SQL Day

14 edycja konferencji SQLDay

9-11 maja 2022, WROCŁAW + ONLINE

Data Community

partner złoty

Future Processing

Infinite DATA

KPMG

partner srebrny

Objectivity

summ-it

UBS

dbWatch DATABASE CONTROL

partner brązowy

elitmind think bright

Data Community

Tomaž Kaštrun, MVP

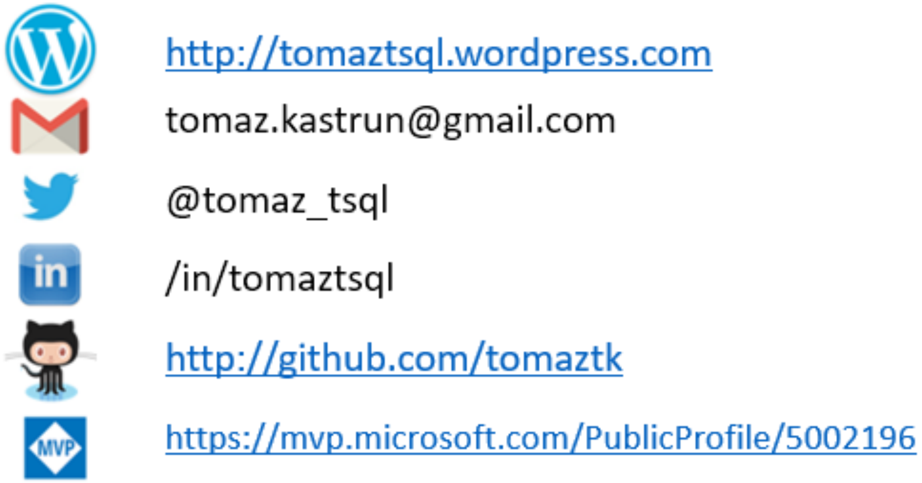
Azure Machine Learning services with Python

SQLDay 2022

SQL Day

About

- BI Developer and Data Scientist
- MSSQL, SAS, R, Py, C#, SAP, SPSS
- 20+years experience MSSQL, DEV, BI, DM
- MVP, MSCA, MCP, MCT
- Avid coffee drinker & Bicycle junkie



http://tomaztsql.wordpress.com

tomaz.kastrun@gmail.com

@tomaz_tsql

/in/tomaztsql

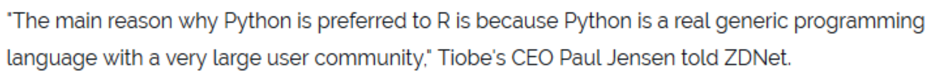
http://github.com/tomaztk

https://mvp.microsoft.com/PublicProfile/5002196

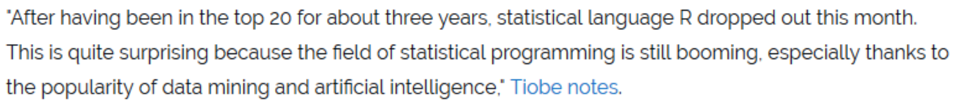
Code for session today:
https://github.com/tomaztk

Where Python is standing...

R vs Python: R's out of top 20 programming languages despite boom in statistical jobs (May, 2019)

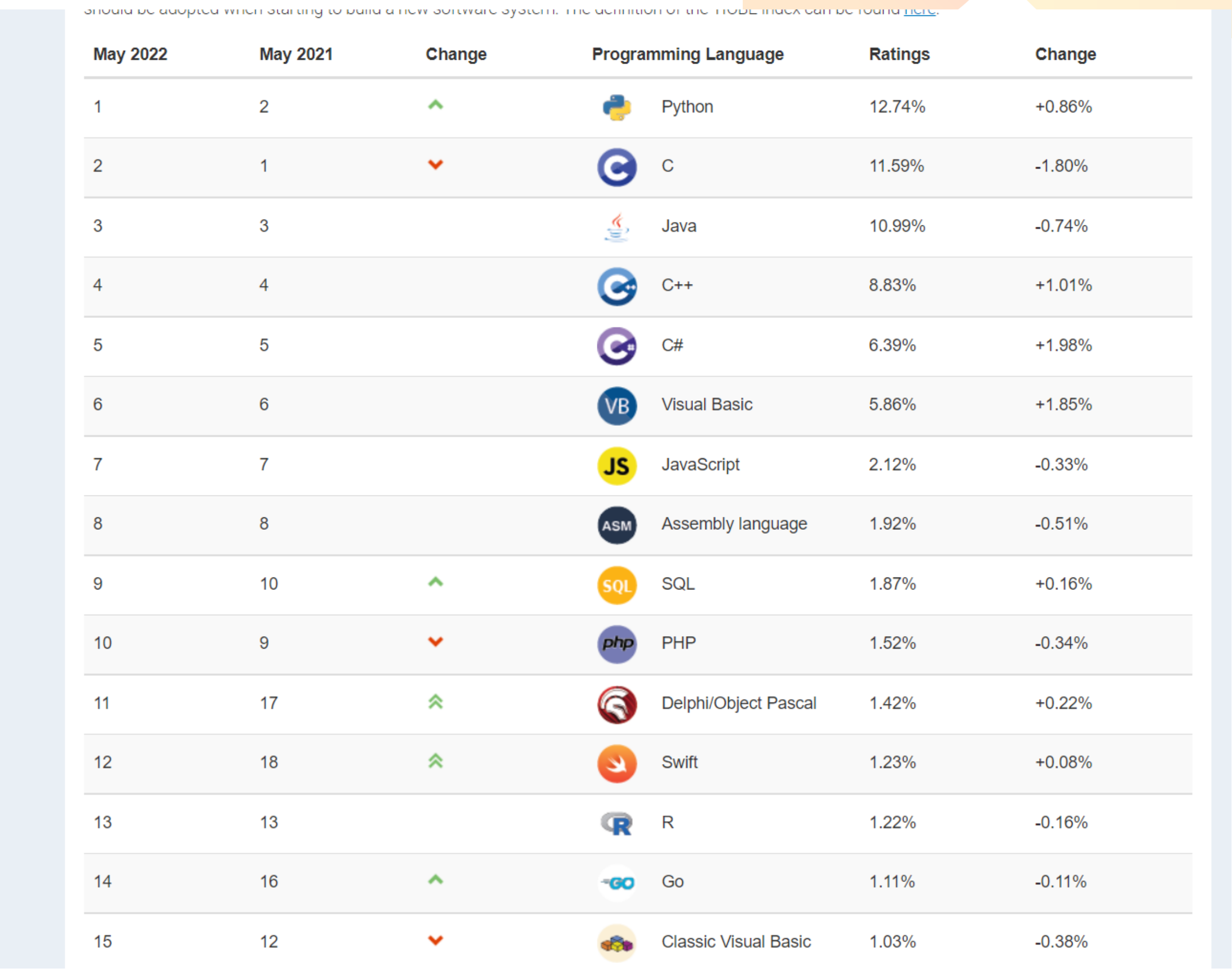


"The main reason why Python is preferred to R is because Python is a real generic programming language with a very large user community," Tiobe's CEO Paul Jensen told ZDNet.



"After having been in the top 20 for about three years, statistical language R dropped out this month. This is quite surprising because the field of statistical programming is still booming, especially thanks to the popularity of data mining and artificial intelligence," Tiobe notes.

- Source: https://www.zdnet.com/google-amp/article/r-vs-python-rs-out-of-top-20-programming-languages-despite-boom-in-statistical-jobs/



should be adopted when starting to build a new software system. The definition of the TIOBE index can be found here.

