Session 1: Supervised Classification with Random Forest

Theory and Practice for Earth Observation

Stylianos Kotsopoulos
EU-Philippines CoPhil Programme

Session Overview

Part A: Theory (1.5 hours)

- Introduction to Supervised Classification
- Decision Trees Fundamentals
- Random Forest Ensemble Method
- Feature Importance
- Accuracy Assessment
- Google Earth Engine Platform

Part B: Hands-on Lab (1.5 hours)

- Sentinel-2 Data Acquisition
- Feature Engineering (Spectral Indices)
- Training Data Preparation
- Model Training & Optimization
- Classification & Validation
- Philippine NRM Applications

Learning Objectives:

- Understand supervised classification workflow for EO data
- Implement Random Forest using Google Earth Engine
- Perform accuracy assessment and interpret results
- Apply classification to Palawan land cover mapping

Part A: Theory

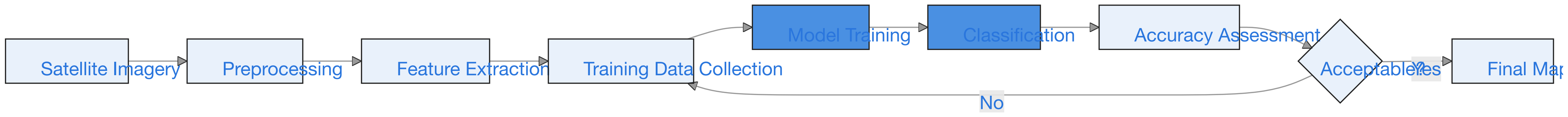
What is Supervised Classification?

- **Goal:** Assign labels to pixels/objects based on their characteristics
- **“Supervised”:** We provide labeled training examples to the algorithm
- **Learning Process:** Algorithm learns patterns from training data
- **Application:** Classify entire image based on learned patterns

Key Concept

Supervised classification requires **labeled training data** (ground truth) to learn the relationship between spectral signatures and land cover classes.

Supervised Classification Workflow



Key Steps:

1. **Preprocessing:** Cloud masking, atmospheric correction
2. **Feature Extraction:** Spectral bands, indices (NDVI, NDWI)
3. **Training Data:** Collect representative samples for each class
4. **Model Training:** Train classifier on training data
5. **Classification:** Apply model to entire scene
6. **Validation:** Assess accuracy with independent test data

Common Land Cover Classes in Philippines

Natural Ecosystems:

- Primary Forest (dipterocarp)
- Secondary Forest
- Mangroves
- Grasslands
- Water Bodies (rivers, lakes, coastal)

Human-Modified:

- Agricultural Land (rice paddies, coconut)
- Urban/Built-up Areas
- Bare Soil
- Mining Areas
- Roads and Infrastructure

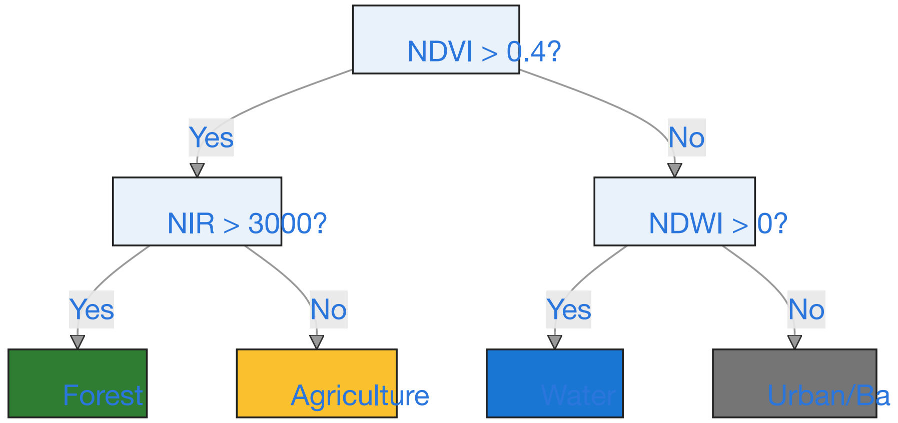
Philippine Context

Accurate land cover classification supports monitoring of **Protected Areas**, **REDD+ programs**, **agricultural expansion**, and **disaster response** (typhoons, floods).

Decision Trees

What is a Decision Tree?

A tree-like structure that makes decisions by asking a series of questions about features.



How it Works:

1. Start at root node
2. Test condition (e.g., NDVI > 0.4?)
3. Branch based on answer
4. Repeat until reaching leaf node
5. Leaf node = predicted class

Decision Tree Splitting

How does a tree decide where to split?

