

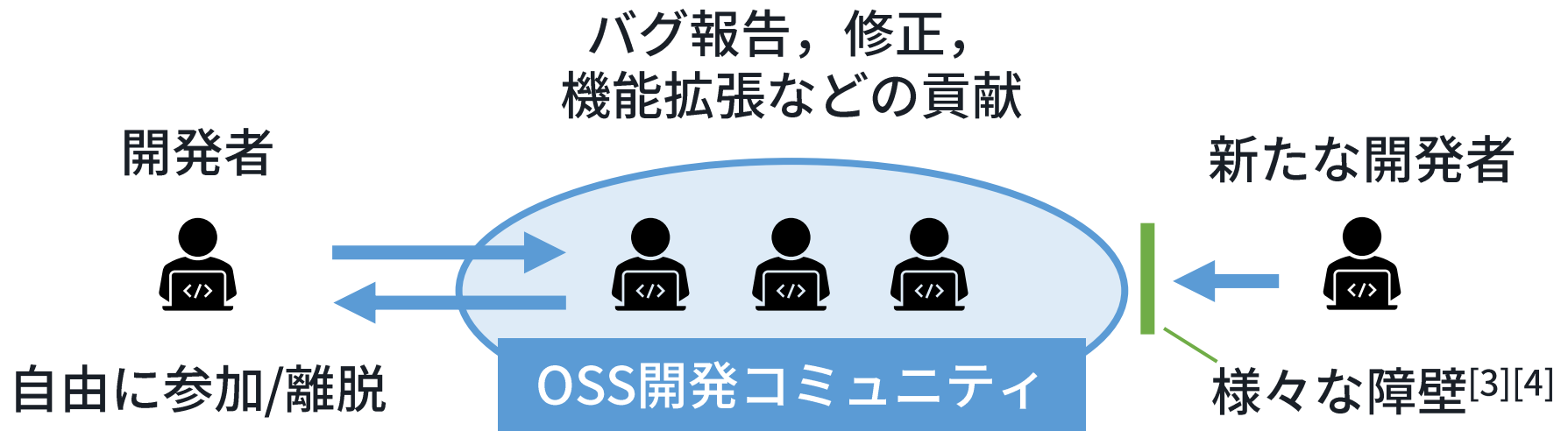
Good First Issueを用いた OSSプロジェクトへの オンボーディング支援

M2 堀口 日向

研究発表

2021/5/12

1. 背景



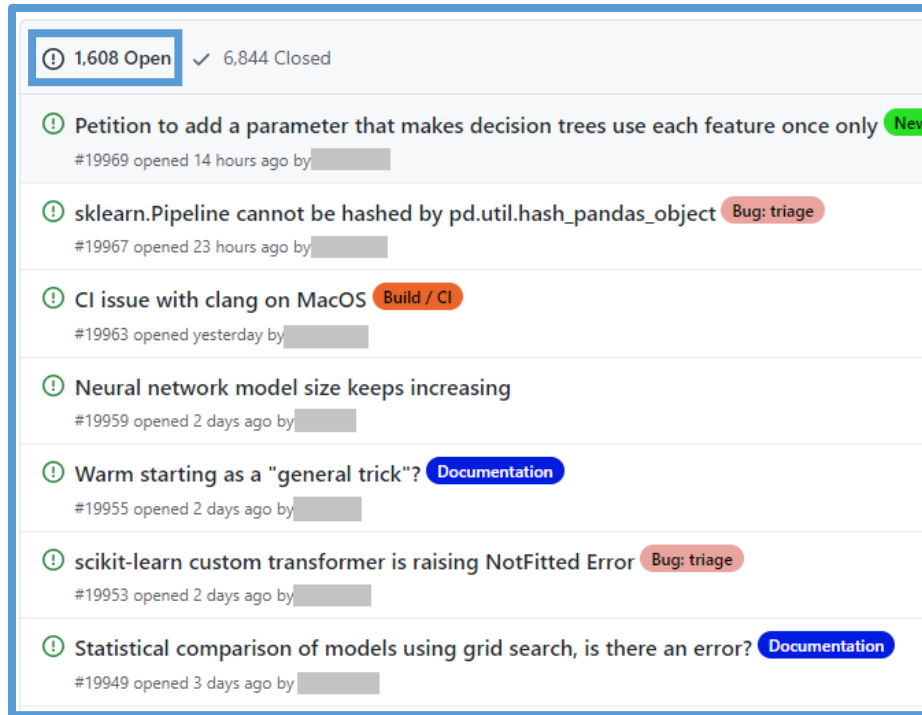
- プロジェクトを持続可能にするため、常に新たな開発者からの貢献を求めている^{[1][2]}
- 特にOSS開発経験の少ない開発者は、障壁の影響が大きく、コミュニティへの参加 (オンボーディング) 自体が困難

