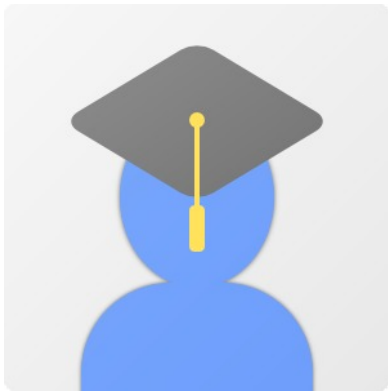


Using Dynamic and Contextual Features to Predict Issue Lifetime in GitHub Projects



Riivo Kikas



Marlon Dumas
University of Tartu



Dietmar Pfahl

MSR '16 Proceedings of the 13th International Conference
on Mining Software Repositories Pages 291-302

背景

- 多くのOSSホスティングサービスでITS(Issue Tracking System)が使われており, 誰でも不具合報告ができる
- 多くの不具合が報告されるが開発者の数には限りがあるため, オープンされたままの不具合も多い
- 不具合の存続期間を予測する研究がされてきている
 - 生存期間: 不具合がクローズされるまでの期間

先行研究

- 文脈的に似ている不具合から存続期間を予測[1]
- 動的特徴を取り入れた予測[2]
 - 動的特徴：ex)コメント数, アクション数
- 静的特徴と機械学習を用いた予測[3]

- [1] Weiss, C., Premraj, R., Zimmermann, T., and Zeller, A. How long will it take to fix this bug?, In Proceedings of the Fourth International Workshop on Mining Software Repositories (2007), MSR '07, IEEE Computer Society, p. 1.
- [2] Giger, E., Pinzger, M., and Gall, H. Predicting the fix time of bugs. In Proceedings of the 2Nd International Workshop on Recommendation Systems for Software Engineering (2010), RSSE '10, ACM, pp. 52–56.
- [3] Panjer, L. D. Predicting eclipse bug lifetimes. In Proceedings of the Fourth International Workshop on Mining Software Repositories (2007), MSR '07, IEEE Computer Society, p. 29.

先行研究の問題点と本研究の目的

- 先行研究の問題点
 - 多くの先行研究は対象プロジェクトが少ない
 - 動的特徴と静的特徴どちらもを用いた予測モデルの構築はされていない
- 本研究の目的
 - 多くのプロジェクトを対象にして, 動的・静的特徴を用いた予測モデルを作る

リサーチクエスチョン

- RQ1
 - 静的・動的・文脈的特徴を用いて、観測点Tから不具合存続期間を予測するモデルの精度はどれほどのものか？
- RQ2
 - 不具合存続期間予測にもっとも重要な特徴はどれか？

データセット

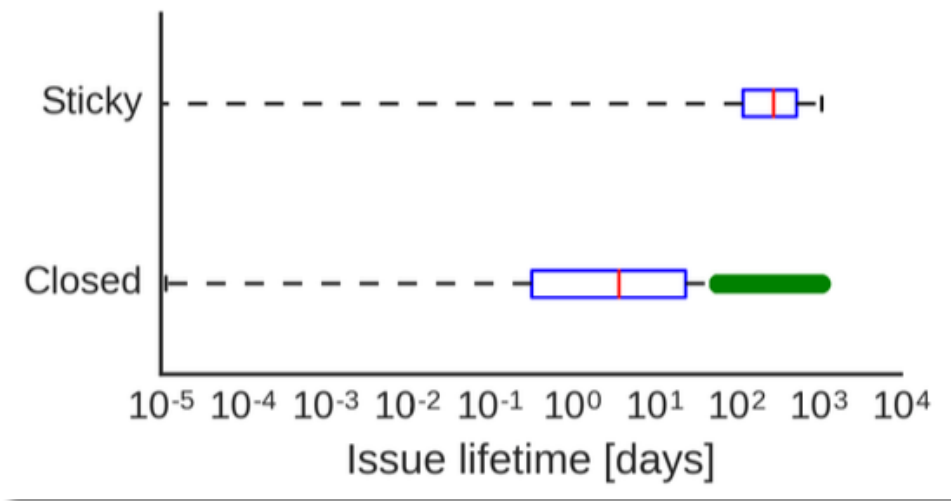
- 2015年4月1日付のGHTorrentのデータを用いる
- 対象期間
 - 2012年1月から2014年12月
- 対象プロジェクト
 - 全リポジトリは700万リポジトリ以上
 - フィルタリングを行い,4,024プロジェクトを対象

