CarveLog Gen 4 – Guida d'Installazione & Utilizzo (Linux + Windows)

1. Cos'è CarveLog Gen 4

CarveLog è un pipeline multi-pass che **classifica** grandi volumi di log tramite TF-IDF streaming, MiniBatch K-Means e Random Forest.

2. Prerequisiti

- Python 3.9 + (consigliato 3.11)
- `pip` (o `pipx`)
- Compiler C/C++ (solo per alcune dipendenze native):
  - **Linux** → `build-essential`, `python3-dev`
  - **Windows** → *Visual Studio Build Tools 2022 → "Desktop development with C++"*
- RAM ≥ 8 GB (per i dataset forniti come esempio, i requisiti aumentano per dataset più grandi)

3. Quick Start

```
# Linux
cd carvelog
python3 -m venv .venv && source .venv/bin/activate
pip install -r requirements.txt
python carvelog_gen4.py        # Training
python carvelog_run2.py        # Inference
```

```
:: Windows (PowerShell)
cd carvelog
py -3 -m venv .venv; .\.venv\Scripts\Activate.ps1
pip install -r requirements.txt
py carvelog_gen4.py
py carvelog_run2.py
```

4. Installazione passo-passo

4.1 Scarica i sorgenti

Clona (o copia) `config.yaml`, `carvelog_gen4.py` e `carvelog_run2.py` in una directory

4.2 Crea l'ambiente virtuale

Linux

```
python3 -m venv .venv
source .venv/bin/activate
```

Windows

```
py -3 -m venv .venv
.\.venv\Scripts\Activate.ps1
```

4.3 Installa le dipendenze

Assicurati di avere `pip (versione 3.9 o successiva)`, poi:

```
pip install --upgrade pip wheel setuptools
pip install numpy scipy scikit-learn tqdm joblib pyyaml
```

> **TIP:** Se compili su Windows senza i Build Tools riceverai errori tipo *"cl.exe non trovato"*.

4.4 Configura `config.yaml`

La configurazione è già pronta per essere usata, non servono modifiche. In ogni caso i parametri piu' importanti sono:

- `training_dataset` / `inference_dataset`
- `models_path` se desideri una directory diversa
- `initial_k_range` per cluster min/max

5. Fase Training – `carvelog_gen4.py`

1. Lancia:

```
python carvelog_gen4.py config.yaml
```

2. Osserva le 9 fasi nel log a schermo:
   - Fase 1 → Costruzione vocabolario (streaming)
   - Fase 3 → Selezione automatica di k (silhouette)
   - Fase 6 → Grid-search RandomForest con CV
3. Output finale:
   `models/km.joblib`, `models/rf.joblib`, `models/meta.npz`

6. Fase Inference – `carvelog_run2.py`

1. Prepara `inference-logs.txt` (oppure usa il file di esempio preconfigurato)
2. Esegui:

```
python carvelog_run2.py config.yaml
```

3. La Progress Bar mostra la distribuzione cluster in tempo reale.
4. Al termine vengono stampati:
   - Distribuzione dei cluster
   - Un certo numero di esempi per cluster (configurabili con `show_examples_per_cluster`)

7. Troubleshooting

- *MemoryError* durante TF-IDF → Riduci `chunk_size` a 10000.
- *ModuleNotFoundError: scipy.sparse* → `pip install scipy`
- Mancata convergenza K-Means → Abilita `expand_k_if_best_is_max`.