****

**本科毕业设计**

**（2022届）**

|  |  |
| --- | --- |
| **题 目** | **GitHub新手任务的采集与分析系统的设计与实现** |
| **学 院** | **计算机学院** |
| **专 业** | **软件工程** |
| **班 级** | **18052711** |
| **学 号** | **18051230** |
| **学生姓名** | **王武俊** |
| **指导教师** | **陈洁** |
| **递交日期** | **2022年5月** |

**诚 信 承 诺**

我谨在此承诺：本人所写的毕业论文《GitHub新手任务的采集与分析系统的设计与实现》均系本人独立完成，没有抄袭行为，凡涉及其他作者的观点和材料，均作了注释，若有不实，后果由本人承担。

**承诺人（签名）：**

**2022 年 5 月 10 日**

# 摘 要

GitHub是一个目前在全球都很受欢迎的开源的在线资源托管平台，其受众主要是技术从事人员，拥有着庞大的代码资源以及其丰富的相关问题讨论。对于GitHub的研究可以更好地宣扬开源精神，从而进一步促进计算机行业的发展。GFI（good first issue）是新手上手GitHub的重要途径之一，截止到目前为止，GFI约有20余万条，其中，有约1/4的GFI尚未被解决。本课题重点研究GitHub的GFI, 通过对其研究，构建基于Random forest(随机森林)分类模型，对GitHub中的GFI(Good First Issue)进行收集以及分析，探寻适合新手解决的GFI，从而帮助新手更好的融入GitHub的圈子中。本文研究步骤主要如下：

1. 基于GitHub Restful API收集459390条GFI，并选取了其中含有PR（Pull Request）的9979条作为数据集。采用了隐式标签的方式，得到正样本（分类问题中的适合新手解决）1247条（12.5%），负样本8732条（87.5%）。
2. 通过收集issue的基本信息（标题长度、评论数等基本信息）以及对收集到的issue的标题和内容作文本可读性分析和LDA主题分析得到共17个特征。
3. 对数据集中的特征进行特征选择，最终确定13个特征，并通过validation\_curve（验证曲线）得到模型的调参范围，用于后续分类模型的训练。
4. 基于sklearn建立随机森林模型，使用GridSearchCV（网格搜索）对上述验证曲线中的参数进行训练，最终模型在训练集和测试集的最优得分（指标为roc\_auc）分别是79.70%和76.29%。

关键词：GitHub; Good First Issue; GFI; 随机森林；LDA主题模型

# ABSTRACT

GitHub is an open source online resource hosting platform that is currently very popular around the world. Its audience is mainly technical personnel, and it has a huge code resource and a wealth of related issues to discuss. Research on GitHub can better promote the spirit of open source, thereby further promoting the development of the computer industry. GFI (good first issue) is one of the important ways for beginners to get started with GitHub. So far, there are about 200,000 GFIs, of which about 1/4 of GFIs have not been resolved. This topic focuses on the GFI of GitHub. Through its research, it builds a classification model based on Random forest, collects and analyzes the GFI (Good First Issue) in GitHub, and explores GFI that is suitable for novices to solve, so as to help novices to update Well integrated into the GitHub circle. The main steps of this study are as follows:

1. Based on the GitHub Restful API, 457052 GFIs were collected, and 9979 GFIs containing PR (Pull Request) were selected as the dataset. Using the implicit label method, 1247 positive samples (12.5%) and 8732 negative samples (87.5%) were obtained (suitable for novice solutions in classification problems).
2. A total of 17 features are obtained by collecting the basic information of the issue (basic information such as title length, number of comments, etc.), and performing text readability analysis and LDA theme analysis on the title and content of the collected issue.
3. Feature selection is performed on the features in the data set, 13 features are finally determined, and the parameter adjustment range of the model is obtained through the validation\_curve (validation curve), which is used for the subsequent training of the classification model
4. Build a random forest model based on sklearn, and use GridSearchCV (grid search) to train the parameters in the above validation curve. The optimal scores of the final model in the training set and test set (the indicator is roc\_auc) are 79.70% and 76.29%, respectively.

**Key words：**GitHub; Good First Issue; GFI; Random Forest; LDA topic model

目 录

[1绪论 1](#_Toc8979)

[1.1 研究背景及研究意义 1](#_Toc6445)

[1.2 国内外研究现状 1](#_Toc2847)

[1.2.1 分类模型方面的研究 1](#_Toc26492)

[1.2.2 基于GitHub的研究 2](#_Toc28411)

[1.2.3 基于GitHub GFI的研究 2](#_Toc10536)

[1.3 论文研究内容 3](#_Toc21839)

[1.4 论文组织结构 4](#_Toc6849)

[2 相关理论基础 5](#_Toc7910)

[2.1 GitHub网站简介 5](#_Toc15989)

[2.2 LDA主题模型 6](#_Toc8064)

[2.3 文本向量化 7](#_Toc12838)

[2.3.1 文本向量化简介 7](#_Toc14864)

[2.3.2 词袋模型 8](#_Toc32558)

[2.4特征选择 8](#_Toc1012)

[2.4.1 特征选择简介 8](#_Toc31027)

[2.4.2 互信息 9](#_Toc20206)

[2.5随机森林分类模型 10](#_Toc11459)

[3 数据获取与处理 12](#_Toc14311)

[3.1 获取原始数据 12](#_Toc9091)

[3.2 数据预处理 14](#_Toc2242)

[3.2.1 文本预处理 15](#_Toc13120)

[3.2.2 定义标签 16](#_Toc28101)

[3.3 LDA主题分析 16](#_Toc30245)

[3.3.1 模型准备 16](#_Toc17260)

[3.3.2 参数优化 18](#_Toc15827)

[3.3.3 结果及预测 21](#_Toc10138)

[4 基于Random Forest的分类模型实现GFI的分类 22](#_Toc8257)

[4.1 模型的构建 22](#_Toc4483)

[4.1.1 特征选择 22](#_Toc29833)

[4.1.2 参数优化 23](#_Toc3530)

[4.1.3 模型构建与结果 25](#_Toc1791)

[4.2 模型比较 28](#_Toc26788)

[4.3 模型分类 29](#_Toc18264)

[4.4 模型落地 29](#_Toc7682)

[5 总结与展望 33](#_Toc2212)

[5.1 研究总结 33](#_Toc26634)

[5.2 研究展望 34](#_Toc27234)

[6 致谢 35](#_Toc18030)

[7 参考文献 36](#_Toc31487)

# 1绪论

## 研究背景及研究意义

随着社会科学技术的不断普及和发展，计算机技术及其应用已经得到了普遍的推广和使用，成为世界各国建设国民经济体系不可或缺的一环。GitHub 是一个资源托管平台，一个基于Git协议而建立起来的Web网站，即面向开源软件、开源社区，也面向私有软件。目前，据GitHub年度官方报告显示，GitHub已经成为世界上最大的代码托管平台，其用户遍布全球各地，多达7300万+，仓库数目突破1亿，是计算机研究的首选。GFI（good first issue）是新手上手GitHub的重要途径之一，截止到目前为止，GFI约有20余万条，其中，有约1/4的GFI尚未被解决。本课题重点研究GitHub的GFI, 通过对其研究，构建分类模型，对GFI进行分类，找出适合新手解决的GFI,意在帮助新手更快更好地融入到GitHub的圈子中。

随着信息技术的迅猛发展和互联网生态环境的日益变化，用户量和用户活跃度等指标已经是软件产品稳定性的重要衡量标准之一，GitHub平台亦是这样。保持大量的新人的涌入对于GitHub来说至关重要**[1]**。GitHub官方为此推出了good first issue的概念，项目开发者通过标记issue为good first issue来向issue参与者表示issue是否适合新手解决，但是，很多新手再尝试后仍然未解决问题，这不仅仅大幅地降低了新手融入GitHub圈子的积极性，也在一定程度上让项目开发者的努力付之东流。

在过去的十年中，已经有越来越多的项目在使用官方推出的这种GFI标签的方式来促进新手入圈，但是，通过一个issue还是需要几天甚至长达近一个月才能被解决。不仅如此，还会有项目的难易程度，新手的技能水平参差不齐等多种相关因素导致这一官方机制的不合理性。于是。本课题诣在从多角度多因素探究适合新手解决的GFI,从而帮助新手更好的参与到开源项目的贡献中来。

## 国内外研究现状

### 分类模型方面的研究

分类问题是数据挖掘处理中的一个主要部分,在计算机知识应用领域,分类问题一般也被看作是监督型学习(supervised learning),换言之,分类问题的主要任务就是通过检查已知样本的一些性质,确定一种新的样本属于哪一种已知的类型。

按照分类结果的数量来划分，可以将分类问题归为二元分类和多元分类两个板块；按照分类对象的属性来划分，还可以将分类问题归为图像分类和文本分类等。目前，比较通用的用于分类的模型算法有这么几个：1.决策树、2.人工神经网络、3.KNN算法、4.支持向量机（SVM）、5.朴素贝叶斯、6.Adaboost、7.逻辑回归、8.随机森林、9.GBDT 等。

基于分类问题的研究也数不胜数。于卫红利用3种文本特征和5中分类模型来探究多类别文本下分类方法效果的比较，最终发现，使用词嵌入这种方式来进行分类对模型速度和效果均有显著优势；李培强探究了特征选择和分类算法，并基于互信息法构造出新的互信息评估函数，并且利用k中心点和粗糙集构建的KNN分类算法，在一定程度上减少了原KNN算法的计算耗时，从时间层面提升了对文本分类的效率；罗康洋使用SVM模型和SMOTE算法，实现了一种表现在大量特征数据的不平衡的数据集中的分类算法，提升了分类器的分类效果和分类性能。

### 基于GitHub的研究

GitHub拥有着庞大的用户群体和数据量，是目前国内外机器学习爱好者的主战场之一，针对GitHub中不同层面不同场景的问题，均有很多学者进行了深入的研究。在当前的研究中，大部分研究者是基于上述分类中的SVM，逻辑回归和随机森林来完成问题的研究。

Thomas Zimmermann等人针对GitHub的issue中的bug类型进行了深入的研究，构建了逻辑回归分类模型，实现了对于bug类型的issue“是否会重新被open”进行了研究、预测与验证**[2]**；Ali Sajedi Badashian等人基于SVM支持向量机和朴素贝叶斯这两个分类模型，对GitHub中的bug类型进行了分类、分析与验证**[3-4]**；Riivo Kikas等人通过研究证明了在对于GitHub issues的模型的研究中，动态的特征属性和上下文相关的特征属性在分类或预测模型中更关键、更有利，对于实现模型的长期预测具有较大的帮助**[5]**。

还有很多学者也在针对于GitHub的研究中发布了很多很有意义的成果**[6-9]**，这些研究在促进互联网领域发展和GitHub开源精神的发展道路上发挥了不可或缺的作用，在一定程度上对于GitHub平台、对于计算机新手从事着都有着很重要的意义和影响，同时也对本课题的研究起到了一定的思想启蒙作用。

### 基于GitHub GFI的研究

GFI是GitHub研究中的重点内容之一，good first issue标签是项目开发者成员手动标记的产物，具有一定的不确定性和不统一性，没有一个具体的评价标准用于识别issue是否适合新手来解决。

在进行本课题的研究之前，已经有众多的学者投入到GitHub的GFI的研究中,并且都得出一些比较有意义的结果。基于前人在此领域的研究成果，有利于本课题实现对GFI分类模型从特征选择、训练调参、训练模型等方面的选择和操作。

黄悦凯等结合这几种模型，采用5组交叉验证，从文本静态特征、文本复杂程度以及语义化等多个层面，采用了多种分类模型，采取了多种准确率指标进行对比，分析出了预设特征中对模型准确率影响最大的特征集合**[10]**；XinTan等人使用逻辑回归的分类模型，从文本、技术、项目、标签四个层面共10个细化指标，跑出了得分非常高的模型**[11]**。除此之外，还有很多优秀的研究者在GitHub GFI的研究中取得了十分不错的成果。

总结目前国内外的研究成果发现，在现有针对GitHub GFI的研究中，研究者的核心方式大多是采用分类模型分析的方法，结合具体情境确定特征、运用特定分类模型，提升模型的处理效率进而实现分类。

