，在用户登录网站之后可以将之前数据管理员处理好的数据文件上传到 TAP 平台的 hdfs 上，同时在 TAP 上建立属于自己的运行实例用于运行用户的应用，预测观察员可以在 TAP 提供的命令窗口里输入自己的代码，使用上传好的数据，选择合适的模型进行训练，训练完成的模型可以用于预测观察员进行高血压的预测，并且查看预测结果以及准确率。预测观察员需求用例图如图 3.2 所示。

本需求着重描述高血压风险预测系统设计与实现。此系统主要包括数据处理量化模块、数据管理模块、运行实例管理模块、高血压预测模块等。

### 3.4 非功能性需求分析

作为高血压风险预测这样一个大型系统，其除了要满足不同的功能需求外，还应该满足以下的一些非功能性需求。

(1) 操作系统：系统必需要有一定规模的用户使用，这样的系统才能快速的在消费者和商家中得到广泛的应用；能支持 windows 应用程序的运行，保证系统正常工作。

(2) 界面需求：对于数据管理员用户，需要使用系统处理大量数据，界面应该有友好的设计，清晰的结构，方便的操作，以及良好的用户体验；同时对预测观察员用户来说，界面需要能够方便的访问，操作要简单快捷，结果显示要清晰明了。

(3) 通信网络：因为要上传大量数据，通信网络应有高的带宽，能够支持数据的迅速上传。

(4) 数据存储：系统需要对大量数据存储，要支持存储量可扩展，要能够方便的对数据进行操作。另外要保证存储的安全性和可靠性。

(5) 设备：搭建云计算平台需要多台性能较好服务器，有足够多的内存以及存储空间，并且服务器之间互相连接，并且可以访问网络。

(6) 操作人员：由于用户直接在 TAP 平台上运行代码，所以要求用户有计算机软件或者相关经验。

### 3.5 系统概要设计

需求决定设计，承接上一节的需求分析，通过对高血压风险预测系统具体需求的分析，以这些需求出发进而得出对整个高血压风险预测系统的概要设计。正是前面提到的这些需求才决定我们的高血压风险预测系统是设计方式，同时也只有这样的设计才能满足这么多复杂的需求。本节将分别介绍系统的总体架构和部分核心模块的架构设计图。

#### 3.5.1 目标与概述

本设计将实现在云计算平台（TAP）上的高血压风险预测系统，系统将集数据填补、数据清理、数据量化、模型训练、结果预测于一身，为医疗机构提供可扩展、高准确率、运行高效的高血压风险预测服务。

本设计在设计完成后需要对以下的功能进行支持：

1. 用户可以在客户端实现数据字段选择、缺失率查看、处理规则修改、数

据处理、数据导出等功能。

2. 用户可以通过浏览器登录 TAP 云计算平台界面，并实现数据的上传、数据删除、运行实例创建、运行实例删除、模型训练、预测结果等功能。

另外，本设计在安全性、可维护性以及稳定性上的要求都必须满足要求。安全性要求为预测观察员对平台的操作，对数据的管理，对运行实例的管理等环节进行访问权限的管理与控制，也就是对用户的合法权的保护。可维护性就是要求系统文档、代码指示等要遵循软件开发的规范。稳定性就是要求系统性能的稳定性和持续性，在用户访问时响应速度必需满足用户需求。

### 3.5.2 系统总体设计

基于云计算平台的高血压风险预测系统主要是为用户高血压风险预测在大量数据处理、模型训练等大运算量的情况下提供一种技术解决方案，克服传统预测方法数据量小，以及模型拟合度不高，预测结果不精确的问题。同时，为了方便用户的使用，系统提供应用程序界面以及 web 界面，用户可以在应用程序界面对海量数据进行处理，同时也可以 web 界面上对平台和项目的一些配置进行管理，实现本系统的预测功能。

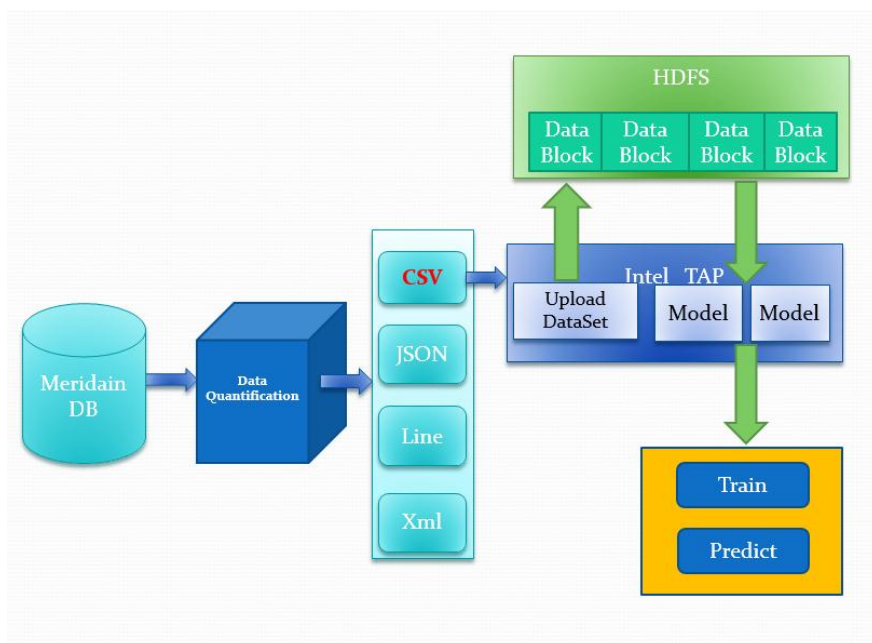


图 3.3 高血压风险预测系统的总体架构

高血压风险预测系统的总体架构如图 3.3 所示。客户可通过应用程序界面进行 Data Quantification 过程对外部数据源（Meridain 公司）发送过来的数据进行整理、填补、量化、清理操作，最后导出为 TAP 平台支持的数据文件格式，然后通过上传 TAP 平台，用户可以直接在平台上使用这些数据，用户在 TAP 平台上可以进一步管理这些数据，同时用这些数据进行训练模型和预测结果。

### 3.5.3 量化程序开发架构

本系统量化程序的体系结构如图 3.4 所示，采用 xml 的方式来保存数据的处理规则，量化程序通过解析 xml 文件获得相应处理规则对数据的分类和量化以及缺省值的填补，Feature 列表文件保存的是我们所选择的字段，量化程序会根据我们选择的字段来处理，而并不对其他字段进行处理，节省运行时间。



图 3.4 量化程序客户端的体系结构

数据管理员通过 windows 应用程序对数据进行操作，视图的界面设计利用 C# 的布局控件进行，通过客户端的功能模块响应控件事件处理来实现与用户的交互。数据上传后，量化程序会计算各字段的缺失率，并显示在界面上。

### 3.5.4 高血压风险预测系统总体模块设计

高血压风险预测系统主要包括四个功能模块：数据处理量化模块、数据管理模块、运行实例管理模块、高血压预测模块。接下来将使用流程图对系统的

总体模块进行简要的描述。

### ● 功能模块流程

系统的主要工作流程如图 3.5 所示。从图中可以看出数据管理员用户首先判断数据是否已经处理过，如果没有处理过就进入数据处理量化模块，对数据进行填补、量化等的操作，否则就将处理过的数据交给预测观察员，预测观察员登录本系统的 TAP 平台，将数据上传到 TAP 平台之中，具体的是 TAP 中的 hdfs 之中，然后预测观察员在 TAP 创建运行实例后，就可以运用运行实例和数据来训练我们的高血压风险预测模型，最后预测结果，整个工作流程大体如此。

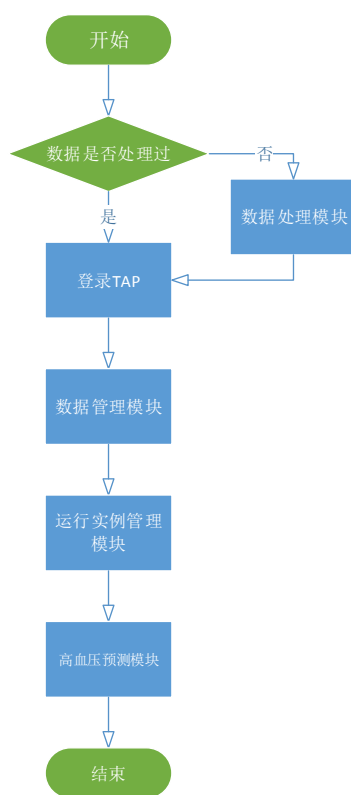


图 3.5 系统的主要工作流程

在图 3.5 的流程图中，描述四个模块：数据处理模块、数据管理模块、运行实例管理模块和高血压预测模块。这四个模块中，数据管理模块和运行实例管理模块是属于 TAP 平台之中的，所以在以下的章节里将不再进行描述，而作为本人主要完成的数据处理量化模块、高血压预测模块不仅是高血压风险系统的核心模块，而且更是本人主要完成的模块，所以下面的章节会着重讲解这两个模块。

### 3.5.5 方法和工具

#### ● 设计目标

在应用运行平台上,采用现阶段最热门的云计算技术 Openstack、CDH、Cloud Foundry 等搭建起云计算平台 TAP 平台,并通过 TAP 的分布式和集群运算来提高应用运行效率。

在数据处理上采用 xml 解析技术,并通过 C#技术实现前台和后台数据逻辑处理及通信。

开发环境为 Visual Studio, Web 容器为 Tomcat server, 数据库为 Hdfs。

#### ● 采用的方法

高血压风险预测的数据处理模块是采用面向对象的设计方法进行开发的, TAP 平台的前后台是使用 http 协议进行通信,而本系统的模型预测模块则使用 ssh 进行连接的。

#### ● 技术难度及特色分析

按照如今云计算技术的发展速度,技术每时每刻都在进行更新,为保证本系统的先进性,开发团队经常对平台进行更新,再加上如今云计算技术并不是相当成熟,给平台的设计和搭建带来很大的挑战。而面对这些崭新的框架和技术的学习,给开发者带来很大的时间成本,这些是本系统开发人员的最大障碍。

除此之外,对预测模型的选择也是一个难点,为了尽可能的找到适合预测高血压风险的模型,开发人员需要学习了解以及使用不同种类的数据模型,这给开发人员带来相当大的学习成本。

最后由于本系统应用的使用直接在 TAP 平台之上,这就要求用户有一定的计算机软件相关知识,这无疑是使用者的一大门槛。

本系统拥有很多特色

1. 本系统基于的 TAP 平台采用了最新的云计算技术,计算资源以及存储容量理论上无限提升。
2. 本系统作为高血压风险预测方面的系统,准确率非常之高。
3. 本系统作为平台级别的系统,可快速的更换模型进行其他方面的预测应用。
4. 支持海量数据的处理。
5. TAP 平台的处理速度快、响应时间短、性能稳定。



### 3.6 本章小结

本章首先从高血压的历史出发，通过介绍高血压检测技术的发展，分析人们对高血压疾病的认识过程，进而概括出高血压预测系统的总体需求。然后将总体需求落实到我们的高血压风险预测系统上，详细的介绍了高血压风险预测系统不同角色的需求，结合这些需求，使用工程化方法使用用例图将需求具体化，从而让这些需求主导了本高血压风险预测系统的设计和实现的整个过程。

之后，本章通过需求进而引出高血压风险预测系统的总体设计，介绍该系统的总体架构设计，总体模块流程设计以及核心模块数据处理模块的总体架构设计，通过这些设计进而关联到开发该系统大概要使用到的技术和工具。

本章从需求出发得出的总体设计为下面章节的系统核心模块的设计奠定基础，另外也为系统核心模块的设计提供依据。

## 第4章 数据处理量化模块的设计

本章主要通过对原始数据的特性进行分析，得出其特征，从而进一步根据这些特征设计相应的数据处理规则，在数据处理规则设计完成后对整个模块的工作流程进行分析，形成数据处理量化模块的总体流程图。

### 4.1 引言

从上一章的系统总体流程图可看到在系统开始时首先要判断的就是原始数据是否经过处理，如果没处理则由数据处理量化模块也即本模块进行处理。在本系统设计之初，本项目所获得的原始数据是从不同医疗体检机构的不同数据库中导出的，并且因为公司政策即安全性方面的原因，我们得到的数据是保存为 TXT 格式的，在数据格式上，内容上都有着复杂的缺陷。

对于这些有缺陷的数据本系统的预测程序是无法直接用于模型的训练和预测的，我们需要对这些原始数据的特征进行分析，并进一步总结其规则，在结合相应的医学知识对这些数据进行处理和量化，这样处理过后的数据才能直接使用。

### 4.2 原始数据特征分析

为便于本论文对原始数据的特征进行分析，并直观的展示出来，本节特意截取了部分数据，如图 4.1 和图 4.2 所示。

## 第 4 章 数据处理量化模块的设计

PE_ID	HPI	GXY	PE_date	Age	Sex	Height	Weight	BMI	YW	WHR	DBP	SBP
iL	ALT	AST	AKP	GGT	ADA	TPO	Aib	Gib	A/G	PA	BA	AMS
1203280179		010104.	001	No	1/14/2013	42	48	男	174	81.6	27	
1204190214			No	1/15/2013		42	女	158	77	30.8		
1208010049			No	2/7/2013		43	女	160.5	70.2	27.3		
1208010054			No	1/31/2013		24	男	178.5	72.5	22.8		
1208140218			No	1/10/2013		42	女	168	62.2	22		
1208170230			No	1/24/2013		31	男	173	83.9	28		
1208230070		012201.	015	Yes	3/4/2013		73	女	162	49.8	19	
1208230210			No	3/4/2013		78	男	168	53.2	18.8		
1208270051		012201.	015	Yes	1/6/2013		53	男	164.5	80.7	29.8	
1208270061			No	1/6/2013		47	男	169	72	25.2		
1208300125			No	2/20/2013		54	男	169	71	24.9		
1209100061			No	5/31/2013		42	男	179.5	87.2	27.1		
1209100069			No	1/11/2013		28	男	176	87.8	28.3		
1209100079			No	1/15/2013		27	男	169.5	65	22.6		
1209110106			No	3/26/2013		49	男	171.5	74.2	25.2		
1209110108			No	3/21/2013		28	男	172.5	73.3	24.6		
1209141341			No	1/25/2013		37	男	183	73.2	21.9		
1209170260			No	1/24/2013		23	女	165.5	47.6	17.4		
1209170263			No	1/29/2013		55	男	174.5	69.9	23		
1209170326		010109.	006	No	3/4/2013		31	女	165	56.2	20.6	
1209210097			No	2/17/2013		24	男	175	73.2	23.9		
1209210174			No	1/31/2013		34	女	176	64.9	21		
1209210617			No	1/11/2013		41	男	180	83	25.6		
1209250034			No	5/14/2013		43	男	176	79.9	25.8		
1209270079			No	1/9/2013		24	女	169.5	59.9	20.8		

图 4.1 部分数据 1

PH	PRO	GIu	KET	BLD	BIL	URO	NIT	SG	LEU	N_QT
Hcy	RF	ASO	CRP	CD	IFN	IgA	IgG	Hs_CRP	IgM	C3
5	5	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	00001
5		000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	
6		000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000068.001;	000010.002;	000010.002;	
8.5		000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	
5		000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000068.001;	000010.002;	000010.002;	
7		000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	
7	5	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	00001
5.5		000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	
	7	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	00001
5.5		000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	
5		000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	
5.5		000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	
5.5		000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	
5.5		000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000041.001;	000010.002;	000010.002;	
6		000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	
5		000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	
7		000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	
6		000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	
5		000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	
	5	000010.002;	000010.002;	000010.002;	000011.005;	000010.002;	000010.002;	000010.002;	00001	
6		000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000082.002;	000010.002;	000010.002;	
7.5		000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	
6		000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	000010.002;	

图 4.2 部分数据 2

在图 4.1 和图 4.2 中，头两行为字段名称，总共有 386 个字段，每个字段代表不同的体检项目，字段之后的行中为数据，每个字段的数据以空格或者“;”隔开，数据格式不一有字母、有汉字、有日期、有数字。同时图 4.2 展示的数据其中的如 000010.002 等样式的数据并不是该字段的体检值而是该项检查症状对应的代码，这些代码并不能作为数据直接用于本项目的高血压风险预测，需要

根据其意义对其进行量化。

在图 4.3 所展示了部分字段代码与检测结果的对应表，原始数据文件中只存放了字段 000010.002 这样格式的代码，在本系统的数据处理量化模块中我们需要根据这个代码所对应的检测结果值并将这个值量化成数字，比如图中黄色部分，我们规定将检测结果为阴性量化为 0，将检测结果为阳性量化为 1。

