**深 圳 大 学**

**联合培养**

**深圳职业技术学院**

本 科 毕 业 论 文（设计）

题目:**基于Vue+ElementUI的电商**

**后台管理及智能推荐系统**

姓名: **陈俊宇**

专业: **计算机科学与技术**

学院: **人工智能学院**

学号: **16240038**

指导教师: **李斌**

职称： **副教授**

2020年 5月 2日

**深圳大学深圳职业技术学院联合培养**

**本科毕业论文（设计）诚信声明**

本人郑重声明：所呈交的毕业论文（设计），题目《基于Vue+ElementUI的电商后台管理及智能推荐系统》 是本人在指导教师的指导下，独立进行研究工作所取得的成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式注明。除此之外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。本人完全意识到本声明的法律结果。

毕业论文（设计）作者签名：

日期： 年 月 日

**目 录**

[摘要 1](#_Toc40815665)

[1 前言 2](#_Toc40815666)

[1.1 研究背景及意义 2](#_Toc40815667)

[1.2 国内外研究及开发现状 2](#_Toc40815668)

[1.2.1 Vue框架 2](#_Toc40815669)

[1.2.2 MVVM 3](#_Toc40815670)

[1.2.3 Axios技术 4](#_Toc40815671)

[1.2.4 Webpack 4](#_Toc40815672)

[1.2.5 深度学习 4](#_Toc40815673)

[1.2.6 TensorFlow框架 4](#_Toc40815674)

[1.3 开发技术 6](#_Toc40815675)

[1.3.1 电商后台管理系统开发技术 6](#_Toc40815679)

[1.3.2 电影推荐系统开发技术 6](#_Toc40815680)

[1.4 开发工具 9](#_Toc40815681)

[1.4.1 电商后台管理系统开发工具 9](#_Toc40815682)

[1.4.2 电影推荐系统开发工具 9](#_Toc40815683)

[2 电商后台管理系统需求分析 10](#_Toc40815684)

[2.1 系统功能性需求 10](#_Toc40815687)

[2.2 系统非功能性需求 10](#_Toc40815691)

[3 电商后台管理系统详细设计 11](#_Toc40815692)

[3.1 界面设计 11](#_Toc40815693)

[3.2 业务流程 12](#_Toc40815694)

[3.3 项目结构 12](#_Toc40815695)

[3.4 接口详情 13](#_Toc40815696)

[3.4.1 接口说明 13](#_Toc40815704)

[3.4.2 支持的请求方法 13](#_Toc40815705)

[3.4.3 通用返回状态说明 13](#_Toc40815706)

[3.4.4 后端api路由 14](#_Toc40815707)

[4 电商后台管理系统实现 16](#_Toc40815708)

[4.1 登录模板的实现 16](#_Toc40815713)

[4.2 用户管理模板的实现 18](#_Toc40815714)

[4.2.1 用户列表模块的实现 18](#_Toc40815715)

[4.3 权限管理模板的实现 21](#_Toc40815716)

[4.3.1 角色列表模块的实现 21](#_Toc40815717)

[4.3.2 权限列表模块的实现 23](#_Toc40815718)

[4.4 商品管理模板的实现 23](#_Toc40815719)

[4.4.1 商品列表模块的实现 23](#_Toc40815720)

[4.4.2 分类参数模块的实现 29](#_Toc40815721)

[4.4.3 商品分类模块的实现 31](#_Toc40815722)

[4.5 订单管理模板的实现 31](#_Toc40815723)

[4.5.1 订单管理模块的实现 31](#_Toc40815724)

[4.6 数据统计模板的实现 32](#_Toc40815725)

[4.6.1 数据统计模块的实现 32](#_Toc40815726)

[4.7 系统测试 33](#_Toc40815727)

[4.8 本章小结 39](#_Toc40815728)

[5 电影推荐系统实现 39](#_Toc40815729)

[5.1 数据收集和准备 39](#_Toc40815731)

[5.2 构建模型 43](#_Toc40815732)

[5.3 模型训练与评估 44](#_Toc40815733)

[5.4 构建完整的电影推荐系统 46](#_Toc40815734)

[5.5 本章小结 46](#_Toc40815735)

[6 总结与展望 46](#_Toc40815736)

[参考文献 48](#_Toc40815737)

[致谢 49](#_Toc40815738)

[Abstract（Key words） 50](#_Toc40815739)

基于Vue+ElementUI的电商后台管理系统及智能推荐系统

计算机科学与科学技术 陈俊宇

学号：16240038

【摘要】从二十世纪以来，电商平台以一种迅猛的速度发展，电商运营的商品成千上万，使得电商后台的信息化管理的功能越来越重要。本论文项目之一的电商后台管理系统采用Vue官方提供的脚手架构建系统，前端页面使用ElementUI实现。本文主要介绍了系统的功能需求及其整体技术架构,浅谈了Vue的MVVM模式的特点、Vue的优缺点、axios以及webpack网络打包工具,针对开发过程中遇到的问题进行分析并提出解决方案,并就实践中使用的技术加以阐述。不仅如此，本人还通过使用TensorFlow构建的基于内容的电影推荐系统,介绍基于内容的推荐系统的原理、基于矩阵分解的推荐系统原理以及基于商品的协同过滤的推荐系统的原理。

**【关键词】**Vue.js；ElementUI；TensorFlow;矩阵分解；协同过滤；

# 前言

## 研究背景及意义

我们即将踏入全面5G时代，新兴的互联网技术大大改变了人们的生活方式。例如在购物方面，19世纪到二十世纪初，人们只能步行或者使用其他交通方式到集市购买生活必需品，但是现在只需要轻轻打开手机，到应用商城里面点击需要购买的东西，就可以足不出户，由快递员送到门前。网购的行为，大大地方便了人们的生活，也促进了经济增长。由于电商的商品形式越来越多样化，我认为一个好的电商后台管理系统至关重要，这不但影响消费者的消费体验，直接影响企业的收入，还影响到企业的风评。这个系统的商品分类、权限管理等功能能很好地帮助电商管理自己的商品。我们需要做的是进一步完善这个电商后台管理系统，使得用户能有更好的浏览体验以及能更好地帮助管理人员管理商品。

现在人工智能应用已经非常广泛，推荐系统就是其中一个典型的例子，比如新闻推荐、音乐推荐、电影推荐。例如网易云的私人FM就是非常好用的个性化系统。为什么要研究人工智能呢？原因有两个，原因之一个是创造智能机器，另一个是更好地理解人类智能的本质。总之，人工智能不是单独一个学科组成的，它既是工程又是属于科学的。人工智能的发展可以使人工智能产品可以更好地造福人类社会。

## 国内外研究及开发现状

### Vue框架

Vue.js在2014年开源后，就备受国内外开发人员关注。Vue.js初始时是一个视图，但随着时代的发展，Vue全家桶的逐渐完善，使得Vue.js慢慢地演变为一个完整的前端框架。从从2009年到2014年，产生了很多优秀的前端框架，比如AngularJS、React、Vue等，这些前端框架的不断出现，说明前端工程框架正在取代传统的开发模式。

Vue.js是一套用于现代化Web项目应用的轻型渐进式脚本框架，是一款MVVM前端框架[[1]](#endnote-2)[1] 。Vue与其他框架不同的地方在于，它是一套轻巧而高性能的用于搭建用户界面的渐进式框架。不仅如此，当与单文件的组件和Vue生态支持的库使用时，Vue完全能够为复杂的单页应用程序提供驱动[[2]](#endnote-3)[2] 。

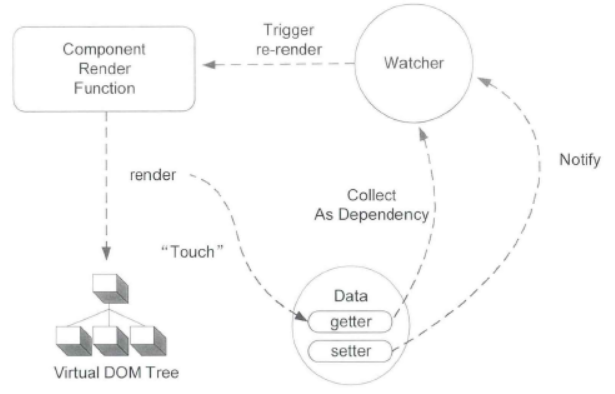


图1 Vue结构原理图

每一个Watcher实例对象与组件实例都存在一一对应的关系，渲染组件时，Watcher实例对象把属性记录为依赖，当setter被调用，Watcher实例对象重新计算[[3]](#endnote-4)[3] 。

一个好的项目，它的前端既不能冗余，也不能太过 简单。既要保证用户的使用体验，又需要将自己好的一面 展示出来[[4]](#endnote-5)[4] 。显而易见，vue.js轻量级脚本可以说是非常适合电商后台管理系统了。Vue框架具有以下特点，轻量级的框架、双向数据绑定、指令、插件化[[5]](#endnote-6)[5] 。

### MVVM

MVVM模式是Model-View-ViewModel的缩写[[6]](#endnote-7)[6]，即模型-视图-视图模型。MVVM模式侧重于开发事件驱动的UI平台[[7]](#endnote-8)[7]，Model-View-ViewModel模型他们三者的关系如图2所示：

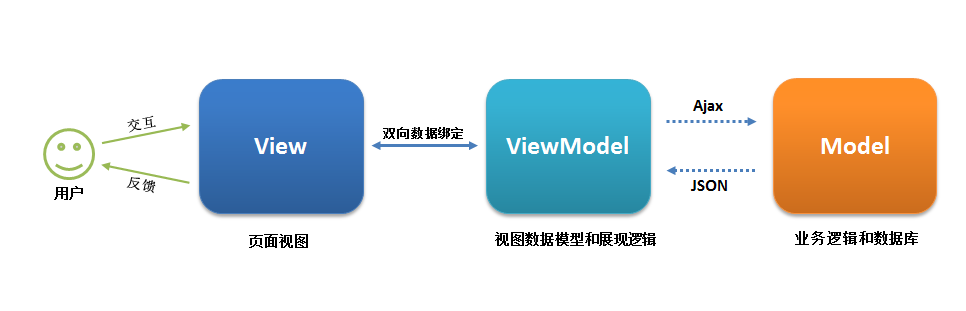


图2 MVVM的工作流程

MVVM模式是MVC模式上进一步发展的产物 [[8]](#endnote-9)[8]。Model指代数据访问层，指的是后端的业务逻辑处理以及数据操控，除了数据本身以外，它不关注其他任何东西；View与MVC的视图一样，是用户在屏幕上看到的UI界面；视图模型是同步视图和模型的对象。

