Classification Models in Software Engineering: From Defect Prediction to Best-Answer Prediction

Fabio Calefato
University of Bari

University of Victoria, 24 Nov. 2016

Community-based Q&A

- Devs more and more seek technical support from experts other than teammates
  - Before: mailing lists and web forums
  - Now: question-and-answer sites
- Benefits
  - Often answered within minutes
  - Gamification leverages community participation
  - Skills acknowledgment

A shift in Q&A sites purpose

Platforms originally aimed at providing quick solutions to the information seeker

Short-term value, mostly for the or original asker

Platforms supporting the process of **community-driven knowledge** creation

Long-term value, for a broader audience

Anderson et al., Discovering Value from Community Activity on Focused Question Answering Sites: A Case Study of Stack Overflow, KDD 2012

Technical Q&A sites

- Important for SE from both professional and educational perspective
- Stack Overflow has ~40M visits per month [1]
  - 16M from professional developers
  - 70% report to be self-taught devs
- Developers read manuals less and less, they rather “search” [2]
  - E.g., SO covers ~87% of Android API [3]
  - E.g., API augmented with contextual insights from SO [4]

[1] http://stackoverflow.com/research/developer-survey-2016
[2] M. Shaw, Progress Toward an Engineering Discipline for Software, ICSE 2016 Keynote
[3] C. Parnin et al., Crowd documentation: Exploring the coverage and the dynamics of API discussions on Stack Overflow, Georgia IT, Tech Report 2012
[4] C. Treude and P. Robillard, Augmenting API Documentation with Insights from Stack Overflow, ICSE 2016

Sentiment Analysis in Software Engineering

CT&W '11 (2011)
Do moods affect programmers' debug performance? (Kahn et al)

ESEC/FSE '13 (2013)
Towards emotional awareness in software development teams. (Guzman and Bruegge)

Peer J '13 (2013)
Happy software developers solve problems better Peer J (Graziotin et al.)

RE'14 (2014)
How Do Users Like this Feature? A Fine Grained Sentiment Analysis of App Reviews (Guzman and Maalej)

MSR '14 (2014)
- Sentiment analysis of commit comments in GitHub: An empirical study (Guzman et al.)
- Security and emotion: sentiment analysis of security discussions on GitHub (Pletea et al.)
- Do developers feel emotions? (Murgia et al.)

CHASE '15 (2015)
- Exploring Causes of Frustration for Software Developers. (Ford and Parnin)
- Towards emotion-based collaborative software engineering (Dewan)

ICSE'15 (2015)
Stuck and Frustrated or In Flow and Happy: Sensing Developers' Emotions and Progress. (Muller and Fritz)

MSR'15 (2015)
Mining Successful Answers in Stack Overflow (Calefato et al.)

XP'15 (2015)
Would you mind fixing this issue? (Ortu et al.)

EmoQuest: Investigating the Role of Emotions in the Social Programmer Ecosystem

- RQ: **getting emotional** while communicating with developers: **good or bad?**
- Model: combining message properties, social factors, and affective factors
- Output:
  - Evidence-based netiquette
  - SE-specific sentiment analysis tool and emotion classifier

Towards Discovering the Role of Emotions in Stack Overflow

Nicole Novielli, Fabio Calefato, Filippo Lanubile
University
Dipartimento d
Bari, I
{nicole.novielli, fabio.calefato

ABSTRACT

Today, people increasingly try to solve domain-specific problems through interaction on online Question and Answer (Q&A) sites, such as Stack Overflow. The growing success of the Stack Overflow community largely depends on the will of their members to answer others' questions. Recent research has shown that the factors that push members of online communities encompass both social and technical aspects. Yet, we argue that also the emotional style of a technical question does influence the probability of promptly obtaining a satisfying answer.

In this paper, we describe the design of an empirical study aimed to investigate the role of affective lexicon on the questions posted in Stack Overflow.

Categories and Subject Descriptors
H.1.2 **[User/Machine Systems]**: Human factors

General Terms
Design, Human Factors.

Keywords
Online Q&A, Technical Forum, Sentiment Analysis, Experimental Design, Stack Overflow.

1. INTRODUCTION

The worldwide diffusion of social media has profoundly changed the way we communicate and access information. Increasingly, people try to solve domain-specific problems through interaction on online Question and Answer (Q&A) sites. The enormous success of Stack Overflow (SO), a community of over 3 million programmers asking questions (~7 millions) and providing answers (~13 millions) about software development, attests this increasing trend. Launched in 2008, Stack Overflow is now part of Stack Exchange, a fast growing network of more than 100 Q&A sites about a broad range of topics, from academic life to traveling and gaming, which originated from the success of Stack Overflow itself.

The growing success of Stack Exchange communities largely depends on the will of their members to answer others' questions.

Mining Successful Answers in Stack Overflow

Fabio Calefato, Filippo Lanubile, Mari
Dipartimento di Informatica, Unive
{fabio.calefato, filippo.lanubile, nicole.novielli

***Abstract*—** Recent research has shown that drivers of success in online question answering encompass presentation quality as well as temporal and social aspects. Yet, we argue that also the emotional style of a technical contribution influences its perceived quality. In this paper, we investigate how Stack Overflow users can increase the chance of getting their answer accepted. We focus on actionable factors that can be acted upon by users when writing an answer and making comments. We found evidence that factors related to information presentation, time and affect all have an impact on the success of answers.

Index Terms* — *Online Q&A, Sentiment Analysis, Knowledge Sharing, Human Factors.

I. Introduction

The enormous success of Stack Overflow (SO) provides data scientists with a huge amount of data about online question answering (QA). Our investigation aims to provide guidelines for writing high-quality contributions and inform the design of tools that support effective knowledge sharing. In this paper, we investigate how an information provider can increase the chance of getting his answer accepted in SO. In particular, we focus on actionable factors that can be acted upon by community members when contributing to answering a question. Hence, our first research question is formulated as follows:

RQ1 – Which actionable factors predict the success of a SO answer?

Social and temporal aspects are among the success factors of an answer [1][4], depending on the answerers' level of expertise and their engagement in the community. More recently, research has begun to investigate linguistic factors too, looking at how answers are formulated [5][7]. In addition, we argue that the path to effective question answering and reputation building passes through emotions too. There is an increasing attention to the impact of emotional awareness on effective collaboration [5][8]. However, existing research on online QA sites has not taken into full consideration the potential contributions from the field of affective computing, with the only notable exception of a large-scale sentiment analysis study on Yahoo! Answers [9]. Therefore, we formulate our second research questions:

RQ2 – Do affective factors influence the success of a SO answer?

While previous research has mostly focused on time, reputation and presentation quality, our study is the first one to investigate the impact of affective factors on the success of answers in SO. This study is part of our ongoing research on investigating the role of emotions in community-based QA,

Success Factors for Effective Knowledge Sharing in Community-based Question-Answering

Fabio Calefato, Filippo Lanubile, Ma
Merolla, Nicole Novielli

Dipartimento di Informatica
Università degli Studi di Bari 'Aldo Moro'
via E. Orabona, 4 – 70125 Bari
{fabio.calefato,filippo.lanubile,nicole.novielli}@uniba.it
m.merolla1@studenti.uniba.it

Structured Abstract

Purpose – Nowadays, people increasingly seek information
and Answer (Q&A) sites. The enormous success of Sta
growing network of Q&A sites, attests this increasing trend.
depends on the will of their members to provide good
questions. We investigate the success factors of Q&A tha
effective knowledge creation and sharing. In particular, we f
acted upon by contributors when writing a question.

Design/methodology/approach – Based on literature in
empirical model of the factors that predict the chance of g
asking a question on a Q&A site. The actionable factors in
three categories of features: *Presentation Quality*, *Time*, and
logistic regression framework for estimating the probabili
based on our set of predictors, that is the metrics that op
presentation quality. Stack Exchange makes user-contribu
under Creative Commons license, which we use in our empir

Originality/value – Previous research shows how the succ
its presentation quality (Treude *et al*. 2011, Asaduzzaman
time in which it is posted (Bosu *et al*. 2013), and on the ask
2014). The influence of affective factors is less evident. Ho
to effective question answering also involves consideration
2014). Our ongoing research aims at filling this gap in liter
the role of affect in Stack Exchange.

Practical implications – The expected output of this ong
driven netiquette for online Q&A sites. It will shed new ligh
facilitates or impairs effective knowledge sharing, leadin
emotional awareness computer-mediated interactions. In de

[1] http://stackexchange.com/
[2] https://archive.org/details/stackexchange

The Challenges of Sentiment Detection in the Social Programmer Ecosystem

Nicole Novielli, Fabio Calefato, Filippo Lanubile
University of Bari
Dipartimento di Informatica
Bari, Italy
{nicole.novielli, fabio.calefato, filippo.lanubile}@uniba.it

ABSTRACT

A recent research trend has emerged to study the role of affect in in the social programmer ecosystem, by applying sentiment analysis to the content available in sites such as GitHub and Stack Overflow. In this paper, we aim at assessing the suitability of a state-of-the-art sentiment analysis tool, already applied in social computing, for detecting affective expressions in Stack Overflow. We also aim at verifying the construct validity of choosing sentiment polarity and strength as an appropriate way to operationalize affective states in empirical studies on Stack Overflow. Finally, we underline the need to overcome the limitations induced by domain-dependent use of lexicon that may produce unreliable results.

Categories and Subject Descriptors
H.1.2 [**User/Machine Systems**]: Human factors

General Terms
Human Factors.

Keywords
Online Q&A, Technical Forum, Sentiment Analysis, Stack Overflow, Social Programmer, Social Software Engineering

1. INTRODUCTION

Software engineering involves a large amount of social interaction, as programmers often need to cooperate with others, whether directly or indirectly. However, we have become fully aware of the importance of social aspects in software engineering activities only over the last decade. In fact, it was not until the recent diffusion and massive adoption of social media that we could witness the rise of the "social programmer" [41] and the surrounding ecosystem [42].

Social media has deeply influenced the design of software development-oriented tools such as GitHub (i.e., a social coding site) and Stack Overflow (i.e., a community-based question answering site) [43]. Stack Overflow, in particular, is an example of an online community where social programmers do networking by reading and answering others' questions, thus participating in the creation and diffusion of crowdsourced documentation. In our previous work, we argued and proved that among the non-technical factors, which can influence the members of online communities, the emotional style of a technical contribution does affect its probability of success [29], [9]. More specifically, our effort is to understand how expressing affective states in Stack Overflow influences the probability for askers of eliciting an accepted answer and the probability for answerers of having an answer accepted.