[1] Israr Qureshi and Yulin Fang. Socialization in Open Source Software Projects: A Growth Mixture Modeling Approach. Organizational Research Methods, pp. 208–238, 2011.
[2] Forte, A. and Lampe, C. Defining, Understanding, and Supporting Open Collaboration: Lessons From the Literature. American Behavioral Scientist 57, 5 2013, pp. 535–547.
[3] I. Steinmacher, T. Conte, M. A. Gerosa, and D. Redmiles, “Social barriers faced by newcomers placing their first contribution in open source software projects,” in CSCW ’15, 2015, pp. 1379–1392.
[4] C. Hannebauer and V. Gruhn, “On the relationship between newcomer motivations and contribution barriers in open source projects,” in OpenSym ’17, 2017, pp. 1–10.

1. 背景 | 課題選択の障壁

プロジェクト内には多くの課題 (Issue) が存在

Issue一覧の例 (Scikit-learn)



•
•
•

OSS開発経験の
少ない開発者

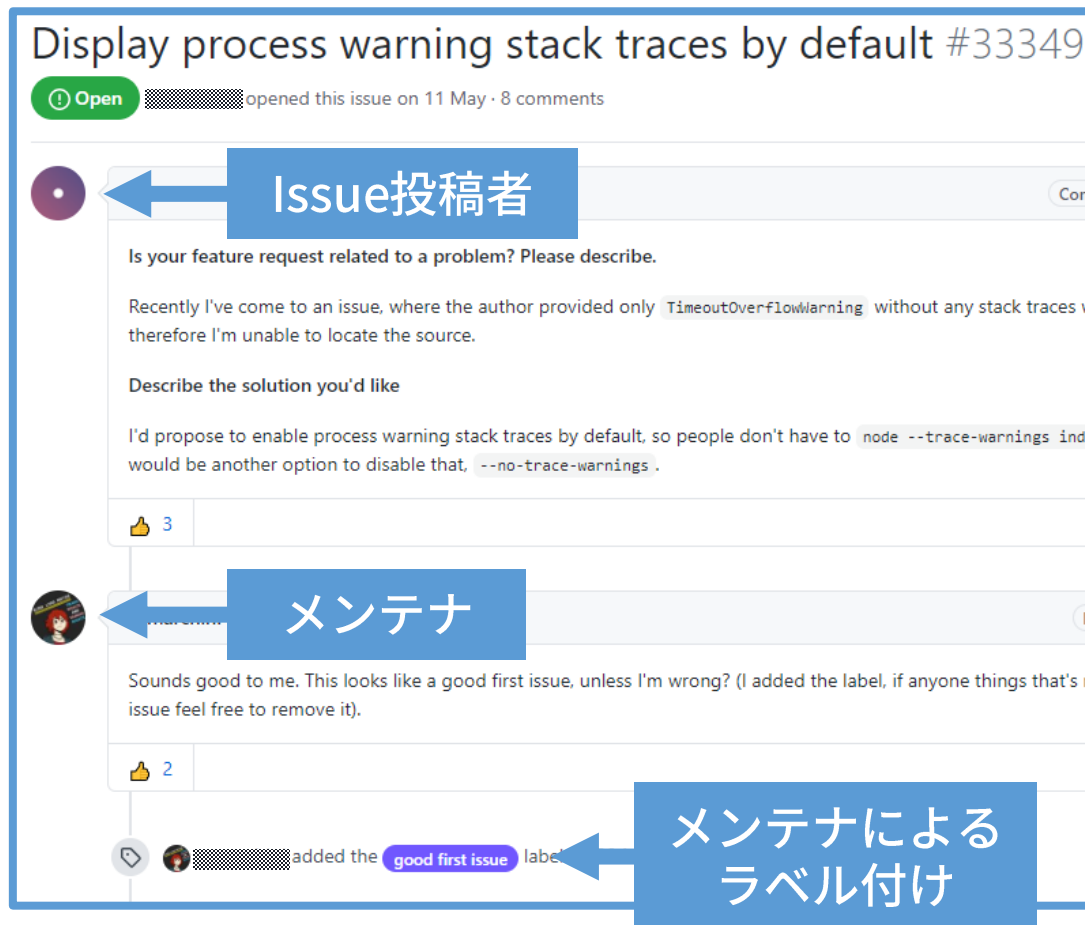


Issueの選択に多くの
時間がかかる^[5]

[5] I. Steinmacher, T. U. Conte and M. A. Gerosa, Understanding and Supporting the Choice of an Appropriate Task to Start with in Open Source Software Communities, 48th Hawaii International Conference on System Sciences, pp. 5299-5308, 2015

1. 背景 | Good First Issue (GFI)

GFIの例 (Node.js)



一部プロジェクトでは
経験の少ない
開発者向けのIssueに
「Good First Issue (GFI)」
ラベルを付ける

メンテナの負担となり
多くのプロジェクトで
利用されていない[6]

