基于机器学习的GitHub企业影响力分析与预测

王明宇[1]，宫庆媛[2]，瞿晶晶[3]，王新[1]
（1. 复旦大学计算机科学技术学院，上海 200438；
2. 复旦大学智能复杂体系基础理论与关键技术实验室，上海 200438；
3. 上海人工智能实验室，上海 201210）

摘　要：企业影响力的高低不仅关系到其行业竞争力，也影响着其社会声誉和未来发展，然而对企业影响力的评价一直没有统一的标准。GitHub是一个代表性的软件开发代码存储库开源平台，现有研究通常使用企业在GitHub发布的项目得到的star总数衡量其影响力高低，但是这种方式难以衡量小微企业和新生企业的潜力。通过引入科学家的影响力衡量指标h指数，以GitHub为信息源进行企业网络建模，同时基于该网络提取特征构建分类器，对企业未来的影响力水平进行预测。在此基础上应用SHAP模型解释技术，判别决定企业影响力的重要特征。实验结果显示，基于XGBoost的模型在GitHub真实数据集上实现了0.92的准确率和0.93的平均AUC，可以准确、可靠地对企业进行影响力预测。

关键词：在线开发者社区；社交网络；机器学习；SHAP
中图分类号：TP39
文献标志码：A
doi: 10.11959/j.issn.2096−6652.202327

Analysis and prediction of GitHub company influence based on machine learning

WANG Mingyu[1], GONG Qingyuan[2], QU Jingjing[3], WANG Xin[1]
1. School of Computer Science, Fudan University, Shanghai 200438, China
2. Research Institute of Intelligent Complex Systems, Fudan University, Shanghai 200438, China
3. Shanghai Artificial Intelligent Laboratory, Shanghai 201210, China

Abstract: The influence of a company is not only related to its industry competitiveness, but also affects its public reputation and future development. However, there has been no unified standard for evaluating the influence of a company. GitHub is a representative open-source platform for software development code repositories. Existing research typically used the total number of stars a company receives for projects posted on GitHub to measure its influence, but this approach is difficult to measure the potential of small, micro, and nascent companies. The paper predicted the future influence level of a company by introducing the scientist's influence measure h-index, using GitHub as the information source, and modeling the company network. Features was extracted features based on this network to build the classifier, which predicted the future influence level of the company. The SHAP model explanation technique was further applied on this basis to identify the important features that determined the influence of a company. The experimental results showed that the XGBoost model achieved an accuracy of 0.92 and an average AUC of 0.93 on the real-world GitHub dataset. In summary, the proposed method could accurately and reliably predict the influence of companies.

Key words: online developer community, social network, machine learning, SHAP

收稿日期：2023−07−01；**修回日期：**2023−08−02
通信作者：宫庆媛，gongqingyuan@fudan.edu.cn
基金项目：国家自然科学基金项目（No.62102094）
Foundation Item: The National Natural Science Foundation of China (No.62102094)

0　引言

GitHub是一个领先的在线开发人员社区，拥有3 100万开发人员和9 600多万个存储库。此外，有大量企业在GitHub拥有官方账号，尤其是科技类企业[1]。他们会将科研成果以项目的形式发布在GitHub账号上，如果开发者对某个项目感兴趣，他们可以通过“star”来推荐它。在很多研究中，学者使用star的数量来代表一个项目[2]的受欢迎程度，即star的数量越大，说明这个项目越受欢迎。对于一家企业来说，其创建的所有项目所获得的star总数越大，说明该企业越受欢迎。但是这种评价方式存在一定的局限性，例如，同样作为知名社交网站，Facebook拥有超过80 000名员工，其star总数高达935 897次，而拥有7 500名员工的Twitter的star数量只有123 067次；同样作为知名互联网公司，创立于2000年的百度的star总数高达64 149次，而创立于2010年的美团的star总数仅有6 404次。显然，使用star总数作为衡量指标，若以员工人数划分，小型企业相比大型企业不占优势；若以创立时间划分，新生企业相比老牌企业不占优势。

本文借鉴科学家的h指数（用于评估科研学者学术水平与学术影响力的混合量化指标，也称h因子），定义企业的h指数：其在GitHub发布的N个项目中有h个项目每个项目至少被star了h次，并且其余N-h个项目每个项目被star的数量均小于或等于h次。根据定义，我们对GitHub上的企业进行h指数计算。表1列举了几家代表性企业的h指数和其他相关指标。从表1中可以看出，h指数受企业规模大小和创立时间因素的影响较小，大大减少了使用star总数评价企业影响力的局限性。例如，Facebook虽然拥有接近100万个star，在全球企业中排名第3，但实际上这些star只集中在不到100个项目中。也就是说其他大量的项目并没有得到很多关注，因此其h指数只有97。而Square虽然只拥有不到2 000名员工，并且只创立了14年，其star总数相对较低，但h指数达到了较高的105，是因为它发展迅速，有很多优质项目得到大量关注。因此，h指数是一个稳健的累积指标，兼顾了企业项目的发布数量和star数量，“质”与“量”并重。单纯的项目数量增加对h指数不产生直接影响，也不容易被某个意外过高或过低star的项目所影响。因此h指数能够科学地对企业的影响力进行综合评价，并进行相应的量化。经过计算，Brand Finance（全球性的独立第三方品牌影响力评估机构）发布的科技类企业排名与GitHub企业的h指数排名之间的Spearman相关系数为0.484，具有一定的相关性。因此h指数可以在一定程度上反映出科技类企业的影响力，尤其是在品牌强度方面的影响力。

表1　GitHub企业信息对比

企业名称	员工数量/名	创立时间	star总数/次	h指数
Tencent	110 715	1998年	448 913	135
Square	1 853	2009年	281 166	105
IBM	282 100	1911年	66 368	103
Apple	154 000	1976年	217 340	101
Facebook	83 553	2004年	935 897	97
JetBrains	1 900	2000年	161 409	97
Uber	29 000	2009年	98 124	82
Twitter	7 500	2006年	123 067	69
Oracle	143 000	1977年	56 345	64

h指数可以衡量企业影响力的高低，而企业影响力的高低反映出其在一定时间内创造的价值，对于企业的社会声誉和未来发展具有重要意义。一般来说，企业影响力是一种无法预判的后验信息，仅能依靠有关特征来确定现有的影响力。为了解决这一问题，我们将企业当前的指标作为依据，对企业未来的影响力进行预测，发掘企业潜力，找到那些暂时处于初创期、发展期的优秀企业，提高其知名度，从而带来关注度和资源倾斜。同时，也可以对处于停滞期和衰退期的企业起到预警作用。

本文使用GitHub提供的REST API收集公开信息，抓取的信息字段包括项目数量、贡献者数量、每个项目拥有的star数量等。在此基础上，使用极限梯度提升树（extreme gradient boosting，XGBoost）作为底层分类器来构建模型，使用SHAP解释框架对特征进行解释，从而对企业未来影响力高低进行可靠预测，并将预测结果与支持向量机（support vector machine, SVM）、多层感知机（multilayer perceptron, MLP）模型进行对比。结果显示，基于XGBoost的模型预测结果显著优于基于支持向量机、多层感知机的模型预测结果。本文的主要贡献总结如下。

（1）使用NetworkX（用于创建、操作和研究复杂网络的结构、动力学和功能的Python包）对486家企业进行网络建模和计算，提取可以反映节

点企业影响力的网络连接特征，而不仅限于从GitHub爬取的企业账户特征。在此基础上，使用3个维度共11个特征基于XGBoost分类算法进行建模。由于加入了网络层面的影响因素，从而可以更准确地预测GitHub企业未来可以达到的影响力水平。

（2）采用目前流行的机器学习模型可解释技术，对构建的企业影响力预测模型进行解释，给出使预测概率上升的重要特征和从整体上影响模型结果的重要特征，以提升模型透明度和可信度，进而构建基于可解释性机器学习（explainable artificial intelligence，XAI）的GitHub企业影响力预测与解释系统。

（3）在从GitHub收集的大规模数据集上评估了所提出的方法。实验结果表明，本文提出的方法在统计学上显著优于其他模型。此外，我们发现项目的数量、贡献者的数量和在企业社交网络中的有效规模取值是决定一个企业是否具有高影响力的重要的3个特征。

1 GitHub企业数据获取

本文的数据来源是GitHub平台，我们通过GitHub的REST API收集企业账号信息，抓取了GitHub上486家企业的相关数据。图1展示了数据抓取的过程。

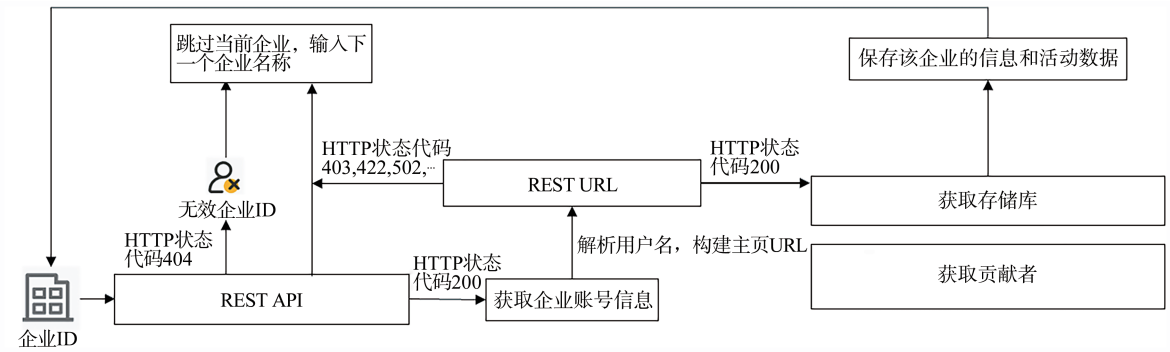


图1 GitHub企业数据抓取过程

GitHub企业群体庞大，且存在企业名称与其GitHub账号名称不一致的问题，因此难以获取全部企业账号列表。以往针对该平台企业账号的研究只能使用一小部分企业对GitHub进行数据驱动研究。本文希望通过扩大整个企业数据集，克服抽样偏差的缺点。Gitstar Ranking是一个GitHub用户和存储库数据实时更新网站，它收集了大量GitHub官方组织账户，并将其按star数量进行排序。本文对该官方组织名单进行了补充，并且排除了开源项目账户，仅使用科技类企业的账户构成数据集。

GitHub为开发者提供了REST API，可以用来创建调用以获取与GitHub集成相关的数据。REST API提供了GitHub的很多常用功能，比如操作、拉取等，但是有些功能需要经过身份验证。由于本文需要对GitHub发出大量请求，而未经过身份验证的客户端每小时只能发出60个请求，因此我们创建个人访问令牌，使得每小时的调用上限达到5 000次，以满足调用需求。

每个GitHub企业账户都有一个唯一账号名。如果知道一个企业的账号名，就可以通过访问URL来查看其对公众开放的资料页面。考虑到GitHub中存在长期没有更新和维护账号数据的企业，本文过滤掉没有更新信息的企业，只选择至少有1个贡献者和1个项目的企业，组成有效企业数据集合。本文实现了一个基于Python的爬虫，这个爬虫应用GitHub提供的REST API来下载每个企业的数据。数据集收集于2022年11月24日至12月15日，历时3周。我们成功抓取了486家科技类企业的账户资料及其在GitHub上的存储库，这些企业累计发布了70 074个项目，共计拥有924 967个贡献者。每个企业的数据包含项目数量、全名、拥有项目的所有者ID、企业所在国家、贡献者数量、开发语言、创建时间等。我们还计算了每两家企业之间的共同贡献者人数，并以此作为企业社交网络建模的基础。因此，本文能够构建具有486个节点和5 040条边的完整的GitHub企业社交网络图。此外，本文以3周的时间为间隔，分别抓取了GitHub上486家企业的项目数量变化量和贡献者数量变化量。

本文尊重GitHub用户的隐私，只收集了公开的信息。通过分析抓取的数据，提取了每个企业的

信息字段的值，例如项目数量、开发语言、所在国家或地区等。此外，对于每个用户，我们都获得了完整的项目列表和贡献者列表。同时，在离线环境中存储和分析以上数据。

2　相关工作

2.1　相关研究成果

随着GitHub的快速发展，研究人员针对其用户影响力进行了一系列衡量和预测[3]。李变[4]对基于链接分析的HITS算法进行改进，提出一个基于用户属性矩阵和HITS算法的社交网络中用户影响力评估算法。Hu等人[5]提出了一种基于Follow-Star-Fork-Activity的方法来衡量用户在GitHub社交网络中的影响力。通过构建社交网络，分析用户在社交网络中的影响力，包括受欢迎程度、中心性、内容价值、贡献和活动，使用波达计数法（Borda count）对用户影响力进行全面量化，提出了一个基于关注关系、星标关系、分叉关系和用户活动的影响力衡量框架。文献[6]则对关注者数量、分叉项目数量和GitHub中的项目观察者数量3个影响指标进行深入比较，其结果表明3个影响指标捕获了两个主要特征：受欢迎程度和内容价值（代码可重用性）。文献[7]将深度学习方法应用到社交网络用户的影响力预测，引入图卷积与图注意力两种方法，学习社交网络模型的隐藏参数。由于兼顾了用户的节点特征与所处子图的结构特征，更全面地了解用户偏好与信息传播模式，大大提升了社交网络中用户影响力的预测精度。

如何对GitHub中存储库的影响力进行预测也引起了广泛关注[8]。Borges等人[9]认为存储库的star数是其影响力的直接衡量标准，揭示了影响GitHub项目star数量的主要因素，并对2 279个GitHub存储库的star数的时间序列进行聚类，得到了流行度增长的4种主要模式。使用多元线性回归来预测GitHub存储库的star数，基于存储库过去6个月得到的star数量训练模型并进行预测。此外，对于star数量较少或者增长缓慢的存储库，使用来自具有相同增长趋势的存储库数据生成特定预测模型。Varuna等人[10]在GitHub上执行趋势预测，使用最相关且重复发生的事件构建多变量时间序列，使用长短期记忆（LSTM）模型预测star数量变化趋势。Han等人[11]从GitHub中提取了35个特征并构建随机森林分类器，以预测GitHub项目的受欢迎程度。在从GitHub收集的大规模数据集上评估模型性能，实现了0.76的平均AUC。Ren等人[12]提出了一种基于star数量的StarIn方法来预测GitHub存储库的影响力。以GitHub中的关注者为基础数据集建立网络，从基本影响、网络动态影响和网络静态影响3个方面共6个特征的角度分析了StarIn与存储库影响力大小的相关性。StarIn在两种不同的分类器下对预测存储库的影响力大小均实现了较高的准确率。