Our research follows a recent trend that has emerged to study the role of affect in social computing. For example, Kucuktunc et al. [19] performed a large-scale sentiment analysis study on Yahoo! Answers to assess the impact of the semantic orientation of a post on its perceived quality. Althoff et al. [1] found that expressing gratitude in a question is positively correlated with success of altruistic requests in Reddit.com. Guzman et al. [17] perform sentiment analysis of commit comments in GitHub and demonstrate that a correlation exists between emotions and other factors such as the programming language used in a project, the geographical distribution of the team and the day of the week. Similarly, Guzman and Bruegge [16] used a sentiment analysis tools for detecting the polarity, i.e., the positive or negative semantic orientation of a text, to investigate the role of emotional awareness in software development teams.

What these studies have in common is that they applied sentiment analysis techniques to crowd-generated content relying on polarity as the only dimension to operationalize affect. However, polarity is only one of the possible dimensions of affect, which could be also modeled in terms of its duration, activation, cognitive triggers, and specificity [11]. Still, polarity is the most used dimension because of its ease of measurement and the availability of open source and robust analysis tools. In this paper, we argue that polarity, if employed alone, is insufficient for detecting the sentiments of programmers in a reliable manner. Furthermore, we highlight and discuss the challenges existing when sentiment analysis techniques are employed to assess the affective load of text containing technical lexicon, as typical in the social programmer ecosystem.

The remainder of the paper is structured as follows. In Section 2, we first provide an overview of detecting affective states from text, including a state-of-the-art in the field of sentiment analysis. Then, in Section 3, we perform a qualitative analysis to show the limits of only using polarity to measure the sentiment expressed in questions and answers in Stack Overflow. The findings from our analysis are then discussed in Section 4, where we also outline the future research directions.

collab.di.uniba.it/fabio

SO problems (

Hi @user12345 if this or any answer has solved your question please consider accepting it by clicking the check-mark. This indicates to the wider community that you've found a solution and gives some reputation to both the answerer and yourself. There is no obligation to do this.

- Despite its popularity (12.6M questions)
  - About 50% are still unresolved questions (5.7M)
  - ~4M unresolved questions have 1+ unaccepted answers
    - Newbie askers not taking actions
    - No perfect solutions

83

You could also try changing your build directory for your project since that is where most of the path issues will arise. In your root build.gradle file

```
allprojects {
    buildDir = "C:/tmp/${rootProject.name}/${project.name}"
    repositories {
       ...
    }
}
```

Android Studio will pick up on the change and still show your new build location in the Project view. It's a lot easier than moving your entire project.

share improve this answer

answered Jan 8 at 15:10
lodlock
1,646 ● 7 ● 12

5 Best solution of all !!! Worked for me. Only changes the build directory, no need to move the entire project. – Nigel Crasto Feb 23 at 7:24

4 This should be the accepted solution, works great and has no impact on the project itself. – Bruno Coelho Apr 11 at 10:22

2 Genius! I had the same problem happen out of the blue after updating my gradle. (Google play services uses

19 answers
The most appreciated not the accepted one

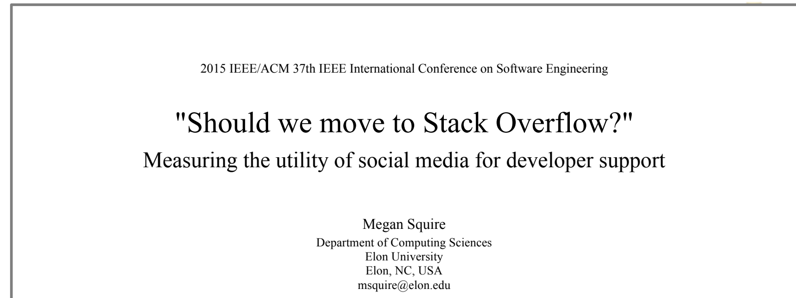
* As of Oct. 2016

Approaching the problem: Best-answer prediction

- Binary (two-class) classification problem of identifying accepted answers (solutions) within question threads
  - Leverage machine learning to build a *best-answer prediction* model
    - Positive class = {accepted answers}
    - Negative class = {non-accepted answers}
- Potential benefits
  - Identify most promising answers in unresolved threads
  - Ensure crowdsourced knowledge is well-curated

SO problems (2/2)

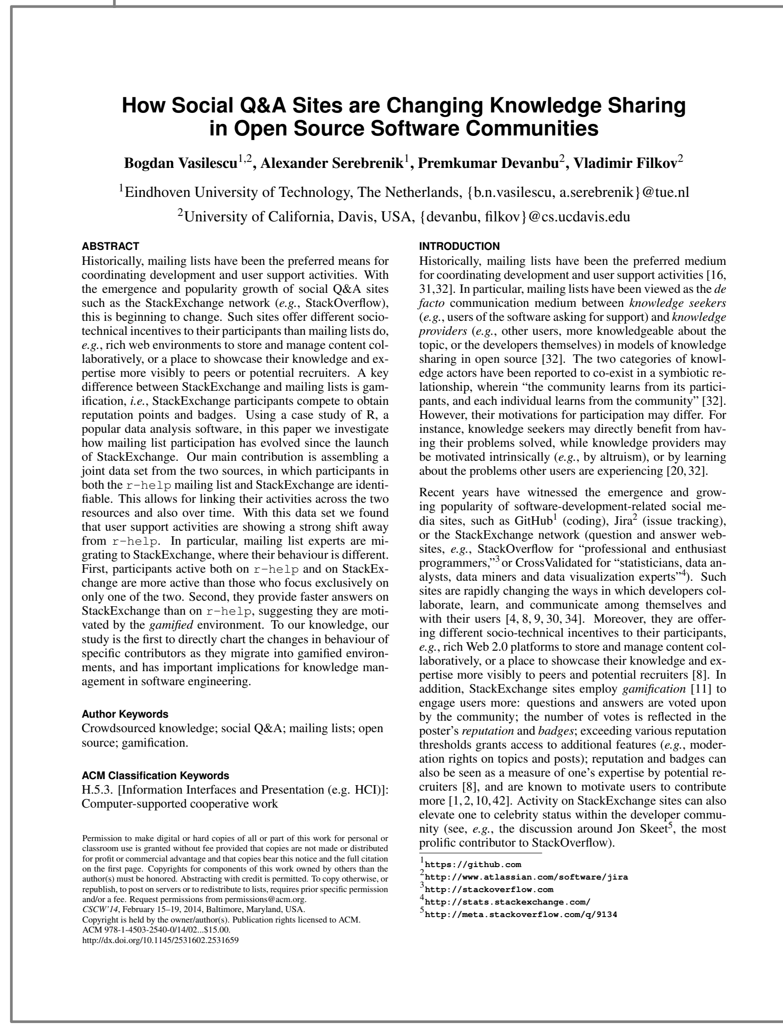
- Popularity side effects
  - Communities abandoning support forums and mailing list over Stack Overflow (e.g., R)
  - Huge amount of crowdsourced knowledge getting lost



2015 IEEE/ACM 37th IEEE International Conference on Software Engineering

"Should we move to Stack Overflow?"
Measuring the utility of social media for developer support

Megan Squire
Department of Computing Sciences
Elon University
Elon, NC, USA
msquire@elon.edu



How Social Q&A Sites are Changing Knowledge Sharing in Open Source Software Communities

Bogdan Vasilescu[1,2], Alexander Serebrenik[1], Premkumar Devanbu[2], Vladimir Filkov[2]

[1]Eindhoven University of Technology, The Netherlands, {b.n.vasilescu, a.serebrenik}@tue.nl
[2]University of California, Davis, USA, {devanbu, filkov}@cs.ucdavis.edu

ABSTRACT
Historically, mailing lists have been the preferred means for coordinating development and user support activities. With the emergence and popularity growth of social Q&A sites such as the StackExchange network (*e.g.*, StackOverflow), this is beginning to change. Such sites offer different socio-technical incentives to their participants than mailing lists do, *e.g.*, rich web environments to store and manage content collaboratively, or a place to showcase their knowledge and expertise more visibly to peers or potential recruiters. A key difference between StackExchange and mailing lists is gamification, *i.e.*, StackExchange participants compete to obtain reputation points and badges. Using a case study of R, a popular data analysis software, in this paper we investigate how mailing list participation has evolved since the launch of StackExchange. Our main contribution is assembling a joint data set from the two sources, in which participants in both the `r-help` mailing list and StackExchange are identifiable. This allows for linking their activities across the two resources and also over time. With this data set we found that user support activities are showing a strong shift away from `r-help`. In particular, mailing list experts are migrating to StackExchange, where their behaviour is different. First, participants active both on `r-help` and on StackExchange are more active than those who focus exclusively on only one of the two. Second, they provide faster answers on StackExchange than on `r-help`, suggesting they are motivated by the *gamified* environment. To our knowledge, our study is the first to directly chart the changes in behaviour of specific contributors as they migrate into gamified environments, and has important implications for knowledge management in software engineering.

Author Keywords
Crowdsourced knowledge; social Q&A; mailing lists; open source; gamification.

ACM Classification Keywords
H.5.3. [Information Interfaces and Presentation (e.g. HCI)]: Computer-supported cooperative work

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CSCW'14, February 15–19, 2014, Baltimore, Maryland, USA.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2540-0/14/02...$15.00.
http://dx.doi.org/10.1145/2531602.2531659

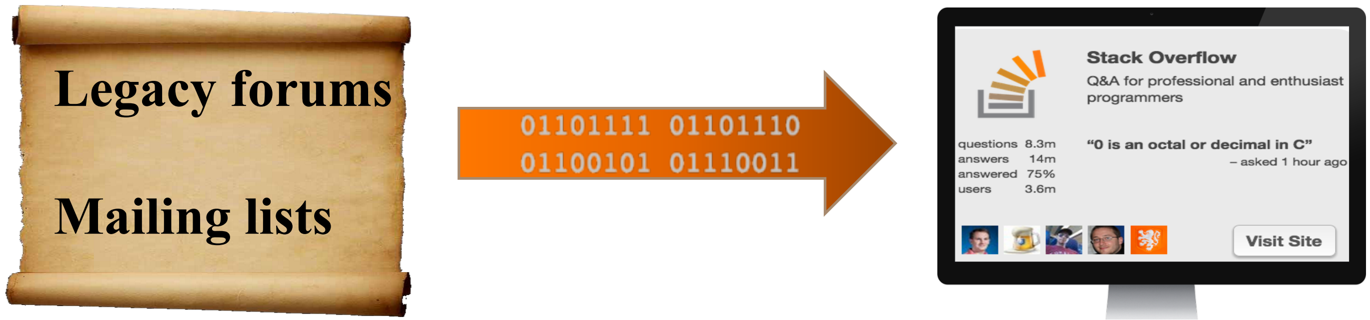
INTRODUCTION
Historically, mailing lists have been the preferred medium for coordinating development and user support activities [16, 31, 32]. In particular, mailing lists have been viewed as the *de facto* communication medium between *knowledge seekers* (*e.g.*, users of the software asking for support) and *knowledge providers* (*e.g.*, other users, more knowledgeable about the topic, or the developers themselves) in models of knowledge sharing in open source [32]. The two categories of knowledge actors have been reported to co-exist in a symbiotic relationship, wherein "the community learns from its participants, and each individual learns from the community" [32]. However, their motivations for participation may differ. For instance, knowledge seekers may directly benefit from having their problems solved, while knowledge providers may be motivated intrinsically (*e.g.*, by altruism), or by learning about the problems other users are experiencing [20, 32].

