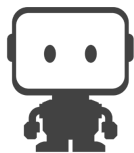


Automated Machine Learning





DataRobot

Ukrainian Catholic University

**APPLIED
SCIENCES
FACULTY** ●

kaggleTM

Machine Learning Engineer

Core Modeling Team

Teach sometimes

AI, Machine Learning, Summer/Winter ML Schools

Compete sometimes

Currently hold an Expert rank, top 2% worldwide



Why This Talk



● **AutoML**
Search term

+ Compare

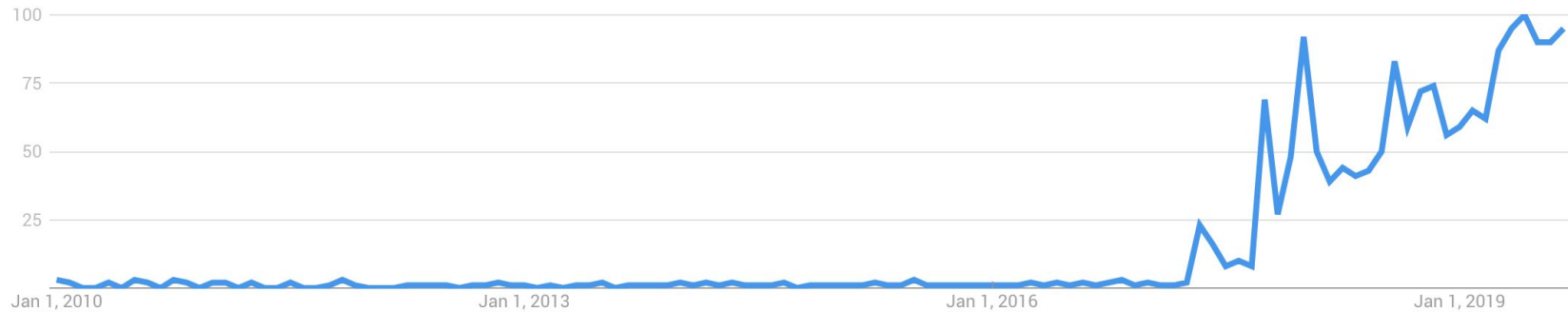
Worldwide ▾

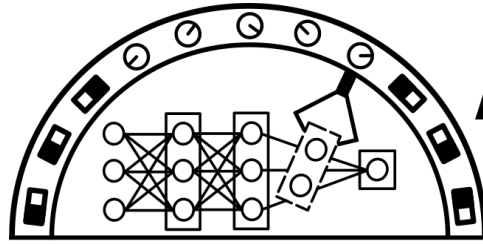
1/1/10 - 9/1/19 ▾

All categories ▾

Web Search ▾

Interest over time





AutoML.org

Freiburg-Hannover

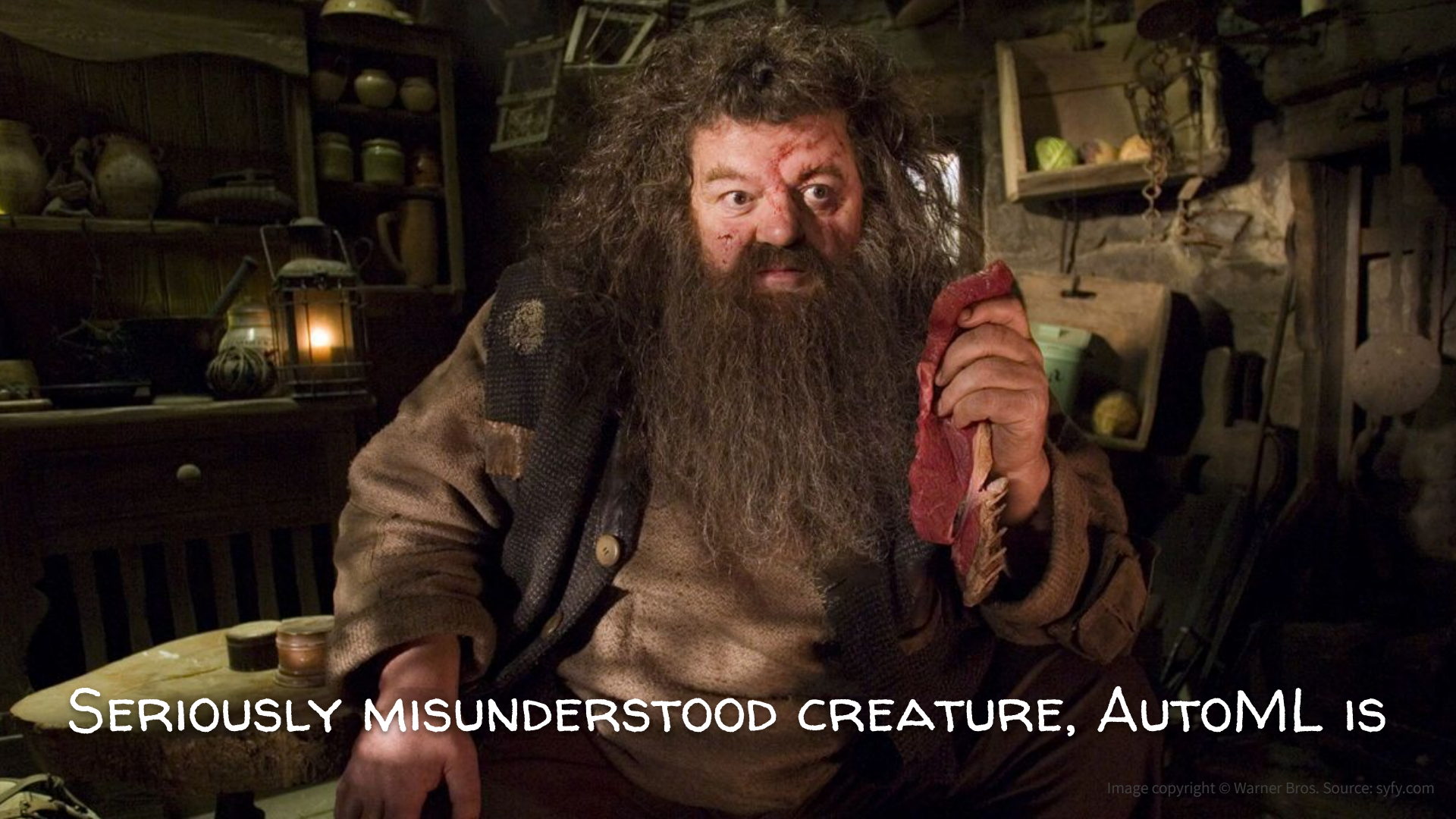
AutoML provides methods and processes to make ML available for non-ML experts, to improve efficiency of ML and to accelerate research on ML.

