Report: Forecasting Yield of Crop Products using MLP

Farhad Oghazian (April 2025)

# A. Performance

To evaluate the prediction performance of our multilayer perceptron (MLP) model, we computed three key regression metrics: R² score, Root mean squared error (RMSE) and mean absolute error (MAE). These metrics assess the model’s prediction accuracy in predicting crop yields on 2022 dataset (i.e., test set). Since crop yields are positively skewed, we applied a logarithmic transformation to the yield values before training, to reduce the influence of extreme outliers, and satisfying the assumption of normally distributed outputs in MLP modelling. Accordingly, model predictions were transformed back using the inverse transformation (np.expm1) prior to computing the evaluation metrics.

Data Overview:

The input dataset consisted of 67760 data instances in total. Each representing a specific cropcountry-year combination with associated climate, soil, and land use features.

For splitting strategy, we followed a temporal split strategy:

* Training set: all data from years 2010 to 2021 excluding data related to a random year between 2010 to 2021 (as validation set) and the year 2022 (i.e., as test set) was used for training. Data used for training had 57445 instances and used to optimize model weights.
* Validation set: data related to one randomly selected year (i.e., 2019) was used as the validation set. Data instances used for validation were 5141 in total. The validation set were used for hyperparameter tuning and early stopping as well.
* Test set: the year 2022 set to evaluate forecasting for a future unseen year, aligning with the task goal. (5174 data instances used for test-set)

Metric Definitions and Justification:

Mean Absolute Error (MAE):

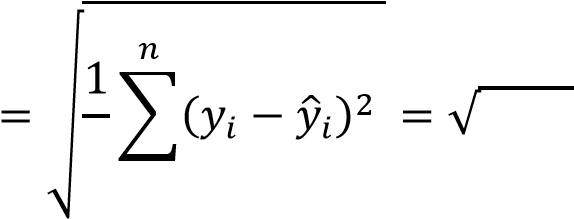
1

𝑀𝐴𝐸 = |𝑦 − 𝑦 |

𝑛

MAE is the average of the absolute differences between predicted and actual values. It is more robust to outliers and provides interpretable units (kg/ha), appropriate to report to stakeholders.

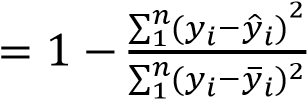
## Root Mean Square Error (RMSE)

 𝑅𝑀𝑆𝐸 𝑀𝑆𝐸

𝑛

RMSE measures the square root of the average of squared prediction errors. It penalizes larger errors more heavily and is useful for detecting the impact of large deviations. Like MAE it has interpretable units(kg/ha).

R² Score (Coefficient of Determination):

𝑅 ,

where 𝑦 is the predicted value and 𝑦 is the actual value, and in python we used sklearn.metrics.r2\_score to compute it efficiently. This function compares predicted and actual yield values in the test set (2022), returning a score between 0 and 1 that shows how well the model explains the variance in the target variable (yield).

In our case, MAE was chosen as the early stopping criterion, as it is more robust to extreme values—important in a dataset with right-skewed yield distributions. Nevertheless, RMSE and R² provide additional insight into error magnitude and variance explanations.

Test Set Performance (2022):

Test performance on the 2022 data (transformed back from logarithmic to real yield values) was as follows:

