

Recommendations for choosing the right data preparation tool in SageMaker AI

Data preparation in machine learning refers to the process of collecting, preprocessing, and organizing raw data to make it suitable for analysis and modeling. This step ensures that the data is in a format from which machine learning algorithms can effectively learn. Data preparation tasks may include handling missing values, removing outliers, scaling features, encoding categorical variables, assessing potential biases and taking steps to mitigate them, splitting data into training and testing sets, labeling, and other necessary transformations to optimize the quality and usability of the data for subsequent machine learning tasks.

Choose a feature

There are 3 primary use cases for *data preparation* with Amazon SageMaker AI. Choose the [use case](#) that aligns with your requirements, and then refer to the corresponding [recommended feature](#).

Use cases

The following are the primary uses cases when performing data preparation for Machine Learning.

- **Use case 1:** For those who prefer a visual interface, SageMaker AI provides ways to explore, prepare, and engineer features for model training through a point-and-click environment.
- **Use case 2:** For users comfortable with coding who want more flexibility and control over data preparation, SageMaker AI integrates tools into its coding environments for exploration, transformations, and feature engineering.
- **Use case 3:** For users focused on scalable data preparation, SageMaker AI offers serverless capabilities that leverage the Hadoop/Spark ecosystem for distributed processing of big data.

Recommended features

The following table outlines the key considerations and tradeoffs for the SageMaker AI features related to each data preparation use case for machine learning. To get started, identify the use case that aligns to your requirements and navigate to its recommended SageMaker AI feature.

Descriptor	Use case 1	Use case 2	Use case 3
SageMaker AI feature	Data Wrangler within Amazon SageMaker Canvas	Data preparation with SQL in Studio	Prepare data using EMR Serverless applications in Studio
Description	SageMaker Canvas is a visual low-code environment for building, training, and deploying machine learning models in SageMaker AI. Its integrated Data Wrangler tool allows users to combine, transform, and clean datasets through point-and-click interactions.	The SQL extension in Studio allows users to connect to Amazon Redshift, Snowflake, Athena, and Amazon S3 to author ad-hoc SQL queries, and preview results in JupyterLab notebooks. The output of these queries can be manipulated using Python and Pandas for additional processing, visualization, and transformation into formats usable for machine learning model development.	The integration between EMR Serverless and Amazon SageMaker Studio provides a scalable serverless environment for large-scale data preparation for machine learning using open-source frameworks such as Apache Spark and Apache Hive. Users can directly access EMR Serverless applications and data from their Studio notebooks to perform their data preparation tasks at scale.
Optimize for	Using a visual interface in which you can: • Create data preparation pipelines • Perform data analysis • Transform data using built-in transforms • Use genAI-powered natural language instructions for data transforms	For users whose data resides in Amazon Redshift, Snowflake, Athena, or Amazon S3 and want to combine exploratory SQL and Python for data analysis and preparation without the need to learn Spark.	For users who prefer a serverless experience with automatic resource provisioning and termination for scaling short-running or intermittent interactive workloads revolving around Apache Spark, while taking advantage of SageMaker AI's machine learning capabilities.

Descriptor	Use case 1	Use case 2	Use case 3
	Optimized for tabular data tasks such as handling missing values, encoding categorical variables, and applying data transformations.		

Descriptor	Use case 1	Use case 2	Use case 3
Considerations	• It may not be the optimal choice if your team already has expertise in Python, Spark, or other languages. • It might not be best suited if you need full flexibility to customize transformations to add complex business logic or require full control over your data processing environment.	• This feature is designed for structured data residing in Amazon Redshift, Snowflake, Athena, or Amazon S3 only. • If the size of your query results exceeds your SageMaker AI instance memory, the following notebook can guide you on getting started with Athena to prepare your data for ingestion by a SageMaker AI algorithm.	• The learning curve for users not familiar with EMR Serverless applications and Spark-based tools can be challenging. • This feature is better suited for interactive data preparation tasks and may not be as efficient as Amazon EMR clusters for large-scale, long-running, or complex data processing requirements involving massive amounts of data, extensive integration with other services, custom applications, or diverse distributed data processing frameworks beyond just Apache Spark. • While serverless computing can be cost-effective for short-lived tasks, it is essential to monitor and manage costs carefully, especially for long-running or resource-intensive workloads.

