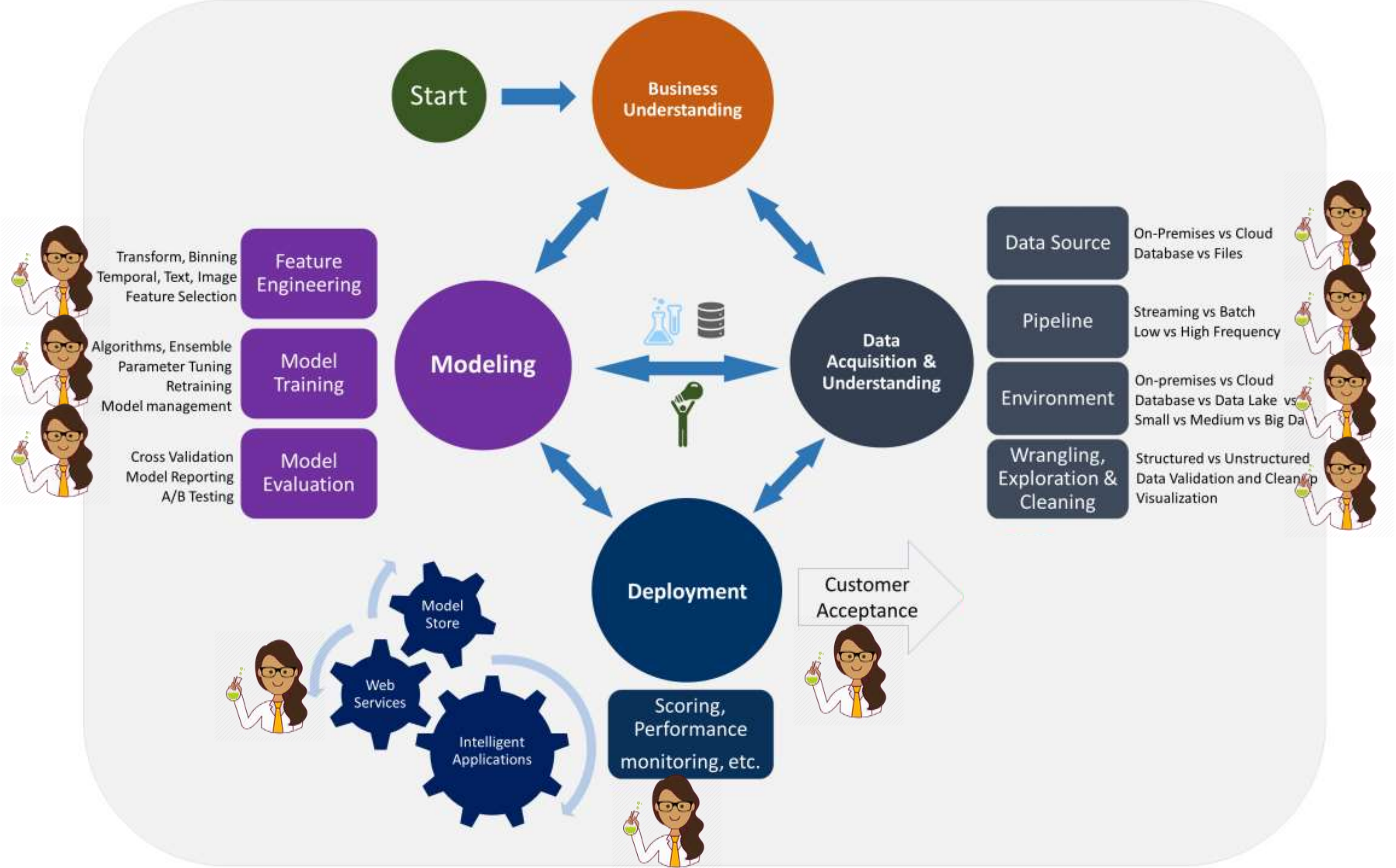
Automatic Machine Learning

Herman Wu

Sr. Software Engineer
Microsoft

Machine Learning LiveCycle



Start
Business Understanding
Data Acquisition & Understanding
Data Source
On-Premises vs Cloud
Database vs Files
Pipeline
Streaming vs Batch
Low vs High Frequency
Environment
On-premises vs Cloud
Database vs Data Lake vs
Small vs Medium vs Big Da
Wrangling, Exploration & Cleaning
Structured vs Unstructured
Data Validation and Clean p
Visualization
Modeling
Feature Engineering
Transform, Binning
Temporal, Text, Image
Feature Selection
Model Training
Algorithms, Ensemble
Parameter Tuning
Retraining
Model management
Model Evaluation
Cross Validation
Model Reporting
A/B Testing
Deployment
Customer Acceptance
Scoring, Performance monitoring, etc.
Model Store
Web Services
Intelligent Applications

