What is automated machine learning (AutoML)?

Article • 10/19/2022 • 10 minutes to read

APPLIES TO: Python SDK azure-ai-ml **v2 (current)**

> - v1

> - v2 (current version)

Automated machine learning, also referred to as automated ML or AutoML, is the process of automating the time-consuming, iterative tasks of machine learning model development. It allows data scientists, analysts, and developers to build ML models with high scale, efficiency, and productivity all while sustaining model quality. Automated ML in Azure Machine Learning is based on a breakthrough from our Microsoft Research division .

- For code-experienced customers, Azure Machine Learning Python SDK . Get started with Tutorial: Train an object detection model (preview) with AutoML and Python.

How does AutoML work?

During training, Azure Machine Learning creates a number of pipelines in parallel that try different algorithms and parameters for you. The service iterates through ML algorithms paired with feature selections, where each iteration produces a model with a training score. The better the score for the metric you want to optimize for, the better the model is considered to "fit" your data. It will stop once it hits the exit criteria defined in the experiment.

Using **Azure Machine Learning**, you can design and run your automated ML training experiments with these steps:

1. **Identify the ML problem** to be solved: classification, forecasting, regression, computer vision or NLP.

2. **Choose whether you want to a code-first experience or a no-code studio web experience**: Users who prefer a code-first experience can use the AzureML SDKv2 or the AzureML CLIv2. Get started with Tutorial: Train an object detection model with AutoML and Python. Users who prefer a limited/no-code experience can use the web interface in Azure Machine Learning studio at https://ml.azure.com . Get started with Tutorial: Create a classification model with automated ML in Azure Machine Learning.

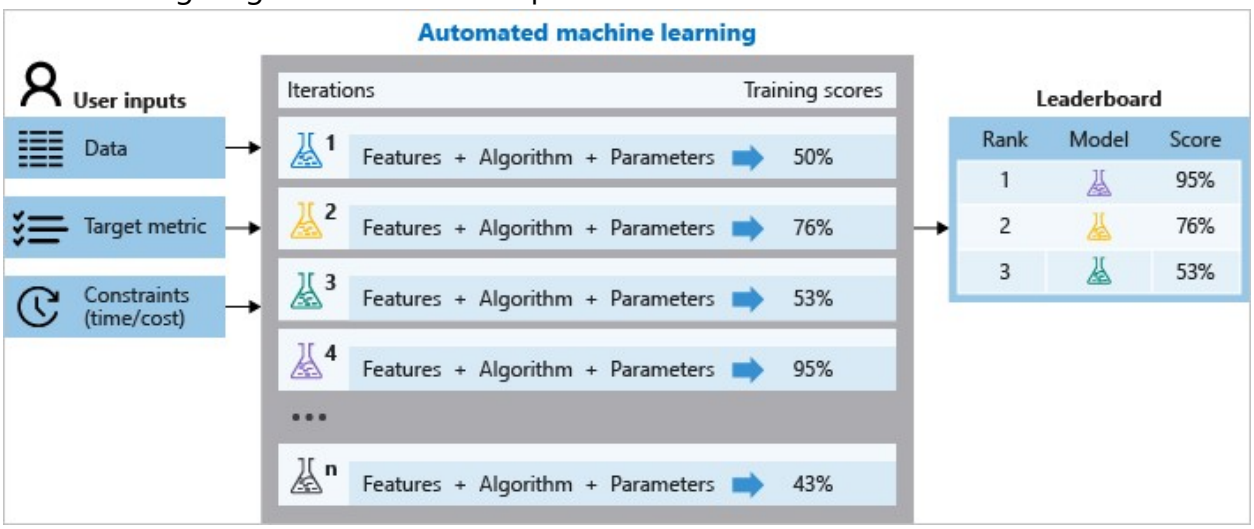
3. **Specify the source of the labeled training data**: You can bring your data to AzureML in many different ways.

4. **Configure the automated machine learning parameters** that determine how many iterations over different models, hyperparameter settings, advanced preprocessing/featurization, and what metrics to look at when determining the best model.

5. **Submit the training job.**

6. **Review the results**

The following diagram illustrates this process.



You can also inspect the logged job information, which contains metrics gathered during the job. The training job produces a Python serialized object (`.pkl` file) that contains the model and data preprocessing.

While model building is automated, you can also learn how important or relevant features are to the generated models.

When to use AutoML: classification, regression, forecasting, computer vision & NLP

Apply automated ML when you want Azure Machine Learning to train and tune a model for you using the target metric you specify. Automated ML democratizes the machine learning model development process, and empowers its users, no matter their data science expertise, to identify an end-to-end machine learning pipeline for any problem.

ML professionals and developers across industries can use automated ML to:

- Implement ML solutions without extensive programming knowledge
- Save time and resources
- Leverage data science best practices
- Provide agile problem-solving

Classification

Classification is a type of supervised learning in which models learn using training data, and apply those learnings to new data. Azure Machine Learning offers featurizations specifically for these tasks, such as deep neural network text featurizers for classification. Learn more about featurization options. You can also find the list of algorithms supported by AutoML here.