## 论文研究内容

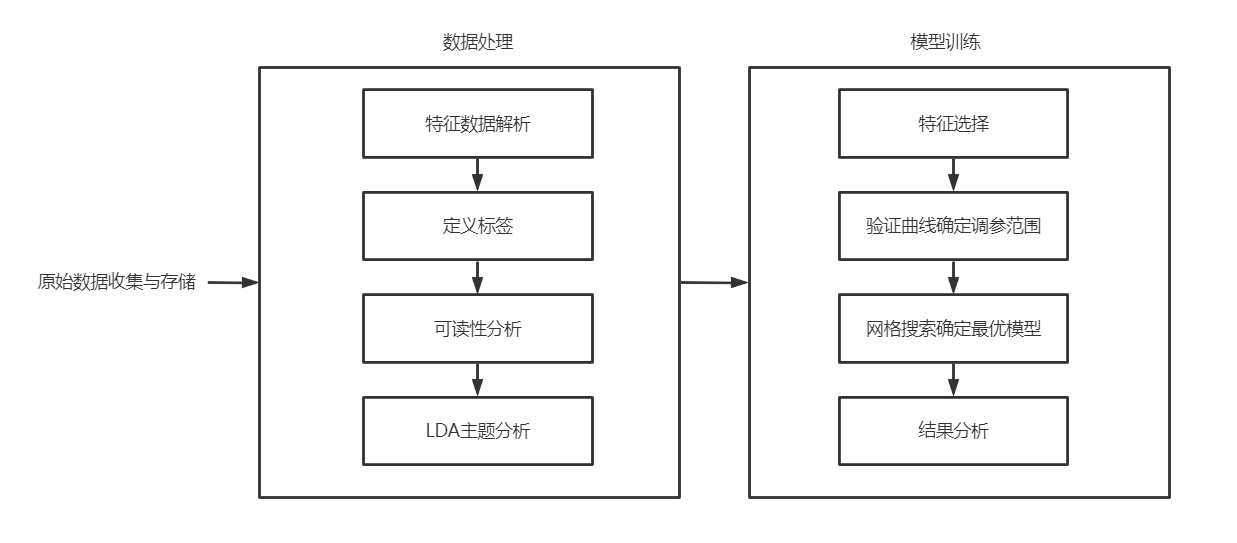
本论文通过对GitHub中的GFI进行收集、处理、分析，构建随机森林分类模型，从多个基本特征以及对issue的title和body进行可读性分析、主题分析。

图1-1 论文研究步骤

如图1-1 所示，本文的研究主要通过以下几个具体的方面进行展开：

1. 了解GitHub网站相关知识及开发者文档，通过GitHub Restful Api (<https://docs.github.com/cn/rest>)，收集自2000年以来的所有含有good first issue标签的issue，并对收集到的数据进行处理，存入到mysql数据库中。
2. 利用收集到的的原始数据中从静态指标和语义指标层面，结合可读性分析和LDA主题分析的方法，解析出模型训练需要的各个特征数据。
3. 使用特征选择的方法，从数学层面进行相关性分析，去除特征和标签之间相关性不大的特征，确定最终模型训练特征；使用验证曲线的多种方法，通过观察参数影响趋势，确定模型训练的调参范围。通过特征选择和验证曲线的方式，大幅度减少模型训练耗时、提升模型性能。
4. 使用sklearn框架中的随机森林模型以及网格搜索的包进行模型训练，采用roc\_auc的准确率指标，得到最优模型，进而分类和模型落地。

## 论文组织结构

本论文的主要组织结构如下：

第一章： 绪论。从科学技术发展、计算机的发展等多个角度交代本课题的研究背景以及研究意义，阐述本课题在实际应用领域的必要性。通过阅读分类模型、GitHub以及GFI相关文献后，介绍国内外这一领域相关学者目前的研究进展和研究方法。简要介绍本论文的研究内容和组织结构。

第二章：相关理论基础。这一章将对在本课题研究之前了解的一些关于课题中使用到的技术和方法进行初步的介绍，有利于读者对下述论文的阅读。主要会简单介绍GitHub网站，以及对LDA主题模型、特征选择和随机森林模型的概念。

第三章：数据获取与处理。这一章会介绍数据处理中的四大步骤，会向读者阐述本课题是如何获取数据、数据获取后的统计分析和可视化、存储数据的方式以及对数据进行可读性分析和LDA主题分析的过程。会重点介绍标签的定义方法和LDA主题模型的调参过程。

第四章：基于Random Forest的分类模型实现GFI的分类。这一章会介绍特征选择，以及验证曲线在参数优化中的具体使用，对模型训练结果进行可视化分析。最后会介绍本课题模型落地的应用和基本使用方式。

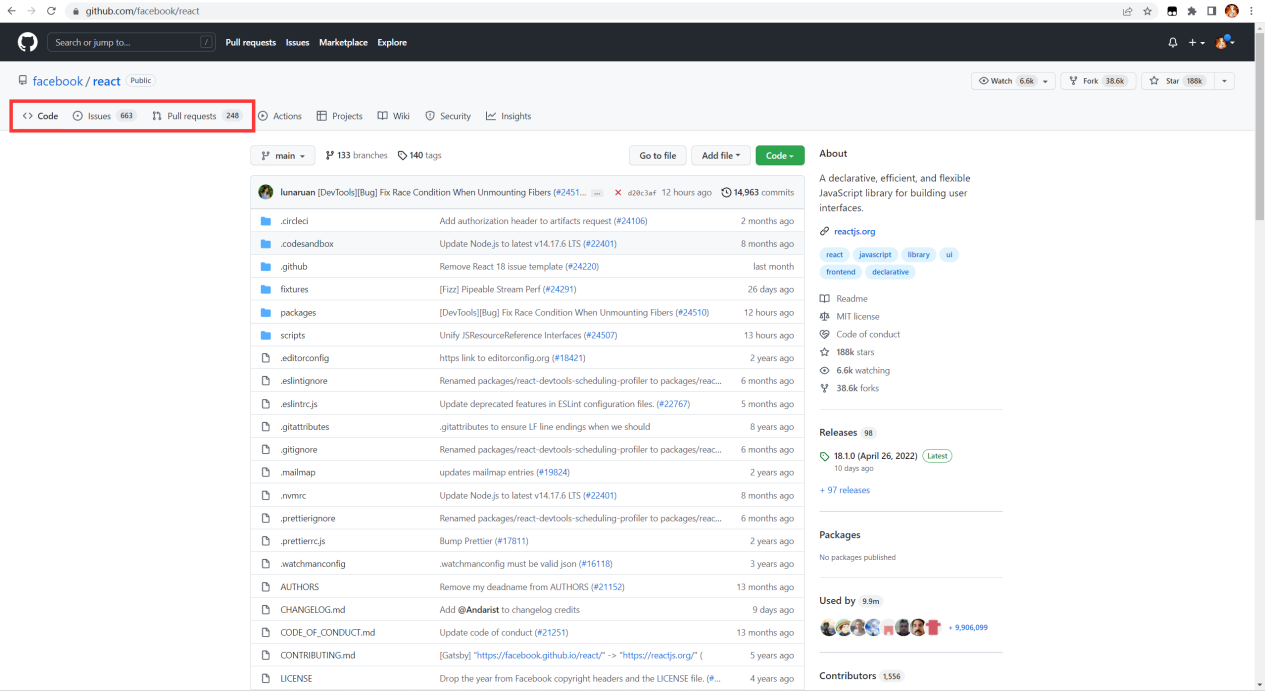
第五章：总结与展望。这一章，我对本课题的研究方法、研究成果进行综合性的总结，并对研究过程中的缺陷和不足进行深刻的反思。同时，结合本课题的研究成果，探究未来针对GitHub GFI的进一步研究的可能性。

# 2 相关理论基础

本章主要介绍GitHub网站、LDA主题模型、文本向量化、特征选择和随机森林模型这五个方面的相关的理论基础，作为论文前置的学习，可以帮助读者进一步理解后续的研究逻辑。

## 2.1 GitHub网站简介

GitHub是基于Git版本工具实现的一个在线的开源平台，目前用户量超1300万+，其主要受众用户为IT行业人员，主要托管资源为项目代码，除此之外，GitHub还提供GitHub Pages、Gist、Student Developer Pack等多种功能。一个Git的Web平台，可以轻松的实现本地文件与远端资源的同步，通过多人基于Git协议对同一份资源进行管理、修改等，可以轻松记录每一个操作成员的操作记录，从而高效地进行项目开发和记录跟踪。

图2-1 GitHub项目界面图

如图2-1所示，本课题设计到的GitHub中的概念为Issues和Pull requests。issue（议题）是由项目成员或者GitHub成员对此项目提出建议、bug等的一种手段；pull requests是一个允许GitHub成员提交代码更改的途径。通过issue的提出与解决、pull requests的提出与合入可以聚集众人的力量，发扬开源精神，促进项目的进一步完善和发展。

## 2.2 LDA主题模型

主题模型，业界称为Topic Model，是自然语言处理和机器学习领域的用来预测文档的主题分布的一种统计模型。通过对文档的词频统计，根据统计的信息来判断当前文档包含了哪些主题以及每个主题下的占比。目前常用的主题模型有：潜在语义分析（LSA）、概率潜在语义分析（pLSA）、潜在狄利克雷分布（LDA）以及基于深度学习的lda2vec，本课题使用的LDA主题模型。

LDA主题模型,主要目的是用来对文档的主题分布做出预估。对一个文本,在经过向量化后,主题模型将能够提供与它相对应的每一个主题的概率,并以此为文本的主题聚类或文本分级作基础。LDA中涉及到很多常用的数学概念，例如二项分布、多项分布、马尔可夫链等，如果你想对LDA作进一步的了解，可以先行了解一下这些数学相关的理论基础。

LDA的图模型结构如图2-2所示：

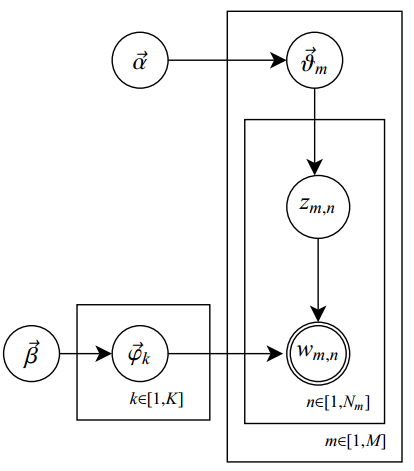


图2-2 LDA图模型结构

其中的参数概念如下：

表2-1 LDA图模型结构参数解释

|  |  |
| --- | --- |
| 参数 | 含义 |
|  | 狄利克雷分布 |
|  | 狄利克雷分布 |
|  | 主题的单词分布 |
|  | 主题分布 |
|  | 主题 |
|  | 单词 |
|  | 续表 2-1 |
| 参数 | 含义 |
|  | 单词个数 |
|  | 主题个数 |
|  | 文本个数 |

在LDA模型中，一篇文档是这样生成的：

（1）利用狄利克雷分布取样，从中生成文档 i 的主题分布

（2）利用主题的多项式分布取样，并生成文档i第j个词的主题

（3）利用狄利克雷分布取样，从中生成主题对应的单词分布

（4）利用单词的多项式分布采样，并最终生成词语

## 2.3 文本向量化

### 2.3.1 文本向量化简介

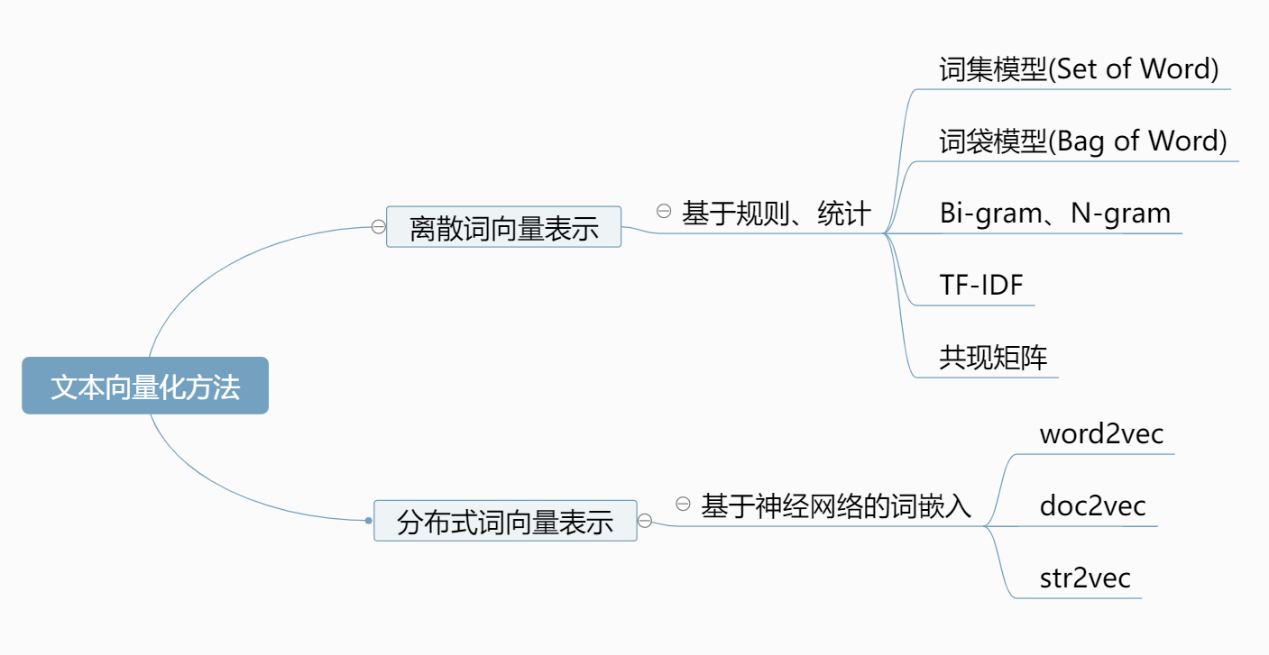
文本向量化又叫“词向量模型”或者“向量空间模型”，也就是将文本用计算机可以识别的实体向量来表示，根据文本的细致程度不一致，又可以将文本特征划分为字、词、句子、篇章等层次。文本向量化的方法也叫做“词嵌入”（word embedding）方法,也就是说将某个文本“嵌入”到空间中，用空间中的向量来表示这个词。

