内部

中国科学技术大学

工程硕士学位论文

基于云计算技术的

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　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

Limited

University of Science and Technology of China

A dissertation for master's degree of engineering

Hypertension Risk System Based On Cloud Computing Technology Design And Implementation

Author's Name： Jun Li

Speciality： Software Engineering

Supervisor： Prof. Haoran Zheng

Advisor： Lijuan Zhang senior engineer

Finished time: Aug 11nd, 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

云计算技术在互联网相关服务的使用、增加和交付模式的改变中发展起来，云计算技术为用户提供动态可扩展的虚拟化资源，从而使用户具有强大的运算处理能力。云计算技术的不断发展来源于用户对大数据处理需求的不断增加，云计算应用的服务范围日益扩大，各行各业都使用云计算技术来促进行业的进步。

在医疗领域，医术已经在传统医学上前行很久，在高血压风险预测方面，国内外很早就尝试将新技术应用于研究和实践，这诞生了很多优秀成果，但由于运算能力的欠缺，以及硬件性能不足等原因使得该领域的发展比较缓慢。

本文描述的高血压风险预测系统是在前人的基础上，结合最新的云计算技术，使用最新的硬件设备，并在详细的研究和设计之后建立起来的一套对用户高血压风险预测的可靠系统。该系统使用多种先进云计算技术，可以使用海量数据对预测模型进行训练，并用此模型对用户进行预测，准确率高，并且可扩展性强。同时本系统的设计方法和实现过程也为其他疾病的预测系统的开发提供参考性建议。

本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云计算技术，以软件工程化的方法和理论，将本文分为需求分析，概要设计、系统设计、实现、测试等五个部分来对本系统进行描述，着重强调高血压风险预测系统的体系架构和处理流程，并以图片和表格的方式形象的描述了本系统的实现过程。

关键词：云计算，高血压，大数据处理

ABSTRACT

Cloud computing technology developed in Internet-related services using, adding and changing in the delivery model. This technology provide users dynamically scalable virtualized resources, and allowing users to have powerful computing capability. The development of cloud computing technology from the user to the increasing demand for large data processing, the range of services cloud computing applications growing, businesses are using cloud computing technology to promote the industry's progress.

In the medical field, medicine has been a long time in the front row of traditional medicine. in hypertension risk prediction, at home and abroad very early to try new technologies applied to the study and practice, w This creates a lot of outstanding achievements, but due to the lack of computing power, and lack of hardware performance and other reasons such developments in the field is relatively slow.

Hypertension risk prediction systems described herein is based predecessors, it combined with the latest cloud computing technology, and using the latest hardware, we set up a system after detailed study, It is user reliable to predict risk of hypertension. The system uses a variety of advanced cloud computing technology that can be used for mass data prediction model training, and use this model to predict, this system is high accuracy and scalability. While the design and implementation method of the system also provides the reference recommendations for the development of prediction systems of other diseases.

Based on the development at home and abroad, and combined with the current mainstream market demand and cloud computing technology, In software engineering methods and theories, the paper split the described into five parts like sub-requirement analysis, design, system design, implementation and testing. Emphasis on the architecture and processing, and use pictures and tabular to describes the implementation process of the system.

Keywords: Cloud computing , 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 主要研究内容 4

1.4 论文的组织 5

第 2 章 背景知识 7

2.1 Trusted Analytics Platform(TAP) 7

2.1.1 TAP 平台架构 7

2.2 OpenStack 10

2.3 CDH 11

2.4 Cloud Foundry 11

2.5 本章小结 12

第 3 章 需求分析与概要设计 13

3.1.1 需求分析 13

3.1.2 功能性需求 14

3.1.3 非功能性需求 16

3.2 概要设计 17

3.2.1 目标与概述 17
3.2.2 系统总体设计 18
3.2.3 量化程序客户端开发架构 19
3.2.4 TAP 平台架构 19
3.3 方法和工具 21
3.4 本章小结 22
第 4 章 高血压风险预测系统的设计 23
4.1 数据模型的选择 23
4.2 随机森林算法原理简述 25
4.3 高血压风险预测系统的设计 26
4.4 功能模块设计 26
4.4.1 数据处理模块的设计 27
4.4.2 数据管理模块的设计 29
4.4.3 运行实例管理模块的设计 31
4.4.4 高血压预测模块的设计 32
4.5 数据处理规则设计 34
4.5.1 汉字的量化 34
4.5.2 连续值的分类量化 35
4.5.3 关联其他字段的字段分类与量化 36
4.6 本章小结 37
第 5 章 高血压风险预测系统的实现 39
5.1 TAP 平台的搭建与实现 39
5.2 数据处理模块程序的实现 41
5.3 高血压风险预测应用程序的实现 52
5.3.1 连接 TAP 平台 53
5.3.2 增加数据源 54

5.3.3 创建并操作 Frames 55
5.3.4 模型训练及预测 56
5.4 本章小结 58
第 6 章 测试与分析 59
6.1 测试方案 59
6.2 测试描述 60
6.2.1 测试需求描述 60
6.2.2 测试成员及任务分配说明 61
6.3 测试执行情况 62
6.3.1 需求测试 62
6.3.2 功能测试 62
6.3.3 兼容性测试 64
6.3.4 性能测试 64
6.3.5 回归测试 64
6.4 测试结果及评估 65
6.4.1 功能性评估 65
6.4.2 可用性评估 65
6.4.3 性能评估 65
6.4.4 设备支持性评估 66
6.5 结果分析 66
6.5.1 高血压风险预测系统的优势 66
6.5.2 高血压风险预测系统的不足 66
6.6 本章小结 67
第 7 章 结 论 68
7.1 总 结 68
7.2 展 望 69

7.3 本章小结 69

参考文献 71

致 谢 74

表目录

表 4-1 国内外知名算法优缺点对比表.....23
表 5-1 高血压风险预测系统硬件开发环境.....39
表 5-2 高血压风险预测系统软件开发环境.....40
表 5-3 Form 类部分方法列表.....42
表 5-4 Form 类部分事件列表.....43
表 6-1 测试需求描述表.....60
表 6-2 测试任务分配表.....61
表 6-3 数据处理模块测试说明表.....62
表 6-4 数据管理模块测试说明表.....63
表 6-5 运行实例管理模块测试说明表.....63
表 6-6 高血压预测模块测试说明表.....63
表 6-7 兼容性测试说明表.....64
表 6-8 回归测试说明表.....65
表 7-1 高血压风险预测系统的优缺点.....68

图目录

图 2-1 TAP 体系架构图.....8
图 3-1 数据管理员需求用例图.....15
图 3-2 预测观察员需求用例图.....16
图 3-3 高血压风险预测系统的总体架构图.....18
图 3-4 量化程序客户端的体系结构图.....19
图 3-5 TAP 平台的体系结构图.....20
图 4-1 系统的主要工作流程图.....27
图 4-2 数据处理模块工作流程图.....28
图 4-3 数据管理模块工作流程图.....30
图 4-4 运行实例管理模块工作流程图.....31
图 4-5 高血压预测模块工作流程图.....33

图 4-6 Sex 字段分析图.....34
图 4-7 Sex 字段规则设计图.....34
图 4-8 CI 字段分析图.....35
图 4-9 CI 字段规则设计图.....35
图 4-10 Cr 字段分析图.....36
图 4-11 Cr 字段规则设计图.....37
图 5-1 数据处理模块主页面图.....44
图 5-2 数据处理模块项目结构图.....45
图 5-3 数据处理模块类图.....46
图 5-4 数据处理窗口图.....47
图 5-5 feature 选择窗口图.....48
图 5-6 数据处理规则编写窗口图.....49
图 5-7 数据处理模块时序图.....51
图 5-8 数据管理模块界面图.....52
图 5-9 运行实例创建界面图.....53
图 5-10 创建证书文件图.....54
图 5-11 随机森林算法预测结果图.....58

第1章 绪 论

随着人们生活水平的提高，对健康医疗方面的关注日益加强，而作为慢性病中最普遍的高血压疾病，对该病的预防和检测研究一直从未停止。在一些传统的检测预测方法中，检测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

互联网技术的不断发展，与人们生活联系的日益紧密，人们日益增长的需求不断推动互联网技术的进步，同时新的互联网技术又能满足人们新的需求。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[1]，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[2]。云计算技术在物联网、智能交通、智慧城市、云计算、手机支付、两化融合、视频监控、医疗信息化[3]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，而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%[4]。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涌现出一系列优秀

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，进而致使对高血压风险预测准确率的降低的问题，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际性的要求，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，为其他基本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，把专家的知识以及经验规则化形式化并存入其中，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[5]。

除此之外，在应用系统建设方面还有很多较为知名的系统。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[6]。

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[7]。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[6]。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[8]。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由于发展时期较早，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参数建立起来的，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检测，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 主要研究内容

本课题主要是针对高血压预测领域传统预测方法准确率不高，计算能力有

限等问题而设计的一个基于云计算平台的高血压风险预测系统，其主要包含以下 5 点内容：

（1）云计算平台（Trusted Analytics Platform，TAP）的部署以及优化。

（2）原始数据的填充，量化等处理。

（3）选择合适用于预测的变量

（4）建立以及训练高血压预测数据模型。

（5）使用模型进行预测及结果查询。

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的设计、高血压风险预测系统的实现、测试与分析以及结论这七个章节来详细介绍高血压风险预测系统。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发展现状和存在的问题。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 背景知识

简单的介绍高血压风险预测系统底层云平台的组织架构，详细的分析云平台所采用的云计算技术，对主要的几个技术以及框架进行进一步的阐述说明，同时分析高血压风险预测系统为什么采用该平台的缘由。

第三章 需求分析与概要设计

结合用户的实际使用需求，对高血压风险预测在使用过程中存在的一些问题和不足进行详细的分析，并得到具体的用户需求，再根据需求分析进行简单的概要设计。

第四章 高血压风险预测系统的设计

根据需求分析和概要设计，明确高血压风险预测系统的设计目标，分析主流数据处理技术的技术特点，介绍目前最优秀的几个数据模型算法，进行对比分析，并从中挑选了最适合本论文的设计高血压风险预测系统的各模块。

第五章 高血压风险预测系统的实现

根据高血压风险预测系统的设计，详细描述高血压风险预测系统的开发环

境，以及数据处理模块，模型训练模块，风险预测模块等模块的实现细节。

第六章 测试

主要描述高血压风险预测系统的测试方案和测试环境，得出了几个实验的测试结果，并针对测试的结果进行分析，最后总结了高血压风险预测系统的优缺点。

第七章 结论和展望

对高血压风险预测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，并展望了其未来的发展情况。

第2章 背景知识

2.1 Trusted Analytics Platform(TAP)

Trusted Analytics Platform(TAP)是一个平台即服务的云计算的平台，在该平台上数据专家和应用开发者能够创建和运行大规模数据分析驱动的特定领域的应用程序。

TAP 平台是简化了在大量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 数据科学计算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和机器学习的任何解决方案合并到工作流中并向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。

2.1.1 TAP 平台架构

本系统的 TAP 平台的体系架构如图 2-1 所示，TAP 平台是一个简化了图像分析和机器学习的云计算平台。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，TAP 平台使用一个可扩展、模块化的框架来为图形建构，图像分析以及机器学习提供一个分析管道，通过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的一些预测数据。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。