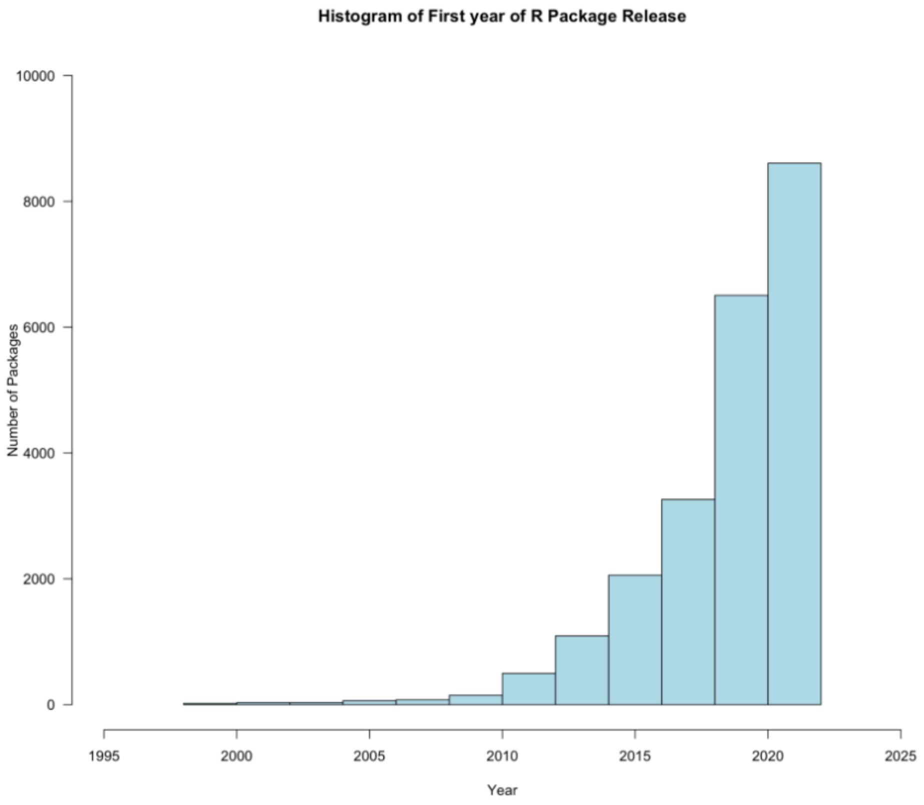
May 2022	May 2021	Change	Programming Language	Ratings	Change
1	2	↑	Python	12.74%	+0.86%
2	1	↓	C	11.59%	-1.80%
3	3		Java	10.99%	-0.74%
4	4		C++	8.83%	+1.01%
5	5		C#	6.39%	+1.98%
6	6		Visual Basic	5.86%	+1.85%
7	7		JavaScript	2.12%	-0.33%
8	8		Assembly language	1.92%	-0.51%
9	10	↑	SQL	1.87%	+0.16%
10	9	↓	PHP	1.52%	-0.34%
11	17	⇈	Delphi/Object Pascal	1.42%	+0.22%
12	18	⇈	Swift	1.23%	+0.08%
13	13		R	1.22%	-0.16%
14	16	↑	Go	1.11%	-0.11%
15	12	↓	Classic Visual Basic	1.03%	-0.38%

[index | TIOBE - The Software Quality Company]

But things are looking better for R...

First release and update dates of R Packages statistics

If years 2016 and 2017 were “data science years” and golden years for R, the decline happened in 2018 but improved back in 2019 and again R is on positive trend. There are many other statistics available to prove this.



Initial release year of R packages

[R package updates and release dates statistics and another rise of R language | by Tomaz Kastrun | Medium]

Agenda

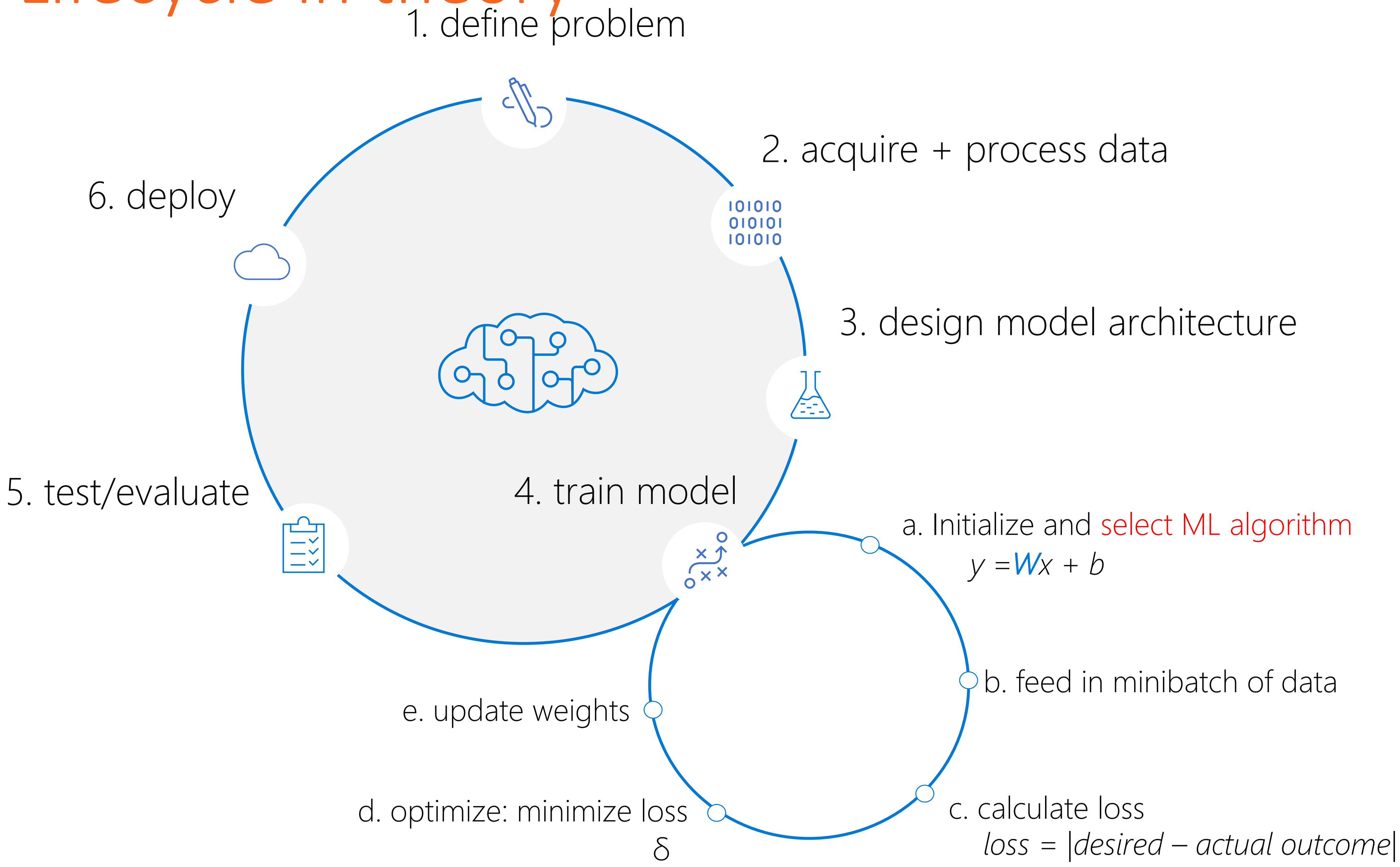
- Azure Analytics family
- Azure Machine Learning and Automated Machine Learning
- How to Start with Azure ML
- Notebooks, VM, Labs
- Demo
  - Automated ML in Action
  - Using Azure ML
  - Python Azure SDK
- Summary

Azure Analytics family

- Azure Databricks Managed, fast, easy and collaborative Apache® Spark™ based analytics platform optimized for Azure
- Azure HD Insight an open-source analytics service that runs Hadoop, Spark, Kafka, and more for big data processing in Azure
- Azure Synapse Analytics brings together enterprise data warehousing and Big Data analytics
- Azure Analysis Services Enterprise grade analytics engine as a service
- Azure Stream analytics Real-time data stream processing from millions of IoT devices
- Machine Learning Service Open and elastic AI development spanning the cloud and the edge
- Cognitive Services API based entreprise solutions
- Azure Datalake Analytics* On-demand pay-per-job analytics service with enterprise-grade security, auditing, and support

* deprecated

Lifecycle in theory



1. define problem

2. acquire + process data

3. design model architecture

4. train model

a. Initialize and select ML algorithm

$y = Wx + b$

b. feed in minibatch of data

c. calculate loss

$loss = |desired - actual\ outcome|$

d. optimize: minimize loss

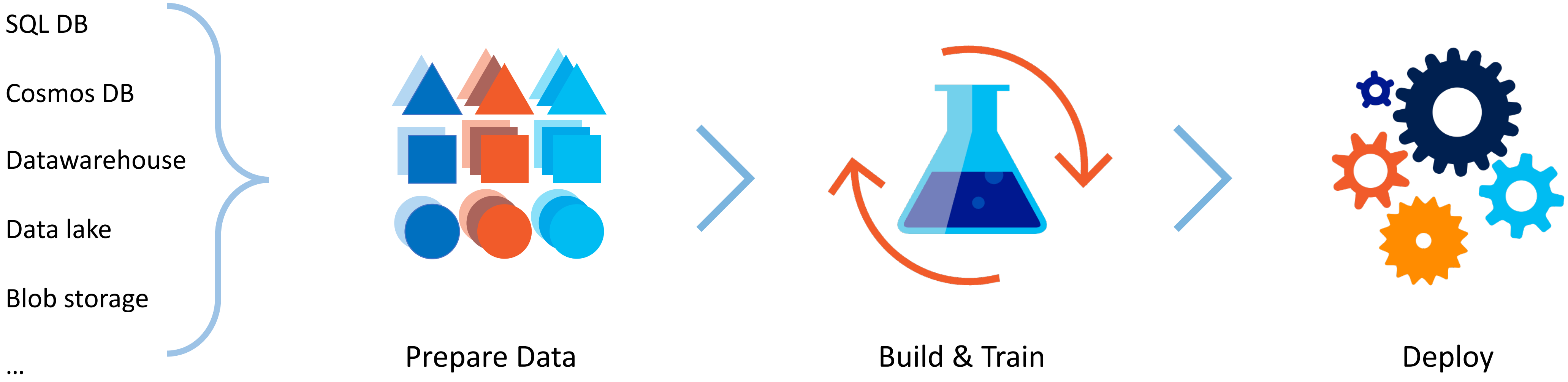
δ

e. update weights

5. test/evaluate

6. deploy

Azure Machine Learning Process



SQL DB

Cosmos DB

Datawarehouse

Data lake

Blob storage

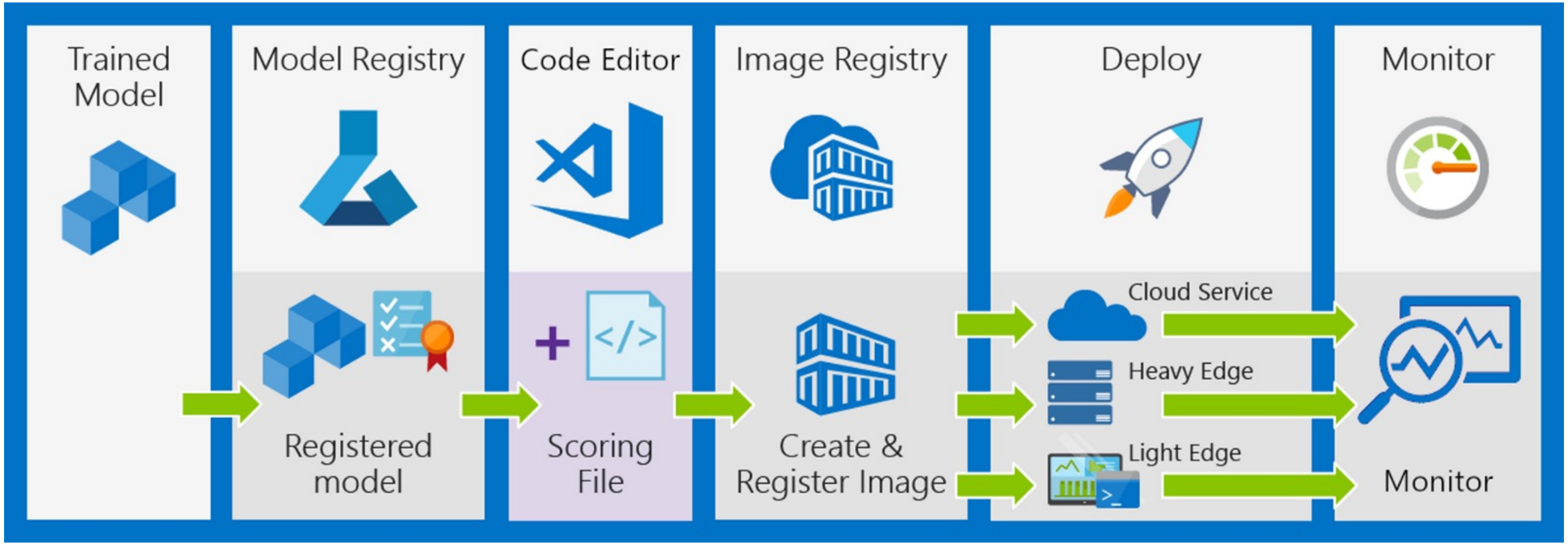
…

Prepare Data

Build & Train

Deploy

Model Lifecycle in practice



What is automated machine learning?

> Automated machine learning (automated ML) picks an algorithm and hyperparameters for you and generates a model ready for deployment. The model can be downloaded to be further customized as well.

Automated ML

Cycle

Data

Data cleaning support

Automated ML currently supports automated data cleaning

Feature

Feature engineering

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

Algorithm

Pick and play

Testing many different algorithms at once.

Tuning

What to leave out

Hyperparameter tuning: what to include what to leave out

Ranking

Ranking

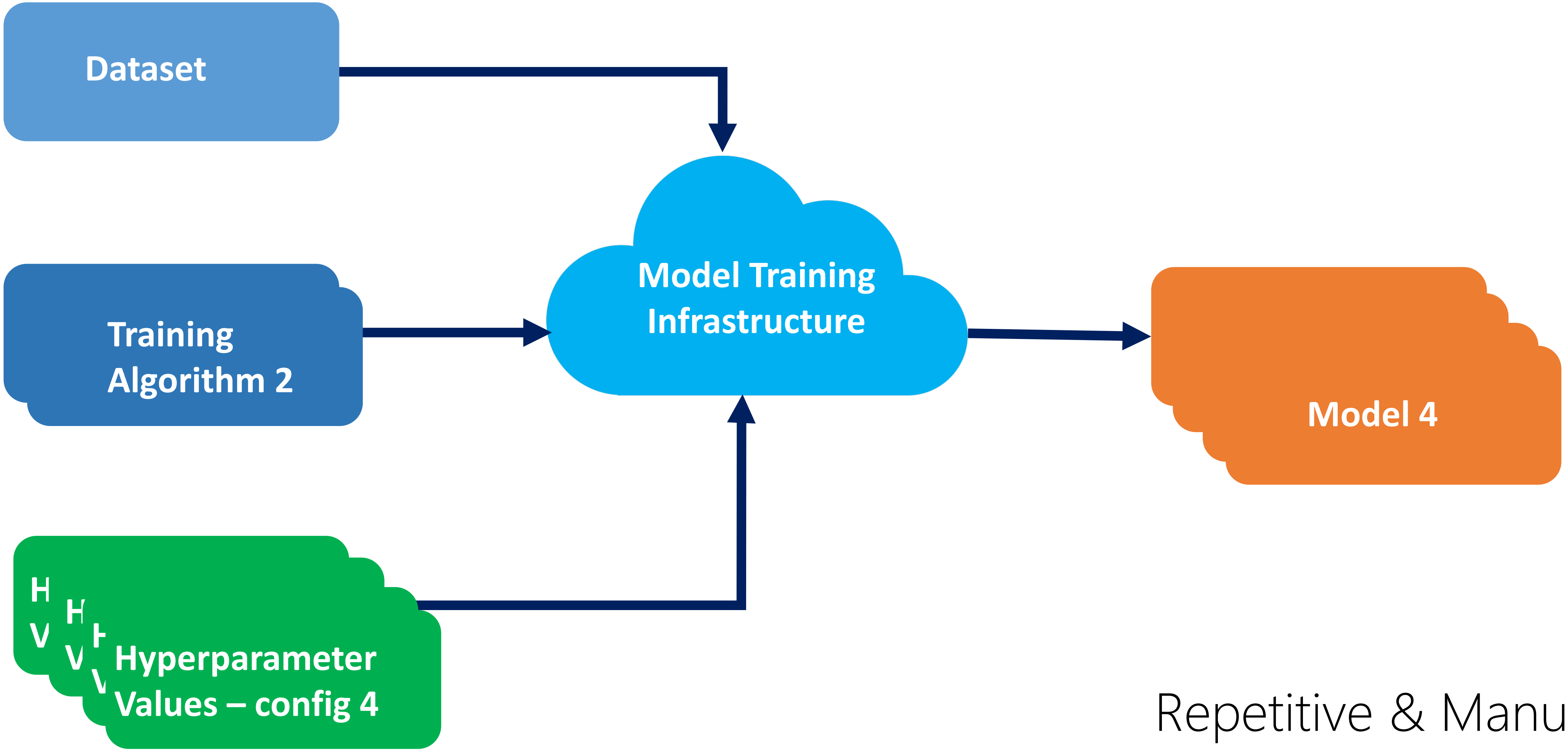
Having an overview of the best performing models based on accuracy & speed.

