Beyond Accuracy: ROI-driven Data Analytics of Empirical Data

ABSTRACT

Background: Solving decision problems starts with designing an analytical solution, based on the data, tools and techniques available. However, analytics might not be always valuable and could be expensive. As being true for any technology, it is crucial to select the most appropriate one based on its efficiency for the problem at hand.

Objective: The objective of this vision paper is to demonstrate the need to consider Return-on-Investment (ROI) as an important factor when performing Data Analytics (DA). Decisions on "How much is needed?" are hard to answer, however, ROI could be used as guidance for decision support for deciding on What? How? and How Much? of analytics for a given problem.

Method: The conceptual framework proposed is validated through two empirical studies for extracting requirements dependencies in evolving software products using Mozilla Firefox data. The two studies: (i) Evaluation of BERT against Naive Bayes and Random Forrest machine learners for binary dependency classification and (ii) Active Learning for *REQUIRES* dependency extraction, a specific form of dependency. For the selected techniques, their investment (cost) is estimated and the achievable benefit from applying DA is predicted.

Results: There is a break-even point for investing in DA. For the binary dependency extraction problem, a fine-tuned BERT model performed superior to Random Forest, provided there is more than 40% of training data available. For extracting *REQUIRES* dependency, Active Learning achieved convergence to higher F1 within fewer iterations and higher ROI compared to Baseline (Random sampling based RF classifier).

Conclusions: Decisions for the depth and breadth of DA of empirical data should not be done solely based on the accuracy measures. The ROI-driven Data Analytics helps to avoid over-analyzing empirical data. Since it provides simple yet effective guidance to discover when to stop further training while considering the cost and value of the analysis, it should be considered when different algorithmic solutions are compared.

KEYWORDS

Data Analytics, Return-on-Investment, Requirements Engineering, Dependency extraction, BERT, Mozilla

Conference'20, ,

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM Reference Format:
. 2020. Beyond Accuracy: ROI-driven Data Analytics of Empirical Data. In *Proceedings of ACM Conference (Conference'20).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Return-on-Investment (ROI) is of great interest in engineering and business for arriving at decisions. The same is true in Software Engineering (SE). For example, Cleland et al. [9] studied the ROI of heterogeneous solutions for the improvement of the ROI of requirements traceability. Recent data explosion in the form of big data and advances in Machine Learning (ML) have posed questions on the efficiency and effectiveness of these processes that have become more relevant. In this paper, we re-performed two empirical studies from the field of requirements dependency analysis and evaluated the benefit of ROI analysis.

Data analytics in SE (also called "Software Analytics" by Bird et al. [7]) has been a field of interest and has been explored to great depth. Most of the solution approaches often present problems ready for the application of tools and techniques which could be instantly applied to them. The reality is messier. Although data processing, labelling, modeling have been identified as a crucial part of any decision problem, they are seldom accounted for before jumping into a solution approach. Thus, in this paper, we argue that ROI should be the driver for solution approach selection and should not just rely on the accuracy measure.

Software Engineering is uncertain in various ways. It is highly human-centric, and the processes are not strictly repeatable. Thus, experimentation and DA are inherently arduous under such circumstances. The famous Aristotle (384 to 322 BC) is widely attributed with saying "It is the mark of an educated mind to rest satisfied with the degree of precision which the nature of the subject admits and not to seek exactness where only an approximation is possible". **Our research hypothesis** that ROI-driven DA helps to avoid excessive emphasis on the precision when it is not possible to achieve it efficiently, even worse when it does not make sense.

Figure 1 shows a typical ROI (cost-benefit) curve of technology usage. Following some phase of increase, curve reaches saturation so, beyond that point, further investment does not pay off. We contemplate that a similar behaviour holds true for applying DA.

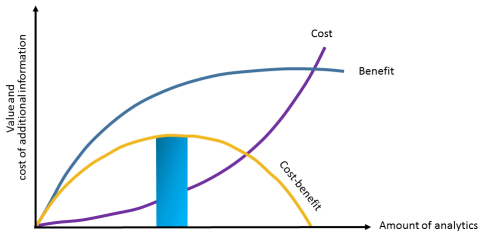


Figure 1: Expected ROI curve from technology investment.