https://rladiestaipei.github.io/R_DragonBall/

Day 1 (8/18) : Data Wrangling & Data Cleaning

Day 2 (8/19) : Exploratory Data Analysis

Day 3 (8/20) : Feature Engineering

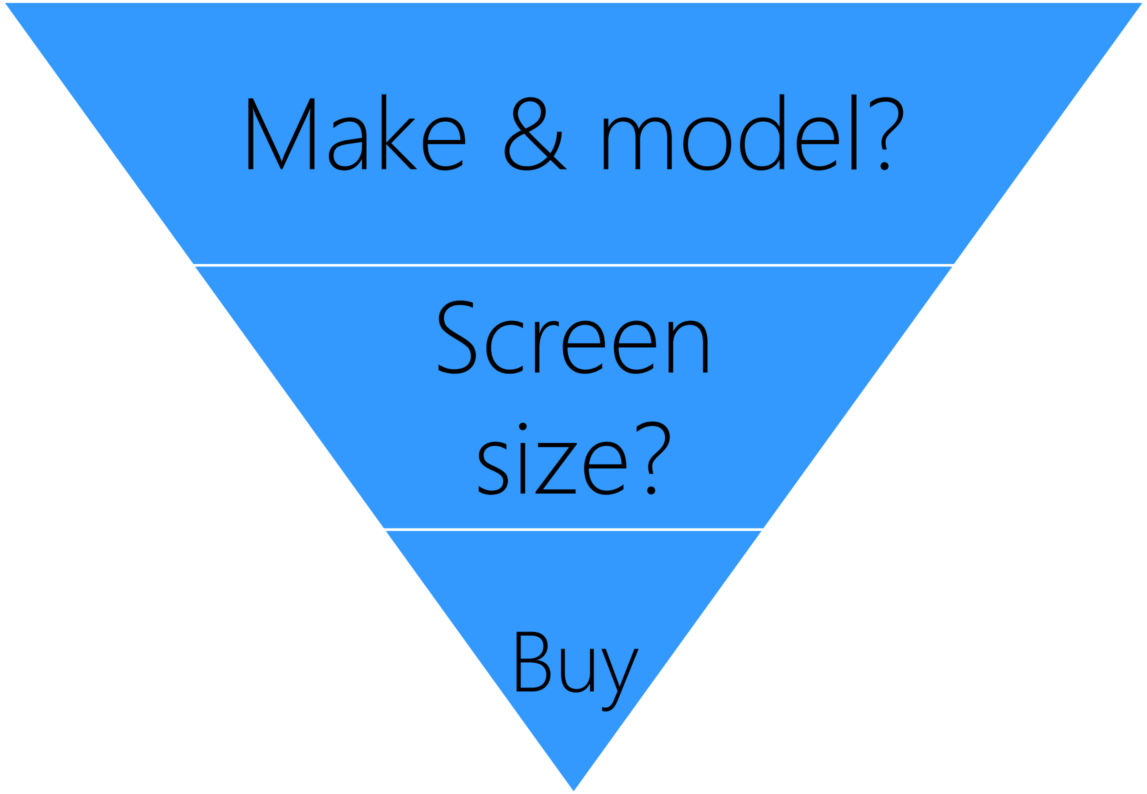
Day 4 (8/21) : Models

Day 5 (8/22) : Cross Validation & Hyperparameter Tuning

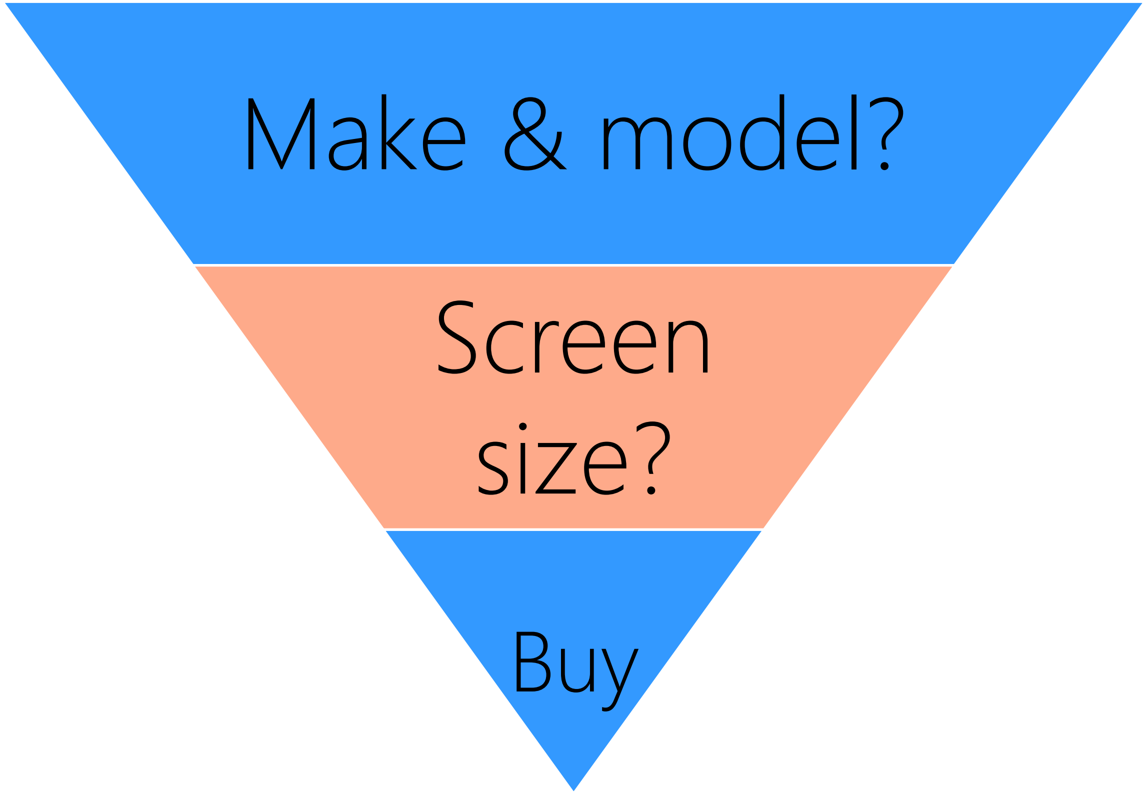
Day 6 (8/23) : Performance Evaluation

Day 7 (8/24) : Shiny

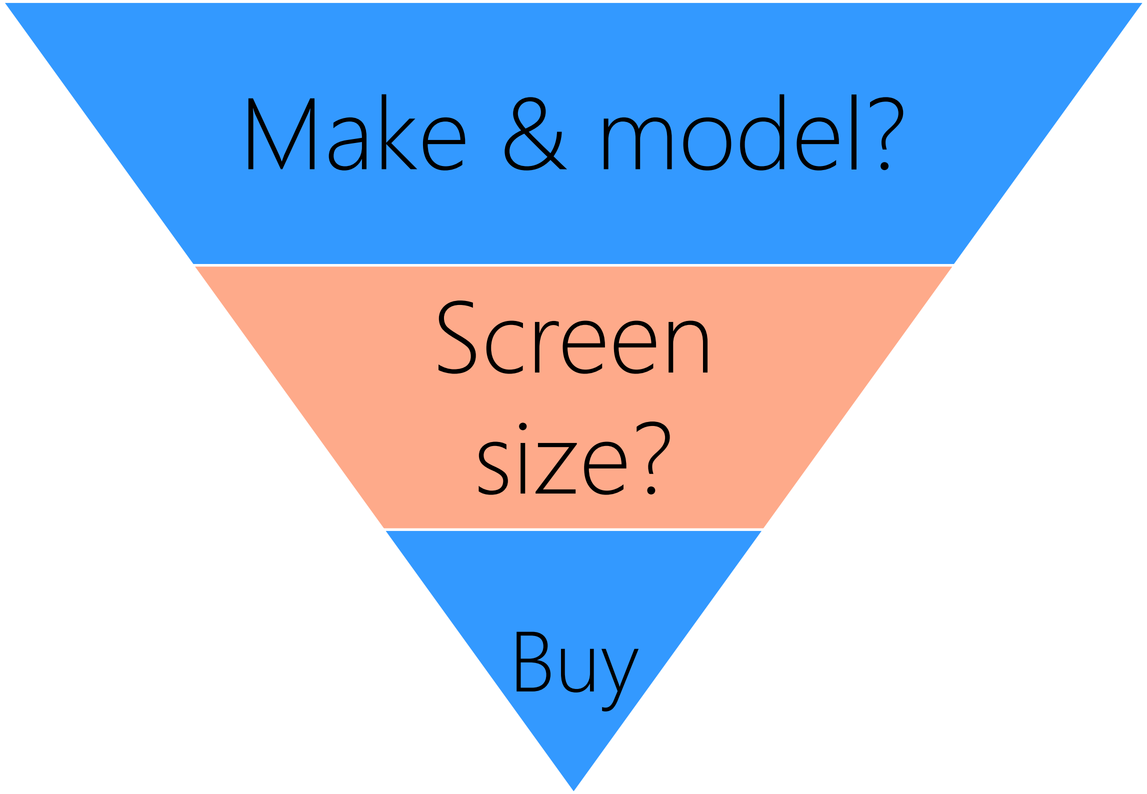
Example: Sales funnel optimization



Example: Sales funnel optimization



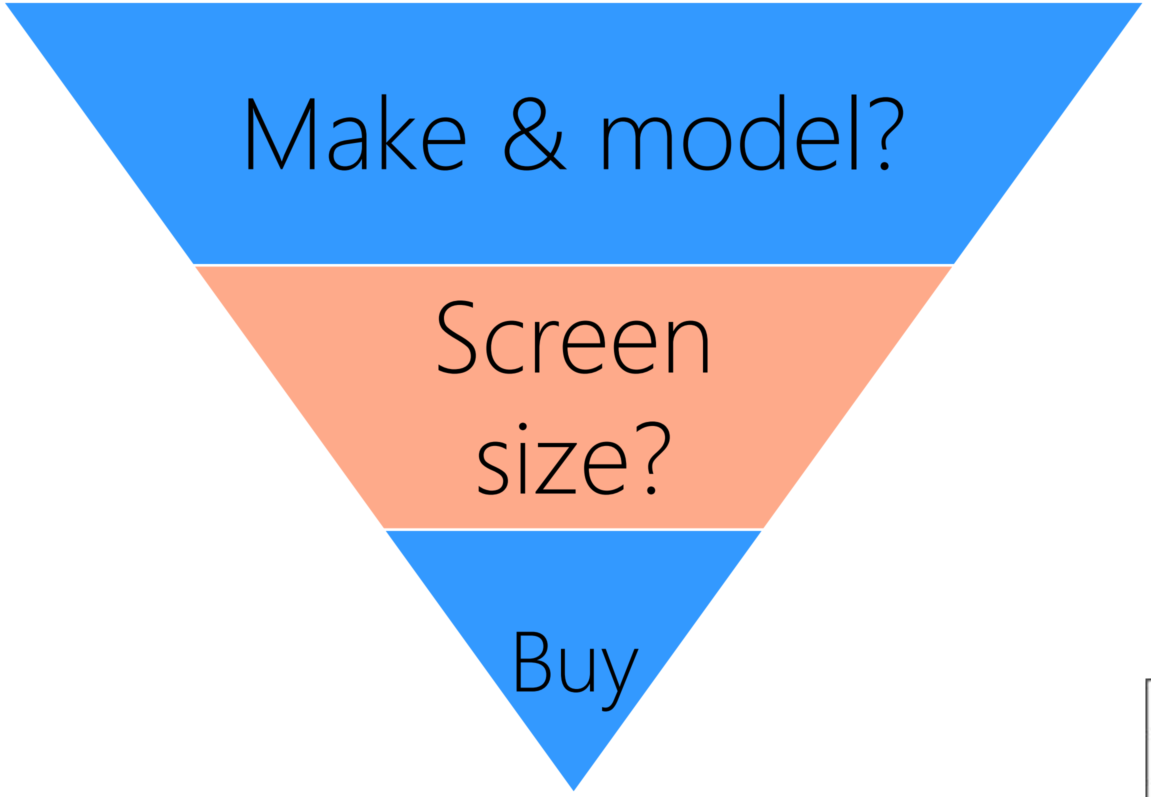
Example: Sales funnel optimization

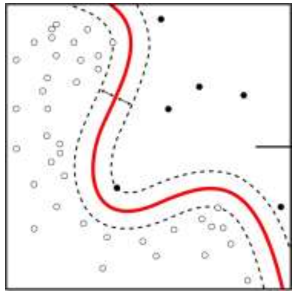


ML?

- Classification
- Regression
- Timeseries forecasting

Example: Sales funnel optimization





Class 1

Class 2

Example: Sales funnel optimization

user_id	user_device	user_os	user_age	zipcode	last_message	time	target
23433	ios	ios_11	30	92505	what ligtweight options do you have?	2018-12-15	13.5in
5223423	android	android_7	65	75240	How do increase default fntsize	2019-01-15	15in
343433	android	android_9		98004		2018-08-01	15in

Example: Sales funnel optimization

user_id	user_device	user_os	user_age	zipcode	last_message	time	target
23433	ios	ios_11	30	92505	what ligtweight options do you have?	2018-12-15	13.5in
5223423	android	android_7	65	75240	How do increase default fntsize	2019-01-15	15in
343433	android	android_9		98004		2018-08-01	15in

Numerical features

- Discretization:
  - k-means clustering
    - n_clusters (2, 3, 4,..?)
  - Equal sized bins
    - n_bins (2, 3, 4,..?)
    - Target encoding on bin-categories.
- Scaling
  - Normalization, percentile-based, …
- Outlier removal

Example: Sales funnel optimization

user_id	user_device	user_os	user_age	zipcode	last_message	time	target
23433	ios	ios_11	30	92505	what ligtweight options do you have?	2018-12-15	13.5in
5223423	android	android_7	65	75240	How do increase default fntsize	2019-01-15	15in
343433	android	android_9		98004		2018-08-01	15in

Numerical features
- Discretization:
    - k-means clustering
        - n_clusters (2, 3, 4,..?)
    - Equal sized bins
        - n_bins (2, 3, 4,..?)
        - Target encoding on bin-categories.
- Scaling
    - Normalization, percentile-based, …
- Outlier removal

Categorical features
- one hot encoding
- Target encoding
    - Cross-validation
    - Regularization params
- Categoricals for trees

Example: Sales funnel optimization

user_id	user_device	user_os	user_age	zipcode	last_message	time	target
23433	ios	ios_11	30	92505	what ligtweight options do you have?	2018-12-15	13.5in
5223423	android	android_7	65	75240	How do increase default fntsize	2019-01-15	15in
343433	android	android_9		98004		2018-08-01	15in

Numerical features
- Discretization:
    - k-means clustering
        - n_clusters (2, 3, 4,..?)
    - Equal sized bins
        - n_bins (2, 3, 4,..?)
        - Target encoding on bin-categories.
- Scaling
