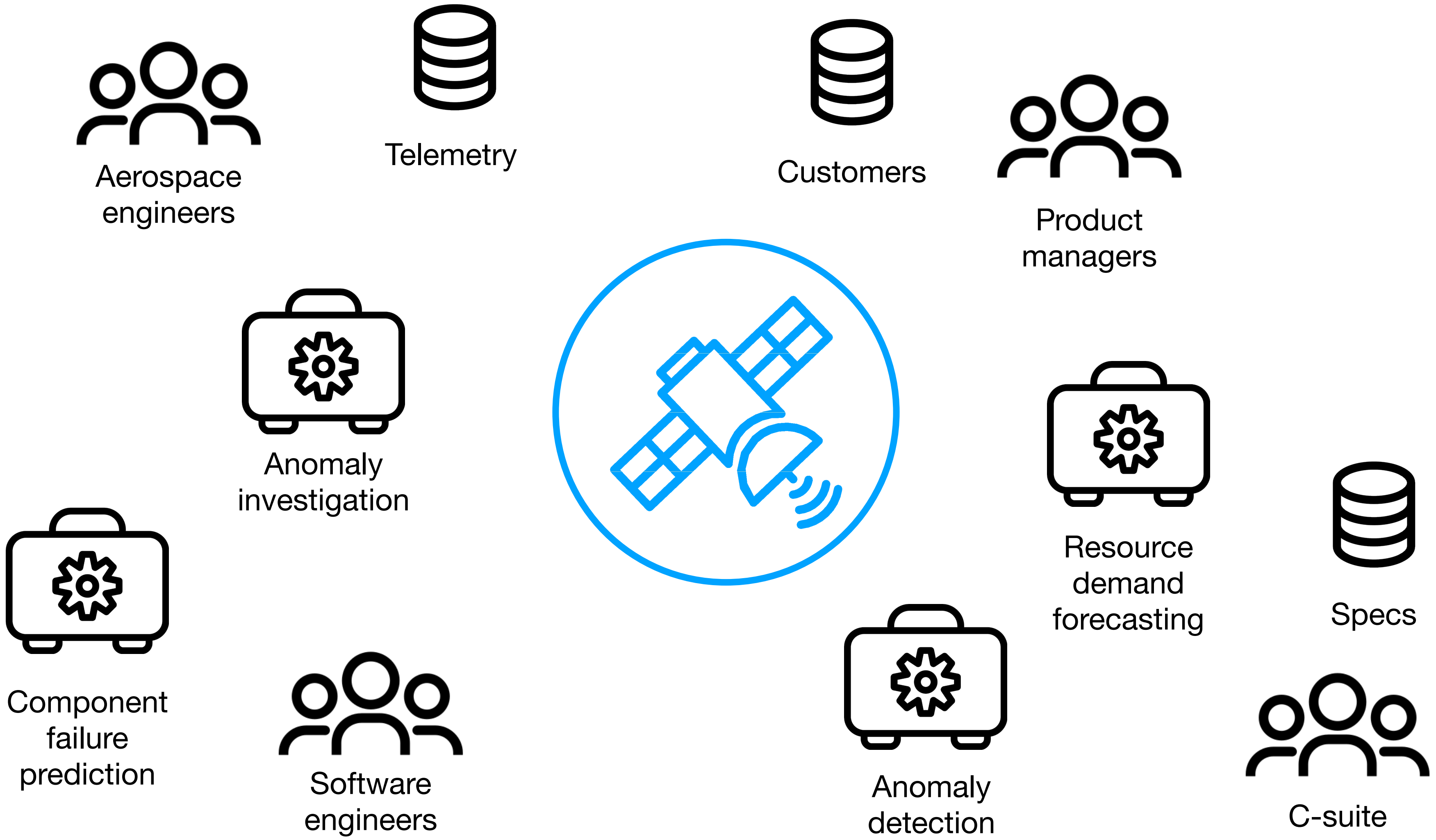
ACM SIGMOD/PODS International Conference on Management of Data, June 14-19, 2020 ~~Portland, OR, USA~~ The Internet

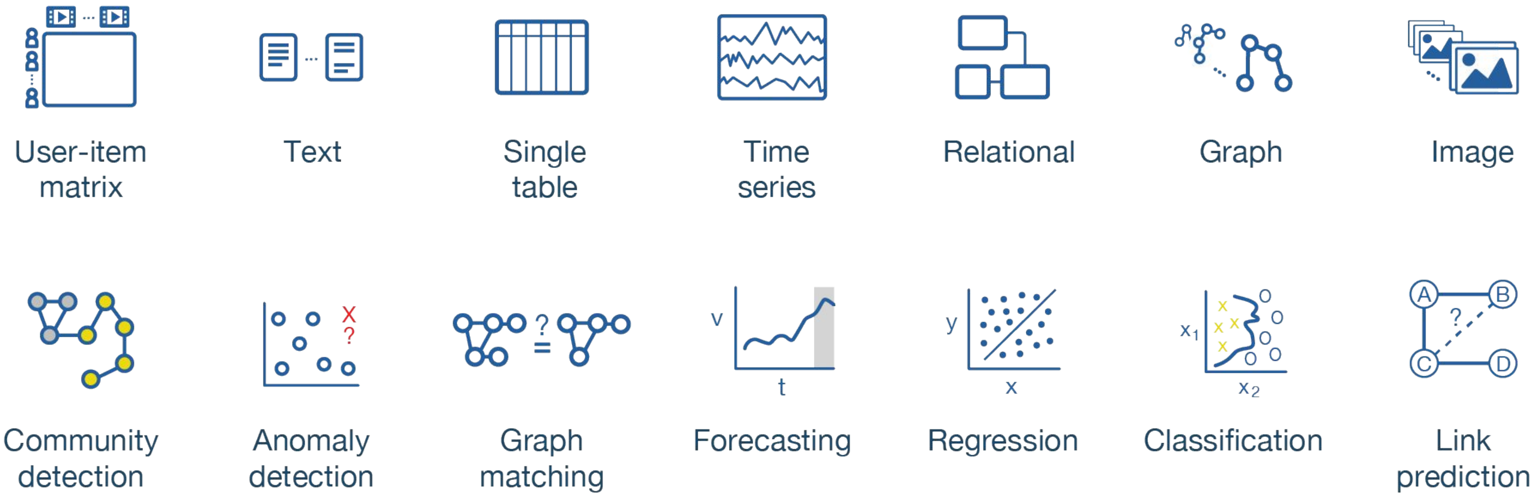
The Machine Learning Bazaar: Harnessing the ML Ecosystem for Effective System Development

Micah J. Smith[1], ***Carles Sala***[1], *James Max Kanter*[2], *Kalyan Veeramachaneni*[1]

[1]MIT, [2]Feature Labs

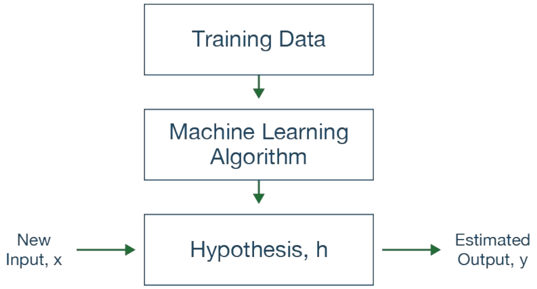
Deploying ML to solve real problems isn't easy.