Explaining

Justification

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

Model Selection & Hyperparameter Tuning



Dataset

Training Algorithm 2

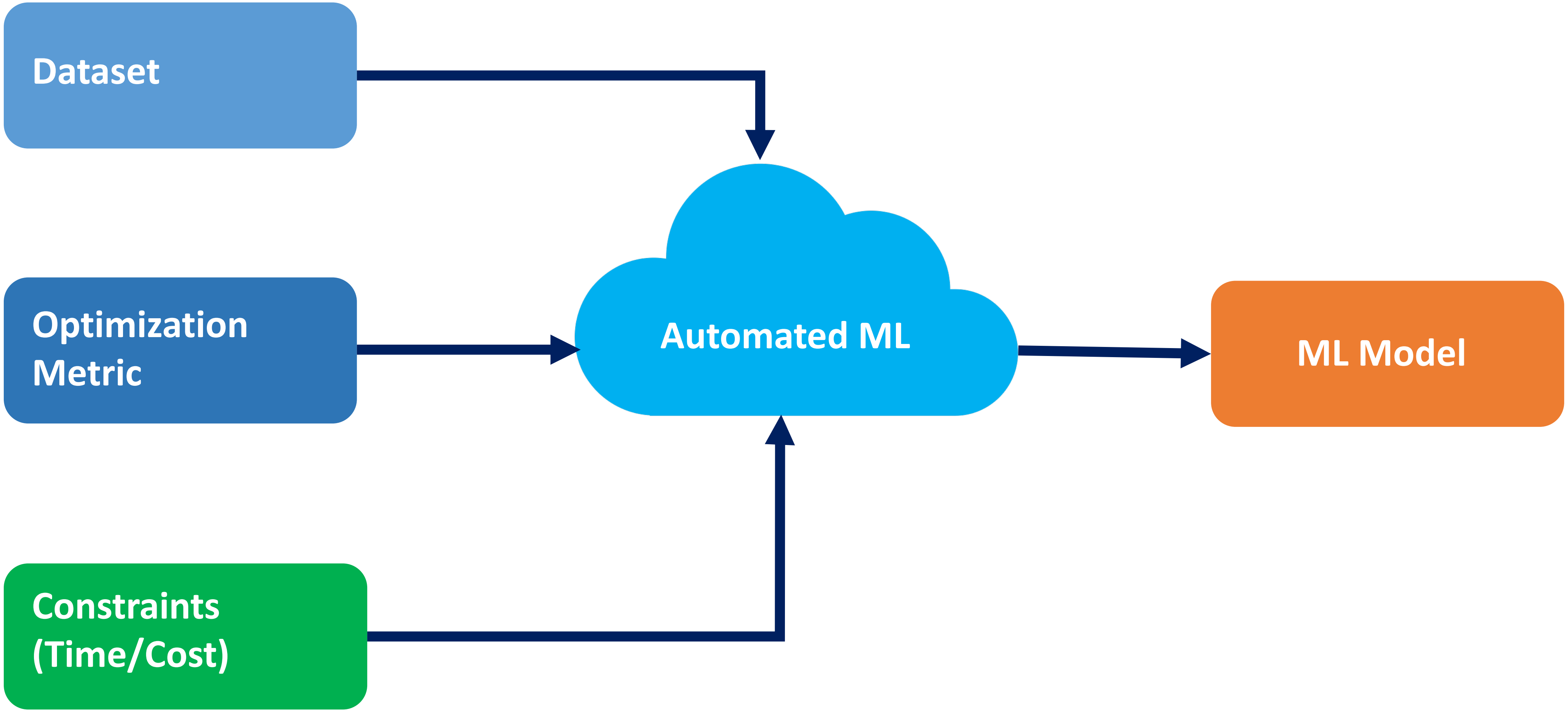
Hyperparameter Values – config 4

Model Training Infrastructure

Model 4

Repetitive & Manual

Introducing Automated Machine Learning



Accessible & Faster

Automated ML Accelerates Model Development

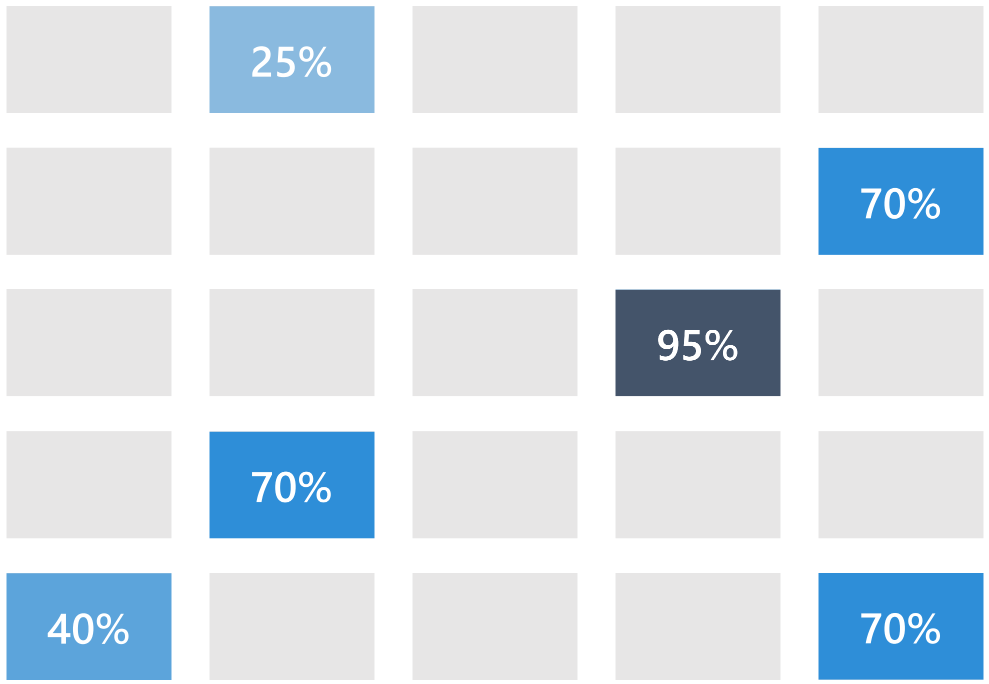
Input

Enter data

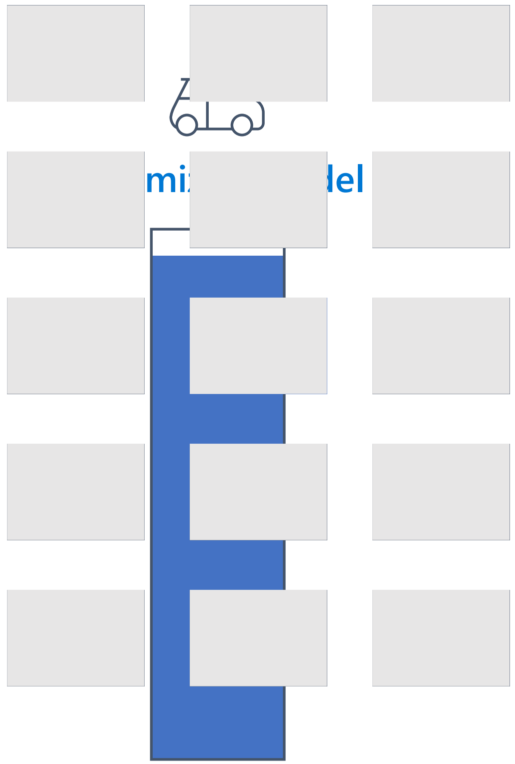
Define goals

Apply constraints

Intelligently test multiple models in parallel



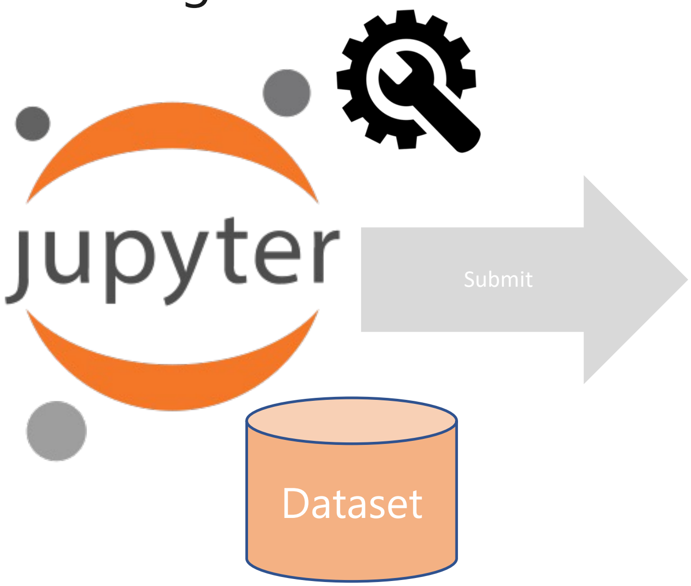
Output



Automated ML – User Experience

It is a „black box“?

Python Script Configuration

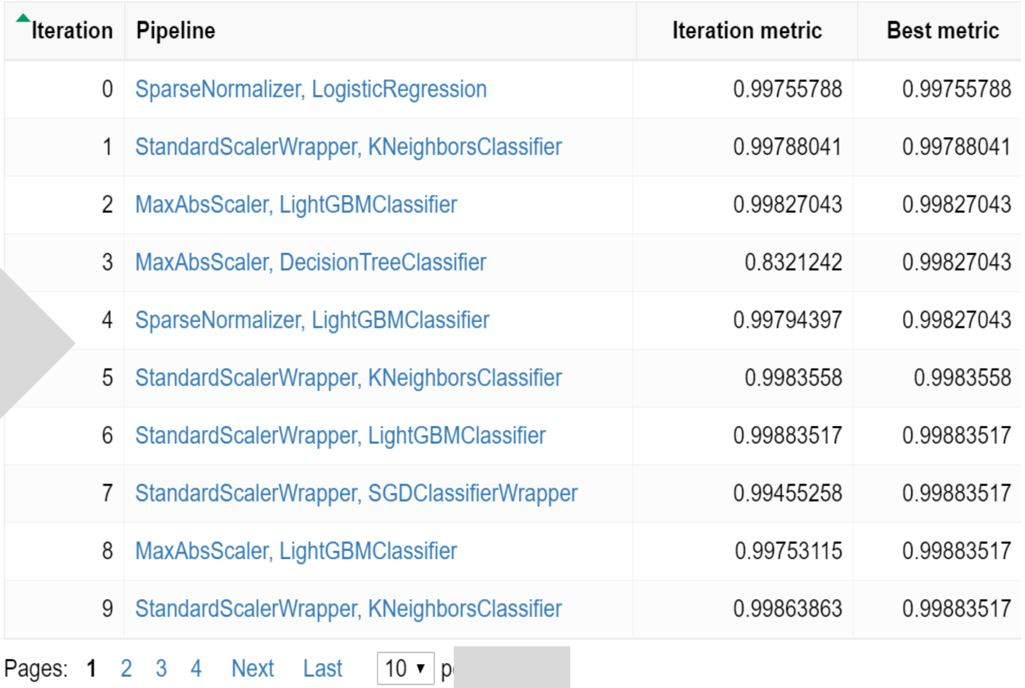


Automated ML

Models & Hyper parameters

Train & Repeat

Models (tuned Local / Cloud)



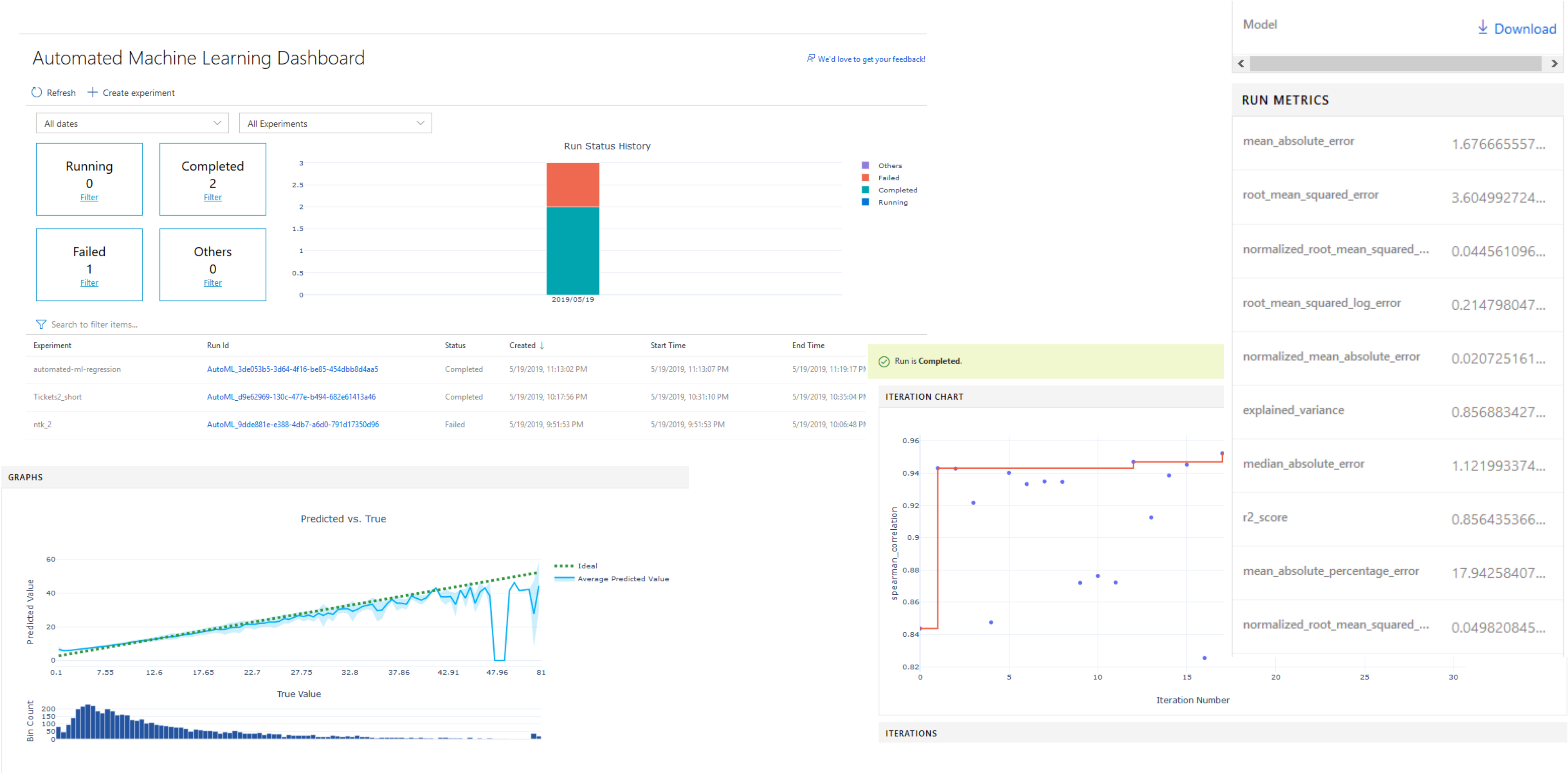
Iteration	Pipeline	Iteration metric	Best metric
0	SparseNormalizer, LogisticRegression	0.99755788	0.99755788
1	StandardScalerWrapper, KNeighborsClassifier	0.99788041	0.99788041
2	MaxAbsScaler, LightGBMClassifier	0.99827043	0.99827043
3	MaxAbsScaler, DecisionTreeClassifier	0.8321242	0.99827043
4	SparseNormalizer, LightGBMClassifier	0.99794397	0.99827043
5	StandardScalerWrapper, KNeighborsClassifier	0.9983558	0.9983558
6	StandardScalerWrapper, LightGBMClassifier	0.99883517	0.99883517
7	StandardScalerWrapper, SGDClassifierWrapper	0.99455258	0.99883517
8	MaxAbsScaler, LightGBMClassifier	0.99753115	0.99883517
9	StandardScalerWrapper, KNeighborsClassifier	0.99863863	0.99883517

Pages: 1 2 3 4 Next Last 10 ▾ p

Solution

High Quality Machine Learning Model

Automated Machine Learning Dashboard



Automated Machine Learning Dashboard

Experiment	Run Id	Status	Created ↓	Start Time	End Time
automated-ml-regression	AutoML_3de053b5-3d64-4f16-be85-454dbb8d4aa5	Completed	5/19/2019, 11:13:02 PM	5/19/2019, 11:13:07 PM	5/19/2019, 11:19:17 PM
Tickets2_short	AutoML_d9e62969-130c-477e-b494-682e61413a46	Completed	5/19/2019, 10:17:56 PM	5/19/2019, 10:31:10 PM	5/19/2019, 10:35:04 PM
ntk_2	AutoML_9dde881e-e388-4db7-a6d0-791d17350d96	Failed	5/19/2019, 9:51:53 PM	5/19/2019, 9:51:53 PM	5/19/2019, 10:06:48 PM

GRAPHS

Model

RUN METRICS

mean_absolute_error	1.676665557...
root_mean_squared_error	3.604992724...
normalized_root_mean_squared_...	0.044561096...
root_mean_squared_log_error	0.214798047...
normalized_mean_absolute_error	0.020725161...
explained_variance	0.856883427...
median_absolute_error	1.121993374...
r2_score	0.856435366...
mean_absolute_percentage_error	17.94258407...
normalized_root_mean_squared_...	0.049820845...