BIL	BIL前6位	文字说明	分类
000010.002	000010	(-)	阴性
000011.005	000011	1+	阳性
000012.002	000012	2+	阳性
阴性：0 阳性：1			
URO	URO前6位	文字说明	分类
000010.002	000010	(-)	阴性
000011.005	000011	1+	阳性
阴性：0 阳性：1			
PRO	PRO前6位	文字说明	分类
000010.002	000010	(-)	阴性
000011.005	000011	1+	阳性
000012.002	000012	2+	阳性
000015.001	000015	+-	弱阳性
阴性：0 阳性：1 弱阳性：2			
Glu	Glu前6位	文字说明	分类
000010.002	000010	(-)	阴性
000011.005	000011	1+	阳性
000013.002	000013	3+	阳性
000015.001	000015	+-	弱阳性
000012.002	000012	2+	阳性
阴性：0 阳性：1 弱阳性：2			
VC	VC前6位	文字说明	分类
000010.002	000010	(-)	阴性
000011.005	000011	1+	阳性
000015.001	000015	+-	弱阳性
000013.002	000013	3+	阳性
000012.002	000012	2+	阳性
阴性：0 阳性：1 弱阳性：2			
HBsAb	HBsAb前6位	文字说明	分类
000015.011	000015	弱阳性	弱阳性
000011.013	000011	阳性	阳性
000010.009	000010	阴性	阴性
000010.002	000010	(-)	阴性
000011.002	000011	(+)	阳性
阴性：0 阳性：1 弱阳性：2			
HBeAg	HBeAg前6位	文字说明	分类
000010.009	000010	阴性	阴性
000010.002	000010	(-)	阴性
000011.013	000011	阳性	阳性
000015.011	000015	弱阳性	弱阳性
阴性：0 阳性：1 弱阳性：2			
HBeAb	HBeAb前6位	文字说明	分类
000010.009	000010	阴性	阴性
000010.002	000010	(-)	阴性
000011.013	000011	阳性	阳性
000015.011	000015	弱阳性	弱阳性
000011.002	000011	(+)	阳性
阴性：0 阳性：1 弱阳性：2			
HBcAb	HBcAb前6位	文字说明	分类
000010.009	000010	阴性	阴性
000010.002	000010	(-)	阴性
000011.013	000011	阳性	阳性
000011.002	000011	(+)	阳性
000015.011	000015	弱阳性	弱阳性
阴性：0 阳性：1 弱阳性：2			

图 4.3 部分字段代码对应说明

除了对字段代码对应结果的量化外，在原始数据中我们还会遇到这样的情况，原始数据中字段存放的数据为实际检测的值，但是这些字段值所对应的结果会根据这些值进行分类，更复杂的情况是有些字段需要依据其他字段的值进行分类，如图 4.4 展示了部分字段的分类要求。

在图中可以看出，BMI 字段表示的是体重指数，其值只需要根据大小进行分类就可以了，该字段小于 18.5 则字段对应结果为偏轻，在 18.5 到 24.9 之间则

对于结果表示正常，大于 24.9 则表示超重。而 ALT 字段所代表的丙氨酸氨基转移酶的值需要工具体检者的性别进行结果的分类，男性小于 12.6 表示偏低，女性小于 9.2 才表示偏低。这些分类情况在本系统的数据处理量化模块中需要进行处理，并且对分类好的结果进行量化。

字段名	数据类型	中文描述	正常值范围	分类
ALT	int	丙氨酸氨基转移酶(ALT)[谷丙转氨酶]	速率法：10-40 U/L；分光光度计 男性：下限0.21 ukat.L <sup>-1</sup> ( 12.6 U.L <sup>-1</sup> )，上限0.75 ukat.L <sup>-1</sup> ( 44.8 U.L <sup>-1</sup> )；女性：下限0.15 ukat.L <sup>-1</sup> ( 9.2 U.L <sup>-1</sup> )，上限0.54 ukat.L <sup>-1</sup> ( 32.3 U.L <sup>-1</sup> )。	偏低：男性-<12.6，女性-<9.2； 正常：男性-12.6~44.8，女性-9.2~32.3； 偏高：男性->44.8，女性->32.3
AST	int	天门冬氨酸氨基转移酶(AST)[谷草转氨酶]	速率法：10-40 U/L；分光光度计 男性：下限：0.31 ukat·L <sup>-1</sup> ( 18.6 U·L <sup>-1</sup> )；上限：0.55 ukat·L <sup>-1</sup> ( 32.7 U·L <sup>-1</sup> )；女性：下限0.26 ukaII·L <sup>-1</sup> ( 15.5 U·L <sup>-1</sup> )；上限：0.47 ukat·L <sup>-1</sup> ( 28.1 U·L <sup>-1</sup> )	偏低：男性-<18.6，女性-<15.5； 正常：男性-18.6~32.7，女性-15.5~28.1； 偏高：男性->32.7，女性->28.1
AST/ALT	decimal(5,2)	谷草/谷丙(比值)	≤1	正常：≤1； 偏高：>1
BASO#	decimal(5,1)	嗜碱性粒细胞绝对值(BASO#)	0-0.06X 10 <sup>9</sup> /L	正常：0~0.06； 偏高：>0.06
BASO%	decimal(5,1)	嗜碱性粒细胞比率(BASO%)	0-1%	正常：0~1； 偏高：>1
BMI	decimal(3,1)	体重指数(BMI)	18.5-24.9	偏轻：<18.5； 正常：18.5~24.9； 超重：25~29.9； 肥胖：≥30
BUN	decimal(5,2)	尿素氮(BUN、UREAN)	成人：3.2-7.1mmol/L；婴儿、儿童：1.8-6.5mmol/L	偏低：成人 ( 年龄≥18 ) -<3.2，婴儿、儿童 ( 年龄<18岁 ) -<1.8； 正常：成人 ( 年龄≥18 ) -3.2~7.1，婴儿、儿童 ( 年龄<18岁 ) -1.8~6.5； 偏高：成人 ( 年龄≥18 ) ->7.1，婴儿、儿童 ( 年龄<18岁 ) ->6.5
Ca2	decimal(5,2)	血清前白蛋白(PA)	2.25~2.58mmol/L	偏低：<2.25； 正常：2.25~2.58； 偏高：>2.58

图 4.4 部分字段值分类

处理了上面所提到的原始数据的处理量化和格式的调整后，因为原始数据的 386 个字段中的有很多字段都缺少值，不同字段缺省的程度不一，在本系统的数据处理量化模块的设计时我们统计了下个字段的缺失率如表 4.1 所示。

表 4.1 部分字段缺失率

Feature	Rate
Sex	0%
GXY	0%
Age	0%

续表 4.1

AST/ALT	1.12%
ALT	1.13%
WBC	2.07%
...	...
ANA	99.99%
ACA	99.99%
ZQGJJC	100%
Zn	100%
...	...

表中省略了部分字段的缺失率，从表中可以看出很多字段的缺失率很大，有的都高达 70% 以上，还有的缺失率是 100%，对于这些字段因为没有任何数据存在，对本系统的高血压风险预测无任何意义，这样的字段应该直接予以删除处理，除了缺失很严重的字段，对于那些缺失不是很多的字段我们也需要对其缺失的字段值进行填补，填补的方法可以是中位数填补也可以是平均数填补。根据字段的缺失率对字段进行选择 and 填补对本系统的意义非常重大，合理的减少字段数可以大大的减少程序的运算量，减少运行时间。

### 4.3 数据处理规则设计

针对上面分析出来的这些原始数据的特征，在本系统的数据处理量化模块的设计时就需要更具不同的字段情况设计一种数据处理的规则，本节将对这数据规则的设计过程进行描述。

由于高血压风险预测系统对数据的处理要求比较严格，加之我们得到的原始数据纷杂繁复，而且有些数据还涉及不同字段之间的分类关系或者有些字段

的值需要进行转化才有实际意义等等各种复杂的情况，本模块针对这些数据特征结果精心的研究，最终决定将这些处理规则形式化为 xml 文件的形式。

本系统的数据处理规则主要是针对原始数据特征使用 xml 方式记录下字段处理时应该遵循的规则，而在处理数据是使用解析 xml 的方式获得这些规则，以此来处理相应数据，下面我就针对实际中遇到的几种情况描述下记录规则的方式，当然实际运用中可能还会遇到其他的一些情形，我们也支持规则修改的功能。

#### 4.3.1 汉字的量化

在数据处理过程中我们首先会遇到的就是汉字的量化问题，在我们获得的原始数据中有些字段采用的是汉字的表示方法，而如果这些字段要参与高血压风险预测决策中的话就需要将其量化成数字，如此才能用于判断，比如如图 4.5 所表示的情况。

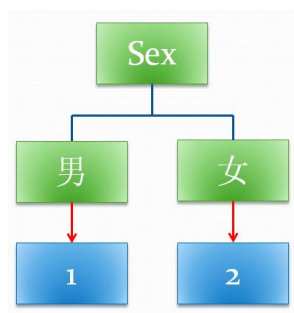


图 4.5 Sex 字段分析

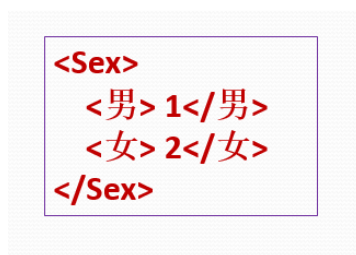


图 4.6 Sex 字段规则设计

典型的如 Sex 字段，该字段在原始数据中一般用汉字表示，而该字段对于

我们高血压风险的预测是不可或缺的，男性还是女性在高血压上的反应有很大的不同，这里我们就需要将其量化为数字。而我们最终形成的就如图 4.6 所表示的一样，我们用 xml 存储 Sex 节点，并在其内对不同汉字与数字进行对应，使用时只要根据字段名称以及其值就可以替换成相应的数字。

#### 4.3.2 连续值的分类量化

在原始获得的数据中很多字段的值都是连续值，而我们用于预测时并不能使用连续值进行预测，我们需要将其划分不断的数据段，比如高、中、低等，并用数字进行表示，比如下面这种情况，如图 4.7 所示。

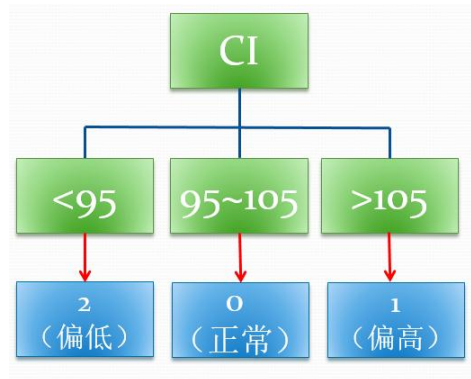


图 4.7 CI 字段分析

```

<CI>
  <lt value="95"> 2</lt>
  <bt value="95,105"> 0</bt>
  <gt value="105 "> 1</gt>
</CI>
  
```

图 4.8 CI 字段规则设计

CI 字段是一个连续性值得字段，字段的实际值可以为任意一个值，在对高血压风险预测是不行的，所以我们根据相关医学知识将 CI 字段划分成三种档次偏低、正常、偏高，并且为不同档次规定相对应的数字表示 2、0、1.最后形成如图 4.8 的 xml 表示形式。在数据处理过程中，程序根据字段名解析 xml 得到相



应 xml 节点，然后根据其值找到相应的子节点，进而获得其相对的数字表示进行替换。不同的字段有不同的分段方式，不一定就如此分，我们可以根据相应医学知识，这样对预测精确度的提示是很有利的。

4.3.3 关联其他字段的字段分类与量化

有些体检数据的结果与多个字段有所关联，同样的结果会因其他字段的  
不同而形成两种不同的情况，而这些不同我们需要区分出来，并且用不同的数  
字就行表示。由于有可能一个字段会依赖于多个字段，或者一个字段依赖的字  
段又依赖其他字段，实际中情况相当复杂。如图 4.9 所示的情形。

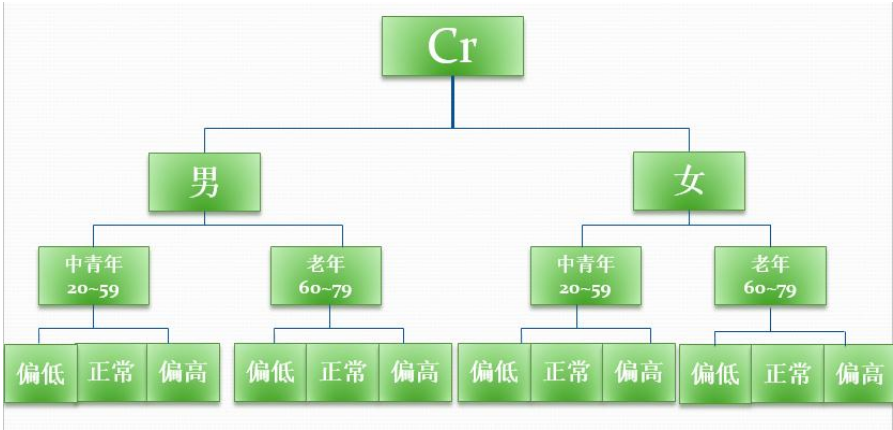


图 4.9 Cr 字段分析

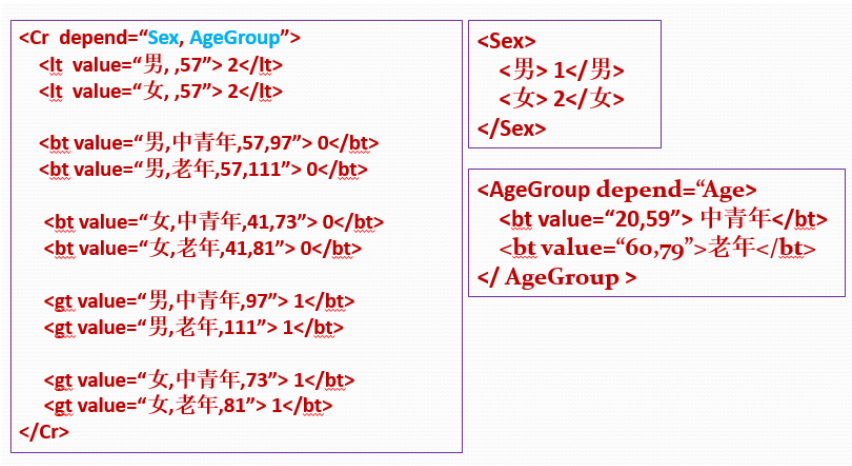


图 4.10 Cr 字段规则设计

从图 4.9 我们可以看出，Cr 字段结果可划分为偏高、正常、偏低三者结果，但是因为患者是男性还是女性以及患者所处的年龄段的不同，相同的 Cr 值有可能分为不同种情况，我们必须对这不同的情况进行不同的表示。如图 4.10 就是我们最终选择的表示方法，我们用属性 `depend` 来记录字段依赖的具体字段名，而在子节点的 `value` 属性里记录不同情况的临界值，如此在数据处理时我们只要根据字段名找出其依赖的字段依次获得其值，最后就能将结果准确的表示成数字显示。

实际遇到的情况还有很多，这里我就不一一赘述，只有根据相关医学知识将数据进行明确的分类、量化，才能使我们训练出来的模型拟合度越高，最终的预测结果越准确。

#### 4.4 数据处理模块总体流程设计

通过对数据处理规则的问题的分析和解决，我们就可以对数据处理量化模块的总体流程进行设计。在数据处理模块中可以对数据中的缺少值、需要量化分类的值进行处理，还能对我们需要的字段进行选择，从而让数据符合我们 TAP 平台中数据要求，使预测观察员能方便的使用这些数据进行训练模型以及预测结果等一系列操作。该模块工作流程如图 4.11 所示。

从图中可以看到，用户进入到数据处理界面后可进行导入数据，当数据导入后程序自动计算出数据的缺失率，用户可以详细的看到数据中各字段的缺失率情况，通过这些字段的缺失率和相应的字段选择规则，之后会对这些字段进行选择，选择出用户需要的字段进行处理，同时用户还可以对用户数据处理的规则进行修改，这些规则是从相应医学知识和经验中获得的，将这些规则具体化，以文件的形式保存下来，用户可以对这些规则进行添加和修改等一系列操作，该模块将使用这些规则将原始数据处理成符合用户和 TAP 平台的要求的格式化数据，最后用户还可以将处理好的数据导出为用户需要的格式，自此数据处理完毕，可以上传 TAP 平台，进入 TAP 平台的数据管理模块。

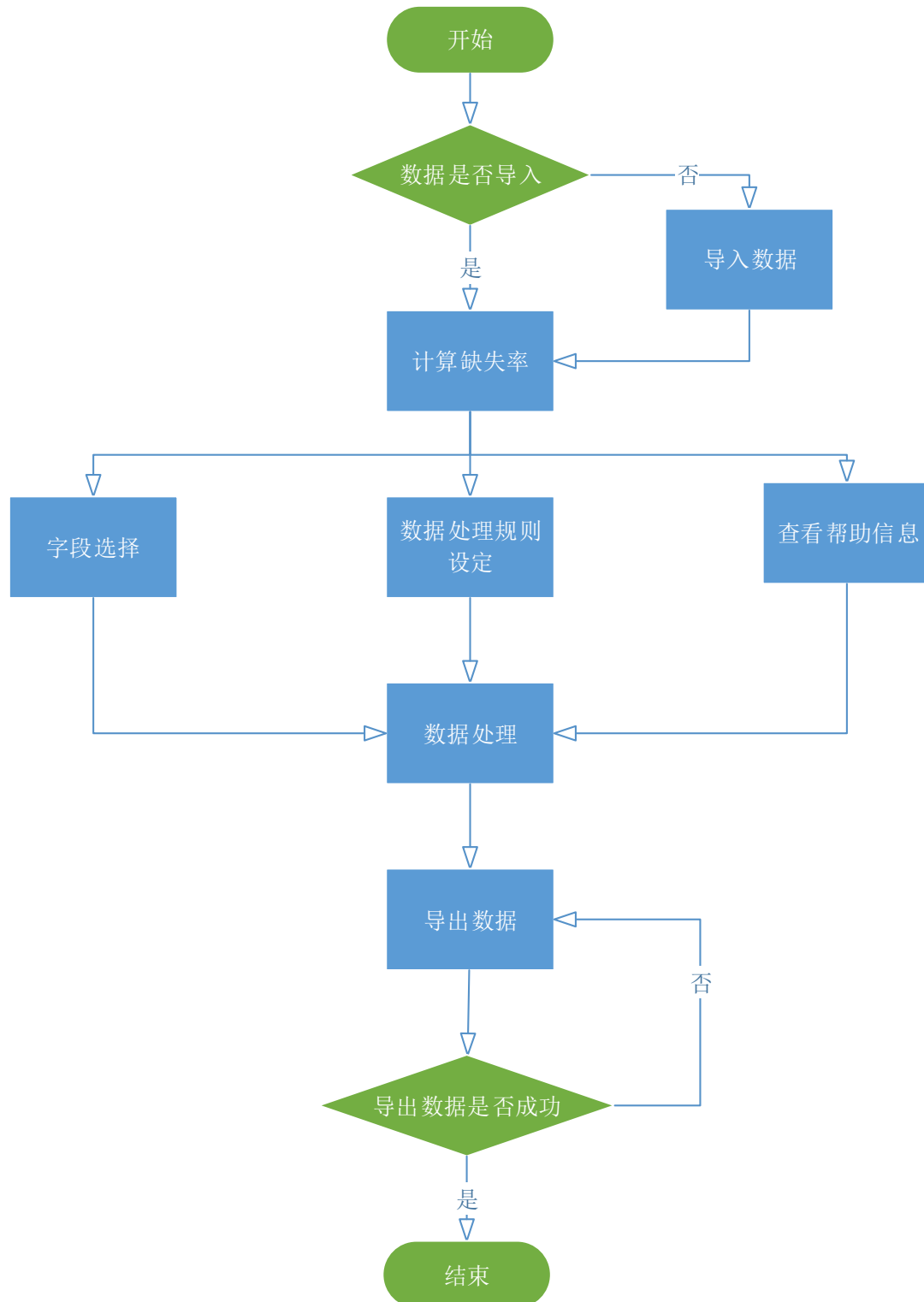


图 4.11 数据处理量化模块工作流程

## 4.5 本章小结

本章主要介绍高血压风险预测系统中核心模块数据处理量化模块的设计过程，通过对原始数据的特征进行分析，进而对相应的数据处理量化规则进行设计。在整个分析过程中形成数据处理量化模块的总体流程图。

