DSC 102
Systems for Scalable Analytics

Rod Albuyeh

Topic 5: Model Building Systems

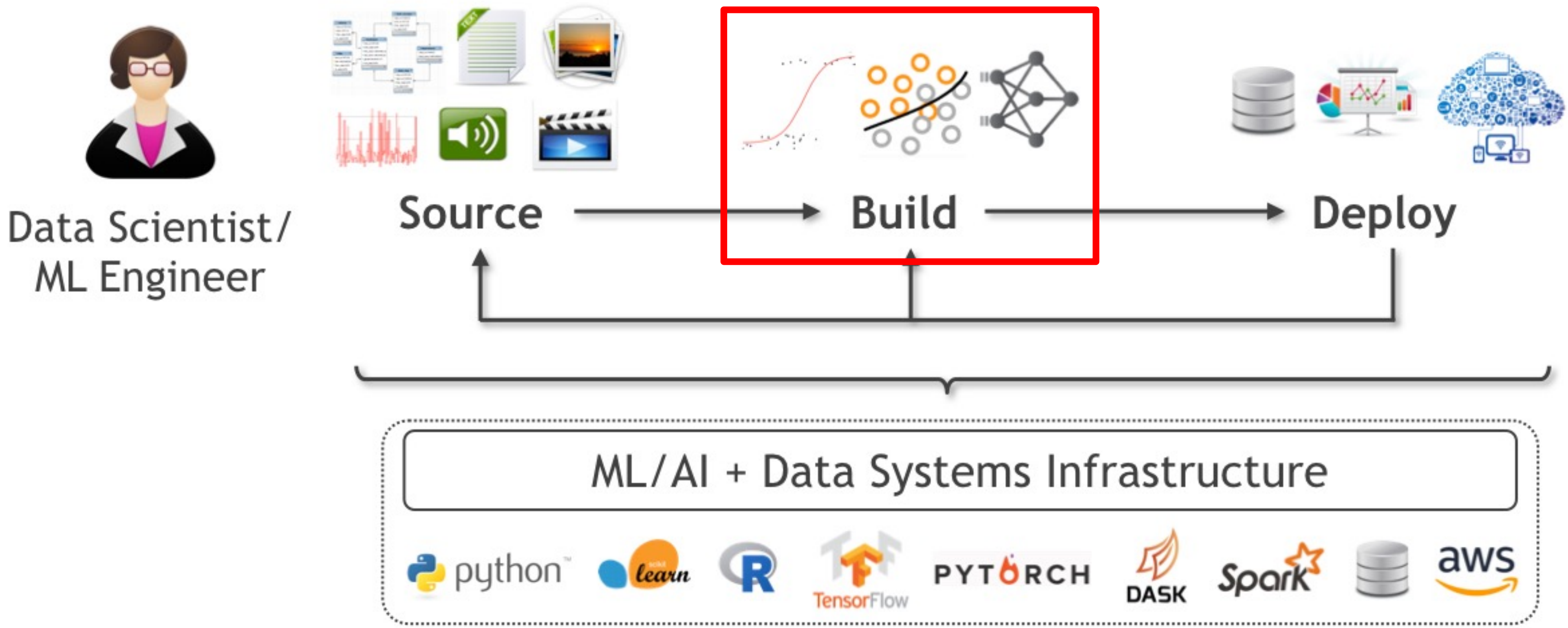
Admin

Reminder: 90%+ CAPE response rate for class yields 0.5% collective boost to final score.

Current response rate as of June 7th 9am is 44.64%.

Help both your instructor and DSC 102 improve!