    - Normalization, percentile-based, ...
- Outlier removal

Categorical features
- one hot encoding
- Target encoding
    - Cross-validation
    - Regularization params
- Categoricals for trees

Text features
- Word ngrams
    - unigrams
    - bigrams
    - trigrams ...
- Character ngrams
    - unigram ...
- vocabulary size, idf, stop-words, casing
- word embedding
    - pretrained word embeddings
        - language?
        - pretrained corpus?
        - dimension?
- text similarity
    - embedding-based
    - ngram-based

Example: Sales funnel optimization

user_id	user_device	user_os	user_age	zipcode	last_message	time	target
23433	ios	ios_11	30	92505	what ligtweight options do you have?	2018-12-15	13.5in
5223423	android	android_7	65	75240	How do increase default fntsize	2019-01-15	15in
343433	android	android_9		98004		2018-08-01	15in

Numerical features

- Discretization:
  - k-means clustering
    - n_clusters (2, 3, 4,..?)
  - Equal sized bins
    - n_bins (2, 3, 4,..?)
    - Target encoding on bin-categories.
- Scaling
  - Normalization, percentile-based, …
- Outlier removal

Categorical features

- one hot encoding
- Target encoding
  - Cross-validation
  - Regularization params
- Categoricals for trees

Text features

- Word ngrams
  - unigrams
  - bigrams
  - trigrams …
- Character ngrams
  - unigram …
- vocabulary size, idf, stop-words, casing
- word embedding
  - pretrained word embeddings
    - language?
    - pretrained corpus?
    - dimension?
- text similarity
  - embedding-based
  - ngram-based

Time series forecasting features:

- lagged features
- Frequency detection

Timestamp features

- Day of week
- Day of month
- Day of year
- Month
- Hour
- Minute
- Holiday
- Quarter

Without automated ML: hard, combinatorial explosion

user_id	user_device	user_os	user_age	zipcode	last_message	time	target
23433	ios	ios_11	30	92505	what ligtweight options do you have?	2018-12-15	13.5in
5223423	android	android_7	65	75240	How do increase default fntsize	2019-01-15	15in
343433	android	android_9		98004		2018-08-01	15in

Numerical features
- Discretization:
  - k-means clustering
    - n_clusters (2, 3, 4,..?)
  - Equal sized bins
    - n_bins (2, 3, 4,..?)
    - Target encoding on bin-categories.
- Scaling
  - Normalization, percentile-based, …
- Outlier removal

Categorical features
- one hot encoding
- Target encoding
  - Cross-validation
  - Regularization params
- Categoricals for trees

Text features
- Word ngrams
  - unigrams
  - bigrams
  - trigrams …
- Character ngrams
  - unigram …
- vocabulary size, idf, stop-words, casing
- word embedding
  - pretrained word embeddings
    - language?
    - pretrained corpus?
    - dimension?
- text similarity
  - embedding-based
  - ngram-based

Time series forecasting features:
- y- lagged features
- Frequency detection

Timestamp features
- Day of week
- Day of month
- Day of year
- Month
- Hour
- Minute
- Holiday
- Quarter

