



guild.ai

Keys to Building Machine Learning Systems

Garrett Smith, Guild AI - April 27-29, 2020

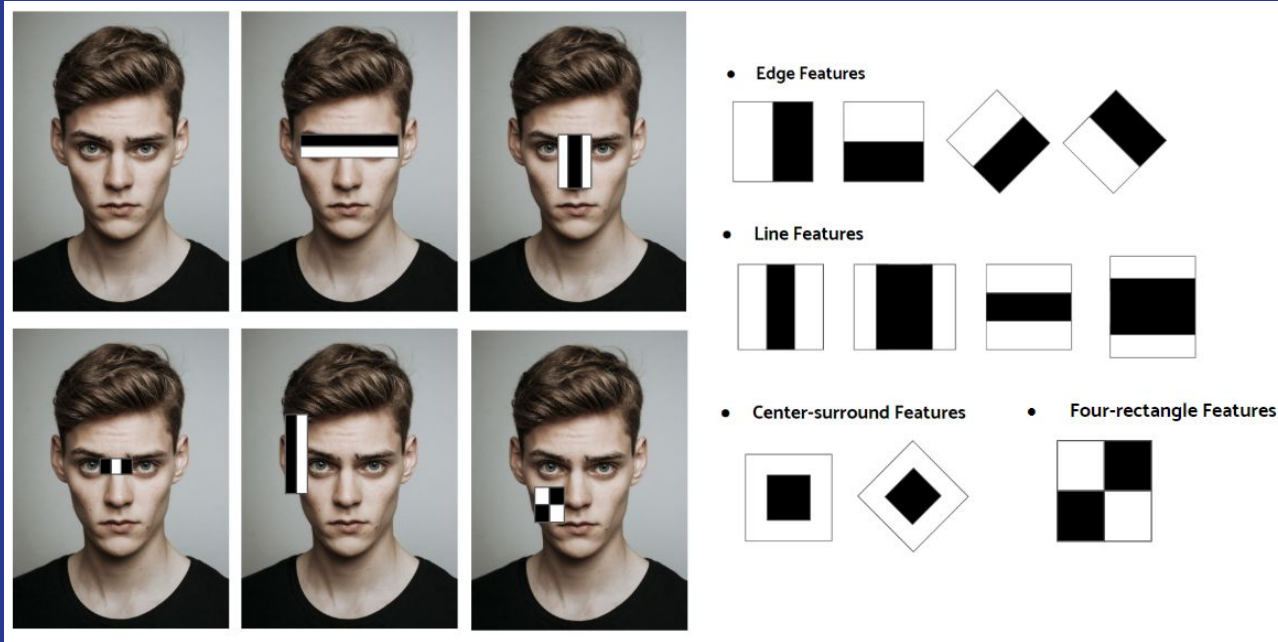
About Me

- Programmer (Python, Erlang, C, C++, Java)
- Founder, Guild AI (ML Engineering Toolkit)
- Founder of Chicago ML (4000 member ML user community)
- Previously CloudBees
 - Architect of CloudBees Jenkins Enterprise
 - Director of PaaS operations
- Stax Networks (Java PaaS, acquired by CloudBees in 2010)

Machine learning lets you
produce software that you
could never write yourself

Machine Learning in Two Slides

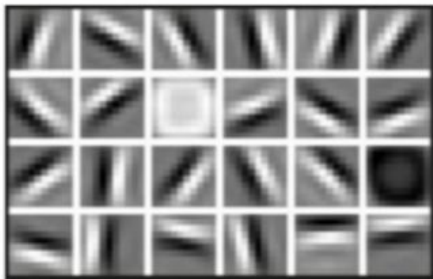
Manual Program (Haar Cascade)



Credit: [Computer Vision for Beginners](#)

Learned Program (Neural Network)

Low Level Features



Lines & Edges

Mid Level Features



Eyes & Nose & Ears

High Level Features



Facial Structure

Credit: [MIT Introduction to Deep Learning](#)

So what's the problem?

AI Adoption in the Enterprise 2020

- 85% evaluating, 15% doing anything
- 50% consider themselves “mature” adopters of AI
- Challenges to adoption
 - Institutional support
 - Use cases
 - Lack of skilled resources
 - Lack of data