本章对高血压风险预测其中的一个核心模块的设计过程进行详细描述，以数据为引，引出下一章对另一个核心模块高血压预测模块的设计过程。

## 第5章 高血压预测模块的设计

本章着重对这些数据模型进行对比，并结合本项目的预测数据的数据特征，从中选择出适合本系统的数据模型。并使用该数据模型，进一步对高血压预测模块的总体流程图进行设计。

### 5.1 引言

在本系统之中高血压预测模块是最核心最重要的一个模块，而在这一模块中数据模型的选择又是重中之重，在 TAP 平台中提供了多个数据模型供平台使用者进行选择，这些数据模型各有其优点，在没有形成自己独创的数据模型之前，本系统使用的是 TAP 平台中目前使用最为广泛的数据模型。这些模型无一不是专家在针对某一问题深入研究后得出的最优的成果，这些数据模型各有优点各有不同。

### 5.2 TAP 平台中数据模型优缺点对比

下面是 TAP 平台中提供的的几种主要的数据模型，根据每个数据模型的算法原理进行对比，其优缺点大致如表 5.1 所示。

表 5.1 国内外知名算法优缺点对比

名称	优点	缺点
分类树	直观，有可以理解的规则；计算量相对来说不大	有较高的方差，数据上的细小变动都会引起完全不同的分裂；预测面会影响回归的效果
主成分分析法	可消除评估指标之间的相关影响；可减少指标选择的工作量	主成分的累计贡献率必须达到一个较高的水平；对主成分必须能给出解释；主成分因子有正有负时，函数意义不太明确

续表 5.1

支持向量机	有坚实理论基础；最终决策函数只由少数的支持向量所确定,计算的复杂性取决于支持向量的数目	SVM 算法对大规模训练样本难以实施
K-Means 算法	简单，易于理解和实现；时间复杂度低	要手工输入分类数目，对初始值的设置很敏感；对噪声和离群值非常敏感；
随机森林	对大量的、高维度的数据进行训练时，很难出现过度拟合现象，速度快；对训练数据中的噪声和错误有很好的鲁棒性。	在某些噪音较大的分类或回归问题上会过拟；属性的数据取值划分较多的属性会对随机森林产生很大的影响

上面的表格中只列出了 TAP 平台中几种主要的数据模型算法，更多的没有列出，这些算法各有优点，各有劣势。而对于本系统来说不是每一个算法都适合，需要就每一个算法的特征以及本系统的数据特点进行适应性分析。

### 5.3 模型训练数据的特征分析

在选择数据模型之前，先明确下本系统用于训练模型和预测的数据所具有的特征，这些数据都是经过数据处理量化模块处理过后的数据，这些数据具有以下一些特点：

- 1) 字段较多，即维度高。在原始数据经过字段选择之后还具有很多字段，这些字段都将用于模型的训练和结果预测。
- 2) 数据密集，集中度高。因为这些数据都是人体检测数据，即使每个人的检测结果不同，都是值之间的差别不会很巨大。
- 3) 噪声小。数据都在正常值上下波动，某数值突变的情况很少。
- 4) 数据相关性低。数据是通过不同人的不同检测结果获得，相关性较低。
- 5) 数据量大。采集不同体检机构的体检数据，体检人数多，数据量大。

## 5.4 各算法模型对本系统适应性分析

结合各个算法模型的算法原理以及本系统所采用的数据的特征，可以对适合本系统的算法模型进行筛选，本节将对 TAP 平台提供的几个主要模型进行相对于本系统的适应性分析。

### 5.4.1 分类树模型适应性分析

分类树算法虽然直观，并且计算量不大，但是数据上一点点细小的变化都会引起完成不同的分裂，进而得出完全不同的结果，从这点出发，分类树算法就很不适合用于高血压风险预测系统的使用。

本系统高血压风险预测模块采用的数据来源于不同人的检测结果，虽然不同人的检测结果不可能一样，使用当前这批人检测到的数据进行模型训练和换一批不同的人检测到的数据进行模型的训练，本系统期望的结果是模型基本一致，而对分类树这一模型来说换一批数据模型训练的结果会和原先得数据模型完全不同，不同的数据模型预测的结果必然不同，这明显不符合本系统的预期，毕竟作为一个风险预测系统首先是要满足结果的可重现性，如果因为使用了其他数据进行模型训练而导致用户预测的结果都不相同，那么这个系统将会是不可信的，也就不会有任何用户会使用它。

### 5.4.2 主成分分析方法模型适应性分析

对于主成分分析方法，该算法的优势在于可以减少评估指标之间的相关性，减少指标选择的工作量，主成分方法通过将相关性大的多个指标通过正交变换等复杂变换操作变换成几个相关性很小的指标成分，但是必需要为这几个主要的指标成分找到符合实际意义的解释，而且当主成分符号有正负的时候，对结果的解释就很难说清楚很难使用户信服。

本系统所采用的数据其特点是相关性较低，并不需要将其再进行正交变换，而且本系统对于指标的选择是通过医学知识以及相关文献和经验非常精心挑选出来的，如果将其变换成主要的几个主要成分的话会损失原始数据的信息，减少预测的精确度，而且最重要的是对于每一个具有实际意义的字段进行变换成

几个主要的成分时将会很难为这些主要成分找到实际的意义，从这一点看，用于高血压风险预测的依据就不容易人使用户接受和信服。所以主成分分析方法也不适合本系统的使用。

#### 5.4.3 支持向量机模型适应性分析

支持向量机算法是非常优秀的一个算法，它有非常坚实的理论基础，该算法完全可以由线性分类函数  $y=ax+b$  出发，经过 **logistic** 函数，等等一系列的数学推演出其数学理论基础。但是该算法支持向量的数目会严重的影响到计算的复杂度，这既是其优点也是其重大缺陷。

高血压风险预测系统将会使用海量的数据对数据模型进行训练，以保证其模型的拟合性，进而保证其预测的准确性，但是因为本系统的支持向量数据多，鉴于支持向量机算法模型的特点，用该模型进行训练在这种情况下会产生极大的运算量，这无疑会拖慢整个系统的运行速度，给用户极其恶劣的用户体验，这是本系统绝不允许的，也是本系统在设计之初坚决杜绝的。

#### 5.4.4 K-Means 算法模型适应性分析

K-Means 算法随着大数据的兴起而广为流传，这种算法以其简单，易于理解而又非常少的时间复杂度而被大家所欢迎，在很多场合都将它作为核心算法。K-Means 算法通过事先输入的分类数目  $k$ ，初始选择  $k$  个中心，然后每次在这  $k$  个中心的基础上重新选择中心点，使周围的离散数据尽可能的靠近中心，最后将原始数据成聚合度极高的  $k$  个分类簇。

首先因为本系统所采用的数据有着数据密集，集中度高的特点，如果使用该算法进行模型训练的话，会很难进行  $k$  个聚类簇之间的划分。此外，该算法如果应用在高血压风险预测系统上，本系统设计人员首先就要确定出我们需要将数据分为多少个分类簇，而分的类数目又会对预测结果产生很大的影响，那么为了使系统达到最佳的预测准确性就需要设计人员反反复复的去尝试，如果还要考虑字段选择等其他的一些变化因素的话，这将会给系统的设计和开发带来极大的困难，即使是这样，每次训练数据模型之初选择的初始中心点也会对结果产生很大的影响，使结果的波动性很大，所以该算法也不太适合本系统的



模型训练。

#### 5.4.5 随机森林算法模型适应性分析

随机森林算法的优点是对大量的、高维度的数据进行训练时，很难出现过度拟合现象，而且速度快，而其缺点是在某些噪音较大的分类或回归问题上会过度拟合；对于有不同取值的属性的数据，取值划分较多的属性会对随机森林产生很大的影响。

考虑到本系统所采用的模型训练的数据的特点是数据量大，字段多，维度高以及噪声小，而且数据比较密集，值的划分比较少的特点。随机森林算法可以很好的发挥其优势，并且规避了其不利的方面，从这方面来说该算法比较符合本系统的要求。

在本系统没有开发出属于自己的专属数据模型的情况下，我们暂时选择了随机森林算法作为我们数据模型训练的核心算法，以后再针对具体的实际、医学知识、医师经验等开发出本高血压风险预测系统专有的算法模型。

### 5.5 高血压预测模块总体流程设计

在选择完成高血压预测模块的核心算法为随机森林算法模型之后，可以以此为核心对高血压风险预测的模块总体流程进行设计，该模块是高血压风险预测系统的核心环节，所有模块都是围绕着该模块而运行。该模块的工作流程如图 5.1 所示。

从图中可以看出，用户首先要创建一个运行终端，并且用生成的秘密登陆相应终端，然后在终端里先要生成连接服务器的密钥，之后的连接都使用该密钥进行连接，在连接服务器后上传我们实现编写完成的应用代码，运行，首先会创建一个 **Frame**，该 **Frame** 即使数据，之后用这些数据训练我们的预测高血压风险的数据模型，最后预测出结果，并且显示给用户。

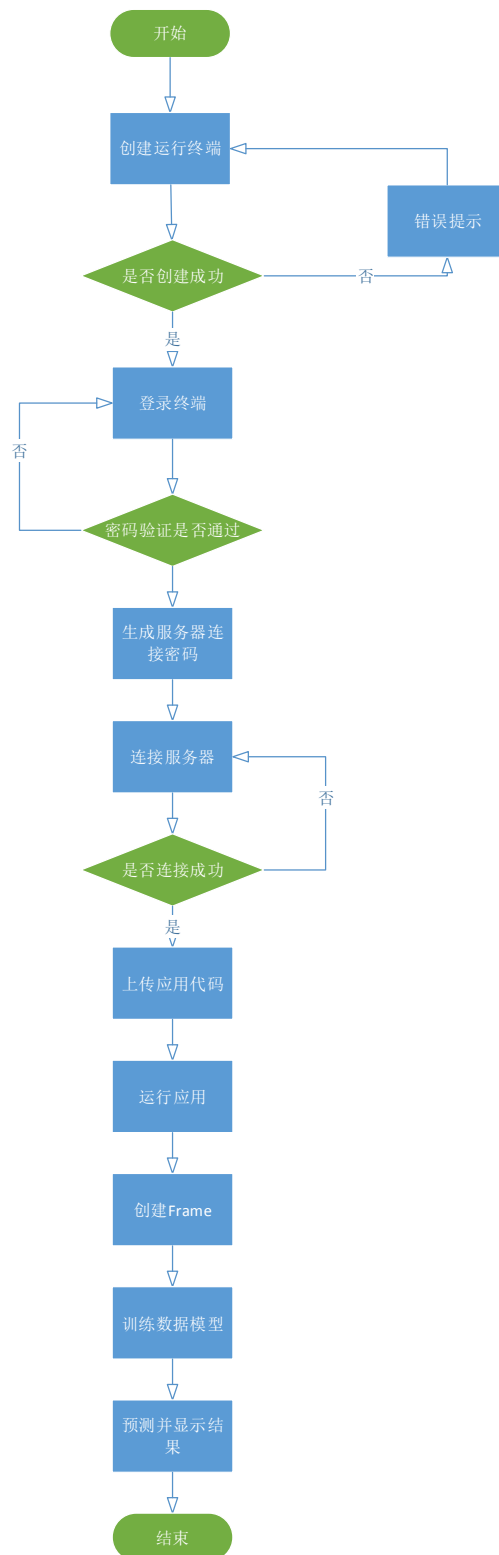


图 5.1 高血压预测模块工作流程

## 5.6 补充--随机森林算法原理简述

随机森林算法在对大量的、高维数据进行训练时，没有过拟合现象，而且速度较快，受训练数据中的噪声和错误影响不大等等一些优点。正是这些优点让我们最终选择了这个算法作为本系统的数据模型算法，基于高血压风险预测系统的实际情况，以及本系统数据的一些特点，随机森林算法满足我们的需求又避开我们的缺陷。作为高血压风险预测系统最重要的一部分，下面将对该算法的原理进行一些简单的介绍。

随机森林算法不仅仅可以用来分类，还可以用来做回归预测，是机器学习等一些领域应用最为广泛的一种算法，该算法是由一个个的决策树构成，但是相比于决策树算法，其在分类和预测方面又有很大的优势，很难出现决策树的过度拟合现象。

决策树作为随机森林的基础元素，在介绍随机森林算法之前先对决策树的原理进行一些简单的说明。决策树是一棵树，可以是二叉树也可以是非二叉树，它的每个非叶子节点都表示一个特征属性，节点上的分支代表特征属性在某个值域上的分类，所以每个叶节点就存放了一个类别。决策树的决策的过程是从根节点开始的，对待分类项中每个的特征属性进行测试，按照测试的值对输出的分支进行选择，一直到达叶子节点为止，叶子节点存放的就为决策的结果。决策树的构造过程大致如下：

1. 首先将所有记录看作是一个节点
2. 遍历每个变量的每种分割方式，找到最好的分割点
3. 利用分割点将记录分割成两个子结点 C1 和 C2
4. 对子结点 C1 和 C2 重复执行步骤 2)、3)，直到满足特定条件为止

从上面的过程可以看出，构建过程是一个递归的过程，构建好的决策树就是一个分类好的树，可以用于预测。

而随机森林算法就是由多个决策树构成的森林，算法最后的分类结果是由这些组成它的决策树投票得到的。在构建随机森林时采用的放回抽样，每构建一颗决策树都是从样本数据集中有放回的选择一个数据集进行决策树的构建的，除此之外，在选择决策树的非叶子节点也就是分割条件时是从字段上无放回的随机抽样而得到的特征子集并以此来构建的。

以上便是随机森林算法的基本原理，从原理可以得知随机森林算法是一个

组合模型，能够有效的防止过度拟合，正是适合本系统的一个模型算法。

## 5.7 本章小结

本章主要描述了高血压风险预测系统核心模块的设计，这是在系统概要设计的基础上做出的更详细的设计。在云计算领域用于分类聚合的算法如随机森林算法，支持向量机算法，主成分分析方等为代表的算法模型都更有所长，适合不同的应用场景，本系统鉴于本身的应用场景，从不同算法相应的算法原理出发，最终根据测试数据的预测准确最终选择了随机森林算法模型。

高血压风险预测系统的核心模块是数据处理模块和高血压预测模块，数据处理模块将来自不同地方的数据处理成 TAP 平台适合的数据形式，高血压预测模块使用这些数据将模型进行训练，从而拿来预测患者的高血压风险。

下一章将会对高血压风险预测系统这两个核心模块的具体实现过程进行描述，向大家展示该系统的具体工作原理。

## 第6章 数据处理量化模块程序的实现

本章主要描述了高血压风险预测系统核心模块的数据处理量化模块的实现过程，通过详细的介绍实现该模块所使用的技术，以及该模块程序类和方法的设计实现过程，并使用类图、时序图等形象的展示整个模块的运行原理。

### 6.1 引言

数据处理模块程序的实现本系统使用的是 `c#` 语言，`c#`这种高级程序语言是微软公司发布的一种面向对象，运行在`.NET Framework` 上面的，一种简单的、稳定的、安全的有 `c` 和 `c++`衍生出来的一种面向对象的语言，`c#`和 `java` 一样都是面向对象的都一样有着自己的自动垃圾收集机制，但不同的是 `c#`有着 `java` 没有的高运行效率，而且 `c#`是面向组件的编程，这在开发程序界面时它的便捷性尤为突出，开发人员可以很简单的进行界面的设计和编程。

在我们高血压风险系统的数据处理模块中用来开发界面主要使用到了 `c#`的 `Form` 类，该类是 `c#`创建 `windows` 窗口时自动会生产的类，该类就是代表窗口的一个类，该类有关于在该窗口上组件的一些属性，还有一些对事件进行处理的函数。比如在数据处理模块程序中本系统使用到的关于 `Form` 类的方法如表 6.1 所示。

表 6.1 `Form` 类部分方法

	名称	说明
	<code>Activate()</code>	激活窗体并给予它焦点。
	<code>AdjustFormScrollbars(Boolean)</code>	根据当前控件位置和当前所选控件调整容器中的滚动条。
	<code>Close()</code>	关闭窗体。
	<code>Dispose()</code>	对 <code>Component</code> 使用的所有资源进行释放。

