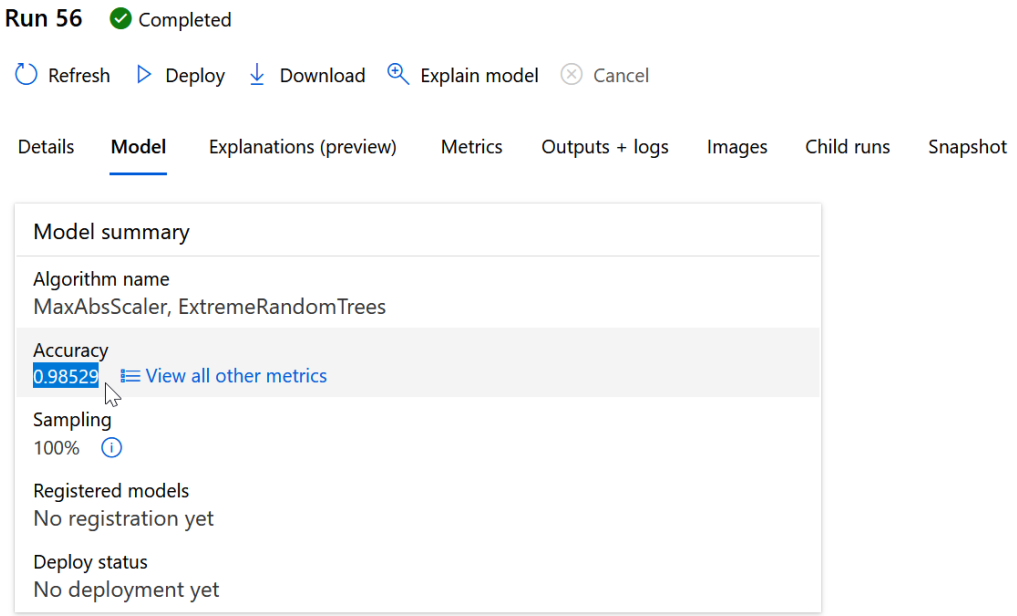
AZEO
talents & technology


# Data Scientist *versus* Automated ML, who's best ?

Automated ML GUI *versus* SDK, who's best ?

*In **less than one hour** and for **less than one dollar of Azure consumption**, I obtained an **accuracy of 0.98529** with an algorithm I didn't even know the name of, while the salary of a Data Scientist is $90k (*) a year minimum!*




*(*) No idea if this value is true...*

AZEO
talents & technology

# Agenda

The common tasks of Data Scientists

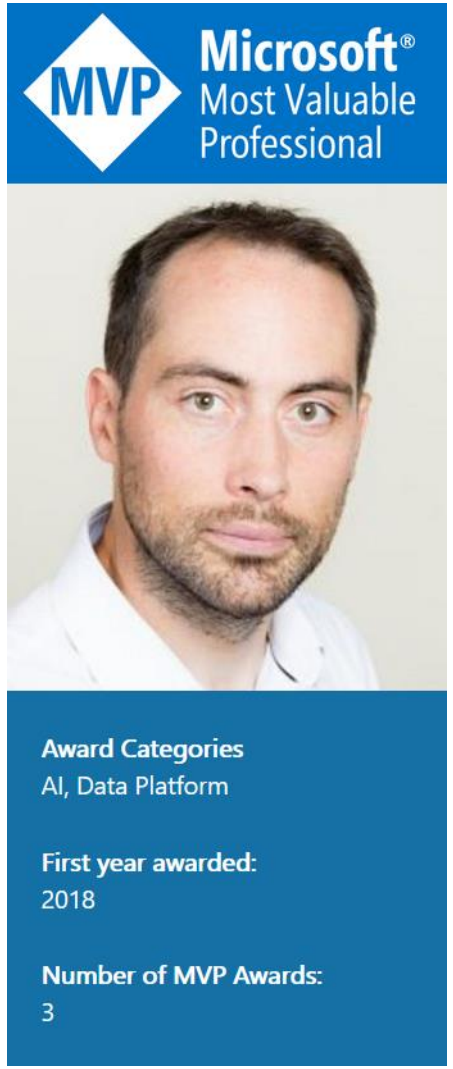
Definition of Automated Machine Learning

1st round : interface under Azure ML studio (new portal)

2nd round: SDK azureml.core

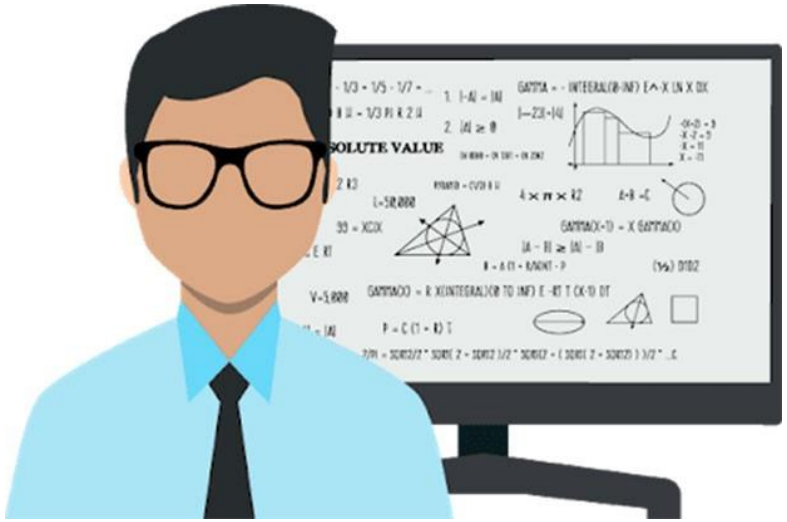
Result of the game

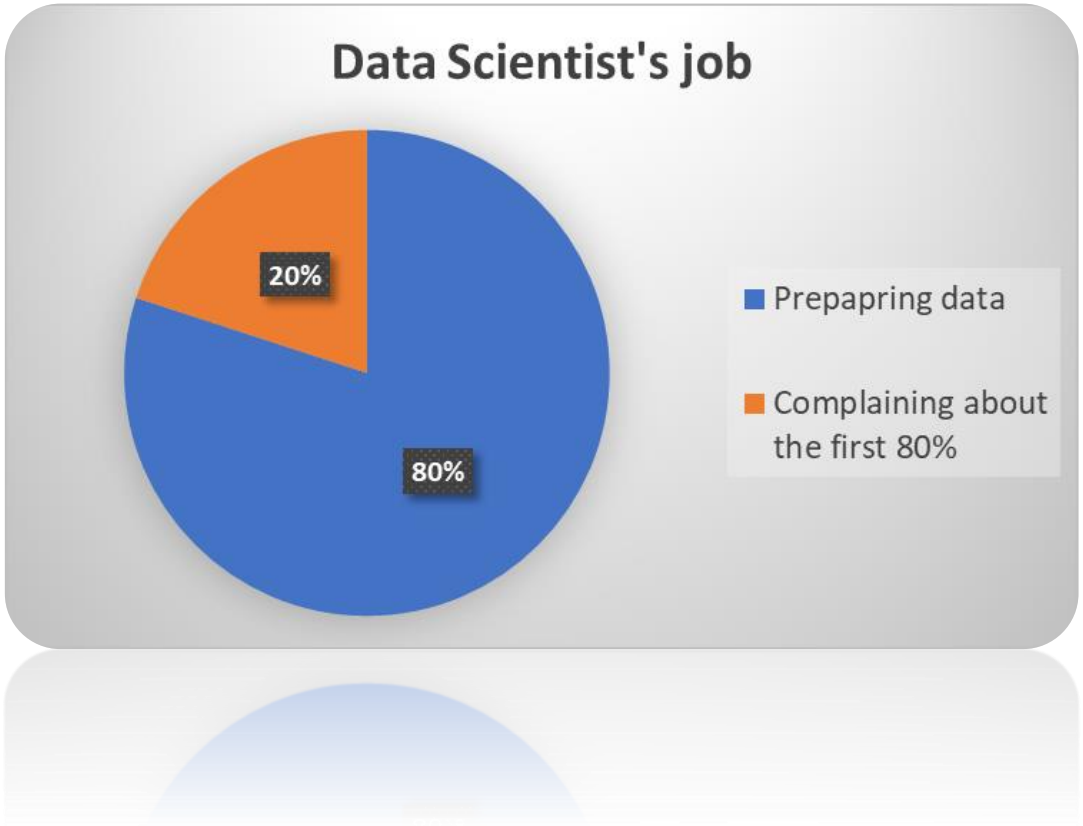
Datasets & notebooks are available on :
https://github.com/methodidacte/azureml




Paul PETON – Tech Lead Data & AI

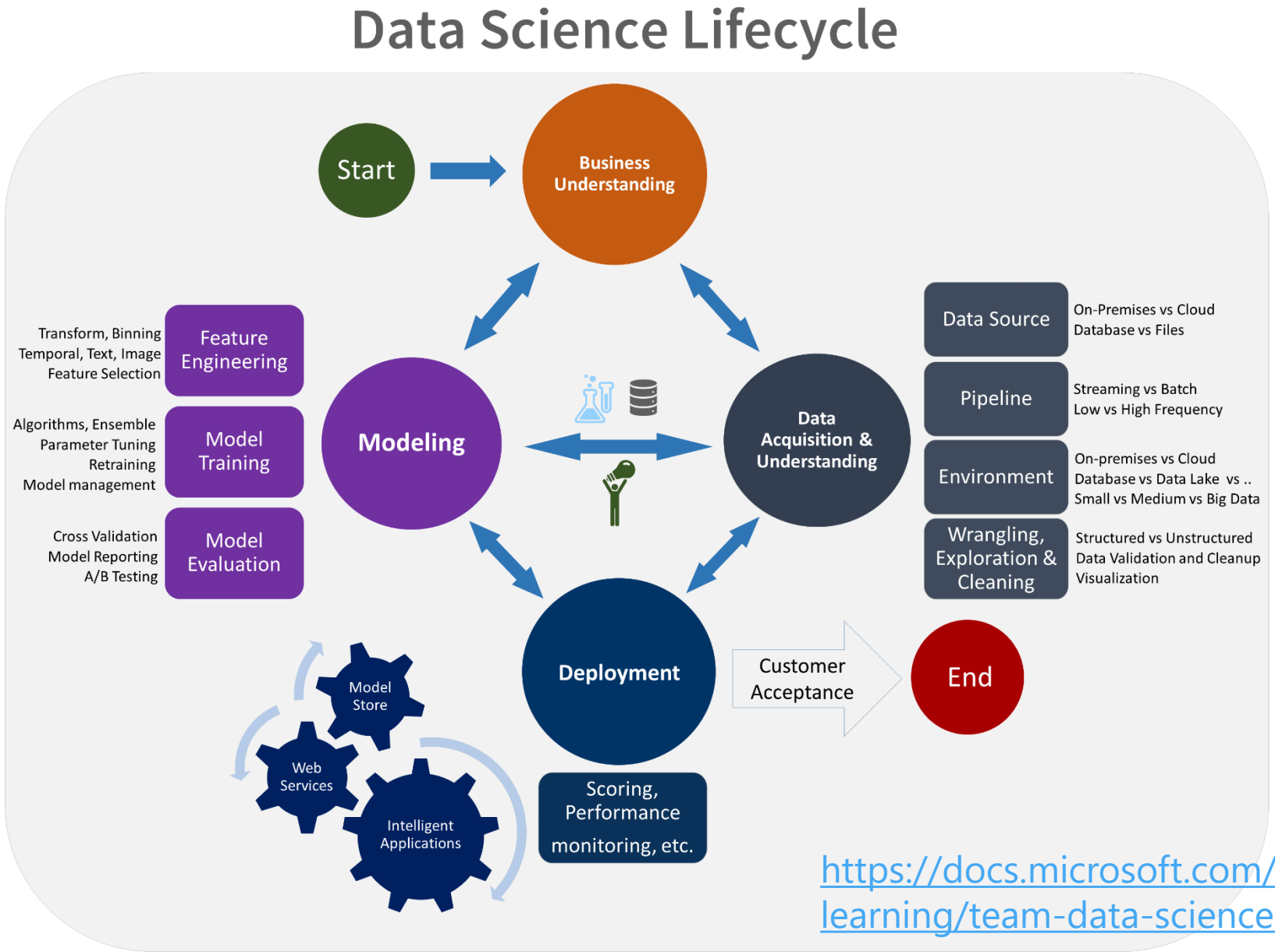
# What is the Pareto's law for Data Scientists ?






Advisory : no data scientist will be harmed during this talk.

# The team data science process




Choose the right model : the most uncertain task

Dataprep : the longest step

https://docs.microsoft.com/fr-fr/azure/machine-learning/team-data-science-process/overview

# Data Scientist whole job

Understand the business need and reformulate it into Data Science process

Carry out data acquisition from one or more sources

Explore the data

Prepare the data :

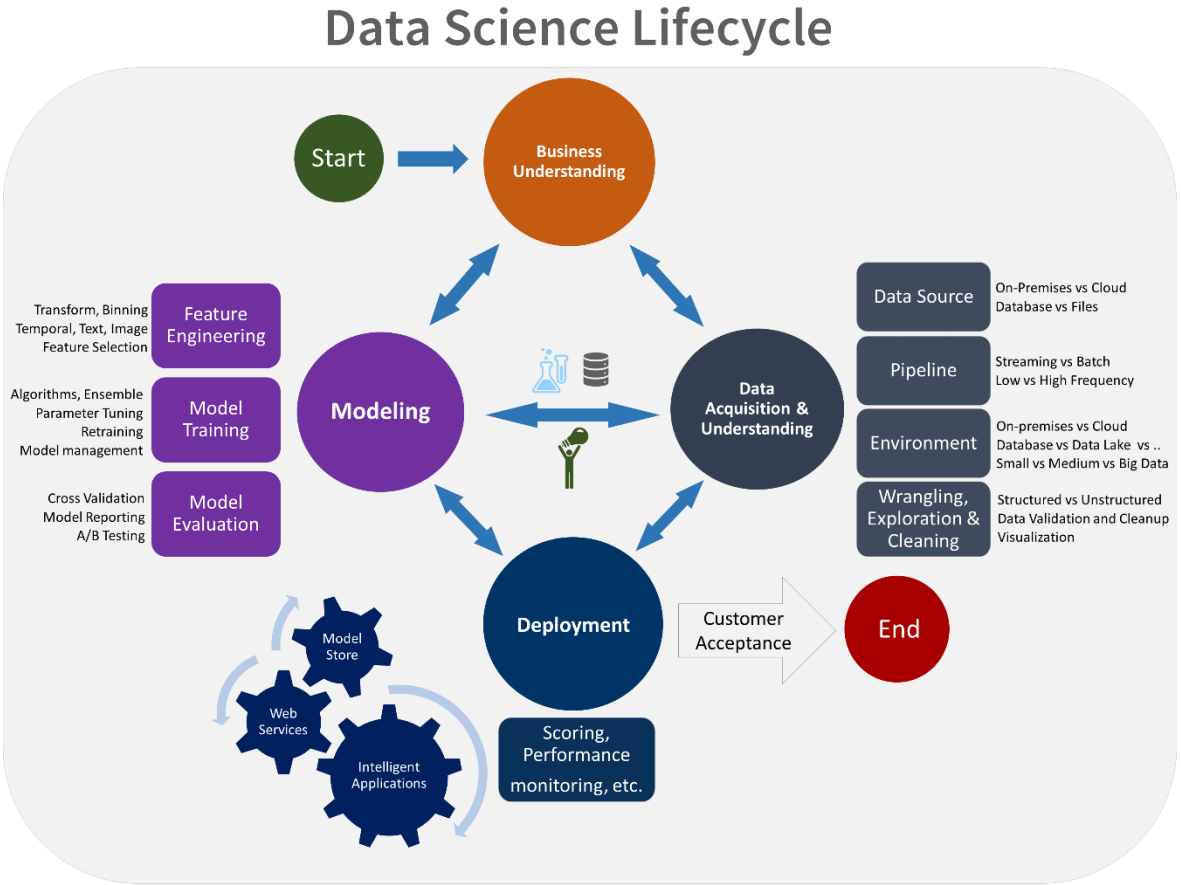
- Clean up missing values or outliers
- Reduce the dimension
- Select the best features

Training several models:

- Trying different approaches (linear or non-linear)
- Select and calculate evaluation metrics
- Identify the best hyperparameters ("*gridsearch*")

Interpret the results obtained

Deploy the models (or help the Data Engineers to do so !)



AZEO
talents & technology

# Automated Machine Learning : definition

## Automated ML is a "brute force" approach to testing many algorithms and keeping the best according to the chosen evaluation metric.

“Theoretical" advantages :

Implementing Machine Learning solutions without having an in-depth knowledge of programming.

Save time and resources.

Leverage best practices in data science.

