KASETSART UNIVERSITY

01418262 Machine Learning Systems

Model Development (Part 1)

Sarach Tuomchomtam, Ph.D.

sarach.t@ku.th

	baths	bedrooms	Area_in_Marla	city_0	city_1	city_2	random	price
44749	2	2	-0.682582	0	0	1	0.723952	1
128534	5	5	0.166235	1	0	0	0.812452	4
29224	3	4	-0.231648	0	1	0	0.917679	2
89063	2	2	-0.496903	0	1	0	0.864875	0
42944	2	2	-0.682582	0	1	0	0.030144	1
...	...	...	...	...	...	...	...	...
30618	2	2	-0.337750	0	1	0	0.270109	0
69578	2	2	-0.496903	1	0	0	0.659784	2
64395	5	5	-0.364275	1	0	0	0.545952	3
65830	7	5	1.492512	1	0	0	0.024733	4
108078	5	5	-0.099020	0	0	1	0.485322	4

Outline

Model Development

Model Selection

Ensembles

Tracking and Versioning

Debugging Models

Deep Neural Network

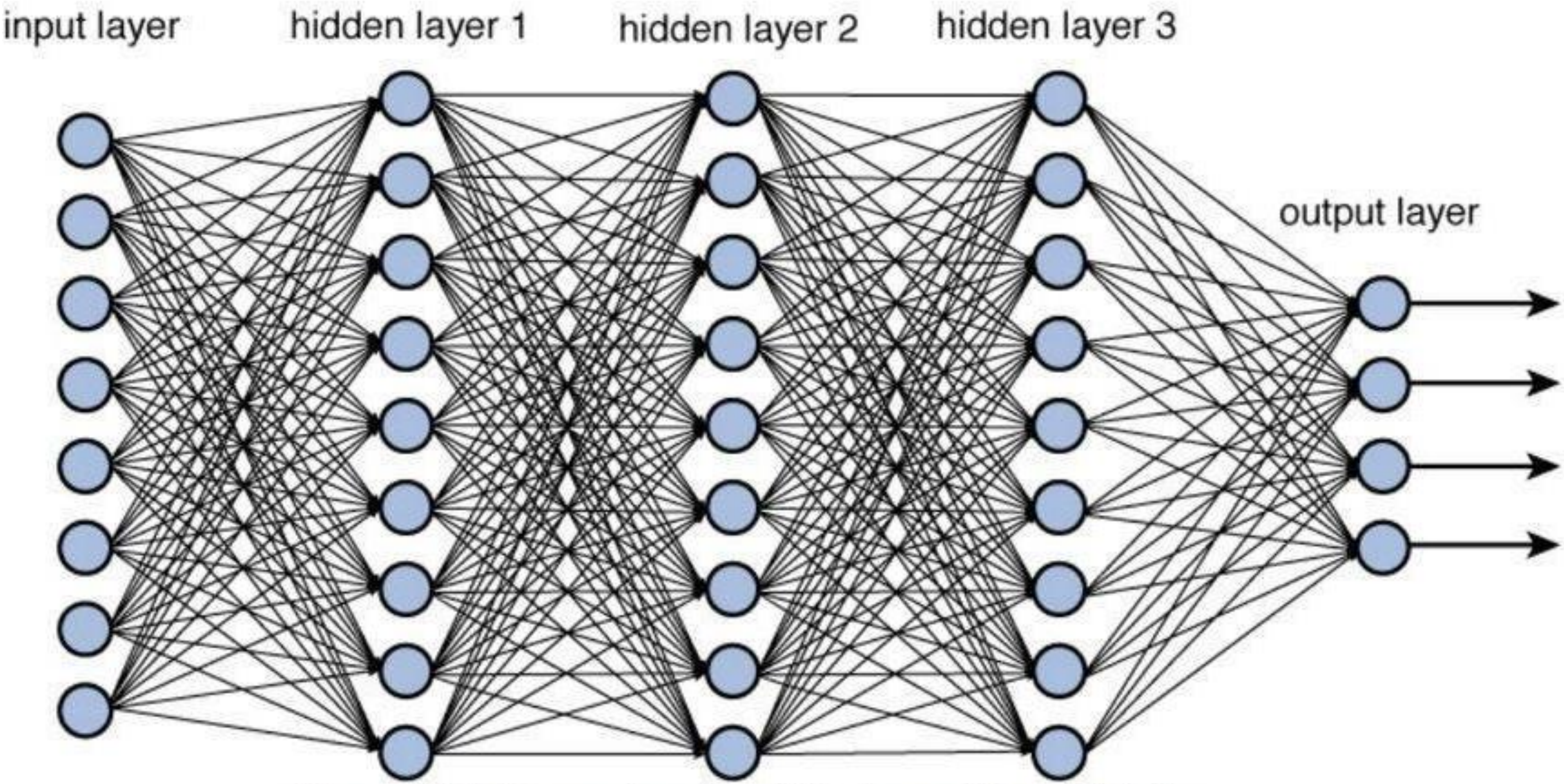


Figure 12.2 Deep network architecture with multiple layers.