リポジトリのフィルタリング

- 対象リポジトリの条件
 - 2012年1月から2014年12月の間に作成されており、8ヶ月以上のヒストリーがあること
 - 2000件以上の不具合がオープンもしくはクローズされていること/1日に500件以上の不具合が報告されていること
 - 既存のプロジェクトのフォークではないこと
 - メインリポジトリに5回以上コミットされていること
 - リポジトリが作成される以前のアクティビティが記録されていないこと

対象とする不具合

- 不具合件数
 - 合計：967,037件
 - クローズされたもの：675,970件
 - クローズされていないもの：291,067件



Sticky : クローズされていない不具合
Closed: クローズされた不具合

モデルに用いた特徴

クラス	特徴	説明
Issue features	nCommentsT	時点Tにおけるコメント数
	nActorsT	時点Tまでにコメント, 参照, 購読した人の数
	nAssignmentsT	時点Tまでの割り当てイベント数
	nLabelsT	時点Tまでに追加されたラベル数
	nMentionedByT	時点Tまでに他の不具合から記述された回数
	nReferencedByT	時点Tまでにコミットメッセージに記述された回数
	nSubscribedByT	時点Tまでに不具合のアップデートを購読している人の数
	meanCommentSizeT	時点Tまでの平均コメント量
	issueCleanedBodyLen	タイトルと本文の長さ
	textScore	不具合のタイトルと中身の分類値

モデルに用いた特徴

クラスタ	特徴	説明
Issue submitter features	nIssuesByCreator	不具合投稿者が不具合がオープンする前3ヶ月間に発行した不具合の数
	nIssues ByCreatorClosed	不具合投稿者が不具合を発行する前の3ヶ月間でクローズした不具合の数
	nCommitsByCreator	不具合投稿者が不具合を発行する前の3ヶ月間でリポジトリにコミットした数
Participant's features	nCommitsByActorsT	不具合発行から時点Tまでの2週間で、リポジトリにコードをコミットした人の数
	nCommits ByUniqueActorsT	不具合発行から時点Tまでの2週間で、リポジトリにコードをコミットした一意の人の数

モデルに用いた特徴

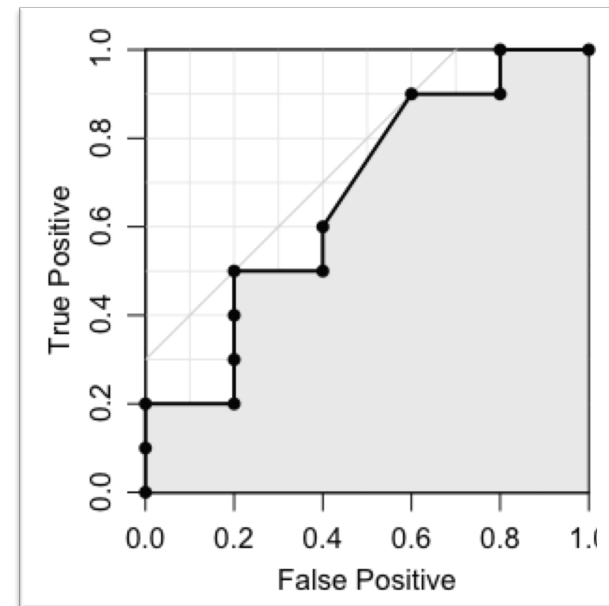
クラス	特徴	説明
Project features	nIssuesCreated InProject	不具合発行の前の3ヶ月間に生成された不具合の数
	nIssuesCreated InProjectClosed	不具合発行の前の3ヶ月間に生成されクローズした不具合の数
	nCommits InProject	不具合発行3ヶ月前のコミット数
	nIssuesCreatedProjectT	不具合発行前から時点Tまでの2週間で生成された不具合の数
	nIssuesCreated ProjectClosedT	不具合発行前から時点Tまでの2週間で生成されクローズされた不具合の数
	nCommitsProjectT	不具合発行前から時点Tまでの2週間のコミット数