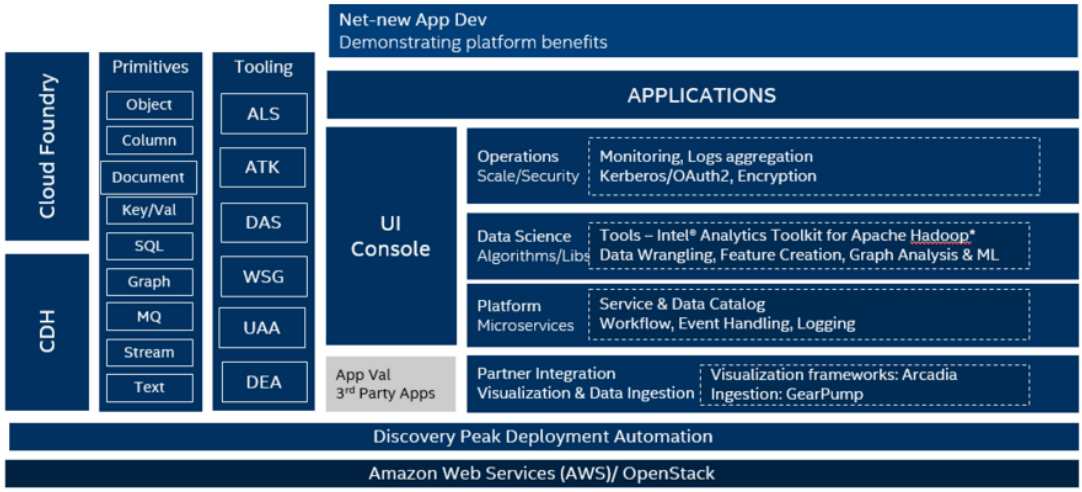

图 2-1 TAP 体系架构图

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，强强联合并且加之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高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，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。

2.2 OpenStack

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 就是一整套开源软件项目的综合，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 作为一个开源的项目，其发展是非常迅速的，无数的开发者在其中贡献出自己的智慧，使得 OpenStack 提供的几种服务尤为强大。Nova 是 OpenStack 的一个弹性计算的控制器，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，虽然 Nova 本身并没有提供任何虚拟化的能力，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，它为 OpenStack 提供一种分布式的持续的虚拟对象存储，类似于 AWS 的 S3，Swift 有着跨越节点进行百万级对象的存储能力，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。

2.3 CDH

CDH 是 cloudera 公司为了简化 hadoop 的安装，对 Hadoop 进行封装，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 cloudera 的官网上就提供了三种安装方式。

Cloudera 的 CDH 是对 hadoop 的封装，同时也在 hadoop 的基础上进行了一些优化，这使得 Cloudera CDH 比 Apache hadoop 在兼容性、安全性以及稳定性上有所增强。

Cloudera CDH 的 hadoop 版本划分的非常清晰，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。

2.4 Cloud Foundry

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 可以使开发人员在短短的几秒之内进行应用的部署以及扩展，而不需要担心任何基础架构的问题。

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。

在 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 Cloud Foundry 中 Cloud Controller 负责其 app 的整个生命周期，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。

2.5 本章小结

本章主要介绍高血压风险预测系统使用到的 OpenStack、CDH、Cloud Foundry 等等一些云计算技术，并对这些技术进行一些比较细致的介绍，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，这也从侧面反应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　　需求分析与概要设计

通过前面的章节对高血压风险预测系统的发展情况的分析，以及从用户实际使用角度出发，我们得到了很多关于高血压风险预测系统的需求，针对这些需求进行分析，并给出简单的概要设计。

3.1.1 需求分析

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，催生出了下面一系列的需求。

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迫切的需求，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一种疾病，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一种疾病，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。

3.1.2 功能性需求

高血压风险预测系统中包括了数据管理员和预测观察员两种用户，根据其不同的角色对系统有不同的需求。

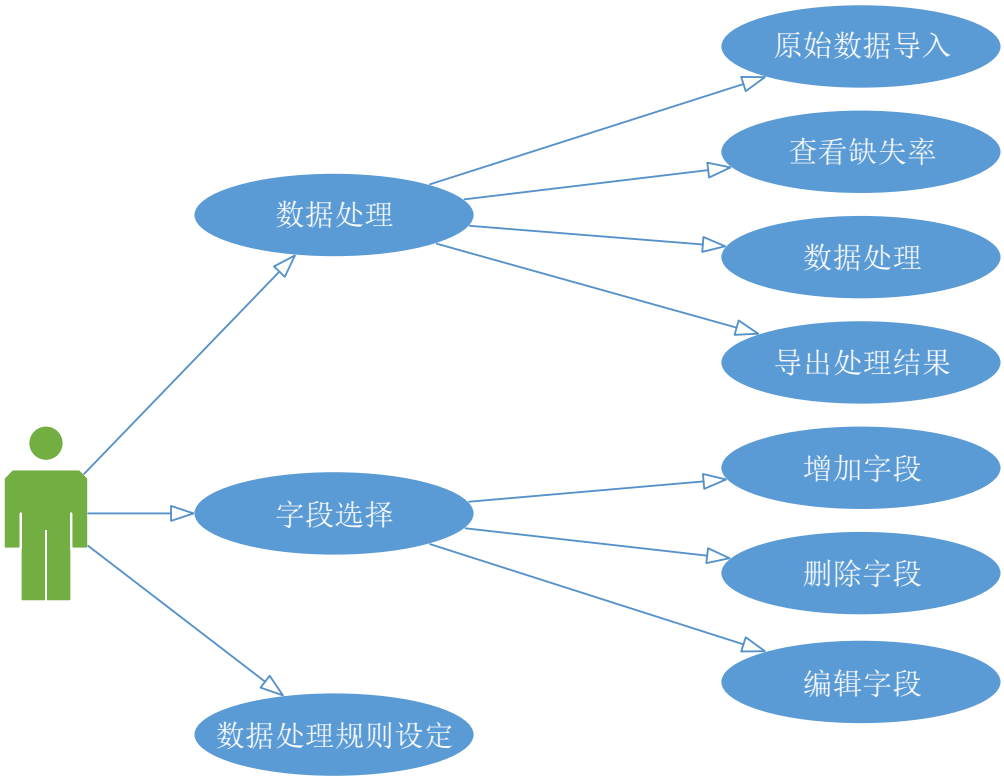

图 3-1 数据管理员需求用例图

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，种类繁杂，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。

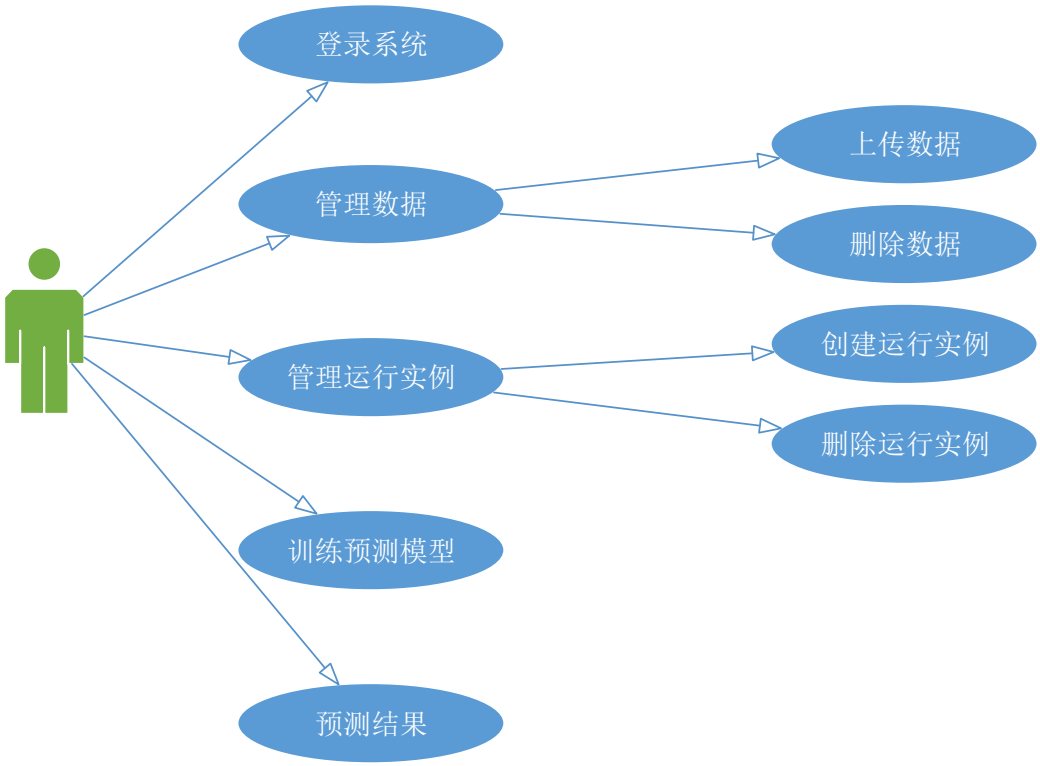

图 3-2 预测观察员需求用例图

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，选择合适的数据模型进行训练，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。

3.1.3 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：根据上传大量数据的需要，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，而且要方便用户对数据的操作。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。

3.2 概要设计

3.2.1 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，本设计要达到安全性、可维护性以及稳定性的三项性能上的要求。安全性就是要求对预测观察员实现对平台的操作，对数据的管理，对运行实例的管理等进行访问权限的管理与控制，对用户的合法权进行保护。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。

3.2.2 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，同时也能在 web 浏览器上对平台相关信息进行管理，并且实现相应的预测功能。

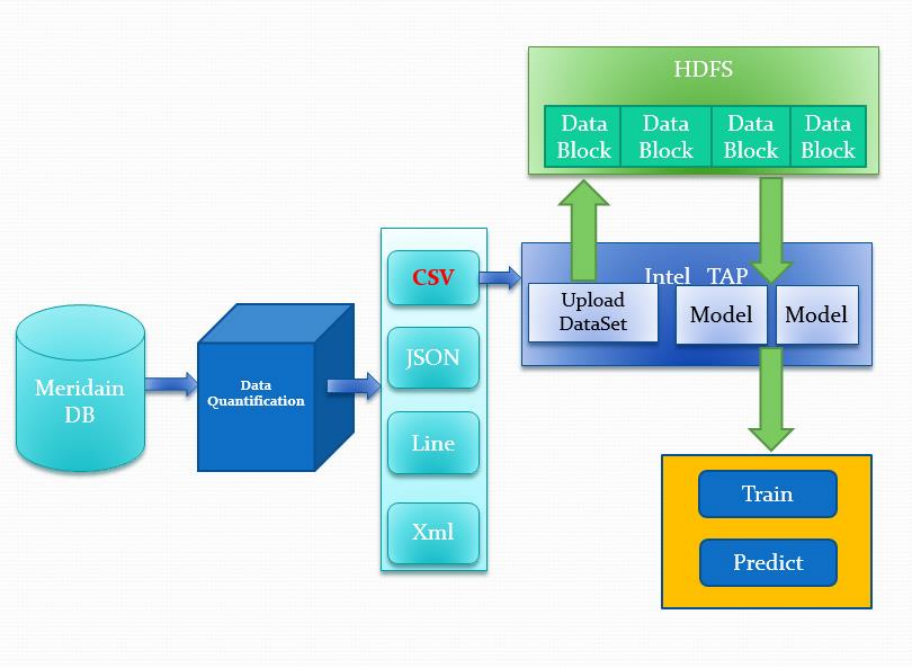

图 3-3 高血压风险预测系统的总体架构图

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。

3.2.3 量化程序客户端开发架构

本系统量化程序客户端的体系结构如图 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。

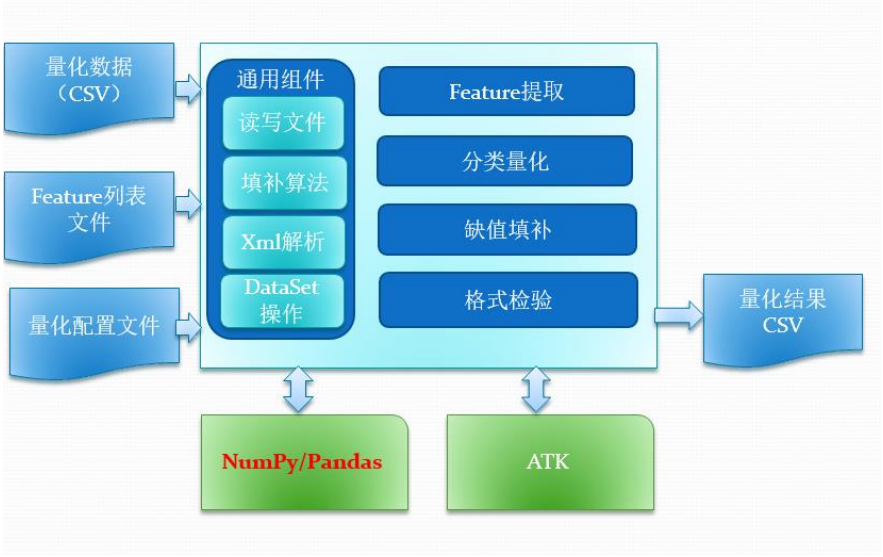

图 3-4 量化程序客户端的体系结构图

数据管理员通过 windows 应用程序对数据进行操作，视图利用 C#的布局控件进行界面设计，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。