The Lifecycle of ML-based Analytics



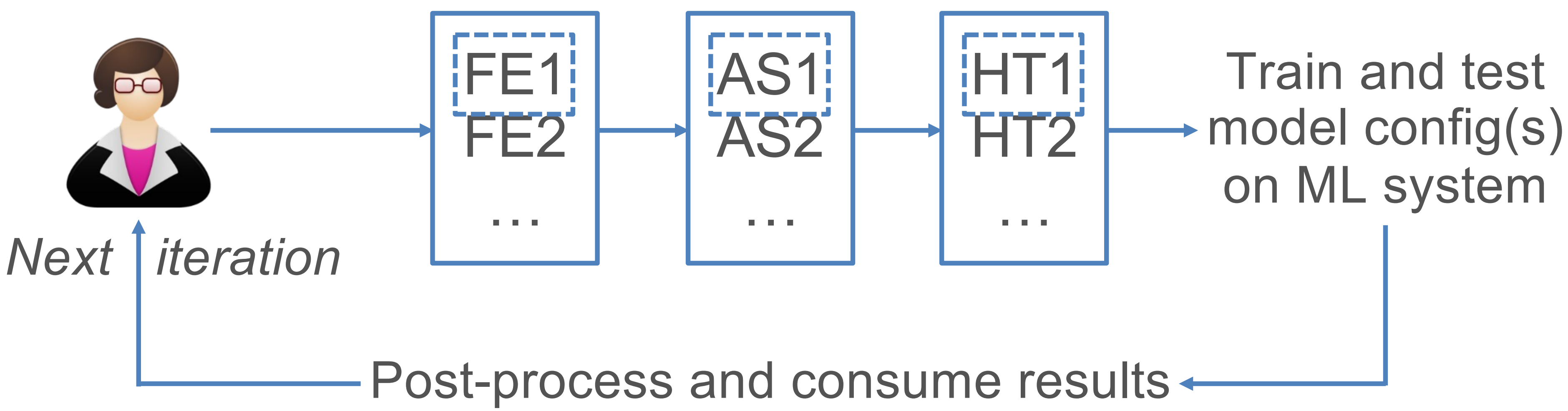
Data acquisition
Data preparation

Feature Engineering
Training & Inference
Model Selection

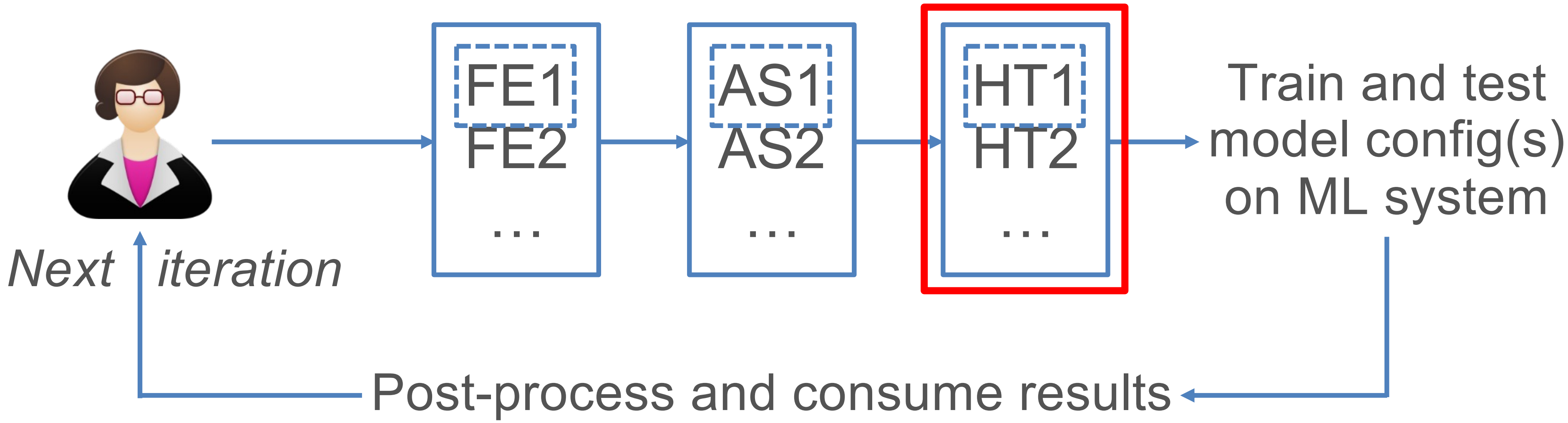
Serving
Monitoring

Model Selection Process

- Model selection is usually an *iterative exploratory* process with human making decisions on FE, AS, and/or HT
- Increasingly, automation of some or all parts possible: **AutoML**



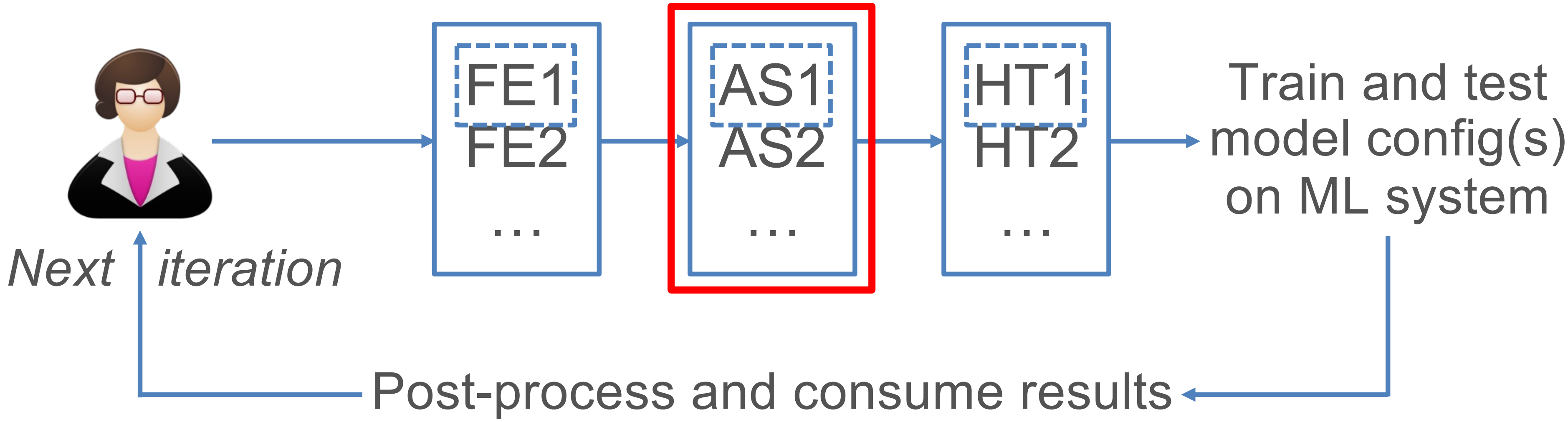
Hyper-Parameter Tuning



Hyper-Parameter Tuning

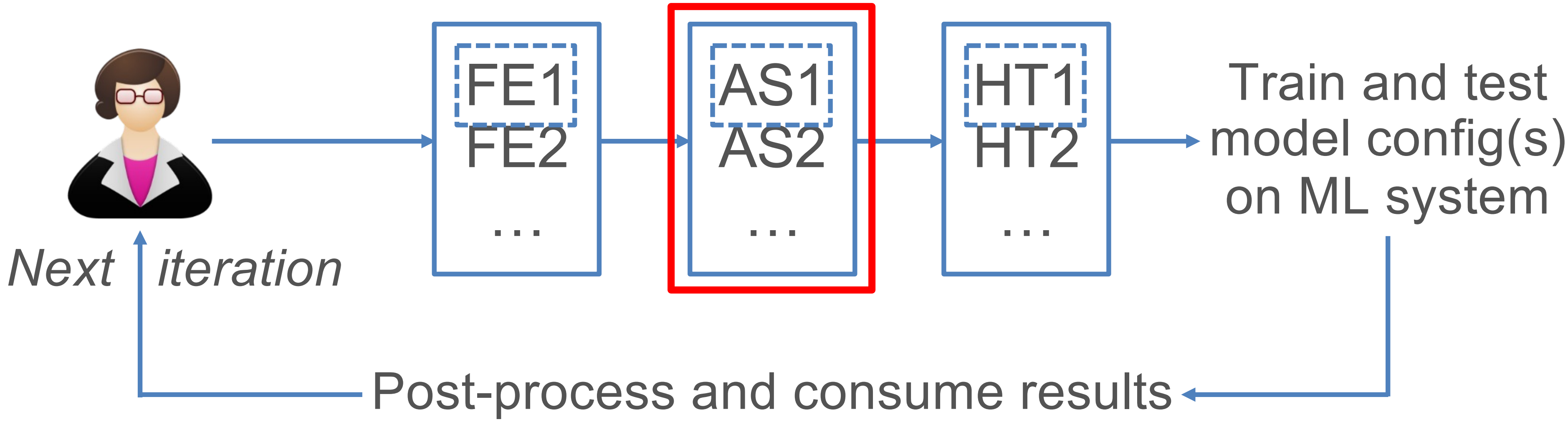
- **Hyper-parameters:** Knobs for an ML model or training algorithm to control *bias-variance* tradeoff in a dataset-specific manner to make learning effective
- **Examples:**
  - GLMs: L1 or L2 *regularizer* to constrain weights
  - All gradient methods: *learning rate*
  - Mini-batch Stochastic Gradient Descent: *batch size*
  - *Others?*
- HT is an "outer loop" around training/inference
- Most common approach: **grid search**; pick set of values for each hyper-parameter and take cartesian product
- Also common: **random search** to subsample from grid
- Complex AutoML heuristics exist too for HT, e.g., Bayesian