2.2　XGBoost

XGBoost通过不断地加入新的弱模拟器来拟合前一个弱评估器训练所产生的误差，使得真实值与预测值的残差不断减小。在模型迭代结束后，将每个评估器上的预测结果进行加权组合即可得到预测结果[13]。XGBoost的预测函数见式（1）。

$$y_i^* = \sum_{i=1}^{K} f_k(x_i), f \in F \tag{1}$$

其中x_i为输入的第i个样本，K为决策树的数目，$f_k(x_i)$为第i个样本在第k棵树上的预测值。XGBoost的目标函数由损失函数与正则化两部分组成，其正则化的目标函数见式（2）和式（3）。其中$\Omega(f_k)$代表树的复杂度，y_i^{k-1}为前k棵树的预测值，γ、λ为控制惩罚程度的超参数，T代表叶子节点个数，w代表每个叶子节点的值。

$$L^k = \sum_{i=1}^{n} L\left[y_i, y_i^{k-1} + f_k(x_i)\right] + \Omega(f_k) \tag{2}$$

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \| w \|^2 \tag{3}$$

2.3　SHAP解释框架

集成学习和深度学习等黑盒模型一般具有良好的精度，但由于模型内部复杂度高，使得建立的模型缺乏可解释性，人们无法直观理解。Lundberg等人[14]于2017年提出SHAP框架，这是一种用于复杂机器学习模型的解释框架。该解释模型与人类直觉一致，近年来在解释医疗和社会现象相关模型方面日益流行。SHAP是一种经典的事后归因解释框架，通过计算每个特征加入模型时的贡献进行模型解释。其核心思想是Shapley value，最早由Shapley于1953年提出，主要是用来解决合作博弈论中的分配均衡问题。SHAP是一类可加性解释模型，将每个特征都视为贡献者，通过计算其具体的贡献值，并将所有特征的贡献值加和得出模型的最终预测。对于每一个样本而言，样本最终的预测值等于

平均预测值与样本各特征的SHAP值之和。这确保了贡献值加和等于最终输出，消除了各个模型间结构带来的解释性差异，能解决很多特征归因方法不一致的问题。此外，传统的特征重要性只能说明哪个特征重要，但并不清楚该特征是怎样影响预测结果的。而SHAP可以反映出各个特征的影响力以及影响的正负性，揭示该特征怎样影响预测结果，这是其最大的优势所在。

3 GitHub企业社交网络分析

随着在线社交应用的普及，社交网络成为目前学者们研究的热点[15]，其中对节点影响力的分析也取得了一系列成果。在社交网络中，已有大量研究通过融入中心性指标，对节点用户的影响力进行识别和评估。一般来说，把用户当作网络中的节点，用户之间的联系就是网络中的边。这种联系可以是用户之间相同的兴趣爱好所产生的连接，也可以是用户之间的合作或竞争关系等。这一概念与本文构建的企业社交网络具有高度相似性。

企业影响力的研究数据主要分为两大类。一类是企业自身的属性，如项目数量、项目增加量、贡献者数量等。另一类是基于企业间的联系产生的属性，如企业之间共同的贡献者、共同参与的项目等，这种联系构成了企业之间的社交网络[16]。目前已有的针对企业影响力的研究只考虑到企业的自身属性，忽略了企业在网络中的位置和重要性，因此本文引入社交网络分析，以企业作为节点，以两家企业之间的共同贡献者作为联系依据，利用NetworkX构建企业社交网络。

本文使用486家企业的数据集，根据企业之间的联系紧密程度建立无向图，并进行可视化，如图2所示。每个节点代表一个GitHub上的企业，若两家企业的共同贡献者多于（或者等于）100人，即联系紧密，增加一条边；若共同贡献者少于100人，即联系不紧密，则不增加边。边的连接代表了企业之间联系的紧密程度。从图2中可以看到，许多节点是高度连接的，并且节点倾向于形成集群，即一些企业占据网络的中心，有很多企业与它们有相连的边，比如Apache、Nokia、Shopify等。也有一些企业几乎不与其他企业相连，比如AOL、安居客、FacePlusPlus等。图例显示了拥有边数最多的前10家企业，它们均为行业代表性企业或者高新技术企业，拥有大量用户、优质声誉以及高度的影响力。

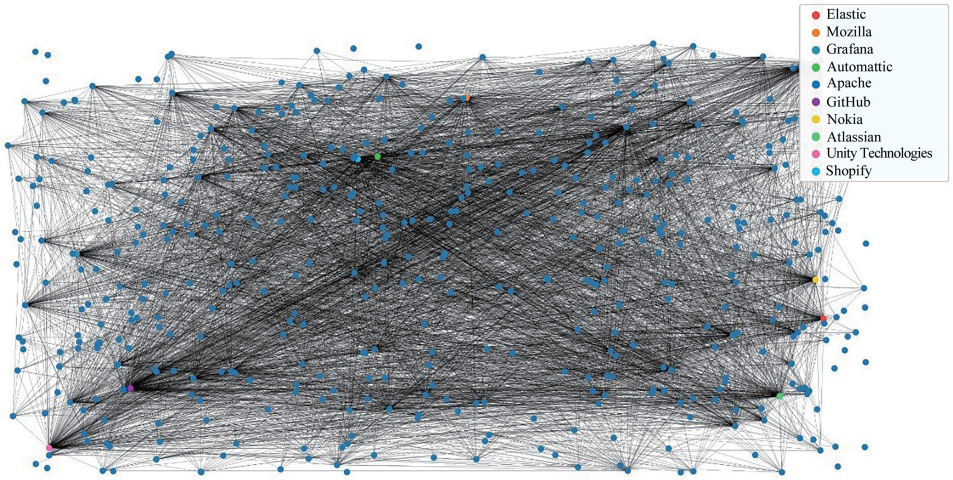


图2 486家企业社交网络可视化（以100作为连接边的阈值）

3.1 中心性度量

节点的中心性（centrality）用于定量地衡量节点在网络中的重要性，因此可用于判定节点在网络中的影响力大小，本文将它们作为企业联系维度的特征进行建模[17]。节点中心性的4个主要度量指标分别为度量中心性、介数中心性、接近中心性和特征向量中心性。表2为各个指标数值排名前15的企业及其具体数值。由表2可以看出，Microsoft在4项中心性度量中均排名第一，说明其位于企业网络的最中心。

