

# 面向 DevOps 的政务大数据分析可视化系统

张冬松<sup>1,2</sup>, 胡秀云<sup>2</sup>, 鄢长安<sup>2</sup>, 毛凤翔<sup>2</sup>

(1.国防科技大学 计算机学院, 湖南 长沙 410073;

2.信阳学院 大数据与人工智能学院, 河南 信阳 464000)

**摘要:** 随着大数据技术的快速发展, 越来越多与政务大数据行业相关的项目正在规划和建立。当前政务大数据建设的重要方面之一, 就是要把政务大数据作为重要资源, 持续深入地挖掘数据价值, 为政务服务、行政监管和社会治理过程中的科学分析、预测和决策提供重要基础。这就对大数据分析及可视化系统建设提出了新的技术要求, 包括集成多种机器学习算法、集成多种可视化工具、支持快速迭代开发部署、支持交互式数据分析任务和可视化展示任务等。面向 DevOps 这一先进的持续集成和持续交付的平台, 提出一种大数据分析及可视化系统, 满足了上述政务大数据行业的新需求。通过政务大数据行业应用, 表明该系统设计合理, 可视化分析结果有效, 具备持续交付能力, 相关技术的集成具有很强的系统指导和应用价值。

**关键词:** 大数据; 分析; 可视化; 持续集成; 持续交付

中图分类号: TP391

文献标识码: A

文章编号: 1673-629X(2020)08-0001-07

doi: 10.3969/j.issn.1673-629X.2020.08.001

## Analysis Visualization System of Government Big Data for DevOps

ZHANG Dong-song<sup>1,2</sup>, HU Xiu-yun<sup>2</sup>, WU Chang-an<sup>2</sup>, MAO Feng-xiang<sup>2</sup>

(1.School of Computer, National University of Defense Technology, Changsha 410073, China;

2.School of Big Data and Artificial Intelligence, Xinyang College, Xinyang 464000, China)

**Abstract:** With the rapid development of big data technology, more and more projects related to the government big data industry are being planned and established. One of the important aspects of current government big data construction is to use government big data as an important resource to continuously and deeply explore the value of data, and provide an important basis for scientific analysis, forecasting and decision-making in the process of government service, administrative supervision and social governance. This puts forward new technical requirements for the construction of big data analysis and visualization system, including integration of multiple machine learning algorithms, integration of multiple visualization tools, support for rapid iterative development deployment, support for interactive data analysis tasks and visual presentation tasks. We propose a big data analysis and visualization system for DevOps, an advanced platform for continuous integration and continuous delivery, to meet the new needs of the above-mentioned government big data industry. The application of government big data industry shows that the proposed system is reasonable in design, effective in visualization analysis results, capable of continuous delivery, which should have strong system guidance and application value for the integration of related technologies.

**Key words:** big data; analysis; visualization; continuous integration; continuous delivery

### 1 概述

大数据分析及可视化是近年来重新兴起的一门多领域交叉学科, 是一类从海量数据中自动分析获得规律, 并利用规律对未知数据进行预测, 同时提供可视化交互方式和展示界面, 在数据挖掘、电子商务、金融分析、政务管理等领域有着十分广泛的应用, 是当前各行各业的一个热点<sup>[1]</sup>。

从大数据行业应用角度来看, 当前大数据应用发展趋势正向着行业纵深层次稳步前行。以政务大数据为例, 用数据治理、用数据说话、用数据发展、用数据改进和用数据创新, 已成为政务大数据公认的价值体系所在。通过大数据分析及可视化技术, 将政务相关的数据整合起来进行有效分析, 提供立体可视化展示能力, 应用到政府业务领域, 赋能政府机构, 提升政务实

收稿日期: 2019-08-30

修回日期: 2019-12-31

基金项目: 国家自然科学基金(61402527)

