ML COMPETENCY - LEARN BY DOING

Infrastructure deliverable

Configuration Management tools

HYDRA VS DYNACONF

Hydra is an **open-source** Python framework that simplifies the development of research and other complex applications. The key feature is the ability to dynamically create a hierarchical configuration by composition and override it through config files and the command line. The name Hydra comes from its ability to run multiple similar jobs - much like a Hydra with multiple heads.

Hydra allows users to define hyperparameters using **YAML** files in a hierarchically structured order. Also, it is possible to change and execute the configuration values directly from the command line and to perform a grid search on hyperparameters with a single command.

Basic Usage

- Write the hyperparameters that need to be managed in a config file in YAML format.

Key features:

- Hierarchical configuration composable from multiple sources
- Configuration can be specified or overridden from the command line
- Dynamic command line tab completion
- Run your application locally or launch it to run remotely
- Run multiple jobs with different arguments with a single command
- Hydra supports Linux, Mac and Windows.
- Use the version switcher in the top bar to switch between documentation versions.
- hyperparameter management in machine learning.

Hydra is highly configurable. Many of its aspects and subsystems can be configured, including:

- The Launcher
- The Sweeper
- Logging
- Output directory patterns
- Application help (--help and --hydra-help)

Getting started | Hydra

DYNACONF

- **Settings management** (default values, validation, parsing, templating)
- Protection of **sensitive information** (passwords/tokens)
- Define comprehensive default values.
- Store parameters in multiple file formats (**.toml**, .json, .yaml, .ini and .py) and customizable loaders.
- Full support for **environment variables** to override existing settings (dotenv support included).
- Optional layered system for **multi environments** [default, development, testing, production] (also called multi profiles)
- Built-in support for **Hashicorp Vault** and **Redis** as settings and secrets storage. external services
- Built-in extensions for **Django** and **Flask** web frameworks.
- Strict separation of settings from code (following 12-factor applications Guide).
- Sensitive **secrets** like tokens and passwords can be stored in safe places like *.secrets* file or *vault server*.
- Simple feature flag system.
- Support for *.env* files to automate the export of environment variables.
- Correct data types (even for environment variables).
- Have **only one** canonical settings module to rule all your instances.
- Powerful **$ dynaconf** CLI to help you manage your settings via console.
- Customizable Validation System to ensure correct config parameters.

Getting Started — dynaconf 2.2.3 documentation

Difference Table:

	Hydra	Dynaconf
Open Source	Yes	Yes
Basic Info	Hydra is an open-source Python framework that simplifies the development of research and other complex applications.	dynaconf a layered configuration system for Python applications from dynaconf
File formats	YAML files, hierarchically structured order	Store parameters in multiple file formats (.toml, .json, .yaml, .ini and .py) and customizable loaders.
simplifies the development of research and other complex applications.	Yes	Yes
CLI	Yes	True Yes
ability to run multiple similar jobs	Yes	Yes
Run your application locally	Yes	No
Supports Linux, Mac and Windows	yes	Windows and Linux
hyperparameter management in machine learning.	Yes	No
External framework Configuration	Yes	Not much
Protection of sensitive information	No	yes
Optional layered system for multi environments	No	yes
Built-in extensions	No	for Django and Flask web frameworks.
Customizable Validation System	No	Yes
Support for 12-factor applications	No	Yes