The rest of the paper is structured as follows: Section 2 discusses related work. The problem formulation is explained in Section 3.

ROI is not only discussed conceptually but also in the concrete context of requirements dependency extraction in (open source) Software Systems. In Section 4, a detailed approach of empirical investigations for two variants of the dependency extraction problems are given. A discussion of the applicability of the results is elaborated in Section 5. Finally, in Section 6, we give an outlook on future research.

2 RELATED WORK

2.1 ROI Analysis in Software Engineering

Evaluating the profitability of expenditure helps to measure success over a period of time thus takes the guesswork away from the concrete decision-making process. For instance, Erdogmus et al. [13] analyzed the ROI of quality investment to bring its importance in perspective and posed important questions, "We generally want to increase a software products quality because fixing existing software takes valuable time away from developing new software. But how much investment in software quality is desirable? When should we invest, and where?". In the crux, exploring answers to these questions is the idea of this paper, although the focus is on DA in particular.

Begel & Zimmermann [4] composed a set of 145 questions - based on a survey with more than 200 developers and testers - that are considered relevant for DA at Microsoft. One of the questions:"How important is it to have a software DA team answer this question?", expected answer on a five-point scale (*Essential* to *I don't understand*). Although it provides a sneak peek of the development and testing environments of Microsoft, it does not prove any emphasis on any form of ROI. Essentially, we speculate that the ROI aspect was softened into asking for the perceived subjective importance through this question.

Boehm et al. [8] presented quantitative results on the ROI of Systems Engineering (SE-ROI) based on the analysis of the 161 software projects in the COCOMO II database. Van Solingen [24] analyzed the ROI of software process improvement and took a macro perspective to evaluate corporate programs targeting the improvement of organizational maturity. Ferrari et al. [14] studied the ROI for text mining and showed that it has not only a tangible impact in terms of ROI but also an intangible benefits - which occur from the investment in the knowledge management solution that is not directly translated into returns, but that must be considered in the process of judgment to integrate the financial perspective of analysis with the non-financial ones.

Ruhe and Nayebi proposed a *Analytics Design Sheet* [18] to focus on formal analytics for solving a decision problem. The four-quadrant template provides direction to formulate the most appropriate analytics as a result of gathering information in every step. In its nature, the sheet is qualitative rather than quantitative.

2.2 Empirical Analysis for Requirements Dependency Extraction

The extraction of dependencies among requirements is an active field of SE research. Previous empirical studies have explored diverse computational methods that used natural language processing (NLP) [11], predicate logic [26] and deep learning [15] techniques. However, none of the approaches considered ROI to decide among techniques and their execution level. The closest one was Samer et al. [19], who analyzed small industry data sets and utilized Latent Semantic Analysis to extract *REQUIRES* type dependencies. However, we speculate that some knowledge regarding efforts consumed and the expected outcome of specific techniques could have changed the direction and depth of the analysis altogether.

3 CONCEPTUAL FRAMEWORK FOR ROI-DRIVEN DATA ANALYTICS

Different models exist that provide guidance to perform DA. Wieringa [25] provides checklist for what he calls the design cycle and the empirical cycle. For this study, we use the term - *Scoping* for defining the problem, selection of the (empirical) data to be analyzed, and the statement of the analysis objectives. Scoping also means defining the boundaries that help to exclude inessential part of the investigation. Thus, scoping means to perform analysis of the potential *Return-on-Investment (ROI)* which could determine the depth and the breadth of any investigation.

3.1 Research Question

DA follows a resource and computation intensive funnel shaped process which has data gathering and processing components at its near end that contribute to the non-trivial proportion of the total research cost. Thus, it is essential to account for these to compute the over all cost-benefit and optimize it further.

Our aim is to study DA as part of empirical studies that have been conducted. In particular, we are interested in requirements dependency analysis (RDA). We define and validate the principal concepts needed for ROI-driven DA. Our research question is as follows:

> **RQ:** What are the benefits of ROI-driven *Data Analytics* in Requirements Dependency Analysis?