图2-3 文本向量化方法思维导图

如图2-3所示，文本向量化目前大致可以从离散词向量和分布式词向量两个方向进行研究。针对离散词向量表示，一般基于规则、统计词频，代表性的方法有词集模型、词袋模型、Bi-gram、N-gram、TF-IDF、共现矩阵等；针对分布式词向量表示，一般基于神经网络来进行词嵌入计算，代表方法有word2vec、doc2vec、str2vec。

本课题基于sklearn的gensim中的corpora方法来进行文本向量化操作，底层使用的是词袋模型的方法，通过统计各词在句子中出现的次数用来作后续的文本向量化计算。

### 2.3.2 词袋模型

词袋模型（Bag of Word,BOW Model）最早是自然语言处理和信息检索领域的概念，其主要利用统计的方法，对文档进行分词、标记操作后，记录文档对应的词典以及词典中每一个词的频率。

例如有两个文档：

A: I like cat,you like too

B: I do not like dog

词袋模型会按照记录文档字典如下：

{‘I’:0,’like’:1,’cat’:2,’you’:3,’too’:4,’do’:5,’not’:6,’dog’:7 }

构建的字典中有8个单词，每一单词有一个唯一的索引，对于A和B两个文档便可以分别表示为：

A: [1,2,1,1,1,0,0,0]

B: [1,1,0,0,0,1,1,1]

向量的生成与文档中单词的顺序无关，而是取决于字典的顺序以及句子中单词出现的频率，所以Bow模型在某种意义上可以看成是一种统计直方图。

从上述过程来看，可以了解到词袋模型仅仅是从统计学的角度思考问题并得出结论，并没有考虑句子中的上下文环境，没有将语义的因素考虑在内。但是这并不影响词袋模型在文本向量化过程中的地位和优秀效果。

## 2.4特征选择

### 2.4.1 特征选择简介

特征选择（Feature Selection）是特征工程里面的一个很关键的方面,它主要的任务就是特征集中查找出最优的特征子集。主要作用是能够降低特征数量、降维,使模型的泛化能力更强,从而减少模型过拟合的问题,还能够加深人们对特征数量与特征值之间关系的认识。而更好的特征字段集合也可以在一定程度上改善模型的特性,更能有助于人们理解数据的基本特征与底层构造信息。好的特征选择能够在一定程度上提高模型的性能，更能帮助我们了解数据的特点和底层结构信息，这对于进一步改善模型以及算法都有着不可磨灭的作用。

据特征选择的形式，可以将其分为以下三大类:

1. 过滤法(filter)。按照特征数据之间的离散性或者是关联性等来分别对于这些不同的特征的特性进行过滤和赋分,选择过滤的阈值或者仅仅是根据最终可以选择或者保留过滤的特征数目等来分别进行过滤。
2. 包装法(wrapper)。按照目标函数的实际预测和效果进行打分,每次可以选择增加若干个特征值和选择删除掉若干个特征。
3. 嵌入法(embedded)。先要利用即将进行训练时的模型算法对系统进行一个预先训练,以便最终得出对每一种特征的权值系数,根据这个系数排序消除。这在某种层度上属于过滤法的一种，例如sklearn框架中提供的决策树和随机森林的模型均提供feature\_importance\_的方法可以查看训练结束以后模型的参数重要性排序。

过滤法主要包含方差过滤和相关性过滤，其中相关性过滤的方式主要有卡法校验（chi2）、F检验（f\_classif）和互信息法（mutual\_info\_classif）。本课题采用第一种过滤法中的相关性过滤下的互信息法进行特征选择。

### 2.4.2 互信息

互信息（Mutual Information,简称MI）,是两个特征之间的依赖性的衡量指标。互信息法是用来衡量每个特征与标签之间的任意关系的过滤的方法，这里的任意关系包括线性关系和非线性关系。

互信息是基于熵提出的一种概念。

熵的定义如下：

 （1）

互信息与熵的关系如图2-4所示，定义如下，：

 (2)

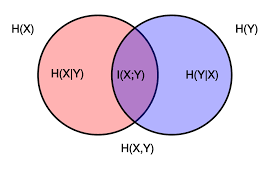


图2-4 互信息公式示意图

公式中的参数解释如下：

表2-2 互信息公式参数含义以说明

|  |  |  |
| --- | --- | --- |
| 参数 | 含义 | 说明 |
| H(X) | 熵 | 事件的信息 |
| H(X|Y) | 条件熵 | 表示在知道事件X的情况下，此时再知道事件 Y 可以提供给我们的信息 |
| H(X,Y) | 联合熵 | 表示知道事件X和Y之后可以提供给我们的信息 |
| I(X;Y) | 互信息 | 表示事X和Y共同提供的信息；也可以理解为知道了事件 X可以对事件Y提供多少信息，反之亦然 |

分类问题中，使用互信息理论对特征进行选择是基于以下的假设来进行的：在某个特定的分类中出现的频率大，但在其他分类中出现的频率比较小的词条，与该类的互信息比较大。通常用互信息作为特征词和类别之间的测度，如果特征词属于该类的话，它们的互信息量最大。由于该方法不需要对特征词和类别之问关系的性质作任何假设，因此非常适合于文本分类的特征和类别的配准工作。

## 2.5随机森林分类模型

了解随机森林模型之前，先简单介绍下集成学习以及集成算法的概念。

集成学习是机器学习中对应个体学习的一个概念，通常来说，对于同一份训练集，使用基于单个算法而实现的个体学习器来说，容易产生过拟合的现象。然后，将多个个体学习器组合成一个群体，由此学习器群体中的每一个成员都来对同一份数据进行训练，再收集所有的训练结果，作出最终判断，这样的学习方式成为集成学习，随机森林正是通过构建了一个随机的决策树群体才被称之为集成学习。

集成算法中主要有两大方法，Bagging和Boosting。

Bagging (Bootstrap AGgregation) 是并行式集成算法中最著名的一个算法，其样本采样的主要核心思路是：有放回的从训练集中取出M个样本形成样本集T，未出现在T中的数据用作验证集V，对于集成学习器中的每一个个体，都采取同样的操作，这样可以保证每一个个体学习器拿到的T和V都不相同且T中可能存在重复数据，T和V的比例约为4:1。此类算法最具代表的是Random Forest。

Boosting是指一个能够直接把一个弱学习器数据转换为一个强学习器数据的算法,其数据采样算法的一个主要的核心思想表现为:首先要使用一个初始的数据集来训练出一个基学习器,然后再使用基学习器去预测，对于每一次预测，记录结果的正确性，并赋予初始权重，对于预测正确的样本，降低其权重，反之升高权重，使预测错误的样本在一次次的迭代中受到更多的关注。如此重复地训练，直到基学习器的个数达到预定值Y,这样得到Y个权重不同的基学习器，后续的分类Y个基学习器的加权结果判断。此类算法最具代表的是Ada Boost。

随机森林（Random Forest）是一种由决策树构成的基于Bagging的并行的集成算法，如图2-5所示，通过多个弱分类器的组合，最终采取投票或者取均值的方式，使得模型的泛化能力得到大幅度地提升。

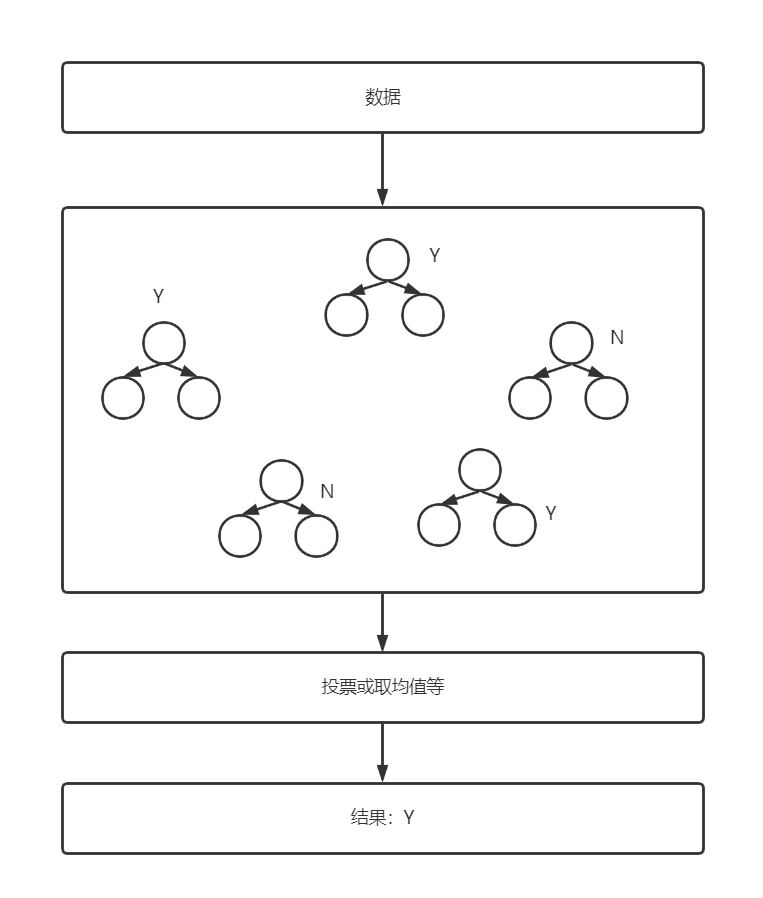


图2-5 随机森林模型概念简图

# 3 数据获取与处理

本章将会从介绍本课题的数据获取和处理的详细流程，从获取原始数据到数据的解析、标签的定义、文本的可读性分析、LDA主题分析进行详细的介绍，并将重点介绍LDA主题模型在本课题中的应用方式和调参过程。结合一定的图表，可视化的分析相关数据。

## 3.1 获取原始数据

本课题通过GitHub Restful Api(<https://docs.github.com/cn/rest>)获取训练需要的原始数据。通过[https://docs.github.com/en/rest/reference/search#search-labels](https://docs.github.com/en/rest/reference/search" \l "search-labels) 来获取good first issue标签。

GitHub官方对OpenApi的调用存在频率限制和数量限制，即每一个用户身份一分钟内只能请求至多30次以及每次数据最多返回1000条。针对与频率限制，本课题通过构建user personal access token池，来提升爬虫的效率和可使用性；针对数量限制，本课题将标签的搜索条件细化到issue创建的每一天，尽可能搜集更多的GFI。

具体的搜集信息如下：

表3-1 原始数据收集字段表

|  |  |
| --- | --- |
| 字段 | 描述 |
| issueId | issue的唯一Id |
| issueRepo | issue 对应的项目 |
| issueTitle | issue 的标题 |
| issueState | issue的状态 open / closed |
| issueLinkedPr | issue是否有链接的pr（Pull Request）信息 0:false 1:true |
| issueLinkedPrInfo | issue 链接的pr具体信息 |
| issueApiUrl | issue具体信息的api链接 |
| issueHtmlUrl | issue github url链接 |
| issueCommentsUrl | issue 评论详情api 链接 |
| issueCreated | issue创建时间 |
| issueUpdated | issue更新时间 |
| issueTitle | issue的标题 |
| issueBody | issue的内容 |
| collectedTime | 收集日期 |

通过程序编程，本课题共收集到459390条GFI,覆盖132382个项目，其中，issue状态为open和closed的数目分别是160650（34.97%）和298740（65.03%）（如图3-1所示），有PR信息的有20693条（4.50%）。