Recent years have witnessed the emergence and growing popularity of software-development-related social media sites, such as GitHub[1] (coding), Jira[2] (issue tracking), or the StackExchange network (question and answer websites, *e.g.*, StackOverflow for "professional and enthusiast programmers,"[3] or CrossValidated for "statisticians, data analysts, data miners and data visualization experts"[4]). Such sites are rapidly changing the ways in which developers collaborate, learn, and communicate among themselves and with their users [4, 8, 9, 30, 34]. Moreover, they are offering different socio-technical incentives to their participants, *e.g.*, rich Web 2.0 platforms to store and manage content collaboratively, or a place to showcase their knowledge and expertise more visibly to peers and potential recruiters [8]. In addition, StackExchange sites employ *gamification* [11] to engage users more: questions and answers are voted upon by the community; the number of votes is reflected in the poster's *reputation* and *badges*; exceeding various reputation thresholds grants access to additional features (*e.g.*, moderation rights on topics and posts); reputation and badges can also be seen as a measure of one's expertise by potential recruiters [8], and are known to motivate users to contribute more [1, 2, 10, 42]. Activity on StackExchange sites can also elevate one to celebrity status within the developer community (see, *e.g.*, the discussion around Jon Skeet[5], the most prolific contributor to StackOverflow).

[1]https://github.com
[2]http://www.atlassian.com/software/jira
[3]http://stackoverflow.com
[4]http://stats.stackexchange.com/
[5]http://meta.stackoverflow.com/q/9134

[1] M Squire, Should we move to stack overflow? ICSE '15
[2] B. Vasilescu et al. How Social Q&A Sites Are Changing Knowledge Sharing in Open Source Software Communities, CSCW '14

Best-answer prediction in legacy forums

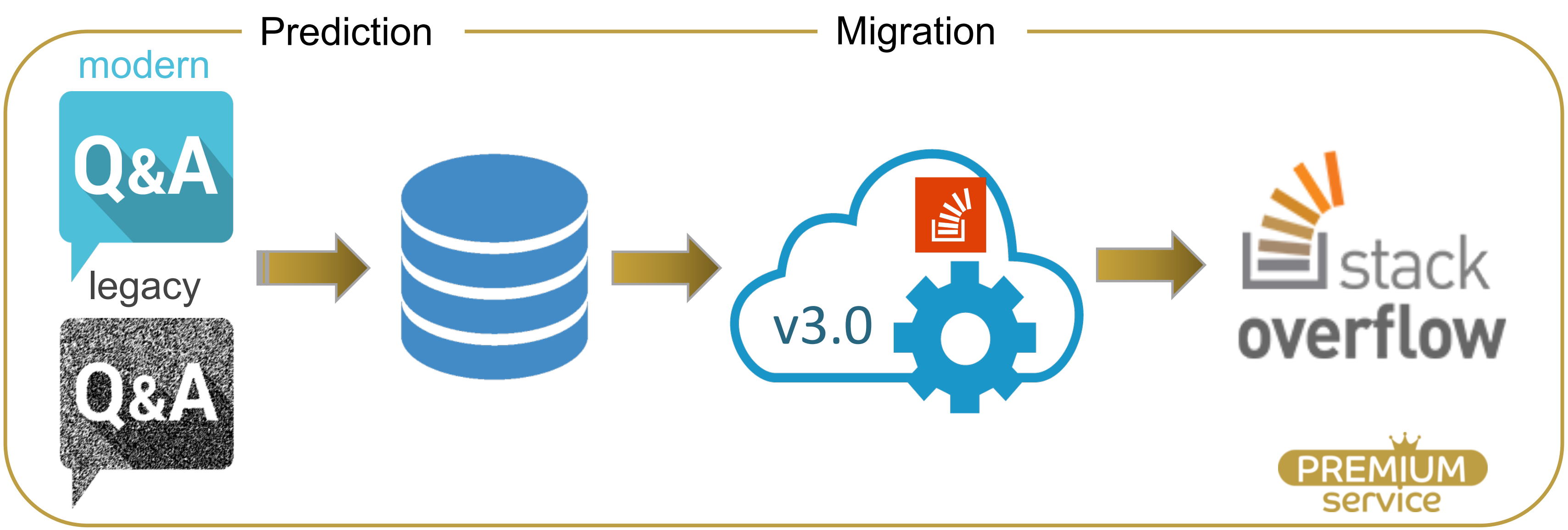


- Can we automatically migrate legacy support channels towards modern Q&A sites?
- Research Challenges:
  - Different interaction styles
  - Quality of imported content
  - Existing user reputation and identities [1]
  - Lack of info about accepted answers / resolved questions
- Potential benefit
  - Save existing crowdsourced knowledge from being lost upon migrations

[1] B. Vasilescu et al. How Social Q&A Sites Are Changing Knowledge Sharing in Open Source Software Communities, CSCW '14, 342–354.

Practical perspective

- Migration from internal legacy forum to modern Q&A site



http://stackapps.com/questions/1999/announcing-api-version-1-1-and-roadmap

Study inception

- Best-answer prediction relatively new problem
  - Limited amount of existing research on building prediction models
- Let’s do like machine learners do!!
  - Let’s use the experience from a more mature Sw. Eng. research field on building prediction models
  - Software Defect Prediction

A step back

BINARY CLASSIFICATION: CHALLENGES AND METRICS

Software Defect Prediction (SDP)

- Disproportionate amount of the cost of developing software spent on maintenance
  - Some industrial surveys claim 90%!
  - Bugs must be found before they can be fixed!
- Use machine learners to build prediction models and identify most defect-prone code
  - Use historical data about known bugs to train the model
  - Fit the defect prediction model to new, unseen code

L. Erlikh, “Leveraging legacy system dollars for e-business,” IT Professional, vol. 2, pp. 17–23, 2000.

SDP research

- Substantial amount published in the last two decades
- Main drivers
  - Economic benefits, especially for the Quality Assurance team [1]
    - Limited testing resource allocated for the most fault-prone code
    - Much more cost-effective than traditional code reviews
  - Availability of public datasets [2]
    - NASA, Eclipse, PROMISE
    - OSS repositories (e.g., APACHE)

[1] Menzies et al., Defect prediction from static code features: current results, limitations, new approaches, Automated Software Engineering 2010
[2] R. Malhotra, A systematic review of machine learning techniques for software fault prediction, Applied Soft Computing 2015

Classification techniques

Technique	Classifier
Regression-based	Logistic Regression
Bayesian	Naïve Bayes
Nearest Neighbors	K-Nearest Neighbors
Decision Trees	C4.5 / J48
Support Vector Machines	Sequential Minimal Optimization
Neural Networks	Radial Basis Functions
Ensemble (Bagging)	Random Forests
Ensemble (Boosting)	Adaptive Boosting

- Most commonly used learners for SDP [1]
- 75% of learners used by primary studies in [2]

[1] R. Malhotra, A systematic review of machine learning techniques for software fault prediction, Applied Soft Computing 2015
[2] R.S. Wahono, A systematic literature review of software defect prediction: Research trends, datasets, methods and frameworks. Journal of Software Engineering 2015

Class imbalance

- Skewness of class instance distribution in a dataset
  - $|Negative\ (majority)\ class| \gg |Positive\ (minority)\ class|$
- Reported through pos/neg (*aka* imbalance) ratio

$$\text{pos/neg ratio} = |Positive\ class| : |Negative\ class|$$

- Typical of (binary) classification problems
  - SW defect prediction, medical screening, fraud and intrusion detection, …
- Impairs classification tasks
  - Learning algorithms performance
  - Performance metrics

Class imbalance: solutions

1. Resampling
2. Cost-sensitive learning
3. Ensemble learning

V. Lòpez et al. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences* 250, 113–141 (2013).

Preprocessing: Classifier settings

- 87% of 30 most commonly used classifiers requires the setting of at least one param [1]
- Parameters often left with default values [2]
  - Data mining toolkits (e.g., R, Weka, scikit-learn) have very different default settings
  - Study replicability seriously limited
- Without param tuning, most classifiers may
  - severely underperform with suboptimal configs [3]
  - build models with statistically indistinguishable performances [4]

[1] C. Tantithamthavorn et al., Automated Parameter Optimization of Classification techniques for Defect Prediction Models, ICSE'16
[2] T. Menzies and M. Shepperd, Special issue on repeatable results in software engineering prediction. ESE 2012
[3] T. Hall et al. A systematic literature review on fault prediction performance in software engineering. TSE 2012
[4] B. Ghotra et al., Revisiting the Impact of Classification Techniques on the Performance of Defect Prediction Models, ICSE'15

Automated param tuning techniques

- Narrow down the space to explore
  - Tuning process requires hours, not days!
- Benefits
  - Boasts prediction models performance
  - Increases models' stability
- Param tuning is *very* dataset-dependent

Dear everyone who has used data miners with their default parameter tunings. #WrongThingToDo [1,2]

And you know all those conclusions you made that learnerA was better than learnerB? Or that attributesA where more important than attributesB cause the learner told you so? Or all those lit reviews and SLRs that made conclusions from reading other people's data mining results? #ReallyWrongThingToDo [1]

Also, (just a heads up), in the near future, data science papers might get rejected if they don't have an auto-tuning pre-study. Further (ping Mark Harman) on that day, when all SE data science needs search to find tunings, then all SE data science will become search-based SE.