表2　中心性度量排名前15名的企业

排名	度量中心性	介数中心性	接近中心性	特征向量中心性
1	'microsoft': 0.522	'microsoft': 0.110	'microsoft': 0.554	'microsoft': 0.164
2	'shopify': 0.470	'shopify': 0.070	'shopify': 0.521	'shopify': 0.162
3	'unity-technologies': 0.406	'unity-technologies': 0.030	'unity-technologies': 0.486	'unity-technologies': 0.158
4	'atlassian': 0.359	'automattic': 0.026	'atlassian': 0.461	'atlassian': 0.151
5	'nokia': 0.346	'atlassian': 0.021	'nokia': 0.456	'nokia': 0.149
6	'github': 0.338	'github': 0.019	'github': 0.454	'github': 0.1 464
7	'apache': 0.326	'apache': 0.019	'apache': 0.446	'apache': 0.144
8	'automattic': 0.282	'nokia': 0.017	'automattic': 0.431	'grafana': 0.139
9	'grafana': 0.280	'mozilla': 0.014	'grafana': 0.429	'segmentio': 0.135
10	'mozilla': 0.274	'JetBrains': 0.009	'mozilla': 0.428	'hashicorp': 0.134
11	'elastic': 0.266	'facebook': 0.009	'elastic': 0.423	'elastic': 0.133
12	'segmentio': 0.264	'intel': 0.008	'segmentio': 0.422	'hubspot': 0.132
13	'facebook': 0.260	'adobe': 0.006	'facebook': 0.421	'appian': 0.131
14	'hashicorp': 0.260	'alibaba': 0.006	'hashicorp': 0.421	'nextdoor': 0.131
15	'hubspot': 0.256	'meteor': 0.006	'hubspot': 0.419	'digitalocean':0.130

3.1.1　度量中心性

度量中心性（degree centrality，DC）根据每个节点存在的连接数分配重要性分数，即节点的中心性越高，连接到特定节点的边就越多，该节点的邻居节点就越多。事实上，节点的中心性程度是它所连接的节点的比例，也就是特定节点连接到网络节点的百分比。在一个包含N个节点的网络中，节点i的度量中心性定义为：

$$DC_i = \frac{l_i}{N-1} \tag{4}$$

其中，l_i表示现有的与节点i相连的边的数量，N-1表示节点i与其他节点都相连的边的数量。图3为企业中心性度量条形图，横坐标为各项中心性指标的具体数值，纵坐标为中心性指标为该数值的企业数量。由图3可以看出，企业的度量中心性的分布显示出长尾效应，大多数企业集中在头部，即度量中心性较低处；只有零散、少量的企业分布在尾部，度量中心性越大，企业数量越少。有3家企业的度量中心性高于0.4，分别是Microsoft、Shopify、Unity-Technologies。这是因为在大多数社会网络中，节点的度遵守幂律分布，度较大的节点数量在网络的节点总数量中占比很少，而度较小的节点的数量却占很大比例。所以许多节点具有极低的度量中心性，因为它们在这个网络中并不相互连接。

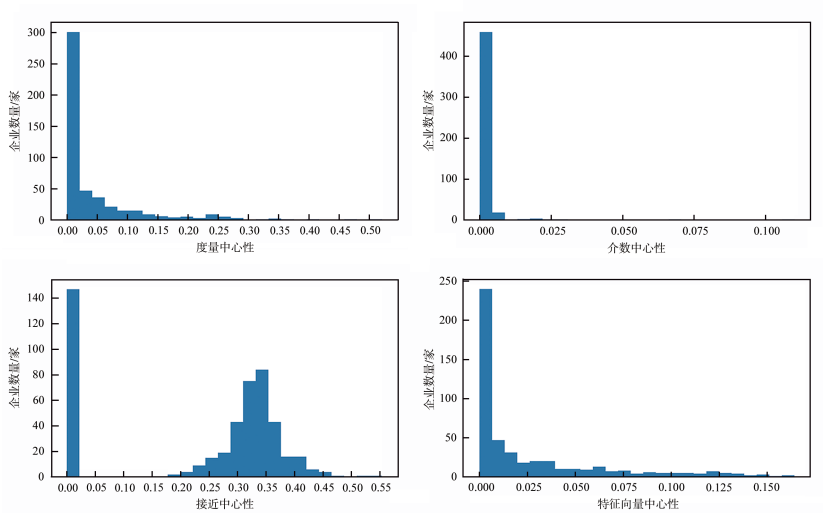


图3　企业中心性度量

3.1.2 介数中心性

介数中心性（betweenness centrality，BC）用于衡量节点影响其他节点的能力，数值越高，说明该企业充当许多非连接企业的桥梁，能够传达较大的信息流量，影响和连接其他企业。节点 i 的介数中心性定义为：

$$BC_i = \sum_{s \neq i \neq t} \frac{n_{st}^i}{g_{st}} \tag{5}$$

其中，n_{st}^i 表示经过节点 i 且为最短路径的路径数量，g_{st} 表示连接 s 和 t 的最短路径的数量。由图 3 可以看出，绝大多数企业的介数中心性低于 0.025，而 Microsoft 拥有最高的介数中心性，为 0.110。介数中心性衡量节点位于其他节点之间最短路径上的次数。节点的介数中心性与节点影响其他节点的能力有关。具有高介数中心性的企业充当许多非连接企业之间的桥梁，因此能够通过传达信息来影响他们，甚至通过企业的圈子连接他们。

3.1.3 接近中心性

接近中心性（closeness centrality，CC）根据每个节点与网络中所有其他节点的“接近度”对每个节点进行评分。对于一个节点，它的接近中心性测量其到所有其他节点的平均远度。即企业的接近中心性越高，它离网络中心越近。接近中心性度量对于监控虚假信息或病毒的传播非常重要，因为虚假信息必须首先到达具有高度接近中心性的节点，然后再将其传播到网络的许多不同部分。一个含有 N 个节点的网络的节点 i 的接近中心性定义为：

$$CC_i = \frac{1}{\frac{1}{N}\sum_{j=1}^{N'} d_{ij}} \frac{N}{\sum_{j=1}^{N'} d_{ij}} \tag{6}$$

其中，d_{ij} 表示节点 i 到节点 j 的距离。由图 3 可以看出，企业按接近中心性的大小聚集成两个集群，一个接近为 0，另一个则集中在 0.175 至 0.475 之间。如果接近中心性较高的企业开始传播错误信息，整个网络将会很快获得错误信息。但是，如果接近中心性接近为 0 的企业传播虚假信息，则虚假信息向整个网络的传播速度会慢得多。

3.1.4 特征向量中心性

特征向量中心性（eigenvector centrality，EC）是显示节点与网络中其他重要节点的连接程度的指标。它根据节点及其相邻节点在网络内的连接程度来衡量节点的影响力。特征向量中心性可以识别对整个网络影响最大的节点。一个节点的重要性既取决于其邻居节点的数量，也取决于其邻居节点的重要性。高特征向量中心性意味着节点连接到本身具有高特征向量中心性的其他节点。节点 i 的特征向量中心性定义为：

$$EC_i = \mathrm{c} \sum_{j=1}^{n} a_{i,j} x_j \tag{7}$$

其中 c 为一个比例常数，当且仅当 i 与 j 相连时 a_{ij}=1，否则为 0。由图 3 可以看出，企业的特征向量中心性同样表现出长尾效应，大多数企业的特征向量中心性接近 0，其余企业的特征向量中心性大致分布在 0 至 0.15 之间，并且数量逐渐减少，其中最大值为 0.164，为 Microsoft。该度量与企业影响整个图的能力相关联，因此具有最高特征向量中心性的企业是该网络中最重要的节点。

3.2 其他指标

3.2.1 聚类系数

节点的聚类系数（clustering coefficient）定义为两个随机选择的节点彼此存在连接的概率，它反映某个节点的邻接节点聚集成团的倾向。在现实网络中，节点总是趋向于建立紧密的组织关系。聚类系数越大，其所在局部网络的凝聚力越大，影响力也就相对较大。