Justification As for any effort investment, it is most important to know how much is enough. There is no incentive in investing in analytics for the sake of just doing analysis. Even though we cannot claim exactness from analysis, it would be useful to get some guidance on where (which techniques) and how far (how much of it) we should go. To make the analysis concrete, we have selected RDA as the area of our specific investigations.

3.2 Effort Factors

Data processing is an umbrella term used to combine data collection (C_{dg}), pre-processing (C_{pp}) and labeling (C_l) under one hood, each one of which is a cost component. However, not all costs are fixed and some vary based on the solution approach used to tackle any decision problem. For example, supervised Machine Learning (ML) requires a large amount of annotated data, to begin with, whereas Active Learning acquires these annotations over a period of time in iterations until a stopping condition for classification operation is reached [20]. Additionally, there is a cost associated with modeling and evaluation (C_e).

3.3 Value Factors

The value returns or "benefits" are defined based on the needs of the decision problem. In the context of dependency extraction, the

Table 1: Parameters used for ROI computation

	Symbol	Meaning	Unit
Cost	C_{dg}	Data gathering time	Minutes
	C_{pp}	Pre-processing time	Minutes
	C_e	Evaluation time	Minutes
	C_l	Labeling time	Minutes
	$C_{resource}$	Human resource cost	$ per hour
Benefit	B_{reward}	Value per True Positive instance	$
	$B_{penalty}$	Penalty per False Negative instance	$
	$BF1_{iteration}$	Iteration wise F1 difference	Number
	$PValue$	Projected value per 1% improvement in F1	$
Others	H	#Human resources	Number
	N_{train}	Size of the training set	Number
	N_{test}	Size of the test set	Number
	N	N_{train} + N_{test}	Number

benefit could be modeled in terms of the ability of the ML model to identify a larger number of dependencies correctly (higher TP: B_{reward}) while limiting misclassification (reduced FN: $B_{penalty}$). Conversely, the benefit could also be determined based on the net value ($PValue$) of change of accuracy ($BF1_{iteration}$) in every iteration, especially when using Active Learning. Table 1 lists the relevant cost components and their corresponding units. These will be utilized to compute the *ROI* later for the two different problems in Sections 4.4 and 4.5.

3.4 ROI

To determine the ROI, we follow the simplest form of its calculation relating to the difference between $Benefit$ and $Cost$ to the amount of $Cost$. Both $Benefit$ and $Cost$ are measured as human effort in person hours.

$$ROI = (Benefit - Cost)/Cost \quad (1)$$

Costa et al. [10] distinguished the "hard ROI" from the "soft ROI". The former refers to the direct additional revenue generated and cost savings. The latter improved productivity, customer satisfaction, technological leadership, and efficiencies.

4 ROI OF TECHNIQUES FOR REQUIREMENTS DEPENDENCY ANALYSIS

We have selected the area of requirements dependency analysis (RDA) to illustrate and initially validate our former conceptual framework. In what follows, we introduce the key terms needed to formulate two Empirical Analysis Studies called EAS 1 resp. EAS 2.

4.1 Problem statement

Following are the definitions of dependency types that are used to state the two analysis problems. For a set of requirements R and a pair of requirements (r, s) ϵ $R \times R$

1) A ***INDEPENDENT*** relationship is absence of any form of relationship between a pair of requirements.
2) A ***DEPENDENT*** relationship is defined as the complement set of INDEPENDENT, i.e., there exists at least one type of the various dependency types such as *REQUIRES, SIMILAR, OR, AND, XOR, value synergy, effort synergy* etc. between r and s.
3) A ***REQUIRES*** is a special form of *DEPENDENT* relationship. If r requires s, or s requires r, then, r and s are in a *REQUIRES* relationship
4) A ***OTHER*** is when (r, s) is ***DEPENDENT*** and dependency type is not REQUIRES (could be any of the other dependency types mentioned in (2))

Problem 1- Binary requirements dependency extraction: For a given set R of requirements and their textual description, the binary requirements dependency extraction problem aims to classify for each pair (r,s) ϵ $R \times R$ if they are *DEPENDENT* or *INDEPENDENT*.

