Candidate number: 291747

Introduction

This assignment consists of the ML Coursework 2025A2 Dataset, a comprehensive agricultural and environmental dataset obtained from FAOSTAT and NASA. This dataset covers over 20 years and 245 countries with 17 CSV files describing crop yield, crop production, land cover, soil moisture, precipitation, and temperature. Each file contains the same key fields, e.g., "country," "longitude," "latitude," and "year," which makes it possible to conduct analysis across them. The aim is to predict crop yield for a particular region one year in advance using a neural network, taking advantage of spatial and temporal features for improved accuracy.

Section 1: Performance

To measure the performance of the model for yield forecasting, I used multiple distance and correlation metrics to measure how close the model's predictions were to the actual ground truth value. Distance metrics are important for measuring the dissimilarity between two data points, whereas correlation metrics are useful for measuring the statistical association between two variables. These included RMSE, MAE, and R^2 values all measured on a held-out test set of data from the year 2021. The available dataset comprised of 62,738 records that represented unique values for latitude, longitude, and year. However, we only have yield data aggregated at a country-year level, so we used that data. All training data supplied to the model has to be perform on data earlier than 2021 as to limit and simulate a real comparison of the model to predicting the next future time value. Prior to measuring the model's performance, the predictions and true target values (yield) needed to be transformed back to their original scale. After training, the MLP model is tested on 2021 data, which is the only data used to evaluate its accuracy.

On the validation set, the model achieved an R^2 value as 0.7466. An RMSE of 7639.53 indicates that, on average, the predicted yield deviates from the actual yield by approximately 7639 units. Likewise, the MAE value of 5821.23 suggests that the average absolute difference between the predicted and actual yields is around 5821 units. The model thus appears to perform reasonably well for predictive purposes.

The MAE, RMSE and R^2 score calculated as shown below:

MAE (Mean Absolute Error):

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

RMSE (Root Mean Squared Error):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

R^2 score:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Where:

n is the number of samples in the validation dataset

$\hat{y}_i$ is represents the predicted yield

$\bar{y}$ is the mean (average) of the actual yield values

y_i is represents the actual yield values

In summary, the model was successful in predicting crop yields well and proved to be a reliable and accurate model at the country-year level, yielding a high R^2 value and low RMSE and MAE values.

Section 2: Multilayer Perceptron (MLP) Model

A multilayer perceptron (MLP) is a type of artificial neural network that is composed of multiple layers of connected neurons. It is frequently used in many different machine learning applications, such as classification and regression.

1. Neuron (Node) and Activation Function:

A neuron (node) in the MLP accepts input from the previous layer, applies weights to the input and calculates an output using an activation function. The activation function adds non-linearity, allowing the network to acquire complicated relationships with the data.

2. Feedforward Operation:

In an MLP, the information flows from nodes in the input layer to nodes in the output layer as there are no feedback connections, meaning that the data only flows in one direction.

3. Layers:

An MLP typically consists of one input layer, one or more hidden layers, and one output layer. The output layer provides the model's predictions, and the output corresponds to the input characteristics. Hidden layers process the incoming data and modifies it, allowing the hidden layers to learn hierarchical representations.

Model Architecture

The model featured a simple architecture that offered a trade-off between learning and generalization. The model began with an input layer hosting all the processed features, these

inputs included numerical inputs such as rainfall, soil temperature and moisture as well as categorical inputs, like country and item, which were one-hot encoded.

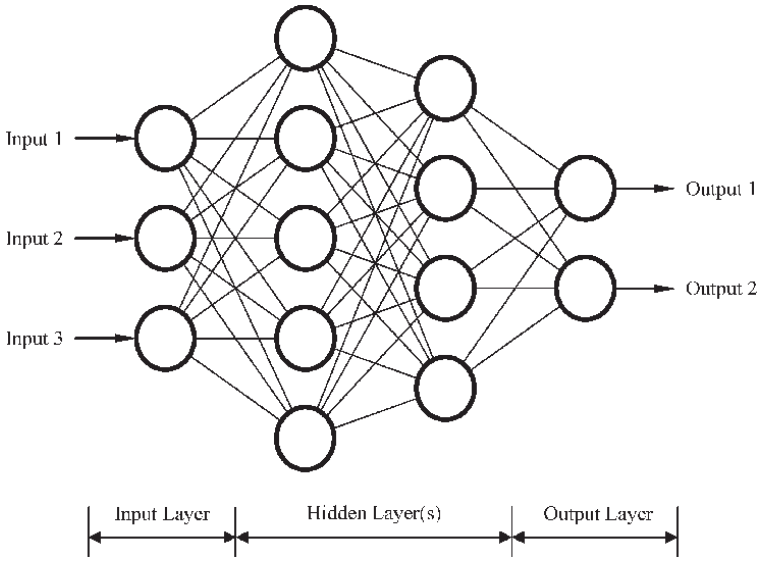

Figure 1: MLP Architecture

In our model, the data pass through two hidden layers, the **first hidden layer** has **64** neurons, in addition to a rectified linear unit (ReLU) activation function to allow the model to recognize more complex synergies for predictions. The **second hidden layer**, which used a ReLU activation, had **32** neurons. Lastly the output layer has 1 neuron, which produces predicted yield, and has also has a **ReLU activation** so the prediction cannot be a negative number, as negative crop yields do not make sense.