[\[www.automl.org\]](http://www.automl.org)



WIKIPEDIA
The Free Encyclopedia

Automated machine learning (AutoML) is the process of automating end-to-end the process of applying machine learning to real-world problems



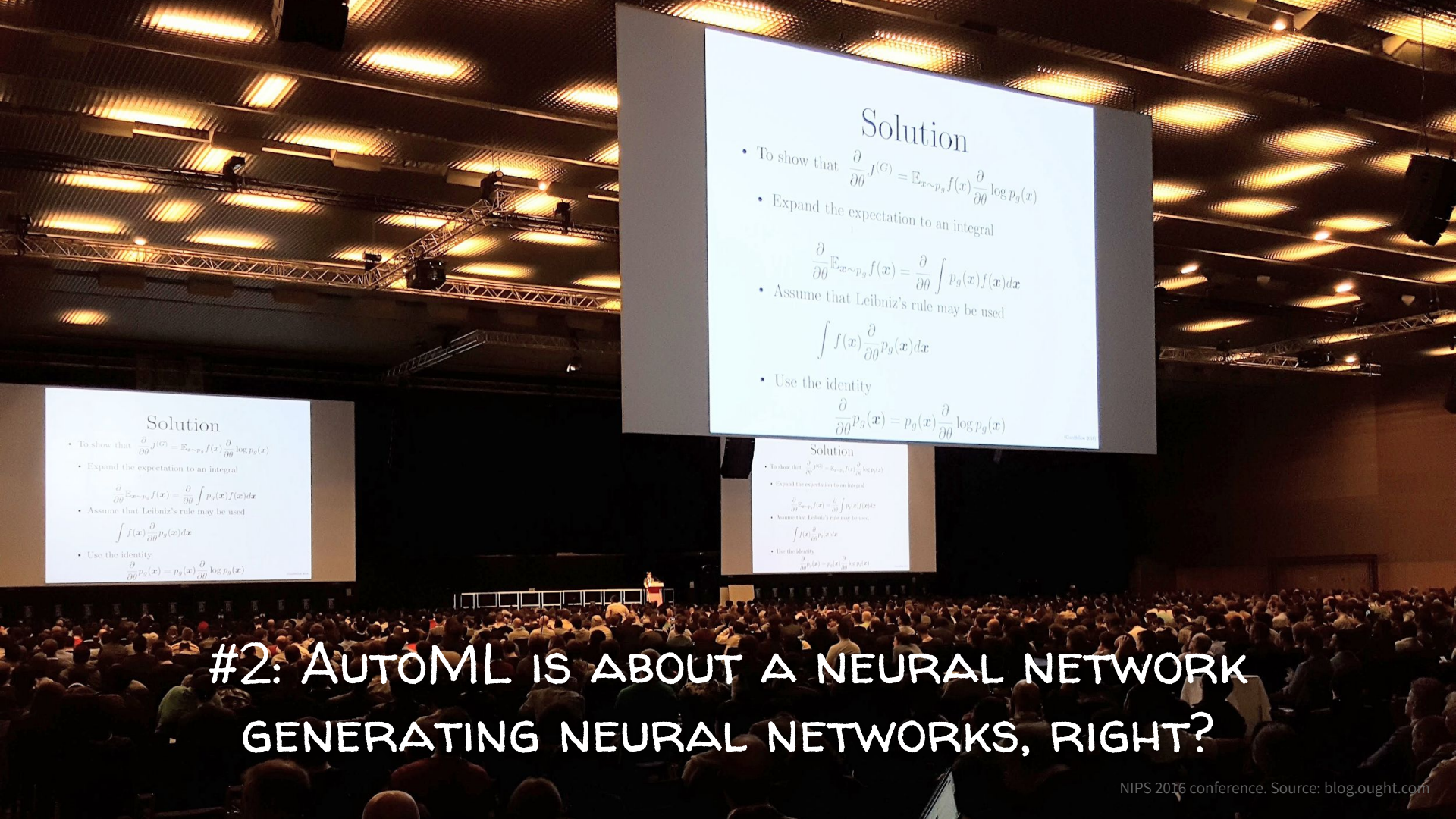
SERIOUSLY MISUNDERSTOOD CREATURE, AUTOML IS



Top 3 Questions Peers Ask Me

A photograph of two men sitting at a desk in an office. The man on the left has a mustache, glasses, and is wearing a white short-sleeved shirt, a patterned tie, and suspenders. He is looking directly at the camera with a serious expression. The man on the right is balding, wearing glasses, a dark suit jacket, a white shirt, and a patterned tie. He is also looking at the camera with a serious expression. The background shows a typical office environment with cubicles and fluorescent lighting.

#1: Will ALL DATA SCIENTISTS LOSE THEIR JOBS SOON?



Solution

- To show that $\frac{\partial}{\partial \theta} J^{(G)} = \mathbb{E}_{x \sim p_{\theta}} f(x) \frac{\partial}{\partial \theta} \log p_{\theta}(x)$
- Expand the expectation to an integral
- Assume that Leibniz's rule may be used
- Use the identity

$$\frac{\partial}{\partial \theta} \mathbb{E}_{x \sim p_{\theta}} f(x) = \frac{\partial}{\partial \theta} \int p_{\theta}(x) f(x) dx$$
$$\int f(x) \frac{\partial}{\partial \theta} p_{\theta}(x) dx$$
$$\frac{\partial}{\partial \theta} p_{\theta}(x) = p_{\theta}(x) \frac{\partial}{\partial \theta} \log p_{\theta}(x)$$

Solution

- To show that $\frac{\partial}{\partial \theta} J^{(G)} = \mathbb{E}_{x \sim p_{\theta}} f(x) \frac{\partial}{\partial \theta} \log p_{\theta}(x)$
- Expand the expectation to an integral
- Assume that Leibniz's rule may be used
- Use the identity

$$\frac{\partial}{\partial \theta} \mathbb{E}_{x \sim p_{\theta}} f(x) = \frac{\partial}{\partial \theta} \int p_{\theta}(x) f(x) dx$$
$$\int f(x) \frac{\partial}{\partial \theta} p_{\theta}(x) dx$$
$$\frac{\partial}{\partial \theta} p_{\theta}(x) = p_{\theta}(x) \frac{\partial}{\partial \theta} \log p_{\theta}(x)$$

Solution

- To show that $\frac{\partial}{\partial \theta} J^{(G)} = \mathbb{E}_{x \sim p_{\theta}} f(x) \frac{\partial}{\partial \theta} \log p_{\theta}(x)$
- Expand the expectation to an integral
- Assume that Leibniz's rule may be used
- Use the identity

$$\frac{\partial}{\partial \theta} \mathbb{E}_{x \sim p_{\theta}} f(x) = \frac{\partial}{\partial \theta} \int p_{\theta}(x) f(x) dx$$
$$\int f(x) \frac{\partial}{\partial \theta} p_{\theta}(x) dx$$
$$\frac{\partial}{\partial \theta} p_{\theta}(x) = p_{\theta}(x) \frac{\partial}{\partial \theta} \log p_{\theta}(x)$$

#2: AUTOML IS ABOUT A NEURAL NETWORK GENERATING NEURAL NETWORKS, RIGHT?