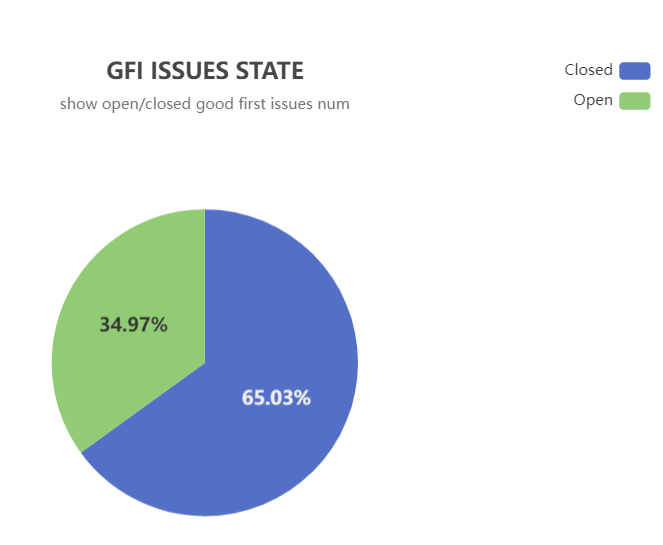


图3-1 GFI open/closed占比图

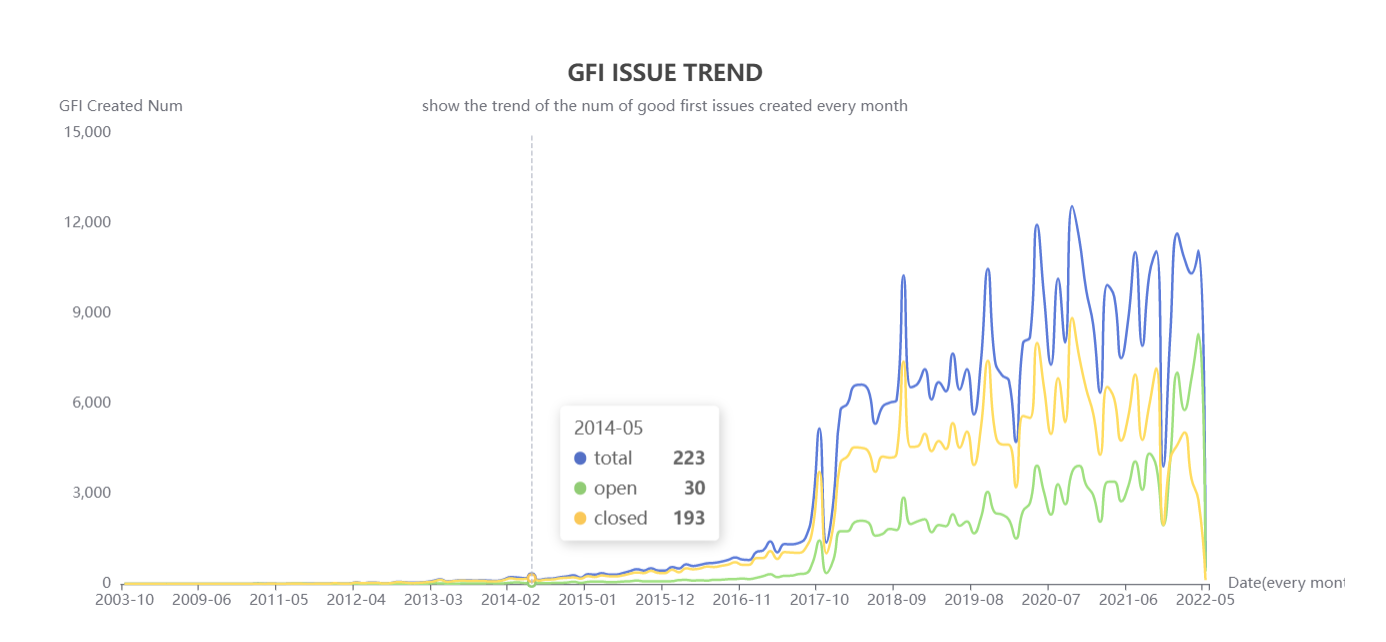
如图3-2所示，通过对收集到的数据进行绘制，可以发现，自2017年10月以来，GFI的创建数目就开始急剧增加，open状态的GFI（图中绿线部分）愈来愈多，这也在一定程度上验证了本课题的必要性。

图3-2 GFI创建数目趋势图

## 3.2 数据预处理

通过收集到的原始数据，我们尝试从两个大方向细化特征指标。

1. 静态指标

a. issue本身

issue自身的信息是issue是否适合新手解决的关键，主要从这几个维度去细化指标：标题长度、内容长度、评论数、评论总长度、参与人数、受让人数。

b. 项目

项目的star数、open issue数、是否是组织下等维度也会间接影响分类效果。

c. issue提出者

提问题的人对问题的影响不可忽略，大致从问题提出者的入圈时间和追随人数来间接表示问题的价值。

1. 语义指标

a. 文本可读性

探究issueTitle和issueBody的文本的可读性对分类的影响

b. 主题

通过对issueTitle和issueBody进行主题分析，尝试探究issue的title和body的主题和主题明确度对分类效果的影响

具体的训练集信息如下：

表3-1 训练集字段表

|  |  |
| --- | --- |
| 字段 | 描述 |
| issueId | issue的唯一Id |
| titleLength | 标题长度 |
| bodyLength | 内容长度 |
| commentsNum | 评论数 |
| commentsTotalLength | 评论总长度 |
| participantsNum | 参与人数 |
| assigneesNum | 受让人数 |
| creatorCreated | 入圈时间 |
| creatorFollowers | 追随人数 |
| starNum | 项目的star数 |
| openIssuesNum | 项目的open issue数 |
| hasOrganization | 是否是组织下的项目 0:false 1:true |
| titleReadability | 标题的可读性 |
|  | 续表 3-1 |
| 字段 | 描述 |
| bodyReadability | 内容的可读性 |
| titleTopic | 标题的主题 |
| titleTopicProbability | 标题是该主题的可能性，可认为是标题主题明确度 |
| bodyTopic | 内容的主题 |
| bodyTopicProbability | 内容是该主题的可能性，可认为是内容主题明确度 |
| isGoodForFreshman | 标签 1:适合新手 0:不适合新手 |
| createdAt | 记录创建时间 |
| updateAt | 记录更新时间 |

对于静态指标，通过对原始数据中的issueApiUrl和issueCommentsUrl解析即可获取到，这里不在阐述实现过程。对于语义化特征层面，本课题使用python进行文本的可读性分析和主题分析。

### 3.2.1 文本预处理

本课题使用python nltk包对issueTitle和issueBody进行文本预处理。

对于可读性分析的文本，我们只需要进行小写化、去除特殊标点、去除换行转义等操作即可。对于主题分析的文本，我们还要增加分词后去除停用词、词干化的操作。

文本预处理的核心程序为：

def text\_precessing(text):

1. 使用string的lower方法将文本中非小写部分小写化
2. 遍历string包的punctuation数组（特殊标点字符数组），如果text中出现这个字符，将text中的此字符全局替换为空
3. 由于GitHub网站的特殊格式导致文本中存在大量换行符,对\r \n这两个出现频率较高的转义进行替换为空
4. 使用nltk的分词包对文本进行分词操作
5. 使用nltk的停用词处理包，去除文本中的英文停用词
6. 对文本进行词干化操作

return {

# 基本处理

'basic':经过小写化、特殊字符处理、换行转义字符处理后的文本 ,用于可读性分析

'filtered': 上述所有过程处理后的结果，用于主题分析

}

对于可读性分析，本课题使用了python的textstat包进行处理。使用flesch\_reading\_ease api进行计算。

可读性分析的核心代码为：

readability = textstat.flesch\_reading\_ease(text\_precessing(text)['basic'])

### 3.2.2 定义标签

定义标签是机器学习中一个繁琐但是很重要的工作。本课题随机选取了10000条含有pr信息的数据（其中有效期内9979条），通过解析每一条issue的提交记录，判断提交记录中的成员中是否含有“新手”。对于“新手”的定义，本课题认为截止到该issue创建时，该成员在此issue对应的项目中的提交次数不超过2次即为新手。

通过采用此隐式标签的方式，编写程序计算，最终我们得到了正样本（分类问题中的适合新手解决）1247条（12.5%），负样本8732条（87.5%）。

## 3.3 LDA主题分析

主题分析是特征语义化层面经常考虑的一个因素，本课题通过分析issueTitle和issueBody的主题，然后将其主题和主题明确度作为分类模型训练的特征。

### 3.3.1 模型准备

我们对issueTitle和issueBody进行文本处理后，使用gensim中的corpora.Dictionary方法构建词袋模型的字典，进行词频统计（如表3-2、表3-3所示）后条绘制数据云图，如图3-3、图3-4所示。

表3-2 issueTitle词频统计Top10频数

|  |  |
| --- | --- |
| 词 | 频数 |
| add | 1099 |
| updat | 1084 |
| fix | 811 |
| ad | 702 |
| readm | 583 |
| creat | 524 |
| md | 482 |
| test | 386 |
|  | 续表3-2 |
| 词 | 频数 |
| file | 360 |
| featur | 319 |

表3-3 issueBody词频统计Top10频数

|  |  |
| --- | --- |
| 词 | 频数 |
| http | 1562 |
| com | 1509 |
| fix | 1302 |
| chang | 1302 |
| test | 1236 |
| issu | 1124 |
| add | 1122 |
| ad | 1068 |
| github | 1014 |
| pr | 993 |

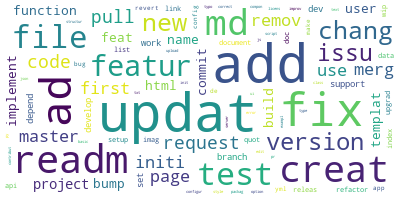


图3-3 issueTitle词频统计云图

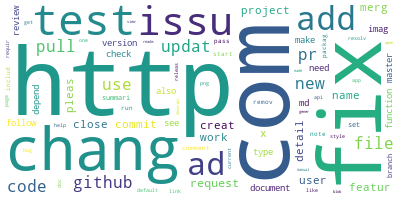


图3-4 issueBody词频统计云图

可以发现issueTitle中经常出现的词为add、update、fix、test、feature等与git commit message规范有关的词语，issueBody中的主流词语与issueTitle有一定重复，除此之外，内容更偏向于http、com等包含的网络链接。

结合词袋模型，我们将文本进行向量化处理：

corpus = [dictionary.doc2bow(text) for text in issues]

在处理好issueTitle和issueBody数据以后，本课题使用python中的Gensim包构建主题模型。我们将文本向量化以后的结果作为LDA主题模型的输入，通过训练来获取最优主题。

### 3.3.2 参数优化

本课题针对主题模型的主题数K={10,20,30,40,50}和模型迭代次数N={500,1000,1500,2000}进行调参，并使用Gensim包中的CoherenceModel来计算模型优劣得分，并结PyLDAvis绘制图形后观察来综合衡量主题模型的效果。

CoherenceModel是gensim针对模型设计到一套评分模型，相比于主题模型的困惑度指标来说，结果更为可靠**[12]**，本课题使用coherence='u\_mass'作为指标，u\_mass指标越大，表示模型的效果越好。

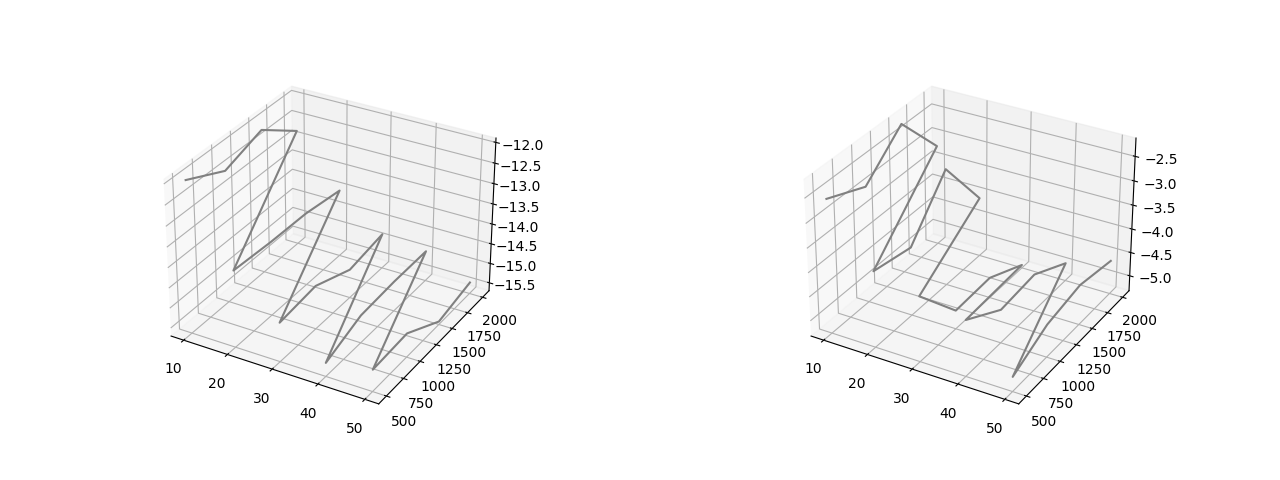
实验的结果如表3-4、表3-5、图3-5所示：

表3-4 issueTitle CoherenceModel模型K-N试验下的得分

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| K  N | 10 | 20 | 30 | 40 | 50 |
| 500 | -11.97 | -13.80 | -14.74 | -15.36 | -15.17 |
| 1000 | -12.44 | -13.87 | -14.62 | -15.02 | -15.11 |
| 1500 | -12.11 | -13.88 | -14.99 | -14.95 | -15.63 |
| 2000 | -12.79 | -14.01 | -14.82 | -14.94 | -15.42 |

表3-5 issueBody CoherenceModel模型K-N试验下的得分

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| K  N | 10 | 20 | 30 | 40 | 50 |
| 500 | -2.57 | -3.75 | -3.98 | -4.16 | -5.02 |
| 1000 | -2.91 | -3.90 | -4.94 | -4.63 | -4.64 |
| 1500 | -2.20 | -2.88 | -4.90 | -4.55 | -4.48 |
| 2000 | -3.20 | -4.07 | -5.26 | -4.96 | -4.62 |

图3-5 issueTitle（左）和 issueBody（右）CoherenceModel模型K-N试验下的得分可视化图

通过数据可以大致发现，issueTitle和issueBody的LDA主题模型在K越小时候表现的整体效果越好，在N=1500时，表现效果也比其他迭代次数稍优一点。

以issueBody的主题模型为例：

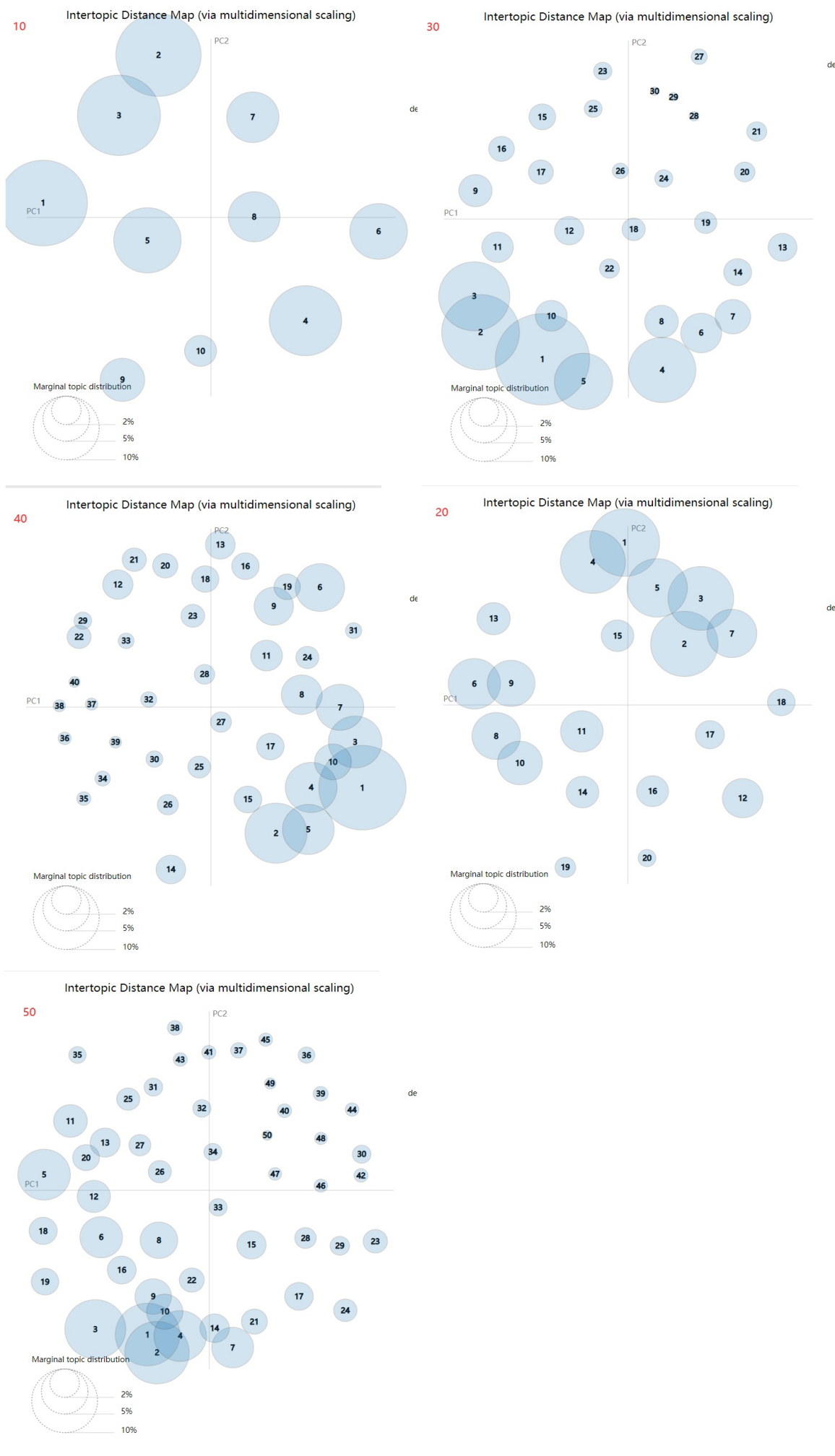


图3-6 N=1500时，issueBody各个主题数下LDA模型效果图

如图3-6所示，通过观察其它共20组模型效果图可以发现，在主题数目较多时，主题的重合数目（重合面积）更多，根据图中的每一个主题的占比和主题之间的距离等因素，我们最终确定了issueTitle和issueBody的LDA主题模型的K/N均为 10/1500时，主题之间的距离越大，每个主题涵盖的数据越多，效果最好。

### 3.3.3 结果及预测

我们利用最终确定下来的K = 10和N = 1500的LDA主题模型，预测了训练集中的9979条数据。

本课题利用主题模型来预测数据集中每一条数据在每一个主题下的概率，并取最高概率的主题作为这条数据的主题，取其对应的概率作为此主题的主题明确度。预测结果如表3-6、表3-7、图3-7所示：

表3-6 数据集中issueTitle字段主题分类结果

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 主题 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 数目 | 9847 | 18 | 41 | 18 | 10 | 38 | 3 | 2 | 2 | 0 |

表3-7 数据集中issueBody字段主题分类结果

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 主题 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 数目 | 7471 | 1510 | 547 | 83 | 115 | 96 | 55 | 76 | 20 | 6 |



图3-7 issueTitle（左）和issueBody（右）LDA主题模型分类后占比

可以发现，在issueTitle的主题模型下，约98%的数据最有可能的主题都是主题1，在issueBody的主题模型下，主题1，2，3的比例更大些，分别为74.87%，15.13%和5.48%。

# 4 基于Random Forest的分类模型实现GFI的分类

本章将介绍基于Random Forest来构建分类模型实现GFI分类的详细过程，从特征选择到验证曲线再到模型的训练，并简单介绍模型的分类与落地的过程和成果。

## 4.1 模型的构建

在对随机森林模型、GFI等知识概念有了一定的了解并对数据进行了处理和分析以后，我们使用sklearn框架开始着手基于Random Forest分类模型的构建。

### 4.1.1 特征选择

特征选择可以在一定程度上避免数据的冗杂，本课题基于特征选择的过滤法中的互信息法，对离散数据的特征与标签之间进行相关性分析，从数学层面移除不相关特征，从而提升分类模型的准确率。

本课题使用python sklearn中的mutual\_info\_classif，计算出每一个特征与标签之间的相关性得分如表4-1、图4-1所示：

表4-1 特征选择互信息法得分结果

|  |  |
| --- | --- |
| 特征 | 互信息法相关性得分 |
| titleLength | 0.0 |
| bodyLength | 0.001604684240608334 |
| commentsNum | 0.0 |
| commentsTotalLength | 0.004364934319135205 |
| participantsNum | 0.0033744825278620993 |
| assigneesNum | 0.0 |
| creatorCreated | 0.09367223124764767 |
| creatorFollowers | 0.02092528114480774 |
| starNum | 0.019155245802792376 |
| openIssuesNum | 0.01948706834514513 |
| hasOrganization | 0.00022137211691952707 |
| titleReadability | 0.0 |
| bodyReadability | 0.009399307765926901 |
| bodyTopic | 0.00555100485939852 |
| titleTopicProbability | 0.01165483776439391 |

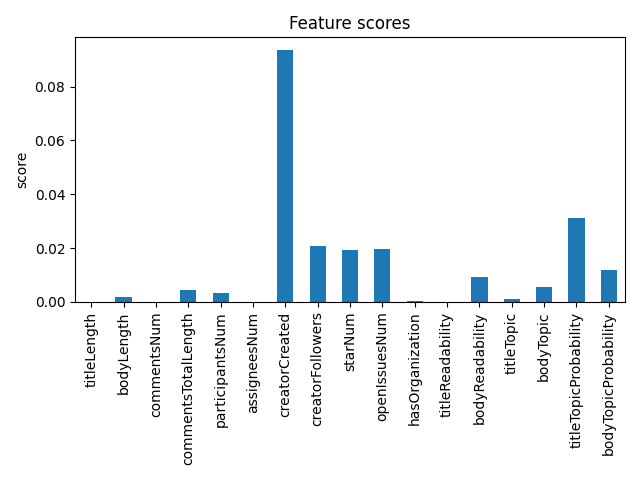


图4-1 特征选择互信息法得分结果可视化图

我们发现，标题长度、评论数、受让人数和标题可读性在预测中与标签的相关性为0，而issue创建人的入圈时间，以及仓库相关特征和主题相关特征的与标签的相关性较大，在预测中更为重要。于是，我们去除了相关性为0的四种特征，最终保留了13个特征用于后续的分类模型的训练。

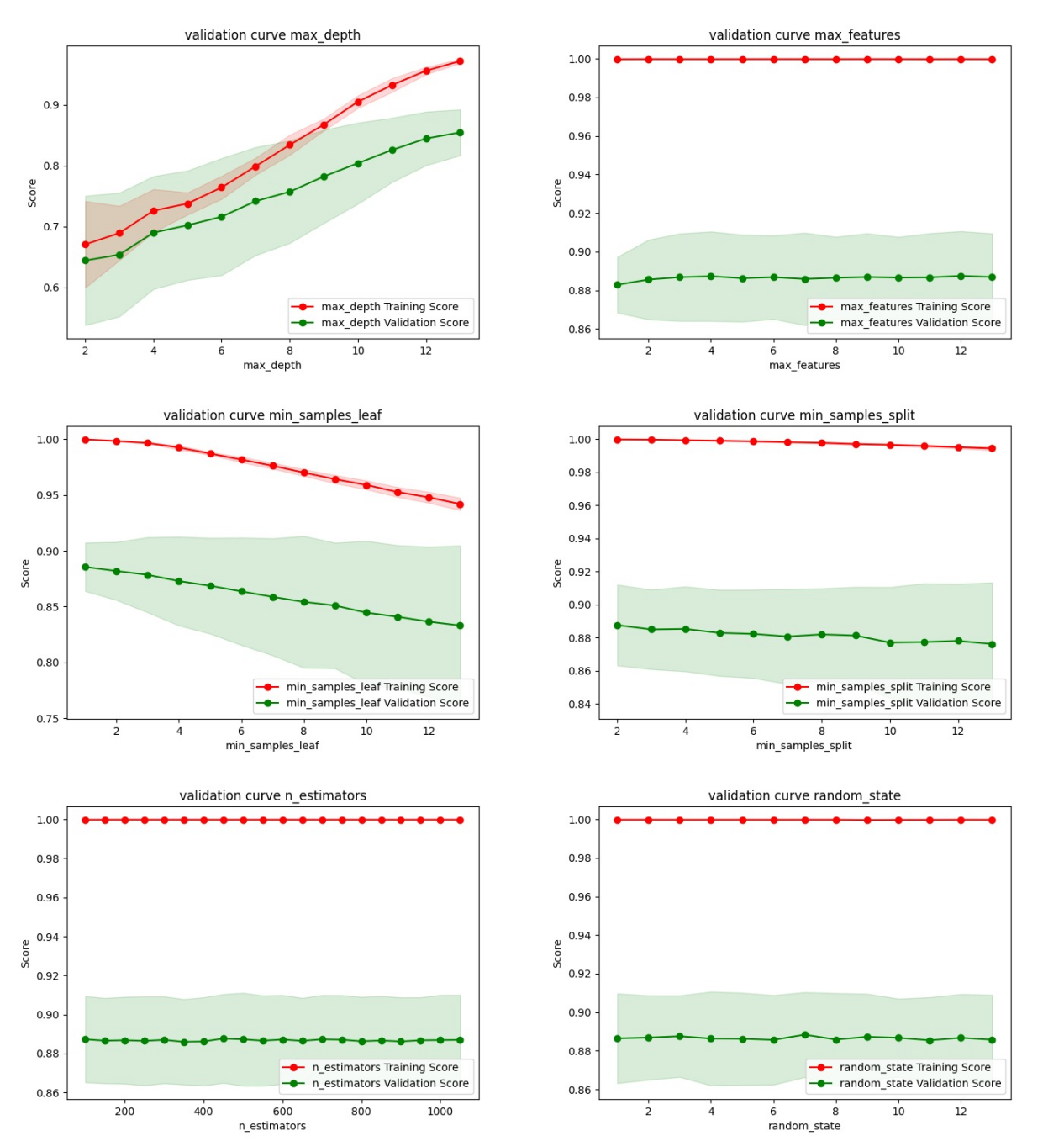
### 4.1.2 参数优化

我们针对随机森林模型的以下几个参数进行调节。

表4-2 随机森林模型拟定调参数据及说明

|  |  |
| --- | --- |
| 参数 | 说明 |
| n\_estimators | 随机森林集成学习中最大的弱学习器的个数 |
| max\_depth | 弱学习器中树的最大深度 |
| min\_samples\_split | 弱学习器中每个节点可分的最小样本数目 |
| min\_samples\_leaf | 弱学习器中树的叶子节点的最小样本数目 |
| max\_features | 构建模型时候选择最优模型时的最大特征数目 |
| random\_state | 随机种子 |

本课题使用验证曲线的方式来帮助模型确定调参范围，验证曲线的使用可以在一定程度上为模型调参作参考，大大地减少了模型训练需要的时间。通过观察和绘制模型的每一个待训练参数的验证曲线，选取曲线上比较效果比较好的点作为后续调参的范围。

图4-2 各参数下验证曲线图

如图4-2所示，我们使用sklearn中的model\_selection包下的validation\_curve方法，采用了10组交叉验证，得出了如图的结果。以图中的random\_state为例，我们通过观察数据在学习曲线和验证曲线之前的距离、以及绘制验证曲线的数据变化弧度来判断，random\_state参数在4、7两个位置的变化弧度较大，学习曲线和验证曲线之间的弧度较小。

最终结合sklearn 中为 Random Forest模型提供的初始参数，例如max\_depth提供了None不限制每一个弱学习器的树的深度、max\_features除了提供float或int的类型外，还提供了“auto”、“sqrt”、“log2”这三种方式等，最终确定了每个参数的调参范围，如表4-3所示。

表4-3 最终的确定调参数据范围

|  |  |
| --- | --- |
| 参数 | 调参范围 |
| n\_estimators | [220,550] |
| max\_depth | [None,5,8] |
| min\_samples\_split | [2,6] |
| max\_features | ['auto','sqrt','log2',5,7] |
| random\_state | [4,7] |

### 4.1.3 模型构建与结果

关于训练集，本课题用sklearn提供的train\_test\_split方法将样本随机分为7:3,70%用作训练集，30%用作测试集，并使用random\_state属性保证每一次划分出来的数据集一致，便于后续不同模型的训练对比。

关于样本不平衡问题，本课题收集的样本中，正样本约12%，这属于样本严重不平衡，相比于使用SMOTE等算法进行过抽样或欠抽样处理，我们使用sklearn随机森林模型为我们提供的class\_weight字段，使用class\_weight=’balanced’来处理。

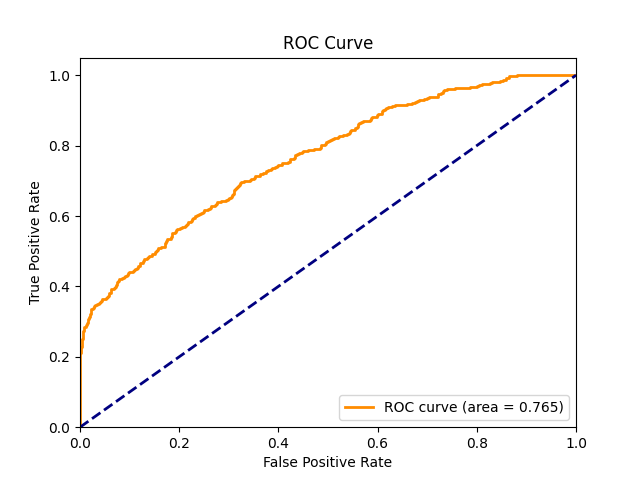
关于过拟合问题，本课题使用cv = 10来进行10组交叉验证**[13]**。

关于准确率评价指标，本课题采用的是roc\_auc指标。准确率评价指标大约有准确率（accuracy）、精确率（precison）、召回率（recall）、F1-score、ROC曲线、AUC面积（ROC曲线下的面积）等。由于ROC、AUC在分类问题中可以解决样本不平衡问题并且不依赖阈值进行评价，相比于accuracy来说，具有跟高的可信度，所以本课题采用框架提供的指标评价方法中的roc\_auc作为指标评价标准。

基于上述的调参范围和处理，结合GridSearchCV（网格搜索），对模型进行训练，最终得到以下结果：

表4-4 模型训练结果信息

|  |  |
| --- | --- |
| 信息 | 说明 |
| 训练耗时 | 524.27s |
| 训练得分(正确率指标为roc\_auc) | 训练集得分：0.7970010253462754  测试集得分：0.7628851208523992 |
| 模型最优参数 | n\_estimators: 550  max\_depth: None  min\_samples\_split: 6  min\_samples\_leaf: 2  max\_features: 5  random\_state: 4 |

模型的ROC曲线如图4-3所示，ROC曲线的曲率越大、AUC的面接越大，模型的效果越好。

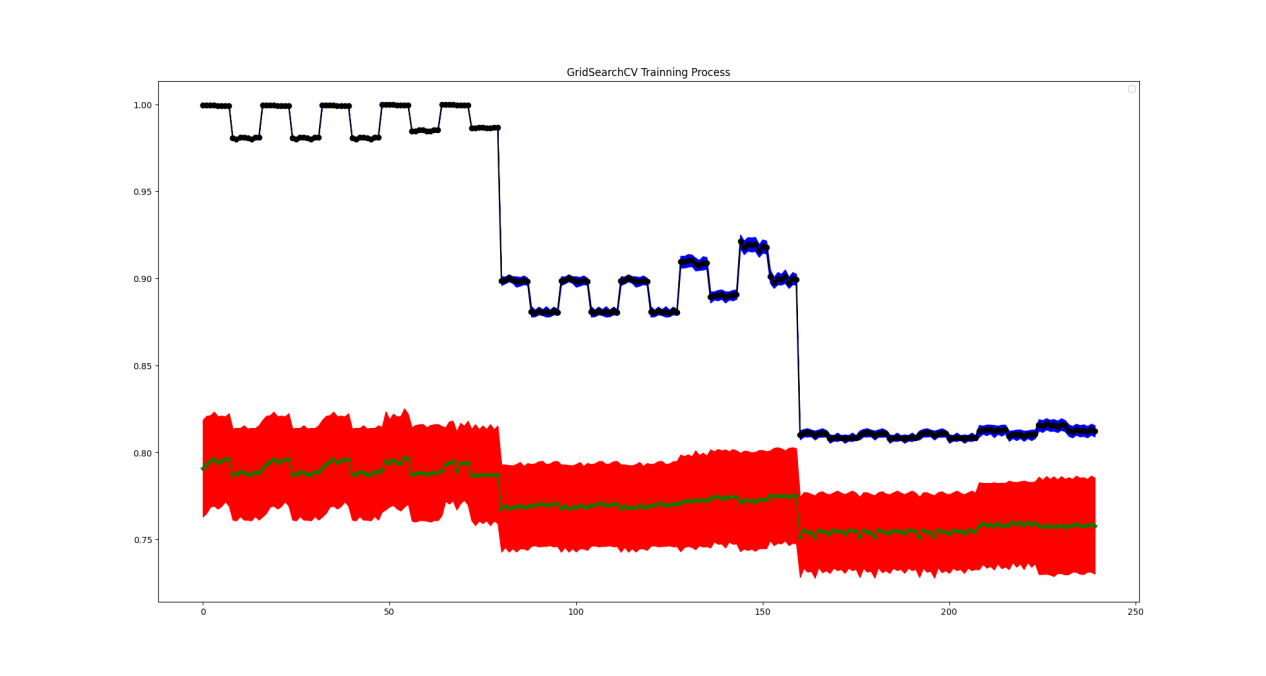
图4-3 最优模型训练ROC曲线

图4-4 训练过程可视化图

如图4-4所示，通过训练过程中的平均训练集得分、平均测试集得分、训练集得分、测试集得分，针对GridSearchCV训练过程进行可视化，可以发现，训练阶段大致分为三层。第一层时，训练集分类效果95%~100%,而测试集分类效果基本在80%左右，这属于严重的过拟合现象；第二层训练集的分类效果降低到90%附近，测试集效果在78%左右；第三层时，训练集分类效果在80%附近，测试集效果在75%左右。通过多次交叉验证，随机森林模型大幅降低了过拟合现象。

针对训练出的最优分类模型，利用feature\_importances\_属性可以查看训练过程中比较重要的特征为issue提出者的入圈时间(creatorCreated)，相关性不大的特征为标题的主题（titleTopic),如表4-5、图4-5所示。

表4-5 训练特征重要性

|  |  |
| --- | --- |
| 特征 | 训练特征的重要性 |
| bodyLength | 0.07115018916864126 |
| commentsTotalLength | 0.05194449525178909 |
| participantsNum | 0.025033809685471097 |
| creatorCreated | 0.29146158544052614 |
| creatorFollowers | 0.09871436995504271 |
| starNum | 0.07687581276259096 |
| openIssuesNum | 0.08595335820357905 |
| hasOrganization | 0.016386382438895602 |
| bodyReadability | 0.06898727214188889 |
| titleTopic | 0.0019585823003357363 |
| bodyTopic | 0.012260552831615076 |
| titleTopicProbability | 0.12515830049119236 |
| titleTopicProbability | 0.07411528932843209 |

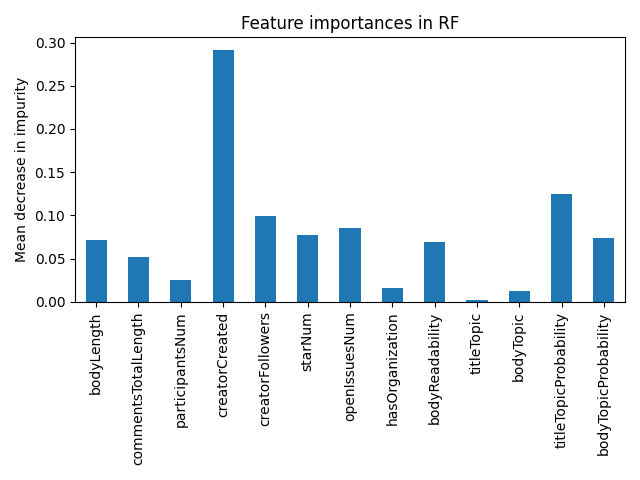


图 4-5 训练特征重要性可视化图

## 4.2 模型比较

在上述特征选择之后，本课题得到了一份特征数为13的训练集，基于该训练集，我们采用了验证曲线的方式对调参范围进行了确定，并利用随机森林模型进行训练，最终在测试集上得到了76.29%的roc\_auc得分。

采用相同的流程，本课题用验证曲线-网格搜索的方式，同样采用roc\_auc的准确率指标，对决策树、SVM和逻辑回归模型也进行了训练，并得出表4-6所示的各个模型的训练信息。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 模型 | 训练耗时 | 训练集得分 | 测试集得分 | 模型最优参数 |
| 决策树 | 3.06s | 0.7228207642186448 | 0.7013207747065084 | max\_depth:8  max\_features:13  min\_samples\_leaf:9  min\_samples\_split:4  random\_state:10 |
| SVM | 117.70s | 0.7114846409134741 | 0.6923649333810925 | C:2  gamma:scale  random\_state:2 |
| 逻辑回归 | 11.28s | 0.7114836987666384 | 0.6923901582608958 | C:1  max\_iter:100  multi\_class:auto  penalty:l2  random\_state:2 |