Aerospace engineers
Telemetry
Customers
Product managers
Anomaly investigation
Resource demand forecasting
Specs
Component failure prediction
Software engineers
Anomaly detection
C-suite

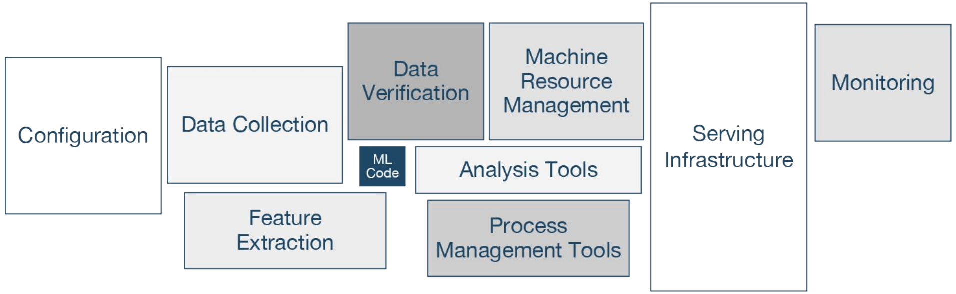

User-item matrix
Text
Single table
Time series
Relational
Graph
Image
Community detection
Anomaly detection
Graph matching
Forecasting
Regression
Classification
Link prediction

pipeline jungles lead to data management, MLOps, and usability challenges

Ideal



Real world



Sculley et al, 2015

How can we make building ML systems easier in practical settings?