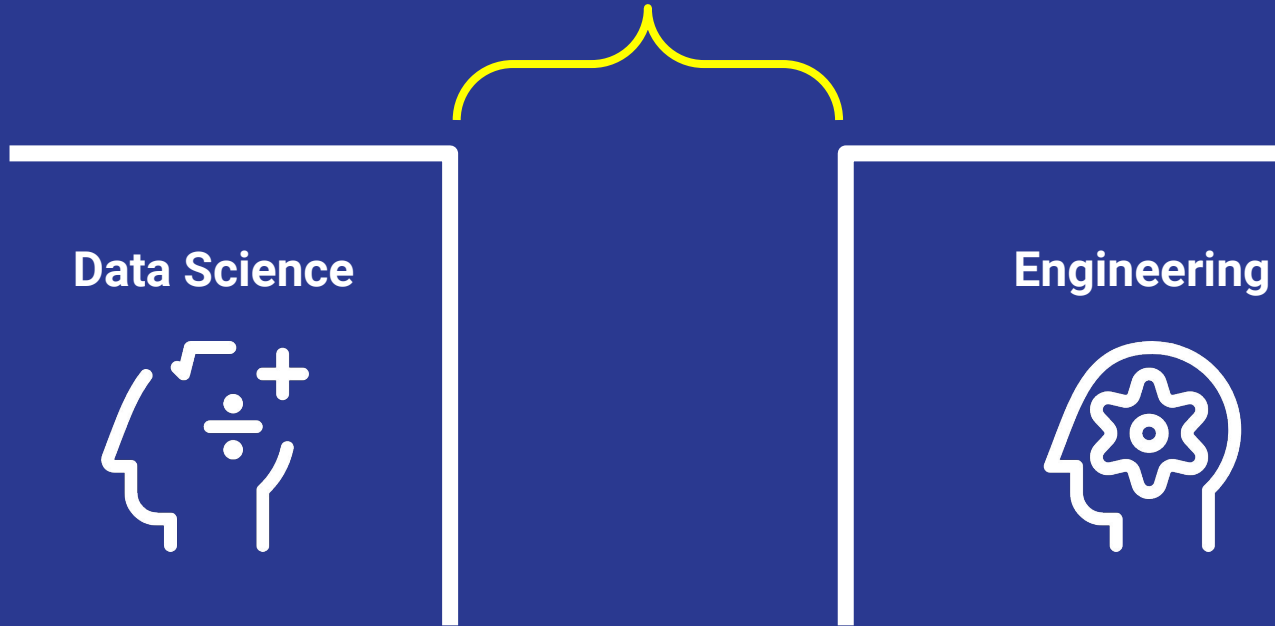
Source: O'Reilly

Assuming institutional
support and a solid
business case...

The problem is NOT data

The problem is NOT the
latest libraries and tools

Fundamental Problem In Most Organizations



Data Science Code vs Traditional Software

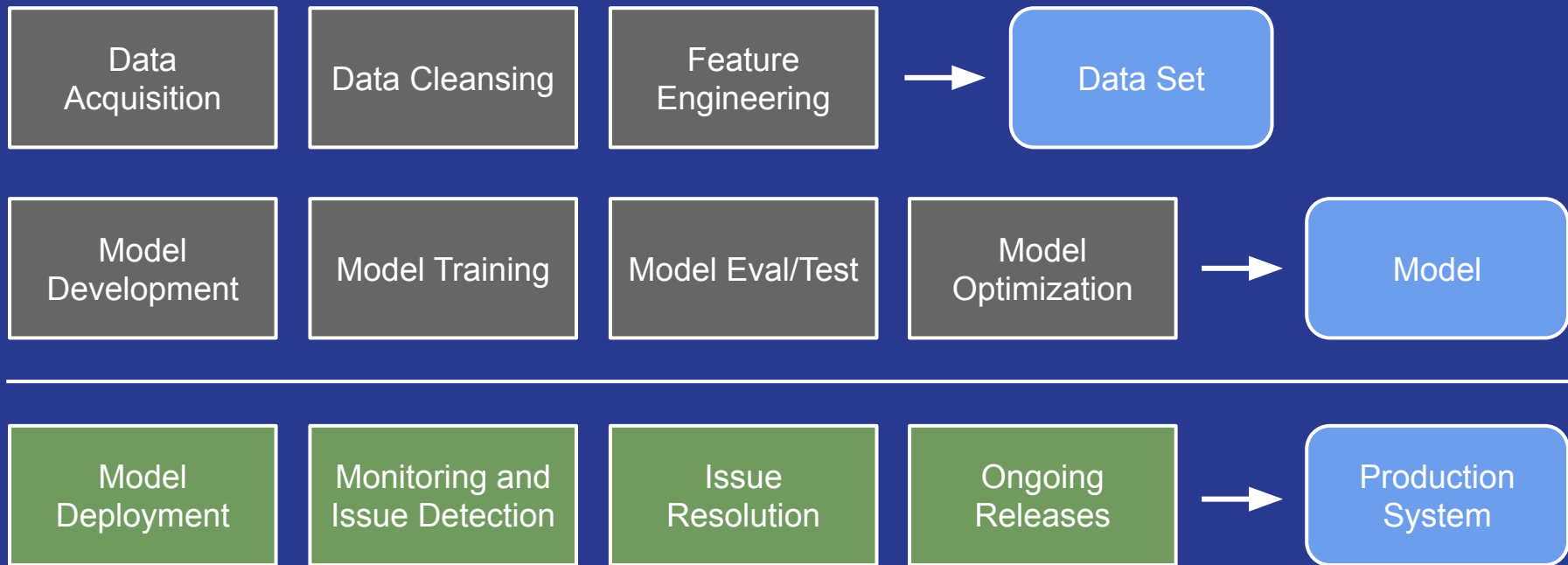
	Data Scientist	Software Engineer
Team Structure	Solo with Some Collaboration	Teams with Extensive Collaboration
Deliverables	Reports, Notebooks	Production Code
Expected Outcome	Uncertain	Certain (lolz)
Quality Measure	Nondeterministic	Deterministic

An Actual Quote

70% - 80% of my code is thrown away.