模型和视图不能直接进行联系，但是视图模型可以作为桥梁，接通模型和视图，我们需要一名观察者充当交警，来监督数据的的处理情况，observer就是这名交警。也就是说，MVVM模式的核心是视图模型。他们之间存在这样一种模式，视图模型会随着视图的变化而更新。将后端的传递的数据转化为显示的UI界面，即模型转化为视图，需要挺高数据绑定的形式实现；要使视图转化为模型，则需要采用DOM事件监听处理，这一过程称之为数据的双向绑定。

总的来说，为了实现数据的双向绑定，当数据在系统的运行中发生改变时，视图的变化会被视图模型所识别，接着通过数据做出改动。Vue采用的就是这种模式。

### Axios技术

Axios基于promise用于浏览器和node.js的http客户端[[9]](#endnote-10)[9]。

Axios有如下特点：支持 Promise；支持node端和浏览器端；配置项多；社区支持[[10]](#endnote-11)[10] ；

### Webpack

Webpack 是一个现代 JavaScript 应用程序的静态模块打包器(module bundler)[[11]](#endnote-12)[11]，它的核心是输出、插件、loader、以及入口。Webpack可以把任何资源，包括js、css、html、图片以及其他任何类型的文件和js模块都可视为模板，通过loader被处理[[12]](#endnote-13)[12] 。打包原理是，把所有依赖打包成bundle.js文件，通过代码分割成单元片段并按需加载[[13]](#endnote-14)[13] 。

### 深度学习

深度学习 [[14]](#endnote-15)[14] 是机器学习的拓展，其训练深度神经网络也会用到有、无监督的方法。近几年，深度学习正在不以一个惊人的速度发展壮大，一些特殊的技术手段不断地被提出，因此越来越多的人将深度学习看作是一个的单独的学习方法。

深度学习在2012年后热度持续上升，一直到2017年也还是处于持续上升的状态当中。国内在2015年开始兴起人工智能，至今，人工智能的技术已经有了很好的发展趋势并且在各个领域当中都得到了很好的应用。2017年，阿尔法围棋以3: 0战胜了当时的世界冠军棋手柯杰。从那时起，人工智能变被人逐渐了解。深度学习是AlphaGo围棋程序的设计原理。近年来，深度学习备受开发人员关注，现如今，它已经是人工智能的一个火爆研究方向。深度学习是一种通过多层次分析和计算获得结果的方法。接下来我们研究的TensorFlow就是深度学习框架之一。

### TensorFlow框架

云计算和大规模并行处理基础加构的共同发展不仅使得机器学习和深度人工智能有更广阔的应用空间，也激发人工智能框架的快速迭代和部署[[15]](#endnote-16)[15] 。

TensorFlow是Google开源软件库[[16]](#endnote-17)[16]，是一个基于数据流编程的符号数字系统，被广泛运用于各类机器学习算法的编程实现。TensorFlow拥有多层级结构，可以分为设备层、网络层、数据操作层、图计算层、API层和应用层，其中设备层、网络层、数据操作层，图计算层是TensorFlow的核心层[[17]](#endnote-18)[17]

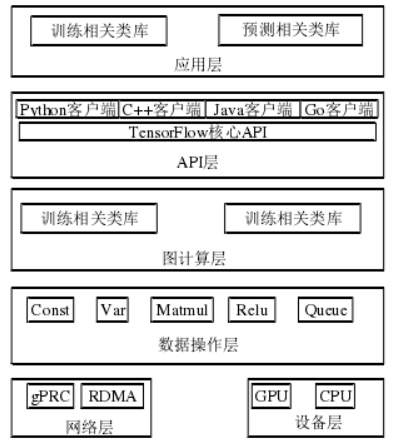


图3 Tensorflow系统架构

如图3所示，最底层是网络通信层和设备管理层。设备管理层主要包括可在TensorFlow中引导运行的设备，如中央处理器(CPU)和图形处理器(GPU)，它们的作用是为数据操作层提供接口。再上一层，即数据操作层，该层包含一些数据操作，如卷积核、Rulu等。一些报错以及图计算、图节点之间的依赖关系，则在图计算层实现。API层则是系统基础的算法实现层。正是这种体系结构使得TensorFlow在深度学习应用中发挥着越来越重要的作用[[18]](#endnote-19)[18]。

对比于其他主流深度学习框架，如Caffe、Theano、Pytorch等，TensorFlow是一款可自动求微分、多语言支持的、拥有高优化性能、高度灵活性的、以及可移植性的的深度学习框架。

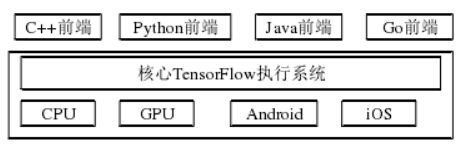


图4 TensorFlow多语言支持

目前，深度学习在计算机视觉和自然语言处理中的应用已经远远超过了传统的机器学习方法[[19]](#endnote-20)[19] 。 TensorFlow已经被广泛利用，尤其是神经网络模型构建和参数优化等问题的研究上[[20]](#endnote-21)[20] 。它提供了一个强大而灵活的网关，可以进行深度学习。本项目中，在通过训练和评估模型的时候发现两个问题。首先，为了体现深度学习的优势，还需要对数据进行大量的训练，但是如果数据的样本量比较小，就会使得深度学习的效果得不到预期值，然而，这种情况却可以使用传统的机器学习处理。其次，在某些领域，深度学习方法显得复杂，只需要使用传统的机器学习就可以很好地解决问题。

## 开发技术



### 电商后台管理系统开发技术

前后端分离技术是本论文中电商后台管理系统的开发理念。后端主要负责的是数据库的操作，以及向前端暴露接口，前端主要负责绘制页面，同时基于Ajax技术，调用后端提供的API接口。前端项目技术栈会使用到Vue，Vue-router路由、ElementUI前端UI组件库、网络数据请求、绘制相关图形报表（Echarts）。后端使用了Mysql、Jwt 、Node.js等技术。

### 电影推荐系统开发技术

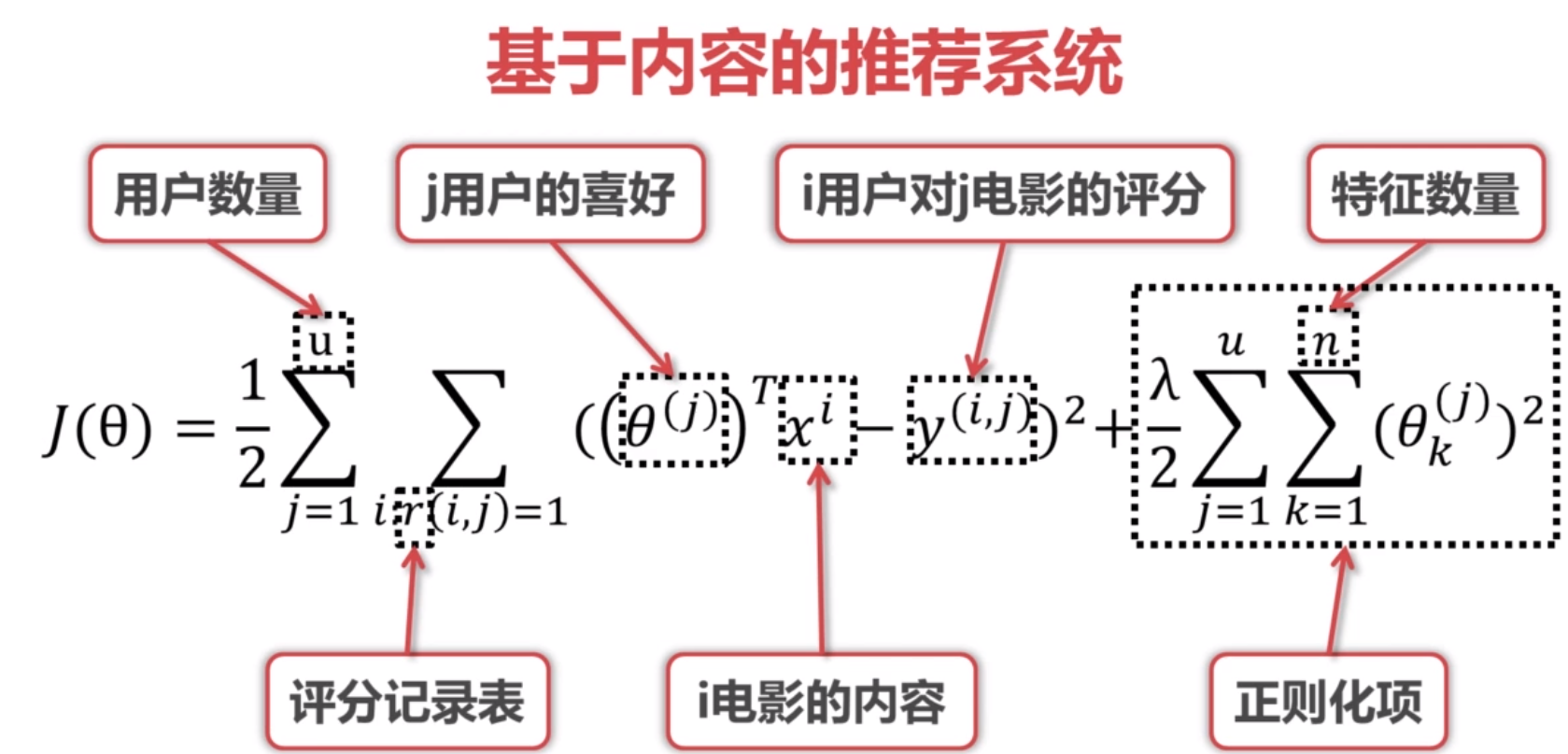
#### 基于内容的推荐系统

电影评分表的第一行是用户列表，第一列是电影列表，表中的数字是用户对电影的评分，用户没有对影片评分用问好表示。根据每个电影和已经被评估的电影的内容来推断每个用户对每部电影的偏好，以此基础来推断每个用户对他们未曾看过的电影的评价，这便是基于内容的推荐系统的原理。

表1 电影评分表与电影内容矩阵X



评分记录表指的是i用户对j电影评分，如果有评分，对应的评分记录表的位置为1，如果没有评分，则为0。θ(j)T与xi相乘即电影推荐系统预测的评分，θ(j)T乘xiy-y(i,j)为误差值。正则化项是用于防止θ构成的模型对原始数据集产生过拟合的箱，通过调整其中的λ防止θ过拟合。用户喜好矩阵代价函数J(θ)公式如下：

（1）

最小化代价函数是我们的研究目的，实现目的的方法是采取梯度下降等算法。再研究基于内容的推荐系统时，我发现了一个问题，电影内容矩阵的构建实在麻烦，要通过人为观察进行计算一部电影的喜剧、武侠、爱情片成分。