3.2.4 TAP 平台架构

本系统的 TAP 平台的体系结构如图 3-5 所示，TAP 平台是一个简化了图像分析和机器学习的云计算平台。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只是挖掘，TAP 平台使用一个可扩展、模块化的框架来为图形建构，图像分析以及机器学习提供一个分析管道，通过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的一些预测

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。

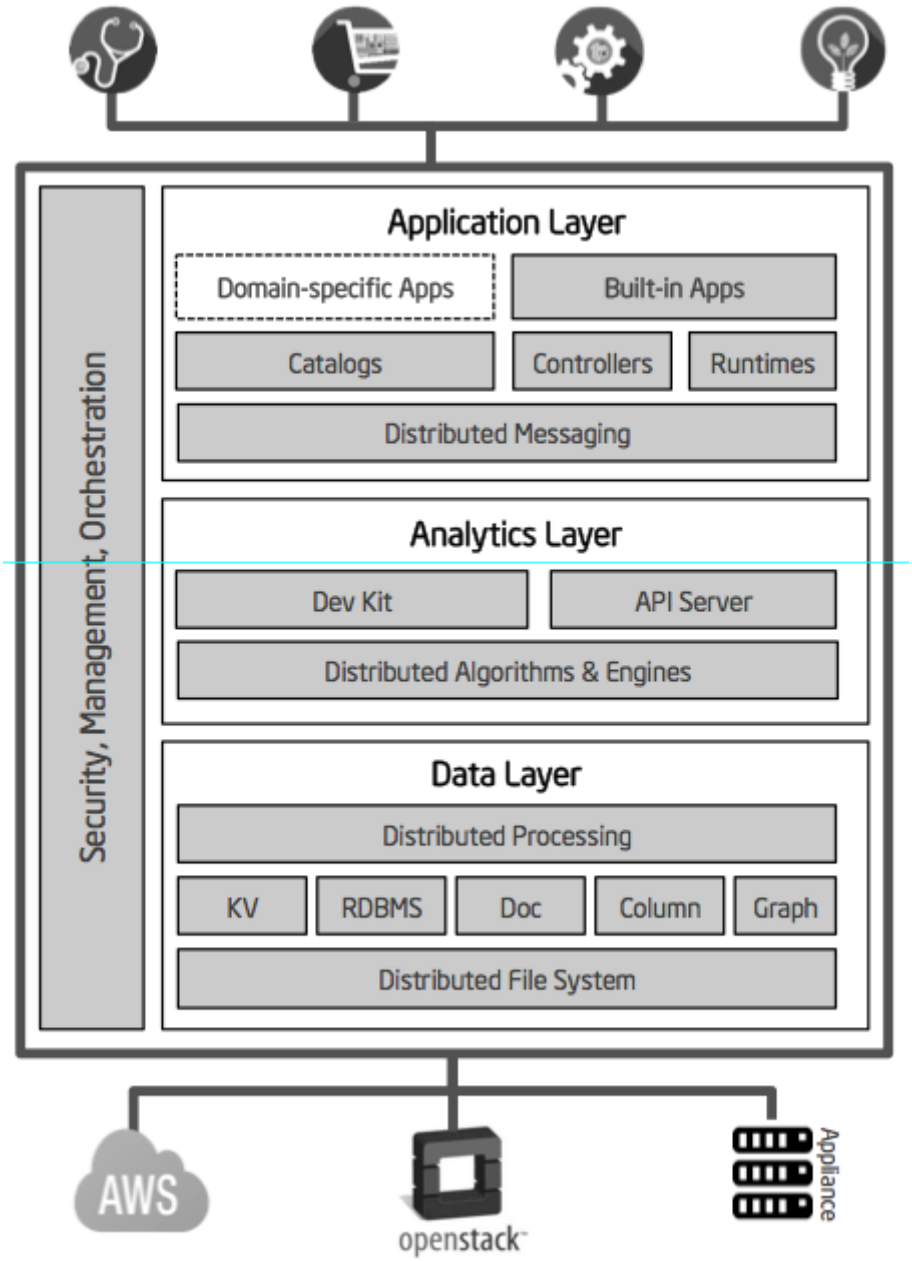

图 3-5 TAP 平台的体系结构图

3.3 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框架，基于 HTTP 协议，实现客户端与服务端通信，并通过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，技术每时每日都在进行更新，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，这些都成为了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. 本系统采用了最新的云计算技术，计算资源以及存储容量无上限。
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. 服务器的处理速度快、响应时间短，性能稳定。

3.4 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，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，通过这些需求进而引出高血压风险预测系统的总体设计，以及开发该系统大概要使用到的技术和工具。

需求决定设计，正是本章提到的这些需求才决定我们的高血压风险预测系统是如此设计，同时也只有这样的设计才能满足这么多复杂的需求。

本章从需求出发得出的总体设计为下一章系统设计奠定基础，另外各详细需求说明也为系统设计中的模块设计提供依据。

第4章 高血压风险预测系统的设计

4.1 数据模型的选择

经过前面的介绍，现在本章将要进行高血压风险预测系统更详细的设计，在介绍具体的设计之初，先介绍下高血压风险预测系统所采用的数据模型。

在本系统之中数据模型的选择是重中之重，在没有形成自己独创的数据模型之前，本系统使用的是现如今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结论，这些数据模型各有优点各有不同，下面是最知名的几种数据模型。如表 4-1 所示。

表 4-1 国内外知名算法优缺点对比表

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，在数据上一点点细小的变动都会导致完全不同的分裂；预测面会影响回归的效果
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；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更大的影响
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	应保证前几个主成分的累计贡献率达到一个较高的水平，其次对主成分必须能给出解释；主成分的因子有正有负时，函数意义就不太明确了。

名称	优点	缺点
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 的算法	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；

上面的表格中只列出了部分优秀数据模型算法，更多的没有列出，这些算法各有优点，各有劣势。从上面的图表中可以看出，分类树算法虽然直观，并且计算量不大，但是较小的数据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，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，如果用户每一次预测的结果都不相同，那么这个系统将会是非常不可信的，也就不会有任何用户会使用它。

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，但是必需要为前几个主要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。从这些特点出发，本系统所采用的字段指标是采集于人体的健康参数，其关联性是非常大的，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进行非常精心的挑选的，所以并不需要将这些变量变换成不相关的几个主成分，而且最重要的是系统开发者不容易为这些主要成分找到实际的意义，更不容易人用户接受这些意义。所以主成分分析方法也不适合高血压风险系统的使用。

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。但是该算法的计算复杂度取决于支持向量的数目，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，但是支

持向量机算法模型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。

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 的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。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。鉴于此，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。

4.2 随机森林算法原理简述

随机森林算法有着在大的、高维数据训练时，不容易出现过拟合而且速度较快的一些优点，而且对训练数据中的噪声和错误鲁棒等等。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。

随机森林算法是机器学习等一些领域应用最为广泛的一种算法，该算法不仅仅可以用来分类，还可以用来做回归预测。随机森林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，不容易出现决策树的过度拟合现象。

决策树是组成随机森林的基础，在介绍随机森林算法之前先简单的较少下决策树的原理。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。

4.3 高血压风险预测系统的设计

根据第 3 章的概要设计，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。

接下来的小节将对这四个部分的设计进行详细的描述。

4.4 功能模块设计

系统的主要工作流程如图 4-1 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 云计算平台界面，将数据上传到 TAP 平台之中，具体的是 TAP

中的 hdfs 之中，然后预测观察员通过 TAP 对数据进行管理，之后创建运行实例，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。接下来的章节将主要详细介绍数据处理模块、数据管理模块、运行实例管理模块、高血压预测模块。

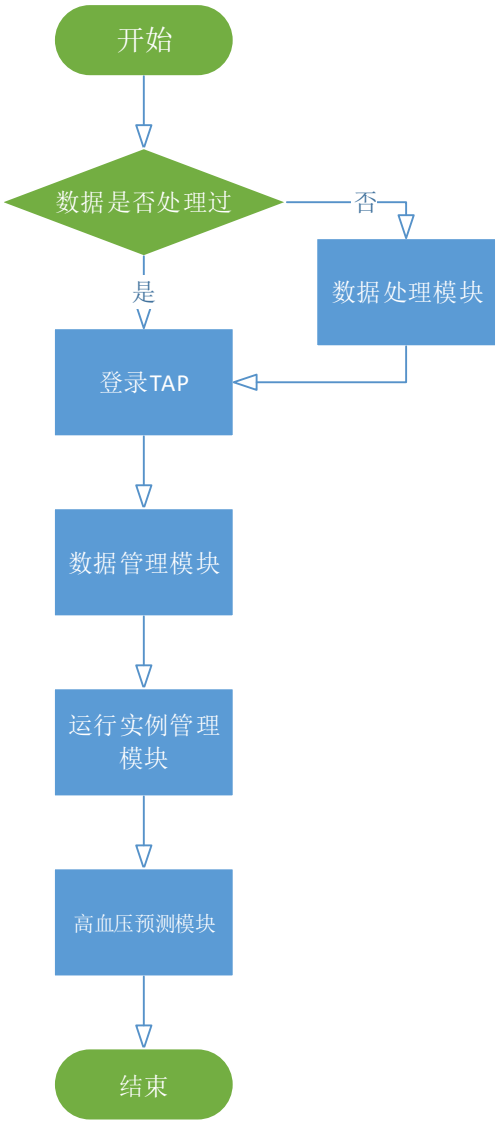

图 4-1 系统的主要工作流程图

4.4.1 数据处理模块的设计

数据处理模块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 4-2 。

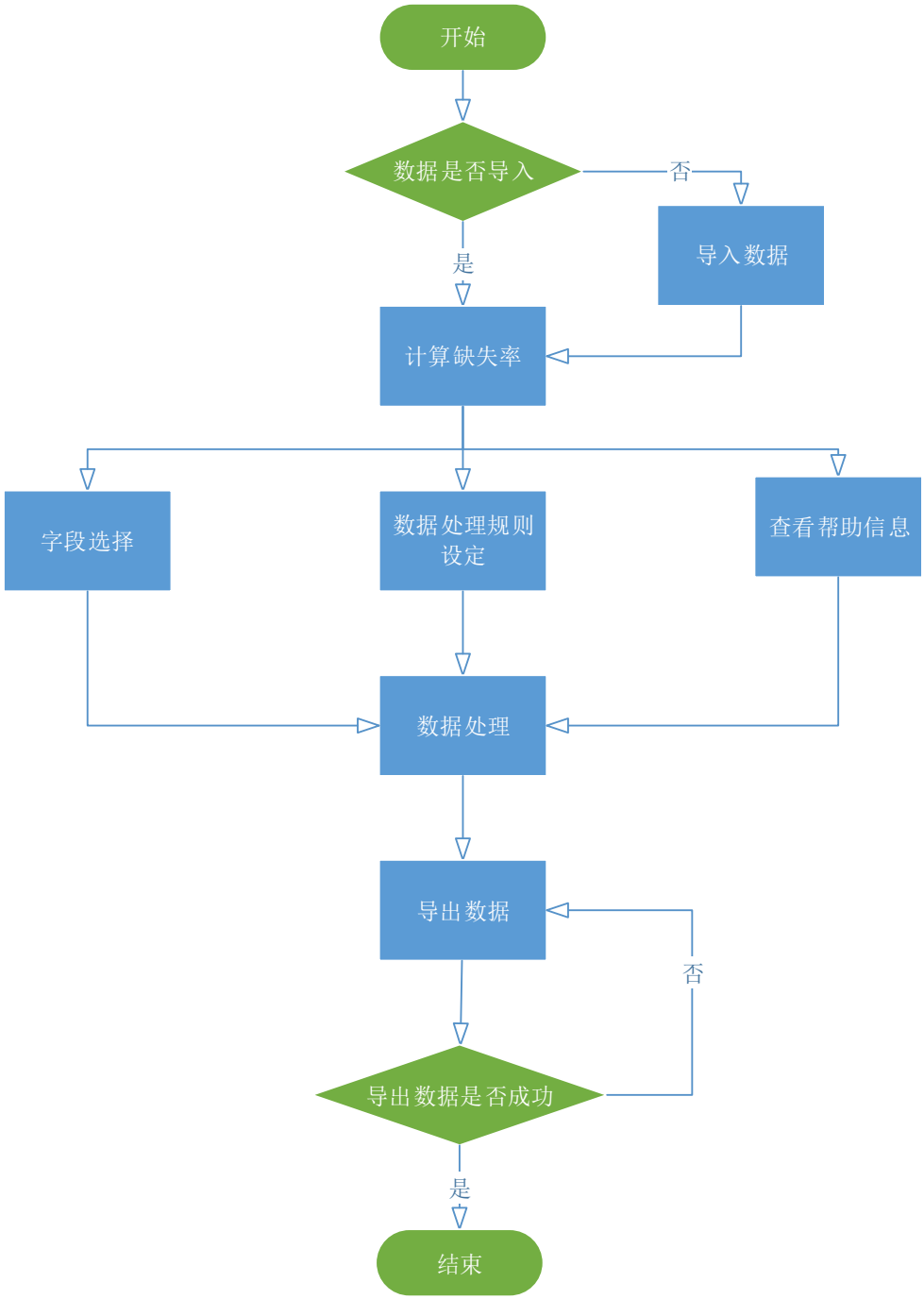

图 4-2 数据处理模块工作流程图

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个字段的缺失率情况，之后也会可以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，以使数据符合用户的要求，最后用户还可以将处理好的数据导出为用户需要的可是，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。

4.4.2 数据管理模块的设计

数据管理模块是 TAP 平台很重要的一部分，在 TAP 平台中数据不仅使用了 Python 数据高级处理的的一些技术，同时云平台支持分布式存储，同时支持集群计算，这也是为什么用户需要将数据上传到 TAP 平台中之后才拿来使用，TAP 平台提供了 web 浏览器界面，在网页上管理数据十分简单。该模块工作流程如图 4-3 。

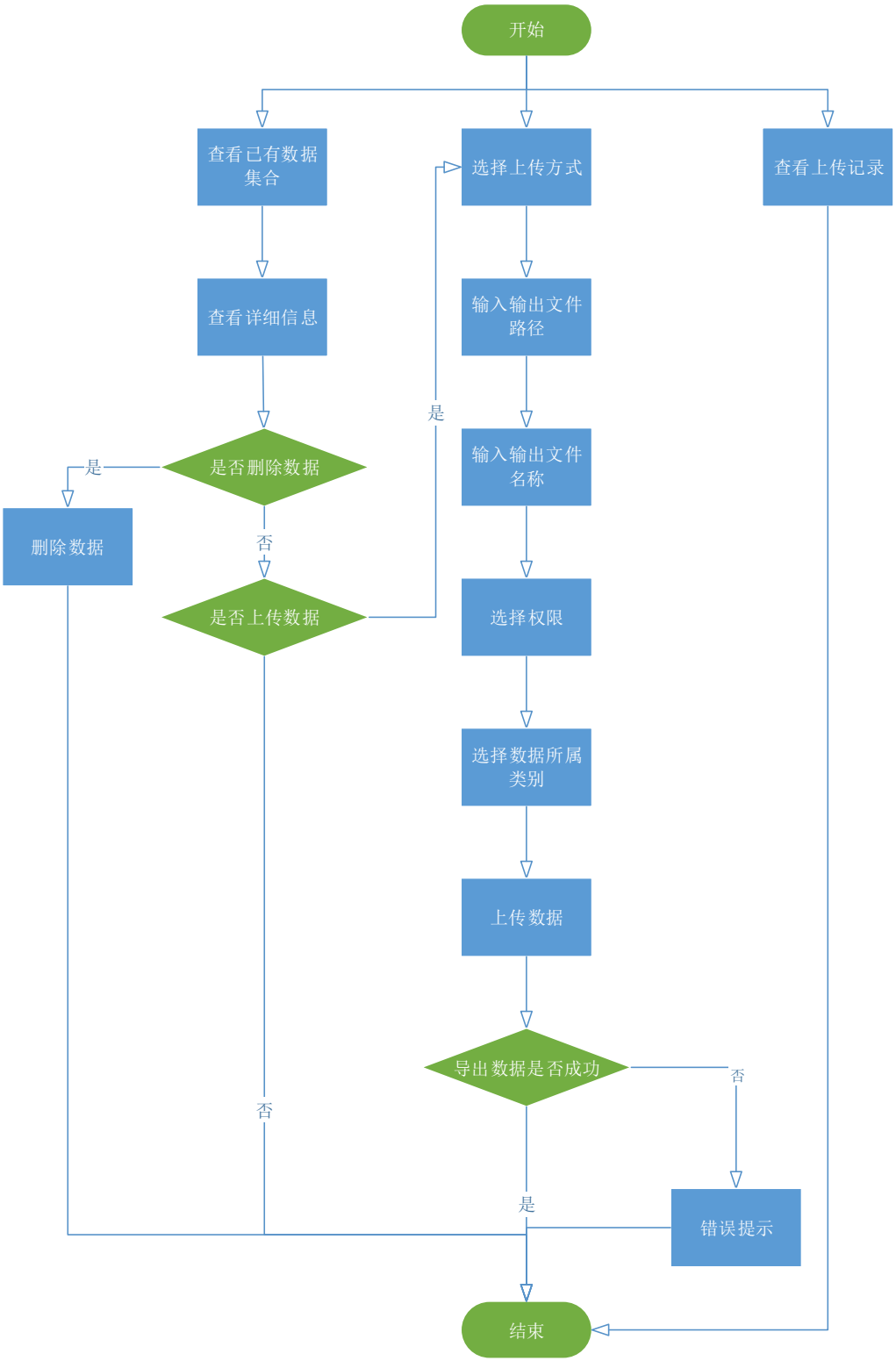

图 4-3 数据管理模块工作流程图

从图中可以看出，用户可以查看以及上传的数据集合，还能查看上传的历史记录，在上传数据的时候可以选择不同的上传方式，可以为数据设置不同的权限，同时也可以为数据进行类别的分类，便于管理。除此之外即使上传数据失败之后系统也会有错误提示，交互方式友好，操作简单易懂。

4.4.3 运行实例管理模块的设计

运行实例管理模块是高血压风险预测系统提供服务的关键。当用户在 TAP 平台上进行运行时首先需要的是要有一个实例，云计算平台上的实例就相当于我们传统工作中的物理机，没有实例就没有相应的运行计算资源，只有在成功创建实例之后才有可能运行我们自己的应用。该模块工作流程如图 4-4 。

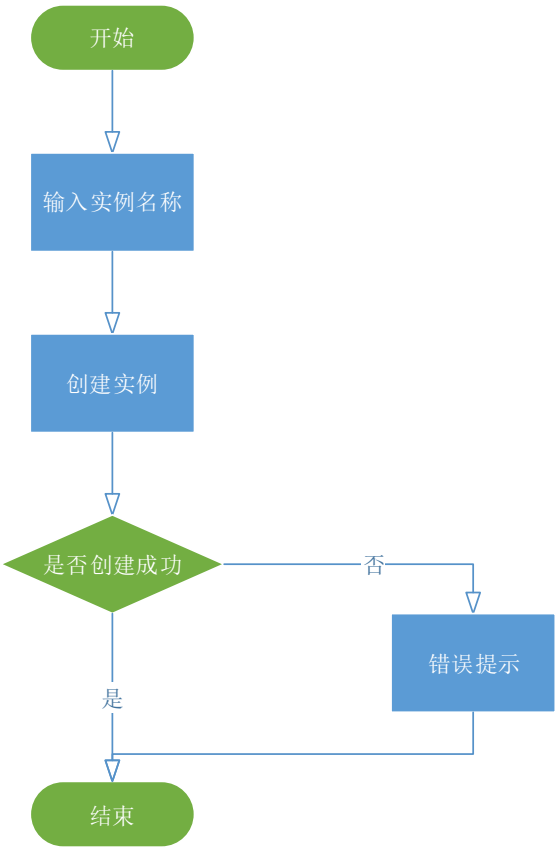

图 4-4 运行实例管理模块工作流程图

从图中可看出，运行实例管理模块的流程相对比较简单，这是因为复杂的

创建工作都有 TAP 平台在后台帮用户完成了。从流程图中可以看出，只要用户输入要创建实例的名称，点击创建，实例就创建成功了，相当简单。

4.4.4 高血压预测模块的设计

高血压风险预测系统最核心的工作都是由高血压预测模块完成的，该模块是这个系统的核心环节，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 4-5 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。

