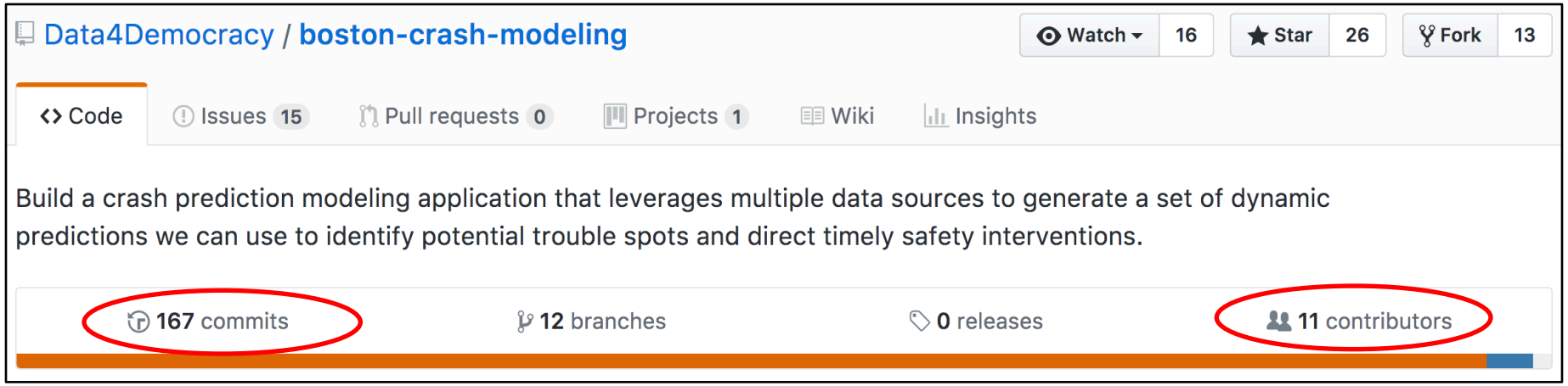
FeatureHub: towards collaborative data science

