

What's new in mlflow™?

A System to Accelerate the Machine Learning Lifecycle

Corey Zumar

Software Engineer, Databricks

Outline

Overview of new developments

Feature deep dives

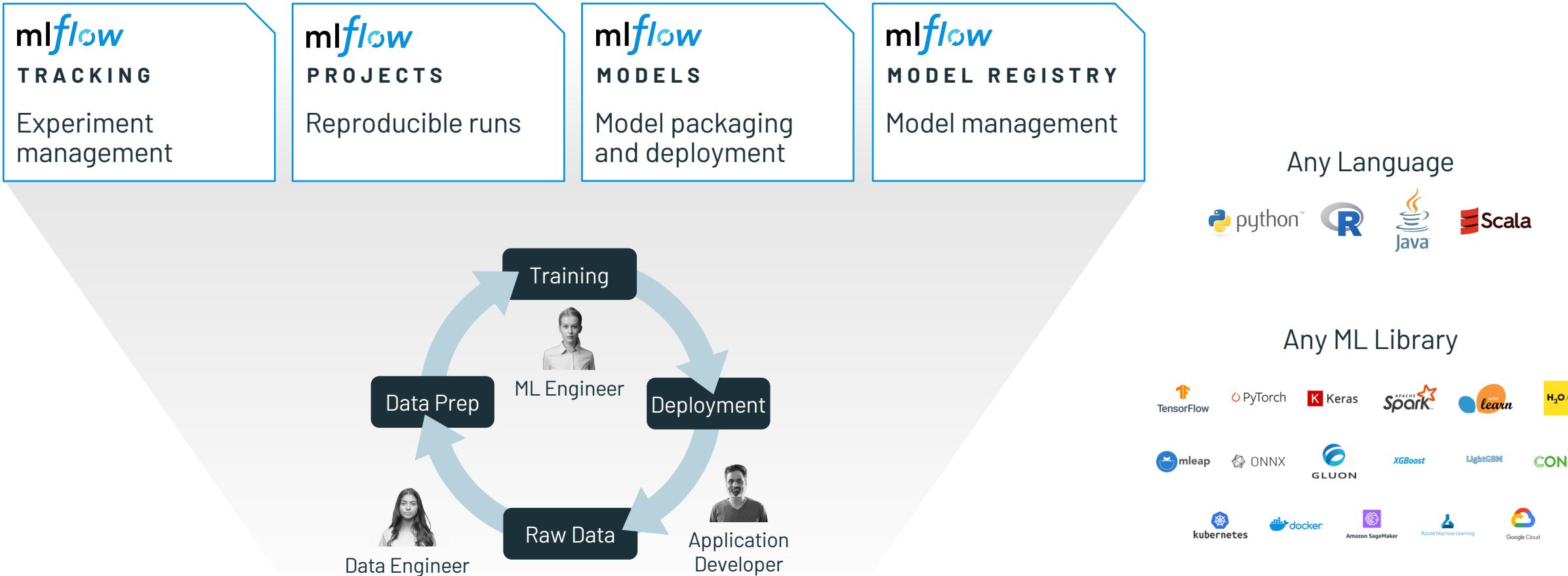
- Autologging for scikit-learn

- Model schemas & input examples

- MLflow's plugin ecosystem

What's next for MLflow?

mlflow: An Open Source ML Platform



mlflow: Overview of new developments

mlflow

TRACKING

- Scikit-learn autologging (1.11)
- Fast.ai autologging (1.9)
- UI accessibility, syntax highlighting for artifacts + PDF support (1.9, 1.11)

mlflow

PROJECTS

- Backend plugin support (1.9)
- YARN execution backend
- Expanded artifact resolution capabilities (1.9)

mlflow

MODELS

- Model schemas & input examples (1.9)
- Deployment plugin support (1.9)
- Spacy model flavor (1.8)

mlflow

MODEL REGISTRY

- Tags for registered models and model versions (1.9)
- Enhanced version comparison UI, including schemas (1.11)
- Simplified model archiving (1.10)

Outline

Overview of new developments

Feature deep dives

- Autologging for scikit-learn

- Model signatures & input examples

- MLflow's plugin ecosystem

What's next for MLflow?

Instrumenting scikit-learn workflows

```
mlflow.start_run()
mlflow.log_param("alpha", alpha)
mlflow.log_param("l1_ratio", l1_ratio)
model = ElasticNet(alpha, l1_ratio)
model.fit(X, y)
y_pred = model.predict(X)
loss = mean_absolute_error(y, y_pred)
mlflow.log_metric("loss", loss)
mlflow.sklearn.log_model(model)
mlflow.end_run()
```

Instrumenting scikit-learn workflows

ElasticNet

```
mlflow.start_run()
mlflow.log_param("alpha", alpha)
mlflow.log_param("l1_ratio",
l1_ratio)
model = ElasticNet(alpha, l1_ratio)
model.fit(X, y)
y_pred = model.predict(X)
loss = mean_absolute_error(y, y_pred)
mlflow.log_metric("loss", loss)
mlflow.sklearn.log_model(model)
mlflow.end_run()
```