Problem 2- Specific requirements dependency extraction of the type *REQUIRES*: For a given set R of requirements and their textual description, the REQUIRES dependency extraction problem aims to classify for each pair (r,s) ϵ $R \times R$ if they are in a *REQUIRES* relationship.

4.2 Empirical Analysis Studies

In this section, we formulate two Empirical Analysis Studies, ESA 1 and ESA 2, to study the above two problems, respectively. We aim to analyze and compare Bidirectional Encoder Representations from Transformers (BERT), and Active Learning (AL) - that are proven to be of interest in general and were pre-evaluated toward their applicability to the stated problems - with traditional ML. For the two studies, we examine the (F1) accuracy, yet analyze the ROI of the whole process of DA simultaneously.

EAS 1: We compare supervised classification algorithms: Naive Bayes (NB) and Random Forest (RF)- two ML algorithms, successfully and prominently used for text classification[16] - with a fine-tuned BERT model [12]. The analysis was performed for varying training set size and how this impacts F1 accuracy and ROI.

BERT (Bidirectional Encoder Representations from Transformers) [12] is a recent technique published by researchers from Google. BERT is applying bidirectional training of Transformer, a popular attention model, to language modeling, which claims to be state-of-the-art for NLP tasks. In this study scenario, we explore the question, "how does fine-tune BERT compare with traditional algorithms on an economical scale?" by comparing models' effectiveness with incurred ROI.

EAS 2: Random sampling (Passive Learning) randomly selects a training set - referred to as *Baseline* in the rest of the paper. Active Learning selects most informative instances using various sampling techniques such as MinMargin, LeastConfidence etc.[20]. We compare *Baseline* with AL using RF as a classifier for this scenario. The analysis was done by adding a few training samples in every iteration concurrently to classify the unlabeled instances to analyze F1 (accuracy) and ROI.

Active Learning (AL) is a ML method that guides the selection of instances to be labeled by an oracle (e.g., human domain expert or

a program) [20]. While this mechanism has been proven to positively address the question, "can machines learn with fewer labeled training instances if they are allowed to ask questions?", through this exploration, we try to answer the question,"can machines learn more economically if they are allowed to ask questions?" [21] by measuring the ROI against the accuracy of the investment.

4.3 Data

The online bug tracking system Bugzilla [2] is widely used in open-source software development. New requirements are logged into these systems in the form issue reports [22] [5] which help software developers to track them for effective implementation [23], testing, and release planning.

In Bugzilla, feature requests are a specific type of issue that is typically tagged as "enhancement" [1]. We retrieved these feature requests or requirements from *Firefox* - a project from Mozilla family of products - using the search engine in the Bugzilla issue tracking system and exported all the related fields such as Title, Type, Priority, Product, Depends_on, and See_also.

Data collection Collecting data from Bugzilla was a substantial effort that was carried out in multiple rounds. We collected 3,704 enhancements from *Firefox* using REST API through a python script such that each one of the enhancements considered for retrieval is dependent on at least another one in the dataset. The data spanned from 08/05/2001 to 09/08/2019.

Data preparation The complete data was analyzed to eliminate special characters and numbers. Then dependent requirement pairs were created based on the depends_on (interpreted as *REQUIRES* dependency) field information for each one of the enhancements. Requirements with no dependency between them were paired to generate *INDEPENDENT* class dataset. Further, sentence pairs that had fewer than three words in them were filtered out resulting in 3,373 *REQUIRES*, 219 *OTHER* and 21,358 *INDEPENDENT* pairs.

Pre-processing and feature extraction: The data was first processed to eliminate stop words and then lemmatized following the traditional NLP pipeline [3]. For supervised and AL ML, we used the Bag Of Words (BOW) [17] feature extraction method, which groups textual elements as tokens. For applying BERT, we retained sentence pairs in their original form (without stop word removal and lemmatization).