Micah J. Smith, Roy Wedge, Kalyan Veeramachaneni
MIT

IEEE DSAA 2017
Tokyo, Japan

Massachusetts Institute of Technology

A tale of two systems



Massive Open Data Science

Thousands of collaborators

Single solution

Range of expertise

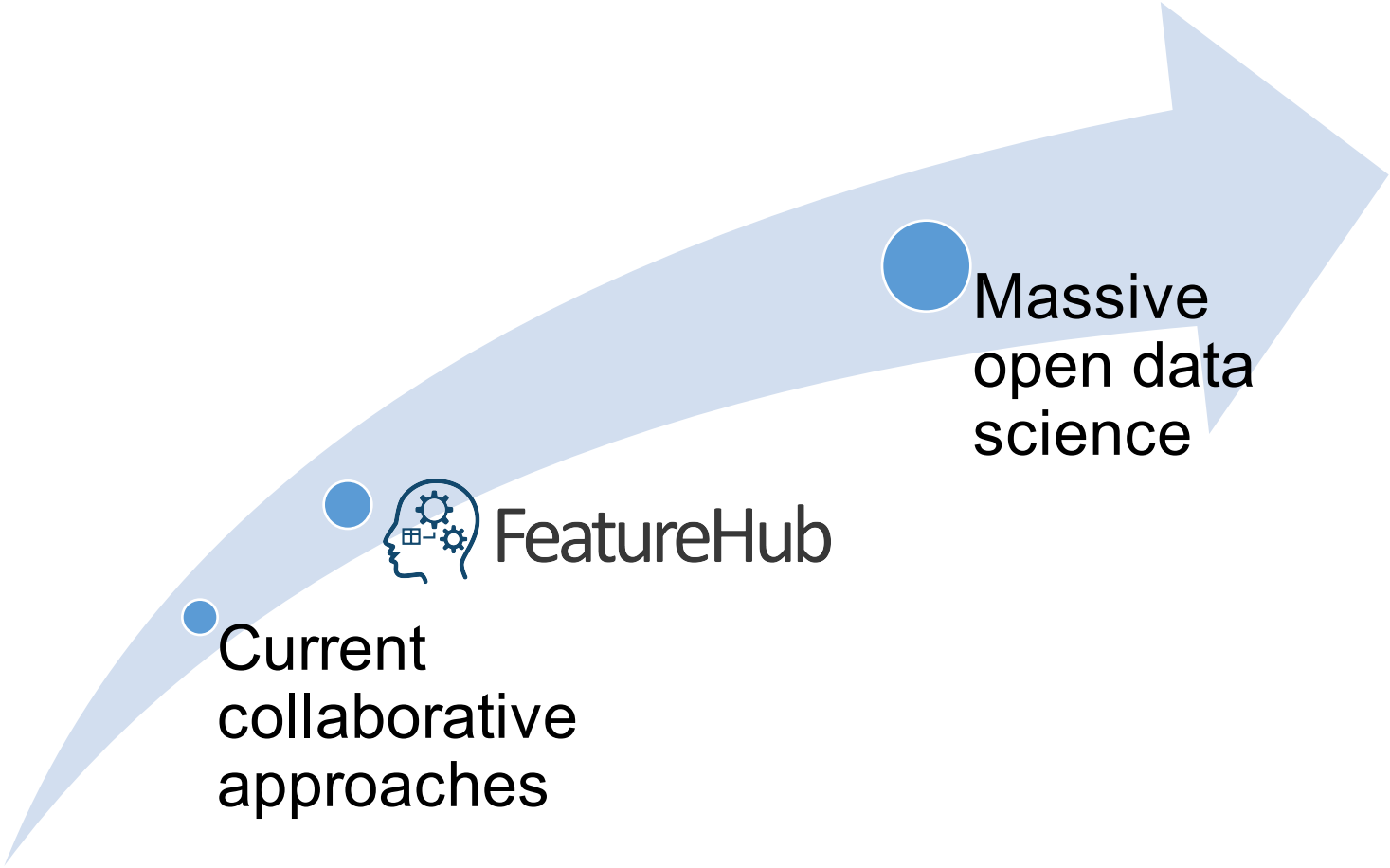
Natural abstractions

Machine-driven automation

The state of collaborative systems

	✓ ease of use ✓ share results	✗ no collaboration ✗ not scalable
	✓ integrated solution ✓ ecosystem of collaboration	✗ wrong abstractions ✗ difficult to use
	✓ ease of use ✓ bookkeeping	✗ not open ✗ expensive
	✓ many competitors	✗ many solutions ✗ no additional structure

Towards this vision



The FeatureHub paradigm

Towards collaboration at scale through feature engineering

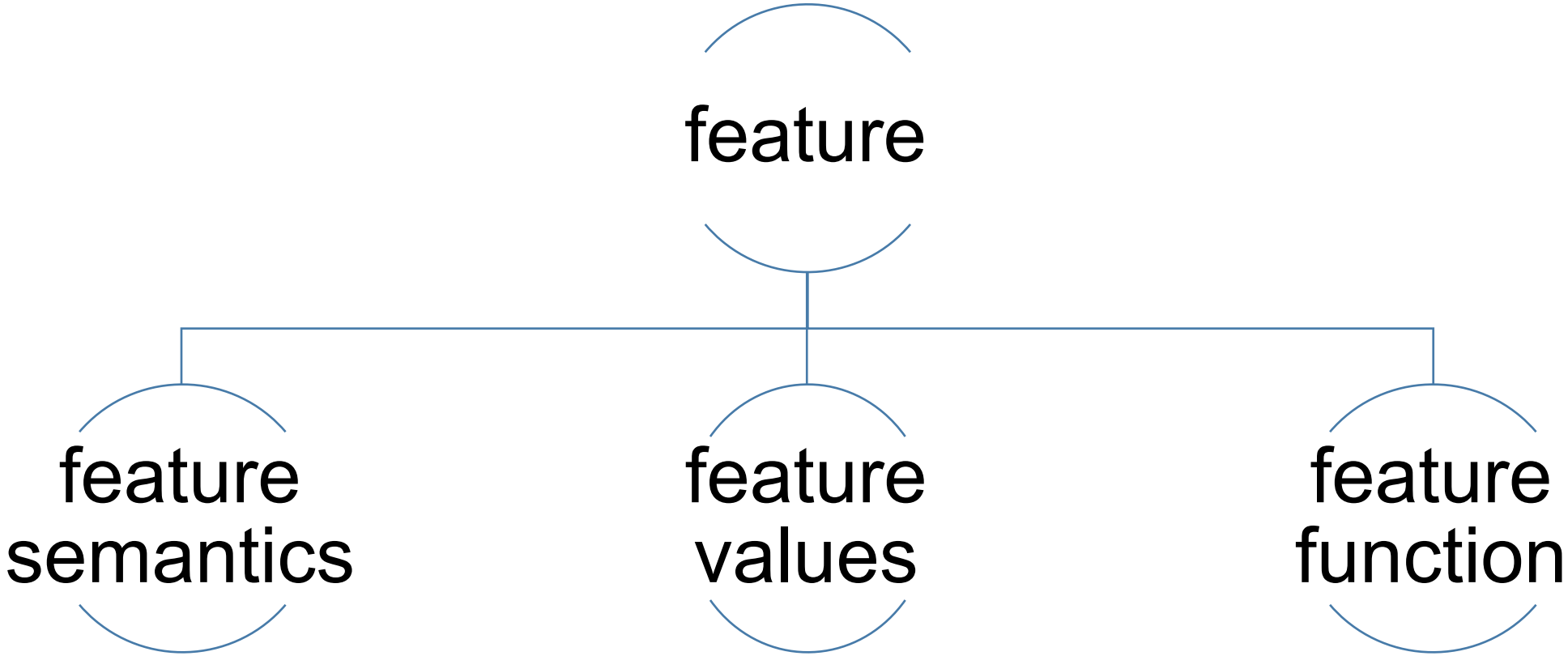
- Isolate and structure feature engineering
- Parallelize across people and features
- Minimize redundant work
- Automate everything else