REFERENCES

[1] "Tuning for Software Analytics: is it Really Necessary?" by Wei Fu, Tim Menzies, Xipeng Shen IST journal 2016, https://goo.gl/5w5GmM

[2] "Automated Parameter Optimization of Classification Techniques for Defect Prediction Models" by Chakkrit Tantithamthavorn, Shane McIntosh , Ahmed E. Hassan, Kenichi Nakamura ICSE'16: http://goo.gl/ae4rQy

GitHub
timm/timm.github.io
timm.github.io - my web site

W. Fu et al., Tuning for software analytics: Is it really necessary? IST 2016

Feature selection techniques

- Enhances classification performance (shorter training times)
- Simplifies the model (interpretability)
- Param tuning change what features are important [1]
- Recommendation:
  use Wrapper methods [2]
  - Alternatively, Correlation Feature Selection (CFS)

http://tiny.cc/timm5

[1] W. Fu et al., Tuning for software analytics: Is it really necessary? IST 2016
[2] T. Menzies, How not to do it: Anti-patterns for data science in SE, ICSE Tech. briefing, 2016

Performance metrics

Confusion matrix		Prediction		
		Positive	Negative	
Actual	Positive	True Positives (TP)	False Negatives (FN)	Pc
	Negative	False Positives (FP)	True Negatives (TN)	Nc

Positive class $= TP + FN$
Negative class $= FP + TN$

Scalar metrics

Metrics (*synonyms*)	Definition	Description
Accuracy	$Acc = \frac{TP + TN}{TP + FN + FP + TN}$	Proportion of correctly classified instances
Error rate	$E = 1 - Acc$	Proportion of incorrectly classified instances
Precision (*Positive Predicted Values*)	$P = \frac{TP}{TP + FP}$	Proportion of instances correctly classified as positive
Recall (*Probability of Detection, True Positive rate, Sensitivity*)	$R = TP_{rate} = \frac{TP}{TP + FN}$	Proportion of positive instances correctly classified
F-measure (*F1-score*)	$F = 2\frac{P \times R}{P + R}$	Harmonic mean of Precision and Recall
True Negative rate (*Specificity*)	$TN_{rate} = \frac{TN}{TN + FP}$	Proportion of negative instances correctly classified
G-mean	$G = \sqrt{TP_{rate} \times TN_{rate}}$	Geometric mean of True Positive rate and True Negative rate
Matthews Correlation Coefficient	$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}}$	Correlation coefficient between observations and predictions (defined in $[-1, +1]$)
False Positive rate (*Probability of False Alarm*)	$FP_{rate} = \frac{FP}{FP + TN}$	Proportion of negative instances misclassified
Balance	$B = 1 - \frac{\sqrt{(0 - FP_{rate})^2 + (1 - TP_{rate})^2}}{\sqrt{2}}$	Distance from the point (0, 1) in the ROC space representing the perfect classification performance
AUC (*AUROC*)	Area under the ROC Curve	Probability to rank a randomly chosen positive instance higher than a randomly chosen negative one

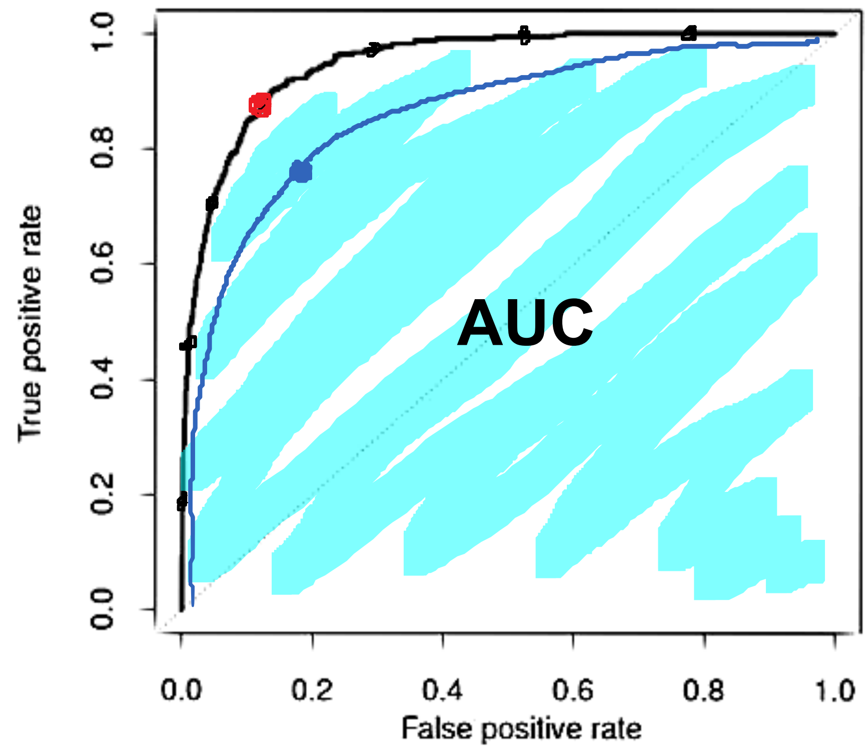
Graphical analysis

- Aggregate scalar metrics improve over single scalar metrics to assess a classification model performance
- However, graphical analysis is better suited to compare multiple models
  - Scalar measures impose a one-dimensional ordering
  - Two-dimensional plots are more capable of preserving performance-related info

C Drummond & R.C. Holte, Cost curves: An improved method for visualizing classifier performance, 2006

Graphical analysis: ROC curve

- Receiver Operating Characteristic
  - Shows the tradeoff between accurate classification of pos instances (*Recall*) and misclassification of neg instances (FP_{rate})
- (0,1) is perfect classification
  - Line connecting (0,0) and (1,0) is the random performance



T. Fawcett, An introduction to roc analysis. Pattern recognition letters 27(8), (2006)

Best model selection

- There is no absolute best prediction model
  - Pick the right model for the given context
- Empirical work must assess the performance of models trained by several classifiers
  - Statistical significance nonparametric test [1]
    - Friedman + Nemenyi post-hoc test: finds groups of mean values statistically different from each other [2]:
    - Scott-Knott: clustering algorithm, finds statistically distinct ranks with no overlapping [3]

[1] Y. Jiang et al, Techniques for evaluating fault prediction models, EMSE 2008
[2] J. Demsar, Statistical comparisons of classifiers over multiple data sets. The Journal of Machine Learning Research 2006
[3] B. Ghotra et al. Revisiting the Impact of Classification Techniques on the Performance of Defect Prediction Models, ICSE’15

Cross-project/company SDP

- What if a project is new or has not collected historical data to build predictive modes?
  - Train models on data from
    - Other (similar?) projects within the same company? Cross-project defect prediction
    - Other (similar?) projects within the other companies? Cross-company defect prediction
- T. Zimmermann et al. Cross-project defect prediction: a large scale experiment on data vs. domain vs. process. ESEC/FSE '09
- B. Turhan, et al., On the relative value of cross-company and within-company data for defect prediction, EMSE 2009
- J. Nam and S. Kim, Heterogeneous defect prediction, ESEC/FSE 2015
- F. Zhang et al. Towards building a universal defect prediction model with rank transformed predictors, EMSE 2016,

SDP: Lessons learned

- Prefer aggregate scalar metrics over single scalar metrics
- Rely on graphical analysis to compare the performance of multiple prediction models on one dataset
- Tune learners’ parameters & select relevant features
- Always include a preliminary assessment to identify most promising learners for the given context
- Select best prediction model informed by statistical significance test
- Cross-prediction possible, but a *much* harder task

Back to the study

FROM DEFECT PREDICTION TO BEST-ANSWER PREDICTION

Observational Study

Best-answers prediction in technical Q&A sites

- Context
  - Within-platform prediction
    - Training and test sets from Stack Overflow
  - Cross-platform prediction
    - Training set from Stack Overflow
    - Test set from both modern Q&A site and legacy support forums
  - Take into due account class imbalance
    - Adequate classification algorithm
    - Adequate performance metrics
- Goal
  - Assess to what extent knowledge could be automatically migrated to Stack Overflow
  - Identify best predicting features for the problem, not the platform

Best answer: definition

- The answer marked as the accepted solution by the original asker
  - i.e., the **fastest, good-enough answer** that satisfies the info seeker
  - Takes into account the time dimension
  - Same conceptualization of Stack Overflow
- A question thread may contain another one considered better by the community (e.g., comments like “*This should be the accepted solution!!*”)
  - ~~absolute best answer~~

Datasets

	Stack Overflow	Docusign	Dwolla	Yahoo! Answers	SAP Comm. Network
Q&A Platform	Modern	Legacy	Legacy	Modern	Modern
Questions threads	507K	1,572	103	41,190	35,544
Questions resolved (%)	279K (~55%)	473 (~30%)	50 (~48%)	29,021 (~70%)	9,722 (~27%)
Answers	1.37M	4,750	375	104,746	141,692
Answers accepted (%)	279K (~20%)	473 %)	50 (~13%)	29,021 (~28%)	9,722 (~6%)
pos/neg ratio	~1:4	~1:10	~1:7	~1:4	~1:15

Datasets

	Extracted Information Elements	Stack Overflow	Docusign	Dwolla	Yahoo! Answers	SCN
Thread content	Type (quest./answer)	Yes	Yes	Yes	Yes	Yes
	Body	Yes	Yes	Yes	Yes	Yes
	Title	Yes	Yes	Yes	Yes	Yes
	Author	Yes	Yes	Yes	Yes	Yes
	~~Tags~~	~~No~~	~~Yes~~	~~No~~	~~No~~	~~No~~
	~~Comments~~	~~Yes~~	~~No~~	~~No?~~	~~?~~	~~?~~
Thread metadata	URL	Yes	Yes	Yes	Yes	Yes
	Question id	Yes	Yes	Yes	Yes	Yes
	Question resolved	Yes	Yes	Yes	Yes	Yes
	Answer count	Yes	Yes	Yes	Yes	Yes
	Accepted answer	Yes	Yes	Yes*	Yes	Yes
	Date / time	Yes	Yes	Yes	Yes	Yes
	~~Answer views~~	~~No~~	~~Yes~~	~~No~~	~~No~~	~~No~~
	Rating score	Yes	Yes	No	Yes	Yes

Features & ranking

Feature type	**Feature name**
Linguistic	Length
	Word count
	No. sentences
	Longest sentence
	Avg. words per sentence
	Avg. chars per word
	Contains hyperlinks
Meta	Age
	Rating score
Vocabulary	*Log-Likelihood normalized (LL_n)*
	Flesch-Kinkaid grade (F-K)
Thread	Answer count