Classifiers: Based on the initial results, we chose NB and RF for further study as SVM did not yield good results. The data was split into train and test (80:20) and balanced between classes for this study. Also, hyperparameter tuning was performed and the results for 10-fold cross-validation were computed followed by testing (on unseen data).

To fine-tune BERT model, we used *NextSentencePrediction*[1], a sentence pair classification pre-trained BERT model, and further fine-tuned it for the RDA specific dataset on Tesla K80 GPU on Google Colab[2].

4.4 ROI Modeling

4.4.1 ***EAS1.*** The classification algorithms such as RF and NB, have been explored in NLP based SE problems. These algorithms are driven by the feature extraction aspect to a great extent. Thus, could influence their effectiveness on classification outcomes. However, feature extraction is problem specific and incurs substantial cost and access to domain expertise.

On the other hand, BERT [12], a state-of-the-art NLP language modeling technique eliminates the need for feature extraction since it is a language model based on deep learning. BERT, pre-trained on a large text corpus, can be fine-tuned on specific tasks by providing only a small amount of domain-specific data.

In this empirical analysis, we conducted classification by utilizing a fraction of the whole dataset for training and testing for a small fixed data set. This was repeated by slowly increasing the fraction of the training set and results were captured.

During every classification, $Cost$ and $Benefit$ were computed using various parameters explained in Table 1. $Cost$ is the sum of the data processing costs ($(C_{dg} + C_{pp} + C_e + C_l)/60$) (in hours) for a fraction (N%) of training set. This is further translated into dollar cost based on hourly charges ($C_{resource}$) of H human resources.

$$Cost = N\% * \frac{(C_{dg} + C_{pp} + C_e + C_l)}{60} * H * C_{resource} \quad (2)$$

$Return$ computations for RDA, assumes reward (B_{reward}) for identifying the dependent requirements (TP) while penalizing ($B_{penalty}$) instances that were falsely identified as independent (FN).

$$Benefit = TP * B_{reward} - FN * B_{penalty} \quad (3)$$

Table 2: Parameter settings for the two empirical analysis scenarios

Parameters	Values
$C_{fixed} = C_{dg} + C_{pp} + C_e$	1 min/sample
C_l	0.5 min/sample
$C_{resource}$	$400/hr
H	1
N	4,586
B_{reward}	$500/TP
$B_{penalty}$	$500/FN
$BF1_{iteration}$	$=F_{cur} - F_{prev}$
$PValue$	$10,000 per percent F1 improvement

4.4.2 ***EAS 2.*** In this empirical analysis, we compared AL with a traditional random sampling based classification- *Baseline* - using the RF ML algorithm.

Beginning with 60 training samples of each class (*REQUIRES, INDEPENDENT and OTHER)*, we developed multi-class classifiers for both AL and Baseline for this empirical study scenario. When AL used MinMargin sampling technique[3] to identify 20[4] most uncertain instance (requirement pair) for oracle to label, baseline randomly selected 20 instances and added to the training set along with their label, thus kept the two approaches comparable in all the 20 iterations. Since data is already labeled, for AL, we pretend

[1]https://huggingface.co/transformers/model_doc/bert.html#bertfornextsentenceprediction
[2]https://colab.research.google.com/

[3]MinMargin sampling technique performed well compared to Least Confidence and Entropy, thus we utilized MinMargin for this study
[4]Tests were also performed with#samples = 10, 15 and 20 but in this study we will discuss results related to #samples=20

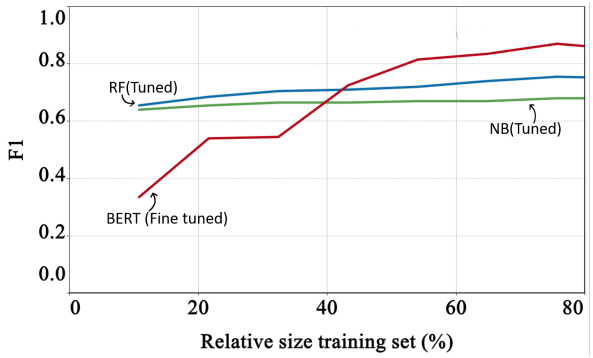


Figure 2: F1 score plot for NB, RF and BERT trained over increasing training set size, F1 improves, but plateaus beyond a certain point

they are unlabeled until queried and labeled by a simulated oracle in this scenario.

Cost is determined by first computing the sum of total processing time in person hours (= *Cost*) taken for data processing ($C_{fixed} = C_{dg} + C_{pp} + C_e$)), labeling ($C_l$) of train set ($N_{train}$) and data processing cost (C_{fixed}) for testing. This is further translated into dollar cost (=C_{total}) based on hourly charges ($C_{resource}$) of H human resources.

$$Cost = \frac{N_{train} * (C_{fixed} + C_l) + N_{test} * C_{fixed}}{60}$$

$$C_{total} = Cost * H * C_{resource} \quad (4)$$

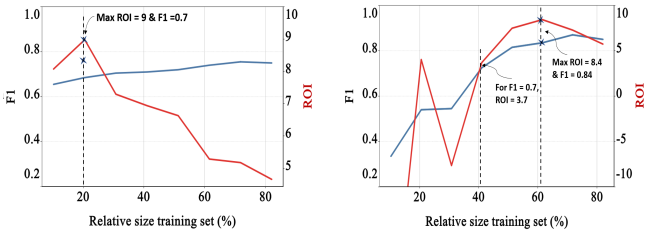
Likewise, $Benefit$ is defined as the monetary value associated with a 1% improvement in F1 score ($BF1_{iteration}$) between subsequent iterations.

$$Benefit = BF1_{iteration} * PValue \quad (5)$$

5 RESULTS

In the real-world, cost and benefit values are hard to get and are uncertain. All the results presented in this section are based on the parameter settings in Table 2. The settings reflect practical experience, but the results are sensitive to these settings. We claim that the principal arguments made in our paper are independent of these settings.

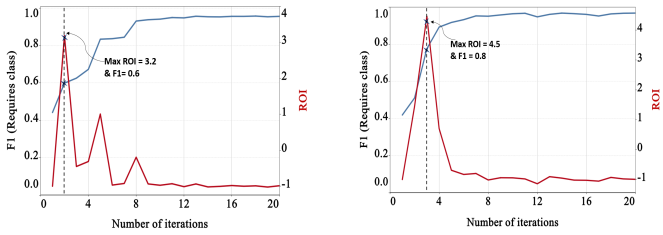
5.1 EAS 1



(a) F1 vs ROI for Random Forest **(b) F1 vs ROI for Fine tuned BERT**

Figure 3: Empirical Analysis Scenario 1 (EAS 1)

Figure 2 provides the "accuracy only view" and shows that F1 gradually increases with the increasing training size for the three ML algorithms: NB, RF, and BERT. However, all three ML algorithms reach a saturation towards larger training set sizes. While BERT performed exceptionally well when training set size exceeded 42%, it could have been ideal to pre-determine "How much training is enough?". Thus we selected the top two classifiers (Figure 2): BERT and RF and applied the monetary values (Table 2) for the various cost and benefit factors defined in Table 1 and computed the ROI function.



(a) F1 vs ROI for Baseline **(b) F1 vs ROI for AL**

Figure 4: Empirical Analysis Scenario 2 (EAS2)

Figure 3a and 3b show the results for RF and BERT, respectively. The ROI behaviour is not monotonous and peaks for both cases. Although RF classification achieved the highest ROI with just 20% of training set and accuracy of F1 = 0.7, highest F1 value of 0.75 was achieved along with the lowest ROI of 4.7.

For RF classification and applying ROI arguments, learning can be stopped with 20% of the training set.

Now looking at BERT classification, best ROI-driven results, F1 = 0.84 and an ROI = 8.43, were achieved with the 60% training set. Although F1 rose to 0.9 with 70% training set size, ROI dropped to 7.27. For the recommendation of 20% of training set size, ROI has a local optimum. BERT in general performs well on the F1, however, is it worth the ROI? needs to be explored.