作者简介: 张冬松(1980-), 男, 博士, 博士后, 通信作者, CCF 会员(41845M), 研究方向为大数据应用、自然语言处理和实时系统。

施效能。

当前,大数据分析可视化可以提供强大的数据读取功能、高效的数据处理方法、可视化的数据探索方式、丰富的数据挖掘算法、简易模型训练等功能,从易用性、功能范围、数据挖掘效果等方面全面超越了传统的分析方法。利用这一先进的大数据分析及可视化技术有助于帮助客户实现海量数据挖掘,利用一系列机器学习算法组合得到模型结果,经过可视化方式展示,以便更直观地从图表中发现数据背后的潜在价值。

政务大数据分析可视化系统是一项系统工程,涉及到数据存储、数据处理、模型建立、模型训练、分析应用、可视化展示等一系列功能模块。这些系统功能模块数量多、交互复杂。开发和运维这样的系统,出现问题是不可避免的。但是,如何快速发现并解决问题是关键。如果采取传统的软件开发模式,开发人员将软件产品开发完成后,交由测试人员进行测试,再由运维人员进行部署。通常只有在最后部署阶段才能发现问题,显然这种反馈的难度和时间成本都很大。

当前政务大数据行业应用的用户需求快速变化,高效自动部署技术是缩短大数据软件产品交付时间的关键。DevOps 技术<sup>[2]</sup>有助于实现快速自动化部署过程,通过不断迭代交付来缩短发布周期。它使用持续集成和持续交付的方式,一旦代码更新提交,就立刻得到反馈;如果发现问题,可以快速发现、定位并解决。这就极大降低了复杂软件快速频繁的更新与部署过程中产生问题的可能性,提升软件的质量与稳定性。

近年来,由于 DevOps 的构建、部署、运维等各阶段都有众多配套工具提供支持,它已经得到非常广泛的应用<sup>[3]</sup>。通常,软件开发工作会根据各自不同的实际情况,按需搭建适合自身的 DevOps 工具链。文献[4]提出了一种高效动态的应用部署引擎 OpsFlow,其目标在于组合多种自动化部署技术以应对异构应用与部署环境,从而实现高效自动化的微服务应用部署。但是,该 DevOps 解决方案仍是一种虚拟机部署方式,并不适合在不同云、不同版本操作系统间进行快速迁移。

当前,一种新的开发部署方式是采用 Docker 容器和镜像共享方式实现的<sup>[5]</sup>,每个容器有自己的运行环境和文件系统,容器之间不会相互影响。相比虚拟机部署方式,这种基于 Docker 容器技术的开发、测试和运维一站式 DevOps 解决方案,更适合云计算环境下跨平台、跨系统快速部署,已经得到广泛应用<sup>[6]</sup>。

因此,面对政务行业领域中的海量数据,文中提出了面向 DevOps 的政务大数据分析可视化系统,以事件为中心,综合各模态数据信息聚合和语义协同的结

果,挖掘跨平台、跨模态信息的内部语义展示方式,研究基于可视化工具的全方位展示技术,搭建统一 DevOps 平台实现自动化开发、测试和运维,以更好地满足政务行业应用对大数据产品快速、高效、多样化的信息可视化需求。

主要贡献如下:

(1) 从政务大数据业务落地出发,提出了一种面向 DevOps 的政务大数据分析可视化系统,主要依托于机器学习算法设计大数据分析方法,利用交互式和大屏可视化技术提出大数据可视化方法;