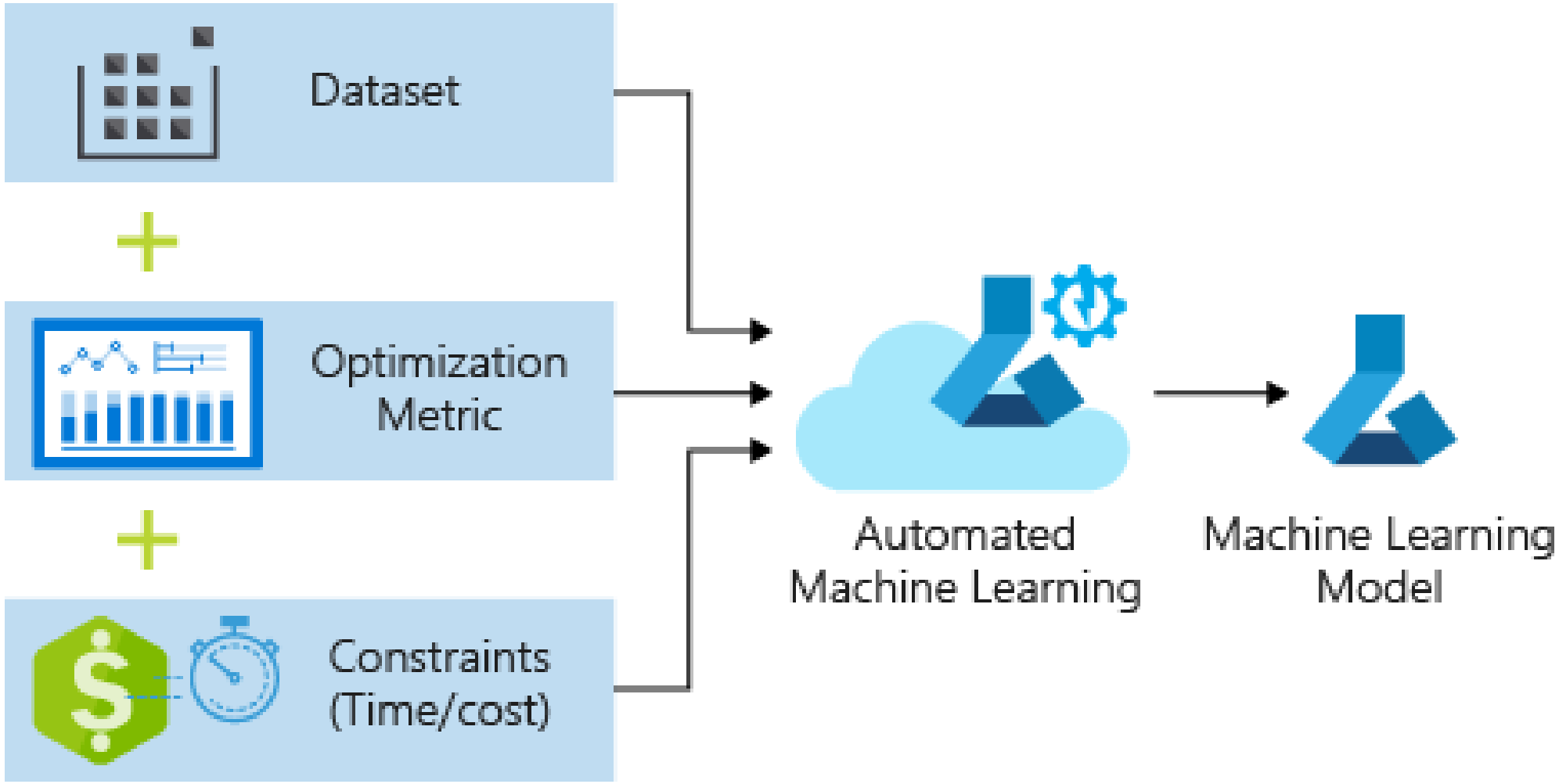
Provide agile problem solving.

Machine Learning benefits from ~~distributed~~ parallel architectures.

Different algorithms can then be executed in parallel, taking advantage of the different nodes in a cluster.

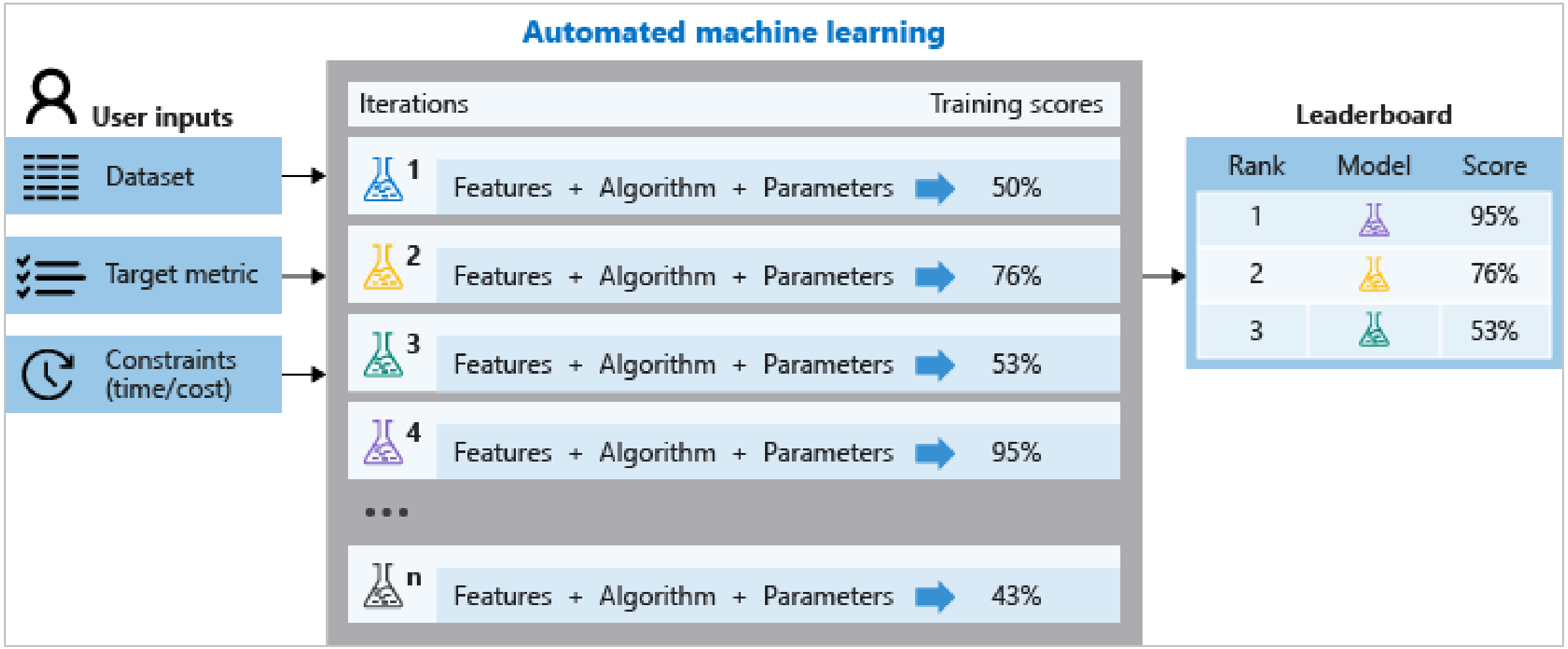
AZEO
talents & technology

# Automated ML : how it works ? (overview)

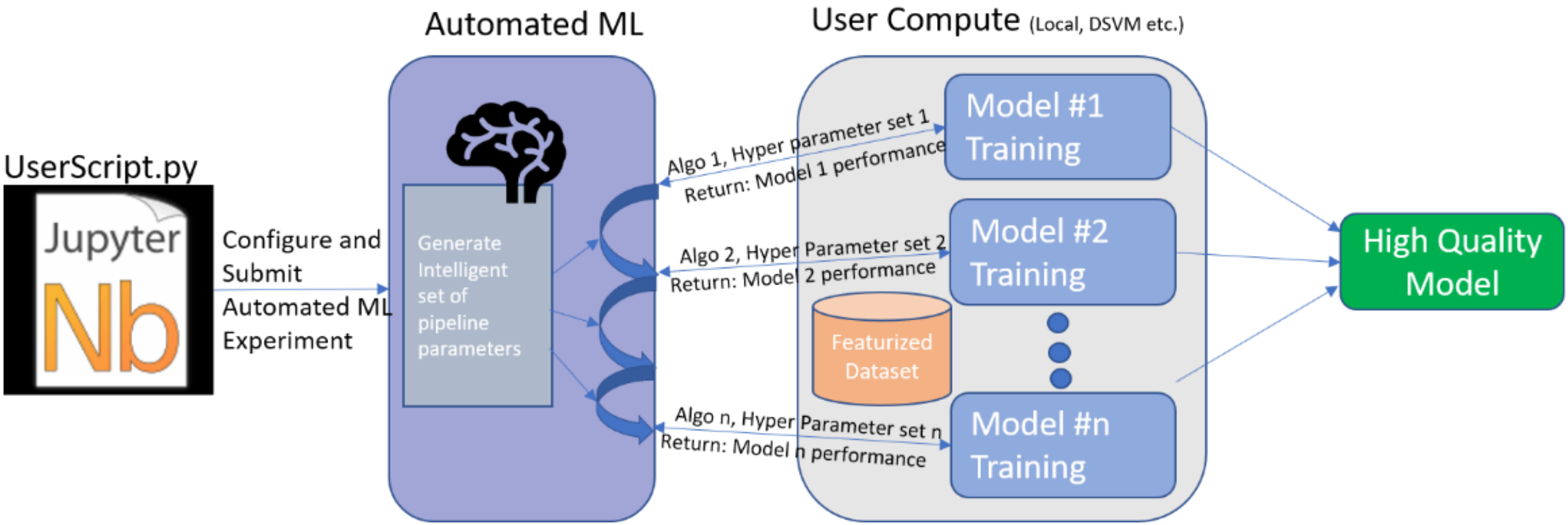


AZEO
talents & technology

# Run many different algorithms in parallel and rank on metrics



# Automated ML : how it works ? (detailled view)



AZEO
talents & technology


PREMIUM licence

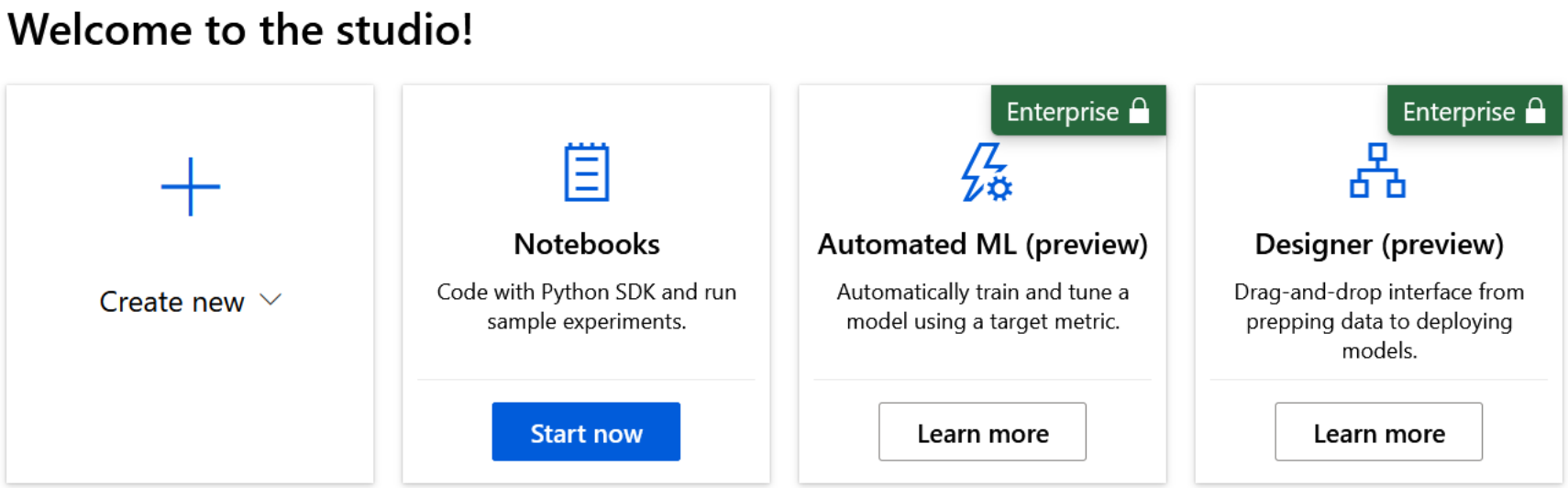
Azure ML SDK Version: 1.10.0

# Automated Machine Learning on Microsoft

AZEO
talents & technology

# Two approaches in Azure Machine Learning

« No code / low code » : GUI but need Enterprise licence




« Code first » : package azureml-sdk (visible code but no contribution possible) and Standard license for storage, monitoring and deployment features.

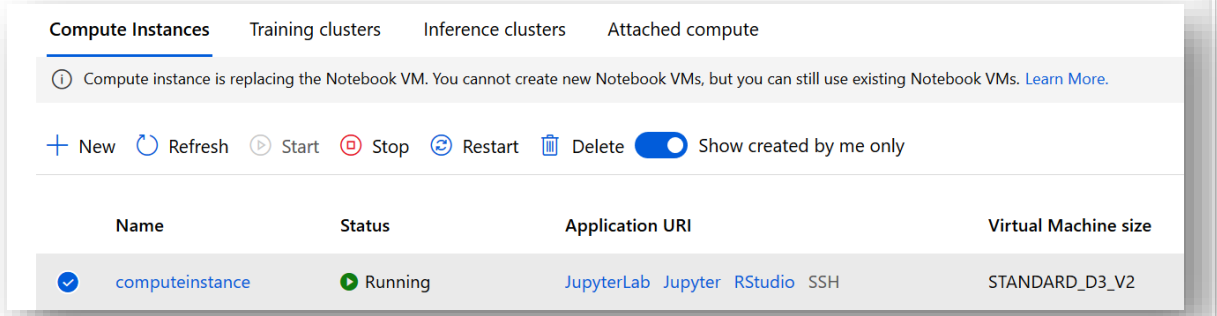
https://pypi.org/project/azureml-sdk/

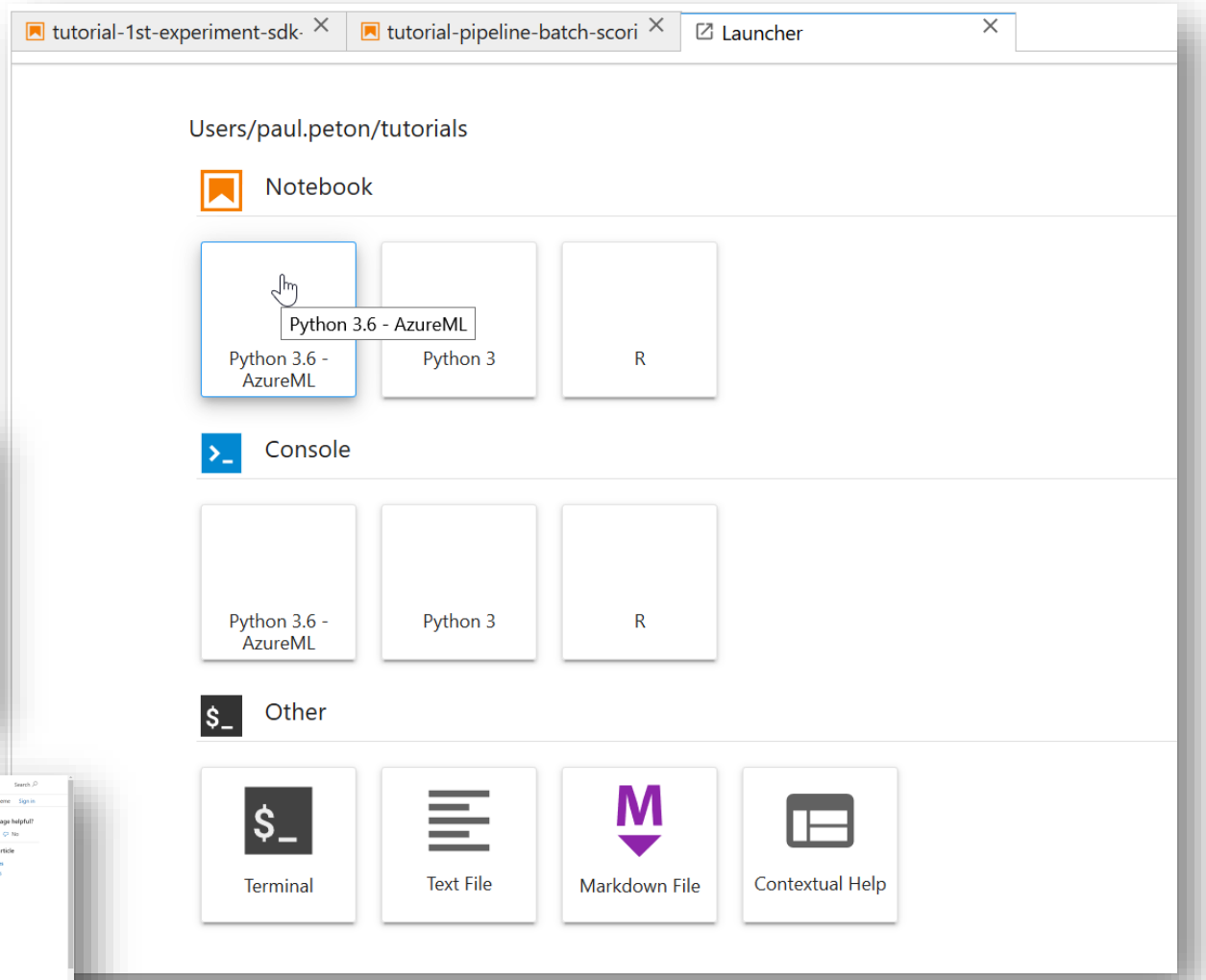
# Python Kernel and azureml.core library