The Machine Learning Bazaar

A framework for designing and developing ML and AutoML systems

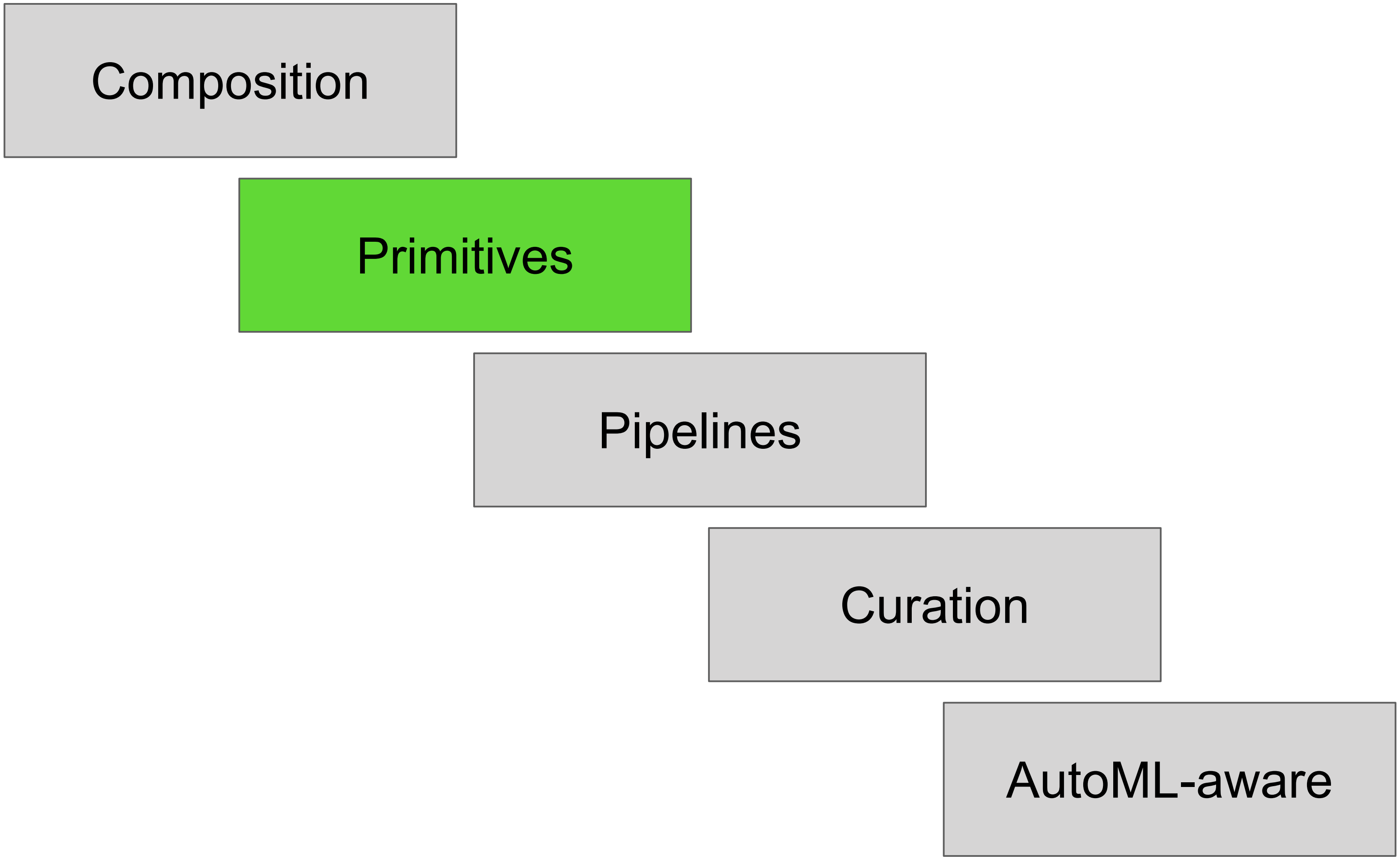
Composition

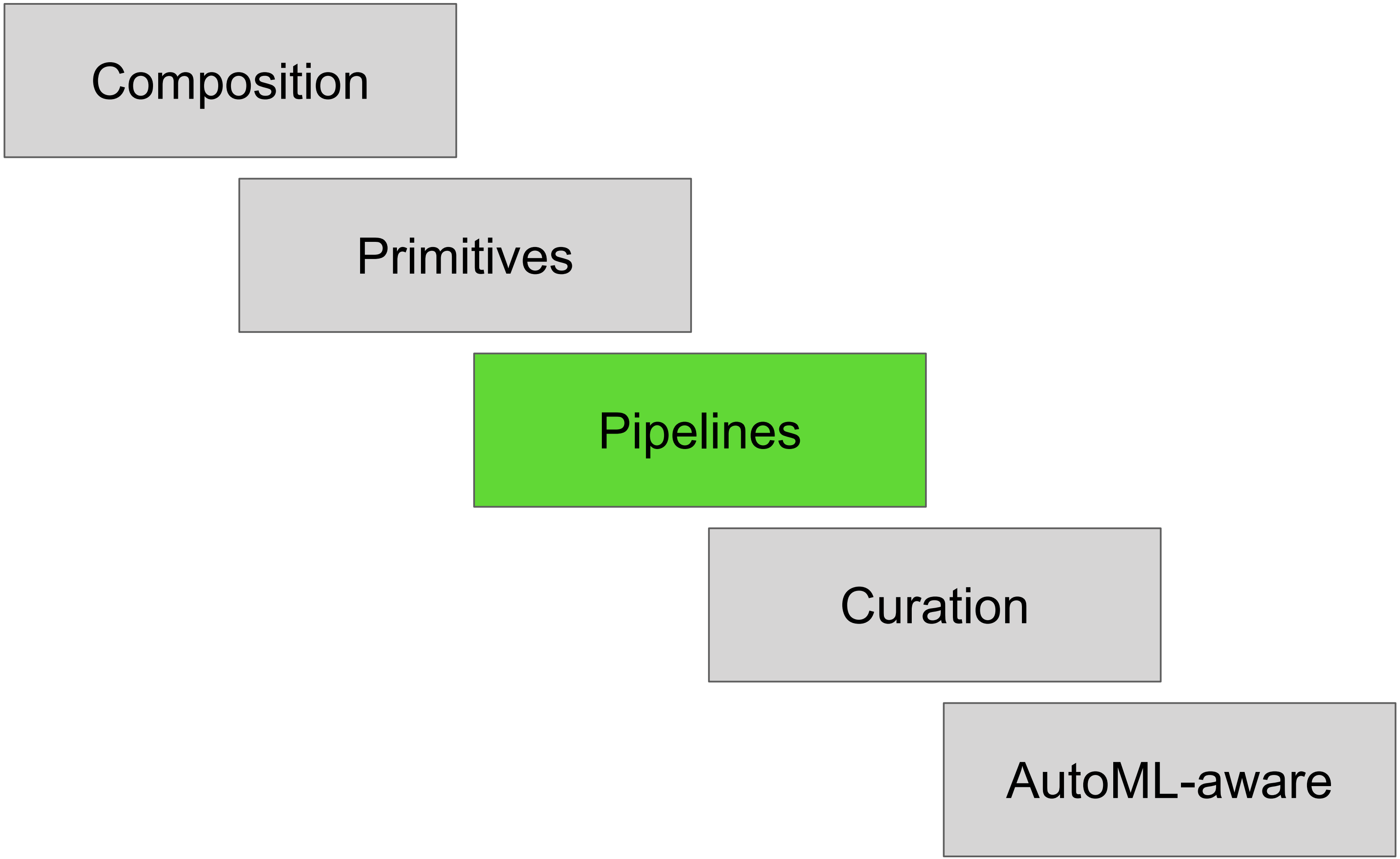
Primitives

Pipelines

Curation

AutoML-aware


Composition
Primitives
Pipelines
Curation
AutoML-aware


Composition
Primitives
Pipelines
Curation
AutoML-aware

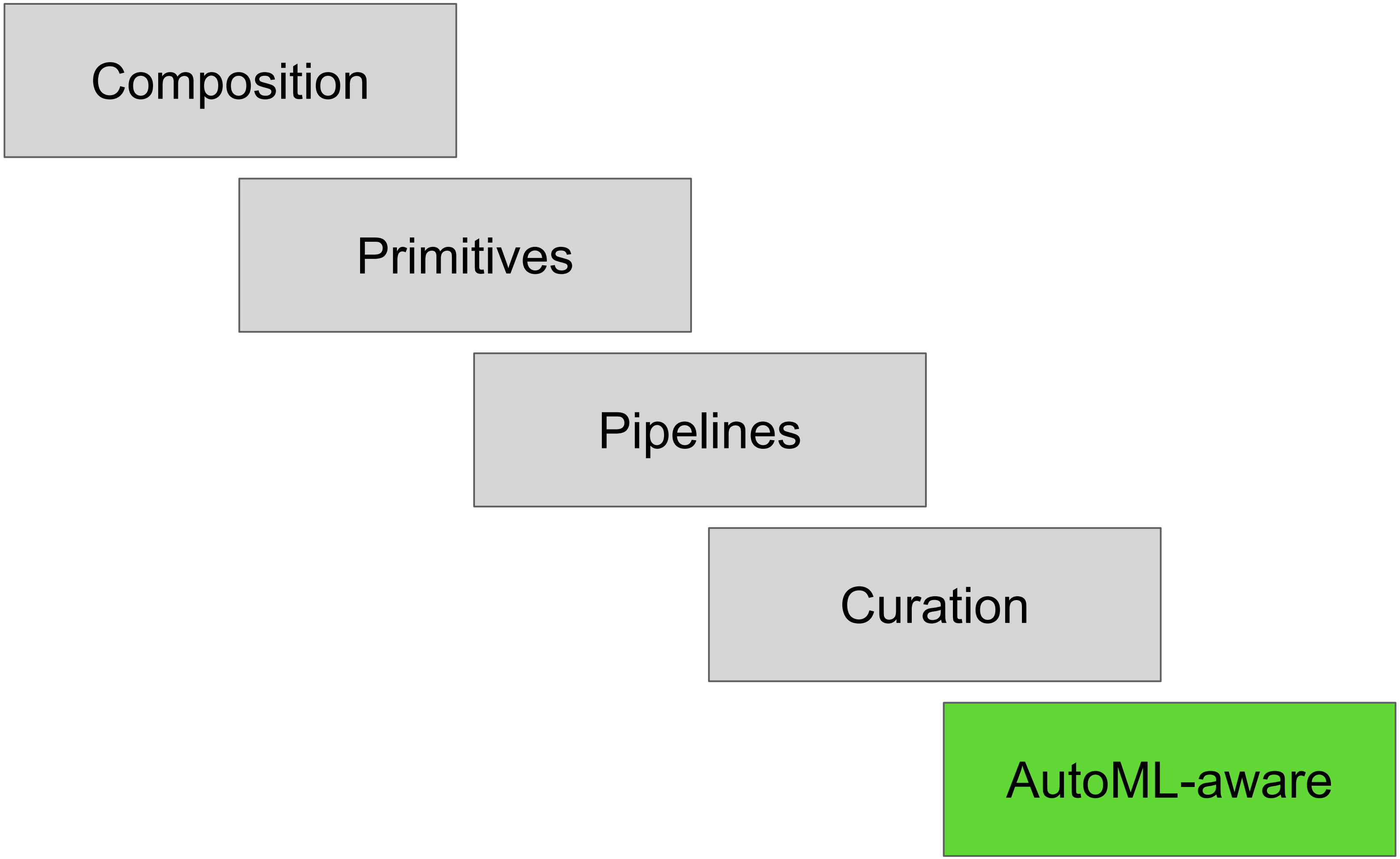
Composition

Primitives

Pipelines

Curation

AutoML-aware


Composition
Primitives
Pipelines
Curation
AutoML-aware

Core framework
- MLPrimitives
- MLBlocks
- BTB

ML applications
- Orion
- Greenguard
- Cardea
- water

AutoML systems
- mit-d3m-ta2
- AutoBazaar

The ML Bazaar

Curated collections
- primitives
- pipeline templates