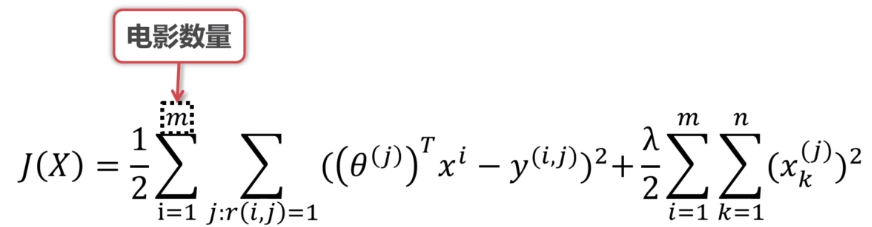
#### 基于矩阵分解的协同过滤

想要推断出每个用户对未观看过的电影的评分，就需要每部电影的内容，而这就要依据每个用户对每部电影类型、评分的电影的偏好来预测。

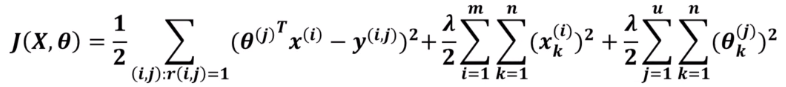
表1.2 电影评分表与用户喜好矩阵θ



通过在线问卷调查获得用户对电影的评价，可以得出用户喜好矩阵，然而这种方法有一个缺陷，就是它不能确保每个用户填写都进行问卷调查，并且不能保证用户填写的信息是正确的，那怎么解决这个问题呢？

（2）

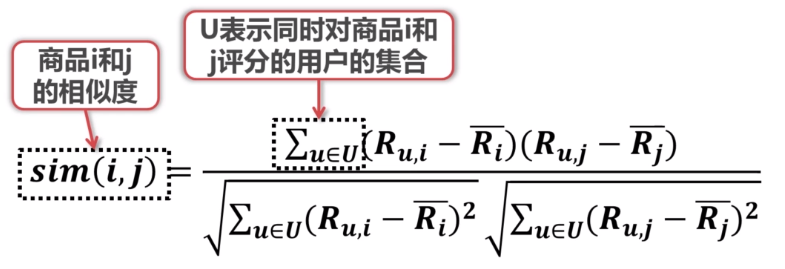
解决方法是，合并公式用户喜好矩阵代价函数（1）和电影内容矩阵代价函数（2），得到公式（3）协同过滤的代价函数J(X,θ)，求解x和θ两个参数。第一，我们要将电影内容矩阵X和用户喜好矩阵θ进行随机初始化，然后通过梯度下降等算法，这样电影评分表得到的X和θ就拥有更好的拟合效果。要实现这一方法，只要有用户对电影的评分即可，用户对于不同类型电影的喜好程度及内容将会被系统自动得识别出来，因此我们没有必要去收集其他信息。这便是基于协同过滤推荐系统的原理。

（3）

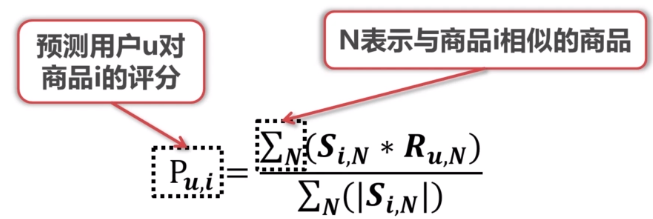
前面实现协同过滤的方法叫作矩阵分解，在使用这种方法后，我们将会得到了两个矩阵，一、用户喜好矩阵θ，二、电影内容矩阵x。将上述两个求解得到的矩阵相乘，一张完整的评分表便被我们所获取到了。此时每个用户对于每部电影都是完整的，可以按照评分，从高到低的顺序对这些电影进行排序，将评分最高的那些电影推荐给用户。这就是本项目采用的推荐方法。

#### 基于item的协同过滤和基于内容的协同过滤

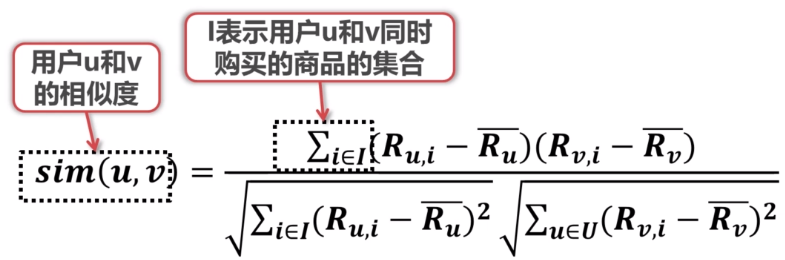
计算商品评分之间的相似度是我们的工作重心，向用户进行推荐的依据是商品评分之间的相似度。用户对商品的评分是我们所需要的。度量商品之间的相似度方式叫作correlation-based similarity，公式如下：

（4）

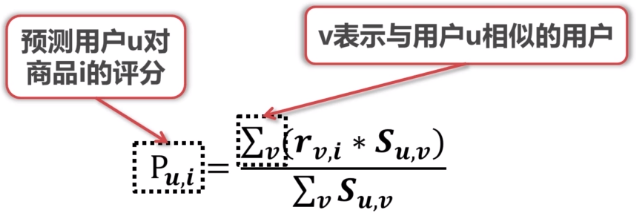
预测用户对商品的评分之前，我们先要计算出商品的相似度。想要使系统选择评分最高的商品推荐给用户，把与商品i相似的商品，需要根据与商品i比较相像的商品来推断用户u对商品的评分，当用户u对商品的评分被系统计算出来后，对商品评分进行排序，便可以实现。预测用户对商品的评分公式如下：

 （5）

把要推荐的用户设为A，找到与A相似的其他用户是基于用户的协同过滤的首要任务，把其他用户购买过的商品推荐给A是重中之重。度量用户的相似度公式如下：

 （6）

计算出用户之间的相似度后，就可以推断出用户对商品的评分。用户评分公式如下：

（7）

#### 冷启动问题

冷启动，当系统出现了新用户时，对该用户或该商品进行推荐是系统不能做到的事情。遇到冷启动问题的时候，我们不仅要考虑到用户冷启动还要同时考虑到商品冷启动，也就是说这两种办法需要同时考虑到。

用户冷启动，当新用户选择了一个商品或者多个商品，并且对商品进行评分之后，系统把该用户的信息加入到评分中，然后系统会对该用户进行推荐。商品冷启动，系统提供热门内容，类似热度，将热门的内容有限推给用户。

#### 基于内容的推荐系统的优缺点

基于内容的推荐系统关键在于构建商品内容矩阵，它不存在商品冷启动问题，可以明确告诉用户推荐的商品哪些属性，但是需要对内容进行透彻的分析，并且很少能给用户带来惊喜。存在用户冷启动问题。

#### 基于协同过滤的推荐的优缺点

不需要对商品有很好的认知，就能够通过各个用户的历史信息推断出商品的质量。缺点在于冷启动问题、gray sheep问题，以及协同过滤的复杂度会随着商品的数量和用户数量的增加而增加，同义词的会影响系统的性能，还有恶意刷分也会影响协同过滤系统的判断。

## 开发工具

### 电商后台管理系统开发工具

本系统主要使用了Vue+ElementUI进行开发，开发工具为 ，使用的vue脚手架版本为4.2.3，通过Vue脚手架创建项目，配置Vue路由，配置axios库，使用第三方接口，接口请求的地址为http://server.sineava.top/api/private/v1/，接口的测试使用Postman完成。使用Win10系统进行开发和测试。

### 电影推荐系统开发工具

本系统主要使用了Python语言进行开发，并采用TensorFlow框架进行搭建。开发工具为Jupyter NoteBook。使用Win10系统进行开发和测试。

# 电商后台管理系统需求分析



## 系统功能性需求

图5 电商后台管理系统模块设计

1. 登录模块：登录与系统主界面。
2. 用户管理模块：包含用户列表子模块，主要用于对用户的管理。拥有对用户进行增删改查、更新用户状态、为用户分配角色的功能。
3. 权限管理模块：包含角色列表和权限列表两个子模块。角色列表拥有对角色进行增删改以及权限分配的功能。权限列表模块，主要进行权限列表的查看。
4. 商品管理模块：包含商品列表、分类参数和商品分类三个子模块。分类参数列表，包含选择商品分类、删除商品参数或属性以及修改商品参数或属性的功能。商品列表，主要是对商品进行增删改查操作。商品分类包含对商品进行增删改的操作。
5. 订单管理模块：包含订单列表子模块，拥有对订单进行查询编辑、以及对商品的物流进行定位的功能。
6. 数据统计模块：包含数据报表子模块，点击数据报表侧边导航栏，查看华东、华南、华北、西部以及其他地区的用户来源情况，也可以报表上方点击华东、华南、华北、西部或其他地区，单独观看各个地区的用户来源情况。

## 系统非功能性需求

为了使用户有更好的浏览体验，就需要对系统进行优化，我们需要考虑以下几点非功能性需求：

1. 生成打包报告：在打包的过程中，还要同时生成报告的目的是为了能够更直接地发现项目的BUG。实现这一步骤的方法有很多，使用命令行参数成报告是一种方式，通过可视化面板查看报告也是一种方法。
2. 采用CDN方式优化前端页面组件：本项目中，按照需求加载前端组件虽然已经是一个很好的优化系统的方法，但是这些组件，依旧占用了较大的空间。将其中的组件，通过CDN的形式来加载，打包后的文件体积将会减少许多。
3. 路由懒加载：指的是当路由被访问时，系统按要求加载对应的组件，所以不同的代码块需要其不同路由的对应组件。