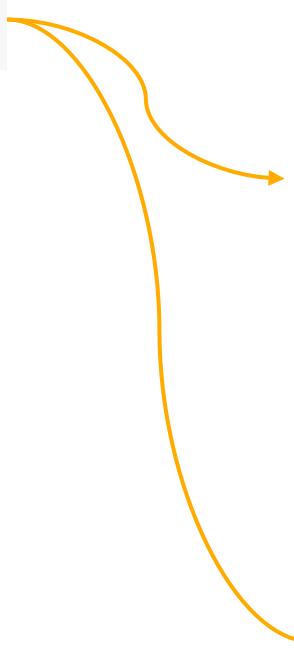
| | | | |
|---------------------------|--------------------------------|--------------------------------|----------------------------|
| ARDRegression | GradientBoostingClassifier | MeanShift | RFE |
| AdaBoostClassifier | GradientBoostingRegressor | MinCovDet | RFECV |
| AdaBoostRegressor | GraphicalLasso | MinMaxScaler | RadiusNeighborsClassifier |
| AdditiveChi2Sampler | GraphicalLassoCV | MiniBatchDictionaryLearning | RadiusNeighborsRegressor |
| AffinityPropagation | GridSearchCV | MiniBatchKMeans | RadiusNeighborsTransformer |
| AgglomerativeClustering | HashingVectorizer | MiniBatchSparsePCA | RandomForestClassifier |
| BaggingClassifier | HistGradientBoostingClassifier | MissingIndicator | RandomForestRegressor |
| BaggingRegressor | HistGradientBoostingRegressor | MultiLabelBinarizer | RandomTreesEmbedding |
| BayesianGaussianMixture | HuberRegressor | MultiOutputClassifier | RandomizedSearchCV |
| BayesianRidge | IncrementalPCA | MultiOutputRegressor | RegressorChain |
| BernoulliNB | IsolationForest | MultiTaskElasticNet | Ridge |
| BernoulliRBM | Isomap | MultiTaskElasticNetCV | RidgeCV |
| Binarizer | IsotonicRegression | MultiTaskLasso | RidgeClassifier |
| Birch | IterativeImputer | MultiTaskLassoCV | RidgeClassifierCV |
| CCA | KBinsDiscretizer | MultinomialNB | RobustScaler |
| CalibratedClassifierCV | KMeans | NMF | SGDClassifier |
| CategoricalNB | KNNGraph | NearestCentroid | SGDRegressor |
| ClassifierChain | KNeighborsClassifier | NearestNeighbors | SVC |
| ColumnTransformer | KNeighborsRegressor | NeighborhoodComponentsAnalysis | SVR |
| ComplementNB | KNeighborsTransformer | Normalizer | SelectFdr |
| CountVectorizer | KernelCenterer | NuSVC | SelectFpr |
| DBSCAN | KernelDensity | NuSVR | SelectFromModel |
| DecisionTreeClassifier | KernelPCA | Nystroem | SelectFwe |
| DecisionTreeRegressor | KernelRidge | OAS | SelectKBest |
| DictVectorizer | LabelBinarizer | OPTICS | SelectPercentile |
| DictionaryLearning | LabelEncoder | OneClassSVM | ShrunkCovariance |
| DummyClassifier | LabelPropagation | OneHotEncoder | SimpleImputer |
| DummyRegressor | LabelSpreading | OneVsOneClassifier | SkewedChi2Sampler |
| ElasticNet | Lars | OneVsRestClassifier | SparseCoder |
| ElasticNetCV | LarsCV | OrdinalEncoder | SparsePCA |
| EllipticEnvelope | Lasso | OrthogonalMatchingPursuit | SparseRandomProjection |
| EmpiricalCovariance | LassoCV | OrthogonalMatchingPursuitCV | SpectralBiclustering |
| ExtraTreeClassifier | LassoLars | OutputCodeClassifier | SpectralClustering |
| ExtraTreeRegressor | LassoLarsCV | PCA | SpectralCoclustering |
| ExtraTreesClassifier | LassoLarsIC | PLSCanonical | SpectralEmbedding |
| ExtraTreesRegressor | LatentDirichletAllocation | PLSRegression | StackingClassifier |
| FactorAnalysis | LedoitWolf | PLSSVD | StackingRegressor |
| FastICA | LinearDiscriminantAnalysis | PassiveAggressiveClassifier | StandardScaler |
| FeatureAgglomeration | LinearRegression | PassiveAggressiveRegressor | TSNE |
| FeatureHasher | LinearSVC | PatchExtractor | TfidfTransformer |
| FeatureUnion | LinearSVR | Perceptron | TfidfVectorizer |
| FunctionTransformer | LocalOutlierFactor | Pipeline | TheilSenRegressor |
| GammaRegressor | LocallyLinearEmbedding | PoissonRegressor | TransformedTargetRegressor |
| GaussianMixture | LogisticRegression | PolynomialFeatures | TruncatedSVD |
| GaussianNB | LogisticRegressionCV | PowerTransformer | TweedieRegressor |
| GaussianProcessClassifier | MDS | QuadraticDiscriminantAnalysis | VarianceThreshold |
| GaussianProcessRegressor | MLPClassifier | QuantileTransformer | VotingClassifier |
| GaussianRandomProjection | MLPRegressor | RANSACRegressor | VotingRegressor |
| GenericUnivariateSelect | MaxAbsScaler | RBFSampler | |

new in 1.11

mlflow Autologging for scikit-learn

One line to record params, metrics and models:

```
mlflow.sklearn.autolog()
```



2020-09-25 00:04:01 CDT

- alpha: 0.5
- copy_X: True
- fit_intercept: True
- l1_ratio: 0.213
- max_iter: 1000
- normalize: False
- positive: False
- precompute: False
- random_state: 42
- selection: cyclic
- tol: 0.0001
- warm_start: False