Algorithm Selection in “classical” ML



- Not much to say; ML user typically picks models/algorithms in advance
- Best practice: first train more simple models (log. reg.) as baselines; then try more complex models (XGBoost)
- **Ensembles:** Build diverse models and aggregate predictions. Even for tabular data, ensembles yield better results and often win Kaggle comps with a few % boost in performance.

Architecture Selection in DL



- More critical in DL; neural arch. is **inductive bias** in classical ML parlance; controls feature learning and bias-variance tradeoff
- Some applications: Many off-the-shelf pre-trained DL models to do "transfer learning," e.g., see models at HuggingFace.co
- Other applications: Swap pain of hand-crafted feature eng. for pain of neural arch. eng.! Neural arch probably a better interview skill ☺

Automated Model Selection / AutoML

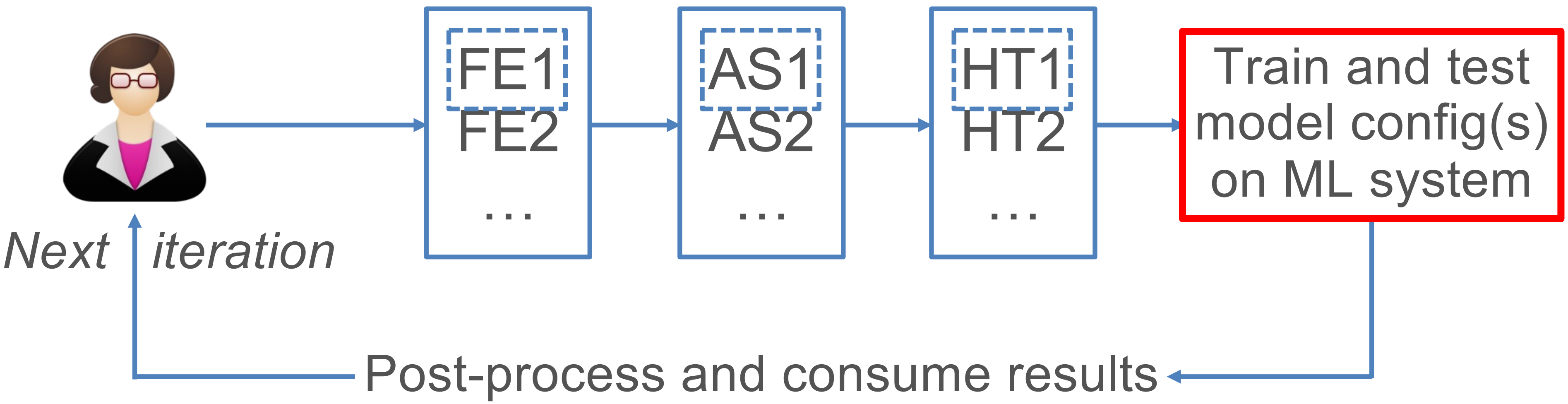
***Q:** Can we automate the whole model selection process?*

- It depends. HT and most of FE already automated mostly in practice; (neural) AS is often application-dictated
- AutoML tools/systems now aim to reduce data scientist's work; or even replace them?! ;)

- **Pros:** Ease of use; lower human cost; easier to audit; improves ML accessibility
- **Cons:** Higher resource cost; less user control; may waste domain knowledge; may leave performance on the table
- Pareto-optima; hybrids possible

But: The data sourcing + feature engineering stage is still very hard to automate and tends to be domain / context specific!

Scalable ML Training and Inference



Then deploy?