From a compute instance :

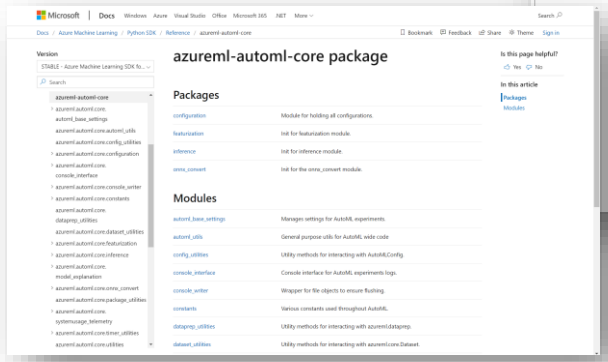
Launch JupyterLab

Use the preconfigured kernel with azureml.core



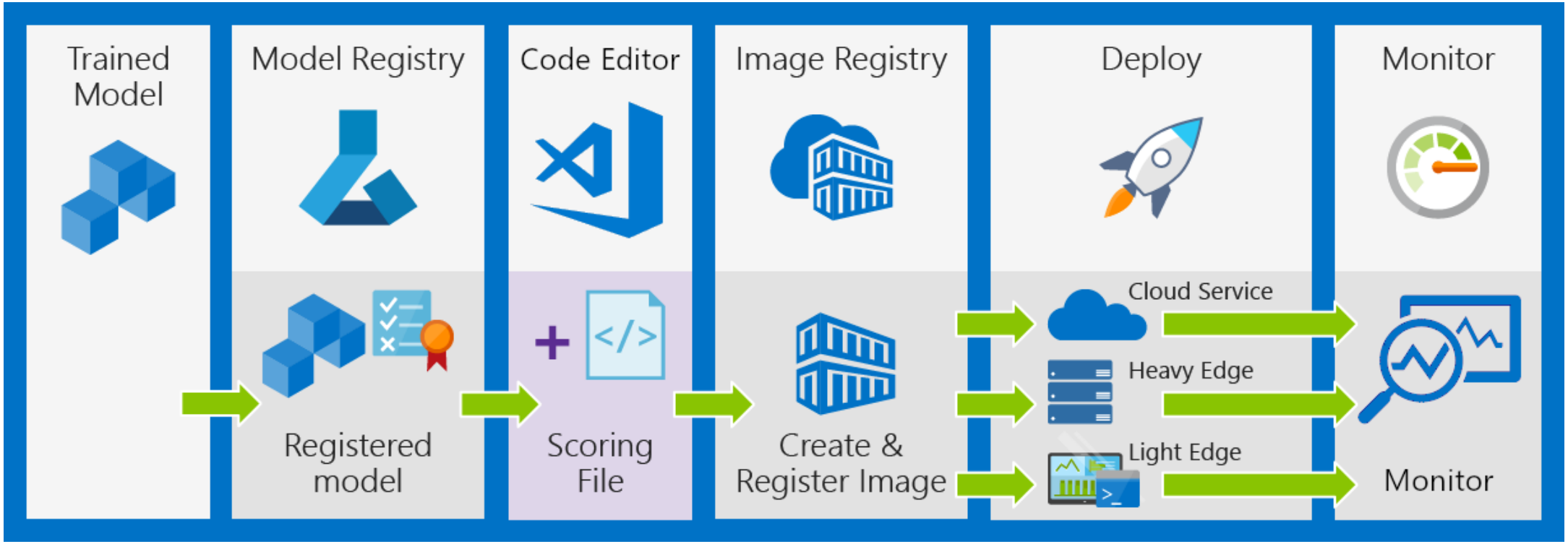




Available documentation



AZEO
talents & technology

# Azure Machine Learning Service features

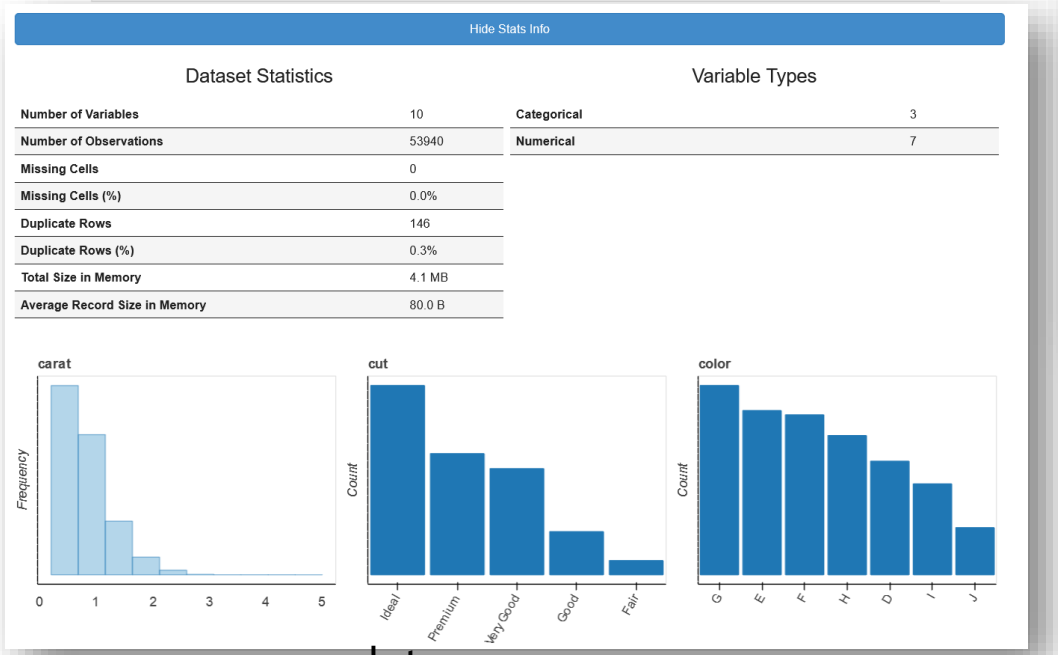


# Don’t forget Exploratory Data Analysis (EDA)

Use packages like :

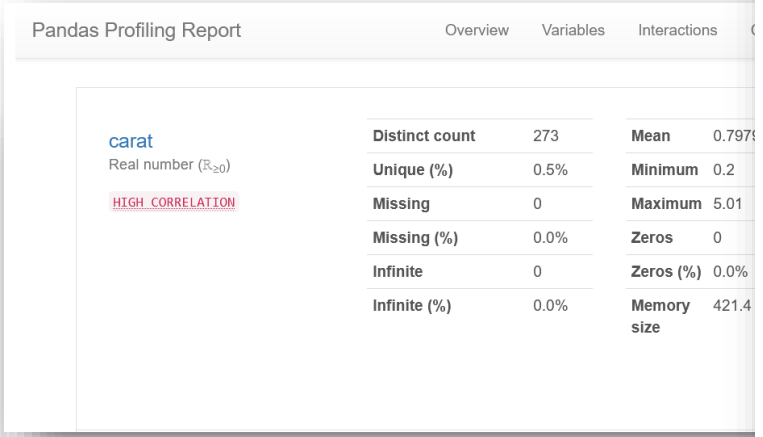
- azureml-dataprep (doc & example)
- pandas_profiling
- dataprep

```
from dataprep.eda import plot, plot_correlation, plot_missing

plot(df)
```




dataprep

```
import pandas_profiling
import pandas.util.testing as tm

pandas_profiling.ProfileReport(df)
```




pandas_profiling

```
from azureml.core import Dataset

dataset = Dataset.get_by_name(ws, name='Penguins')
df = dataset.to_pandas_dataframe()
```

```
import azureml.dataprep as dprep

dataset.get_profile()
```

```
ColumnProfile:
    column_name: Culmen Depth (mm)
    type: FieldType.DECIMAL

    min: 32.1
    max: 59.6
    count: 344.0
    missing_count: 2.0
    not_missing_count: 342.0
    percent_missing: 0.005813953488372093
    error_count: 0.0
    empty_count: 0.0
    unique_values: 165


    Quantiles (est.):
         0.1%: 32.1
           1%: 36.57
           5%: 36.5
          25%: 39.28
          50%: 44.38
          75%: 48.5
          95%: 52.0
          99%: 55.808
        99.9%: 59.6

    mean: 43.92192982456136
    std: 5.459583713926547
    variance: 29.807054329371987
    skewness: 0.052653027315875206
    kurtosis: -0.8931396557023659
    whisker_top: 59.6
    whisker_bottom: 32.1
```

azureml-dataprep

AZEO
talents & technology

# Supervised machine learning tasks

## Tasks

### Regression

How long will the trip take?

### Classification

Is the station out of order?

### Time series

How many rentals will there be next month?

AZEO
talents & technology

# Available algorithms (scikit-learn library)

| classification ; | régression ; | Prévision de séries chronologiques |
|---|---|---|
| Logistic Regression | Elastic Net | Elastic Net |
| Light GBM | Light GBM | Light GBM |
| Gradient Boosting | Gradient Boosting | Gradient Boosting |
| Decision Tree | Decision Tree | Decision Tree |
| K Nearest Neighbors | K Nearest Neighbors | K Nearest Neighbors |
| Linear SVC | LARS Lasso | LARS Lasso |
| Support Vector Classification (SVC) | Stochastic Gradient Descent (SGD) | Stochastic Gradient Descent (SGD) |
| Random Forest | Random Forest | Random Forest |
| Extremely Randomized Trees | Extremely Randomized Trees | Extremely Randomized Trees |
| Xgboost | Xgboost | Xgboost |
| Classifieur DNN | Régresseur DNN | Régresseur DNN |
| Classifieur linéaire DNN | Régression linéaire | Régression linéaire |
| Naive Bayes | FastLinearRegressor | Auto-ARIMA |
| Stochastic Gradient Descent (SGD) | OnlineGradientDescentRegressor | Prophet |
| AveragedPerceptronClassifier | | ForecastTCN |
| LinearSVMClassifier | | |

# Data sources and datasets

- Azure file systems
  - Azure Blob Container
  - Azure File Share
  - Azure Data Lake
  - Azure Data Lake Gen2
- Azure managed databases
  - Azure SQL Database
  - Azure Database for PostgreSQL
  - Azure Database for MySQL
- Databricks File System


# DEMO 1 – Automated ML GUI

AZEO
talents & technology

# Which dataset will we use ?


**Iris Versicolor**

**Iris Setosa**

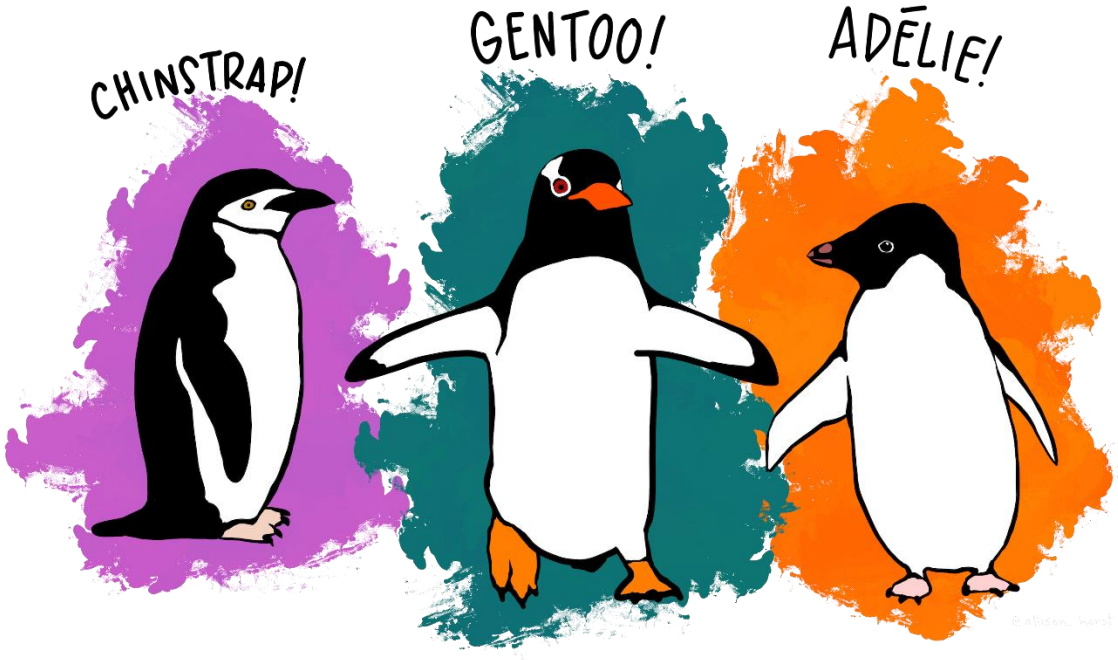
**Iris Virginica**

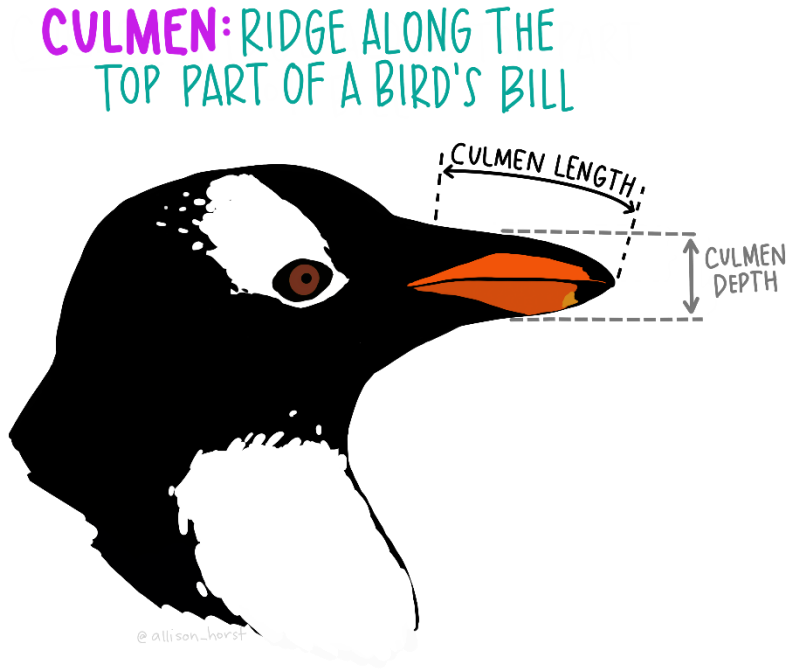
# Datasets - classification

## Palmer penguins

Data were collected and made available by Dr. Kristen Gorman and the Palmer Station, Antarctica LTER, a member of the Long Term Ecological Research Network.