The main goal of classification models is to predict which categories new data will fall into based on learnings from its training data. Common classification examples include fraud detection, handwriting recognition, and object detection.

See an example of classification and automated machine learning in this Python notebook: Bank Marketing .

Regression

Similar to classification, regression tasks are also a common supervised learning task. AzureML offers featurization specific to regression problems. Learn more about featurization options. You can also find the list of algorithms supported by AutoML here.

Different from classification where predicted output values are categorical, regression models predict numerical output values based on independent predictors. In regression,

the objective is to help establish the relationship among those independent predictor variables by estimating how one variable impacts the others. For example, automobile price based on features like, gas mileage, safety rating, etc.

See an example of regression and automated machine learning for predictions in these Python notebooks: Hardware Performance .

Time-series forecasting

Building forecasts is an integral part of any business, whether it's revenue, inventory, sales, or customer demand. You can use automated ML to combine techniques and approaches and get a recommended, high-quality time-series forecast. You can find the list of algorithms supported by AutoML here.

An automated time-series experiment is treated as a multivariate regression problem. Past time-series values are "pivoted" to become additional dimensions for the regressor together with other predictors. This approach, unlike classical time series methods, has an advantage of naturally incorporating multiple contextual variables and their relationship to one another during training. Automated ML learns a single, but often internally branched model for all items in the dataset and prediction horizons. More data is thus available to estimate model parameters and generalization to unseen series becomes possible.

Advanced forecasting configuration includes:

- holiday detection and featurization
- time-series and DNN learners (Auto-ARIMA, Prophet, ForecastTCN)
- many models support through grouping
- rolling-origin cross validation
- configurable lags
- rolling window aggregate features

See an example of forecasting and automated machine learning in this Python notebook: Energy Demand .

Computer vision

Support for computer vision tasks allows you to easily generate models trained on image data for scenarios like image classification and object detection.

With this capability you can:

- Seamlessly integrate with the Azure Machine Learning data labeling capability
- Use labeled data for generating image models
- Optimize model performance by specifying the model algorithm and tuning the hyperparameters.
- Download or deploy the resulting model as a web service in Azure Machine Learning.
- Operationalize at scale, leveraging Azure Machine Learning MLOps and ML Pipelines capabilities.

Authoring AutoML models for vision tasks is supported via the Azure ML Python SDK. The resulting experimentation jobs, models, and outputs can be accessed from the Azure Machine Learning studio UI.

Learn how to set up AutoML training for computer vision models.

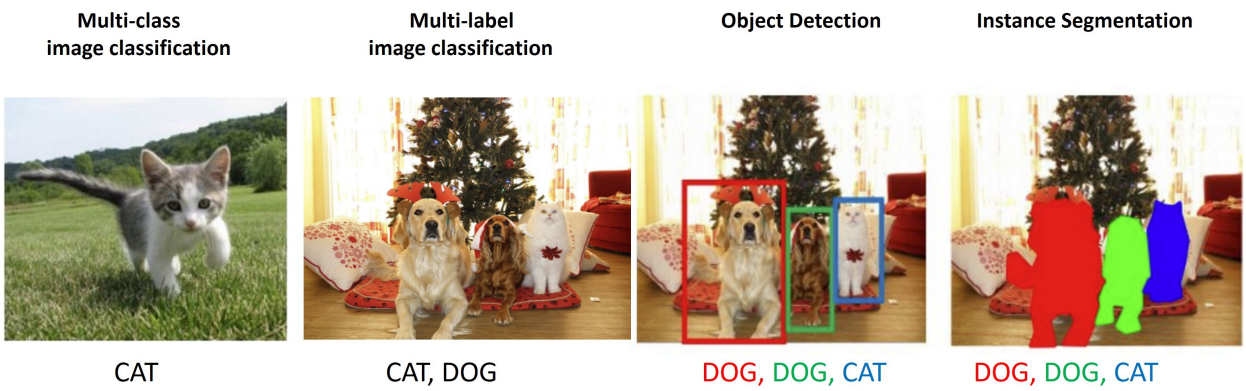


Image from: http://cs231n.stanford.edu/slides/2021/lecture_15.pdf

Automated ML for images supports the following computer vision tasks:

Task	Description
Multi-class image classification	Tasks where an image is classified with only a single label from a set of classes - e.g. each image is classified as either an image of a 'cat' or a 'dog' or a 'duck'
Multi-label image classification	Tasks where an image could have one or more labels from a set of labels - e.g. an image could be labeled with both 'cat' and 'dog'
Object detection	Tasks to identify objects in an image and locate each object with a bounding box e.g. locate all dogs and cats in an image and draw a bounding box around each.

Task	Description
Instance segmentation	Tasks to identify objects in an image at the pixel level, drawing a polygon around each object in the image.

Natural language processing: NLP

Support for natural language processing (NLP) tasks in automated ML allows you to easily generate models trained on text data for text classification and named entity recognition scenarios. Authoring automated ML trained NLP models is supported via the Azure Machine Learning Python SDK. The resulting experimentation jobs, models, and outputs can be accessed from the Azure Machine Learning studio UI.

