Forecasting the export value of crop products
Using
Multilayer Perceptron Model

1. Performance Evaluation:

The performance of the model can be evaluated using various metrics, but one common metric for regression tasks like predicting export values is Mean Squared Error (MSE). MSE measures the average squared difference between the predicted values and the actual values.

Mathematically, MSE is calculated as:

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

Where:

- n is the total number of instances in the dataset.
- y_i is the actual export value for instance i.
- $\hat{y}_i$ is the predicted export value for instance i.

Mean Squared Error on Validation Set: 0.00546

R^2 (coefficient of determination) metric can also be used to evaluate the goodness of fit of the model. R^2 measures the proportion of the variance in the dependent variable (export values) that is predictable from the independent variables (features) in the model. It ranges from 0 to 1, with higher values indicating better model performance.
The reported performance of the model can be computed from the model outputs by comparing the predicted export values ($\hat{y}_i$) with the actual export values (y_i) using the chosen evaluation metric (e.g., MSE or R^2).

For the validation set, the R-squared value is: R^2=0.994547
These metrics indicate that the model performs exceptionally well on the validation set, explaining approximately 99.45% of the variance in the data.

The performance of my model on the test set is evaluated using two key metrics: Mean Squared Error (MSE) and R-squared (R^2).

1. **Mean Squared Error (MSE)**:
    - **Value**: 0.07640475699173555
    - **Interpretation**: This indicates that the average squared difference between the predicted export values and the actual export values is approximately 0.0764. A lower MSE indicates better predictive accuracy, but the acceptable range of MSE depends on the scale of the target variable.
2. **R-squared (R^2)**:
    - **Value**: 0.9237332838044297

- **Interpretation**: This value signifies that approximately 92.37% of the variance in the export values is explained by the model. An R^2 value close to 1 indicates a good fit, meaning the model's predictions are closely aligned with the actual values.

The total number of instances used in the model training and evaluation process is essential for understanding the scope and scale of the analysis. Here is how the instances were distributed across the training and test sets:

1. **Total Number of Instances:**
    - Total instances in the dataset: 167,649
2. **Training and Test Set Split:**
    - **Training Set:** Typically, a certain percentage of the data is allocated to the training set to train the model. For example, in this scenario, a common split might be 80% of the data allocated to the training set.
    - **Test Set:** The remaining percentage of the data is allocated to the test set to evaluate the trained model's performance. In this scenario, it would be 20% of the data.
3. **Number of Instances in Training Set:**
    - Number of instances in the training set: $0.8 \times 167,649 = 134,119$$0.8 \times 167,649 = 134,119$
4. **Number of Instances in Test Set:**
    - Number of instances in the test set: $0.2 \times 167,649 = 33,530$$0.2 \times 167,649 = 33,530$

These sets were derived by randomly splitting the original dataset into training and test sets while ensuring that each set is representative of the overall dataset's distribution. This ensures that the model is trained on a diverse range of examples and evaluated on unseen data to assess its generalization performance.

2. MLP model:

Model Description:

The multilayer perceptron (MLP) model described in the provided output consists of three dense layers with 100, 50, and 1 units, respectively. Each dense layer is followed by a dropout layer, which helps prevent overfitting by randomly dropping a fraction of the units during training.

- **Activation Function**: The activation function for the output layer is not explicitly specified in the provided output. However, for regression tasks like predicting export values, a commonly used activation function is the linear activation function ($f(x)=xf(x)=x$), which preserves the continuous nature of the output.

- **Loss Function**: The loss function used to train the model is typically Mean Squared Error (MSE) for regression tasks, where the goal is to minimize the squared difference between the predicted and actual values.