A promotional image for the TV show 'Suits' featuring two main characters, a man in a grey suit sitting in a chair and another man in a blue suit standing by a window overlooking a city skyline. The text is overlaid at the bottom of the image.

#3: DS/ML REQUIRES SERIOUS HUMAN EXPERTISE.
HOW CAN AUTOMATION EVER BE “BETTER”?

Three Levels of Scope

1

Academic AutoML

Advance human knowledge in fundamental AutoML methods
Get publications, citations, degrees, inspire R&D

2

Libraries and Open-Source AutoML Software

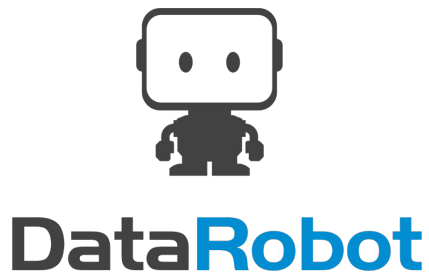
Refine academic ideas to technical feasibility, gain product engineering experience
Find peers, validate ideas with early adopters, build a community of practitioners

3

Commercial AutoML Product

Build a profitable business by solving real-world problems and delivering value at scale
(from small businesses and NGOs to largest corporations and governments)

 *focus of this talk*



Some Background



Unicorn startup from Boston, MA



Developing AutoML products since 2012



\$430M of investments (Series E)



Hundreds of enterprise customers (including 1/3 of Fortune 50)



1.3 billion ML models built so far



1000 employees @ ~50 locations around the globe

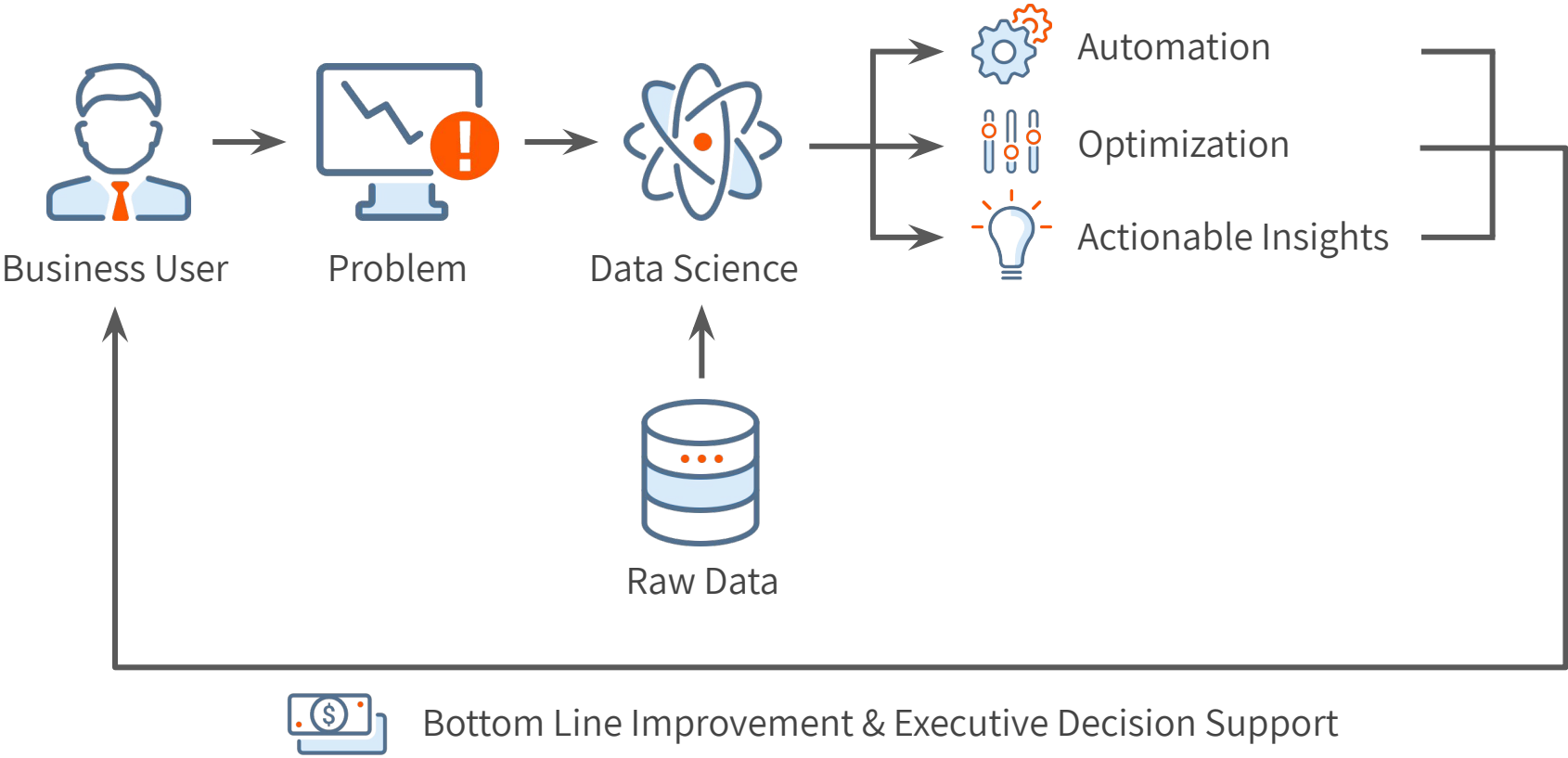
“DataRobot sets the standard for augmented data science and machine learning”