(2) 动态打通了机器学习分析工具、自动运维工具与数据分析可视化,相关技术的集成具有很强的系统指导和应用价值。

## 2 系统组成

本节主要介绍政务领域大数据分析可视化系统的基本结构与主要功能,并且将 DevOps 的方法引入系统的开发运维全生命周期中。

### 2.1 功能架构

本系统由云基础设施、DevOps 平台和政务大数据分析可视化系统构成,如图 1 所示。

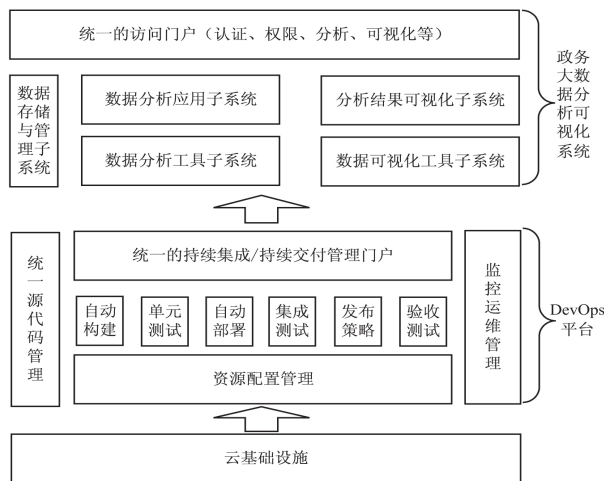
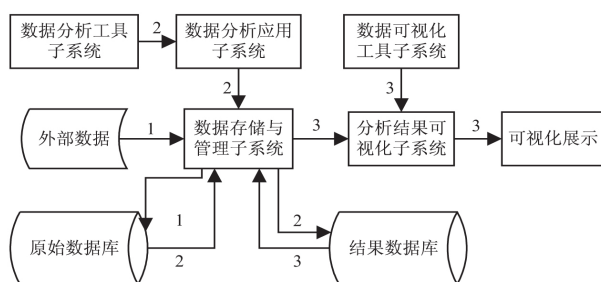


图 1 整体功能架构

其中,底层是云基础设施层,它表示整个系统的硬件资源环境。理想情况下它要么是公有云环境,要么是私有云环境,但在实际行业场景中,云基础环境大多是异构环境,比如是由前两者构成的混合云。

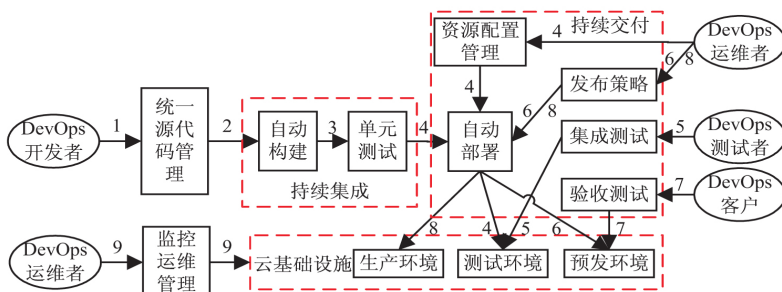
为了解决底层基础环境的异构问题,中间层设计了一个 DevOps 平台。它利用 Docker 容器引擎工具,提供一系列如创建容器镜像和运行容器等核心方法,可无视基础环境的多元性,实现统一源代码管理、资源配置管理、监控、运维管理等基本功能;同时,它通过统一的持续集成和持续交付管理门户,完成自动构建、单元测试、自动部署、集成测试、发布策略和验收测试等关键功能。

上述政务大数据分析可视化系统中各个子系统的工作流程如图 2 所示。



(3) 分析结果可视化子系统调用相关的数据可视化工具,并通过数据存储与管理子系统从结果数据库中获取数据,进行可视化图表展示。

上述 DevOps 平台中各个模块之间的交互流程如图 3 所示。



(5) 集成测试: DevOps 测试者在单元测试的基础上, 将所有应用程序按照设计要求, 组装成一个子系统或系统, 进行集成测试, 目标在于测试多种应用程序的

(8) 部署到生产环境: 通过所有测试后, DevOps 运维者便可以利用现有持续交付系统中发布策略管理功能, 使用灰度发布方法将预发布环境中最新的大数据分析可视化系统软件版本自动部署到真实生产环境中; 这里, 灰度发布又称金丝雀发布 (Canary release)<sup>[12]</sup>, 是在原有版本可用的情况下, 同时部署一

个新版本系统作为“金丝雀”,验证新版本的性能和表现,以保障整体系统稳定的情况下,尽早发现、调整问题;一旦新版本验证通过,逐步替换所有系统为新版本;