- tuners
- selectors

Benchmarking
- ML Bazaar Task Suite
- mit-d3m
- d3m-dataset-downloader
- d3m-dataset-manager

Meta-analysis
- ml-pipelines data
- piex library
- ml-bazaar-analysis

Core framework
- MLPrimitives
- MLBlocks
- BTB

ML applications
- Orion
- Greenguard
- Cardea
- water

AutoML systems
- mit-d3m-ta2
- AutoBazaar

The ML Bazaar

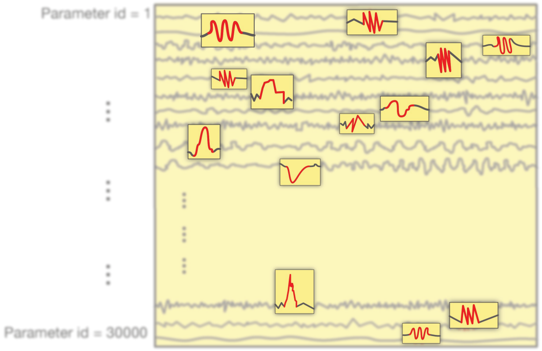
Curated collections
- primitives
- pipeline templates
- tuners
- selectors

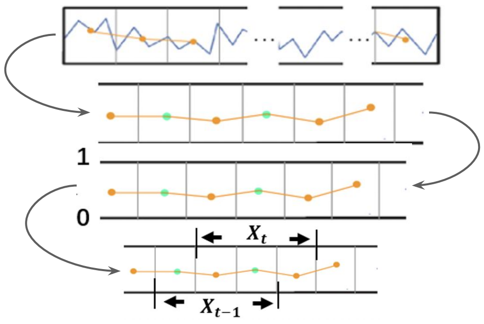
Benchmarking
- ML Bazaar Task Suite
- mit-d3m
- d3m-dataset-downloader
- d3m-dataset-manager

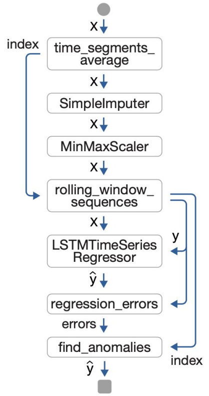
Meta-analysis
- ml-pipelines data
- piex library
- ml-bazaar-analysis

Orion Project

ML for telemetry data generated by satellites







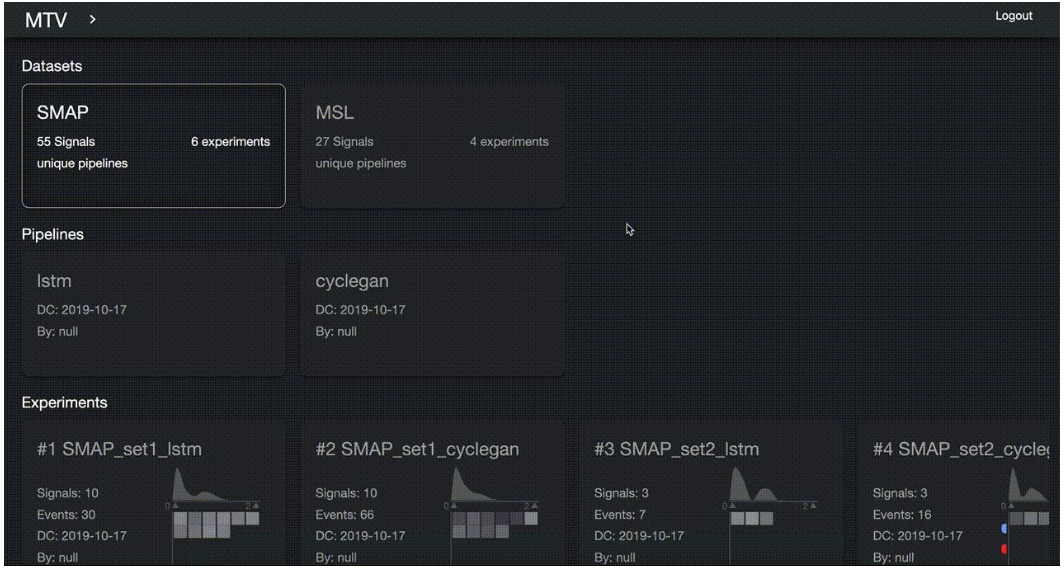
https://github.com/signals-dev/orion

Alnegheimish, Geiger, Liu, Sala, Veeramachaneni 2020

orion project

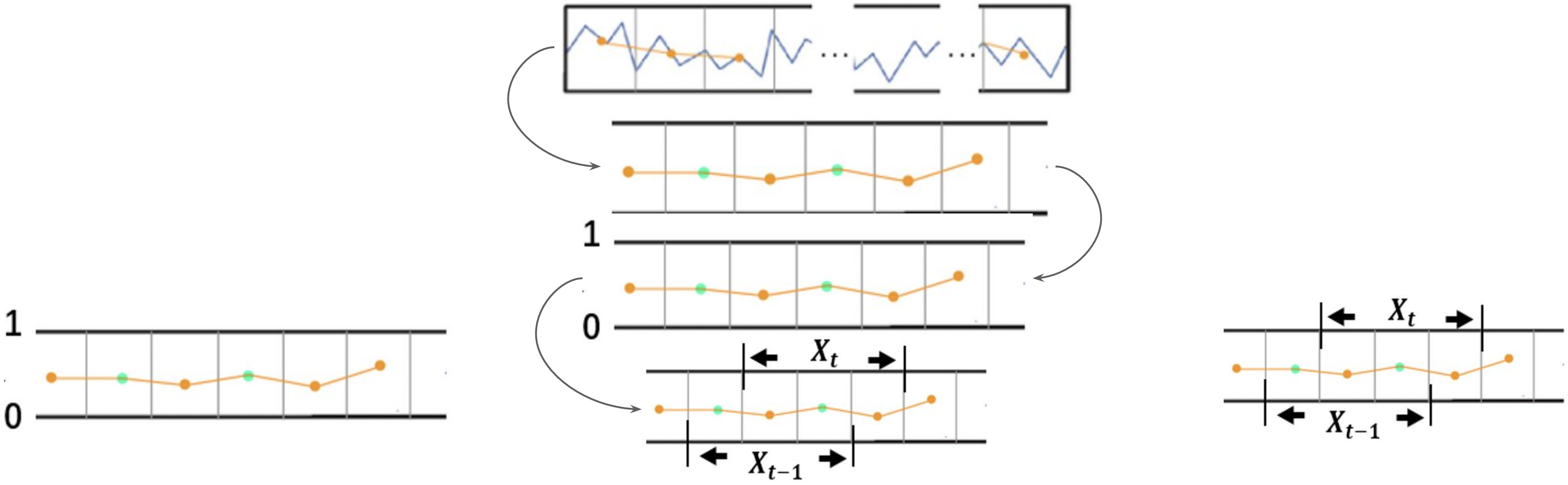
ML for telemetry data generated by satellites

pipeline	rank	holdout	accuracy	f1	precision	recall	elapsed
cyclegan.json	1	True	0.813596	0.388817	0.398299	0.684344	11679.3
cyclegan.json	2	False	0.835993	0.386045	0.420367	0.631183	11897.2
lstm_dynamic_threshold.json	3	False	0.862858	0.335507	0.418613	0.486285	2485.91
lstm_dynamic_threshold.json	4	True	0.833355	0.315074	0.402231	0.501001	1926.47
dummy.json	5	True	0.706477	0.255769	0.365821	0.460047	0.00778921
dummy.json	6	False	0.8049	0.229079	0.517508	0.286709	0.00542795
arima.json	7	False	0.52847	0.165833	0.211291	0.224588	537.567
arima.json	8	True	0.528483	0.165808	0.211391	0.224592	294.973
sum_24h_lstm.json	9	True	0.715023	0.059131	0.093512	0.0776259	3424.04
mean_24h_lstm.json	10	True	0.702763	0.0562528	0.0830615	0.109077	1808.26
median_24h_lstm.json	11	True	0.702342	0.0527362	0.0774	0.102952	2101.01
mean_24h_lstm.json	12	False	0.790045	0.0387521	0.103517	0.0432888	1848.14
sum_24h_lstm.json	13	False	0.821669	0.028044	0.0809271	0.0189294	3463.38
median_24h_lstm.json	14	False	0.803412	0.0245045	0.074956	0.033353	2138.08
skew_24h_lstm.json	15	False	0.831571	0.00575738	0.0178371	0.00343754	2244.59
skew_24h_lstm.json	16	True	0.733722	0.00412963	0.0110799	0.00257553	2185.13

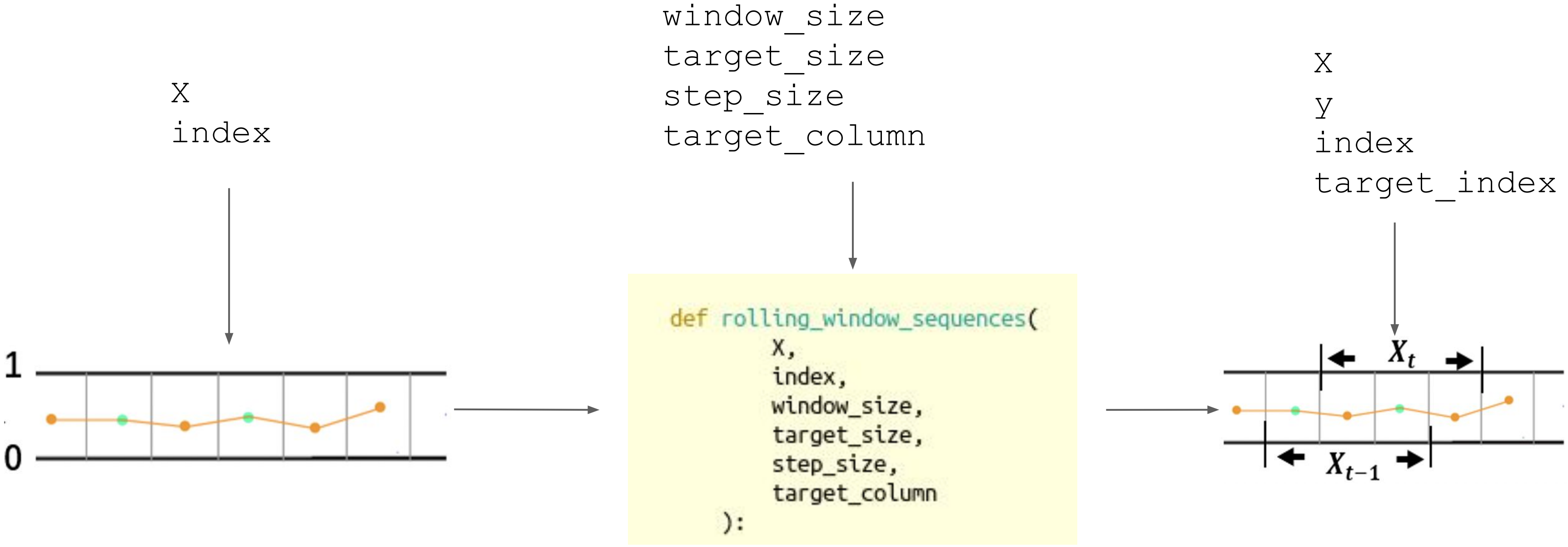


https://github.com/signals-dev/orion

Alnegheimish, Geiger, Liu, Sala, Veeramachaneni 2020



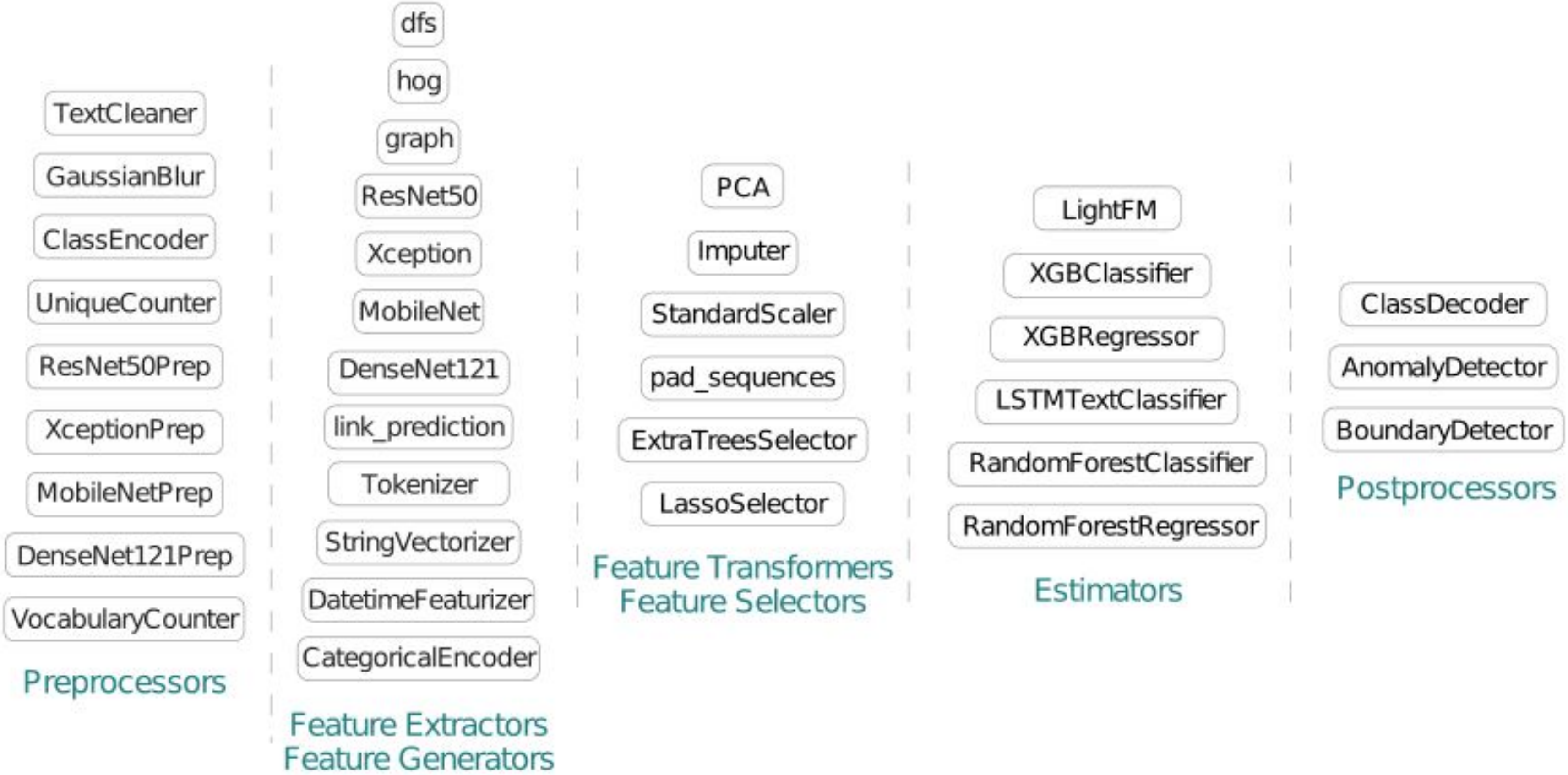
1
0
1
0
X_t
X_{t-1}
X_t
X_{t-1}


X
index
1
0
window_size
target_size
step_size
target_column
def rolling_window_sequences(
X,
index,
window_size,
target_size,
step_size,
target_column
):
X
y
index
target_index
X_t
X_{t-1}

Primitive

- Software component
- Self-contained
- Context agnostic
- Reusable

- Expect some input data
- Produce new data
- May learn from data
- May have hyperparameters


TextCleaner
GaussianBlur
ClassEncoder
UniqueCounter
ResNet50Prep
XceptionPrep
MobileNetPrep
DenseNet121Prep
VocabularyCounter
Preprocessors
dfs
hog
graph
ResNet50
Xception
MobileNet
DenseNet121
link_prediction
Tokenizer
StringVectorizer
DatetimeFeaturizer
CategoricalEncoder
Feature Extractors
Feature Generators
PCA
Imputer
StandardScaler
pad_sequences
ExtraTreesSelector
LassoSelector
Feature Transformers
Feature Selectors
LightFM
XGBClassifier
XGBRegressor
LSTMTextClassifier
RandomForestClassifier
RandomForestRegressor
Estimators
ClassDecoder
AnomalyDetector
BoundaryDetector
Postprocessors

