

Documentazione del Progetto Icon-24-25: Airlytics → Dati e previsioni sulla qualità dell'aria di Achille Carbonara

La Visione di Airlytics

Immagina una città in cui l'aria che respiri è sempre monitorata, in cui puoi sapere con precisione quando e dove l'inquinamento raggiunge livelli critici. Immagina un sistema intelligente che non solo analizza i dati ambientali, ma prevede la qualità dell'aria prima che diventi un problema. Questo è Airlytics: un ecosistema avanzato per il monitoraggio e la previsione della qualità dell'aria, pensato per enti pubblici, aziende e cittadini consapevoli.

Grazie all'intelligenza artificiale, Airlytics non si limita a raccogliere dati, ma li trasforma in informazioni utili per la tutela della salute pubblica. Con una pipeline completamente automatizzata, il sistema analizza, pulisce e modella i dati per fornire previsioni affidabili sulla qualità dell'aria nelle aree urbane e industriali.

Introduzione

Il progetto **Airlytics** è un sistema di **monitoraggio e previsione della qualità dell'aria**. Il flusso principale prevede:

- **Ingestione dei dati** ambientali da file CSV.

- **Preprocessing e pulizia dei dati.**
- **Addestramento di un modello di machine learning** per predire valori di inquinanti.
- **Previsione sui nuovi dati.**
- **Valutazione del modello.**

Il sistema è modulare ed estendibile, permettendo l'aggiunta di nuovi modelli e strategie di previsione.

Struttura del Progetto

Il progetto è organizzato come segue:

```
Progetto-Icon-24-25/  
├── index.py                # Entry point principale del progetto  
├── requirements.txt        # Librerie necessarie  
├── src/  
│   ├── data_ingestion/    # Modulo di acquisizione e preprocessing dati  
│   │   ├── ingest_data.py # Caricamento dati  
│   │   ├── preprocessing/ # Modulo di pulizia e trasformazione dati  
│   │   │   ├── clean_data.py # Pulizia e normalizzazione dei dati  
│   │   │   └── models/       # Modelli di machine learning  
│   │   │       ├── train_model.py # Addestramento modello  
│   │   │       ├── predict.py    # Predizioni  
│   │   └── evaluation/        # Valutazione delle prestazioni del modello  
│   │       └── evaluate_model.py # Metriche di valutazione  
│   ├── utils/              # Funzioni di supporto  
│   └── logger.py           # Logging del sistema  
├── dataset/                # Dati grezzi e processati  
├── models/                 # Modelli addestrati salvati  
└── venv/                  # Ambiente virtuale (opzionale)
```

Setup dell'Ambiente

Per avviare il progetto, segui questi passi:

1. Clonare il repository:

```
git clone https://github.com/tuo_repo/Progetto-Icon-24-25.git
```

```
cd Progetto-Icon-24-25
```

2. Creare e attivare un ambiente virtuale:

```
python3 -m venv venv
source venv/bin/activate # Mac/Linux
venv\Scripts\activate # Windows
```

3. Installare le dipendenze:

```
Pip install -r requirements.txt
```

4. Eseguire il progetto:

```
python3 index.py
```



Flusso di Esecuzione

1. **Ingestione dei dati** → ingest_data.py legge i dati grezzi.
 2. **Pulizia e preprocessing** → clean_data.py rimuove outlier e normalizza.
 3. **Addestramento del modello** → train_model.py addestra il modello.
 4. **Previsione** → predict.py genera previsioni sui dati.
 5. **Valutazione** → evaluate_model.py calcola metriche di accuratezza.
-



Spiegazione dei Moduli



data_ingestion/

- **ingest_data.py** → Legge i dati grezzi e li salva in dataset/raw/.
- **preprocessing/clean_data.py** → Pulizia dei dati con rimozione di outlier, normalizzazione e salvataggio in dataset/processed/.



models/

- **train_model.py** → Addestra un modello di machine learning e lo salva in models/.
- **predict.py** → Usa il modello addestrato per generare previsioni.



evaluation/

- **evaluate_model.py** → Calcola metriche di accuratezza del modello.



utils/

- **logger.py** → Sistema di logging per tenere traccia delle operazioni svolte.
-

Tecnologie Utilizzate

- **Python 3.x**
 - **Pandas, NumPy** → Manipolazione e analisi dati.
 - **Scikit-learn** → Modelli di machine learning.
 - **Matplotlib** → Visualizzazione dati.
 - **Logging** → Gestione dei log di sistema.
-

Possibili Miglioramenti

- **Ottimizzazione del modello** → Testare modelli più avanzati.
 - **Automazione del pipeline** → Creare una pipeline completa con Airflow.
 - **Visualizzazione dati** → Dashboard interattiva per i risultati.
 - **Supporto per più fonti di dati** → Collegare API di dati in tempo reale.
-

Copyright

© 2025 Airlytics. Tutti i diritti riservati. Questo documento e il codice contenuto nel progetto sono protetti da copyright. È vietata la riproduzione, la distribuzione o la modifica non autorizzata senza il consenso esplicito dell'autore.