Without automated ML: hard, combinatorial explosion
With automated ML: easy, tailored to your dataset

user_id	user_device	user_os	user_age	zipcode	last_message	time	target
23433	ios	ios_11	30	92505	what ligtweight options do you have?	2018-12-15	13.5in
5223423	android	android_7	65	75240	How do increase default fntsize	2019-01-15	15in
343433	android	android_9		98004		2018-08-01	15in

Numerical features
- Discretization:
  - k-means clustering
    - n_clusters (2, 3, 4,..?)
  - Equal sized bins
    - n_bins (2, 3, 4,..?)
    - Target encoding on bin-categories.
- Scaling
  - Normalization, percentile-based, …
- Outlier removal

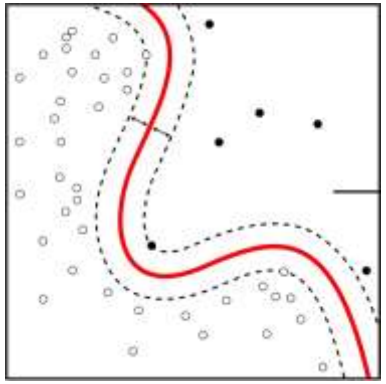
Categorical features
- one hot encoding
- Target encoding
  - Cross-validation
  - Regularization params
- Categoricals for trees

Text features
- Word ngrams
  - unigrams
  - bigrams
  - trigrams …
- Character ngrams
  - unigram …
- vocabulary size, idf, stop-words, casing
- word embedding
  - pretrained word embeddings
    - language?
    - pretrained corpus?
    - dimension?
- text similarity
  - embedding-based
  - ngram-based

Time series forecasting features:
- y- lagged features
- Frequency detection

Timestamp features
- Day of week
- Day of month
- Day of year
- Month
- Hour
- Minute
- Holiday
- Quarter

What Machine Learning algorithm will best separate 13.5in users from 15in users?



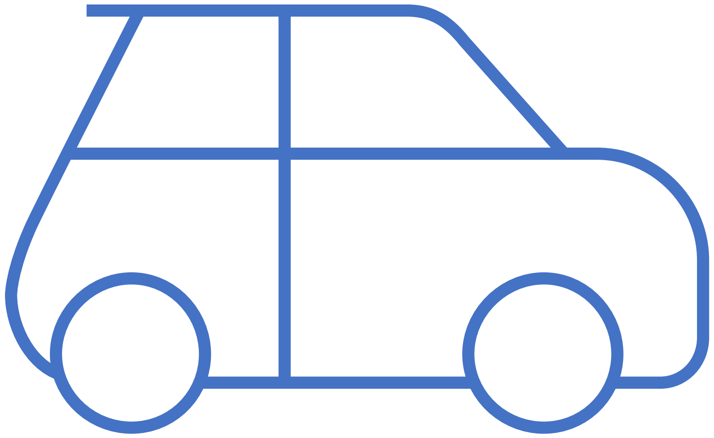
xgboost alone: ~ 10^10 possible parameter configurations

Compute cost > 10^5 years


DEMO
H2O AutoML


DEMO
AutoSKLearn

Machine Learning Problem Example



How much is this car worth?

Model Creation Is Typically Time-Consuming

Which features?

Mileage

Condition

Car brand

Year of make

Regulations

…

Which algorithm?

Gradient Boosted

Nearest Neighbors

SVM

Bayesian Regression

LGBM

…

Which parameters?

Criterion
Parameter 1

Loss
Parameter 2

Min Samples Split
Parameter 3

Min Samples Leaf
Parameter 4

Others

30%

Model

Model Creation Is Typically Time-Consuming

Which features?

- **Mileage**
- **Condition**
- **Car brand**
- **Year of make**
- Regulations
- …

Which algorithm?

- **Gradient Boosted**
- **Nearest Neighbors**
- SVM
- Bayesian Regression
- LGBM
- …

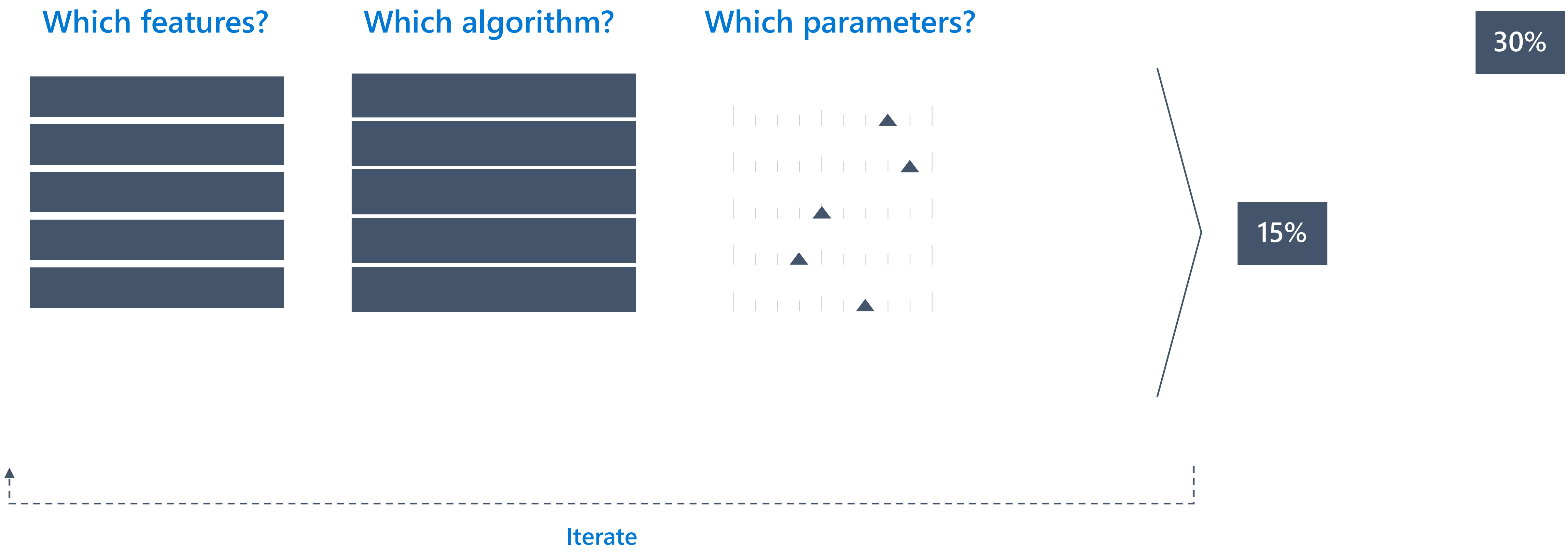
Which parameters?

