

“How not to do it”: Anti-patterns for Data Science in Software Engineering

Tim Menzies, NC State University, tim.menzies@gmail.com

DESCRIPTION OF TOPIC: Many books and papers describe how to do data science. While those texts are useful, it can also be important to reflect on anti-patterns; i.e. common classes of errors seen when large communities of researchers and commercial software engineers use, and misuse data mining tools. This technical briefing will present those errors and show how to avoid them. The errors will be presented at four levels:

- For researchers:
 1. *Prior to the study*: how not to design a study, how to miss important results;
 2. *After the study*: how to make unfounded conclusions, how to get a paper rejected from EMSE journal, ICSE, ASE;
 3. *Research directions*: speculations on how to use these data mining anti-patterns as a novel research direction for data science.
- For industrial practitioners:
 4. Ways to design your data mining team in order to ensure the failure of your project.

Material for these points will come from the latest research in SE analytics:

- At ASE'15, the Actionable Analytics community was convened by Tim Menzies to discuss all that was right, and all that was wrong, with SE data science. This tech briefing will present the anti-patterns found at that workshop.
- At Dagstuhl'14, the “Software Development Analytics” workshop convened by Tim Menzies (and Harald Gall, Laurie Williams and Thomas Zimmermann, see <https://goo.gl/hvdAtS>) distilled the collective wisdom of good, and bad, practice in data science of SE down to a list of 69 “mantras”. The mantras relating to anti-patterns will be presented here.
- Tim Menzies is an associate editor and reviewer for IEEE TSE, ASE journal, EMSE, and the Software quality journal. Each year he rejects dozens of papers about data science in SE-- usually for all the same reasons (those reasons will be presented at this technical briefing).
- Other material for this workshop will come from joint work with Tom Zimmermann, specifically, the relevant parts of the Inductive Engineering Manifesto¹ and Zimmermann's more recent work on how to staff a data mining project for SE².



¹ Tim Menzies, Christian Bird, Thomas Zimmermann, Wolfram Schulte, and Ekrem Kocaganeli. 2011. The inductive software engineering manifesto: principles for industrial data mining. In *Proceedings of the International Workshop on Machine Learning Technologies in Software Engineering* (MALETS '11). ACM, New York, NY, USA, 19-26. DOI=<http://dx.doi.org/10.1145/2070821.2070824>

² Miryung Kim and Thomas Zimmermann and Robert DeLine and Andrew Begel, The Emerging Role of Data Scientists on Software Development Teams, MSR-TR-2015-30, Microsoft Research, <http://research.microsoft.com/apps/pubs/default.aspx?id=242286>, 2015

WHY THE TOPIC WOULD BE OF INTEREST TO A BROAD SECTION OF THE SE COMMUNITY:

To say the least, a lot of industrial workers and researchers in SE use data mining. Data mining is widely used in the software industry. Graduates from data mining master degrees are in high demand. Any recent conference on SE contains dozens of papers on data science. In the SE journals of (TSE, the ASE journal, EMSE), data mining papers are the (first, first, and fourth) most cited papers^{3,4}.

While there are many Data Mining 101 tutorials for software engineering, there is a dearth of material devoted to advanced users of these tools. The main selling point of this tech briefing is the concept of “Anti-Pattern”. As we mature as a field, it is becoming realized by the broader community that we need only share our **successes- but also our failures**. Hence, it is important to report “how not to do it”.

As to why this particular topic would be seen as different to other ICSE’16 tutorials, I checked the ICSE’15 roster and offer the following notes on how this tutorial differs to some of the other proposals you might receive:

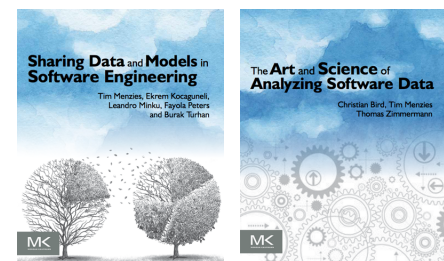
- **Big(ger) Data in Software Engineering:** focused on the technical dimensions enabled by the massive data sets that are now widely available. That tech briefing was an exciting exploration of new possibilities while this tech briefing is a more reflective, more careful, catalog of the certain weaknesses with our current approaches
- **The Use of Text Retrieval and Natural Language Processing in Software Engineering:** focused on an important topic (text mining). The scope of this tech briefing is far broader.
- **The Art and Science of Analyzing Software Data: Quantitative Methods:** presented some of the latest findings and methods by Leandro, Fayola and me. That briefing was a good data mining 101 introduction. But this briefing is more a cautionary tale about what happens when data mining methods are over-used.