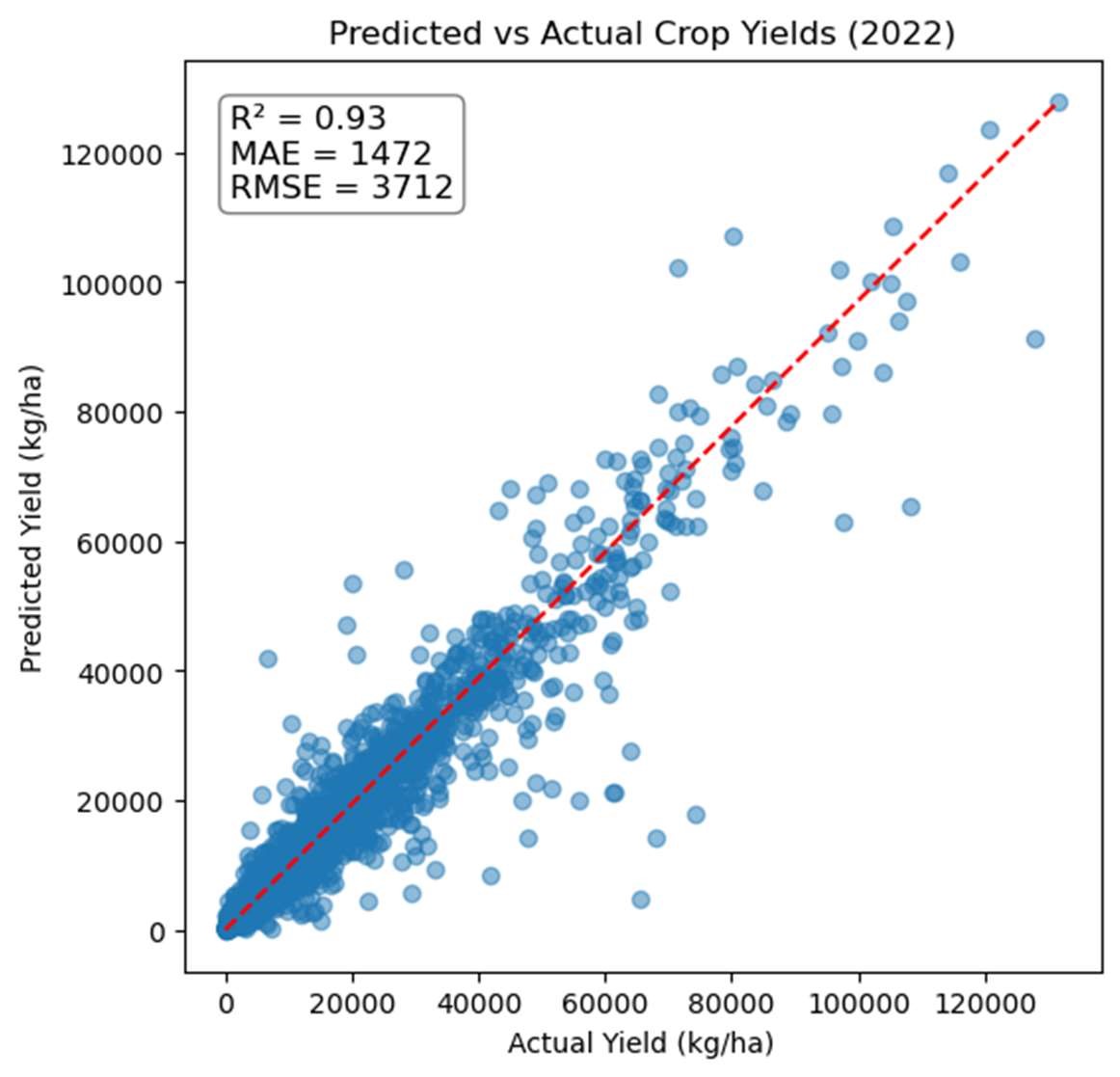
|  |  |  |
| --- | --- | --- |
| Metric | Value | Interpretation |
| MAE | 1472 kg/ha | On average, predictions were off by ~1.47 tonnes/ha |
| RMSE | 3711 kg/ha | Penalizes larger errors more; a few outliers contribute significantly to this value. |
| R² | 0.93 | The model explains 93% of the variance in yields across countries and crop types for 2022. |

.

These results demonstrate the effectiveness of our embedding-based categorical handling, log-transformations, and careful feature engineering in modelling a reliable crop yield forecasting tool.

## Prediction vs. Actual Yield

The scatter plot below compares the predicted yield values from the trained MLP model against actual yields for the year 2022. The dashed line shows the ideal line (i.e., perfect accuracy). Most points are closely aligned along this line, indicating strong performance (R² = 0.93), and low error (MAE = 1,472 kg/ha) across diverse crop-country combinations.



A few under and over predictions are visible at higher yield ranges (especially >75,000 kg/ha), and, we observe a small number of outliers with very high predicted and actual yields. These likely reflect:

 Unusually high-yielding crops in specific countries (e.g., watermelons in the Dominican Republic, or potential data quality issues in isolated cases. While these outliers were rare and did not significantly affect model performance (as reflected by high R² and low MAE), they do require an additional data cleaning with a focused on yield data quality.

To keep the report concise we focused on the core modelling objectives, we chose not to remove all the outliers (only 21 instance removed), or adjust these entries in this assignment. However, for a professional level prediction tool, it is advisable to inspect and verify extreme yields, and retrain the model on a refined dataset.