For training set sizes of at least 40% of the size of the whole set, BERT performed better than RF in terms of both accuracy and ROI.

5.2 EAS 2

We analyzed the ROI for *Baseline* against AL for classifying the *REQUIRES* class. The results are shown in Figure 4a and Figure 4b. Similar to EAS 1, we applied the values from Table 2 and equations (4) and (5) to compute cost and benefit at every iteration for both the approaches. For the *Baseline* approach, ROI peaked at 3.2 and F1 = 0.6, in the very 2nd iteration. Onwards, ROI drastically decreased which indicated lesser value for increasing training set by random sampling (*Baseline*) method.

Similar behavior was observed for the AL approach. shown in Figure 4b. The peak here was after three iterations with values ROI = 4.5 and F1 = 0.8.

Both Baseline and AL showed best ROI performance in the early iterations. However, higher F1 accuracy needs additional human resources, thus, it substantially reduce ROI further on.

6 DISCUSSION

For the problem of RDA, we explored the potential value of ROI-driven decisions. When chasing higher accuracy, there is a risk of over analyzing empirical data. In the sense that the added value of increased accuracy is not justifiable by the additional effort needed for it.

6.1 What does a high or low ROI mean for DA?

If available, a high ROI ratio indicates that there is a substantial benefit expected from following the recommendations derived from DA. Assuming that the ROI-driven suggestions are really implemented, the small improvements achieved for solving the decision problems with high impact could justify the effort invested. Analysis related to effort and benefit, targeting high ROI also implies simplicity first. Advanced methods are needed, but they are hard to justify practical application, if a similar type of insight could be reached from a much simpler analysis, e.g., from descriptive statistics.

6.2 What is the risk of ignoring analysis?

The calculation of ROI is based on the value and effort estimates and thus only provides an approximation. In all types of exploratory data analysis, the emphasis is mainly on creating new research hypotheses or validating existing assumptions. In these cases, the notion of ROI is not the main concern. Also, estimates for value and effort needed are highly dependent, hence, the ROI might only serve as a soft recommendation. On the other hand, whenever the ROI can be determined as a reasonable estimate, even after using intervals of best and worst-case performances, then, ignoring ROI means to potentially waste effort for analysis that does not pay off the investment made. For EAS 1, if the training size set was limited to 30%, RF could be considered as a better choice over BERT. However, with the possibility to increase the training set size, the BERT approach could be favored.

7 CONCLUSIONS AND FUTURE WORK

For empirical studies, through this research, we envision to complement Data Analytics with ROI analysis to increase its impact. To validate the need, we performed an analysis of accepted papers of ESEM conferences between 2015 and 2019 and found that 51 papers out of 190 papers (27%) were addressing some form of DA. Among them, 39% included some consideration of cost, value or benefit. However, none of them directly explored or discussed ROI or used cost-benefit analysis to decide the degree of DA needed. Firstly, from a decision-making perspective, choosing one out of many techniques is an arduous task in itself. Additionally, for a given technique, deciding the termination mark for analysis means exponentially expanding the scope from one to multiple criteria.

Beyond accuracy, reflecting the benefit, it is important to look into the investment as well. Exclusively looking into the different aspects of accuracy is important, but does not provide the full picture as it ignores the effort consumption and impact. Effort estimation is well studied, but not so much is known about the prediction of value [6]. Even rough estimates may be helpful to decide how much further investment into DA is reasonable. To make this agenda successful, economical, business, and social concepts need to be taken into account, apart from just the technical aspects.

