

Documentazione del Progetto: Dataset Builder Ontology

Vincenzo Buttari

4 febbraio 2025

Indice

1	Introduzione	2
2	Struttura del Progetto	2
3	Popolamento dell'Ontologia	3
4	Generazione del Dataset	3
5	Modello Predittivo	3
6	Interfacce Utente	3
7	Istruzioni per l'Installazione e l'Esecuzione	4
7.1	Prerequisiti	4
7.2	Esecuzione via CLI	4
7.3	Esecuzione dell'Interfaccia Web	4

1 Introduzione

Questo documento descrive il progetto **Dataset Builder Ontology**, un'applicazione che consente di:

- Popolare un'ontologia OWL con dati generati automaticamente, sfruttando la libreria `Owlready2`.
- Estrarre informazioni dall'ontologia per generare un dataset utilizzabile per modelli predittivi.
- Addestrare e valutare modelli predittivi (ad esempio, utilizzando `scikit-learn`).
- Interagire con il sistema tramite una CLI e una semplice interfaccia web sviluppata con `Flask`.

2 Struttura del Progetto

La gerarchia delle cartelle del progetto è la seguente:

```
dataset_builder_ontology/  
  data/  
    ontology.owl      % L'ontologia di partenza  
    dataset.csv       % Dataset generato (output)  
  src/  
    owl/  
      logger_config.py      % Configurazione del logging  
      populate_ontology.py  % Popolamento dell'ontologia  
      ontology_manager.py   % Caricamento e ragionamento sull'ontologia  
      dataset_generator.py  % Estrazione e costruzione del dataset  
      text_feature_extractor.py % Estrazione di feature testuali dai corsi  
    predictive_model/  
      predictive_model.py    % Addestramento e validazione  
                             del modello predittivo  
    app.py                % Interfaccia web (Flask)  
    cli.py                % Interfaccia a riga di comando  
  templates/  
    index.html  
    dataset.html  
    train.html  
  requirements.txt        % Dipendenze del progetto  
  README.md
```

3 Popolamento dell'Ontologia

La funzione `populate_ontology` (in `src/owl/populate_ontology.py`) si occupa di:

- Creare e definire le classi OWL (ad esempio, `Person` e `Course`).
- Definire le proprietà (ad es. `has_name`, `has_age`, `teaches`, `takes`, `course_title`, `course_description`).
- Creare e popolare automaticamente gli individui con dati realistici, utilizzando la libreria `Faker`.

4 Generazione del Dataset

Il modulo `dataset_generator.py` si occupa di estrarre le informazioni dagli individui dell'ontologia e di costruire un dataset in formato CSV. Le informazioni estratte includono:

- Nome e età della persona.
- Lista dei corsi seguiti (`takes`) e, se presente, dei corsi insegnati (`teaches`).

Il dataset viene salvato in `data/dataset.csv`.

5 Modello Predittivo

Il modulo `predictive_model.py` (in `src/predictive_model/`) definisce, addestra e valida un modello predittivo. Ad esempio:

- Viene utilizzata una regressione logistica per prevedere se una persona è un insegnante (basandosi sulle feature come l'età e il numero di corsi seguiti).
- Il dataset viene suddiviso in training e test set, le feature vengono normalizzate e il modello viene valutato tramite metriche standard (accuracy, classification report).

6 Interfacce Utente

Il progetto offre due modalità di interazione:

- **CLI:** Un'interfaccia a riga di comando (`src/cli.py`) che permette di:
 - Popolare l'ontologia (`populate`).
 - Estrarre il dataset (`extract`).
 - Addestrare il modello predittivo (`train`).
- **Interfaccia Web:** Un'applicazione web sviluppata con Flask (`app.py`) che consente di eseguire le stesse operazioni attraverso una interfaccia grafica e visualizzare:
 - Il dataset estratto (in `dataset.html`).
 - I risultati del modello predittivo (in `train.html`).

7 Istruzioni per l'Installazione e l'Esecuzione

7.1 Prerequisiti

- Python 3.6 o superiore
- Dipendenze elencate in `requirements.txt` (installabili con `pip install -r requirements.txt`)
- spaCy e il modello `it_core_news_sm` (installabili con `python -m spacy download it_core_news_sm`)

7.2 Esecuzione via CLI

Per utilizzare l'interfaccia a riga di comando:

```
python src/cli.py populate
python src/cli.py extract
python src/cli.py train
```

7.3 Esecuzione dell'Interfaccia Web

Per avviare l'applicazione web:

```
python app.py
```

Quindi aprire il browser all'indirizzo `http://127.0.0.1:5000/`.