- training_mae: 0.567
- training_mse: 0.503
- training_r2_score: 0.204
- training_rmse: 0.709
- training_score: 0.204

Full Path: /tmp/mlruns/0/17d6d66505624f758881dcc650...
Size: 1009B

artifact_path: model
flavors:
python_function:
env: conda.yaml
loader_module: mlflow.sklearn
model_path: model.pkl
python_version: 3.7.5
sklearn:
pickled_model: model.pkl
serialization_format: cloudpickle
sklearn_version: 0.22.1
run_id: 17d6d66505624f758881dcc650c773cc
saved_input_example_info:
artifact_path: input_example.json
pandas_orient: split
type: dataframe

| | | Parameters | Metrics > | | |
|--|---------------------|--------------|-----------|-----------------|------------------|
| | Start Time | Source | C | mean_test_score | mean_train_score |
| | 2020-09-25 00:27:23 | sklearn_pres | - | - | - |
| | 2020-09-25 00:27:23 | sklearn_pres | 5 | 0.987 | 0.982 |
| | 2020-09-25 00:27:23 | sklearn_pres | 5 | 0.98 | 0.978 |
| | 2020-09-25 00:27:23 | sklearn_pres | 10 | 0.973 | 0.978 |
| | 2020-09-25 00:27:23 | sklearn_pres | 1 | 0.98 | 0.982 |
| | 2020-09-25 00:27:23 | sklearn_pres | 1 | 0.967 | 0.973 |
| | 2020-09-25 00:27:23 | sklearn_pres | 10 | 0.98 | 0.985 |

jupyter scikit-learn autologging Last Checkpoint: 3 hours ago (unsaved changes) 

File Edit View Insert Cell Kernel Widgets Help Trusted

Trusted

In []: `import pandas as pd
X = pd.read_csv('housing_data.csv')
y = pd.read_csv('housing_labels.csv')`

In []: `import numpy as np
from sklearn.linear_model import ElasticNet

model = ElasticNet(alpha=0.5, l1_ratio=0.25)
model.fit(X, y)`

jupyter scikit-learn autologging Last Checkpoint: 3 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted

In []: `import pandas as pd
X = pd.read_csv('housing_data.csv')
y = pd.read_csv('housing_labels.csv')`

In []: `import mlflow
mlflow.sklearn.autolog()`

In []: `import numpy as np
from sklearn.linear_model import ElasticNet

model = ElasticNet(alpha=0.5, l1_ratio=0.25)
model.fit(X, y)`

Default

Experiment ID: 0

Artifact Location : /tmp/artifacts/0

▼ Notes 

None

Search Runs: metrics.rmse < 1 and params.model = "tree" and tags.mlflow.source.type = "LOCA

?

State:

Active ▾

Search

Clear

Showing 1 matching run

Compare

Delete

[Download CSV](#)

1

6

Columns

| | | Parameters > | | | Metrics < | | | |
|--------------------------|---------------------|--------------|-------|--------|---------------|--------------|--------------|-------------------|
| | Start Time | User | alpha | copy_X | fit_intercept | training_mae | training_mse | training_r2_score |
| <input type="checkbox"/> | 2020-09-26 14:41:04 | czumar | 0.5 | True | True | 3.496 | 24.99 | 0.704 |

▼ Parameters

| Name | Value |
|---------------|--------|
| alpha | 0.5 |
| copy_X | True |
| fit_intercept | True |
| l1_ratio | 0.25 |
| max_iter | 1000 |
| normalize | False |
| positive | False |
| precompute | False |
| random_state | None |
| selection | cyclic |
| tol | 0.0001 |
| warm_start | False |

| | |
|--------------|--------|
| random_state | None |
| selection | cyclic |
| tol | 0.0001 |
| warm_start | False |

▼ Metrics

| Name | Value |
|---|-------|
| training_mae  | 3.496 |
| training_mse  | 24.99 |
| training_r2_score  | 0.704 |
| training_rmse  | 4.999 |
| training_score  | 0.704 |

▼ Tags

training_mse 

24.99

training_r2_score 

0.704

training_rmse 

4.999

training_score 

0.704

▼ Tags

| Name | Value | Actions |
|-----------------|---|---|
| estimator_class | sklearn.linear_model._coordinate_descent.ElasticNet |   |
| estimator_name | ElasticNet |   |

Add Tag

▼ Artifacts

File Edit View Insert Cell Kernel Widgets Help

Trusted



```
In [ ]: import pandas as pd  
X = pd.read_csv('housing_data.csv')  
y = pd.read_csv('housing_labels.csv')
```

```
In [ ]: import numpy as np  
from sklearn.linear_model import ElasticNet  
from sklearn.model_selection import GridSearchCV  
  
search_cv = GridSearchCV(ElasticNet(), {"alpha": np.arange(0.1, 1.1, .1)})
```

```
In [ ]: search_cv.fit(X, y)
```

Search Runs: metrics.rmse < 1 and params.model = "tree" and tags.mlflow.source.type = "LOCAL"

State:

Active ▾

Search

Clear

Showing 11 matching runs

Compare

Delete

Download CSV 

Columns

| | | | Parameters | Metrics | |
|--------------------------|---------------------|--------|------------|-----------------|--|
| | Start Time | User | alpha | mean_test_score | |
| <input type="checkbox"/> | 2020-09-26 15:29:03 | czumar | - | - | |
| <input type="checkbox"/> | 2020-09-26 15:29:03 | czumar | 0.2 | 0.466 | |
| <input type="checkbox"/> | 2020-09-26 15:29:03 | czumar | 0.7 | 0.483 | |
| <input type="checkbox"/> | 2020-09-26 15:29:03 | czumar | 0.3 | 0.477 | |
| <input type="checkbox"/> | 2020-09-26 15:29:03 | czumar | 1.0 | 0.473 | |
| <input type="checkbox"/> | 2020-09-26 15:29:03 | czumar | 0.9 | 0.476 | |
| <input type="checkbox"/> | 2020-09-26 15:29:03 | czumar | 0.6 | 0.485 | |
| <input type="checkbox"/> | 2020-09-26 15:29:03 | czumar | 0.1 | 0.444 | |
| <input type="checkbox"/> | 2020-09-26 15:29:03 | czumar | 0.4 | 0.483 | |
| <input type="checkbox"/> | 2020-09-26 15:29:03 | czumar | 0.8 | 0.48 | |
| <input type="checkbox"/> | 2020-09-26 15:29:03 | czumar | 0.5 | 0.485 | |

None

▼ Parameters

| Name | Value |
|--------------------|---|
| best_alpha | 0.5 |
| cv | None |
| error_score | nan |
| estimator | ElasticNet() |
| iid | deprecated |
| n_jobs | None |
| param_grid | {'alpha': [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]} |
| pre_dispatch | 2*n_jobs |
| refit | True |
| return_train_score | False |
| scoring | None |
| verbose | 0 |

▼ Metrics

None

▼ Parameters

| Name | Value |
|--------------------|---|
| best_alpha | 0.5 |
| cv | None |
| error_score | nan |
| estimator | ElasticNet() |
| iid | deprecated |
| n_jobs | None |
| param_grid | {'alpha': [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]} |
| pre_dispatch | 2*n_jobs |
| refit | True |
| return_train_score | False |
| scoring | None |
| verbose | 0 |

▼ Metrics

| | |
|--------------------|----------|
| pre_dispatch | 2*n_jobs |
| refit | True |
| return_train_score | False |
| scoring | None |
| verbose | 0 |

▼ Metrics

| Name | Value |
|---|-------|
| best_cv_score  | 0.485 |
| training_mae  | 3.48 |
| training_mse  | 24.85 |
| training_r2_score  | 0.706 |
| training_rmse  | 4.985 |
| training_score  | 0.706 |

