Project Report

CDS3005 – Foundations of Data Science

Class ID: BL2025260100756

Slot: B11+B12+B13+B14+E14

Fall Semester 2025-26

Submitted by

Disha Naveen

(22BCE100776)

Submitted to

Ms. Prerna

Research Scholar (IFT),

School of Computing Science Engineering

Index

Sr. No	Content	Page No.
1.	Introduction	5
2.	Dataset Description	6
3.	Project File structure	8
4.	Methodology	9
5.	Results	11
6.	Observations	13
7.	Inference	13
8.	Conclusion	13

1. Introduction

Crop rotation is an essential agricultural practice that enhances soil fertility, reduces pest infestations, and improves crop yield. However, farmers often face difficulty in selecting the best crop for the next season due to overlapping pests, unsuitable soil types, or season mismatches.

Picking the next crop is not just about what grows—it's about what else grows with it: pests. In Indian agriculture, where seasons (Kharif, Rabi, Zaid) and diverse soil types matter, choosing the wrong rotation can invite overlapping pests and yield loss.

Goal: build a simple, reproducible system that recommends next-season crops with minimal pest overlap, while staying season-appropriate and soil-compatible—usable by farmers and hobbyists without lab soil tests.

This project aims to assist farmers and agricultural hobbyists in making informed decisions about next-season crop selection by predicting the most suitable crops based on:

- Current crop
- Season
- Soil type
- Pest overlap

The model uses machine learning (RandomForest) to recommend the optimal crop rotation strategy while minimizing pest risks.

2. Dataset Description

The dataset was curated manually with a focus on Indian agricultural practices, accounting for the three primary Indian crop seasons:

- Rabi (winter crops such as Wheat, Barley, Gram)
- Kharif (monsoon crops such as Rice, Maize, Cotton)
- Zaid (summer crops such as Melons, Vegetables)

Each crop entry includes:

- Crop Name
- Season (Rabi, Kharif, Zaid)
- Soil Type (e.g., Alluvial, Black, Arid & Desert, Laterite, Red & Yellow)
- Associated Pests (comma-separated list of pests affecting the crops.

Files

- indian_crops_dataset.xlxs: full crop dataset with crop, season, soil, and pest information

Crop Name	Season	Soil Type	Pests
Wheat	Kharif	Arid and Desert	Armyworm, Cutworm, Whitefly, Mealybug
Rice	Rabi	Arid and Desert	Whitefly, Mealybug, Leaf miner, Cutworm
Maize	Kharif	Alluvial	Pod borer, Cutworm
Barley	Rabi	Arid and Desert	Grasshopper, Stem borer
Sorghum	Rabi	Laterite	Cutworm, Whitefly
Bajra	Zaid	Red	Armyworm, Thrips, Leaf miner
Pulses	Rabi	Laterite	Pod borer, Stem borer, Leaf miner, Grasshopper
Gram	Zaid	Arid and Desert	Leaf miner, Aphids
Lentil	Zaid	Arid and Desert	Root-knot nematode, Whitefly
Mustard	Zaid	Arid and Desert	Stem borer, Whitefly, Fruit fly, Grasshopper
Sugarcane	Rabi	Arid and Desert	Leaf miner, Mealybug
Cotton	Zaid	Arid and Desert	Fruit fly, Mealybug, Whitefly
Jute	Rabi	Red	Armyworm, Root-knot nematode
Groundnut	Zaid	Black	Thrips, Pod borer, Shoot borer
Soybean	Kharif	Arid and Desert	Mealybug, Cutworm, Armyworm, Stem borer

- pairwise_crops_dataset.xlxs: pairwise combinations for pest overlap calculations

Current Crop	Current Crop	Soil Type	Current Season	Next Season	Pest Overlap
Wheat	Barley	Arid and Desert	Kharif	Rabi	0
Barley	Cotton	Arid and Desert	Rabi	Zaid	0
Barley	Gram	Arid and Desert	Rabi	Zaid	0
Barley	Turmeric	Arid and Desert	Rabi	Zaid	0
Barley	Lentil	Arid and Desert	Rabi	Zaid	0
Maize	Sunflower	Alluvial	Kharif	Rabi	0
Sorghum	Cauliflower	Laterite	Rabi	Zaid	0
Sorghum	Onion	Laterite	Rabi	Zaid	0
Sorghum	Ginger	Laterite	Rabi	Zaid	0
Pulses	Ginger	Laterite	Rabi	Zaid	0
Gram	Soybean	Arid and Desert	Zaid	Kharif	0
Gram	Wheat	Arid and Desert	Zaid	Kharif	0

- validation.csv: validation split for testing the model - This was derived from both datasets via code

3. Project File Structure

```
CropRotationPestControl/
├── src/
│   ├── train.py              # trains RandomForest, saves model
│   ├── predict.py            # makes predictions & recommends next crops
│   ├── data_preprocessing.py # parsing pests, normalizing strings, encoders
│   ├── model_utils.py        # saving/loading model, encoders, datasets
├── data/
│   ├── Originaldata.csv
│   ├── pairwise.csv
│   ├── validation.csv
├── models/
│   ├── crop_recommender.pkl      # trained RandomForest model
│   ├── label_encoders.pkl        # encoders for categorical features
│   ├── original_dataset.pkl      # reference dataset
│   ├── model_architecture.json   # saved metadata
├── docs/
│   ├── README.md
│   ├── report.pdf
├── output/
│   ├── predictions.csv       # generated recommendations
│   ├── metrics.json          # accuracy/validation results
├── requirements.txt
└── .gitignore
```

4. Methodology

4.1 Preprocessing

- Label Encoding(sklearn.preprocessing.LabelEncoder): categorical features such as crop name, soil type, and season were encoded using LabelEncoder.
- Pest Parsing: pests stored as comma-separated values were split into sets for comparison.
- String Normalization: crop names and soil types were normalized (lowercased, whitespace removed).

4.2 Model Training

- Algorithm: RandomForest Classifier was chosen for its robustness and ability to handle categorical + mixed-type data.

- Features Used: soil type, season, and pest encodings.

- Target Variable: the recommended next crop.

- Training/Validation Split: data split into train and validation subsets (validation.csv)
- Why Random Forest?
    - Handles mixed/categorical-derived features robustly.
    - Nonlinear interactions between crop × season × soil are captured without complex feature engineering.
    - Good default bias–variance tradeoff; stable with small-to-medium tabular datasets.
- Key parameters (chosen for reliability over tuning):
    - n_estimators=100: adequate ensemble size for stability.
    - random_state=42: exact reproducibility.
    - Other RF defaults kept standard to avoid overfitting with a hand-curated dataset.
- Training target: Pest Overlap (classification).
- Validation: standard train_test_split with test_size=0.2.

4.3 Plantability Score Formula(Decision Layer)

To recommend the next crop, a Plantability Score was calculated:

Plantability Score= (0.5*pest_score)+ (0.25*seasonscore) + (0.25*soilscore)

Explanation: The pest score is inversely proportional to no of pest overlaps, the season score full(25/25) if the current crop season map to the rotation cycle of the next crop and same goes with soil, 25/25 is rewarded is current crop soil type matches with the next crop soil type.

- pest_score (0–50): inversely proportional to common pests/
- season_score (0 or 25): full points if the candidate's season equals the mapped next season.
- soil_score (0 or 25): full points if the candidate's soil equals the current soil.

This weighting reflects reality: pest avoidance is the primary driver (50%), but planting window (season) and soil suitability are decisive constraints (25% each).

4.4 Rotation Logic (Domain Prior)

- Next-season map encodes basic Indian rotation cycle: Kharif → Rabi, Rabi → Zaid, Zaid → Kharif.
- Candidate generation: for a given current crop, all other crops in Originaldata.csv are considered as candidates.

4.5 Persistence & Reproducibility

We store all artifacts for deterministic reuse:

- models/crop_recommender.pkl — trained RandomForest.
- models/label_encoders.pkl — dict of fitted LabelEncoders (one per categorical field).
- models/original_dataset.pkl — cached Originaldata for quick candidate lookup.
- models/model_architecture.json — JSON metadata (model type, parameters, feature list).

4.6 Visualization of Crop Rotation

The system can generate a rotation recommendation table showing:

- Current crop and its season/pest
- Recommended next-season crops
- Pest overlap count
- Plantability score

5. Results

1. *Example Recommendation Output*

a. Wheat → Top-5

```
recommend_next_crops_with_score("Wheat")
```

```
🌱 Current Crop Information:
   Crop: Wheat
   Season: Kharif
   Soil Type: Arid and Desert
   Pests: Armyworm, Cutworm, Mealybug, Whitefly

📅 Next Planting Season: Rabi

🌾 Recommended Next Crops:
```

Crop	Season	Soil Type	Pests in Common	Common Pests	Plantability Score
Barley	Rabi	Arid and Desert	0	None	100.00
Sugarcane	Rabi	Arid and Desert	1	Mealybug	87.50
Cabbage	Rabi	Arid and Desert	1	Cutworm	87.50
Pulses	Rabi	Laterite	0	None	75.00
Gram	Zaid	Arid and Desert	0	None	75.00

b. Tea→ Top-5

```
recommend_next_crops_with_score("Tea")
```

```
🌱 Current Crop Information:
   Crop: Tea
   Season: Rabi
   Soil Type: Laterite
   Pests: Leaf miner, Whitefly

📅 Next Planting Season: Zaid

🌾 Recommended Next Crops:
```

Crop	Season	Soil Type	Pests in Common	Common Pests	Plantability Score
Cauliflower	Zaid	Laterite	0	None	100.00
Millets	Zaid	Laterite	0	None	100.00
Ginger	Zaid	Laterite	0	None	100.00
Chilli	Zaid	Laterite	0	None	100.00
Groundnut	Zaid	Black	0	None	75.00

Bajra → Top-5

```
recommend_next_crops_with_score("Bajra")
```

🌱 Current Crop Information:
Crop: Bajra
Season: Zaid
Soil Type: Red
Pests: Armyworm, Leaf miner, Thrips

📅 Next Planting Season: Kharif

🌾 Recommended Next Crops:

Crop	Season	Soil Type	Pests in Common	Common Pests	Plantability Score
Coffee	Kharif	Red	1	Armyworm	83.33
Maize	Kharif	Alluvial	0	None	75.00
Sesame	Kharif	Laterite	0	None	75.00
Peas	Kharif	Arid and Desert	0	None	75.00
Garlic	Kharif	Arid and Desert	0	None	75.00

2. Validation Metrics

```
Accuracy: 0.9117647058823529

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        17
           1       0.82      1.00      0.90        14
           2       0.00      0.00      0.00         3

    accuracy                           0.91        34
   macro avg       0.61      0.67      0.63        34
weighted avg       0.84      0.91      0.87        34
```

6) Observations

1. Zero-overlap isn't enough: Even with 0 pest overlaps, a candidate can score below 100 if its season doesn't match the rotation cycle or soil differs. The weighting makes this explicit.

2. Indian soil types matter: Recommendations are grounded in the soil taxonomy used in India (e.g., Arid & Desert, Alluvial, Laterite), so outputs are locally meaningful.

3. Season-aware recommendations: The Kharif→Rabi→Zaid rotation map enforces planting windows, preventing unrealistic suggestions.

4. Accessible, not lab-dependent: The system does not require N–P–K (Nitrogen/Phosphorus/Potassium) measurements. It's intentionally generalized so smallholders and hobbyists can apply it without a soil lab.

7) Inference

- When there is a trade-off, avoiding pests is prioritized (50% of score), but being in-season and soil-appropriate (25% each) are decisive filters.

- The model and scoring layer together reflect three constraints—biological (pests), temporal (season), and edaphic (soil)—which aligns with agronomic best practices for rotation.

8) Conclusion

A compact, interpretable workflow can offer practical rotation advice:

- It blends learning (RandomForest on curated pairs) with domain logic (rotation map + plantability scoring).

- It yields Indian-context suggestions without requiring lab tests, making it usable by farmers and hobbyists alike.
- Future steps: enrich with regional pest incidence, weather, and yield economics to move from risk-aware to profit-aware rotation planning.