- No user-related features
- All features computationally inexpensive

Feature ranking: Examples

Answers	Word count
a1	100
a2	310
a3	209
a4	145

Answers	Word count ranked
a2	1
a3	2
a4	3
a1	4

ascending

Answers	Age
a1	3 min
a2	4 min
a3	9 min
a4	15 min

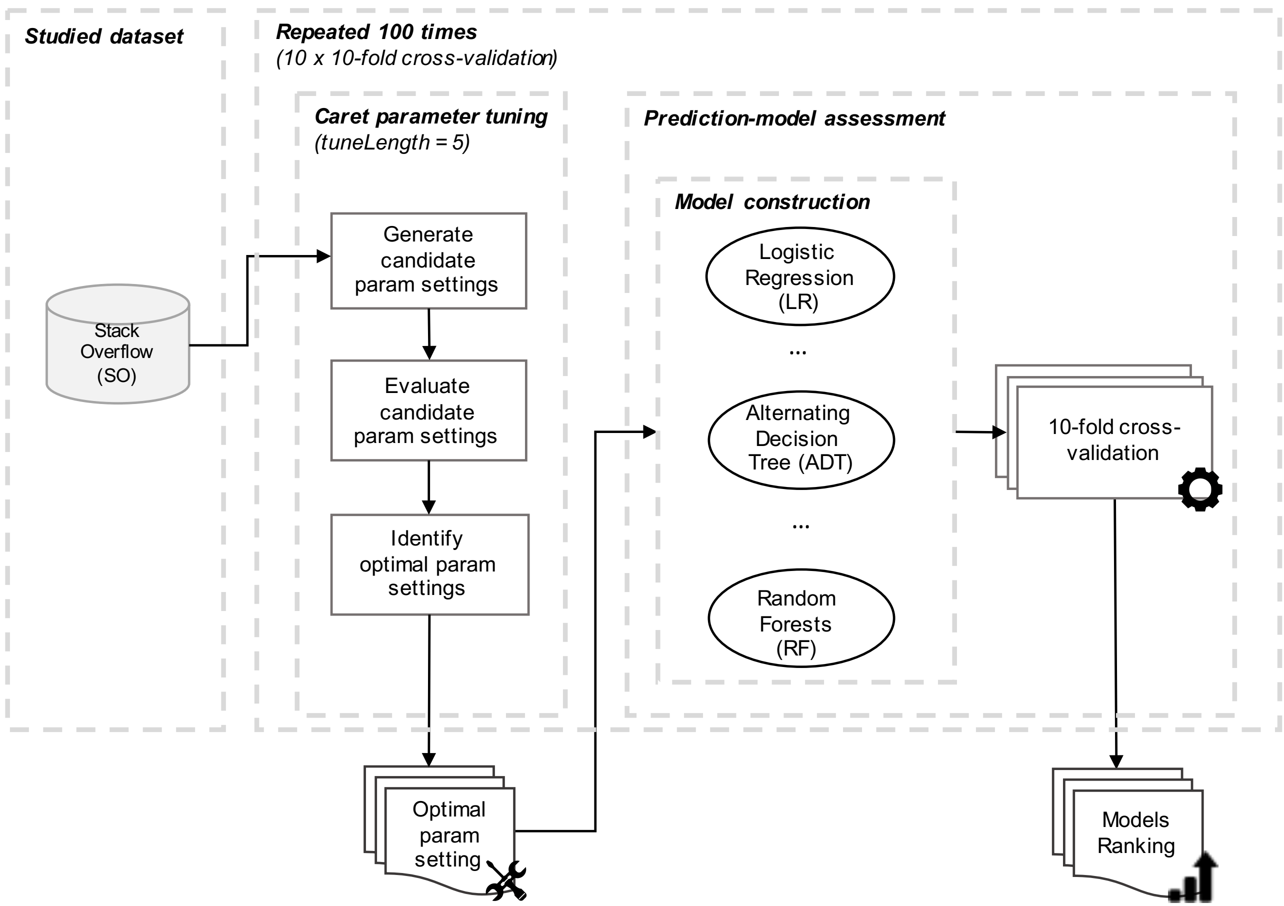
Answers	Age ranked
a1	1
a2	2
a3	3
a4	4

descending

Study execution

Step 1:

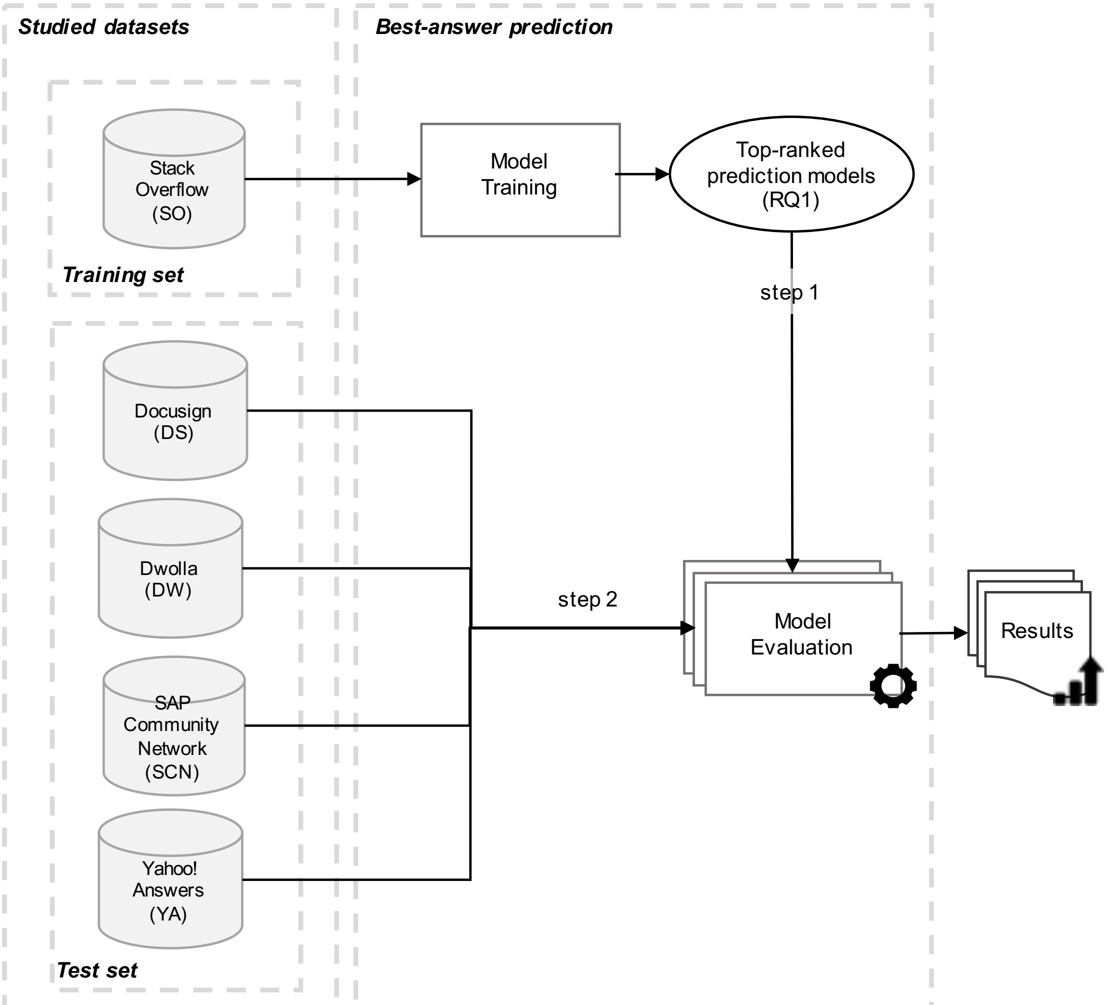
Best-answer prediction within Stack Overflow



Study execution

Step 2:

Cross-platform best-answer prediction



Family	Classifier (short name)	Parameters	Description
Regression-based	Generalized Linear Models (glm)	-	
	Multivar. Adaptive Regression Splines (earth)	degree nprune	Max degree of interaction Max # of terms in model
Bayesian	Naïve Bayes (nb)	fL usekernel?	Laplace correction factor Use kernel density estimate
Nearest Neighbor	K -Nearest Neighbor (knn)	k	# Clusters
Discrimination Analysis	Linear Discriminant Analysis (lda)	-	
	Penalized Discriminant Analysis (pda)	lambda	Shrinkage penalty coefficient
	Flexible Discriminant Analysis (fda)	degree nprune	Max degree of interaction Max # of terms in model
Decision Trees	C4.5-like trees (J48)	C	Confidence factor for pruning
	Logistic Model Trees (LMT)	iter	# Iterations
	Classification and Regression Trees (rpart)	cp	Complexity penalty factor
Support Vector Machines	SVM with Linear Kernel (svmLinear)	C	Cost penalty factor
Neural Networks	Standard (nnet)	size decay	# Hidden units Weight decay penalty factor
	Feature Extraction (pcaNNet)	size decay	# Hidden units Weight decay penalty factor
	Model Averaged (avNNet)	bag? size decay	Apply bagging at each iteration # Hidden units Weight decay penalty factor
	Multi-layer Perceptron (mlp)	size	# Hidden units
	Voted-MLP (mlpWeightDecay)	decay size	Weight decay penalty factor # Hidden units
	Penalized Multinomial Regression (multinom)	decay	
Rule-based	Repeated Incremental Pruning Reduction (JRip)	NumOpt	# Optimization iterations
Bagging	Random Forests (rf)	mtry	# Predictors sampled
	Bagged CART (treebag)	-	
Boosting	Gradient Boosting Machine (gbm)	n.trees interact. depth shrinkage n.minobsinnod	# Trees to fit Max depth of var. interactions Param. applied to tree expansion Min # terminal nodes
	Adaptive Boosting (AdaBoost)	mfinal maxdepth coeflearn	# Boosting iterations Max tree depth Weight updating coefficient
	General. Additive Models Boost. (gamboost)	mstop prune?	# Initial boosting iterations Apply pruning w/ stepwise feat. selection
	Logistic Regression Boosting (LogitBoost)	nIter	# Boosting iterations
	eXtreme Gradient Boosting Tree (xgbTree)	nrounds maxdepth eta	Max # iterations Max tree depth Step-size shrinkage coefficient
	C5.0 (C50)	trials model winnow?	# Boosting iterations Decision trees or rule-based Apply predictor feature selection

