

Amazon SageMaker

Developer Guide

▶ What Is Amazon SageMaker?

▶ Get Started

▶ SageMaker Studio

▶ Notebook Instances

▶ Autopilot: Automated ML

▶ Label Data

▶ Prepare and Analyze Datasets

▶ Process Data

▶ Create, Store, and Share Features

▼ Training

▶ Choose an algorithm

▼ Use built-in algorithms

▶ Common Information

▶ BlazingText

▶ DeepAR Forecasting

▶ Factorization Machines

▶ Image Classification Algorithm

▶ IP Insights

▶ K-Means Algorithm

▶ K-Nearest Neighbors (k-NN) Algorithm

▶ Latent Dirichlet Allocation (LDA)

▶ Linear Learner Algorithm

Use Amazon SageMaker Built-in Algorithms

[PDF](#) | [Kindle](#) | [RSS](#)

Amazon SageMaker provides a suite of built-in algorithms to help data scientists and machine learning practitioners get started on training and deploying machine learning models quickly. For someone that is new to SageMaker, choosing the right algorithm for your particular use case can be a challenging task. The following table provides a quick cheat sheet that shows how you can start with an example problem or use case and find an appropriate built-in algorithm offered by SageMaker that is valid for that problem type. Additional guidance organized by learning paradigms (supervised and unsupervised) and important data domains (text and images) is provided in the sections following the table.

Table: Mapping use cases to built-in algorithms

| Example problems and use cases | Learning paradigm or domain | Problem types | Data input format | Built-in algorithms |
|--|-----------------------------|--|-------------------|---|
| Predict if an item belongs to a category: an email spam filter | Supervised Learning | Binary/multi-class classification | Tabular | Factorization Machines Algorithm, K-Nearest Neighbors (k-NN) Algorithm, Linear Learner Algorithm, XGBoost Algorithm |
| Predict a numeric/continuous value: estimate the value of a house | | Regression | Tabular | Factorization Machines Algorithm, K-Nearest Neighbors (k-NN) Algorithm, Linear Learner Algorithm, XGBoost Algorithm |
| Based on historical data for a behavior, predict future behavior: predict sales on a new product based on previous sales data. | | Time-series forecasting | Tabular | DeepAR Forecasting Algorithm |
| Drop those columns from a dataset that have a weak relation with the label/target variable: the color of a car when predicting its mileage. | Unsupervised Learning | Feature engineering: dimensionality reduction | Tabular | Principal Component Analysis (PCA) Algorithm |
| Detect abnormal behavior in application: spot when an IoT sensor is sending abnormal readings | | Anomaly detection | Tabular | Random Cut Forest (RCF) Algorithm |
| Protect your application from suspicious users: detect if an IP address accessing a service might be from a bad actor | | IP anomaly detection | Tabular | IP Insights |
| Improve the data embeddings of the high-dimensional objects: identify duplicate support tickets or find the correct routing based on similarity of text in the tickets | | Embeddings: convert high-dimensional objects into low-dimensional space. | Tabular | Object2Vec Algorithm |
| Group similar objects/data together: find high-, medium-, and low-spending customers from their transaction histories | | Clustering or grouping | Tabular | K-Means Algorithm |
| Organize a set of | | Topic modeling | Text | Latent Dirichlet |

On this page**Supervised learning algorithms**

Unsupervised Learning

Textual Analysis

Image Processing

| | | | | |
|--|------------------|--------------------------------------|-------|--|
| documents into topics (not known in advance): tag a document as belonging to a medical category based on the terms used in the document. | | | | Allocation (LDA) Algorithm, Neural Topic Model (NTM) Algorithm |
| Assign pre-defined categories to documents in a corpus: categorize books in a library into academic disciplines | Textual Analysis | Text classification | Text | BlazingText algorithm |
| Convert text from one language to other: Spanish to English | | Machine translation algorithm | Text | Sequence-to-Sequence Algorithm |
| Summarize a long text corpus: an abstract for a research paper | | Text summarization | Text | Sequence-to-Sequence Algorithm |
| Convert audio files to text: transcribe call center conversations for further analysis | | Speech-to-text | Text | Sequence-to-Sequence Algorithm |
| Label/tag an image based on the content of the image: alerts about adult content in an image | Image Processing | Image and multi-label classification | Image | Image Classification Algorithm |
| Detect people and objects in an image: police review a large photo gallery for a missing person | | Object detection and classification | Image | Object Detection Algorithm |
| Tag every pixel of an image individually with a category: self-driving cars prepare to identify objects in their way | | Computer vision | Image | Semantic Segmentation Algorithm |

For important information about Docker registry paths, data formats, recommended Amazon EC2 instance types, and CloudWatch logs common to all of the built-in algorithms provided by SageMaker, see [Common Information About Built-in Algorithms](#).

The following sections provide additional guidance for the Amazon SageMaker built-in algorithms grouped by the supervised and unsupervised learning paradigms to which they belong. For descriptions of these learning paradigms and their associated problem types, see [Choose an Algorithm](#). Sections are also provided for the SageMaker built-in algorithms available to address two important machine learning domains: textual analysis and image processing.

