

本文目标：使用机器学习方法研究GitHub中的代码和职责的关系

角色：后端、前端、全栈、移动、运维、数据。

角色划分来源：

The Annual Developer Survey conducted by Stack Overflow in 2018

（问卷问题：Which technical roles they associate with the tasks they normally perform?）

BACKEND (58%), FULLSTACK (48%), FRONTEND (38%), and MOBILE (20%), DEVOPS (10.4%) and DATASCIENCE (7.7%)

由于目标是划分Github，因此采用Stack overflow做基本事实（ground truth）。

截止2020年6月2日，使用SEDE工具收集含Github信息的stack overflow用户，发现了27051个开发者。解析GitHub用户名，使用GraphQL API获取Github数据。删除无效用户名，得到了24889个开发者。

标签从overflow上的介绍中提取。共有1802个开发者标记了自己的角色。去掉140个少于5个项目的用户，还剩1662个开发者，他们标记了2022个角色。

在开发者的标记中，前端最多，移动和后端最少。详细表格（分析）：



调查：

角色是招聘某一特定职位时招聘者首要考察的因素。有64%的工作在他们的描述中提到说需要至少一种角色。

6/7的响应者会去查看被试人的Github用于招聘（该问卷有23%的响应率），具体信息如下：



图中还显示了GitHub的其他用途：辨别开发者正在使用哪些技术开源过程中的行为。

实验数据提取：

编程语言：

实验结果1：

最佳模型：precision=0.75 auc=0.7 recall=0.49 f1=0.59

（后两项很低，但是这个模型中准确度模型显然更为重要，即正确分类了符合需求的应试者，相比于辨认他们的需求）

准确率：数据0.86和前端0.77最好，后端0.62最差。

Note:



（<1，准确率更重要；>1召回率更重要；=1，同样）

实验结果2：

数据的前10相关特性有33.2%的代表性（与其它特征不同）。后端仅6.8%。程序语言与所有角色都相关。

实验结果3：

训练120个分类器，观察改进，recall+0.05，precision-0.05

实验结果4：

全栈的分类器效果很好，precision=0.99，recall=0.71；基于这一分类继续分前后端，有更好的准确率（0.87front & 0.86end）

特征转换

协方差分析

* 477个语言和1000个项目依赖
* pd.corr()>0.7 => 去掉较高cor的260个语言和202个依赖

词袋模型（文本->特征）

* 手工去除<…>、标点、数字
* sklearn.text移除stop words(his, and…)
* sklearnTfidfVectorizer生成特征
* 随机森林：随机选择文档，100个不同的词袋配置
* 去除不在以下频率的特征：[0.04,0.15] for Projects’ Descriptions, [0.03, 0.25] for Projects’ Names, [0.01, 0.25] for Projects’ Topics, and [0.01, 0.20] for Short Bio

**标签值的转换**

注意到标签可以是重合的，即y=(frontend and data)是合法标签。（通常分类问题仅有一个分类）

转换方法：

* BR：对每个属性回答是或否，分成5张表进行分析
	+ 好处：支持独立或合并（所有模型上的加权平均）显示结果
* CC：将y逐行转化为特征放入x中
	+ 好处：解决了BR的标签独立问题



分类器

随机森林

* 对噪声和异常值的健壮性

Naive Bayes

* 对文本信息的处理较好，例如垃圾邮件过滤和新闻分类

分类效果

* 均采用10折交叉验证

结论描述变量

* Precision、Recall、F1、AUC、Jaccard Coefficient（正确预测了所有标签的概率）、Hamming Loss（平均预测错了多少个标签）

结论

RQ1: How accurate are machine learning classifiers in identifying technical roles?

BR模式整体：随机森林表现更好（除了recall都好）



BR模式局部：数据0.86和前端0.77最好，后端0.62最差。



**RQ2:** What are the most relevant features to distinguish technical roles?

对每个类别做一次随机森林，找前10个较好的feature。



DATASCIENCE: *scientist (Bio), Jupyter Notebook (total), Jupyter Notebook (rate), data (Bio), R (total)*, and *R (rate)*.

*devops (Bio)* for DEVOPS

*mobile (Bio)* for MOBILE,

*JavaScript (author)* for FRONTEND

BACKEND: none

**RQ3:** Do technical roles influence each other during classification?

CC模式中有120种选择，用RQ1作为基线0绘图。

小改进：

Recall：+0.05上限 Precision：-0.05上限 F1 & JC：+0.02上限 AUC：-0.01（更差了）

HL：浮动



**RQ4:** How effectively can we identify full-stack developers?

Stack Overflow上提取Full-stack：853/2284 are FULLSTACK (783未完全标记，加标签).

特征扩展：1567 features: 819 from Projects’ Dependencies category, 219 from Programming Languages, 212 from Projects’ Descriptions, 146 from Projects’ Topics, 101 from Projects’ Names, and 70 from Short Bio.

结论：所有指标都更好

可能的原因：

* 补上了标签
* 增加了1个新特征



可能的改进

角色：有限数量的，但是已经是最流行的6个了（覆盖了64%）

基本事实：仅包括了同时在Github和overflow上有信息的人。可能可以增加数据源。

基本事实：仅依赖于overflow描述，可能有误。尝试半自动化标签或半监督学习

多标签：Label PowerSet

分类技术：XGBoost

阈值：pearson>=0.7相对保守

Icml2020

Why bigger is not always better: on finite and infinite neural networks

Laurence Aitchison, University of Bristol, Bristol, UK.

Correspondence to: Laurence Aitchison.