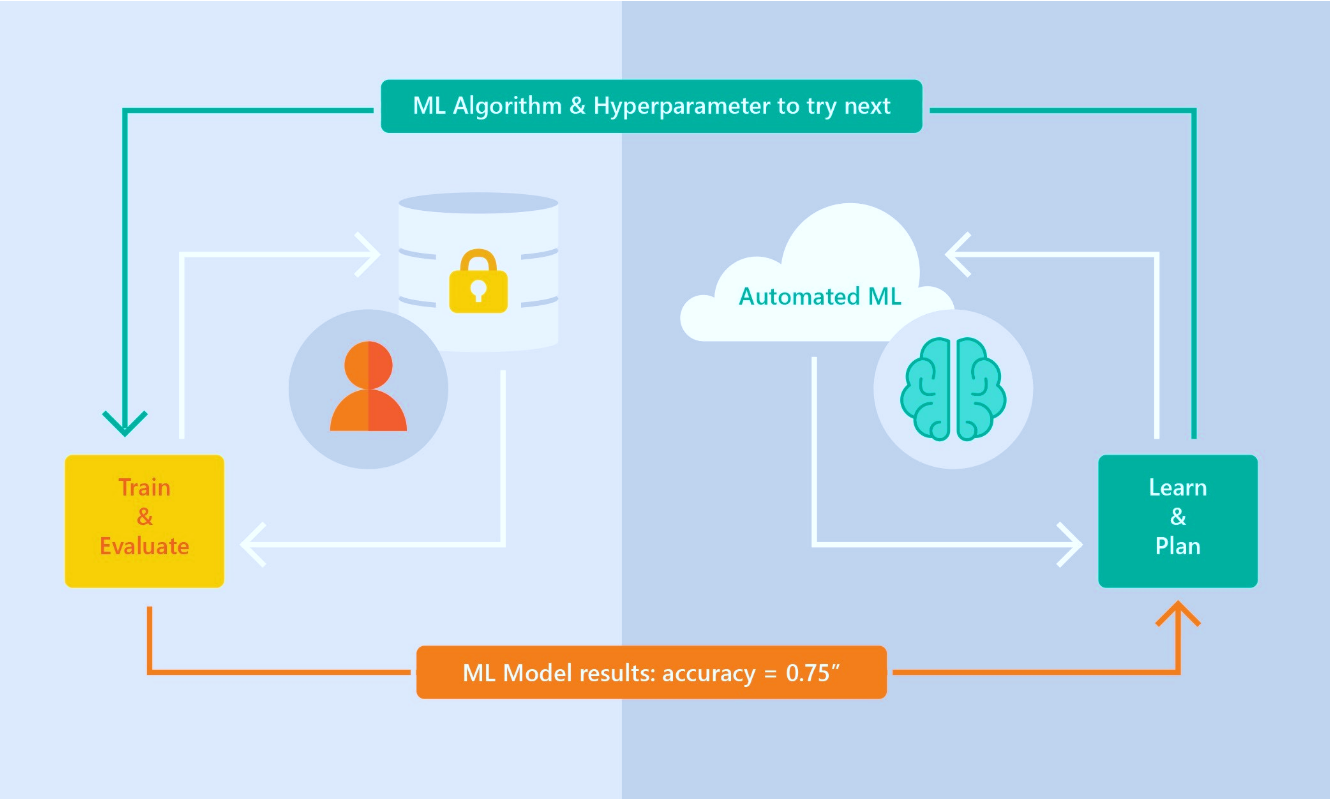
Run is Completed.

ITERATION CHART

ITERATIONS

Automated ML Capabilities

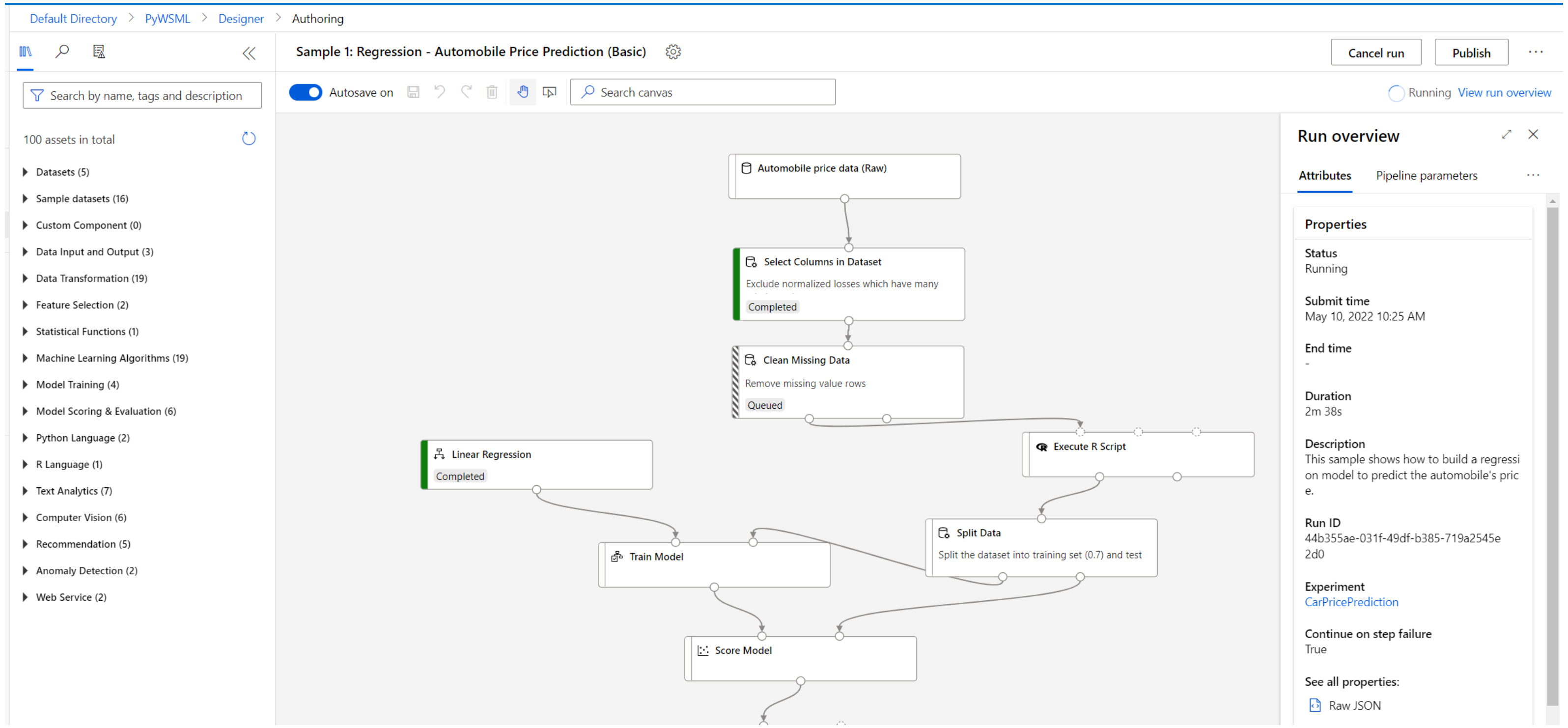
- ML Scenarios: Classification & Regression, Forecasting*
- Integration: Azure Machine Learning, Azure Notebooks, Jupyter Notebooks
- Data Type: Numeric, Text
- Languages: Python SDK for deployment and hosting for inference
- Training Compute: Local Machine, Remote Azure DSVM (Linux), Azure Compute, Azure Databricks*
- Transparency: View run history, model metrics
- Scale: Faster model training using multiple cores and parallel experiments



ML Algorithm & Hyperparameter to try next
Automated ML
Train & Evaluate
Learn & Plan
ML Model results: accuracy = 0.75”

Visual Interface

- Drag-n-Drop building ML models
- No limit to size or compute capacity for model training
- Powerful R and Python support
- One-Click deploy web service
- Rich support
- Community engagement



Properties

Status
Running

Submit time
May 10, 2022 10:25 AM

End time
-

Duration
2m 38s

Description
This sample shows how to build a regression model to predict the automobile's price.