https://allisonhorst.github.io/palmerpenguins/



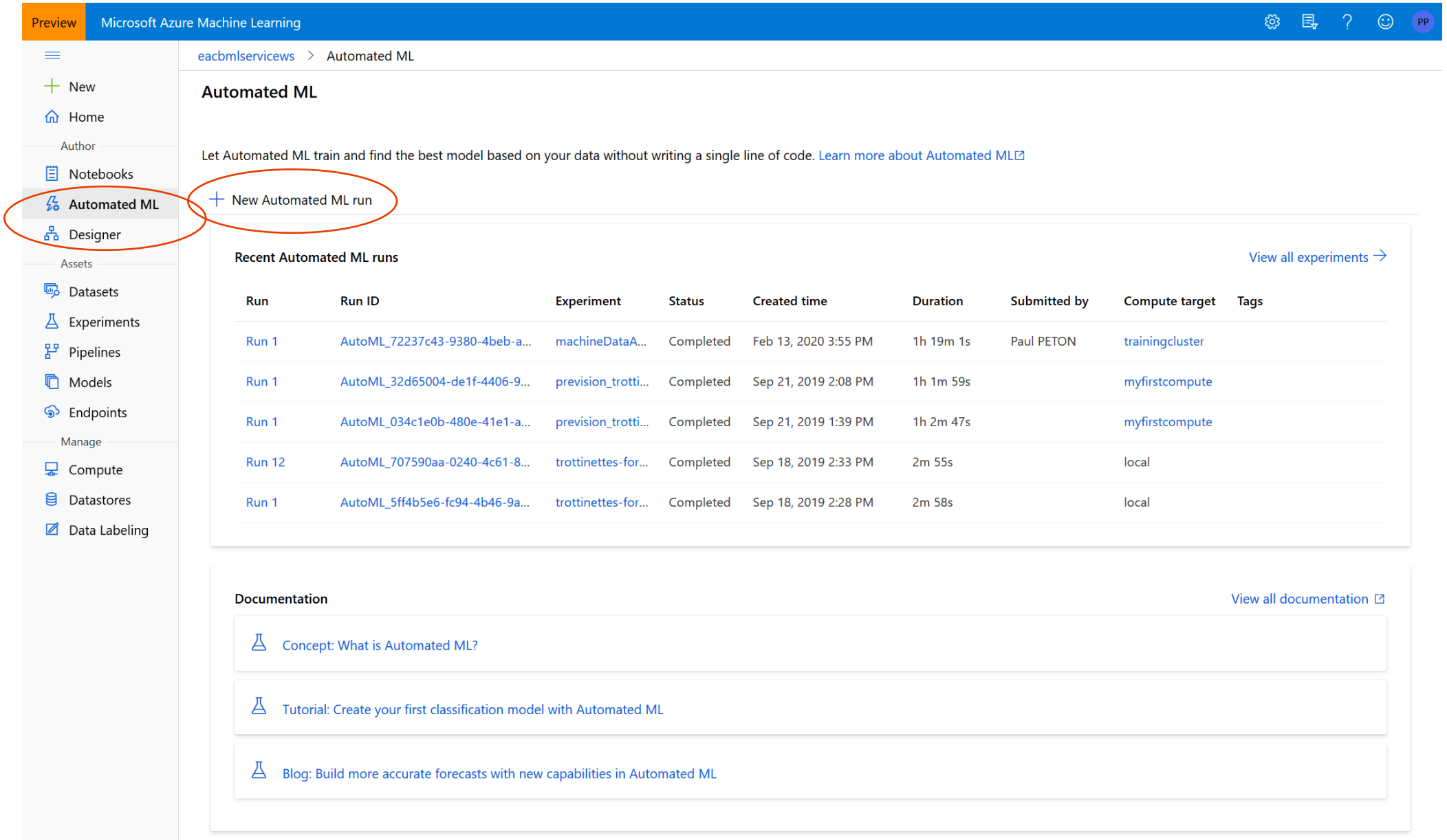

https://data-enhanced.com/2020/06/10/two-cheers-for-penguins-data/

AZEO
talents & technology

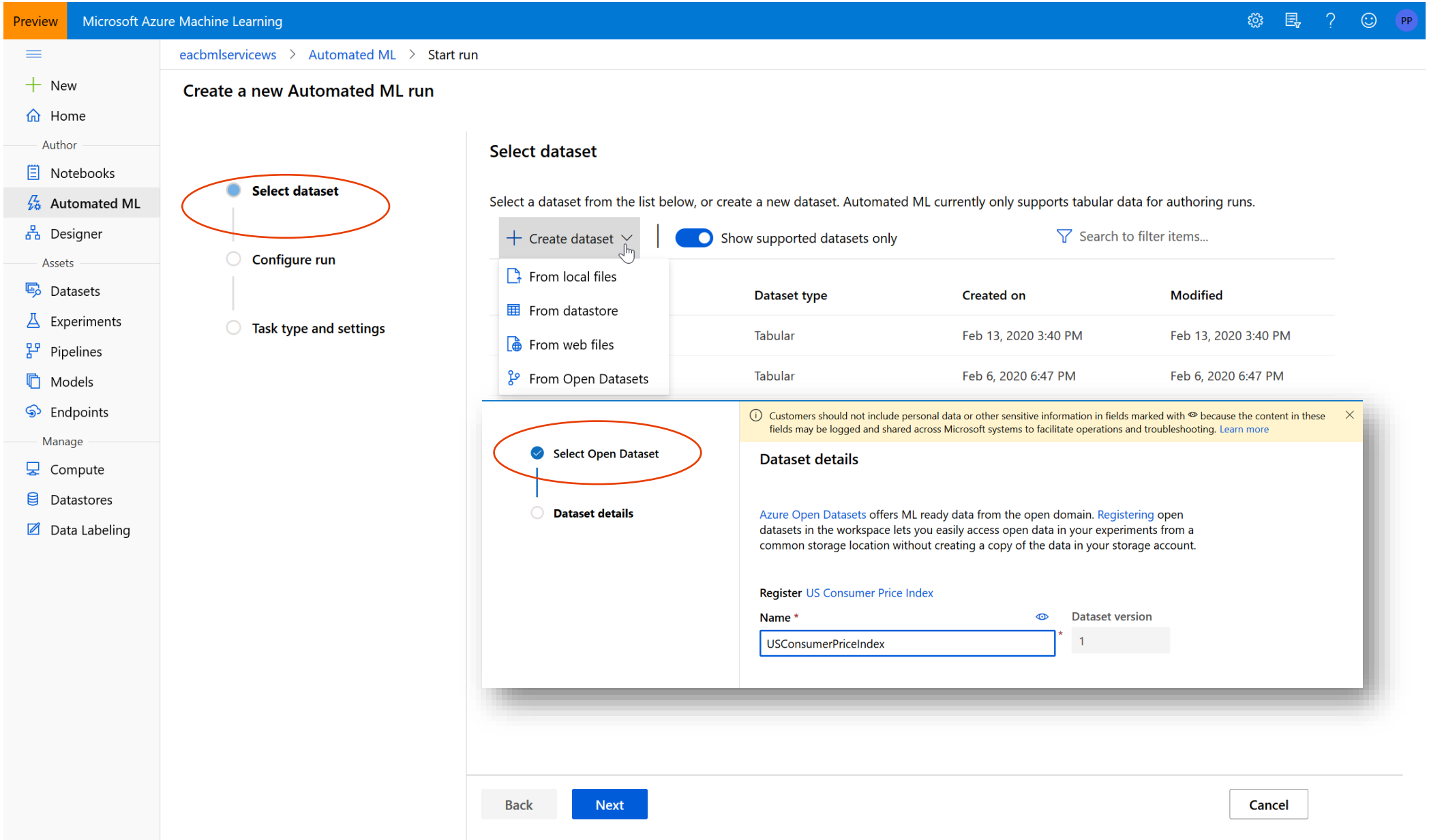
Now, we are playing the game of…

# The Seven Common Mistakes with an Automated ML GUI

# Launch a new Automated ML run

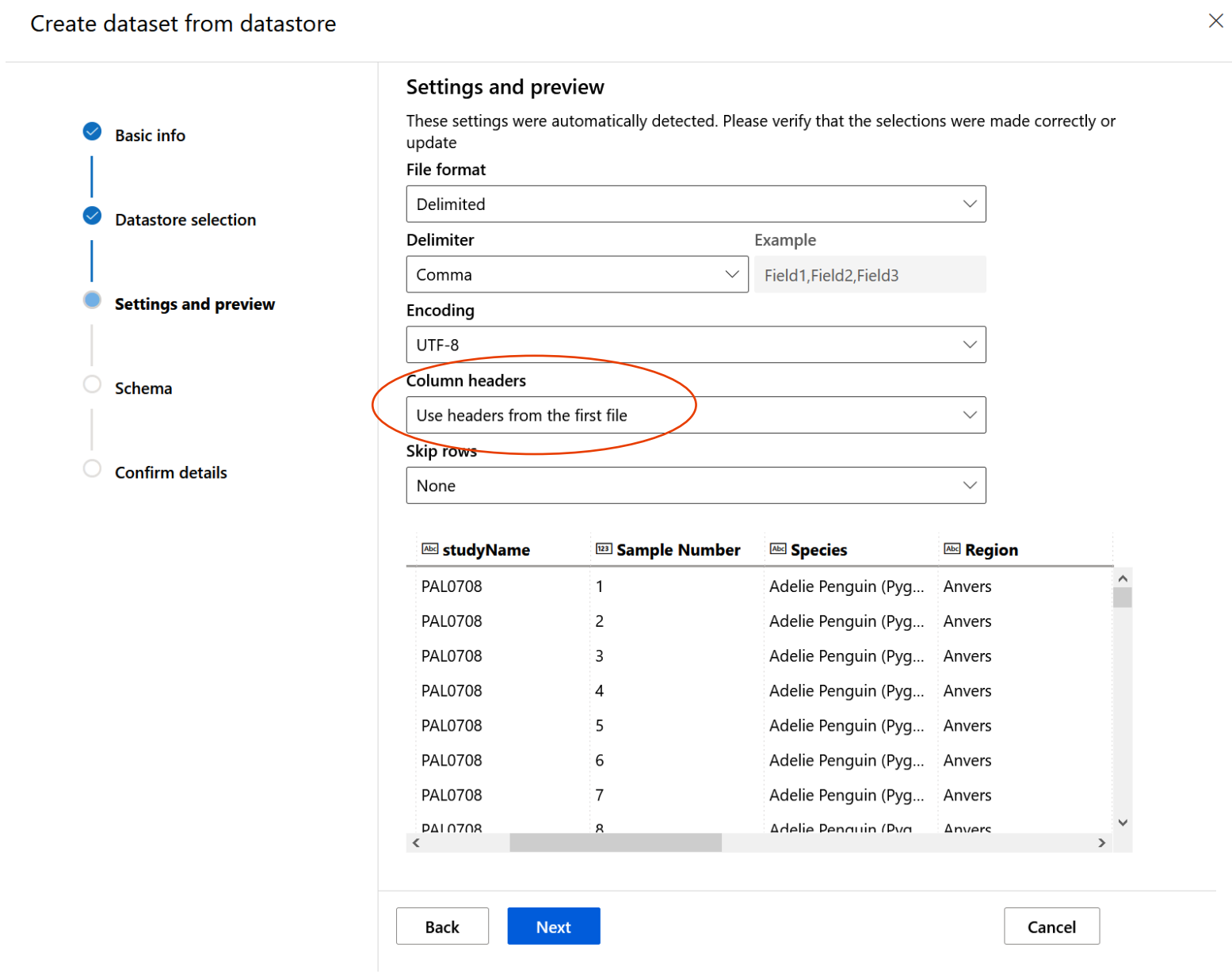


# Choose training dataset



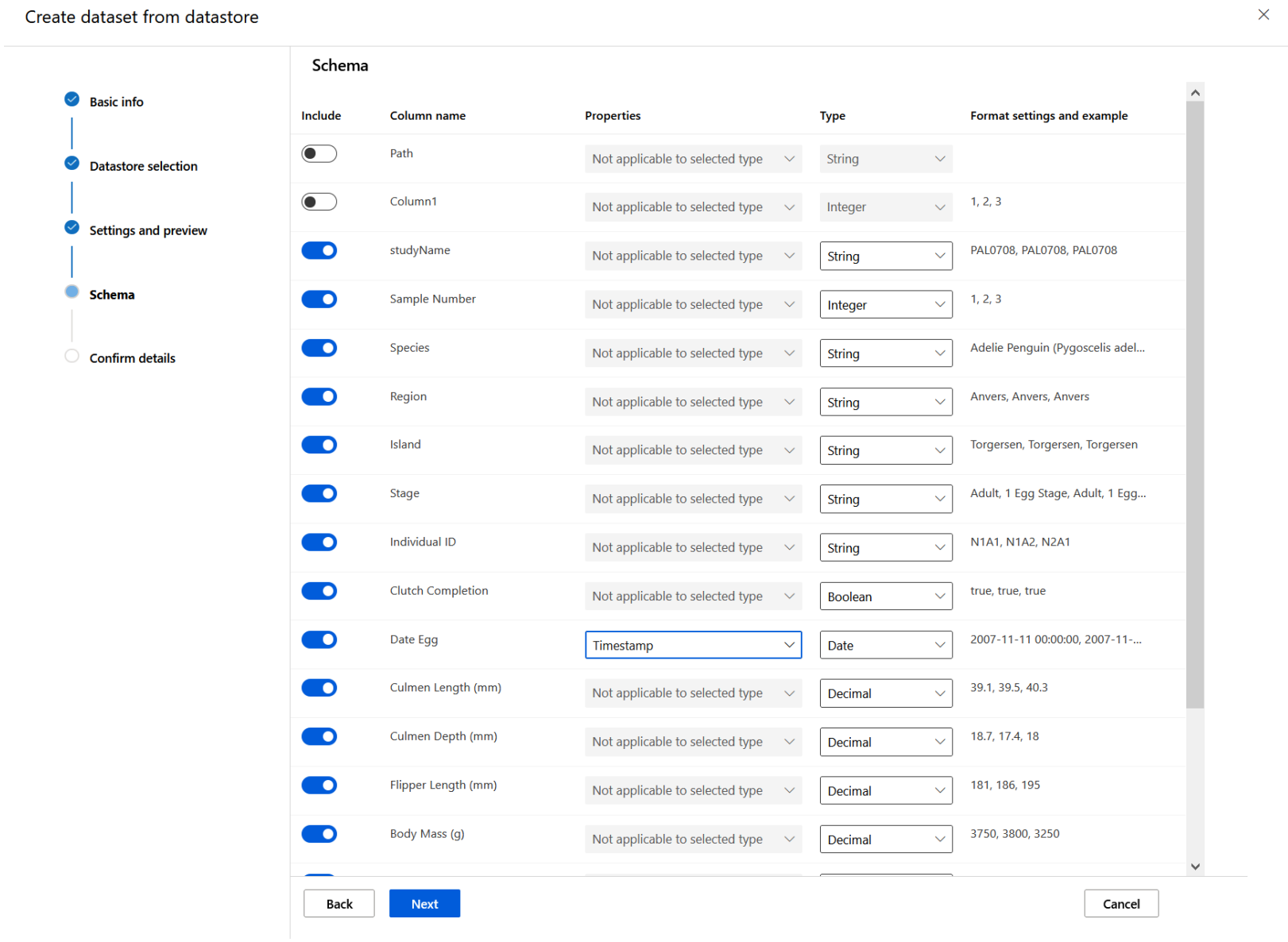
# Create a new dataset

First mistake : use a raw dataset. It's better to use a prepared one.

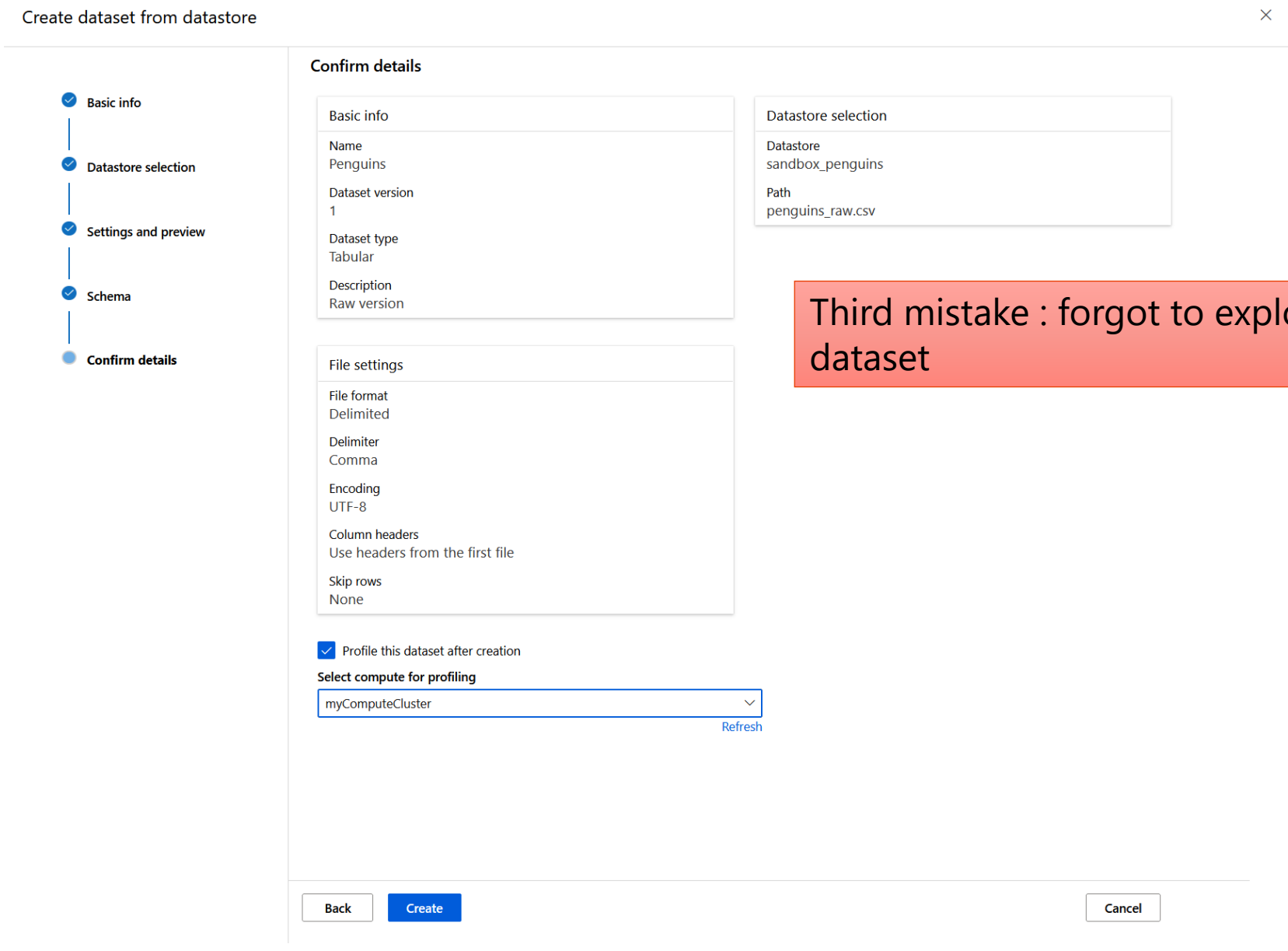


# Include columns and format settings

Second mistake : don't verify each feature type. Be careful with missing values.