What is a feature?

A *feature* is a quantitative, measurable property of a particular entity.

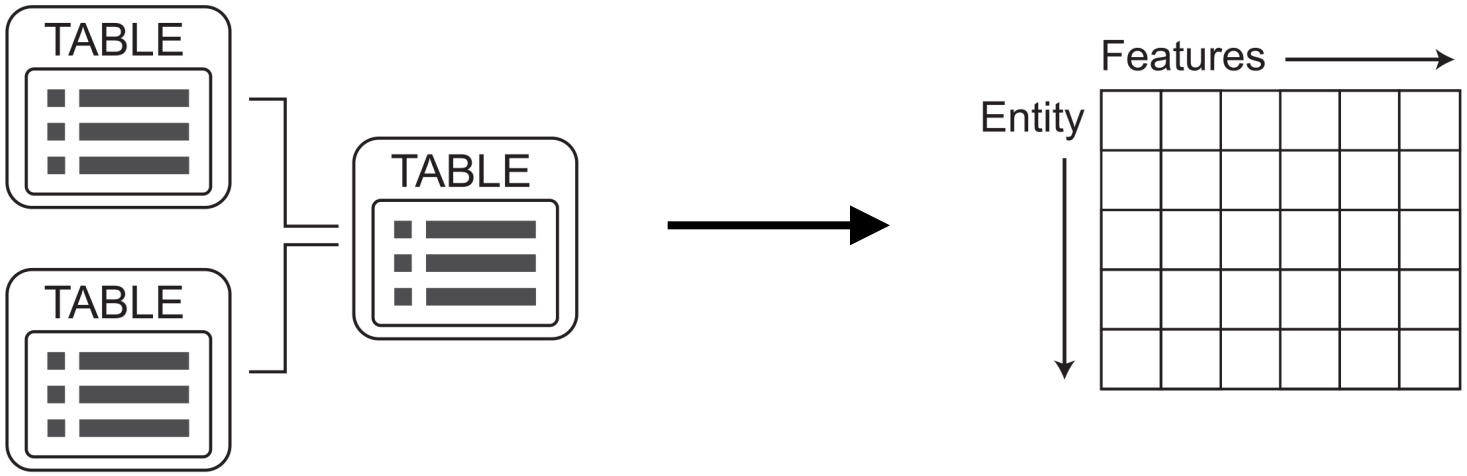
id	Closest traffic light (meters)
Beacon St @ Prentiss	470
Vassar St @ Main	25
Newbury St @ Mass Ave	0
…	
Memorial Drive @ Ames	130

What is a feature?



What is feature engineering?

Feature engineering is the process of ideating *feature semantics*, and writing *feature functions* to extract *feature values* from a raw data source.