NAME AND SHORT BIOGRAPHY OF THE SPEAKER: Tim Menzies (Ph.D., UNSW, 1995) is a full Professor in CS at North Carolina State University where he teaches software engineering and automated software engineering. His research relates to synergies between human and artificial intelligence, with particular application to data mining for software engineering.

He is the author of over 230 referred publications; and is one of the 100 most cited authors in software engineering out of over 80,000 researchers (<http://goo.gl/BnFJs>). In his career, he has been a lead researcher on projects for NSF, NIJ, DoD, NASA, USDA, as well as joint research work with private companies.

Working closely with the software engineering community, Dr. Menzies has recently co-edited and released two summaries of the state of the art in data science for software engineering. These two books, shown at right, are published by Morgan Kaufmann. A third book (“Perspectives on Data Science for Software Engineering”) is being developed with a particular focus on industrial practitioners.

Prof. Menzies is the co-founder of the PROMISE conference series devoted to reproducible experiments in software engineering (<http://openscience.us/repo>). He is an associate editor of IEEE Transactions on Software Engineering, Empirical Software Engineering, the Automated Software Engineering Journal and the Software Quality Journal. In 2015, he served as co-chair for the ICSE’15 NIER track. In 2016, he serves as co-general chair of ICMSE’16. For more, see his vita (<http://goo.gl/8eNhYM>) or his list of publications <https://goo.gl/qNQA1q> or his home page <http://menzies.us>.



³ BTW, just to speak to my qualifications to present this material, those particular three papers were written by me.

⁴ Over the last decade, measured in cites/year, as documented by Google Scholar, as of Nov 2015.

Not in this talk: not what everyone else is talking about

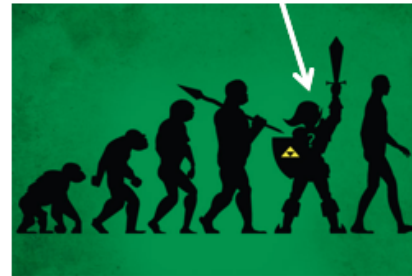
- Principles for designing case studies
- Visualizations
- Data mining
- Big Data
- Qualitative methods see parts 1+2



1

02:00

The talk... adding in some missing bits



NC STATE UNIVERSITY slides.net

2

02:00

So many data repositories

Repository	URL
Big Prediction Dataset	http://big.itsf.net
Edgex Big Data	www.edgex.net
FLCOSMOS	http://flcosmos.org
FLCOSMOS	http://flcosmos.org
International Software Benchmarking Standards Group (ISBSG)	www.isbsg.org
ISBSG	www.isbsg.org
PRODIGE	http://prodige.cs.gatech.edu
Qualitas Corpus	http://qualitas.corpus.com
Software Artifact Repository	http://sar.srli.edu
SourceForge Research Data	http://sourceforge.net
SourceForge Project	http://sourceforge.net
Tokubuku	www.tokubuku.com
Ullmann Dataset Database	http://null.ullmann.org

- What's next?
- What tools would we need for an "debate"-oriented repository?

NC STATE UNIVERSITY slides.net

3

3

02:00

Let's talk tools



To design those tools, ask:

- What problems are seen when people try to share data and conclusions?
- What minimal data structures address those problems?

NC STATE UNIVERSITY slides.net

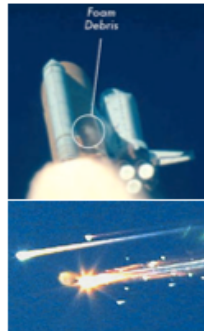
4

02:00

Models have "certification envelopes"

Bad things happen when you stretch the envelope

- Columbia ice strike
 - Size: 1200 m²
 - Speed: 477 mpg (relative to vehicle)
- Certified as "safe" by the CRATER micro-meteorite model.
 - A experiment in CRATER's DB:
 - Size: 30m²
 - Speed: under 100 mpg
- Columbia, and crew, dies on re-entry
- Lesson: conclusions should come with a "certification envelope"
 - If new tests outside of the envelope of the training set
 - Raise an alert



NC STATE UNIVERSITY slides.net

Goals matter

Learners learn for X, users want Y

- Learners work this way
 - Users want it that way
- Waste of time learning models users do not want
 - Better to tune learning methods to goals of users
- Enter search-based software engineering
 - Multi-goal optimization



Figure 25.3 Different users value different things. Note that some of these goals are defined in terms of Figure 25.1.