# B. Model

For modelling, a 3-layer Multilayer Perceptron (MLP) for regression used, and built using PyTorch. The structure includes embedding layers for categorical variables with fully connected layers for numerical inputs. The model was trained to predict the log-transformed yield (log(1 + Yield)).

Input → hidden\_layer\_1 → hidden\_layer\_2 → Output

##  Inputs

The model (YieldMLP) accepts numerical and categorical inputs. Categorical inputs include ‘country’ with 163 unique values (embedded to 50 dimensions) and ‘Item’ with 102 unique values (embedded to 50 dimensions). These are passed through nn.Embedding layers to learn representations. Numerical inputs are 8 features including ‘Rainf\_mean’, ‘SoilMoi’, etc. (see Section C) that were concatenated into a single input tensor.

The Dataset class (YieldDataset), wraps numeric features, categorical indices, and targets and returns a tuple: (X\_num, X\_cat, y) for each sample.

##  Hidden layers

Layer\_1 has 256, and layer\_2 256 neurons. LeakyReLU activation (with alpha = 0.01) used on both hidden layers which by trial found to have better performance compared to ReLU on our dataset.

##  Output layer

It is a single neuron for regression output (i.e., log(1+Yield))

We applied several strategies to prevent overfitting and ensure the model generalizes well to unseen data including:

Early stopping: training stops if validation MAE does not improve for 5 consecutive epochs (patience =5)

Weight decay (L2 Regularization): we used weight decay = 0.00001, in Adam optimizer. It will penalize large weights and prevents overfitting by adding regularization to the training process.

Log transformation of Yield (output): reduces the effect of extreme outliers by compressing the range of yield values into an almost normal distribution that supports the assumption of normally distributed labels for MLP modelling.

Separate validation Set: We held out an entire year of data as validation set to ensure unbiased hyperparameter tuning and as a criterion for early stopping during training and grid search tuning.

We did not use dropout, because the above strategies provided sufficient overfitting prevention and strong validation and test performance.

##  Hyperparameters & Optimization

To adjust hyperparameters we performed a grid search over the followings:

|  |  |
| --- | --- |
| Hyperparameter | Values Explored |
| Learning Rate | 0.0001, 0.001, 0.01 |
| Batch Size | 100, 200, 400 |
| Hidden Sizes | (256,256), (512,256), (512,512) |
| Weight Decay | 0.00001, 0.0001, 0.001 |

Best hyperparameters found to be, Learning Rate: 0.0001, Batch Size: 400, Hidden Sizes: (512, 512) and Weight Decay: 0.0001. The best configuration selected based on lowest MAE and highest R2 on the validation set, with performance evaluated on log-transformed yields and reported on the original scale. Also, Loss function was set to nn.MSELoss and its graph for training and validation is shown in the next page.

We used the Adam optimizer with weight decay for L2 regularization (to manage overfitting) for training. SGD did not produce satisfactory results, and its performance was very sensitive

to hyperparameters (especially the learning rate). Adam produced much more consistent and faster convergence across different configurations in grid search.