- **Goal:** Create pure nodes (all samples belong to one class)
- **Metric:** Information Gain or Gini Impurity
- **Process:** Test all possible splits, choose the best one
- **Recursion:** Repeat for each branch until stopping criteria

Gini Impurity Formula:

$$Gini = 1 - \sum_{i=1}^{n} (p_i)^2$$

Where p_i is the probability of class i in the node.

- **Gini = 0:** Pure node (all samples same class) ✓
- **Gini = 0.5:** Maximum impurity (50/50 split) ✗

Decision Tree Example: Spectral Splitting

Split 1 | Split 2 | Split 3

Root Node: All 1000 samples

Test: **NDVI > 0.4?**

- **Left branch (NDVI ≤ 0.4):** 400 samples → Mostly Water, Urban, Bare
- **Right branch (NDVI > 0.4):** 600 samples → Mostly Forest, Agriculture

Information Gain: High ✓ (classes becoming more separated)

Decision Tree Advantages & Limitations

Advantages:

- ✓ Easy to understand and visualize
- ✓ No data normalization needed
- ✓ Handles non-linear relationships
- ✓ Feature importance easily extracted
- ✓ Fast prediction

Limitations:

- ✗ **Overfitting:** Can memorize training data
- ✗ **High variance:** Small data changes → big tree changes
- ✗ **Instability:** Greedy algorithm (local optima)
- ✗ **Bias:** Favor features with many levels

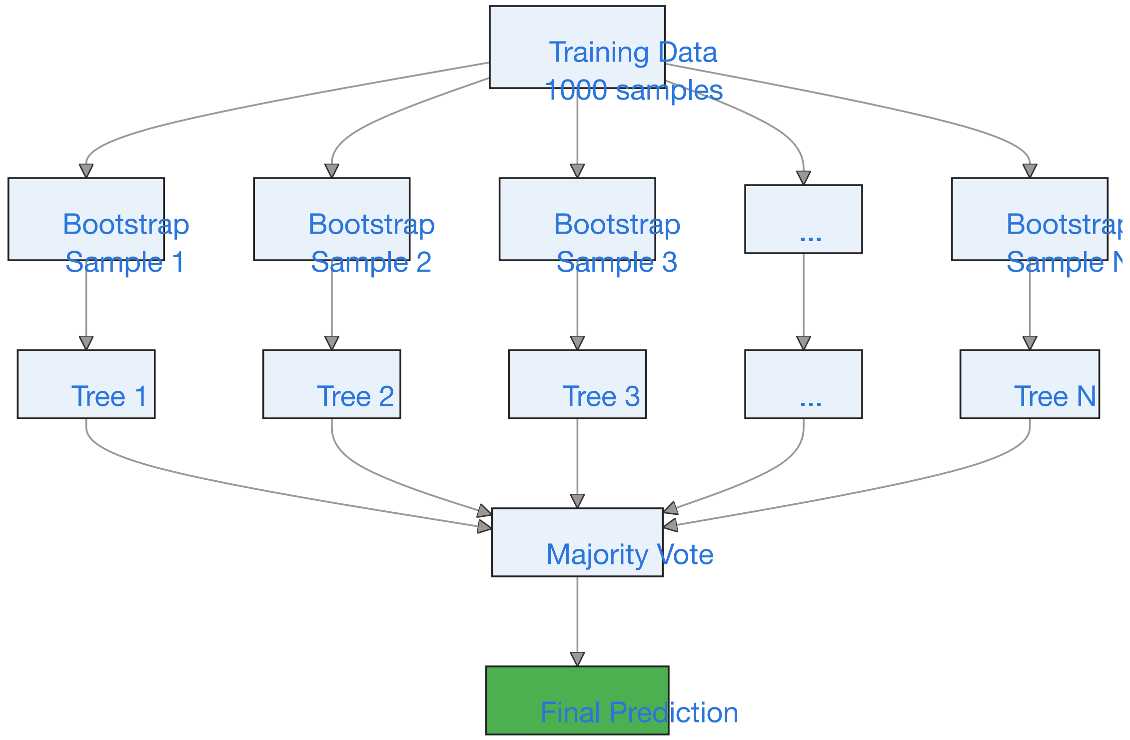
The Solution

Random Forest addresses these limitations by combining many decision trees!

Random Forest

What is Random Forest?

An **ensemble learning** method that combines many decision trees to improve accuracy and reduce overfitting.