# 电商后台管理系统详细设计

## 界面设计



图6 界面设计图

## 业务流程

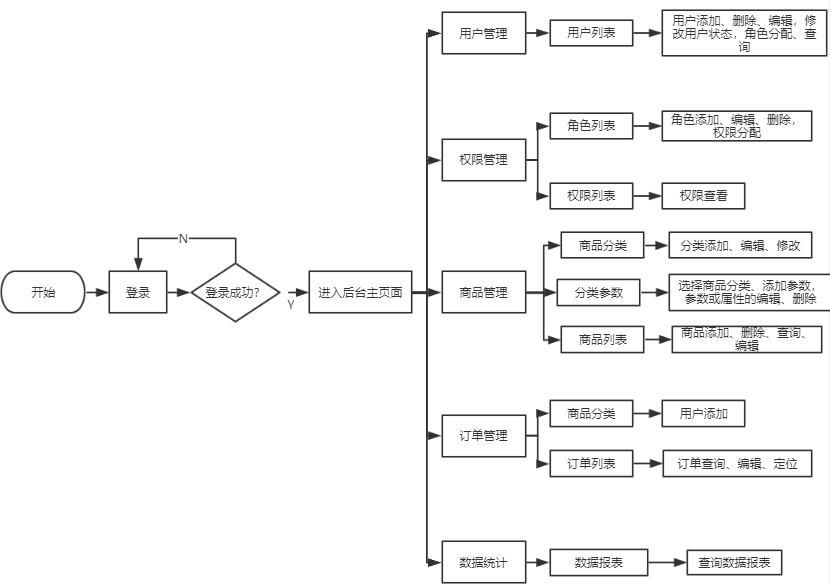


图7 业务流程图

## 项目结构

系统采用前后端分离方式进行，使用Vue脚手架对项目前端进行搭建，并对路由进行如表所示的划分：

表3-1 路由划分表

|  |  |  |
| --- | --- | --- |
| 路由 | 组件相对路径 | 说明 |
| /login | ../components/Login.vue | 登录 |
| /home | ../components/Home.vue | 主页 |
| /users | ../components/user/Users.vue | 用户列表 |
| /rights | ../components/power/Rights.vue | 权限列表 |
| /roles | ../components/power/Roles.vue | 角色列表 |
| /cate | ../components/goods/Cate.vue | 商品分类 |
| /params | ../components/goods/Params.vue | 分类参数 |
| /goods | ../components/goods/List.vue | 商品列表 |
| /goods/add | ../components/goods/Add.vue | 商品添加 |
| /others | ../components/order/Order.vue | 订单管理 |
| /reports | ../components/report/Report.vue | 数据统计 |

## 接口详情



### 接口说明

- 接口基准地址：` http://server.sineava.top/api/private/v1/ `

- 服务端已开启 CORS 跨域支持

- API V1 认证统一使用 Token 认证

- 需要授权的 API ，必须在请求头中使用 `Authorization` 字段提供 `token` 令牌

- 使用 HTTP Status Code 标识状态

- 数据返回格式统一使用 JSON

### 支持的请求方法

- GET（SELECT）：从服务器取出资源（一项或多项）。

- POST（CREATE）：在服务器新建一个资源。

- PUT（UPDATE）：在服务器更新资源（客户端提供改变后的完整资源）。

- PATCH（UPDATE）：在服务器更新资源（客户端提供改变的属性）。

- DELETE（DELETE）：从服务器删除资源。

- HEAD：获取资源的元数据。

- OPTIONS：获取信息，关于资源的哪些属性是客户端可以改变的。

### 通用返回状态说明

表3-2 通用返回状态说明表

|  |  |  |
| --- | --- | --- |
| 状态码 | 含义 | 说明 |
| 200 | OK | 请求成功 |
| 201 | CREATED | 创建成功 |
| 204 | DELETED | 删除成功 |
| 400 | BAD REQUEST | 请求的地址不存在或者包含不支持的参数 |
| 401 | UNAUTHORIZED | 未授权 |
| 403 | FORBIDDEN | 被禁止访问 |
| 404 | NOT FOUND | 请求的资源不存在 |
| 422 | Unprocesable entity | [POST/PUT/PATCH] 当创建一个对象时，发生一个验证错误 |
| 500 | INTERNAL SERVER ERROR | 内部错误 |

### 后端api路由

表3-3 后端api路由详情说明表

|  |  |  |
| --- | --- | --- |
| 路由 | 方法 | 说明 |
| login | post | 登录验证 |
| users | get | 用户数据列表 |
| users | post | 添加用户 |
| users/:uId/state/:type | put | 修改用户状态 |
| users/:id | get | 根据 ID 查询用户信息 |
| users/:id | put | 编辑用户提交 |
| users/:id | delete | 删除单个用户 |
| users/:id/role | put | 分配用户角色 |
| rights/:type | get | 所有权限列表 |
| menus | get | 左侧菜单权限 |
| roles | get | 角色列表 |
| roles | post | 添加角色 |
| roles/:id | get | 根据 ID 查询角色 |
| roles/:id | put | 编辑提交角色 |
| roles/:id | delete | 删除角色 |
| roles/:roleId/rights | post | 角色授权 |
| roles/:roleId/rights/:rightId | delete | 删除角色指定权限 |
| categories | get | 商品分类数据列表 |
| categories | post | 添加分类 |
| categories/:id | get | 根据 id 查询分类 |
| categories/:id | put | 编辑提交分类 |
| categories/:id | delete | 删除分类 |
| categories/:id/attributes | get | 参数列表 |
| categories/:id/attributes | post | 添加动态参数或者静态属性 |
| categories/:id/attributes/:attrid | delete | 删除参数 |
| categories/:id/attributes/:attrId | get | 根据 ID 查询参数 |
| categories/:id/attributes/:attrId | put | 编辑提交参数 |
| goods | get | 商品列表数据 |
| goods | post | 添加商品 |
| goods/:id | get | 根据 ID 查询商品 |
| goods/:id | put | 编辑提交商品 |
| goods/:id | delete | 删除商品 |
| goods/:id/pics | put | 同步商品图片 |
| goods/:id/attributes | put | 同步商品属性 |
| upload | post | 图片上传 |
| orders | get | 订单数据列表 |
| orders/:id | put | 修改订单状态 |
| orders/:id | get | 查看订单详情 |
| /kuaidi/:id | get | 查看物流信息 |
| reports/type/1 | get | 基于时间统计的折线图 |

# 电商后台管理系统实现



## 登录模板的实现

登录页面由文本框以及按钮等组件组成。点击登录，通过表单数据验证，实现登录功能，将重定向到指定跳转页面。点击重置按钮，将清空文本框中的用户名及密码。该部分实现了登录以及控制页面访问权限功能。

主页面由顶部，侧边栏以及内容主体构成。顶部由logo以及登出按钮组成，侧边栏由下拉导航组件组成，采用双层for循环实现左侧菜单数据的加载，并实现侧边栏的折叠与展开效果，内容主体由面包屑导航栏、按钮、文本框、分页等组件构成。

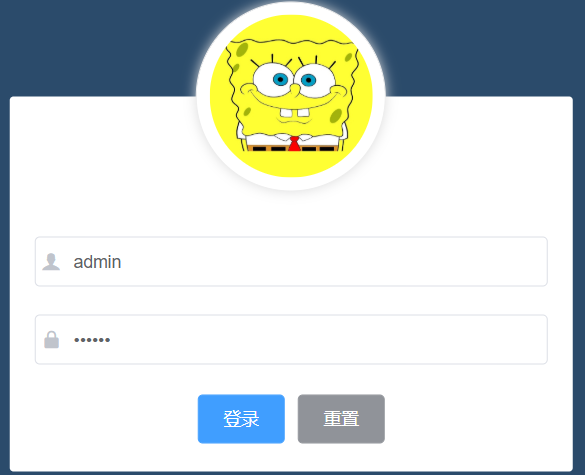


图8 首页界面

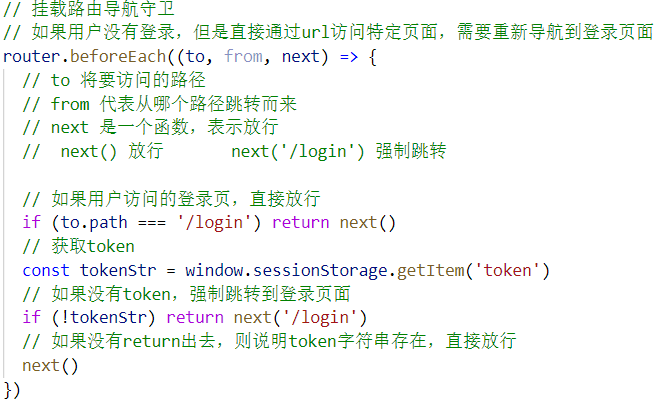


图9 主页面

首页输入账号密码，根据后台的响应状态发起axios登录请求。代码实现如下：



想要控制用户的访问权限，需要实现路由导航守卫的功能。倘若用户处于注销状态，但是想要直接通过输入地址的方式特定页面，系统就会重新跳转到登录页面，强迫用户登录，增加系统的安全性。实现代码如下：



## 用户管理模板的实现

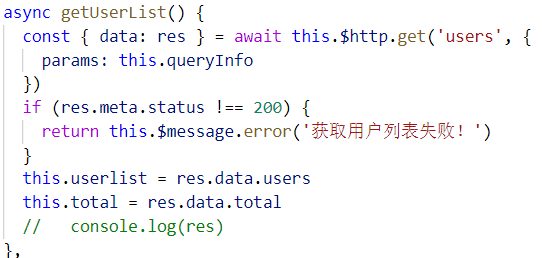
### 用户列表模块的实现

用户列表由面包屑导航、按钮、文本框、卡片视图、分页等组件实现其UI界面，实现了用户查询、添加、删除、修改、为用户分配角色、修改状态的功能。用户列表组件，会由路由的形式渲染出来。用户列表数据需要当用户列表被渲染出来后获取。



图10 用户列表界面

由API接口详情可知，获取用户列表数据，需要使用get请求user路径，代码实现如下：



用户查询功能只需在查询按钮绑定一个单击事件到getUserList()即可。

添加用户时需要进行表单预验证，验证通过后，点击确定按钮，发起添加用户的网络请求，添加用户成功。代码实现如下：



对用户的信息进行修改，实现代码如下：



根据id删除对应的用户信息功能，实现代码如下：



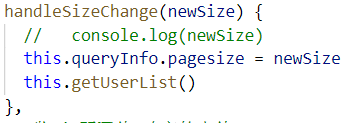
实现点击按钮分配角色，代码如下：



监听switch开关状态的改变，即实现点击按钮修改用户状态的功能，代码如下：



监听页码值改变事件，代码如下：



## 权限管理模板的实现

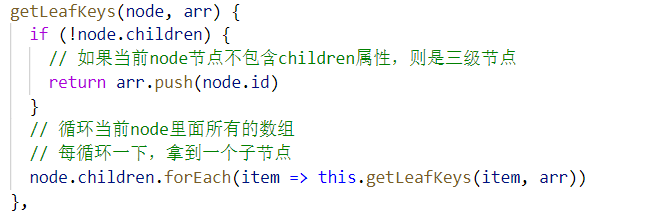
### 角色列表模块的实现

角色列表由面包屑导航、按钮、卡片视图等组件构成UI界面。实现了添加角色、对角色进行编辑、删除以及对角色进行权限分配的功能。路由的功能不但渲染角色列表，而且还要对基本布局进行结构，与此同时还要获取列表数据。当完成角色列表数据的显示后，使用三重for循环渲染三级权限，最后完成其他几个按钮的功能。