- Criterion
- N Neighbors
- Loss
- Weights
- Metric
- Min Samples Split
- P
- Min Samples Leaf
- Others

30%
Model

Iterate

Model Creation Is Typically Time-Consuming



Automated ML Accelerates Model Development

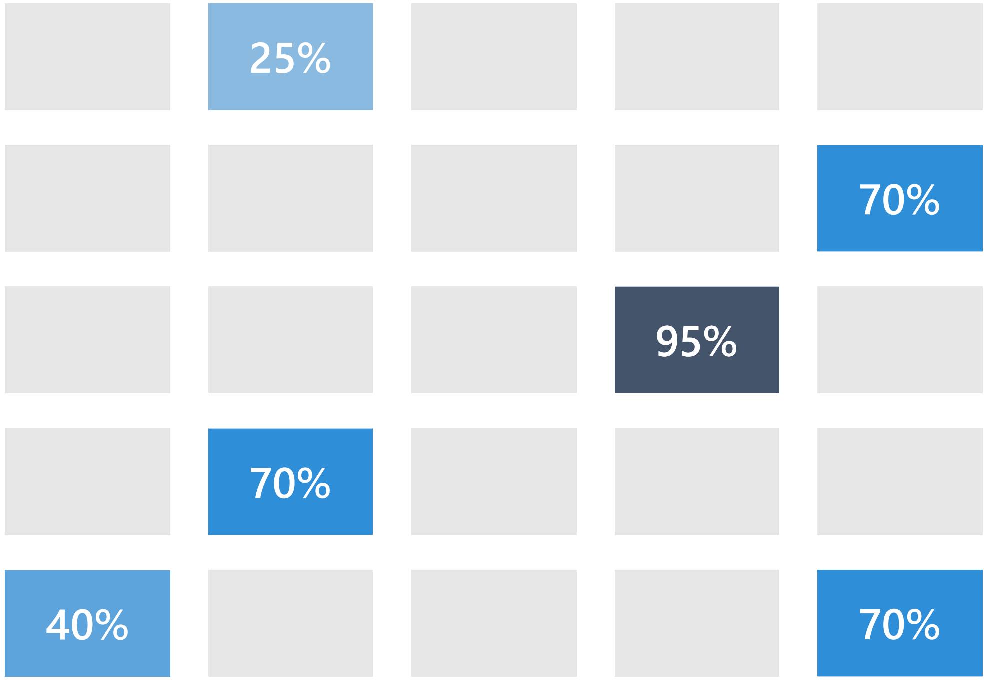
Input

Enter data

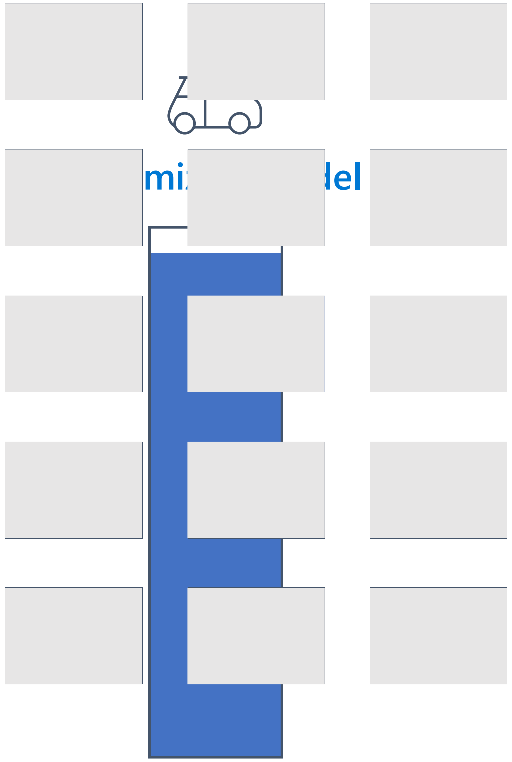
Define goals

Apply constraints

Intelligently test multiple models in parallel



Output



Automated ML

1. **Data Preprocessing**

   Automated ML currently supports automated data cleaning

2. **Feature Engineering**

   Most time-consuming part when done manually can now be done within minutes.

3. **Algorithm Selection**

   Testing many different algorithms at once.

4. **Hyper-parameter Tuning**

   Hyperparameter tuning what to include what to leave out

5. **Model Recommendation**

   Having an overview of the best performing models based on accuracy & speed.

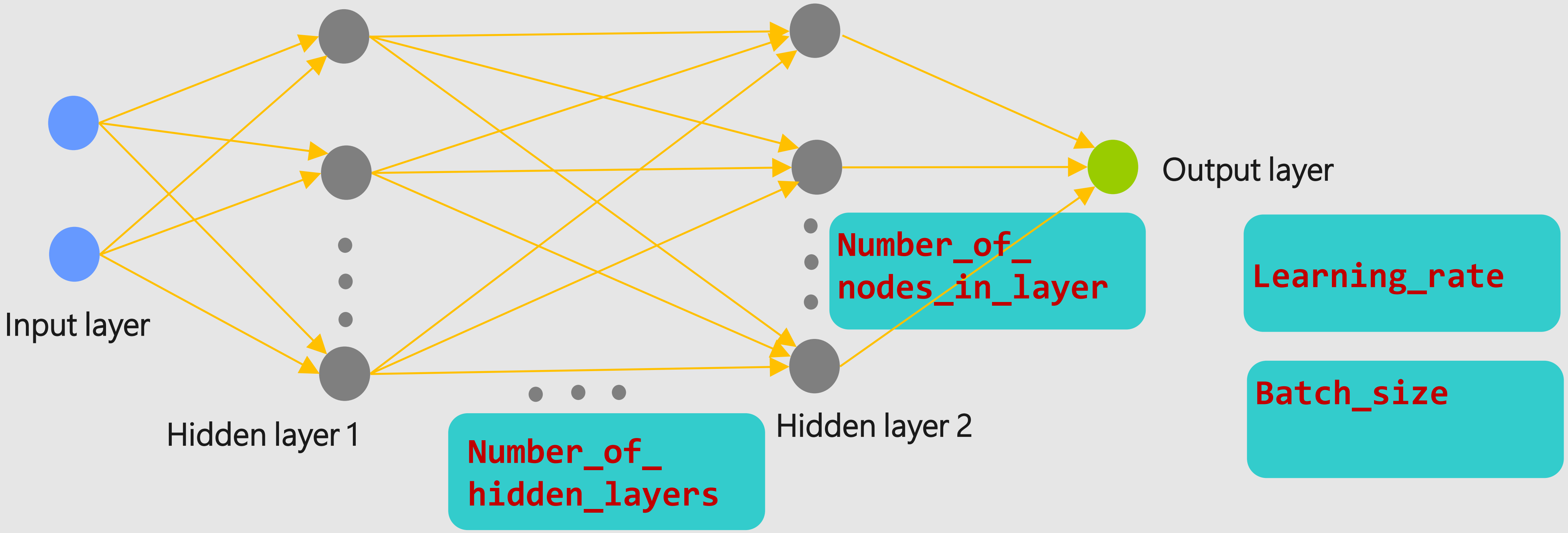
6. **Interpretability & Explaining**

   Being able to explain what created an outcome and what features had the most significant impact

What are hyperparameters?

- Adjustable parameters that govern model training
- Chosen prior to training, stay constant during training

E.g.



Common Approach

- Grid / random search
  - Run configurations with equal chance and predefined length
  - Unable to terminate poor configurations early → wasted resources
  - Unable to identify good configurations early → ineffective exhaustive search
- Human expert
  - Rely on human expert to iteratively experiment and refine hyperparameters
  - Not scalable to help many models
  - Not scalable to help models with many hyperparameters

Hyperparameter exploration

- Search across various hyperparameter configurations
- Find the configuration that results in best performance

Challenges

- Huge search space to explore
- Sparsity of good configurations
- Expensive evaluation
- Limited time and resources