Key Ideas:

1. **Bootstrap Aggregating (Bagging)**
   - Random sampling with replacement
   - Each tree sees different data
2. **Random Feature Selection**
   - Each split uses random subset of features
   - Reduces correlation between trees
3. **Majority Voting**
   - Classification: Most common class
   - Regression: Average prediction

Random Forest: The "Forest" Analogy

- **One tree (Decision Tree):** One expert's opinion
  - Can be very confident but sometimes wrong
  - Might overfit to specific training examples
- **Forest (Random Forest):** Committee of experts
  - Each expert sees slightly different data
  - Each expert considers different features
  - Final decision: Majority vote
  - **Wisdom of crowds:** Group decision more reliable than individual

Intuition

If you ask 100 independent experts and 75 say "Forest", you can be more confident than if only 1 expert says "Forest".

Bootstrap Aggregating (Bagging)

Bootstrap Sampling:

- Original training set: 1000 samples
- Each tree gets: 1000 samples (with replacement)
- Some samples repeated, some never selected (~37% out-of-bag)

Original Data:

```
Sample IDs: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
```

Tree 1 Bootstrap:

```
Sample IDs: 1, 3, 3, 5, 7, 7, 7, 9, 9, 10
```

Tree 2 Bootstrap:

```
Sample IDs: 2, 2, 4, 5, 5, 6, 8, 8, 9, 10
```

Why Bootstrap?

- Introduces diversity between trees
- Each tree specializes on different samples
- Reduces overfitting
- Enables Out-of-Bag (OOB) validation

Out-of-Bag Samples: - Samples not used by a tree (~37%) - Used for internal validation - No separate validation set needed

Random Feature Selection

At each split, only consider a **random subset** of features.

All Features (13 for Sentinel-2 + indices):

- B2 (Blue)
- B3 (Green)
- B4 (Red)
- B8 (NIR)
- B11 (SWIR1)
- B12 (SWIR2)
- NDVI
- NDWI
- NDBI
- EVI
- SAVI
- Texture features
- Elevation

Random Subset at Each Split:

Typical: $\sqrt{n}$ features

For 13 features: $\sqrt{13} \approx 4$ features

Tree 1, Split 1: {NDVI, B4, B11, Elevation}

Tree 1, Split 2: {B8, NDWI, B3, SAVI}

Tree 2, Split 1: {NDBI, B12, NDVI, B2}

. . .

Result: - Trees decorrelated - Prevents strong features from dominating - Better generalization

Random Forest Prediction: Majority Voting

Example Classification:

Classify a pixel with spectral signature: **NDVI=0.65, NIR=4500, SWIR=2000**

100 Trees Vote:

- Tree 1 → **Forest** 🌲
- Tree 2 → **Forest** 🌲
- Tree 3 → **Agriculture** 🌾
- Tree 4 → **Forest** 🌲
- Tree 5 → **Forest** 🌲
- …
- Tree 100 → **Forest** 🌲

Vote Count:

- **Forest:** 78 votes
- **Agriculture:** 22 votes

Final Prediction: Forest (78%)

Confidence: 78% confidence in prediction

> ⓘ **Prediction Confidence**
>
> The proportion of votes can be interpreted as **confidence**. Higher consensus → more confident prediction.

Random Forest Hyperparameters

Key parameters to tune:

Parameter	Description	Typical Values	Impact
n_trees	Number of trees in forest	50-500	More trees → better performance (diminishing returns)
max_depth	Maximum depth of each tree	10-50 or None	Deeper → more complex, risk overfitting
min_samples_split	Min samples to split node	2-10	Higher → simpler trees, less overfitting
max_features	Features per split	$\sqrt{n}$ or $\log_2(n)$	Balance between accuracy and diversity
bootstrap	Use bootstrap sampling	True	Almost always True for RF

Google Earth Engine Default

GEE's `ee.Classifier.smileRandomForest()` defaults: - **numberOfTrees:** 100 - **variablesPerSplit:** $\sqrt{n}$ (automatic) - **minLeafPopulation:** 1

Random Forest Advantages

1. **High Accuracy**
    - Often achieves excellent performance out-of-the-box
    - Handles complex non-linear relationships
2. **Robust to Overfitting**
    - Ensemble averaging reduces variance
    - Harder to overfit than single decision tree
3. **Feature Importance**
    - Quantifies which features matter most
    - Helps understand classification drivers
4. **Handles Missing Data**
    - Can work with incomplete feature sets
    - Robust to noisy data
5. **No Normalization Needed**
    - Works with features on different scales
    - Simplifies preprocessing
6. **Efficient**
    - Fast training (parallelizable)

Feature Importance

Understanding Feature Importance

Question: Which spectral bands/indices contribute most to classification accuracy?

