Title: Comprehensive Analysis and Implementation of MLP Model for Agricultural Export Value Prediction

Introduction

This is an extensive analysis of a Python-based machine learning project that aims to predict the 'Export Value' of agricultural products for a specific area three years into the future. Libraries such as pandas, numpy, sklearn, seaborn, and matplotlib are utilized. The process starts with the acquisition of data, which is loaded into pandas DataFrames. This is followed by data preprocessing, including handling missing values and duplicates, standardizing the data, and dividing the dataset into training and testing sets. An Exploratory Data Analysis (EDA) is performed, utilizing seaborn and matplotlib for data visualization. Key features for predicting the 'Export Value' are identified using Lasso regression, and a Multi-Layer Perceptron (MLP) regressor is employed to model the relationship between these features and the 'Export Value'. The model's performance is evaluated using metrics like Mean Squared Error (MSE), Mean Absolute Error (MAE), and the R-squared score. The trained model is then used to make future predictions for the next three years, and the results are saved to CSV files for further analysis.

Performance

The model's efficacy was evaluated using three primary metrics: Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared score.Mean Squared Error (MSE) is the average of the squared differences between the actual and predicted values. It is a popular metric for regression models and is computed with the formula:

Here, 'n' represents the total number of instances, 'actual' is the actual value, and 'prediction' is the value predicted by the model. The squaring is done to penalize large errors.

Mean Absolute Error (MAE), on the other hand, is the average of the absolute differences between the actual and predicted values. It is less sensitive to outliers and large errors compared to MSE, and is calculated as follows:

Here, '|actual - prediction|' denotes the absolute difference between the actual and predicted values.

R-squared score, also known as the coefficient of determination, is a statistical measure that shows the proportion of the dependent variable's variance that is predictable from the independent variables. It is computed with the formula:

In this formula, SSR is the sum of squares of the residual errors, and SST is the total sum of squares. If the R² of a model is 0.50, then approximately half of the observed variation can be explained by the model's inputs.

The calculated performance metrics for the model include Mean Squared Error, Normalized Mean Squared Error, Mean Absolute Error, Normalized Mean Absolute Error, and the R-squared score.

## Results of Performance:

The results of the model's performance are presented in the table below:

| Metric | Value |
| --- | --- |
| Mean Squared Error (MSE) | 125718908324.67043 |
| Normalized Mean Squared Error | 0.13428035987886638 |
| Mean Absolute Error (MAE) | 166230.1403851357 |
| Normalized Mean Absolute Error | 0.023873779714042406 |
| R^2 Score | 0.8657196401211337 |

These metrics provide a comprehensive view of the model's performance. The relatively low values for Mean Squared Error and Mean Absolute Error indicate that the model's predictions are close to the actual values. The R^2 score of approximately 0.87 suggests that the model explains about 87% of the variance in the target variable, which is a strong indication of the model's efficacy.

Data Splitting process:

The total number of instances used in the model is the length of the merged DataFrame, which is obtained by merging multiple datasets on common columns. The data is then split into training and testing sets.

In the process of data splitting, the total number of instances used from the data was divided such that 20% of the data was allocated for the testing set, while the remaining 80% was used for the training set. This splitting was done using a standard method to ensure that the same sequence of random numbers is generated each time the code is run. Consequently, the number of instances in the training set is the length of the training features or training target. The number of instances in the test set is the length of the testing features or testing target.

Multilayer Perceptron Model

The Multilayer Perceptron (MLP) model utilized in this project is a variant of artificial neural networks. It comprises several layers of nodes, configured in a directed graph, where each layer is fully connected to the succeeding one.

The activation function used in the output layer is the Rectified Linear Unit, also referred to as ReLU. The ReLU function operates in such a way that if the input is greater than zero, it returns the input; if not, it returns zero. This function is commonly utilized in deep learning models as it efficiently introduces non-linearity without necessitating complex computations.

The loss function employed for training the model is the Mean Squared Error or MSE. This is a commonly used loss function for regression problems. The MSE is computed using the formula:

MSE = (1/n) \* Σ(actual - prediction)²

In this formula, 'n' denotes the total number of instances, 'actual' refers to the actual value, and 'prediction' is the value predicted by the model. The squaring is done to penalize large errors. Therefore, the MSE calculates the average of the squared differences between the actual and predicted values, making it sensitive to outliers.

The Multilayer Perceptron (MLP) model incorporates a single unit in the output layer, which presents the predicted value, serving as an indicator of the model's performance. The model consists of multiple layers of nodes in a directed graph, each layer fully connected to the next one.

