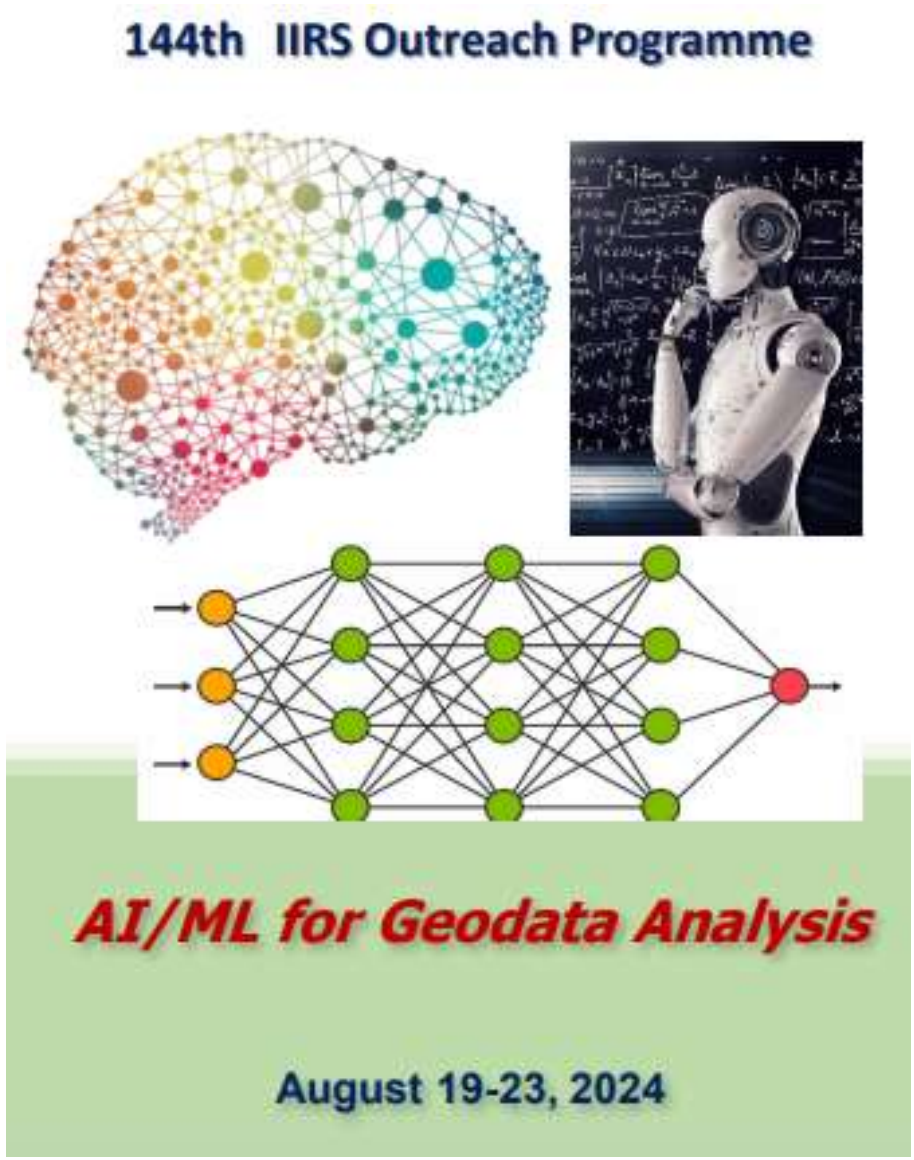


Machine Learning through Google Earth Engine

kamal@iirs.gov.in

Learning Objectives

Machine Learning task accomplished through GEE

Basic Coding in GEE for ML

Machine learning through Google Earth Engine

Machine Learning using Google Earth Engine is a powerful approach for analyzing and interpreting large-scale geospatial data

GEE is a cloud-based platform that provides access to a vast archive of satellite imagery and geospatial datasets, enabling users to perform planetary-scale analysis