Feature Importance measures the contribution of each feature to the model's predictions.

Calculation Methods:

1. **Mean Decrease in Impurity (MDI)**
   - How much each feature reduces impurity (Gini)
   - Averaged across all trees
   - Default in most implementations
2. **Permutation Importance**
   - Measure accuracy drop when feature is randomly shuffled
   - More reliable but slower

Interpretation:

- **High importance:** Feature strongly discriminates classes
- **Low importance:** Feature adds little information
- **Zero importance:** Feature not used by any tree

Example: Feature Importance for Palawan

Land Cover Classification (7 classes)

Top Features:

Rank	Feature	Importance	Use Case
1	**NDVI**	0.285	Forest vs. non-forest
2	**NIR (B8)**	0.192	Vegetation density
3	**SWIR1 (B11)**	0.156	Moisture content
4	**NDWI**	0.128	Water detection
5	**Red (B4)**	0.089	Vegetation health
6	**NDBI**	0.067	Urban areas
7	**Elevation**	0.045	Topographic context

Insights:

- **NDVI dominant:** Vegetation indices most important
- **NIR crucial:** Distinguishes vegetation types
- **SWIR useful:** Separates forest from agriculture
- **NDWI essential:** Water body identification
- **Elevation helps:** Mountains → forest, lowlands → agriculture

Actionable: - Focus on acquiring high-quality NIR and SWIR data - Ensure accurate NDVI calculation - Include DEM for improved accuracy

Feature Importance Visualization

Typical Output:

```
# Example feature importance from trained RF
Feature Importances:

NDVI                               0.285
NIR                                0.192
SWIR1                              0.156
NDWI                               0.128
Red                                0.089
NDBI                               0.067
Elevation                          0.045
Blue                               0.020
Green                              0.018
```

Applications:

- **Feature selection:** Remove low-importance features
- **Data collection priorities:** Focus on important bands
- **Model interpretation:** Understand classification logic
- **Domain validation:** Does importance match EO theory?

Accuracy Assessment

Training/Test Data Splitting

Critical Decision: How to split data for training vs. validation?

Common Split Ratios:

80/20	80%	20%	Standard (sufficient data)
70/30	70%	30%	More robust validation
60/40	60%	40%	Limited training data
50/50	50%	50%	Very small datasets

Google Earth Engine: Use `.randomColumn()` to assign splits

Splitting Strategies:

1. Random Split (most common)

```
# Add random column
data = data.randomColumn('random')

# Split 80/20
training = data.filter(ee.Filter.lt('random', 0.8))
testing = data.filter(ee.Filter.gte('random', 0.8))
```

2. Stratified Split (recommended) - Maintain class proportions in both sets - Important for imbalanced datasets - Ensures all classes in test set

3. Spatial Split - Training from one region, testing from another - Tests geographic transferability - More realistic for operational use

Why Accuracy Assessment?

- **Quantify performance:** How good is the classification?
- **Compare models:** Which classifier performs better?
- **Identify weaknesses:** Which classes are confused?
- **Build confidence:** Can we trust the map for decisions?
- **Report to stakeholders:** Scientific credibility

> **Golden Rule**
>
> **ALWAYS** use **independent test data** that was NOT used for training. Otherwise, you’re measuring memorization, not generalization.

Confusion Matrix

A table showing **predicted classes** vs. **actual classes** for test data.

Example: 5-class classification

	Forest	Agriculture	Water	Urban	Bare		
Actual ↓							
Forest	**85**	8	0	2	5	100	85%
Agriculture	12	**73**	0	5	10	100	73%
Water	0	1	**95**	2	2	100	95%
Urban	3	7	3	**82**	5	100	82%
Bare	5	11	2	9	**78**	105	74%
Total	105	100	100	100	100	505	
Producer's Acc	81%	73%	95%	82%	78%		**OA: 82.6%**

Accuracy Metrics Explained

Overall Accuracy | Producer's Accuracy | User's Accuracy | Kappa Coefficient

Definition: Percentage of correctly classified samples

$$\text{Overall Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Samples}} = \frac{85 + 73 + 95 + 82 + 78}{505} = \frac{413}{500} = 82.6\%$$

Interpretation: - Simple, intuitive metric - **Limitation:** Can be misleading with imbalanced classes

Example: 95% accuracy sounds great, but if 95% of pixels are forest, a "classify everything as forest" model achieves 95%!