图11 角色列表界面

添加角色按钮功能如通用户列表的添加用户功能，只需要将请求用户列表的接口地址和请求方法修改为角色列表的请求地址和请求方法即可。对角色进行删除和编辑亦如上。点击权限分配，弹出权限分配窗口，点击选中的复选框后点击确定，同时更新角色列表下的视图。实现方式为通过递归的形式，获取角色下所有三级权限的id，并保存到defKeys数组中，代码如下：



展示分配权限的对话框，实现代码如下：

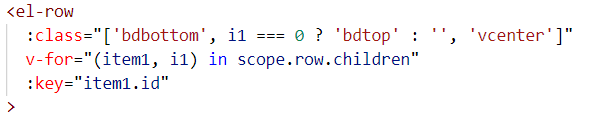


最后，为角色分配按钮添加单击事件，点击角色分配权限实现权限分配按钮功能。



图12 权限分类按钮

实现角色列表下的权限列表界面，需要先使得界面实现栅格系统，通过三重for循环渲染角色列表下的视图的功能。一级权限，每循环一次为el-row添加一个el-tag标签；二级权限如一级权限方式实现；三级权限为在二级权限嵌套el-col，实现代码如下：





### 权限列表模块的实现

通过调用API接口获取权限列表的数据，然后渲染到页面中。



图13 权限列表

## 商品管理模板的实现

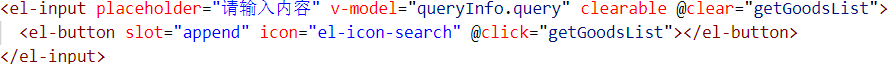
### 商品列表模块的实现

商品列表组件会被路由加载到前端页面中，在系统获取商品列表数据之后，渲染到商品表格中，并且使用全局过滤器处理时间格式，通过编程式导航跳转到商品添加页面。



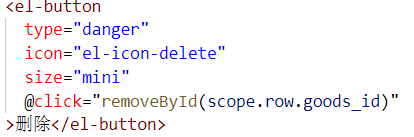
图14 商品列表界面

点击搜索框，输入需要搜索的商品，点击查询按钮，返回查询数据，这时候，点击文本框右侧的交叉按钮，可以实现搜索清空功能。实现搜索功能需要通过调用商品列表数据，将输入框的数据和查询的数据query实现双向绑定。清空功能，只需要在组件加上clearable属性即可，实现代码如下：





点击删除按钮，弹出提示框，询问用户是否删除，如若点击确定，则根据列表的ID删除商品。实现该功能需要为删除按钮添加单击事件。商品删除功能实现代码如下：





商品添加的具体流程如图15所示：



图15 添加商品的基本信息界面



图16 添加商品的商品参数界面



图17 添加商品的商品属性界面



图18 添加商品的商品图片界面

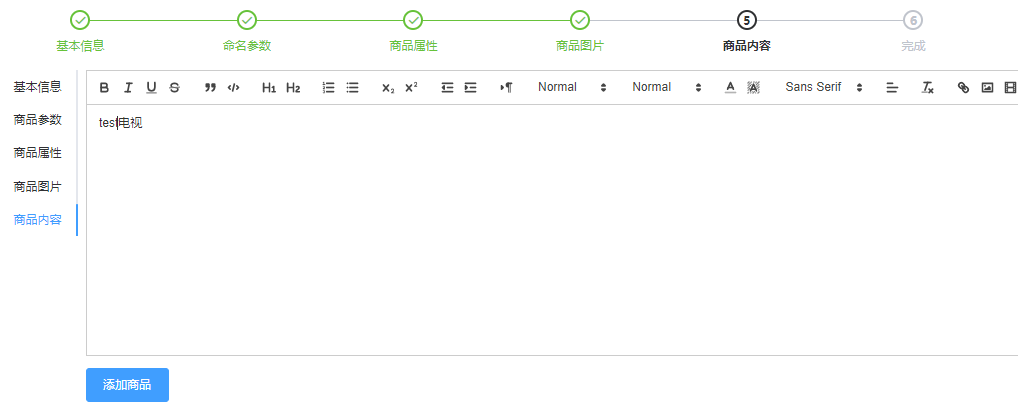


图19 添加商品的商品内容界面



图20 添加商品后

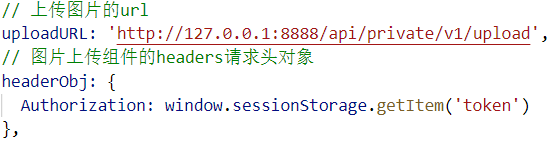
基本信息界面，基本信息面板的前端页面结构可以通过表单的组成部分分析出来。获取商品分类数据，绘制商品分类的联机选择器，并且限定，只能选中三级商品分类，还设置了阻止页面切换的功能。实现代码如下：



商品参数界面，获取动态参数列表数据，可以实现商品参数面板的复选框组件功能。商品属性面板的UI结构的渲染，需要获取静态属性列表数据。实现代码如下：

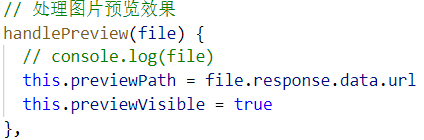


商品图片界面，使用upload上传组件，为其绑定headers请求头，监听upload组件的on-success事件以及on-remove事件，即实现下方的handleSuccess和handleRemove方法。实现代码代码如下：

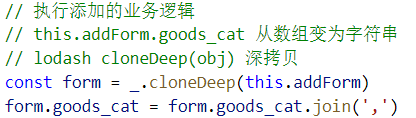




图片预览功能实现代码如下：



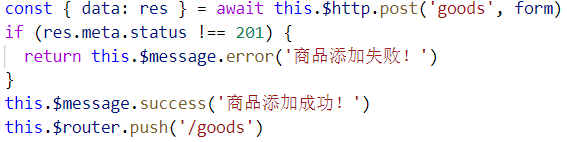
商品内容界面，需要先安装配置vue-quill-editor，当通过表单数据预验证后，就准备对数据发起请求。发起请求之前，需要对表单的数据进行处理，由API接口可知，goods\_cat是以英文逗号的分类列表。由于级联选择器，只允许通过v-model双向绑定到goos\_cat数组。 所以在执行添加函数时，需要在将goods\_cat转化为字符串之前，需要将addForm(添加商品的表单数据对象)深拷贝，实现代码如下：



处理动态参数列表和静态参数列表，代码如下：



发起请求添加商品，商品的名称必须是位移的，代码实现如下



### 分类参数模块的实现

分类参数模块由下拉选项、按钮、卡片视图等组件组成UI界面，实现了选择商品分类后返回商品动态参数和静态属性的功能，还能对参数或者属性进行编辑和修改。该页面首先先调用API接口渲染参数下的可选项，然后为每一行数据提供单独的inputVisuble和inputValue，实现文本框与按钮的切换显示，通过点击按钮，实现参数和属性的展示折叠，并且还能点击“+New tag”添加标签。参数的添加、编辑、删除的实现，逻辑上与用户添加、编辑、删除是一样的。