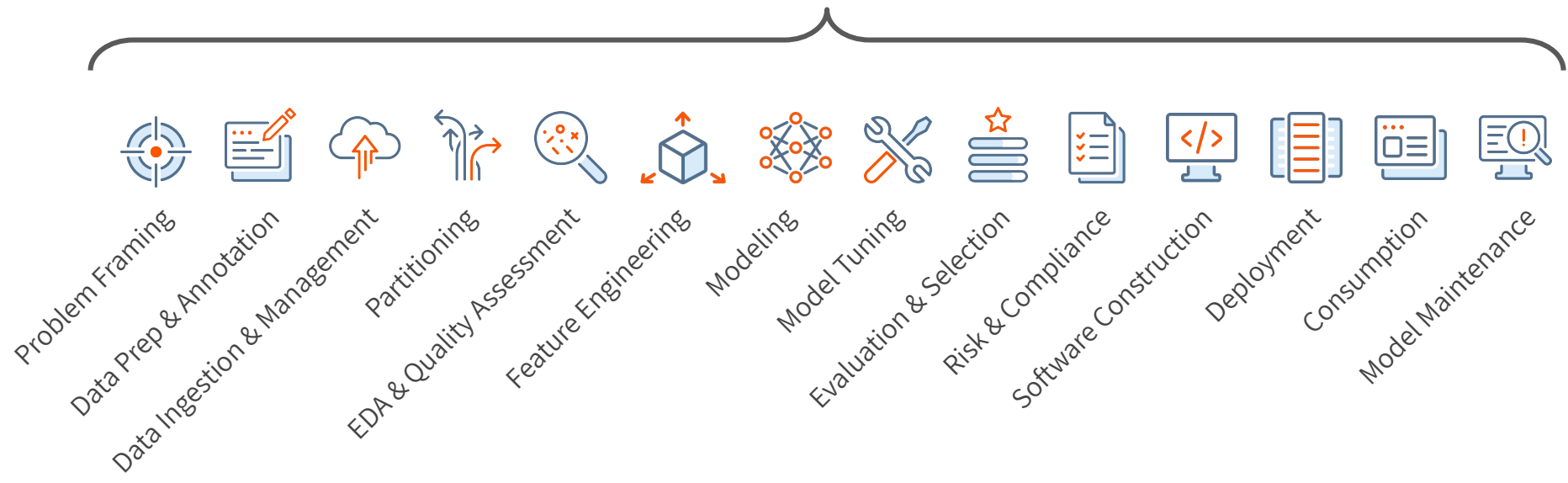
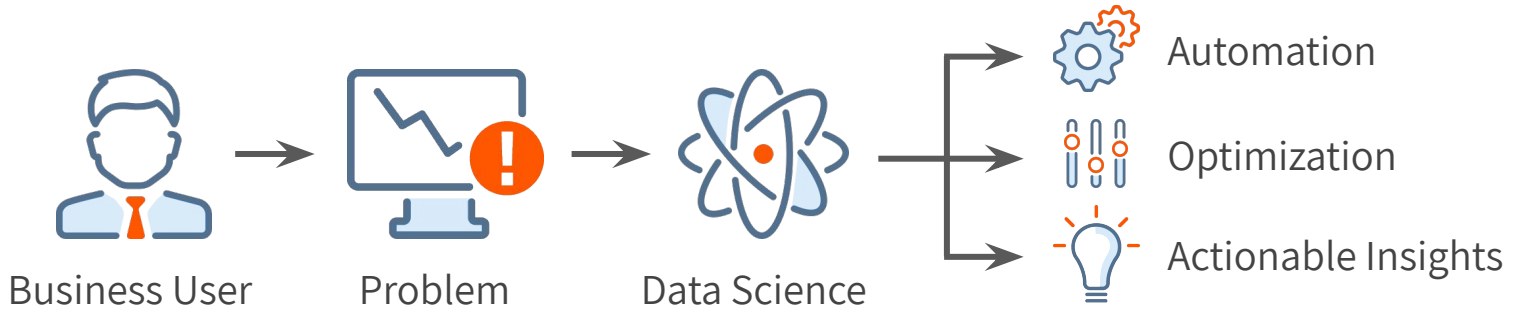
– Gartner Magic Quadrant for DS and ML Platforms, 2019

“DataRobot leads the pack with a broad set of robust capabilities”

– Forrester New Wave, Automation-Focused ML Solutions, Q2 2019

Recap: DS Value Generation





Persona: Data Scientist



Needs domain knowledge to do right

Hates doing

Enjoys doing and wants to keep doing it

Often lacks skills or methodology to do right



Problem Framing

Data Prep & Annotation

Data Ingestion & Management

Partitioning

EDA & Quality Assessment

Feature Engineering

Modeling

Model Tuning

Evaluation & Selection

Risk & Compliance

Software Construction

Deployment

Consumption

Model Maintenance

*In large organizations, a lot of “throwing over the wall” happens here
~85% of DS projects never make it to production [bit.ly/30PGOZM]*



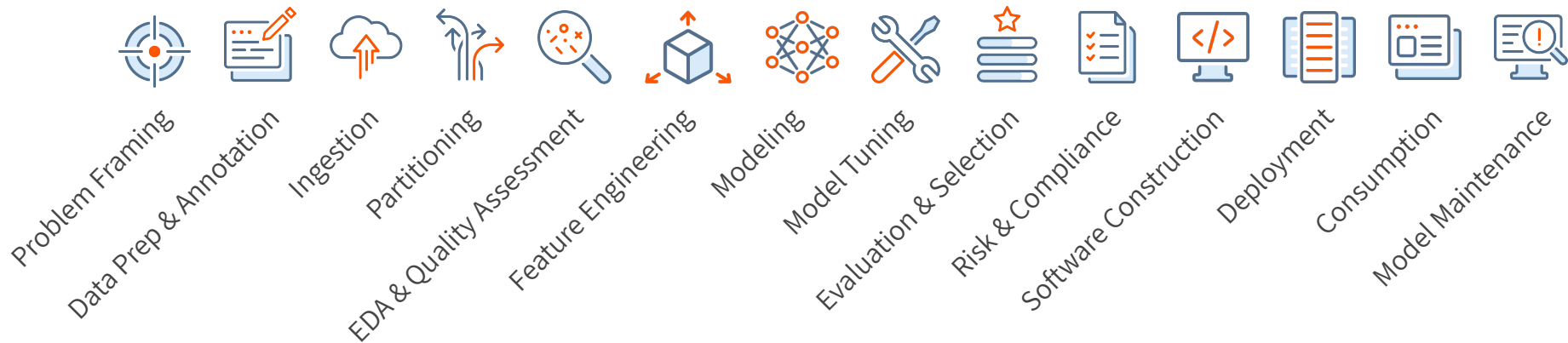
Recall The Earlier Definitions:

1. “Accessible for non-ML experts”
2. “End-to-end automation”

Actually needed to deliver value in the real world

*Vast majority of AutoML research
and emerging products focused here*

*Vast majority
of ML research
focused here*



Part I AutoML Methods

1	Hyperparameter Optimization	3
	Matthias Feurer and Frank Hutter	
2	Meta-Learning	35
	Joaquin Vanschoren	
3	Neural Architecture Search	63
	Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter	

