



Module 3: Remote Sensing with AI

- Satellite, drone, and LIDAR data sources
- Automated image classification and land use analysis
- Tools: Google Earth Engine + Python; QGIS with ML plugins
- 📌 Demo: Land cover classification using Earth Engine + ML
- 📌 Group work: Brainstorm ideas for AI-powered remote sensing applications

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”

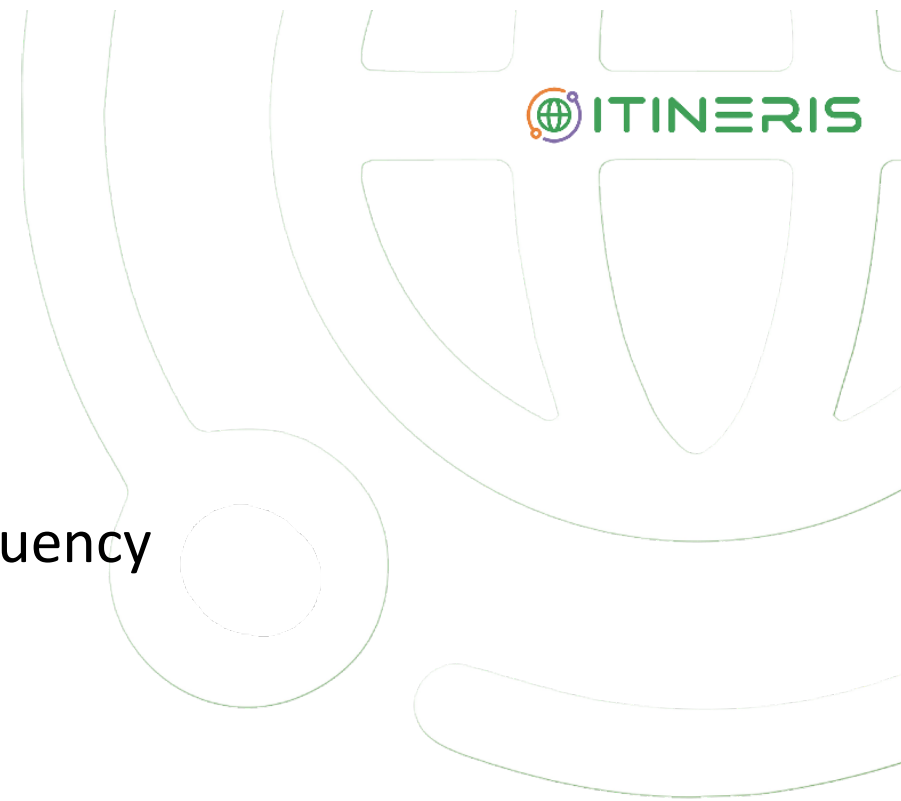


Why Remote Sensing + AI?

- 🌐 Importance of environmental monitoring and land use mapping
- 🌐 Limitations of manual analysis at large scales
- 🌐 AI's role in accelerating insights from remote sensing data

Data Sources Overview

- 🌐 Satellites: Sentinel, Landsat, MODIS
- 🌐 Drones: High-res, local observations
- 🌐 LIDAR: 3D terrain and vegetation structure
- 🌐 Trade-offs: resolution, coverage, update frequency