(9) 监控运维: DevOps 运维者还可以通过监控运维管理系统,对云基础设施中各种资源,进行实时监控、运维管理、日志分析等操作。

### 3 基于分析工具的大数据分析应用

在本节中,详细介绍大数据分析工具子系统中所用到的常用分析工具类型,以及如何根据给定的数据和分析工具,先训练后分析,最后实现大数据分析应用子系统的方法。

#### 3.1 分析工具

数据分析工具子系统通常由分类、聚类、回归、特征提取、时间序列分析、推荐、关联等分析模块构成,每个模块包含若干分析算法<sup>[13]</sup>。分类模块包含逻辑回归、支持向量机、朴素贝叶斯、神经网络、随机森林等算法;聚类模块包括 K 均值、离群点检测等算法;回归模块包括线性回归、K 近邻、决策树、神经网络等算法;特征提取包括循环特征消减选择(RFE)等算法<sup>[14]</sup>;时间序列分析包括趋势分析、自回归滑动平均模型(ARMA)等算法<sup>[15]</sup>;推荐模块包括基于用户和基于物品的协同过滤算法;关联模块包含先验知识(Apriori)关联分析算法。

需说明,上述所有算法分析工具均与真实政务行

业应用无关。算法分析工具接收来自原始数据库中数据作为输入后,会输出分析结果,供数据分析应用子系统中的分析应用软件调用。

#### 3.2 分析过程

为取得更好的分析效果,大数据分析过程通常需要先训练后分析。其中,训练阶段主要是:从原始数据库中读取一系列样本数据,经由数据分析工具中的训练模块,生成相应的分析模型文件,将模型文件存储到指定文件夹。分析阶段主要是:从指定文件夹中获取分析工具模型文件,读取待分析的数据文件,再经由数据分析工具中的分析模块,生成相应的分析结果,将结果数据存储到结果数据库中。

#### 3.3 应用方法

真实政务大数据行业应用类型繁多,为此提出的数据分析应用子系统是由多个应用类型模块构成,每类应用模块又包含多项分析内容,如图 4 所示。

这里,每项分析内容可以看作一个应用分析软件,它也分为训练和分析两部分。训练部分的主体是调用数据分析工具子系统中相应算法工具的训练模块,并读入与此相匹配的样本数据;此外,何时进行训练,则由训练策略决定,即可以人工实时启动,也可以自动定时启动。分析部分的主体是调用数据分析工具子系统中相应算法工具的分析模块,并读入与此相匹配的分析数据;同样,何时进行分析,由分析策略决定,可以人工实时启动,也可以自动定时启动。

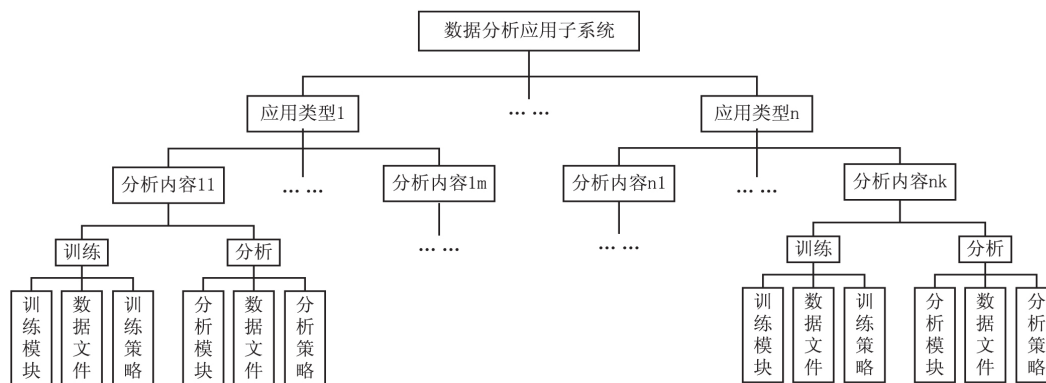


图 4 数据分析应用子系统的组成

### 4 支持交互式和大屏的大数据分析结果可视化

根据实际政务大数据行业应用的需求,分析结果可视化子系统又可细分为两种子系统:交互式可视化和大屏可视化。

#### 4.1 可视化工具

数据可视化工具子系统是由柱状图、散点图、折线图、饼图、二维矩阵图、分布直方图、分类图、时间序列

图、雷达图、词云图等图表工具模块构成。所有图表工具模块可以与真实政务行业应用无关。图表工具模块接收模型文件作为数据输入后,会输出指定类型的图表结果,再提供给分析结果可视化子系统。