Common Confusion Patterns

Example: Forest vs. Agriculture confusion

	Predicted Forest	Predicted Agriculture
Actual Forest	85	**8** ← Confusion
Actual Agriculture	**12** ← Confusion	73

Why confusion occurs:

1. **Spectral Similarity**
   - Tree crops (coconut, fruit trees) look like forest
   - Young forest regeneration looks like agriculture
2. **Mixed Pixels**
   - Agroforestry systems
   - Forest edges with agriculture
3. **Temporal Variability**
   - Agriculture changes rapidly (planting, harvesting)
   - Single-date imagery may miss phenology
4. **Class Definition Ambiguity**
   - Where does “forest” end and “tree plantation” begin?

Best Practices: Training Data Collection

Practical Tips for High-Quality Training Samples:

Sample Size Guidelines:

Common (forest)	50	100-200	More coverage
Moderate (agriculture)	50	100-150	Capture variability
Rare (bare soil)	30	50-100	Get what you can

Sampling Strategies:

1. Stratified Random: - Distribute samples across study area - Avoid clustering in one region - Ensure all sub-types represented

2. Purposive Sampling: - Target known pure pixels - Use high-resolution imagery (Google Earth) - Field visits when possible

Quality Criteria:

✓ **Pure Pixels** - Homogeneous within polygon - Avoid edges and mixed areas - Use ≥3x3 pixel minimum areas

✓ **Clear Definition** - Unambiguous class membership - Document class definitions - Use consistent interpretation rules

✓ **Temporal Match** - Training data date matches imagery - Account for phenology (crops) - Update for multi-temporal analysis

Philippine-Specific Tips: - Use **PhilSA Space+ Dashboard** for recent imagery - Leverage **NAMRIA land cover** for reference - Consult **LGU land use plans** for urban areas - Use **Google Street View** for ground truth

Improving Classification Accuracy

Better Training Data | More Features | Better Model | Post-Processing

- **More samples:** 50-100 per class minimum
- **Better quality:** Pure pixels, clear boundaries
- **Balanced:** Equal samples per class
- **Representative:** Cover all variations within class
- **Distributed:** Spatial coverage across study area

Google Earth Engine

Why Google Earth Engine?

Challenges with Desktop GIS:

- ✗ Downloading large satellite data
- ✗ Storage requirements (TBs)
- ✗ Computational limitations
- ✗ Manual preprocessing
- ✗ Time-consuming workflows

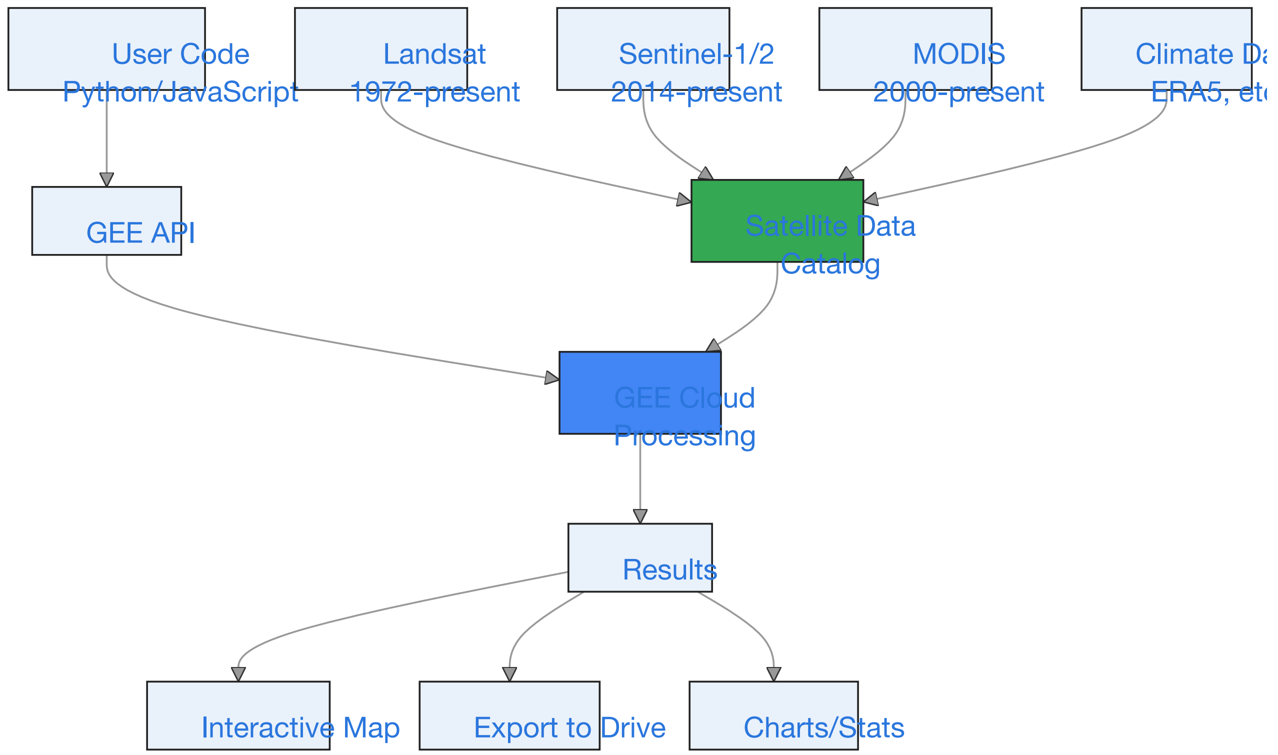
Google Earth Engine Solutions:

- ✓ **Petabyte-scale catalog** (Landsat, Sentinel, MODIS…)
- ✓ **Cloud computing** (no downloads)
- ✓ **Pre-processed data** (atmospherically corrected)
- ✓ **Scalable processing** (parallel)
- ✓ **Free for research** & education

Perfect for This Course

GEE enables us to process years of Sentinel-2 data for entire provinces in minutes!

Google Earth Engine Architecture



Key Concepts:

- **Server-side processing:** Code runs on Google servers, not your laptop
- **Lazy evaluation:** Operations queued, executed only when needed (e.g., map display, export)
- **Parallel processing:** Automatically distributed across many machines

GEE Data Catalog Highlights

Relevant for Philippine EO:

Optical Imagery:

- **Sentinel-2 MSI:** 10m, 13 bands, 5-day revisit
- **Landsat 8/9 OLI:** 30m, 11 bands, 16-day revisit
- **MODIS:** 250-500m, daily, long time series

Radar:

- **Sentinel-1 SAR:** 10m, cloud-free, day/night

Terrain:

- **SRTM DEM:** 30m elevation
- **ALOS World 3D:** 30m (better for SE Asia)

Climate:

- **ERA5:** Hourly reanalysis (temp, precip)
- **CHIRPS:** Daily rainfall
- **MODIS LST:** Land surface temperature

Pre-processed Products:

- **Hansen Global Forest Change:** Annual tree cover loss
- **ESA WorldCover:** Global 10m land cover
- **Global Surface Water:** Water occurrence

GEE Code Editor vs. Python API

JavaScript (Code Editor)

- **Pros:**
  - Browser-based (no installation)
  - Interactive map interface
  - Built-in visualization
  - Great for exploration
- **Cons:**
  - Limited to GEE environment
  - Harder to integrate with other tools
  - Less powerful for data science

Use Case: Quick exploration, visualization

Python API

- **Pros:**
  - Integrate with NumPy, Pandas, scikit-learn
  - Jupyter notebooks
  - Reproducible workflows
  - Version control (Git)
  - Advanced analysis
- **Cons:**
  - Requires installation/setup
  - Slightly steeper learning curve

Use Case: Reproducible research, production workflows

Our Approach

We'll use **Python API** with **geemap** library for best of both worlds: Python ecosystem + interactive maps!

GEE Random Forest Workflow

High-level workflow for today's lab:

```python
import ee
import geemap

# 1. Initialize GEE
ee.Initialize()

# 2. Load Sentinel-2 imagery
s2 = ee.ImageCollection('COPERNICUS/S2_SR') \
    .filterBounds(palawan_boundary) \
    .filterDate('2024-01-01', '2024-12-31') \
    .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 20))

# 3. Compute composite and indices
composite = s2.median()
ndvi = composite.normalizedDifference(['B8', 'B4']).rename('NDVI')
ndwi = composite.normalizedDifference(['B3', 'B8']).rename('NDWI')

# 4. Stack features
```

Philippine NRM Applications

Forest Monitoring (DENR, REDD+)

Challenges:

- 7,641 islands, 30 million hectares
- Cloud cover year-round
- Rapid deforestation in some areas
- Limited ground-based monitoring

RF Classification Helps:

- Annual forest cover maps
- Deforestation hotspot detection
- REDD+ MRV (Monitoring, Reporting, Verification)
- Protected area encroachment

Example: Palawan Biosphere Reserve

- **Area:** 1.1 million hectares
- **Protection:** UNESCO MAB, NIPAS
- **Threats:** Illegal logging, mining, agriculture

Workflow:

1. Annual Sentinel-2 composites (2016-2024)
2. RF classification (primary forest, secondary, non-forest)
3. Change detection (forest loss/gain)
4. Alert system for encroachment
5. Reports for PCSDS, DENR