3.2.2 约束度

约束度（constraint）以节点对其他节点的依赖程度作为评价标准，指网络中各个节点的联系在多大程度上是冗余的。数值越大，约束性越强，网络中存在的结构洞越少；依赖性越强，则能力越小，跨越结构洞的可能性就越小。

3.2.3 有效规模

有效规模（effective size）用于测算节点的整体影响力，该指标可以在一定程度上定量地衡量结构洞节点的重要性。

4 系统设计与构建

本文提出了一种融合 XGBoost 与 SHAP 框架的 GitHub 企业影响力预测与解释系统，利用机器学习方法构建预测模型，融合 SHAP 解释框架对各个特征的重要性进行量化和归因，进而阐释影响影响力预测结果的重要因素及其作用方式。

4.1 系统概述

本文从企业账号信息与历史数据中提取特征，将 h 指数的高低作为预测变量，构建基于极端梯度提升决策树、支持向量机和多层感知机的 3 个企业

影响力预测模型。其中，h指数高于30的企业为高影响力企业，h指数低于或等于30的企业为低影响力企业。使用网格搜索对模型的超参数进行寻优，使用5折交叉验证评估不同参数模型的能力，将得到的最优参数组合带入模型进行预测分析。针对最优预测模型，使用目前流行的SHAP解释方法，对各个特征预测结果的影响给出解释，找到对企业影响力判断影响较大的特征，并明确影响方向，提升预测模型的可靠性[18]。具体而言，预测系统的研究框架分成4个模块，如图4所示。

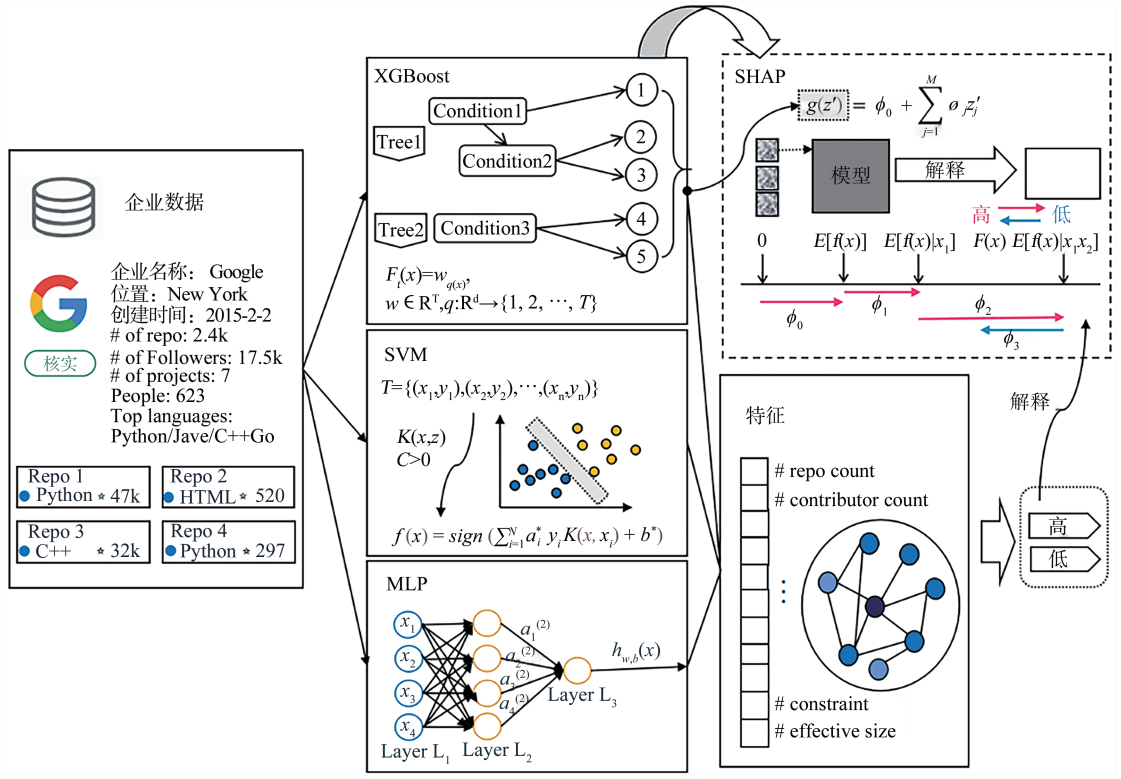


图4　GitHub企业影响力预测与解释系统框架

（1）特征提取模块。爬取GitHub信息源中的数据，筛选、剔除无效数据后，计算得出网络指标。在此基础上，进行特征提取和筛选，确定用于建模的特征。

（2）模型构建模块。使用确定的特征进行建模，分别构建基于极端梯度提升决策树、支持向量机和多层感知机的企业未来影响力预测模型。

（3）模型评估模块。将3个主流机器学习算法进行性能比较，将准确率、精准率、召回率和F1分数4个指标作为模型评估指标，找出综合最优模型。

（4）模型解释模块。引入SHAP解释框架，对综合最优模型进行可视化和解释说明，识别影响企业影响力高低的重要因素及其作用方式。

4.2　特征提取

本文对企业数据进行特征提取与特征筛选，确定特征项及其数值，并根据企业影响力相关特征的不同特性，将其分为企业账户特征、时间变化特征、企业关联特征3个维度。企业账户特征是指与企业GitHub账户本身相关的特征，是企业账号的固有特征，例如项目数量、贡献者数量、项目所有者ID、员工数量、主要开发语言、注册所在国家等。企业账号的固有特征有很多，考虑到模型目标是对企业影响力的高低进行预测，因此本文剔除了难以进行量化和大小比较的特征，例如项目所有者ID、主要的开发语言、注册所在国家等，保留了可以进行量化和大小比较的特征，例如项目数量和贡献者数量。时间变化特征是指随着时间变化而发生演变的动态特征，包括项目数量、贡献者数量在一定时期内的变化情况。企业关联特征是指通过网络建模和计算得到的反映企业之间关联度的特征，可用于衡量企业与其他企业的连接程度。本文将所有企业之间的联系构建成一个企业社交网络，每个节点代表一个企业，每一条边代表相邻的两个企业拥有超过100个共同贡献者。企业的影响力越强，

则它在网络中的节点越重要。企业关联特征包括度量中心性、介数中心性、聚类系数等。

通过人工筛选的方式，本文剔除了 6 家长期无更新的企业，最终数据集包括 480 家企业。通过对企业进行网络建模和计算，得到相关性度量。最后，共确定了 11 个特征用于建立预测模型，分别是项目数量、贡献者数量、项目变化量、贡献者变化量、度量中心性、介数中心性、接近中心性、特征向量中心性、聚类系数、约束度和有效规模。

4.3 预测模型构建与调优

根据企业的项目数量和 star 数量计算其 h 指数，以 h 指数为标签，以 30 为阈值，将高于该值的企业标记为高影响力企业，低于或等于该值的企业标记为低影响力企业。使用 XGBoost 模型对企业影响力进行建模，并将支持向量机、多层感知机这两个常用的主流机器学习模型作为对比模型进行性能比较，以验证 XGBoost 模型是否能提供最佳结果。

参数调优不仅可以最大化 XGBoost 模型的性能，对于防止过拟合也具有重要作用。网格搜索（GridSearch）是应用最广泛的参数搜索算法，通过循环遍历查找搜索范围内所有的参数，从而确定最优值。交叉验证可以在一定程度上减小过拟合，并从有限的数据中获取尽可能多的有效信息。因此我们使用网格搜索和交叉验证（GridSearchCV）来自动化参数的调优过程。实验结果显示，交叉验证后的最佳 XGBoost 超参数值为：learning_rate: 0.5，n_estimators: 80，min_child_weight: 5，subsample: 0.8。

4.4 模型评估

随机将数据集中 80% 的样本划分为训练集，20% 划分为测试集，预测模型仅使用来自训练集的数据建立。将训练集不重复抽样随机分成 5 份，每次使用其中 4 份训练模型，余下 1 份用来验证 4 次模型训练的准确率，重复该步骤 5 次，直到每个子集都有一次机会作为验证集。计算 5 组测试结果的平均值作为模型精度的估计，并作为当前 5 折交叉验证下模型的性能指标。分别对 3 个分类模型进行训练和分类预测。使用准确率（accuracy）、精准率（precision）、召回率（recall）、F1 分数（F1-score）以及指标受试者工作特征曲线（receiver operating characteristic curves）下面积 AUC 评估模型性能，从而找出综合最优模型。

准确率指正确分类的样本数与总样本数之比，精准率指在所有被预测为正的样本中实际为正的样本的概率，召回率指在实际为正的样本中被预测为正样本的概率，F1 分数为综合评价指标。计算方法为：

$$\text{accuracy} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} \tag{8}$$

$$\text{precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} \tag{9}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} \tag{10}$$

$$\text{F1\_score} = \frac{2\times\text{precision}\times\text{recall}}{\text{precision}+\text{recall}} \tag{11}$$

其中 TP，FP，TN，FN 分别表示真阳性、假阳性、真阴性、假阴性。

4.5 模型解释

在完成企业影响力预测后，在综合最优模型上应用 SHAP 解释框架进行可视化和解释说明，大大增强预测结果的可解释性。该框架可以分析影响企业影响力预测结果的重要因素，并结合 SHAP3 类可视化图表来阐明各特征如何影响预测结果，具体如下。

（1）揭示影响的重要性，即各个特征对提升整个预测模型的贡献程度。它可以直观地反映特征对模型的影响程度，发现对预测模型影响较大的特征。运用 SHAP 特征重要性条形图，来反映特征对模型的影响程度，发现对企业影响力影响较大的特征。

（2）反映影响的总体方向性。将特征重要性与特征效应结合分析，刻画出特征值与预测结果的总体正负关系，明确特征对预测结果的影响方向。通过绘制 SHAP 概要图，刻画出特征值与 h 指数的总体正负关系，明确特征值对企业影响力高低的影响方向。

（3）特征依赖分析。通过分析各特征对企业影响力的边际效应，可以反映该特征对企业影响力判断的影响方式是线性的、单调的还是复合的，并展示特征数据变动对企业影响力预测值影响的大小，即抬升力度或降低力度的大小。借助 SHAP 部分依赖图，分析各特征的边际效应，剖析各特征变量具体取值如何影响企业影响力高低，明确特征变量的影响方式及其影响力度（抬升力度或降低力度）。

5 实验结果

5.1 预测模型实验结果

为了实现 XGBoost 模型构建，本文使用 scikit-

learn，这是一个基于Python的机器学习库。关于XGBoost模型的参数，我们将弱学习器的数量设置为80，学习率设置为0.5。

本文比较了XGBoost和其他两种流行的机器学习算法的性能，在训练集上分别对3种预测模型进行训练，并在测试集上进行性能测试。首先将阈值设置为30，h指数高于30的企业被判断为高影响力企业，h指数低于或等于30的企业被判断为低影响力企业。使用准确率、精准率、召回率和F1分数4种评价指标分别对3种模型的预测结果进行评估，结果如表3所示。从表3可以看出，基于XGBoost的预测模型各项指标均高于其他两种模型。

表3　预测模型性能对比（阈值为30）

预测模型	参数	accuracy	recall	F1-score	precision
XGBoost	learning_rate=0.5, n_estimators=80, min_child_weight=5, subsample=0.8	0.918	0.914	0.889	0.865
SVM	kernel=`linear`, C=16	0.800	0.886	0.861	0.838
MLP	solver=`lbfgs`, hidden_layer_sizes=(5,2), max_iter=1000	0.837	0.714	0.756	0.806

为了确保实验结果对不同阈值均有效，本文将划分企业影响力高低的阈值更改为15、20、25、30、35和40，并重复进行实验。图5显示了当阈值分别为这6个值时，3种模型所能达到的准确率、精准率、召回率和F1分数。由图5可以看到，基于XGBoost的模型在不同阈值下均表现出更稳定的性能和更准确的预测结果。XGBoost分类模型在准确率、精准率、召回率以及F1分数上整体优于其他对比模型。当阈值为25时，XGBoost分类模型的召回率略低MLP分类模型。

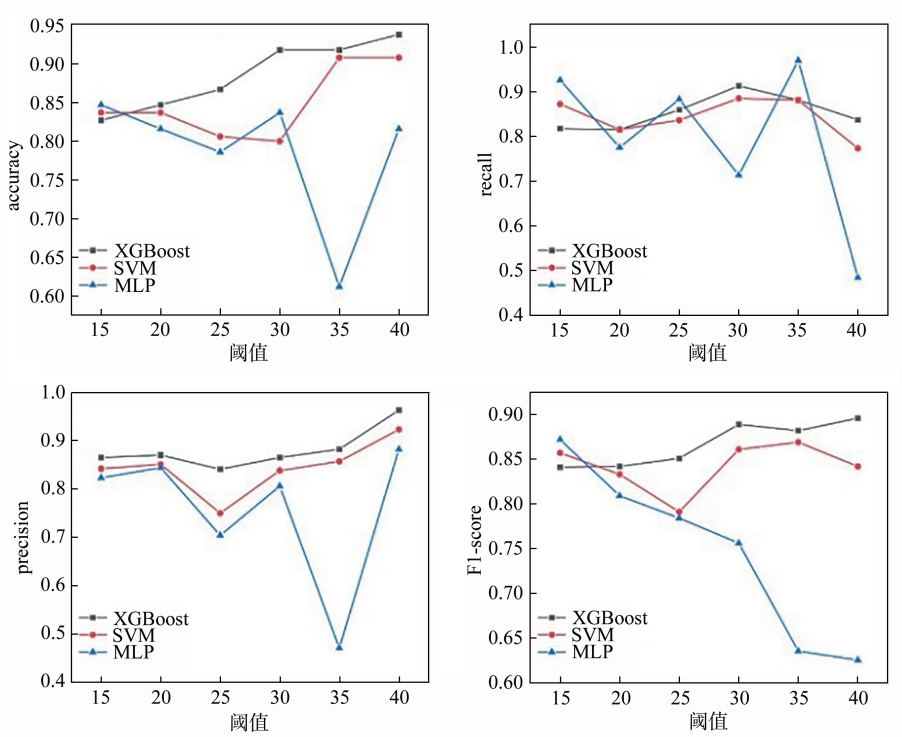


图5　不同阈值下3种预测模型的预测结果对比

此外，本文在不同阈值下计算了模型的AUC。AUC是综合度量指标，为受试者操作特征曲线下的面积，而该曲线图是反映敏感性与特异性之间关系的曲线。横坐标X轴为1-特异性，也称为假阳性率（误报率），X轴越接近零准确率越高；纵坐标Y轴称为敏感度，也称为真阳性率（敏感度），Y轴越大代表准确率越好。