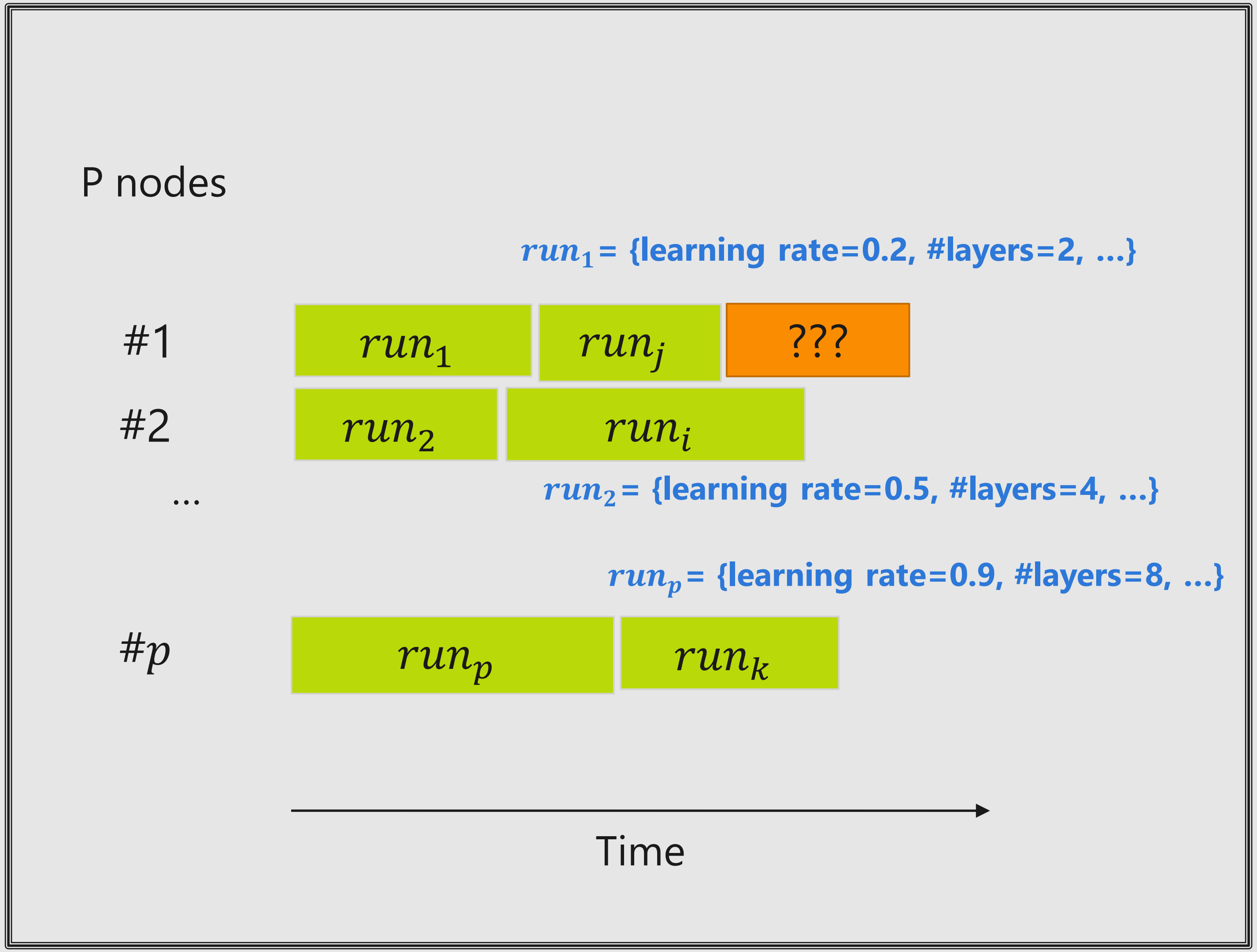
How it works

Launch multiple parallel training runs

(A) Generate new runs
- Which parameter configuration to explore?

(B) Manage resource usage of active runs
- How long to execute a run?

Hyperparameter Tuning runs in Azure ML



Generate new runs

```
{
    “learning_rate”: uniform(0, 1),
    “num_layers”: choice(2, 4, 8)
    …
}
```

Define hyperparameter search space

Sampling algorithm

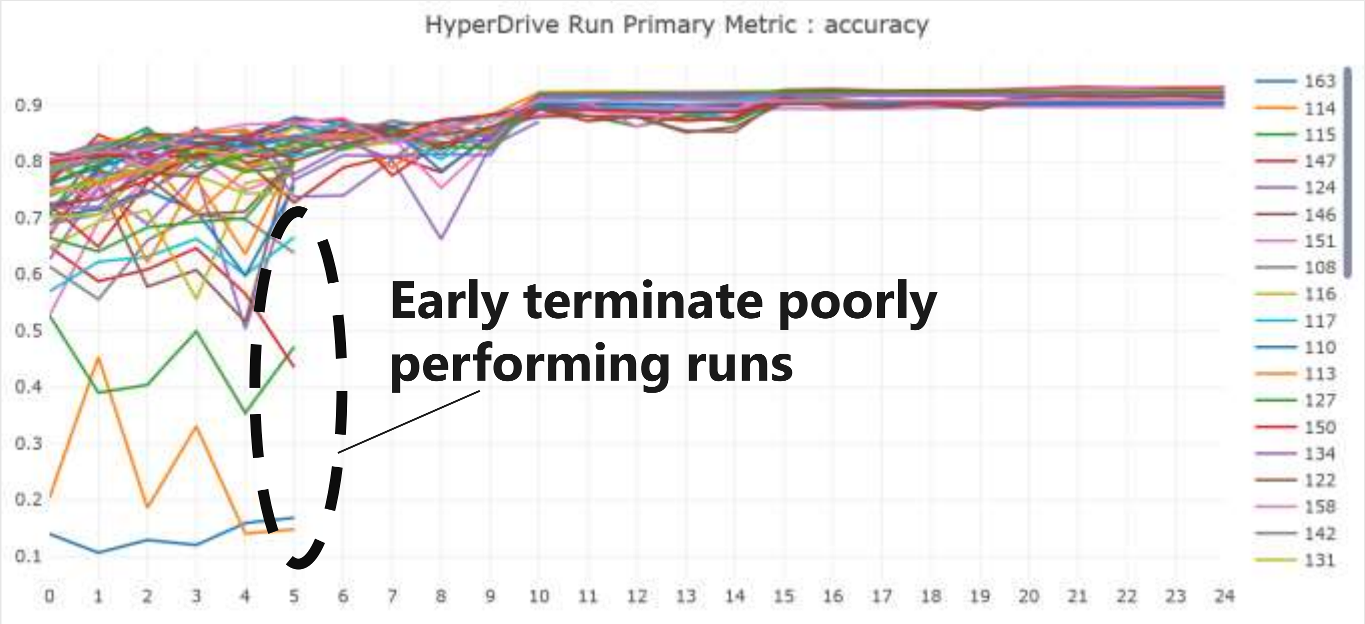
```
Config1= {“learning_rate”: 0.2,
“num_layers”: 2, …}

Config2= {“learning_rate”: 0.5,
“num_layers”: 4, …}

Config3= {“learning_rate”: 0.9,
“num_layers”: 8, …}

…
```