予測モデルの構築

- ランダムフォレストを用いて予測
 - 一番精度が良かった
- 学習データ
 - 不具合の観測点
 - 不具合がオープンされてから経過した日数
 - 予測範囲
 - 不具合がクローズされた日数

評価方法

- クローズされている不具合についての予測精度を評価する
- 評価基準
 - Precision
 - Recall
 - F1値
 - AUROC
 - ROC : True positiveとfalse positiveの割合
 - AUC : 曲線以下の面積



結果：RQ1

Table 3: AUC scores for different Prediction horizon and observation point (OP) values.

OP	Prediction horizon(days)						
	1	7	14	30	90	180	365
0	0.653	0.660	0.663	0.665	0.666	0.681	0.707
1		0.641	0.644	0.649	0.658	0.680	0.688
7			0.639	0.644	0.653	0.680	0.671
14				0.646	0.653	0.681	0.665
30					0.653	0.681	0.659
90						0.687	0.661
180							0.684

AUCの値は全て0.64~0.707の値であり、
短期予測よりも長期予測の方がいい

結果：RQ1

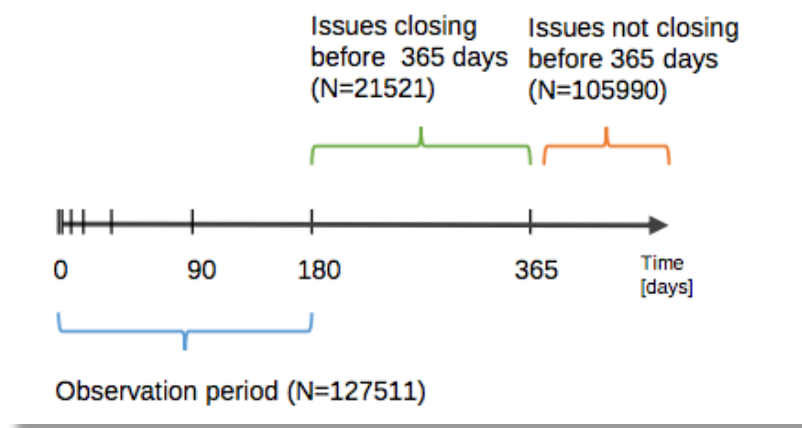
Table 4: F1 scores for different Prediction horizon and observation point (OP) values.

OP	Prediction horizon(days)						
	1	7	14	30	90	180	365
0	0.437	0.604	0.659	0.715	0.781	0.830	0.898
1		0.392	0.478	0.571	0.695	0.766	0.863
7			0.236	0.402	0.587	0.702	0.806
14				0.278	0.513	0.657	0.771
30					0.395	0.574	0.712
90						0.337	0.589
180							0.387