Major ML Model Families/Types

Generalized Linear Models (GLMs); from statistics

Bayesian Networks; inspired by causal reasoning

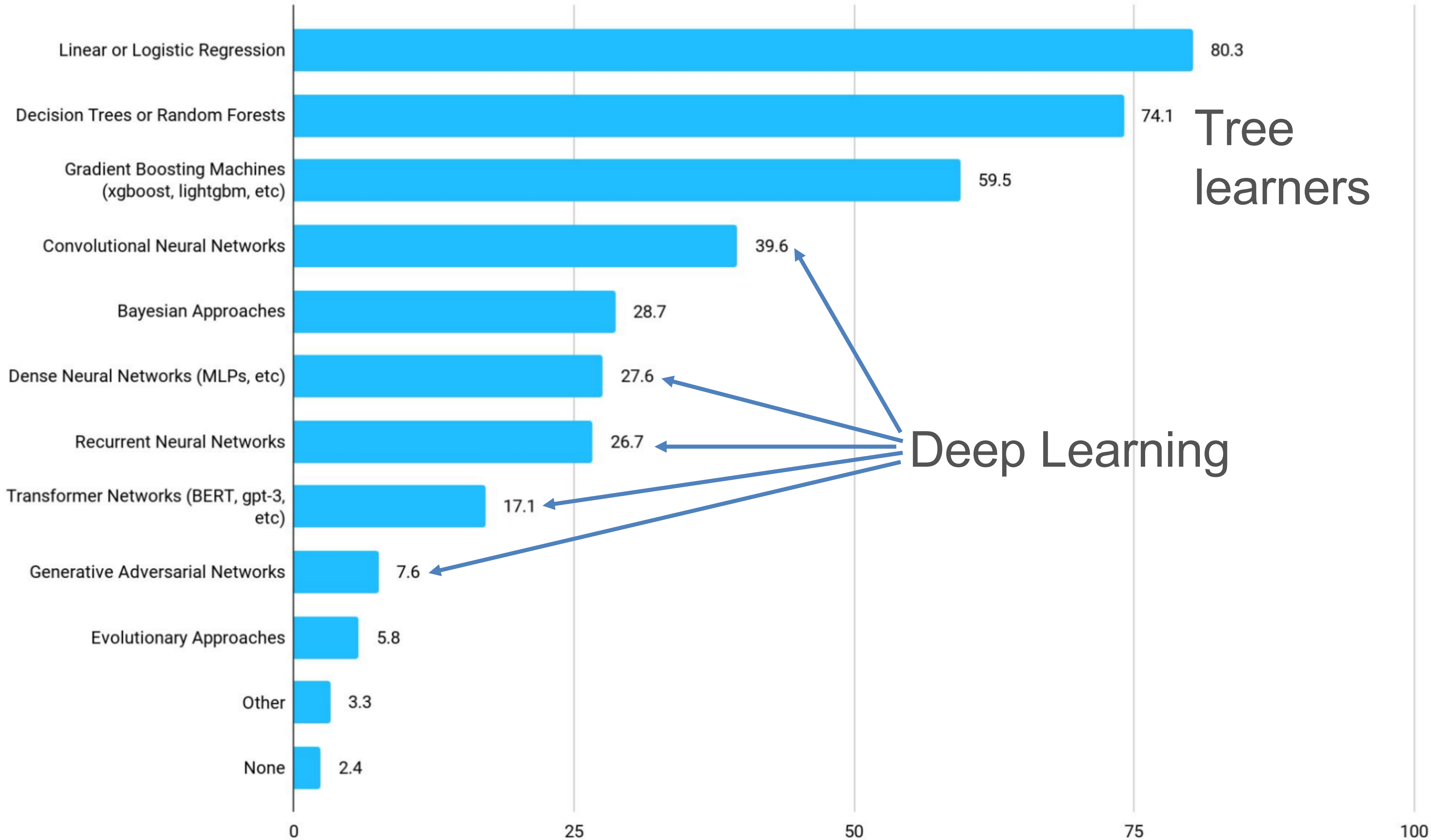
Decision Tree-based: CART, Random Forest, Gradient-Boosted Trees (GBT), etc.; inspired by symbolic logic

Support Vector Machines (SVMs); inspired by psychology

Artificial Neural Networks (ANNs): Multi-Layer Perceptrons (MLPs), Convolutional NNs (CNNs), Recurrent NNs (RNNs), Transformers, etc.; inspired by brain neuroscience

Unsupervised: Clustering (e.g., K-Means), Matrix Factorization, Latent Dirichlet Allocation (LDA), etc.

ML Models in Kaggle 2021 Survey



https://tinyurl.com/3b2pp6nr

Scalable ML Training Systems

- Scaling ML training is involved and model type-dependent
- Orthogonal Dimensions of Categorization:

1. **Scalability:** In-memory libraries vs Scalable ML system (works on larger-than-memory datasets)
2. **Target Workloads:** General ML library vs Decision tree-oriented vs Deep learning, etc.
3. **Implementation Reuse:** Layered on top of scalable data system vs Custom from-scratch framework

Major Existing ML Systems

General ML libraries:

In-memory: scikit-learn, R, RAY

Disk-based files: sas, DASK

Layered on RDBMS/Spark: MADlib, Apache Spark MLlib

Cloud-native: Azure Machine Learning, Amazon SageMaker

"AutoML" platforms: DataRobot, H2O.ai

Decision tree-oriented:

dmlc XGBoost, Microsoft LightGBM

Deep learning-oriented:

TensorFlow, PyTorch

Scalable ML Inference

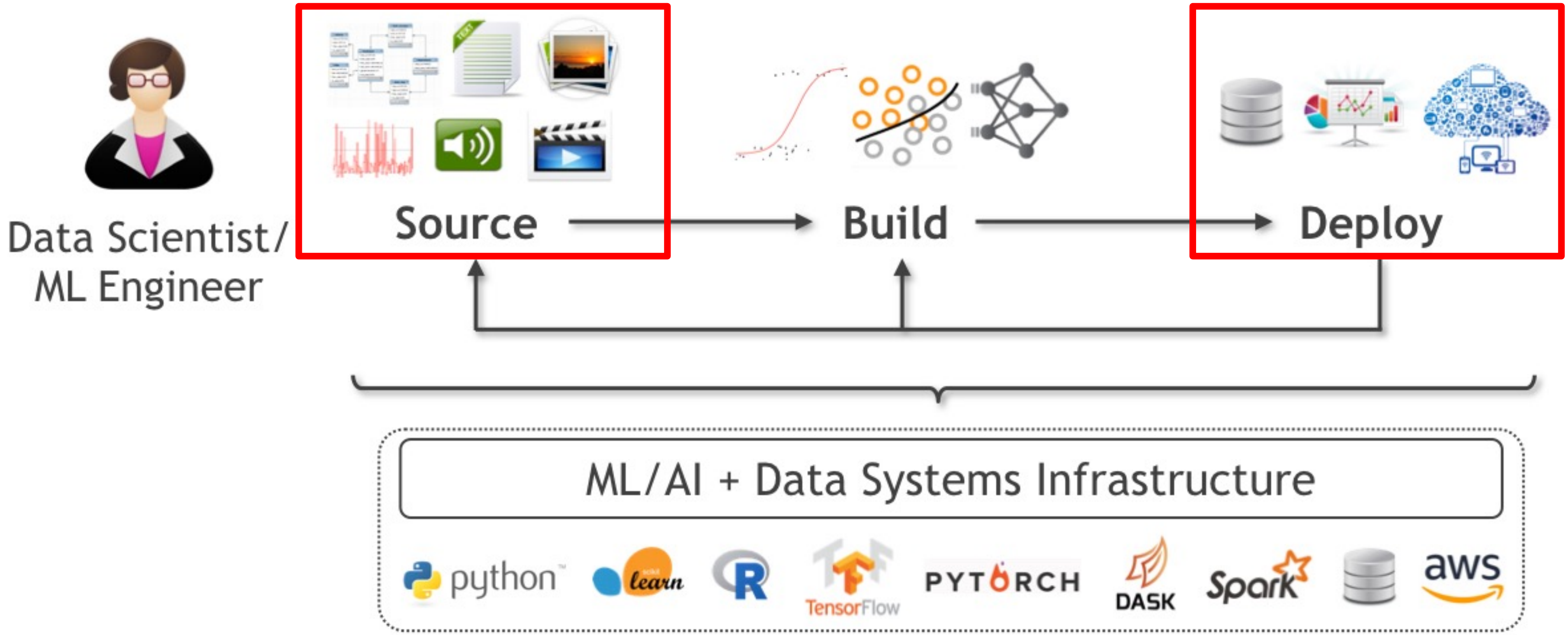
- A trained/learned ML model is just a prediction function:

$$f : \mathcal{D}_X \rightarrow \mathcal{D}_Y$$

Q: *Given large dataset of examples, how to scale inference?*

- Assumption 1: An example fits entirely in DRAM
- Assumption 2: *f* fits entirely in DRAM
- If both hold, trivial access pattern: single filescan, apply per-tuple function *f*, write output. How to do this with MapReduce?
- If either fails, access pattern becomes more complex and dependent on breaking up internals of *f* to stage access to data for partial computations

The Lifecycle of ML-based Analytics



Data acquisition
Data preparation

Serving
Monitoring

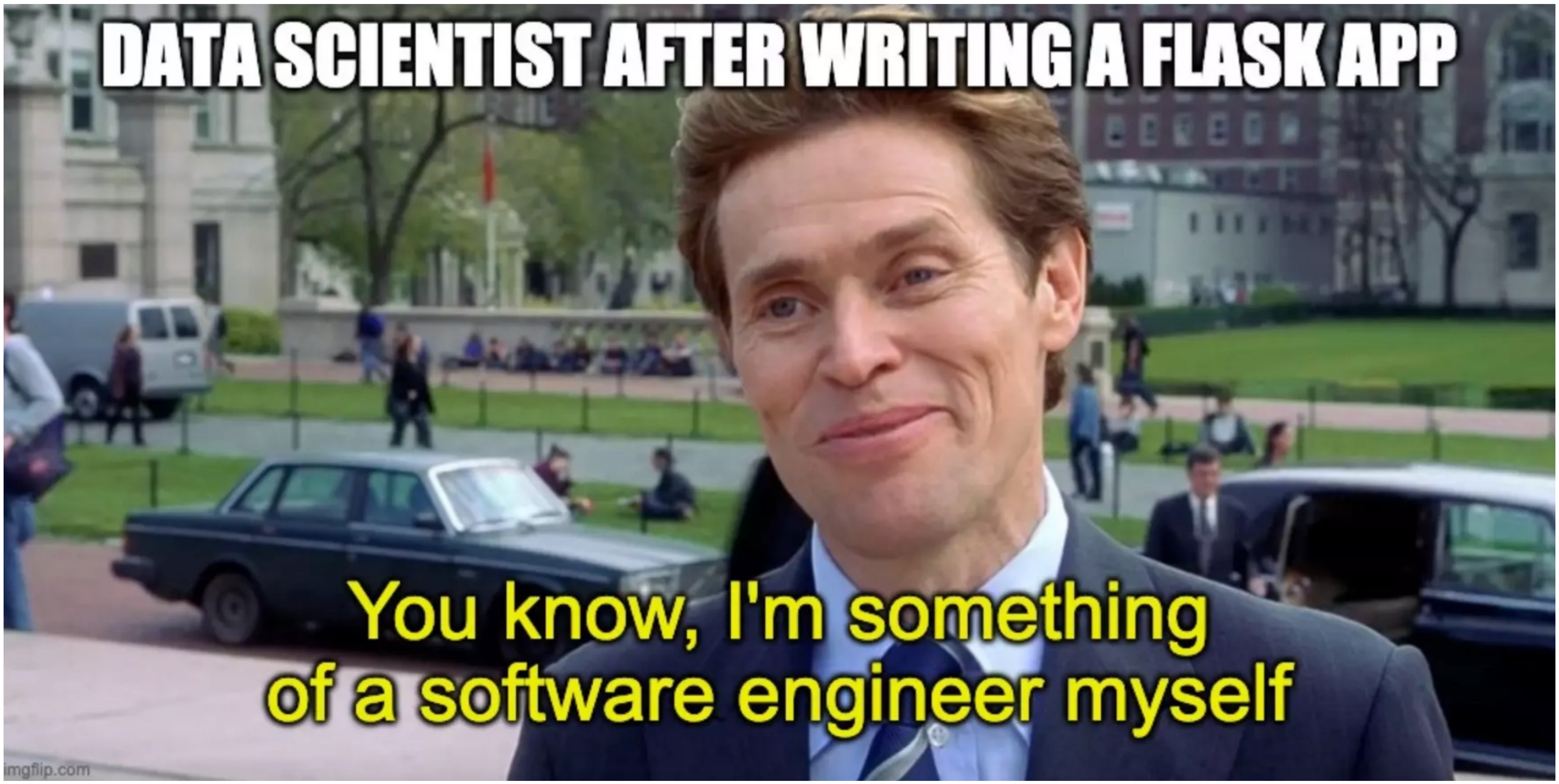
Deployment involves re-applying any transformations and the final f()

