

COMPARISON BETWEEN MODELARTS & VERTEX AI

What is ModelArts?

A product of Huawei Cloud which is a one-stop AI development platform enabling developers and data scientists of any skill level to rapidly build, train and deploy models anywhere, from the cloud to the edge as well as manage full-lifecycle AI workflows. ModelArts fosters AI innovation with capabilities that include data preprocessing and auto labeling, distributed training, automated model building and one-click workflow executing.

What is Vertex AI?

Vertex AI is a machine learning (ML) platform from Google Cloud that lets you train and deploy ML models and AI applications, and customize large language models (LLMs) for use in your AI-powered applications. Vertex AI combines data engineering, data science, and ML engineering workflows, enabling your teams to collaborate using a common toolset and scale your applications using the benefits of Google Cloud.

Similarities

- **AI development platforms that allow training and deployment of ML models and AI applications**
- **Possible to train models without having to write code (using AutoML for Vertex AI and ExeML for ModelArts)**

ExeML is the process of automating model design, parameter tuning, and model training, model compression, and model deployment with the labeled data. The process is code-free and does not require developers to have experience in model development. A model can be built in three steps: labeling data, training a model, and deploying the model.

AutoML lets you train tabular, image, text, or video data without writing code or preparing data splits.

- **Provide SDK (Software Development Kit) for Python:**

ModelArts SDKs have been integrated into ModelArts notebook.

The Vertex AI SDK for Python helps you automate data ingestion, train models, and get predictions on Vertex AI. The Vertex AI SDK uses Python code to access the Vertex AI API so that you can programmatically accomplish most of what you can do in the Google Cloud console. The Vertex AI SDK for Python is recommended for an experienced machine learning (ML) and artificial intelligence (AI) engineer or a data scientist who wants to programmatically automate his workflow.

- **Availability of notebook based environments for compiling code**

Vertex AI provides two options for running your code in notebooks, Colab Enterprise and Vertex AI Workbench. ModelArts DevEnviron uses cloud native resources and integrates the development tool chain to provide better in-cloud AI development experience for AI development, exploration, and teaching.

ModelArts notebook for seamless in-cloud and on-premises collaboration

- In-cloud JupyterLab, local IDE, and ModelArts plug-ins for remote development and debugging, tailored to your needs
- In-cloud development environment with AI compute resources, cloud storage, and built-in AI engines
- Custom runtime environment saved as an image for training and inference

- **Supports the entire development process, including data processing, model training, management, and deployment**

The machine learning workflow of Vertex AI involves data preparation, model training, model iteration and evaluation, model deployment and serving and model monitoring. The processes are common in ModelArts as well for building and deploying models

- **Supports images, text, speech, tabular and video data**

ModelArts supports datasets of images, audio, text, tables, videos, and other types for the following purposes:

- Images
 - Image classification: identifies a class of objects in images.
 - Object detection: identifies the position and class of each object in an image.
 - Image segmentation: identifies the outline of each object in an image
- Audio
 - Sound classification: classifies and identifies different sounds.
 - Speech labeling: labels speech content.
 - Speech paragraph labeling: segments and labels speech content.
- Text
 - Text classification: assigns labels to text according to its content.
 - Named entity recognition: assigns labels to named entities in text, such as time and locations.
 - Text triplet: assigns labels to entity segments and entity relationships in the text.
- Tables
 - Table: applies to structured data processing such as tables. The file format can be CSV. Tables cannot be labeled but you can preview up to 100 data records in a table.

- Videos
 - Video labeling: identifies the position and class of each object in a video. Only the MP4 format is supported.

Differences between ModelArts & Vertex AI

WORKFLOW

ModelArts supports the entire development process, including data processing, and model training, management, and deployment. It also provides AI Gallery for sharing models.

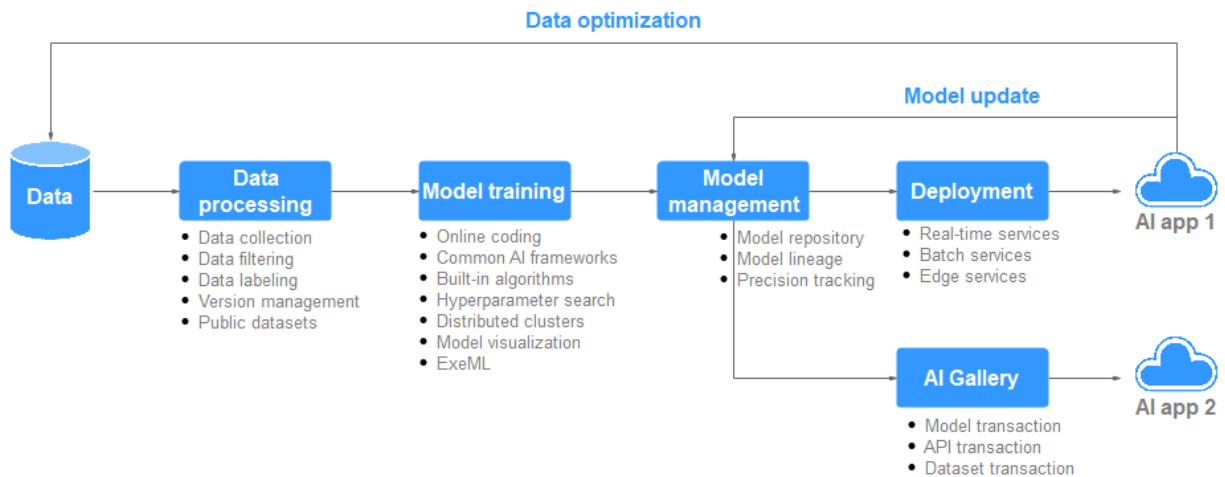


Figure 1 ModelArts Standard Architecture

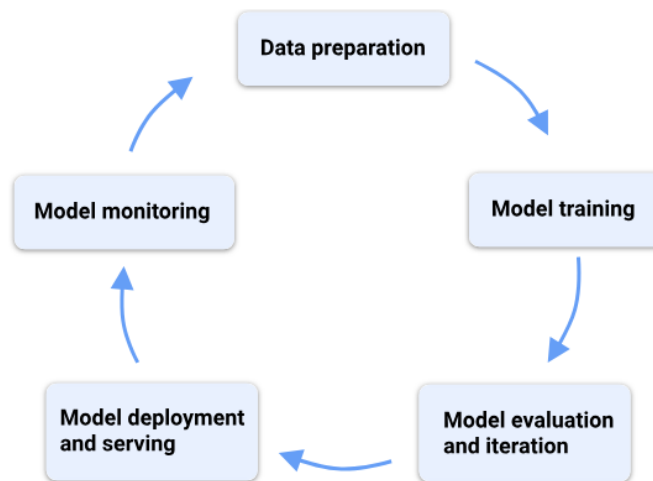


Figure 2 Vertex AI workflow

INTERFACES TO ACCESS THE SERVICES

One can access ModelArts through the web-based management console or by using HTTPS-based application programming interfaces. Interaction with Vertex AI is possible through a wide range of interfaces. Some Vertex AI operations are only available through specific interfaces during your workflow.

Example: One must use the API to log data to an experiment run , but the results can be viewed on the console.

Vertex AI	ModelArts
<ul style="list-style-type: none">▪ Console It can be used for managing your managed datasets, models, endpoints and jobs. Cloud Storage and BigQuery can be accessed through the console.▪ gcloud Use the Google Cloud CLI when you want to manage your Vertex AI resources from the command line or through scripts and other automation.▪ Terraform One can define the Vertex AI resources and permissions for his/her Google Cloud project in a Terraform configuration file. Then use Terraform to apply the configuration to the project by creating new resources and updating existing resources.▪ Python Vertex AI SDK for Python allows programmatically automating Vertex AI workflow▪ Client libraries Client libraries use each supported language's natural conventions to call the Vertex API and reduce boilerplate code.	<ul style="list-style-type: none">▪ Management Console ModelArts Standard features a simple and easy-to-use management console, and provides a host of functions including ExeML, data management, development environment, model training, AI application management, AI Gallery, and service deployment. You can complete end-to-end AI development on the management console.▪ Using SDKs For integrating ModelArts into a third-party system for secondary development.▪ Using APIs If one wants to integrate ModelArts into a third-party system for secondary development. APIs can be used to access ModelArts.

<ul style="list-style-type: none"> ▪ REST The Vertex AI REST API provides RESTful services for managing jobs, models and endpoints and for making predictions with hosted models on Google Cloud. It has to be used if one needs to use own libraries to call the Vertex AI API from one's application. 	
---	--

FUNCTIONS

Vertex AI	ModelArts
<ul style="list-style-type: none"> ▪ Orchestrate workflows Vertex AI Pipelines helps automate, monitor and govern ML workflows ▪ Track the metadata used in an ML system Vertex ML Metadata lets you record the metadata, parameters and artifacts that are used in your ML system. You can then query that metadata to help analyze, debug and audit the performance of your ML system or the artifacts that it produces. ▪ Identify the best model for a use case Vertex AI Experiments lets you track and analyze different model architectures, hyperparameters and training environments for identifying the best model for your use case ▪ Manage model versions Vertex AI Model Registry provides an overview of your models so that you can better organize, track and train new versions. One can evaluate models, deploy models to an endpoint, create batch predictions and view 	<ul style="list-style-type: none"> ▪ Data governance Manages data preparation, such as data filtering, labeling and dataset versions. ▪ Rapid and simplified model training Enables high-performance distributed training and simplifies coding with the self-developed MoXing deep learning framework. ▪ Multi-scenario deployment Deploy models in various production environments such as devices, the edge and the cloud and supports real-time batch inference. ▪ Auto Learning Enables model building without coding and supports image classification, object detection and predictive analytics. ▪ AI Gallery

<p>details about specific models and model versions.</p> <ul style="list-style-type: none"> ▪ Manage features Vertex AI feature store provides a centralized repository organizing, storing and serving ML features. Using a central featurestore enables an organization to re-use ML features at scale and increase the velocity of developing and deploying new ML applications ▪ Monitor model quality Vertex AI Model Monitoring monitors models for training-serving skew and prediction drift and sends you alerts when the incoming prediction data skews too far from the training baseline. You can use alerts and feature distributions to evaluate whether you need to retrain your model. ▪ Scale AI and Python applications Ray on Vertex AI is designed so you can use the same open source Ray code to write programs and develop applications of Vertex AI with minimal changes. You can then use Vertex AI integrations with other Google Cloud services such as Vertex AI Prediction and BigQuery as part of your machine learning workflow. 	<p>Supports commonly used algorithms and datasets, and internal and public sharing of models.</p>
---	---

INTEGRATED FRAMEWORKS

Vertex AI	ModelArts
<ul style="list-style-type: none"> ▪ PyTorch and Tensorflow <p>Vertex AI's PyTorch and TensorFlow integration makes it easier for one to train, deploy and orchestrate PyTorch and TensorFlow models in production.</p>	<ul style="list-style-type: none"> ▪ MoXing <p>A lightweight distributed framework developed by the ModelArts team and built on deep learning engines such as TensorFlow, PyTorch, MXNet, and MindSpore. It improves performance of</p>

	<p>these engines and makes them easier to use.</p> <p>MoXing contains many components. MoXing Framework is a basic common component that can be used to access OBS. It is decoupled from AI engines and can be used in all ModelArts-supported AI engines such as Tensorflow, PyTorch, MXNet and MindSpore.</p>
--	---