観測点(OP)が早いとF1値は高い
長期予測には適している

追加実験

- 前に述べた二つの結果は，観測点と予測範囲の組み合わせのサンプル数が異なるため単純に比較することはできない



- そこで，テストセットの数を揃えて追加実験を行う

結果：RQ1

Observation point	AUC	Precision	Recall	F1	TP	FP	FN	TN
Prediction horizon of 365 days (N=127511)								
0	0.629	0.187	0.851	0.306	18311	79679	3210	26311
1	0.631	0.185	0.905	0.308	19484	85702	2037	20288
7	0.629	0.188	0.914	0.311	19674	85136	1847	20854
14	0.635	0.195	0.921	0.322	19811	81582	1710	24408
30	0.637	0.204	0.912	0.334	19635	76484	1886	29506
90	0.654	0.244	0.875	0.381	18831	58433	2690	47557
180	0.684	0.249	0.874	0.387	18802	56839	2719	49151
Prediction horizon of 180 days (N=196976)								
0	0.618	0.181	0.814	0.297	24757	111782	5661	54776
1	0.633	0.206	0.783	0.326	23830	91756	6588	74802
7	0.656	0.208	0.814	0.331	24767	94540	5651	72018
14	0.664	0.207	0.826	0.331	25117	96029	5301	70529
30	0.675	0.208	0.836	0.333	25416	97003	5002	69555
90	0.687	0.211	0.834	0.337	25374	94884	5044	71674
Prediction horizon of 90 days (N=280862)								
0	0.615	0.251	0.793	0.381	47477	141823	12375	79187
1	0.616	0.263	0.746	0.389	44643	124887	15209	96123
7	0.630	0.268	0.700	0.388	41912	114349	17940	106661
14	0.639	0.268	0.682	0.385	40801	111222	19051	109788
30	0.653	0.272	0.721	0.395	43179	115474	16673	105536

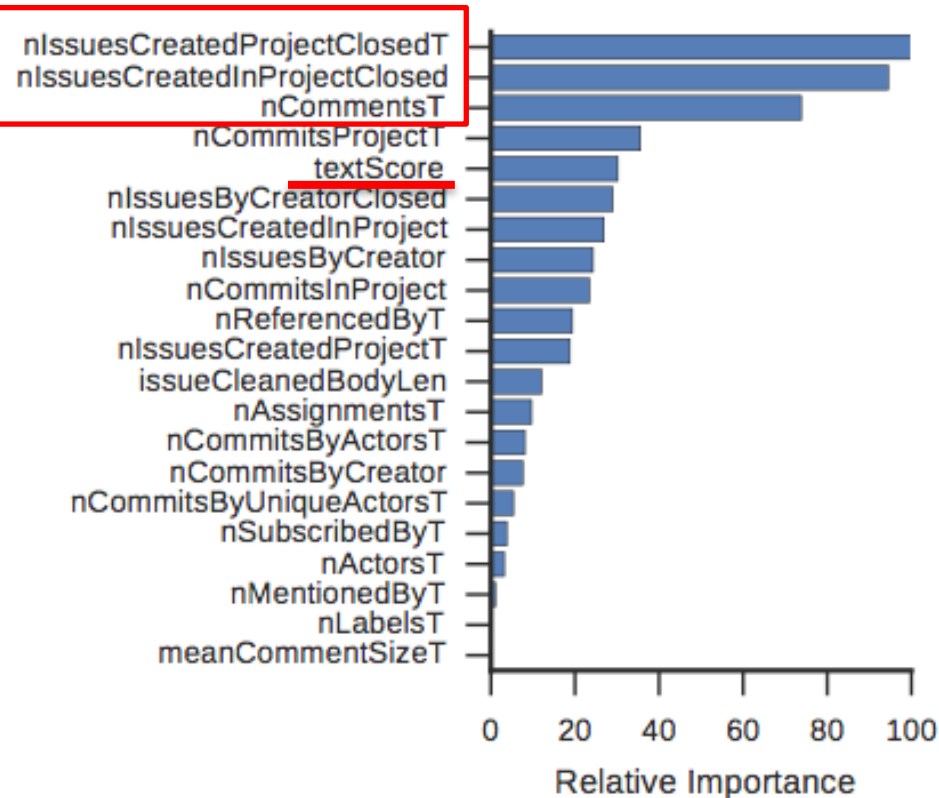
結果：RQ1

Observation point	AUC	Precision	Recall	F1	TP	FP	FN	TN
Prediction horizon of 365 days (N=127511)								
0	0.629	0.187	0.851	0.306	18311	79679	3210	26311
1	0.631	0.185	0.905	0.308	19484	85702	2037	20288
7	0.629	0.188	0.914	0.311	19674	85136	1847	20854
14	0.635	0.195	0.921	0.322	19811	81582	1710	24408

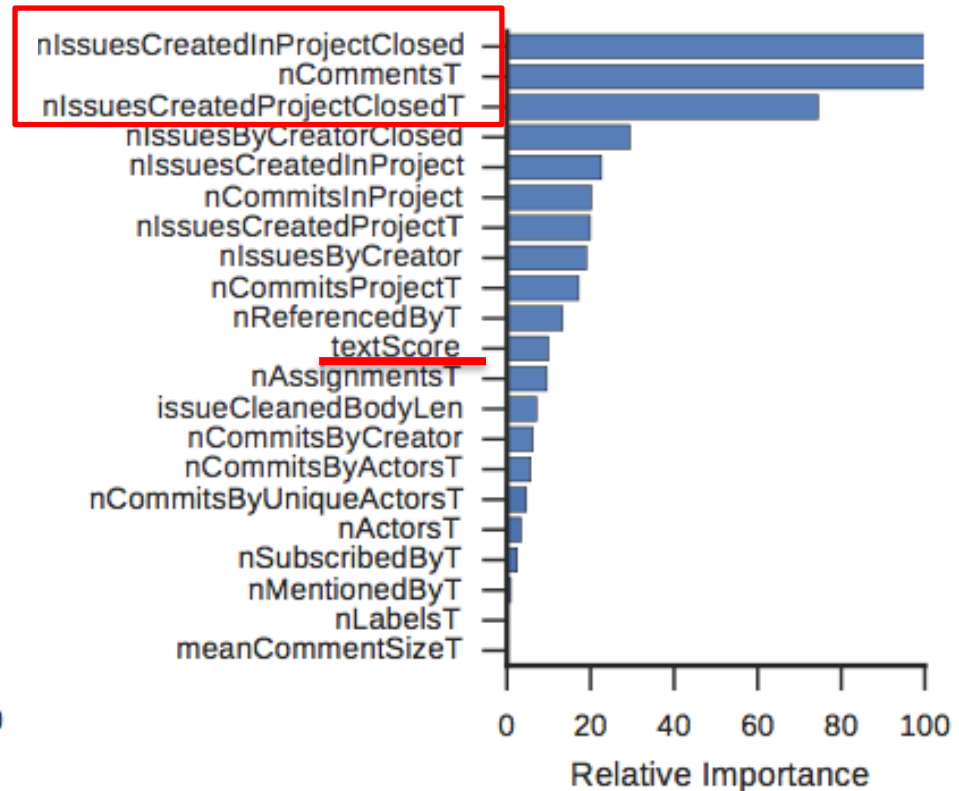
観測点が遅くなると、AUCとPrecisionが増加する
しかし、Recallは減少

30	0.675	0.208	0.836	0.333	25416	97003	5002	69555
90	0.687	0.211	0.834	0.337	25374	94884	5044	71674
Prediction horizon of 90 days (N=280862)								
0	0.615	0.251	0.793	0.381	47477	141823	12375	79187
1	0.616	0.263	0.746	0.389	44643	124887	15209	96123
7	0.630	0.268	0.700	0.388	41912	114349	17940	106661
14	0.639	0.268	0.682	0.385	40801	111222	19051	109788
30	0.653	0.272	0.721	0.395	43179	115474	16673	105536

結果：RQ2



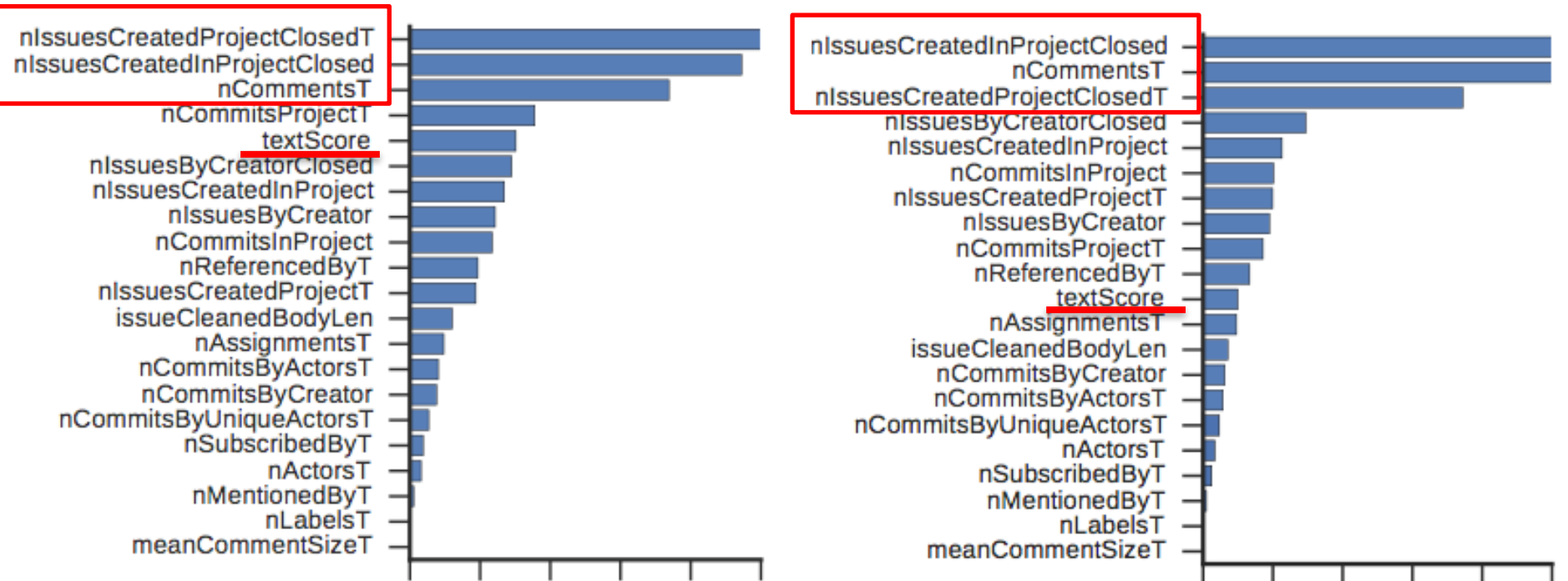
(a) 30-days horizon



(b) 180-days horizon

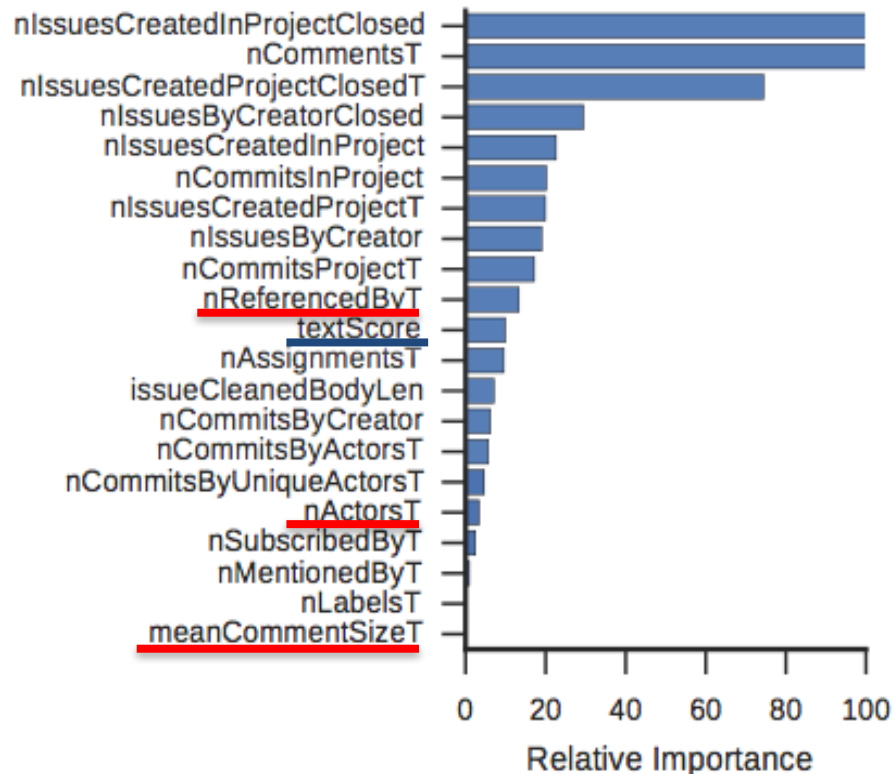
0 day model

結果：RQ2

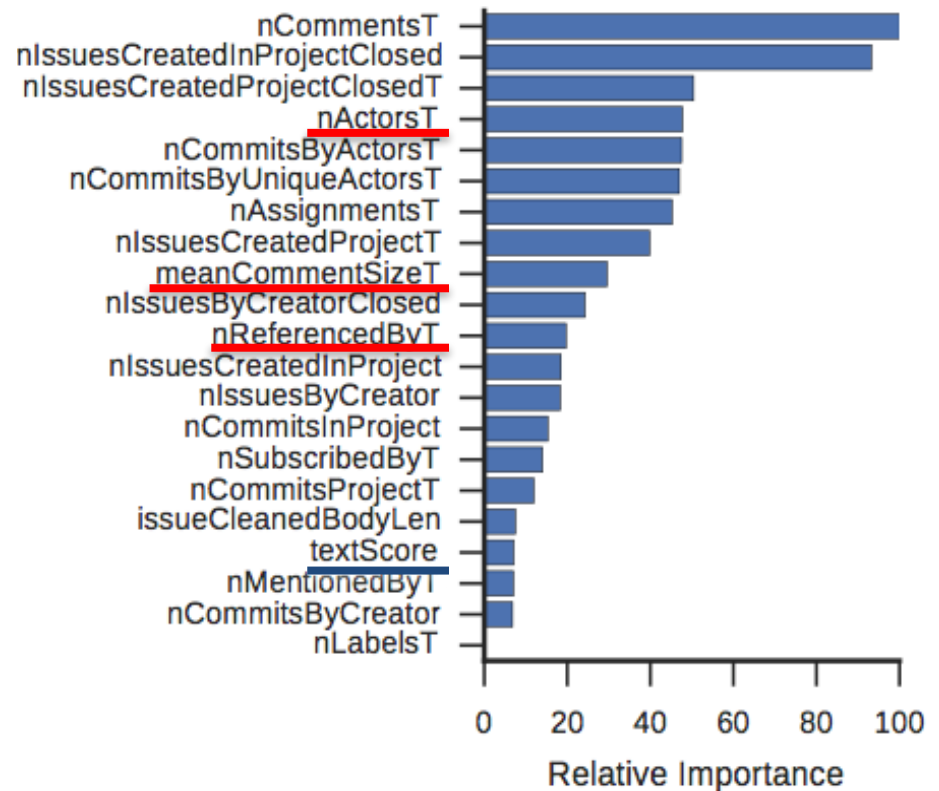


観測点が遅くなるとtextScoreの重要性が下がる