续表 6.1







	Focus ()	设置控件的输入焦点。
	OnActivated (EventArgs)	引发 Activated 事件。
	OnClick (EventArgs)	引发 Click 事件。
	OnEnter (EventArgs)	引发 Enter 事件。
	Select ()	激活控件。
	Show ()	向用户显示控件。

表 6.1 只列出了我数据处理模块使用到的部分 Form 类的方法，除了这些方法还使用到其他一些我们自己定义的方法。通过这些方法我们对窗口上发生的一些时间进行反应，从而实现程序与用户的互动。在数据处理程序中有这样的一些事件程序会对其进行处理，下面列出部分 Form 类上的一些事件。

表 6.2 Form 类部分事件

	名称	说明
	Activated	激活窗体并给予它焦点。
	Click	在单击控件时发生。
	Closed	关闭窗体时发生。
	CursorChanged	发生在 Cursor 属性的值有更改时。
	Deactivate	当窗体失去焦点并不再是活动窗体时发生。
	DoubleClick	发生在双击控件时。
	FormClosed	关闭窗体后发生。
	HandleCreated	在为控件创建句柄时发生。
	HelpButtonClicked	单击“帮助”按钮时发生。

## 6.2 数据处理量化模块程序的实现

通过上面介绍的 Form 类窗口对事件的处理以及一些方法的使用，再加上一些我们自己编写的业务处理类，这就构成了我们数据处理模块的整个程序，最终形成如图 6.1 所示的数据处理模块的主页面。

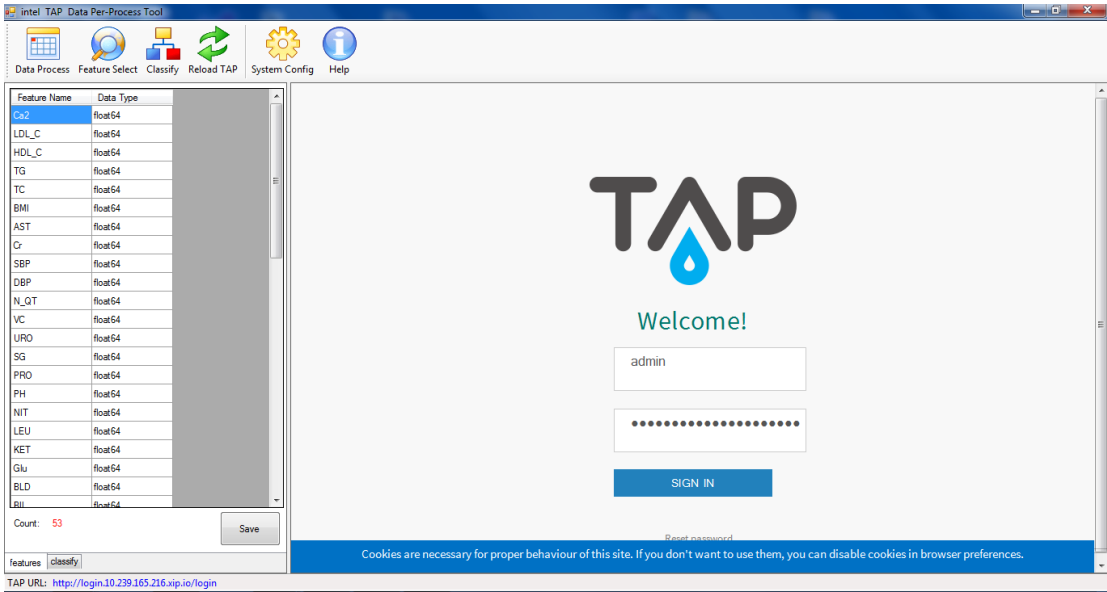


图 6.1 数据处理量化模块主页面

在上面的数据处理模块主页中左上角的 **Data Process** 是数据处理程序入口，点击将进入具体数据处理的窗口中进行详细的处理过程，接下来的 **Feature Select** 是用于字段选择具体处理窗口的入口，点击进入的字段选择窗口将用于用户进行字段选择的具体操作，在原始采集到的数据中包含着很多字段，而并不是所有的字段我们都需要，如果不先将这些不需要的字段进行数据处理，在本系统数据量比较大的情况下将产生多余的性能消耗，所以在 **Feature Select** 中用户可以通过设置选择出本系统需要的一些字段，而数据处理程序将根据这些字段详细的对这些字段的值进行处理。

除了 **Data Process** 和 **Feature Select**，还有 **Classify** 和 **Reload TAP** 这两个比较重要的组件，**Classify** 是数据处理程序的核心，在 **Classify** 弹出的窗口中是对数据处理规则的具体设定，其中存储的是本系统以 xml 形式存储的对数据处理的具体规则。而 **Reload TAP** 则是在主界面的右下角中打开 TAP 平台的页面，数据

处理完成之后可以通过页面将数据上传到 TAP 平台的 hdfs 之中。

除了上面介绍的一些比较重要的组件在主页面上还可以 System Config 系统设置组件和 Help 帮助组件，值得一提的是在主界面左下角的地方还有显示字段名称、属性以及填充方式的窗口，其中数据填充方式中 avg 代表均值填充即以平均值作为填充空白字段的方式，median 代表中位数填充，mode 代表统计频率填充即以列出该字段中出现频率最高的数值并以该值填充到空白字段之中，如果填充方式是 no 的话代表关闭本列字段的填补，一般很少使用，这些填补方式是可以通过窗口输入进行修改的。

要产生如上的界面以及其内部的具体处理程序，我们进行了大量的代码编写工作，最后数据处理程序的项目结构如图 6.2 所示。

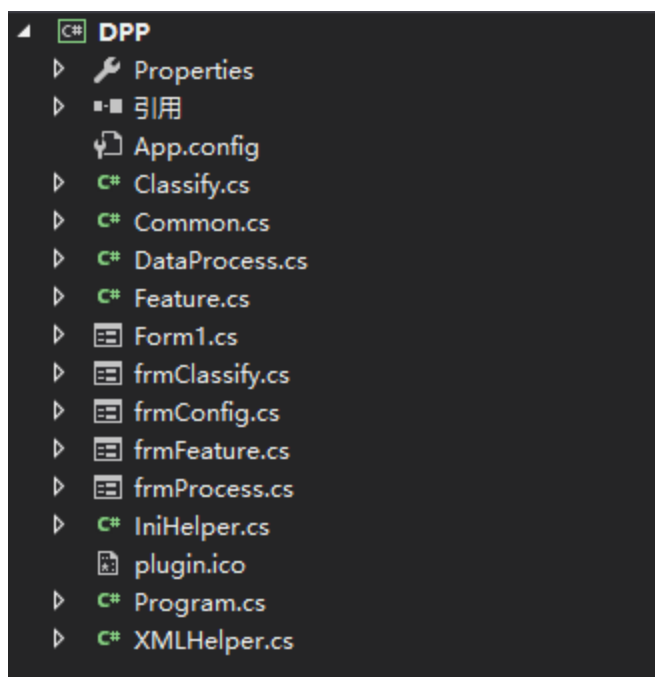


图 6.2 数据处理量化模块项目结构

### 6.2.1 数据处理量化模型程序相关类的设计与实现

从项目的结构上看本数据处理模块程序有 Form1、frmClassify、frmFeature、frmProcess 四个窗口类，除了这四个窗口类还有 Classify 类用于对处理规则的实现，DataProcess 类用于对数据处理的具体实现，Feature 类用于对字段列进行处



理的实现，Common 类是对一些排序、求平均、求频率以及读写文件的公共操作进行实现的一个类，而 XMLHelper 类是数据处理程序对 xml 进行解析的一个帮助类。

这些类之间的关系等情况用图 6.3 所示的类图进行展示。

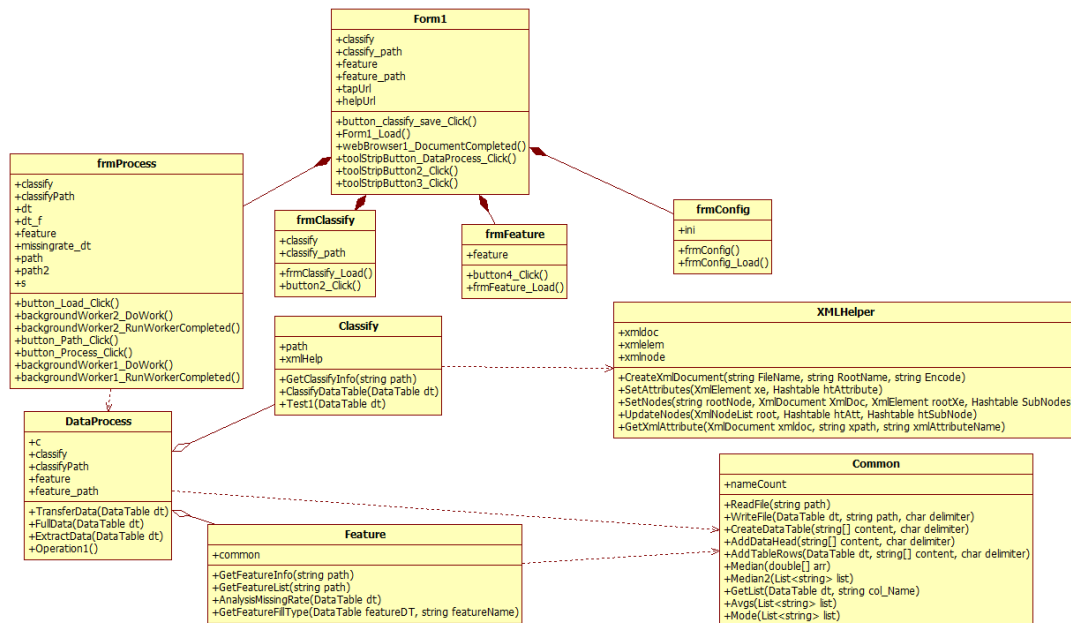


图 6.3 数据处理模块类

图 6.3 清晰的展示的是数据处理模块程序主要的几个类，图中类与类之间的关系一目了然，Form1 类和 frmProcess 类、frmClassify 类、frmFeature 类和 frmConfig 类是组合关系，其组合关系表现为 Form1 类代表的主页面窗口是由其它几个类所代表的组件所组成。而 frmProcess 类和 DataProcess 类是依赖关系，是使用的关系，组件类调用具体的实现类去实现相应的功能。其中 XMLHelper 类只和 Classify 类为依赖关系而和其他类没有任何关系，主要是因为只有在 Classify 类才需要对 xml 进行解析从而使用到 XMLHelper 类，和 XMLHelper 类不同，Common 类就分别和 DataProcess 类和 Feature 类是依赖关系，因为 DataProcess 类和 Feature 类都使用到了 Common 类的某些方法。在所有类里面 DataProcess 类的关系最多，它不仅和 Classify 类和 Feature 类有聚合关系，而且还和 Common 类有依赖关系。DataProcess 类是数据处理模块的核心模块，数据

处理的主体功能都是在这个类里面实现，而其他类则都被其所用，如此才能完成数据处理的任务。

下面将通过对每个类进行详细的介绍来展示数据处理模块程序是怎么实现其功能的。

### ● Form1 类

Form1 类是数据处理模块主界面的类，该类有着 `classify`、`classify_path`、`feature`、`feature_path`、`tapUrl` 和 `helpUrl` 这样一些属性，这些属性代表着 Form1 类可以打开界面上所有的一些组件，比如展示选择过的字段 `feature` 和进行过处理的数据处理规则 `classify`，出错之外还可以通过 `tapUrl` 和 `helpUrl` 分别打开 TAP 平台页面和帮助页。

而该类拥有的如 `toolStripButton_DataProcess_Click()`、`toolStripButton2_Click()`、`webBrowser1_DocumentCompleted()` 等的这些方法是对在窗口是按钮进行点击事件的具体执行操作，比如打来 `feature` 选择界面，打开数据规则编写界面等等。

### ● frmProcess 类

frmProcess 类是数据处理窗口的类，数据处理的具体过程是在该窗口内发生，就如图 6.4 所示，图 6.4 就是数据处理的窗口。



图 6.4 数据处理窗口

frmProcess 类就是上面这个窗口的类，该类有着 `classify`、`feature`、`dt`、`path`

等很多的属性，这么多的属性也表明类该类是该模块最重要的一个类，该类有着 `backgroundWorker2_DoWork()`、`button_Process_Click()`、`backgroundWorker1_DoWork()`等等一些方法，这些都是用来处理窗口上按钮组件发生点击事件，通过按钮的点击去出发相应的数据处理工作，而在这些方法中又通过使用其他的比如 `DataProcess` 类这样的具体功能实现类来完成其任务。

● `frmClassify` 类

`frmClassify` 类是选择字段 `feature` 的窗口的类，对要使用哪些 `feature` 来进行处理进行选择，在该窗口内可以对这些 `feature` 进行增加、删除，修改的操作，就像图 6.5 展示的这样。

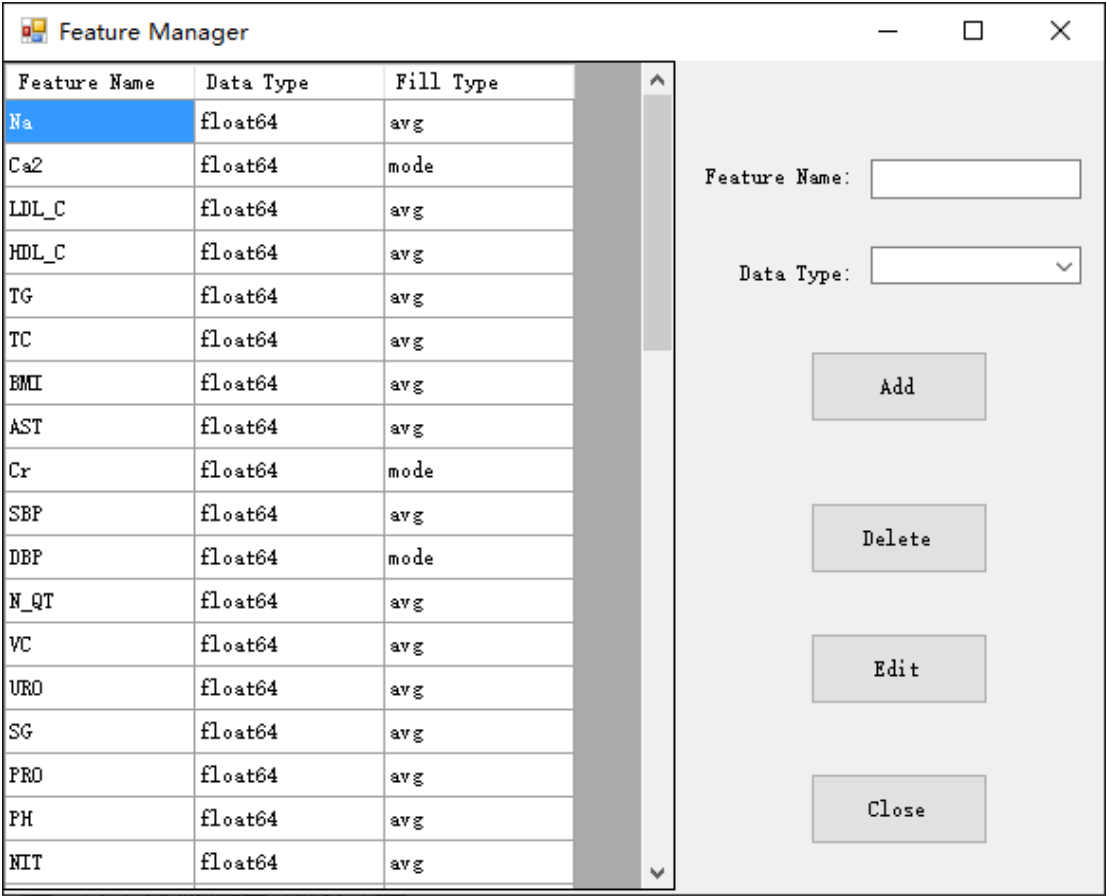


图 6.5 feature 选择窗口

`frmClassify` 类只有 `feature` 属性和 `frmFeature_Load()`还有 `button4_Click()`等这

几个方法，通过这些属性和方法对 feature 进行增加、删除、修改的处理。

### ● frmClassify 类

frmClassify 类是对数据处理规则进行设计的窗口的类，在上一章的高血压风险预测系统的设计中我们也提到了，本系统的数据处理规则是通过 xml 存储的，将相应的数据处理规则保存在 xml 之中，在处理数据是通过解析对应 xml 文件来告知程序该安装何种处理规则处理，而本类就是这样一个 xml 处理规则添加和修改的类，该类所代表的窗口如图 6.6 所示。

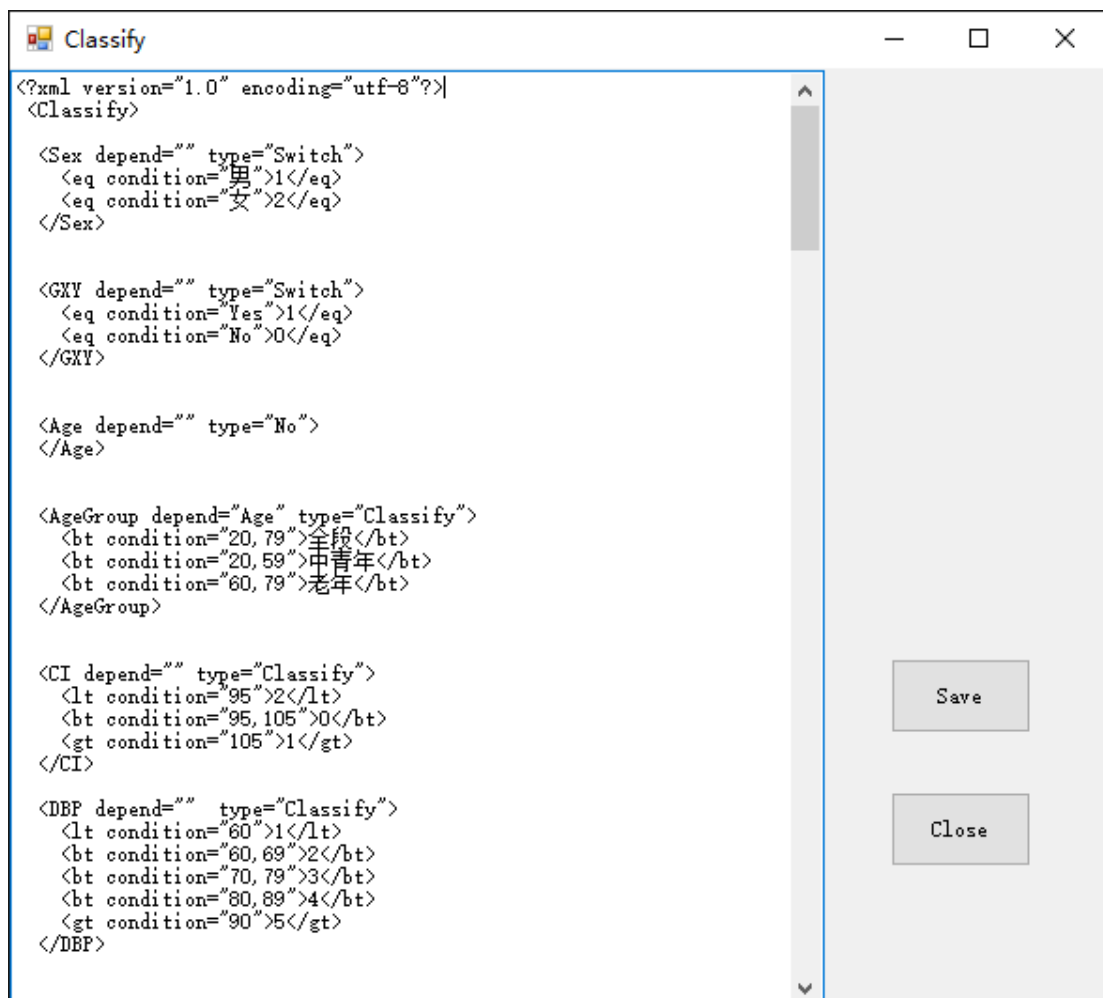


图 6.6 数据处理规则编写窗口

图 5.6 所展示的这样，frmClassify 类比较简单，只有 classify、classify\_path 这两个属性和 frmClassify\_Load()、button2\_Click()这两个方法，通过在窗口坐标

的 xml 里进行修改然后点击右边的“Save”按钮触发 button2\_Click()方法来进行规则的保存等操作。

- DataProcess 类

DataProcess 类有着 c、classify、classifyPath、feature、feature\_path 这些属性，还有 TransferData(DataTable dt)、FullData(DataTable dt)、ExtractData(DataTable dt)、Operation1()这样的一些方法，该类通过这些方法和属性所代表的 Classify 类和 Feature 类为数据处理的进行具体操作过程。

- Classify 类

Classify 类主要是为 DataProcess 类提供一些数据处理规则的一些信息，该类通过解析数据处理规则的 xml 文件来获知某一字段值改对应何值，一次来为数据处理提供可靠依据，该类有 path、xmlHelp 这样一些属性和 GetClassifyInfo(string path)、ClassifyDataTable(DataTable dt)这些方法。从这些属性和方法中就可以了解到该类的主要功能作用。

- Feature 类

Feature 类是用来获取实现选择好的那些 feature 字段信息的一个类，该类的 GetFeatureInfo()、GetFeatureList()、GetFeatureFillType(DataTable featureDT, string featureName)和 AnalysisMissingRate(DataTable dt)这些方法就是原来获取这些 feature 信息的。

- Common 类

Common 类有些很多公用的一些操作，比如 Avgs(List<string> list)求平均、Median(double[] arr)求中位数、Mode(List<string> list)求众数等等一些比较常用的操作，以供 DataProcess 类、Feature 类或者其他类的使用。

- XMLHelper 类

XMLHelper 类是对 xml 进行解析的一个类，该类可以用来读取 xml 节点属性值，或者设置和更新一些节点属性值的一些操作。

### 6.2.2 数据处理量化模块程序运行过程实现

在介绍过数据处理模块程序主要的一些类及其它它们之间的关系之后，下面来介绍一下数据处理模块的具体处理过程，数据处理模块程序的时序图如下图所示。

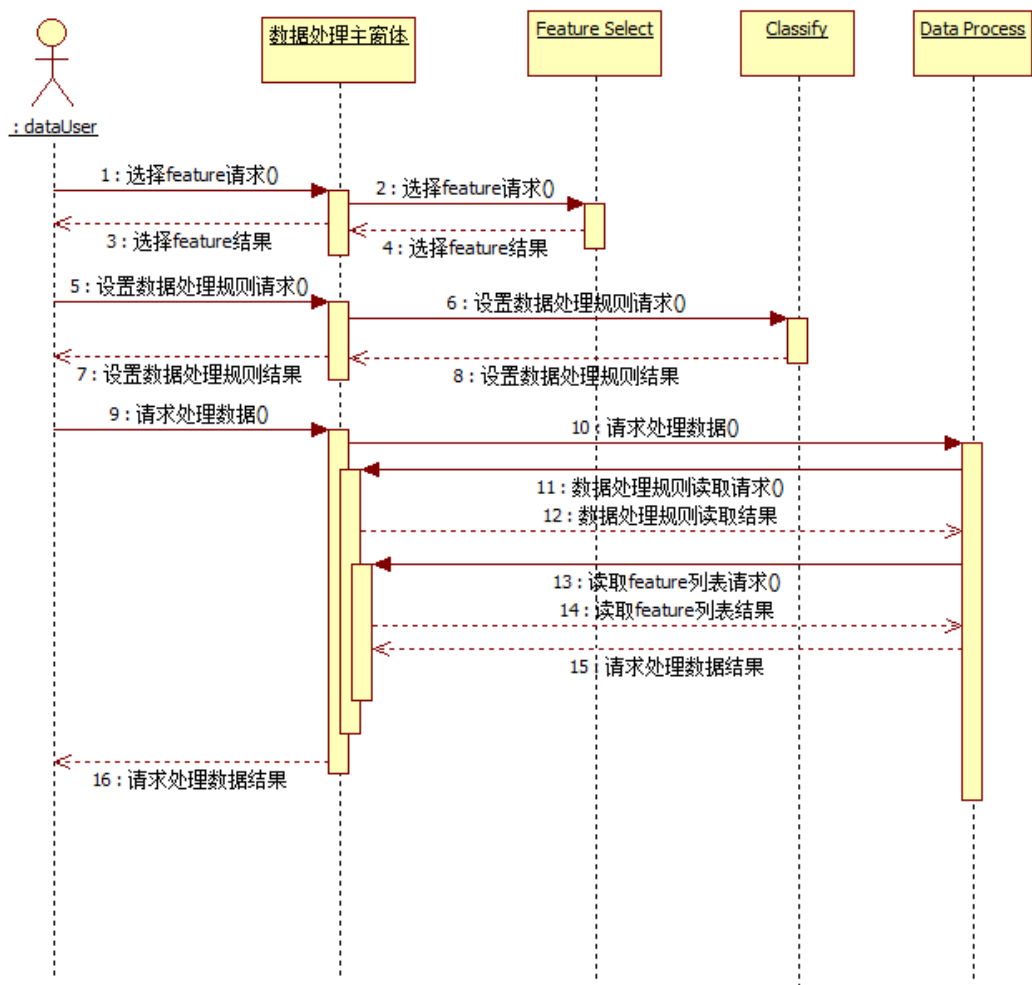


图 6.7 数据处理模块时序

在图 6.7 的时序图中描述的是数据处理的具体过程，首先用户要通过数据处理主窗体向选择 feature 窗体进行 feature 的选取这一请求，在完成 feature 选择设置完毕之后，用户还要通过数据处理主窗体向 Classify 数据处理规则发出设置处理规则的请求，在 Classify 相应这一请求，在用户设置好处理规则之后才算数据处理的前期工作准备完成，

在准备完成之后用户发出数据处理请求，主窗体在接到这一请求之后想数据处理程序发出请求，而 Data Proess 在接到这一请求之后要读取数据处理规则以及之前用户设置好的 feature 列表，只有在这两个条件下才进行数据的处理，

在 Data Process 结果数据填充，数据转换，等等一系列操作之后将原始的杂乱的数据处理成格式统一的数据，从而满足用户的请求。

至此，数据处理模块的具体实现介绍完毕。路过数据处理模块处理之后的数据是格式满足 TAP 以及项目要求的数据，且数据内部的字段值都已经转换成对数据模型训练以及预测有意义的数值。

### 6.3 本章小结

本章主要介绍高血压风险预测系统核心模块之一的数据处理量化模块的实现过程，通过对模块程序中事件处理，类的描述，以及相应方法的介绍将数据处理量化模块的实现原理展现出来，并结合类图，时序图等工程化图标，让程序的运行过程一目了然。

在本章介绍完成之后将进入下一章对高血压风险预测系统另一个核心模块高血压预测模块的实现的介绍。

## 第7章 高血压预测模块的实现

本章着重介绍高血压预测模块的实现过程。通过对 TAP 平台的搭建以及相关接口的描述，实现高血压风险预测模块连接 TAP 平台，创建 Frame，训练模型，结果预测等一系列功能的实现。

### 7.1 引言

高血压预测模块是基于 TAP 平台而开发的，该模块的数据模型取自于 TAP 平台，模型的训练和预测也是通过调用 TAP 平台的 API 实现其相应的功能，通过该模块的运行也依赖于 TAP 平台，高血压预测模块与 TAP 平台密不可分。

### 7.2 TAP 平台的搭建与运行

本论文高血压风险预测程序是基于云计算平台 TAP 平台而设计的，其实现的开发环境主要包括硬件环境和软件环境。

#### 7.2.1 硬件环境准备

高血压风险预测系统的硬件开发环境主要包括：开发主机一台，服务器五台，以及其他网络设备若干，详细清单参见表 7.1。

表 7.1 高血压风险预测系统硬件开发环境

序号	硬件	数量
1	台式主机	1 台
2	服务器	5 台
3	液晶电视显示器	2 台



续表 7.1

4	RJ45 网线	数根
5	鼠标	1 个
6	键盘	1 个
7	交换机	1 个

### 7.2.2 软件环境准备

高血压风险预测系统的软件开发环境主要包括：开发主机 Ubuntu14.04 LTS 的 64 位操作系统，服务器 CentOS 6.7 的 64 位操作系统，以及其他集成开发和调试工具。详细开发调试工具清单参看表 7.2 所示。

表 7.2 高血压风险预测系统软件开发环境

序号	软件	说明
1	主机操作系统	Ubuntu14.04 LTS (64 位)
2	服务器操作系统	CentOS 6.7 (64 位)
3	Pycharm	Python 集成开发环境
4	visual studio	C#集成开发环境
5	putty	串行接口连接软件
6	Git	代码管理工具

### 7.2.3 TAP 平台的部署和运行

TAP 平台既可以部署在 AWS 上也可以部署在 Openstack 上，在本次 TAP 平台的部署中鉴于成本及性能等多方面因素，我们将 TAP 平台部署在 Openstack 之上。

- 安装 Openstack

首先为事先准备的 5 台服务器安装好系统，这里我们按要求安装的是 CentOS 6.7（64 位）的操作系统。用网线以及交换机将 5 台服务器连接，使其能够互相连通，为每台机器分配可用的 ip 地址。选出其中一台作为 Controller node，其他四台作为 Storage node，按照官方文档安装配置好 Openstack，这里因与主题相关性不大，我们略过。

### ● 创建 Stack

首先为创建 Stack 下载 heat 模板，这里我们下载 TAP 平台的 TAP-FullVM.yaml 文件用来以完整的虚拟机类型的方式安装。

接下来登录进 Openstack 的 Horizon web 界面，创建一个新的 Openstack 项目，并且为项目设置 Volumes, Vol Snapshots, Total size of Vols 和 Security Groups 这些。创建新的 Openstack 用户，并且为其分配管理员权限，之后登出本用户，重新以刚才创建的用户登录 Horizon。

进入我们之前创建的项目页面，Import 进 ssh 密钥对，然后分配并且记录下 Floating IP，同时记录下 API 地址。

做完这些之后就可以 Launch 起 Stack 了，之后还要做一点点配置，比如提供一个模板文件，减少 timeout 时间为 300 分钟，设置公共 ip 地址等。

### ● 部署平台

当上面的 Stack 起来之后，用 SSH 命令登录进 Jump Box，登陆命令如下：

```
“ssh ubuntu@<jumpbox_server_ip> -i <ssh_key.pem>”
```

登录进入后直接运行 TAP 平台的安装 shell 脚本，在禁用 Kerberos 认证的情况下运行脚本的命令是

```
“sudo -i curl -Sso tqd.sh https://s3.amazonaws.com/trustedanalytics/tqd.sh &&
sudo -i bash tqd.sh”
```

在不禁用 Kerberos 认证的情况下运行

```
“sudo -i curl -Sso tqd.sh https://s3.amazonaws.com/trustedanalytics/tqd.sh &&
sudo -i KERBEROS_ENABLED=True bash tqd.sh”
```

我们选择的是启用 Kerberos 认证的方式，整个部署过程大约要花费 2 到 5 个小时。当安装脚本在没有任何错误的情况下运行结束后，就可以通过 https://console.DOMAIN\_NAME\_YOU\_CHOSE 和我们之前在 Openstack 项目上设置的用户名和密码登录进 TAP 平台了。

至此 TAP 平台部署完毕。可以在该平台上进行应用开发了。

### 7.3 高血压风险预测应用程序的实现

在搭建并成功运行 TAP 平台时候，我们使用数据处理量化模块处理过的字段值符合实际要求且格式符合 TAP 平台要求的数据，将这些数据通过数据管理模块上传到 TAP 平台的 hdfs 之中，并记录下数据的 hdfs url，这个 url 将用于之后高血压风险预测模块模型训练和预测的数据查找。在图 7.1 所示的界面上传我们处理过的数据。

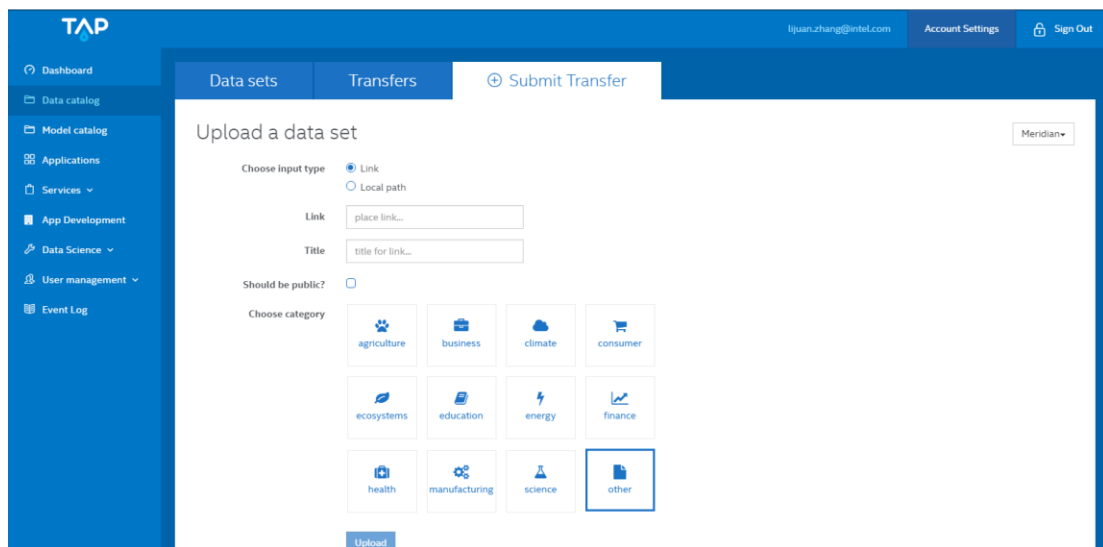


图 7.1 数据上传界面

如图 7.1 所示通过 Data catalog 的 Submit Transfer 进行数据的上传，在数据上传到 TAP 平台之后，并不是马上就可以运行高血压风险预测模块进行模型训练、风险预测，而是需要先创建属于我们的运行实例之后才能在运行实例上运行我们的应用。运行实例 instance 是我们运行高血压风险预测应用的载体，运行实例的创建过程极其简单，只要在如图 7.2 的界面里输入要创建的 instance 名称，点击之后的创建按钮就可以创建过程。

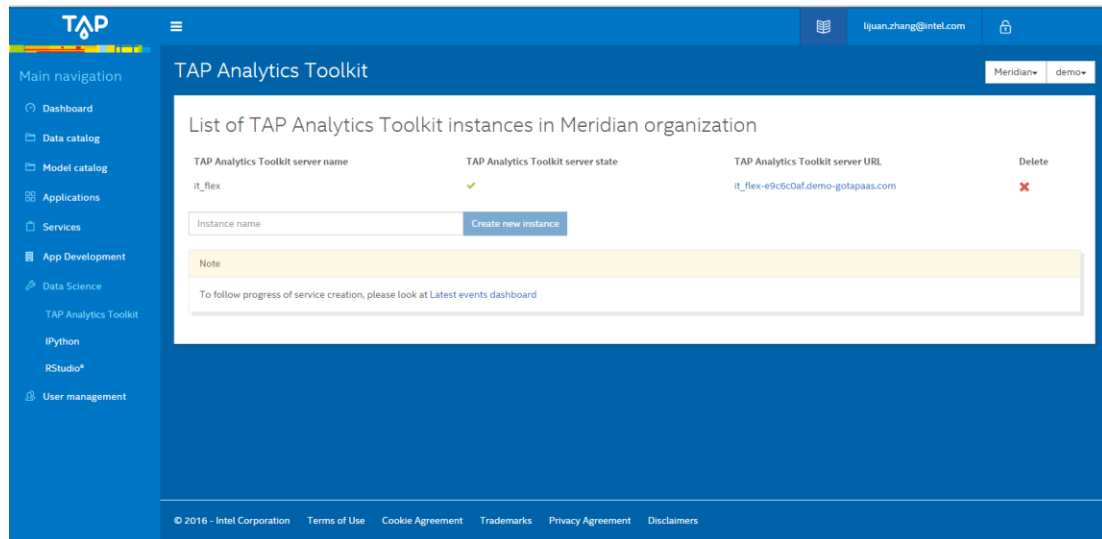


图 7.2 运行实例创建界面

创建运行实例的时候需要稍微等待几分钟，因为需要等待 TAP 平台分配好该运行实例的内存等的资源。在创建成功后的运行实例后面会有一个 server url，这个 url 将用于高血压风险预测模块进行 TAP 平台的连接。

鉴于数据管理模块和运行实例管理模块都同属于 TAP 平台的一部分，而且比较简单，这里我们介绍的比较少，而高血压风险预测模块是本系统的核心模块，下面详细的进行介绍。

### 7.3.1 高血压预测模块交互过程

在本模块中，功能的实现依赖于 TAP 平台提供的 api，模块程序通过调用 api 来完成功能的实现，其交互过程如图 7.3 所示。

程序通过调用 api 和 TAP 平台进行交互，从而让 TAP 平台按照程序执行相应的任务，在执行完成后将结果返回给模块程序，最后显示出来。

整个过程就是一个不断调用 api，不断执行任务的过程。

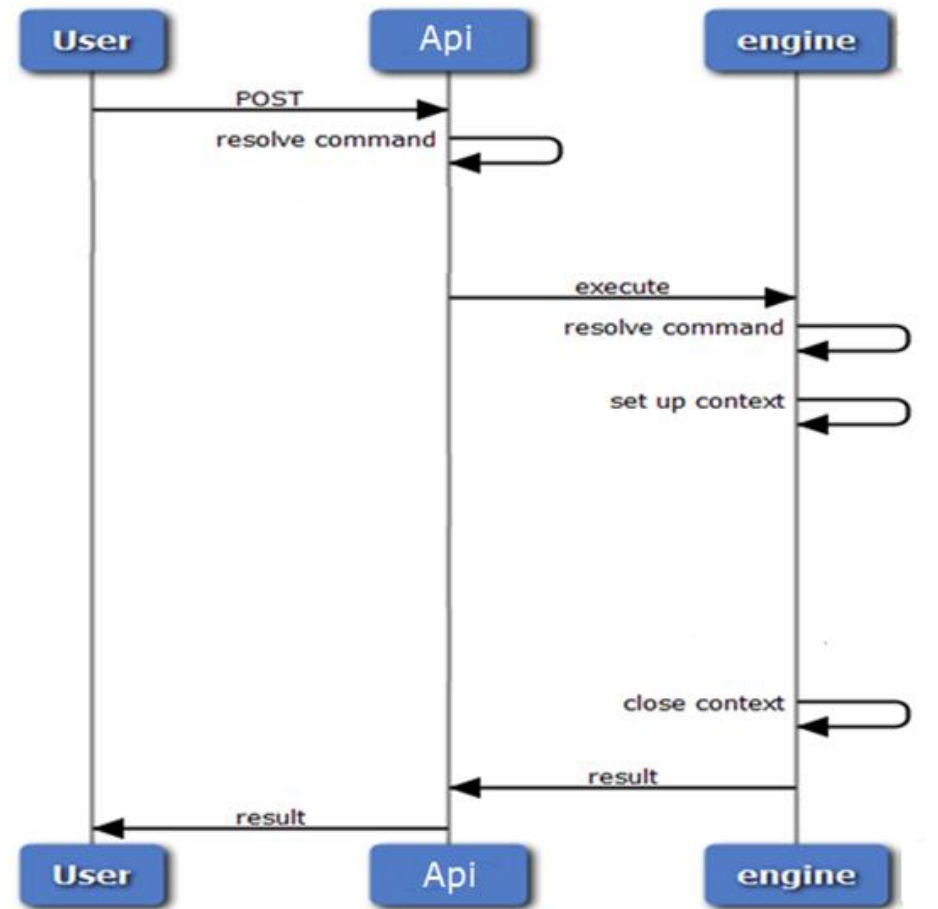


图 7.3 高血压预测模块交互过程

### 7.3.2 连接 TAP 平台

TAP 平台提供了很多 api 给开发者进行使用, 不仅提供了 Python API 而且还提供了 REST API, 本高血压风险预测模块程序使用的是 python 进行编写的, 所以我们只使用了 Python API。

- API

```
trustedanalytics.connect(self, credentials_file=None)
```

该 api 用于连接 TAP 平台的 server, 通过这个方法调用 server, 下载 API 信息, 并且动态的应用一些必要的 python 包, 在每次调用 server 是运行该 api 是必要的。credentials\_file 参数是连接 TAP server 必要的证书文件地址。

### ● 连接

在使用上面的 api 之前需要先在本地创建连接 TAP server 的证书文件，通过下面的命令进行证书文件的创建。

```
$ python2.7

>>> import trustedanalytics as ta
>>> ta.create_connect_file('~/.ta/demo.creds')
OAuth server URI: uaa.my-tap-domain.com
user name: dscientist9
Password: *****

Credentials created at '/home/dscientist9/.ta/demo.creds'
```

图 7.4 创建证书文件

在创建完成证书文件后就可以使用上面的 api 连接到 server 了，

- `import trustedanalytics as ta`
- `ta.server.uri = "atk-bcc31ff0-b3c2-4059-9749.10.239.165.216.xip.io"`
- `ta.connect()`

上面的三个命令完成了连接 TAP server 的操作，第二个命令制定的是我们在 TAP 平台创建的运行实例的 url。

### 7.3.3 增加数据源

TAP 平台对多种数据源提供支持，CsvFile、HiveQuery、HBaseTable、JdbcTable、JsonFile、LineFile、Pandas、UploadRows、XmlFile 等不同源的数据 TAP 平台都能很好的将数据进行导入，而我们的数据是以 csv 格式存放在 hdfs 之中的，所以用的是 CsvFile。

### ● API

```
trustedanalytics.CsvFile(file_name, schema, delimiter=',', skip_header_lines=0)
```

*file\_name* 就是保存在 hdfs 里的数据文件，其值即为在上传到 TAP 平台时记录下来的 hdfs url，该文件保存了量化好的数据。*Schema* 是一个元组，该

元组是对要导入的数据的一个描述，以(name, type)的形式，那么为字段 feature 的名字，type 为 feature 的类型，TAP 平台支持的数据类型包括 datetime\float32\float64\int32\int64\unicode\vector 其中。skip\_header\_lines 是设置跳过数据的前几行，比如 skip\_header\_lines=1 跳过首行，即一般为字段名行。

#### ● 导入数据源

导入数据源的操作比较简单，只需要一条命令就可以完成。

```
➤ csv=ta.CsvFile("hdfs://nameservice1/org/intel/hdfsbroker/userspace/275ed74
a-df39-4ff0-b3b9-049acccf904e/1178fa5c-6d05-47ca-9ed7-488786b21934/0
00000_1", schema=[("GXY",ta.int32), ("Age",ta.int32),("Sex",ta.int32)],
skip_header_lines=1)
```

在这一条命令中这里省略了 schema 的其他一些元组，schema 中的元组就是我们将用来进行数据模型训练和预测的 feature。而 hdfs url 也在之前有过记录。同时可以同时导入多个数据源，一个用来数据模型训练，一个用来测试数据模型准确性，一个用来预测。

### 7.3.4 创建并操作 Frames

在上一步将数据源导入成功后，接下来就是使用这些数据创建 Frames 并且对这些 Frames 进行操作。Frame 就是一个存储数据的一个大的表格，在 python 的高级数据处理中也有相同的 Frame 概念，使用 Frame 可以方便对数据进行处理。TAP 平台的 Frame 包含了大量的对数据的操作方法。

#### ● API

Frame( source=None, name=None)	创建 Frame
add_columns(self,func,schema[, columns_accessed])	为 Frame 增加一列
append(self, data)	为 Frame 增加更多的数据
column_median(self,data_column[, weights_column])	计算该列的中位数
count(self, where)	计算符合条件的数目
drop_columns(self, columns)	删除列

还有很多 api 在这里不在一一列出，比如 **Frame** 还有排序方法，还有改名方法等等，这么多的 api 基本上满足本系统的开发。

#### ● 创建 Frame

本系统因为在之前的数据处理模块上已经将数据进行量化、格式化，所以在该步只需要创建 **Frame** 就可以了，并不需要对 **Frame** 进行操作。而在创建 **Frame** 之前要先为每个 **Frame** 进行命名，如果在本应用中有同名的 **Frame** 名字就需要删除旧的了。

```
➤ frame_name = 'Random_forest_SampleFrame'
➤ exist_frames = ta.get_frame_names()
➤ if frame_name in exist_frames:
    ta.drop_frames(frame_name)
```

上面几条命令是检测是否错在同名的 **Frame**，如果存在则删除旧的 **Frame**，防止冲突。

```
➤ frame = ta.Frame(csv, frame_name)
```

创建 **Frame** 的语句及其简单，只有一条语句，创建 **Frame** 时要指明数据源是什么，并同时为该 **Frame** 进行赋值。

### 7.3.5 模型训练及预测

该部分是高血压风险预测模块的重点，在该部分中完成对数据模型的训练工作，并可以使用测试数据或者实际数据进行预测。因为 **TAP** 平台为方便开发者使用，提供了很多经典数据模型的 api，开发人员可以通过这些 api 来选择相应的数据模型进行操作，比如就有 **PrincipalComponentsModel**、**KMeansModel**、**SvmModel**、**LogisticRegressionModel**、**NaiveBayesModel**、**RandomForestClassifierModel** 等等的一些数据模型，这些模型是平台已经帮我们实现过的，所以并不需要开发人员编写算法，非常的方便。除此之外 **TAP** 平台也允许开发人员自行添加自己的算法，这也为本系统以后开发并使用自己的数据模型提供可能性。这里因为本系统暂未开发出属于自己的专有算法模型，所以在结果详细的选择对比之后本系统选择的是随机森林算法模型。

下面从随机森林算法模型简单的 api 开始介绍。

#### ● API



<code>RandomForestClassifierModel(name=model_name)</code>	创建新的 model
<code>predict(self, frame[, observation_columns])</code>	进行数据预测
<code>test(self, frame, label_column[, observation_columns])</code>	用测试数据测试模型的准确性，最后返回准确率等统计信息
<code>train(self, frame, label_column, observation_columns[, num_classes, ...])</code>	对数据密码进行训练

TAP 平台将对数据模块的操作基础为什么几个 api，而对开发者隐藏起内部算法，简化了开发过程。

#### ● 模型训练及预测

首先要使用下面这一条语句来创建数据模型。

➤ `classifier = ta.RandomForestClassifierModel(name=model_name)`

创建模型成功后就可以使用数据对该模型进行训练，训练的命令也比较简单。

➤ `classifier.train(frame, 'GXY', ['Age', 'Sex', 'BMI', 'DBP', 'SBP', 'HCT', 'MCV'], num_classes=2)`

在这条命令中 `frame` 为之前创建的 `frame`，'GXY'即为本系统要进行预测的字段高血压字段，而在该字段之后的运用模型训练的其他字段，当预测是就是根据这些字段来预测'GXY'字段的值。

➤ `metrics=classifier.test(frame_predict, 'GXY', ['Age', 'Sex', 'BMI', 'DBP', 'SBP', 'Na'])`

上面的这条命令是用来测试该模型预测的准确性，在随机森林算法运用于本系统的预测结果比较好，准确率很高，结果页面如图 7.5 所示。

```

[100.00% Tasks retries:0 Time 0:00:04]
create frame
[api model_random_forest_classifier.py[247] global/get_frame_names get_frame_names
[api model_random_forest_classifier.py[243] frame/init (Frame:7fd3986c3d90, init [source='file_name': 'hdfs://nameservice1.org/ncel/hdfsbroker/user-space/275ed94a-df39-44ff-b3b8-042a60cf0
0e6/cf794949-89d6-10d1-2577712b/aa3/000000_1', 'delimiter': ',', 'skip_header_lines': 1, 'schema': [({'GXY', <type 'numpy.int32'>), ('Age', <type 'numpy.int32'>), ('Sex', <type 'numpy.int32'>),
('BMI', <type 'numpy.float64'>), ('SBP', <type 'numpy.float64'>), ('DBP', <type 'numpy.float64'>), ('HCT', <type 'numpy.float64'>), ('MCV', <type 'numpy.float64'>), ('RDW_SD', <type 'numpy.float64
'>), ('RDW_CV', <type 'numpy.float64'>), ('HGB', <type 'numpy.float64'>), ('MCH', <type 'numpy.float64'>), ('MCHC', <type 'numpy.float64'>), ('RBC', <type 'numpy.float64'>), ('WBC', <type 'numpy.fl
oat64'>), ('PLT', <type 'numpy.float64'>), ('MONO', <type 'numpy.float64'>), ('MONO2', <type 'numpy.float64'>), ('EO2', <type 'numpy.float64'>), ('BASO1', <type 'numpy.float64'>), ('NETI2', <ty
pe 'numpy.float64'>), ('MONO2', <type 'numpy.float64'>), ('EO2', <type 'numpy.float64'>), ('BASO2', <type 'numpy.float64'>), ('PLT', <type 'numpy.float64'>), ('PDW', <type 'numpy.float64'>), ('MPV',
<type 'numpy.float64'>), ('P_LCR', <type 'numpy.float64'>), ('PCT', <type 'numpy.float64'>), ('PH', <type 'numpy.float64'>), ('PRQ', <type 'numpy.float64'>), ('Glu', <type 'numpy.float64'>), ('KET
', <type 'numpy.float64'>), ('HDL', <type 'numpy.float64'>), ('SIL', <type 'numpy.float64'>), ('URO', <type 'numpy.float64'>), ('KET', <type 'numpy.float64'>), ('SD', <type 'numpy.float64'>), ('ES
', <type 'numpy.float64'>), ('H_QT', <type 'numpy.float64'>), ('VC', <type 'numpy.float64'>), ('FSG', <type 'numpy.float64'>), ('TC', <type 'numpy.float64'>), ('TG', <type 'numpy.float64'>), ('LDL
_C', <type 'numpy.float64'>), ('HDL_C', <type 'numpy.float64'>), ('ALT', <type 'numpy.float64'>), ('AST', <type 'numpy.float64'>), ('AST_ALT', <type 'numpy.float64'>), ('BUN', <type 'numpy.float64'>
), ('Cr', <type 'numpy.float64'>), ('Ca2', <type 'numpy.float64'>), ('Na', <type 'numpy.float64'>)]], name='Random_forest_Samplerframe_product', _info=None)
[100.00% Tasks retries:0 Time 0:00:04]
Initializing a RandomForestModel object
[api model_random_forest_classifier.py[279] global/get_model_names get_model_names
[api model_random_forest_classifier.py[282] global/drop_models drop_models(item='POCRandom_forest_SampleModel')
[api model_random_forest_classifier.py[284] model:random_forest_classifier/new (RandomForestClassifierModel:7fd3986c3d90, new(name='POCRandom_forest_SampleModel', _info=None)
[100.00% Tasks retries:0 Time 0:00:00]
Training the model on the frame
[api model_random_forest_classifier.py[293] model:random_forest_classifier/train (RandomForestClassifierModel:7fd3986c3d90, train(frame=Frame:7fd398763b50, label_column='GXY', observation_column
='Age', 'Sex', 'BMI', 'DBP', 'SBP', 'HCT', 'MCV', 'HGB_Sp', 'RDW_CV', 'HGB', 'MCH', 'MCHC', 'RBC', 'WBC', 'PLT', 'LINF8', 'MONO1', 'EO1', 'BASO1', 'NETI2', 'MONO2', 'EO2', 'BASO2', 'PLT', 'PDW
', 'MPV', 'P_LCR', 'PCT', 'PH', 'PRQ', 'Glu', 'KET', 'HDL', 'SIL', 'URO', 'KET', 'SD', 'ESQ', 'H_QT', 'VC', 'FSG', 'TC', 'TG', 'LDL_C', 'HDL_C', 'ALT', 'AST', 'AST_ALT', 'BUN', 'Cr', 'Ca2', 'Na'], n
um_classes=2, num_trees=1, impurity='gini', max_depth=4, max_bin=100, seed=507463409, categorical_features_info=None, feature_subset_category=None)
[100.00% Tasks retries:0 Time 0:00:07]
Predicting on the frame
[api model_random_forest_classifier.py[313] model:random_forest_classifier/test (RandomForestClassifierModel:7fd3986c3d90, test(frame=Frame:7fd3986c3d90, label_column='GXY', observation_column=
'Age', 'Sex', 'BMI', 'DBP', 'SBP', 'HCT', 'MCV', 'RDW_SD', 'RDW_CV', 'HGB', 'MCH', 'MCHC', 'RBC', 'WBC', 'NETI1', 'LINF8', 'MONO1', 'EO1', 'BASO1', 'NETI2', 'MONO2', 'EO2', 'BASO2', 'PLT', 'PDW',
'ESQ', 'P_LCR', 'PCT', 'PH', 'PRQ', 'Glu', 'KET', 'HDL', 'SIL', 'URO', 'KET', 'SD', 'ESQ', 'H_QT', 'VC', 'FSG', 'TC', 'TG', 'LDL_C', 'HDL_C', 'ALT', 'AST', 'AST_ALT', 'BUN', 'Cr', 'Ca2', 'Na'])
[100.00% Tasks retries:0 Time 0:00:04]
Precision: 0.846153846154
Recall: 0.371875
Accuracy: 0.945
FMeasure: 0.285714285714
Confusion Matrix

