Data Science in Software Engineering

Increasing Software Productivity

Dr. Eray Tüzün
Department of Computer Engineering
Bilkent University
@eraytuzun

@tuzuneray
http://linkedin.com/in/tuzuneray

@eraytuzun
http://twitter.com/eraytuzun
Software is Eating the World

Software Updates

Our cars regularly receive over-the-air software updates that add new features and enhance existing ones over Wi-Fi.

When updates become available, you’ll receive a notification on your center touchscreen display, with the option to install the update immediately or schedule for later. To ensure the fastest and most reliable delivery of software updates, connect your car to Wi-Fi.

Frequently Asked Questions

What are over-the-air software updates?
Over-the-air software updates introduce new features and updates to your car—making your car safer and more capable over time.

Where can I find information on the latest software updates?
You can find the latest information on your car’s software in the release notes. Tap the Tesla ‘T’ at the top center of your car’s touchscreen to access. After a new window opens, tap ‘Release Notes’ on the bottom right hand side of the page.

How can I confirm which software version I have?
To check the latest software version on your car, tap the Tesla ‘T’ at the top center of your car’s touchscreen. This will display a summary of your car’s software, including version number, in the bottom left hand corner. This summary also includes your vehicle avatar, battery configuration and VIN.

When do software updates become available?
Software updates occur on a rolling basis. If your car is due for an update, Tesla will notify you through an alert on your car’s touchscreen and Tesla mobile app. The notification will also tell you the estimated time required to complete the update.

Can I drive my car during a software update?
During a software update, you are unable to drive your car as a safety measure. We recommend that you park your car in a safe location and complete the update.
Software Engineering Goals

- Increase Productivity
- Improve Quality
Why is software development productivity important?

Large-Scale Software Engineering
- Fixed release schedule.
- Fixed resources (time and money)
- Way too much work to finish.
- Develop more valuable products for lower costs ($)
- Decrease time-to-market for competition
- Hard to find (good) developers
How to achieve Productivity

Doing things right vs. Doing the Right Things
Engineering Complexity

7.300,000 kg, 
2.5 million rivets, all of them as old as the tower itself, hold the latticed structure together.
Engineering Complexity

5-10 million lines of code
The failure of any line of code could lead to total system failure
Space of inputs even to the smallest app on the phone $> 10^{80}$
Software development is complex
Software Practitioners rely on their prior experiences to plan software project projects, fix bugs, prioritize testing, etc.
What are key “Decisions” we are after?

• Guess the location of undetected bugs
• Effort Estimation
• Who should review my code?
• Finding duplicated bugs
• Finding potential reopen bugs
• Who is the Most Valuable Developer in my team?
• Which features are mostly used in my application?
• Who in the team is most likely to leave in the next 2-4 weakest?

• When will this project be ready to ship?
• Which components of our application most need to be tested?
• Who should fix this bug?
• What parts of my API do people find hardest to use?
• Will My Patch Be Accepted?
• Finding code clones
• Why my bugs are reopened?
• ....
Decision Making

Data Driven vs Gut Feeling
Data Science

The field of study that uses various **methods** to extract useful insights and knowledge from the **data** to make data-driven **decisions**.

Methods can include/require; domain expertise, programming skills (i.e. scripting to process data), statistical modelling (i.e. machine learning algorithms), visualization techniques.
What is Data Science?

Data Science makes use of AI, ML, DL

Data Science in Software Engineering

Transform static record-keeping SE data to **active** data
Make SE data actionable by uncovering hidden **patterns** and **trends**
Gain empirically-based understanding of software development
Predict, plan, and understand various aspects of a project
Support future development and project management activities
Definitions

“Software analytics is analytics on software data for managers and software engineers with the aim of empowering software development individuals and teams to gain and share insight from their data to make better decisions.”


“Software analytics is to enable software practitioners to perform data exploration and analysis in order to obtain insightful and actionable information for data-driven tasks around software and services (and software practitioners typically include software developers, testers, usability engineers, and managers, etc.).”

Terminology

• Data Science in Software Engineering
• Software (Data) Analytics
• Software Intelligence
• Software Development Analytics
• Mining Software Repositories
What do we mean by “data” in Software Engineering?

• Large amount of artefacts are generated in the software development process
• Increased amount of data available in software archives through large open-source projects
Where is the Data?
How does the data look like?

- [https://github.com/apache/hadoop/pulls](https://github.com/apache/hadoop/pulls)
Types of SE Data

- Runtime traces
- Program logs
- System events
- Perf counters

- Usage log
- User surveys
- Online forum posts
- Blog & Twitter

- Source code
- Bug history
- Check-in history
- Test cases
How big is the data?