[6] <https://github.blog/2020-01-22-how-we-built-good-first-issues/>

2. 目的とアプローチ

目的

OSS開発経験の少ない開発者向けのIssueを分類する
機械学習モデルを構築

アプローチ

Step 1. 教師データ作成のためのGFI分析 ← 前回発表の内容

Step 2. ランダムフォレストを使った分類モデルの構築

Step 3. モデルの解釈と定性的な評価 ← 今後の予定

3. 前回発表までの成果

GFIの現状

- アプローチ

GFI数が多い上位11プロジェクトのGFIを分析

- 結果

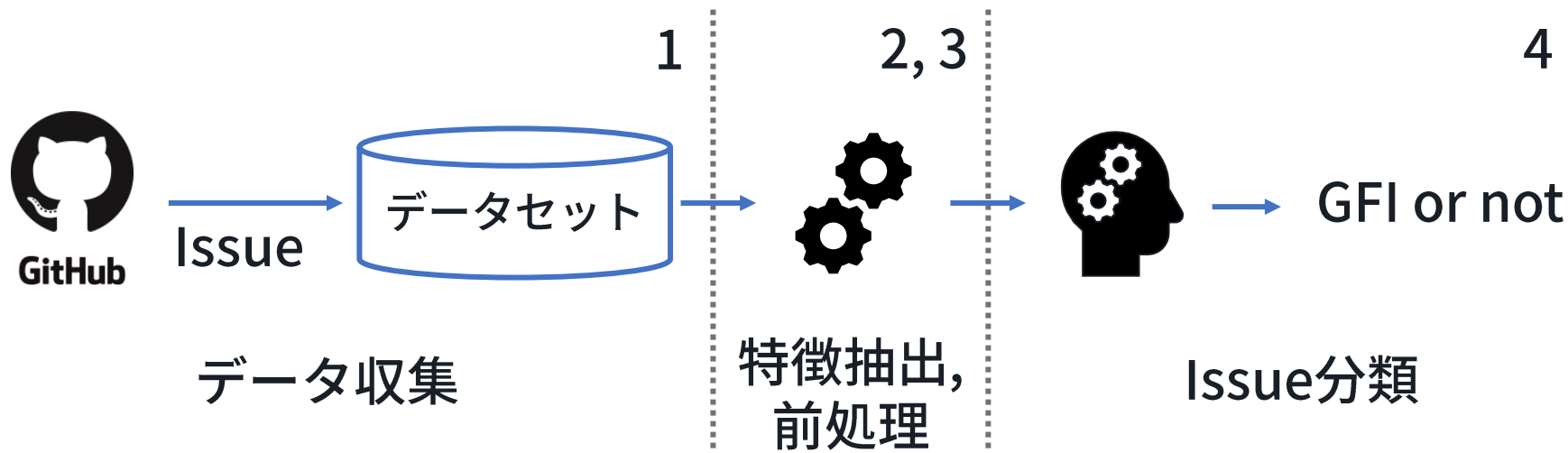
- 経験の少ない開発者に適しており，通常のIssueよりも解決しやすい

⇒ 経験の少ない開発者向けのIssue分類モデル構築のための
正例として利用できる

- GFIの内容によっては，GFI解決後に通常Issueへの貢献につながらない場合がある

⇒ GFIをスクリーニングする必要がある

4. 分類モデルの構築 | 概要



1. 通常のIssueとGFIを収集しデータセットを作成
2. モデルに入力する特徴量の抽出
3. Issueタイトルや本文の前処理
4. 分類モデルの学習と定量的な評価

4. 分類モデルの構築 (1 / 4) | データセット作成

1. GitHub上のGFIが存在するプロジェクト名を取得
→ 約7万プロジェクト
2. プロジェクトのフィルタリング
7万プロジェクトのすべてのIssueを収集するのは時間がかかる
現段階では、GFI数 > 500のプロジェクトのみ対象
→ 15プロジェクト
3. 対象プロジェクトの通常IssueとGFIを収集
最終的なデータ件数

	データ件数[件]
通常Issue (負例)	142,651
GFI (正例)	8,849

4. 分類モデルの構築 (2 / 4) | 入力する特徴量 (1 / 2)