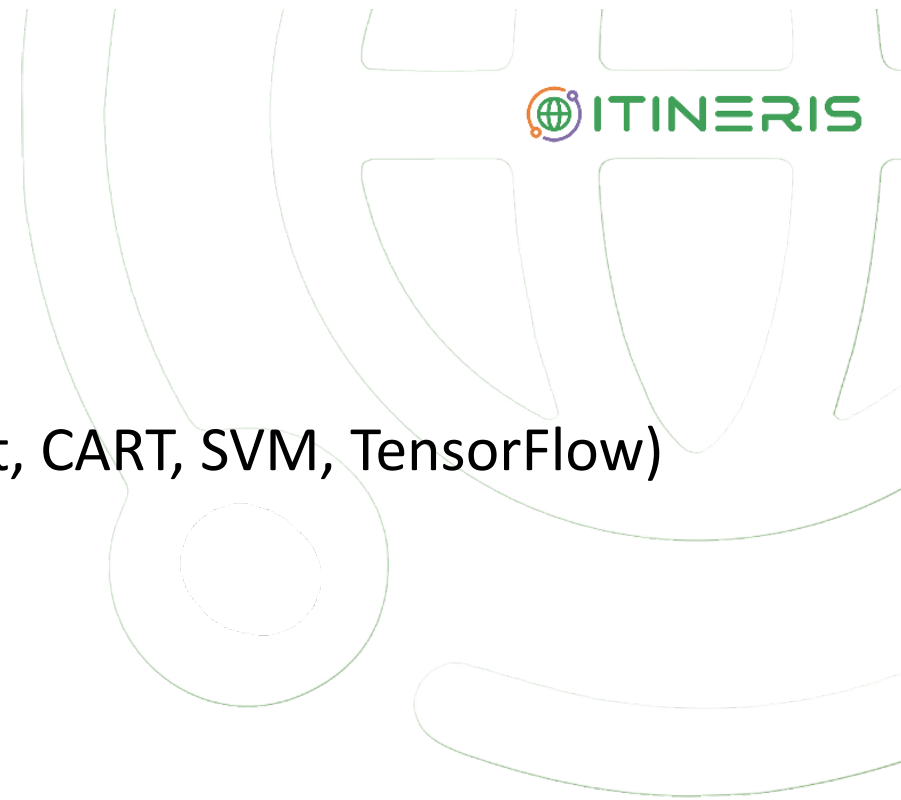


What AI Can Do with This Data

- 🌐 Automated land cover classification
- 🌐 Change detection (deforestation, urban expansion)
- 🌐 Object detection (buildings, roads, crops)
- 🌐 Terrain analysis using elevation and canopy structure

Google Earth Engine (GEE)

- 🌐 Cloud-based geospatial processing platform
- 🌐 Integrates global satellite archives
- 🌐 Machine Learning integration (random forest, CART, SVM, TensorFlow)
- 🌐 Visual programming and Python API



Python & QGIS Tools

- 🌐 Python: scikit-learn, TensorFlow, geemap for Earth Engine
- 🌐 QGIS: Semi-Automatic Classification Plugin, ML plugins
- 🌐 Combining open-source tools for end-to-end workflows

Demo – Land Cover Classification with Earth Engine



Step-by-step:

- Load Sentinel-2 imagery
- Train classifier (e.g., Random Forest)
- Visualize classified map (forest, water, urban, agri)