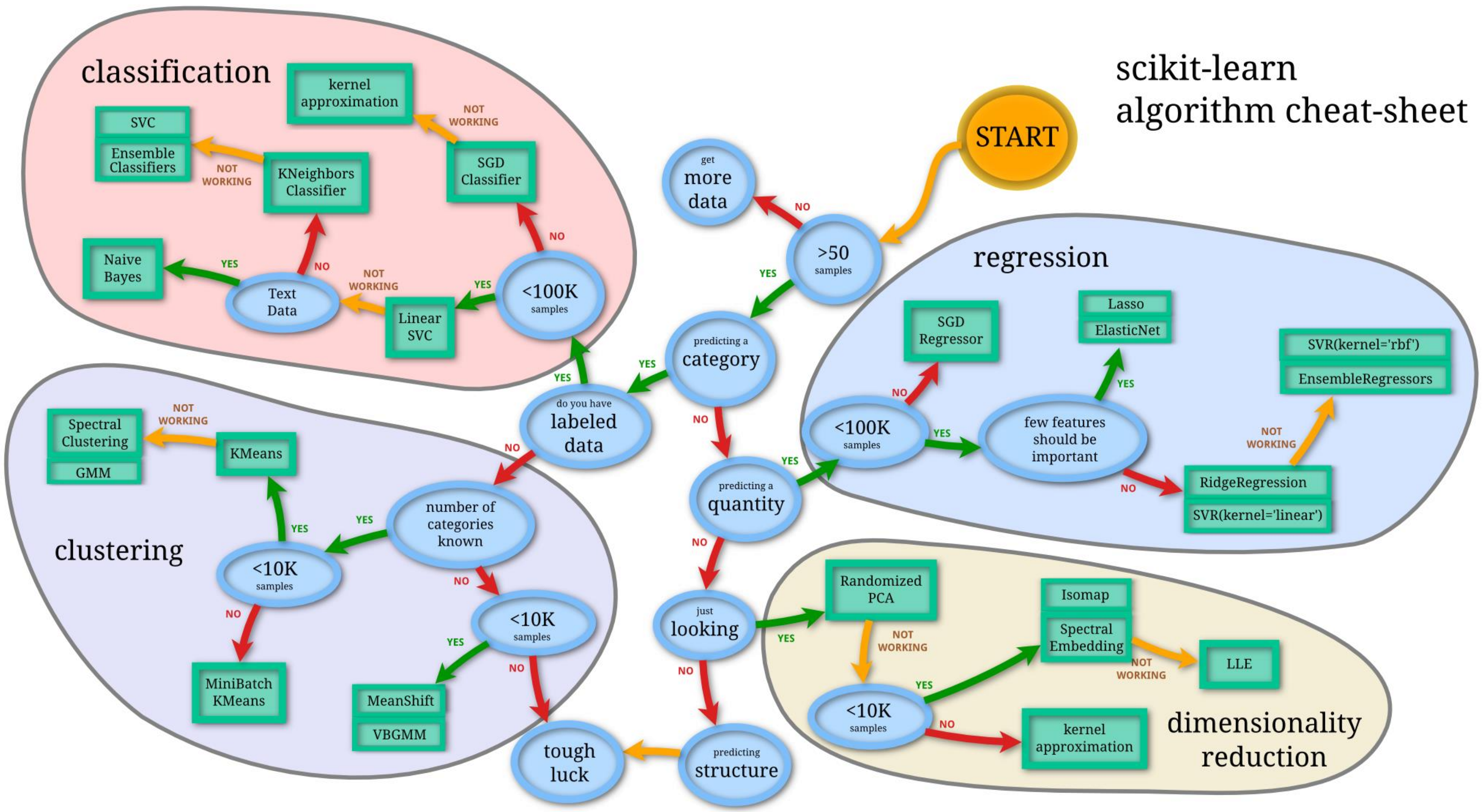
Model Selection

Machine vs. Deep Learning Algorithms

Classical ML algorithms are not going away.

Focus on a set of models suitable for your problem.

Non-neural network algorithms tend to be more explainable.



Six Tips for Model Selection

1. Avoid the state-of-the-art trap.
2. Start with the simplest models.
3. Avoid human biases in selecting models.
4. Evaluate good performance now versus good performance later.
5. Evaluate trade-offs.
6. Understand your model's assumptions.

Six Tips for Model Selection

1. Avoid State-of-the-Art Trap

Researchers often only evaluate models in academic settings.

“It performs better than existing models on some static datasets.”

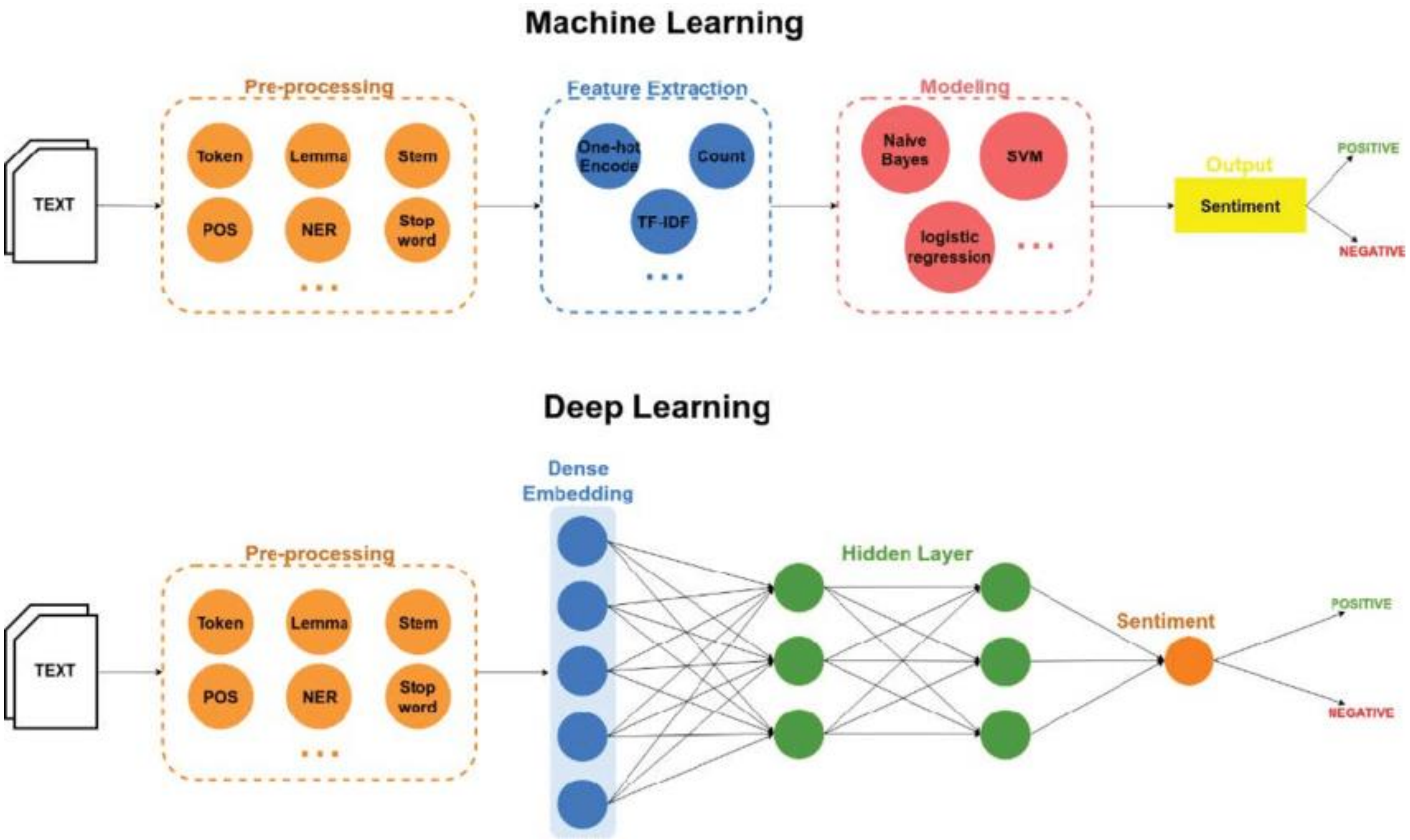
Use the simpler solution if it can solve your problem that much cheaper than state-of-the-art models.

Task	Leading Methods
Semantic Segmentation	HRNet-OCR | Efficient-Net-L2 | ResNeSt-269 | VMVF
Image Classification	FixEfficientNet | BiT-L | Wide-ResNet-101 | Branching CNN
Object Detection	Efficient-Det-D7x | Rodeo | Patch Refinement | IterDet
Sentiment Analysis	BERT | T5–3B | NB-weighted-BON + dv-cosine
Language Modeling	Megatron-LM | GPT-3 | GPT-2
Text Classification	XLNet | USE_T + CNN | SGC
Question Answering	T5–11B | SA-Net on Albert | TANDA-RoBERTa
Machine Translation	Efficient-Det-D7x | Rodeo | Patch Refinement | IterDet
Recommender System	Bayesian time SVD++ // flipped w/ Ordered Probit Reg | EASE | H+Vamp Gated
Speech Recognition	ContextNet + Noisy Student | ResNet + BiLSTMs | LiGRU | Large-10h-LV-60k

2. Start with the Simplest Models

Simpler models are easier to deploy, understand, and debug.