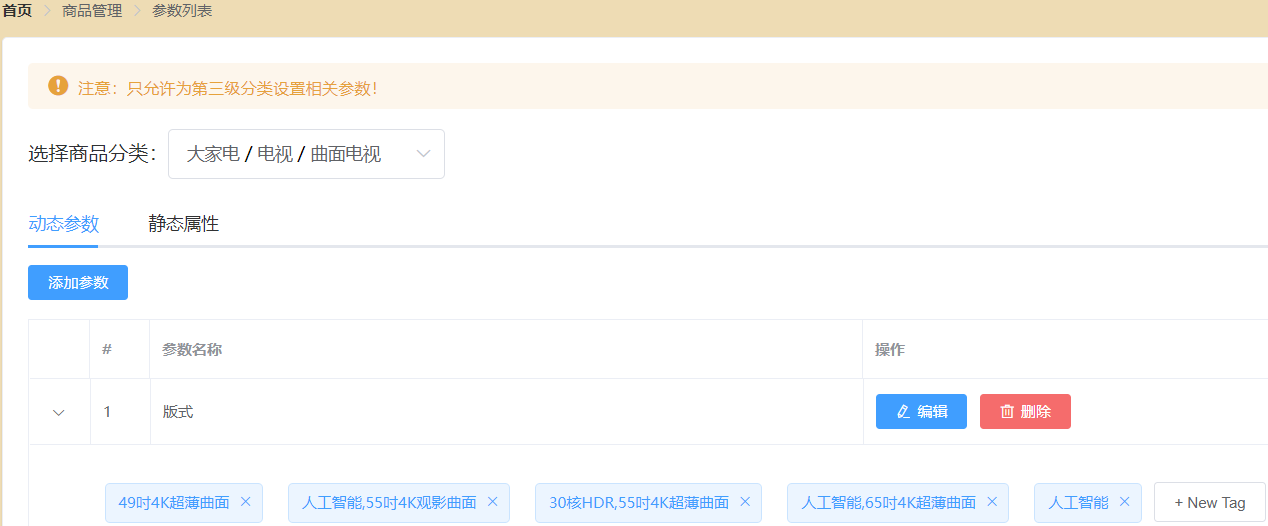


图21 参数列表界面

点击按钮，展示展开行，代码如下：



### 商品分类模块的实现

商品分类模块由面包屑导航、按钮、卡片视图等组件UI界面。分类的添加、编辑、删除，逻辑上与用户添加、编辑、删除一样。



图22 商品分类界面

## 订单管理模板的实现

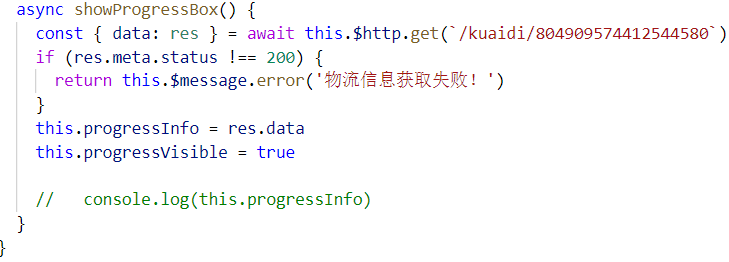
### 订单管理模块的实现

订单列表UI界面如图23所示：



图23 订单列表界面

定位功能，实现的代码如下：



物流进度如图24所示：



图24 图物流进度图

## 数据统计模板的实现

### 数据统计模块的实现

该模块由面包屑导航、卡片视图等组件组成，其中可视化视图需要安装Echarts工具。

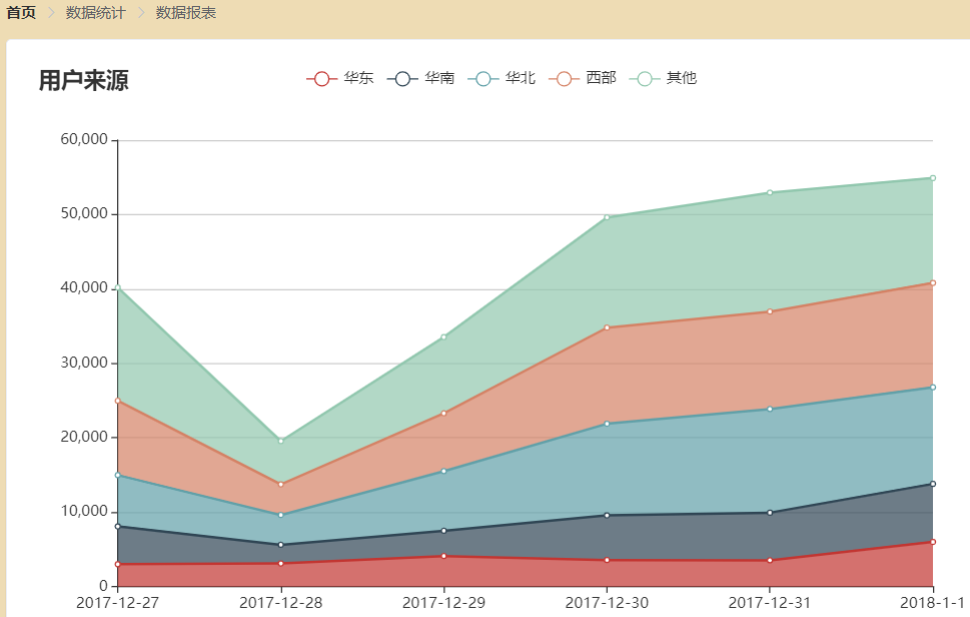


图25 数据报表界面

由API接口可知，基于事件统计的折线图，发起请求成功后，会返回一个折线图的数据，这时，不要把服务器返回的数据交给图表，应该把服务器返回的数据与option数据交给图表，获取折现图数据并渲染图表的代码实现如下：s



## 系统测试

本小结将对电商后台管理系统的功能模块进行测试。测试是软件工程种的重要步骤之一，只有通过测试我们能确认产品的性能、业务是否根需求分析、系统设计的最初标准一致，达到用户使用要求。

电商后台管理系统的功能测试用例表如下：

1. 功能测试

表4-1 登录测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例 | 预置条件 | 测试步骤 | 预测结果 | 是否通过 |
| 登录 | 无 | 1. 输入正确账号及密码，点击登录按钮 2. 2.输入错误账号密码，点击登录按钮 | 1. 登录成功 2. 登录失败 | 是 |
| 重置 | 无 | 点击重置按钮 | 重置成功 | 是 |
| 登出 | 登录 | 点击主页面登出按钮 | 登出成功，返回登录界面 | 是 |

表4-2 用户列表测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例 | 预置条件 | 测试步骤 | 预测结果 | 是否通过 |
| 用户列表UI界面 | 登录 | 点击权限列表 | 进入用户列表界面 | 是 |
| 搜索栏 | 登录 | 搜索系统用户，点击搜索 | 搜索成功，返回搜索结果 | 是 |
| 添加用户 | 登录 | 输入用户名、密码、邮箱、手机，点击确定 | 添加用户成功 | 是 |
| 修改用户状态 | 登录 | 点击修改用户状态按钮 | 修改用户状态成功 | 是 |
| 编辑用户信息 | 登录 | 编辑用户邮箱或手机号，点击确定 | 修改用户信息成功 | 是 |
| 删除用户 | 登录 | 点击删除用户按钮后点击确定按钮 | 删除用户成功 | 是 |
| 角色分配 | 登录 | 点击角色分配按钮，给用户分配新角色 | 角色分配成功 | 是 |

表4-3 角色列表测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例 | 预置条件 | 测试步骤 | 预测结果 | 是否通过 |
| 角色列表UI界面 | 登录 | 点击角色列表 | 进入角色列表界面 | 是 |
| 添加角色 | 登录 | 输入角色名称和角色描述 | 添加角色成功 | 是 |
| 权限分配 | 登录 | 点击权限分配按钮，为角色选中全选 | 权限分配成功 | 是 |
| 编辑角色 | 登录 | 编辑角色名称和角色描述，点击确定 | 编辑角色成功 | 是 |
| 删除用户 | 登录 | 点击删除用户按钮后点击确定按钮 | 删除用户成功 | 是 |
| 权限列表可视化UI界面 | 登录 | 点击角色左边下拉按钮，展开角色所对应的权限列表 | 权限列表可视化UI界面无误 | 是 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例 | 预置条件 | 测试步骤 | 预测结果 | 是否通过 |
| 权限列表UI界面 | 登录 | 点击权限列表 | 进入权限列表界面 | 是 |

表4-4 商品列表测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例 | 预置条件 | 测试步骤 | 预测结果 | 是否通过 |
| 商品列表UI界面 | 登录 | 点击角色列表 | 进入商品列表界面 | 是 |
| 搜索栏 | 登录 | 搜索商品名称，点击搜索 | 搜索成功，返回搜索结果 | 是 |
| 添加商品 | 登录 | 1. 填写商品基本信息（商品名称、价格、重量、数量、分类） 2. 输入商品参数 3. 输入商品属性（主体-品牌、价格-规格、时效-功效） 4. 商品图片上传 5. 输入商品内容 | 添加商品时UI界面无误，商品上传时可图片预览，最后点击添加商品按钮，商品添加成功， | 是 |
| 编辑商品信息 | 登录 | 编辑商品名称、价格、重量，点击确定 | 修改商品信息成功 | 是 |
| 删除商品 | 登录 | 点击删除商品按钮后点击确定按钮 | 删除用户商品 | 是 |

表4-5 分类参数测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例 | 预置条件 | 测试步骤 | 预测结果 | 是否通过 |
| 分类参数UI界面 | 登录 | 点击分类参数 | 进入分类参数界面 | 是 |
| 选择商品分类 | 登录 | 点击下拉框，选择商品分类 | 商品分类选择成功 | 是 |
| 添加动态参数 | 登录 | 点击添加参数按钮，输入动态参数 | 动态参数添加成功 | 是 |
| 编辑动态参数 | 登录 | 输入动态参数 | 修改动态参数成功 | 是 |
| 删除动态参数 | 登录 | 点击删除动态参数按钮 | 删除动态参数成功 | 是 |
| 添加属性 | 登录 | 点击添加属性按钮，输入静态属性 | 静态属性添加成功 | 是 |
| 编辑静态属性 | 登录 | 输入静态属性 | 修改静态属性成功 | 是 |
| 删除静态属性 | 登录 | 点击删除静态属性按钮 | 删除静态属性成功 | 是 |

表4-6 商品分类测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例 | 预置条件 | 测试步骤 | 预测结果 | 是否通过 |
| 商品分类UI界面 | 登录 | 点击分类参数 | 进入商品分类界面 | 是 |
| 添加分类 | 登录 | 输入分类名称，选择父级分类 | 添加分类成功 | 是 |
| 编辑商品分类 | 登录 | 输入商品分类名称 | 修改商品分类成功 | 是 |
| 删除商品分类 | 登录 | 点击删除商品分类按钮后点击确定按钮 | 删除用户商品分类成功 | 是 |

表4-7 订单列表测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例 | 预置条件 | 测试步骤 | 预测结果 | 是否通过 |
| 订单列表UI界面 | 登录 | 点击分类参数 | 进入订单列表界面 | 是 |
| 搜索栏 | 登录 | 搜索订单内容，点击搜索 | 搜索成功，返回搜索结果 | 是 |
| 编辑订单地址 | 登录 | 输入地址信息 | 修改订单地址成功 | 是 |
| 删除订单 | 登录 | 点击删除订单按钮 | 删除订单成功 | 是 |

表4-8 数据报表测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例 | 预置条件 | 测试步骤 | 预测结果 | 是否通过 |
| 数据报表UI界面 | 登录 | 点击分类参数 | 进入数据报表界面 | 是 |

1. 性能测试

渲染时间、总下载时间、每一帧时间是前端页面响应速度的几个关键指标，我采用Chrome Devtool工具对电商后台管理系统进行性能测试，以下性能分析图是由插件的录制功能得到的。

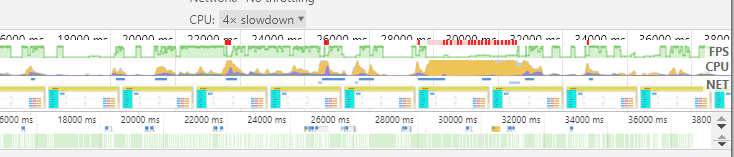


图26 性能分析图

从FPS（帧率）曲线可以看出，FPS基本高于60，说明页面没有卡顿的现象。

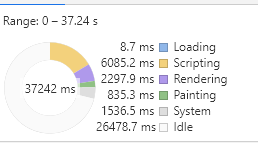


图27 性能分析时间图

从上图来看，系统的加载时间为8.7ms，整体页面渲染时间为2230ms，可以看出，整体的时耗较低，性能较高。

## 本章小结

在基于项目需求于和项目系统实现的前提下，对系统的登录、用户管理等六个模板的功能代码实现进行了详细的说明，对系统的各个页面以及系统的各项功能做出简要描述，以及展示页面内容。对系统的功能测试和性能测试进行了较详细的说明。