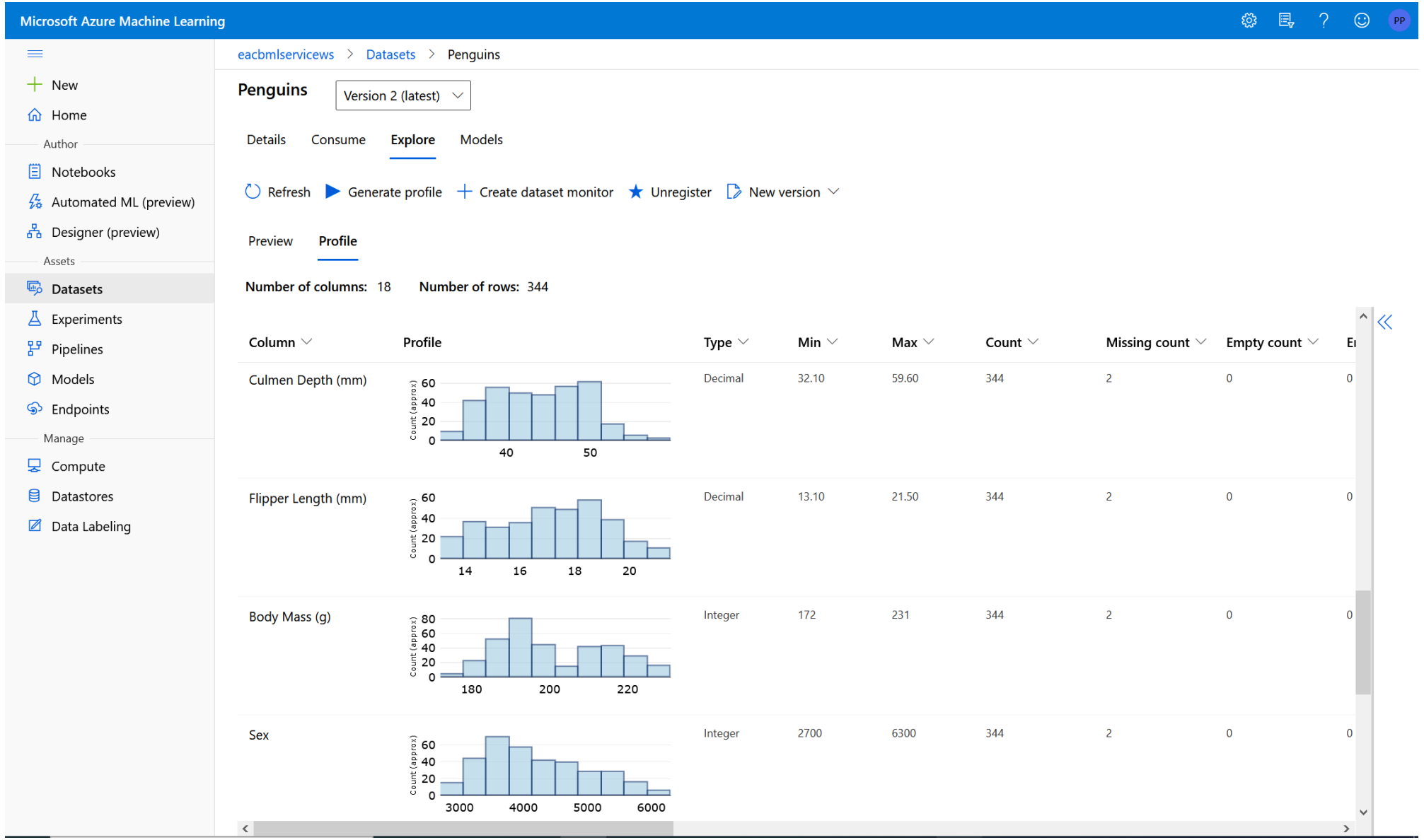


# Confirm details and profile this dataset

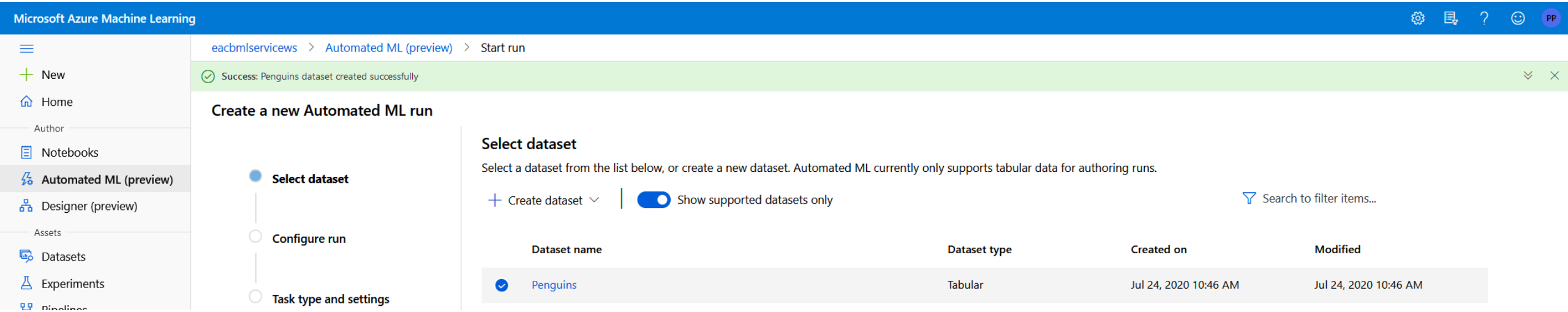



Third mistake : forgot to explore the dataset

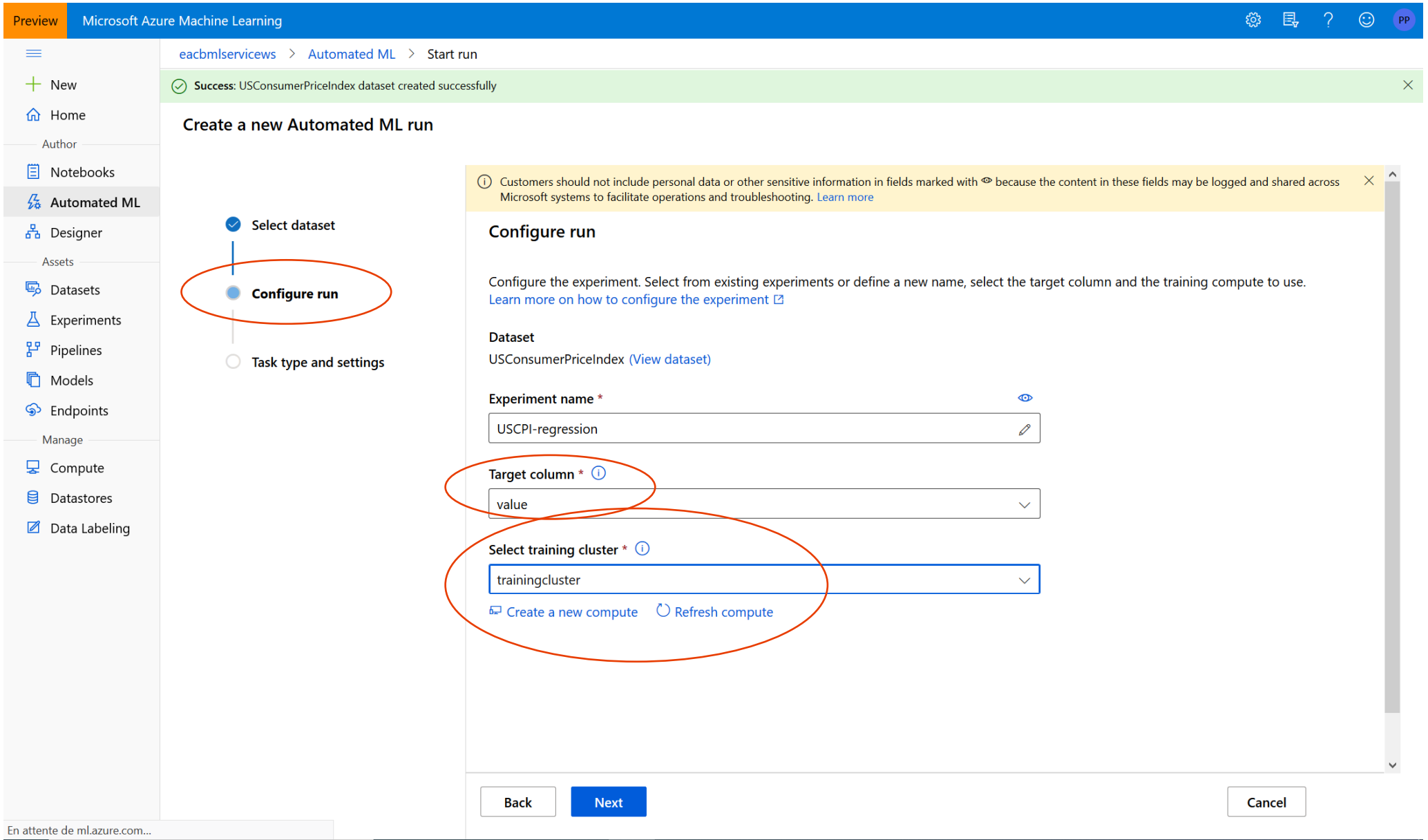
# Profile : some help for Exploratory Data Analysis



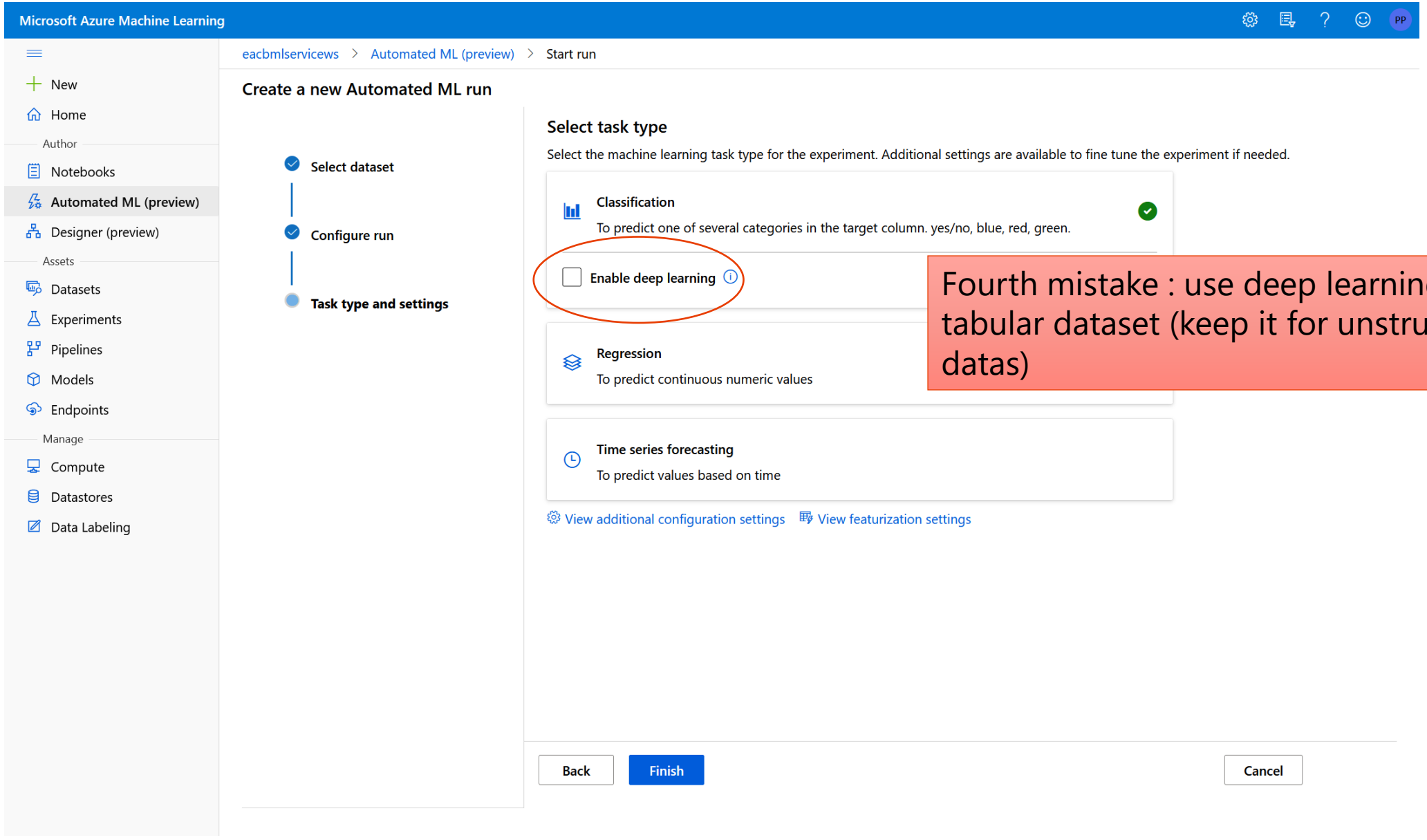
# Select dataset



# Choose the target



# Select task type (classification, regression, time series forecast)




Fourth mistake : use deep learning for tabular dataset (keep it for unstructured datas)

# Métrique d'évaluation (classification)

Additional configurations

Primary metric

AUC weighted

Explain best model

Blocked algorithms

A list of algorithms that Automated ML will not use during training.

Exit criterion

Training job time (hours) 1

Metric score threshold Metric score threshold

Validation

Validation type Train-validation split

Percentage validation of data * 30

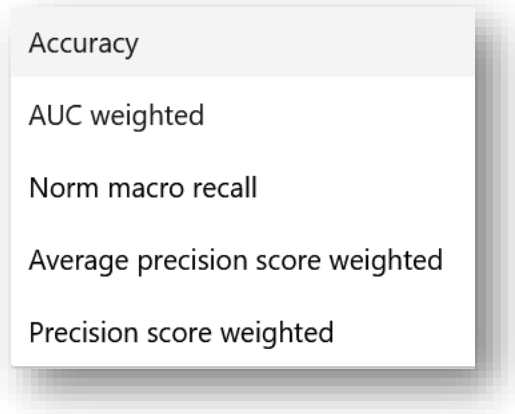
Automated ML recommends the percentage validation of data to be between 10 and 30