 Option: Compare with ground truth or historical images

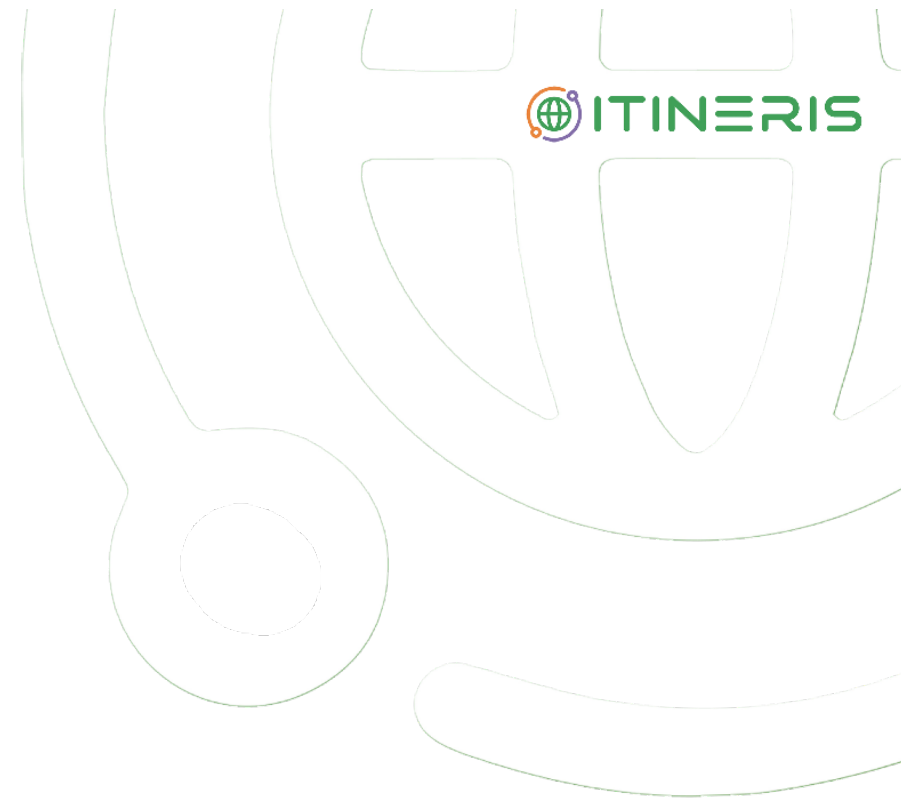
Key Considerations in Classification

- 🌐 Selecting good training samples
- 🌐 Handling cloud cover and seasonal changes
- 🌐 Accuracy assessment: confusion matrix, kappa coefficient
- 🌐 Edge cases (e.g., mixed land types)



Use Cases & Impact





- 🌐 Forest monitoring (REDD+)
- 🌐 Urban planning and smart cities
- 🌐 Disaster response (flood zones, landslides)
- 🌐 Agriculture (crop health, yield estimation)





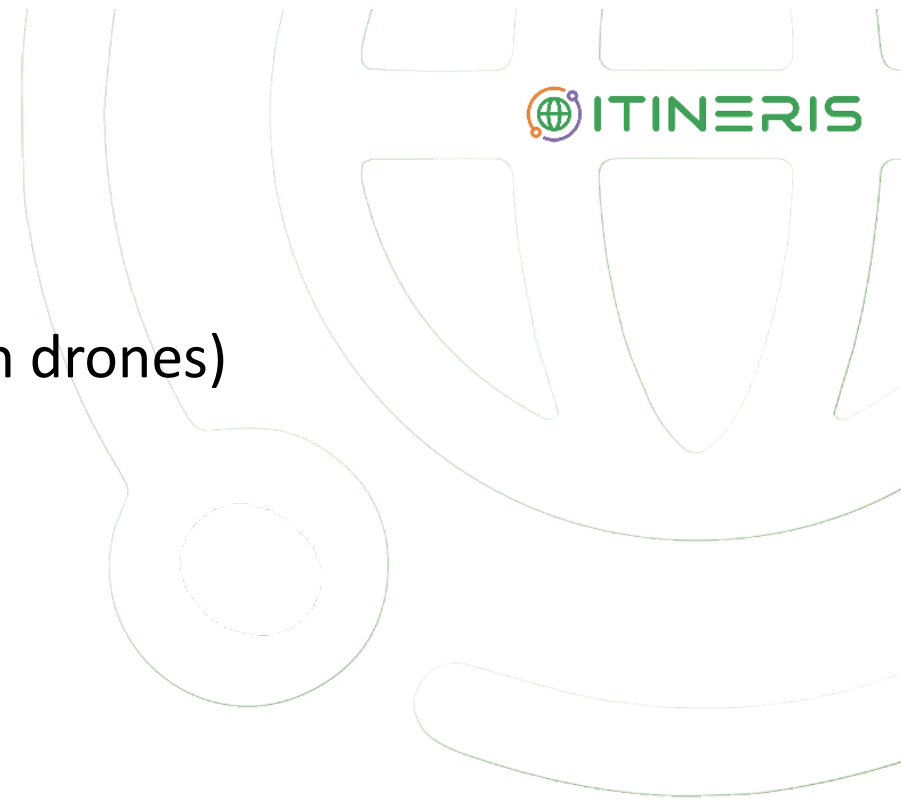
Group Activity - AI + Remote Sensing Idea Storm