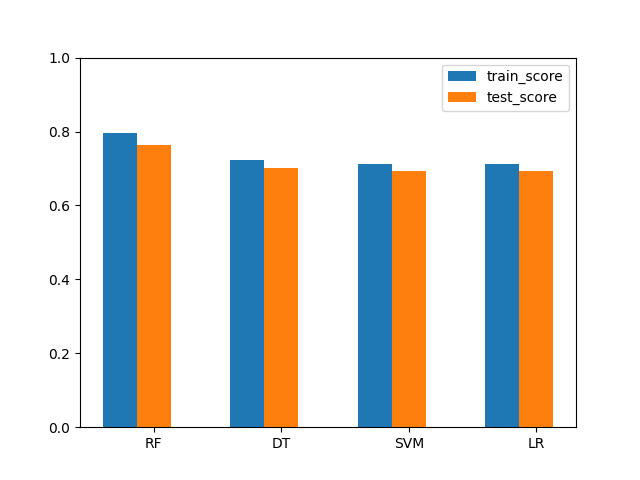


图4-6 各模型下训练集/测试集roc\_auc指标对比图

如图4-6所示，随机森林在分类训练中表现的最好，决策树、SVM、逻辑回归分类在训练集和测试集上的roc\_auc得分基本相同，决策树具有一定的优势。

本课题最终采用效果最好的随机森林来进行后续的模型分类工作和模型落地工作。

## 4.3 模型分类

本课题从收集到的160650条状态为open的issue中选取了22273条按照解析基本数据、计算文本可读性、主题分析的步骤处理数据，并使用最优分类模型进行分类，其中，有939条(4%)数据被分类到“适合新手解决”。

## 4.4 模型落地

模型落地是模型面向用户的一个重要步骤，优秀好用的应用载体可以将模型的使用价值发挥到更好。

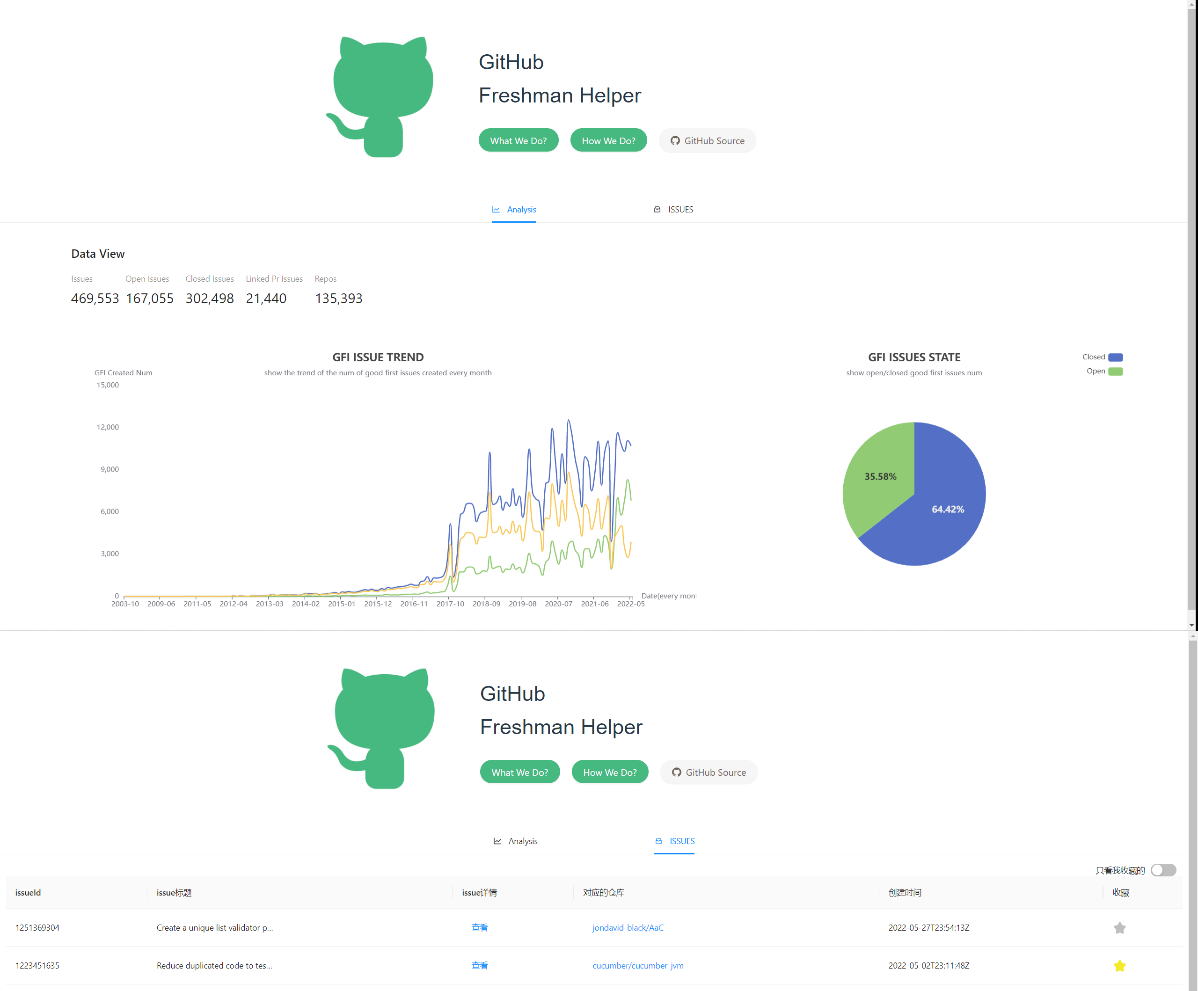
本课题基于此随机森林分类的模型，采取前端-后端-模型一体化的方式构建模型落地系统，使用前端技术：React+AntDesign、后端技术：Nestjs、模型技术：Sklearn+Random Forest,实现主要包括以下前端用户界面、后台管理界面：

图4-7 前端用户界面

如图4-7所示，前端用户界面主要有两大功能，一：对收集到的数据作趋势展示和open/closed issue占比展示给用户；二：将模型分类出的939条适合新手解决的GFI作展示，并结合用户体验，提供收藏的功能，用户可以通过此平台适合自己解决的GFI，更好地融入GitHub的圈子中。

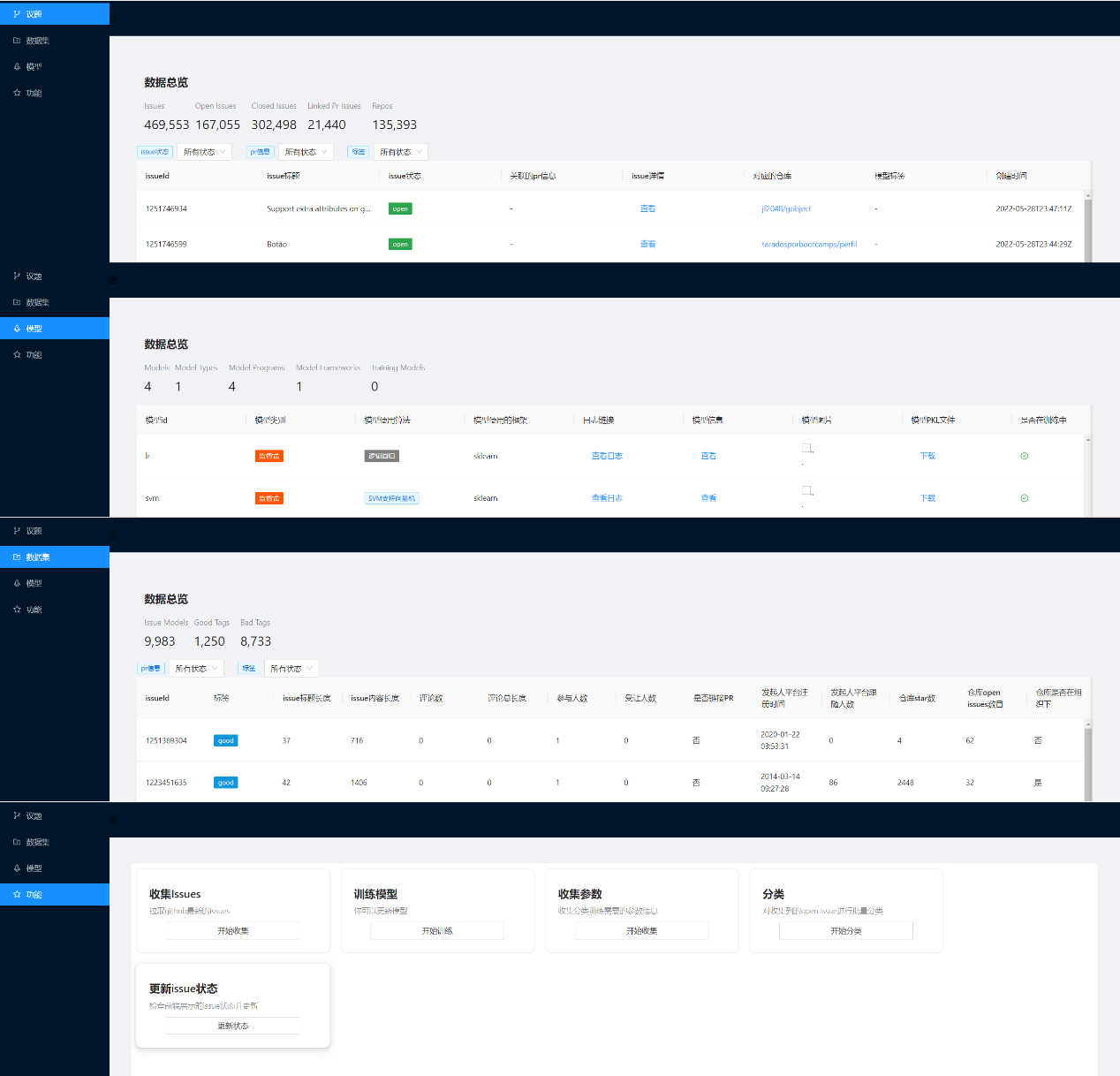
后台管理界面主要对收集到的issue、数据集和训练的模型进行分页展示，并提供收集issue、训练模型、收集参数、分类、更新issue状态等功能提供可视化的操作模型的训练、模型的分类的途径。如图4-8所示。