Supported sampling algorithms –

- Grid Sampling
- Random Sampling
- Bayesian Optimization

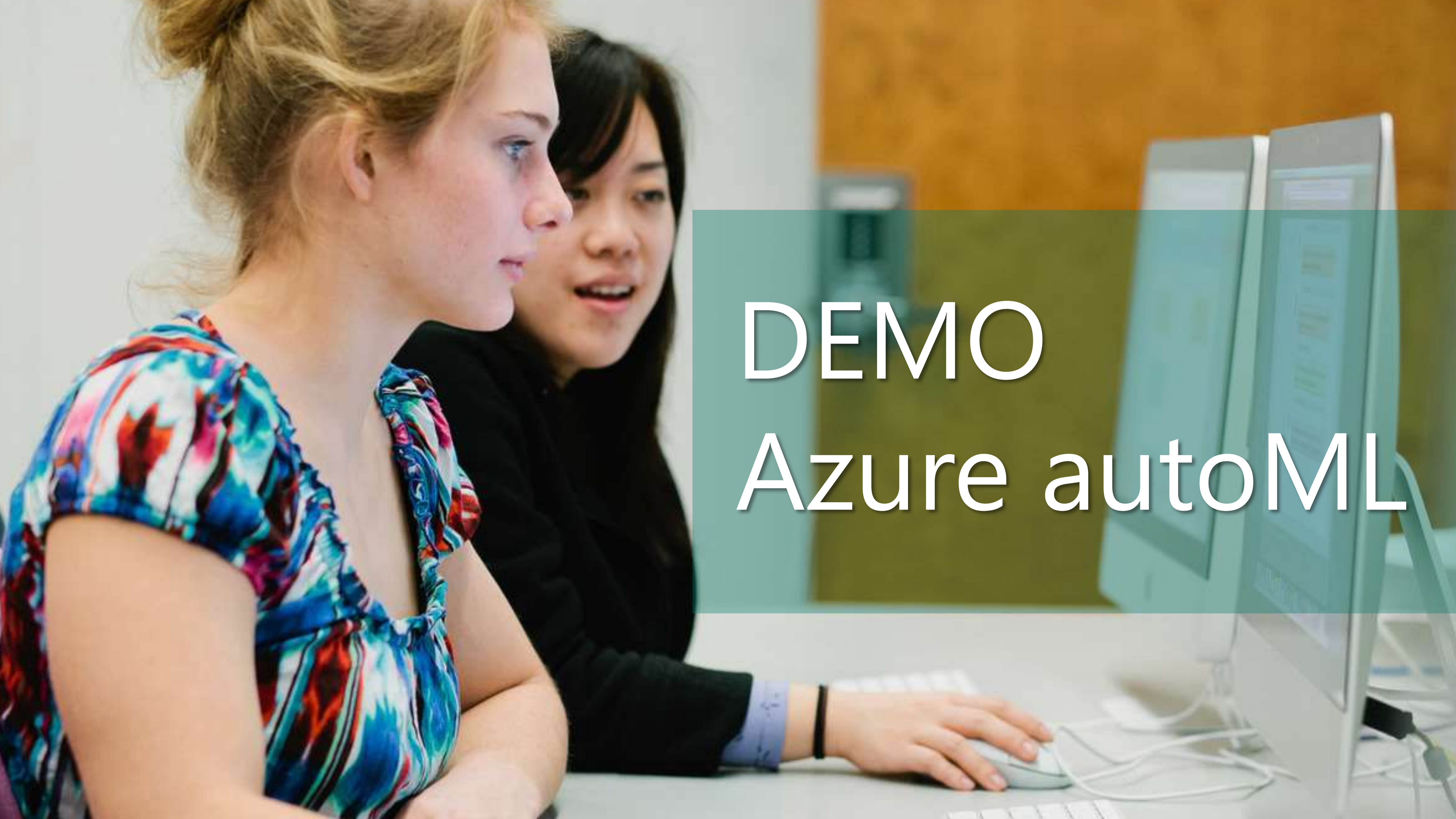
Manage Active Jobs

- Evaluate training runs for specified primary metric
- Early terminate poor performing training runs
- Use resources to explore new configurations



Supported early termination algorithms –

- Bandit policy
- Median Stopping policy
- Truncation Selection policy


DEMO
Azure autoML

Tool	Platform	Input data sources		Data pre-processing	Data types detected					Feature engineering				ML Tasks		Model selection and Hyperparameter optimization					Quick start / early stop			Model evaluation / Result analysis/ Visualization		
		Spreadsheet datasets	Image, text		Numerical	Categorical	Datetime	Time-series	Other (Hierarchical types) (7*)	Datetime, categorical processing	Imbalance, missing values	Feature selection, reduction	Advanced feature extraction (8*)	Supervised learning (9*)	Unsupervised learning (10*)	Ensemble	Genetic algorithm	Random search	Bayesian search	Neural architecture search	Quick finding of starting model	Allow maximum limit search time	Restrict time consuming combination of components	Model dashboard	Feature importance	Model explainability and interpretation, and reason code (11*)
TransmogrifAI	Apache Spark	Y	N	Y(*)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	N	Y	Y	N	N			Y	Y	
H2O-AutoML	H2O clusters	Y	N	Y	Y	Y	Y	Y	N	Y	Y	Y	N	Y	N	Y	N	Y	N	N	N	Y	Y	Y	Y	Y
Darwin (+)	GCP	Y	N	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	N	N	Y	Y	Y	N	Y	Y	Y
DataRobot (+)	Datarobot & AWS	Y	Y	Y	Y	Y	N	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	Y		Y	Y	Y
Google AutoML (+)	Google Cloud	N	Y	Y						N	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Auto-sklearn		Y	N	N	N	N	N	N	N	Y(2*)	Y	Y	Y	Y	N	Y	N	Y	Y	N	Y	Y	Y	Y	Y	Y
MLjar (+)	MLJAR Cloud	Y(3*)	N	Y	Y	Y	N	N	N	Y	Y(4*)	N	N	Y(5*)	N	Y	N	Y	N	N	N	N	N	Y	Y	N
Auto_ml		Y	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	N	Y	N	Y	Y	N	N	N	N	Y	Y	Y
TPOT		Y	N	N	N	N	N	N	N	N	Y	N	Y	Y	N	Y	Y	N	N	N	N	Y	N	Y	Y	N
Auto-keras		Y	Y	N	N	N	N	N	N	N	Y	Y	N	Y	N	N	N	Y	Y	Y	Y	Y	N	Y	N	Y
Ludwig		Y	Y	Y(*)	Y	Y	N	Y	Y	N	Y	Y	Y	Y	N	Y	N	Y	Y	Y	Y	N	N	Y	Y	N
Auto-Weka		Y	N	N	Y	Y	N	N	N	N	Y	Y	N	Y	N	Y	N	Y	Y	N	N	Y	Y	Y	N	N
Azure ML (+)	Azure	Y	Y	Y(6*)	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	N	Y	N	Y	Y	N		Y	Y	Y	Y	
Sagemaker (+)	AWS	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	N		Y	N	Y	Y	Y
H2O-Driverless AI (+)	H2O clusters	Y(3*)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	Y	Y	Y	Y

Neural Architecture Search: State-of-the-art Overview

Model	Depth	Parameters	Error rate (%)
Network in Network (Lin et al., 2013)	-	-	8.81
All-CNN (Springenberg et al., 2014)	-	-	7.25
Deeply Supervised Net (Lee et al., 2015)	-	-	7.97
Highway Network (Srivastava et al., 2015)	-	-	7.72
Scalable Bayesian Optimization (Snoek et al., 2015)	-	-	6.37
FractalNet (Larsson et al., 2016)	21	38.6M	5.22
with Dropout/Drop-path	21	38.6M	4.60
ResNet (He et al., 2016a)	110	1.7M	6.61
ResNet (reported by Huang et al. (2016c))	110	1.7M	6.41
ResNet with Stochastic Depth (Huang et al., 2016c)	110	1.7M	5.23
	1202	10.2M	4.91
Wide ResNet (Zagoruyko & Komodakis, 2016)	16	11.0M	4.81
	28	36.5M	4.17
ResNet (pre-activation) (He et al., 2016b)	164	1.7M	5.46
	1001	10.2M	4.62
DenseNet $(L = 40, k = 12)$ Huang et al. (2016a)	40	1.0M	5.24
DenseNet$(L = 100, k = 12)$ Huang et al. (2016a)	100	7.0M	4.10
DenseNet $(L = 100, k = 24)$ Huang et al. (2016a)	100	27.2M	3.74
DenseNet-BC $(L = 100, k = 40)$ Huang et al. (2016b)	190	25.6M	3.46
Neural Architecture Search v1 no stride or pooling	15	4.2M	5.50
Neural Architecture Search v2 predicting strides	20	2.5M	6.01
Neural Architecture Search v3 max pooling	39	7.1M	4.47
Neural Architecture Search v3 max pooling + more filters	39	37.4M	3.65

Table 1: Performance of Neural Architecture Search and other state-of-the-art models on CIFAR-10.