Issue投稿時に含まれる情報のみを利用

→ タイトル, 本文, 投稿者

タイトル, 本文

自然言語 (主に英語) で記述される

⇒ 1単語ずつ (1-gram) に分割,

Bag of Wordsによるベクトル化

また, タイトル, 本文の長さは, Issueの内容が詳細に説明されているかに関連

⇒ タイトル, 本文の文字数

4. 分類モデルの構築 (2 / 4) | 入力する特徴量 (2 / 2)

Issue投稿時に含まれる情報のみを利用

→ タイトル，本文，投稿者

投稿者

プロジェクトオーナーやメンテナは，新たな開発者を惹きつけるために，GFIになりえるIssueを投稿する可能性

⇒ 投稿者とプロジェクトとの関係を数値化

(オーナー: 0, 共同開発者: 1, 関係なし: 2, etc…)

4. 分類モデルの構築 (3 / 4) | 前処理

Markdown形式で記述

コード片

インラインコード

問題の詳細な説明

Issueの例 (<https://github.com/pandas-dev/pandas/issues/39701>)

commented on 10 Feb • edited ▾

☒ I have checked that this issue has not already been reported.
☒ I have confirmed this bug exists on the latest version of pandas.
☐ (optional) I have confirmed this bug exists on the master branch of pandas.

Note: Please read [this guide](#) detailing how to provide the necessary information for us to reproduce your bug.