Model Serving / Deployment

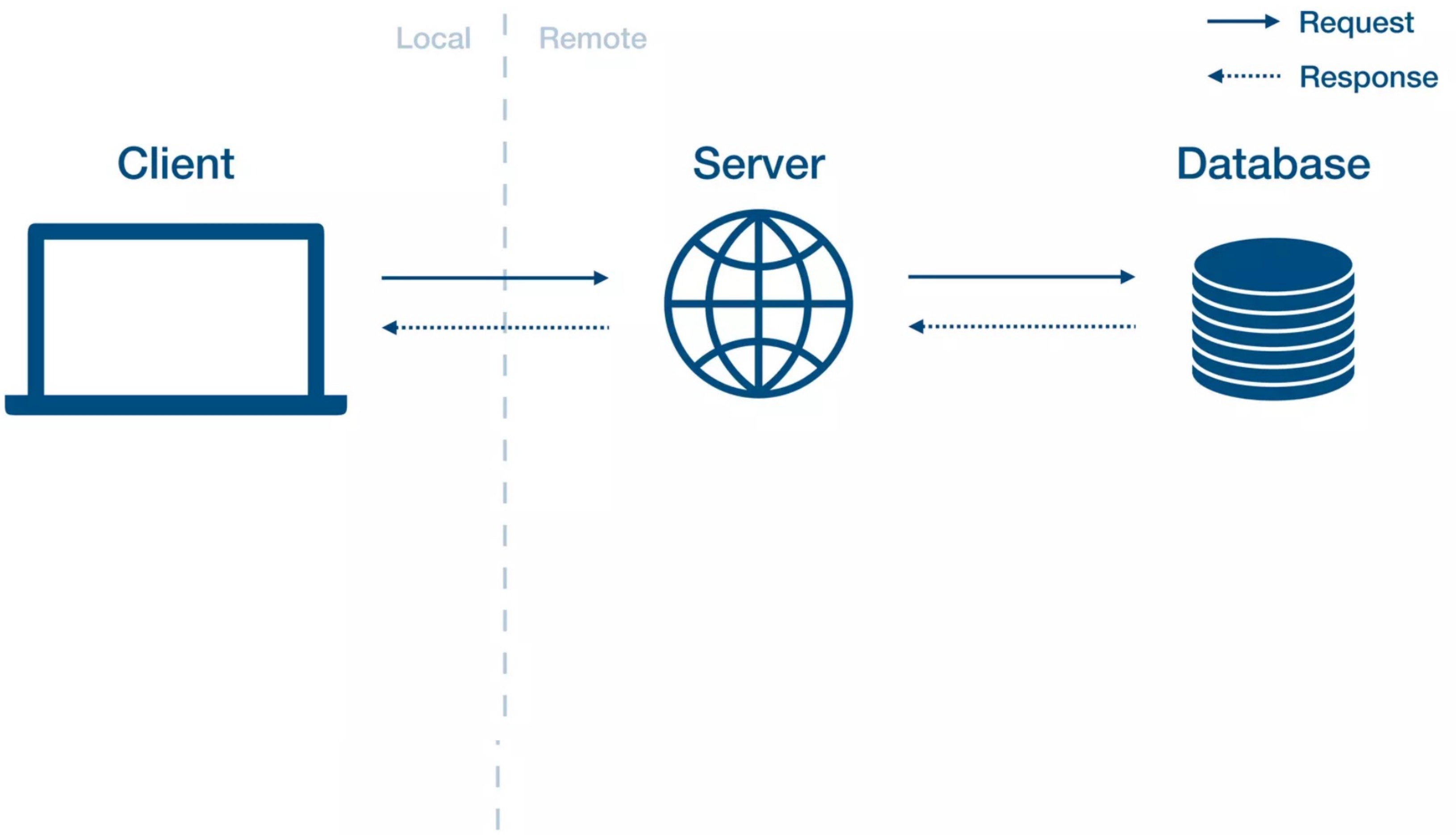
- A trained/learned ML model is just a prediction function:

$$f : \mathcal{D}_X \to \mathcal{D}_Y$$

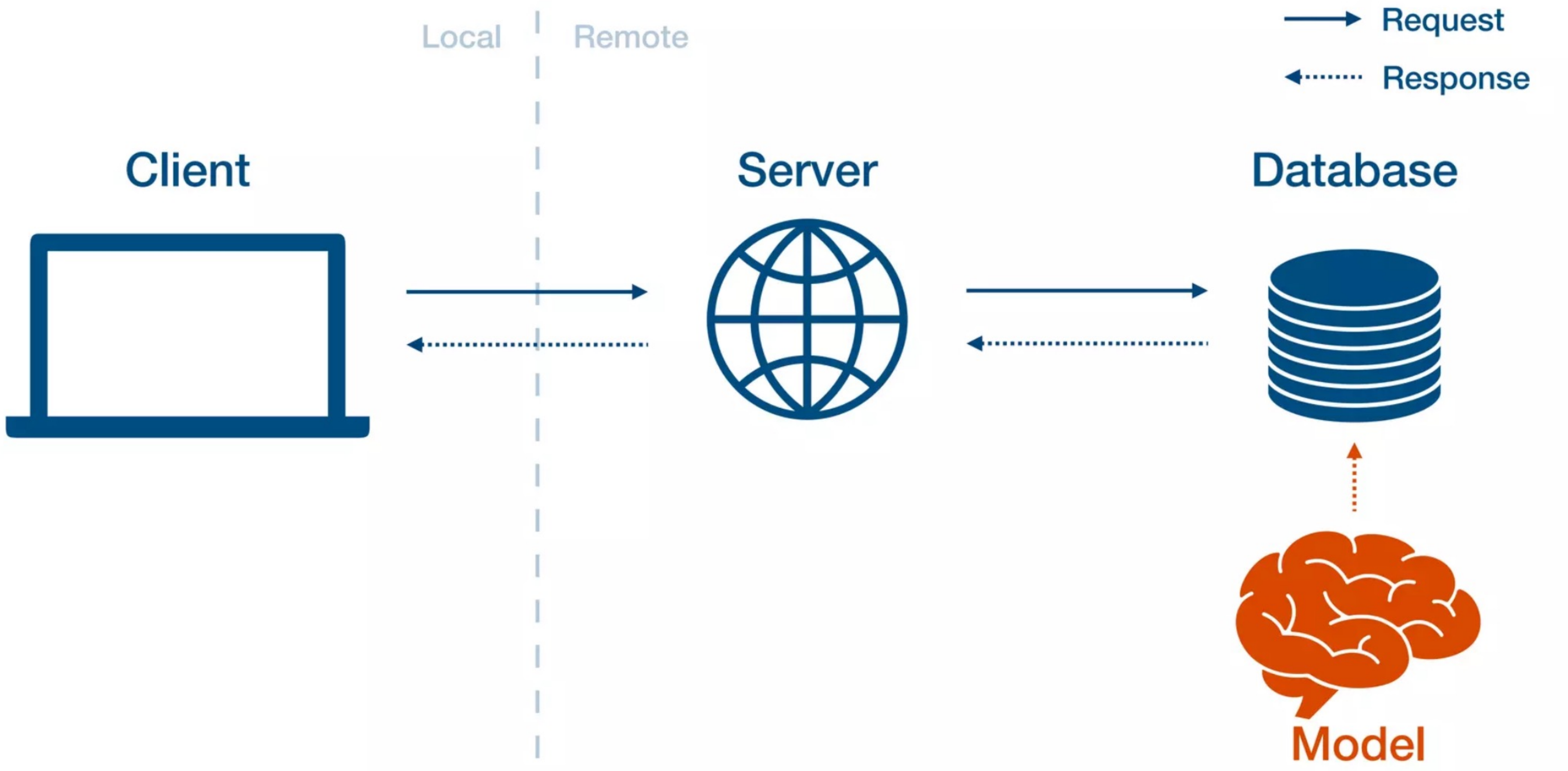
- A major consideration is, online/realtime vs. offline/batch.
- In the offline scenario, serving a model is more trivial where it is another processing function that we apply.
- In the online scenario, we become concerned with millisecond latency for responses, setting up APIs, load balancing, and monitoring.