Efficient Neural Architecture Search via Parameter Sharing (Pham et al, 2018)

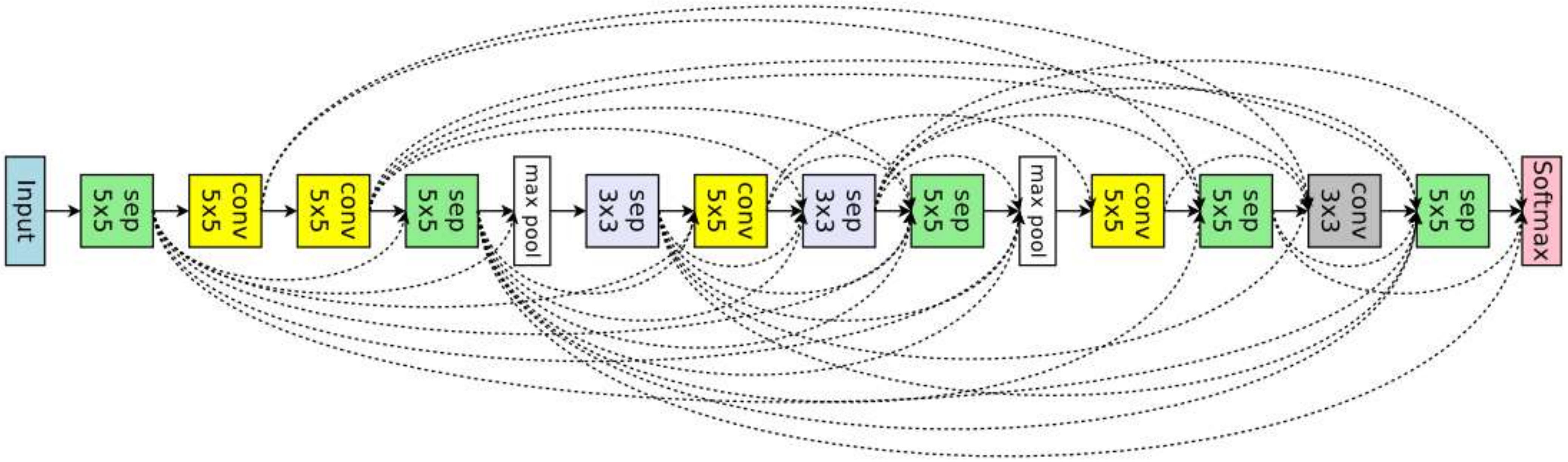


Figure 7. ENAS's discovered network from the macro search space for image classification.

Progressive Neural Architecture Search (Liu et al. 2018)

DARTS: Differentiable Architecture Search (Liu et al. 2018)


DEMO
Auto Keras

NNI (Neural Network Intelligence) toolkit

	Frameworks & Libraries	Tuning Algorithms	Training Services
Built-in	• Supported Frameworks ◦ PyTorch ◦ Keras ◦ TensorFlow ◦ MXNet ◦ Caffe2 More... • Supported Libraries ◦ Scikit-learn ◦ XGBoost ◦ LightGBM More... • Examples ◦ MNIST-pytorch ◦ MNIST-tensorflow ◦ MNIST-keras ◦ Auto-gbdt ◦ Cifar10-pytorch ◦ Scikit-learn More...	Tuner • General Tuner ◦ Random Search ◦ Naïve Evolution • Tuner for **HPO** ◦ TPE ◦ Anneal ◦ SMAC ◦ Batch ◦ Grid Search ◦ Hyperband ◦ Metis Tuner ◦ BOHB ◦ GP Tuner • Tuner for **NAS** ◦ Network Morphism ◦ ENAS Assessor ◦ Median Stop ◦ Curve Fitting	• Local Machine • Remote Servers • Kubernetes based services ◦ OpenPAI ◦ Kubeflow ◦ FrameworkController on K8S (AKS etc.)
References	• Python API • NNI Annotation • Supported OS	• CustomizeTuner • CustomizeAssessor	• Support TrainingService • Implement TrainingService

Neural Network Intelligence

Overview   Trials Detail

Experiment

Name
example_mnist

ID
KCiBytKB

Start Time
10/31/2018, 8:16:15 PM

End Time
none

LogPath
/home/quzha/nni/experiments/KCiBytKB/log

TrainingPlatform
local

Status

Status
EXPERIMENT_RUNNING

Duration 32h 40min
0 MaxDuration: 100h

TrialNum 412
0 MaxTrialNumber: 10000

Best Accuracy
0.992200

Time Spent
32h 40min

Remaining Time
67h 19min

Duration
100h

Succeed Trial
403

Stopped Trial
0

Failed Trial
9

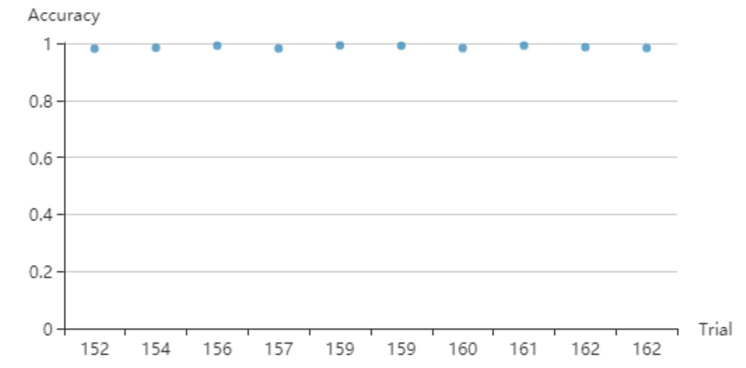
Search Space


```
{
  "dropout_rate": {
    "_type": "uniform",
    "_value": [
      0.1,
      0.5
    ]
  },
  "hidden_size": {
    "_type": "choice",
    "_value": [
      124,
      512,
      1024
    ]
  },
  "learning_rate": {
```


Trial Profile


```
{
  "revision": 11768,
  "authorName": "default",
  "trialConcurrency": 1,
  "clusterMetaData": [
    {
      "key": "codeDir",
      "value": "/home/quzha/nni/nni/examples/trials/mnist-hyperband/."
    },
    {
      "key": "command",
      "value": "python3 mnist.py"
    }
  ]
}
```


Optimization Progress



Top10 Trials

	Trial No.	Id	Duration	Status	Default Metric
+	159	t0I7T	13min 37s	SUCCEEDED	0.992200
+	161	ITVup	23min 27s	SUCCEEDED	0.991800
+	156	SPNVb	13min 17s	SUCCEEDED	0.991500
+	159	BLy3T	13min 58s	SUCCEEDED	0.991000
+	162	YeI3O	23min 34s	SUCCEEDED	0.986100
+	154	FnDZH	4min 15s	SUCCEEDED	0.984000
+	162	sEzTu	14min 17s	SUCCEEDED	0.983000
+	160	Uvk7y	14min 49s	SUCCEEDED	0.982600
+	157	gL2JW	6min 30s	SUCCEEDED	0.981800
+	152	g1WQs	6min 56s	SUCCEEDED	0.981000

Q&A

Deployment Options

- Microservice deployment (AKS)
- IOT Edge
- Different run-time: ONNX
    - How do you deploy inferencing/prediction code in different run times?
    - With ONNX you can
        - scikit-learn/tensorflow trained model → prediction code directly in C# or Java applications
        - AutoML can generate ONNX formatted models
        - For more info, go to the ONNX session (code: BRK3012, today@12:30pm)

Auto-SKLearn