Study results

BEST-ANSWER PREDICTION WITHIN STACK OVERFLOW

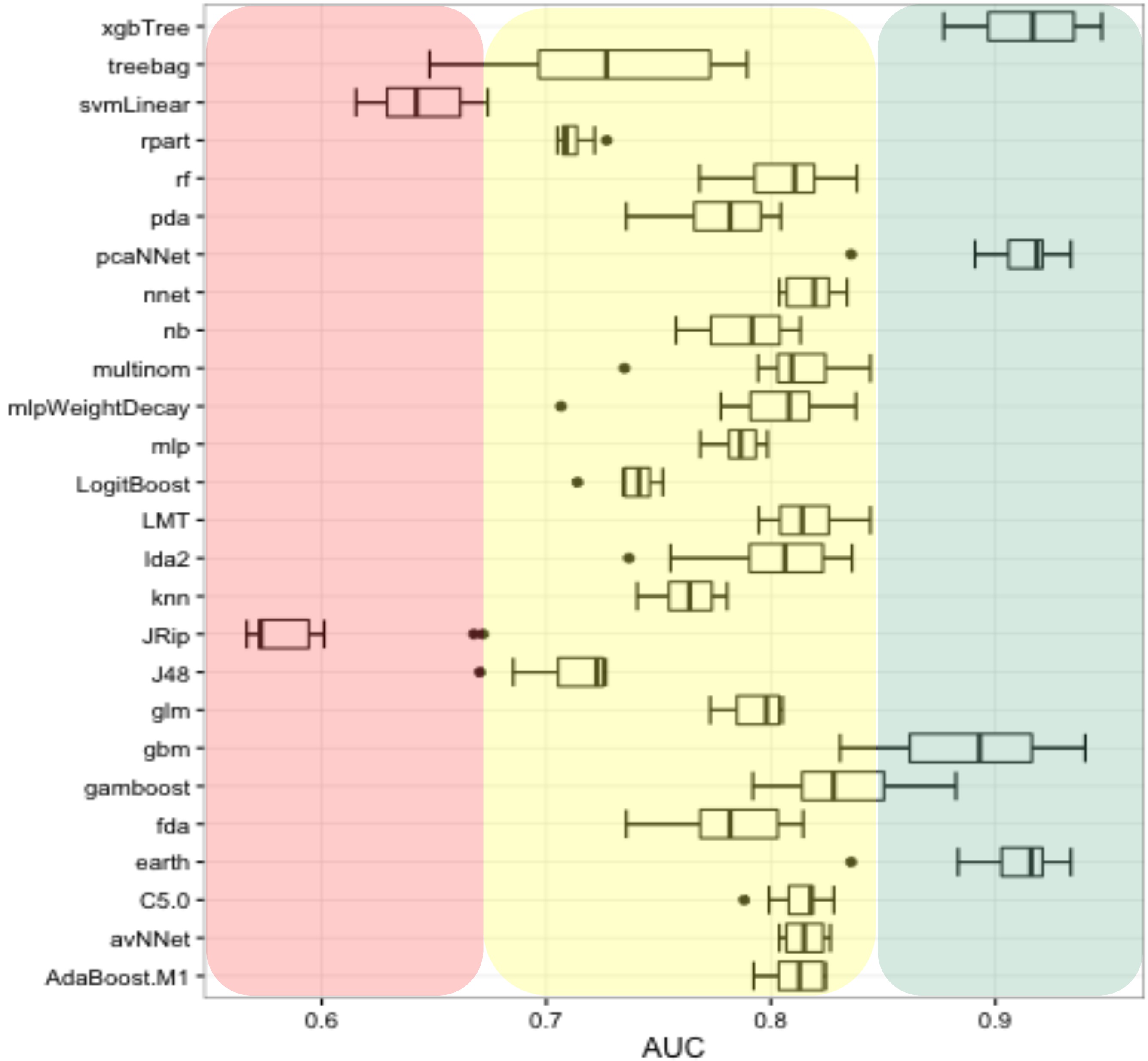
Scott-Knott test

- 6 statistically distinct ranks
- AUC performance increase due to tuning up to 43%

Rank	Prediction model	Mean AUC	Max	Min	SD
1	**xgbTree**	**0.91**	**0.95**	**0.88**	**0.02**
	pcaNNet	**0.91**	**0.93**	**0.84**	**0.03**
	Earth	**0.91**	**0.93**	**0.83**	**0.03**
	gbm	**0.90**	**0.94**	**0.83**	**0.04**
2	gabmboost	0.83	0.88	0.79	0.03
	nnet	0.82	0.83	0.80	0.01
	LMT	0.82	0.84	0.79	0.02
	avNNet	0.82	0.83	0.80	0.01
	C5.0	0.81	0.83	0.79	0.01
	AdaBoost	0.81	0.82	0.79	0.01
	multinom	0.81	0.84	0.73	0.03
	rf	0.81	0.84	0.77	0.02
	lda	0.80	0.84	0.74	0.03
	mlpWeightDecay	0.80	0.84	0.71	0.04
3	glm	0.79	0.81	0.77	0.01
	nb	0.79	0.81	0.76	0.02
	mlp	0.79	0.80	0.77	0.01
	fda	0.78	0.81	0.74	0.02
	pda	0.78	0.80	0.74	0.02
	knn	0.76	0.78	0.74	0.01
4	LogiBoost	0.74	0.75	0.71	0.01
	treebag	0.73	0.79	0.65	0.05
	J48	0.71	0.73	0.67	0.02
	rpart	0.71	0.73	0.70	0.01
5	svmLinear	0.64	0.67	0.62	0.02
6	JRip	0.59	0.67	0.57	0.04

Boxplot

- No pattern observed related to classification techniques

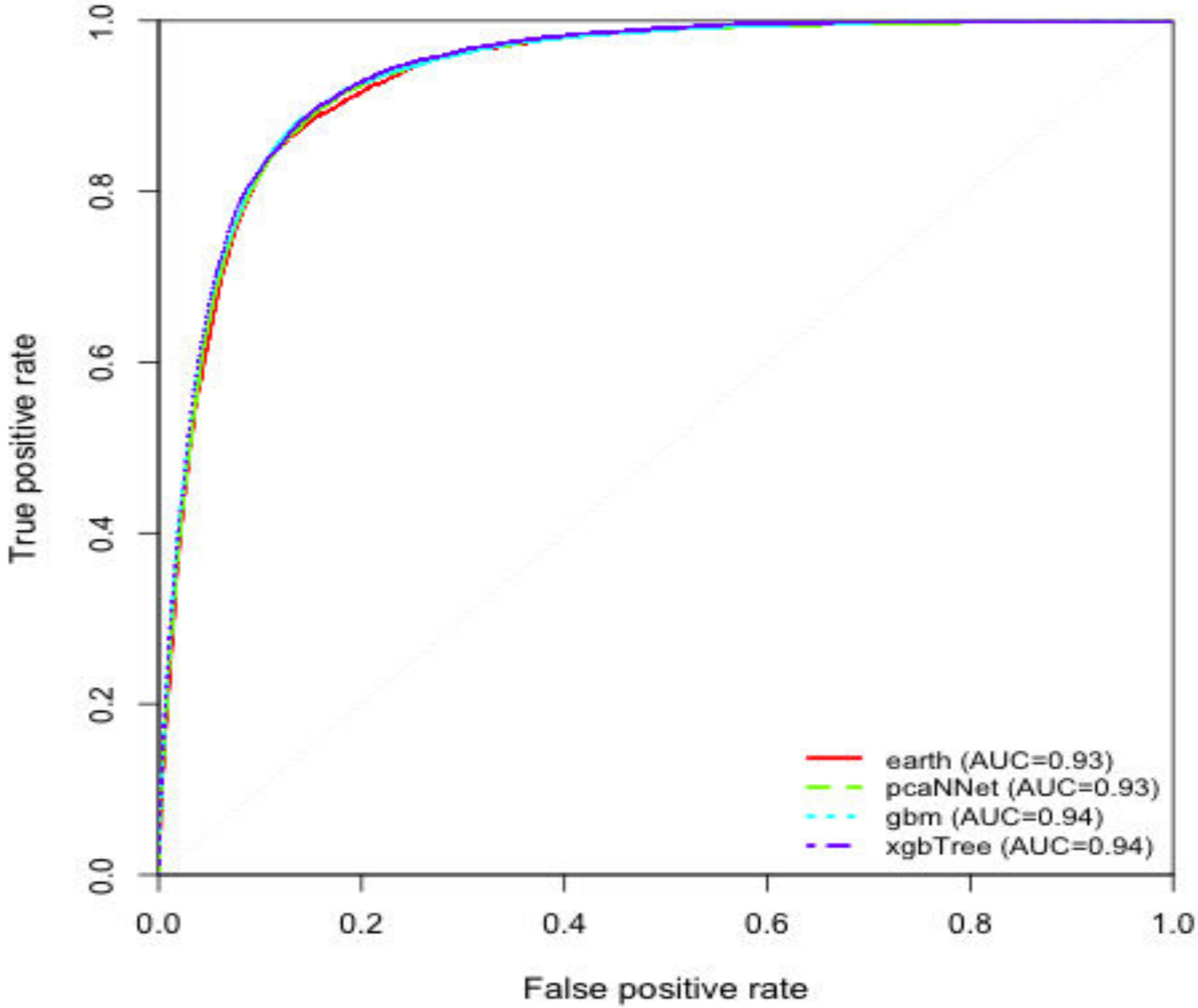


Parameter tuning

Prediction Model	Optimal parameter configuration	Default parameter configuration	Overall tuning runtime
xgbTree	`nrounds = 200`	`nrounds = 100`	6h 47m
	`max_depth = 4`	`max_depth = 1`	
	`eta = 0.1`	`eta = 0.3`	
pcaNNet	`size = 7`	`size = 1`	2h 20m
	`decay = 0.1`	`decay = 0`	
earth	`nprune = 15`	`nprune = NULL`	3h 53m
	`degree = 1`	`degree = 1`	
gbm	`n.trees = 250`	`n.trees = 100`	8h 44m
	`interaction.depth = 3`	`interaction.depth = 1`	
	`shrinkage = 0.1`	`shrinkage = 0.1`	
	`n.minobsinnode = 10`	`n.minobsinnode = 10`	
...	...	...	...