Agricultural Monitoring (DA, PhilRice)

Rice Production Monitoring:

- **Goal:** Estimate planted area and yield
- **Importance:** Food security planning
- **Traditional method:** Field surveys (slow, expensive)

RF Approach:

- Multi-temporal Sentinel-2 (capture crop phenology)
- Training data from field surveys
- Classify: Rice, Other crops, Non-ag
- Area calculation per province/municipality
- Early warning for production shortfalls

Example: Central Luzon Rice Bowl

Classes: - Rice (wet season) - Rice (dry season) - Vegetables - Fallow/bare - Non-agricultural

Features: - NDVI time series (captures growth cycle) - LSWI (Land Surface Water Index) - EVI (Enhanced Vegetation Index)

Validation: - PhilRice field surveys - DA crop cut experiments

Urban Expansion Monitoring (NEDA, HLURB)

Metro Manila & Major Cities:

- **Rapid urbanization:** 3-5% annual growth
- **Planning needs:** Infrastructure, transport, housing
- **Environmental concerns:** Loss of green space, flooding

RF Classification:

- Urban/built-up
- Roads and infrastructure
- Vegetation (parks, trees)
- Bare soil (construction sites)
- Water bodies

Applications:

1. **Urban growth tracking**
    - Compare 2015 vs. 2024
    - Identify sprawl patterns
    - Predict future expansion
2. **Green space monitoring**
    - Urban vegetation loss
    - Park accessibility analysis
3. **Flood risk**
    - Impervious surface mapping
    - Drainage planning
4. **Compliance**
    - Illegal construction detection
    - Zoning violations

Water Resources (NWRB, LGUs)

Applications:

Surface Water Mapping:

- Rivers, lakes, reservoirs
- Seasonal variations
- Drought monitoring
- Flood extent mapping

RF Advantages: - NDWI as strong predictor - Multi-temporal captures seasonal changes - Can detect small water bodies

Watershed Management:

- Land cover within watersheds
- Forest cover (water regulation)
- Agriculture (erosion risk)
- Urban (runoff)

Example: Angat Dam Watershed - Critical for Metro Manila water supply - Monitor forest cover changes - Detect encroachment - Sediment risk assessment

Disaster Response (NDRRMC, PAGASA)

Post-Typhoon Damage Assessment:

Challenge: - Philippines: ~20 typhoons/year - Rapid assessment needed for relief - Cloud-free imagery rare after storms

RF Classification Approach:

1. **Pre-event baseline:** Land cover map
2. **Post-event imagery:** First clear Sentinel-2
3. **Damage classes:**
    - Intact forest/vegetation
    - Damaged vegetation
    - Exposed soil/landslides
    - Flooded areas
    - Building damage (requires very high res)

Example: Typhoon Odette (2021)

- **Affected:** Visayas, Mindanao
- **Assessment needs:**
    - Agricultural damage (coconut, rice)
    - Forest destruction
    - Coastal erosion
    - Flooded areas

RF Workflow: - Pre-typhoon: December 2021 composite - Post-typhoon: January 2022 composite - Classify: Intact, Damaged, Destroyed - Area statistics per municipality - Priority areas for relief

Recap: Session 1 Theory

What We Learned:

✓ Supervised classification workflow

✓ Decision trees: Intuitive but limited

✓ Random Forest: Ensemble of trees - Bootstrap sampling - Random feature selection - Majority voting

✓ Feature importance: Which bands matter?

✓ Accuracy assessment: - Confusion matrix - Overall, Producer's, User's accuracy - Kappa coefficient

✓ Google Earth Engine: Cloud-based EO

Key Takeaways:

1. Random Forest is **powerful** for EO classification
   - High accuracy
   - Handles non-linear relationships
   - Robust to overfitting
2. **Training data quality** is critical
   - Representative samples
   - Balanced classes
   - Sufficient quantity
3. **Feature engineering** improves results
   - Spectral indices (NDVI, NDWI)
   - Multi-temporal data
   - Auxiliary data (DEM)
4. **Accuracy assessment** builds confidence
   - Always use independent test data
   - Understand confusion patterns

Break

15-minute break before hands-on lab

- 🔶 Stretch
- 🔶 Coffee/water
- 🔶 Check your setup: - C

Coming up: Hands-on lab with Palawan land cover classification!