▼ Tags

| Name | Value | Actions |
|------|-------|---------|
|------|-------|---------|

mlflow Autologging: supported libraries



Outline

Overview of new developments

Feature deep dives

Autologging for scikit-learn

Model schemas & input examples

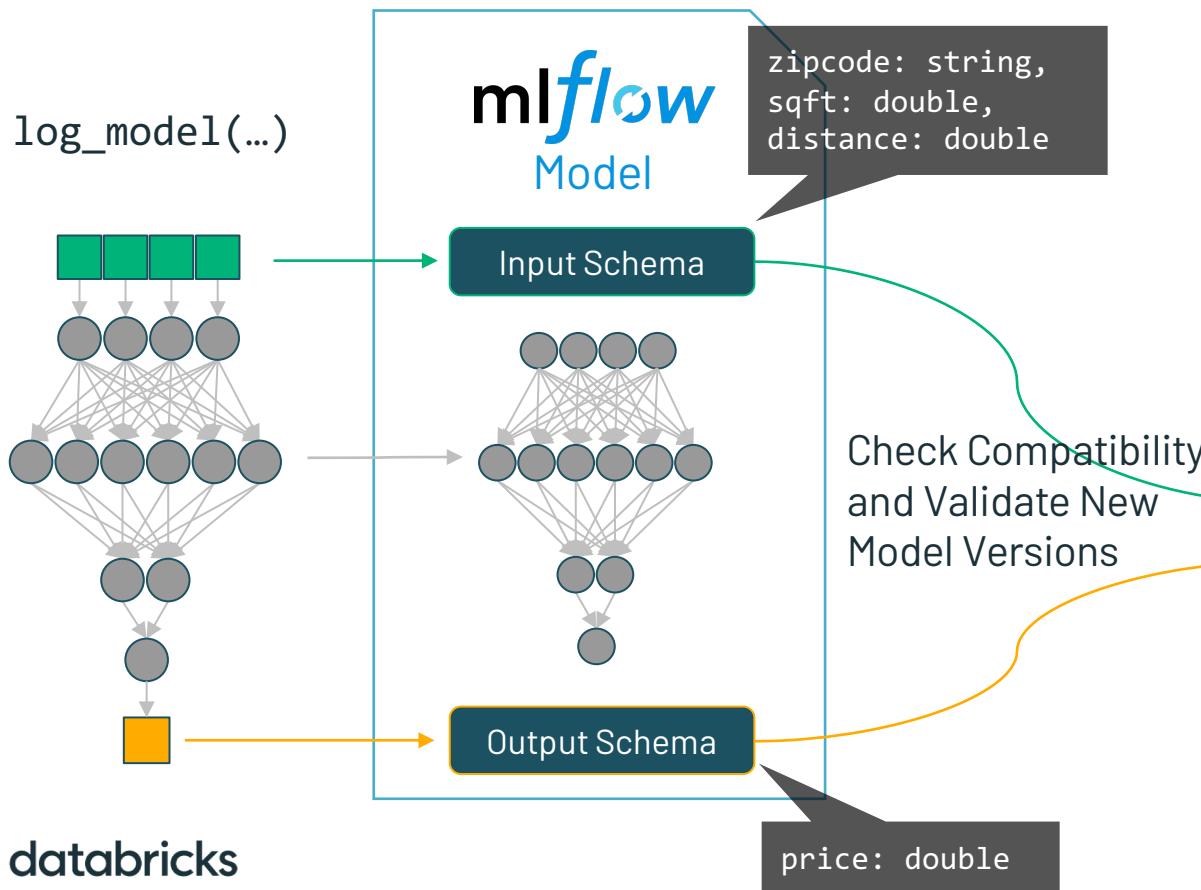
MLflow's plugin ecosystem

What's next for MLflow?

new in 1.9

mlflow Model Schemas

Specify input and output data types for models



Registered Models > SchemaTest > Comparing 2 Versions

| Run ID: | 954568e525e842f4865ae11398be0d06 | c088997da44f45f7bbd659b7d4bf0bb1 |
|--------------------------|----------------------------------|----------------------------------|
| Model Version: | 1 | 2 |
| Run Name: | | |
| Start Time: | 2020-05-26 14:20:58 | 2020-05-26 14:23:07 |
| ▼ Parameters | | |
| Parameters are identical | | |
| Incompatible schemas! | | |
| ▼ Schema | | |
| Inputs | | |
| sqft | string | long |
| zipcode | string | long |
| Outputs | | |
| price | string | long |
| ▼ Metrics | | |
| Metrics are identical | | |

Show diff only

Compare by column name Show diff only

Show diff only

mlflow Model Schemas

Infer model input / output signature from data

```
infer_signature(  
    inputs,  
    outputs  
)
```



```
inputs: [  
    'year built': long,  
    'year sold': long,  
    'lot area': long,  
    'zip code': long,  
    'quality': long  
]
```

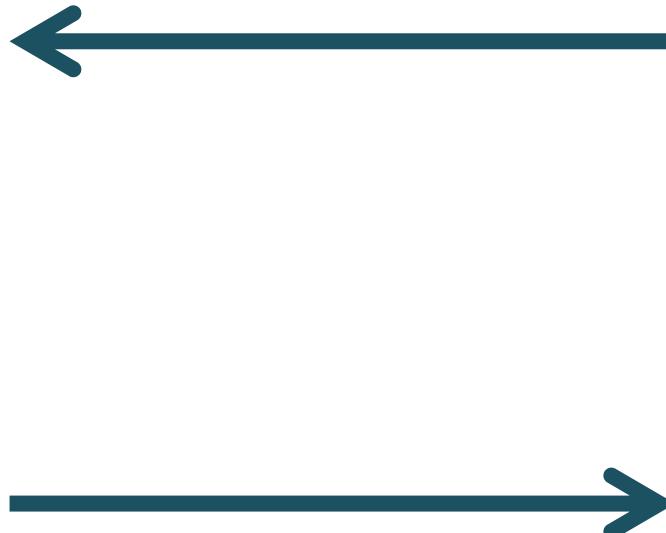
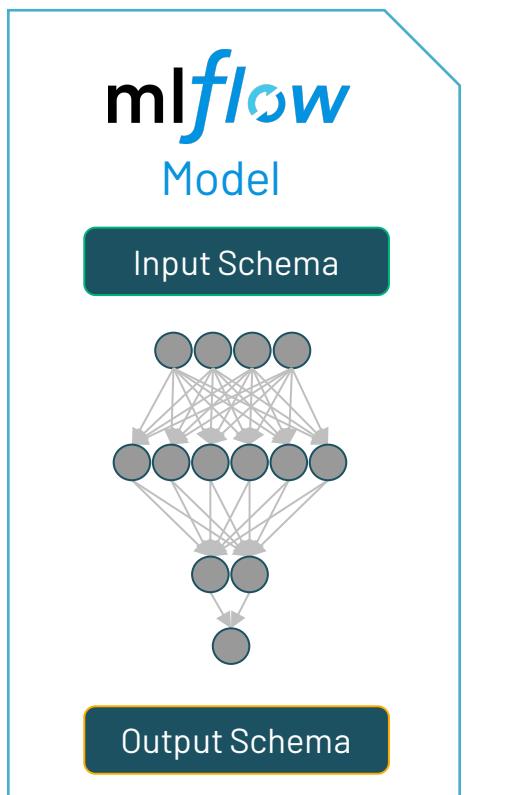
```
outputs: ['sale price': double]
```



```
log_model(  
    ...,  
    signature  
)
```

mlflow Model Schemas

Validate inputs against schema during inference



```
input_frame = pd.DataFrame.from_dict({  
    "year built": data["year built"],  
    "year sold": data["year sold"],  
    "lot area": data["lot area"],  
    "zip code": data["zip code"],  
    "condition": data["condition"],  
})
```