Part II AutoML Systems

automl.org/book/

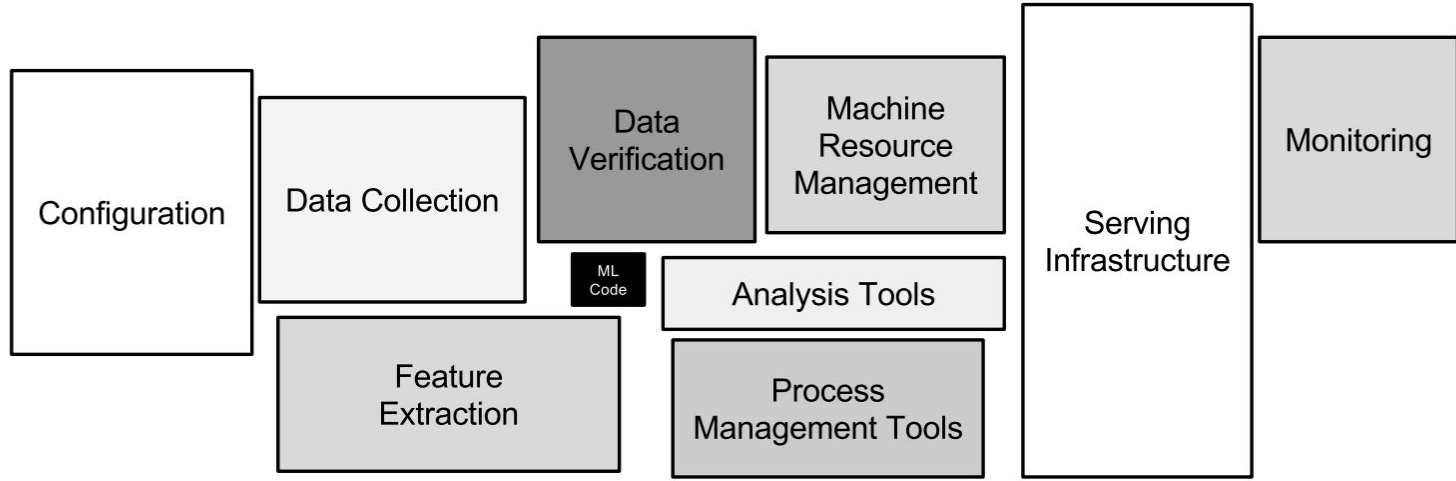


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

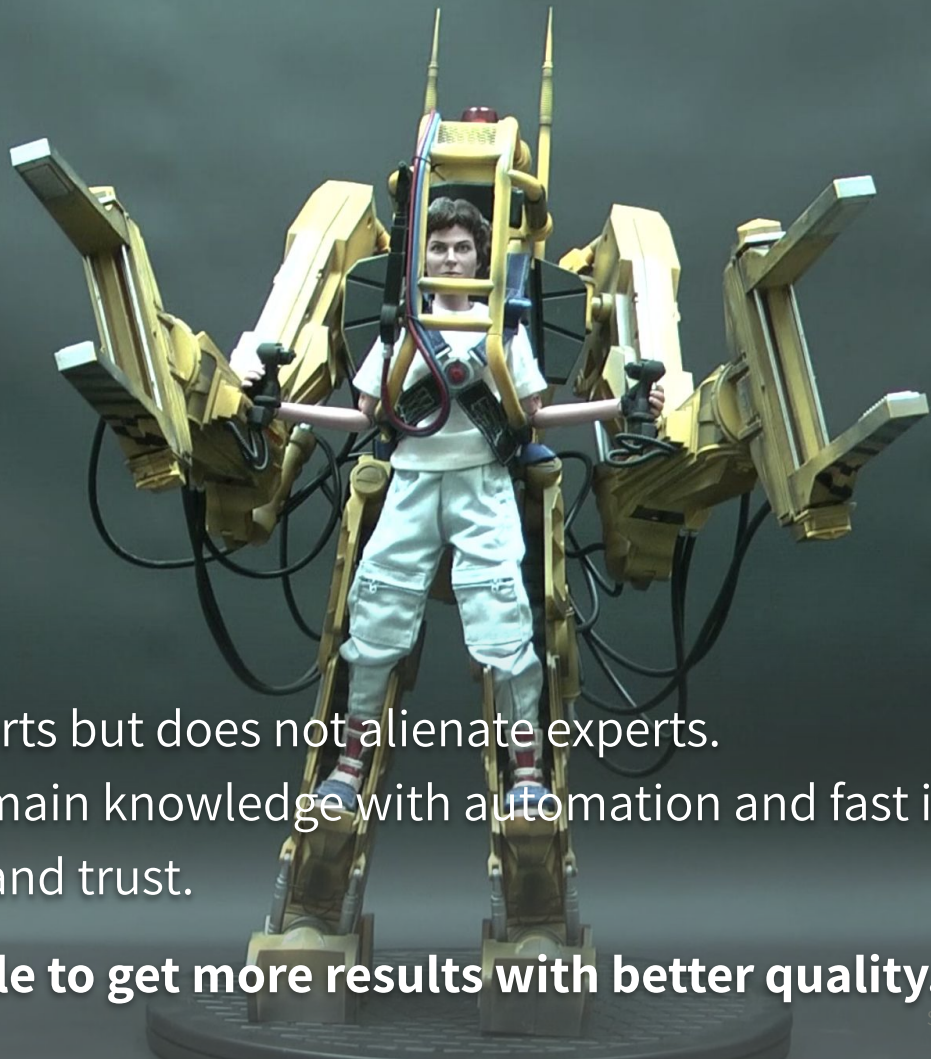
Sculley et al. (Google)

“Hidden Technical Debt in Machine Learning Systems” [NIPS 2015]

Ideal Goal



- Lots of capable and motivated people in non-DS teams that know the domain and can deliver value
- Data scientists focus on strategic projects, mentor “citizen data scientists”, and help with problem setup

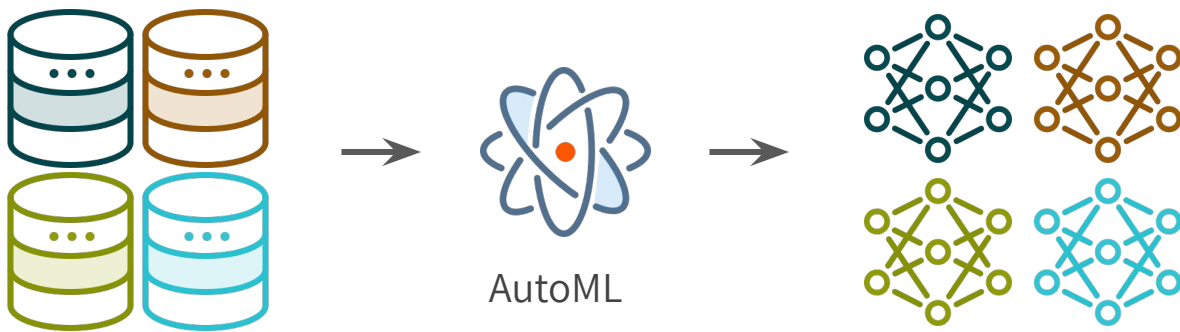


Good AutoML:

1. Empowers non-experts but does not alienate experts.
2. Augments user's domain knowledge with automation and fast iteration.
3. Provides guardrails and trust.