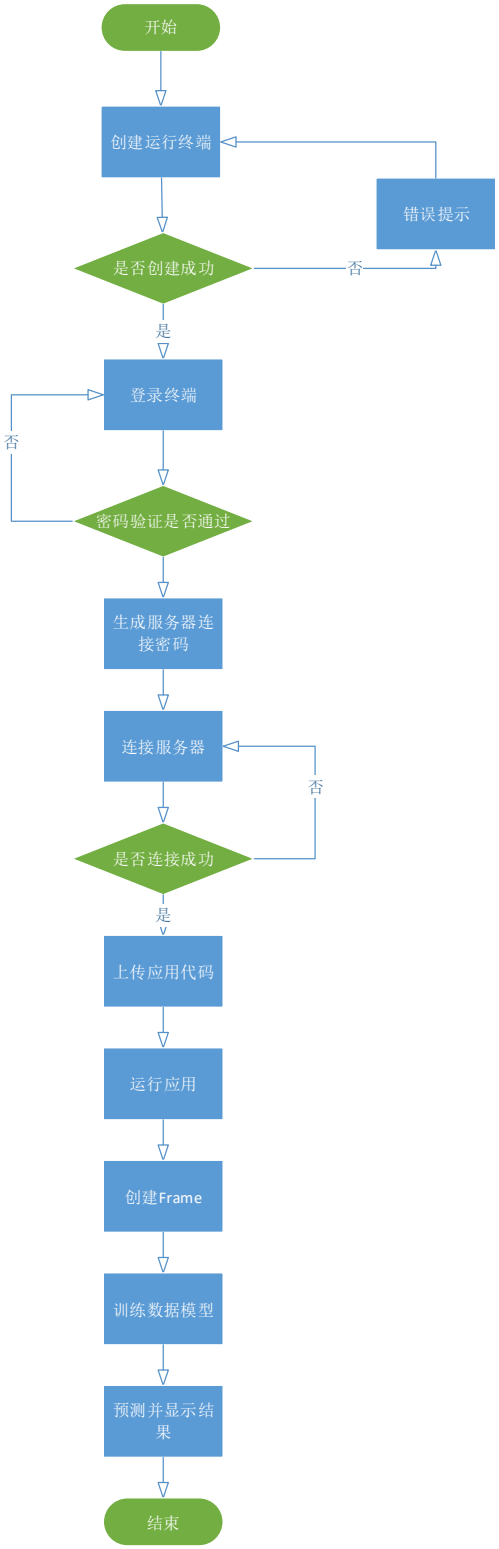

图 4-5 高血压预测模块工作流程图

4.5 数据处理规则设计

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，所以这里我着重写一下。

本系统的数据处理规则主要是针对实际情况使用 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。

4.5.1 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 4-6 所表示的情况。

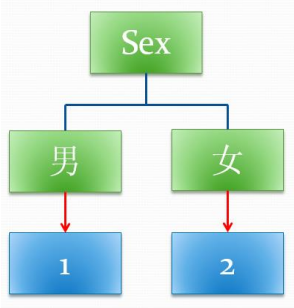

图 4-6 Sex 字段分析图

```
<Sex>
  <男> 1</男>
  <女> 2</女>
</Sex>
```

图 4-7 Sex 字段规则设计图

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 4-7 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。

4.5.2 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、底等，并用数字进行表示，比如下面这种情况，如图 4-8 所示。

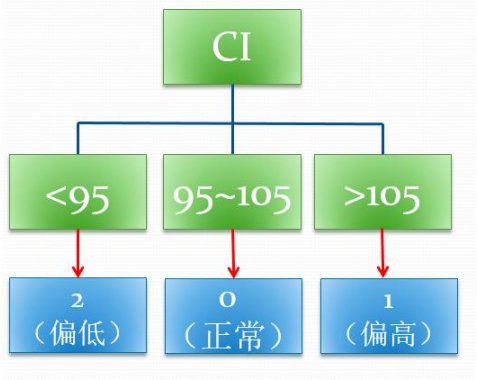

图 4-8 CI 字段分析图

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

图 4-9 CI 字段规则设计图

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 4-9 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。

4.5.3 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，同一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。比如如下图 4-10 所示的情形。

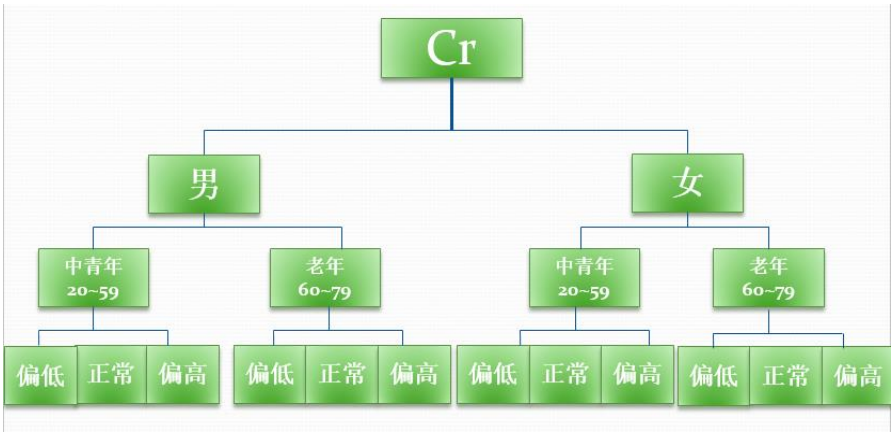

图 4-10 Cr 字段分析图

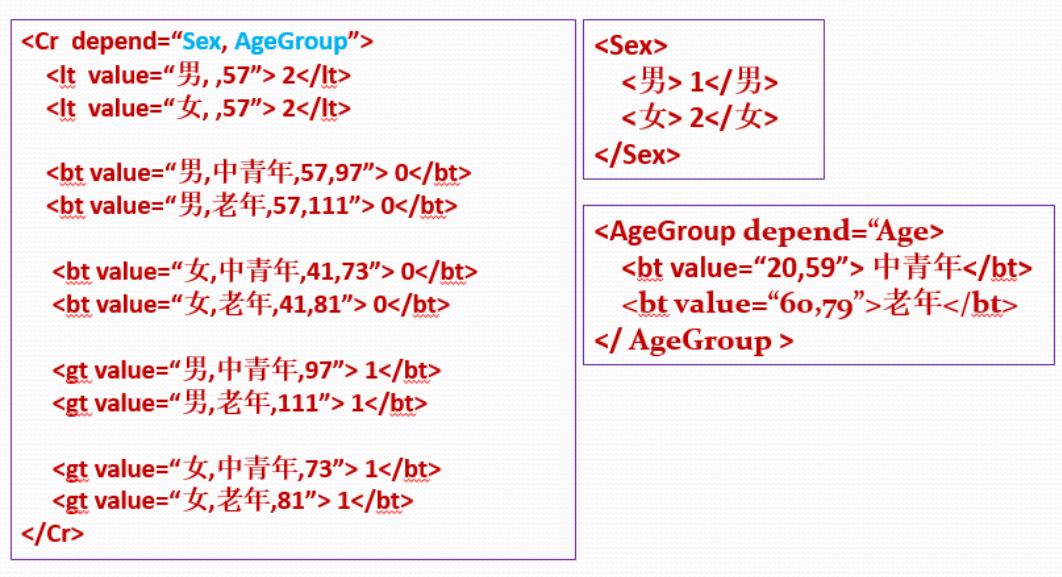

图 4-11 Cr 字段规则设计图

从图 4-10 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 4-11 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。

4.6 本章小结

本章主要描述了高血压风险预测系统的模块设计，这是在第三章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，高血压预测模块将模型进行训练，从而拿来预测患者的高血压风险。

本章通过介绍高血压风险预测系统的不同模块的运作流程来向大家展示该系统的工作原理，具体实现将在下一章进行详细的介绍。

第5章　高血压风险预测系统的实现

高血压风险预测系统主要分为数据处理模块、数据管理模块、运行实例管理模块、高血压预测模块这四个模块构成，这几个模块中除了数据处理模块运行于 TAP 平台之外，其他的几个模块都是运行在 TAP 平台之中，而数据管理模块和运行实例管理模块作为 TAP 平台的一部分我们在这里就不再详细讲述，本章主要向大家介绍数据处理模块和高血压预测模块的具体实现过程，以使大家有更深一步的了解。

5.1 TAP 平台的搭建与实现

本论文高血压风险预测程序是针对基于云计算平台 TAP 平台而设计的，其实现的开发环境的搭建也是基于此搭建的，主要包括硬件环境和软件环境。

1．硬件环境

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 5-1 表 5-1 高血压风险预测系统硬件开发环境。

表 5-1 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
4	RJ45 网线	数根
5	鼠标	1 个
6	键盘	1 个

7	交换机	1 个

2．软件环境

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 5-2。

表 5-2 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

3. TAP 平台的部署

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，使其能够互相联通，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。

5.2 数据处理模块程序的实现

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 5-3 所示。

表 5-3 Form 类部分方法列表

	名称	说明
	Activate()	激活窗体并给予它焦点。
	AdjustFormScrollbars(Boolean)	根据当前控件位置和当前所选控件调整容器中的滚动条。
	Close()	关闭窗体。
	Dispose()	对 Component 使用的所有资源进行释放。
	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 5-3 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 5-4 Form 类部分事件列表

	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。
	Enter	发生进入控件时。
	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。

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 5-1 所示的数据处理模块的主页面。

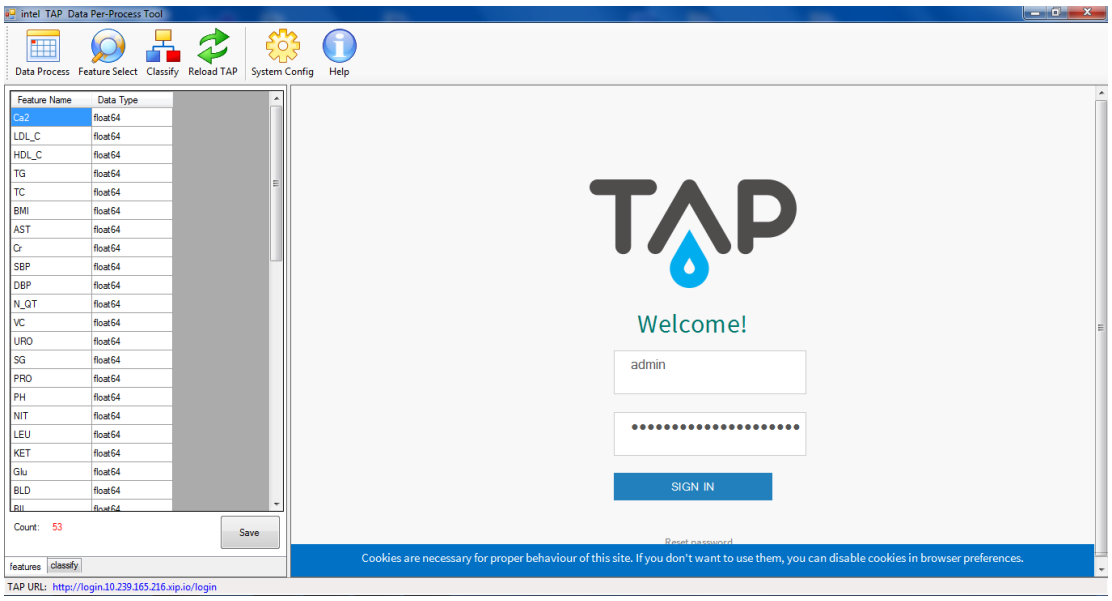

图 5-1 数据处理模块主页面图

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就进行数据处理的话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 5-2 所示。