Here's a breakdown of the first ten predictions along with their actual values:

Year	Value	Predicted Export Value (USD)	Actual Export Value (USD)
108916	1.471928	0.122713	0.192980
9838	0.908217	0.005008	-5.064062e-17
145571	0.767289	0.000000	-0.276745
74792	-1.346629	0.000000	-0.284379
59297	0.908217	0.000000	-0.235571
4979	-0.641990	0.000000	-0.345557
199463	1.331000	0.000000	-0.344540
17905	0.344505	0.000000	-0.303056
69998	-0.078278	0.000000	-0.345578
167898	-1.346629	0.000000	-0.317130

Model: "sequential_5"

Layer (type)	Output Shape	Param #
dense_15 (Dense)	(None, 100)	27700
dropout_10 (Dropout)	(None, 100)	0
dense_16 (Dense)	(None, 50)	5050
dropout_11 (Dropout)	(None, 50)	0
dense_17 (Dense)	(None, 1)	51

Total params: 32801 (128.13 KB)
Trainable params: 32801 (128.13 KB)
Non-trainable params: 0 (0.00 Byte)

Preventing Overfitting:

To prevent overfitting, several steps can be taken:

1. **Dropout Regularization**: Dropout layers randomly drop a fraction of the units during training, forcing the network to learn more robust features. Mathematically, dropout can be represented as:
   $$\text{Output} = \text{input} \times \text{mask output} = \text{input} \times \text{mask}$$
   where the mask is a binary vector with randomly selected values.

2. **Early Stopping**: Early stopping is a technique that monitors the model's performance on a validation set during training and stops training when the performance stops improving. This helps prevent the model from overfitting to the training data.
3. **Reduce Learning Rate on Plateau**: This technique involves reducing the learning rate when the model's performance on the validation set plateaus. Lowering the learning rate can help the model converge to a better solution and prevent overfitting.

3. Features & Labels:

For the model development, a comprehensive set of features was meticulously selected to capture various dimensions influencing crop export values. There are a total number of 19 features used. These features were chosen based on their theoretical relevance and practical availability in the dataset. Below is a detailed description of the features used in the model:

- In deriving labels for the model, the export value (USD) of crop products served as the primary **target variable**. This export value was directly provided within the dataset, eliminating the need for any mathematical derivation. Each entry in the

dataset included a numerical value representing the export value of crop products for a specific geographical region. Therefore, the label was readily available and required no further calculation or manipulation.

- Year (Temporal Dimension):

  Total Number Used: 1 feature
  Derivation: Directly extracted from the 'Year' column in the dataset, indicating the specific year of observation. This temporal feature allows for the analysis of trends and patterns in crop export values over time.

- Area (Geographical Dimension):

  Total Number Used: 1 feature
  Derivation: Extracted from the 'Area' column, identifying the geographical regions or countries under consideration. This feature provides essential contextual information about the location of crop production and trade activities.

- Months:

  Total Number Used: 1 feature
  Derivation: The month in which the data was recorded, extracted from the 'Months' column in the dataset.

- Value (Numerical Indicator):

  Total Number Used: 1 feature
  Derivation: Directly obtained from the dataset, representing various numerical metrics associated with crop production, trade volumes, or economic performance. This feature serves as a fundamental indicator of crop export values.

- Average Annual Inflation:

  Total Number Used: 1 feature
  Derivation: Calculated based on aggregated inflation rates over time, reflecting the average annual inflation rate in a given geographical region. This feature captures the economic environment's stability and its potential impact on crop export values.

- Inflation Volatility:

  Total Number Used: 1 feature
  Derivation: Derived from the dataset by assessing the variability or fluctuations in inflation rates over time. This feature provides insights into the economic conditions affecting crop trade and export dynamics.

- Average Temperature Change:

Total Number Used: 1 feature
Derivation: Utilizes data from land temperature datasets to assess climate-related impacts on crop production and export. This feature accounts for climate variability and its effects on agricultural productivity, influencing crop export values.

- Average Production Value:

  Total Number Used: 1 feature
  Derivation: Represents the average production value of crop products for each geographical region. Derived from the crop production dataset by aggregating production values over time.

- Total Production Value:

  Total Number Used: 1 feature
  Derivation: Represents the total production value of crop products for each geographical region. Calculated by summing the production values over time.

- Total Food Security Index:

  Total Number Used: 1 feature
  Derivation: Represents the total food security index for each geographical region. Calculated by summing the food security index values over time.

- Average Emissions:

  Total Number Used: 1 feature
  Derivation: Represents the average emissions level for each geographical region. Derived from the emissions dataset by aggregating emissions values over time.

- Total Emissions:

  Total Number Used: 1 feature
  Derivation: Represents the total emissions level for each geographical region. Calculated by summing the emissions values over time.