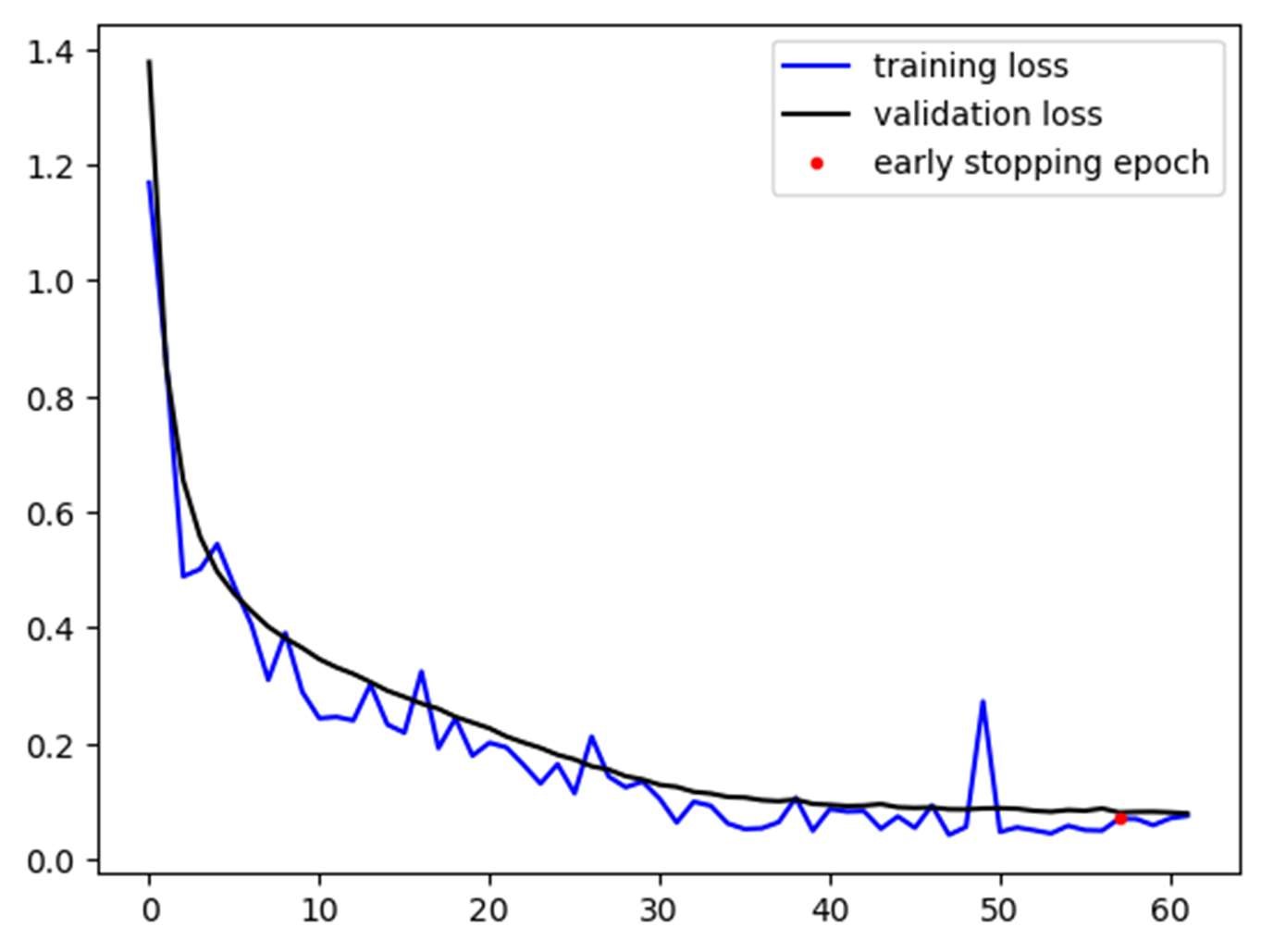


Figure 1Training vs. validation loss graph in different epochs

The graph above shows a clear decrease over epochs, in validation and training loss and early stopping happened at epoch 62. The relatively small gap between training and validation loss suggests minimal overfitting and good generalizability to unseen data. Based on test results (MAE=1472, R² = 0.93) it confirms strong generalization on unseen 2022 data. So our feature engineering, embedding, MLP structure, log-transformation of the labels and hyperparameters tuned from our grid search (learning rate, batch size, weight decay, hidden sizes) are effective and working quite well.

### C. Features & Labels

* Output variable: The target variable for our MLP regression is yield\_log:

this is log(1+Yield) per crop, per country, per year where Yield is the original values in the dataset (kg/ha). This log transformation helps our model to generalize better across both high and low yield of crops. Before taking Log we added +1 to Yield to prevent having zero in the logarithm. After prediction, the output transformed back to its original form using np.expm1 (i.e., exp(yield\_log)−1). This ensures final results are in the original scale (kg/ha), which is required for interpretability and calculating performance metric (i.e., R2, MEA, RMSE metrics).

* Input Features (Xi): We used a combination of categorical and numerical features as model inputs. The total feature set was carefully selected based on relevance and collinearity among features, and extracted from 17 datasets provided in the coursework.

#### 1. Categorical Features

ML models require numerical inputs, but country and Item are categorical (strings). We have many categories (e.g., 163 countries, 102 crops). One solution is to use embedding-based encoding for categorical variables. Categorical features like 'country' and 'Item' are first converted into integer indices using label encoding.

{data[col + '\_idx'] = data[col].cat.codes}, This assigns each unique category an integer ID. Inside the neural network, these integer indices are passed to nn.Embedding layers, which map each index to a learnable dense vector (e.g., 50dim). The size of each learnable dense embedding vector for categorical variables was chosen as min(50, (number of categories + 1) // 2).