Run ID
44b355ae-031f-49df-b385-719a2545e2d0

Experiment
CarPricePrediction

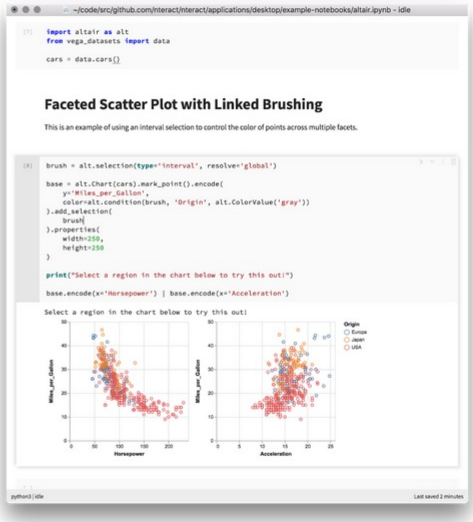
Continue on step failure
True

See all properties:

Notebooks

- Notebooks offer flexibility to users
- Easier managibility
- Repetability
- How Netflix built analytics around Notebooks -> https://medium.com/netflix-techblog/notebook-innovation-591ee3221233

Jupyter Notebooks



```
import altair as alt
from vega_datasets import data

cars = data.cars()
```

Faceted Scatter Plot with Linked Brushing

This is an example of using an interval selection to control the color of points across multiple facets.

```
brush = alt.selection(type='interval', resolve='global')

base = alt.Chart(cars).mark_point().encode(
    y='Miles_per_Gallon',
    color=alt.condition(brush, 'Origin', alt.ColorValue('gray'))
).add_selection(
    brush
).properties(
    width=250,
    height=250
)

print("Select a region in the chart below to try this out!")

base.encode(x='Horsepower') | base.encode(x='Acceleration')
```

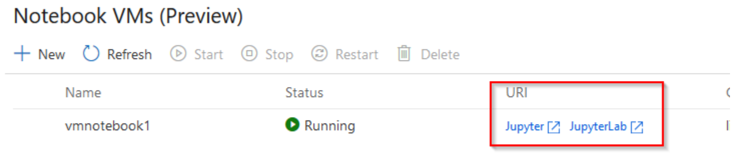
```
Select a region in the chart below to try this out!
```

Project Jupyter began in 2014 with a goal of creating a consistent set of open-source tools for scientific research, reproducible workflows, computational narratives, and data analytics. Those tools translated well to industry, and today Jupyter notebooks have become an essential part of the data scientist toolkit. To give you a sense of its impact, Jupyter was awarded the 2017 ACM Software Systems Award—a prestigious honor it shares with Java, Unix, and the Web.

Notebook VMs and Jupyter Lab

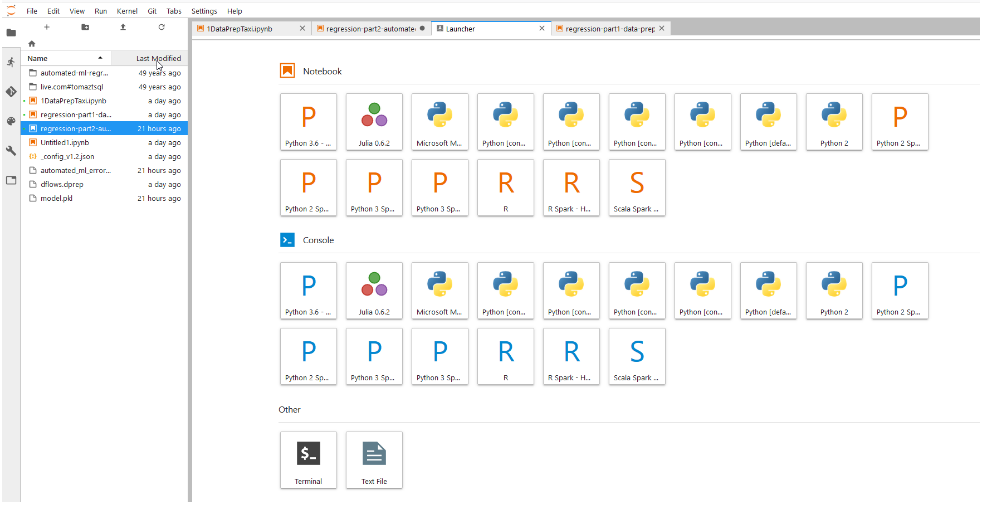
- Dedicated Virtual Machine for Machine Learning
- Easy setup the configurations and extentions
- Continous delivery and model maintanance (DevOps and pipelines)
- All the usuals: disaster recovery, update management, backup, Auto-shutdown, change tracking
- Serves as input or output | source or destination
- Git integration
- Jupyter Notebook integration
- Jupyter Labs (!!!!) (notebook, Console, terminal,…)
- 20+ pre-prepared Python/Scala/R/Julia environments
- Awesome Jupyter Labs extensions

Data Science Virtual Machine



Notebook VMs (Preview)

Name	Status	URI
vmnotebook1	Running	Jupyter JupyterLab



Name	Last Modified
automated-ml-regr...	49 years ago
live.com#tomaztsql	49 years ago
1DataPrepTaxi.ipynb	a day ago
regression-part1-da...	a day ago
regression-part2-au...	21 hours ago
Untitled1.ipynb	a day ago
_config_v1.2.json	a day ago
automated_ml_error...	21 hours ago
dflows.dprep	a day ago
model.pkl	21 hours ago

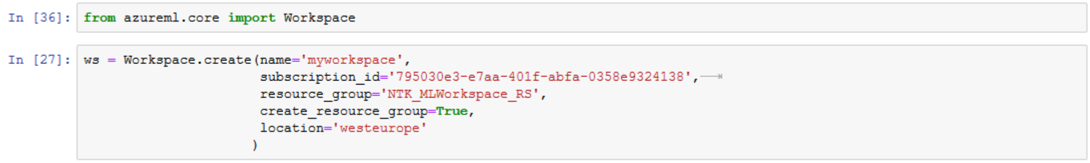
Azure Machine Learning SDK for Python

- Python SDK + Azure ML = BFF
- SDK to build and run machine learning workflows using Azure Machine Learning Services
- Explore, prepare and manage the lifecycle of your datasets used in machine learning experiments
- Manage cloud resources for monitoring, logging, and organizing your machine learning experiments
- Train models either locally or by using cloud resources, including GPU-accelerated model training.
- Use automated machine learning, which accepts configuration parameters and training data. It automatically iterates through algorithms and hyperparameter settings to find the best model for running predictions.
- Deploy web services to convert your trained models into RESTful services that can be consumed in any application.