Hyperparameters and Optimization

I used **Adam** optimizer to train MLP model with a learning rate of **0.0001**, which ensures weight updates occur in a stable and measured way. The **loss** function was mean squared error (**MSE**), which is appropriate for regression and emphasizes larger errors. The model was trained with mini-batches of **32** to improve efficiency and allow the model to generalize better with added stochasticity. The model was trained for **10 epochs**, as test loss did not improve beyond that epoch. To avoid **overfitting**, I monitored training with respect to validation loss and kept weights from the epoch with the best validation loss; essentially applying a form of early stopping manually to return the most generalizable model.

Steps Taken to Prevent Overfitting

Overfitting is a major issue if the data is limited.

1. Chronological Train-Test Split:

Rather than randomly splitting the dataset, the dataset was split using time as the indicator: all of the records prior to 2021 were used for training, until the 2021 data was reserved for

validation. This mimics a real-world forecasting setting, ensuring the model was evaluated on data that was genuinely unseen.

2. Simplified Architecture:

The Multilayer Perceptron (MLP) model used an intentionally abbreviated number of layers and neurons, which works to reduces the model's ability to simply memorize the training data, and improve generalization to new inputs.

3. Manual Early Stopping:

Training was stopped at the time when the validation loss was the least, preventing overfitting of the training model by stopping early before it began to learn noise in the train data.

4. Feature Standardization:

All of the numerical input features were standardized, using z-score normalization. Standardizing makes all numbers similar in scale so that no single feature has more influence than others when the model learns.

5. Small batches for training: The data is broken down into batches. This provides some randomness to the model's training, which was helpful as it improved the model's ability to escape local minima and offered better generalization.

The well-tuned and simple MLP model was designed to generalize well, helping it performs accurately on new data and avoid overfitting—important for real-world tasks like crop yield prediction.

Section 3: Features & Labels

Feature Selection:

Feature selection is the process of selecting the appropriate input variables for a machine learning model. Selecting only the relevant input variables provides several benefits to performance by:

- Increased accuracy: By eliminating irrelevant or misleading data.
- Reduced overfitting: By reducing the dimensionality of the model.
- Increasing training speed: By reducing computation time.
- Increasing interpretive ability: By making the model easier to understand.
- Reducing dimensionality: By increased efficiencies in large datasets.

In order to create a robust model for predicting crop yield, it was important to carefully outline what data would be served as the inputs (features) and as the outputs (label). Since the model functioned to predict crop yield for each country, I organized the data at the country-year level to facilitate the association of environmental conditions from a particular year against the yield outcomes from that same year and the subsequent year (for future forecasting).

At the country-year level, the features and the labels were composited to make them comparable at the same granularity and degree of measurement as the target variable, crop yield. This was needed to remove errors associated with label leakage. For example, to apply one country level yield values to multiple latitude-longitude points would result in large errors as those yields are un-averaged. The use of annual averages for environmental data also helped create a smoothing effect for anomalies and short-term signals (e.g., unusual snowfall pattern) that have less impact on yield status over time. This helped the model to focus on the long-term predictive nature of the features that influence crop yield, as monthly yield would be driven by short-term factors, which is not the intended purpose for the ML process.

Finally, putting the numerical environmental variables alongside the categorical contextual variables (e.g., related to the country and crop type) gave the model all contextual information to replicate the potential yield conditions. Thus, these features and labels composited together formed the hedonic pricing model for a meaningful model for forecasting with regard to crop yields, which was the overall purpose of this project.

Categorical Features

In addition to environmental and soil conditions, the model includes two categorical features.

- Country: This feature indicates which country the data record comes from. One-hot encoding was used for the country feature, such that each country in the country variable has its own binary vector. This allows the model to learn the patterns specific to each country and how they influence crop yields.
- Item (Crop Type): This feature refers to the different types of crops that one might find in the dataset, such as maize or wheat. The crop type feature acknowledges that different crops may react differently to different environmental conditions. One-hot encoding allows the model to change its predictions based on the crop type in the record.

One-hot encoding is beneficial for using the categorical features, as it will ensure that the model treats each category distinctly, and non-ordinally, while also maintaining the model's assumption of a linear relationship between variables, as the one-hot variables do not imply any distance between the categories. By using this technique, we were able to use the categorical data in the Multilayer perceptron model (MLP) version and maximize the quality of the predictions by allowing it to account for nuanced differences across categories.