DATA SCIENTIST AFTER WRITING A FLASK APP
You know, I'm something
of a software engineer myself
imgflip.com

Where to host the model?

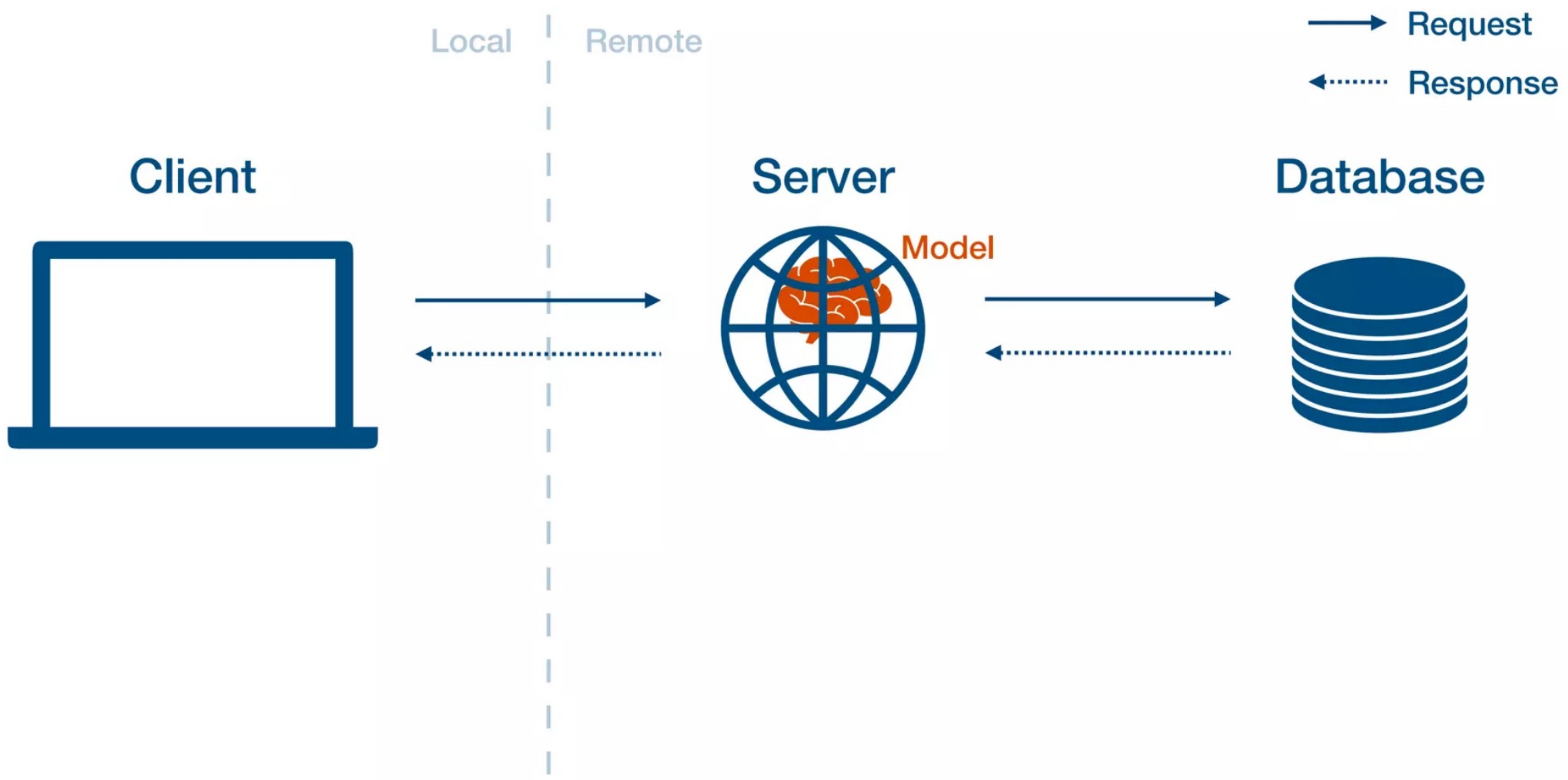


Batch/offline prediction



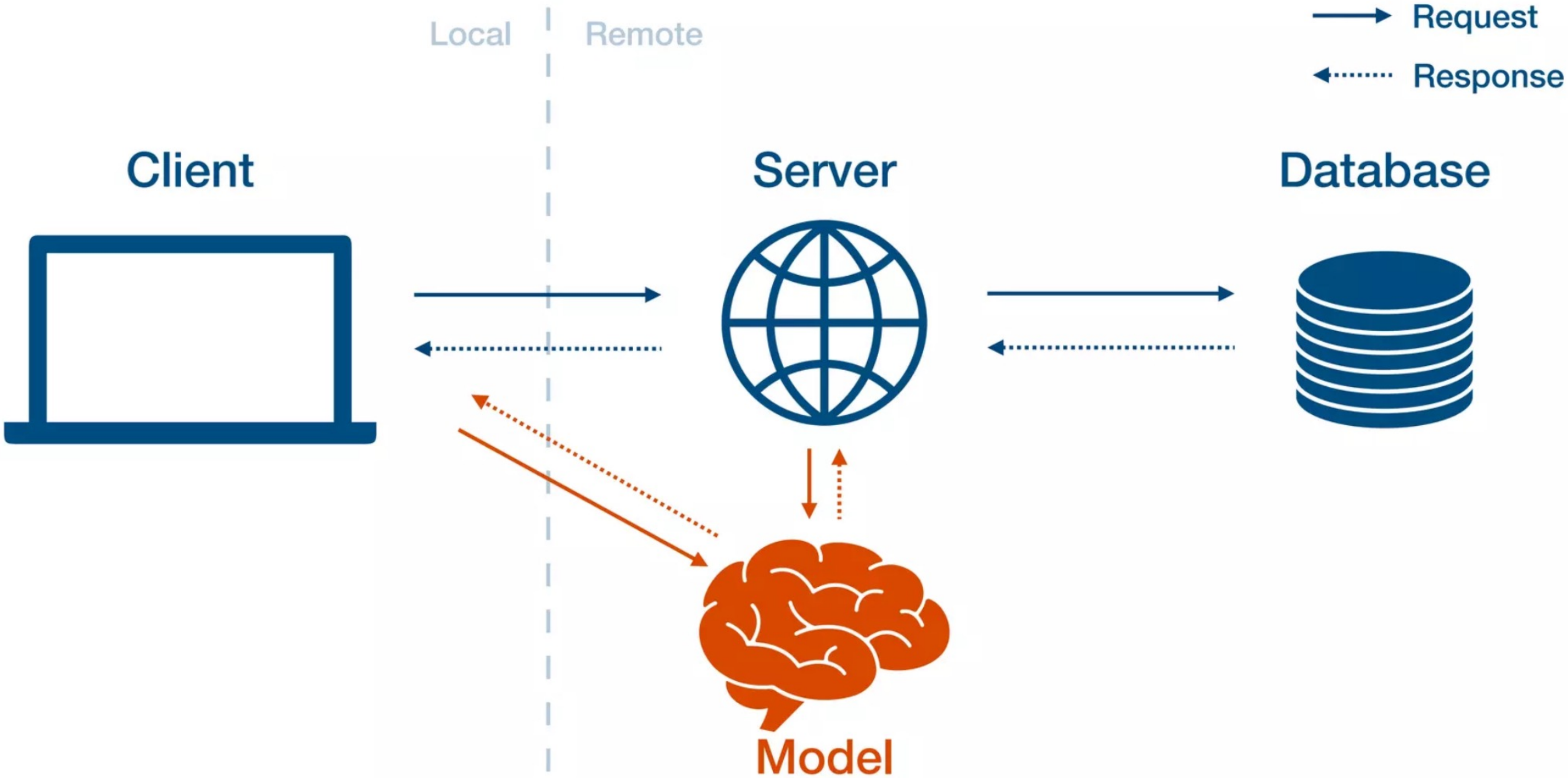
- Periodically run your model on new data and cache the results in a database
- Works if the universe of inputs is relatively small (e.g., 1 prediction per user)

Realtime/Online prediction



Embedded within an application? What if the application is in Java?

Model-as-a-Service



- Run your model on its own web server
- The backend (or the client itself) interact with the model by making requests to the model service and receiving responses back

Want more info on deployment?

140 (excellent) slides with associated videos to be found here:
https://fullstackdeeplearning.com/spring2021/lecture-11/

Week	Topic and Papers	Slides, Videos; Review Forms, Deadlines
0	**Introduction, ML Lifecycle Overview, and Basics**	Slides: **PDF PPTX** **Video 1**; **Video 2**; **Video 3**
	Readings: SIGMOD tutorial 1, SIGMOD tutorial 2, Berkeley report	
1-2	**Topic 1: Classical ML Training at Scale**	Slides: **PDF PPTX** **Video 1**; **Video 2**
	For review: Parameter Server	**Review 1 Form**; due 10/6
	For review: XGBoost	**Review 2 Form**; due 10/13
	More readings: MADlib, MLlib, Mahout, GraphLab, AWS Sagemaker	-
1	**No class on 10/8**	-
3	**Topic 2: Deep Learning Systems**	Slides: **PDF PPTX** **Video 1**; **Video 2**; **Video 3**
	For review: TensorFlow (Talk slides)	**Review 3 Form**; due 10/20
	More readings: Horovod, Distributed PyTorch, TVM	-
4-5	**Topic 3: Feature Engineering and Model Selection Systems**	Slides: **PDF PPTX** **Video 1 Video 2**
	For review: Cerebro	**Review 4 Form**; due 10/27