- At least one param tuned from default config
- Tuning took hours, not days

ROC plots



True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0
earth (AUC=0.93)
pcaNNet (AUC=0.93)
gbm (AUC=0.94)
xgbTree (AUC=0.94)

Scalar metrics

Models	F	G-mean	AUC	Balance
xgbTree	0.91	0.87	0.94	0.87
gbm	0.92	0.88	0.94	0.87
pcaNNet	0.90	0.86	0.93	0.85
earth	0.86	0.84	0.93	0.82

Feature selection

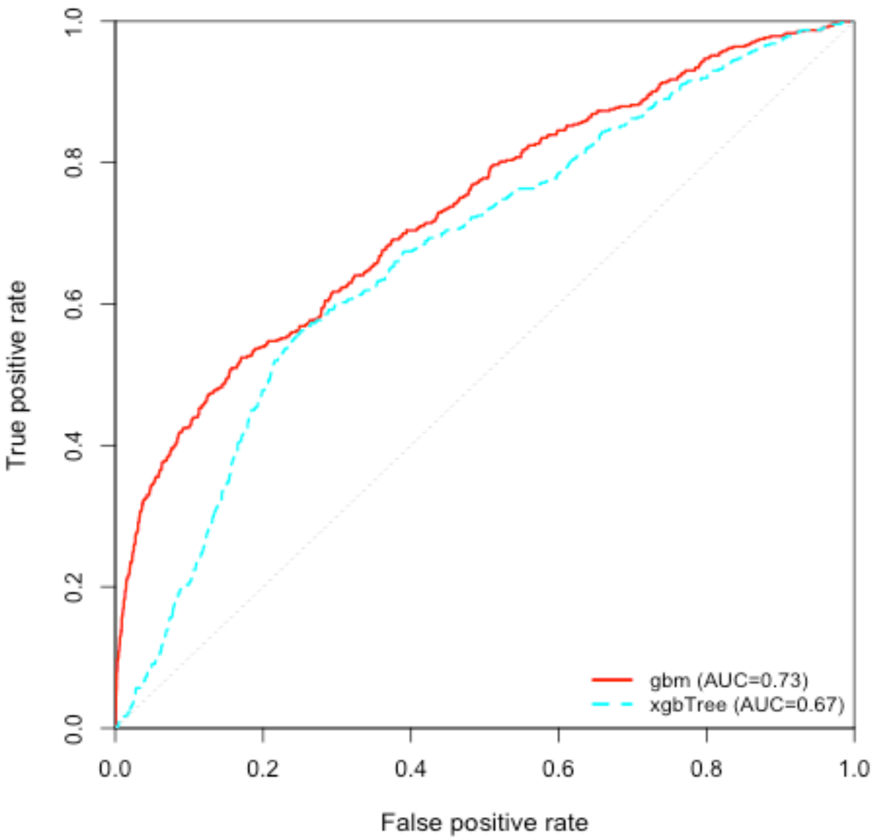
Feature category	Feature name	Pos.	Feature importance (Z)	Gain (ranked – non-ranked) Pos.	Gain (ranked – non-ranked) Importance
Linguistic	Length	16	38.64	+7	+9.85
	Length_ranked	9	48.49		
	Word count	14	40.06	+3	+2.92
	Word count_ranked	11	42.98		
	No. of sentences	13	41.7	+5	+7.57
	No. of sentences_ranked	8	49.34		
	Longest sentence	18	38.21	+11	+13.83
	Longest sentence_ranked	7	50.29		
	Avg. words per sent.	12	42.27	+6	+12.58
	Avg. words per sent._ranked	6	54.85		
	Avg. chars per word	19	36.87	+15	+28.09
	Avg. chars per word_ranked	4	64.26		
	Contains hyperlinks	22	0.02	N/A	N/A
Meta	Age	5	57.11	+2	+17.85
	Age_ranked	3	74.96		
	Rating score	2	118.24	+1	+16.59
	Rating score_ranked	1	134.83		
Vocabulary	LL$_n$	21	12.14	+1	+1.22
	LL$_n$_ranked	20	13.36		
	F-K	15	39.55	+5	+8.72
	F-K_ranked	10	48.27		
Thread	Answer count	17	38.63	N/A	N/A

Study results

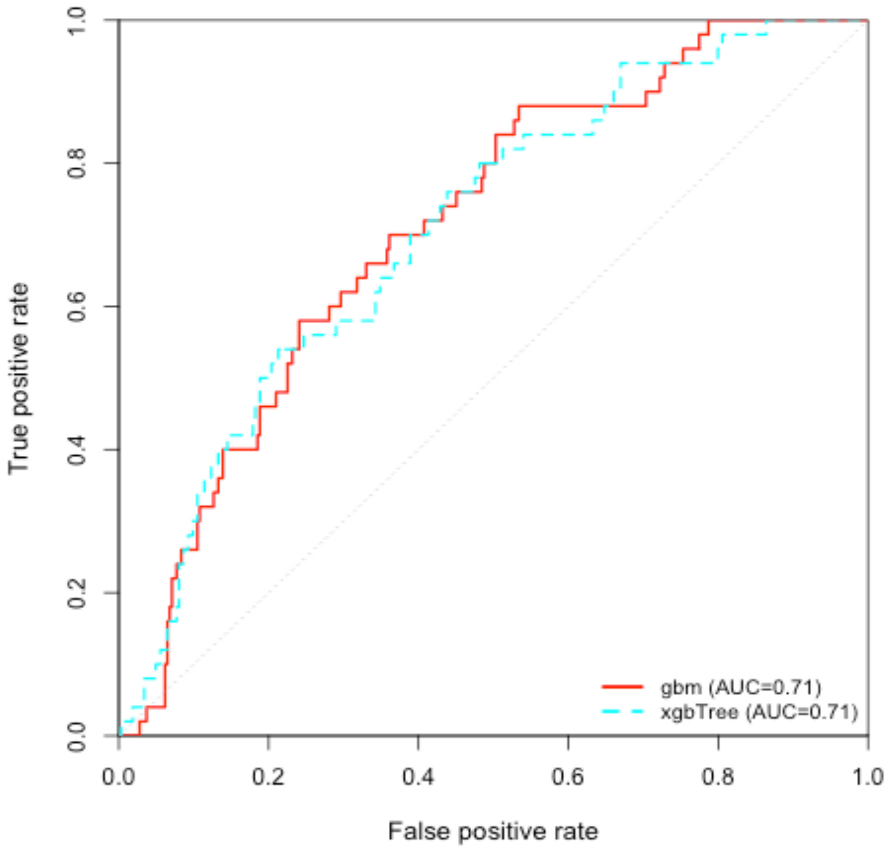
CROSS-PLATFORM BEST-ANSWER PREDICTION

ROC plots:
Legacy platforms

Docusign

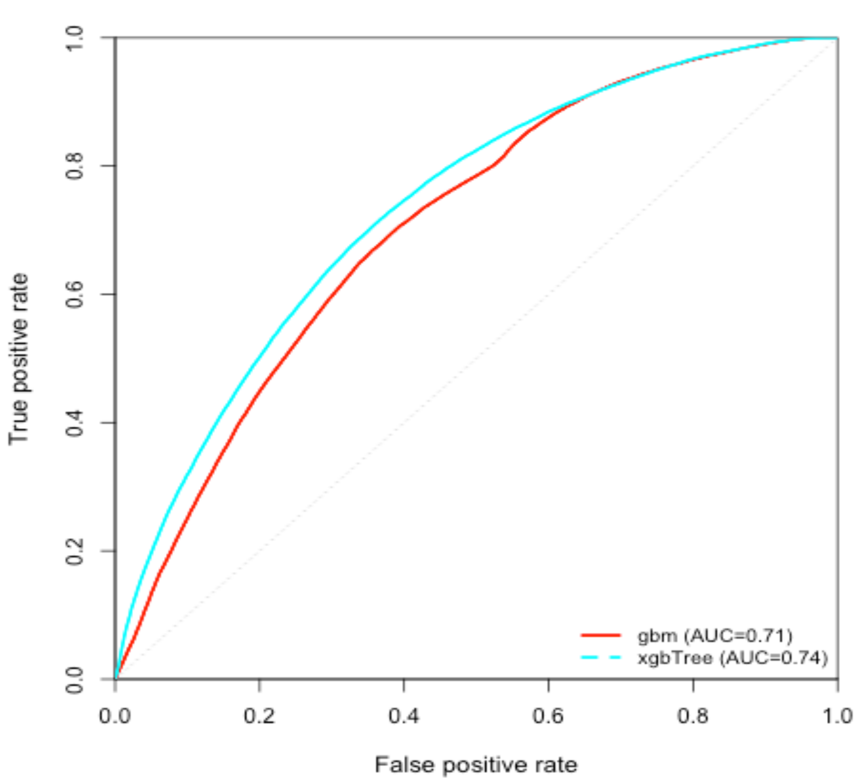


Dwolla

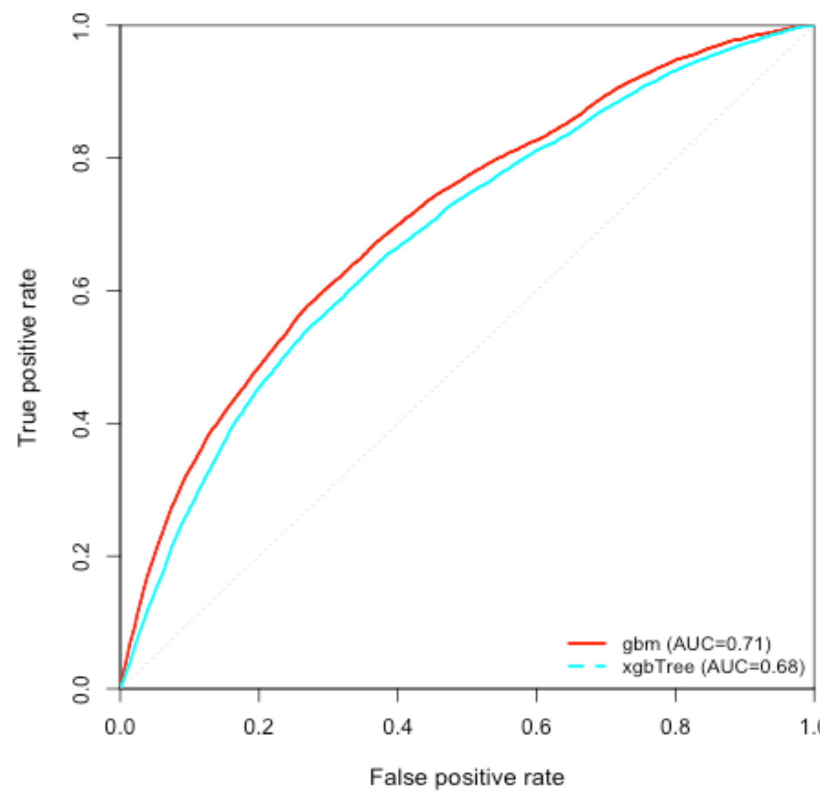


ROC plots: Modern platforms

Yahoo! Answers



SCN



Scalar metrics

	Test set (pos/neg)	xgbTree				gbm			
		F	G-mean	AUC	Balance	F	G-mean	AUC	Balance
I	Docusign (1:10)	0.86	0.62	0.67	0.61	0.82	0.65	0.73	0.64
II	Dwolla (1:7)	0.85	0.63	0.71	0.62	0.80	0.65	0.71	0.65
III	Yahoo (1:4)	0.66	0.65	0.74	0.64	0.72	0.65	0.71	0.65
IV	SCN (1:15)	0.84	0.62	0.68	0.62	0.80	0.65	0.71	0.65
	Avg.	0.80	0.63	0.70	0.62	0.79	0.65	0.72	0.65
	S.D.	0.10	0.01	0.02	0.01	0.04	0.00	0.01	0.01