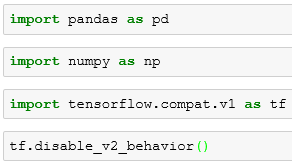
# 电影推荐系统实现



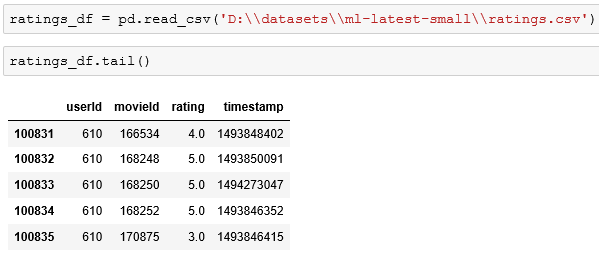
## 数据收集和准备

数据收集地址：<http://group.org/datasets/movielens/>

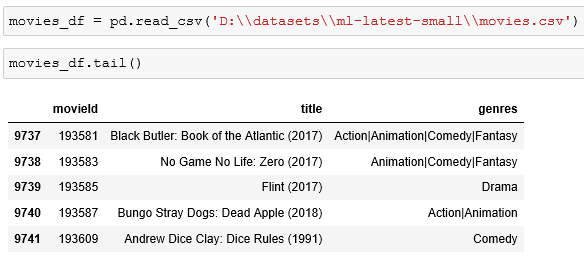
准备必要数据，导入相关包：



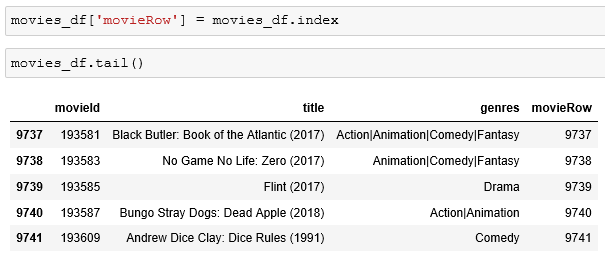
ratings\_df为电影评分矩阵，加载电影评分数据集：



加载电影数据集：



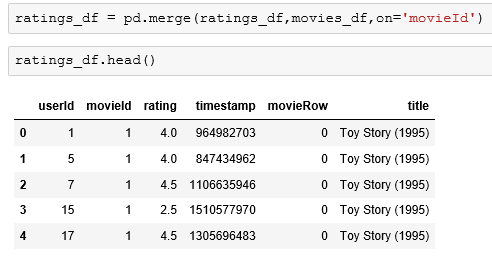
由电影数据集来看，movieId远大于行号，如果使用movieId最大值构建评分表，评分表将非常大的稀疏矩阵，十分浪费内存，所以用行号来标识电影。添加新的属性movieRow行号信息代码如下：



用户是否评分的情况存放在record矩阵中，每个用户对电影的评分情况存放在reting矩阵中。如果评过分，对应位置就是1，否则为0。为了获得这两个矩阵，我们需要筛选movies\_df中的特征：



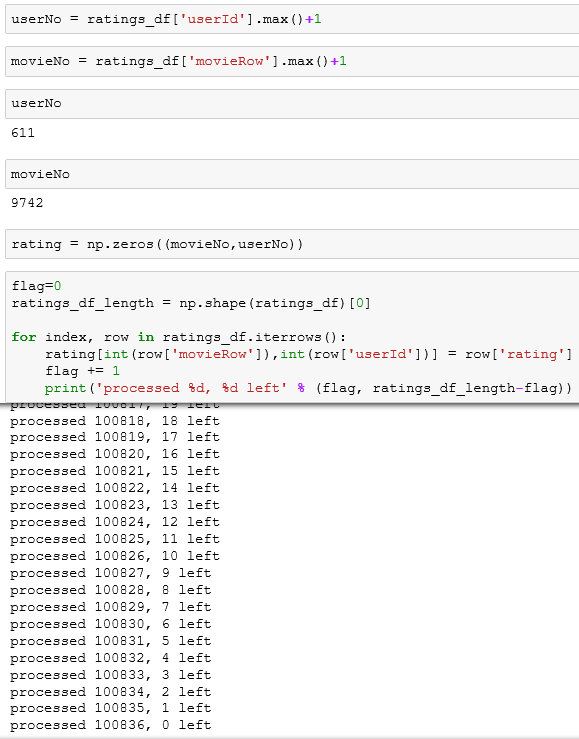
要把ratings\_df中的movieId替换为对应的行号，首先要将movies\_df合并到rating\_df中：



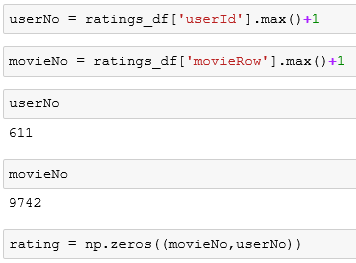
筛选需要用到的特征：



创建电影评分矩阵rating，行数为电影最大编号数量，列数是用户的最大编号数量，实现代码如下：



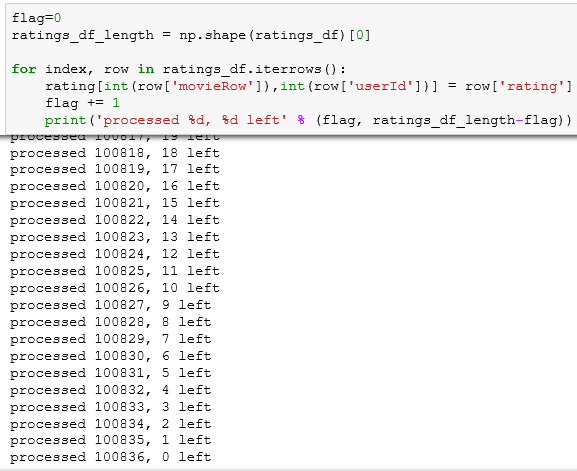
获取用户最大编号和电影最大编号：



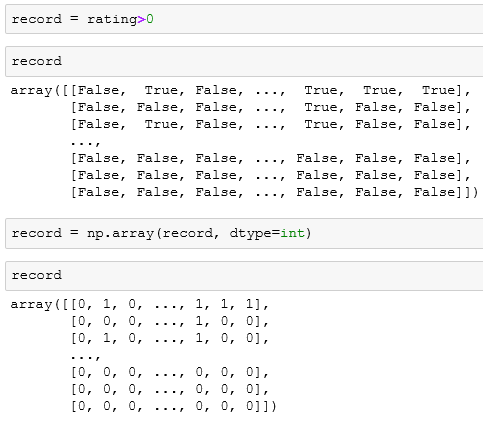
创建rating矩阵，行数为电影最大编号数量，列数为用户的最大编号数量：



把ratings\_df的内容填写到rating矩阵中，定义两个变量，flag用来记录处理进度，ratings\_df\_length是ratings\_df的样本个数。

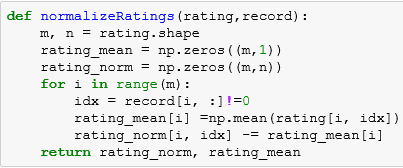


再电影评分表中，所有为0的表示没有评分，大于0为已经评分。为了更直观感受record，将record的布尔值转换为0和1。

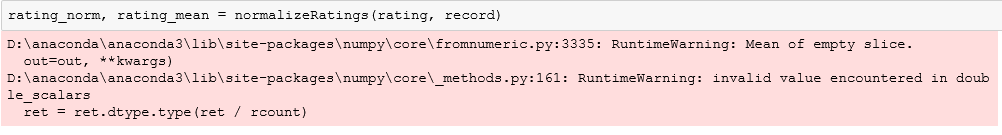


## 构建模型

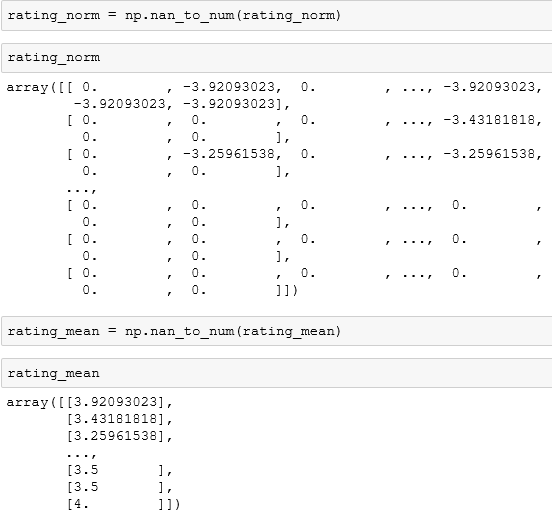
为了使推荐系统的功能更好一些，我们需要定义一个函数normalizeRatings来优化评分取值范围，电影数量用字母m表示，用户数量用字母n表示，之后计算对于每部电影每个用户的评分的平均值，电影的平均评分使用rating\_mean表示，rating\_norm存放处理后的数据，电影评分用户的下标用idx表示。函数代码如下：



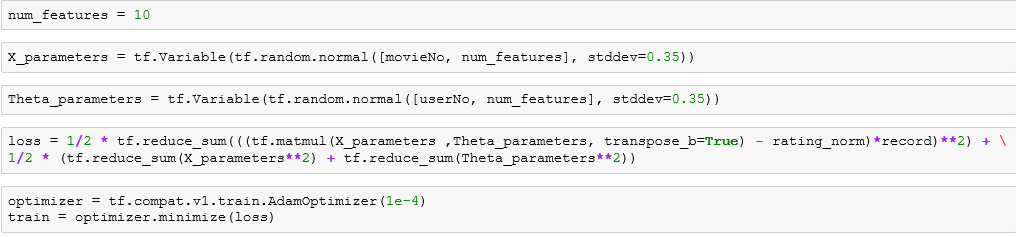
执行函数后，提示RuntimeWarning，原因是数据集中有的行全部是0，因此对于这样的行，使用np.mean计算出来的结果是NaN，如果不处理，将会后面评分的过程出现问题。



如何解决这个问题呢？将NaN变成0，其他不变。



初始化X、θ矩阵，这里产生的参数都是正态分布的随机数。代价的表达式已经再本文的开发技术说明。

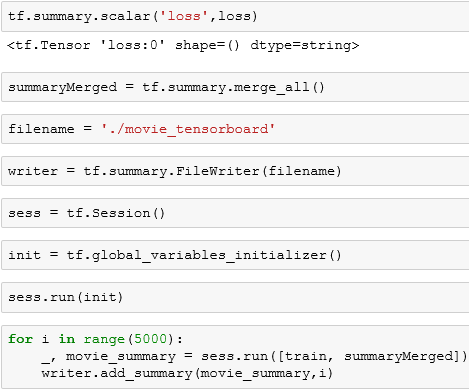


创建优化器和优化目标



## 模型训练与评估

希望能够看待模型的整个训练过程，即希望看到代价值随着迭代次数的增加的变化情况，所以使用tensorboard将整个训练格式化，想要将tensorflow的数据到处，需要用到tensorboard里面有一个叫tf.summary模块，它可以让数据变得可视化。loss是标量。tf.summary.merge\_all()是用户将summary信息汇总