Some goals relate to aspects of data prediction:

- Minimize critical system risk and/or safety and/or security. This is not a goal in itself, but it is a goal that can be achieved by many different means. It is a goal that can be achieved by many different means.
- Cost-effectiveness. This is a goal that can be achieved by many different means. It is a goal that can be achieved by many different means.
- Accuracy. This is a goal that can be achieved by many different means. It is a goal that can be achieved by many different means.
- Speed. This is a goal that can be achieved by many different means. It is a goal that can be achieved by many different means.

Related data prediction user goals that combine data prediction with other non-data factors:

- Accuracy & Speed [12]. (Holland & Wierwille [1982] and Holland et al. [1972] say that data prediction should maximize accuracy, i.e., find the best data of value that contains the most bugs).
- In other words, Kupper et al. are concerned about accuracy. They don't care about speed for performance reasons of the code that are not relevant to them [2013]. Such accuracy does not play a role in their model, but it is a goal that can be achieved by many different means.
- In other words, Kupper et al. are concerned about accuracy. They don't care about speed for performance reasons of the code that are not relevant to them [2013]. Such accuracy does not play a role in their model, but it is a goal that can be achieved by many different means.
- In other words, Kupper et al. are concerned about accuracy. They don't care about speed for performance reasons of the code that are not relevant to them [2013]. Such accuracy does not play a role in their model, but it is a goal that can be achieved by many different means.

All the other measures relate to the tendency of a prediction to find something. Another goal of accuracy would be to check the variability of that prediction.


6. In fact, only a representative of all models, such as finding and finding accuracy using the coefficient of variation ($CV = \frac{\sigma}{\mu}$) or using the variance. They defined reproducibility as $\frac{\sigma}{\mu}$ [1982].

NC STATE UNIVERSITY slides.net

slides= tiny.cc/se15

Locality matters

Not general models ,but general methods for local models



- Devanbu et al. ASE'11
Ecological Inference
- Betternburg et al. MSR'12
Think local, act global,
- Menzies et al. TSE'13
Local versus Global learning,
- Yang et al. IST'13
Handling local bias,
- Minku et al. ICSE'14
Best Use of Cross-Company Data


NC STATE UNIVERSITY slides= tiny.cc/se15 7 02:00

slides= tiny.cc/se15

Sharing matters

Given enough eyes, all bugs are shallow

When (2013)	What
Mar 15	"Better cross-company learning" accepted to MSR'13
Mar 29	Camera-ready submitted
?Apr 10	Pre-prints go on-line
Apr 29	Hyeonmin Jeon, graduate student at Pusan Natl. Univ. emailed us: can't reproduce result
May 4	Fayola Peters, checking code, found error. Manic week of experiments follow
May 11	We conclude results definitely wrong
May 12	Email MSR organizers. Our penalty? Present paper and its error.



- How was the error found so fast?
 - Open science

NC STATE UNIVERSITY slides= tiny.cc/se15 8 02:00

slides= tiny.cc/se15

Compression and privacy matter

Squeezing and secrets

- Facebook, Google, Netflix etc
- Small X% of all users are subjects in continual experiments: testing new features
- Data from studies, retained indefinitely, warehoused
 - Problems with volume (needs compression)
 - Problems with confidentiality (needs privacy)
- If I want to challenge the conclusions made by Facebook, Google, Netflix, etc
 - I need to be able to access, privately, that data
 - (needs trusted sharing)

NC STATE UNIVERSITY slides= tiny.cc/se15 9 02:00

slides= tiny.cc/se15

Lessons learned

What matters?

- Certification envelopes (when not to trust conclusions)
- Goals matter (not everything is "classification")
- Locality matters (when their conclusions do not hold for you)
- Need "streaming tools" (continually stream over a never ending sequence of new data)
- Need repair tools (to fix broken ideas)
- Verification matters (sooner or later, we all screw up)
- Need to transfer data (get by with a little help from your friends)
- Need compression tools (to save space)
- Need privacy tools (so you can share)

NC STATE UNIVERSITY slides= tiny.cc/se15 10 02:00