**Model Development V2 — Script Usage Guide**

This repository provides 5 command-line scripts for automated model/config generation and MLflow logging for predictive maintenance solutions. Each script is designed for a specific fault type or anomaly detection use case. All scripts support tagging runs as @production or @testing for MLflow tracking.

**General Usage**

* Place your dataset CSV files in the [data](vscode-file://vscode-app/c:/Users/ghosh/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) folder.
* Run the script from the command line, providing required arguments.
* Use --run-tag @production for production runs, or --run-tag @testing for test runs (default).
* All outputs and configs are saved in the [artifacts](vscode-file://vscode-app/c:/Users/ghosh/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) folder and logged to MLflow.

**Example Commands**

**1. Bearing Fault Model Training**

Detects bearing faults and generates a YAML config.

python bearing\_model\_training.py --tenant-id "28" --machine-id "257" --dataset-filename "iotts.harmonics\_257.csv" --shaft-rpm 1480 --ball-diameter 8.0 --pitch-diameter 40.0 --num-elements 8 --run-tag @production

**2. Gearbox Fault Model Training**

Detects gearbox faults using harmonic profiling.

python gearbox\_model\_training.py --tenant-id "28" --machine-id "257" --dataset-filename "iotts.harmonics\_257.csv" --run-name "harmonic\_profiling\_kstest\_v1" --use-parity-profiles --run-tag @production

**3. Winding Fault Model Training**

Detects winding faults and generates a config for phase imbalance.

python winding\_model\_training.py --tenant-id "28" --machine-id "257" --dataset-filename "iotts.harmonics\_257.csv" --run-tag @production

**4. Rotor Fault Model Training**

Detects rotor faults using PCA and SVM.

python rotor\_model\_training.py --tenant-id "28" --machine-id "257" --dataset-filename "rotor\_data.csv" --run-tag @production

**5. Anomaly Health Model Training**

Trains an LSTM autoencoder for anomaly detection in energy data.

python anomaly\_health\_model\_training.py --tenant-id "27" --machine-id "242" --dataset-filename "iotts.energy\_242.csv" --epochs 50 --window-size 15 --latent-dim 32 --run-tag @production

**Documentation & Parameters**

* Each script requires specific parameters (see script help via -h or --help).
* All scripts support --run-tag for MLflow tagging.
* Output artifacts (model/config) are saved in [artifacts](vscode-file://vscode-app/c:/Users/ghosh/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) and logged to MLflow.
* Only runs tagged as @production should be promoted for use in production systems.

**Notes**

* Make sure your dataset files are named and formatted as expected by each script.
* For testing, simply omit or set --run-tag @testing.
* For more details on parameters, run any script with --help.

**Example for help:**

python bearing\_model\_training.py –help

This guide aggregates the usage for all scripts in the repository. Use the relevant command for your equipment and fault type.