{self.embeddings = nn.ModuleList([ nn.Embedding(num\_categories, emb\_dim) for num\_categories, emb\_dim in emb\_dims])}

1. country (country name, e.g., “Sweden”) with 163 categories
2. Item (crop type, e.g., “Apple”) with 102 categories

country and Item were label-encoded to integer indices using .astype('category').cat.codes.

These were passed through PyTorch embedding layers to learn dense vector representations during training. This avoids the sparsity of one-hot encoding (i.e., another technique for handling categorical features) while the model captures relationships between similar categories. We tried one-hot encoding and did not get satisfactory results. (e.g., R2 was negative)

#### 2. Numerical Features

The following table summarizes the 8 numerical features used, along with a short description of each. Note that year only used as a data splitting criterion and not a feature.

|  |  |
| --- | --- |
| Feature | Description |
| Land\_cover\_ percent\_mean | Average % of land classified as cropland (MODIS class 12 + 14) per country and year |
| CanopInt\_mean | Average water on plant surface(kg/m²) in each country and year |
| Rainf\_mean | Monthly average rate of rainfall (kg/m²/s) per country and year |
| ESoil\_mean | Average evaporation from soil (W/m²) per country and year |
| Snowf\_mean | Monthly Average snowfall rate (kg/m²/s) per country and year |
| TWS\_mean\* | Monthly Average Terrestrial water storage (mm) per country and year |
| SoilMoi | Monthly Average of soil moisture over 12 months, further averaged over 4 depths per each country and year (kg/m²) |
| SoilTMP | Monthly average of soil temperature over 12 months, further averaged over 4 depths per each country and year (K) |

\* TWS is expressed in millimetres of water equivalent, which represents the depth of water stored over a unit area, therefore average per country used

It is worth mentioning that all numerical features were standardized using StandardScaler, fitted(fit.transform) on the training set and applied (.transform) to validation and test set.

Feature selection

## Monthly to yearly aggregation

Soil temperatures (i.e., SoilTMP at 4 different depths), moistures (i.e., SoilMoi at 4 different depths), CanopInt\_inst\_data, ESoil\_tavg\_data, Rainf\_tavg\_data, Snowf\_tavg\_data, TVeg\_tavg\_data and TWS\_inst\_data were originally monthly per latitude per longitude, we computed yearly averages across all 12 months to match the temporal resolution of Yield\_and\_Production data.

## Land cover

For the land cover percentage data, only class 12 and class 14 were included and summed up in the analysis. Out of the 17 MODIS land cover classes class 12 corresponds to Croplands and class 14 represents Cropland/Natural Vegetation Mosaics.[[1]](#footnote-1)

Including both classes allows a more accurate representation of agricultural activity, especially in tropical regions where cropland is often mixed with natural vegetation.

## Multicollinearity reduction

To handle redundant features, we computed pairwise Spearman correlations across all numeric features and identified highly correlated soil moisture and temperature layers (ρ > 0.9). The result showed that averaged moisture features across depths and also averaged temperatures across depths are extremely correlated. To address multicollinearity among soil-related features, we aggregated highly correlated soil moisture and soil temperature across depths into two new features (SoilMoi and SoilTMP) using their annual averages. o SoilMoi: monthly average of all soil moistures at 4 depths per country per year

o SoilTMP: monthly average of all soil temperatures at 4 depths per country per year

These two new features preserve the underlying signal and allow us to remove 6 redundant inputs/features. After consolidating soil moisture and temperature features, we checked multicollinearity again and found one remaining high correlation pair: TVeg\_mean and Rainf\_mean. We confirmed the strong correlation between TVeg\_mean (TVeg : Evaporation of water from plant) and Rainf\_mean (Rainf : Rate of rainfall) using a Pearson correlation test (r= 0.94, p < 0.05). Therefore we statistically proved another redundancy and kept only Rainf\_mean in the feature set. (It seems logical because evaporation of water from plant inherently linked to water availability -i.e., rainfall)

### D. Preprocessing

Preprocessing was a critical and time intensive part of this project due to the size, resolution, and structure of the given datasets. Below is a description of preprocessing steps taken, along with their rationale and implementation strategy.

## Loading and Merging Raw Data

We began by loading 17 separate CSV files that included data on:

* Crop yields and production (from FAO)
* Monthly climate and soil features (4 CSV file for monthly SoilTMP and 4 file for SoilMoi each file related to a depth/level , CanopInt\_inst\_data, ESoil\_tavg\_data,

Rainf\_tavg\_data, Snowf\_tavg\_data, TVeg\_tavg\_data and TWS\_inst\_data )

* Land cover data
* Geospatial lookup table for country centroids

Each dataset had its own format, resolution, and anomalies (i.e, Null, zero) requiring alignment before merging.

Data cleaning and wrangling:

Summary statistics used to find anomalies in datasets and investigate raw data better, we took notes and proceeded as follows:

* In country look up table we checked for duplicates, Null values and removed countries that have 'centroid radius' = 0 (7 rows removed from lookup table)
* Only class 12 (Croplands) and class 14 (Cropland/Natural Vegetation Mosaics) columns were retained from the Land\_cover\_percent\_data for analysis, as they are most relevant to agricultural land use

##  Averaging monthly columns in climate and soil datasets per year

Based on the given units for SoilTMP [Kelvin], SoilMoi [Kg/m2], Rainf\_tavg [Kg/m2S], TVeg\_tavg [Watts/m2], ESoil\_tavg [Watts/m2], CanopInt\_inst [ Kg/m2 ], and TWS\_inst [mm], and also considering their biological importance, these variables reflect ongoing conditions that impact crops regularly, so in this report average behaviour of monthly values over the year were considered.

##  Yield\_and\_Production\_data

The formula to compute yield is Yield= Production/AreaHarvested, so the variable Production is already included in Yield, so it is decided to remove production data from the dataset. (Including both Production and Yield as features can lead to data leakage. Production is not a true input feature but a result of the same outcome we're trying to predict). Therefore, we filtered the Element column to contain only yield. And to assess the quality of the yield data, we examined the flag descriptions provided alongside each entry. According to the FAO website, the five distinct flags in our dataset are:

A: Official figure

E:Estimated value

I: Imputed value

M: Missing value (data cannot exist, not applicable) X: Figure from international organizations.

The Flag M is not acceptable for our analysis. (although flags 'I' and 'E' are estimated and imputed and represent non-official data, they were kept for modelling purposes to preserve sample size). Therefore, only data points with flag 'M' were excluded from the dataset before training.

And also irrelevant columns Domain, Element (e.g., "production"), Item code (CPC), and Flag description removed. At the end we left with a clean target variable (Yield), the key identifiers (Country, Year) and one feature (Item).

 Country Assignment Based on Coordinates:

Longitude and latitude and year were common keys between datasets, except for crop Yield\_and\_Production dataset. Assigning a country enables merging these with Yield\_and\_Production\_data. Therefore, all the datasets can be connected to each other by using appropriate keys in lookup table. For datasets containing latitude and longitude information, a custom function defined to assign each data point (longitute/latitude) to a country. The function first extracts centroid coordinates and country names from a lookup table to match spatial data points in to countries. Then assigns each (latitude, longitude) point to the nearest country centroid, using Euclidean distance under a flat-Earth assumption. If a point falls within the radius of one or more centroids, it is assigned to the nearest one; otherwise, it's matched to the closest centroid. And finally, latitude and longitude columns are dropped from datasets after country assignment and all their data grouped by country and year (i.e., averaged). That is because our target (Yield) is at country-year level and environmental features were previously at coordinate level, so aggregating to country-year makes them compatible before join/merging.

##  Merging

all cleaned and aligned datasets merged on country and year using an outer join, to retain all possible country-year combinations. And final output named merged\_df served as our master modelling dataset. However it needed:

 More cleaning o checking for nulls: the rows with null value in Yield and Item column removed

(3442 instances) o checking for nulls again: 7692 rows that were present in all climate and soil columns removed.

o After investigating multicollinearity among features and creating new features that was described in section C, We also investigate the distribution of Yield. Its graph showed extremely right skewed distribution. Checking for possible outliers we identify a small number of crop-country pairs (29 instances) exhibited unrealistically high yield values (e.g., Watermelons in the

Dominican Republic, Papayas in Guyana, etc.), with yields exceeding 200,000 – 300,000 kg/ha). Therefore it is decided to remove values above 99.97% percentile.[[2]](#footnote-2)

1. According to the reference in the dataset's README file: "In areas of the tropics where cropland sizes tend to be much smaller than a MODIS pixel, agriculture is sometimes underrepresented (i.e., labeled as natural vegetation)." Given this, class 14 is defined as: "Mosaics of small-scale cultivation (40–60%) with natural tree, shrub, or herbaceous vegetation." [↑](#footnote-ref-1)
2. Note: Although extreme yields were retained for realism, but should be reviewed by domain experts if model is going to be implemented in operational level [↑](#footnote-ref-2)