- [Supervised Learning](#)
- [Unsupervised Learning](#)
- [Textual Analysis](#)
- [Image Processing](#)

Supervised Learning

Amazon SageMaker provides several built-in general purpose algorithms that can be used for either classification or regression problems.

- [Linear Learner Algorithm](#)—learns a linear function for regression or a linear threshold function for classification.
- [Factorization Machines Algorithm](#)—an extension of a linear model that is designed to economically capture interactions between features within high-dimensional sparse datasets.
- [XGBoost Algorithm](#)—implementation of the gradient-boosted trees algorithm that combines an ensemble of estimates from a set of simpler and weaker models.
- [K-Nearest Neighbors \(k-NN\) Algorithm](#)—a non-parametric method that uses the k nearest labeled points to assign a label to a new data point for classification or a predicted target value from the average of the k nearest points for regression.

Amazon SageMaker also provides several built-in supervised learning algorithms that are used for more specialized tasks during feature engineering and forecasting from time series data.

- [Object2Vec Algorithm](#)—a new highly customizable multi-purpose algorithm used for feature engineering. It can learn low-dimensional dense embeddings of high-dimensional objects to produce features that improve training efficiencies for downstream models. While this is a supervised algorithm, as it requires labeled data for training, there are many scenarios in which the relationship labels can be obtained purely from natural clusterings in data, without any explicit human annotation.
- [DeepAR Forecasting Algorithm](#)—a supervised learning algorithm for forecasting scalar (one-dimensional) time series using recurrent neural networks (RNN).

Unsupervised Learning

Amazon SageMaker provides several built-in algorithms that can be used for a variety of unsupervised learning tasks such as clustering, dimension reduction, pattern recognition, and anomaly detection.

- [Principal Component Analysis \(PCA\) Algorithm](#)—reduces the dimensionality (number of features) within a dataset by projecting data points onto the first few principal components. The objective is to retain as much information or variation as possible. For mathematicians, principal components are eigenvectors of the data's covariance matrix.
- [K-Means Algorithm](#)—finds discrete groupings within data, where members of a group are as similar as possible to one another and as different as possible from members of other groups.
- [IP Insights](#)—learns the usage patterns for IPv4 addresses. It is designed to capture associations between IPv4 addresses and various entities, such as user IDs or account numbers.
- [Random Cut Forest \(RCF\) Algorithm](#)—detects anomalous data points within a data set that diverge from otherwise well-structured or patterned data.

Textual Analysis

SageMaker provides algorithms that are tailored to the analysis of textual documents used in natural language processing, document classification or summarization, topic modeling or classification, and language transcription or translation.

- [BlazingText algorithm](#)—a highly optimized implementation of the Word2vec and text classification algorithms that scale to large datasets easily. It is useful for many downstream natural language processing (NLP) tasks.
- [Sequence-to-Sequence Algorithm](#)—a supervised algorithm commonly used for neural machine translation.
- [Latent Dirichlet Allocation \(LDA\) Algorithm](#)—an algorithm suitable for determining topics in a set of documents. It is an *unsupervised algorithm*, which means that it doesn't use example data with answers during training.
- [Neural Topic Model \(NTM\) Algorithm](#)—another unsupervised technique for determining topics in a set of documents, using a neural network approach.

Image Processing

SageMaker also provides image processing algorithms that are used for image classification, object detection, and computer vision.

- [Image Classification Algorithm](#)—uses example data with answers (referred to as a *supervised algorithm*). Use this algorithm to classify images.
- [Semantic Segmentation Algorithm](#)—provides a fine-grained, pixel-level approach to developing computer vision applications.
- [Object Detection Algorithm](#)—detects and classifies objects in images using a single deep neural network. It is a supervised learning algorithm that takes images as input and identifies all instances of objects within the image scene.

Topics

- [Common Information About Built-in Algorithms](#)
- [BlazingText algorithm](#)
- [DeepAR Forecasting Algorithm](#)
- [Factorization Machines Algorithm](#)
- [Image Classification Algorithm](#)
- [IP Insights](#)
- [K-Means Algorithm](#)
- [K-Nearest Neighbors \(k-NN\) Algorithm](#)
- [Latent Dirichlet Allocation \(LDA\) Algorithm](#)
- [Linear Learner Algorithm](#)
- [Neural Topic Model \(NTM\) Algorithm](#)
- [Object2Vec Algorithm](#)
- [Object Detection Algorithm](#)
- [Principal Component Analysis \(PCA\) Algorithm](#)
- [Random Cut Forest \(RCF\) Algorithm](#)
- [Semantic Segmentation Algorithm](#)
- [Sequence-to-Sequence Algorithm](#)
- [XGBoost Algorithm](#)



Did this page help you?

Yes

No

Provide feedback

Edit this page on GitHub [Edit this page on GitHub](#)

Previous topic: [Choose an algorithm](#)

Next topic: [Common Information](#)

Need help?

- [Try the forums](#)
- [Connect with an AWS IQ expert](#)