Code Sample, a copy-pastable example

```
import contextlib, io, json
import pandas as pd

redirect = io.StringIO()
with contextlib.redirect_stdout(redirect):
    pd.show_versions(as_json=True) # produces invalid json
pd_versions = redirect.getvalue()
pd_versions = json.loads(pd_versions) # raises JSONDecodeError
```

Problem description

`pd.show_versions(as_json=True)` is supposed to print valid JSON, but it really doesn't. It is supposed to work as advertised but it doesn't. It prints invalid JSON that is not parsable. The following error is printed when attempting to parse it:

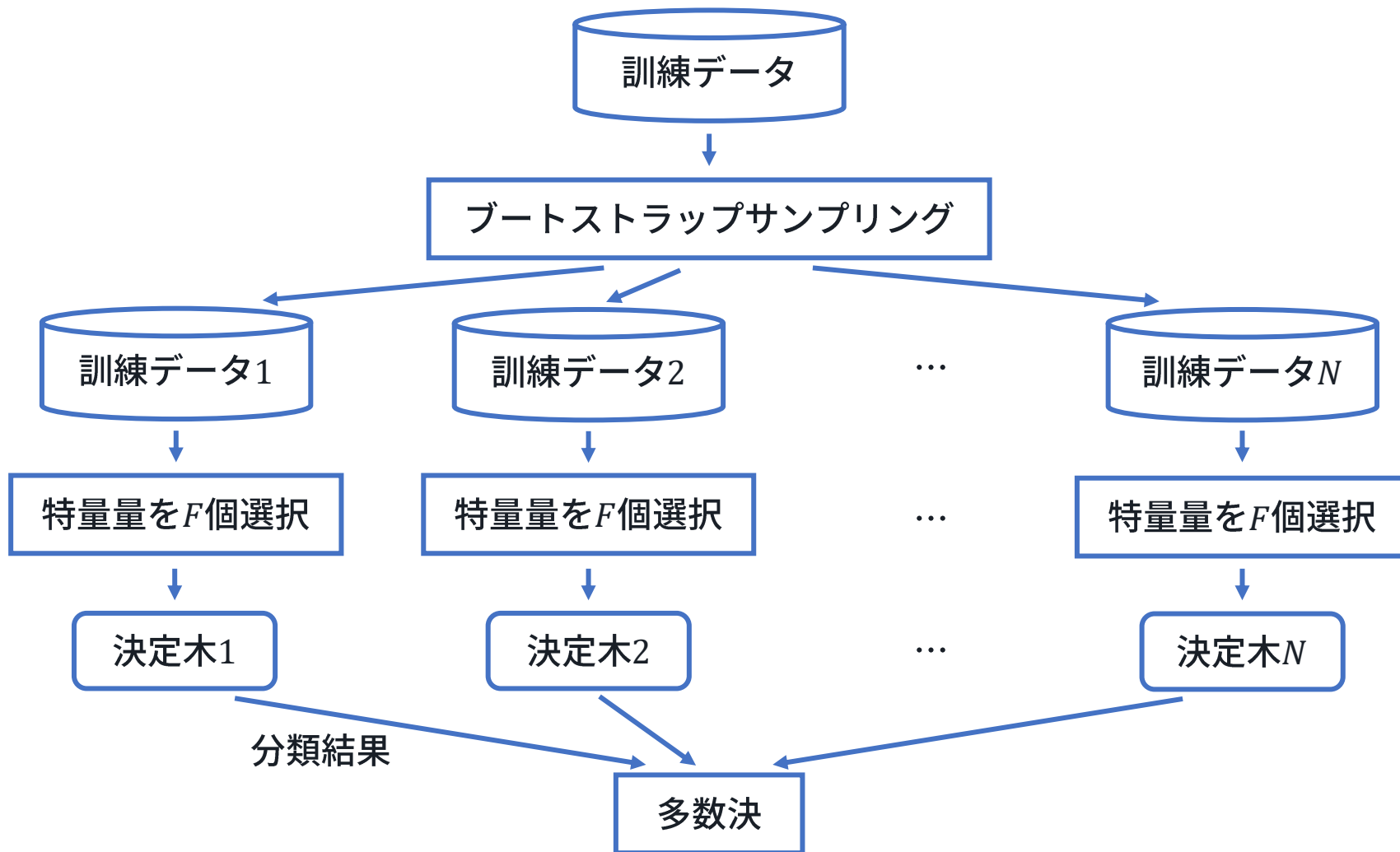
```
json.decoder.JSONDecodeError: Expecting property name enclosed in double quotes: line 1 column 2 (char 1)
```

This functionality is important because when running a job in the cloud, it is not feasible to print a multiline output. This is because every printed line has significant attached metadata. It is better to have valid JSON output instead in this case. The implementation of the `show_versions` function looks to be very hacky and poorly reviewed, given that it fails in this very basic way.

- コード片，インラインコード，ハイパーリンクなどの置換
- ストップワードや記号の除去，数字や単語の正規化など

4. 分類モデルの構築 (4 / 4) | 機械学習アルゴリズム (1 / 2)

ランダムフォレスト^[7]



[7] Leo Breiman, Random Forests, Machine Learning, 45, pp. 5-32, 2001

4. 分類モデルの構築 (4 / 4) | 機械学習アルゴリズム (2 / 2)

パラメータ

発案者らによって推奨されている値^[8]を利用 (今後チューニングを行う予定)

- ・ 決定木の個数 : $N = 200$
- ・ 特徴量の個数 : $F = \sqrt{M}$ M : すべての特徴量の個数

不均衡データへの対処

	データ件数[件]
通常Issue (負例)	142,651
GFI (正例)	8,849

正例，負例の偏りが大きいので，決定木のノード分割を行う際，**不純度** (ジニ不純度を利用) に対して，データ件数に反比例する重みづけ

[8] Hastie, Trevor; Tibshirani, Robert; Friedman, Jerome (2008). The Elements of Statistical Learning (2nd ed.). Springer. ISBN 0-387-95284-5.

4. 分類モデルの構築 (4 / 4) | モデルの学習・評価 (1 / 3)

入力データ

- 説明変数 (116,624次元)
 - タイトルのBag of Words (55,580次元)
 - 本文のBag of Words (61,041次元)
 - タイトルの文字数 (1次元)
 - 本文の文字数 (1次元)
 - 投稿者とリポジトリの関係 (1次元)
- 目的変数 (1次元)

GFI : 1, 通常Issue : 0

4. 分類モデルの構築 (4 / 4) | モデルの学習・評価 (2 / 3)

評価方法

データセットを9:1に分割し、9割を使って層化10分割交差検証、残り1割を評価用データとして、*Precision*, *Recall*を計算

混同行列

		分類結果	
		GFI	通常Issue
実際のクラス	GFI	<i>True Positive</i>	<i>False Negative</i>
	通常Issue	<i>False Positive</i>	<i>True Negative</i>

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \dots \text{GFIと分類したIssueが実際にGFIだった割合}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \dots \text{実際のGFIのうち、GFIに分類できた割合}$$

4. 分類モデルの構築 (4 / 4) | モデルの学習・評価 (3 / 3)

層化10分割交差検証の結果

	スコア
<i>Mean Precision</i>	0.91
<i>Mean Recall</i>	0.33

評価用データの分類結果

		分類結果				
		GFI	通常Issue			スコア
実際の クラス	GFI	302	628	→	<i>Precision</i>	0.91
	通常Issue	30	14,974		<i>Recall</i>	0.32

*Precision*は高いが，その分*Recall*は低い

GFIの中に通常Issueが多く紛れているより，見逃しがあっても
より高い確率でGFIであるIssueを分類できる方が重要

5. 今後の予定

- 誤分類の原因や，GFIと通常Issueを見分ける特徴を理解するため，分類モデルを解釈
 - 特徴量の重要度
 - モデルの簡略化^[9]
- GFIを運用していないプロジェクトへの分類モデル適用
 - メンテナへのアンケートで定性的評価

[9] Making Tree Ensembles Interpretable: A Bayesian Model Selection Approach, Satoshi Hara, Kohei Hayashi, Proceedings of the 21th International Conference on Artificial Intelligence and Statistics (AISTATS'18), p. 77-85, 2018