• Size
  • Linux >10M LOC

• Developer
  • >1K in Windows

• Code comments
  • 2 Million in Eclipse, 1 million in Mozilla

• Commit logs
  • 222K for Linux, 31K for PostgreSQL

• Bug reports
  • 641K in Mozilla, 18K in Linux, 7K in Apache
Software Data is Connected
Who uses the data?

Target audience – software practitioners

Program Manager
Developer
Management personnel
Designer
Tester
Support engineer
Operation engineer
Usability engineer
Overview of Data Science in SE

- software engineering tasks helped by data mining
  - programming
  - defect detection
  - testing
  - debugging
  - maintenance
  - ...

- Data Science techniques
  - NLP
  - classification
  - association/patterns
  - clustering
  - Deep Learning
  - ...

- software engineering data
  - code bases
  - change history
  - program states
  - structural entities
  - bug reports
  - ...

- Programming, defect detection, testing, debugging, and maintenance are software engineering tasks helped by data mining techniques such as NLP, classification, association/patterns, clustering, and Deep Learning.

- Software engineering data includes code bases, change history, program states, structural entities, and bug reports.
Questions

• How can we make data useful to a wide audience, not just to developers but to anyone involved in software?
• What can we learn from the vast amount of unexplored data?
• How can we learn from incomplete or biased data?
• How can we better tie usage analytics to development analytics?
• When and what lessons can we take from one project and apply to another?
• How can we establish smart data science as a discipline in software engineering practice and research as well as education?
Software Artifacts Traceability Graph
How does RSTrace work?

Create an artifact traceability graph using the available repositories

For an artifact that requires reviewer recommendation

  Run “the algorithm” that assigns scores to developer-artifact pairs

  Sort developers by scores in descending order

  Recommend top-k developers
The Algorithm

We define a new metric called **know-about** that measures the knowledge of a developer about an artifact (e.g. source code file).

Know-about score of a developer for an artifact is calculated as follows:

\[ KA_{Developer-Artifact} = \sum_i \frac{1}{\text{Length}(\text{Path}_i(\text{Developer} - \text{Artifact}))} \]

Know-about score of a developer for a changeset is calculated as follows:

\[ KA_{Developer-Changeset} = \sum_i KA_{Developer-Artifact(i)} \]

where \( \text{Artifact}(i) \in \text{Changeset} \)
Calculate know-about score of Alice and Bob for A.java

Alice: \( \frac{1}{2} + \frac{1}{5} = 0.70 \)

Bob: \( \frac{1}{2} + \frac{1}{3} = 0.83 \)

Bob is a better reviewer for A.java file because his know-about score is higher than Alice’s.
GitHub Bot

Reviewer Recommendation Tool Flow Diagram

<table>
<thead>
<tr>
<th>Developer</th>
<th>Bot</th>
<th>Collaborator</th>
<th>Reviewer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commit changes</td>
<td>Find appropriate reviewers by analyzing the pull request</td>
<td>Request a code review from the recommended reviewer</td>
<td>Review the PR and submit feedback</td>
</tr>
<tr>
<td>Create a new pull request</td>
<td>Write the recommended reviewers as a comment to the PR and mention them.</td>
<td>Requires extra commits?</td>
<td></td>
</tr>
<tr>
<td>YES</td>
<td>NO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Push new commits according to feedback</td>
<td>Decide whether PR is merged or not according to reviewer's comment</td>
<td>Ask opinion about the reviewer suggestion quality</td>
<td>Close PR</td>
</tr>
</tbody>
</table>

Rounding error fixed #2

Closed alice wants to merge 1 commit into mary:master from alice:patch-2

Conversation 2  Commits 1  Checks 0  Files changed 1

Alice commented on Dec 4, 2018

No description provided.

Update SocketInterface.java

RSTraceBot bot commented on Dec 4, 2018

Recommended reviewers: @bob, @mary and @john

Alice closed this on Dec 4, 2018

RSTraceBot bot commented on Dec 4, 2018

Thanks for using the bot. Would you like to take a survey? https://goo.gl/forms/PvrHlLw8cRFBp2cpl
Understand the Application Domain

**Domain Knowledge + Close(r) Inspection**

- Make sure you manually examine the repositories. Do not fully automate the process!
Data Science Pipeline

Question → Data

Data → Wrangle → Clean → Explore

Explore → Pre process → Model → Validate

Validate → Tell the Story → Actionable Insight

Image credit Wolfram Research
Facebook - How Facebook tools learn to fix bugs automatically

GitHub CoPilot

• [https://copilot.github.com](https://copilot.github.com)
References

Selected Literature

“Perspectives on Data Science for Software Engineering”, 1st Edition, Tim Menzies, Laurie Williams, Thomas Zimmermann

Data Science in Software Engineering

Analyze software data

Make informed data-driven decisions

Supporting decision making using facts instead of fortune tellers!

Software Archive

Code  Bug  E-Mail/Chat  User Reviews  Others
Slide Credits

• Tao Xie
• Ahmed Hassan