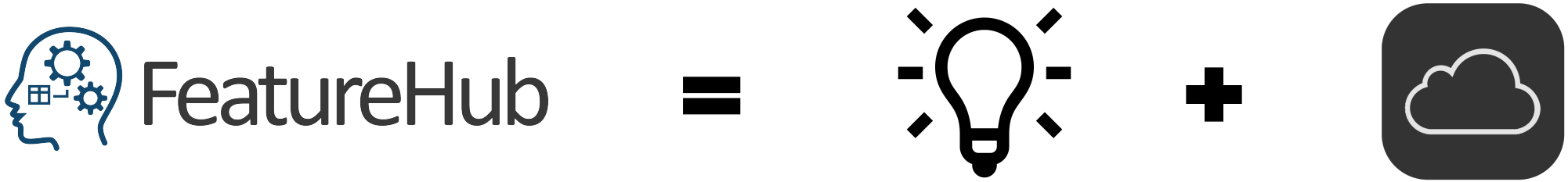


Why feature engineering?

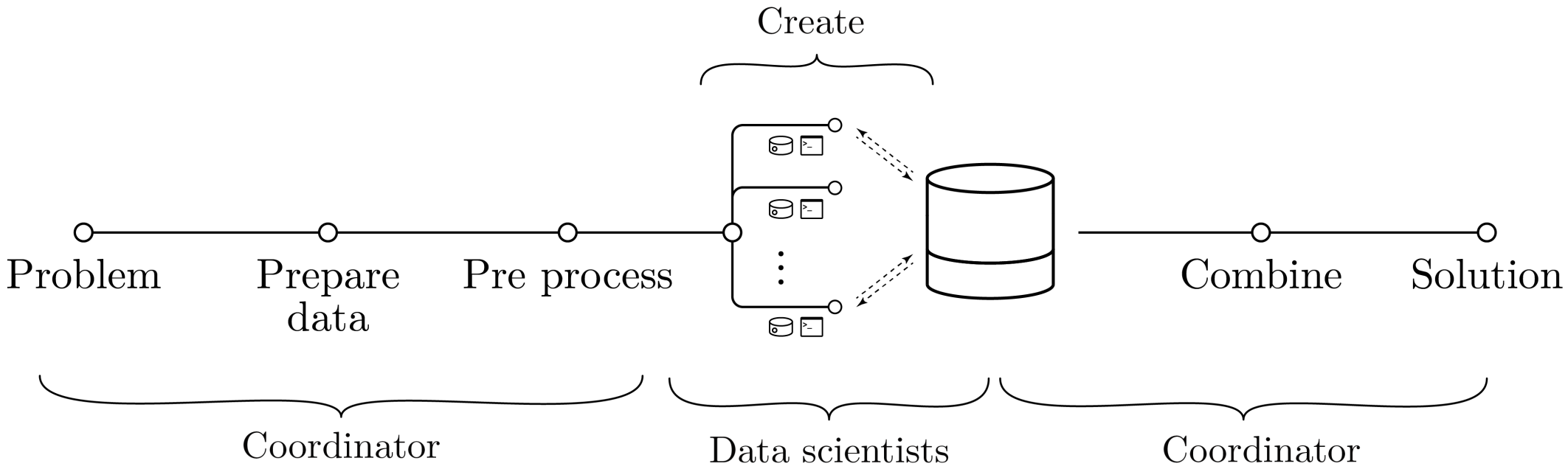
- Features very important to modeling success
- Challenging!
  - Needs human intuition and domain expertise
  - Automation difficult in many circumstances
  - Collaboration can help uncover key ideas
- Can structure into more natural units of work

Our goal

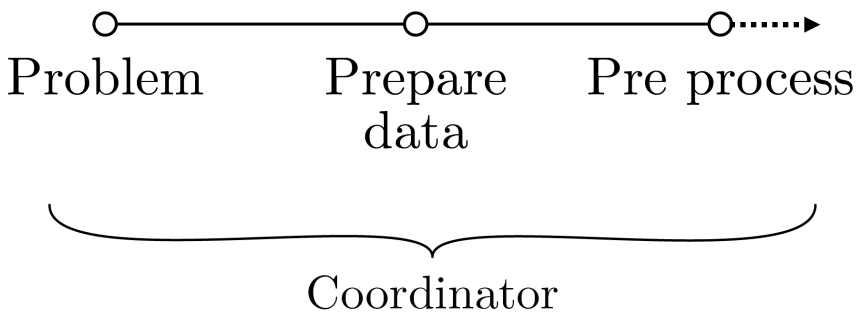
Develop a system to enable collaborative data science under the FeatureHub paradigm.