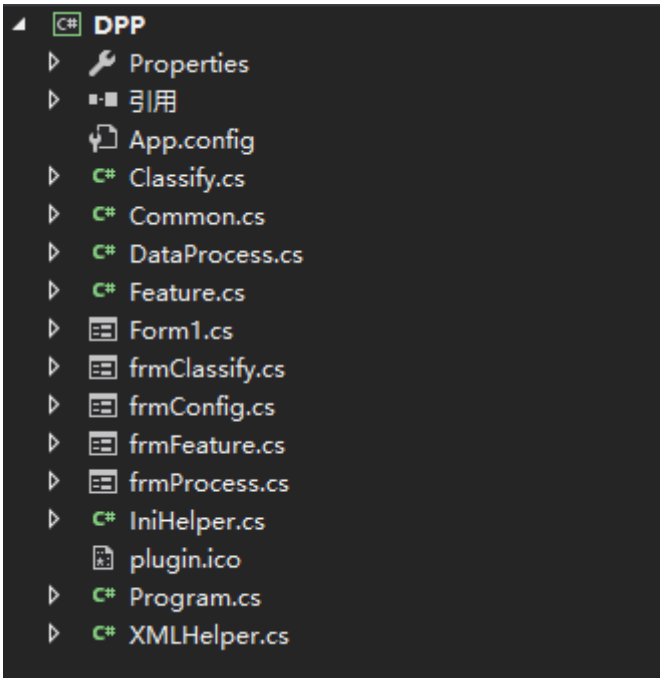

图 5-2 数据处理模块项目结构图

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 5-3 所示的类图进行展示。

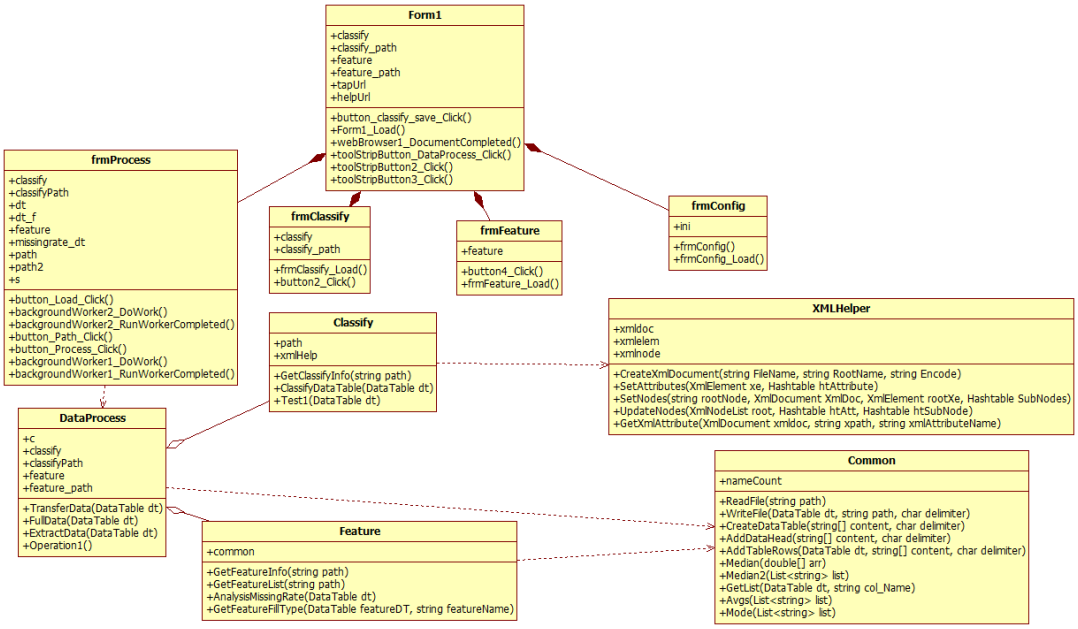

图 5-3 数据处理模块类图

图 5-3 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 From1 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 这样一些熟悉，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 5-4 所示，图 5-4 就是数据处理的窗口。

图 5-4 数据处理窗口图

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，这么多的熟悉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 5-5 展示的这样。

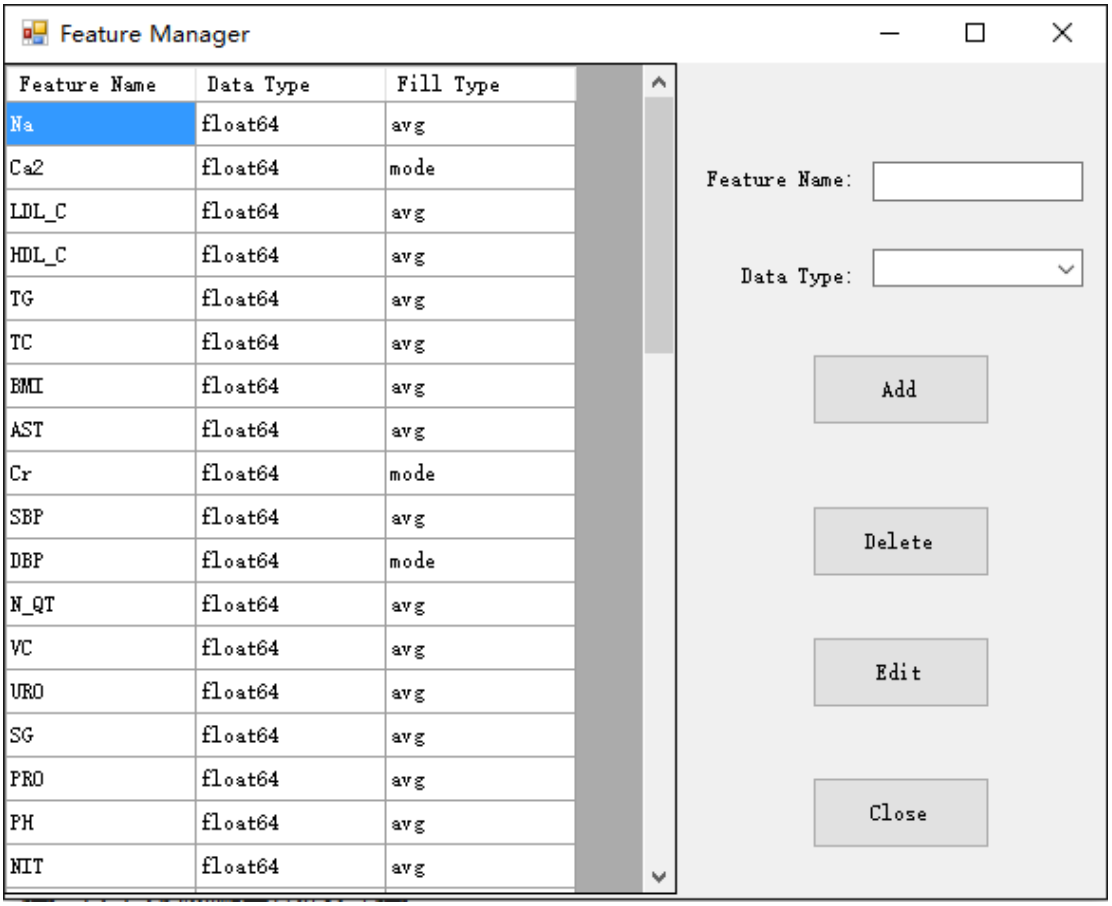

图 5-5 feature 选择窗口图

frmClassify 类只有 feature 熟悉和 frmFeature_Load()还有 button4_Click()等这几个方法，通过这些熟悉和方法对 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 5-6 所示。

