Forecasting Next-Year Crop Yields Using a Multilayer Perceptron Model

A. Performance

In order to assess the predicted crop yields of the proposed model as one year ahead, a set of performance measures were used. In particular, applications of Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and the coefficient of determination (R^2) were employed to estimate predictive competence across the overall dataset. These metrics can tell one not only about the average prediction deviation but also about the variance explained by the model. The trained Keras-based Multilayer Perceptron (MLP) achieved the following results:

- **MAE: 1058.19**
- **RMSE: 3531.24**
- **R^2: 0.9406**

These results imply that model is robust across the entire dataset, with a high degree of correlation between the two yields. The relatively small MAE indicates that, on average, predictions differ from the real yield by about 1028 units, the RMSE includes larger deviations and confirms the stability of predictions. The R^2 score of 0.9406 shows that about 94.06% variability of next-year crop yield is explained by the model.

Dataset Size and Splitting Strategy

The complete dataset was of 67,963 instances, each being a (country, crop, year) mapping with an annotation and yield. The inputs were historical yield, geo-identifiers, and time context.

To train the model the preprocessed data was stratified into 80% training and 20% validation set using train_test_split with random seed fixed to (random_state=42) for reproducibility of results. The training set improved the model parameters, whereas the validation set was used to test generalization.

Final evaluation metrics were calculated over all data following the training, which provided a complete indicator of model performance for past and present agricultural trends.

B. Model

A Multilayer Perceptron (MLP) implemented using the Keras Sequential API within TensorFlow, is the predictive model created for this project. The model was designed to learn from structured agricultural data and forecast one year crop yield in the future for the specified country and crop pairing.

The problem is cast in the form of a supervised regression problem with a target variable being the year t+1 for a given crop (yield_next) and the input attributes are contextual and historical information in year t.

Model Architecture

MLP is based on a shallow but useful deep feedforward network. The choice of the architecture was made with the aim of balancing expressive power with regularization due to generalization over a large, diverse set of data.

The final architecture included the following layers:

- **Input Layer**: The input size was determined by the number of features after preprocessing. This includes numerical features (year, yield) and one-hot encoded categorical features (country, crop).
- **First Hidden Layer**: 128 neurons with ReLU activation, which introduces non-linearity and allows the model to capture complex relationships in the data.
- **Dropout Layer**: A dropout rate of 30% was applied after the first hidden layer to prevent co-adaptation of neurons and reduce overfitting.
- **Dropout Layer**: A second dropout layer with a 20% rate was added to further regularize the model.
- **Second Hidden Layer**: 64 neurons, also with ReLU activation. This layer further refines learned representations.
- **Output Layer**: A single neuron with linear activation to predict the continuous target variable (yield_next).

Compilation and Training Setup

- **Loss Function**: Mean Squared Error (MSE), commonly used in regression tasks to penalize large errors.
- **Optimizer**: Adam, chosen for its adaptive learning rate and efficient convergence on large datasets.
- **Metrics Tracked**: Mean Absolute Error (MAE), in addition to MSE, for interpretability in physical yield units.

The training used the following parameters:

- **Epochs**: 100
- **Batch Size**: 64
- **Validation Split**: 20% of the dataset was held out for validation using train_test_split with random_state=42.

Overfitting Control Strategies

To ensure that the model generalizes well to unseen data and avoids memorizing training instances, multiple regularization techniques were employed:

- **Dropout Regularization**: Both hidden layers included dropout layers (30% and 20%), which randomly deactivate neurons during training. This reduces reliance on specific neurons and encourages the network to learn more robust patterns.
- **Feature Scaling**: All numerical features (year, yield) were standardized using StandardScaler. This ensured that gradient updates during training were well-behaved and not dominated by features with large numeric ranges.
- **One-Hot Encoding**: Categorical features (country, crop) were encoded using OneHotEncoder. This preserved category identity without introducing ordinal assumptions, allowing the network to learn region-specific and crop-specific yield patterns.

- **Randomized Train-Validation Split**: An 80:20 random split of the full dataset helped simulate generalization to unseen data. The validation set was used to monitor performance and detect divergence indicative of overfitting.