Schema Mismatch Error





File Edit View Insert Cell Kernel Widgets Help

Trusted



```
In [ ]: from mlflow.models import Model  
Model.load('runs:/d6cd66f93d5942c0bcf80db06efdbb6a/model').signature
```

```
In [ ]: import mlflow.pyfunc  
model = mlflow.pyfunc.load_model('runs:/d6cd66f93d5942c0bcf80db06efdbb6a/model')
```

```
In [ ]: import pandas as pd  
x = pd.DataFrame.from_records([  
    "Year Sold": 2007,  
    "Year Built": 1983,  
    "Lot Area": 2650,  
    "Zip Code": 6, # categorically encoded  
    "Condition": 7 # wrong field  
])  
x
```

```
In [ ]: model.predict(x)
```

```
In [ ]:
```

File Edit View Insert Cell Kernel Widgets Help

Trusted



```
In [1]: from mlflow.models import Model  
Model.load('runs:/d6cd66f93d5942c0bcf80db06efdbb6a/model').signature
```

```
Out[1]: inputs:  
        ['Year Sold': long, 'Lot Area': long, 'Year Built': long, 'Zip Code': long, 'Quality': long]  
outputs:  
        ['Sale Price': double]
```

```
In [ ]: import mlflow.pyfunc  
model = mlflow.pyfunc.load_model('runs:/d6cd66f93d5942c0bcf80db06efdbb6a/model')
```

```
In [ ]: import pandas as pd  
x = pd.DataFrame.from_records([  
    "Year Sold": 2007,  
    "Year Built": 1983,  
    "Lot Area": 2650,  
    "Zip Code": 6, # categorically encoded  
    "Condition": 7 # wrong field  
])  
x
```

```
In [ ]: model.predict(x)
```

```
In [ ]:
```

File Edit View Insert Cell Kernel Widgets Help

Trusted



```
model = mlflow.pyfunc.load_model('runs:/d6cd66f93d5942c0bcf80db06efdbb6a/model')
```

```
In [3]: import pandas as pd
x = pd.DataFrame.from_records([{
    "Year Sold": 2007,
    "Year Built": 1983,
    "Lot Area": 2650,
    "Zip Code": 6, # categorically encoded
    "Condition": 7 # wrong field
}])
x
```

Out[3]:

| | Year Sold | Year Built | Lot Area | Zip Code | Condition |
|---|-----------|------------|----------|----------|-----------|
| 0 | 2007 | 1983 | 2650 | 6 | 7 |

```
In [ ]: model.predict(x)
```

In []:

File Edit View Insert Cell Kernel Widgets Help

Trusted



outputs:
['Sale Price': double]

In [2]: `import mlflow.pyfunc
model = mlflow.pyfunc.load_model('runs:/d6cd66f93d5942c0bcf80db06efdbb6a/model')`

In []: `import pandas as pd
x = pd.DataFrame.from_records([
 "Year Sold": 2007,
 "Year Built": 1983,
 "Lot Area": 2650,
 "Zip Code": 6, # categorically encoded
 "Condition": 7 # wrong field
}])
x`

In []: `model.predict(x)`

In []:

File Edit View Insert Cell Kernel Widgets Help

Trusted



```
model = mlflow.pyfunc.load_model('runs:/d6cd66f93d5942c0bcf80db06efdbb6a/model')
```

```
In [3]: import pandas as pd
x = pd.DataFrame.from_records([{
    "Year Sold": 2007,
    "Year Built": 1983,
    "Lot Area": 2650,
    "Zip Code": 6, # categorically encoded
    "Quality": 7
}])
x
```

Out[3]:

| | Year Sold | Year Built | Lot Area | Zip Code | Quality |
|---|-----------|------------|----------|----------|---------|
| 0 | 2007 | 1983 | 2650 | 6 | 7 |

```
In [ ]: model.predict(x)
```

In []:

Registered Models > pricing_model > Version 1 ▾

Registered At: 2020-09-28 21:59:43

Creator:

Stage: None ▾

Last Modified: 2020-09-28 21:59:43

Source Run: Run d6cd66f93d5942c0bcf80db06efdbb6a

▼ Description 

None

▼ Tags

| Name | Value | Actions |
|----------------|-------|---------|
| No tags found. | | |

Add Tag

▼ Schema

▼ Tags

| Name | Value | Actions |
|----------------|-------|---------|
| No tags found. | | |

Add Tag

| | | |
|---------------------------|----------------------------|------------------------------------|
| <input type="text"/> Name | <input type="text"/> Value | <input type="button" value="Add"/> |
|---------------------------|----------------------------|------------------------------------|

▼ Versions

All

Active(0)

Compare

| <input type="checkbox"/> | Version | Registered at | Created by | Stage |
|--------------------------|-----------|---------------------|------------|-------------------------------------|
| <input type="checkbox"/> | Version 1 | 2020-09-28 21:59:43 | | <input type="button" value="None"/> |
| <input type="checkbox"/> | Version 2 | 2020-09-28 21:59:50 | | <input type="button" value="None"/> |

Run ID:

d6cd66f93d5942c0bcf80db06efdbb6a

1aaf8140aed14c3188152cfa2dc25630

Model Version:

1

2

Run Name:

Start Time:

2020-09-28 00:58:21

2020-09-28 21:58:52

▶ Parameters

 Show diff only

▼ Schema

 Ignore column ordering Show diff only

Inputs

inputs [0]

Year Sold: long

Year Sold: long

inputs [1]

Lot Area: long

Lot Area: long

inputs [2]

Year Built: long

Zip Code: long

inputs [3]

Zip Code: long

Year Built: long

inputs [4]

Quality: long

Condition: long

Outputs

outputs [0]

Sale Price: double

Sale Price: double

▶ Metrics

 Show diff only

Run ID:

d6cd66f93d5942c0bcf80db06efdbb6a

1aaf8140aed14c3188152cfa2dc25630

Model Version:

1

2

Run Name:

Start Time:

2020-09-28 00:58:21

2020-09-28 21:58:52

Parameters

 Show diff only

Schema

 Ignore column ordering Show diff only

Inputs

inputs [2]

Year Built: long

Zip Code: long

inputs [3]

Zip Code: long

Year Built: long

inputs [4]

Quality: long

Condition: long

Outputs

Schema outputs are identical

Metrics

 Show diff only[Scatter Plot](#)[Contour Plot](#)[Parallel Coordinates Plot](#)

Outline

Overview of new developments

Feature deep dives

Autologging for scikit-learn

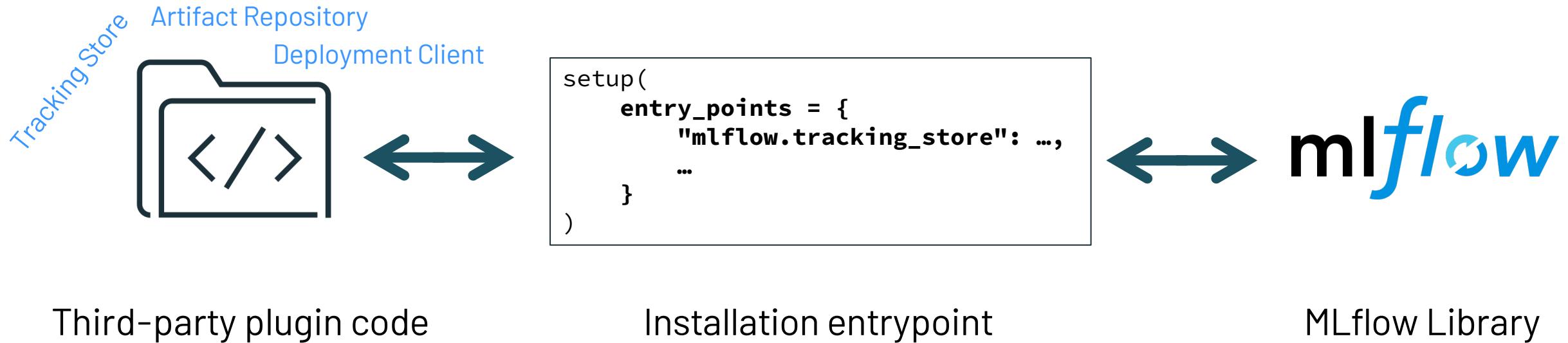
Model schemas & input examples

MLflow's plugin ecosystem

What's next for MLflow?

mlflow Plugins

Seamlessly integrate third-party code & infrastructure with MLflow



mlflow Plugins

Supported plugin types:

Tracking storage & search ([Tracking Store](#))

Tracking metadata collection ([Run Context Provider](#))

Model registry storage & search ([Model Registry Store](#))

Project execution ([Project Backend](#))

Model deployment ([Deployment Client](#))

Creating a plugin

1. Implement the plugin interface
(<https://mlflow.org/docs/latest/plugins.html>)
2. Add an MLflow plugin entrypoint & upload your package to PyPI
3. Add your plugin to the MLflow docs:
<https://mlflow.org/docs/latest/plugins.html#community-plugins>

mlflow Deployments API

new in 1.9

Pluggable way to create and manage deployment endpoints in MLflow

Used in new  RedisAI endpoint

Other integrations being ported:



Amazon
SageMaker



Azure Machine Learning

```
mlflow deployments create -t redisai -n spam  
-m models:/SpamScorer/production
```

```
mlflow deployments predict -t redisai -n spam  
-f emails.json
```

mlflow Project Backend Plugins

new in 1.9

Pluggable way to execute MLflow Projects on a variety of compute resources

Used for new YARN backend

```
mlflow run --backend yarn  
https://github.com/mlflow/mlflow#examples/pytorch
```

Other integrations being ported:



mlflow Community Plugins

Elastic Search backend for MLflow Tracking (experimental)

(<https://pypi.org/project/mlflow-elasticsearchstore/>)

Model deployment to RedisAI (<https://pypi.org/project/mlflow-redisai/>)

Project execution on YARN (<https://pypi.org/project/mlflow-yarn/>)

Artifact storage in SQL Server

(<https://pypi.org/project/mlflow-dbstore/>)

Artifact storage in Alibaba Cloud OSS

(<https://pypi.org/project/aliyunstoreplugin/>)

Outline

Overview of new developments

Feature deep dives

- Autologging for scikit-learn

- Model schemas & input examples

- MLflow's plugin ecosystem

What's next for MLflow?