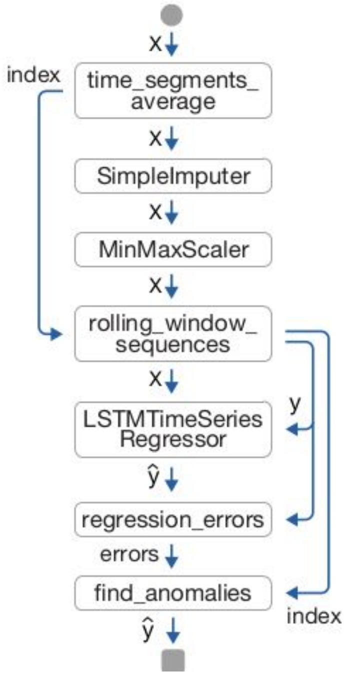
Primitive JSON

- Metadata
- Python Primitive
- Inputs
- Outputs
- Hyperparameters

```
{
  "name": "mlp.custom.ts_pre.rolling_window_sequences",
  "description": "Create rolling window sequences out of timeseries data.",
  "classifiers": {"type": "preprocessor", "subtype": "feature_extractor"},
  "modalities": ["timeseries"],
  "primitive": "mlp.custom.ts_pre.rolling_window_sequences",
  "produce": {
    "args": [
      {"name": "X", "type": "ndarray"},
      {"name": "index","type": "ndarray"}
    ],
    "output": [
      {"name": "X", "type": "ndarray"},
      {"name": "y", "type": "ndarray"},
      {"name": "index", "type": "ndarray"},
      {"name": "target_index", "type": "ndarray"}
    ]
  },
  "hyperparameters": {
    "fixed": {
      "target_size": {"type": "int", "default": 1},
      "target_column": {"type": "str or int", "default": 1}
    },
    "tunable": {
      "window_size": {"type": "int", "default": 250, "range": [1, 1000]},
      "step_size": {"type": "int", "default": 1, "range": [1, 1000]}
    }
  }
}
```

Pipelines

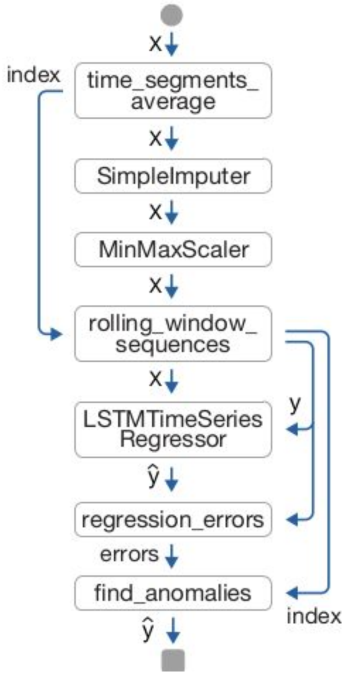
- Collection of Primitives
- Directed Acyclic Graph
- Single computational unit



Pipeline JSON

- Primitives List
- User specified parameters
- Variable Mappings

```
{
  "primitives": [
    "mlp.custom.ts_pre.time_segments_average",
    "sklearn.impute.SimpleImputer",
    "sklearn.preprocessing.MinMaxScaler",
    "mlp.custom.ts_pre.rolling_window_sequences",
    "keras.Sequential.LSTMTimeSeriesRegressor",
    "mlp.custom.ts_anomalies.regression_errors",
    "mlp.custom.ts_anomalies.find_anomalies"
  ],
  "init_params": {
    "mlp.custom.ts_pre.time_segments_average#1": {
      "time_column": "timestamp",
      "interval": 21600
    },
    "sklearn.preprocessing.MinMaxScaler#1": {
      "feature_range": [-1, 1]
    },
    "mlp.custom.ts_pre.rolling_window_sequences#1": {
      "target_column": 0,
      "window_size": 250
    },
    "keras.Sequential.LSTMTimeSeriesRegressor": {
      "epochs": 35
    }
  },
  "input_names": {
    "mlp.custom.ts_anomalies.find_anomalies#1": {"index": "target_index"}
  },
  "output_names": {
    "keras.Sequential.LSTMTimeSeriesRegressor#1": {"y": "y_hat"}
  }
}
```



MLPrimitives

- Primitives and pipelines repository
- Custom Python primitives and third party tools
- Tunable hyperparameters curated by experts
- Community driven contributions

Source	Count	Source	Count
scikit-learn	39	XGBoost	2
MLPrimitives (custom)	27	LightFM	1
Keras	25	OpenCV	1
pandas	16	python-louvain	1
Featuretools	4	scikit-image	1
NumPy	3	statsmodels	1
NetworkX	2		

Primitives in the curated catalog of MLPrimitives. Catalogs maintained by individual projects may contain more primitives.

https://github.com/HDI-Project/MLPrimitives, MLPrimitives v0.2.4

MLBlocks

- Uniform API for any Python library
- Simplify construction of complex Pipelines
- Machine-readable tunable hyperparameters
- JSON language for primitives and pipelines

```python
from mlblocks import MLPipeline
from mlprimitives.datasets import load_dataset

dataset = load_dataset('census')
X_train, X_test, y_train, y_test = dataset.get_splits(1)

primitives = [
    'mlprimitives.custom.preprocessing.ClassEncoder',
    'mlprimitives.custom.feature_extraction.CategoricalEncoder',
    'sklearn.impute.SimpleImputer',
    'xgboost.XGBClassifier',
    'mlprimitives.custom.preprocessing.ClassDecoder'
]
pipeline = MLPipeline(primitives)

pipeline.fit(X_train, y_train)
predictions = pipeline.predict(X_test)

dataset.score(y_test, predictions)
```

https://github.com/HDI-Project/MLBlocks,
MLBlocks v0.3.4

DARPA D3M Program

design and implement AutoML systems that can produce a solution to arbitrary ML tasks with minimal human involvement

Lippman et al, 2016,
https://gitlab.com/datadrivendiscovery,
https://www.darpa.mil/program/data-driven-discovery-of-models

DARPA D3M Program

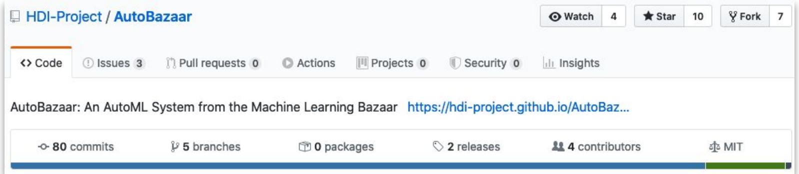
design and implement AutoML systems that can produce a solution to arbitrary ML tasks with minimal human involvement

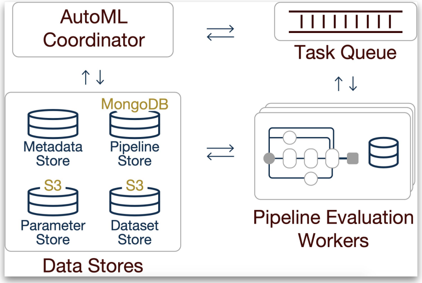
System	Top pipeline	Beats Expert 1	Beats Expert 2	Rank
System 1	29	57	31	1
ML Bazaar	18	56	28	2
System 3	15	47	22	3
System 4	14	46	21	4
System 5	10	42	14	5
System 6	8	43	15	6
System 7	8	33	12	7
System 8	6	24	11	8
System 9	4	25	13	9
System 10	2	27	12	10