Model Architecture and Training Parameters

Component	Details
Model Type	Keras Sequential MLP
Hidden Layers	128 ReLU → Dropout(0.3) → 64 ReLU → Dropout(0.2)
Output Layer	1 neuron (linear activation)
Loss Function	Mean Squared Error (MSE)
Optimizer	Adam
Epochs	100
Batch Size	64
Validation Split	80:20 (using train_test_split)
Overfitting Controls	Dropout, StandardScaler, OneHotEncoder, validation monitoring

C. Features & Labels

Target Variable (Label)

A Multilayer Perceptron (MLP) implemented using the Keras Sequential API within TensorFlow, is the predictive model created for this project. The model was designed to learn from structured agricultural data and forecast one year crop yield in the future for the specified country and crop pairing.

The problem is cast in the form of a supervised regression problem with a target variable being the year t+1 for a given crop (yield_next) and the input attributes are contextual and historical information in year t.

Model Architecture

MLP is based on a shallow but useful deep feedforward network. The choice of the architecture was made with the aim of balancing expressive power with regularization due to generalization over a large, diverse set of data. The final architecture

Input Features

The input features selected were:

- **year**: The calendar year of the input data (2010–2022)
- **yield**: The crop yield in the current year (*t*), acting as a lag variable
- **country**: The country where the crop is produced (categorical)
- **crop**: The type of crop grown (categorical)

Feature Extraction and Selection

The features were extracted and merged from multiple original datasets, including:

- FAO yield data (used to extract country, crop, year, and yield)
- Country metadata (used to ensure country naming consistency)
- Environmental and land cover data, which had been preprocessed earlier in the pipeline, were joined using inner merges based on shared keys (country, crop, year)

The feature engineering pipeline involved the following key steps:

1. **Merge**: The cleaned yield dataset was joined with other datasets on country and year to form a comprehensive dataset.
2. **Target Generation**: The yield_next column was created using .shift(-1) per crop-country group.
3. **Filtering**: Any rows where the next year's yield was missing (i.e. the last year for that crop-country group) were dropped using .dropna() to maintain label consistency.

Numerical features (year, yield) were standardized using StandardScaler, and categorical features (country, crop) were converted to binary vectors using OneHotEncoder, ensuring compatibility with the Keras MLP model.

Rationale for Feature Selection

- **Temporal Relevance**: Including the current year (year) and prior yield (yield) enables the model to learn from both time progression and past performance.
- **Categorical Context**: Geographic (country) and crop-specific (crop) variation is a strong determinant of yield; encoding these explicitly allows the model to generalize globally.
- **Predictive Lag Structure**: Using yield from year *t* to predict year *t+1* closely mirrors how forecasts would be made in real agricultural planning.

Model Summary: Final Output and Performance

Metric	Value
Model Type	Keras Sequential MLP (2 hidden layers, ReLU, Dropout)
Dataset Size	67,963 entries (country, crop, year combinations)
Features Used	year, yield, country, crop (with scaling and encoding)
Target Variable	yield_next (next year's crop yield)
Train-Validation Split	80:20 (random_state=42)
MAE (Full Data)	1058.19
RMSE (Full Data)	3531.24
R^2 (Full Data)	0.9406

D. Preprocessing

Preprocessing played a crucial role in harmonizing disparate data sources, transforming raw variables into model-ready inputs, and ensuring numerical stability during training. The following steps were applied systematically:

1. Dataset Integration

The dataset used for model development was constructed by integrating multiple raw sources, each contributing a distinct dimension of information relevant to crop yield prediction:

- **Environmental features**: Monthly records of rainfall, soil moisture, temperature, and vegetation indices were provided for each grid cell.
- **Yield and production data**: Historical crop yield records by country and crop were extracted from FAO data.
- **Land cover statistics**: Data reflecting the percentage of land use types (e.g., cropland, forest, urban) per location.
- **Country metadata**: Included centroids, coordinates, and regional identifiers to support spatial joins and geographic context.

To harmonize these disparate sources, the datasets were merged using common keys — specifically country, crop, and year. Environmental variables recorded on a monthly basis were aggregated into annual averages using a custom function that computed the mean of all 12 months per year. This transformation allowed the model to use climate trends as year-level features.