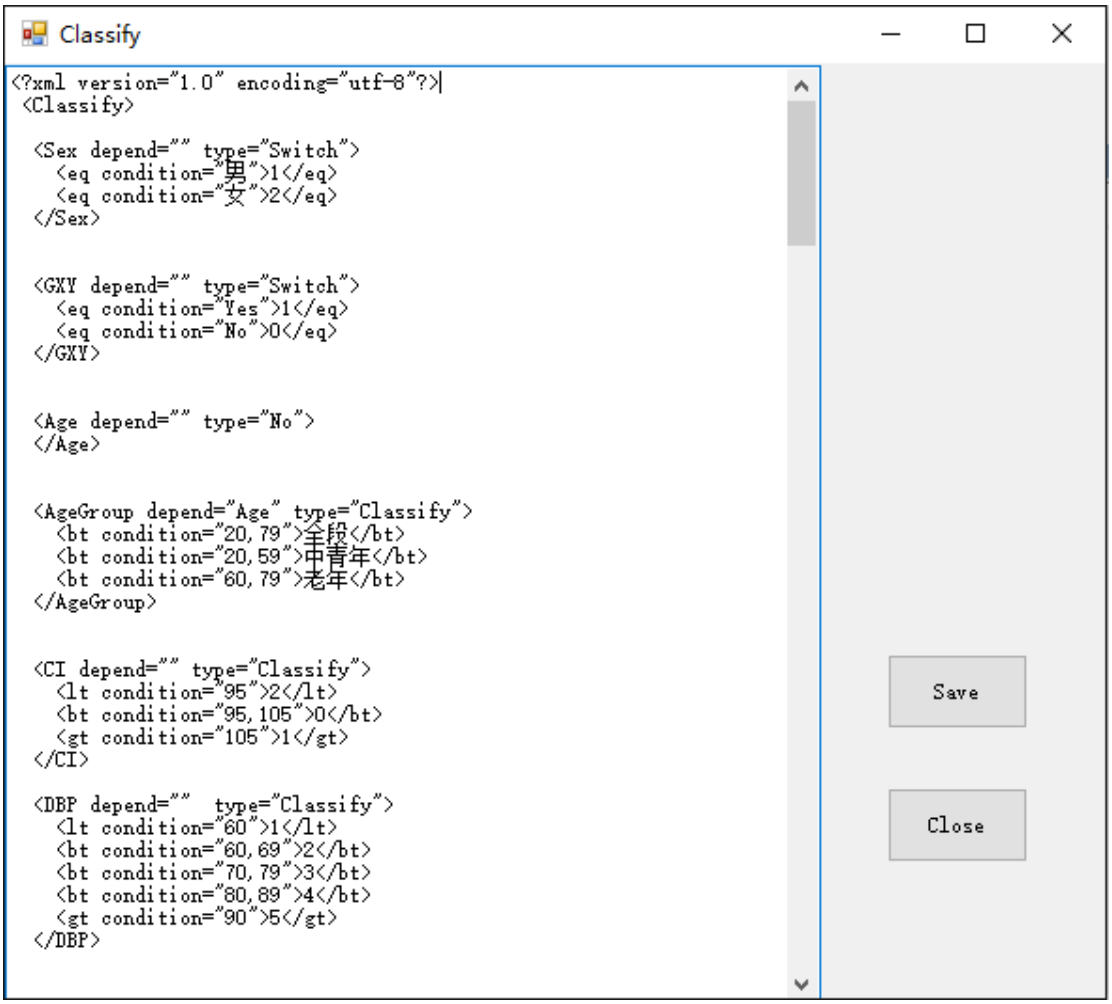

图 5-6 数据处理规则编写窗口图

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。

在介绍过数据处理模块程序主要的一些类及其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。

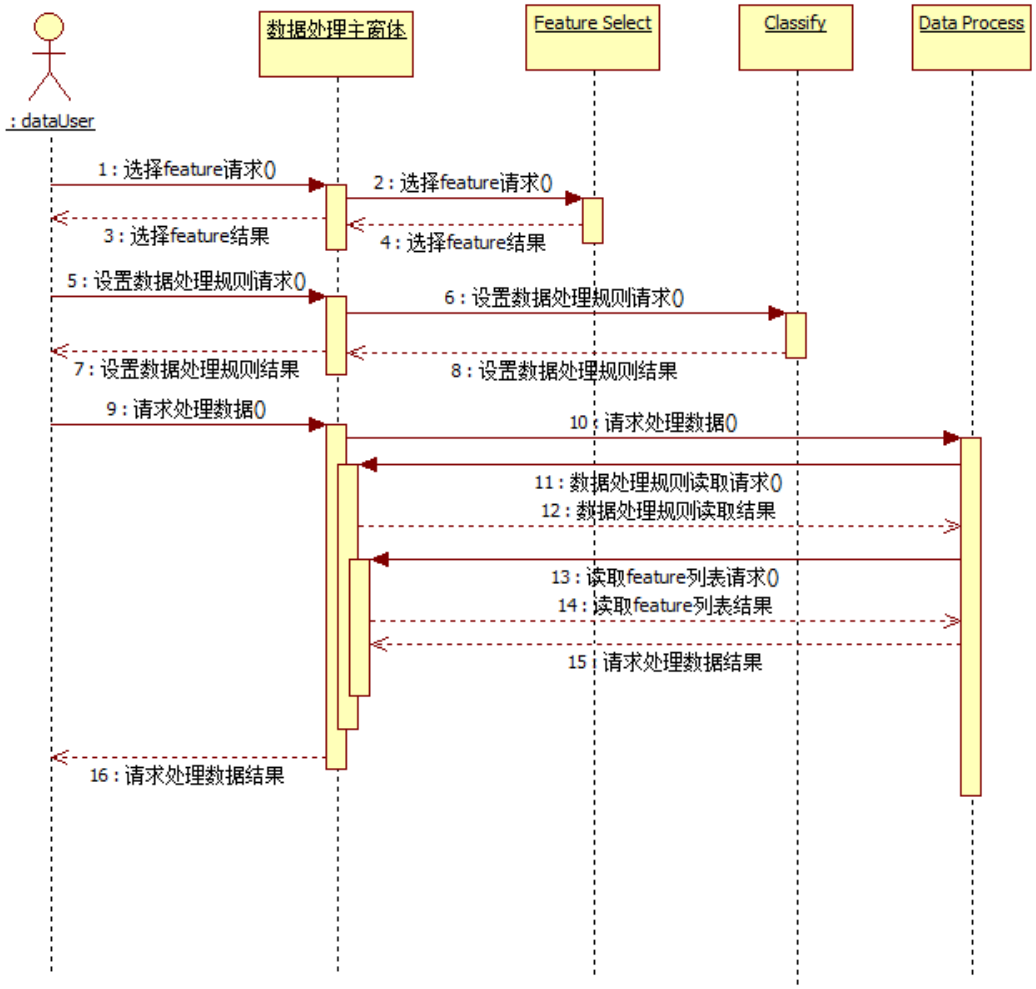

图 5-7 数据处理模块时序图

在图 5-7 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据值。

5.3 高血压风险预测应用程序的实现

通过上面的数据处理模块的处理，我们已经得到字段值满足实际要求，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查找用于模型训练和预测的数据。在图 5-8 所示的界面上传我们处理过的数据。

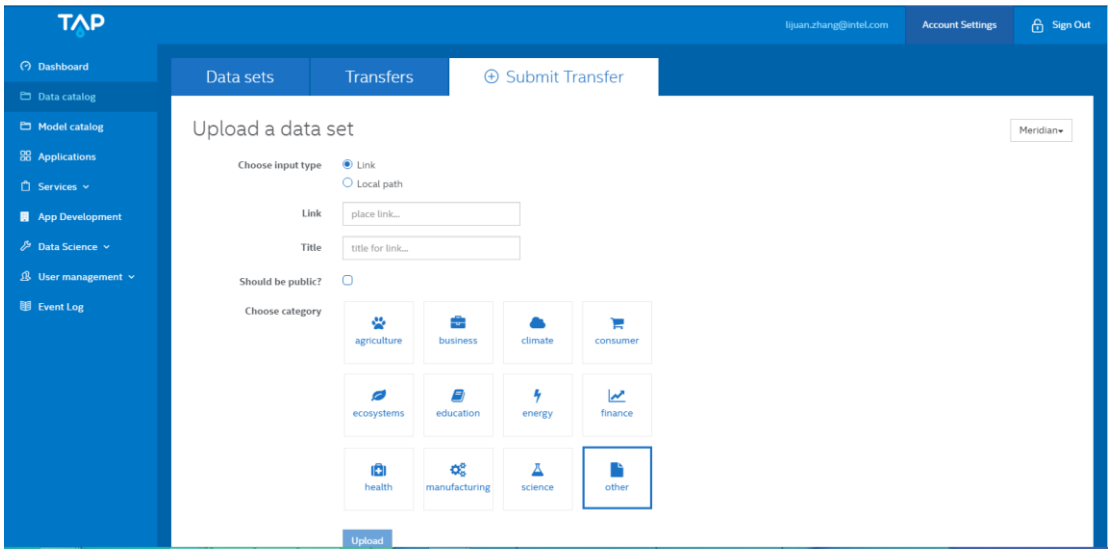

图 5-8 数据管理模块界面图

如图 5-8 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、风险预测等模块，而是需要先通过运行实例管理模块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 5-9 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。

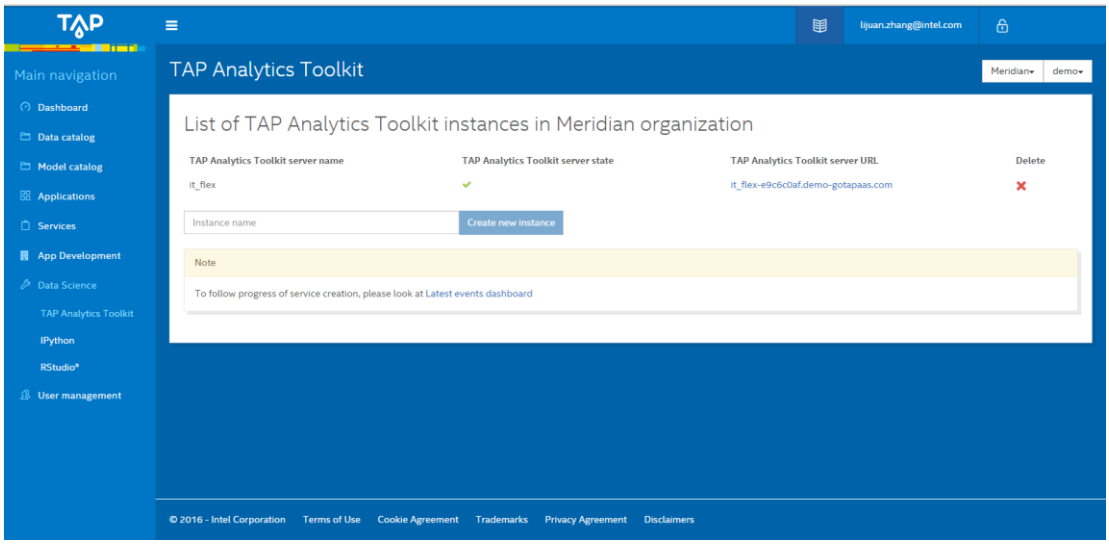

图 5-9 运行实例创建界面图

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。

5.3.1 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 5-10 创建证书文件图

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。

5.3.2 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，所有用的是 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/275ed74a-df39-4ff0-b3b9-049acccf904e/1178fa5c-6d05-47ca-9ed7-488786b21934/000000_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那些实现选择出来的 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。

5.3.3 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。

5.3.4 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

RandomForestClassifierModel(name=model_name)	创建新的 model
predict(self, frame[, observation_columns])	进行数据预测
test(self,frame,label_column[, observation_columns])	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
train(self,frame,label_column, observation_columns[, num_classes, ...])	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_prodict,'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 5-11 所示。

图 5-11 随机森林算法预测结果图

从图 5-11 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_prodict)

frame_prodict 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。

5.4 本章小结

本章主要介绍的是高血压风险预测系统的实现过程，首先简要的介绍了 TAP 平台的搭建过程。其次一点点的介绍了高血压风险预测系统各个模块的实现过程，重点介绍了数据处理模块和高血压风险预测模块的实现过程，通过类图，时序图等 UML 工具来详细的阐述模块内部的运作原理。

本章系统性、着重性的介绍了整个高血压风险预测系统的实现过程，深刻的揭露出其实现原理，使整个系统一目了然。为下一章的系统测试做准备，也为测试中将会遇到的问题提供原理性的参考。

第6章 测试与分析

前面的章节介绍了高血压风险预测系统从需求到设计在到实现的这个过程，而本章将从前几章的基础出发对高血压风险预测系统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，为以后进一步发展高血压风险预测系统提供依据。

高血压风险预测系统使用在医疗健康体检机构采集到的最近导出的一万条数据进行本系统的测试，最后分析测试结果并做出总结。

6.1 测试方案

测试的目的

第四章和第五章中分别从高血压风险预测系统的设计和实现两个角度，对本系统的原理进行阐述，当实现出程序并未经过检验，只在理论上论证其可行性。本章的测试就是要通过实际的数据，系统的检测，以量化的数据来检验本高血压风险预测系统的准确性，优越性。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。

测试的方案

本测试将通过需求测试、功能测试、兼容性测试、性能测试以及回归测试系统的对高血压风险预测系统的方方面面进行测试。需求测试主要是对提出高血压风险预测系统的需求进行明确和区分，测试这些需求的清晰度和相关度。功能测试主要是对高血压风险预测系统程序实现的个个功能进行测试，测试个功能是否正常。兼容性测试是通过将高血压风险预测系统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。最后进行的回归测试则是对高血压风险预测系统程序修改过的 bug 进行重新测试，确保程序执行的准确性。

6.2 测试描述

6.2.1 测试需求描述

表 6-1 描述了高血压风险预测系统各模块测试的优先级和测试的项目。

表 6-1 测试需求描述表

模块与功能		优先级别	需求测试	功能测试	兼容性测试	性能测试	回归测试
数据处理模块	Feature 选择	高	√	√	√	√	√
	数据处理规则设置						
	数据处理						
数据管理模块	上传数据	中	×	√	×	√	√
	删除数据	低	×	√	×	√	√
运行实例管理模块	创建运行实例	中	×	√	×	√	√
	删除运行实例	中	×	√	×	√	√
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，比如其中的需求测试和兼容性测试就不是每个模块都需要进行的。

- 数据处理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每一项的测试。
- 数据管理模块是属于 TAP 平台内置的模块，则对该模块就不需要进行需求测试和兼容性测试了，但是对该模块的功能测试和性能测试已经回归测试都是必需要进行的。
- 运行实例管理模块和数据管理模块一样都是属于 TAP 平台内置的模块，所以也不需要进行需求测试和兼容性测试，只对其功能和性能进行测试，最后做必要的回归测试。
- 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。

6.2.2 测试成员及任务分配说明

高血压风险预测系统的测试工作主要由三名人员进行，其各自的分工如表 6-2 所示。

表 6-2 测试任务分配表

职位	测试成员	测试任务
测试组长	测试组长	测试计划及测试方案的制定；测试资源的管理与任务分配；参与功能测试、性能测试的执行；质量分析与质量保证；测试过程监督与管理
测试组员	组员 A	进行数据处理模块的所有测试项目和数据管理模块和运行实例管理模块的性能测试、功能测试以及回归测试
测试组员	组员 B	进行高血压预测模块的所有测试项目并完成整个测试报告

6.3 测试执行情况

下面从需求测试开始进行对每个模块进行测试，知道完成功能测试、、兼容性测试、性能测试和回归测试整个过程。

6.3.1 需求测试

需求测试以需求文档学习，需求讨论会议等的形式进行的，由开发人员介绍，测试组长详述项目需求，经过组员热烈讨论，最后交由测试组织总结，使所有组员对高血压风险预测项目的需求都清晰明了。

6.3.2 功能测试

功能测试采用黑盒测试的方式以测试人员手工的进行，对每一个模块每一个功能至少测试三次，每个模块至少进行一次回归。

- 数据处理模块

表 6-3 是对数据处理模块功能测试的描述说明。

表 6-3 数据处理模块测试说明表

测试目标	Feature 选择功能的准确 数据处理规则设置功能的正确 数据处理功能的正确
完成标准	所计划的测试全部执行。 所有缺陷全部解决。 数据处理模块能够正常使用并能得到预期结果。
完成情况	数据处理模块的测试过程中发现了 5 处缺陷，5 处缺陷修复完成测试通过，所有测试计划通过，结果满足预期。