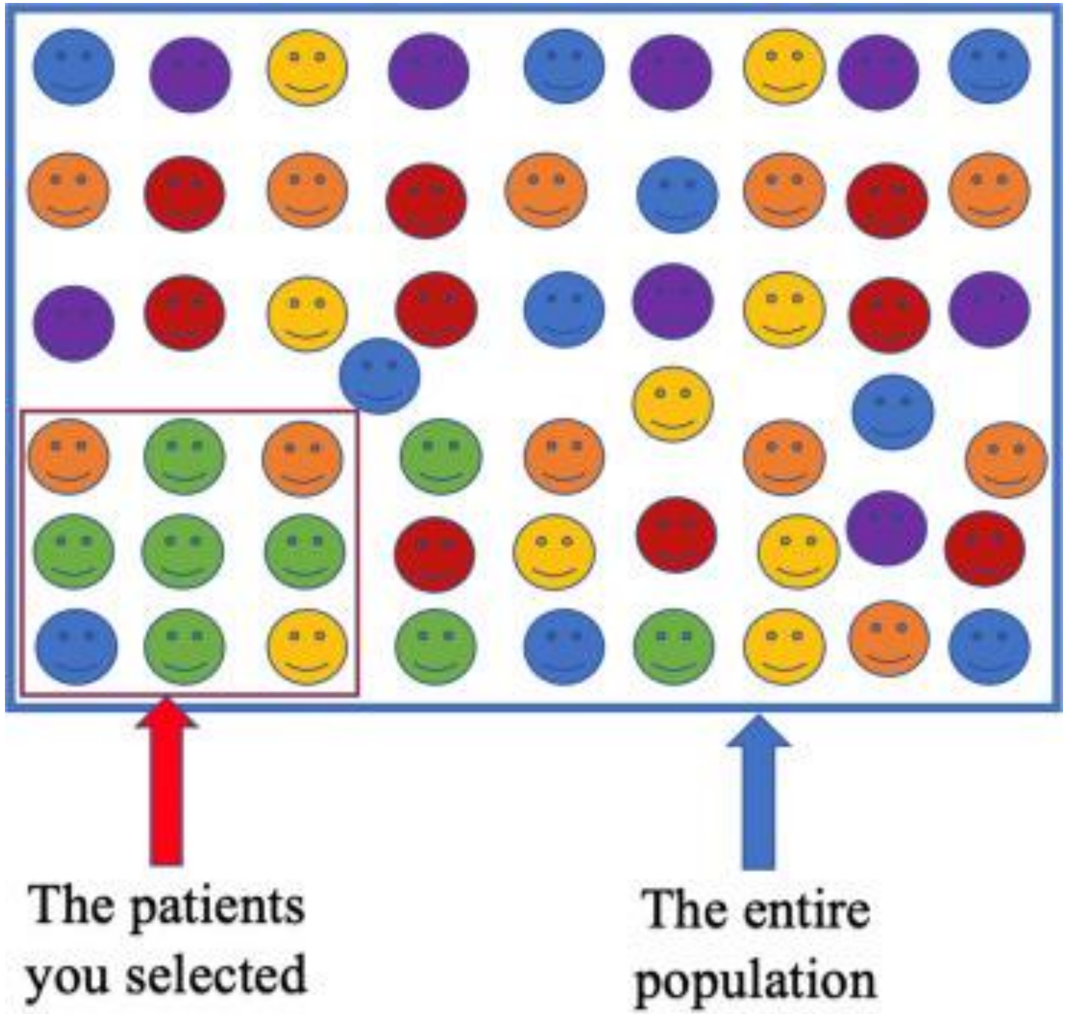
The simplest model serves as a baseline to which you can compare your more complex models.



3. Avoid Human Biases in Selecting Models

Part of the process is to experiment with different features and hyperparameters.

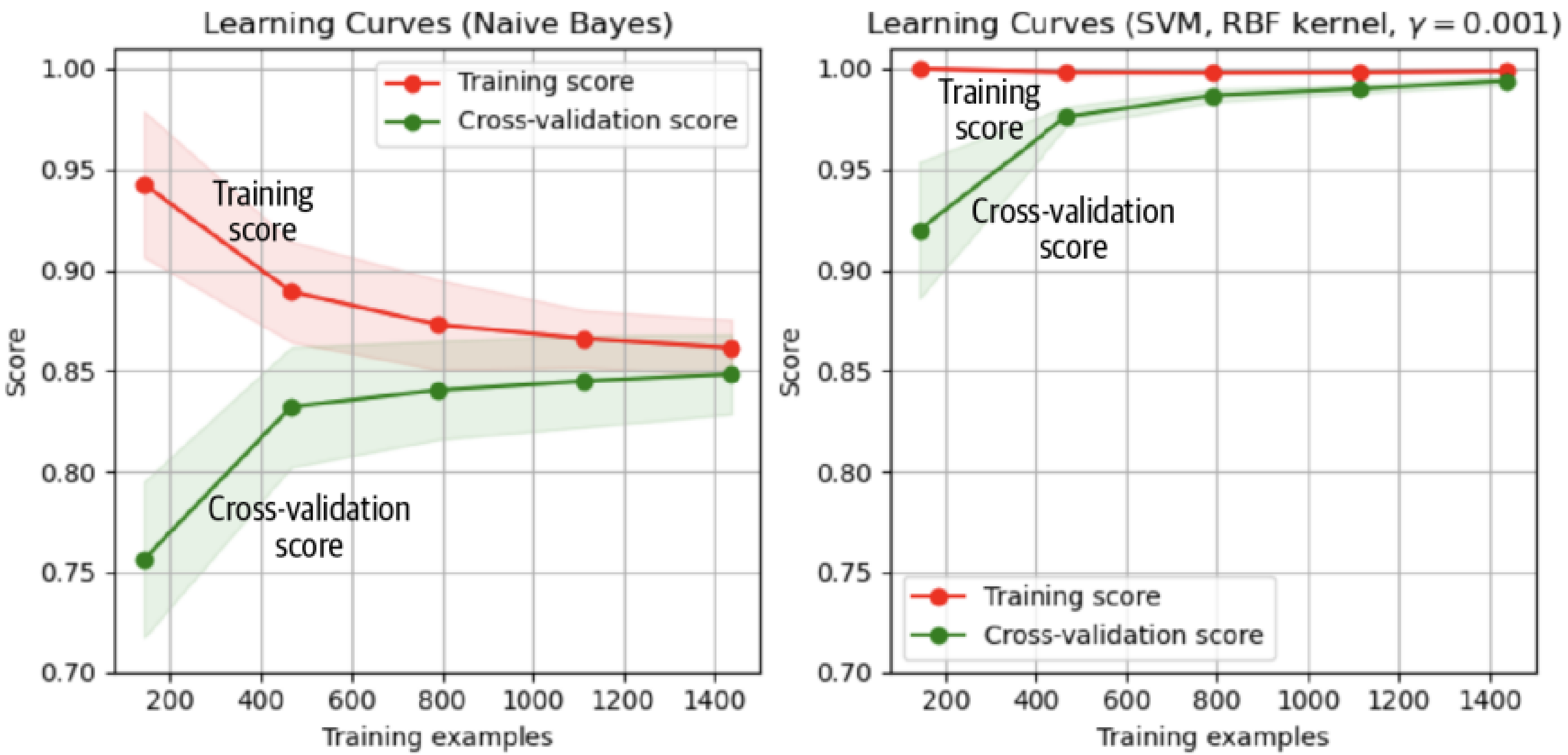
If you run 100 experiments for an architecture, you might need to run 100 experiments for the other architecture too.