How it works


Create
Problem
Prepare data
Pre process
Combine
Solution
Coordinator
Data scientists
Coordinator

Launch



- `setup`: Setup problem and platform
- `prepare_dataset`: Minimal cleaning, extract metadata
- `preextract_features`: Preprocess features

CREATE: Scaffolding feature functions

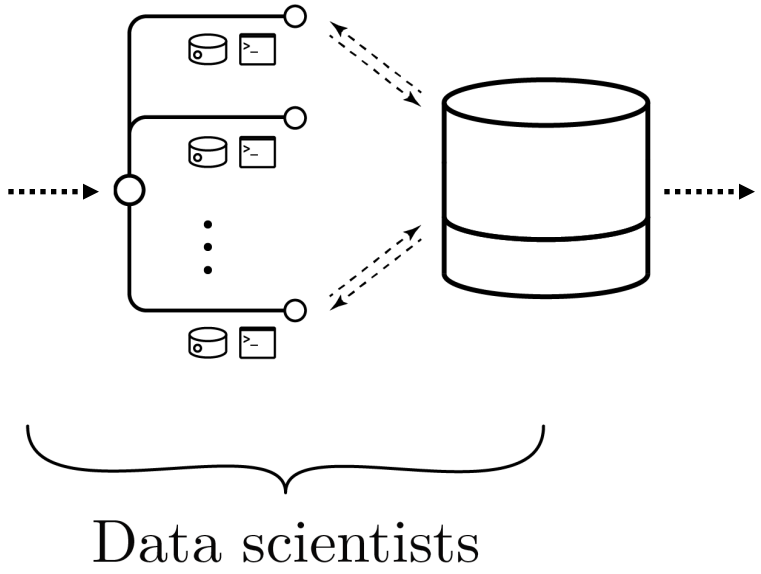
```
1 def hi_lo_age(dataset):
2 """Whether users are older than 30 years"""
3     from sklearn.preprocessing import binarize
4     threshold = 30
5     return binarize(dataset["users"]["age"], threshold)
```

- Input: single collection of data tables
- Output: single column of values – one value per entity

Bookkeeping

- Actually “works”
- Self-contained

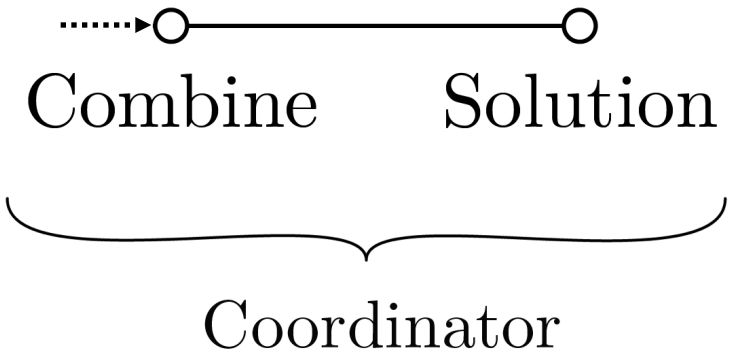
CREATE



- Log in to hosted Jupyter Notebook environment
- `get_dataset`: Acquire dataset
- `discover_features`: Collaborate on new features at integrated forum, “fork” existing features
- `evaluate`: Write and evaluate features
- `submit`: Submit feature functions (source code) to evaluation system and feature database