当阈值分别为20、25、30时，XGBoost分类模型的AUC如图6所示。实验结果显示，在3种阈值下，XGBoost分类模型的AUC均达到了0.9以

上，表现出较高的稳定性和准确性。综上所述，XGBoost是综合最优模型。

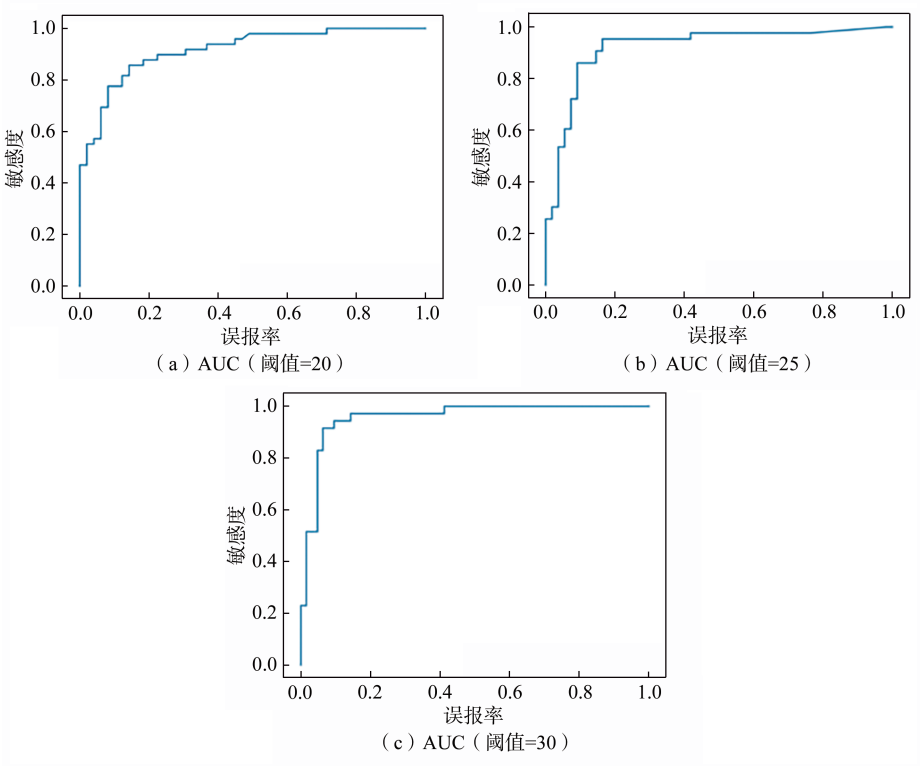


图6　不同阈值下XGBoost模型的AUC值

5.2　解释模型实验结果

SHAP特征概要图可以直观地反映特征对模型的影响程度。特征重要性越高，该特征对企业影响力判断的影响就越大[19]。基于XGBoost的预测模型的特征概要如图7所示，每一行代表一个特征，横坐标为SHAP值。若SHAP值大于0，代表正向影响；若SHAP值小于0，代表负向影响。特征的排序按照SHAP的平均绝对值，从上到下重要性依次降低。图中每一个点代表一个样本，宽的地方表示有大量的样本聚集，颜色越红说明特征本身数值越大，颜色越蓝说明特征本身数值越小。以第一行为例，高项目数量（红色）对预测是正向影响，低项目数量（蓝色）对预测是负向影响。此外，项目数量特征的样本分散非常广，因此对于模型非常重要。而项目变化量特征基本围绕在SHAP=0附近，说明对模型影响很小。

从图7中可以看到，特征的重要性从大到小为：项目数量、贡献者数量、有效规模、聚类系数、约束度、贡献者变化量、介数中心性、度量中心性、接近中心性、特征向量中心性和项目变化量。也就

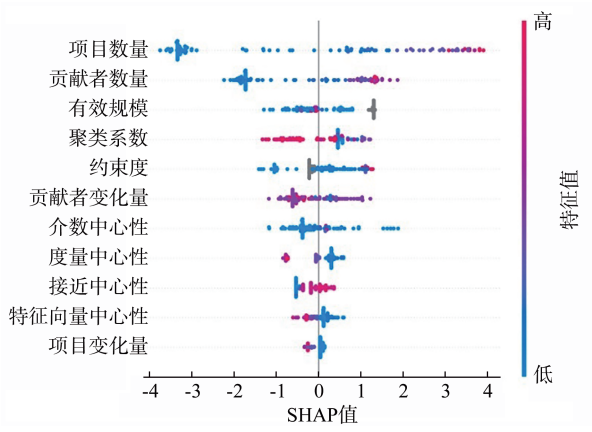


图7　SHAP特征概要图

是说，企业影响力高低受项目数量和贡献者数量影响较大，受项目变化量影响较小。其中，企业本身的特征是影响企业影响力高低的重要特征维度。

5.3　与其他研究成果的比较

在相关工作中，Han等人[11]的研究与本文工作最为相关。他们提取了GitHub项目的3个维度共35个特征，使用随机森林建立预测模型，探索各种特征与GitHub项目影响力高低之间的关系。他们将star数量大于100的项目定义为高影响力项目，

否则定义为低影响力项目，从GitHub项目中提取了2015年的特征，并用它们来预测一个项目在未来是否会流行。与Han等人[11]的研究相比，本文旨在解决一个相关但不同的问题。本文的研究对象是GitHub上的企业，我们提出了一种自动化的方法来预测企业的受欢迎程度。为了进行比较，我们同样利用随机森林作为底层分类器，基于他们提出的特征来构建预测模型。为了与企业账号的特征相对应，同时避免潜在的偏差，我们删除了个人项目特有的信息，比如readme文件的大小、commits的数量、代码行数的增加量以及项目的主题分布等。

本文使用准确率、精准率、召回率和F1分数作为评估指标。评估结果如表4所示。对于各项指标，本文提出的模型效果均优于基于随机森林构建的模型，其中本文预测模型可以实现0.914的召回率，比Han等人提出的模型提升了约10%。

表4　GitHub影响力预测模型的比较

模型	accuracy	recall	F1-score	precision
本文预测模型	0.918	0.914	0.889	0.865
Han等人提出的模型[11]	0.865	0.816	0.827	0.838

6　结论与展望

6.1　主要结论

本文提出一种融合XGBoost与SHAP的可解释的企业影响力预测模型，并以GitHub平台的486家企业为例，进行实证研究。结果发现，XGBoost能较好地预测企业未来影响力高低，模型精度较高，且在准确率、精准率、召回率和F1指数4项性能指标上均优于对比模型。在此基础上，模型融入SHAP可解释性框架，对各特征的重要性进行量化和归因，可以精准识别其作用方式。以往基于GitHub平台的研究不多，且绝大部分集中在针对个人账户的研究[20]。本文致力于对社交平台上的企业影响力进行公平、全面、可信的综合评价和量化，将不变特征与演变特征有机结合，从多个维度解释了特征如何影响企业影响力高低，进一步明确企业影响力评判体系[21]，能够丰富有关GitHub平台的研究成果，对于发掘高潜力优质企业具有重大意义。