The MLP model structure includes three hidden layers with 64, 32, and 16 units respectively, allowing the model to learn intricate patterns in the data without succumbing to overfitting. The activation function used in these layers is the Rectified Linear Unit (ReLU) function, which outputs the input directly if positive, otherwise, it outputs zero.For regression tasks, the activation function in the output layer is the identity function, outputting the input as is. The Mean Squared Error (MSE) loss function is used to train the MLP model, with 'adam' as the chosen solver for weight optimization. Adam, an algorithm for first-order gradient-based optimization of stochastic objective functions, is efficient and has little memory requirement. It works well in practice and compares favorably to other adaptive learning-method algorithms as it converges very fast and the learning speed of the Model is quite high. The maximum number of iterations for the solver to converge is set to 10000, and the pseudo random number generator's seed is set to 42.

To prevent overfitting, a common issue in machine learning models, multiple strategies were employed. The data was split into two distinct sets - the training and testing sets. This allows for an accurate evaluation of the model's performance on unencountered data. All features were standardized to a mean of zero and a standard deviation of one to ensure equal contribution and prevent any single feature from becoming overly influential.

L2 regularization was introduced to add a penalty term to the loss function, discouraging overly complex models by effectively shrinking the parameters. Early stopping was also implemented, halting the training process when the validation score ceases to improve, thereby avoiding learning noise inherent in the training data. These measures ensure a balance between complexity and performance, allowing the model to learn without overly adapting to the training data.

Features and labels

(10 marks max) Describe (as a list) how the label(s) was/were derived from the given data (using maths formulae when appropriate).

• (25 marks max) Report the features that you used for your model, making sure to specify the features used and the total number used, clarify why these features were selected, and describe (as a list) how they were derived from the given data, if applicable.

1. Derivation of Labels:

The target variable or label is 'Export Value,' representing the food export value from a specific area in a given year. The label for the MLP model was derived from the 'Export Value' column in the merged DataFrame. This column represents the export value of the agricultural products. The model is trained on other features, learning a mathematical function that maps the relationship between these features and the 'Export Value.' This function, represented as y = f(X) + e, where y is the target variable, f(X) represents the learned relationship between the predictors X and the target y, and e is the error term, signifying the difference between the actual and predicted target variable. The form of f(X) depends on the type of model used. In this case, the model involves multiple layers of linear transformations and non-linear activation functions, with the exact form of f(X) determined during the training process. The model learns the weights and biases that minimize the difference between the actual and predicted target variable.

The process of deriving features from data entails a series of steps, each of which is explained in more detail below:

1. Understanding the Data: This is the initial step, which involves deep diving into the data. One needs to understand the structure, statistical properties, and the presence of missing values in the data. This is usually done by using various functions that provide a snapshot of the data, its statistical summary, and the count of missing values in each column.
2. Correlation Analysis: In this step, a correlation matrix is calculated for the data. This matrix provides a measure of how strongly each pair of variables is related. To avoid redundancy (as the matrix is symmetric), only the upper triangle of this matrix is selected. If any feature pairs have a correlation greater than a set threshold (for instance, 0.7), they are listed for removal. This is because high correlation between features can lead to multicollinearity in the model, which can be problematic for certain types of models.
3. Feature Removal: This step involves removing the highly correlated features identified in the previous step. Removing these features can help prevent the issues associated with multicollinearity. After the correlated features are dropped, the modified data is stored separately for further use.
4. Feature Selection with Lasso Regression: Here, a Lasso regression model is fitted to the data. One column is treated as the dependent variable and the rest as independent variables. The Lasso model is a type of linear regression that uses a particular type of regularization (L1 regularization) which can shrink some coefficients to zero. This effectively simplifies the model by excluding those features. The degree of sparsity of the estimated coefficients, i.e., the number of coefficients that are zero, is controlled by a parameter.
5. Coefficient Analysis: In this step, the names of features corresponding to non-zero coefficients in the Lasso model are identified. These are the selected features, i.e., the ones that the model considers important. If a particular feature is present among the selected features, it is explicitly removed.
6. Compilation of Selected Features: This is the final step where all the selected features are compiled. They are first added to a list. This list is then converted to a set, which automatically removes any duplicates because sets only contain unique elements. Optionally, this set can be converted back to a list.

The final output of this process is a list of unique selected features. These features are considered important for the model as they have been chosen by the model itself and are not highly correlated with each other, thereby minimizing the risk of multicollinearity.