#### 4.2 交互式可视化

交互式可视化需求主要是针对政务大数据行业内部复杂、海量的业务数据和外部数据,基于大数据云平台,如何通过良好的人机交互操作进行数据可视化分析,实现探索式分析、全维度数据钻取等功能。因此,



支持分析结果的终端交互式可视化展示功能,很可能是交互式可视化设计中需要关注的关键点。

交互式可视化子系统是由 Web 页面、Server 服务器、Scheduler 调度器和 Visual Engine 可视化引擎四部分组成。该子系统采用 W-S-S-VE 循环流程实现人机交互式展示分析结果,具体内容如下:

(1) Web 页面负责接收客户操作,向 Server 服务器发送可视化操作请求;这里的 Web 页面其实可以作为统一访问门户的一部分;

(2) 当 Server 服务器接收到可视化操作请求后会创建可视化任务,然后把该任务传送给 Scheduler 调度器;

(3) Scheduler 调度器接收到传递的可视化任务后,先调度这些任务并为其分配资源,再向 Visual Engine 可视化引擎发送可视化数据处理的请求;

(4) Visual Engine 可视化引擎接收到可视化数据请求后,会调用数据可视化工具子系统中相关图表工具,然后加载从结果数据库中获取的模型数据文件,绘制图表,最后把处理后的图表传回 Web 页面,由 Web 页面向客户展示处理绘制数据的可视化结果。

#### 4.3 大屏可视化

大屏可视化需求主要是指在电视大屏上对大数据分析结果进行可视化展现。在大数据背景下,大屏可视化需求背后存在着复杂的数据来源问题,如何保证复杂数据处理性能与多屏页面美工效果之间的合理平

衡,很可能是大屏可视化设计中应该关注的重要指标。

大屏可视化子系统是由多个不同布局形式的显示屏构成,每个显示屏又由若干显示块构成,每个显示块是最小显示单元,它可以展示某个图表工具绘制出来的图形或表格。显示块的主体是调用数据可视化工具子系统中相应的图表工具模块,加载需要大屏可视化展示的分析结果数据;这些分析结果数据在显示前会保存在模型文件中;还需要显示控制策略来控制该显示块中图表的显示时机、显示位置、尺寸、颜色等外部形态。

实现大屏可视化子系统的基本流程包括:

(1) 业务专家确定大屏内容显示方案。

根据业务场景抽取关键指标,然后定义分析指标的维度,最后定义不同指标的图表类型和页面划分。

(2) 美工人员设计整体布局和风格。

根据业务专家划分的页面进行指标布局设计,再根据用户要求的色系进行样图设计,最后根据用户定稿的样图实施切图。

(3) 开发人员研发前端页面及优化。

根据切好的图,结合样式需求制作静态网页模板,然后调用后端数据接口获取数据,开发前端页面,最后整体大屏显示细节调整与优化。

#### 4.4 分析结果可视化子系统的工作流程

不论交互式可视化方法,还是大屏可视化需求,分析结果可视化子系统的工作流程如图 5 所示。

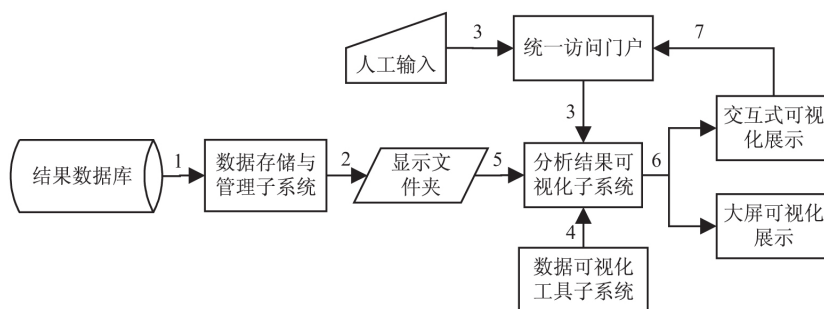


图5 分析结果可视化子系统的工作流程

(1) 可以采取周期性定时方法或者按需实时方法,从结果数据库中获取指定的数据,传送给数据存储与管理子系统中;