Combine

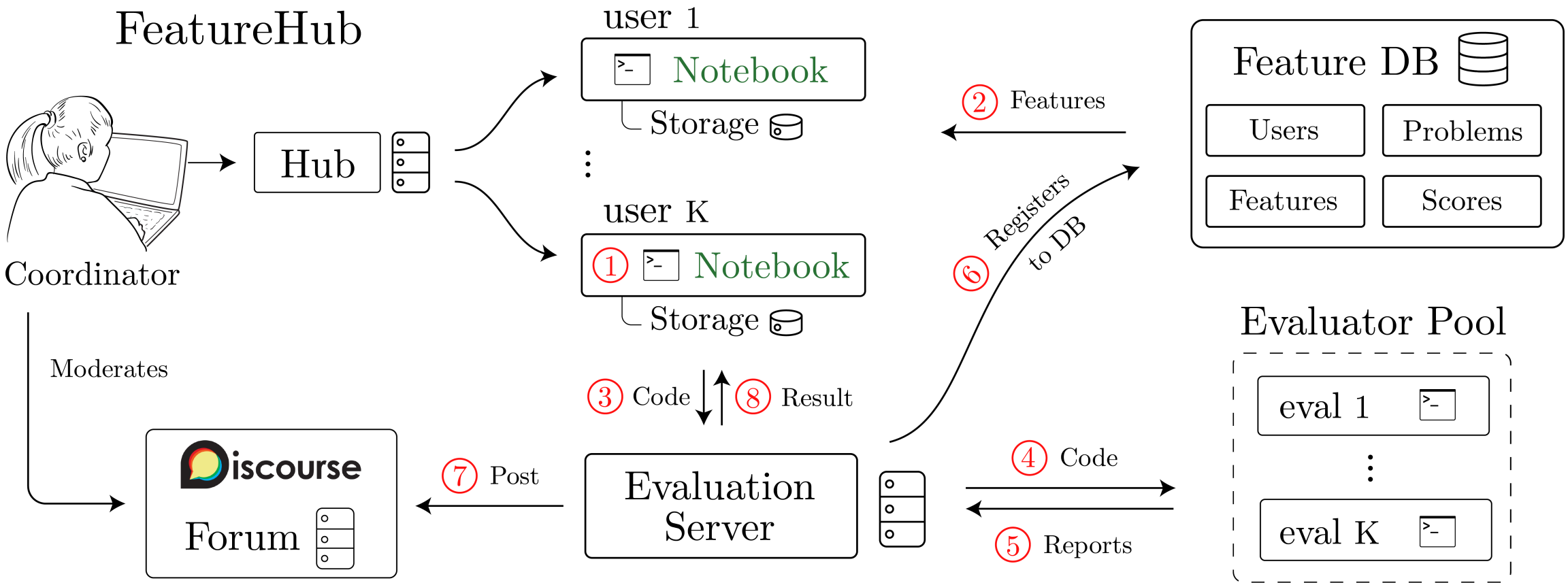
- `extract_features`: *Automatically* execute feature functions to extract values on train and test sets
- `learn_model`: *Automatically* build and evaluate models using AutoML
- *Automatically* produce solution (predictions on new data points)



Implementation challenges

- Integrating untrusted source code
  - Quality
  - Security
- High-quality contributions
  - Metrics to reward good work
  - Adversarial behavior
- Minimize redundant work while scaling
- Appropriate use of automation technologies

Platform architecture



Experiments

Hired 41 crowd data scientist workers from Upwork

- Beginner to intermediate experience/skill, hourly rates between 7 to 45 USD per hour
- Write features on FeatureHub: two prediction problems, five hours total
  - *airbnb:* Predict the destination country of Airbnb users (Source: Kaggle)
  - *sberbank:* Predict selling price for houses and apartments (Source: Kaggle)
- Assign to experimental groups to assess different collaborative functionality
- Bonus payments for high quality features