6.2　不足与展望

GitHub是一个全球领先的在线开发人员社区，企业开设官方账户发布公开可见的项目，既是展示自身实力和成果的机会，也给了社会大众观察和评判的空间，这既是机遇也是挑战[22]。本文构建的GitHub企业影响力预测可解释系统能够发现哪些企业具有发展潜力，哪些企业可能处于衰退阶段，从而进行社会资源的倾斜和扶持，促进优质企业快速成长。

本文结论仅适用于GitHub平台，但构建的企业影响力可解释性预测系统的思路可以被广泛迁移用于其他社交平台。后续将从以下几个方面进一步探索：一是对预测模型进行改进，进一步提高预测的准确度；二是继续定期抓取数据进行建模预测，本文只使用了3周的时间变化数据，未来将加入半年乃至一年内的变化量作为时间维度的特征；三是将寻找更有代表性的数据源，使用更大规模的样本数据开展研究。

参考文献：

[1] GONG Q Y, LIU Y S, ZHANG J Y, et al. Detecting malicious accounts in online developer communities using deep learning[C]//Proceedings of IEEE Transactions on Knowledge and Data Engineering. Piscataway: IEEE Press, 2023: 1-17.

[2] LIU C, YANG D, ZHANG X H, et al. Recommending GitHub projects for developer onboarding[J]. IEEE Access, 2018, 6: 52082-52094.

[3] 李占山, 刘兆赓. 基于XGBoost的特征选择算法[J]. 通信学报, 2019, 40(10): 101-108.
LI Z S, LIU Z G. Feature selection algorithm based on XGBoost[J]. Journal on Communications, 2019,40(10):101-108.

[4] 李变. 基于GitHub社交网络中用户影响力评估算法的研究[D]. 西安: 西安电子科技大学, 2015.
LI B. Research on user influence evaluation algorithm in GitHub-based social networks[D]. Xi'an: Xidian University, 2015.

[5] HU Y, WANG S, REN Y, et al. User influence analysis for GitHub developer social networks[J]. Expert Systems with Applications, 2018, 108:108-118.

[6] BADASHIAN A S, STROULIA E. Measuring user influence in GitHub: the million follower fallacy[C]//Proceedings of the 3rd International Workshop on Crowd Sourcing in Software Engineering. New York: ACM, 2016: 15-21.

[7] LI S Y, YANG J, LUENG G, et al. SybilFlyover: heterogeneous graph-based fake account detection model on social networks[J]. Knowledge-Based Systems, 2022, 258: 110038.

[8] ZHANG P C, XIONG F, LEUNG H, et al. FunkR-pDAE: personalized project recommendation using deep learning[J]. IEEE Transactions on Emerging Topics in Computing, 2021, 9(2): 886-900.

[9] BORGES H, HORA A, VALENTE M T. Predicting the popularity of GitHub repositories[C]//Proceedings of the 12th International Conference on Predictive Models and Data Analytics in Software Engineering. New York: ACM, 2016: 1-10.

[10] VARUNA T V, MOHAN A. Trend prediction of GitHub using time series analysis[C]//Proceedings of 2019 10th International Conference on

Computing, Communication and Networking Technologies (ICCCNT). Piscataway: IEEE Press, 2019: 1-7.
[11] HAN J X, DENG S G, XIA X, et al. Characterization and prediction of popular projects on GitHub[C]//Proceedings of 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC). Piscataway: IEEE Press, 2019: 21-26.
[12] REN L M, SHAN S M, XU X J, et al. StarIn: an approach to predict the popularity of GitHub repository[M]//Communications in Computer and Information Science. Singapore: Springer Singapore, 2020: 258-273.
[13] RAHMAN M M, ROY C K, COLLINS J A. CORRECT: code reviewer recommendation in GitHub based on cross-project and technology experience[C]//Proceedings of 2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C). Piscataway: IEEE Press, 2017: 222-231.
[14] LUNDBERG S M, ERION G, CHEN H, et al. From local explanations to global understanding with explainable AI for trees[J]. Nature Machine Intelligence, 2020, 2(1): 56-67.
[15] 贾俊, 张斌, 李志远, 等. 基于用户行为分析的个性化推荐算法[J]. 智能科学与技术学报, 2019, 1(4): 421-426.
JIA J, ZHANG B, LI Z Y, et al. Personalized recommendation algorithm based on user behavior analysis[J]. Chinese Journal of Intelligent Science and Technology, 2019, 1(4): 421-426.
[16] THUNG F, BISSYANDÉ T F, LO D, et al. Network structure of social coding in GitHub[C]//Proceedings of 2013 17th European Conference on Software Maintenance and Reengineering. Piscataway: IEEE Press, 2013: 323-326.
[17] 陈妍, 罗雪琴, 梁伟, 等. 基于情感信息融合注意力机制的抑郁症识别[J]. 智能科学与技术学报, 2022, 4 (4): 600-609.
CHEN Y, LUO X Q,LUENG W, et al. Depression recognition based on emotional information fused with attentional mechanism[J]. Chinese Journal of Intelligent Science and Technology, 2022, 4 (4): 600-609.
[18] 陈德旺,蔡际杰,黄允浒. 面向可解释性人工智能与大数据的模糊系统发展展望[J]. 智能科学与技术学报, 2019, 1(4): 327-334.
CHEN D W, CAI J J, HUANG Y H. Development prospect of fuzzy system oriented to interpretable artificial intelligence and big data[J]. Chinese Journal of Intelligent Science and Technology, 2019, 1(4): 327-334.
[19] ZHAO G,DA C, ZOU Y, et al. Improving the pull requests review process using learning-to-rank algorithms[J]. Empirical Software Engineering,2019,24(4) : 2140-2170.
[20] RIQUELME F, VERA J A. A parameterizable influence spread-based centrality measure for influential users detection in social networks[J]. Knowledge-Based Systems, 2022, 257: 109922.
[21] PANCHENDRARAJAN R, SAXENA A. Topic-based influential user detection: a survey[J]. Applied Intelligence, 2023, 53(5): 5998-6024.
[22] 张钹. 人工智能进入后深度学习时代[J]. 智能科学与技术学报, 2019, 1(1): 4-6.
ZHANG B. Artificial intelligence is entering the post deep-learning era[J]. Chinese Journal of Intelligent Science and Technology,2019, 1(1): 4-6.

[作者简介]



王明宇（1998-），女，2022 年在格拉斯哥大学计算机科学学院获得硕士学位，2022—2023 年在复旦大学计算机科学技术学院从事科研助理工作。主要研究方向为社交网络、机器学习和大数据分析。



宫庆媛（1991− ），女，博士，2020 年在复旦大学计算机科学技术学院获得博士学位，2020—2022 年在复旦大学计算机学院继续从事博士后研究工作。2022 年起在复旦大学智能复杂体系基础理论与关键技术实验室担任青年副研究员。主要研究方向为在线社交网络用户行为大数据。



瞿晶晶（1990− ），女，博士，上海人工智能实验室治理研究中心副研究员。主要研究方向为人工智能创新与治理、科技与社会、科学学。



王新（1973− ），男，博士，复旦大学计算机科学技术学院党委书记、教授、博士生导师，主要研究方向为新一代互联网体系结构、无线与移动网络、数据中心网络、社交网络、网络编码的应用。