```

	Predicted Pos	Predicted Neg
Actual Pos	11	53
Actual Neg	2	934

```

Run 1 test in 167.178s

```

图 7.5 随机森林算法预测结果

从图 7.4 可以看出，准确了高达 84% 以上，符合本系统的准确性要求。

测试过模型的准确性之后就可以对患者进行真实的预测了，预测的命令也很简单。

➤ `output = classifier.predict(frame_product)`

`frame_product` 是以用于预测的数据创建的 `Frame`，用该 `Frame` 进行预测后的结果将会输出出来，通过查看结果我们就可以活着用户是否患有高血压疾病了。

高血压风险预测模块就是一这样的顺序和原理进行运行的，在本系统中，上面所提到的指令都是集成在一个 `python` 文件中的，预测是知道启动该 `python` 脚步即可输出结果，完成这个预测过程。

## 7.4 本章小结

本章主要介绍了高血压预测系统核心模块之一的高血压预测模块的实现过程。本章首先对高血压预测模块基于的 TAP 平台的搭建过程进行描述，接着结合 TAP 平台提供的 API 对高血压预测模块的具体实现进行详细的介绍。

本章详细的介绍了高血压预测模块的实现过程，充分展示了高血压风险预测系统的运行原理，结合上一章的数据处理量化模块的实现，本系统的两个核心模块设计和实现介绍完毕，下一章将对这两个模块进行测试和相应的分析。

## 第8章 系统核心模块的测试与分析

本章主要向大家介绍数据处理量化模块和高血压预测模块的具体测试过程，本章的测试就是要通过实际的数据，系统的检测，来检验上两章所实现的程序的准确性。通过这些结果可以进一步的对本高血压风险预测系统的缺点提出修改意见，进行优化。通过对系统核心模块的测试以及相关测试数据的分析，本章可以使大家对本系统有更深一步的了解。

### 8.1 引言

前面介绍了高血压风险预测系统核心模块实现的整个过程，而本节将从核心模块的实现出发对模块的各个方面进行测试和评估，从而对整个系统的准确性，优势以及劣势有个直观的了解，为以后进一步优化核心模块提供依据。

本系统核心模块的测试所采用的数据是在医疗健康体检机构采集到的最近导出的一万条数据，使用真实的数据进行本系统核心模块的测试，最后分析测试结果并做出总结。

### 8.2 测试的方案

本测试因为只对核心模块进行的测试，所以只进行功能测试、兼容性测试和性能测试这三项测试，通过这三项对系统核心模块的方方面面进行测试。功能测试主要是对高血压风险预测系统程序实现的各个功能进行测试，测试各个功能是否正常，功能测试使用的是黑盒测试。兼容性测试是通过将模块放入不同的运行环境，从而测试其程序对运行环境的兼容程度。性能测试是通过高血压风险预测系统运行模块的时间等数据对高血压风险预测系统的性能表现进行测试。

### 8.3 测试需求描述

表 8.1 描述了系统核心模块测试的优先级和测试的项目。

表 8.1 测试需求描述

模块与功能		优先级	功能测试	兼容性测试	性能测试
数据处理模块	Feature 选择	中	√	×	×
	数据处理规则设置	高	√	×	×
	数据处理	高	√	√	√
高血压预测模块	连接 TAP 并导入数据源	中	√	√	√
	创建 Frame	中	√	√	√
	模型训练	高	√	√	√
	数据预测	高	√	√	√

因为不同的模块有很大的不同，所有需要对不同的模块区别对待，比如其中的兼容性测试和性能测试就不是每个模块都需要进行的。

- 数据处理量化模块是在高血压风险预测系统中是比较独立的一个模块，对于该模块就需要进行详细的测试。
- 高血压预测模块是整个系统的核心模块，对该模块的每一项测试都是不可或缺的，而且大部分测试优先级都是高的。

## 8.4 测试执行情况

下面从功能测试开始进行对每个模块进行测试，直到完成功能测试、兼容性测试、性能测试整个过程。

### 8.4.1 功能测试

功能测试采用黑盒测试的方式手工进行的，对每一个模块每一个功能至少测试三次。

#### (1) 数据处理量化模块

数据处理量化模块主要有 Feature 选择、数据处理规则设置和数据处理三大功能构成，针对这三大功能进行下列测试。

测试一：

测试用例：Feature 选择

目的：测试 Feature 选择功能

内容：Feature Name 输入、Data Type 输入。点击 Feature 的 add、delete、Edit 操作检查是否正常，测试用例表，如表 8.2 所示。

表 8.2 Feature 选择测试用例

测试用例	输入	输出
1	Feature Name 为空、Data Type 为空，点击 Add	错误--提示 Name 和 Type 未输入
2	Feature Name 为空、Data Type 为 float64， 点击 Add	错误-提示 Name 未输入
3	Feature Name 为 Na、Data Type 为空，点击 Add	错误-提示 Type 未输入
4	Feature Name 为 Na、Data Type 为 float64， 点击 Add	正确--提示 Feature 添加 成功
5	Feature Name 为空、Data Type 为空，点击 Delete	错误--提示 Name 和 Type 未输入

续表 8-2

6	Feature Name 为空、Data Type 为 float64, 点击 Delete	错误-提示 Name 未输入
7	Feature Name 为 Na、Data Type 为空, 点击 Delete	错误-提示 Type 未输入
8	Feature Name 为 Na、Data Type 为 float64, 点击 Delete	正确--提示 Feature 删除成功
9	Feature Name 为空、Data Type 为空, 点击 Edit	错误--提示 Name 和 Type 未输入
10	Feature Name 为空、Data Type 为 float64, 点击 Edit	错误-提示 Name 未输入
11	Feature Name 为 Na、Data Type 为空, 点击 Edit	错误-提示 Type 未输入
12	Feature Name 为 Na、Data Type 为 float32, 点击 Edit	正确--提示 Feature 修改成功

测试二:

测试用例: 数据处理规则设置

目的: 测试数据处理规则设置功能

内容: 在界面左侧编辑框内修改数据处理规则对应的 xml, 点击 Save 按钮, 检查其功能是否正常。测试用例表, 如表 8.3 所示。

表 8.3 数据处理规则设置测试用例

测试用例	输入	输出
1	xml 中没有增加任何处理规则节点, 点击 Save	错误--提示未添加任何处理规则
2	xml 中增加一个处理规则节点, 点击 Save	正确--提示处理规则添加成功

测试三：

测试用例：数据处理

目的：测试数据处理功能

内容：在 **Source Data** 添加原始数据路径，在 **Export Path** 添加处理完成数据导出路径，点击 **Process** 按钮，检测其功能是否正常。测试用例表，如表 8.4 所示。

表 8.4 数据处理测试用例

测试用例	输入	输出
1	Source Data 为空，Export Path 为空，点击 Process	错误—提示未添加任何数据
2	Source Data 为 G:\SourceData\Data.txt，Export Path 为空，点击 Process	错误—提示未添加数据输出位置
3	Source Data 为空，Export Path 为 G:\DataResult\Data.csv，点击 Process	错误—提示未添加任何数据
4	Source Data 为处理过的数据 G:\DataResult\Data.csv，Export Path 为 G:\DataResult\Data2.csv，点击 Process	错误—提示原始数据已经处理过
5	Source Data 为 G:\SourceData\Data.txt，Export Path 为 G:\DataResult\Data2.csv，点击 Process	正确—提示数据处理成功

## （2）高血压预测模块

高血压预测模块主要有连接 TAP 并导入数据源、创建 Frame、模型训练和数据预测四大功能构成，针对这四大功能进行下列测试。

测试一：

测试用例：连接 TAP 并导入数据源

目的：测试连接 TAP 并导入数据源功能

内容：在程序中修改对应的 TAP 运行实例的 **server.uri** 和连接 TAP 平台的验证文件路径，以及数据的 **hdfs.url**，运行程序，检查其功能是否正常。测试用

列表，如表 8.5 所示。

表 8.5 连接 TAP 并导入数据源测试用例

测试用例	输入	输出
1	修改 ta.server.uri、验证文件路径、hdfs 的 url 为空，运行程序	错误—提示连接不成功
2	修改 ta.server.uri 为 atk-bcc31ff0-b3c2-4059-9749.10.239.165.216.xip.io、验证文件路径为/root/demo.creds、hdfs 的 url 为空，运行程序	错误—提示 TAP 平台连接成功但数据为成功导入
3	修改 ta.server.uri 为空、验证文件路径为 /root/demo.creds、hdfs 的 url 为正确路径，运行程序	错误—提示 TAP 平台连接不成功
4	修改 ta.server.uri 为空、验证文件路径为 /root/demo.creds、hdfs 的 url 为正确路径，运行程序	错误—提示 TAP 平台连接不成功
5	修改 ta.server.uri 为 atk-bcc31ff0-b3c2-4059-9749.10.239.165.216.xip.io、验证文件路径为空、hdfs 的 url 为正确路径，运行程序	错误—提示 TAP 平台连接不成功，未通过验证
6	修改 ta.server.uri 为 atk-bcc31ff0-b3c2-4059-9749.10.239.165.216.xip.io、验证文件路径为/root/demo.creds、hdfs 的 url 为正确路径，运行程序	正确—提示 TAP 平台连接成功并成功导入数据

测试二：

测试用例：创建 Frame

目的：测试创建 Frame 功能

内容：在程序中修改 Frame Name，运行程序，检查其功能是否正常。测试用例表，如表 8.6 所示。



表 8.6 创建 Frame 测试用例

测试用例	输入	输出
1	修改 Frame Name 为空，运行程序	错误--提示创建 Frame 不成功
2	修改 Frame Name 为 LinearRegressionSampleFrame，运行程序	正确--提示创建 Frame 成功
3	修改 Frame Name 为已经存在的 frame 名称，运行程序	正确--提示创建 Frame 成功

测试三：

测试用例：模型训练

目的：测试模型训练功能

内容：在程序中修改模型训练相应的参数如 feature 字段数组，目标 feature，运行程序，检查其功能是否正常。测试用例表，如表 8.7 所示。

表 8.7 模型训练测试用例

测试用例	输入	输出
1	修改 feature 字段数组为空，目标 feature 为空，运行程序	错误--提示模型训练不成功
2	修改 feature 字段数组为正确数组，目标 feature 为空，运行程序	错误--提示模型训练不成功
3	修改 feature 字段数组为空，目标 feature 为 GXY，运行程序	错误--提示模型训练不成功
4	修改 feature 字段数组为正确数组，目标 feature 为 GXY，运行程序	正确--提示模型训练成功

测试四：

测试用例：数据预测

目的：测试数据预测功能

内容：在程序中修改数据预测相应的参数如 **feature** 字段数组，目标 **feature**，运行程序，检查其功能是否正常。测试用例表，如表 8.8 所示。

表 8.8 数据预测测试用例

测试用例	输入	输出
1	修改 feature 字段数组为空，目标 feature 为空，运行程序	错误--提示数据预测不成功
2	修改 feature 字段数组为正确数组，目标 feature 为空，运行程序	错误--提示数据预测不成功
3	修改 feature 字段数组为空，目标 feature 为 GXY，运行程序	错误--提示数据预测不成功
4	修改 feature 字段数组为正确数组，目标 feature 为 GXY，运行程序	正确--提示模型训练成功，输出预测结果

#### 8.4.2 兼容性测试

兼容性测试主要分为两个部分，第一部分是对在 **ubuntu** 平台运行的兼容性进行测试，第二部分是对 **windows** 平台运行的兼容性进行测试。表 8.9 描述了兼容性测试的详细说明。

表 8.9 兼容性测试说明

测试目标	<p>确保数据处理模块程序在 windows 平台能够正常运行</p> <p>确保高血压预测模块程序在 ubuntu 平台可以正常运行。</p> <p>确保高血压预测模块程序在 windows 平台可以正常运行。</p>
完成标准	在 ubuntu 和 windows 上都能得到很好的运行

续表 8.9

完成情况	高血压预测模块程序在 ubuntu 和 windows 平台上都能得到很好的运行，而数据处理模块只能在 windows 平台上运行，该问题尚未解决。
------	--

### 8.4.3 性能测试

#### (1) 数据处理量化模块

在对数据处理量化模块的性能测试中使用的是 10000 条未经处理的数据进行的测试，该模块的性能测试主要测试了在数据导入阶段和数据处理阶段。分别对这两个阶段进行了 10 次测试，他们所使用的时间如图所示，纵坐标表示为所消耗的时间，单位为秒。

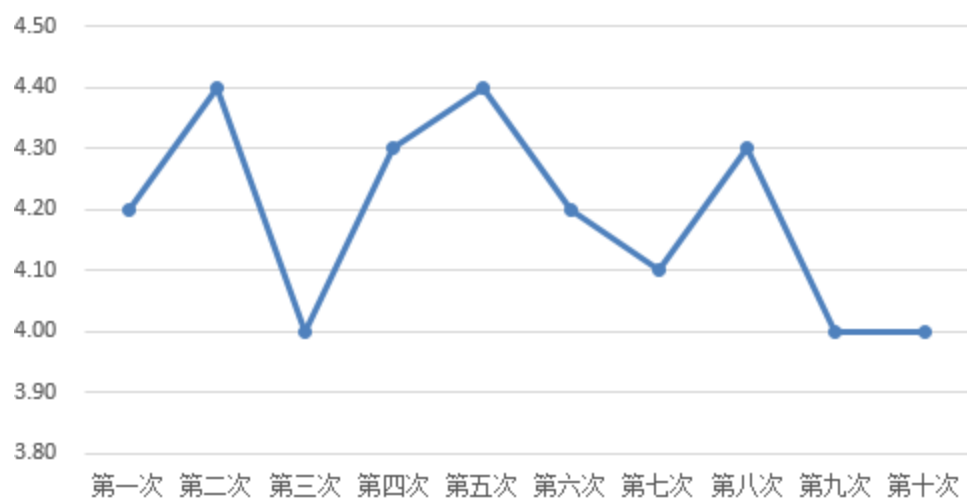


图 8.1 数据导入时间折线

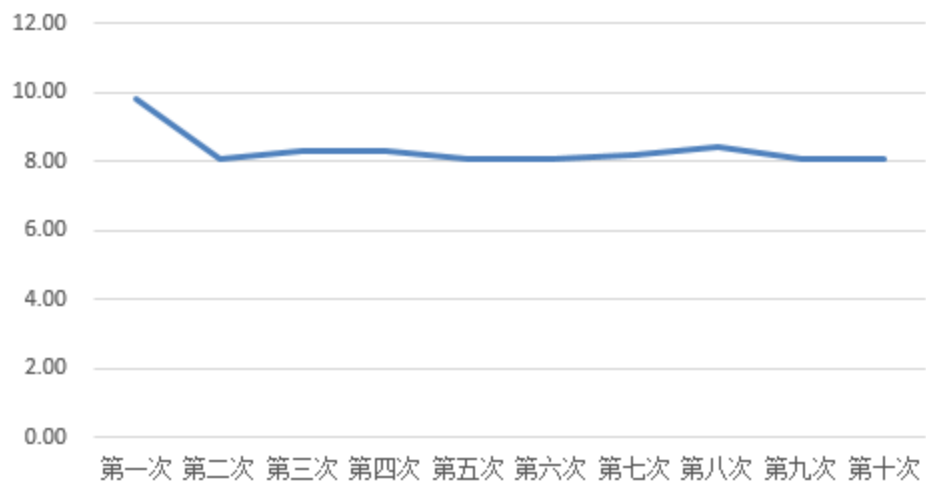


图 8.2 数据处理时间折线

从图 8.1 和图 8.2 中可以看出,数据量化处理模块所使用的时间是比较少的,一万条数据载入时间平均需要 4 秒,数据处理时间平均需要 8 秒,表现比较好,但是还有很大的提高空间。

(2) 高血压预测模块

高血压预测模块使用的是数据处理模块处理完成的一万条数据,其中 8000 用于预测模型的训练,2000 条用于检测预测的准确性。分别运行 20 次,各个功能模块所使用的平均时间如表 8.10 所示。

