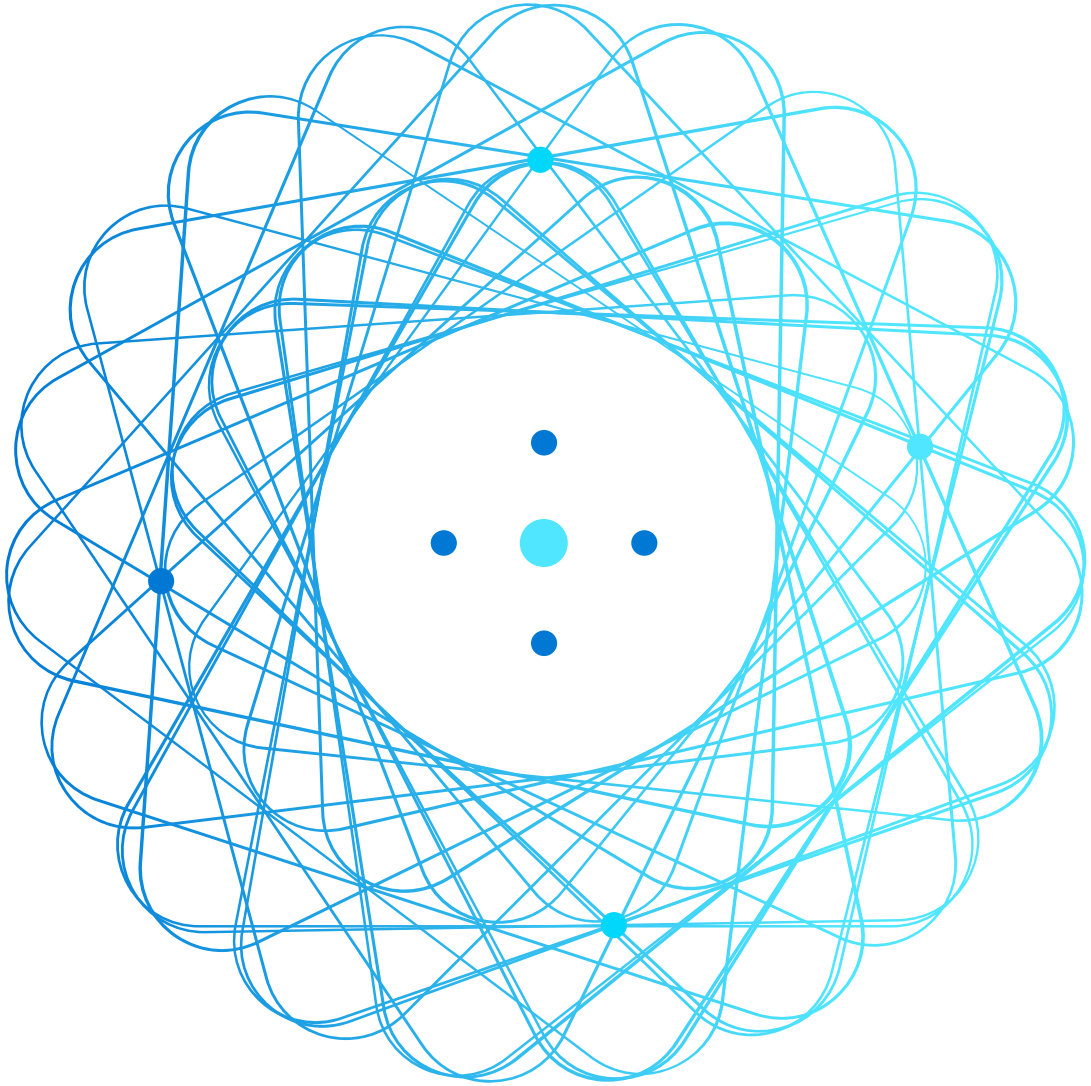
Microsoft Azure

Find the best model with Automated Machine Learning

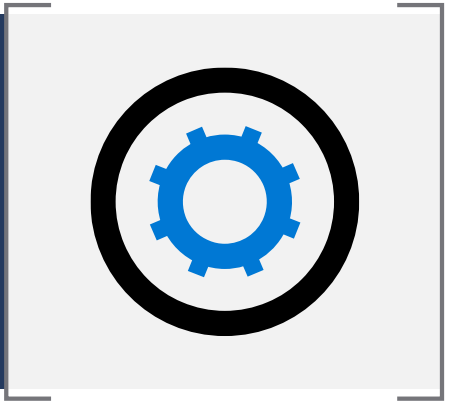


Module Agenda

Explore Automate Machine Learning

Find the best classification model with Automated Machine Learning

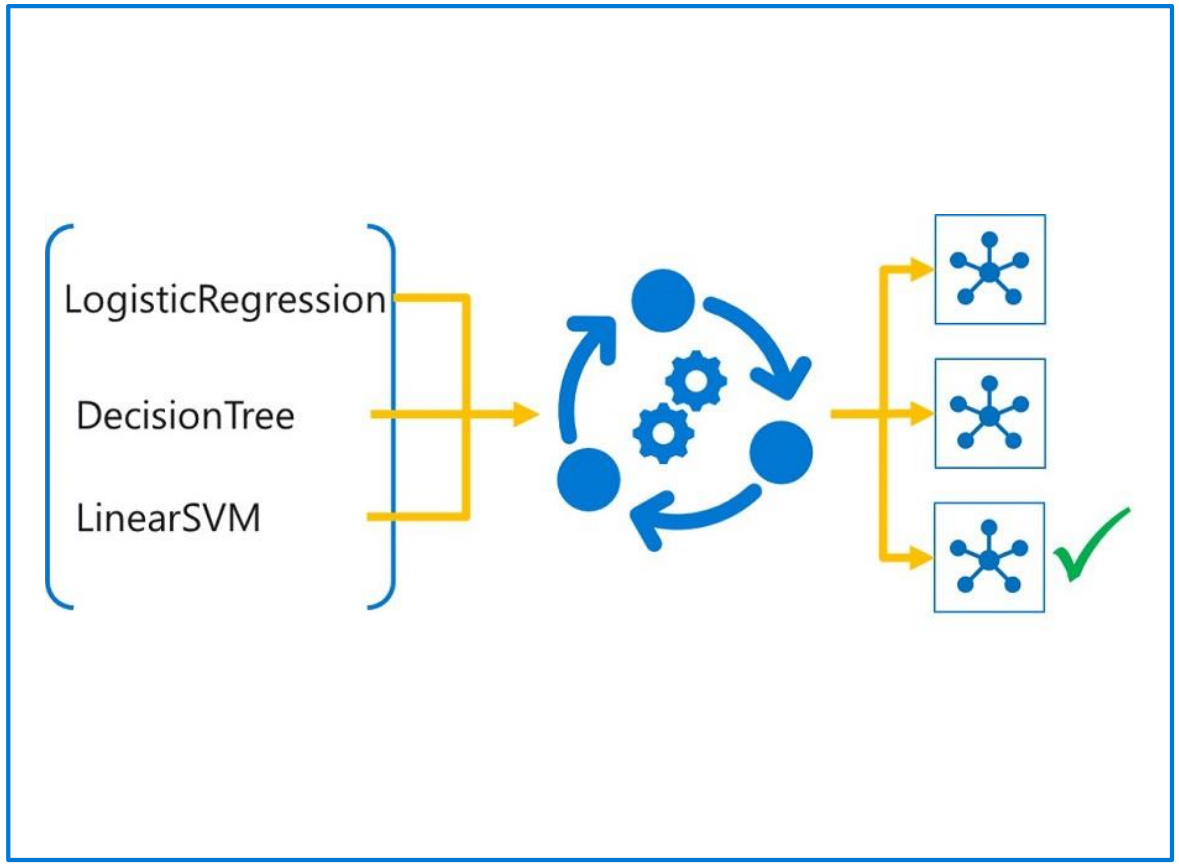
Explore Automate Machine Learning



Explore Automated Machine Learning

Instead of manually having to test and evaluate various configurations to train a machine learning model, you can automate it with automated machine learning or AutoML.

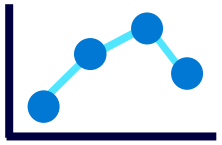
- Train multiple models in parallel, varying preprocessing and algorithm selection.
- Find the "best" model based on a specific performance metric.



Choose a task

Classification

Predict a categorical value.

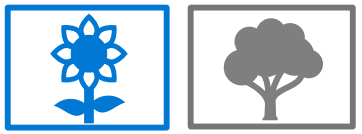


Regression

Predict a numerical value.

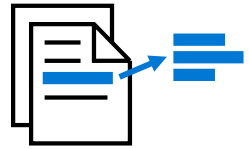
Time-series forecasting

Predict future numerical values based on time-series data.



Computer vision

Classify images or detect objects in images.



Natural language processing (NLP)

Text classification or named entity recognition.

The training data, featurization options, algorithms, and performance metrics will depend on the task you choose.