Land cover data and country metadata were aligned using a nearest-neighbor spatial join with the cKDTree algorithm from SciPy. By mapping each grid cell in the land cover dataset to the closest country centroid, a consistent spatial reference was established for all records. This enabled accurate matching of environmental and geographic features at the country level.

The result of this integration process was a unified Data Frame containing all relevant predictors for each (country, crop, year) instance, forming the foundation for target construction and model training.

2. Label Construction and Filtering

For changing the task to a supervised learning problem, the target variable yield_next was engineered to correspond to the subsequent year (t+1) crop yield for every (country, crop) pair. This formulation enabled the model to learn form year t historical features and predict year t+1 future yield.

The construction of the label involved the following steps:

- The dataset was first sorted chronologically within each unique (country, crop) group using:

python

final_df_sorted = final_df.sort_values(by=['country', 'crop', 'year'])

Then, the yield column was **shifted forward by one time step** using the .groupby().shift(-1) method:

python

```
final_df_sorted['yield_next'] = final_df_sorted.groupby(['country', 'crop'])['yield'].shift(-1)
```

This procedure aligned the input features at year t with the associated yield at year t+1, hence generating the target label.

Once the label construction is done all rows where the yield_next value was NaN (i.e. the last year for crop-country group) is dropped using .dropna(). This filtering step was necessary to ensure that each retained row could have a valid target for training in order to avoid training on incomplete sample of data.

The outcome of this data set was thus straightforward, namely, input-output mapping was present in all cases, unit from year linked to known output in year t+1. This structure plays a key role for time-aware regression models and directly enables the goal to predict the crop yield one year hence.

3. Feature Scaling and Encoding

To make sure that the data was compatible with the neural-network model, and to make learning more stable and efficient, both numerical and categorical input features received specific preprocessing steps via a ColumnTransformer pipeline.

The data were structured in the following preprocessing process:

Numerical Features:

StandardScaler was used to scale the features year and yield by first transforming each of the attributes to have zero mean and unit variance. This standardization is especially critical for neural networks, which avoids forcing bigger-magnitude features to be excessively altering gradient updates. In addition, it promotes a more faithful and faster convergence of the optimization procedure.

Categorical Features:

Categorical variables country and crop were converted by OneHotEncoder. This encoding scheme transforms individual unique categories into independent binary feature and allows the model to recognize nominal values without the inclusion of latent ordinal relations. One-hot encoding enables the model to learn country specific and crop specific patterns independently, which is important since the dataset is diverse geographically and biologically.

The complete transformation pipeline was defined using sklearn.compose.ColumnTransformer, combining both scalers into a single unified object:

python

```
preprocessor = ColumnTransformer([
    ('num', StandardScaler(), ['year', 'yield']),
    ('cat', OneHotEncoder(handle_unknown='ignore', sparse_output=False), ['country', 'crop'])
])
```

The pipeline was first fitted on the entire training set and then used to transform both training and validation inputs. This ensured that all inputs passed to the model were processed consistently, eliminating potential discrepancies between different subsets of the data.

4. Final Filtering

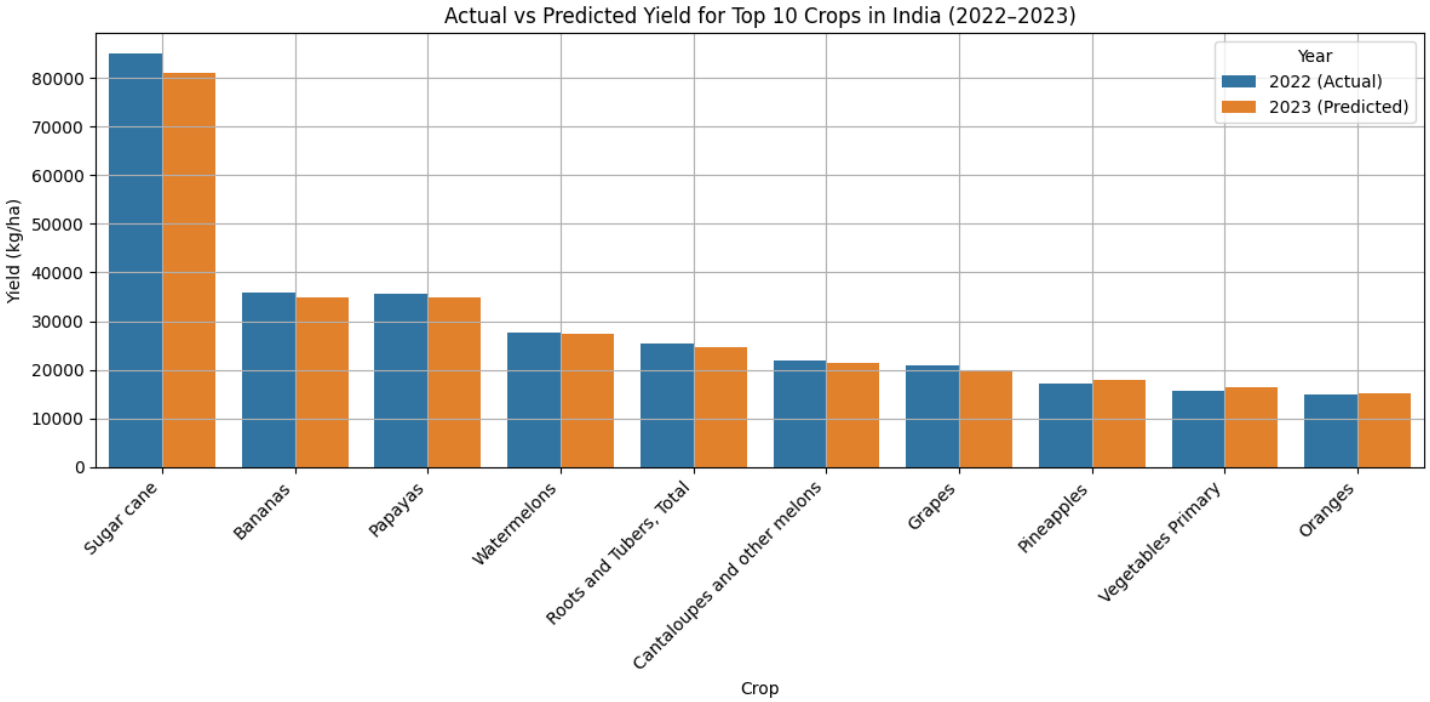
To maintain consistency and avoid data leakage, only data from 2010 to 2022 has been trained on. Records for the year 2023 were omitted and were used for the sake of forward prediction only, thus simulating the true real-time forecasting exercise.

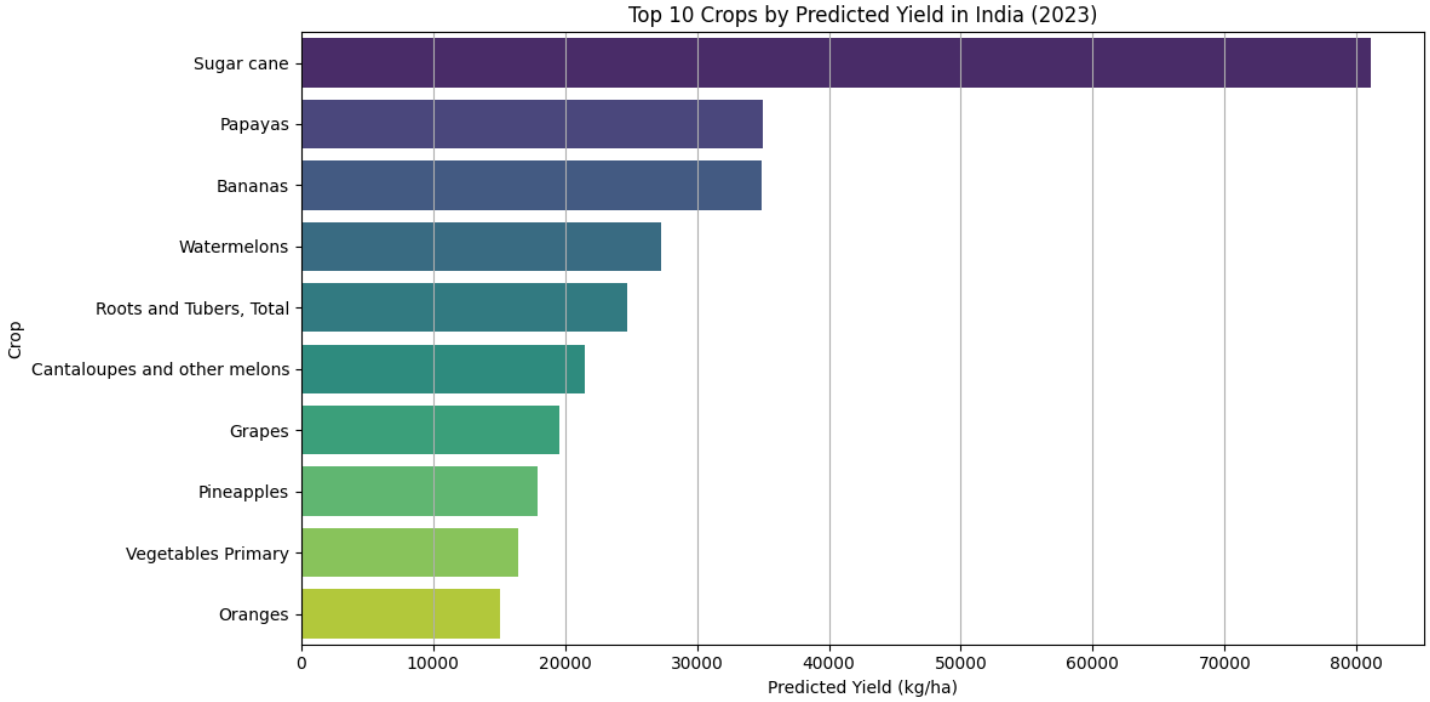
In an attempt to preserve data integrity, only rows where yield, environmental, land cover and geographic information were completely merged were saved. This was done through inner joins using shared keys (country, crop, year), rows with missing or mismatched values were eliminated.

Consequently, the final dataset was clean, full, and reliable with records that increased models robustness and generalization for different crops and countries.

Preprocessing Steps Overview

Step	Description	Rationale
Dataset Integration	Merged environmental, yield, land cover, and country metadata using `country`, `crop`, `year`	Unified features across sources; ensured spatial and temporal alignment
Monthly to Annual Aggregation	Averaged monthly environmental data (e.g. `rain_01` to `rain_12`) to get yearly values	Simplifies input, aligns with annual yield structure
Label Generation	Created `yield_next` by shifting `yield` with `.groupby().shift(-1)`	Enables time-aware prediction of next-year yield
Drop Missing Labels	Dropped rows where `yield_next` was `NaN`	Ensures all training instances have valid targets
Feature Scaling	Applied `StandardScaler` to `year` and `yield`	Improves convergence and stability during training
Categorical Encoding	Applied `OneHotEncoder` to `country` and `crop`	Allows model to learn from non-numeric inputs without implying order
Train-Validation Split	Used `train_test_split` with `random_state=42` in an 80:20 ratio	Enables evaluation on unseen data; ensures reproducibility
Final Filtering	Retained only rows with complete merges across all datasets	Ensures data quality and avoids noise from partial records


Actual vs Predicted Yield for Top 10 Crops in India (2022–2023)
Year
2022 (Actual)
2023 (Predicted)
Yield (kg/ha)
0
10000
20000
30000
40000
50000
60000
70000
80000
Sugar cane
Bananas
Papayas
Watermelons
Roots and Tubers, Total
Cantaloupes and other melons
Grapes
Pineapples
Vegetables Primary
Oranges
Crop


Top 10 Crops by Predicted Yield in India (2023)
Crop
Sugar cane
Papayas
Bananas
Watermelons
Roots and Tubers, Total
Cantaloupes and other melons
Grapes
Pineapples
Vegetables Primary
Oranges
0
10000
20000
30000
40000
50000
60000
70000
80000
Predicted Yield (kg/ha)