Features (Input Data)

The original data consisted of varying sources which were collected in 16 different CSV files which also contained varying environmental, land and soil condition information. The files included time series data such as snow cover, soil temperature, soil moisture, vegetation activity, and precipitation. The majority of the files reported values on a monthly basis — for example, ESoil_tavg_month_1, ESoil _tavg_month_2.... up to ESoil_tavg_month_12. Instead of using all 12 monthly columns as separate features (due to complexity), I compute the annual average for each of the monthly series. Every monthly series produces one representative feature per variable per year (for example, ESoil_tavg_annual_avg). This

averaging procedure extracts the entire year's environmental condition, simplifies model input, and maintains significant information.

After calculating the yearly averages, the next step was to match the latitude-longitude-level data to the country-level yield data. This required aggregating the spatial data all the way to the country-year level, as yield data is not publicly available at finer spatial resolution. To accomplish this, I referred file country_latitude_longitude_area_lookup.csv, which included the centroid longitude and latitude by country. For every environmental data point (with its own longitude and latitude), I calculated the Euclidean distance to each country centroid and assigned the data point to the closest country. This allowed me to group and average the environmental features by country-year. This was an important step to ensure that the input features were aligned with the spatial and temporal resolution of the label.

Here following input data features used:

1. longitude: Represents the longitudinal coordinate.
2. latitude: Represents the latitudinal coordinate.
3. year: Represents the year.
4. CanopInt_inst_month_1 to CanopInt_inst_month_12: Canopy Interception for each month of the year.
5. ESoil_tavg_month_1 to ESoil_tavg_month_12: Soil Evaporation for each month.
6. land_cover_max: The maximum percentage among all land cover classes, representing the dominant land cover type.
7. Rainf_tavg_annual_avg: Annual average rainfall.
8. Snowf_tavg_annual_avg: Annual average snowfall.
9. TWS_inst_annual_avg: Annual average Terrestrial Water Storage.
10. Tveg_tavg_annual_avg: Annual average Vegetation Water Content.
11. SoilMoi*annual_avg(all 4 files): Annual average soil moisture at different depths.
12. SoilTMP*_inst_annual_avg(all 4 files): Annual average soil temperature at different depths.

Labels(Output Data)

The output variable, or label, was the crop yield for a certain country in a certain year. In Yield_and_Production_data dataset, there are two measurements, "yield", and "production", both under the element column. Since I am only predicting crop yield, I filtered the dataset to yield data only, removing any chance of confusion with production data. After filtering, the actual yield values were in the value column, which I will retain as the target for prediction. I also retained the other columns country, year, and item (item specifies the crop type, such as wheat, maize, etc.) needed to join and indicate features. As I am using the target yield variable for model preparation, I will refer to the target variable yield, to avoid confusion with the other value columns in previous feature files.

To enable forecasting, the target variable (yield) was shifted so that the model learns to predict future outcomes based on past data. In this setup, conditions from year *t* are used to predict yield in year *t+1*, reflecting real-world scenarios where future performance is estimated using historical information.

Section 4: Preprocessing

The following preprocessing steps are applied to each dataset before model training:

1. Calculating Annual Averages: For environmental variables (e.g., soil temperature, soil moisture, snowfall) I calculated the yearly average from given data of 12 monthly columns. This helped in reducing dimensionality and prepared the data more stable for learning.
2. Land Cover Data: It has 13 columns for class percentage, I computed the highest percentage across all land cover types and created new feature (land_cover_max), after these 13 non-required columns were dropped.
3. Filtering Yield Data: The element column in yield data included "Yield" and "Production" entries. I cleaned the dataset by keeping only "Yield" records, removing irrelevant columns like domain, flag,
4. Handling Missing Values and Duplicates: Duplicate entries and null values were also removed to maintain a quality of data.
5. Merging to Country-Year: A centroid file for each country was used to calculate the Euclidean distance from each grid point to its nearest country centroid, then the environmental features were averaged by country and year.
6. Encoding Categorical Features: I applied One-Hot Encoding to some of the categorical features, such as country, item. This ensured the model would use these fields without assigning ordinal numbering.
7. Feature Scaling: All numerical features were standardized using StandardScaler to a mean of zero and standard deviation of one. This is important for ensuring stable training of the neural network.
8. Train-Test Split: Data collected during the years prior to 2021 would make up the training set, while 2021 would be the testing set. This more accurately reflects a prediction problem and avoids any potential data loss.

Data preprocessing is an important step which assures clean data, that which is usable with machine learning systems and meaningful in order to draw insights and accurately predict a parameter of interest from the data set.