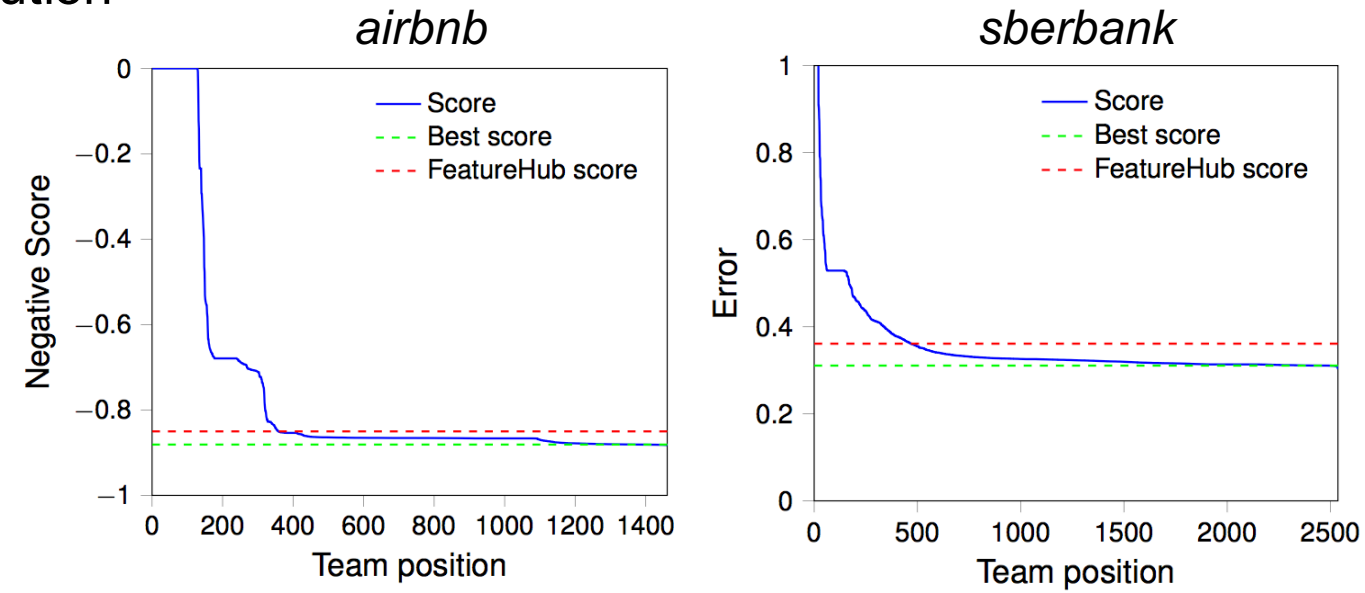
Data collected

- 171 hours spent on platform
- 1952 features submitted
- Detailed survey administered

Experiments

Combined model competes with expert data scientists

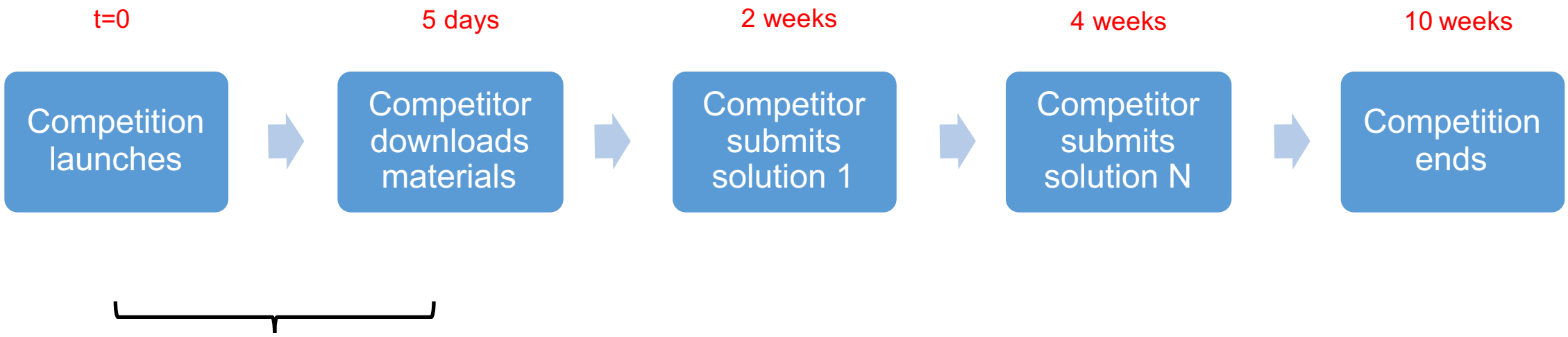
- Pitted FeatureHub predictions against those of “expert” data scientists on Kaggle
- Model uses combined feature matrix with 6 hours of `auto-sklearn`
- With these limited resources, beats 25% of experts and scores within 0.03 to 0.05 points of winning solution



Experiments

Substantially decreases "time to solution"