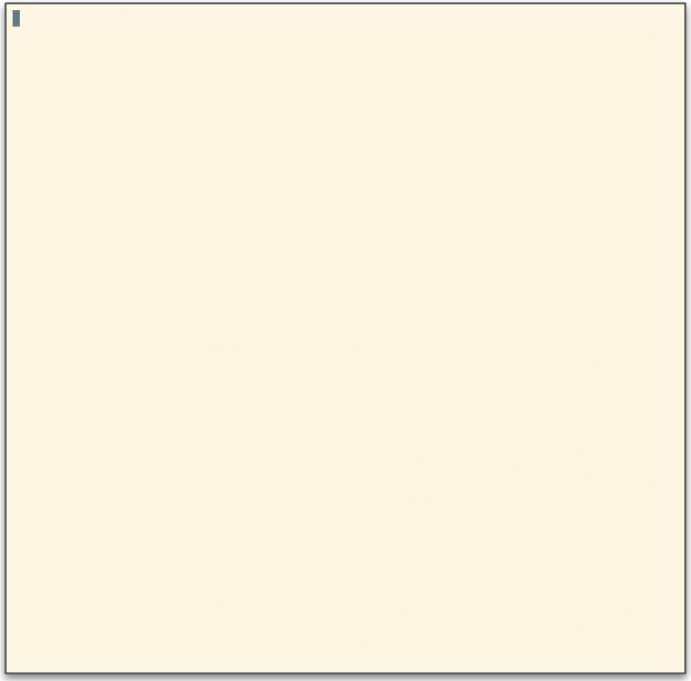
Results from DARPA D3M Summer 2019 evaluation. Highlight System 6 (Shang et al, 2019) and System 7 (Drori et al, 2018) as publicly presented comparisons.

Lippman et al, 2016,
https://gitlab.com/datadrivendiscovery,
https://www.darpa.mil/program/data-driven-discovery-of-models

AutoBazaar







https://github.com/HDI-Project/AutoBazaar

Evaluation

1. ML Bazaar Task Suite
2. AutoBazaar computational performance
3. AutoBazaar AutoML performance
4. Expressiveness of curated primitives and pipelines
5. Case study of ML primitive comparison
6. Case study of AutoML primitive comparison
7. Lessons from 5 applications
8. Evaluation from DARPA D3M program

https://github.com/micahjsmith/ml-bazaar-analysis

Evaluation

1. ML Bazaar Task Suite
2. AutoBazaar computational performance
3. AutoBazaar AutoML performance
4. Expressiveness of curated primitives and pipelines
5. Case study of ML primitive comparison
6. Case study of AutoML primitive comparison
7. Lessons from 5 applications
8. Evaluation from DARPA D3M program

https://github.com/micahjsmith/ml-bazaar-analysis

ML Bazaar Task Suite

Data Modality	Problem Type	Tasks
graph	community detection	2
	graph matching	9
	link prediction	1
	vertex nomination	1
image	classification	5
	regression	1
multi table	classification	6
	regression	7
single table	classification	234
	collaborative filtering	4
	regression	87
	timeseries forecasting	35
text	classification	18
	regression	9
timeseries	classification	37

	min	p25	p50	p75	max
Number of examples	7	202	599	3,634	6,095,521
Number of classes[†]	2	2	3	6	115
Columns of X	1	3	9	22	10,937
Size (compressed)	3KiB	21KiB	145KiB	2MiB	36GiB
Size (inflated)	22KiB	117KiB	643KiB	7MiB	42GiB

mlbazaar.github.io

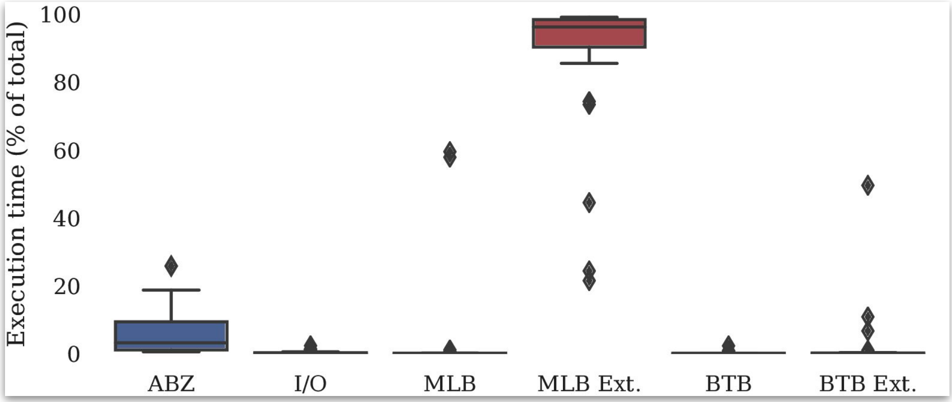
AutoBazaar evaluation

- search and score 2.5 million pipelines on 456 tasks
- independent and parallel execution on heterogeneous cluster of 400 AWS EC2 nodes depending on task workload size

AutoBazaar evaluation

almost zero runtime overhead due to ABZ in searching for pipelines

opportunities for further pipeline-aware optimizations



Execution time of AutoBazaar pipeline search attributable to different components. The box plot shows quartiles of the distribution, 1.5x IQR, and outliers. `Ext` refers to calls to external libraries providing underlying implementations like sklearn `GaussianProcessRegressor`.

Crankshaw et al, 2015; Boehm et al, 2016; Shoumik et al, 2018

Thank you!

Read our paper: `bit.ly/mlbazaar-paper`
Browse the project: `mlbazaar.github.io`
Use our libraries: `pip install mlbazaar`
Talk to us: `mlbazaar@mit.edu`

Special thanks to: Plamen Kolev, Laura Gustafson, William Xue, Akshay Ravikumar, Ihssan Tinawi, Alexander Geiger, Sarah Alnegheimish, Saman Amarasinghe, Dongyu Liu, Stefanie Jegelka, Zi Wang, Benjamin Schreck, Seth Rothschild, Manual Alvarez Campo, Sebastian Peral, Peter Fontana, and Brian Sandberg.

The Machine Learning Bazaar: Harnessing the ML Ecosystem for Effective System Development

Micah J. Smith[1], ***Carles Sala***[1], *James Max Kanter*[2], *Kalyan Veeramachaneni*[1]

[1]MIT, [2]Feature Labs