4. Performance Now vs. Performance Later

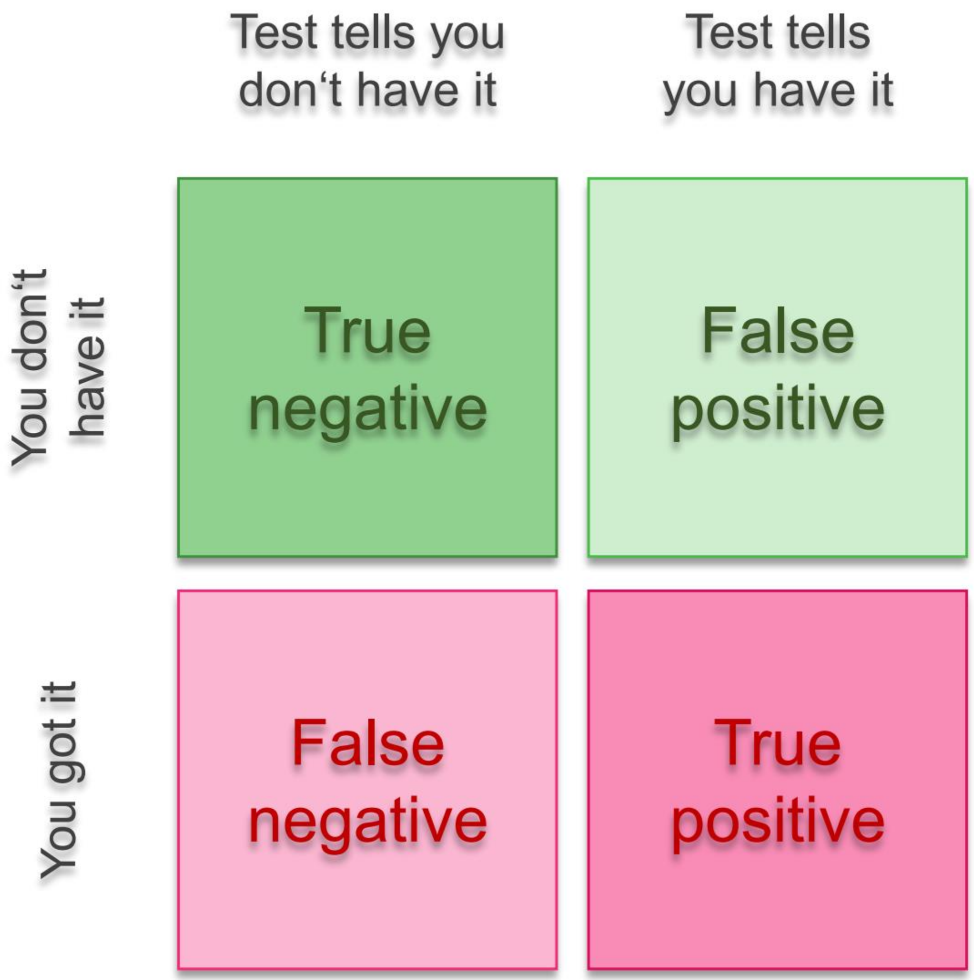
The best model now does not always mean the best model two months from now.

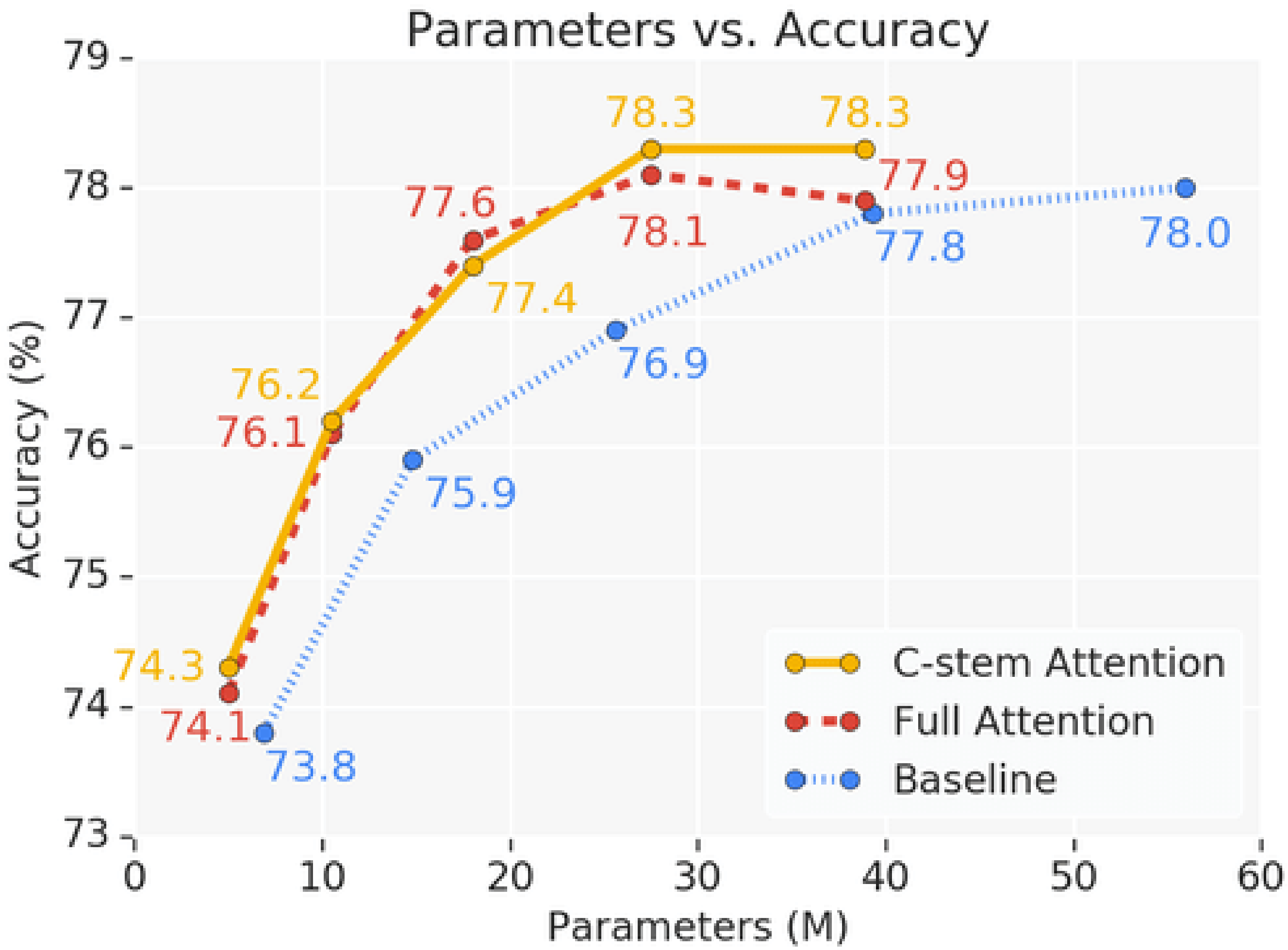
A learning curve is a plot of performance against the number of training samples.



Six Tips for Model Selection

5. Evaluate Trade-offs





6. Understand Your Model's Assumptions

- It's possible to predict Y based on X.
- Examples are drawn from the same joint distribution.
- Similar inputs are transformed into similar outputs.
- Tractable to compute the probability P(Z|X).
- Decision boundaries are linear for linear a classifier.
- Features are independent of each other given the class.
- Data is normally distributed.