在cmd中找到movie\_tensorboard文件夹的位置，执行tensorboard --logdir=./，用浏览器打开cmd中提示的ip地址。

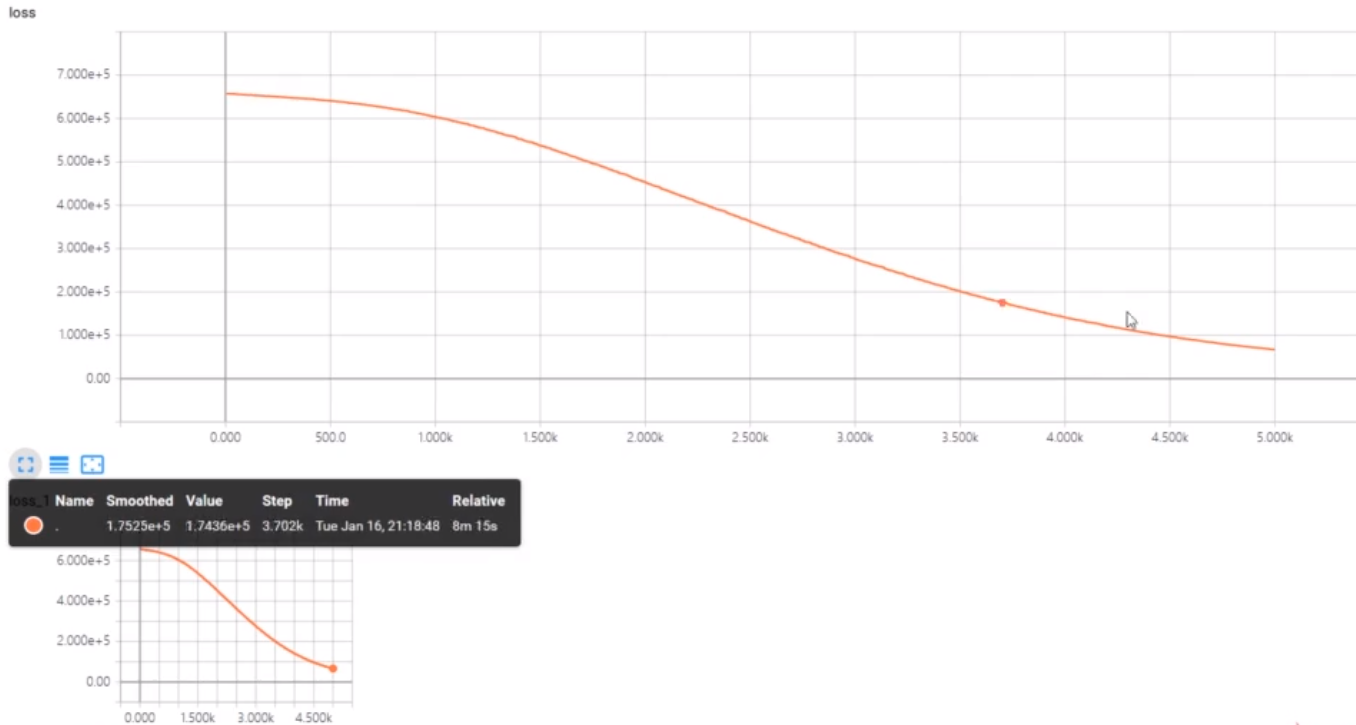
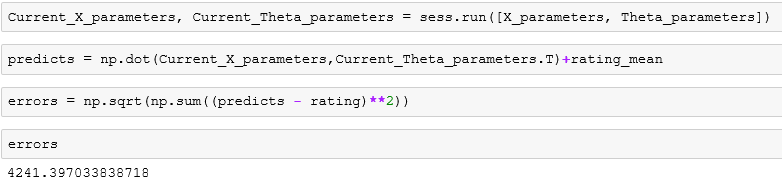


图5.1 模型训练结果图

对模型进行评估，将上述得到的参数将电影评分表填满。第一步，获取X和θ两个矩阵，然后两个矩阵相乘，加上每一行的均值。算出误差值error，随着迭代次数的增加，error回不断减小：



## 构建完整的电影推荐系统

获取用户编号，获取对该用户的电影评分列表，然后向用户推荐评分最高的20部电影，代码以及演示过程如下：



## 本章小结

本章在开发技术的基础上，对实现电影推荐系统的实现进行了详细的描述，实现了基于用户的电影推荐系统。

# 总结与展望

本文主要论述了基于Vue+ElementUI的电商后台管理系统的设计与实现，还简单实现了基于Tensor Flow的电影推荐系统。主要介绍了电商后台管理系统的研究背景、项目需求分析、项目总体设计、系统详细实现等内容，以及电影推荐系统的研究背景、开发现状、所需要的开发技术与代码实现等内容。

本文研究的电商后台管理系统，能够清晰简单地帮助电商管理员对商品进行管理，该系统主要包括登录、用户管理、权限管理、商品管理、订单管理、数据统计六大模块。在系统实现方面，以模块为维度，采用了前后端分离的架构。详细说明了用户登录认证和系统核心模块的实现过程，并且在需求文档给出了系统优化的方案，优化系统速度，在最后对系统进行了一系列系统测试。

电影推荐系统，想要为用户提供二十部适合该用户的口味的最高评分的电影，只需要输入用户编号即可。该系统是基于内容的思想实现的，并且采用了矩阵分解的协同过滤的思想，结合冷启动问题，对系统进行了简单实现。

对于电商后台管理系统，功能很完整，但是还存在很大的提升空间。主要是因为工作量比较大，该系统的很多细节部分做得还不够完善，组件的数量及功能有限，后续还需要不断进行优化；还可以收集更多用户需求，在当前框架中添加新的实现，这样才能扩大系统功能，使其变得更强大与全面，以满足更多的开发项目的使用需求；Vue框架也在不断更新完善中，系统要跟进框架的更新，使系统更有利于维护。对于电影推荐系统，基础功能已经实现，但是该比较简单，将电影推荐系统修改为商品推荐系统，并且应用到电商后台管理系统，是对该系统未来的一个展望。

# 【参考文献】

致谢

转眼间，四年的大学生活即将结束，也意味着我的学生时代的结束。回望过去四年的生活，自己收获颇多，不仅收获了知识，还认识了很多良师益友，他们在我遇到困难时为我解惑，在我需要帮助的时候及时伸出援手。在大学期间遇到的人和事都是我医生中最宝贵的财富。

由衷地感谢李斌老师在这次毕业设计中，在选题或是论文撰写上给了我许多宝贵的意见，也在我遇到困难的时候帮助我解决问题。 我的毕业设计及论文撰写的完成，都离不开李老师的悉心指导。

感谢我的同学，和我一起度过四年的本科生活，在我的学习和生活上给予了我很大的帮助。

感谢我的父母对我的养育之恩，给我无尽的鼓励与支持。

感谢深圳大学和深圳职业技术学院为我提供优秀的学习环境。

最后，感谢所有在百忙之中评阅本论文的评委老师。

E-commerce background management system and intelligent recommendation system based on Vue + ElementUI

【Abstract】Since the twentieth century, e-commerce platforms have developed at a rapid rate, with thousands of commodities operated by e-commerce, making the information management function of e-commerce background more and more important. The e-commerce background management system, one of the projects in this thesis, uses the scaffolding architecture provided by Vue official to build the system, and the front-end page is implemented using ElementUI. This article mainly introduces the system's functional requirements and its overall technical architecture. It briefly discusses the characteristics of Vue's MVVM model, the advantages and disadvantages of Vue, axios and webpack network packaging tools. And explain the techniques used in practice. Not only that, I also introduced the principles of content-based recommendation system, matrix decomposition-based recommendation system and commodity-based collaborative filtering recommendation system by using TensorFlow's content-based movie recommendation system.

**【Key words】**Vue.js；ElementUI；TensorFlow; Matrix decomposition; collaborative filtering；

指导教师：李斌副教授

1. 毛炎. 基于Vue.js框架的Web方言地图的设计与开发[D].武汉大学,2018. [↑](#endnote-ref-2)
2. 王胜,张靖. 基于Vue.js高速路政管理系统的设计与实现[J]. 电脑知识与技术,2017,13(21):86-88+101. [↑](#endnote-ref-3)
3. 焦鹏珲. 基于SpringBoot和Vue框架的电子招投标系统的设计与实现[D].南京大学,2018. [↑](#endnote-ref-4)
4. 吕品田. 基于vue.js的共享空间平台的设计[J]. 数字通信世界,2018(03):199+280. [↑](#endnote-ref-5)
5. 王志任. 基于Vue.js的开发平台的设计与实现[D].广东工业大学,2018. [↑](#endnote-ref-6)
6. 柴青山. 基于MVVM模式的Vue.js框架在物流软件自动化测试系统中的应用研究[D].北京邮电大学,2019. [↑](#endnote-ref-7)
7. 易剑波. 基于MVVM模式的WEB前端框架的研究[J]. 信息与电脑(理论版),2016(19):76-77+84. [↑](#endnote-ref-8)
8. 王鹤琴,朱珍元. 基于MVVM模式的Web开发研究[J]. 菏泽学院学报,2019,41(02):7-13. [↑](#endnote-ref-9)
9. 1. cool\_cl. axios中文文档[EB/OL]. https://www.jianshu.com/p/7a9fbcbb1114, 2018.02.16.

   [↑](#endnote-ref-10)
10. axios中文网[EB/OL]. http://www.axios-js.com/, 20204,26 [↑](#endnote-ref-11)
11. webpack中文文档[EB/OL]. https://www.webpackjs.com/concepts/, 2020,4,26 [↑](#endnote-ref-12)
12. 刘翔宇. 基于Vue的数据可视化系统的设计与实现[D].北京邮电大学,2018. [↑](#endnote-ref-13)
13. 薛定谔的panda.浅谈webpack打包原理. https://blog.csdn.net/u014168594/article/details/77198729, 2017-08-15 [↑](#endnote-ref-14)
14. Shijie Hao, Yuan Zhou, Yanrong Guo. A Brief Survey on Semantic Segmentation with Deep Learning[J]. Neurocomputing, 2020 [↑](#endnote-ref-15)
15. 费宁；张浩然.TensorFlow架构与实现机制的研究[J].计算机技术与发展,2019,29(9) [↑](#endnote-ref-16)
16. Liyang Hao, Siqi Liang, Jinmian Ye, Zenglin Xu,TensorD: A tensor decomposition library in TensorFlow[D],Neurocomputing,Volume 318,2018,Pages 196-200. [↑](#endnote-ref-17)
17. Abadi M、BarhamP,ChenJ,ctal.Tensor Flow:ASyst cm for L argc-S calc Machin cLearning[C] //OSDI.2016,16:265-283. [↑](#endnote-ref-18)
18. 李海龙. 基于概率矩阵分解深度推荐算法的研究[D].燕山大学,2018. [↑](#endnote-ref-19)
19. 李新炜,殷韶坤. 深度学习在文字识别领域的应用[J]. 电子技术与软件工程,2018(24):40. [↑](#endnote-ref-20)
20. 鞠云霞,王希常,陈祥喜,郑伟. 基于TensorFlow平台的卷积神经网络对OMR识别问题研究[J]. 无线互联科技,2019,16(04):54-57. [↑](#endnote-ref-21)