REFERENCES

[1] [n.d.]. BugFields. https://wiki.mozilla.org/BMO/UserGuide/BugFields#bug_type
[2] [n.d.]. Bugzilla, a bug-tracking system. https://www.bugzilla.org/
[3] Andres Arellano, Edward Zontek-Carney, and Mark A Austin. 2015. Frameworks for natural language processing of textual requirements. *International Journal On Advances in Systems and Measurements* 8 (2015), 230–240.
[4] Andrew Begel and Thomas Zimmermann. 2014. Analyze this! 145 questions for data scientists in software engineering. In *ICSE*. 12–23.
[5] Tanmay Bhowmik and Sandeep Reddivari. 2015. Resolution trend of just-in-time requirements in open source software development. In *2015 IEEE Workshop on Just-In-Time Requirements Engineering (JITRE)*. IEEE, 17–20.
[6] Stefan Biffl, Aybuke Aurum, Barry Boehm, Hakan Erdogmus, and Paul Grünbacher. 2006. *Value-based software engineering*. Springer Science & Biz Media.
[7] Christian Bird, Tim Menzies, and Thomas Zimmermann. 2015. *The art and science of analyzing software data*. Elsevier.
[8] Barry Boehm, Ricardo Valerdi, and Eric Honour. 2008. The ROI of systems engineering: Some quantitative results for software-intensive systems. *Systems Engineering* 11, 3 (2008), 221–234.
[9] Jane Cleland-Huang, Grant Zemont, and Wiktor Lukasik. 2004. A heterogeneous solution for improving the return on investment of requirements traceability. In *Proc. 12th IEEE Requirements Engineering Conference, 2004.* IEEE, 230–239.
[10] Antonio Costa, Giulio Pasta, and Giovanni Bergamaschi. 2005. Intraoral hard and soft tissue depths for temporary anchorage devices. In *Seminars in orthodontics*, Vol. 11. Elsevier, 10–15.
[11] Johan et al. Dag. 2002. A feasibility study of automated natural language requirements analysis in market-driven development. *Requirements Engineering* 7, 1 (2002), 20–33.
[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
[13] Hakan Erdogmus, John Favaro, and Wolfgang Strigel. 2004. Return on investment. *IEEE Software* 21, 3 (2004), 18–22.
[14] Mascia Ferrari et al. 2005. ROI in text mining projects. *WIT Transactions on State-of-the-art in Science and Engineering* 17 (2005).
[15] Jin Guo, Jinghui Cheng, and Jane Cleland-Huang. 2017. Semantically enhanced software traceability using deep learning techniques. In *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*. IEEE, 3–14.
[16] Christopher Manning, Prabhakar Raghavan, and Hinrich Schütze. 2010. Introduction to information retrieval. *Natural Language Engineering* 16, 1 (2010), 100–103.
[17] Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, Vol. 242. New Jersey, USA, 133–142.
[18] G Ruhe and M Nayebi. 2016. What counts is decisions, not numbers — Toward an analytics design sheet. In *Perspectives on Data Science for SE*. Elsevier, 111–114.
[19] R. Samer, M. Stettinger, M. Atas, A. Felfernig, G. Ruhe, and G. Deshpande. 2019. New Approaches to the Identification of Dependencies between Requirements. In *31st Conference on Tools with Artificial Intelligence (ICTAI '19)*. ACM, 6.
[20] Burr Settles. 2009. *Active learning literature survey*. Technical Report. University of Wisconsin-Madison Department of Computer Sciences.
[21] Burr Settles. 2011. From theories to queries: Active learning in practice. In *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010*. 1–18.
[22] Lin Shi, Celia Chen, Qing Wang, Shoubin Li, and Barry Boehm. 2017. Understanding feature requests by leveraging fuzzy method and linguistic analysis. In *Proc. of the 32nd Conf. on ASE*. IEEE Press, 440–450.
[23] Yonghee Shin, Jane Huffman Hayes, and Jane Cleland-Huang. 2015. Guidelines for benchmarking automated software traceability techniques. In *Proceedings of the 8th Int. Symposium on Software and Systems Traceability*. IEEE Press, 61–67.
[24] Rini Van Solingen. 2004. Measuring the ROI of software process improvement. *IEEE software* 21, 3 (2004), 32–38.
[25] Roel J Wieringa. 2014. *Design science methodology for information systems and software engineering*. Springer.
[26] Wei Zhang, Hong Mei, and Haiyan Zhao. 2005. A feature-oriented approach to modeling requirements dependencies. In *13th IEEE Conf. Requirements Engineering*. IEEE, 273–282.