Descriptor	Use case 1	Use case 2	Use case 3
Recommended environment	Getting started with using SageMaker Canvas	Launch Studio	Launch Studio

Additional options

SageMaker AI offers the following additional options for preparing your data for use in machine learning models.

- [the section called “Data preparation using Amazon EMR”](#): For long-running, computationally intensive, large-scale data processing tasks, consider using Amazon EMR clusters from SageMaker Studio. Amazon EMR clusters are designed to handle massive parallelization and can scale to hundreds or thousands of nodes, making them well-suited for big data workloads that require frameworks like Apache Spark, Hadoop, Hive, and Presto. The integration of Amazon EMR with SageMaker Studio allows you to leverage the scalability and performance of Amazon EMR, while keeping your full ML experimentation, model training and deployment, centralized and managed within the SageMaker Studio environment.
- [Prepare data using glue interactive sessions](#): You can use the Apache Spark-based serverless engine from AWS Glue interactive sessions to aggregate, transform, and prepare data from multiple sources in SageMaker Studio.
- [Identify bias in training data](#) using Amazon SageMaker Clarify processing jobs: SageMaker Clarify analyzes your data and detect potential biases across multiple facets. For example, you can use Clarify API in Studio to detect if your training data contains imbalanced representations or labeling biases between groups such as gender, race, or age. Clarify can help you identify these biases before training a model to avoid propagating biases into the model's predictions.
- [Create, store, and share features](#): Amazon SageMaker Feature Store optimizes the discovery and reuse of curated features for machine learning. It provides a centralized repository to store feature data that can be searched and retrieved for model training. Storing features in a standardized format enables reuse across ML projects. The Feature Store manages the full lifecycle of features including lineage tracking, statistics, and audit trails for scalable and governed machine learning feature engineering.

- [Label data with a human-in-the-loop](#): You can use SageMaker Ground Truth to manage the data labeling workflows of your training datasets.
- [Use SageMaker Processing API](#): After performing exploratory data analysis and creating your data transformations steps, you can productionize your transformation code using [SageMaker AI Processing jobs](#) and automate your preparation workflow using [SageMaker Model Building Pipelines](#).

Data preparation with SQL in Studio

Amazon SageMaker Studio provides a built-in SQL extension. This extension allows data scientists to perform tasks such as sampling, exploratory analysis, and feature engineering directly within their JupyterLab notebooks. It leverages AWS Glue connections to maintain a centralized data source catalog. The catalog stores metadata about various data sources. Through this SQL environment, data scientists can browse data catalogs, explore their data, author complex SQL queries, and further process the results in Python.

This section walks through configuring the SQL extension in Studio. It describes the capabilities enabled by this SQL integration and provides instructions for running SQL queries in JupyterLab notebooks.

To enable SQL data analysis, administrators must first configure AWS Glue connections to the relevant data sources. These connections allow data scientists to seamlessly access authorized datasets from within JupyterLab.

In addition to the administrator-configured AWS Glue connections, the SQL extension allows individual data scientists to create their own data source connections. These user-created connections can be managed independently and scoped to the user's profile through tag-based access control policies. This dual-level connection model - with both administrator-configured and user-created connections - provides data scientists with broader access to the data they need for their analysis and modeling tasks. Users can set up the necessary connections to their own data sources within the JupyterLab environment user interface (UI), without relying solely on the centralized connections established by the administrator.

Important

The user-defined connections creation capability is available as a set of standalone libraries in PyPI. To use this functionality, you need to install the following libraries in your JupyterLab environment: