



BentoML

Usage Sample

Train and save a model with:

```
import bentoml
... ## Model training code goes here
bento_model = bentoml.sklearn.save_model('model', model)
```

Assuming there is a service.py file that specifies the environment, loads the model and defines the model's API, we can serve the model with:

```
$ bentoml serve .
```

Companies using BentoML

Neurolabs



tomtom

bigdata
republic



NAVER

BentoML is an open-source platform designed to simplify deploying, managing, and scaling machine learning models. It bridges the gap between model development and production, simplifying those transitions for data scientists and engineers. With support for a variety of frameworks, developer friendly features, scaling optimizations, and integration with common tools, BentoML can ease and fortify a variety of ML systems.

Core Features



Model Serving and Deployment

Supports REST and gRPC APIs, dynamic auto-scaling, and containerization via Kubernetes and Docker for production environments.



Model Management

Centralized model store with versioning, dependency tracking, and standardized packaging for consistent and reproducible deployments.



Scalability and Performance

High-throughput serving with GPU acceleration, batch processing, and parallel request handling.



Developer Experience

Python-first APIs, live service development with auto-reloading, and built-in Swagger UI for API discoverability and documentation.

Strengths



Unified Deployment Framework

Simplifies the ML model serving and deployment process.



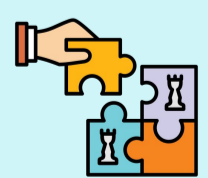
Framework Agnostic

Compatible with a wide range of ML frameworks and libraries.



High Performance

Features like adaptive micro-batching, GPU acceleration, and parallel request handling make for scalable workloads.



Streamlined CI/CD Integration

Automates building, testing, and deploying models through tools like GitHub Actions.



Clear Ownership

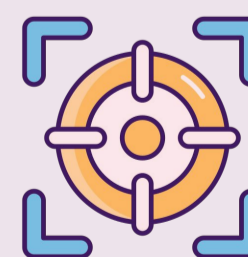
Isolated services for each model ensure clarity in maintenance and operations.

Weaknesses



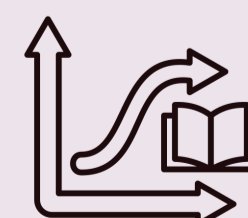
Complexity

Serving configurations can be verbose, manual and require inside knowledge of BentoML.



Focused Scope

Prioritizes serving and deployment, offering limited features for experiment tracking compared to tools like MLFlow.



Learning Curve

Advanced orchestration with Kubernetes may require DevOps expertise.