Six Tips for Model Selection

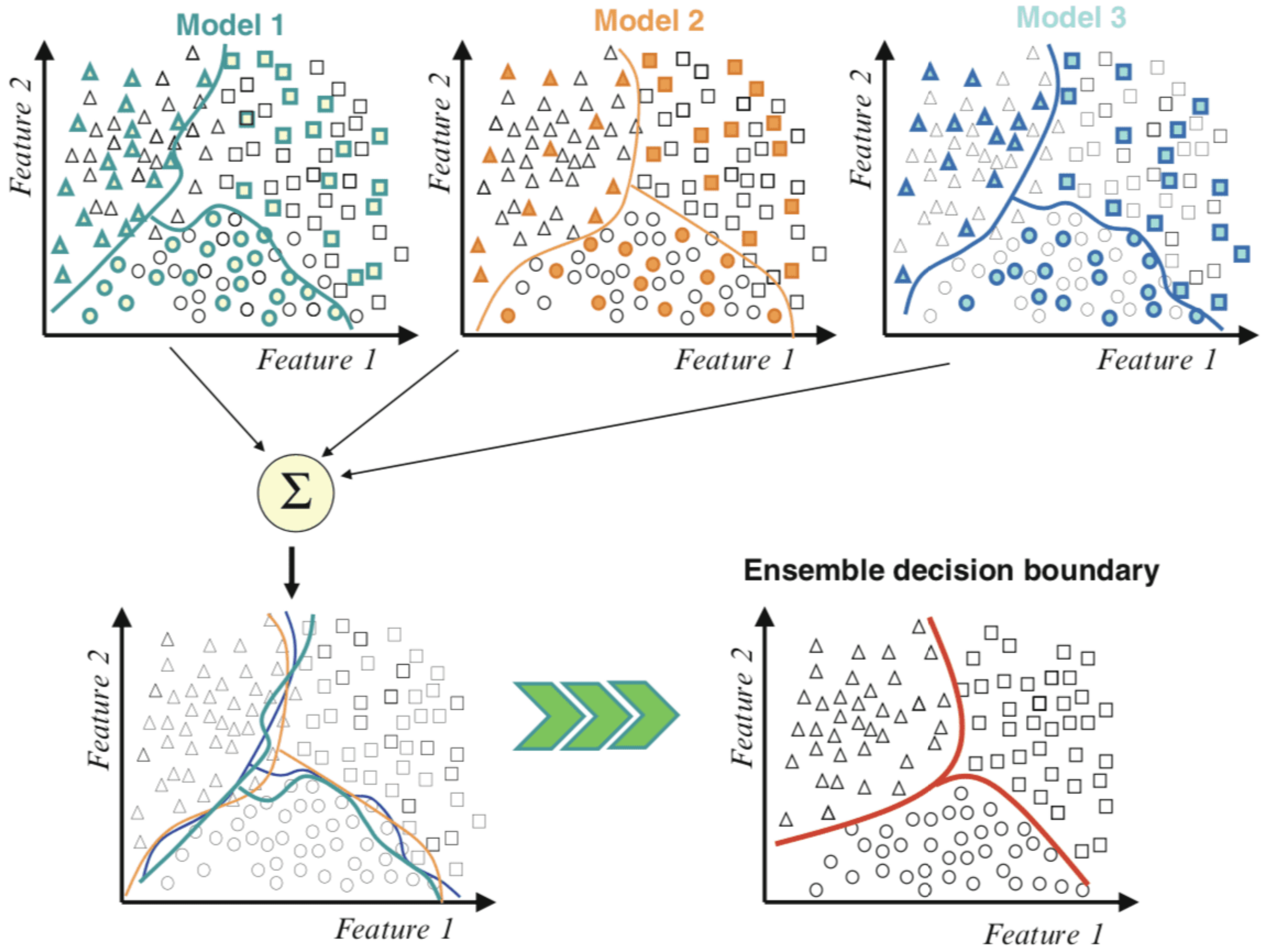
1. Avoid the state-of-the-art trap.
2. Start with the simplest models.
3. Avoid human biases in selecting models.
4. Evaluate good performance now versus good performance later.
5. Evaluate trade-offs.
6. Understand your model's assumptions.

Ensembles

Use multiple models instead of one to make predictions.

Each model in the ensemble is called a base learner.

There are three ways to create an ensemble: bagging, boosting, and stacking.



Model Development

Ensembles

Ensembles are less favored in production because they are more complex to deploy and harder to maintain.

However, they are still common for tasks where a small performance boost can lead to a huge gain.

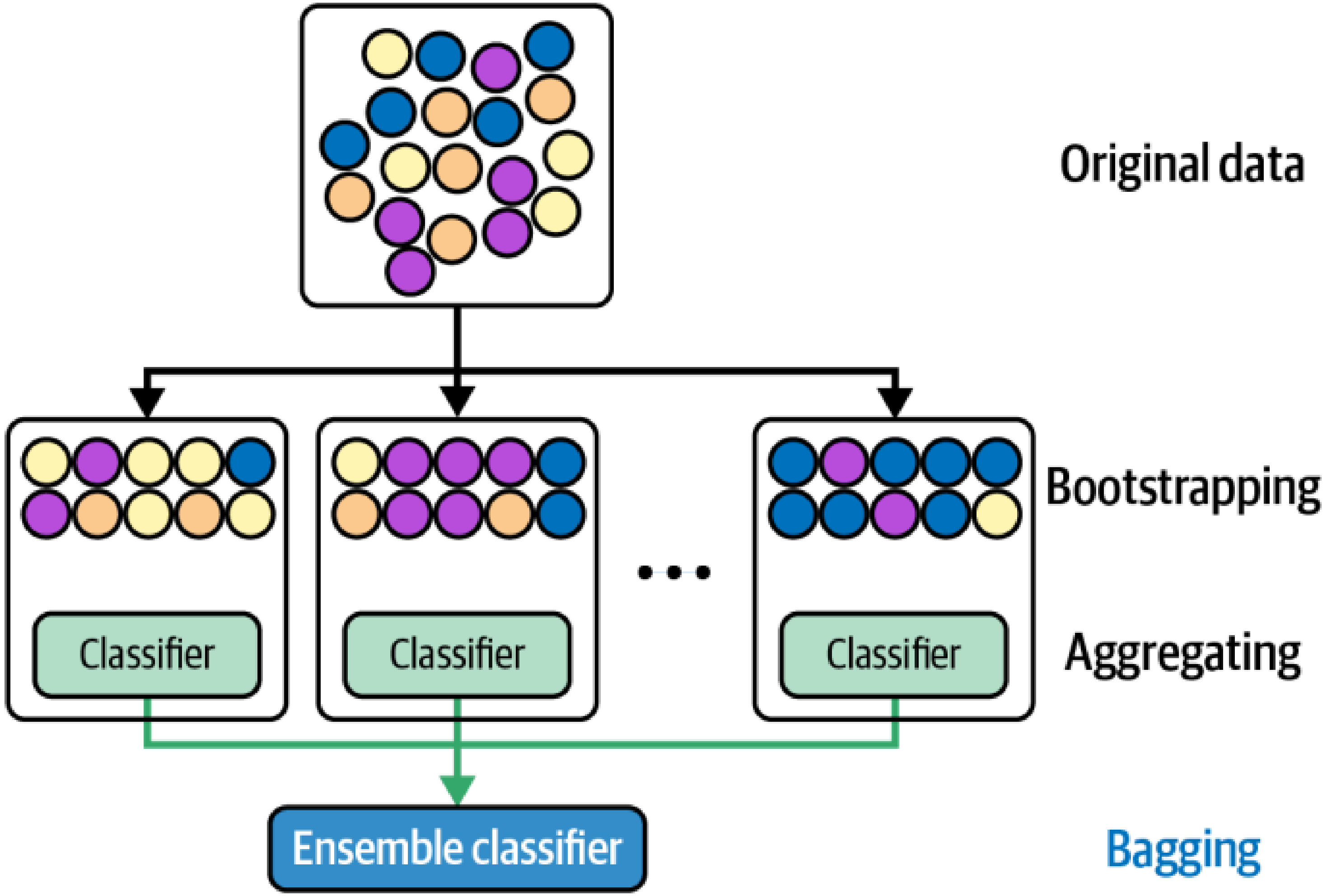
Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar & Jia et al. '18)	86.831	89.452
1 Jun 04, 2021	IE-Net (ensemble) *RICOH_SRCB_DML*	**90.939**	**93.214**
2 Feb 21, 2021	FPNet (ensemble) *Ant Service Intelligence Team*	90.871	93.183
3 May 16, 2021	IE-NetV2 (ensemble) *RICOH_SRCB_DML*	90.860	93.100
4 Apr 06, 2020	SA-Net on Albert (ensemble) *QIANXIN*	90.724	93.011
5 May 05, 2020	SA-Net-V2 (ensemble) *QIANXIN*	90.679	92.948
5 Apr 05, 2020	Retro-Reader (ensemble) *Shanghai Jiao Tong University* http://arxiv.org/abs/2001.09694	90.578	92.978

Bagging (Bootstrap Aggregating)

Reduces variance and helps to avoid overfitting.