- Data scientist at Guild AI

ML Systems Development



The problem is *process*

Keys to Building ML Systems

Unify Data Science and Engineering Teams

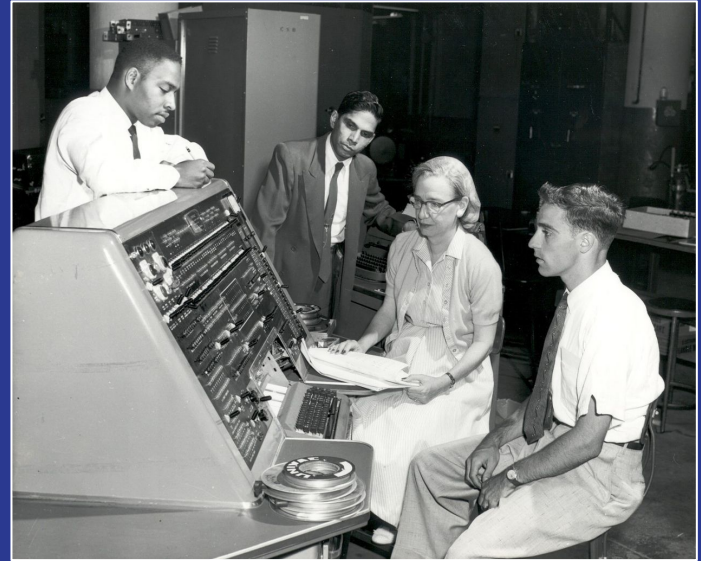
- Embed data scientists in software projects
- Consider colocation for on-site work
- Consider organizational change



Credit: www.freekibble.com

Use *Software* Best Practices

- Data science “best practices” a work-in-progress
- Iterations
- Automated build-and-test
- Joint code ownership
- Revision control, linting, code formatting, etc.



Operator's console of Univac I computer and four programmers. From left to right: Donald Cropper, K. C. Krishnan, Grace Hopper, and Norman Rothberg
www.computerhistory.org

Identify Model Success Criteria Metrics

- What is a minimally viable model?
- What is a game changing model?
- How is progress measured?
- How is regress measured?



Credit: Wikimedia Commons

Solve a Simple Problem First

- Start with the simplest model that can possibly work (doesn't have to be learned!)
- Use first model to drive success criteria and automation
- Improve incrementally using past models as baselines



Credit: www.nps.gov

Automate End-to-End *on Day 1*

- From model code change to deployment/integration
- Improvements are *incremental* and *visible*
- Encourage confident experimentation and refactoring



Credit: Wikimedia Commons

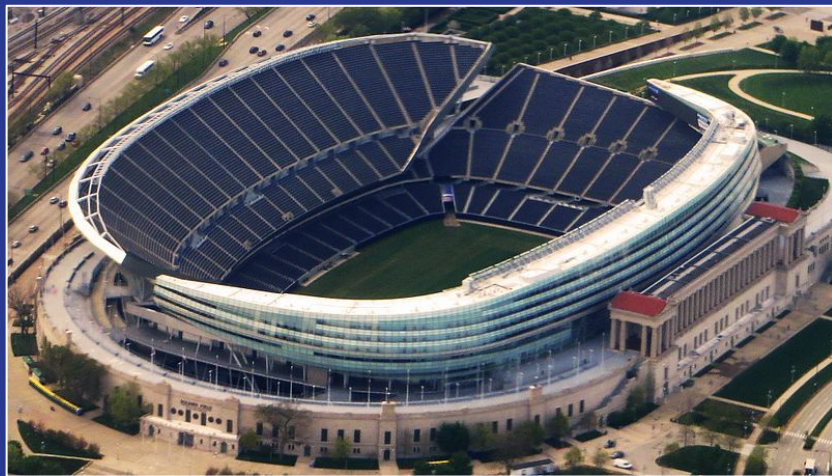
Software Development

*I think my commit fixed the issue.
I wonder what it broke.*

- Everyone

Ongoing Refactoring with Tests

- Strategies
 - Pair Programming
 - Backfill Refactoring
 - Arm's Length CodeAdaptation
- Track ML experiments for audit and comparison



Credit: Wikimedia Commons

Keys to Building ML Systems

- Unify data science and software engineering teams
- Apply software engineering methods to data science
- Identify and measure success metrics up front
- Solve a simple data science problem first
- Automate end-to-end build-and-test pipeline on day 1
- Ongoing refactoring supported by automated tests

Online Resources

r/MachineLearning	https://www.reddit.com/r/MachineLearning
ML Engineering Slack	https://bit.ly/ml-engineering-slack
Chicago ML Slack	https://bit.ly/chicago-ml-slack
Chicago ML Meetup	https://chicago.ml

Discussion