

National University of Sciences and Technology
School of Electrical Engineering and Computer Science
Department of Computing

CS 471 Machine Learning
Fall 2024

Term / Semester Project

NDVI-Based Crop Classification for Rice and Cotton
Using Machine Learning

Announcement Date: 24th Oct 2024

Due Date: 18th Dec 2024 at 11:55 PM (ON LMS)

Table of Contents

1	Overview	2
2	Problem Description: Agriculture Crop Type Classification (Rice and Cotton)	2
	2.1 Dataset Description for Rice and Cotton Classification	3
	2.1.1 Dataset Details	3
	2.2 Cross-Validation Strategy	4
	2.3 Techniques	4
3	Deliverable	5
	3.1 The code files	5
	3.1.1 Implementation Details	6
	3.1.2 Training and Testing Code	6
	3.1.3 Output in Printed Format	7
	3.2 Report	7
	3.3 Presentation:	8
4	Submission Instructions:	8
5	Grading Rubric	8
6	Extra Credit:	8

1 Overview

Machine learning is a powerful tool used to address complex classification problems across various fields, enhancing human sustainability. In this project, students will apply machine learning techniques to tackle a crop classification problem using NDVI data, focusing specifically on distinguishing between rice and cotton crops.

Teams: The project will be done in teams. Each team can be comprised of at-least one member and no more than two members.

Register your groups on this [google form](#) by 28th Oct 2024

<https://forms.gle/PqKC6Lfj1wX2drmk6>

2 Problem Description: Agriculture Crop Type Classification (Rice and Cotton)

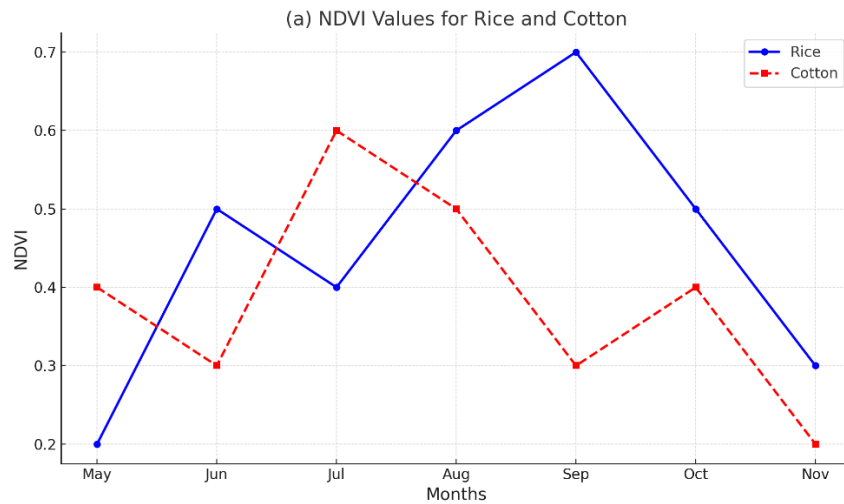
Effective crop classification is essential for better agricultural management, resource allocation, and policy formulation. This problem focuses on classifying two specific crop types—rice and cotton—based on their spectral characteristics, particularly using Normalized Difference Vegetation Index (NDVI) values over a growing season.

Normalized Difference Vegetation Index:

NDVI (Normalized Difference Vegetation Index) measures vegetation health by comparing Near-Infrared (NIR) and Red light reflectance using the formula:

$$NDVI = \frac{(NIR - Red)}{(NIR + Red)}$$

It is crucial for crop classification as it differentiates crop types based on growth patterns, tracks unique spectral signatures of crops like rice and cotton, and aids in precise resource allocation and yield prediction. The given graph shows NDVI values plotted against months from **May to November** for **rice** and **cotton**. Rice's NDVI peaks later in **September**, while cotton's NDVI peaks earlier around **July-August**, reflecting their distinct growth cycles. This visualization highlights how NDVI tracks crop health over the growing season.



2.1 Dataset Description for Rice and Cotton Classification

The dataset contains NDVI values collected for rice and cotton crops over a specific growing season across three years. NDVI (Normalized Difference Vegetation Index) is a widely used vegetation index derived from satellite imagery to monitor crop health and growth patterns.

2.1.1 Dataset Details

NDVI Time Series:

1. The dataset comprises 12 NDVI values per crop type for a complete season, representing two NDVI measurements per month over a six-month period.
2. Each NDVI time series reflects the unique growth patterns of either rice or cotton, showcasing distinct spectral signatures over the growing season.
3. The dataset is labeled, indicating whether the NDVI series corresponds to rice or cotton.

Data Across Three Years: Data Across Three Years

- The dataset contains NDVI data collected for rice and cotton crops over **three consecutive years**: 2021, 2022, and 2023.
- The table below shows the **distribution of data** for each year:

Year	2021	2022	2023
Rice	420	4688	920
Cotton	2884	12412	11778

Explanation of Data Imbalance

- As seen in the table, the dataset is **highly imbalanced**, with significantly fewer samples for **rice** compared to **cotton** across all three years.

- This imbalance can impact model performance, potentially causing bias towards the majority class (cotton).

Mitigation Strategies

To handle the data imbalance, techniques like **oversampling** (increasing rice samples), **undersampling** (reducing cotton samples), or **SMOTE (Synthetic Minority Oversampling Technique)** can be applied to balance the class distribution, improving model performance and generalization.

Dataset availability

https://drive.google.com/drive/folders/13tEXAYJtGd2vgN_thpKlhgKEZUv5Nqku?usp=sharing

Use your SEECs Google Workspace ID to access this dataset. Do not Send requests to access the data from your private emails.

2.2 Cross-Validation Strategy

To evaluate the performance of classification models, a cross-validation approach using the three-year data is proposed. The approach involves training the model on data from two years and testing it on data from the remaining year, covering all possible combinations.

Cross-Validation Combinations

- Train on Year 1 and Year 2, Test on Year 3
- Train on Year 1 and Year 3, Test on Year 2
- Train on Year 2 and Year 3, Test on Year 1

This cross-validation strategy ensures that the model is evaluated thoroughly across different years, helping to assess its generalization capability and robustness over varying growing conditions.

2.3 Techniques

Student will apply the following techniques on the dataset. Students will also apply Grid Search to optimize the parameters for each algorithm.

1. Supervised Learning Algorithm

Supervised learning is suitable here since labelled NDVI time series data is available. The model can be trained to learn patterns from this labelled data and classify unseen NDVI time series.

- i. Ensemble Methods (Bagging / Boosting)
 - a. XGBoost (Extreme Gradient Boosting)
 - b. Bagging (Bootstrap Aggregation)
 - c. Random Forest
- ii. Support Vector Machine (SVM)

2. Grid Search for Parameter Optimization

Grid Search is a systematic method for hyperparameter optimization that evaluates all possible combinations of specified parameter values to identify the best-performing configuration for a machine learning model.

Algorithm	Parameters to Search
XGBoost (Boosting)	learning_rate (eta), max_depth, n_estimators
Bagging (Bootstrap Aggregation)	n_estimators, base_estimator__max_depth
Random Forest	n_estimators, max_depth, min_samples_leaf
Support Vector Machine (SVM)	kernel, C (regularization), gamma (for RBF kernel)

2. Unsupervised Learning Approach

Students will implement the unsupervised learning approach to identify inherent patterns in the NDVI time series, potentially distinguishing between rice and cotton without using the labels of the data. Students will also report the clustering results using **Principal Component Analysis (PCA)** for dimensionality reduction and also **without using** the Principal Component Analysis.

- K-Means Clustering
- Hierarchical Clustering
- DB SCAN
- Gaussian Mixture Models

3 Deliverable

Three deliverables

- Code files
- Short presentation
- Short Report

3.1 The code files

- Implement your task using an IDE, Python Notebook, or Google Colab. Submit your work as a **.py file**. You may use PyCharm, an IDE for Python development, to complete the task.
- Neatly Written Code:
 - The code should be clean, organized, and easy to read.
 - Follow Python coding standards, with consistent indentation and clear variable naming.
- Documented Code:
 - Add clear and meaningful comments explaining each step of the process.

- Include docstrings for each function and module to provide context about their functionality and usage.
- **Modular Code:**
 - Break down the code into modular functions and classes.
 - Create separate modules for data preprocessing, model training, evaluation, and visualization.

3.1.1 Implementation Details

- **Data Augmentation (if needed):**
 - If data augmentation is deemed necessary to enhance the model's performance (e.g., for supervised techniques), include augmentation strategies such as adding noise to NDVI values, time-series shifting, or scaling.
 - Clearly document the augmentation process, specifying how it modifies the **original** NDVI data.

3.1.2 Training and Testing Code

For each supervised and unsupervised technique, the following components must be demonstrated:

- **Supervised Techniques:**
 - Implement and train all models given above
 - Show the feature importance for models that support this functionality (e.g., Random Forest, XGBoost).
 - Report Grid Search parameter optimization results for each model.
 - Use the specified **cross-validation strategy** (train on two years, test on one year).
 - Include the following evaluation metrics for each run:
 - **Accuracy**
 - **F1-Score**
 - **Precision**
 - **Recall**
 - **Confusion Matrix** (for both classes: rice and cotton, as well as overall performance)
- **Unsupervised Techniques:**
 - Implement and test algorithms given above
 - Report results without PCA and with PCA:
 - Conduct clustering directly on the original NDVI time-series dataset (without PCA).
 - Apply PCA for dimensionality reduction and report the clustering results after PCA.
 - For evaluation, use metrics such as:
 - **Cluster Purity**
 - Confusion Matrix (for classes identified after clustering against true labels)

3.1.3 Output in Printed Format

The Python code or Notebook should print the following metrics after each model's training and testing:

- **Accuracy, F1-Score, Precision, Recall for each class** (rice and cotton) as well as the overall dataset.
- **Confusion Matrix**, visually displayed using heatmaps for easier interpretation.

3.2 Report

The report should contain the following:

- **Pre-processing**
 - Apply **Normalization, Data Cleaning**, and **Time-Series Augmentation** to prepare the NDVI dataset.
- **Performance Metrics**
 - For **Supervised Learning**:
 - Evaluate the performance of each algorithm using:
 - **Accuracy**
 - **Recall**
 - **F1-Score**
 - **Precision**
 - **Confusion Matrix** (for both individual classes like rice and cotton and overall performance).
 - Report the **Grid Search results**, including the best parameters and cross-validation scores for each algorithm.
 - For models that support it (e.g., Random Forest, XGBoost), include **Feature Importance** to highlight the contribution of each feature.
 - For **Unsupervised Learning**:
 - Evaluate the clustering performance using:
 - **Cluster Purity**
 - **Confusion Matrix** (compare clusters against true labels: rice and cotton).
 - Perform clustering on:
 - **Original Data (without PCA)**.
 - **Reduced Data (with PCA)**, applying dimensionality reduction before clustering.
- **Model Comparison**
 - Compare and contrast the **top-performing models** for supervised and unsupervised learning tasks, explaining their strengths and weaknesses.

3.3 Presentation:

Presentation Make a short presentation of your work illustrating a coherent flow. You need to give presentation after submission. It is also the part of evaluation

4 Submission Instructions:

The code files, short presentation and a short report should be zipped and uploaded to LMS as a single ZIP file Please name your submission ZIP file as <YourCMS-ID>_<YourName>.ZIP [example [5153452_Ahmad-Khan.zip](#)]

5 Grading Rubric

The project is graded out of **100 points** in total, distributed as follows:

- **Supervised Algorithm:** 40 points (40%)
 - Implementation(include augmentation and data balancing): 10 points
 - Model Accuracy & Evaluation: 10 points
 - Feature Importance: 4 points
 - Grid Search Parameter Optimization 4
 - Results Comparison: 2 points
- **Unsupervised Algorithm:** 20 points (20%)
 - Implementation(include augmentation and data balancing): 10 points
 - Cluster Purity & Evaluation (Without PCA): 4 points
 - Cluster Purity & Evaluation (Without PCA): 4 points
 - Results Comparison: 2 points
- **Report:** 20 points (20%)
 - Clarity & Structure: 5 points
 - Analysis & Insights: 10 points
 - Proper use of metrics and comparisons: 5 points
- **Presentation:** 20 points (20%)
 - Clarity of explanation: 10 points
 - Quality of slides/visual aids: 5 points
 - QA Session about your work: 5 points

This rubric ensures fair grading based on implementation, evaluation, reporting, and presentation quality.

6 Extra Credit:

4% Extra credit will be given to the students implemented a modular and documented solution.

1% Extra credit given to the student participating in the class during presentation for healthy and fruitful discussion during project presentations.