Samples with replacement to create different datasets, called bootstraps, and train a model on each of these bootstraps.

The final prediction is decided by the majority vote or the average.



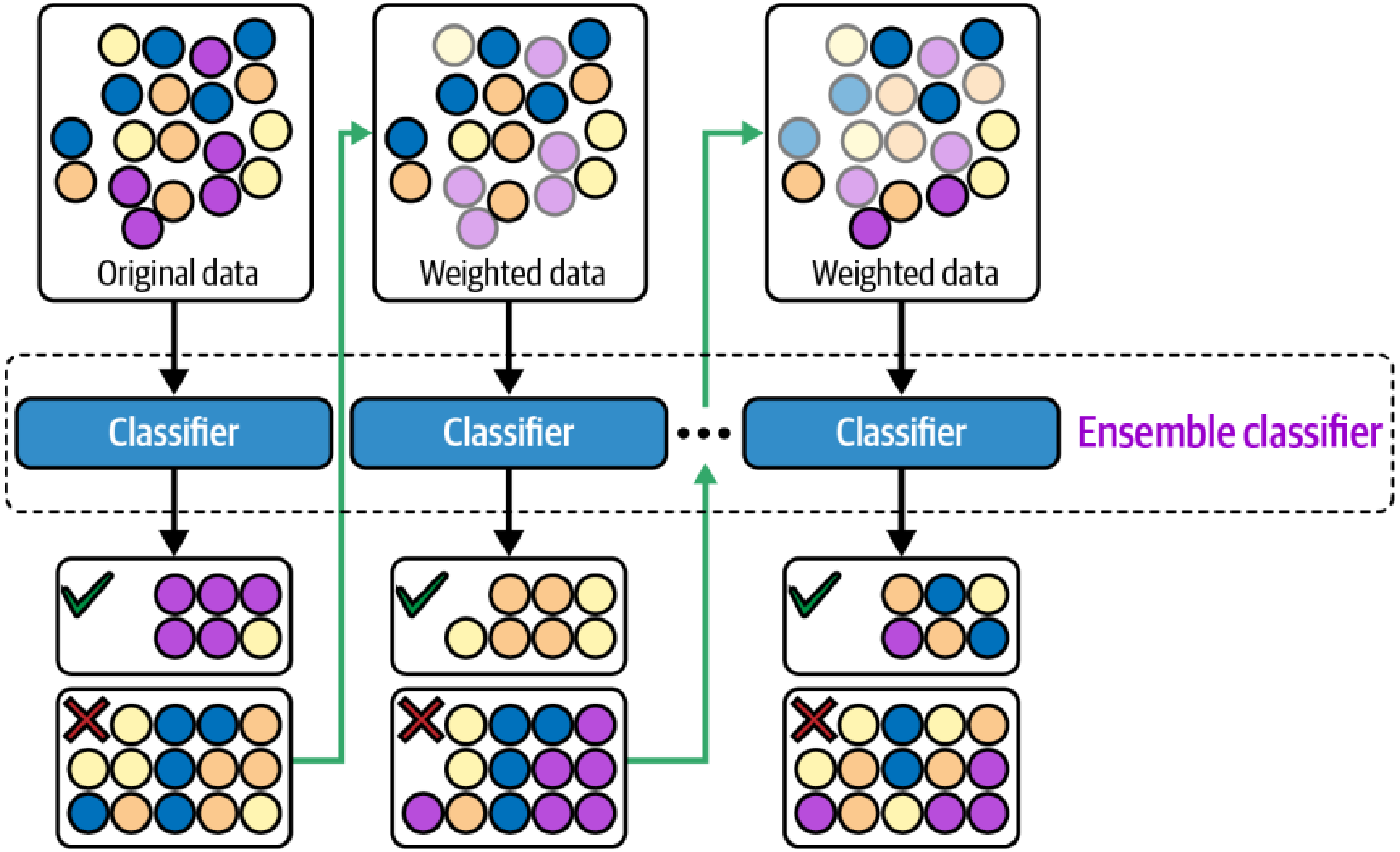
Ensembles

Boosting

Converts weak learners into strong ones.

Trains each learner on the same set of samples, with varying sample weights across iterations.

Future weak learners focus more on the examples that previous weak learners misclassified.



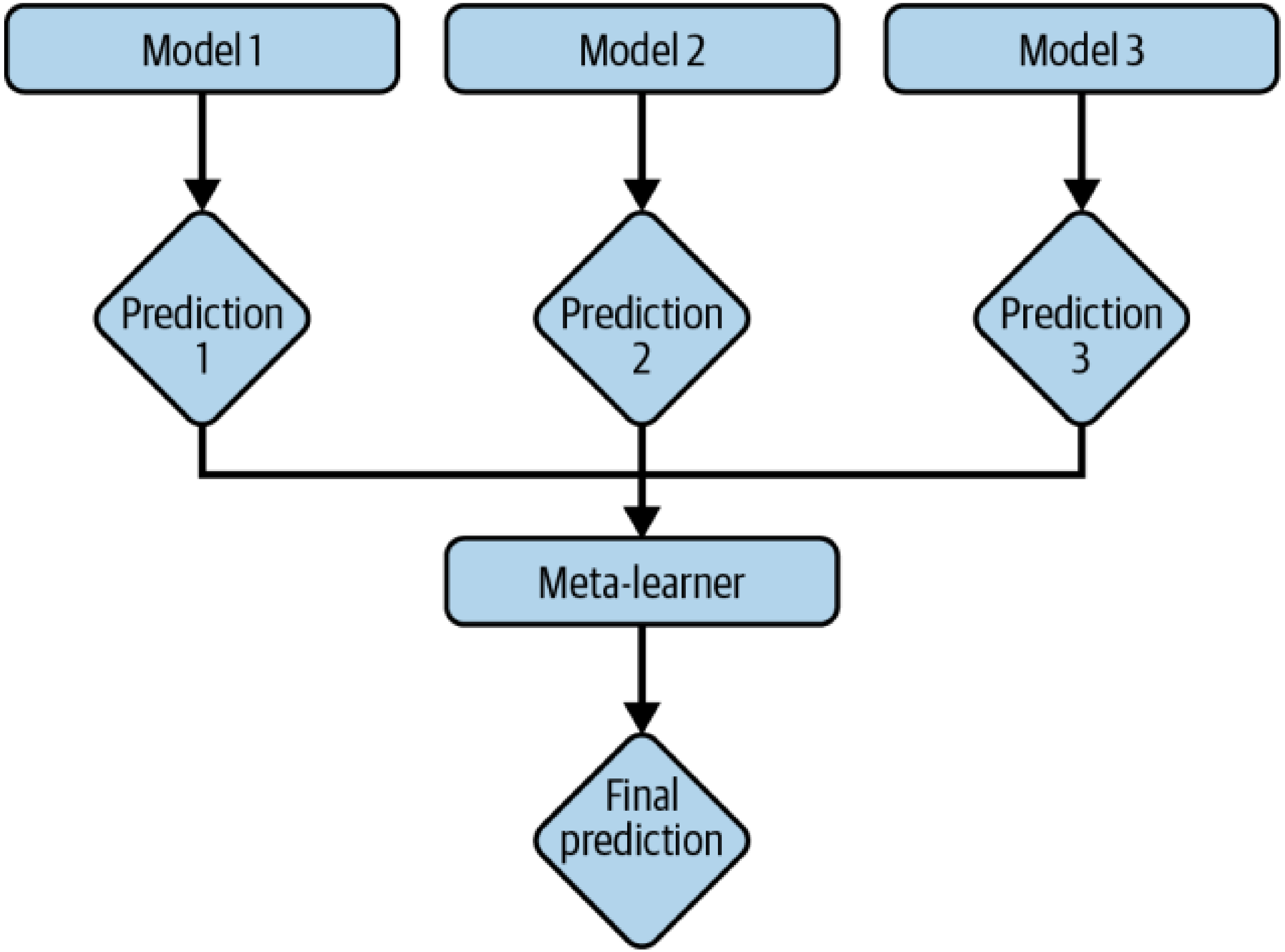
Ensembles

Stacking

Trains base learners then create a meta-learner that combines the outputs to yield final predictions.

The meta-learner can be as simple as a heuristic: the majority vote or the average vote.

Can be another model, such as regression models.



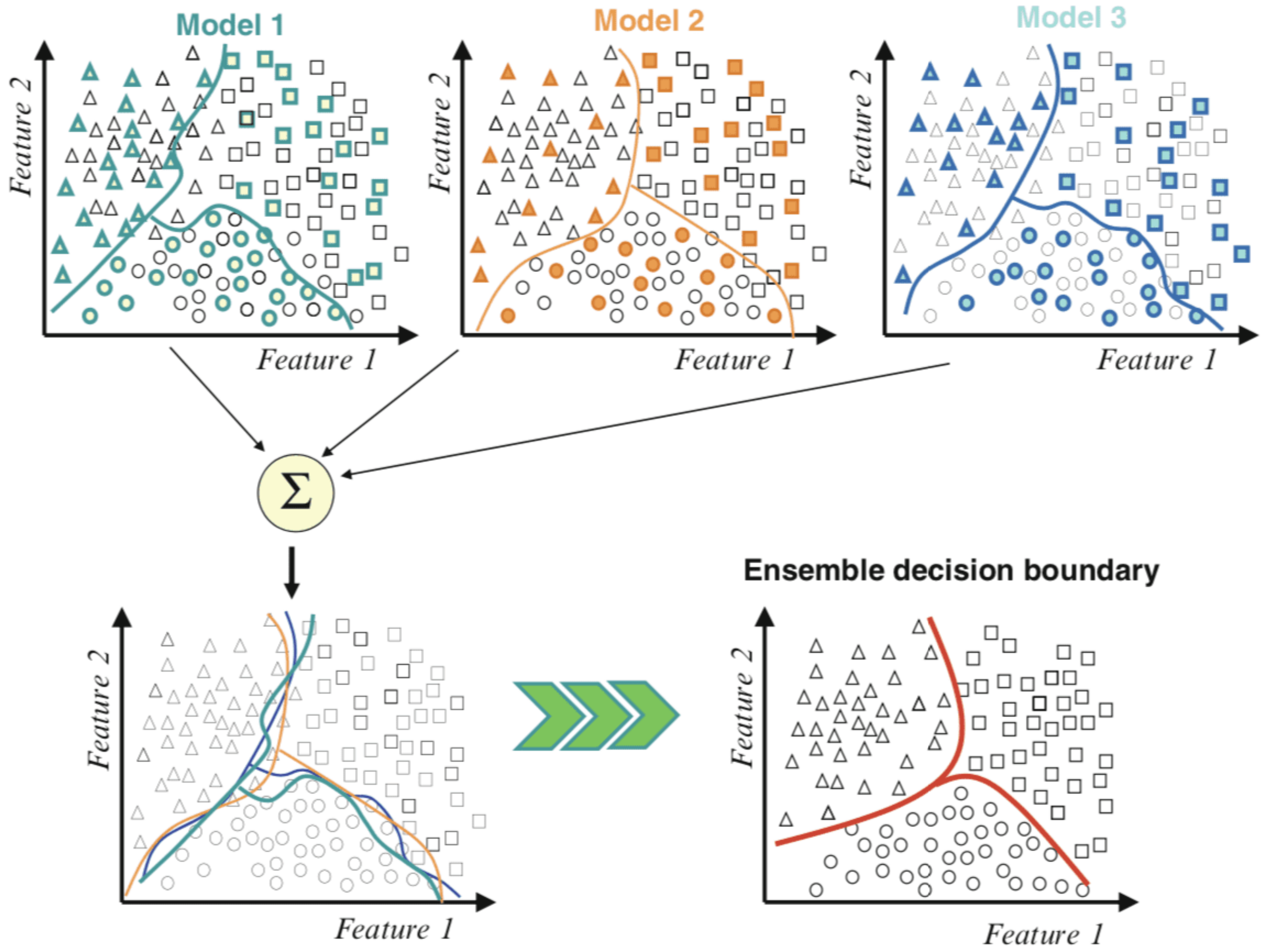
Model Development

Ensembles

Use multiple models instead of one to make predictions.

Each model in the ensemble is called a base learner.

There are three ways to create an ensemble: bagging, boosting, and stacking.

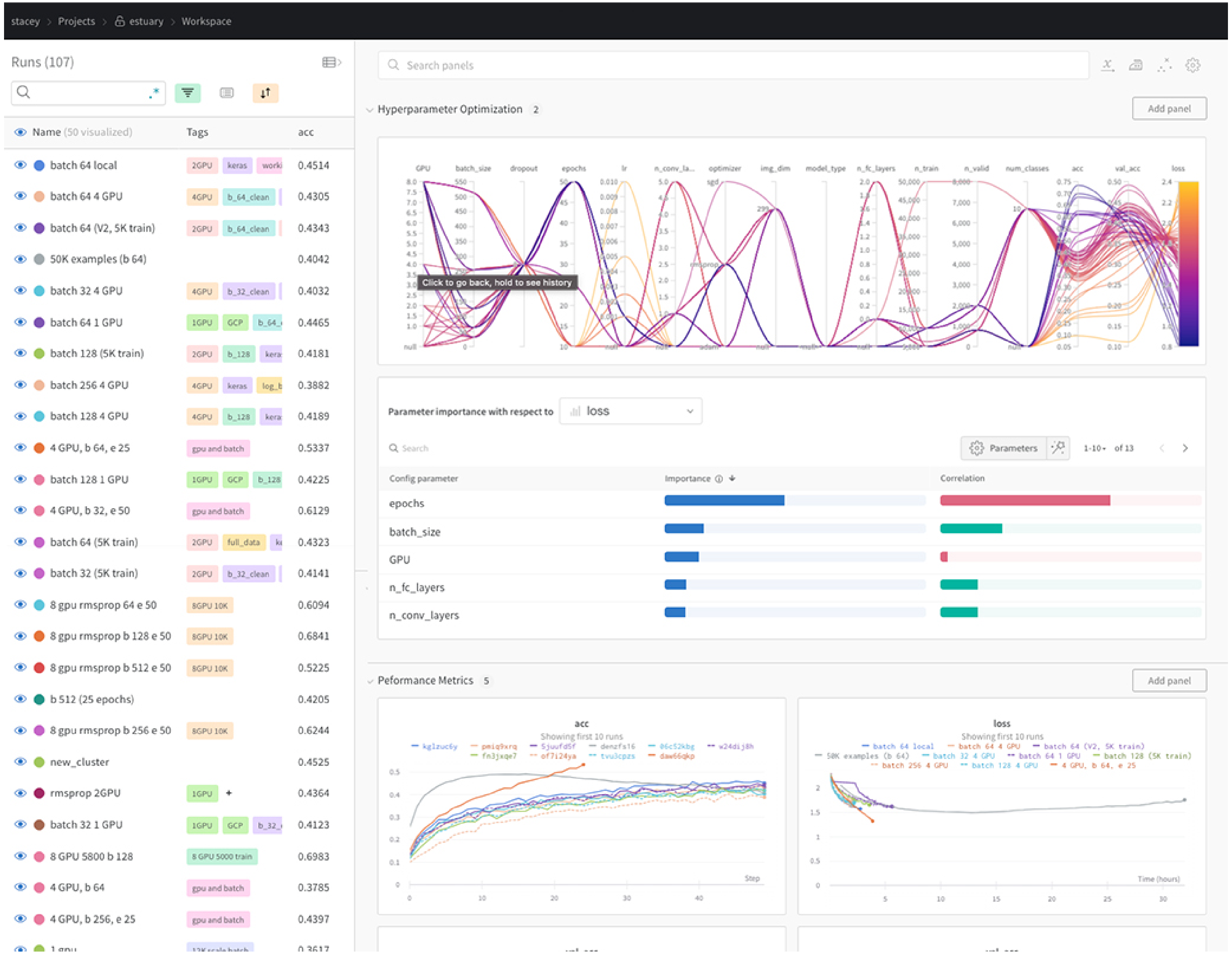


Experiment Tracking

A large part of training an ML model is babysitting.

Tracking gives you observability into the state of your model

A simple way is to make copies of all the files and log all outputs with their timestamps.



Experiment Tracking

A large part of training an ML model is babysitting.

Tracking gives you observability into the state of your model

A simple way is to make copies of all the files and log all outputs with their timestamps.

Things you might consider tracking:

- Loss curve
- Model performance metrics
- Sample, prediction, and ground truth
- Training and inference speed
- Memory and CPU utilization
- Parameters and hyperparameters

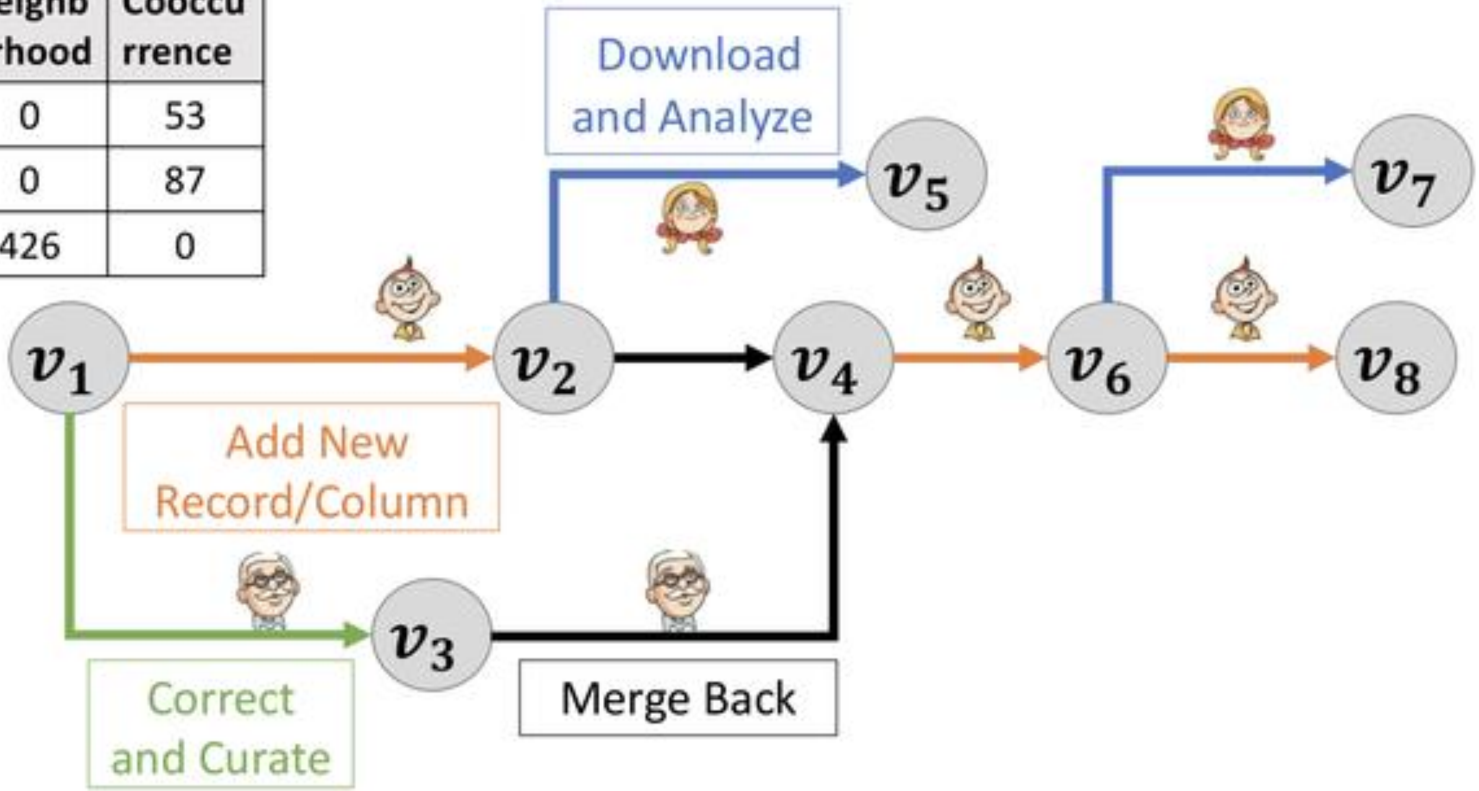
Experiment Versioning

Code versioning has more or less become a standard.

Data versioning is challenging.

Experiment tracking and versioning helps with reproducibility.

Protein2	Neighborhood	Cooccurrence
ENSP2	0	53
ENSP3	0	87
ENSP5	426	0



Causes of Failure in ML Models

1. **Theoretical constraints**

   The data it learns from doesn't conform to its assumptions.

2. **Poor implementation**

   Using off-the-shelf models makes this become less of a problem.

3. **Poor hyperparameters**

   Might render state-of-the-art models useless.

4. **Data problems**

   Data being incorrectly labeled, noisy labels, features normalized using outdated statistics.

5. **Poor choice of features**

   Too many might cause your models to overfit, Too few might lack predictive power.

Debugging Techniques

There is, unfortunately, still no scientific approach to debugging in ML.

These are some of the techniques published by experienced ML engineers and researchers.

1. **Start simple and gradually add more**

   Problem could have been caused by any of the many components in the model.

2. **Overfit a single batch**

   If it can't overfit a small amount of data, there might be something wrong.

3. **Set a random seed**

   Allows you and others to reproduce your errors and results.

The End

Summary

- Aspects to consider to decide on which model is best for you.
- Ensembles of models, a technique widely used in competitions.
- Tracking and versioning of your many experiments are important.