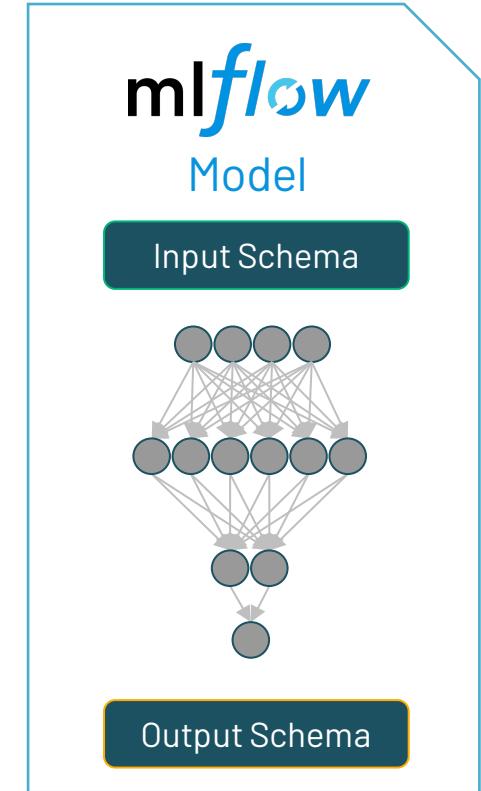
What's next for **mlflow** ?

Model explainability integrations (see [Data + AI Summit Europe 2020](#))

Support for tensor input / output schemas

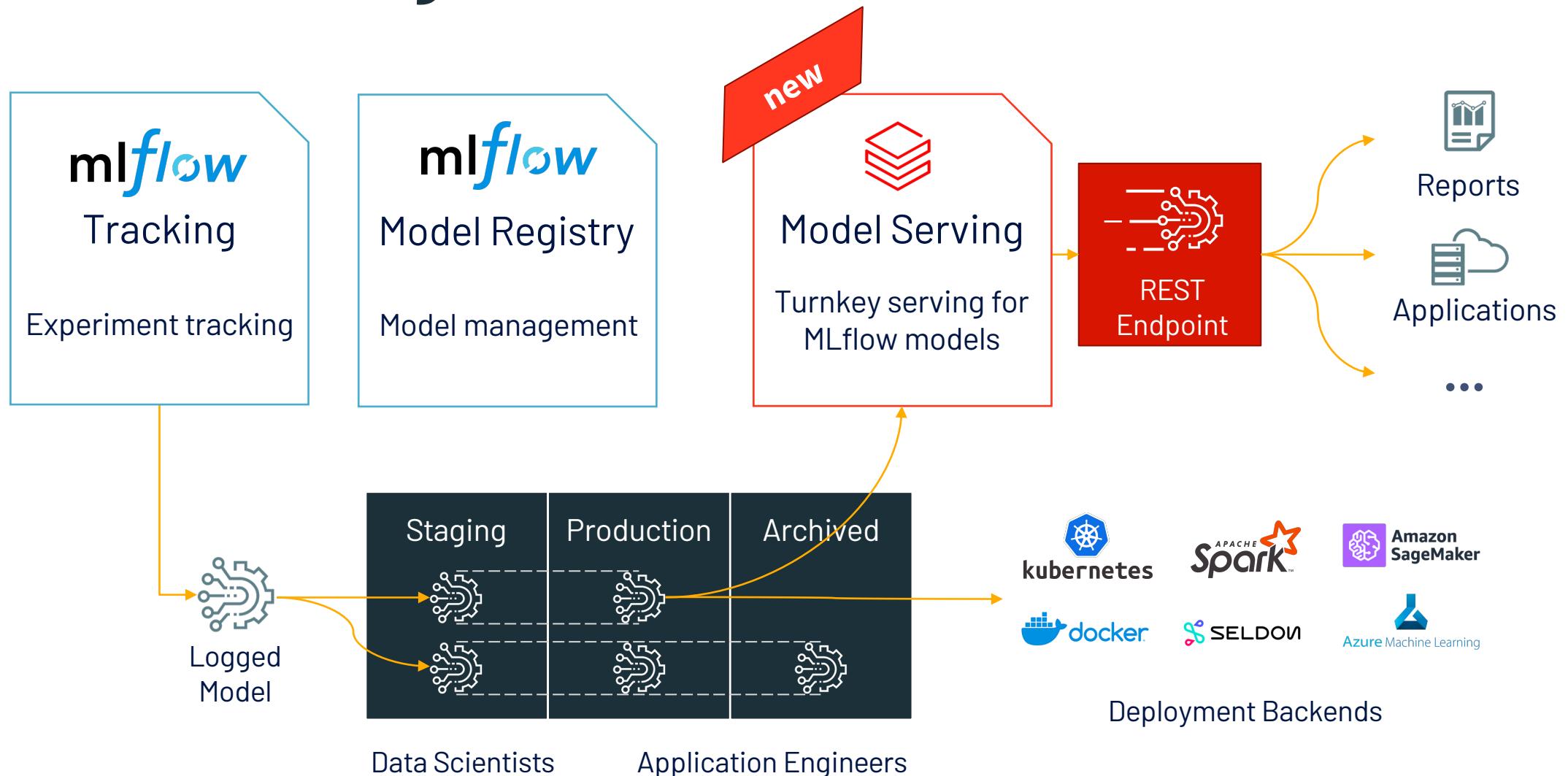
Expanded input example & schema collection for autologging (XGBoost, TensorFlow, and more)

`autolog()`



public preview

Model Serving on Databricks



DATA+AI SUMMIT EUROPE

17-19 NOVEMBER | VIRTUAL

FORMERLY KNOWN AS SPARK+AI SUMMIT

ORGANIZED BY



databricks

[DATABRICKS.COM/DATAAISUMMIT](https://databricks.com/dataaisummit)

Thank you!



`pip install mlflow`

GitHub Repository: github.com/mlflow/mlflow

Website: mlflow.org

Slack server: <https://tinyurl.com/mlflow-slack-server>