- 数据管理模块

表 6-4 是对数据管理模块功能测试的描述说明。

表 6-4 数据管理模块测试说明表

测试目标	上传数据功能的正确 删除数据功能的正确
完成标准	所计划的测试全部执行 所发现的缺陷全部解决
完成情况	在数据管理模块的测试过程中，没有发现任何缺陷，所有功能正常。

- 运行实例管理模块

表 6-5 是对运行实例管理模块功能测试的描述说明。

表 6-5 运行实例管理模块测试说明表

测试目标	创建运行实例功能的正确 删除运行实例功能的正确
完成标准	所计划的测试全部执行 所发现的缺陷全部解决
完成情况	在运行实例管理模块的测试过程中，没有发现任何缺陷，所有功能正常。

- 高血压预测模块

表 6-6 是对高血压预测模块功能测试的描述说明。

表 6-6 高血压预测模块测试说明表

测试目标	连接 TAP 平台并导入数据源功能的正确 创建 Frame 功能的正确 模型训练功能的正确 数据预测功能的正确
完成标准	所计划的测试全部执行 所发现的缺陷全部解决
完成情况	在高血压预测模块的测试过程中，共发现 2 处缺陷，缺陷已修复正常，所有功能正常。

6.3.3 兼容性测试

兼容性测试分主要分为两个部分，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 6-7 描述了兼容性测试的详细说明。

表 6-7 兼容性测试说明表

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

6.3.4 性能测试

高血压风险预测系统的性能测试熊执行效率和运行稳定性两个方面进行测试。在执行效率的测试上主要测试的是高血压风险预测系统对各种操作的响应时间，在这一点上高血压风险预测系统基本上是毫秒级的，只在数据预测时稍微比较忙需要用户等待几十秒，当在大数据量的前提下，用户完全可以接受。而在稳定性测试方面，高血压风险预测系统尚未发现有系统崩溃或者突然卡顿，以及预测结果发生突变等的不稳定情况，系统运行比较稳定。

6.3.5 回归测试

回归测试就是对测试过程中发生的执行错误的测试用例进行重新测试，直到高血压风险预测系统各模块的功能都通过验证。表 6-8 描述了游戏的回归测试

具体说明。

表 6-8 回归测试说明表

测试目标	高血压风险预测系统所有模块所有功能的验证测试
完成标准	高血压风险预测系统计划的所有测试计划都执行完成 所有的回归测试用例都执行并通过
完成情况	测试过程中发现的失败用例都已经成功执行并且通过，发现的缺陷都已经关闭

6.4 测试结果及评估

高血压风险预测系统在测试过程中发现缺陷较少，并且缺陷都已经关闭，足以看出该系统比较优秀稳定。下面将从功能性，可能性，性能以及对设备支持性四个方面对高血压风险预测系统的测试结果进行评估。

6.4.1 功能性评估

高血压风险预测系统所有模块的所有功能都通过测试，仅有的缺陷都已经关闭，回归测试也都完成并且通过。这足以见得高血压风险预测系统功能的稳定性。

6.4.2 可用性评估

在测试过程中高血压风险预测系统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。从这可以看出本系统可用性极高。

6.4.3 性能评估

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。

6.4.4 设备支持性评估

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。

6.5 结果分析

6.5.1 高血压风险预测系统的优势

从对高血压风险预测的结果上看，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 86%以上，并且在对模型训练数据的比例已经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。除此之外本系统操作比较简单，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。

6.5.2 高血压风险预测系统的不足

在高血压风险预测系统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，这一点也是需要在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。

6.6 本章小结

本章主要介绍了对高血压风险预测系统的测试情况，通过对高血压风险预测系统的需求测试、功能测试、性能测试、兼容性测试和回归测试五个方面进行测试，通过这五个方面的测试结果，对高血压风险预测系统的功能性，可能性，性能以及对设备支持性四个方面进行评估，最终对高血压风险预测系统的优势和劣势进行总结。

本章是对前几章设计和实现的高血压风险预测系统的验证，是对高血压风险预测系统能力的一种证明。

第7章 结 论

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。

7.1 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，要么从现今科技水平出发，所提出的高血压风险预测方案会因不同医师经验或者当前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。

本文介绍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，工具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 7-1 表所示。

表 7-1 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	对数据的处理对大数据支持性有缺陷

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。

7.2 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。

- 对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，但在不久的将来将会越来越精确，越来越优秀。

7.3 本章小结

本章主要对本论文进行总结，介绍高血压风险预测系统的优缺点，并在此基础上对本系统的未来进行展望，在分析高血压风险预测系统的过程中，突出本系统相对其他系统的优势之处，在对问题的解决过程中规划下本系统未来的

发展方向，高血压风险预测系统将更好的为用户提供服务。

参考文献

[1] 张新生. 云计算. 中国通信学会. 2006/03
[2] 中国 IDC 圈. 2016 年云计算领域发展趋势. 中国 IDC 圈. 2016/01/26
[3] 前瞻产业研究院. 2013-2017 年中国云计算产业市场前瞻与投资机会分析报告. 前瞻产业研究院. 2014/02/14
[4] 刘力生. 中国高血压防治指南 2010. 中国高血压防治指南修订委员会. 2011/08
[5] 刘冬. 基于遗传算法的 BP 网络在医疗诊断中的应用. 吉林大学. 2006/03
[6] 马里兰. 冠心病风险评估方法研究进展. 大理学院附属医院心内科. 2013/07
. 2012/04/12
[7] 张晔、蔡心轶. 科学生活：用“冷加压”预测高血压准吗？. 科技日报. 2012/05/28
[8] nizhidaolaiyou. 基于分类器集成技术的高血压预测与诊疗的研究. 资料网圈圈
[9] Bazilian Morgan D. Modelling of a photovoltaic heat recovery system and its role in a design decision support tool for building professionals. Renewable Energy. 2002
[10] Monique Frize, Colleen M. Clinical decision support systems for intensive care units:using artificial neural networks. Medical Engineering & Physics. 2001
[11] Valafar H, Valafar F. Data mining and knowledge discovery in proton nuclear magnetic resonance spectra using frequency to information transformation Knoeledge-Based Systems. 2002
[12] William G Baxt. Application of Artificial Neural Networks to clinical Medicine Lancet. 1995
[13] Zheng L, Sun Z, Zhang X, Li J. Framingham 高血压风险预测模型在中国农村人口中的预测价值. 中国高血压杂志.2014/03
[14] 孙艳秋，刘钢. 基于大数据分析的潜在高血压病预测研究. 辽宁中医药大学信息工程学院，沈阳师范大学教育技术学院. 2014/12/05
[15] 李现文, 李春玉. 决策树与 Logistic 回归在高血压患者健康素养预测中的应用. 美国 John Hopkins 大学. 2012/01/19
[16] 王重建, 李玉倩. 人工神经网络在个体患原发性高血压预测中的应用. 郑州大学公共卫生学院流行病与卫生统计学系. 2010/12
[17] Rajiv Agarwal, MD, Allen R. Prevalence,Treatment,and Control of Hypertension in Chronic Hemodialysis Patients in the United States. Excerpta Medical Inc. 2003/09
[18] Jie Su. Data Mining Based Evaluation Method for Hypertension Disease. Zhejiang University,Hangzhou,P.R.China. 2006/05
[19] 杨洋. 利用人工神经网络模型预测原发性高血压的研究. 中国医科大学. 2010/05/04
[20] 赵秀丽，胡大一. 中国 14 省高血压现状的流行病学研究. 北京大学人民医院. 2006/04/25
[21] 程遥，万隧人. 基于 BP 神经网络的高血压诊疗预测分析. 东南大学生物医学工程学院. 2014/03
[22] Li Bo. A study of hypertension epidemiological characteristics and influence factors among

Chinese adult. Huazhong University of Science & Technology. 2014/05
[23] Lixuan Gui. Genetic risk score predicts coronary heart disease risk in a Chinese Han population. Huazhong University of Science & Technology. 2014/01
[24] 晁灵. 分类树模型与 Logistic 回归在儿童高血压预测中的应用. 新乡医学院公共卫生学院. 2015/01/15
[25] MA Liang-Liang,TIAN FU-Peng. Application of time series analysis in the prediction of hypertension incidence. School of Computer and Information,Northwest University for Nationalities,Lanzhou. 2010/07
[26] 党红刚. 基于 ARIMAX 模型的海西州地区高血压月发病率预测. 天水师范学院数学与统计学院. 2011/06/11
[27] Mahmud Mavaahebi,Ken Nagasaka. A Network and Expert System Based Model for Measuring Business Effectiveness of Information Technology Investment. Department of Electronic Engineering,Tokyo University of Agriculture and Technology,Tokyo,Japan. 2012/11/20
[28] Qeethara Kadhim AI-Shayea. Artifical Neural Networks in Medical Diagnosis. MIS Department, AI-Zaytoonah University of Jordan Amman,Jordan. 2011/03
[29] 俞浩，郭志荣. 代谢综合评分与 Framingham 风险评分预测心血管疾病的比较. 苏州大学放射医学与公共卫生学院. 2010/02
[30] 马亮亮. 基于 PCA_ARIMA 模型的高血压发病率预测. 攀枝花学院数学与计算机学院. 2012/12/12
[31] 张军跃. 高血压智能防控平台建设构想. 中日友好医院. 2015/03/01
[32] 马光志. 基于神经网络的高血压在线风险评估系统. 华中科技大学计算机学院. 2008/04/15
[33] 刘力生. 高血压研究四十年. 中国医学科学院. 2002/05/14
[34] 顾东风. 中国成年人高血压患病率_知晓率_治疗和控制状况. 中国医学院科学院中国协和医科大学. 2002/12/31
[35] 王文绢. 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. 中国预防医学科学院. 2001/08/29
[36] YU Jing,JU Bin. Application of Data Mining Technology in Regional Health Information Platform Chronic Disease Management. China Digital Medicine. 2012/09/24
[37] Liqiang Zheng. Validation and Construction a Hypertension Risk Prediction Model in Rural Areas of Fuxin Country with the High Incidence of Hypertension:Result from Liaoning Province. China Medical University. 2014/03
[38] 刘冰. 中国 35_45 岁人群高血压前期检出率及影响因素分析. 北京协和医学院. 2009/08/21
[39] 陈亦敏. 6830 名老年人体质指数_腰围_腰臀比与高血压的关系研究. 浙江中医结合杂志. 2013/11/12
[40] 王霄飞. 基于 OpenStack 构建私有云计算平台[D]. 华南理工大学.2012.
[41] 李知杰, 赵健飞. OpenStack 开源云计算平台[J]. 软件导刊.2012,/11/12
[42] Corradi A, Fanelli M, Foschini L. VM consolidation: A real case based on OpenStack

Cloud[J]. Future Generation Computer Systems. 2014

[43] Sefraoui, Omar, M. Aissaoui, and M. Eleuldj. "OpenStack: Toward an Open-source Solution for Cloud Computing." *International Journal of Computer Applications* .2012

[44] Wuhib, Fetahi, R. Stadler, and H. Lindgren. "Dynamic resource allocation with management objectives: implementation for an OpenStack cloud."*Network and Service Management* IEEE. 2012.

[45] Fifield, Tom, et al. "OpenStack Operations Guide." O'Reilly Media, Inc. 2014.

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

答辩后论文修改情况说明