

Framework

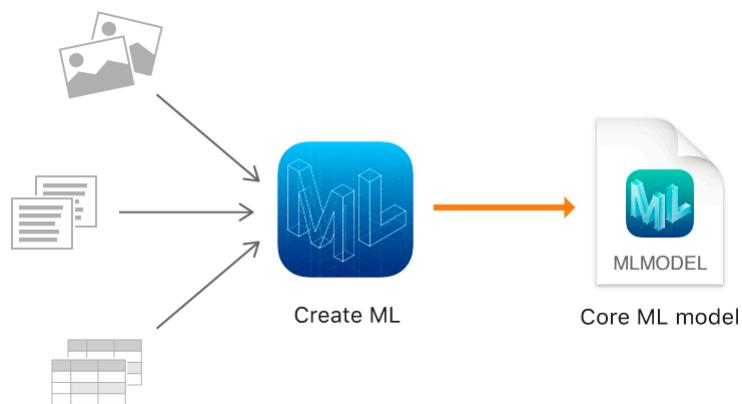
Create ML

Create machine learning models for use in your app.

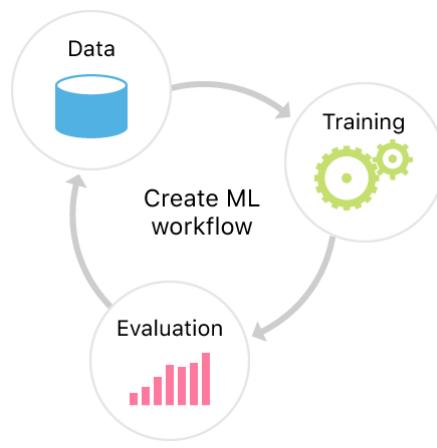
iOS 15.0+ | iPadOS 15.0+ | Mac Catalyst 15.0+ | macOS 10.14+ | tvOS 16.0+ | visionOS 1.0+

Overview

Use Create ML with familiar tools like Swift and macOS playgrounds to create and train custom machine learning models on your Mac. You can train models to perform tasks like recognizing images, extracting meaning from text, or finding relationships between numerical values.



You train a model to recognize patterns by showing it representative samples. For example, you can train a model to recognize dogs by showing it lots of images of different dogs. After you've trained the model, you test it out on data it hasn't seen before, and evaluate how well it performed the task. When the model is performing well enough, you're ready to integrate it into your app using [Core ML](#).



Create ML leverages the machine learning infrastructure built in to Apple products like Photos and Siri. This means your image classification and natural language models are smaller and take much less time to train.

Topics

Image models

Creating an Image Classifier Model

Train a machine learning model to classify images, and add it to your Core ML app.

`struct MLImageClassifier`

A model you train to classify images.

`struct MLObjectDetector`

A model you train to classify one or more objects within an image.

`struct MLHandPoseClassifier`

A task that creates a hand pose classification model by training with images of people's hands that you provide.

Video models

Creating an Action Classifier Model

Train a machine learning model to recognize a person's body movements.

Detecting human actions in a live video feed

Identify body movements by sending a person's pose data from a series of video frames to an action-classification model.

```
struct MLActionClassifier
```

A model you train with videos to classify a person's body movements.

```
struct MLHandActionClassifier
```

A task that creates a hand action classification model by training with videos of people's hand movements that you provide.

```
struct MLStyleTransfer
```

A model you train to apply an image's style to other images or videos.

Text models

📄 Creating a text classifier model

Train a machine learning model to classify natural language text.

📄 Creating a word tagger model

Train a machine learning model to tag individual words in natural language text.

```
struct MLTextClassifier
```

A model you train to classify natural language text.

```
struct MLWordTagger
```

A word-tagging model you train to classify natural language text at the word level.

```
struct MLGazetteer
```

A collection of terms and their labels, which augments a tagger that analyzes natural language text.

```
struct MLWordEmbedding
```

A map of strings in a vector space that enable your app to find similar strings by looking at a string's neighbors.

Sound models

```
struct MLSoundClassifier
```

A machine learning model you train with audio files to recognize and identify sounds on a device.

Motion models

```
struct MLActivityClassifier
```

A model you train to classify motion sensor data.

Tabular models

{ } Creating a model from tabular data

Train a machine learning model by using Core ML to import and manage tabular data.

```
enum MLClassifier
```

A model you train to classify data into discrete categories.

```
enum MLRegressor
```

A model you train to estimate continuous values.

```
struct MLRecommender
```

A model you train to make recommendations based on item similarity, grouping, and, optionally, item ratings.

Tabular data

```
struct MLDataTable
```

A table of data for training or evaluating a machine learning model.

```
enum MLDataTableValue
```

The value of a cell in a data table.

:≡ Data visualizations

Render images of data tables and columns in a playground.

Model accuracy

📄 Improving Your Model's Accuracy

Use metrics to tune the performance of your machine learning model.

```
struct MLClassifierMetrics
```

Metrics you use to evaluate a classifier's performance.

```
struct MLRegressorMetrics
```

Metrics you use to evaluate a regressor's performance.

```
struct MLWordTaggerMetrics
```

Metrics you use to evaluate a word tagger's performance.

```
struct MLRecommenderMetrics
```

Metrics you use to evaluate a recommender's performance.

```
struct MLObjectDetectorMetrics
```

Metrics you use to evaluate an object detector's performance.

Model training Control

```
class MLJob
```

The representation of a model's asynchronous training session you use to monitor the session's progress or terminate its execution.

```
class MLTrainingSession
```

The current state of a model's asynchronous training session.

```
struct MLTrainingSessionParameters
```

The configuration settings for a training session.

```
struct MLCheckpoint
```

The state of a model's asynchronous training session at a specific point in time during the feature extraction or training phase.

Supporting types

```
enum MLCreateError
```

The errors Create ML throws while performing various operations, such as training models, making predictions, writing models to a file system, and so on.

```
struct MLModelMetadata
```

Information about a model that's stored in a Core ML model file.

```
enum MLSplitStrategy
```

Data partitioning approaches, typically for creating a validation dataset from a training dataset.

Articles

☰ Data visualizations

Render images of data tables and columns in a playground.

{ } Detecting human actions in a live video feed

Identify body movements by sending a person's pose data from a series of video frames to an action-classification model.

📄 Gathering Training Videos for an Action Classifier

Collect quality example videos that effectively train action classifiers.

Functions

~~func show(_:_)~~

Generates a streaming visualization of the untyped column.

Deprecated

~~func show(_:_:_:_)~~

Generates a streaming plot visualization of the two untyped columns.

Deprecated

Enumerations

enum MLBoundingBoxAnchor

A location within a bounding box that an annotation's coordinates use as their reference point.

enum MLBoundingBoxCoordinatesOrigin

The location within an image that an annotation's coordinates use as their origin.

enum MLBoundingBoxUnits

The units a bounding box annotation uses to define its position and size.