Key Concepts

Google Earth Engine

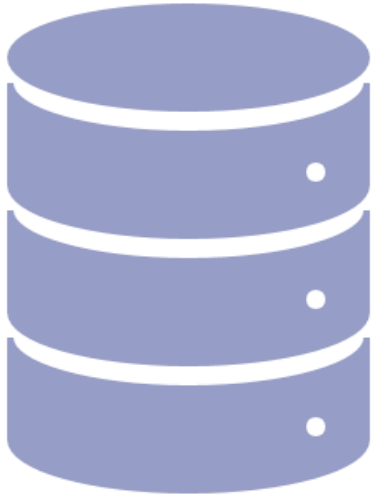
A cloud-based platform for processing and analyzing geospatial data	Provides a large repository of satellite imagery, including Landsat, Sentinel, MODIS, and more	Allows users to write and execute JavaScript or Python code directly in the web-based code editor

Machine Learning

A subset of artificial intelligence that involves training algorithms to learn from and make predictions or decisions based on data	In GEE, ML is typically used for tasks like classification, regression, clustering, and anomaly detection

Common ML Techniques in GEE

- Supervised Classification
  - Example: Land cover classification
  - Process
    - Training Data Collection: Create or import training samples
    - Feature Extraction: Extract relevant features from the imagery
    - Model Training: Train a classifier using the training data
    - Prediction: Apply the trained model to classify the entire image or a region of interest
    - Accuracy Assessment: Evaluate the classification results using validation data



Common ML Techniques in GEE

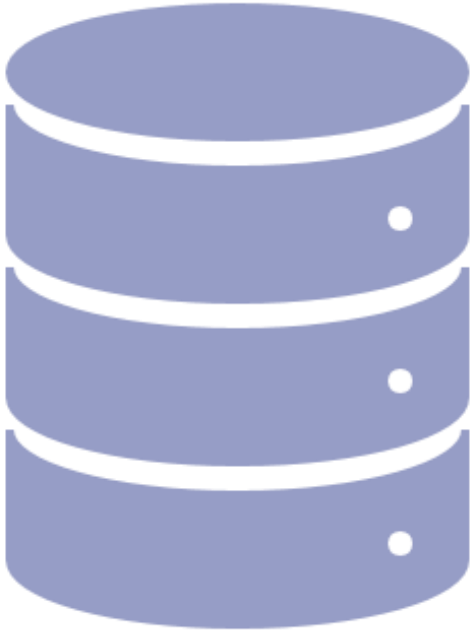
- Unsupervised Classification
  - Example: Clustering similar pixel values in an image without prior labels
  - Process
    - Feature Extraction: Select relevant features
    - Clustering: Apply clustering algorithms to group pixels into classes based on similarity
    - Interpretation: Manually label the clusters based on expert knowledge or additional data

Common ML Techniques in GEE

- Regression Analysis
  - Example: Estimating continuous variables like vegetation biomass or soil moisture
  - Process
    - Training Data Collection: Gather training samples with known values of the target variable
    - Feature Extraction: Extract features related to the target variable
    - Model Training: Train a regression model
    - Prediction: Use the trained model to predict the target variable across the region of interest

Common ML Techniques in GEE

- Time Series Analysis
  - Example: Monitoring changes in vegetation over time
  - Process
    - Data Collection: Compile time-series data
    - Feature Extraction: Calculate temporal features
    - Model Training: Train models to detect changes or anomalies over time




इसरो isro
iirs
Example: Land Cover Classification using Random Forest in GEE
DEMO

Key Steps in the Example

- Load and Preprocess Data: Sentinel-2 imagery is loaded, filtered, and preprocessed
- Training Data: Training data is collected, typically using polygons labeled with the correct land cover class
- Model Training: A Random Forest classifier is trained using the training data
- Classification: The trained model is applied to classify the entire region
- Visualization: The classified map is displayed with different colors representing different land cover classes

Tools and Resources

GEE CODE EDITOR:

The primary interface for writing and running GEE scripts

GEE PYTHON API:

Allows integration with python, useful for combining GEE with other ML libraries like tensorflow or scikit-learn

TRAINING DATASETS:

Can be created manually or imported from external sources

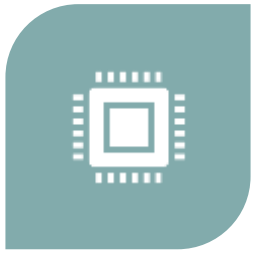
DOCUMENTATION AND TUTORIALS:

GEE provides extensive documentation and community-contributed tutorials

Advantages of Using GEE for ML

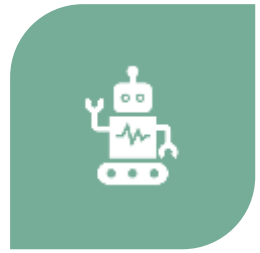
SCALABILITY:

GEE handles large-scale data processing on the cloud, removing the need for local computation resources



ACCESSIBILITY:

Provides easy access to global datasets, making it ideal for environmental monitoring, disaster management, and more



INTEGRATION:

Combines powerful geospatial analysis with machine learning tools, facilitating complex workflows

Conclusion

- Machine learning in Google Earth Engine enables the effective analysis of geospatial data, offering powerful tools for tasks such as land cover classification, regression analysis, and time series monitoring

- By leveraging GEE's cloud-based infrastructure, users can perform large-scale analyses efficiently and apply sophisticated ML models to derive meaningful insights from satellite imagery and other geospatial data

CODE for Random Forest Classifier

```
// Load Sentinel-2 image collection.
var image =  imageCollection.filterDate('2023-01-01', '2023-12-
31').filterBounds(ddn).median();
// Define a region of interest.
// Select bands and create an image with spectral indices.
var bands = ['B2', 'B3', 'B4', 'B8'];
var image = image.select(bands).addBands(image.normalizedDifference(['B8',
'B4']).rename('NDVI'));

var displayparameters = {
  min: 1000,
  max: 4500,
  bands: ['B8', 'B4', 'B3'],
};

Map.addLayer(image,displayparameters,"Image");

// Load training data (e.g., land cover classes).
var label = "Class";
var training = Water.merge(Forest).merge(Urban);
```

CODE for Random Forest Classifier

```
// Extract features for training.
var trainingimage = image.sampleRegions({
  collection: training,
  properties: [label],
  scale: 10
})

// Divide input samples into Training & Testing
var traingData = trainingimage.randomColumn();
var trainSet = traingData.filter(ee.Filter.lessThan('random',0.8));
var testSet = traingData.filter(ee.Filter.greaterThanOrEquals('random',0.8));

// Train a Random Forest classifier.

//Define the classifier parameters:
var classifier = ee.Classifier.smileRandomForest({numberOfTrees:100, variablesPerSplit: 2,
minLeafPopulation: 1, bagFraction: 0.5,
seed: 0});
```

CODE for Random Forest Classifier

```
//Train the classifier on the training dataset
var classifier = ee.Classifier.smileRandomForest(100).train(trainSet, label, bands);

// Classify the image
// Use the trained classifier to classify the entire dataset or the ROI

var classified = image.classify(classifier);

// Display the results.
Map.centerObject(ddn, 10);
Map.addLayer(classified, {min: 1, max: 3, palette: ['green', 'blue', 'red']}, 'Land
Cover');

//Get information about the trained classifier
print('Results of trained classifier', classifier.explain());

//Get a confusion matrix and overall accuracy for the training sample
var trainAccuracy = classifier.confusionMatrix();
print('Training error matrix', trainAccuracy);
print('Training overall accuracy', trainAccuracy.accuracy());
```

CODE for Random Forest Classifier

```
//Get a confusion matrix and overall accuracy for the validation sample.
testSet = testSet.classify(classifier);
var validationAccuracy = testSet.errorMatrix(label,'classification');
print('Validation error matrix', validationAccuracy);
print('Validation accuracy', validationAccuracy.accuracy());


Export.table.toAsset({
  collection: training,
  description: 'LCsample2023',
  assetId: 'LCsample2023'
});


Export.table.toDrive({
  collection: training,
  description: 'LCsample2023',
  fileFormat: 'SHP'
});
```

Thank you

Questions?