Concurrency

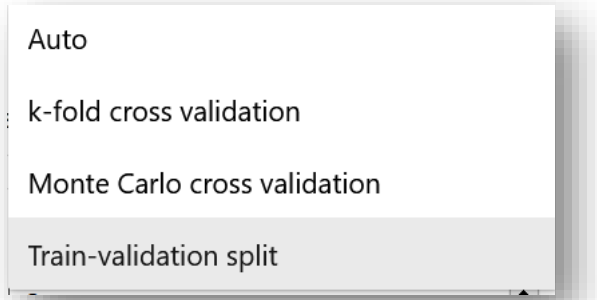
Max concurrent iterations 2

Save

Fifth mistake (bis) : choose accuracy for imbalanced dataset




Sixth mistake : choose train-validation on too small dataset , prefer k-fold CV



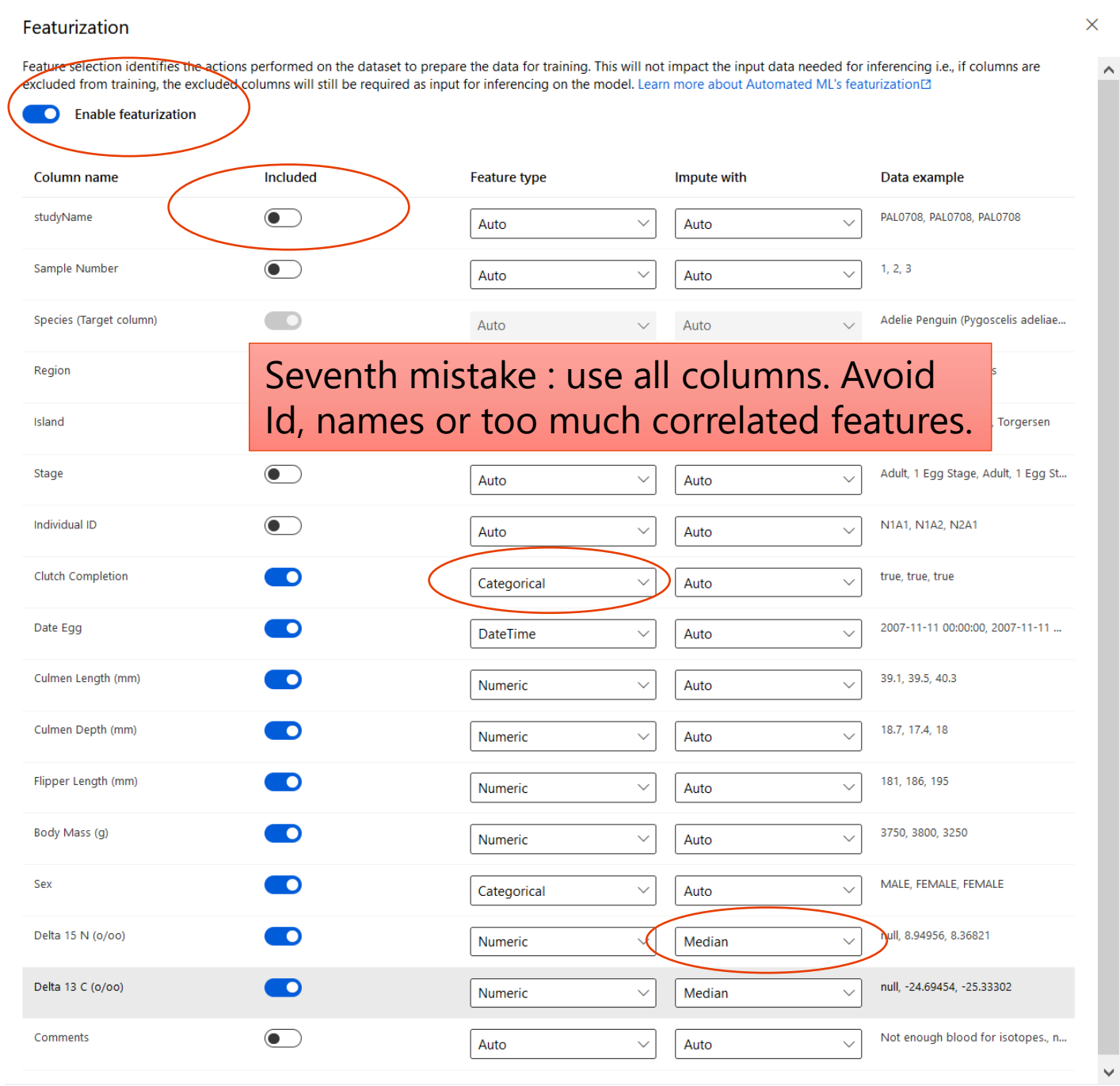
AZEO talents & technology

# Featurization

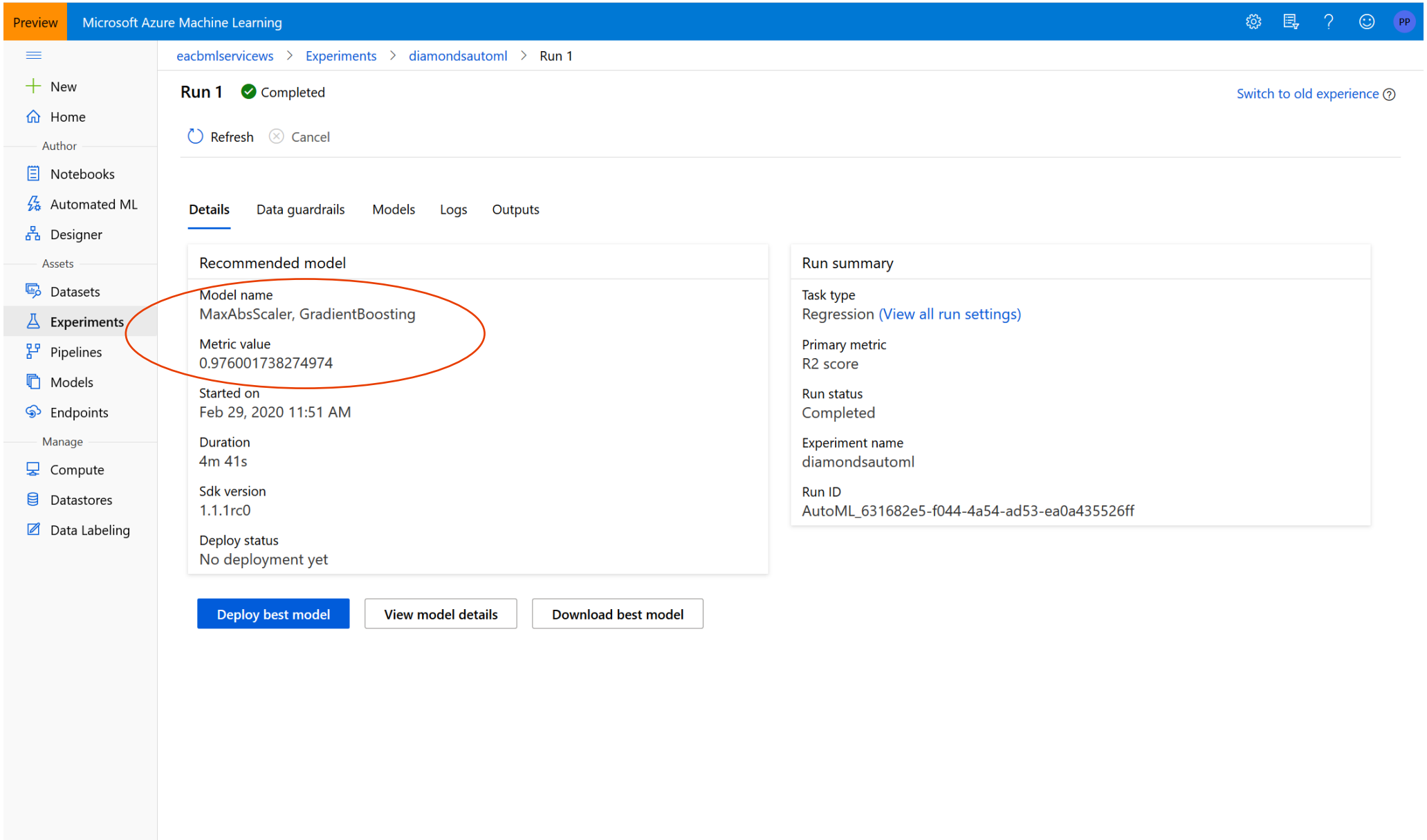
Select features

Select feature type

Select imputation method

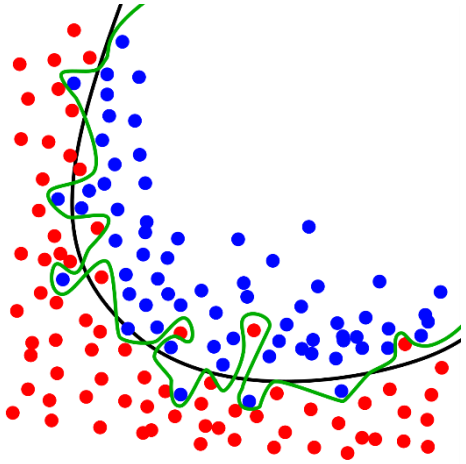


# The « best » model according to the primary metric



AZEO
talents & technology

# Prevent over-fitting



| Model | Train accuracy | Test accuracy |
|---|---|---|
| A | 99.9% | 95% |
| B | 87% | 87% |
| C | 99.9% | 45% |

The best way to prevent over-fitting is to follow ML best-practices including:

- Using more training data
- Preventing target leakage
- Using fewer features
- Regularization and hyperparameter optimization
- Model complexity limitations
- Cross-validation

AZEO talents & technology

# Training and cross validation

| Training data size | Validation technique |
|---|---|
| **Larger than 20,000 rows** | Train/validation data split is applied. The default is to take 10% of the initial training data set as the validation set. In turn, that validation set is used for metrics calculation. |
| **Smaller than 20,000 rows** | Cross-validation approach is applied. The default number of folds depends on the number of rows.<br>**If the dataset is less than 1,000 rows**, 10 folds are used.<br>**If the rows are between 1,000 and 20,000**, then three folds are used. |

AutoMLConfig parameters :

- training_data
- validation_data
- validation_size
- n_cross_validations
- cv_split_column_names

# Featurization (automatic or customize)

*Feature engineering* is the process of using domain knowledge of the data to create features that help machine learning algorithms to learn better.

In Azure Machine Learning

- data-scaling and
- normalization techniques

are applied to make feature engineering easier.

Collectively, these techniques and this feature engineering are called *featurization* in automated machine learning experiments.

| Featurization steps | Description |
|---|---|
| **Drop high cardinality or no variance features*** | Drop these features from training and validation sets. Applies to features with all values missing, with the same value across all rows, or with high cardinality (for example, hashes, IDs, or GUIDs). |
| **Impute missing values*** | For numeric features, impute with the average of values in the column.<br><br>For categorical features, impute with the most frequent value. |
| **Generate additional features*** | For DateTime features: Year, Month, Day, Day of week, Day of year, Quarter, Week of the year, Hour, Minute, Second.<br><br>For Text features: Term frequency based on unigrams, bigrams, and trigrams. Learn more about how this is done with BERT. |
| **Transform and encode*** | Transform numeric features that have few unique values into categorical features.<br><br>One-hot encoding is used for low-cardinality categorical features. One-hot-hash encoding is used for high-cardinality categorical features. |
| **Word embeddings** | A text featurizer converts vectors of text tokens into sentence vectors by using a pretrained model. Each word's embedding vector in a document is aggregated with the rest to produce a document feature vector. |
| **Target encodings** | For categorical features, this step maps each category with an averaged target value for regression problems, and to the class probability for each class for classification problems. Frequency-based weighting and k-fold cross-validation are applied to reduce overfitting of the mapping and noise caused by sparse data categories. |
| **Text target encoding** | For text input, a stacked linear model with bag-of-words is used to generate the probability of each class. |
| **Weight of Evidence (WoE)** | Calculates WoE as a measure of correlation of categorical columns to the target column. WoE is calculated as the log of the ratio of in-class vs. out-of-class probabilities. This step produces one numeric feature column per class and removes the need to explicitly impute missing values and outlier treatment. |
| **Cluster Distance** | Trains a k-means clustering model on all numeric columns. Produces *k* new features (one new numeric feature per cluster) that contain the distance of each sample to the centroid of each cluster. |

AZEO talents & technology

# AutoMLConfig parameters

| Property | Description |
|---|---|
| **task** | forecasting |
| **primary_metric** | This is the metric that you want to optimize.<br>Forecasting supports the following primary metrics<br>*spearman_correlation*<br>*normalized_root_mean_squared_error*<br>*r2_score*<br>*normalized_mean_absolute_error* |
| **experiment_timeout_hours** | Experimentation timeout in hours. |
| **enable_early_stopping** | If early stopping is on, training will stop when the primary metric is no longer improving. |
| **training_data** | Input dataset, containing both features and label column. |
| **label_column_name** | The name of the label column. |
| **compute_target** | The remote compute for training. |
| **n_cross_validations** | Number of cross-validation folds to use for model/pipeline selection |
| **enable_voting_ensemble** | Allow AutoML to create a Voting ensemble of the best performing models |
| **enable_stack_ensemble** | Allow AutoML to create a Stack ensemble of the best performing models |
| **debug_log** | Log file path for writing debugging information |
| **time_column_name** | Name of the datetime column in the input data |
| **grain_column_names** | Name(s) of the columns defining individual series in the input data |
| **max_horizon** | Maximum desired forecast horizon in units of time-series frequency |
| **featurization** | 'auto' / 'off' / FeaturizationConfig Indicator for whether featurization step should be done automatically or not, or whether customized featurization should be used. Setting this enables AutoML to perform featurization on the input to handle *missing data*, and to perform some common *feature extraction*. |
| **max_cores_per_iteration** | Maximum number of cores to utilize per iteration. A value of -1 indicates all available cores should be used. |


# DEMO 2 – Python SDK

AZEO talents & technology

# Automated ML with azureml SDK

```
[ ]: !pip install --upgrade azureml-sdk
```

```
[ ]: !pip install --upgrade azureml-widgets
```

```
[ ]: !pip install --upgrade azureml-train-automl-runtime
```

```
[1]: import pandas as pd
import numpy as np
import azureml.core
```

```
[2]: from azureml.core import Workspace
ws = Workspace.from_config()
```

```
[3]: from azureml.core import Experiment
experiment_name = 'penguins-automatedML'

experiment = Experiment(workspace=ws, name=experiment_name)
```

•••

```
[14]: from azureml.core import Dataset

dataset = Dataset.get_by_name(ws, name='Penguins').drop_columns(['Individual ID', 'studyName','Sample Number', 'Stage','Region','Island','Comments'])
dataset.take(3).to_pandas_dataframe()
```

[14]:

| | Species | Clutch Completion | Date Egg | Culmen Length (mm) | Culmen Depth (mm) | Flipper Length (mm) | Body Mass (g) | Sex | Delta 15 N (o/oo) | Delta 13 C (o/oo) |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | Adelie Penguin (Pygoscelis adeliae) | True | 2007-11-11 | 39.10 | 18.70 | 181 | 3750 | MALE | nan | nan |
| **1** | Adelie Penguin (Pygoscelis adeliae) | True | 2007-11-11 | 39.50 | 17.40 | 186 | 3800 | FEMALE | 8.95 | -24.69 |
| **2** | Adelie Penguin (Pygoscelis adeliae) | True | 2007-11-16 | 40.30 | 18.00 | 195 | 3250 | FEMALE | 8.37 | -25.33 |

# Automated ML with azureml SDK

[21]:
```
import logging

automl_settings = {
    "iteration_timeout_minutes": 1,
    "iterations": 40,
    "primary_metric": 'norm_macro_recall', #precision_score_weighted, average_precision_score_weighted
    "featurization": 'auto', #off, auto, FeaturizationConfig
    "verbosity": logging.INFO,
    "enable_local_managed": True # to submit a local conda or local docker run
}
```

[22]:
```
#https://docs.microsoft.com/en-us/python/api/azureml-train-automl-client/azureml.train.automl.automlconfig.automlconfig?view=azure-ml-py
from azureml.train.automl import AutoMLConfig

automl_config = AutoMLConfig(task='classification',
                             debug_log='automated_ml_errors.log',
                             #compute_target = compute_target,
                             training_data=train_data,
                             validation_data=test_data,
                             label_column_name=label,
                             blocked_models=['ExtremeRandomTrees','SVM'], # https://docs.microsoft.com/en-us/python/api/azureml-train-automl-client/azureml.train
                             **automl_settings)
```

```
featurization_config = FeaturizationConfig()
featurization_config.blocked_transformers = ['LabelEncoder']
featurization_config.drop_columns = ['aspiration', 'stroke']
featurization_config.add_column_purpose('engine-size', 'Numeric')
featurization_config.add_column_purpose('body-style', 'CategoricalHash')
#default strategy mean, add transformer param for for 3 columns
featurization_config.add_transformer_params('Imputer', ['engine-size'], {"strategy": "median"})
featurization_config.add_transformer_params('Imputer', ['city-mpg'], {"strategy": "median"})
featurization_config.add_transformer_params('Imputer', ['bore'], {"strategy": "most_frequent"})
featurization_config.add_transformer_params('HashOneHotEncoder', [], {"number_of_bits": 3})
```

Featurization in SDK needs the enterprise licence of Azure Machine Learning.

# Local vs remote target (only with the SDSK)

**Local compute**: Training occurs on your local laptop or VM compute.

**Remote compute**: Training occurs on Machine Learning compute clusters.

| | Pros (Advantages) | Cons (Handicaps) |
|---|---|---|
| **Local compute target** | • No environment start up time | • Subset of features<br>• Can't parallelize runs<br>• Worse for large data.<br>• No data streaming while training<br>• No DNN-based featurization<br>• Python SDK only |
| **Remote ML compute clusters** | • Full set of features<br>• Parallelize child runs<br>• Large data support<br>• DNN-based featurization<br>• Dynamic scalability of compute cluster on demand<br>• No-code experience (web UI) also available | • Start up time for cluster nodes<br>• Start up time for each child run |

```
import logging

automl_settings = {
    "iteration_timeout_minutes": 1,
    "iterations": 10,
    "primary_metric": 'norm_macro_recall', #precision_score_weighted, average_precision_score_weighted
    "featurization": featurization_config, #off, auto, FeaturizationConfig
    "verbosity": logging.INFO,
    "enable_local_managed": True # to submit a local conda or local docker run
}
```

```
#https://docs.microsoft.com/en-us/python/api/azureml-train-automl-client/azureml.train.automl.automlconfig
from azureml.train.automl import AutoMLConfig

automl_config = AutoMLConfig(task='classification',
                             debug_log='automated_ml_errors.log',
                             #compute_target = compute_target,
                             training_data=train_data,
                             validation_data=test_data,
                             label_column_name=label,
                             blocked_models=['ExtremeRandomTrees','SVM'], # https://docs.microsoft.com/en-
                             **automl_settings)
```

# Automated ML with azureml SDK

```
[23]: import azureml.train.automl.runtime

local_run = experiment.submit(automl_config, show_output=True)
```

```
Running on local machine
Parent Run ID: AutoML_36f1217c-24f4-42e8-b5da-4b502d13b222

Current status: DatasetEvaluation. Gathering dataset statistics.
Current status: FeaturesGeneration. Generating features for the dataset.
Current status: DatasetFeaturization. Beginning to fit featurizers and featurize the dataset.
Current status: DatasetFeaturizationCompleted. Completed fit featurizers and featurizing the dataset.

****************************************************************************************************
DATA GUARDRAILS:

TYPE:         Class balancing detection
STATUS:       PASSED
DESCRIPTION:  Your inputs were analyzed, and all classes are balanced in your training data.
              Learn more about imbalanced data: https://aka.ms/AutomatedMLImbalancedData

****************************************************************************************************

TYPE:         Missing feature values imputation
STATUS:       DONE
DESCRIPTION:  If the missing values are expected, let the run complete. Otherwise cancel the current run and use a script to customize the handling of missing feature values that may
be more appropriate based on the data type and business requirement.
              Learn more about missing value imputation: https://aka.ms/AutomatedMLFeaturization
DETAILS:
+---------------------------------+---------------------------------+---------------------------------+
|Column name                      |Missing value count              |Imputation type                  |
+=================================+=================================+=================================+
|Culmen Length (mm)               |1                                |mean                             |
|Culmen Depth (mm)                |1                                |mean                             |
|Flipper Length (mm)              |1                                |mean                             |
|Body Mass (g)                    |1                                |mean                             |
|Sex                              |8                                |                                 |
|Delta 15 N (o/oo)                |10                               |mean                             |
|Delta 13 C (o/oo)                |9                                |mean                             |
+---------------------------------+---------------------------------+---------------------------------+

****************************************************************************************************

TYPE:         High cardinality feature detection
STATUS:       PASSED
DESCRIPTION:  Your inputs were analyzed, and no high cardinality features were detected.
              Learn more about high cardinality feature handling: https://aka.ms/AutomatedMLFeaturization

****************************************************************************************************
Current status: ModelSelection. Beginning model selection.

****************************************************************************************************
```

```
****************************************************************************************************
Current status: ModelSelection. Beginning model selection.

****************************************************************************************************
ITERATION: The iteration being evaluated.
PIPELINE: A summary description of the pipeline being evaluated.
DURATION: Time taken for the current iteration.
METRIC: The result of computing score on the fitted pipeline.
BEST: The best observed score thus far.
****************************************************************************************************

 ITERATION   PIPELINE                                       DURATION      METRIC      BEST
         0   MaxAbsScaler LightGBM                          0:00:17       0.9800    0.9800
         1   StandardScalerWrapper SGD                      0:00:16       0.9800    0.9800
         2   StandardScalerWrapper RandomForest             0:00:17       0.9522    0.9800
         3   MinMaxScaler SGD                               0:00:16       0.9444    0.9800
         4   StandardScalerWrapper SGD                      0:00:17       0.9800    0.9800
         5   MinMaxScaler RandomForest                      0:00:17       0.9800    0.9800
         6   MinMaxScaler SGD                               0:00:18       1.0000    1.0000
         7   StandardScalerWrapper SGD                      0:00:17       0.9800    1.0000
         8   StandardScalerWrapper SGD                      0:00:12       0.9800    1.0000
         9   MinMaxScaler RandomForest                      0:00:18       0.9800    1.0000
        10   StandardScalerWrapper SGD                      0:00:12       0.9600    1.0000
```