- Total Employment:

  Total Number Used: 1 feature
  Derivation: Represents the total employment rate for each geographical region. Derived from the employment dataset by aggregating employment values over time.

- Total Exchange Rate:

  Total Number Used: 1 feature

Derivation: Represents the total exchange rate for each geographical region. Calculated by summing the exchange rate values over time.

- Total Fertilizers Use:

  Total Number Used: 1 feature
  Derivation: Represents the total amount of fertilizers used for each geographical region. Calculated by summing the fertilizers use values over time.

- Total Food Balances:

  Total Number Used: 1 feature
  Derivation: Represents the total food balances for each geographical region. Calculated by summing the food balances values over time.

- Total FDI:

  Total Number Used: 1 feature
  Derivation: Represents the total foreign direct investment for each geographical region. Calculated by summing the FDI values over time.

- Total Land Use:

  Total Number Used: 1 feature
  Derivation: Represents the total land use for each geographical region. Calculated by summing the land use values over time.

- Total Pesticides Use:

  Total Number Used: 1 feature
  Derivation: Represents the total amount of pesticides used for each geographical region. Calculated by summing the pesticides use values over time.

The selection of these features was driven by their significance in capturing economic, environmental, and agricultural dimensions that influence crop export dynamics. By incorporating a diverse set of indicators, the model aims to provide a comprehensive understanding of the factors shaping crop export trends, enabling more accurate forecasting and informed decision-making in agricultural trade scenarios.

4. Preprocessing:

- **Handling Missing Values**:

  Missing values in datasets are common and can adversely affect the performance of machine learning models.

For numerical features, one common approach is to replace missing values with the mean of the available data. This helps maintain the overall distribution of the feature. For categorical features, missing values are often replaced with the most frequent value (mode) since it provides a reasonable estimate of the missing information without introducing bias.
For numerical columns: Let X represent the numerical feature column. The mean μ of X is calculated as:

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i$$

The missing values $x_{missing}$ are replaced with μ.

For categorical columns: Let X represent the categorical feature column. The most frequent value mode(X) is calculated. The missing values are replaced with mode(X).

- **Feature Scaling:**

Feature scaling is essential to ensure that numerical features are on a similar scale, preventing certain features from dominating others during model training.
'StandardScaler' is a widely used method that scales features to have a mean of 0 and a standard deviation of 1. This transformation ensures that features have comparable ranges.
The formula for standardization involves subtracting the mean of the feature and then dividing by its standard deviation. This centers the data around 0 and scales it to have a unit variance.

Let X represent the numerical feature column.
The mean μ and standard deviation σ of X are calculated as:

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i$$

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)^2}$$

The scaled feature X_{scaled} is obtained as:

$$X_{scaled} = \frac{X - \mu}{\sigma}$$

- **Encoding Categorical Variables**:

Many machine learning algorithms require numerical inputs, making it necessary to encode categorical variables into a numerical format.
One-hot encoding (or dummy encoding) is a common technique used to convert categorical variables into a binary format. Each category becomes a separate binary feature, indicating its presence or absence in the observation.
This process prevents the model from assuming ordinal relationships between categories and ensures that each category receives equal treatment. This step is crucial as most machine learning algorithms require numerical input data.

Let X represent the categorical feature column with k unique categories.
Dummy variables are created for each category i:

1 if the observation belongs to the category i
0 otherwise

- **Train-Test Split:**

  Finally, we split the preprocessed data into training and testing sets using `train_test_split` from scikit-learn.
  Before training a machine learning model, it is essential to evaluate its performance on unseen data to assess its generalization capabilities.
  The dataset is split into training and testing sets, with the majority of data used for training and a smaller portion reserved for testing.
  Common ratios for the train-test split include 80-20 or 70-30, where the training set constitutes the majority of the data, and the testing set serves as an independent evaluation dataset.

  Let N represent the total number of samples. The data is split into training and testing sets with a specified ratio (e.g., 80% training and 20% testing):

  Training Set Size = $\frac{80}{100}$ X N

  Testing Set Size = $\frac{20}{100}$ X N

Overall, these preprocessing steps ensure that the data is cleaned, standardized, and properly formatted for training the machine learning model. They help to mitigate issues such as missing values, feature scales, and categorical variables, which can affect the model's performance if not handled appropriately.