Knowledge check

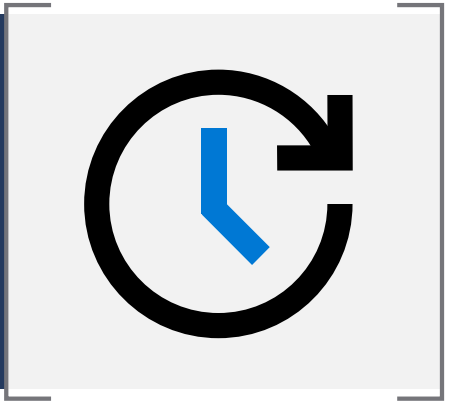
Your marketing team wants to predict whether a customer will churn or not. Which task should you use in Automated Machine Learning?

- ❑ Forecasting
- ❑ Regression
- ☑ Classification

A medical company wants to create an application that can detect abnormalities in X-ray images. Which task should be used in Automated Machine Learning?

- ❑ Forecasting
- ☑ Computer vision
- ❑ Natural language processing

Find the best classification model with Automated Machine Learning



Preprocess data and configure featurization

Before you can run an AutoML experiment, you need to prepare your data.

- Classification requires tabular data.
- Create a **data asset** in Azure Machine Learning.
- Create a **MLTable** data asset: Store your data in a folder together with a MLTable file.

Python

```
from azure.ai.ml.constants import AssetTypes
from azure.ai.ml import Input

my_training_data_input = Input(type=AssetTypes.MLTABLE, path="azureml:input-data-automl:1")
```

Understand scaling and normalization

AutoML applies scaling and normalization to numeric data automatically, helping prevent any large-scale features from dominating training.

During an AutoML experiment, multiple scaling or normalization techniques will be applied.

Configure optional featurization

You can choose to have AutoML apply preprocessing transformations, such as:

Missing value imputation to eliminate nulls in the training dataset

Categorical encoding to convert categorical features to numeric indicators

Dropping high-cardinality features, such as record IDs

Feature engineering (for example, deriving individual date parts from DateTime features)

Run an Automated Machine Learning experiment

The algorithms AutoML uses will depend on the task you specify.

When you want to train a classification model, AutoML will choose from a list of classification algorithms:

Logistic Regression

Light Gradient Boosting Machine (GBM)

Decision Tree

Random Forest

Naive Bayes

Linear Support Vector Machine (SVM)

XGBoost

And others...

Restrict algorithm selection

- By default, AutoML will randomly select from the full range of algorithms for the specified task.
- You can choose to block individual algorithms from being selected; which can be useful if you know that your data isn't suited to a particular type of algorithm.
- You also may want to block certain algorithms if you have to comply with a policy that restricts the type of machine learning algorithms you can use in your organization.

Configure an AutoML experiment (1/2)

When you use the Python SDK (v2) to configure an AutoML experiment or job, you configure the experiment using the *automl* class.

Specify the primary metric: The "best" model is based on the **primary_metric**

Set the limits: To minimize costs and time spent on training, you can set limits to an AutoML experiment by using *set_limits()*

Set the training properties: AutoML will try various combinations of featurization and algorithms to train a machine learning model

Configure an AutoML experiment (2/2)

To configure an AutoML experiment for classification, use the **automl.classification** function:

Python

```
from azure.ai.ml import automl

classification_job = automl.classification(
                compute="aml-cluster",
                experiment_name="auto-ml-class-dev",
                training_data=my_training_data_input,
                target_column_name="Diabetic",
                primary_metric="accuracy",
                n_cross_validations=5,
                enable_model_explainability=True)
```

Set the limits

There are several options to set limits to an AutoML experiment:

- **timeout_minutes**: Number of minutes after which the complete AutoML experiment is terminated.
- **trial_timeout_minutes**: Maximum number of minutes one trial can take.
- **max_trials**: Maximum number of trials, or models that will be trained.
- **enable_early_termination**: Whether to end the experiment if the score isn't improving in the short term.

Python

```
classification_job.set_limits(
        timeout_minutes=60,
        trial_timeout_minutes=20,
        max_trials=5,
        enable_early_termination=True)
```

Submit an AutoML experiment

You can submit an AutoML experiment with the following code:

Python

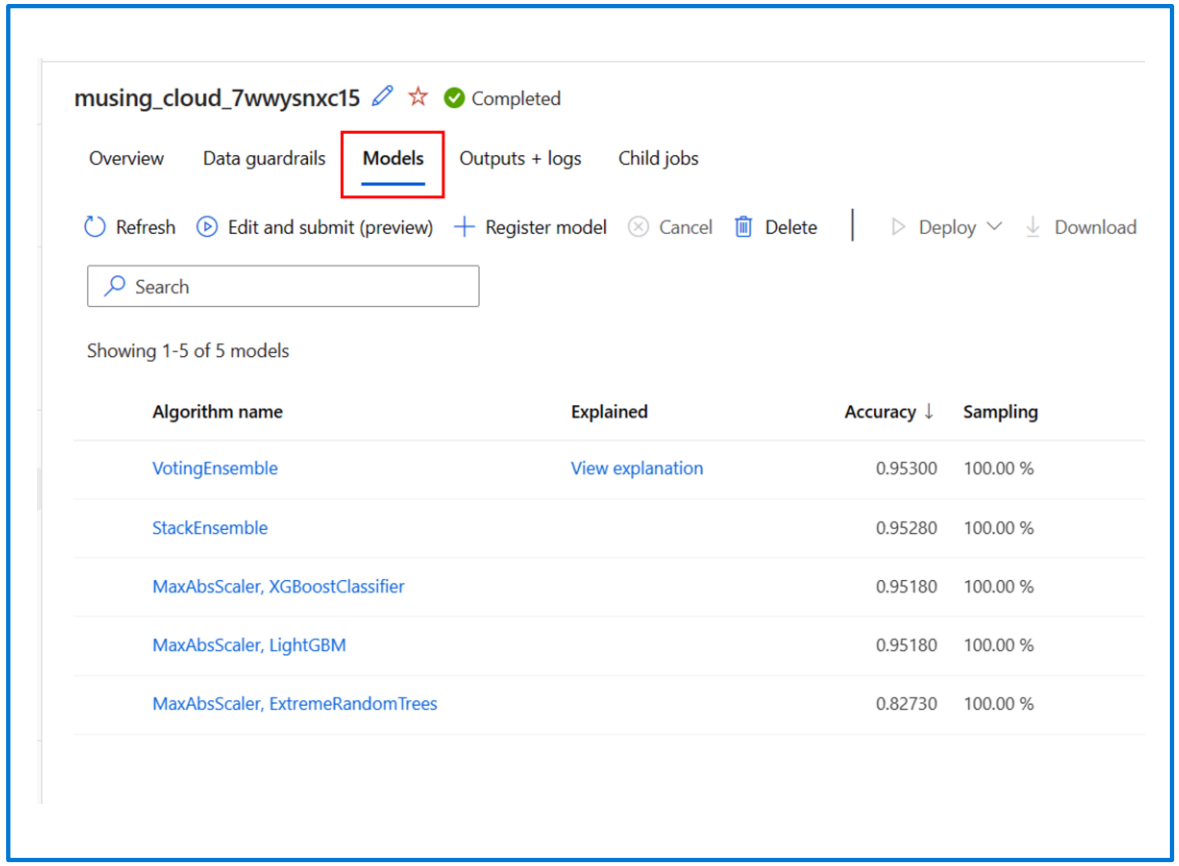
```
returned_job = ml_client.jobs.create_or_update(classification_job)
```

The experiment will consist of child jobs:

- Featurization is performed in a child job.
- Each model is trained in a separate child job.

Evaluate and compare models

- In the Azure Machine Learning studio, you can select an AutoML experiment to explore its details.
- On the **Overview** page of the AutoML experiment run, you can review the input data asset and the summary of the best model. To explore all models that have been trained, you can select the **Models** tab:



Explore preprocessing steps

When you've enabled featurization for your AutoML experiment, data guardrails will automatically be applied too.

The three data guardrails that are supported for classification models are:

- Class balancing detection
- Missing feature values imputation
- High cardinality feature detection

Each of these data guardrails will show one of three possible states:

- **Passed:** No problems were detected and no action is required
- **Done:** Changes were applied to your data. You should review the changes AutoML has made to your data
- **Alerted:** An issue was detected but couldn't be fixed. You should review the data to fix the issue

Retrieve the best run and its model

When you're reviewing the models in AutoML, you can easily identify the best run based on the primary metric you specified.

In the **Models** tab of the AutoML experiment, you can **edit the columns** if you want to show other metrics in the same overview.

To explore a model even further, you can generate explanations for each model that has been trained.

Exercise - Find the best classification model

In this exercise, you will:

Task 1: Prepare the data

Task 2: Configure an Automated Machine Learning experiment

Task 3: Run an Automated Machine Learning job

Instructions

Follow these instructions to complete the exercise:

1. View the exercise repo at https://microsoftlearning.github.io/mslearn-azure-ml/.
2. Complete the **Find the best classification model with Automated Machine Learning** exercise.

Knowledge check

A data scientist wants to use automated machine learning to find the model with the best AUC_weighted metric. What parameter of the classification function should be configured?

- ❑ task
- ❑ target_column_name
- ☑ primary_metric

A data scientist has preprocessed the training data and wants to use automated machine learning to quickly iterate through various algorithms. The data shouldn't be changed. What should be the featurization mode to train a model without letting automated machine learning make changes to the data?

- ❑ auto
- ❑ custom
- ☑ off

Microsoft Azure