# Automated ML with azureml SDK

[25]:
```
local_run.wait_for_completion()
```

```
...
```

[26]:
```
best_run, fitted_model = local_run.get_output()

print(best_run)
print(fitted_model)
```

```
Run(Experiment: penguins-automatedML,
Id: AutoML_36f1217c-24f4-42e8-b5da-4b502d13b222_6,
Type: None,
Status: Completed)
Pipeline(memory=None,
     steps=[('datatransformer', DataTransformer(enable_dnn=None, enable_feature_sweeping=None,
        feature_sweeping_config=None, feature_sweeping_timeout=None,
        featurization_config=None, force_text_dnn=None,
        is_cross_validation=None, is_onnx_compatible=None, logger=None,
        obser...    n_jobs=1, penalty='none', power_t=0.2222222222222222,
           random_state=None, tol=0.001))])
Y_transformer(['LabelEncoder', LabelEncoder()])
```

[27]:
```
print(best_run.get_file_names())
```

```
['accuracy_table', 'confusion_matrix', 'explanation/b2f7213f/classes.interpret.json', 'explanation/b2f7213f/eval_data_viz.interpret.json', 'explanation/b2f7213f/
expected_values.interpret.json', 'explanation/b2f7213f/features.interpret.json', 'explanation/b2f7213f/global_names/0.interpret.json', 'explanation/b2f7213f/glob
al_rank/0.interpret.json', 'explanation/b2f7213f/global_values/0.interpret.json', 'explanation/b2f7213f/local_importance_values.interpret.json', 'explanation/b2f
7213f/per_class_names/0.interpret.json', 'explanation/b2f7213f/per_class_rank/0.interpret.json', 'explanation/b2f7213f/per_class_values/0.interpret.json', 'expla
nation/b2f7213f/rich_metadata.interpret.json', 'explanation/b2f7213f/visualization_dict.interpret.json', 'explanation/b2f7213f/ys_pred_proba_viz.interpret.json',
'explanation/b2f7213f/ys_pred_viz.interpret.json', 'outputs/conda_env_v_1_0_0.yml', 'outputs/env_dependencies.json', 'outputs/model.pkl', 'outputs/pipeline_grap
h.json', 'outputs/scoring_file_v_1_0_0.py']
```

[25]:
```
# register best model
from azureml.core.model import Model

model = best_run.register_model(model_name='penguins_class_best_model', model_path='outputs/model.pkl')

print(model.name, model.version, sep='\t')
```

```
penguins_class_best_model       4
```

# Widget in Jupyter notebooks

```
[24]: from azureml.widgets import RunDetails

      RunDetails(local_run).show()
```

AutoML_36f1217c-24f4-42e8-b5da-4b502d13b222:

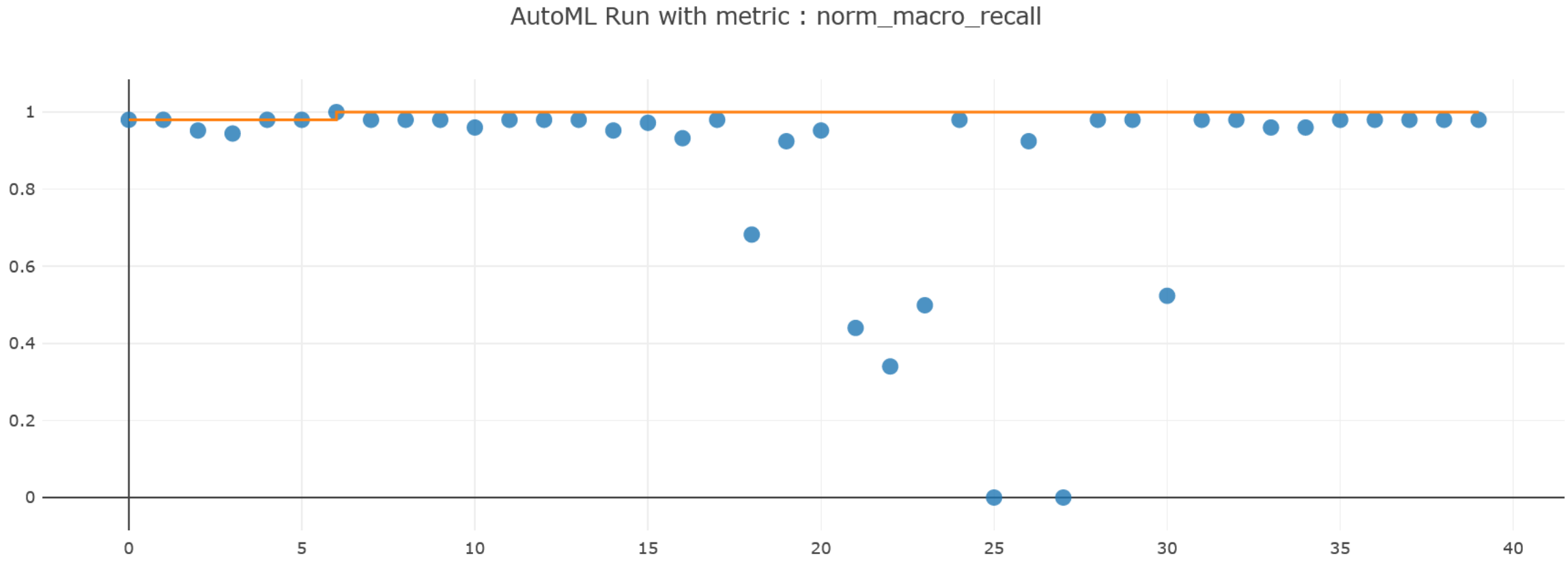
Status: Completed


| Iteration | Pipeline | Iteration metric | Best metric | Status | Duration | Started | Run Id |
|---|---|---|---|---|---|---|---|
| 6 | MinMaxScaler, SGD | 1 | 1 | Completed | 0:00:17 | Jul 28, 2020 7:46 AM | |
| 0 | MaxAbsScaler, LightGBM | 0.98 | 0.98 | Completed | 0:00:16 | Jul 28, 2020 7:44 AM | |
| 1 | StandardScalerWrapper, SGD | 0.98 | 0.98 | Completed | 0:00:15 | Jul 28, 2020 7:44 AM | |
| 4 | StandardScalerWrapper, SGD | 0.98 | 0.98 | Completed | 0:00:16 | Jul 28, 2020 7:45 AM | |
| 5 | MinMaxScaler, RandomForest | 0.98 | 0.98 | Completed | 0:00:16 | Jul 28, 2020 7:45 AM | |

Pages: **1** 2 3 4 5 6 7 8 Next Last 5 per page

norm_macro_recall

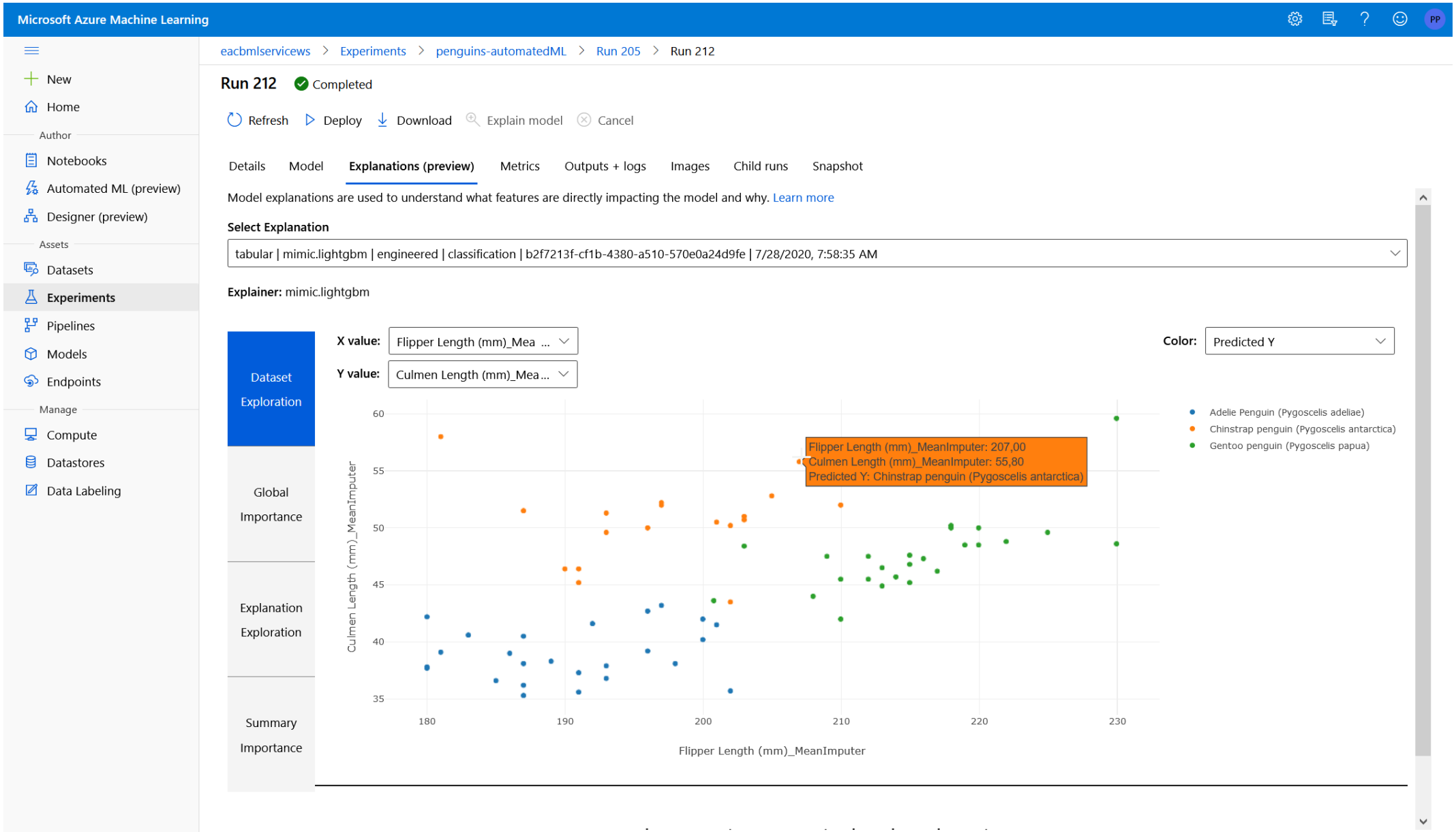
# Widget in Jupyter notebooks



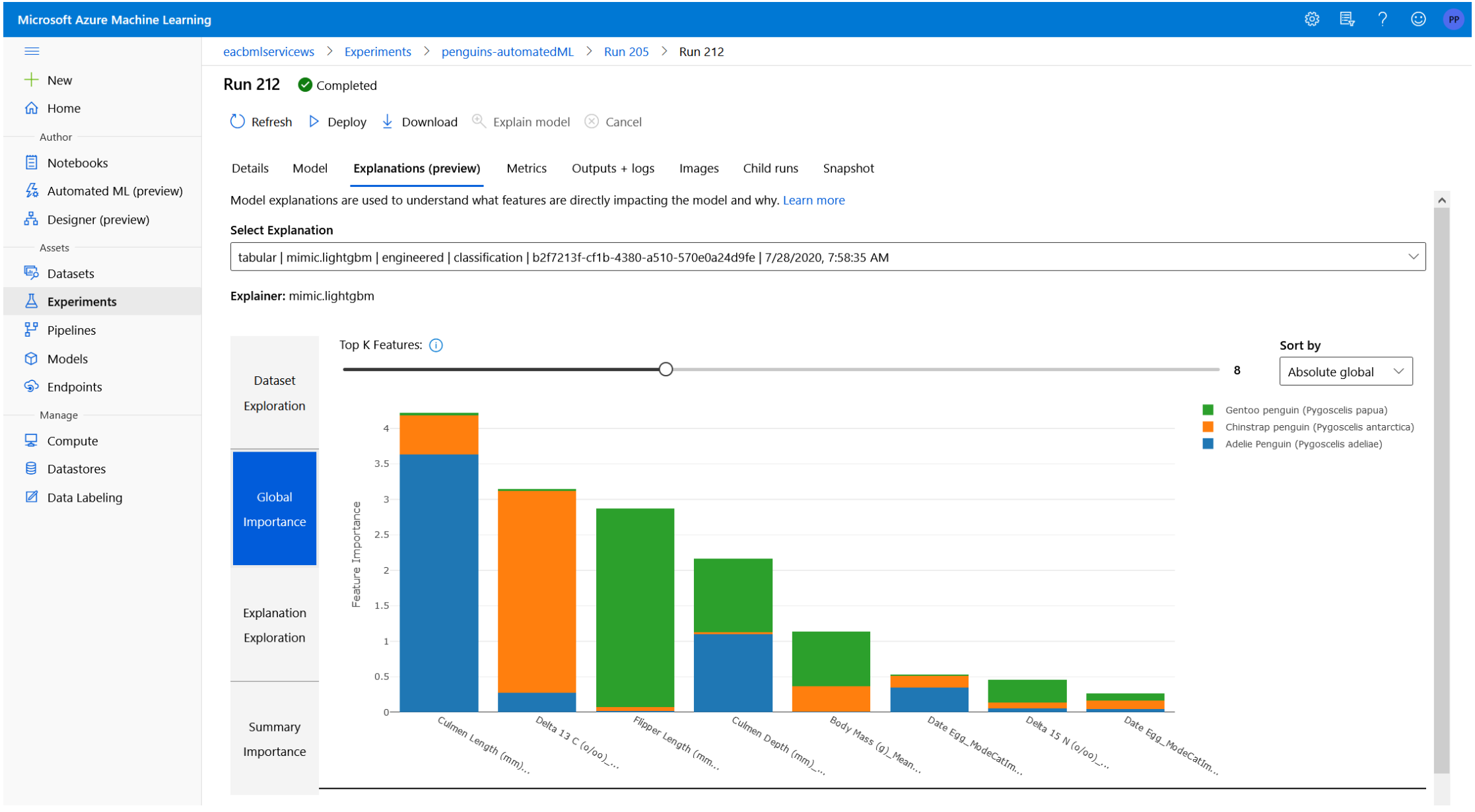

Click here to see the run in Azure Machine Learning studio


# DEMO 3 – Explanation and deployment

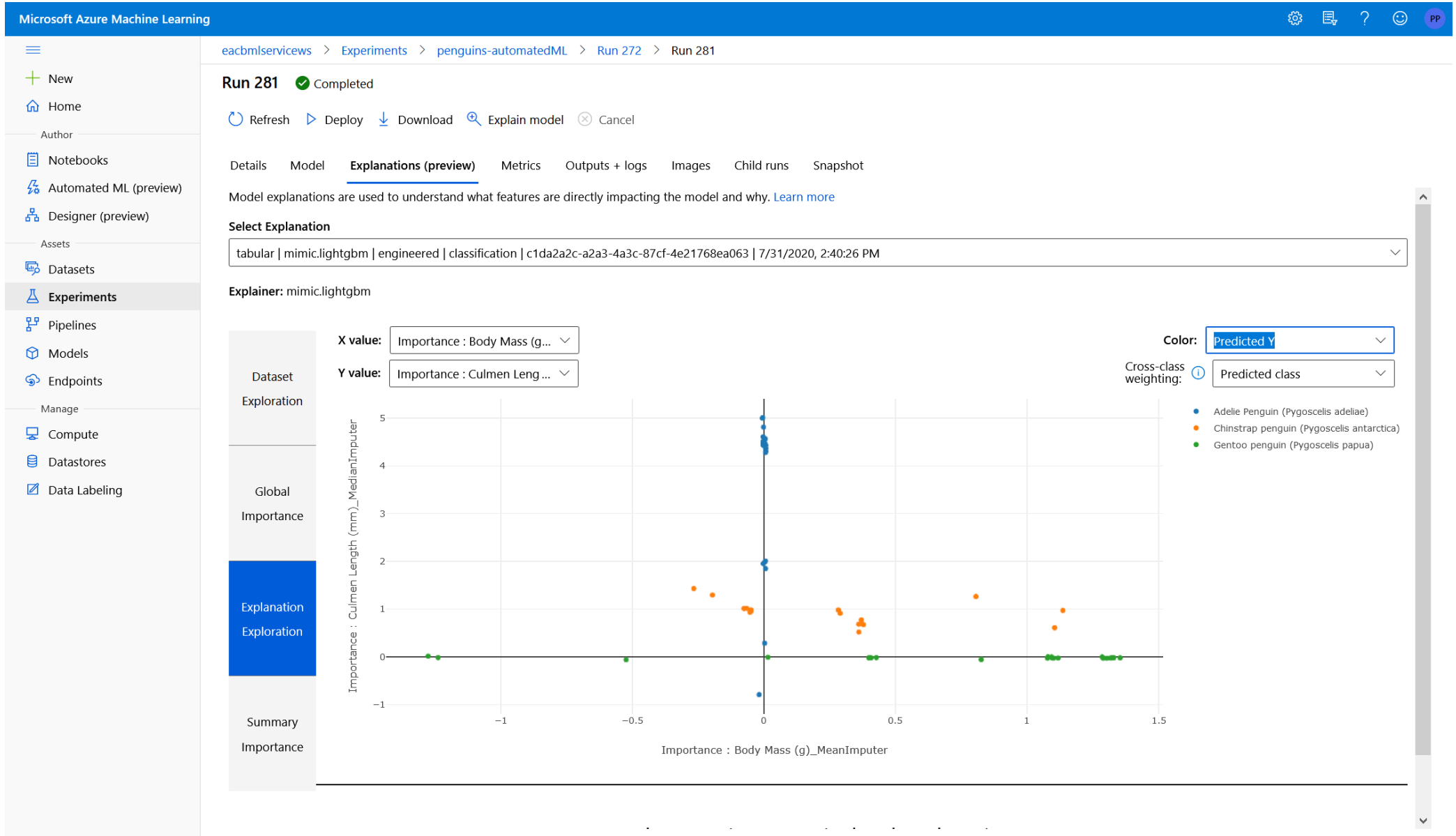
# Explanations (dataset exploration)



