Module 4: Big Environmental Data Analysis

- Handling air/water/soil quality datasets
- Data cleaning, feature extraction, anomaly detection
- Libraries: Pandas, Scikit-learn, TensorFlow for environmental data
- 📌 Example: Detecting pollution anomalies using ML in air quality data

IR0000032 – ITINERIS, Italian Integrated Environmental Research Infrastructures System
(D.D. n. 130/2022 - CUP B53C22002150006) Funded by EU - Next Generation EU PNRR-Mission 4 "Education and Research" - Component 2: "From research to business" - Investment 3.1: "Fund for the realisation of an integrated system of research and innovation infrastructures"

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani

Why Analyse Environmental Quality Data?

- Environmental health and public safety depend on accurate, timely monitoring
- Massive sensor datasets are hard to interpret manually
- ML helps identify patterns, outliers, and trends at scale

Environmental Data Sources

- Air: PM2.5, NO_2, O_3 sensors (e.g., AirNow, OpenAQ)
- Water: pH, turbidity, contaminants (e.g., USGS, EEA)
- Soil: nutrient levels, moisture, toxins (e.g., FAO, SoilGrids)
- Time-series, geospatial, multivariate

Workflow Overview

- Ingest data (CSV, APIs, sensor streams)
- Clean & preprocess (handle missing data, normalize)
- Extract features (trends, rolling averages, domain transformations)
- Train ML models (classification, regression, clustering)
- Detect anomalies & generate alerts

Data Cleaning Essentials

- Handling missing values (interpolation, dropping, imputation)
- Dealing with noise and outliers
- Standardizing timestamps and units
- Tools: pandas, numpy, scikit-learn.preprocessing

Feature Engineering & Domain Context

- Rolling means, gradients, time-of-day effects
- Derived variables: AQI from raw pollutants
- Encoding seasonality, weather influence
- Use domain knowledge to engineer meaningful features

Example – Detecting Anomalies in Air Quality

- Dataset: PM2.5 and NO_2 sensor data from OpenAQ
- Model: Isolation Forest or One-Class SVM
- Steps:
  - Normalize and window the data
  - Train unsupervised model
  - Flag abnormal spikes not explained by season/weather
- Visual output: line chart with anomaly markers

ML Models for Environmental Data

- Supervised: regression (e.g., predict pollutant levels), classification (safe vs hazardous)
- Unsupervised: clustering, anomaly detection
- Time-series: LSTM for forecasting pollutant trends
- Model selection based on task + data availability

Python Tools & Libraries

- pandas: data wrangling
- scikit-learn: models, pipelines, validation
- tensorflow/keras: deep learning (LSTM, autoencoders)
- matplotlib, seaborn, plotly: visualizations
- prophet, tsfresh: time-series specific tools

Use Cases in Action

- Predicting PM2.5 spikes for health advisories
- Detecting illegal discharges in rivers
- Identifying agricultural soil depletion
- AI-powered environmental monitoring dashboards

Challenges in Environmental ML

- Noisy sensors and missing values
- External confounders (e.g., traffic, wind)
- Label scarcity for supervised models
- Data drift: models degrade over time without retraining

Best Practices & Validation

- Cross-validation for time-series (e.g., TimeSeriesSplit)
- Correlate anomalies with known events or news reports
- Alert thresholds + expert feedback loops
- Emphasize explainability and transparency

Resources & Datasets

- Datasets: OpenAQ, EPA, Water Quality Portal, SoilGrids
- Tutorials: Earth Data Science, DataCamp (environmental ML), Google Colab notebooks
- Community: GitHub, Kaggle, DrivenData environmental challenges

Takeaways

- ML unlocks hidden insights in environmental quality data
- Start small: clean data → extract features → anomaly detection
- Python ecosystem offers robust tools for all stages

Q&A and Discussion Prompt

- Prompt:
  - “What environmental issue in your community could benefit from ML-based monitoring?”
- Invite questions, feedback, shared concerns