テキストは短期間の予測には重要である

結果：RQ2



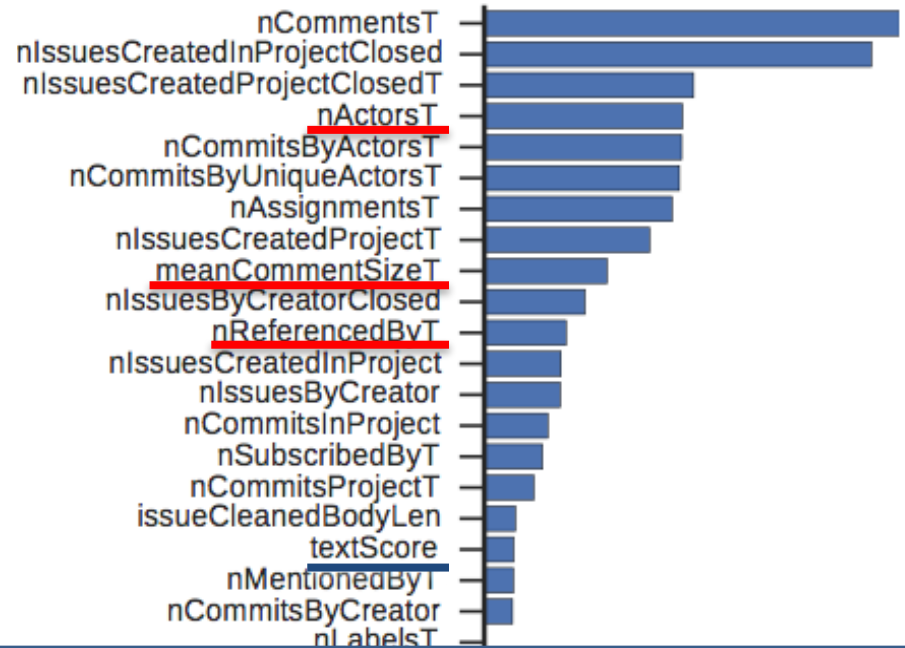
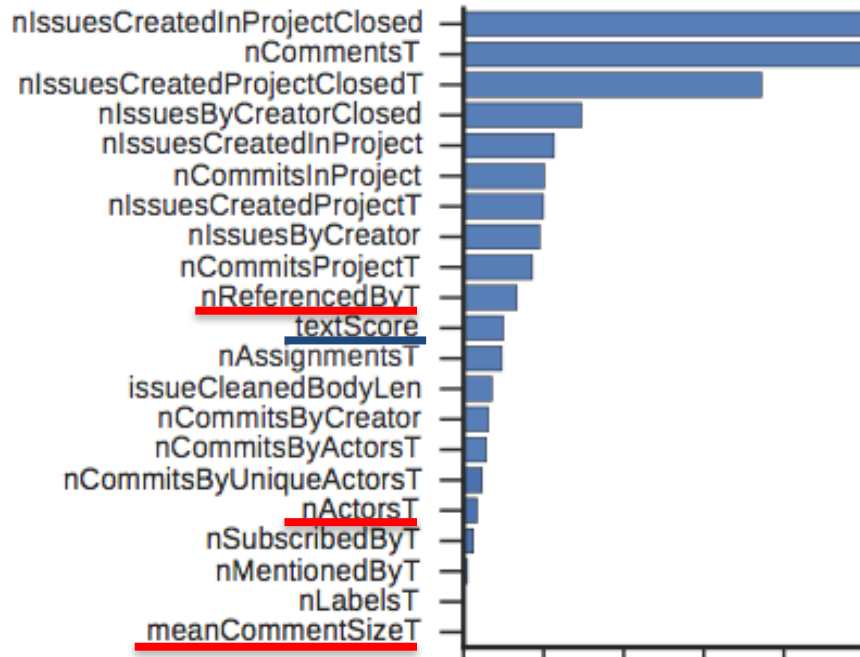
(b) 180-days horizon



(a) 180-days horizon

7 day model

結果：RQ2



textScoreの重要性が減少し動的特徴の重要性が増加
長期予測において動的特徴が静的特量を補っている

妥当性の脅威

- 4,000個以上のプロジェクトを対象にしたが、プロジェクトごとの特色を考慮していないため結果の一般性がない
- 交差検証を行っていない

まとめと所感

- まとめ
 - Githubプロジェクトを対象に、動的特徴と文脈的特徴を用いた不具合存続期間予測モデルを構築した
 - 動的特徴と文脈的特徴は重要である
 - プロジェクトのタイプに適した予測モデルを作る必要がある
- 所感
 - とりあえず、全部まとめてやってみた感
 - Precisionそれでいいのかな？