Enables more people to get more results with better quality.

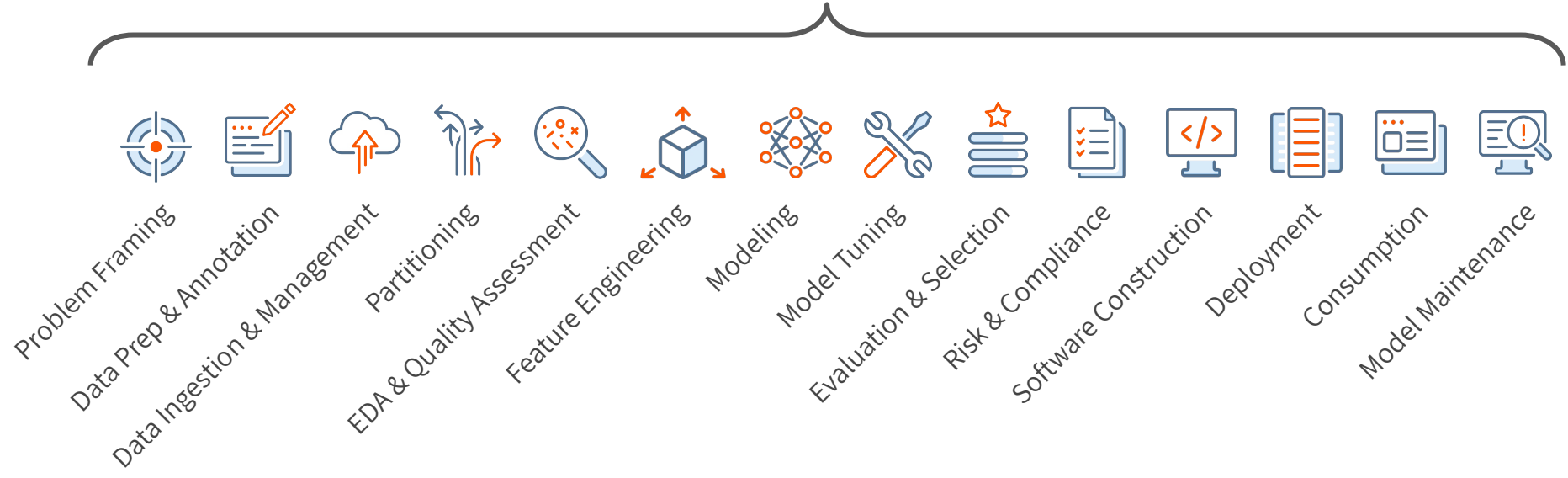
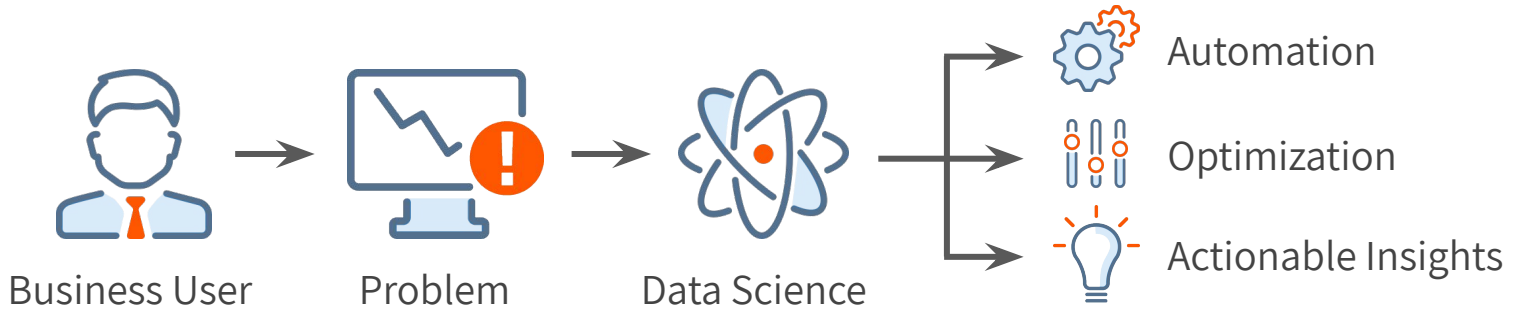
Interesting Use Case: Model Factory



- Models specific to data subsets (e.g. propensity per SKU)
- Models specific to time ranges (e.g. +1 day, +1 month forecast)
- Short-lived models with rapid refresh cycle (e.g. fraud, malware)



Interesting Challenges of Building an AutoML Product





Problem Framing

- Automatic detection of the modeling problem from data layout (regression, binary, multiclass, multilabel, ranking, recommendation, ...)
- Are there datetime features in the data? Maybe it's a time series forecasting problem? Maybe there are multiple series along the same axis?
- Maybe there's no target at all? (E.g., user is interested in anomaly detection)
- If there's a target, can we figure out its distribution and recommend a reliable optimization metric?
- Are there any prior constraints? (E.g., prediction range, monotonicity, weights)



Data Preparation and Annotation

- Does the data have valid tabular shape? Are there various data sources to merge?

① Deep Feature Synthesis: automatic generation of features from snowflake-schema relational data

J. Kanter, K. Veeramachaneni (2015), Deep feature synthesis: Towards automating data science endeavors. DSAA 2015.

“featuretools” Python package: <https://github.com/Featuretools/featuretools>

- Is the target defined everywhere? Do we need weak supervision or active learning?

① Snorkel: rapid training data creation with weak supervision

<https://github.com/snorkel-team/snorkel>

<https://arxiv.org/abs/1711.10160>



Data Acquisition and Management

- Connectivity to lots of enterprise data sources; cataloging and snapshot-ability
- Managing data access drivers and data connection security
- Automatic recognition of ingested data format (including proprietary formats)
- Is there metadata available? We can use it for modeling later!
(data type annotations, constraints, table keys/relationships)
- Ingestion at scale: need to be smart and get to initial results quickly to iterate faster
- User expects the data sources and formats used for training to be usable for production, without any special handling on user's part



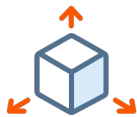
Partitioning

- Automatically recommend a problem-aware validation schema
- Are there group relationships between rows? Need different validation
- Is datetime an important dimension in the dataset? Need different validation
- Seasonal time series detected? Validation needs to account for the seasonal cycles
- Do we need to oversample/undersample/stratify/augment?
- Do not reuse the same validation set for multiple purposes (HPO, ES, model ranking)
- The entire modeling pipeline must be robust enough to *never* peek into the holdout until the final model deployment



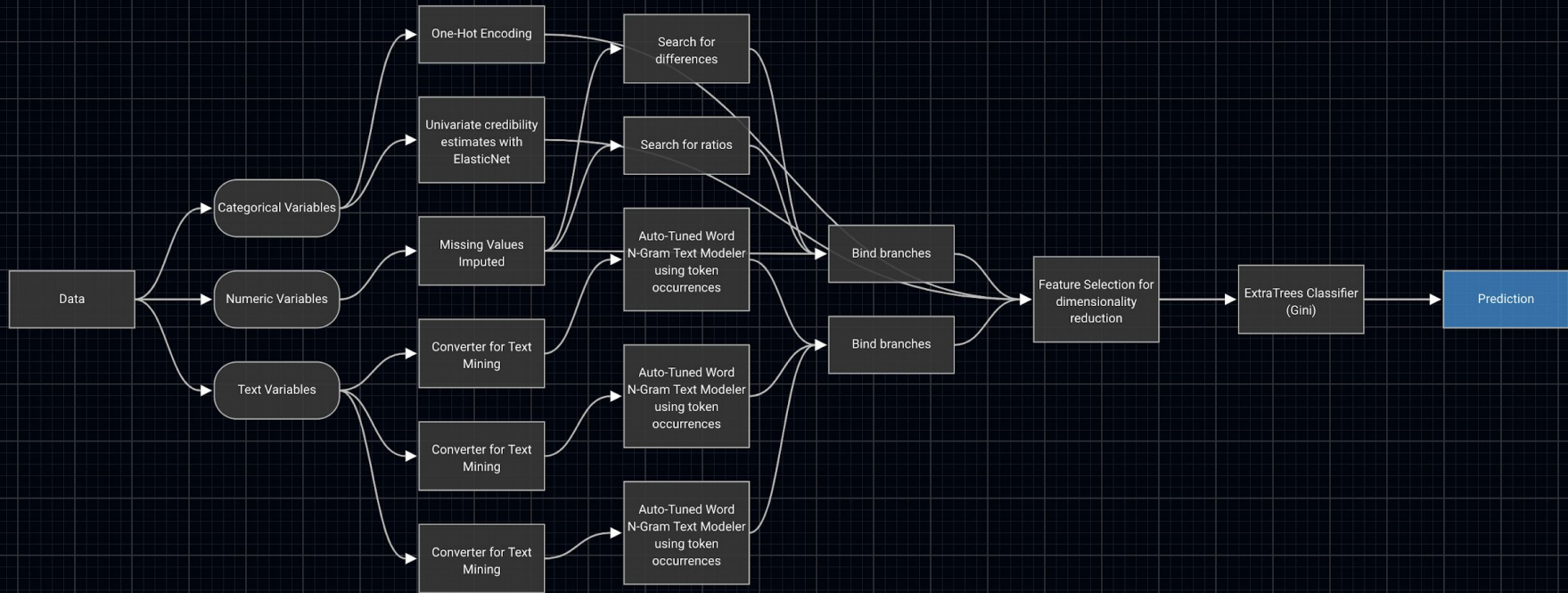
EDA & Quality Assessment

- Automatic data type / column intent detection
 - ① Exercise: think how you would distinguish between numerics, ordinals, categoricals, text, datetime
- Are there features without meaningful information?
(IDs, constants, duplicates, extreme cardinality or sparsity, noise)
- Are there features that are a potential source of leakage?
 - ① Watch my earlier talk :P <https://github.com/YuriyGuts/odsc-target-leakage-workshop>
- Is the format of the data consistent over time? (typical issue for long-lived systems)
- Are there outliers that are dangerous for the chosen optimization objective?
- Can be super insightful to view the data over time, over space, over target label



Feature Engineering

- Needs to be model-aware! Linear, tree-based, neural, FM, classic time series require different preprocessing and benefit from different feature engineering techniques
- Needs to be datatype-aware
 - ① For example, correctly distinguishing between a text feature and a categorical feature pays off here. By the way, language matters for text. We should auto-detect it too and derive features accordingly.
- Needs to be leakage-free (no peeking into test set, *very* careful peeking at the target)
- Needs to work at prediction time when the model is deployed, using the same raw data format but with no ground truth available
- Resources are finite! Latency and scalability are just as important as accuracy





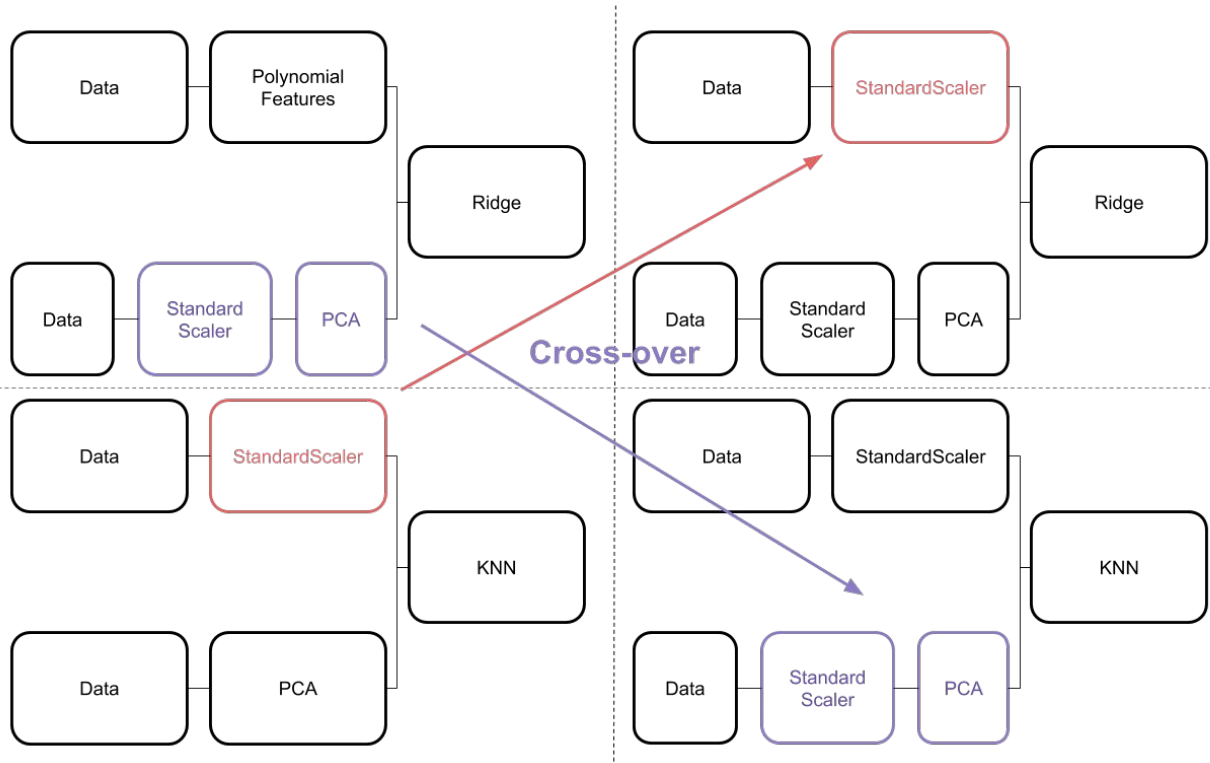
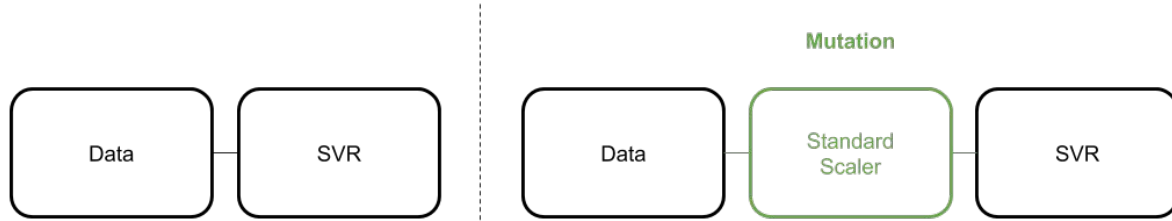
Modeling

- Accuracy is a must. Every percent pays off. Auto-ensembling can help too.
 - ① Steward Healthcare: www.datarobot.com/casestudy/reducing-costs-with-datarobot-at-steward-health-care/
More accurate predictions: **-1%** in nurse hours saves **\$2,000,000/year**; **-0.1%** of patient stay saves **\$10,000,000/year**
- No Free Lunch Theorem is very relevant, especially with prior business constraints
- Not enough to just have a “list of models”: need to construct pipelines dynamically.
 - ① Zoubin Ghahramani. Keynote at ICML 2018 AutoML workshop.
- Training from scratch / exhaustive search vs. transfer learning / metalearning.
- Efficient data usage, CPU/GPU and RAM usage, training time, and prediction latency are just as important as accuracy. Model search can also be constrained by time.
- Every model must be serializable, transferable, reproducible, autonomous.



Model Tuning

- Automated hyperparameter optimization (both for preprocessing and models)
 - ① An extensively studied problem in AutoML research.
See www.automl.org/book/ for current approaches and libraries.
tl;dr: scikit-optimize, hyperopt, BOHB.
- Automated feature reduction / redundancy detection
- Models need to have well-calibrated probability outputs
 - ① Guo et al. On Calibration of Modern Neural Networks, ICML 2017 arxiv.org/abs/1706.04599
- Pipeline optimization (also: Neural Architecture Search)
 - ① Also a subject of extensive academic interest
See www.automl.org/book/ for current approaches
Pipeline optimization AutoML powered by genetic programming: TPOT <https://github.com/EpistasisLab/tpot>





Evaluation and Selection

- Fair model comparison and ranking on out-of-sample data
- Analysis of data efficiency (learning curves), resource usage, prediction throughput
- Analysis of model stability out-of-sample
Typical issue: how well a time series model handles different forecasting horizons
- Recommending the best model, considering accuracy, transparency, and speed
- Making use of the data: retraining the best model on more data if needed
① Quiz: what to do with hyperparameters?
- Fair “apples-to-apples” comparison with externally developed models



Risk and Compliance

- Explaining feature importance, feature interactions, partial dependence
- Explaining the kinds and ranges of tuned hyperparameters and optimal values
- Explaining individual predictions in terms of original features
- Feature sensitivity analysis (effect of perturbations on predictions)
- “What-if” simulations and analysis (e.g. for ethical evaluation)
- Access to preprocessed/final modeling data for external reproducibility
- Auto-documenting the methodology, results, and insights!
- All of the above should be available for **every** model!




Software Construction

- Model needs to use the same dependencies it used during training.
OSS scientific packages also have bugs and breaking changes!
- Edge computing may require the model to be exportable and available offline
 - ① Exercise: think how you would make a full model pipeline available for scoring on iOS, Android, Raspberry Pi, ...
- Models need to expose an API and/or a UI
- Need to distinguish service health vs. input data health vs. model health
- Application needs to be generated according to the initial business problem setup (e.g. do we need to explain, predict, or prescribe/optimize)



Deployment

- Various physical locations (cloud, on-premise)
- Various deployment modes (bare metal, IaaS, PaaS, managed Hadoop service)
- Load balancing and rack awareness (leverage multiple nodes; leverage parallel / accelerated hardware if available; leverage optimized CPU instructions if available)
- Single sign-on, enterprise authentication providers, local PKI
- Integration with other BI/DS software for model consumption (Tableau, Qlik, ...)
- IT policies and compliance have the same relevance here as for any other enterprise software. OSS and security audit. Legacy software compatibility



CLOUD-NATIVE,
DOCKER,
KUBERNETES...

CENTOS 6



Model Maintenance

- Automated feature drift / response drift detection
The world never stops changing
- Feedback cycle detection
And we never stop changing the world
- Continuous learning
- Challenger models / auto-fallback to a more robust model

References

1. Rich Caruana (Microsoft Research). Open Research Problems in AutoML
<https://sites.google.com/site/automlwsicml15/>
2. AutoML: Methods, Systems, Challenges
<http://automl.org/book/>
3. Peter Prettenhofer: AutoML Class @ UCU Data Science School 2019
<https://github.com/pprett/aml-class-19>



yuriy.guts@gmail.com

[linkedin.com/in/yuriyguts](https://www.linkedin.com/in/yuriyguts)

datarobot.com/blog