Cross-platform models performance benchmarking

Datasets (pos/neg) \ Models		xgbTree				gbm			
		Trivial rejector	Cross-platform model	Within-platform model	Cross- vs. within-platform models performance variation	Trivial rejector	Cross-platform model	Within-platform model	Cross- vs. within-platform models performance variation
Docusign (1:10)	F	0.85	0.86	0.95	–9%	0.85	0.82	0.95	-14%
	G	0.95	0.62	0.32	+94%	0.95	0.65	0.37	+76%
	Bal	0.36	0.61	0.37	+65%	0.36	0.64	0.39	+64%
	AUC	0.49	0.67	0.74	–9%	0.49	0.73	0.75	–3%
Dwolla (1:7)	F	0.80	0.85	0.95	–11%	0.80	0.80	0.93	–14%
	G	0.93	0.63	0.52	+21%	0.93	0.65	0.44	+48%
	Bal	0.38	0.62	0.48	+29%	0.38	0.65	0.43	+51%
	AUC	0.49	0.71	0.83	–14%	0.49	0.71	0.83	–14%
Yahoo (1:4)	F	0.58	0.66	0.93	–29%	0.58	0.72	0.92	–22%
	G	0.84	0.65	0.90	–28%	0.84	0.65	0.89	–27%
	Bal	0.46	0.64	0.90	–29%	0.46	0.65	0.96	–32%
	AUC	0.50	0.74	0.97	–24%	0.50	0.71	0.96	–26%
SCN (1:15)	F	0.80	0.84	0.96	–13%	0.80	0.80	0.96	–17%
	G	0.96	0.62	0.12	+417%	0.96	0.65	0.12	+442%
	Bal	0.34	0.62	0.30	+107%	0.34	0.65	0.77	–16%
	AUC	0.50	0.68	0.78	-13%	0.50	0.71	0.77	–8%

Reference	Dataset (# quest./answ.)	pos/neg ratio	Feature categories (total #)	Feature ranking?	Experimental setting	Param tuning?	Other classifiers compared	Performance results	Graphical assessment
Adamic et al. (2008)	Yahoo! Answers – Programming & Design (N/A)	N/A	user, thread, linguistic (4)	No	10-fold cross-validation with Logistic Regression	No	No	Acc= ~73%	No
Shah and Pomerantz (2010)	Yahoo! Answers** (~1.3K/5K)	N/A	user, thread, meta, linguistic (21)	No	10-fold cross-validation with Logistic Regression	No	No	Acc= ~84%	No
Cai and Chakravarthy (2011)	Stack Overflow (1K/5K)*	1:4	textual, user (22)	No	10-fold cross-validation with SVM	No	No	P=.55	No
Tian et al. (2013)	Stack Overflow (~103K/196K)	N/A	thread, meta, linguistic (16)	No	2-fold cross-validation with Random Forests	No	No	Acc= ~72%	No
Burel et al. (2012)	SCN† (~95K/427K) Server Fault (SF)‡ (~36K/95K)	N/A	user, thread, meta, linguistic, vocabulary (19† /23‡)	No	10-fold cross-validation with ADT	No	J48, Random Forests, ADT, Random Trees	P=.83, R=.84, F=.83 AUC=.88 (SCN)	No
								P=.85, R=.85, F=.80 AUC=.91 (SF)	
Shah (2015)	Yahoo! Answers** (23K Q/A pairs)*	1:4	textual, user (12)	No	70/30% training/test set split with Bayesian Network	No	No	Acc=89.2 P=.97, R=.86 AUC=.98	ROC plot
Gkotsis et al. (2014, 2015)	21 Stack Exchange sites (incl. Stack Overflow)** (~3M/7M)	N/A	thread, meta, linguistic, vocabulary (14)	Yes	10-fold cross-validation with ADT	No	Yes (unspecified)	P=.82, R=.66, F=.73 AUC=.85 (SO only)	ROC plot
					Cross-site leave-one-out with ADT			P=.84, R=.70, F=.76 AUC=.87 (avg)	
This study	Stack Overflow (507K/1.37M)	~1:4	thread, meta, linguistic, vocabulary (22)	Yes	10-fold cross-validation with 26 classifiers	Yes	Yes	AUC=.94	ROC plots
	Docusign (~1.5K/~4.7K)	~1:10			Cross-site training vs. test set			AUC=.73, F=.82, G=.65, Bal=.64	
	Dwolla (103/375)	~1:7						AUC=.71, F=.80, G=.65, Bal=.65	
	Yahoo! Answers – Progr. & Design (~41.2K/~105K)	~1:4						AUC=.74, F=.66, G=.65, Bal=.64	
	SCN (~35.5K/~141.7K)	~1:15						AUC=.71, F=.80, G=.65, Bal=.65	

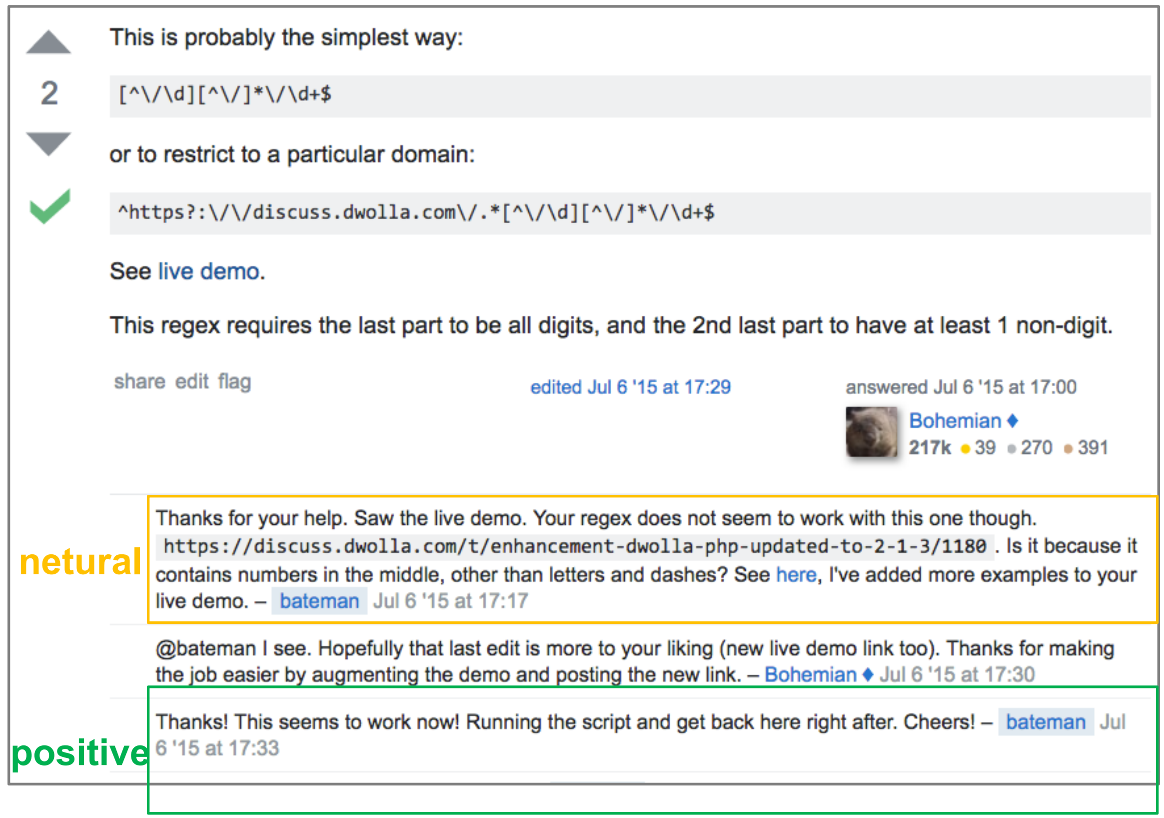
*Opportunistically sampled for selecting question threads with 1 best answer and 4 non-accepted answers. **Dataset mixes technical and non-technical help requests.

Contributions

- First-attempt as cross-platform best-answer prediction
  - Analysis of multiple classifiers
  - in both modern and legacy platforms
  - Cross-platform prediction statistically above the baseline and similar to the upperbound models (AUC)
- (Some) prior work
  - Built prediction models using one classification technique only
  - Failed to visually assess differences by plotting performance curves
  - Reported performance by single scalar measures, sometimes unstable and sensitive to dataset imbalance
- Reliable benchmark for further studies on best-answer prediction
  - Recommended measures and performance baseline

Back to "emotions in SE"

- Use NLP techniques to extract new "shallow" linguistic features from text
  - Leverage sentiment analysis and seek shifts in polarity (+/=/-)
  - E.g. tone of asker's comments before and after a working solution is provided

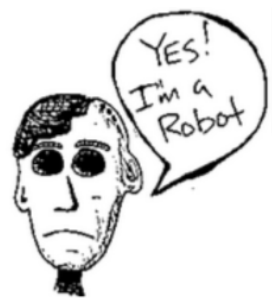


Future work (?)

- Use knowledge to actually provide a best answers to questions that are still open because unresolved
  - E.g., Q&A bot?
- Credits
  - A. Zagalsky :-)

1

Among the Machines: Human-Bot Interaction on Social Q&A Websites



Answer_Bot

I am an experimental bot created at the University of Antwerp. Part of an ongoing academic research project, we are trying to understand whether a certain type of questions posted on Stack Overflow (those that are seemingly duplicates) can be replied to automatically. If a good answer to your question already exists on Stack Overflow, I will link to it when answering your question. If not, I won't bother you.

"*One day the AIs are going to look back on us the same way we look at fossil skeletons on the plains of Africa.*" (Nathan; Ex Machina, 2015)

Abstract

With the rise of social media and advancements in AI technology, human-bot interaction will soon be commonplace. In this paper we explore human-bot interaction in Stack Overflow, a question and answer website for developers. For this purpose, we built a bot emulating an ordinary user answering questions concerning the resolution of `git` error messages. In a first run this bot impersonated a human, while in a second run the same bot revealed its machine identity. Despite being functionally identical, the two bot variants elicited quite different reactions.

Author Keywords

Social Bot; Stack Overflow; Turing Test

ACM Classification Keywords

H.5.m [HCI]: Miscellaneous

Introduction

Ever since the Turing test [25] and the ELIZA experiment [27] the prospect of having meaningful interactions with artificial intelligence (AI) agents has been firing human imagination. While AI agents that circulate inconspicuously among

A.Murgia, et al. Among the Machines: Human-Bot Interaction on Social Q&A Websites, *CHI 2016*

Thanks!

Contacts

fabio.calefato@uniba.it

collab.di.uniba.it/fabio

@fcalefato