Part B: Hands-on Lab

Lab Overview

What We'll Build: Palawan Land Cover Classification using Random Forest

Steps:

1. Setup and authentication
2. Load Sentinel-2 imagery
3. Create cloud-free composite
4. Calculate spectral indices
5. Prepare training data
6. Train Random Forest model
7. Generate classification map
8. Validate accuracy
9. Analyze results

Duration: ~1.5 hours

Study Area: Palawan Province

- **Location:** Western Philippines
- **Area:** ~14,649 km^2
- **Significance:** UNESCO Biosphere Reserve
- **Diversity:** Forest, mangroves, agriculture, urban

Classes: 1. Forest 2. Agriculture 3. Water 4. Urban 5. Bare Soil

Spectral Indices for Classification

Key Features Beyond Raw Bands:

Vegetation Indices:

NDVI	(NIR - Red) / (NIR + Red)	Vegetation vigor
EVI	2.5 × (NIR - Red) / (NIR + 6×Red - 7.5×Blue + 1)	Enhanced sensitivity in high biomass

```
# Calculate NDVI
ndvi = image.normalizedDifference(['B8', 'B4']).rename('NDVI')

# Calculate EVI
evi = image.expression(
    '2.5 * ((NIR - RED) / (NIR + 6*RED - 7.5*BLUE + 1))',
    {'NIR': image.select('B8'),
     'RED': image.select('B4'),
     'BLUE': image.select('B2')
    }).rename('EVI')
```

Water & Built-up Indices:

NDWI	(Green - NIR) / (Green + NIR)	Water bodies
MNDWI	(Green - SWIR) / (Green + SWIR)	Water/wetlands (better separation)
NDBI	(SWIR - NIR) / (SWIR + NIR)	Built-up areas

```
# Calculate water indices
ndwi = image.normalizedDifference(['B3', 'B8']).rename('NDWI')
mndwi = image.normalizedDifference(['B3', 'B11']).rename('MNDWI')

# Calculate built-up index
ndbi = image.normalizedDifference(['B11', 'B8']).rename('NDBI')
```

Why MNDWI? Better separates water from built-up areas than NDWI (uses SWIR instead of NIR)

Lab Instructions

Follow along in Jupyter notebook:

```
../notebooks/session1_hands_on_lab_student.ipynb
```

Student version: With TODO markers for exercises

Instructor version: Complete solutions

Tips for Success

- Read markdown cells carefully before running code
- Experiment with parameters
- Visualize intermediate results
- Ask questions when stuck!

Expected Outputs

By the end of the lab, you will have:

1. ✓ Interactive map of Palawan with Sentinel-2 composite
2. ✓ Calculated spectral indices (NDVI, NDWI, NDBI)
3. ✓ Trained Random Forest classifier (100 trees)
4. ✓ Land cover classification map
5. ✓ Confusion matrix and accuracy metrics
6. ✓ Feature importance ranking
7. ✓ Area statistics per land cover class
8. ✓ Exported classification to Google Drive

Accuracy Target: >80% overall accuracy

Session 1 Summary

Theory Concepts:

- Supervised classification workflow
- Decision trees → Random Forest
- Bootstrap aggregating
- Random feature selection
- Feature importance
- Accuracy assessment metrics
- Confusion matrix interpretation

Tools:

- Google Earth Engine
- Python API (geemap)
- Sentinel-2 imagery

Practical Skills:

- GEE authentication
- ImageCollection filtering
- Composite generation
- Spectral index calculation
- Training data preparation
- RF model training
- Classification execution
- Accuracy validation
- Map visualization

Philippine Context:

- Palawan land cover mapping
- DENR forest monitoring
- DA agricultural mapping
- NDRRMC disaster response

Next Session Preview

Session 2: Advanced Palawan Land Cover Lab

- Multi-temporal composites (dry/wet season)
- Advanced feature engineering (GLCM texture)
- Topographic features (DEM)
- 8-class detailed classification
- Hyperparameter tuning
- Change detection (2020 vs. 2024)
- Deforestation analysis
- Stakeholder reporting

Preparation:

- Complete Session 1 exercises
- Review confusion matrix analysis
- Think about classification improvements

Resources

Documentation:

- Google Earth Engine: https://developers.google.com/earth-engine
- geemap: https://geemap.org
- Sentinel-2: https://sentinel.esa.int/web/sentinel/missions/sentinel-2
- Random Forest paper: Breiman (2001) - Machine Learning 45:5-32

Philippine EO:

- PhilSA: https://philsa.gov.ph
- NAMRIA: https://namria.gov.ph
- DOST-ASTI PANDA: https://panda.stamina4space.upd.edu.ph

Course Materials:

- GitHub: [repository link]
- Datasets: [Google Drive link]

Thank You!

Questions?

- Email: skotsopoulos@neuralio.ai
- Office Hours: [schedule]

🚀 Open: `session1_hands_on_lab.ipynb`