(2) 数据存储与管理子系统将收到的来自结果数据库中数据生成模型文件,传送到显示文件夹中保存起来;

(3) 系统客户可以通过统一访问门户,在网页上采取人工输入自动化脚本或直接操作等方式,实现对分析结果可视化展示的数据内容、展示效果等的有效控制;

(4) 分析结果可视化子系统响应人工输入的自动化脚本或实时操作,从数据可视化工具子系统中调用

相关的图表工具,为该图表工具设置模型文件所在的显示文件夹路径;

(5) 分析结果可视化子系统从显示文件夹中获取指定的分析结果模型文件;

(6) 根据人工输入所提供的显示控制方式,分析结果可视化子系统把指定的分析结果模型文件加载到对应的图表工具中,然后将图表工具绘制出来的图形或表格,传送给交互式可视化子系统或大屏可视化子系统,实现最终的可视化展示效果;

(7) 如果上一步是完成交互式可视化展示,那么交互式可视化子系统还会将得到的图表返回给统一访问门户中的 Web 页面,以便将可视化效果展示给客

户,实现与客户之间的循环交互。

## 5 DevOps 平台设计

随着容器技术的发展,Docker 容器已经成为使用量增长最快的工具<sup>[16]</sup>。通过 Docker 提供的 DevOps 系统,可以将软件系统的运行环境标准化、统一化,以镜像和在镜像基础上构建的容器为基础,同一个镜像可以部署到开发、测试、预发、生产等环境的容器中。这一过程都可自动完成,不需要重复的安装部署,提高了开发和运维效率。

与此同时,与 Docker 容器相关的 DevOps 工具种类繁多、用法各异、适用场景也不同<sup>[17]</sup>。因此,在本节中详细介绍一种基于 Docker 容器技术的 DevOps 平台,以持续集成、持续交付和容器云为核心,允许 DevOps 开发者、测试者和运维者通过一个统一的管理门户来使用该平台,从而打通开发、测试、运维全过程,如图 6 所示。

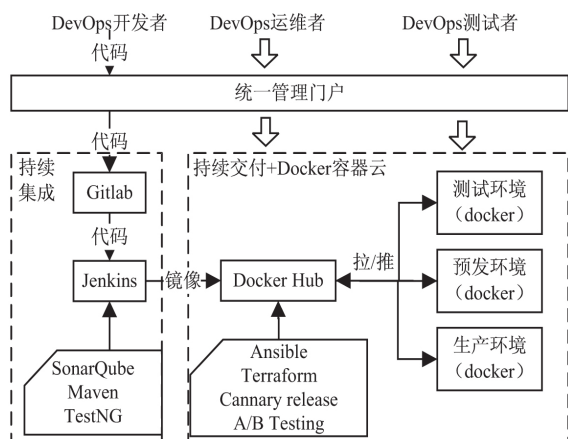


图 6 基于 Docker 容器的持续集成和持续交付

图 6 中,持续集成以源代码版本管理软件 Gitlab 和开源持续集成软件 Jenkins 为核心;持续交付是以 Jenkins 和容器部署配置软件 Docker Hub 为核心;容器云是以测试环境、预发环境和生产环境等三种 Docker 容器部署方式为核心。

### 5.1 持续集成

当 DevOps 开发者通过统一门户将源代码提交给 Gitlab 进行版本管理和项目文档的配置管理后,Jenkins 和相关工具插件会持续对源代码进行检查、编译和测试,这便是持续集成过程,具体内容如下:

(1) Jenkins 可以在 Repository URL 中配置好 Gitlab 代码仓库地址,使在 Gitlab 中每次提交后触发拉起最新版本的源代码;

(2) Jenkins 可以通过配置 Maven 工具插件地址,利用 Maven 等插件对源代码进行编译,实现自动构建;

(3) Jenkins 在对源代码构建完毕后,还可以调用 SonarQube<sup>[18]</sup>、TestNG<sup>[19]</sup>等工具插件,进行各项自动化

检查。其中,SonarQube 用于代码质量管理检查,TestNG 用于代码自动化测试,包括功能测试、性能测试、安全测试等。利用工具实现自动化检查,可以有效提高源代码的规范性,发现潜在的代码缺陷,提高代码质量。

### 5.2 基于 Docker 容器云的持续交付

当 DevOps 运维者通过统一门户对应用镜像进行 Docker 容器云资源的配置管理和镜像部署管理后,Jenkins 和 Docker Hub 会持续将构建好的 Docker 应用镜像,自动部署到 Docker 容器云中相应的测试环境、预发环境和生产环境,这便是基于 Docker 容器云的持续交付过程。同时,DevOps 测试者会通过统一门户,指定相关的测试方式,实现自动测试或人工测试。具体内容如下:

(1) Jenkins 还可以通过配置 Docker 容器服务器(Docker Hub)地址,根据源代码应用中编写好的 Dockerfile 文件<sup>[16]</sup>,自动构建应用镜像;这里,通过执行 Dockerfile 文件中基础镜像信息、维护者信息、镜像操作和容器启动等 4 种类型指令,构建一个虚拟容器,实现快速构建;

(2) Docker Hub 服务器软件可以利用 Terraform 开源工具插件<sup>[20]</sup>,根据应用系统的拓扑、依赖关系等对容器云资源进行配置管理;还可以利用 Ansible 开源工具插件,根据不同的部署方式,将应用镜像自动部署到不同的 Docker 容器云环境中;

(3) 当应用镜像通过预发环境中 DevOps 测试者对其的 A/B 测试<sup>[21]</sup>后,DevOps 运维者可以指定金丝雀发布策略<sup>[12]</sup>,实现预发环境中最新版本的应用镜像自动部署到生产环境中;

(4) 为实现持续交付过程,Docker Hub 会与三种类型容器云环境中应用镜像进行多次拉镜像和推镜像操作,可在任意环境实现多个应用系统的快速部署。

## 6 系统应用示例

限于政务数据安全要求,在本节构建了大数据分析可视化应用的演示示例,用于预先验证提出的方法。

基于 Python 编程语言的数据分析可视化系统的交互演示界面如图 7(a) 所示。该交互演示界面是以图形插件的形式在 Python3 环境下运行,集成了 4 种数据预处理方法、12 种数据分析工具和 12 种数据可视化工具。图 7(b) 为当选三维点图后,该交互界面根据相关数据,实时绘制出来的三维点图。三维点图所展示的内容都是持续动态变化,这是通过可以与后端的分析结果数据服务进行数据实时交互而得到的。经验证分析,该交互演示界面能够较好地反映出真实场景中数据分析工具子系统和数据可视化工具子系统的运行。

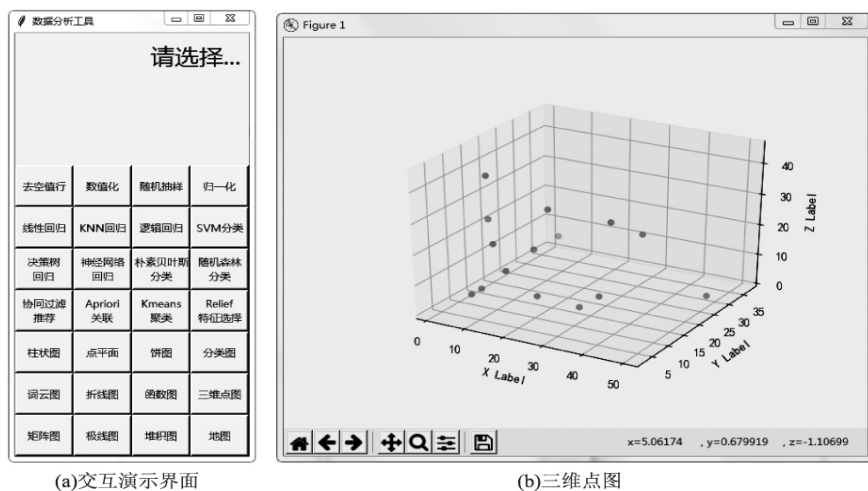


图7 数据分析可视化系统的交互演示界面和三维点图示例

## 7 结束语

提出了一种面向 DevOps 自动开发部署平台的政务大数据分析可视化系统,依托于机器学习算法设计大数据分析方法和利用交互式和大屏可视化技术提出大数据可视化方法。该系统已部署在某政府部门,根据真实场景下系统运行情况及应用结果表明,提出的大数据分析可视化系统,可以通过数学模型和机器学习帮助客户挖掘大数据更深层次的价值和内涵,满足了前期的政务大数据分析与可视化需求。该系统后续会持续迭代、开发、部署,以期更好地服务于客户的业务体系。

### 参考文献:

- [1] ZHOU L, PAN S, WANG J, et al. Machine learning on big data: opportunities and challenges [J]. Neurocomputing, 2017, 237: 350-361.
- [2] HUTTERMANN M. DevOps for developers [M]. New York: Apress, 2012.
- [3] 乔玮, 赵文瑞. DevOps 发展现状及趋势研究 [J]. 数字技术与应用, 2018, 36(4): 74-76.
- [4] 李超, 花磊, 宋云奎. OpsFlow: 一种面向 DevOps 的应用自动化部署引擎 [J]. 计算机与数字工程, 2019, 47(1): 190-194.
- [5] 凌云. 基于 Docker 平台的 DevOps 运维系统的研究与改进 [J]. 电脑知识与技术, 2018, 14(26): 209-211.
- [6] 樊炼, 廖振松. 一种云计算时代的 DevOps 自动化运维平台 [J]. 电信工程技术与标准化, 2018, 31(11): 63-66.
- [7] HETHEY J M. GitLab repository management [M]. Birmingham: Packt Publishing Limited, 2013.
- [8] Jenkins Project Team. Jenkins user documentation [EB/OL]. [2019-06-15]. <https://jenkins.io/doc/>.
- [9] MOHAAN M, RAITHATHA R. Learning ansible [M]. Birmingham: Packt Publishing Ltd., 2014.
- [10] Docker Inc. Docker hub: build and ship any application anywhere [EB/OL]. [2019-06-15]. <https://hub.docker.com/>.
- [11] DAVIS F D, VENKATESH V. Toward preprototype user acceptance testing of new information systems: implications for software project management [J]. IEEE Transactions on Engineering Management, 2004, 51(1): 31-46.
- [12] SATO D. Canary release [EB/OL]. [2019-06-15]. <http://martinfowler.com/bliki/CanaryRelease.html>.
- [13] TSAI C W, LAI C F, CHAO H C, et al. Big data analytics: a survey [J]. Journal of Big Data, 2015(2): 21.
- [14] CHANDRASHEKAR G, SAHIN F. A survey on feature selection methods [J]. Computers & Electrical Engineering, 2014(1): 16-28.
- [15] CHATFIELD C. The analysis of time series: an introduction [M]. London: Chapman and Hall, 2003.
- [16] TURNBULL J. The Docker book: containerization is the new virtualization [EB/OL]. [2019-06-15]. <https://dockerbook.com/>.
- [17] 文太益. 华数 DevOps 建设之路 [J]. 广播电视信息, 2017(10): 107-110.
- [18] CAMPBELL G, PAPAPETROU P P. SonarQube in action [M]. Greenwich: Manning Publications Co., 2013.
- [19] BEUST C, SULEIMAN H. Next generation java testing: TestNG and advanced concepts [M]. New York: Addison-Wesley Professional, 2007.
- [20] Hashi Corp, Inc. Terraform: write, plan, and create infrastructure as code [EB/OL]. [2019-06-15]. <https://terraform.io>.
- [21] SHAHIN M, BABAR A M, ZHU L. Continuous integration, delivery and deployment: a systematic review on approaches, tools, challenges and practices [J]. IEEE Access, 2017(5): 3909-3943.