图4-8 后台管理界面

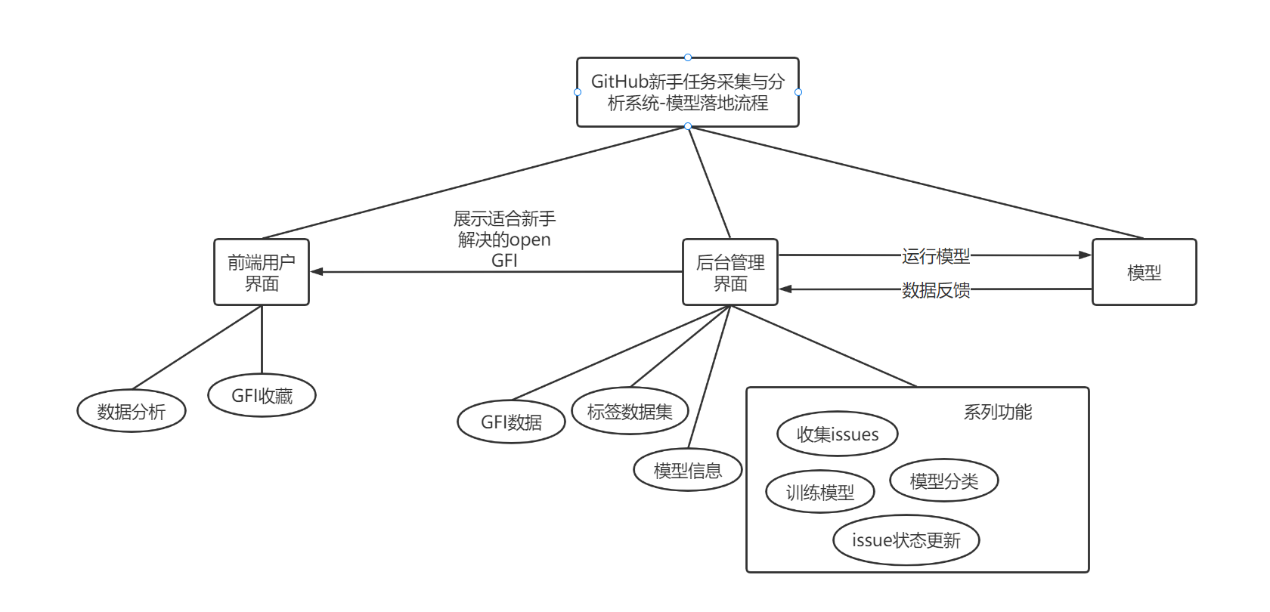
模型落地系统设计图如图4-9所示：

图4-9 模型落地系统设计图

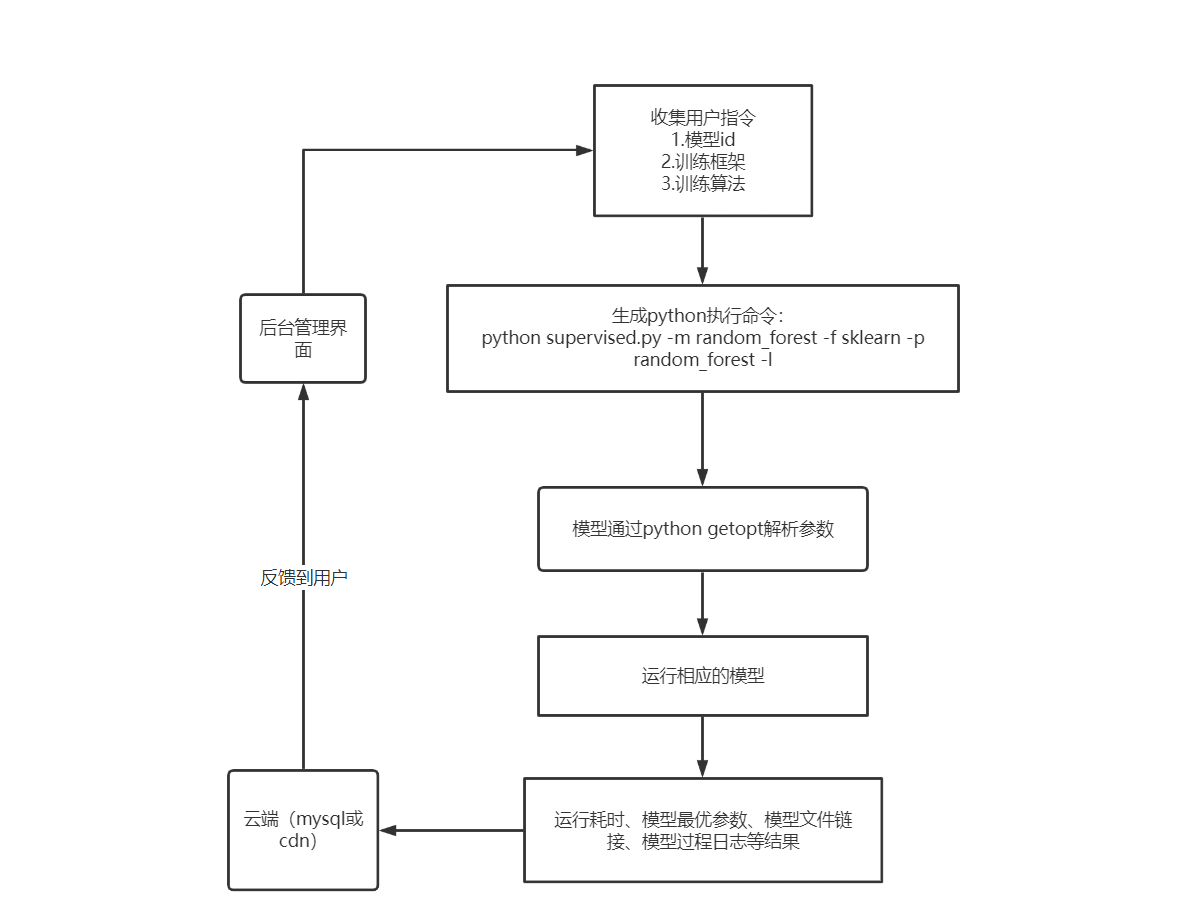


图4-10 模型落地过程中模型训练功能的设计闭环图

以模型训练功能为例介绍，如图4-10所示，系统的核心是实现是由model层到server层的关联，本课题通过python包getopt来对python程序代码进行模块化分装，由server层进行参数传递：

1. 训练随机森林模型

python supervised.py -m random\_forest -f sklearn -p random\_forest -l

1. 训练SVM模型

python supervised.py -m svm -f sklearn -p svm -l

命令行参数m、f、p、l 分别为模型id、模型使用框架、模型使用算法、模型运行环境，通过server层从用户层面获取到传到model层，再由model层内部程序对这些参数进行消化，运行相应部分的逻辑，从内部将运行结果存到云端（mysql或者是cdn空间）。

模型落地很重要，但并不是很课题的研究重点，关于系统层面的设计和代码具体实现，可以在<https://github.com/mengxun1437/github_freshman_helper>进行查看和跟进，本论文将不作进一步的描述和介绍。

# 5 总结与展望

## 研究总结

本课题通过结合LDA主题模型和随机森林分类模型，对目前全球最大的代码资源托管平台GitHub中包含GFI标签的数据进行了收集、分类和分析。通过特征选择、验证曲线等多种方式进行参数调优，对GFI进行分类后构建相应的模型落地系统，意在帮助新手小白更好地融入GitHub的技术圈子中。本研究主要完成的内容如下：

（1）了解GitHub的相关知识和GitHub GFI的相关理念，通过GitHub Restful Api，采取相应的数据爬取手段，收集了459390条原始数据，并对数据进行了趋势分析和占比分析，验证了本课题研究的必要性。

（2）随机选取了9979条有效数据，通过隐式标签的方式，对数据进行了标签化，得到正样本1247条（12.5%），负样本8732条（87.5%），利用这9979条数据作为后续的数据集。

（3）从静态指标和语义指标两个层面共17个特征解析原始数据。从GitHub Api层面，通过网络请求的形式得到11个静态特征指标；对GFI标题和内容文本预处理进行可读性分析，文本向量化后进行主题分析，得到语义层面的6个特征指标。

（4）通过特征选择、验证曲线的方式优化模型。通过特征选择，利用互信息法，从数学层面去除特征与标签相关性不大的4个特征指标，大大提升了后续分类模型的分类性能，减少了模型的训练耗时；通过验证曲线的方式，得到模型训练中的调参点的范围，大幅度地减少了模型训练的所需时间，为模型的工程应用型层面降低了成本。

（5）通过网络搜索的方式，基于sklearn框架，使用随机森林模型，对样本进行训练集划分和权重平衡，对参数进行10组交叉验证，最终得到最优分类模型。利用框架提供的特性，分析与模型训练相关性不大的特征。

（6）采用同种过程，结合决策树、SVM、逻辑回归等模型对同一份数据集进行分析，得到各个模型训练后的效果对比。

（7）使用最优模型随机森林对采取到的另外22273条数据进行分类，得到939条适合新手解决的GFI，并构建web-server-model一体化的用户系统，将模型成果落地。

本课题主要的创新点如下：

（1）目前研究的学者几乎是基于GitHub Issue来进行研究，本文着手GFI的研究，可以在含有good first issue的标签中分类出真正适合新手解决的issue，分类对象更具体，使得结果更可靠。

（2）本课题通过主题模型对文本进行建模。目前主题模型的优劣主要评判标准是困惑度，而困惑度的标准在工程领域的实用性较差，而本课题采用CoherenceModel来量化模型效果，并结合PyLDAvis工具绘制可视化效果图对最优模型的选择结果进行了双重的验证。

（3）本课题通过先分析后训练的方式，结合特征选择和验证曲线，相比于传统的直接暴力参数调优来说，大幅度减少模型的训练耗时。

（4）本课题通过各种模型的训练结果对比，最终选择了随机森林模型。并结合当前热度较高的Web形式来进行模型落地，从模型到产品，形成了一套完整的工程应用。

## 5.2 研究展望

回首整个课题研究过程，有欣喜也有遗憾。欣喜的是经过近一个月的代码编写、实验与验证，最终实现了一套可以用于交付和使用的模型和产品。但在整个实验过程结束后，认真思考后，发现总体还是存在一些缺陷和局限：

（1）本课题在数据收集步骤中，没有做好技术调研和特征确定。在技术调研中，没有发现一些基于GitHub Restful Api的爬虫包，这些爬虫包内部解决了频率限制和数量限制的问题，而我们自己采用token池和细化搜索的方式，虽然内部核心与这些爬虫包基本一致，效果也达到了要求，但是在一定程度上，没有使用爬虫包给我们带来了编码时间上的代价；在特征确定上，没有考虑好后续模型训练需要的特征前，进行了初步特征提取，导致后续特征增加后，出现数据爬取返工的现象。

（2）本课题研究使用主题模型进行文本主题分析，但是在主题模型的选择上采取的是比较古老的基于LSA和pLSA的LDA模型，并没有采取其他模型进行训练效果的对比参考，在主题分析步骤中，可能会导致分析结果具有一定的不可靠性。

（3）虽然采用了多种模型的训练效果对比，但是没有对准确率评价指标、交叉验证折数等进行控制变量，没有探究评价指标等对同一模型或不同模型的影响。

（4）由于模型落地不是本课题的重点研究步骤，所以在模型落地的产品设计上，没有过多的考虑，这将导致用户体验性有一定程度上的降低，落地系统的可靠性和安全性也有待提高。

针对这些问题，我们将在后续升级迭代中予以改善和解决。

# 6 致谢

经过近一个月的实验研究和验证，本课题终于得到了不错的研究成果，在这漫长的过程中，我得到了很多的帮助，在此表示由衷的感谢。

感谢我的毕设指导老师——陈洁老师，陈老师在我比较困惑的时候，给我提供了研究的方向，并对我的研究内容进行了引导，是您宝贵的指导意见让我一步步取得了现在的成果。

感谢前人学者们在相关课题领域的研究，站在巨人的肩膀上，我得到了很多的启发，受益匪浅。

感谢一直陪伴的我的同学们和家人们，感谢你们对我的鼓励和支持。

最后，感谢在百忙之中来参与答辩的答辩组各种老师和同学们，你们的指导意见对我提供了很大的帮助。

# 7 参考文献

1. Allaho M Y, Lee W C. Analyzing the social ties and structure of contributors in open source software community[C]//Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. 2013: 56-60.
2. Golzadeh M, Decan A, Legay D, et al. A ground-truth dataset and classification model for detecting bots in GitHub issue and PR comments[J]. Journal of Systems and Software, 2021, 175: 110911.

[3] Kikas R, Dumas M, Pfahl D. Issue dynamics in github projects[C]//International Conference on Product-Focused Software Process Improvement. Springer, Cham, 2015: 295-310.

[4] Salo R, Poranen T, Zhang Z. Requirements management in GitHub with a lean approach[C]//SPLST. 2015: 164-178.

[5] Di Sorbo A, Canfora G, Panichella S. " Won't We Fix this Issue?" Qualitative Characterization and Automated Identification of Wontfix Issues on GitHub[J]. arXiv preprint arXiv:1904.02414, 2019.

1. Zimmermann T, Nagappan N, Guo P J, et al. Characterizing and predicting which bugs get reopened[C]//2012 34th International Conference on Software Engineering (ICSE). IEEE, 2012: 1074-1083.
2. Sajedi Badashian A, Hindle A, Stroulia E. Crowdsourced bug triaging: Leveraging q&a platforms for bug assignment[C]//International Conference on Fundamental Approaches to Software Engineering. Springer, Berlin, Heidelberg, 2016: 231-248.
3. Badashian A S, Hindle A, Stroulia E. Crowdsourced bug triaging[C]//2015 IEEE International Conference on Software Maintenance and Evolution (ICSME). IEEE, 2015: 506-510.
4. Kikas R, Dumas M, Pfahl D. Using dynamic and contextual features to predict issue lifetime in github projects[C]//2016 IEEE/ACM 13th Working Conference on Mining Software Repositories (MSR). IEEE, 2016: 291-302.
5. Huang Y, Wang J, Wang S, et al. Characterizing and Predicting Good First Issues[C]//Proceedings of the 15th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM). 2021: 1-12.
6. Tan X, Zhou M, Sun Z. A first look at good first issues on github[C]//Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. 2020: 398-409.
7. Brassard G, Méthot A A. Can quantum-mechanical description of physical reality be considered in complete?[J]. International Journal of Quantum Information, 2006, 4(01): 45-54.
8. Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection[C]//Ijcai. 1995, 14(2): 1137-1145.