2. Features Used:

The model used 15 specific features: 'Standard Deviation', 'Area', 'Use per value of agricultural production', 'Value', 'Agricultural Use', 'Local currency units per USD', 'Year Code', 'Use per area of cropland', 'Value US$', 'Export Quantity', 'Yield', 'Emissions (CO2)', 'Crops total (Emissions CH4)', 'Food', and 'Import Value'. These were selected based on a combination of correlation analysis and Lasso regression. The correlation analysis removed highly correlated features to avoid multicollinearity, and Lasso regression excluded non-informative features.

The process of deriving these features from the given data involved several steps:

1. Each dataset was individually cleaned and preprocessed, involving the removal of unnecessary columns, duplicates, and converting necessary columns to categorical data types.
2. Each dataset was grouped by 'Area Code (M49)', 'Year Code', and 'Element', and the mean of 'Value' was calculated for each group.
3. The grouped data was pivoted to create a new DataFrame with 'Area Code (M49)' and 'Year Code' as the index, 'Element' as the columns, and 'Value' as the values.
4. Missing values in the pivoted data were filled with the mean of the respective column.
5. All pivoted DataFrames were merged into a single DataFrame based on 'Area Code (M49)' and 'Year Code'.
6. A correlation matrix was calculated, and features with a correlation coefficient greater than 0.7 were dropped.
7. Lasso regression was performed on the remaining features, excluding any with a coefficient of zero.
8. The remaining features were stored in the unique\_selected\_features list and used in the model.

The exact features and their total number depended on the specific data and the results of the correlation analysis and Lasso regression.

Pre-processing

The process of pre-processing the features involved a series of steps. The first step was data cleaning, which involved the removal of unnecessary columns that did not contribute to the analysis or model. This step effectively reduced the dimensionality of the datasets and aided in focusing on the relevant features. Additionally, duplicate rows were removed to prevent any bias that their presence could introduce. Dealing with missing values was another key aspect of pre-processing. The missing values in the datasets were treated by filling them with the mean of the corresponding column values. This simple and commonly used imputation method can yield satisfactory results for many features. However, it operates under the assumption that the data is missing completely at random, an assumption that may not always hold true.

Following this, data aggregation was performed by grouping the data by 'Area Code (M49)', 'Year Code', and 'Element', and calculating the mean of 'Value'. This was an important step to consolidate the data and make it suitable for analysis. Finally, the data reshaping stage involved using the pivot operation. This operation transformed the long-format data into a more workable wide format where each 'Element' became a separate column.

All these pre-processing steps were integral in ensuring the cleanliness and usability of the data for model building. They facilitated focus on the most relevant features, the handling of missing values, and the formatting of data in a way that made it was more readable and a essential preparation into explanatory data analysis (EDA).

During EDA, was when he missing values where first checked and cleaned out. Then follows a calculation of absolute correlation matrix for each data frame. The matrix shows the correlation coefficients between each pair of features in the dataframe. The upper triangle of the correlation matrix was then selected, and feature pairs with a correlation greater than 0.7 were listed. A high correlation between two features often indicates that they provide similar information, so one of them can be removed to reduce redundancy. Hence, features that were highly correlated were dropped to prevent multicollinearity, which can lead to unstable estimates and difficulty in determining the effect of individual features in models. This step was crucial for ensuring the robustness of the subsequent model building process.

The mathematical representation of the correlation calculation can be described by Pearson's correlation coefficient formula:

This formula gives a value between -1 and 1, where 1 represents total positive linear correlation, 0 represents no linear correlation, and -1 represents total negative linear correlation.

After dropping some features that had high correlation values, it was then proceeded to performing L1 regularization with Lasso regression analysis which is used to select features om multiple data frames.Lasso (Least Absolute Shrinkage and Selection Operator) regression is a type of linear regression that includes a penalty term. This penalty term encourages the model to keep only the most important features and set the coefficients of the other features to zero in order to extract features. ALos, during regularization the smoothing parameter was set to 0.1.

Lasso regression formula:

In addition, heatmap was used for the visual representation of the correlation matrix. It allows quick identification of which features are most correlated with each other. The darker the color, the higher the correlation. This is useful for understanding the relationships between different features.

Conclusion

In conclusion, this project aimed at employing an MLP model to predict the 'Export Value' of agricultural products for a specific area three years into the future. The project utilized a number of Python libraries to preprocess the data, perform exploratory data analysis, and implement the machine learning model. The performance metrics confirmed the model's efficiency in making accurate predictions. The model's ability to make these predictions will be instrumental in guiding agricultural planning and policies in the future. It will allow stakeholders to anticipate changes in export values and adjust their strategies accordingly. However, continual refinement and validation with new data will be critical to ensure the model's ongoing effectiveness. The model could also be expanded to include more features or be adapted to other similar prediction tasks in the future.