The NLP capability supports:

- End-to-end deep neural network NLP training with the latest pre-trained BERT models
- Seamless integration with Azure Machine Learning data labeling
- Use labeled data for generating NLP models
- Multi-lingual support with 104 languages
- Distributed training with Horovod

Learn how to set up AutoML training for NLP models.

Training, validation and test data

With automated ML you provide the **training data** to train ML models, and you can specify what type of model validation to perform. Automated ML performs model validation as part of training. That is, automated ML uses **validation data** to tune model hyperparameters based on the applied algorithm to find the combination that best fits the training data. However, the same validation data is used for each iteration of tuning, which introduces model evaluation bias since the model continues to improve and fit to the validation data.

To help confirm that such bias isn't applied to the final recommended model, automated ML supports the use of **test data** to evaluate the final model that automated ML recommends at the end of your experiment. When you provide test data as part of your AutoML experiment configuration, this recommended model is tested by default at the end of your experiment (preview).

Important

Testing your models with a test dataset to evaluate generated models is a preview feature. This capability is an experimental preview feature, and may change at any time.

Learn how to configure AutoML experiments to use test data (preview) with the SDK or with the Azure Machine Learning studio.

Feature engineering

Feature engineering is the process of using domain knowledge of the data to create features that help ML algorithms learn better. In Azure Machine Learning, scaling and normalization techniques are applied to facilitate feature engineering. Collectively, these techniques and feature engineering are referred to as featurization.

For automated machine learning experiments, featurization is applied automatically, but can also be customized based on your data. Learn more about what featurization is included and how AutoML helps prevent over-fitting and imbalanced data in your models.

Note

Automated machine learning featurization steps (feature normalization, handling missing data, converting text to numeric, etc.) become part of the underlying model. When using the model for predictions, the same featurization steps applied during training are applied to your input data automatically.

Customize featurization

Additional feature engineering techniques such as, encoding and transforms are also available.

Enable this setting with:

- Azure Machine Learning studio: Enable **Automatic featurization** in the **View additional configuration** section with these steps.
- Python SDK: Specify featurization in your AutoML Job object. Learn more about enabling featurization.

Ensemble models

Automated machine learning supports ensemble models, which are enabled by default. Ensemble learning improves machine learning results and predictive performance by combining multiple models as opposed to using single models. The ensemble iterations appear as the final iterations of your job. Automated machine learning uses both voting and stacking ensemble methods for combining models:

- **Voting**: predicts based on the weighted average of predicted class probabilities (for classification tasks) or predicted regression targets (for regression tasks).
- **Stacking**: stacking combines heterogenous models and trains a meta-model based on the output from the individual models. The current default meta-models are LogisticRegression for classification tasks and ElasticNet for regression/forecasting tasks.

The Caruana ensemble selection algorithm with sorted ensemble initialization is used to decide which models to use within the ensemble. At a high level, this algorithm initializes the ensemble with up to five models with the best individual scores, and verifies that these models are within 5% threshold of the best score to avoid a poor initial ensemble. Then for each ensemble iteration, a new model is added to the existing ensemble and the resulting score is calculated. If a new model improved the existing ensemble score, the ensemble is updated to include the new model.

See the AutoML package for changing default ensemble settings in automated machine learning.

AutoML & ONNX

With Azure Machine Learning, you can use automated ML to build a Python model and have it converted to the ONNX format. Once the models are in the ONNX format, they can be run on a variety of platforms and devices. Learn more about accelerating ML models with ONNX.

See how to convert to ONNX format in this Jupyter notebook example . Learn which algorithms are supported in ONNX.

The ONNX runtime also supports C#, so you can use the model built automatically in your C# apps without any need for recoding or any of the network latencies that REST

endpoints introduce. Learn more about using an AutoML ONNX model in a .NET application with ML.NET and inferencing ONNX models with the ONNX runtime C# API .

Next steps

There are multiple resources to get you up and running with AutoML.

Tutorials/ how-tos

Tutorials are end-to-end introductory examples of AutoML scenarios.

- **For a code first experience**, follow the Tutorial: Train an object detection model with AutoML and Python
- **For a low or no-code experience**, see the Tutorial: Train a classification model with no-code AutoML in Azure Machine Learning studio.

How-to articles provide additional detail into what functionality automated ML offers. For example,

- Configure the settings for automatic training experiments
  - Without code in the Azure Machine Learning studio.
  - With the Python SDK.
- Learn how to train computer vision models with Python.
- Learn how to view the generated code from your automated ML models.

Jupyter notebook samples

Review detailed code examples and use cases in the [GitHub notebook repository for automated machine learning samples](https://github.com/Azure/azureml-examples/tree/main/sdk/python/jobs/automl-standalone-jobs .

Python SDK reference

Deepen your expertise of SDK design patterns and class specifications with the AutoML Job class reference documentation.

Note

Automated machine learning capabilities are also available in other Microsoft solutions such as, ML.NET, HDInsight, Power BI and SQL Server