-  Prompt: “Design a remote sensing project using AI for your local environment. What problem would it solve?”
-  Groups of 2–3
-  10 min brainstorm
-  2 min per group shareback

Challenges & Ethics

- 🌐 Data access inequality
- 🌐 Surveillance and privacy risks (especially with drones)
- 🌐 Interpretability and trust in ML decisions
- 🌐 Need for local validation



Resources & Datasets

- 🌐 Earth Engine datasets: Sentinel, Landsat, CHIRPS, etc.
- 🌐 LIDAR: USGS 3DEP, OpenTopography
- 🌐 Drones: OpenDroneMap, DroneDeploy
- 🌐 Tutorials: Google Dev Docs, Mapbox, Radiant Earth


Recap & Takeaways

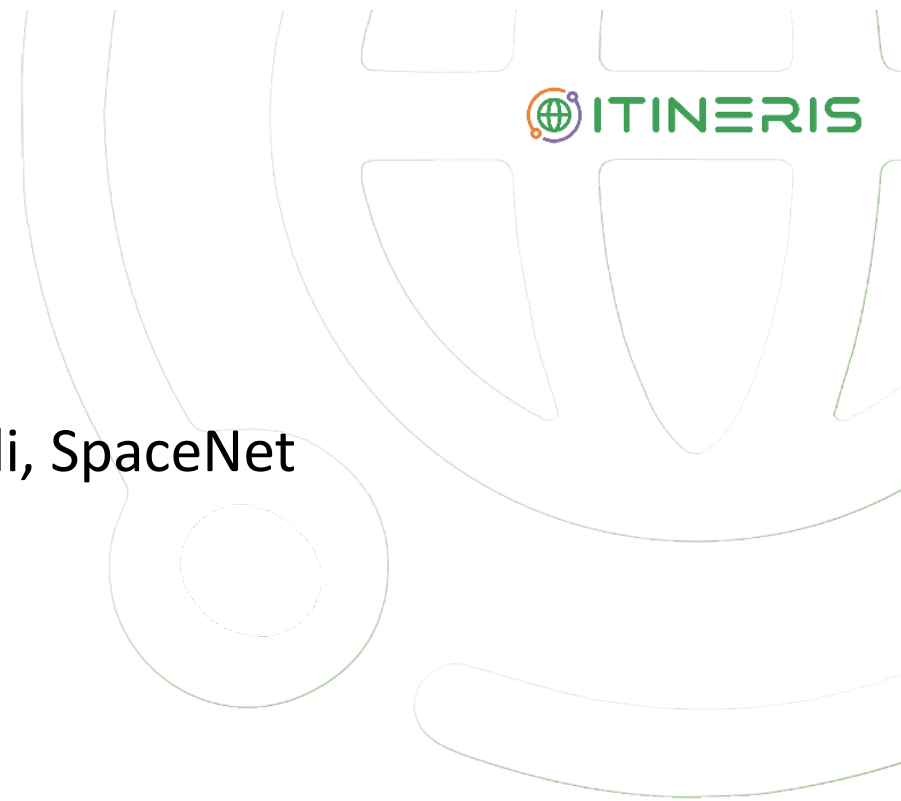
- 🌐 AI + remote sensing unlocks global and local-scale insights
- 🌐 Earth Engine + Python/QGIS = powerful, accessible combo
- 🌐 Always balance automation with ground truth and ethics

What's Next?

 Learn [GEE Python API](#)


 Try QGIS with ML

 Explore open competitions: DrivenData, Zindi, SpaceNet



Q&A / Feedback

 Invite discussion

 Ask: “What application would you prototype if you had a satellite feed for your neighborhood?”