	More readings: MSMS, Hyperband, ASHA, Vizier, Columbus, Vista	-
5	**Review Session 1 on 11/3 (tentative)**	Slides: **PDF**
5	**Exam 1 on 11/5**	-
6	**Topic 4: Data Sourcing and Organization for ML**	Slides: **PDF PPTX** **Video 1**; **Video 2**; **Video 3**
	For review: TFDV	**Review 5 Form**; due 11/3
	More readings: Deequ, Snorkel, Ground, SortingHat, Hamlet	-
7	**Guest Lecture by Matei Zaharia (Databricks and Stanford) on MLFlow on 11/17**	**Video**; **Slides PDF**
7-9	**Topic 5: ML Deployment**	Slides: **PDF PPTX** **Video 1**; **Video 2**
	For review: Clipper	**Review 6 Form**; due 11/12
	More readings: TF Serving, Uber PyML, Hummingbird, Federated ML	-
8	**Guest Lecture by Angela Jiang (Determined AI) on Determined DL Platform on 11/24**	**Video**; **Slides PDF**
8	**Thanksgiving Holiday on 11/26**	-
9	**Guest Lecture by Joshua Patterson (NVIDIA) on RAPIDS on 12/1**	**Video**; **Slides PDF**
9-10	**Topic 6: ML Platforms and Feature Stores**	Slides: **PDF PPTX** **Video1**; **Video 2**
	For review: ML systems technical debt	**Review 7 Form**; due 11/17
	For review: TensorFlow Extended	**Review 8 Form**; due 12/3
	More readings: MLFlow, Michelangelo	-

CSE 234/291 from Fall'20 with lecture videos on Youtube

https://cseweb.ucsd.edu//classes/fa20/cse291-d/schedule.html

DSC 102 focuses on thinking about the <u>fundamentals of scalable analytics systems</u>

1. "**Systems**": What resources does a computer have? How to store and efficiently compute over large data? What is cloud?
2. "**Scalability**": How to scale and parallelize data-intensive computations?
3. **For "Analytics"**:
   1. **Source**: Data acquisition & preparation for ML
   2. **Build**: Model selection & deep learning systems
   3. **Deploying** ML models
4. Hands-on experience with scalable analytics tools