表 8.10 模块功能消耗平均时间

模块功能	平均消耗时间（单位秒）
创建 Frame	4
模型训练	8
数据预测	5

从表中可以看出,在数据量大,计算量很大的情况,该模块的性能都很优秀,其所使用的时间远远少于传统方法所消耗的时间,这只是本系统区别领先

于其他系统的所在。

高血压预测模块除了时间性能优秀外，其预测的精确度也相当优秀，在本次性能测试中，对该模块进行的 20 次测试，测试结果用图 8.3 所示，纵坐标为准确率。

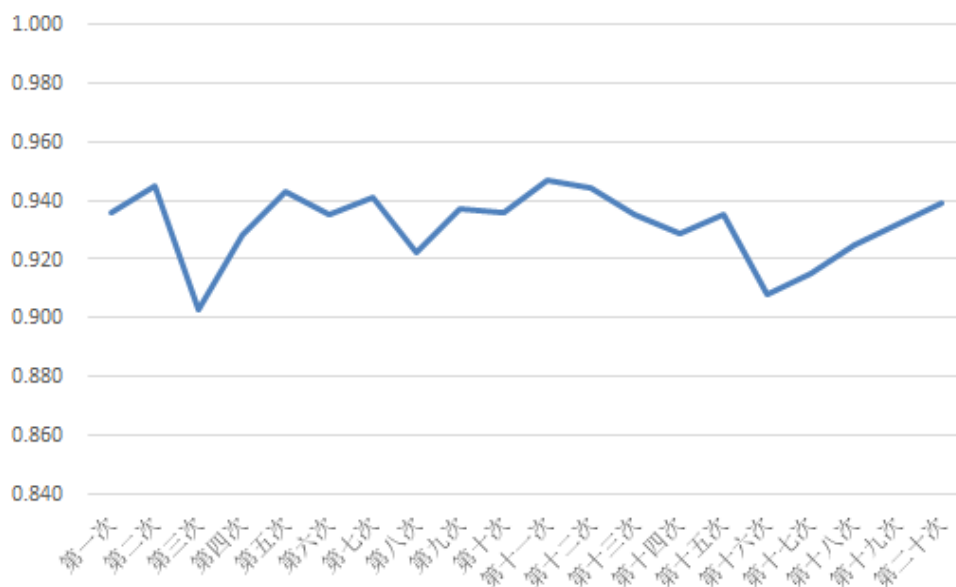


图 8.3 预测准确率折线

从图中可以看出本系统模块的数据模型预测准确率在 90% 以上，平均在 93% 左右，预测准确率较高，系统值得信赖。

#### 8.4.4 结果分析

测试结果表明我所开发的高血压风险预测系统核心模块的所有功能都通过测试，在测试过程中也没有发生崩溃或者严重卡顿，又或者预测出的结果变动性较大的情况。在测试高血压风险预测系统的对各种操作的反应中，98% 的操作都能做到毫秒级别，只有数据处理、模型训练以及数据预测几个操作消耗时间比较长，但尚在可接受范围之内，考虑到本系统是对大量数据进行操作以及执行很复杂的算法操作，本系统的性能优势比较突出。

在高血压风险预测系统的各个模块中数据处理模块由于所采用的语言对平台的依赖性比较到，导致数据处理模块只能在 windows 平台上进行运行，从这

一点看，高血压风险预测系统对设备的支持性存在不足，需要在这些地方加以改进。

#### ● 优势

从对模块的结果上看，本系统性能突出，运行稳定，响应时间比较短，并且最后预测的结果也比较准确高达 90% 以上，并且在模型训练数据的比例以及数量进行改变之后该准确性还会进一步提高，这对作为高血压预测的系统来说尤为重要。除此之外这两个模块操作比较简单，通过把复杂的逻辑封装到内部，只展现给用户简单易懂的操作界面，比较友好。

#### ● 不足

在模块的测试过程中也充分的暴露除了高血压风险预测系统的不足之处，首先一点是本系统对平台的支持性不够，数据处理模块只能在 windows 平台上运行，这需要在今后的版本中进行改进，其次一点是没有将数据处理模块集成到 TAP 平台之后，对数据的处理还只是使用平台程序进行处理，这在以后数据量增大的情况下，系统将会非常吃力，会成为整个系统的瓶颈，这一点也是需要下一版本的高血压风险预测系统的开发中进行改进，将其集成到 TAP 云计算平台之中。最后一点不足是整个程序没有统一的界面展示，这不利于系统的发展，迫切需要进行改进。

### 8.5 本章小结

本章主要介绍了对高血压风险预测系统核心模块的数据处理量化模块和高血压风险预测模块这两个模块的测试过程，通过对这两个模块的测试，具体进行功能测试、性能测试和兼容性测试来验证模块实现的准确性以及模块性能的优越性。

本章对高血压风险预测系统的核心模块进行测试，并从测试结果中获得对本系统直观的了解，为下一章的总结提供事实依据。

## 第9章 总结与展望

本章主要是对全文进行总结，通过传统高血压风险预测方案与本文的高血压风险预测系统的对比进一步明确设计本系统的初衷，通过对本系统的优势和劣势进行概括总结，进而对高血压风险预测系统的实际价值进行评估。最后根据在现有云计算技术以及高血压风险预测技术的现状基础上的本系统的不足之处出发，对本高血压风险预测系统的发展进行展望。

### 9.1 总结

传统的高血压风险预测方案要么是从当前医疗水平出发，要么从当时科技水平出发，所提出的高血压风险预测方案会因不同医师经验或者当时对高血压的认知水平影响，而使结果具有很大的不确定性，其预测结果并不能给用户有很大的信服度。近些年发展起来的高血压风险预测系统也由于对国人的不适应性而在国内无法采用。

本文设计并实现的高血压风险预测系统是从与高血压疾病无直接相关的因素出发，使用最新的云计算技术，根据国人的体质数据而定制的一套对高血压疾病进行预测的系统。通过实际测试，总结本系统的优缺点如表 9.1 所示。

表 9.1 高血压风险预测系统的优缺点

优点	缺点
响应时间短，结果预测准确率高	对平台兼容性不好
性能好，运行速度快	没有统一的界面
对海量数据支持性好	数据模型有待改善
可扩展性好，便于优化	数据的处理对大数据支持性有限

本系统在高血压预测准确性，相应速度以及性能上都具有很大的优势，这也是本系统领先于其他高血压风险预测方案的地方。使用本系统进行对用户的

高血压风险进行预测能够很好的满足用户的需求。

## 9.2 展 望

本高血压风险预测系统虽然有很大的优势，但是本系统在某些方面也存在一些不足之处，本节将根据测试的结果针对本系统存在的缺陷对未来进行展望。

- 对平台兼容性不好

高血压风险预测系统有一大缺陷就是对平台兼容性不好，这主要体现在数据处理模块只能运行在 windows 平台上，而针对这一点，本系统在以后将该用平台支持性好的语言来编写数据处理程序，从而使整个高血压风险预测系统可以对多平台进行支持。

- 没有统一的界面

目前的高血压风险预测系统尚无统一的界面供用户进行操作，这非常不方便，而在本系统的项目开发计划中将会开发出 web 界面和移动端界面，如此用户可以在时尚友好的界面中来预测高血压的风险性。

- 数据模型有待改善

本系统暂时使用的数据模型是随机森林算法，虽然该算法可以满足需求，当在今后的时间里，本系统将会考虑多种算法，如神经网络、Weibull 回归、Cox 风险比例回归以及相关的高血压预测公式等，并在这些算法的基础上提出自己独创性的算法模型，使用该模型来预测，这将会进一步提高本系统预测的准确性。

- 数据的处理对大数据支持性有缺陷

本系统因为数据处理模块尚未使用云计算的相关技术来处理，这将会给我们的系统带来性能上的瓶颈，在今后的时间里，开发人员将致力于把数据处理模块使用 TAP 平台自带的 api 进行处理，从而消除这一瓶颈。

总之，高血压风险预测系统虽然目前还存在不少缺陷，但在不久的将来会越来越精确，越来越优秀。



## 参考文献

- [1]. 罗珮允. 2013. 云计算环境下著作权侵权保护研究[D]: [硕士学位论文]. 大连: 大连海事大学
- [2]. 中国 IDC 圈. 2016 年云计算领域发展趋势 [Z]. 中国 IDC 圈. [http :  
//cloud.idcquan.com/yzx/83754.shtml](http://cloud.idcquan.com/yzx/83754.shtml). [2016-01-26]
- [3]. 前瞻产业研究院. 2014. 2013-2017 年中国云计算产业市场前景与投资机会分析报告 [J]. 前瞻产业研究院. 北京: 年鉴社出版社
- [4]. 刘力生. 2011. 中国高血压防治指南 2010[M]. 中国高血压防治指南修订委员会. 北京: 人民军医出版社
- [5]. 刘冬. 2007. 基于遗传算法的 BP 网络在医疗诊断中的应用[D]: [硕士学位论文]. 吉林: 吉林大学
- [6]. 马里兰. 2013. 冠心病风险评估方法研究进展[J]. 中国保健营养旬刊, 23(7): 10~15
- [7]. 张晔、蔡心轶. 科学生活: 用“冷加压”预测高血压准吗? [Z]. 科技日报. [http :  
//www.gov.cn/fwxx/kp/2012-05/28/content\\_2146610.html](http://www.gov.cn/fwxx/kp/2012-05/28/content_2146610.html). [2012-05-28]
- [8]. 李冬. 2012. 基于分类器集成技术的高血压预测与诊疗的研究[D]: [硕士学位论文]. 北京: 北京理工大学
- [9]. Bazilian Morgan D. 2002. Modelling of a photovoltaic heat recovery system and its role in a design decision support tool for building professionals[J]. Renewable Energy, 27(1): 57~68
- [10]. Monique Frize, Colleen M. 2001. Clinical decision support systems for intensive care units: using artificial neural networks[J]. Medical Engineering & Physics, 23(3): 22~217
- [11]. Valafar, Valafar. 2002. Data mining and knowledge discovery in proton nuclear magnetic resonance spectra using frequency to information transformation (fit). Knowledge-Based Systems[R], 15(4): 251~259
- [12]. William G Baxt.. 1995. Application of Artificial Neural Networks to clinical Medicine Lancet[R]. Lancet, 346(8983): 81~135
- [13]. Zheng L, Sun Z, Zhang X, Li J. 2014. Framingham 高血压风险预测模型在中国农村人口中的预测价值[J]. 中国高血压杂志, 25(5): 18~26
- [14]. 孙艳秋, 刘钢. 2014. 基于大数据分析的潜在高血压病预测研究[D]: [硕士学位论文]. 辽宁: 辽宁中医药大学信息工程学院
- [15]. 李现文, 李春玉. 2012. 决策树与 Logistic 回归在高血压患者健康素养预测中的应用[D]: [硕士学位论文]. 美国: 美国 John Hopkins 大学
- [16]. 王重建, 李玉倩. 2010. 人工神经网络在个体患原发性高血压预测中的应用[D]: [硕士学位论文]. 郑州: 郑州大学公共卫生学院流行病与卫生统计学系
- [17]. Rajiv Agarwal, MD, Allen R. Prevalence. 2003. Treatment and Control of Hypertension in Chronic Hemodialysis Patients in the United States. Excerpta Medical Inc[J]. American

- Journal of Medicine, 115(4): 7~291
- [18]. Jie Su. 2006. Data Mining Based Evaluation Method for Hypertension Disease[D]: [master degree papers] . Hangzhou: Zhejiang University, Hangzhou, P.R.China
- [19]. 杨洋. 2010. 利用人工神经网络模型预测原发性高血压的研究[D]: [硕士学位论文]. 沈阳: 中国医科大学
- [20]. 赵秀丽, 胡大一. 2006. 中国 14 省高血压现状的流行病学研究[J]. 中华医学杂志, 86(16): 1148~1152
- [21]. 程遥, 万隧人. 2014. 基于 BP 神经网络的高血压诊疗预测分析[D]: [硕士学位论文]. 南京: 东南大学生物医学工程学院
- [22]. Li Bo. 2014. A study of hypertension epidemiological characteristics and influence factors among Chinese adult[D]: [master degree papers] . Huazhong: Huazhong University of Science & Technology
- [23]. Lixuan Gui. 2014. Genetic risk score predicts coronary heart disease risk in a Chinese Han population[D]: [master degree papers]. Huazhong: Huazhong University of Science & Technology
- [24]. 晁灵. 2015. 分类树模型与 Logistic 回归在儿童高血压预测中的应用[D]: [硕士学位论文]. 河南: 新乡医学院公共卫生学院
- [25]. MA Liang-Liang, TIAN FU-Peng. 2010. Application of time series analysis in the prediction of hypertension incidence[D]: [master degree papers]. Lanzhou: School of Computer and Information, Northwest University for Nationalities, Lanzhou
- [26]. 党红刚. 2011. 基于 ARIMAX 模型的海西州地区高血压月发病率预测[D]: [硕士学位论文]. 甘肃: 天水师范学院数学与统计学院
- [27]. Mahmud Mavaahebi, Ken Nagasaka. 2012. A Network and Expert System Based Model for Measuring Business Effectiveness of Information Technology Investment. Department of Electronic Engineering[D]: [master degree papers] . Japan: Tokyo University of Agriculture and Technology, Tokyo, Japan
- [28]. Qeethara Kadhim Al-Shayea. 2011. Artificial Neural Networks in Medical Diagnosis[D]: [master degree papers]. Amman :MIS Department, Al-Zaytoonah University of Jordan Amman, Jordan
- [29]. 俞浩, 郭志荣. 2010. 代谢综合评分与 Framingham 风险评分预测心血管疾病的比较 [D]: [硕士学位论文]. 苏州: 苏州大学放射医学与公共卫生学院
- [30]. 马亮亮. 2012. 基于 PCA\_ARIMA 模型的高血压发病率预测[D]: [硕士学位论文]. 攀枝花: 攀枝花学院数学与计算机学院
- [31]. 张军跃. 2015. 高血压智能防控平台建设构想[J]. 中日友好医院学报, 29(2): 120~122
- [32]. 马光志. 2008. 基于神经网络的高血压在线风险评估系统[D]: [硕士学位论文]. 武汉: 华中科技大学计算机学院
- [33]. 刘力生. 2002. 高血压研究四十年[J]. 中国医学科学院学报, 24(4): 401~408
- [34]. 顾东风. 2002. 中国成年人高血压患病率\_知晓率\_治疗和控制状况[D]: [硕士学位论文]

- 文] . 北京: 中国医学院科学院中国协和医科大学
- [35]. 王文娟. 2001. 体重指数\_腰围和腰臀比预测高血压\_高血糖的实用价值及其建议值探讨[D]: [硕士学位论文] . 北京: 中国预防医学科学院
- [36]. YU Jing, JU Bin. 2012. Application of Data Mining Technology in Regional Health Information Platform Chronic Disease Management. China Digital Medicine[J]. International Journal of Chronic Obstructive Pulmonary Disease, 7(2) : 82~271
- [37]. Liqiang Zheng. 2014. Validation and Construction a Hypertension Risk Prediction Model in Rural Areas of Fuxin Country with the High Incidence of Hypertension[D]: [master degree papers] . China : China Medical University
- [38]. 刘冰. 2009. 中国 35\_45 岁人群高血压前期检出率及影响因素分析[J]. 中华高血压杂志, 2010(2) : 187~192
- [39]. 陈亦敏. 2013. 6830 名老年人体质指数\_腰围\_腰臀比与高血压的关系研究[J]. 浙江中西医结合杂志, 2014(4) : 303~305
- [40]. 王霄飞. 2012. 基于 OpenStack 构建私有云计算平台[D]: [硕士学位论文] . 广州: 华南理工大学
- [41]. 李知杰, 赵健飞. 2012. OpenStack 开源云计算平台[J]. 软件导刊, 11(12) : 10~12
- [42]. Sefraoui, Omar, M. Aissaoui, M. Eleuldj. 2012. OpenStack: Toward an Open-source Solution for Cloud Computing[J]. International Journal of Computer Applications , 55(3) : 38~42
- [43]. Wuhib, Fetahi, R. Stadler, and H. Lindgren. 2012. Dynamic resource allocation with management objectives : implementation for an OpenStack cloud Communication Networks[R], 610: 309~315
- [44]. Chadwick, D. W. , Siu K. , Lee C, Fouillat, Germonville D. 2013. Adding federated identity management to openstack[J]. Journal of Grid Computing, 12(1) : 3~27
- [45]. Corradi A, Fanelli M, Foschini L. 2014. VM consolidation: A real case based on OpenStack Cloud[J]. Future Generation Computer Systems, 32(1) : 118~127.

## 致 谢

首先，我想向我的指导老师郑浩然先生送以最真挚的敬意，是您在我进行论文编写的过程中提供及时细致的支持，是您为我的论文提供最实用的建议，在您的帮助下我的论文才最终得以诞生。郑浩然老师在论文开题之时就为我的选题，构思提供了很好的指导，在郑老师的指导下我的开题报告才得以顺利通过学院的审核。在论文编纂阶段，郑老师对我论文的结构，格式以及词句的细节进行认真的审核，并热情的之处其中的不足之处。郑老师严谨的作风和一丝不苟的做事态度给我留下深刻的印象，每每在我意想不到的地方指正出错误，正是在郑老师这样的帮助之下我的论文才得以顺利完成，再次，我想再次对郑老师给予的帮助表示感谢。

同时，我也想感谢我的企业导师张丽娟女士，从我进入公司实习的第一天起，您就细心的教导我，在我工作和技术学习上给予很大的帮助，回想起这实习的一年，在您的帮助下我学到了很多，您对技术刻苦钻研的精神指引我不断的前进，最后也是在您的帮助之下我才顺利完成本文的项目。在此，我送上衷心的祝福。

除此之外，我还想感谢那些在我论文写作过程中给予我很大帮助的同事、同学以及家人，谢谢你们在我困难的时候陪伴着我，给予我支持，给予我鼓励。有你们的陪伴真的很好、很好。

最后，我也向各位在百忙之中对本文进行评审的专家们表示衷心的感谢。

于上海

2016年8月11日

## 盲审意见修改情况说明

论文设计并实现了一种高血压风险预测系统，选题具有工程背景和应用价值。论文的主要工作包括：

在现有研究的基础上，基于云计算 TAP 平台，通过使用海量数据对预测模型进行训练，设计并实现了一套高血压风险预测系统，并对系统进行了测试。但论文在组织、格式规范与表述上存在不少问题，希望作者认真加以修改。简单的罗列一些如下：

- (1) 英文题目翻译不准确，摘要需要进行修改；
- (2) 存在多处不规范或者不切当的表达或者表述；
- (3) 图名称表达不规范，应该删去其中的“图”字，它与前面的“图…”重复；
- (4) 文中应该尽可能用文字叙述设计实现原理与思想等；
- (5) 参考文献排版不符合规范。

## 答辩后论文修改情况说明