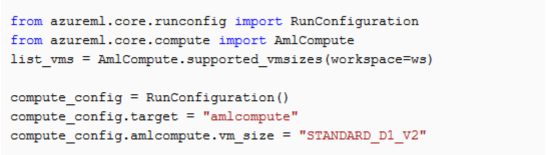
Azure Machine Learning SDK for Python #2

Most important Namespaces:

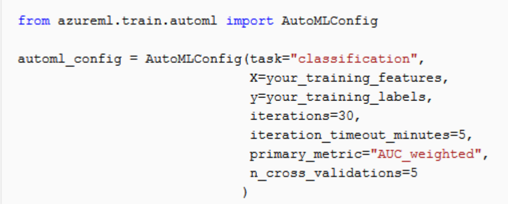
- Workspace (azureml.core.workspace.Workspace)
- Experiment (azureml.core.experiment.Experiment)
- Run (azureml.core.experiment.Run)
- Model (azureml.core.experiment.Model)
- Dataprep (azureml.dataprep.dataflow)
- For Management
  - ComputeTarget (azureml.core.compute.ComputeTarget)
  - RunConfiguration (azureml.core.compute.RunConfiguration)
  - ScriptRunConfig (azureml.core.compute.ScriptRunConfig)
  - Logging (azureml.core.logging.Logging)
- AutoMLConfig (azureml.train.automl.automlconfig.AutoMLConfig)
- Webservice (azureml.core.webservice.Webservice)



```
In [36]: from azureml.core import Workspace

In [27]: ws = Workspace.create(name='myworkspace',
                               subscription_id='795030e3-e7aa-401f-abfa-0358e9324138',
                               resource_group='NTK_MLWorkspace_RS',
                               create_resource_group=True,
                               location='westeurope'
                              )
```

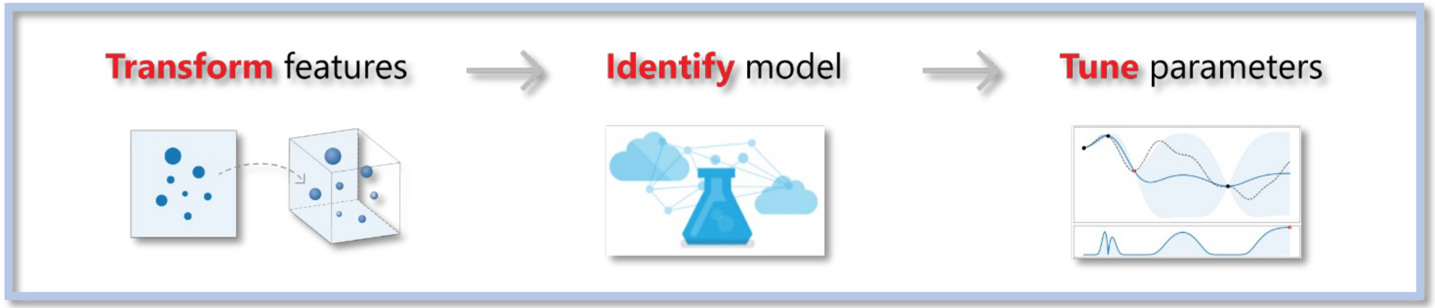


```
from azureml.core.runconfig import RunConfiguration
from azureml.core.compute import AmlCompute
list_vms = AmlCompute.supported_vmsizes(workspace=ws)

compute_config = RunConfiguration()
compute_config.target = "amlcompute"
compute_config.amlcompute.vm_size = "STANDARD_D1_V2"
```



```
from azureml.train.automl import AutoMLConfig

automl_config = AutoMLConfig(task="classification",
                             X=your_training_features,
                             y=your_training_labels,
                             iterations=30,
                             iteration_timeout_minutes=5,
                             primary_metric="AUC_weighted",
                             n_cross_validations=5
                            )
```

Feature Engineering



Transform features → **Identify** model → **Tune** parameters

Dropping high cardinality or no variance features

- Features with no useful information are dropped from training and validation sets. These include features with all values missing, same value across all rows or with extremely high cardinality (e.g., hashes, IDs or GUIDs).

Missing value imputation

- For categorical features, missing values are imputed with most frequent value. For numerical features, missing values are imputed with average of values in the column

Generating additional features

- For DateTime features: Year, Month, Day, of week/ of year, Quarter, Week of the year, Hour, Minute, Second.
- For Text features: Term frequency based on word unigram, bi-grams and Char tri-char, Count vectorizer

Transformations and encodings

- Numeric features with very few unique values are transformed into categorical features. Depending on cardinality of categorical features label encoding or (hashing) one-hot encoding is performed

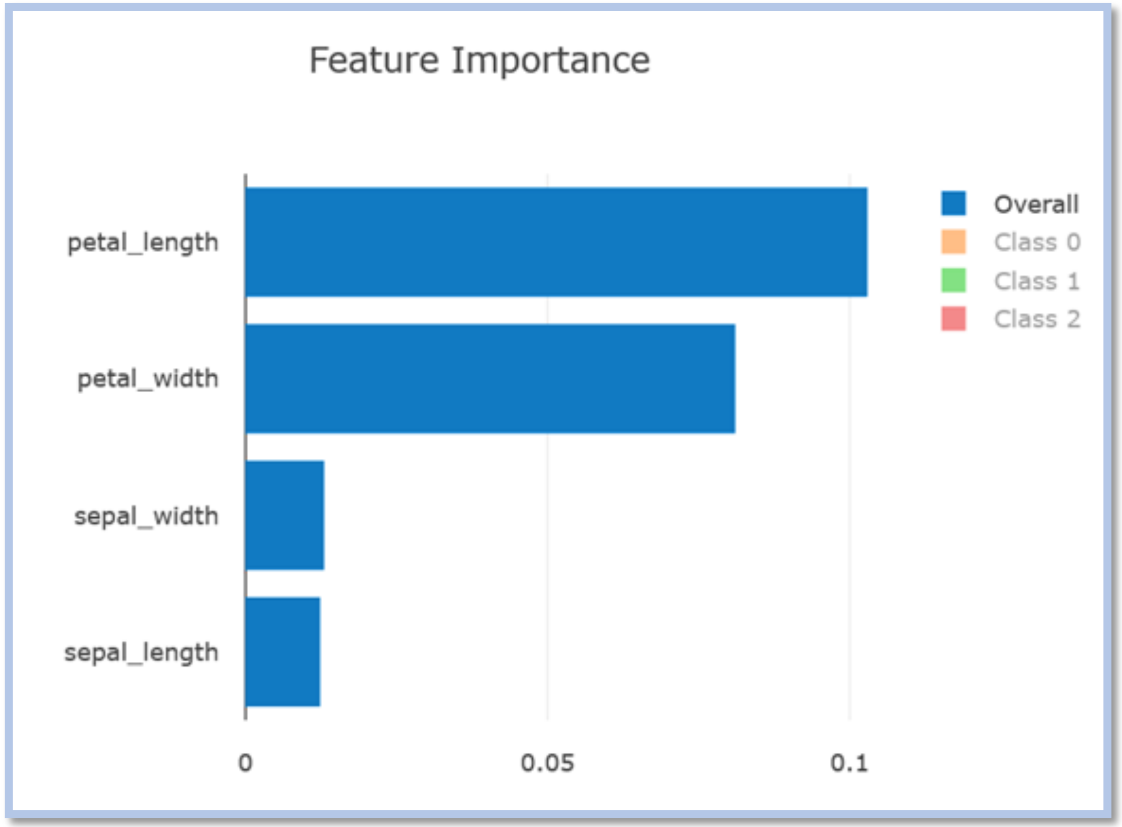
Model Explainability and Interpretability

GA:

- Feature importance as part of training
- Simple UX for feature importance for a selected iteration
- Local feature importance for a given sample

Post GA:

- Importance of Raw data columns
- Accuracy and performance improvements



Time Series Support

- Grain Index Featurization & Grouping
- Missing row imputation, including target column
- Improved time index featurization
- Guidance on time-series specific train/validation/test split
- Drop Column
- Lagging features
- Time aggregations, sliding window features

Model deployment, frameworks and environments

- More than 20 predefined AzureML environments (GPU, CUDA, Ubuntu, ...)
- Creating custom environments
- Use different model frameworks (H20, ONNX, PySpark, PyTorch, TensorFlow, TFKeras,...)
- Model versioning and artifacts
- Model endpoints and consumption

Key take-aways

- AutoML new generation of Machine Learning
- Let algoritm select the best strategy to get ML model
- Azure Machine Learning one of the best ML platform
- Build your enterprise around Notebooks (!)
- Use Azure Python SDK to build your enterprise ML Experience

Thanks!

http://tomaztsql.wordpress.com

tomaz.kastrun@gmail.com

@tomaz_tsql

/in/tomaztsql

http://github.com/tomaztk

https://mvp.microsoft.com/PublicProfile/5002196

SQL Day

14 edycja konferencji SQLDay

9-11 maja 2022, WROCŁAW + ONLINE

Data Community

partner złoty

Future Processing

Infinite DATA

KPMG

partner srebrny

Objectivity

summ-it

UBS

dbWatch DATABASE CONTROL

partner brązowy

elitmind think bright