- Achieve potential turnaround time of <1 day

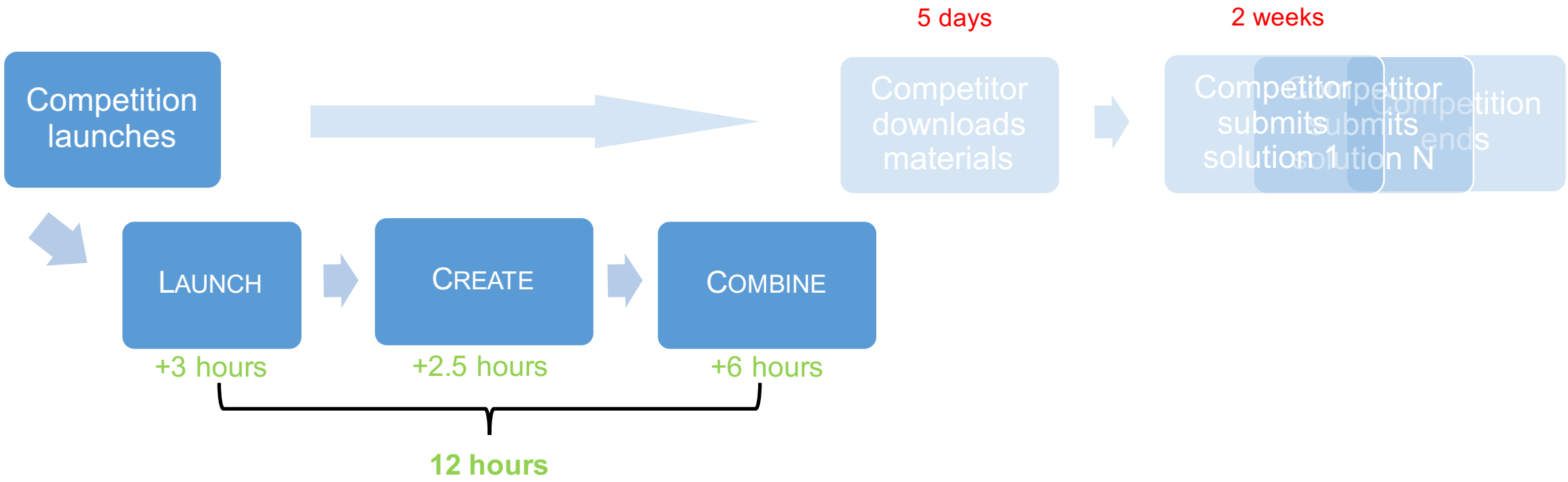


What can we accomplish with FeatureHub?

Experiments

Substantially decreases "time to solution"

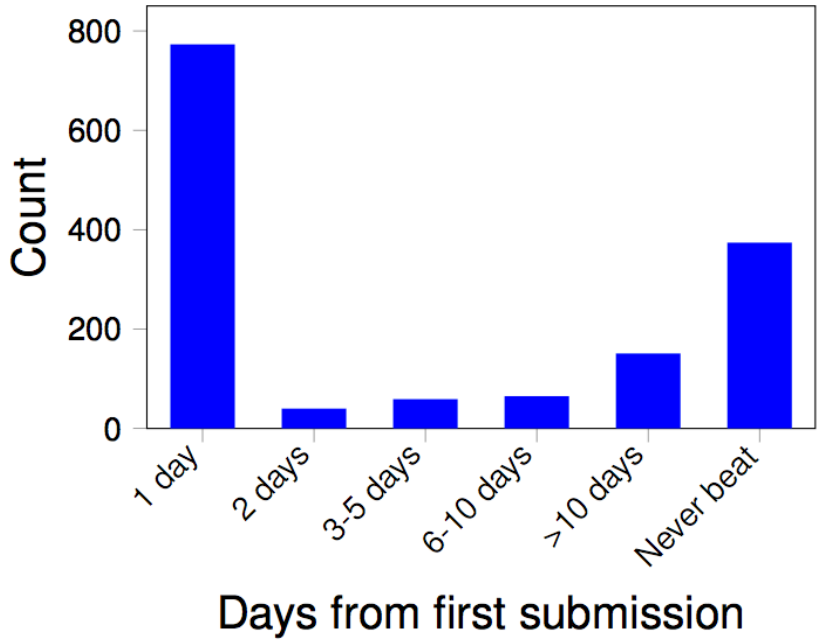
- Achieve potential turnaround time of <1 day



Experiments

Substantially decreases "time to solution"

- (Very conservatively) 47% of experts are not able to achieve FeatureHub-level performance as quickly



Summary

- Propose a new approach to collaborative feature engineering
- The approach is simple but powerful:
  1. Focus creative effort of data scientists working in parallel on feature engineering
  2. Integrate source code contributions into a single model
  3. Automate everything else and produce output quickly
- Engineer a cloud platform to do crowdsourced feature engineering with automated modeling
- Experimental results show we can leverage crowd data scientists using FeatureHub to generate competitive predictive models using limited resources

FeatureHub: towards collaborative data science

Micah J. Smith, Roy Wedge, Kalyan Veeramachaneni
MIT

Source code: https://github.com/HDI-Project/FeatureHub
Correspondence: Micah Smith (micahs@mit.edu, @micahjsmith)

Massachusetts Institute of Technology