# Explanations (global importance)

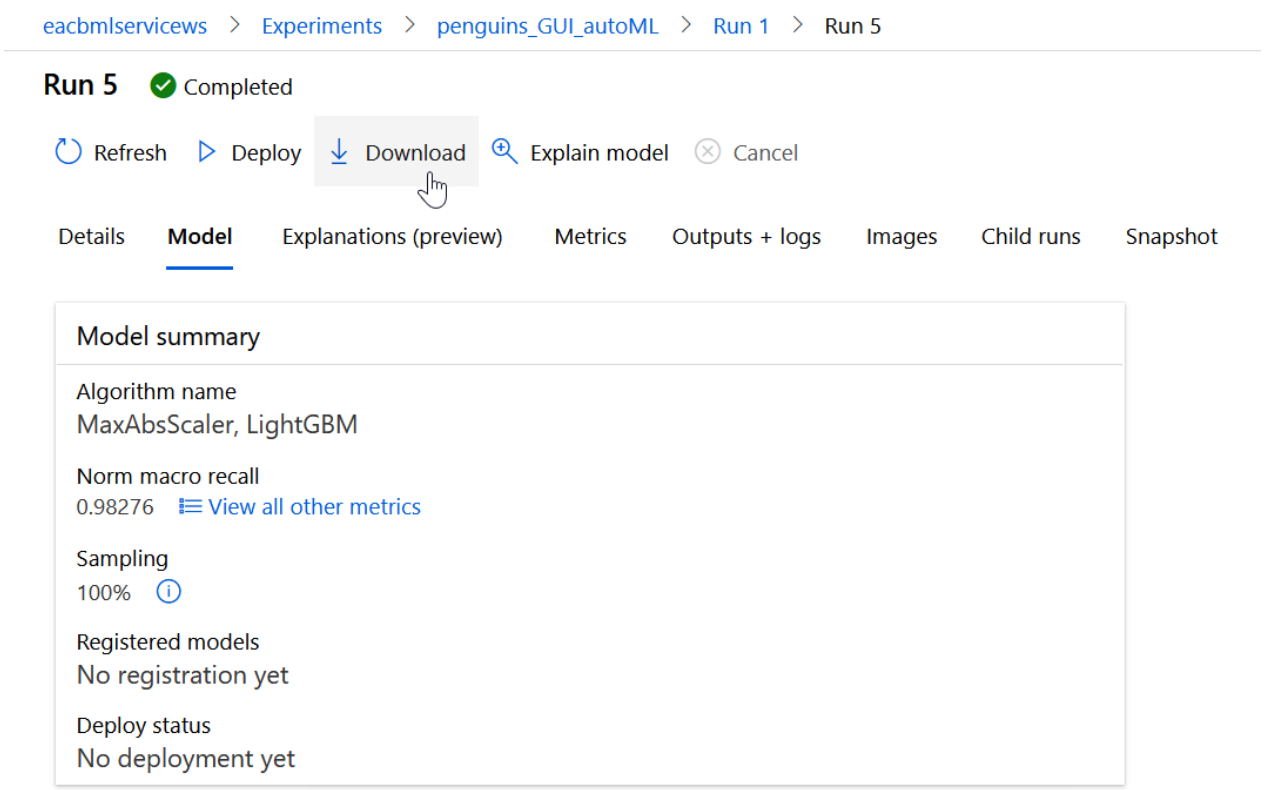


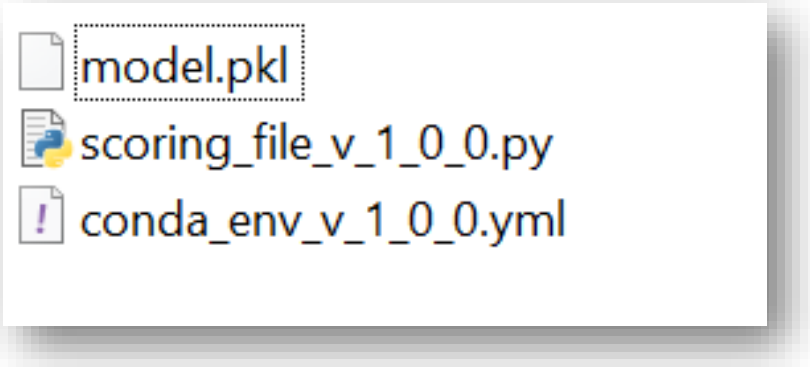
# Explanations (explanation exploration)



# Explanations (summary importance)

# Save the (best) model (GUI *versus* code)




```
best_run, fitted_model = run.get_output()
```

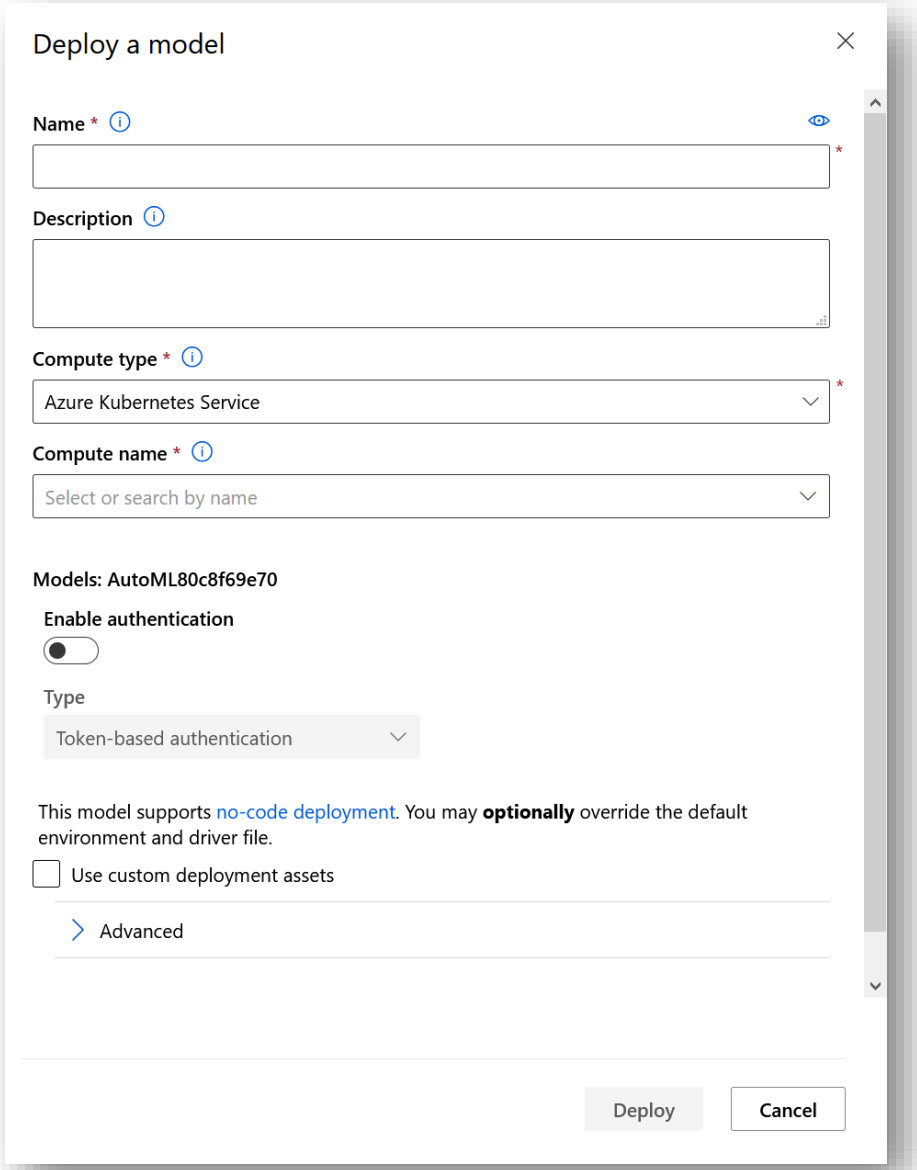
```
from azureml.core.model import Model
```

```
model =
best_run.register_model(model_name='penguins_
class_best_model',
model_path='outputs/model.pkl')
```

```
y_predict =
fitted_model.predict(test_data.to_pandas_data
frame().drop(label, axis=1))
```

```
class_prob =
fitted_model.predict_proba(test_data.to_panda
s_dataframe().drop(label, axis=1))
```

# Model deployment (GUI *versus* code)




Define an inference configuration

- Package dependencies
- Scoring script

Deploy on Azure : ACI or AKS

```
from azureml.core import Webservice
from azureml.core.webservice import AciWebservice
from azureml.exceptions import WebserviceException


service_name = 'penguins-prediction-service'

# Remove any existing service under the same name.
try:
    Webservice(ws, service_name).delete()
except WebserviceException:
    pass

aci_config = AciWebservice.deploy_configuration(cpu_cores=1, memory_gb=1, auth_enabled=True)

service = Model.deploy(workspace=ws,
                       name=service_name,
                       models=[model],
                       inference_config=inference_config,
                       deployment_config=aci_config)

service.wait_for_deployment(show_output=True)
```

AZEO
talents & technology

# CONTEST(S)

AZEO
talents & technology

# GUI (low code) vs SDK (code first), who's best ?

GUI :

- Common mistakes are easier
- Can't modify a run, must redo every steps

SDK :

- Real data preparation with pandas
- Featurization only with enterprise licence
- Choice between local and remote compute (including Azure Databricks)
- Widget in notebooks
- Choose the best model not only on the primary metric
- ONNX model compatibility
- Integration in a more complex pipeline (with hyperdrive for the best model and test)

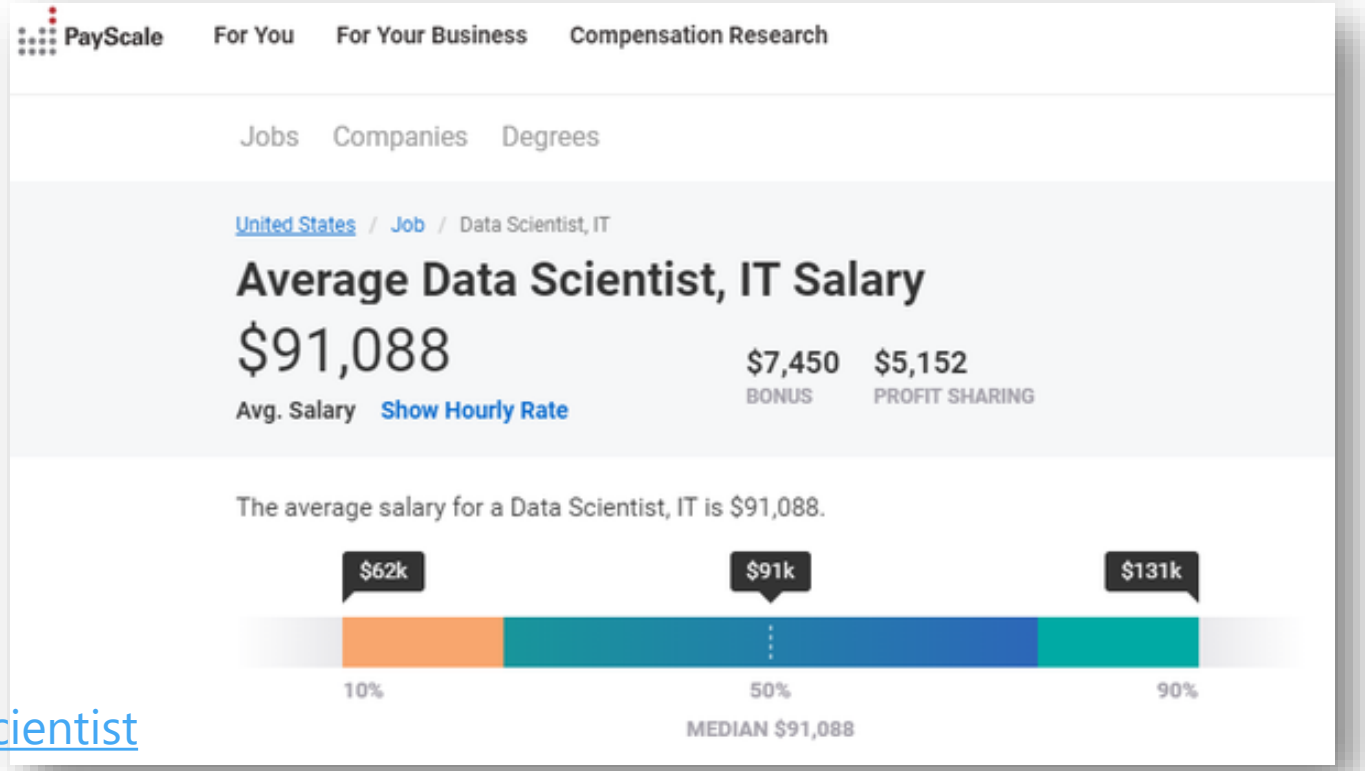
# Comparative pricing

The costs of Azure Machine Learning are related to the VMs used to

- launch notebooks (calculation instance),
- train models (training cluster) or
- deploy models (inference cluster).

Three additional resources will be deployed at additional cost:

- Azure Container Registry Standard account
- Azure Block Blob Storage (general purpose v1)
- Key Vault




https://www.quora.com/What-is-the-salary-of-a-data-scientist

AZEO
talents & technology

# AutoML vs Data Scientist, that is not the good question

This is not an opposition: automated ML is a tool that must be used with discernment.

It is always essential to :

- Understand the business issues and its stakes
- Understanding the data made available
  - Evaluate its quality, improve it if necessary
  - Remove information that is not relevant to the model
  - Transforming certain data to better exploit them
- To know which algorithm families are relevant according to the use case
  - Linear or non-linear models
  - More or less complex models
  - Ensemble (but beware of overlearning)
- Have a complete view of the evaluation metrics
  - Integrate training time and cost
- Interpret the selected

AZEO
talents & technology

# So, do we still need Data Scientists ?

YES!

- To ask the right question for supervised algorithms
- For complex data preparation
- To select features in a non-automated way (based on business criteria)
- For Machine Learning other than supervised: unsupervised, reinforcement learning, transfer, etc.
- To read the interpretation of the models with a critical eye

In conclusion:

Azure ML and the Designer = the *playground* of Citizen Data Scientists

Azure ML and its (Python) SDK = the Data Engineers new best friend !

AZEO
talents & technology

# Webographie

https://azure.microsoft.com/en-us/services/machine-learning-service/

https://docs.microsoft.com/en-us/azure/machine-learning/service/concept-automated-ml

https://azure.microsoft.com/en-us/pricing/details/machine-learning-service/

Forum : https://social.msdn.microsoft.com/Forums/en-US/home?forum=AzureMachineLearningService

AZEO
talents & technology

# Samples notebooks

https://github.com/Azure/MachineLearningNotebooks/tree/master/how-to-use-azureml/automated-machine-learning

https://github.com/Azure/MachineLearningNotebooks/blob/master/tutorials/regression-automl-nyc-taxi-data/regression-automated-ml.ipynb

https://github.com/Azure/MachineLearningNotebooks/blob/master/how-to-use-azureml/automated-machine-learning/forecasting-energy-demand/auto-ml-forecasting-energy-demand.ipynb

R'way

## SIÈGE SOCIAL

52 Avenue André Morizet

92100 Boulogne-Billancourt

Paris – Bordeaux – Lyon - Nantes - Toulouse

AZEO

talents & technology


Paul Péton
Tech Lead Data & AI
MVP AI & Data Platform
paul.peton@azeo.com
06 11 10 22 01

Jean-Pierre Riehl
Directeur de l'Innovation
MVP Data Platform
jean-pierre.riehl@azeo.com
06 62 50 61 24